

Digital Whisper

גליון 63, אוגוסט 2015

מערכת המגזין:

אפיק קסטיאל, ניר אדר

מייסדים:

אפיק קסטיאל

מוביל הפרויקט:

אפיק קסטיאל

עורכים:

כתבים: 0x3d5157636b525761, יובל סיני, יהודה גרסטל, איאן מילר, עמית סרפר ואלכס פרייזר.

יש לראות בכל האמור במגזין Digital Whisper מידע כללי בלבד. כל פעולה שנעשית על פי המידע והפרטים האמורים במגזין Digital Whisper הינה על אחריות הקורא בלבד. בשום מקרה בעלי Digital Whisper /או הכותבים השונים אינם אחראים בשום צורה ואופן לתוצאות השימוש במידע המובא במגזין. עשיית שימוש במידע המובא במגזין הינה על אחריותו של הקורא בלבד.

פניות, תגובות, כתבות וכל הערה אחרת - נא לשלוח אל editor@digitalwhisper.co.il

דבר העורכים

ברוכים הבאים לגיליון ה-63 של Digital Whisper!

לפני מספר חודשים בעת ביקור בבית הורי, ישבתי עם אחותי הקטנה מול המחשב והראתי לה מספר משחקי מחשב שהייתי משחק בהם כשהייתי בערך בגילה, משחקים כדוגמת סדרת המשחקים [Commander Keen](#), או [Supaplex האגדי](#) כשאת רב הילדות שלי העברתי מול כל אחד מ-111 השלבים שלו. הראתי לה את אותם המשחקים גם מפני שאני חושב שלמרות כל המשחקים החדשים שיש היום אותם משחקים ישנים עדיין מעולים, וגם כנראה מהרצון לתת לה פרספקטיבה על אילו דברים היו נחשבים מגניבים ומעניינים פעם - עוד בתקופה שמערכת ההפעלה הנפוצה ביותר הייתה DOS וסייר הקבצים שהיה נחשב המילה האחרונה היה "Norton Comander"... חלק מהמשחקים הורדתי מכל מיני איזורים נידחים ברשת והרצתי על גבי DOSBox, ולחלק אף מצאתי גרסאות On-Line (אני לא מדבר על משהו בסיגנון של [jDosBox](#), אלא ממש מריצים את המשחק על השרת עצמו).

אחרי שישבנו והצגתי לה את המשחקים, נזכרתי שבאחד המשחקים שהיו לנו פעם בבית הספר, מצאנו איזה "כשל אבטחה" שאפשר לנו לצאת מהמשחק עצמו ולהגיע למצב של הרצת פקודות מערכת הפעלה (בבית-הספר היינו מריצים מעין פורטל בממשק DOS-י [שמזכיר קצת את הממשקים של עולם ה-BBS-ים], והיינו יכולים לבחור מתוך רשימה של משחקים או תוכנות, אבל לא באמת הייתה לנו גישה למערכת ההפעלה עצמה), ובעזרת ניצול של אותו "כשל אבטחה" יכולנו להגיע למערכת ההפעלה עצמה, ולהריץ תוכנות או משחקים אחרים, שהיו נמצאים על המחשב אך לא היו קיימים ברשימת התוכנות של אותו הפורטל.

סתם מתוך סקרנות, חיפשתי את אותו המשחק והפעלתי אותו On-Line, אחרי ההפעלה בדקתי האם אני זוכר כיצד לטרגר את אותו הכשל (מסתבר שדברים כאלה לא שוכחים 😊), גם אם נתקלת בהם בגיל 13...), ובאמת לאחר מכן - הגעתי למצב שאני נגיש למערכת ההפעלה עצמה יכולתי להריץ פקודות מערכת. כמובן ששם עצרתי.

למה אני מספר לכם את זה? גם בגלל שאני מת על נוסטלגיה, אבל בעיקר בשביל לתת דוגמא לנקודה שעליה רציתי לדבר בדברי הפתיחה: חוזק האבטחה של הארגון שלנו היא כחוזק האבטחה של החוליה החלשה ביותר בארגון שלנו. מי שינסה להכנס לארגון שלנו, בסבירות מאוד גבוהה לא ינסה לעשות זאת דרך אותם מקומות שעליהם שמנו את הדגש. אפשר לבנות חומות גבוהות לאין-שיעור, אבל כל עוד אנחנו פשוט לוקחים "קופסא שחורה" ומכניסים אותה לארגון שלנו מבלי באמת לבדוק מה יש בפנים - שלא נתפלא אחר-כך שכל אותן חומות לא עזרו לנו.



הדוגמא שנתתי היא דוגמא די מוזרה, אבל אפשר לתת עוד דוגמאות שהעקרון שלהן דומה, כגון: שימוש בספריות קוד ושילובן במוצר שלנו מבלי לבדוק אותן, שימוש בטכנולוגיות או פלאגינים למערכות שונות שנכתבו מחוץ לארגון ולא נבדקו, שימוש במוצרים שונים (שרת ניהול פרסומות? פלטפורמה לניהול קוד? שרת מיילים ארגוני? וכו') שאומנם מספקים את העבודה כמו שצריך, אבל אף אחד לא יכול להבטיח לנו מה הם מביאים לארגון שלנו חוץ מכל אותם פיצ'רים מעולים.

אני לא אומר שצריך להמציא את הגלגל מחדש, ובסבירות גבוהה, ברב המקרים ניסיון ליצור בעצמנו ספריה או מוצר קיים מאפס הוא לא בטוח הכיוון הנכון, אבל מכאן ועד הכנסה של מוצר "כמו שהוא" ולהתקין אותו על השרתים שלנו - יש מרחק.

התחלה טובה יכולה להיות בביצוע מחקר שוק ובדיקה אילו מוצרים קיימים היום יכולים לפתור את הבעיה שלנו ובנוסף, לבדוק אילו כשלי אבטחה פורסמו בכל אחד מהמוצרים עד כה (וכמובן שלא בטוח שהמוצר שבו נמצאו הכי פחות כשלים הוא המוצר הבטוח ביותר). שווה לבדוק תוך כמה זמן לקח לצוות הפיתוח של אותו מוצר לפרסם עדכון, והאם נמצאו כשלי אבטחה באותם איזורים שתוקנו בעבר וכו'. ויכול להיות שהכיוון הזה הוא רק בזבוז זמן, כל מקרה לגופו.

חשוב לזכור שזה שהמוצר לא פותח בשורותינו, לא אומר שהוא בטוח במאה אחוז, ויש מצב שדווקא עליו הייתי עושה יותר בדיקות. שיהיה חודש טוב ושקט (עד כמה שחודש אוגוסט יכול להיות שקט, כן?). ☺

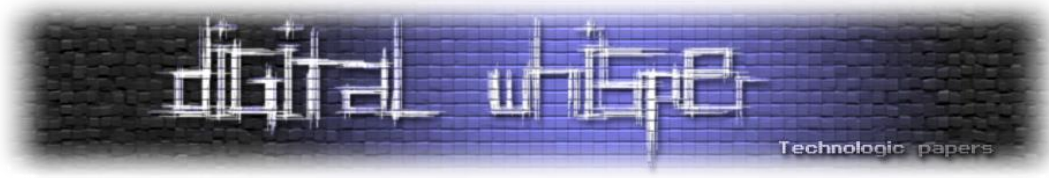
קריאה מהנה!

ניר אדר ואפיק קסטיאל.



תוכן עניינים

2	דבר העורכים
4	תוכן עניינים
5	הנדסה-לאחור: שרשרת העלייה של Windows 7 חלק שני - VBR
22	שיקולים בפיתוח והפעלת נשק קיברנטי
34	ניהול סממאות וזהויות ברשתות מיקרוסופט
57	זליגת שיטות התקיפה של קבוצת HackingTeam: שידרוג מידי לכל האקר מתחיל
69	דברי סיכום



הנדסה-לאחור: שרשרת העלייה של Windows 7 חלק שני - VBR

מאת 0x3d5157636b525761

רקע

בחלק הקודם דיברנו על ה-MBR, על הטעינה שלו על ידי ה-BIOS ועל כל הפעולות שהוא ביצע והכין לפני העברת האחריות ל-VBR. בחלק זה נתמקד ב-VBR. ה-VBR (קיצור של Volume Boot Record) הוא הסקטור הראשון של המחיצה הלוגית. כאמור, הוא נטען על ידי ה-MBR, שהוא הסקטור הראשון של הדיסק הפיזי שעלה.

במהלך המדריך אני אעבוד על Windows 7 SP1, x64, ולמעשה אמשיך מהמצב בו הפסקנו לפני כן.

הערת צד: כל ניסוי שהקורא מחליט לבצע כתוצאה מקריאת מאמר זה - על אחריותו בלבד!

חשיפת ה-VBR

בניגוד לקריאת הדיסק הפיזי (PhysicalDrive0), המצב הרבה יותר קל. בהנחה שהכונן הראשי הוא C:

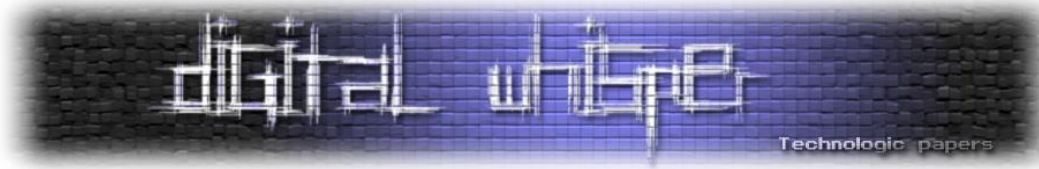
```
ActivePython 2.7.6.9 (ActiveState Software Inc.) based on
Python 2.7.6 (default, Feb 27 2014, 14:13:40) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> vbr = open('\\\\.\\C:', 'rb').read(512)
>>> open('C:\\vbr.bin', 'wb').write(vbr)
>>>
```

ניתוח ראשוני עם IDA

נפתח את הקובץ ב-IDA. כמקודם, IDA לא יודעת לנתח את הקובץ ישירות כי מדובר בקוד טהור ולא בפורמט מוגדר ולכן ננתח את הקובץ כקוד 16 ביט. כמובן, נבצע rebase לכתובת 7C00 (אליה נטען ה-VBR על ידי ה-MBR) ונלחץ על C בהתחלה כדי לנתח כקוד. כמקודם, ה-VBR מסתיים ב-0xAA55, כצפוי.

VBR - חלק שני Windows 7 הנדסה-לאחור: שרשרת העלייה של

www.DigitalWhisper.co.il



חלק א': ישר קופצים?

הדבר הראשון שנראה זה קפיצה מעל אזור די גדול אל - 0x7C54... מדוע? יש תשובה טובה לכך - אנחנו כבר לא בארץ ה-MBR הכיפי - מדובר כאן במחיצה, ולכן היא צריכה להכיל מערכת קבצים! כל מערכת קבצים מתחילה ב-header כלשהו שמתאר פרמטרים שונים בה, וכמובן - Windows עובד על NTFS. מכאן, כדאי להכנס למדריך כלשהו - אני נכנסתי אל <http://ntfs.com/ntfs-partition-boot-sector.htm> שמסביר יפה מאד איך ה-layout הבסיסי נראה. כמובן, לא ניכנס לכאן אל NTFS, אבל זה יספיק לענייננו.

הערות צד: המונח BPB שמופיע בהמון מקומות נקרא גם BIOS PARAMETER BLOCK, והוא מתאר את מערכת הקבצים. הוא שומש באופן מסורתי ב-FAT12 עבור floppies ולאחר מכן ב-FAT16, FAT32 ועכשיו גם ב-NTFS. יש לציין שישנן "גרסאות" ל-BPB, כאשר כל גרסה מרחיבה את הקודמת לה. זה של NTFS גדול במיוחד, בגודל 0x54 בתים!

לאחר שאנחנו מבינים מדוע קיימת קפיצה, אפשר לראות מה קורה לאחר מכן ב-0x7C54. אפשר לראות שהעניינים נראים די דומים (לפחות בהתחלה) למה שהיה עם ה-MBR:

```
seg000:7C54 ;
seg000:7C54
seg000:7C54  lblMain:                ; CODE XREF: seg000:lblStartlj
seg000:7C54          cli
seg000:7C54          ; Build stack
seg000:7C54          ;
seg000:7C54          ;
seg000:7C54          xor     ax, ax
seg000:7C54          mov     ss, ax
seg000:7C54          mov     sp, 7C00h
seg000:7C54          sti
seg000:7C54          ;
seg000:7C54          ; Make CS=DS=0x07C0
seg000:7C54          ;
seg000:7C54          push   7C0h
seg000:7C54          pop    ds
seg000:7C54          assume ds:nothing
seg000:7C54          push  ds
seg000:7C54          push  (offset lblStartParsingNTFS - offset lblStart)
seg000:7C54          retf
seg000:7C54          ;
seg000:7C66  lblStartParsingNTFS:    ; DATA XREF: seg000:7C62lo
seg000:7C66          mov     ds:0Eh, dl
seg000:7C66          ;
seg000:7C66          ; Make sure the OEM is "NTFS"
seg000:7C66          ;
seg000:7C66          cmp     dword ptr ds:3, 'SFTN'
seg000:7C66          jnz    short lblPrintErrorAndHangCaller
seg000:7C66          ;
seg000:7C66          ; Check for LBA mode reading
seg000:7C66          ;
seg000:7C66          mov     ah, 41h ; 'A'
seg000:7C66          mov     bx, 55AAh
seg000:7C66          int     13h                ; DISK - Check for INT 13h Extensions
seg000:7C66          ; BX = 55AAh, DL = drive number
seg000:7C66          ; Return: CF set if not supported
seg000:7C66          ; AH = extensions version
seg000:7C66          ; BX = AA55h
seg000:7C66          ; CX = Interface support bit map
seg000:7C66          jb     short lblPrintErrorAndHangCaller
seg000:7C66          cmp     bx, 0AA55h
seg000:7C66          jnz    short lblPrintErrorAndHangCaller
seg000:7C66          test   cx, 1
seg000:7C66          jnz    short loc_7C80
seg000:7C66          ;
seg000:7C8A  lblPrintErrorAndHangCaller: ; CODE XREF: seg000:7C73lj
seg000:7C8A          ; seg000:7C7Clj ...
seg000:7C8A          jmp     lblPrintErrorAndHang
```

VBR - חלק שני Windows 7 הנדסה-לאחור: שרשרת העלייה של

www.DigitalWhisper.co.il



- את הודעות השגיאה אני יודע להסיק מתוך ה-0x7D6A למשל: מכיוון ש-DS הוא 0x07C0 אז DS:1F8 הוא למעשה 0x7DF8. הבית שמופיע שם הוא 0x8C, וניתן לראות שבתחילת הפונקציה PrintMessage שמים בתוך AH את הערך 1, מה שאומר שכל AX יהיה 0x018C ולכן SI יהיה גם ערך זה. כמובן שאנחנו בסגמנט 0x07C0 ולכן ES:SI הוא 0x7D8C ושם כתוב ב-ANSI את המשפט A disk read error occurred. לכן, PrintMessage מצפה שאוגר AL יחזיק את ה-offset של ההודעה להדפסה, בעוד 0x7DF8 משמשת כטבלת offset-ים עבור הודעות השגיאה.

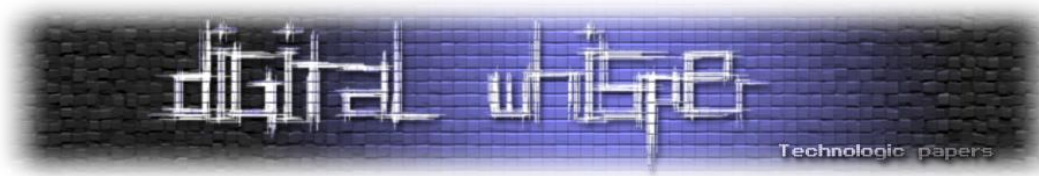
חלק ב': ההכנות לקראת הטעינה הבאה!

קדימה לחלק הבא:

```
seg000:7C8D
seg000:7C8D lblGetDriveParams:          ; CODE XREF: seg000:7C88j
seg000:7C8D         push     ds
seg000:7C8E         ;
seg000:7C8E         ; Make room for the buffer on the stack
seg000:7C8E         ; Size of buffer = 0x1A
seg000:7C8E         ;
seg000:7C8E         sub      sp, 18h
seg000:7C91         push     1Ah          ; 0x1A because it's WORD more than 0x18 + the way PUSH works
seg000:7C94         ;
seg000:7C94         ; Function 48 (get drive paramters)
seg000:7C94         ; DL = drive number
seg000:7C94         ; DS:SI = buffer to fill
seg000:7C94         ;
seg000:7C94         mov     ah, 48h      ; 'H'
seg000:7C96         mov     dl, ds:0Eh  ; DL = Drive number from [0x0E]
seg000:7C9A         mov     si, sp
seg000:7C9C         push   ss
seg000:7C9D         pop    ds
seg000:7C9E         assume ds:nothing
seg000:7C9E         int    13h         ; DISK - IBM/MS Extension - GET DRIVE PARAMETERS (DL - drive, DS:SI - buffer)
seg000:7CA0         ;
seg000:7CA0         ; Clear buffer from stack
seg000:7CA0         ;
seg000:7CA0         lahf   sp, 18h
seg000:7CA1         add    sp, 18h
seg000:7CA4         sahf
seg000:7CA5         pop    ax          ; Bytes per sector
seg000:7CA6         pop    ds          ; Restore DS
```

ניתוח:

- בקטע קוד זה קוראים ל-int13 (דיסק) עם פונקציה מספר 0x48. ניתן לקרוא תיעוד מלא ב-RBIL, אבל בכל מקרה מה שחשוב לדעת זה שמספר הכונן (drive number) נמצא ב-DL, הבאפר שיתמלא נמצא על DS:SI וגודלו בבתים נשמר ב-WORD הראשון שלו (זה ה-0x1A Push).
- נקודה עדינה:** מדוע עושים SUB SP על 0x18? התשובה היא ש-SI יקבל את ערכו של SP. תזכורת חשובה: SP מצביע על ראש המחסנית, על הכתובת האחרונה שבשימוש (inclusive). ה-PUSH לאחר ה-SUB דוחף את הגודל שעליו דיברנו, ולכן בסך הכל הורדנו את SP ב-0x1A בתים (ישנם גדלים נוספים הנתמכים, זו הדרך שבה הפסיקה מבצעת versioning). אגב, כל הסיפור עובד כי המחסנית גדלה לכיוון כתובות נמוכות -- אם לא, היינו צריכים לבצע push 0x1A לפני פעולת ה-SUB.



- נשים לב שמנקים רק 0x18 ואז מבצעים POP AX. במבנה שחוזר ניתן לראות שזה בדיוק ה- bytes per sector. הנה המבנה המלא (כאשר הוא בגודל 0x1A):

Offset (HEX)	Description	Size (bytes)
0x0000	Size of buffer (0x1A)	2
0x0002	Information flags	2
0x0004	Number of physical cylinders on drive	2
0x0008	Number of physical heads on drive	2
0x000C	Number of physical sectors on drive	2
0x0010	Number of total sectors on drive	8
0x0018	Bytes per sector	2

אז למעשה מה שביצענו הוא לקרוא ל-GetDriveParams על DL, וממנו לשלוף את Bytes per sector. כמובן שיש צורך לבצע בדיקות שהקריאה הצליחה וכו', וזה בדיוק מה שיתבצע בהמשך הקוד:

```

000:7CA7 ;
000:7CA7 ; Check for failures and mismatches:
000:7CA7 ; 1. INT13 failure
000:7CA7 ; 2. Bytes per sector failure
000:7CA7 ;
000:7CA7         jb     short 1b1PrintErrorAndHangCaller
000:7CA9         cmp     ax, ds:0Bh ; [0x0B] is exactly bytes per sector in the BPB
000:7CAD         jnz     short 1b1PrintErrorAndHangCaller

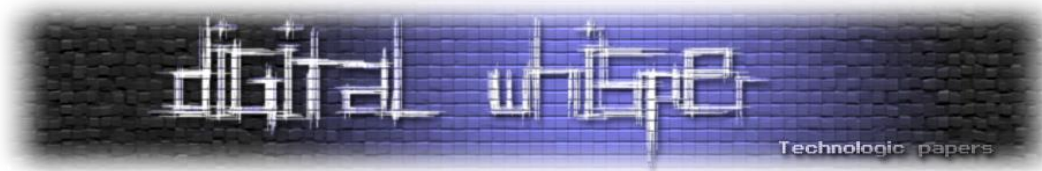
```

לאחר מכן, בהמשך ניתן לראות כמה הכנות ואז תחילת לולאה. אלו הן הכנות לטעינת BOOTMGR, ולמעשה לא מכילות המון לוגיקה. מייקרוסופט כתבו קוד יחסית גנרי, ולכן הוא מתייחס לנתונים מתוך ה-BPB. אף על פי כן, אנחנו נתייחס לנתונים האלה כקבועים:

```

seg000:7CAF ;
seg000:7CAF ; Copy bytes per sector to 0x7C0F and shift
seg000:7CAF ; Shifting by 4 is just like division by 16
seg000:7CAF ; This puts 0x20 in 0x7C0F
seg000:7CAF ;
seg000:7CAF         mov     ds:0Fh, ax
seg000:7CB2         shr     word ptr ds:0Fh, 4
seg000:7CB7 ;
seg000:7CB7 ; Preparations for loading BOOTMGR
seg000:7CB7 ;
seg000:7CB7         push   ds
seg000:7CB8         pop     dx ; DX = 0x07C0
seg000:7CB9         xor     bx, bx
seg000:7CBB         mov     cx, 2000h ; CX = 16 sectors
seg000:7CBE         sub     cx, ax ; CX - AX = 15 sectors
seg000:7CC0         inc     dword ptr ds:11h ; [0x7C11] = 1 (was zero)

```



יש כאן כמה דברים מעניינים שנזכרו לשלב מאוחר יותר:

- ה-WORD בכתובת 0x7C0F מקבלת 0x20, שזה גודל סקטור חלקי 16. ניתן לראות כאן שמייקרוסופט עשו קוד גנרי (AX מכיל בשלב זה את גודל הסקטור) ולא סתם הציבו 0x20 במקום המתאים.
 - אוגר DX מקבל את הערך 0x07C0.
 - אוגר BX מאופס.
 - אוגר CX מקבל גודל של 15 סקטורים, בהנחה שגודל סקטור הוא 512 (זה נכון לפי ה-BPB). דווקא כאן משעשע לראות שהקוד לא גנרי בכלל: אם גודל סקטור ישתנה בעתיד, הקוד הזה יידפק.
 - ה-DWORD בכתובת 0x7C11 (שנמצאת ב-BPB) "הופך" למשתנה גלובאלי, והוא מקבל את הערך 1.
- הקוד לאחר מכן ישר קורא לפונקציה שנמצאת בכתובת 0x7D1D, אז החלק הבא ינתח אותה.

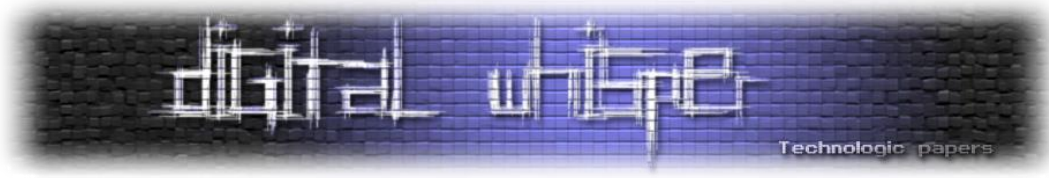
חלק ג': ExtendedRead, שוב...

הפעם נשתולל לגמרי ונעשה חלקים די גדולים (הוספתי הערות כמובן):

```
seg000:7D1D
seg000:7D1D
seg000:7D1D ExtendedDiskRead proc near ; CODE XREF: seg000:7CCF↑p
; All 32 bit registers are pushed
; Backup DS and ES
pushad
push ds
push es
seg000:7D21
seg000:7D21 lblReadAttempt: ; CODE XREF: ExtendedDiskRead+46↓j
; Loads the block counter
mov eax, ds:11h
; Adds the base block number
add eax, ds:1Ch
seg000:7D2A ;
seg000:7D2A ; Prepare buffer for extended read operation
seg000:7D2A ;
; Save DS
push ds
; Starting block number HI
push large 0
; Starting block number LO
push eax
; Transfer buffer HI
push es
; Transfer buffer LO
push bx
; Number of blocks to transfer
push 1
; Size = 0x10, reserved = 0
push 10h
seg000:7D38 ;
seg000:7D38 ; Prepare parameters for extended read
seg000:7D38 ;
; Function #52 - extended read
mov ah, 42h ; 'B'
; Drive number
mov dl, ds:0Eh
; DS = 0 (because SS = 0)
push ss
pop ds
; SI points to the buffer
mov si, sp
seg000:7D45 ;
seg000:7D45 ; Perform the interrupt
seg000:7D45 ;
; DISK - IBM/MS Extension - EXTENDED READ
int 13h
seg000:7D47 ;
seg000:7D47 ; Cleanups
seg000:7D47 ;
; 0x0110
pop ecx
; Old BX
pop bx
; Old ES
pop dx
; Old EAX
pop ecx
; 0
pop ecx
; Restore DS
pop ds
```

VBR - חלק שני 7 Windows הנדסה-לאחור: שרשרת העלייה של

www.DigitalWhisper.co.il



ניתוח:

- תחילת הפונקציה מגבה את כל ה- General purpose registers וכן את DS ו-ES. אפשר לראות שבסוף הפונקציה יש POP-ים מתאימים.
 - מכאן מתחילה לולאה שמשמשת בכמה משתנים גלובאליים.
 - בכתובת 0x7C11 נשמר מספר, שבאיטרציה הראשונה הוא 1. אפשר לראות שהוא גדל כל פעם באחד (בכל איטרציה).
 - בכתובת 0x7C1C נשמר ערך שלא משתנה בין איטרציות (0x800).
 - מכיוון ששני הערכים מחוברים לתוך EAX (ומכיוון ש-EAX הופך למספר הבלוק בדיסק שממנו נבצע את הקריאה), ניתן להסיק שהערך שגדל כל פעם באחד הוא block counter ואילו הערך השני הוא "כתובת בסיס".
 - מכאן מתבצעת הכנת ה-input buffer עבור הקריאה (פונקציה 0x42, פסיקה 0x13). עברנו במאמר על ה-MBR על פונקציה זו ועל המבנה של הבאפר בפירוט, ולכן לא אחזור עליו. כהרגלי, אפנה את הקורא המלומד אל RBIL ובו יש פירוט מלא.
 - ביצוע הפסיקה וניקוי הבאפר ממנה. משעשע לראות שהניקוי הוא לא ADD SP סתם אלא ממש פקודות POP, כאשר ECX משמש כאוגר "זבל" ל-DWORD-ים סוררים.
- אם לסכם, קראנו בלוק יחיד מהדיסק שנלקח מתוך EAX (והוא נלקח מתוך משתנה שמתחיל ב-0x32801 וגדל כל פעם באחד). הערכים הישנים של ES ו-BX נשמר ב-DX ו-BX, בהתאמה. להזכירם, אלו הם הרגיסטרים ששומשו לשמירת הבלוק שנקרא מתוך הדיסק. כעת, אנחנו מצפים לוידוא שהקריאה הצליחה ולהכנה לקראת האיטרציה הבאה, וכך אכן מתבצע:

```
seg000:7D50 ;
seg000:7D50 ; Validation
seg000:7D50 ;
seg000:7D50         jb         lblPrintErrorAndHang
seg000:7D54 ;
seg000:7D54 ; Post iteration operations
seg000:7D54 ;
seg000:7D54         inc         dword ptr ds:11h ; Increase block counter
seg000:7D59         add         dx, ds:0Fh ; Increase buffer pointer
seg000:7D5D         mov         es, dx
seg000:7D5F         dec         word ptr ds:16h ; Decrease iteration flag
seg000:7D63         jnz         short lblReadAttempt
seg000:7D65 ;
seg000:7D65 ; Restore registers and return
seg000:7D65 ;
seg000:7D65         pop         es
seg000:7D66         pop         ds
seg000:7D67         popad
seg000:7D69         retn
seg000:7D6A ; -----
seg000:7D6A lblPrintErrorAndHang: ; CODE XREF: seg000:lblPrintErrorAndHangC
seg000:7D6A ; ExtendedDiskRead+33]j
seg000:7D6A         mov         al, ds:1F8h
seg000:7D6D         call        PrintMessage
seg000:7D70         mov         al, ds:1FBh
seg000:7D73         call        PrintMessage
seg000:7D76 ;
seg000:7D76 lblHang: ; CODE XREF: seg000:7D77]j
seg000:7D76         hlt
seg000:7D76 ExtendedDiskRead endp
seg000:7D76 ;
seg000:7D77 ; -----
seg000:7D77         jmp         short lblHang
```

VBR - חלק שני Windows 7 הנדסה-לאחור: שרשרת העלייה של

www.DigitalWhisper.co.il



ניתוח:

- ביצוע JB כדי לוודא שהפסיקה הצליחה (אם היא נכשלה אז היא משנה את CF).
 - הגדלה של ערך ES בערך שנמצא תחת 0x7C0F. ערך זה יהיה 0x20, כפי שניתחנו לפני כן.
 - הורדה של הערך תחת 0x7C16. ערך זה יסמן לנו את מספר האיטרציות לביצוע - כאשר הוא מגיע לאפס, הפונקציה מסתיימת.
 - כמובן, בכל מקרה של שגיאה תודפסנה הודעות שגיאה ונגיע ללולאה אינסופית של HLT ו-JMP. ראינו דוגמא דומה בניתוח של ה-MBR.
- אם לסכם, פונקציה זו מבצעת קריאה סדרתית (block-by-block) של הדיסק. מספר הבלוקים נקבעים לפי המשתנים 0x7C11 ו-0x7C1C, מספר האיטרציות נקבע לפי 0x7C16, ובאפר היעד יהיה ES:BX.

חלק ד': יאללה ל-bootmgr!

עכשיו יהיה מעניין להסתכל על הקוד שקרא לפונקציה שכעת ניתחנו. הפלא-ופלא, זה בדיוק קוד ההמשך שלנו:

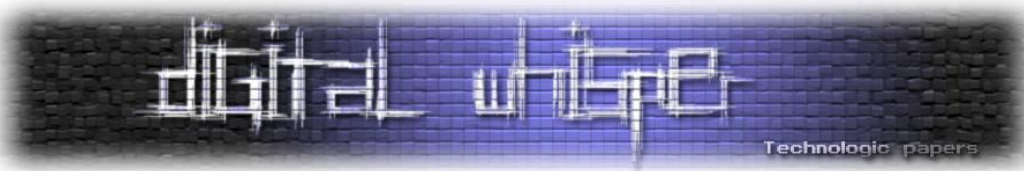
```
seg000:7CC5 ;
seg000:7CC5 ; Load BOOTMGR code
seg000:7CC5 ;
seg000:7CC5 ;
seg000:7CC5 lblLoadChunkFromDisk: ; CODE XREF: seg000:7CD4↓j
seg000:7CC5 add dx, ds:0Fh ; DX is increased by 0x20 (0x07E0, 0x0800, 0x0820, ..., 0x9A0)
seg000:7CC9 mov es, dx ; ES = DX
seg000:7CCB inc word ptr ds:16h ; 0x7C16 = 1 = one iteration
seg000:7CCF call ExtendedDiskRead
seg000:7CD2 sub cx, ax ; 0x2000 initially, each time decreased by 0x200
seg000:7CD4 ja short lblLoadChunkFromDisk
```

ניתוח:

- הערך של DX עולה כל פעם ב-0x20 (זה גודל סקטור חלקי 16 שחישבנו לפני כן). בהתחלה הוא היה 0x07C0, ולכן באיטרציה הראשונה הוא יקבל 0x07E0 וכן הלאה. לאחר מכן אנחנו מבינים ש-DX סתם היה אוגר ביניים, וכל מה שאמרתי עד כה היה עבור ES.
- **נקודה עדינה:** זה זמן טוב להיזכר ש-BX מאופס. מכיוון ש-ES:BX משמשים כבאפר שבו ייכתב המידע, בפועל המידע ייכתב לכתובות הפיזיות 0x7E00, 0x8000 וכן הלאה! לכן היה צורך לחלק את גודל הסקטור ב-16, וכאן ניתן לראות כבר שהקוד שמתמלא ייכנס היישר אל 0x7C00. נזכור גם ש-BX (ולמעשה, כל שאר האוגרים) לא מושפע מקריאה לפונקציה עקב השימוש ב-PUSHAD.
- ייחסנו חשיבות גדולה אל 0x7C16, והפונקציה ExtendedDiskRead מתייחסת אליו מאד ברצינות, אבל בפועל הוא תמיד יהיה 1... לכן קראתי לו בשם "Iteration flag" - משעשע לראות שמייקרוסופט כתבו פונקציית ExtendedRead גנרית מאד שתומכת במספר רב של איטרציות, אך בפועל קוראת לו כל פעם עם איטרציה אחת...

VBR - חלק שני 7 Windows הנדסה-לאחור: שרשרת העלייה של

www.DigitalWhisper.co.il



לאחר מכן בודקים תמיכה ב-Trusted platform:

```

seg000:7CD6 ;
seg000:7CD6 ; TCG, if possible
seg000:7CD6 ;
seg000:7CD9 mov ax, 0BB00h
seg000:7CD9 int 1Ah ; Trusted Computing Group call - TCG_StatusCheck
seg000:7CD9 ; Return: EAX = 0 if supported
seg000:7CD9 ; EBX = 41504354h ('TCPA')
seg000:7CD9 ; CH:CL = TCG BIOS Version
seg000:7CD9 ; EDX = BIOS TCG Feature Flags
seg000:7CD9 ; ESI = Pointer to Event Log
seg000:7CD9 ;
seg000:7CDB and eax, eax
seg000:7CDE jnz short lblRunBootMgr ; Best-effort
seg000:7CE0 cmp ebx, 'APCT'
seg000:7CE7 jnz short lblRunBootMgr ; Best-effort
seg000:7CE9 cmp cx, 102h
seg000:7CFD jnb short lblRunBootMgr ; Best-effort

```

ניתוח:

- קריאה ל-TCG_StatusCheck (על ידי AX=0xBB00), מחזירה ב-EAX את הערך 0 אם יש תמיכה. אם לא, ממשיכים הלאה.
- בדיקה ש-EBX מחזיק את ערך החזרה הנכון. אם לא, ממשיכים הלאה.
- בדיקה שהגרסה היא 1.02 ומעלה. אם לא, ממשיכים הלאה.
- הייתי רוצה שנזכור אם המשפט "אם לא, ממשיכים הלאה", שחזר אחרי עצמו 3 פעמים. אנחנו ננצל את המשפט הזה להערות הסיום.

אם הכל טוב עד כה, נגיע ל-0x7CEF. בבלוק זה תתבצע קריאה אל TCG_CompactHashLogExtendEvent ולאחר מכן יבוצע fallback אל lblRunBootMgr. הקריאה מתוארת באתר של Trusted Computing Group תחת:

https://www.trustedcomputinggroup.org/files/resource_files/CB0B2BFA-1A4B-B294-D0C3B9075B5AFF17/TCG_PCClientImplementation_1-21_1_00.pdf

הנה הקוד שקורא לפונקציה זו:

```

seg000:7CFE ;
seg000:7CFE ; TCG_CompactHashLogExtendEvent
seg000:7CFE ;
seg000:7CFE push ss
seg000:7CF8 push 0BB07h ; EAX = 0x0000BB07
seg000:7CF3 push ss
seg000:7CF4 push 0E70h ; ECX = 0x00000E70
seg000:7CF7 push ss
seg000:7CF8 push 9 ; EDX = 0x00000009
seg000:7CFB push ebx ; EBX
seg000:7CFD push ebx ; Dummy ESP
seg000:7CFE push ebp ; EBP
seg000:7D01 push ss
seg000:7D02 push ss ; ESI = 0x00000000 = informative value to be placed into the event field
seg000:7D03 push ss
seg000:7D04 push 188h ; EDI = 0x00000188 = offset of buffer to be hashed
seg000:7D07 popad
seg000:7D09 push cs
seg000:7D0A pop es ; ES = CS = segment of buffer to be hashed
seg000:7D0B int 1Ah ; Trusted Computing Group call - TCG_StatusCheck
seg000:7D0B ; Return: EAX = 0 if supported
seg000:7D0B ; EBX = 41504354h ('TCPA')
seg000:7D0B ; CH:CL = TCG BIOS Version
seg000:7D0B ; EDX = BIOS TCG Feature Flags
seg000:7D0B ; ESI = Pointer to Event Log
seg000:7D0B ;

```

VBR - חלק שני Windows 7 הנדסה-לאחור: שרשרת העלייה של

www.DigitalWhisper.co.il

ניתוח:

- הקוד דוחף מלא ערכים למחסנית ואז מבצע POPAD, כדי להשפיע על כל האוגרים. נשים לב שהאוגרים יוצאים בסדר הפוך.
- כל השדות מתוארים הן בקוד והן באתר של ה-TCG. ספציפית, השדות המאד מעניינים הם המצביע לבאפר שעליו עושים HASH (כתובת 07C0:01B8) וגודלו (0xE70).

לאחר מכן מגיעים אל סוף התכנית, lblRunBootMgr:

```

seg000:7D0D
seg000:7D0D  lblRunBootMgr:                ; CODE XREF: seg000:7CDE1j
seg000:7D0D                                ; seg000:7CE71j ...
seg000:7D0D          xor     ax, ax
seg000:7D0F ;
seg000:7D0F ; Zero filling
seg000:7D0F ;
seg000:7D0F          mov     di, 1028h
seg000:7D12          mov     cx, 0FD8h
seg000:7D15          cld
seg000:7D16          rep stosb
seg000:7D18 ;
seg000:7D18 ; Pass control to bootmgr
seg000:7D18 ;
seg000:7D18          jmp     near ptr 7E7Ah
seg000:7D1B ; -----
seg000:7D1B          nop
seg000:7D1C          nop

```

כאן מתבצע Zero filling, אף על פי שלא ברור מדוע יש צורך במילוי הזיכרון באפסים ולכאורה ניתן להסתדר בלעדיו. בכל מקרה, לאחר מכן, מעבירים את השליטה אל הכתובת 0x7E7A, שאליה נטען BOOTMGR.

כאן למעשה נגמר הקוד הראשי של ה-VBR. את שתי הפונקציות (קריאה מהדיסק והדפסה) ניתחנו.

נקודות מעניינות ותובנות

- יש לי חוב קטן: איך אני יודע שמדובר ב-bootmgr בכלל? ישנן כמה תשובות טובות, והכנתי אותן בסגנון פסח, כי זה הסתדר לי לא רע:
 1. **חכם** - מה הוא אומר? דיבגתי ואכן ראיתי ש-bootmgr עולה.
 2. **רשע** - מה הוא אומר? הרי קיימת בקוד הודעת השגיאה "BOOTMGR is missing".
 3. **תם** - מה הוא אומר? שרשרת העלייה של Windows מתעדת שהרכיב הבא בתור הוא bootmgr ואני מאמין לתיעוד.
 4. **ושאינו יודע לשאול?** כנראה שלא הגיע לחלק זה של המאמר בכל מקרה...

- אני רוצה שניזכר בנקודה המעניינת על ה-TCG - שימו לב שמתבצעות מספר קריאות, אבל הן best-effort, כלומר, אם אין תמיכה - לא נורא. חלק גדול מה-bootkits עשו hooking על int 0x13 ועל ידי כך החזירו בלוקים שקריים מהדיסק, עם נתונים שלהם. ניתן בהחלט להמציא bootkit שיבצע hooking על הקריאות אל ה-TCG, מכיוון שאף על פי שהן נקראות - נראה שלשרשרת העלייה לא כל כך אכפת אם הן מצליחות או לא. יהיה מעניין כפרוייקט לעשות hooking על int 0x1A.

סיכום

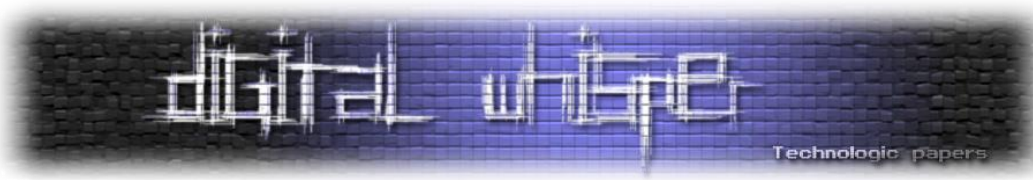
- המשכנו את תהליך העלייה של Windows, הפעם התמקדנו בחלק השני: ה-VBR.
- התעמקנו בנושאים הבאים:
 - ה-BPB
 - קריאות אל ה-TPM
 - טעינת bootmgr מהדיסק עם Extended Read
 - העברת שליטה אל bootmgr
- הוספתי נספח של הקוד השלם. הייתי מוסיף IDB, אבל גרסאות שונות של IDA לא תומכות בהכרח בכל IDB וגם נחמד שזה יגיע יחד עם המסמך. מדובר בקוד הסופי, כולל ההערות, כמובן.

על המחבר

0x3d5157636b525761, עושה Reversing ופיתוח Low Level למחיתו. ניתן ליצור איתי קשר ב:

0x3d5157636b525761@gmail.com





נספח א': הקוד המלא כולל הערות

```

seg000:7C00 ;
seg000:7C00 ; Input MD5 : C1C0E5D5E2701CBDA3FD4292CD32D6C2
seg000:7C00 ; Input CRC32 : 2A6D0660
seg000:7C00 ; -----
seg000:7C00 ; File Name : vbr.bin
seg000:7C00 ; Format : Binary file
seg000:7C00 ; Base Address: 0000h Range: 0000h - 0200h Loaded length: 0200h
seg000:7C00
seg000:7C00 .686p
seg000:7C00 .mmx
seg000:7C00 .model flat
seg000:7C00 ; =====
seg000:7C00 ; Segment type: Pure code
seg000:7C00 seg000 segment byte public 'CODE' use16
seg000:7C00 assume cs:seg000
seg000:7C00 ;org 7C00h
seg000:7C00 assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:nothing
seg000:7C00
seg000:7C00 lblStart: ; DATA XREF: seg000:7C62o
seg000:7C00 jmp short lblMain
seg000:7C02 ; -----
seg000:7C02 nop
seg000:7C02 ; -----
seg000:7C03 g_dwOemLo dd 'SFTN' ; DATA XREF: seg000:7C6Ar
seg000:7C03 ; OEM ID
seg000:7C07 dd ' '
seg000:7C0B dw 200h ; Bytes per sector
seg000:7C0D db 8 ; Sectors per cluster
seg000:7C0E g_wReservedSectors dw 0 ; DATA XREF: seg000:lblStartParsingNTFSw
seg000:7C0E ; seg000:7C96r
seg000:7C0E ; Reserved sectors
seg000:7C10 db 0, 0, 0 ; 0 ; Zero
seg000:7C13 dw 0 ; Unused
seg000:7C15 db 0F8h ; ° ; Media descriptor
seg000:7C16 dw 0 ; Zero
seg000:7C18 dw 3Fh ; Sectors per track
seg000:7C1A dw 0FFh ; Number of heads
seg000:7C1C dd 32800h ; Hidden sectors
seg000:7C20 dd 0 ; Unused
seg000:7C24 dd 800080h ; Unused
seg000:7C28 dq 1866D7FFh ; Total sectors
seg000:7C30 dq 0C0000h ; Logical cluster number for the file
$MFT
seg000:7C38 dq 2 ; Logical cluster number for the file
$MFTMirr
seg000:7C40 dd 0F6h ; Clusters per file record segment
seg000:7C44 db 1 ; Clusters per index buffer
seg000:7C45 db 0, 0, 0 ; 0 ; Unused
seg000:7C48 dq 88B4B852B4B8448Ah ; Volume serial number
seg000:7C50 dd 0 ; Checksum
seg000:7C54 ; -----
seg000:7C54
seg000:7C54 lblMain: ; CODE XREF: seg000:lblStartj
seg000:7C54 cli
seg000:7C55 ;
seg000:7C55 ; Build stack
seg000:7C55 ;

```

VBR - חלק שני Windows 7 הנדסה-לאחור: שרשרת העלייה של

www.DigitalWhisper.co.il

```

seg000:7C55
seg000:7C55          xor     ax, ax
seg000:7C57          mov     ss, ax
seg000:7C59          mov     sp, 7C00h
seg000:7C5C          sti
seg000:7C5D ;
seg000:7C5D ; Make CS=DS=0x07C0
seg000:7C5D ;
seg000:7C5D          push   7C0h
seg000:7C60          pop     ds
seg000:7C61          assume ds:nothing
seg000:7C61          push   ds
seg000:7C62          push   (offset lblStartParsingNTFS - offset lblStart)
seg000:7C65          retf
seg000:7C66
seg000:7C66 lblStartParsingNTFS:          ; DATA XREF: seg000:7C62o
seg000:7C66          mov     ds:0Eh, dl          ; Save drive number in [0x0E]
seg000:7C6A ;
seg000:7C6A ; Make sure the OEM is "NTFS"
seg000:7C6A ;
seg000:7C6A          cmp     dword ptr ds:3, 'SFTN'
seg000:7C73          jnz     short lblPrintErrorAndHangCaller
seg000:7C75 ;
seg000:7C75 ; Check for LBA mode reading
seg000:7C75 ;
seg000:7C75          mov     ah, 41h ; 'A'
seg000:7C77          mov     bx, 55AAh
seg000:7C7A          int     13h                ; DISK - Check for INT 13h Extensions
                                ; BX = 55AAh, DL = drive number
                                ; Return: CF set if not supported
                                ; AH = extensions version
                                ; BX = AA55h
                                ; CX = Interface support bit map
seg000:7C7A          jb     short lblPrintErrorAndHangCaller
seg000:7C7C          cmp     bx, 0AA55h
seg000:7C7E          jnz     short lblPrintErrorAndHangCaller
seg000:7C82          test    cx, 1
seg000:7C88          jnz     short lblGetDriveParams
seg000:7C8A
seg000:7C8A lblPrintErrorAndHangCaller:          ; CODE XREF: seg000:7C73j
seg000:7C8A          ; seg000:7C7Cj ...
seg000:7C8A          jmp     lblPrintErrorAndHang
seg000:7C8D ; -----
seg000:7C8D
seg000:7C8D lblGetDriveParams:          ; CODE XREF: seg000:7C88j
seg000:7C8D          push   ds
seg000:7C8E ;
seg000:7C8E ; Make room for the buffer on the stack
seg000:7C8E ; Size of buffer = 0x1A
seg000:7C8E ;
seg000:7C8E          sub     sp, 18h
seg000:7C91          push   1Ah                ; 0x1A because it's WORD more than 0x18 +
the way PUSH works
seg000:7C94 ;
seg000:7C94 ; Function 48 (get drive paramters)
seg000:7C94 ; DL = drive number
seg000:7C94 ; DS:SI = buffer to fill
seg000:7C94 ;
seg000:7C94          mov     ah, 48h ; 'H'
seg000:7C96          mov     dl, ds:0Eh        ; DL = Drive number from [0x0E]
seg000:7C9A          mov     si, sp
seg000:7C9C          push   ss
seg000:7C9D          pop     ds
    
```

VBR - חלק שני Windows 7 הנדסה-לאחור: שרשרת העלייה של

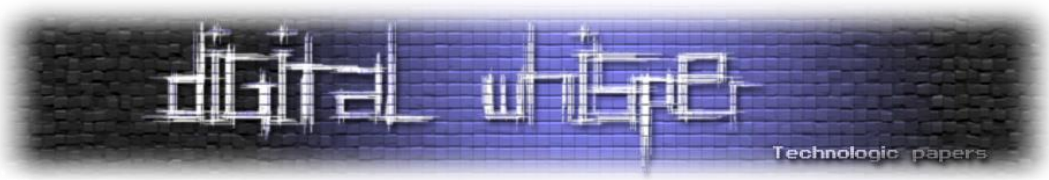
www.DigitalWhisper.co.il

```

seg000:7C9E          assume ds:nothing
seg000:7C9E          int     13h          ; DISK - IBM/MS Extension - GET DRIVE
PARAMETERS (DL - drive, DS:SI - buffer)
seg000:7CA0 ;
seg000:7CA0 ; Clear buffer from stack
seg000:7CA0 ;
seg000:7CA0          lahf
seg000:7CA1          add     sp, 18h
seg000:7CA4          sahf
seg000:7CA5          pop     ax          ; Bytes per sector
seg000:7CA6          pop     ds          ; Restore DS
seg000:7CA7 ;
seg000:7CA7 ; Check for failures and mismatches:
seg000:7CA7 ;     1. INT13 failure
seg000:7CA7 ;     2. Bytes per sector failure
seg000:7CA7 ;
seg000:7CA7          jb     short lblPrintErrorAndHangCaller
seg000:7CA9          cmp     ax, ds:0Bh ; [0x0B] is exactly bytes per sector in
the BPB
seg000:7CAD          jnz     short lblPrintErrorAndHangCaller
seg000:7CAF ;
seg000:7CAF ; Copy bytes per sector to 0x7C0F and shift
seg000:7CAF ; Shifting by 4 is just like division by 16
seg000:7CAF ; This puts 0x20 in 0x7C0F
seg000:7CAF ;
seg000:7CAF          mov     ds:0Fh, ax
seg000:7CB2          shr     word ptr ds:0Fh, 4
seg000:7CB7 ;
seg000:7CB7 ; Preparations for loading BOOTMGR
seg000:7CB7 ;
seg000:7CB7          push   ds
seg000:7CB8          pop     dx          ; DX = 0x07C0
seg000:7CB9          xor     bx, bx
seg000:7CBB          mov     cx, 2000h   ; CX = 16 sectors
seg000:7CBE          sub     cx, ax      ; CX - AX = 15 sectors
seg000:7CC0          inc     dword ptr ds:11h ; [0x7C11] = 1 (was zero)
seg000:7CC5 ;
seg000:7CC5 ; Load BOOTMGR code
seg000:7CC5 ;
seg000:7CC5          lblLoadChunkFromDisk:
seg000:7CC5          add     dx, ds:0Fh ; CODE XREF: seg000:7CD4j
0x0800, 0x0820, ..., 0x9A0) ; DX is increased by 0x20 (0x07E0,
seg000:7CC9          mov     es, dx      ; ES = DX
seg000:7CCB          inc     word ptr ds:16h ; 0x7C16 = 1 = one iteration
seg000:7CCF          call   ExtendedDiskRead
seg000:7CD2          sub     cx, ax      ; 0x2000 initially, each time decreased
by 0x200
seg000:7CD4          ja     short lblLoadChunkFromDisk
seg000:7CD6 ;
seg000:7CD6 ; TCG, if possible
seg000:7CD6 ;
seg000:7CD6          mov     ax, 0BB00h
seg000:7CD9          int     1Ah        ; Trusted Computing Group call -
TCG StatusCheck
seg000:7CD9          ; Return: EAX = 0 if supported
seg000:7CD9          ; EBX = 41504354h ('TCPA')
seg000:7CD9          ; CH:CL = TCG BIOS Version
seg000:7CD9          ; EDX = BIOS TCG Feature Flags
seg000:7CD9          ; ESI = Pointer to Event Log
seg000:7CD9          ;
seg000:7CDB          and     eax, eax
seg000:7CDE          jnz     short lblRunBootMgr ; Best-effort
    
```

VBR - חלק שני Windows 7 הנדסה-לאחור: שרשרת העלייה של

www.DigitalWhisper.co.il



```

seg000:7CE0      cmp     ebx, 'APCT'
seg000:7CE7      jnz     short lblRunBootMgr ; Best-effort
seg000:7CE9      cmp     cx, 102h
seg000:7CED      jb     short lblRunBootMgr ; Best-effort
seg000:7CEF      ;
seg000:7CEF      ; TCG_CompactHashLogExtendEvent
seg000:7CEF      ;
seg000:7CEF      push   ss
seg000:7CF0      push   0BE07h           ; EAX = 0x0000BB07
seg000:7CF3      push   ss
seg000:7CF4      push   0E70h           ; ECX = 0x00000E70 = length of buffer to
be hashed
seg000:7CF7      push   ss
seg000:7CF8      push   9               ; EDX = 0x00000009 = PCR index
seg000:7CFB      push   ebx             ; EBX
seg000:7CFD      push   ebx             ; Dummy ESP
seg000:7CFF      push   ebp             ; EBP
seg000:7D01      push   ss
seg000:7D02      push   ss             ; ESI = 0x00000000 = informative value to
be placed into the event field
seg000:7D03      push   ss
seg000:7D04      push   1B8h           ; EDI = 0x000001B8 = offset of buffer to
be hashed
seg000:7D07      popad
seg000:7D09      push   cs
seg000:7D0A      pop     es             ; ES = CS = segment of buffer to be
hashed
seg000:7D0B      int     1Ah           ; Trusted Computing Group call -
TCG_StatusCheck
seg000:7D0B      ; Return: EAX = 0 if supported
seg000:7D0B      ; EBX = 41504354h ('TCPA')
seg000:7D0B      ; CH:CL = TCG BIOS Version
seg000:7D0B      ; EDX = BIOS TCG Feature Flags
seg000:7D0B      ; ESI = Pointer to Event Log
seg000:7D0B      ;
seg000:7D0D      ; CODE XREF: seg000:7CDEj
seg000:7D0D      ; seg000:7CE7j ...
seg000:7D0D      xor     ax, ax
seg000:7D0F      ;
seg000:7D0F      ; Zero filling
seg000:7D0F      ;
seg000:7D0F      mov     di, 1028h
seg000:7D12      mov     cx, 0FD8h
seg000:7D15      cld
seg000:7D16      rep stosb
seg000:7D18      ;
seg000:7D18      ; Pass control to bootmgr
seg000:7D18      ;
seg000:7D18      jmp     near ptr 7E7Ah
seg000:7D1B      ; -----
seg000:7D1B      nop
seg000:7D1C      nop
seg000:7D1D      ; ===== S U B R O U T I N E =====
seg000:7D1D      ;
seg000:7D1D      ExtendedDiskRead proc near ; CODE XREF: seg000:7CCFp
seg000:7D1D      pushad ; All 32 bit registers are pushed
seg000:7D1F      push   ds ; Backup DS and ES
seg000:7D20      push   es
seg000:7D21      ;
seg000:7D21      lblReadAttempt: ; CODE XREF: ExtendedDiskRead+46j

```

Windows 7 שני - חלק שני - VBR הנדסה-לאחור: שרשרת העלייה של

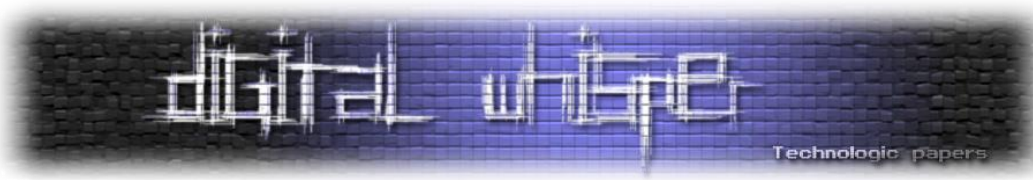
www.DigitalWhisper.co.il

```

seg000:7D21      mov     eax, ds:11h      ; Loads the block counter
seg000:7D25      add     eax, ds:1Ch      ; Adds the base block number
seg000:7D2A      ;
seg000:7D2A      ; Prepare buffer for extended read operation
seg000:7D2A      ;
seg000:7D2A      push   ds                ; Save DS
seg000:7D2B      push   large 0           ; Starting block number HI
seg000:7D31      push   eax               ; Starting block number LO
seg000:7D33      push   es                ; Transfer buffer HI
seg000:7D34      push   bx                ; Transfer buffer LO
seg000:7D35      push   1                 ; Number of blocks to transfer
seg000:7D38      push   10h               ; Size = 0x10, reserved = 0
seg000:7D3B      ;
seg000:7D3B      ; Prepare parameters for extended read
seg000:7D3B      ;
seg000:7D3B      mov     ah, 42h ; 'B'    ; Function #52 - extended read
seg000:7D3D      mov     dl, ds:0Eh      ; Drive number
seg000:7D41      push   ss                ;
seg000:7D42      pop     ds                ; DS = 0 (because SS = 0)
seg000:7D43      mov     si, sp           ; SI points to the buffer
seg000:7D45      ;
seg000:7D45      ; Perform the interrupt
seg000:7D45      ;
seg000:7D45      int     13h              ; DISK - IBM/MS Extension - EXTENDED READ
(DL - drive, DS:SI - disk address packet)
seg000:7D47      ;
seg000:7D47      ; Cleanups
seg000:7D47      ;
seg000:7D47      pop     ecx               ; 0x0110
seg000:7D49      pop     bx                ; Old BX
seg000:7D4A      pop     dx                ; Old ES
seg000:7D4B      pop     ecx               ; Old EAX
seg000:7D4D      pop     ecx               ; 0
seg000:7D4F      pop     ds                ; Restore DS
seg000:7D50      ;
seg000:7D50      ; Validation
seg000:7D50      ;
seg000:7D50      jb     lblPrintErrorAndHang
seg000:7D54      ;
seg000:7D54      ; Post iteration operations
seg000:7D54      ;
seg000:7D54      inc     dword ptr ds:11h ; Increase block counter
seg000:7D59      add     dx, ds:0Fh       ; Increase buffer pointer
seg000:7D5D      mov     es, dx
seg000:7D5F      dec     word ptr ds:16h  ; Decrease iteration flag
seg000:7D63      jnz    short lblReadAttempt
seg000:7D65      ;
seg000:7D65      ; Restore registers and return
seg000:7D65      ;
seg000:7D65      pop     es
seg000:7D66      pop     ds
seg000:7D67      popad
seg000:7D69      retn
seg000:7D6A      ; -----
seg000:7D6A      ;
seg000:7D6A      lblPrintErrorAndHang:    ; CODE XREF:
seg000:lblPrintErrorAndHangCallerj
seg000:7D6A      ; ExtendedDiskRead+33j
seg000:7D6A      mov     al, ds:1F8h
seg000:7D6D      call   PrintMessage
seg000:7D70      mov     al, ds:1FBh
seg000:7D73      call   PrintMessage
seg000:7D76
    
```

VBR - חלק שני Windows 7 הנדסה-לאחור: שרשרת העלייה של

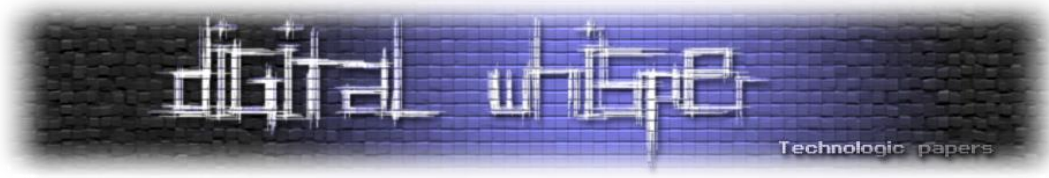
www.DigitalWhisper.co.il



```

seg000:7D76 lblHang: ; CODE XREF: seg000:7D77j
seg000:7D76 hlt
seg000:7D76 ExtendedDiskRead endp
seg000:7D76 ; -----
seg000:7D77 jmp short lblHang
seg000:7D79 ; ===== S U B R O U T I N E =====
seg000:7D79 PrintMessage proc near ; CODE XREF: ExtendedDiskRead+50p
seg000:7D79 ; ExtendedDiskRead+56p
seg000:7D79 mov ah, 1
seg000:7D7B mov si, ax
seg000:7D7D ;
seg000:7D7D ; Handle the next character
seg000:7D7D ;
seg000:7D7D lblNextChar: ; CODE XREF: PrintMessage+10j
seg000:7D7D lodsb
seg000:7D7E cmp al, 0
seg000:7D80 jz short lblFinishPrinting
seg000:7D82 ;
seg000:7D82 ; Perform TTY printing and handle the next character
seg000:7D82 ;
seg000:7D82 mov ah, 0Eh
seg000:7D84 mov bx, 7
seg000:7D87 int 10h ; - VIDEO - WRITE CHARACTER AND ADVANCE
CURSOR (TTY WRITE)
seg000:7D87 ; AL = character, BH = display page
(alpha modes)
seg000:7D87 ; BL = foreground color (graphics modes)
seg000:7D89 jmp short lblNextChar
seg000:7D8B ; -----
seg000:7D8B lblFinishPrinting: ; CODE XREF: PrintMessage+7j
seg000:7D8B retn
seg000:7D8B PrintMessage endp
seg000:7D8B ; -----
seg000:7D8C aDiskReadError db 0Dh,0Ah
seg000:7D8C db 'A disk read error occurred',0
seg000:7DA9 aBootmgrIsMissi db 0Dh,0Ah
seg000:7DA9 db 'BOOTMGR is missing',0
seg000:7DBE aBootmgrIsCompr db 0Dh,0Ah
seg000:7DBE db 'BOOTMGR is compressed',0
seg000:7DD6 aPressCtrlAltDe db 0Dh,0Ah
seg000:7DD6 db 'Press Ctrl+Alt+Del to restart',0Dh,0Ah,0
seg000:7DF8 db 8Ch
seg000:7DF9 db 0A9h ; -
seg000:7DFA db 0BEh ; -
seg000:7DFB db 0D6h ; -
seg000:7DFC db 0
seg000:7DFD db 0
seg000:7DFE db 55h ; U
seg000:7DFE db 0AAh ; -
seg000:7DFE seg000 ends
seg000:7DFE
seg000:7DFE
seg000:7DFE end

```



שיקולים בפיתוח והפעלת נשק קיברנטי

מאת יובל סיני

הקדמה

עידן המידע, יצר עבור רבים כר של הזדמנויות חדשות, דבר אשר כלל בין השאר את לידתה של כלכלת המידע, האצת הגלובליזציה והחדשנות. לצד היתרונות הגלומים בעידן המידע, נוצרה תלות מלאה של מרבית הציבור ומדינות העולם בטכנולוגיה ובזמינות תשתיות קריטיות ([Critical Infrastructure](#)), כדוגמת תשתית החשמל והתקשורת, אשר בתורן משמשות כבסיס לתשתיות מידע קריטיות (Critical Infrastructure Information).

המרחב הקיברנטי (Cyber Space)¹, אשר בהתאם להגדרת מאמר זה כולל בחובו את העולם הווירטואלי אשר האדם יצר באמצעות טכנולוגית מידע ותקשורת (Information and Communication Technology), מאפשר לשחקנים (קואליציות, מדינות, ארגונים², קבוצות, פרטים) לבחור באסטרטגיות פעולה מגוונות ודינמיות. הבחירה של שחקן באסטרטגיה נגזרת משורה של קריטריונים, כדוגמת צרכים עסקיים-פוליטיים, זמן ותמונת המצב הקוגניטיבית אשר כל שחקן בונה לעצמו. הבחירה במונח "תמונת המצב הקוגניטיבית" אינה מקרית, אלא היא באה לחדד כי אין לשחקן אפשרות לקבל תמונה מצב מלאה וריאלית, ולפיכך בעת הבניית העולם השחקן נאלץ להסתמך על מודלים הסתברותיים ולא אקסיומות.

האסטרטגיה הדינמית אשר כל שחקן יבחר תשפיע על התנהלותו כלפי שחקן אחר, כדוגמת: שיתוף פעולה ויצירת בריתות, "ישיבה על הגדר", עימות גלוי, עימות עמום. כתוצאה מכך, כל שחקן בונה לעצמו את מטריצת ההתנהלות שלו כלפי השחקנים האחרים, כאשר שני היתרונות הבולטים במרחב הקיברנטי הינה היכולת לשנות את האסטרטגיה הנבחרת בקצב מהיר ובעלות נמוכה יחסית, וביכולת שחקן לאמץ בו זמנית מספר אסטרטגיות כלפי שחקן אחר, וזאת תוך צמצום האפשרות של השחקן שכנגד לחשיפתו של המשחק הכפול.

¹ ישנה סבירות גבוהה כי הגבולות בין המרחב הקיברנטי (Cyber Space), לעולם האלקטרומגנטי המופשט המוכר לנו כיום יטשטשו בעתיד.

² Non-State Entity

המונח נשק אינו חדש, ומטרתו להכליל את רשימת האמצעים שבהם צד יוכל להשתמש על מנת להטיל את מרותו החד צדדית של פלוני על האחר, וזאת לשם השגת מטרות כאילו ואחרות. המונח לוחמת מחשב³ (Cyber Warfare) כולל בחובו שורה של פעולות התקפיות אשר שחקן יכול ליזום כלפי שחקן אחר במרחב הקיברנטי.

כניסתו של הנשק הקיברנטי לזירה הרחיב את מרחב ההזדמנויות והכלים אשר עומדים לרשות כל שחקן, ואין פלא כי המרחב הקיברנטי זכה להכרה בעיני רבים כמימד החמישי (The fifth dimension) של שדה הקרב המודרני. בהתאם לכך מאמר זה סוקר, על קצה המזלג את עיקר השיקולים בעת ההחלטה להפעיל נשק קיברנטי.

שיקולים בהפעלת נשק קיברנטי

עלות פיתוח ותפעול

עלות פיתוח אמצעי לחימה מסורתיים, עשויה להגיע למאות, אם לא לעשרות מיליארדי דולרים, כאשר זמן הפיתוח עשוי להגיע לא פעם אף לעשרות שנים. אף עלות הייצור עשויה להסתכם בסכומים לא נמוכים לפריט, כאשר ראוי לציין כי השימוש באמצעי לחימה מסורתיים עשוי להגיע בנקל לעלות של 10,000 - 30,000 דולר לשעה. אף אורך חיי אמצעי לחימה מסורתיים נמוך מעשור במוצא, דבר המחייב חידוש מלאי באופן תקופתי.

בהתאם למספר מחקרים אשר בוצעו בשנים האחרונות, התגלה כי עלות פיתוח ה-[Stuxnet](#) מוערכת בכ- 10-20 מיליון דולר, דבר אשר מציג כי ישנם מקרים רבים בהם ניתן לפתח ולהשתמש בנשק קיברנטי, וזאת ללא צורך בהשקעה תקציבית גבוהה. לאור העובדה כי מדובר בעלות לא גבוהה יחסית, מדובר בפתרון אידיאלי עבור גורמים רבים שאינם בעלי גב כלכלי ענף, כדוגמת ארגוני פשיעה וטרור.

אנונימיות הפיתוח, המכירה והשימוש בנשק קיברנטי

פיתוח נשק קיברנטי אינו תלוי מקום גיאוגרפי וזמן, ולפיכך ניתן לפתח אותו אף ללא קשר ישיר בין גורמי הפיתוח. הלכה למעשה, מרבית האמצעים לפיתוח נשק קיברנטי זמינים מזה שנים רבות למרבית הציבור, ואף גורמים בעל כישורים טכניים ממוצעים יכולים לפתח כיום נשק קיברנטי אפקטיבי.

אפשרויות תשלום מבוססות מטבע וירטואלי, כדוגמת [ביטקוין](#) מקלות על תהליכי מכירה ורכישה של נשק קיברנטי ב"שוק השחור", ולפיכך ניתן לזהות מגמה של מכירה ורכישה של נשק קיברנטי בין שחקנים שונים, כאשר למרבה ההפתעה התגלה לא פעם כי אף מדינות (כדוגמת אייזרביג'ן אשר שמה עלה לדיון [בפרשת הפריצה ל-Hacking Team](#)) רוכשות נשק קיברנטי ממקורות שונים ומגוונים.

³המונח Cyber Warfare זכה לתרגום עברי נוסף - "לוחמה קיברנטית".

המרחב הקיברנטי מקשה מטבעו על איתור פעילות השחקנים, ביחוד כאשר שחקנים אלו מאמצים טכניקות של חמקנות, עמימות והסוואה. לפיכך, לגורמי אכיפה וביטחון ישנו קושי ניכר לאתר ולפגוע בשחקנים המשתמשים בנשק קיברנטי, כאשר יש לזכור כי מרבית התקיפות הקיברנטיות מתבצעות תוך זמן קצר יחסית, דבר אשר מקטין את ההסתברות לאיתור יוזם התקיפה ע"י גורמי האכיפה והביטחון.

אורך חיים קצר של נשק קיברנטי

הצלחתו של הנשק הקיברנטי תלויה במספר פקטורים מהותיים, כדוגמת קיומה של פגיעות שלא תוקנה או לא ידועה ([Zero Day Attack](#)), כשל במעגל אבטחה אשר ניתן לניצול לרעה, אי מימוש מנגנוני אבטחה וכשל אנושי. הדינמיות בעולם המחשוב משאירה לשחקן "חלון הזדמנויות" צר יחסית, ועל כל שחקן לבחון האם הוא רוצה ומסוגל לנצל את "חלון ההזדמנויות" הצר, שלאחר סגירתו, הנשק הקיברנטי אשר ברשותו יהיה חסר תועלת.

התמקדות במטרות איכות

אחד היתרונות הבולטים של הנשק הקיברנטי הינו היכולת להתמקד (להתביית בעגה הצבאית) ב"מטרות איכות" לשם השגת מטרות מוגדרות, תוך צמצום ההשפעה הרחבתית על תשתית הארגון המותקף. רוצה לומר, שיטת פעולה זו מצמצמת את חתך החשיפה של הפעילות העוינת, דבר המקשה על איתורה. כמו כן, באמצעות התמקדות במטרות איכות, השחקן אשר בוחר להשתמש בנשק קיברנטי מגדיל את הוודאות כי במקרה כי התקיפה תצלח, הנזק אשר יגרם לשחקן המותקף יהיה גבוה. עם זאת, התמקדות במטרות איכות מחייבת מודיעין מדויק, ואף מעמידה רף קושי גבוה יותר לחציה, וזאת מכיוון שבארגונים רבים מוטמעות בקרות מפצות רבות לשם הגנה על ישויות אשר מוגדרות כמטרות איכות פוטנציאליות.

שיבוש פעילות ודיסאינפורמציה

לאור העובדה כי עידן המידע יצר תלות גוברת במחשוב, הנשק הקיברנטי מאפשר שיבוש של פעילות נורמלית של שירות עסקי, תוך יצירת אשליה למפעיל כי השירות עסקי פועל באופן תקין. יתרה מכך, באמצעות שינוי נתונים ולאו הזנת מידע כוזב (דיסאינפורמציה) במערכות עסקיות, כדוגמת [Big Data](#), השחקן התוקף יכול להשפיע על תהליך קבלת ההחלטות בשחקן המותקף, דבר אשר עשוי לגרום לטעויות אסטרטגיות, כדוגמת קבלת החלטה על השקעה גבוהה בפתרון לא אפקטיבי ואופטימלי. דוגמא אחרת, הערכות לא נכונה מבחינת סד"כ לפעילות לחימה האמורה להתרחש במרחב הפיסי עשויה להוביל לתבוסה גורפת. וכדוגמא אחרונה אציין את היכולת ליצור אנדרלמוסיה כלכלית במדינה פלונית, וזאת ע"י הזנת מידע כוזב במערכות הפיננסיות ולאו המדיה.



ריגול, איסוף מידע ובניית פרופיל פסיכולוגי - התנהגות

עידן המידע הביא עמו תלות גוברת והולכת בזמינות, סודיות, מהימנות ואמינות המידע. נדיר לראות כיום ארגון אשר אינו מאחסן מידע באופן דיגיטלי. יתרה מכך, שירותים עסקיים רבים תלויים באופן ישיר במערכות המחשוב. מטבע הדברים, במהלך השנים מערכות המחשוב נהפכו ליעד תקיפה מועדף, אשר באמצעותו ניתן להפיק מידע איכותי, וזאת כדוגמת גניבת תוכניות מטוס ה-F-35 האמריקאי ע"י סין⁴. בהתאם לכך, ניתן לראות כי נשק הקיברנטי (דבר הכולל לא פעם שילוב של תקיפות מסוג "הנדסה חברתית") מתמקד לא פעם במציאת דרכים לאיתור מידע איכותי והוצאתו ממתחם הארגון, ובכלל זה באיתור אנשי מפתח בארגון, והוצאת מידע איכותי מרשותם.

בנוסף, ניתן להשתמש במידע הדיגיטלי הטמון במרחב הקיברנטי לשם בניית פרופיל פסיכולוגי - התנהגותי של פלוני, ובכך להכין את הקרקע למימוש תקיפות קיברנטיות שכיחות, ובכלל זה ניתן להשיג יכולת חיזוי מסוימת לגבי התנהגותו של פלוני במצבים מסוימים. וכך לדוגמה, הסטארטאפ [Crystal Project Inc.](#) מציע שירות המציע יכולת בניית פרופיל פסיכולוגי - התנהגותי של פלוני, וזאת על סמך מידע דיגיטלי הטמון במרחב הקיברנטי.

מן הראוי אף לציין כי בהתאם לפרסומים זרים, ה-NSA (National Security Agency) פרץ לאלפי מכשירי טלפון ניידים, וזאת במטרה לאסוף מהם צילומים מזירות אירוע בהן התרחשו אירועי טרור, דבר אשר אפשר לממשלת ארה"ב למנף את תהליכי החקירה. לפיכך, ניתן לזהות מגמה שבה שירותי ביטחון מנצלים יכולות תקיפה קיברנטיות לשם איסוף מודיעיני מאזרחי המדינה בה הם פועלים, דבר המעלה לדיון סוגיות מהותיות בנושא זכויות אזרח והגנת הפרטיות.

סחיטה ("כופר"), הונאה (Fraud) והלבנת הון (Money Laundering)

בשנים האחרונות החלו ארגוני פשיעה (בעיקר) להשתמש בתוכנת כופר (Ransomware) לשם סחיטת ארגונים ואנשים פרטיים, דבר הכולל הצפנת מידע חיוני, ודרישה לתשלום כופר לשם שחרור המידע הנמצא בחזקת התוקף. בנוסף, ניתן לראות מקרים שבהם בעלי אתרי אינטרנט (לדוגמה) נדרשים לשלם כופר לשם מניעת הישנות של תקיפות משביתות שירות, אשר פוגעות בפעילות העסקית של אתר האינטרנט.

כמו כן, באמצעות ניצול פגיעויות שונות, ארגוני פשיעה (בעיקר) משתמשים בכלים שונים לשם השתלטות על ציוד המחשוב ומכשירים ניידים, דבר המאפשר להם להפיק מידע אשר באמצעותו ניתן לסחוט את המותקף. דוגמה קלאסית לתקיפה מסוג זו היא הפעלת מצלמת המחשב באופן בלתי רצוני, וזאת לשם הכנת "סרט מביך" אשר יאפשר את סחיטת הצד המצולם. יוער כי שימוש בשיטות סחיטה מקובל מזה

⁴ [New Snowden Documents Reveal Chinese Behind F-35 Hack, Franz-Stefan Gady, 2015](#)



אלפי שנים בעת גיוס מקורות מודיעין (כדוגמת "משתפי פעולה"), והנשק הקיברנטי מקל ברמה מסוימת על גיוס מקורות מודיעין, וזאת תוך מתן אפשרות להסתיר הפרטים האמיתיים של הגורם המפעיל.

באמצעות שימוש בזהויות בדויות, ואף באמצעות התחזות \ גניבת זהויות, גורמים שונים יכולים לבצע מעשי הונאה (Fraud) והלבנת הון (Money Laundering) אשר מטרתם להעשיר את הצד התוקף. Zeus Trojan⁵ מהווה דוגמה קלאסית לכלי תקיפה קיברנטי אשר מאפשר לתוקף לגנוב את פרטי האימות של חשבון הבנק של אדם פלוני, ובכך לאפשר לתוקף לבצע פעולות פיננסיות בשם הקורבן.

עמימות

ארסנל הנשק הקיברנטי ניתן להסתרה בקלות יחסית, ולפיכך ישנו קושי רב לדעת מהן היכולות הפרקטיות של גוף פלוני. לאור העובדה כי ניתן לייצר נשק קיברנטי ללא סממנים המזהים את המפתח המקורי, ואף ניתן להשתמש בנשק קיברנטי באופן אנונימי (כדוגמת הפעלת הנשק מכתובת IP הרשומה על שם מדינה זרה, גרימה לצד שלישי שאינו מעורב בסכסוך בין הצדדים להפעיל את הנשק קיברנטי), דבר המקשה על הצד המותקף להוכיח מיהו התוקף⁶. יתרה מכך, באמצעות שתילת סממנים מזהים כוזבים בנשק הקיברנטי ניתן לגרום לכך שהצד המותקף יחשוד בגורם צד שלישי, שאינו קשור כלל לתקיפה. לפיכך הנשק הקיברנטי יכול לסייע ביצירת חשדנות ומתיחות, ואף במקרים מסוימים לייצר עימות לא רצוני בין גורמים אשר במקור לא תכננו להחריף את מערכת היחסים ביניהם.

יצירת רשת דארקנט ("רשת אפלה") ומחשוב סריגי (Grid Computing)

באמצעות שימוש בנשק קיברנטי, גורמים שונים יכולים להקים רשת דארקנט ("רשת אפלה") פרטית, אשר עצם קיומה ופעילותה מוסתר וממוסך תחת פעילות לגיטימית של משתמשים. במאמר מוסגר יצוין כי מעבר להקמת רשת דארקנט ("רשת אפלה"), כלי הנשק הקיברנטיים מאפשרים להשתמש בכוח מחשוב של משתמשים לגיטימיים לשם ביצוע פעולות הדורשות כוח עיבוד רב, כדוגמת כריית כסף וירטואלי ופענוח של מידע מוצפן.

הקדמה למלחמה מסורתית - "ערפל המלחמה" (The Fog of War)

הנשק הקיברנטי מאפשר לשחקן אשר מעוניין ליזום לחימה מסורתית לנקוט בשורה של צעדים מקדימים, כדוגמת שיתוק תשתיות קריטיות, פגיעה בשרשרת האספקה (Supply Chain) ובמערכות לוגיסטיקה, הסתרת שלבי ההכנה ליציאה לקרב וזריעת פאניקה בצד המותקף, דבר אשר מאפשר לצד היזום להשיג עליונות על השחקן המותקף, ובכך לשנות את כללי המשחק בזירה⁷. דוגמה קלאסית לתקיפה קיברנטית מסוג זו הינה השבתת פעילות של מערכת המחשוב האחראית לזימון כוחות מילואים בחירום, המתבססת

⁵ [Kaspersky Lab Discovers Chthonic: A New Strain of Zeus Trojan Targeting Online Banks Worldwide, 2014](#)

⁶ מושג שכיח המתאר את בעיה זו הינו "בעיית הייחוס" (Problem Attribution)

⁷ Cyberwarfare and Information Warfare Shock Doctrine, Yuval Sinay

על ממשקים חיצוניים המאפשרים המצאת זימון אוטומטי לחייל המילואים באמצעות פנייה קולית ולא דוא"ל ולא SMS (מסרון).

חלופה למלחמה מסורתית

קרל פון קלאוזביץ, מאבות תורת הלחימה המודרנית הטביע את המשפט - "המלחמה אינה אלא המשך המדיניות באמצעים אחרים". עם זאת, השימוש במלחמה מסורתית מחייב את הצד היוזם ליטול סיכונים מרובים, ובכלל זה להשקיע משאבים רבים על מנת להיערך לחימה, אשר זמן תחילתה ידוע, אך תאריך סיומה תלוי בערפל. יתרה מכך, קיומן של בריתות (כדוגמת "ברית נאט"ו") ואמצעי לחימה לא קונבנציונליים עשוי לגרור את הצד היוזם לעימות רוחבי, אשר בסופו עשוי להיגרם לצד היוזם נזק משמעותי, המקטין את כדאיות השימוש במלחמה מסורתית. כחלופה לכך, ניתן לזהות כי בשנים האחרונות גובר השימוש בנשק קיברנטי בין מדינות עוינות (כדוגמת העימות הנוכחי בין רוסיה לאוקראינה, והעימות בין איראן לאזרבייג'ן לפני שנים ספורות), וזאת כחלופה למלחמה מסורתית. היתרונות הגלומים בשימוש בנשק קיברנטי במקרה הנדון כוללים בין השאר את האפשרות להגביל את הפגיעה בצד המותקף (כדוגמת מניעת אובדן חיי אדם ופגיעה הרסנית בתשתיות פיזיות), אך עם זאת לשמר את היכולת לגרום לנזק מהותי (כדוגמת השבתת פעילות תשתית האינטרנט אשר משמשת לטובת פעילות עסקית) לצד המותקף.

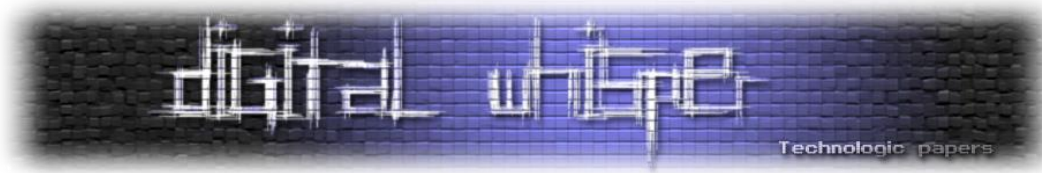
במאמר מוסגר יצוין כי דוגמא מעניינת לשימוש בעימות קיברנטי כחלופה למלחמה מסורתית הינו העימות אשר התקיים לפני פחות משנה בין ארה"ב לצפון קוריאה, אשר נסב אחר סרטה של חברת סוני - "[ראיון סופי](#)". במקרה הנדון נטען ע"י מקורות זרים כי מנהיג צפון קוריאה ("כוכב הסרט"), קים ג'ונג און נעלב מכך שהוא מוצג בסרט כאדם ילדתי ונלעג. עקב כך נטען כי צפון קוריאה יזמה מתקפה קיברנטית כנגד חברת סוני, דבר אשר כלל הדלפת מידע רגיש ממערכות המחשוב של החברה. בתגובה לכך נטען ע"י אותם מקורות, ארה"ב השביתה את פעילות האינטרנט של צפון קוריאה למשך יממה, וזאת מעבר לנקיטת שורה של צעדי ענישה נוספים כנגד צפון קוריאה.

בשנת 2011 אירן פרסמה כי היא הצליחה ליירט מל"ט (מטוס ללא טייס) אמריקאי, וזאת באמצעות שימוש בתקיפה מסוג GNSS-Spoofing⁸. צוות מחקר מאוניברסיטת טקסס באוסטין שחזר את מימוש תקיפה זו בשנת 2012⁹, דבר המעיד כי המרחב הקיברנטי חשוף לשורה של תקיפות מתקדמות הכוללות בחובן אף תקיפות ל"א (לוחמה אלקטרונית), אשר מאפשרות לתוקף להשיג את מטרותיו בדרכים מגוונות ויצירתיות.

מן הראוי לציין כי למרות שלל היתרונות בשימוש בלחימה קיברנטית ביחס למלחמה מסורתית, הסיכון כי מלחמה במרחב הקיברנטי תזלוג למרחב הפיסי שריר וקיים, בייחוד במצבים בהם הפגיעה במרחב הקיברנטי תגרום לפגיעה בחיי אדם, וזאת כדוגמת פגיעה במערכות מחשוב רפואיות במתקני רפואה

⁸ כיצד עלה בידי האיראנים ליירט מל"ט אמריקאי ומהי ההגנה הראויה? חיים רביב, 2015

⁹ חוקרים מאוניברסיטת טקסס "חטפו" מל"ט באמצעות זיוף אותות GPS, 2012



אזרחיים, גם אם בשגגה. ובמילים אחרות, הלחימה קיברנטית יכולה להסלים בקלות יחסית ללחימה מסורתית, ולפיכך ישנו צורך לבחון באופן מיטבי את ההשלכות האפשריות של תקיפה קיברנטית על הצד המותקף, ובהתאם לנקוט בצעדים הנדרשים לשם צמצום ההשלכות השליליות למינימום.

חדלון החוק הבינלאומי

דיני המלחמה מציגים שורה של חוקים והסדרים אשר מקובלים על מרבית מדינות העולם, ואף על ארגונים לא ממשלתיים, כדוגמת האו"ם. עם זאת, דיני המלחמה אשר שרירים ותקפים במלחמה מסורתית מתקשים לספק מענה הולם למלחמה במרחב הקיברנטי, דבר המקל על הצדדים לנהל "מלחמה וירטואלית", וזאת ללא הגבלות משפטיות של ממש. יתרה מכך, דיני המלחמה הנוכחיים מתקשים להתמודד עם שורה של סוגיות משפטיות-מעשיות, כדוגמת מהי התגובה הראויה שעל מדינה לאמץ במקרה שתוצאות תקיפה במרחב הקיברנטי שלה משפיעות על המרחב הפיסי שלה, וזאת עקב טעות אנוש מצדו של הצד התוקף. דוגמא אחרת הינה סוגיית אחריות מדינה אשר דרך מערכת התקשורת העוברת במרחב הטריטוריאלי שלה בוצעה תקיפה קיברנטית ע"י מדינה פלונית כנגד מדינה אלמונית. ודוגמא אחרונה הינה השאלה מהם הגבולות של זכות ההגנה העצמית של מדינה אשר חווה תקיפה קיברנטית אשר מקורה ביוזמה עצמאית של אזרח ממדינה פלונית.

יוער כי ניסיונה של ברית נאט"ו¹⁰ להתאים את דיני מלחמה הנוכחיים למרחב הקיברנטי, תוך השגת הסכמה בינלאומית לא זכה להצלחה יתרה.

אקטיביזם

באמצעות שימוש בנשק קיברנטי, גופים שונים יכולים לנקוט בצעדים אקטיביסטים שונים, כדוגמת השתלטות על אתר אינטרנט מרכזי לשם פרסום משנתם, ואף לשם הענשת הגורם אשר לטענת אותם אקטיביסטים אינו פועל כמצופה ממנו. ולראיה פעילות קבוצת [Anonymous](#) המהווה דוגמא לסנונית הראשונה לפעילות אקטיביסטית "לא פוליטית" (כהגדרת חברי הקבוצה) במרחב הקיברנטי, אשר בהתאם למטרות המוצהרות של הקבוצה ברצונה לעורר מודעות חברתית לנושאים מהותיים, תוך חתירה לצדק.

טרור

יכולותיו של הנשק קיברנטי, ואופי השימוש בו, הופכים את הנשק הקיברנטי לפתרון אטרקטיבי עבור גופים המעוניינים להשליט טרור על מדינה ולא ציבור מסוים. תוצאות פעולת טרור במרחב הקיברנטי יכולות לכלול בין השאר: פגיעה בתדמית, חשיפת מידע מביך אישים פוליטיים, חשיפת מידע אשר עשוי לגרום לסכסוך עם מדינה פלונית, פגיעה ביציבות המערכת הפיננסית ויצירת מצב של אי אמון בין העם לשלטון. מגבלות משפטיות ומעשיות (כדוגמת מגבלות טכנולוגיות) מקשות על גופי אכיפה וביטחון לספק מענה

¹⁰ Tallinn Manual on the International Law Applicable to Cyber Warfare, Michael N. Schmitt, Cambridge University Press; Reprint edition, 2013



הולם לסוג איום זה, דבר המגדיל הסבירות להצלחת תקיפות קיברנטיות מטעם גורמים אלו, כאשר גורמים אלו מודעים לכך כי הסבירות כי הם יענשו בגין מעשיהם נמוכה.

במאמר מוסגר יצוין כי בשנים האחרונות התגלה קשר ישיר בין ארגוני טרור לארגוני פשע מאורגן, דבר הכולל בין השאר רכישת כלי נשק קיברנטיים אשר פותחו ע"י ארגוני פשע מאורגן, אשר מאפשרים לארגוני טרור להשיג עצמאות כלכלית. סוגיה זו זוכה לחשיבות יתרה לאור העובדה כי היא מאפשרת ל"מפגע יחיד" להשיג גב כלכלי החיוני למימוש תקיפה פיסית מסיבית, וזאת תוך הסתרת עצם קיומו ופעילותו מגורמי אכיפה וביטחון.

הנשק הקיברנטי כמכפיל כוח

הנשק הקיברנטי מאפשר לכל גורם להכפיל את כוח הלחימה שלו, וזאת בהשקעה מינימלית. כמו כן, הגישה המקובלת בעת קיומו של מרוץ חימוש (Arms Race) בין גופים שונים היא שאם כלי נשק יכול להגיע לידי הצד האחר, אזי חלה חובה לפתח יכולות דומות. בנוסף, עקרון ההדדיות בלחימה מאיץ את השימוש בכלי נשק בעלי יכולות דומות בשלבי הלחימה הראשונים תוך מתן אפשרות לתוקף המפתיע את המותקף להשיג הישגים מהותיים כבר במערכה הראשונה, דבר ההופך את הנשק הקיברנטי לאטרקטיבי בעיני רבים.

השלם גדול מסך חלקיו

אחת היכולות היותר מעניינות בנשק קיברנטי היא היכולת לרתום את כוחם של אחרים, וזאת אף ללא ידיעתם והסכמתם, לשם מימוש המתקפה. וכך לדוגמא, ממשלת סין פיתחה נשק קיברנטי בשם [Great Cannon](#). נשק זה מנצל פעילות של משתמשים לגיטימיים, אשר מפעילים ללא ידיעתם סקריפט מבוסס JavaScript המאפשר יצירת מתקפה משביתת שירות מסוג DDoS (Distributed Denial-of-Service) Attack כנגד אתר פלוני. דוגמא אחרת הינה מצב שבו גורם עוין משתלט על מערכות המחשוב של מטוס נוסעים ו\או מגדל פיקוח, ובאמצעות מתן הנחיות מרחוק הוא גורם לפיגוע המונים, נוסח פיגוע הטרור במגדלי התאומים בארה"ב מה-11 בספטמבר 2001.

שליטה על התודעה וצנזורה

הנשק הקיברנטי מאפשר למדינות וארגונים להחיל את משנתם במרחב הקיברנטי, וזאת באמצעות החלת ניטור ושלל הגבלות על פעילות המשתמשים. פתרון ה-Great Firewall אשר פותח על ידי ממשלת סין הוא דוגמא קלאסית למימוש פעולות ניטור והגבלת פעילות משתמשים. [Edward Snowden](#) חשף את קיומו של כלי נשק קיברנטי בשם [QUANTUM](#) אשר מטרתו העיקריות כוללות בין השאר; לאפשר לממשלת ארה"ב לפצח מידע מוצפן, ולאפשר לממשלת ארה"ב לשתול Malware במיליוני מחשבים, וזאת תוך זמן קצר. למותר לציין כי אופי פעילות הכלי מעיד כי הוא תוכנן במקור להפצת בוטס, אך במקביל הוא מסוגל להתקין תוכנות מעקב במחשבים וטלפונים ניידים של קבוצות יעד גדולות. יוער כי לאחרונה אף הועלתה

שיקולים בפיתוח והפעלת נשק קיברנטי

www.DigitalWhisper.co.il



טענה כי ה-FBI השתמש בסט הכלים של חברת ה-[Hacking Team](#) לשם השתלת¹¹ כלי מעקב במחשבים טלפונים ניידים של חשודים.

מישל פוקו¹², הוגה דעות צרפתי הציג את [הפנאופטיקון](#) כמודל הפיקוח אולטימטיבי, הגורם לאסיר להפנים את ההתנהגות הרצויה (הראויה), וזאת ללא שימוש באמצעי ענישה פיזיים. רוצה לומר, עצם העובדה כי אדם פלוני יודע כי הוא נתון למעקב פוטנציאלי בכל זמן נתון, דיה בכדי ליצור שינוי התנהגותי, ויכולת זו ניתנת להשגה במרחב הקיברנטי וזאת באמצעות שימוש בנשק קיברנטי.

מן הראוי אף לציין כי ישנו צפי כי השימוש בממשק אדם-מכונה יגבר בעתיד הקרוב, ולפיכך גורמים עוינים יוכלו את ממשק זה על מנת להפוך את "האדם" לממשק המקשר בין הנשק הקיברנטי למערכת המחשוב המותקפת. הדור הראשון של כלי הנשק אשר מאפשר השתלטות מוגבלת על אדם פלוני מרחוק זמין בשוק, והוא מוכר בשם "נשק אנרגיה ישירה" ([Directed Energy Weapon](#)). עם זאת, נכון לזמן כתיבת מאמר זה, ובכפוף למידע החשוף לנחלת הכלל, הדור הראשון אינו כולל יכולת לניצול לרעה של ממשק אדם-מכונה.

לחימה היברידית

אסטרטגיות ודוקטרינות הלחימה החדשות מציעות שילוב בין לחימה מסורתית לבין לחימה קיברנטית, וזאת בהתאם לצורך. וכך לדוגמא, לאחרונה פורסם כי חברת בואינג¹³ מפתחת מל"ט (מטוס ללא טייס) הכולל רכיב חומרה בשם TNI (Tactical Network Injector) אשר מהווה יחידת אחסון לנשק קיברנטי (סביר להניח שיחידת האחסון מכילה Framework הדומה ביכולותיו ל-[Metasploit](#)), אשר ביכולתו לאפשר החדרת קוד זדוני לציוד המחובר לרשתות אלחוטיות (Wi-Fi), וזאת במטרה לאפשר מימוש למתקפת (Man In the Middle) MiTM וניצול Exploits. בנוסף, למל"ט ישנה יכולת לביצוע פעולות ריגול, ניטור ומעקב. למרות שלא פורסם מידע רשמי בנדון, סביר להניח כי המל"ט מצויד בראש נפץ אשר מקנה למל"ט יכולת לפגוע בעת הצורך ב"מטרות איכות", כדוגמת מערכות תקשורת ומכ"ם (מגלה כיוון ומרחק), וזאת בנוסף לקיומו של מנגנון השמדה עצמי מובנה.

¹¹ [FBI Used Hacking Team's Help to Track Tor User](#), Adarsh Verma, 2015

¹² לפקח ולהעניש - הולדת בית הסוהר, מישל פוקו, רסלינג הוצאת ספרים, 2015 (גרסה בצרפתית ובאנגלית של הספר פורסמה בחו"ל בשנת 1975)

¹³ [Hacking Team and Boeing Built Cyber Weaponized Drones to Spy on Targets](#)



סיכום

עידן המדע הפך את המרחב הקיברנטי לשדה לחימה, אשר שחקנים רבים יכולים לנצלו לשם השגת מטרותיהם. מאמר זה סקר על קצה המזלג את השיקולים העיקריים בפיתוח והפעלת נשק קיברנטי, כאשר יש לזכור כי לאור המציאות הדינמית, כניסתן של טכנולוגיות ומערכות אקולוגיות מתקדמות (כדוגמת [IoT-IoE](#)), סביר להניח כי רשימת השיקולים תגדל בעתיד. ניתן אף להניח כי בעתיד הקרוב השימוש בלחימה היברידיית יגבר, וכי עידן חדש של מרוץ חידוש נכנס לזירה.

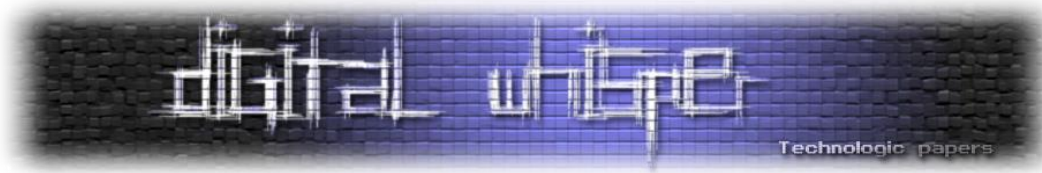
"If you spend more on coffee than on IT security, you will be hacked. What's more, you deserve to be hacked.", [Richard A. Clarke](#)

על המחבר

[יובל סיני](#) הינו מומחה אבטחת מידע, סייבר, מובייל ואינטרנט, חבר קבוצת SWGDE של משרד המשפטים האמריקאי. כמו כן, יובל סיני קיבל הכרה מחברת [Microsoft](#) העולמית כ-MVP בתחום Enterprise Security.

מילות מפתח

Armed Conflict, Critical Infrastructure Information, CII, Critical Infrastructure, CI, Cyber Conflict, Cyber Power, Cybersecurity, Cyber Space, Cyber Strategy, Cyber Warfare, Electronic Warfare, EW, Homeland Security, Hybrid War, Information and Communication Technology, ICT, International Law, Strategic Thinking



ביבליוגרפיה

ביבליוגרפיה בעברית

- [כיצד עלה בידי האיראנים ליירט מל"ט אמריקאי ומהי ההגנה הראויה? חיים רביב, 2015](#)
- [איום ארגוני הטרור במרחב הסייבר, גבי סיבוני, דניאל כהן, אביב רוטברט, צבא ואסטרטגיה, כרך 5, גיליון 3, דצמבר 2013](#)
- [מבוא ל-Web 3.0 Security, יובל סיני, Digital Whisper, 2013](#)
- [תפוצת נשק קיברנטי במרחב הסייבר, דניאל כהן, מבט על, גיליון 444, 08 יולי 2013](#)
- [חוקרים מאונ' טקסס "חטפו" מל"ט באמצעות זיוף אותות GPS, 2012](#)
- [מבט בינתחומי על אתגרי הביטחון בעידן המידע, יצחק בן-ישראל, ליאור טבנסקי, צבא ואסטרטגיה | כרך 3 | גיליון 3 | דצמבר 2011](#)
- [הגנה על תשתיות קריטיות מפני איום קיברנטי, ליאור טבנסקי, צבא ואסטרטגיה | כרך 3 | גיליון 2 | נובמבר 2011](#)
- [הוודאות האבודה של הטבע והאחדות הקוואנטית, צבי ינאי, מחשבות 55-56 | אפריל 1988](#)

ביבליוגרפיה באנגלית

מאמרים:

- [Police bust huge hacker black market](#)
- [China's Great Cannon](#)
- [Cyber Strategy - United States Department of Defense](#)
- [Here's What a Cyber Warfare Arsenal Might Look Like](#)
- [New Snowden Documents Reveal Chinese Behind F-35 Hack, Franz-Stefan Gady, 2015](#)
- [4 Arrested in Schemes Said to Be Tied to JPMorgan Chase Breach](#)
- [WATCH: Is Anonymous becoming the 'modern-day technological Robin Hood'?](#)
- [Hacking Team and Boeing Built Cyber Weaponized Drones to Spy on Targets](#)
- [On Cyberwarfare, Fred Schreier, DCAF Horizon 2015 Working Paper Series \(7\)](#)
- [Critical Infrastructure Protection against Terrorist Attacks, Course Report, NATO COE DAT, Ankara Turkey, 3-7 November 2014 \(Mon-Fri\)](#)
- [Kaspersky Lab Discovers Chthonic: A New Strain of Zeus Trojan Targeting Online Banks Worldwide, 2014](#)

שיקולים בפיתוח והפעלת נשק קיברנטי
www.DigitalWhisper.co.il



- [Why cyber warfare is so attractive to small nations](#)
- [How the NSA Plans to Infect Millions of Computers with Malware, 2014](#)
- [A Treaty for Governing Cyber-Weapons: Potential Benefits and Practical Limitations, Louise Arimatsu, International Law Programme, Chatham House, London, UK, 2012](#)
- [Social Business Systems: Beyond Engagement](#)
- [Threat Assessment & Remediation Analysis \(TARA\), Methodology Description Version 1.0 , Jackson Wynn, Joseph Whitmore, Geoff Upton, Lindsay Spriggs, Dan McKinnon, Richard McInnes, Richard Graubart, Lauren Clausen, MITRE, October 2011](#)
- [Cyberwarfare and International Law, Nils Melzer, 2011](#)
- [The UK Cyber Security Strategy Protecting and promoting the UK in a digital world](#)
- [Civilians in Cyberwarfare: Conscripts, Susan W. Brenner - University of Dayton & Leo L. Clarke - Grand Rapids, Michigan, Vanderbilt Journal of Transnational Law, \[Vol. 43:1011\], 2010](#)
- [BEHIND THE GREAT FIREWALL: THE INTERNET AND DEMOCRATIZATION IN CHINA, Xiaoru Wang, University of Michigan, 2009](#)
- [CIP Program Discussion Paper Series, George Mason University, February 2007](#)

ספרים:

- Understanding Cyber Warfare and Its Implications for Indian Armed Forces, Col R Tyagi, Vij Books, 2013
- Cyberpower and National Security, Ed by Franklin D, Kramer, Stuart H Starr and Larry Wentz, Vij Books, 2009
- Defense Strategies for Protection of People & Facilities against Bioterrorism, James Afshar, 2006

ניהול סממאות וזהויות ברשתות מיקרוסופט

מאת יהודה גרסטל

הקדמה

כפי שאני מכיר את קהל הקוראים שלי (כלל לא) יש סיכוי לא רע שלא תצליחו לשרוד עד סוף המאמר ולכן החלטתי לחלק חלק מההקדשות והתודות כבר פה בהתחלה. אז תודה מיוחדת להוריי ואשתי שהביאוני עד הלום, להם ולילדיי היקרים שבזכותם יש טעם לחיי.

וכעת למאמר. המאמר מחולק לשלושה חלקים:

1. תיאור המערכת עצמה
2. חולשות וסקירת פרצות
3. התמודדות עם החולשות

בכל אחד משלושת החלקים האלו יופיעו שלושה נושאים אופייניים, כלומר:

- שמירת סיסמאות במערכת לטווח ארוך (נדון בכך בשלושה אופנים: גם כיצד זה עובד בתכלס, גם אלו חולשות ופרצות קיימות במנגנון הזה וגם כיצד מגינים מפני כך).
- שמירה ושימוש מקומי בסימאות בזיכרון (ושוב לכל אורך שלושת הנושאים)
- כיצד הזהות וההזדהות עוברים ברשת.

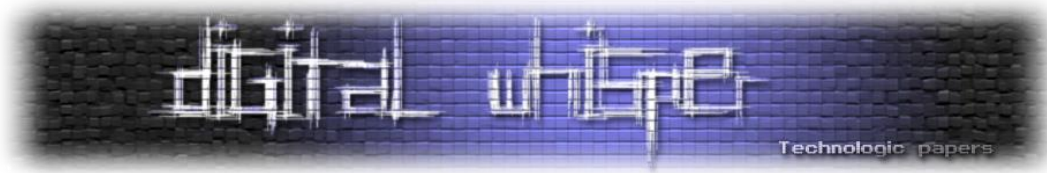
נתחיל מדברים על קצה הרלוונטי ונגיע עד ימינו אנו.

אז איך כל הסיפור הזה עובד? מה קורה בעצם כשאתם מתחברים למערכת חלונות? אתם מזינים את שם המשתמש והסיסמא... ומה אז?

האשים - גיבוב.

בסעיף זה נדון באופן שבו נוהגים לשמור ולאמת סיסמאות בעולם המחשבים - באמצעות גיבוב. אם העיקרון הזה מוכר וברור לכם, הרגישו בנוח לדלג הלאה לכותרת הבאה.

אחת הטעויות השכיחות שיכולים לבצע מפתחים היא שמירת מידע רגיש בצורה גלויה. למשל, לפתח אפליקציה רפואית עם משתמשים רשומים ולשמור את פרטי המשתמשים כמו המידע הרפואי הפרטי שלהם בצורה לא מוצפנת. כאשר גורם לא מורשה מצליח להגיע איכשהו אל מסד הנתונים, הוא מסוגל



לשלוף כמויות אדירות של מידע רגיש ומסווג ללא קושי. הפתרון הפשוט כדי להימנע מגניבה מעין זו הוא להצפין את המידע הרגיש.

באופן דומה, כדי שנוכל לאמת התחברות, לכאורה על הסמאות להיות שמורות במערכת היכן שהוא - כדי שאפשר יהיה לבצע השוואה בין הפרטים שמסר המשתמש המזדהה לאלו המקוריים שהוכנסו בפעם הראשונה. כאן, בסיסמאות, יש למתמטיקאים טריק נוסף, שונה מהצפנה. מאחר והמידע המבוקש הוא ברור, ידוע ונקודתי והלקוח/המשתמש גם מספק אותו בעצמו, אין צורך לשמור אותו בצורה כזו שנוכל ממש לקרוא אותו - עלינו רק לבדוק האם הנתונים שסיפק הלקוח הם אותם הנתונים שקבע הוא בעצמו בפעם הראשונה. מה כן עושים?

את המחרוזת של הסיסמא מעבירים תהליך מתמטי חד כיווני. בניגוד להצפנה אין כאן מפתח והתהליך לא ניתן לשחזור ופענוח. מאפיין נוסף של התהליך הוא שלא משנה מה אורך הקלט - תהא זו סיסמא באורך שמונה תווים או מחרוזת קובץ באורך של חמישים מגה-בייט - הפלט של הפונקציה המתמטית יהיה באורך זהה (כדי להשיג תוצאה זו משתמשים ב-"ריפוד" אבל לא נכנס לפרטים כרגע). התהליך הזה נקרא גיבוב, או בלע"ז 'האש' - HASH.

כך בערך זה נראה: בפתיחת חשבון חדש המשתמש מזין את שמו והסיסמא שלו. מאחורי הקלעים המערכת לוקחת את הסיסמא ומעבירה אותה בפונקציית גיבוב מסוימת. רק תוצאת הגיבוב נשמרת במסד הנתונים המקומי ולא הסיסמא עצמה. בזמן אימות גישה למערכת - הלקוח מזין שוב את הסיסמא שקבע בפתיחת החשבון, מתבצע שוב תהליך הגיבוב והמערכת משווה את המחרוזת שנוצרה זה עתה מול זו שקיימת כבר במסד הנתונים שלה. בצורה כזו הסיסמא עצמה אינה נשמרת במערכת לעולם בשום צורה ולמעשה גם נמצאת בשימוש בצורה גלויה בזמן הקצר ביותר האפשרי. אם פורץ הצליח להגיע למסד הנתונים (או כל איזור אחסון שמכיל את גיבובי הסיסמאות) - הוא לא יכול לעשות עם זה דבר.

כמעט...

אין לתוקף דרך להוציא את הסיסמא מתוך הגיבוב, אבל הוא כן יכול לנסות ליצור גיבובים מכל סיסמא אפשרית ולהשוות את התוצאות שלו עד שימצא את הסיסמא המתאימה - נרחיב על כל זה בחלק שדן בתקיפה (בחלק השני).

אלגוריתמי גיבוב במייקרוסופט

עכשיו, אחרי שדנו קצת בגיבובים בואו נדבר על איך מייקרוסופט עושים את זה. בסעיף זה נדון בשלושת הגיבובים הקיימים במערכות חלונות:

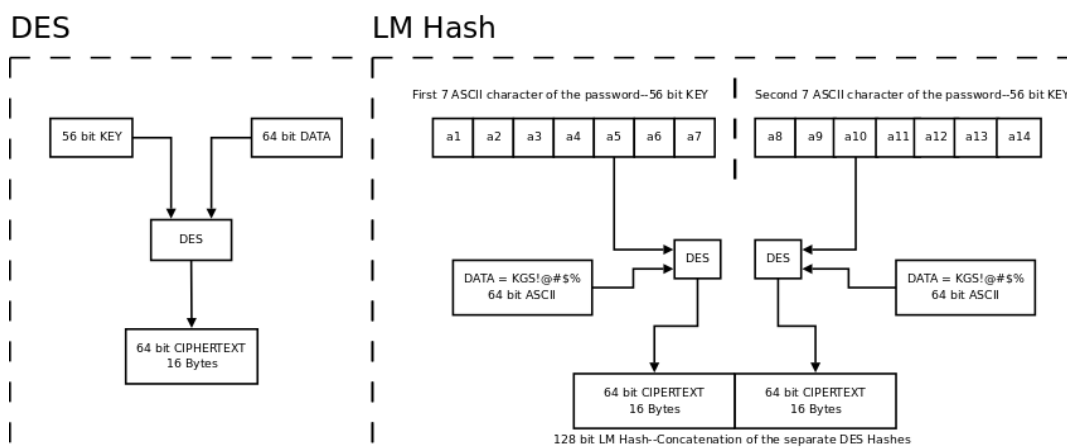
- גיבוב/הצפנה LM
- גיבוב NTLM
- גיבוב MS-CACHE

:LM

נחיל מההתחלה הכי רחוקה שעדיין רלוונטית: מערכות חלונות 98 ו-NT. במערכות אלו פיתחה מיקרוסופט מנגנון אימות שנקרא LAN Manager או בקיצור LM. מנגנון זה אינו מבצע ממש גיבוב במובן המלא משום שהוא משתמש בפונקציה שנעזרת במפתח הצפנה - זו הצפנה חד-כיוונית. ההצפנה היא מסוג DES והמפתח המשמש להצפנה הוא הסיסמא שלכם עצמה. ההצפנה מתבצעת על מחרוזת קבועה, מפורסמת וידועה: **KGS!@#%\$**.

לפני ביצוע ההצפנה המערכת מחלקת את המחרוזת של הסיסמא לשני חלקים של שבעה תווים. (אם אתם יודעים חשבון בסיסי של כיתה ג' הצלחתם להסיק נכון שסיסמאות במערכות שקדמו לחלונות 2000 אינן תומכות בסיסמאות שארוכות מ-14 תווים). במידה ואחד החלקים אינו באורך שבעה תווים מלאים - המערכת משלימה את האורך החסר ו"מרפדת" אותו בתווי NULL.

בנוסף, חשוב לדעת כי המנגנון אינו תומך בתווים מיוחדים, כלומר הוא מוגבל ל-128 סוגי תווים. את שני החלקים השווים של שבעת התווים מעביר ה-LM לאותיות גדולות ורק אז מפעיל את פונקציית ההצפנה על המחרוזת הקבועה שהזכרנו קודם (מכל חלק נוצרת מחרוזת בת 16 תווים). לבסוף האלגוריתם פשוט מחבר את שני החלקים שנוצרו לכדי מחרוזת אחת בת 32 בתים, להלן תרשים אודות מנגנון זה:



[במקור: https://courses.cit.cornell.edu/ece576/FinalProjects/f2008/tt236/tt236/high_level_design.html]

ניהול ססמאות וזהויות ברשתות מיקרוסופט

www.DigitalWhisper.co.il



לדוגמא, אם הסיסמא שלי היא "thisSmypass". המערכת תפרק את הסיסמא שלי לשני חלקים:

- החלק הראשון - thisSm
- החלק השני - ypass
- החלק השני **אינו** באורך של שבעה תווים ולכן אנחנו מרפדים אותו בערך כך: ypassXX

כל אחד משני החלקים משמש כמפתח והמחרוזת עוברת תהליך של הצפנה חד כיוונית -

- המילה thisSm הופכת ל-D478C5B5AB58795A
- המילה ypassXX הופכת ל-B7624FF226D45722
- והמחרוזת המלאה היא: D478C5B5AB58795AB7624FF226D45722

אתם יכולים לשחק עם הנושא [באתר הזה](#)

הערת צד: כאשר הסיסמא קצרה משמונה תווים, למעשה כל החלק השני ריק ומרופד בתווי NULL כך שהתוצאה תמיד תהיה: 0xAAD3B435B51404EE. באופן הזה ניתן לדעת במבט קצר על מחרוזת הגיבוב האם הסיסמא ארוכה משבעה תווים או אפילו ריקה לגמרי.

עד כאן לגבי שיטת הגיבוב של LM.

NTLM

מאז חלונות 2000 קיים סוג נוסף של גיבוב בשימוש על ידי ה-LAN Manager שלנו. שיטה זו נקראת NT או באריכות - NTLM (ה-NT מגיע מכך שמחלונות 2000 מיקרוסופט החלו לכנות את הטכנולוגיה שלה כ-New Technology). שיטה זו פשוטה למדי והיא משתמשת בפונקציית גיבוב ידועה ומוכרת בשם - MD4. הסיסמא שלנו מוזנת כקלט לפונקציה הנ"ל ושבה אלינו כמחרוזת שונה לחלוטין באורך קבוע של 32 תווים ב-UNICODE. בשיטה זו ניתן להשתמש בסיסמא באורך של עד 127 תווים (מגבלה שקשורה לאורך הקלט בתיבת הזנת הסיסמא, מבחינה תכנותית אפשר ליצור סיסמאות ארוכות יותר). שימו לב שבשיטה זו ניתן גם להשתמש בכל התווים הנתמכים ב-UNICODE (כולל סיסמאות בעברית לדוגמא).

MSCACHE וקרברוס

השיטה השלישית והאחרונה שנדון בה אינה שונה במהותה משיטת ה-NTLM. שיטה זו פועלת בסביבות דומיין של מיקרוסופט. לאלו מכם שלא מכירים מדובר בסביבה "עסקית" של רשת מחשבי חלונות. הסביבה מאפשרת ניהול מסודר ומרוכז של המשתמשים ומשאבי הרשת כמו גם מדיניות ונהלים שיחולו ברמת המחשבים והמשתמשים. בסביבה "מתחם" שכזו קיים תמיד שרת מרכזי המשמש בין שאר תפקידיו גם כשרת לאימות הזדהות. ברשתות מיקרוסופט מבוססות דומיין החל מיונידוס 2000 והלאה מתבצע אימות בעזרת פרוטוקול שנקרא קרברוס. הפרוטוקול משמש לזיהוי, אימות ולקבלת הרשאות בין



מערכות ולכן נדון בו מעט יותר בנושא האופקי השלישי של "שימוש בסיסמאות בפועל", כרגע אנחנו רוצים להתרכז באיך הנתונים נשמרים במערכת לאורך זמן.

מטמון

כפי שציינו ממש עכשיו האימות ברשתות ארגוניות מתבצע מול שרת מרכזי (שירות הקרברוס מופעל בדרך כלל על השרת המשמש כ-"Domain Controller" וברשתות גדולות קיימים אף מספר שרתים המשמשים לתפקיד זה), אבל מה קורה כאשר השרתים אינם זמינים? כדי שניתן יהיה להשתמש במחשב ובשירותים מרוחקים בכל זמן, מערכת חלונות מאפשר אימות מול "מטמון" / זמני - CACHE. השם של אימות זה נקרא בפשטות MS-CACHE והוא המשך של מנגנון ה-NTLM אותו כבר הזכרנו. מחרוזת הגיבוב של הסיסמא אותה יצרנו ב-MD4 בשיטת ה-NTLM עוברת תהליך נוסף. אל המחרוזת המגובבת של הסיסמא מצורף שם המשתמש באותיות קטנות ללא שם המתחם (דומיין) והמחרוזת המשולבת עוברת דרך הפונקציה של MD4 פעם נוספת.

אז איפה כל זה נשמר?

כמובן וכאמור בהקדמה, כדי להצליח לאמת את בקשת ההזדהות, כל ההאשים האלו נשמרים בקבצי מערכת במקום כלשהו.

"הרישום" של מיקרוסופט הידוע בכינוי Registry נשמר במספר קבצי מערכת. קובץ SAM שנמצא בנתיב הסטנדרטי הבא: C:\Windows\System32\Config מכיל את כל ההאשים משלושת הסוגים:

- LM
- NTLM
- MS-CACHE

במקרה של MS-CACHE הגיבובים השמורים מוצפנים באמצעות מפתח LSA. כמו כן מדובר במטמון ולכן המערכת שומרת למעשה רק את הגיבובים של עשרת המשתמשים האחרונים שהזדהו.

גרסאות - תמיכה לאחור

שיטת NTLM משמשת במערכות מסוג חלונות 2000 ו-XP ובגרסאות השרתים המקבילות: 2000 ו-2003. חשוב לדעת אמנם, כי לצרכי תמיכה בשיטות ישנות, מערכות אלו תומכות כברירת מחדל בשיטת LM הישנה - כך שלמעשה הסיסמא נשמרת בשתי התצורות גם יחד.

החל מגרסת Windows Vista והלאה, מערכת חלונות אינה תומכת ב-LM כברירת מחדל, אולם ניתן לשנות זאת ולאפשר תמיכה גם באימות מסוג LM. ניתן לעשות זאת (עם כי זה לא מומלץ) ע"י שימוש ב-

Group Policy.

ניהול ססמאות וזהויות ברשתות מיקרוסופט

www.DigitalWhisper.co.il

סביבת מתחם (Microsoft Domain)

בסביבות מתחם של מיקרוסופט יש למעשה שני סוגי חשבונות משתמשים:

- משתמשי וקבוצות מתחם / משתמשי רשת או Active Directory בלע"ז.
 - משתמשים וקבוצות מקומיים (הרגילים שקיימים בכל מערכת הפעלה של מיקרוסופט)
- סמאות של כל המשתמשים הרשתיים (משתמשי Active Directory) נשמרות בתצורת ה-NTLM שלהן בקובץ מרכזי בשם NTDS.DIT. הקובץ מצוי בכל שרת מסוג Domain-Controller שמשמש לאימות, אך פרטי המשתמשים המקומיים המוגדרים על עמדות הקצה עצמן עדיין שמורים מקומית בכל תחנה ותחנה.

איך התהליך מתרחש בפועל כאשר המחשב דולק?

עכשיו אחרי שסיימנו את הנושא האופקי הראשון - איפה זה נשמר - בואו נדבר על איך הסיסמאות נשמרות בטווח הקצר ונעשה בהן שימוש מקומי (השלב הבא יהיה שימוש במרחב הרשת).
כאשר המחשב נדלק, מופעל רכיב בשם LSA, ראשי תיבות: Local Security Authority. הרכיב אחראי בין היתר גם על טעינת הגיבובים מאמצעי האחסון ושמירתם במיקום זמין בזיכרון. רכיב ה-LSA מריץ תהליך במערכת שאולי נתקלתם בו מספר פעמים ועכשיו גם תדעו מהו - תהליך בשם LSASS, כלומר Local Security Authority Subsystem Service. LSASS אחראי על כל תהליכי האימות, בין אם מדובר בהזדהות מקומית בכניסה למחשב, בגישה למשאבים שונים ברשת או בניסיונות גישה מרוחקים למשאבים במחשב הנוכחי (כמו קבצים ומדפסות).

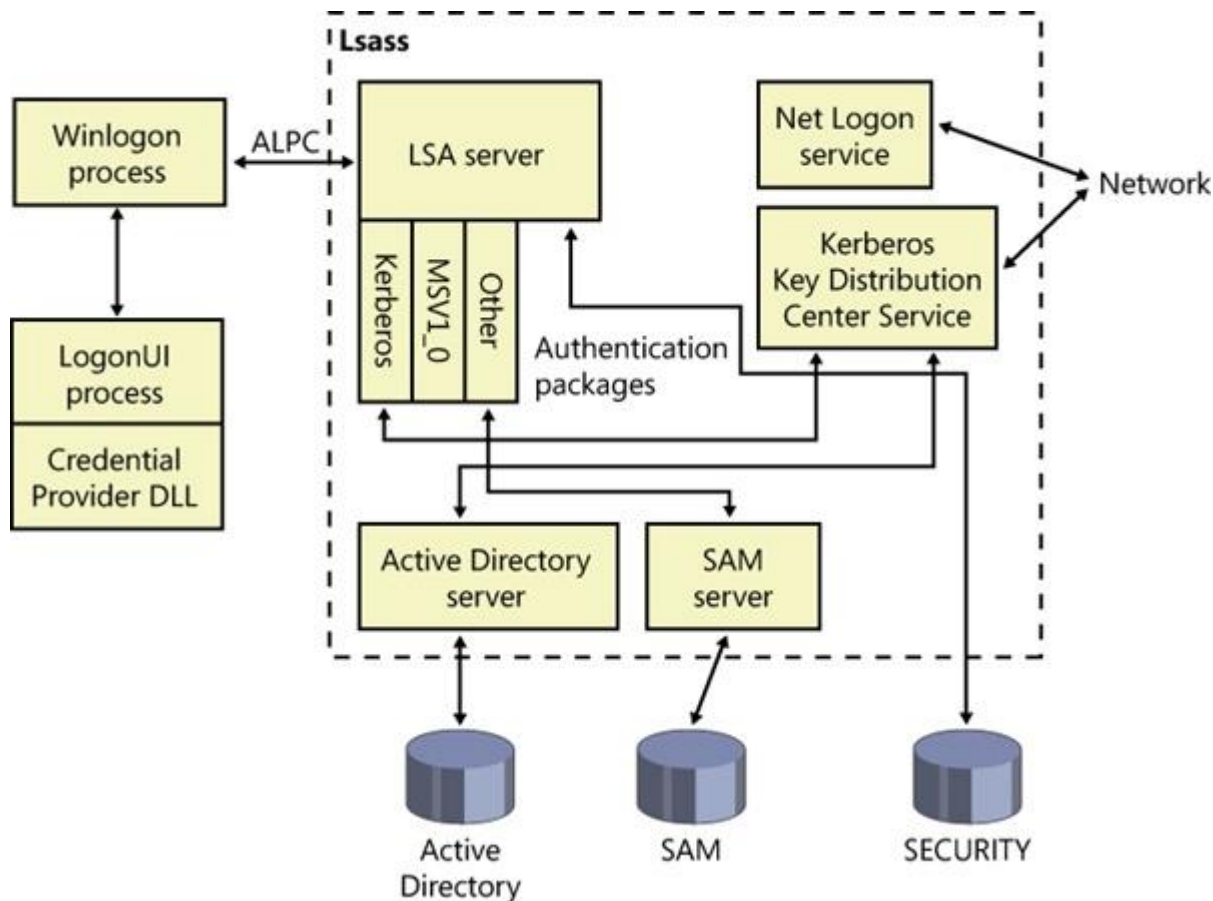
כאשר מופיע מסך כניסה בעליית מערכת חלונות, מאחורי הקלעים תהליך בשם WINLOGON מציג את הבקשה להזנת משתמש וסיסמא. התהליך מעביר את הקלט של המשתמש אל תהליך ה-LSASS יחד עם הגדרה לחבילת האימות שבה יש להשתמש (קרברוס, LM, או NTLM). תהליך ה-LSASS מעביר את הנתונים שקיבל לקבצי DLL רלוונטיים לפי החבילה שמעבדים את הקלט ובמקרה שלנו משווים את התוצאה לתוכן שמופיע בקובץ ה-SAM ומחזירים תשובה אל תהליך ה-LSASS.

שיטות האימות בהן תומך תהליך ה-LSASS בהתקנה סטנדרטית הן:

1. LM, NTLM (MSV_1.0)
2. Kerberos ticket
3. WDigest
4. Terminal Services (TsPkg)
5. PKU2U
6. SCHANNEL

בכל תהליך הזדהות, LSASS פונה לחבילות האבטחה הרלוונטיות (קבצי DLL במערכת) ושומר אצלו את תוצאות האימות של השיטות הללו.

להלן תרשים של כלל הרכיבים המשתתפים בעת תהליך האימות האינטרקטיבי (כאשר המשתמש יושב מול המחשב פיזית ומקליד סיסמה ב-Logon Screen):



[מקור: <https://www.microsoftpressstore.com/articles/article.aspx?p=2228450&seqNum=8>]

בסביבת דומיין כל גיבוי הסימאות נשמרים בשרת המרכזי של סביבת מיקרוסופט, Domain Controller או בקיצור DC. מי שאחראי על תהליך האימות בשרת גם הוא תהליך/שירות LSASS

קישורים להרחבה בנושא:

תהליך הזדהות אינטראקטיבי במערכת:

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa376107\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa376107(v=vs.85).aspx)



המשך תהליכי הזדהות מבוססים על ההזדהות האינטראקטיבית הראשונה:

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa378779\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa378779(v=vs.85).aspx)

תהליך האימות מול חבילות האבטחה:

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa378338\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa378338(v=vs.85).aspx)

סיימנו לפרט את אופן השימוש הסטטי לטווח ארוך וכן את האחסון של הסמאות ופרטי ההזדהות כאשר המערכת רצה. עכשיו נתחיל לדבר על כיצד מועברות הזהויות בפעולות הדורשות תקשורת וגישה בין רכיבי רשת שונים.

שימוש ברשת

יש לא מעט שימושי רשת לפרטי ההזדהות של מיקרוסופט. ברשתות ביתיות אנחנו מכירים בעיקר את שיתוף הקבצים והמדפסות, אבל ברשתות ארגוניות לפרטי ההזדהות יש לא מעט תפקידים נוספים:

- הרצת פקודות מרוחקת - RPC
- גישה גרפית מרוחקת - RDP/Terminal Services
- כניסה למסדי נתונים מסוג MSSQL Server
- גישה מרוחקת למערכת הרישום של מיקרוסופט (Registry)
- אימות לשירותי WEB בשרתי מיקרוסופט IIS (דוגמה לשימוש נפוץ: Sharepoint)
- מערכות צד שלישי רבות שמתממשקות בפרוטוקול LDAP אל ה- Active Directory

התפקוד של כל הרשת מנוהל ממקום מרכזי אחד - זה הכוח (וגם החולשה) של רשתות דומיין.

איך מתבצע תהליך האימות?

האימות הבסיסי נכון לשיטות LM ו-NTLM הראשונות היה פשוט למדי. כדי לא להעביר את הסיסמא ברשת וגם לא את הגיבוב משתמשים בפרוטוקול challenge-response הבא:

1. הלקוח (מבקש השירות) שולח בקשת נתונים לשרת ("השרת" בהקשר זה הוא הגדרה לוגית - יכול להיות שמדובר במחשב, או שירות שפועל על אותו מחשב עצמו)
2. השרת מגיב באתגר - הוא שולח מחרוזת נתונים רנדומלית (NONSE) ומבקש מהלקוח להצפין אותה.
3. הלקוח מקבל את המחרוזת ומצפין אותה באמצעות הגיבוב של המשתמש הנוכחי שהזדהה.
4. הלקוח שולח את המחרוזת המוצפנת יחד עם שם המשתמש של מבקש השירות.
5. השרת בודק האם קיים אצלו משתמש בשם זה ושולף את הגיבוב שקיים אצלו לפרוטוקול.
6. השרת משתמש בגיבוב ששלף כדי להצפין את מחרוזת ה-NONSE ששלח בעצמו ומשווה את התוצאה למחרוזת המוצפנת שקיבל מהלקוח.

7. במידה והתוצאות שוות, סימן שהלקוח מחזיק באותו גיבוב שהשרת מחזיק וכנראה מחזיק גם בסיסמא הנכונה - הלקוח מקבל את התוכן שביקש.

8. חבילת האבטחה יוצרת LOGON SESSION ומעבירה אותו לתהליך ה-LSA

9. תהליך ה-LSA יוצר טוקן שמכיל LUID קיצור של LOGON ID

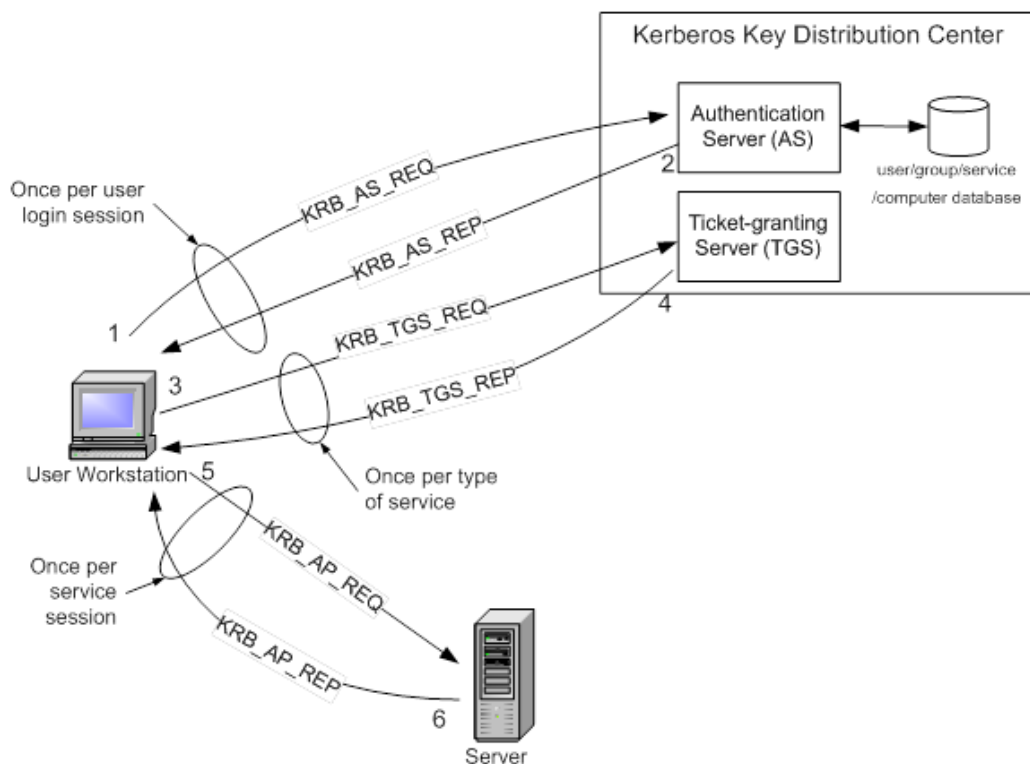
הפרוטוקול החדש יותר מסוג NTLMv2 מוסיף לתהליך הזה שלב אחד קטן בלבד ומאפשר ללקוח להוסיף Nonse משל עצמו, נדון בסיבות לכך בחלק השני שידון בפריצה.

בסביבת דומיין, העניינים מתנהלים קצת אחרת. הפרוטוקול הינו קרברוס. לא נכנס כאן לכל ההסברים של הפרוטוקול משום שזה כמעט מאמר בפני עצמו, רק אקצר ואומר שהפרוטוקול נשען על מספר דברים עקרוניים:

- אימות מול שרת מרכזי שמכיל את פרטי ההזדהות של כל המשתמשים ברשת.
- שימוש בטיקטים בכל תהליך האימות וההתקשורת לאחר ההזדהות
- שימוש בחתימות זמן

תהליך ההזדהות בשלבים:

להלן ציור אשר ממחיש את הצורה שבה עוברת בקשת שירות בין לקוח לשרת בשיטת קרברוס





דוגמאות לשימושים נוספים

קיימים עוד שימושים רבים בשיטות אימות אלו מעבר לשימוש ה"קלאסי" ברשתות מקומיות. מפאת קוצר היריעה (כל נושא הינו פוטנציאל למאמר בסגר גודל של המאמר הנ"ל) לא אפרט על נושאים אלו, אך אתן כותרות וקישורים לשם ההבנה והרחבה למתעניינים:

- אופן פעולה NTLM בגישה לשירותי WEB:

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

<http://www.innovation.ch/personal/ronald/ntlm.html>

[https://technet.microsoft.com/en-us/library/cc778868\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc778868(v=ws.10).aspx)

- הזדהות בפרוטוקול RDP, אופן פעולה:

[http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-](http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/[MS-RDPBCGR].pdf)

[A4F81802D92C/\[MS-RDPBCGR\].pdf](http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/[MS-RDPBCGR].pdf)

כאמור, אלו שתי דוגמאות קיימות, אך יש עוד מספר לא קטן של מנגנונים במערכת ההפעלה (ואף במערכות נוספות) אשר עושות שימוש במנגנוני אימות אלו.

כאן סיימנו את הנושא "האנכי" הראשון.

ועכשיו הגענו לחלק המעניין - אז איך שוברים את כל התהליכים האלו?

שלב האחסון לטווח ארוך

אם יש לנו גישה כלשהי למערכת הקבצים כאשר המערכת כבויה, ניתן לשלוף בקלות את הגיבובים מתוך קובץ ה-SAM המרכזי. בהינתן הרשאות ניהול מקומיות על מערכת ניתן לשחרר את הנעילה של קובץ ה-SAM המרכזי גם בזמן שהמערכת עובדת ולשלוף את הגיבובים. כאמור, סוגי הגיבובים הנשמרים תלויים במערכת ובסביבת הרשת בה היא נמצאת. אבל מה הם עוזרים לנו אם אין אפשרות לבצע פעולה חוזרת הפוכה של תהליך הגיבוב או ההצפנה (במקרה של LM) ולהוציא את הסיסמא?

כדי לקבל את הסיסמא שיצרה את הגיבוב הדרך היחידה המוכרת כיום היא ליצור את כל הגיבובים האפשריים ולהשוות אותם לגיבוב ששלפנו מתוך המערכת. שיטה זו עשויה להיות איטית למדי ונדבר קצת על האופנים השונים של ניצולה.

הדרך הראשונה לתקוף סיסמאות ובמקרה הזה גם גיבובים הוא לנסות סיסמאות קלות לניחוש:

1. מבוסס מילון - שימוש בסיסמאות נפוצות כמו "123456" וכדו'
2. מבוסס מידע אישי - תאריך לידה, מספר זהות, מספר טלפון, שמות ילדים וכדו'
3. ניסוי כל הסיסמאות האפשריות על בסיס טווח תווים הגיוני (למשל אותיות קטנות בלבד או שילוב של אותיות קטנות ומספרים בלבד)

הדרך השנייה היא ניסוי כל הסיסמאות האפשריות ללא שום מגבלה או צמצום טווח האפשרויות.

הערה: שימו לב שהתיאוריה אומרת שקשה למצוא שתי מחרוזות שיצרו את אותו הגיבוב, אך בפועל הדבר אפשרי. כמות האפשרויות שמכסה **מחרוזת התוצאה** של גיבוב NTLM לדוגמא היא 16 בחזקת 32 (מחרוזת באורך 32 תווים כשבכל תו יש 16 אפשרויות - תו הקסדצימאלי כלשהי), שזה בעצם 2 בחזקת 128 אפשרויות. בעוד כמות הסיסמאות האפשריות מצד המשתמש היא לפחות 256 בחזקת 128 (אם ניקח רק את תווי ASCII בלי להתחייח לכל ה-UNICODE), שזה בעצם 2 בחזקת 2,048. כלומר יש לפחות 2 בחזקת 1920 אפשרויות שהן **בוודאות גמורה כפולות של מחרוזות אחרות**. מחקר מעניין שנעשה הציג "התנגשויות" שכאלה בתהליך גיבוב מסוג MD5. ניתן לקרוא על כך במאמר הבא:

<https://eprint.iacr.org/2013/170.pdf>

פיצוח גיבובי LM

כפי שחלקכם כבר הבין לבד, את הגיבובים שנשמרים בשיטת LM ניתן לפצח די בקלות. מדובר בסיסמאות מוגבלות למדי בטווח התווים האפשרי ובנוסף, לפני ביצוע הגיבוב מתבצעת העברה לאותיות גדולות. כלומר שטווח כל הסיסמאות האפשריות שלנו בסך הכול מגיע **לסדר גודל של 64**

ניהול סיסמאות וזהויות ברשתות מיקרוסופט

www.DigitalWhisper.co.il

בחזקת 14 ובוודאי שזה גם מקל על פיצוח מבוסס מילון או מידע אישי. מעבר לכך, הסיסמא אינה באמת באורך של 14 תווים אלא שני חלקים של שבע, מה שאומר שאם נפצח את כל הסיסמאות האפשריות המורכבות מאותיות גדולות מספרים ותווים מיוחדים באורך של שבעה תווים בלבד - נוכל לפתוח כל גיבוב של LM.

עדיין מדובר בלא מעט זמן עבודת עיבוד ומיד נדון בשיטה לקיצור התהליך הזה, אבל קודם בואו נדבר על הגיבובים האחרים.

פיצוח גיבובי NTLM

שיטת NTLM כבר קשה יותר לפיצוח, מדובר בטווח תווים גדול יותר משמעותית ואורך סיסמא כמעט לא מוגבל. למעשה ההגבלה של שיטת הגיבוב עצמה קצרה יותר מכמות הסיסמאות האפשריות בשיטת NTLM (כפי שהוזכר בהערה). עדיין, גם במקרה זה, השימוש במילונים ובמידע אישי עובד לא מעט פעמים ומחזיר אותנו אל הבעיה האמיתית שהיא בחירת סיסמאות נכונה מצד המשתמשים/המערכת. גם בתקיפה בזמן אמת של גיבובי NTLM החישוב עצמו של האלגוריתם אורך מעט יותר זמן ולמעשה מאט משמעותית את קצב התקיפה. שוב, כדי להתמודד עם בעיות אלה מיד נדון בשיטה לפיצוח גיבובים שבה לא נעשה חישוב בזמן אמת.

סוג הגיבוב השלישי שדיברנו עליו הוא... גיבובי סביבת דומיין זמניים שנקראים גם MS-CACHE. כדי לדון בנוסיונות הפריצה לגיבוב הזה הגיע הזמן לדבר קצת על SALT ועל פיצוח בשיטה של עיבוד מוקדם.

פיצוח גיבובים לא מקוון ועיבוד מוקדם

רמת הביצוע של תקיפה בסגנון של ניסיון סיסמאות אפשריות משתנה בהתאם לסוג הגיבוב אותו מנסים לתקוף, בכלי התקיפה שבו משתמשים, באיכות הסיסמא ובמחשב (או מחשבים) שבאמצעותם מתבצעת שיטת הפיצוח. (תודו שאתם מתים על העברית שלי).

בכל המקרים אם נוכל לבצע את התקיפה מול מחרוזת הגיבוב עצמה ולא מול מערכת חיה יהיה לנו הרבה יותר קל. חישוב והשוואת מחרוזות היא פעולה קלה בהרבה מאשר התחברות לשרת לוגי כלשהי וציפייה לתגובה. מלבד זאת קיימות הגנות שמונעות ניסיונות התחברות חוזרים ונשנים. לכן, לאחר שהשגנו את מחרוזת הגיבוב מקובץ ה-SAM, נרצה לייבא אותה אל מחשב או רשת מחשבים שתבצע בשבילנו את עבודת החישוב באופן "לא מקוון" - כלומר ללא ניסיון ממשי להתחבר למערכת.



המהירויות הסטנדרטיות נעות בין נסיונות של כמה אלף סיסמאות לשנייה ועד כ-600 מיליון סיסמאות לשנייה על מחשב בודד עם הכלים הנכונים:

<http://blog.distracted.nl/2009/05/entibr-ntlm-password-brute-forcer.html>

שימוש בשיטת פיצוח סיסמאות מבוצרת יכולה לקחת אותנו קדימה לפי כמות המחשבים, כלומר גודל האוניברסיטה שהשתלטתם עליה או גודל הבוטנט שהצלחתם ליצר לעצמכם באמצעים חוקיים כאלו ואחרים - ועשוי להגיע עד קצבים מטורפים של כמה מאות בליוני סיסמאות לשנייה.

אם ניקח את שיטת הגיבוב הישנה ביותר הקיימת היום - LM ונססה לתקוף מחשב שהרגע השגנו גישה לגיבובים שלו מדובר בכחצי שעת עבודה מפרכת בקצב הזוי של 300 מיליון סיסמאות לשנייה כדי לעבור על כל האופציות ההגיוניות (26 אותיות גדולות + 10 ספרות + כ-14 תווים מיוחדים בשימוש סטנדרטי) לא רע, אבל לא תמיד יש לנו חצי שעה לבזבז ברוב המקרים אין לנו כוח עיבוד מספק בשביל הקצב הנזכר.

ניקח את שיטת NTLM החדשה לדוגמא (תזכורת: LM מופיע רק במערכות XP ומטה) - מדובר בלא מעט זמן, שימוש בסיסמא המורכבת משמונה תווים עם שילוב של אותיות מספרים ותווים מיוחד מביא אותנו לכ-76 בחזקת 8 אפשרויות. גם בקצב המטורף של 300 מיליון סיסמות בשנייה ללא עצירה אנחנו מדברים כאן על כחודש וחצי חישוב.

כדי להתמודד עם הבעיה הזו, במקום לפרוץ את הסיסמאות בזמן התקיפה, אנחנו פורצים אותן לפני.

מה!? איך פורצים סיסמאות לפני שהשגנו את הגיבובים?! פשוט מאד - הפרוטוקול קבוע. אם נבצע עיבוד של כל הסיסמאות האפשריות ונשמור אותן כטקסט קריא, נוכל בקלות בזמן תקיפה להשוות את הגיבוב ששלפנו מהמערכת של הקרבן למסד הנתונים האדיר שלנו עם כל הסיסמאות האפשריות ונבדוק מה הסיסמא שיוצרת את הגיבוב הרלוונטי. השיטה הזו נקראת "עיבוד מקדים" - Precompiled password attack. השיטה למעשה ממירה זמן נפח אחסון. במקום לבזבז זמן בניסיון תקיפה, אנחנו משתמשים בכל הזמן שעומד לרשותנו ומאחסנים את התוצאות - הרבה מאד נפח אחסון אבל חוסך זמן בעת הצורך.

כאשר משווים את הפרס של יכולת שליפת סיסמאות בקלות מכל מערכת מיקרוסופטית קיימת - מדובר בהשקעה משתלמת. אפשר להשקיע חודש וחצי ואפילו שנה כדי לכסות יותר ויותר אפשרויות. למעשה כדי להוכיח את הנקודה הזו והחולשה של הפרוטוקול יש לא מעט שירותים באינטרנט שמחזיקים מאגרים כאלו ומאפשרים שירות בחינם ובתשלום. הנה כמה לדוגמא:

<http://www.hashkiller.co.uk/ntlm-decrypter.aspx>

<http://www.onlinehashcrack.com/list-cracked-hash.php?h=ntlm>

<https://crackstation.net>

בנוסף, קיימות גם קהילות שמשותפות פעולה בהרחבת הטבלאות הקיימות.

ניהול סיסמאות וזהויות ברשתות מיקרוסופט

www.DigitalWhisper.co.il



אתם יכולים לדמיין באיזה נפחי אחסון אסטרונומיים מדובר וכדי להתגבר על בעיית נפח האחסון משתמשים בשיטה שנקראת "טבלאות קשת בענן" (או בלע"ז: Rainbow Tables). ההסבר על אופן הפעולה הוא מחוץ למסגרת של מאמר זה ואתם מוזמנים לפנות ל**כאן ולכאן** כדי להבין איך זה עובד. אפשרות העיבוד המקדים מעמידה את כל שיטת השימוש בגיבובים בסכנה וכדי להימנע מתקיפה זו עושים שינוי קטן בתהליך ההזדהות שנקרא המלחה - מלשון מלח (Salt) ☺.

הבעיה שהמלחה באה לפתור היא שייצוג הגיבובים זהה בכל המערכות בעולם בכל המצבים ולכן יש לנו אפשרות לבצע חישוב מקדים בידיעה שהמערכת הנתקפת תשתמש באותה שיטה בדיוק אותה אנו מכירים. כדי להימנע מהייצוג הקבוע מוסיפים אלמנט רנדומלי שכתוקפים לא נוכל לצפות אותו מראש.

תאוריה - המלחה

בזמן שמירת הגיבוב הראשונית המערכת מייצרת מחרוזת אקראית קצרה שנקראת SALT ומוסיפה אותה לסיסמא בתהליך הגיבוב. המערכת שומרת גם את המלח וגם את הגיבוב הסופי באותו מקום. כאשר משתמש מבקש להזדהות המערכת שולפת את "המלח" מתוך האחסון ומבצעת את תהליך הגיבוב עם אותו "מלח". כל מערכת מייצרת מלח משלה ולכן כתוקפים אנחנו לא יכולים לצפות מראש את הפרוטוקול ולהשתמש בשיטה של עיבוד מקדים.

זה בדיוק מה שקורה עם שמירת הגיבובים של משתמשי דומיין בשיטת MSCACHE. כמו שהזכרנו בחלק הראשון - MSCACHE הוא למעשה גיבוב בשיטת NTLM שחיברו אליו שם משתמש והריצו את תהליך הגיבוב פעם נוספת. השיטה הזו מכריחה אותנו כתוקפים לבצע את הפיצוח **בזמן אמת** רק **לאחר** שהשגנו את "המלח" שהוא שם המשתמש.

נניח ופרצתי למחשב בסביבת דומיין וקיבלתי את ה-Hashים של פרטי חשבונות מסוימים, את LM ו-NTLM אני יכול לחפש בקלות בטבלאות מוכנות מראש. את הגיבובים של MS-CACHE אני צריך לקחת יחד עם שם המשתמש ולבצע תקיפה בזמן אמת. הבחור הנורווגי הנחמד [הזה](#) יעשה לכם עבודה מהירה יותר עם הפריצה הזו, אבל עדיין זה עשוי לקחת המון המון זמן... אז מה עושים? עזבו אתכם שטויות, למה לשבור את הראש על גיבובים שנשמרים בדיסק כאשר הסיסמא נשמרת באופן גלוי...

שלב האחסון בזיכרון

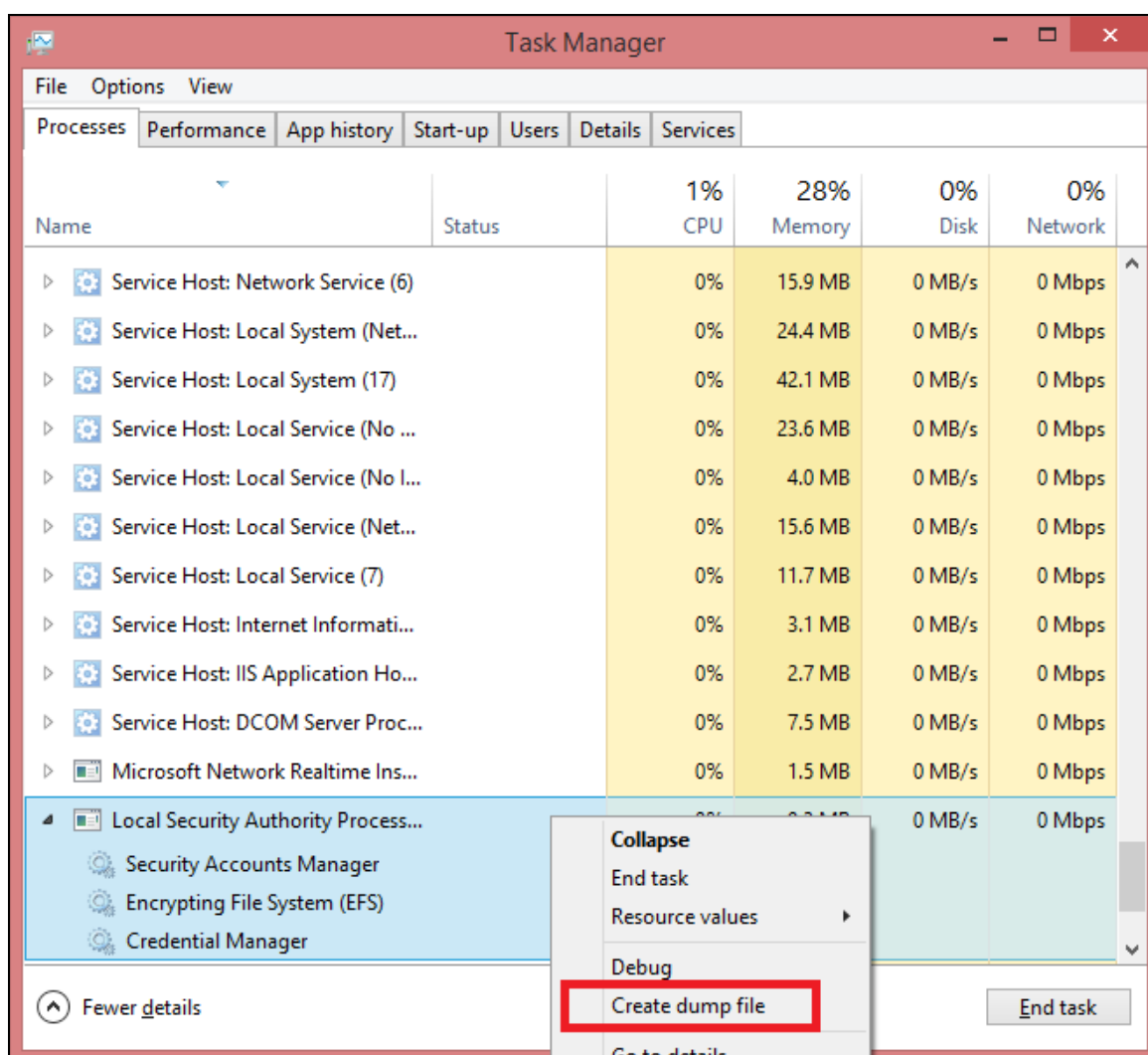
לצערנו (או לשמחתנו), בשלב האחסון בזיכרון יש מעט מאד הגנות. החלק שמעניין אותנו בסיפור הזה הוא שמופיעים גם פרטי אימות זמניים של משתמשים שהתחברו מרחוק מה שמעלה את הסיכויים למצוא פרטי הזדהות של משתמשים לא מקומיים לשרת/תחנת קצה שנפרצה. דוגמא בולטת לכך היא יכולת לשלוף סיסמאות גם של חיבורי RDP מסוג שירותי טרמינל - שירות החיבור המרוחק של מיקרוסופט שמנהלי מערכת עושים בו שימוש רב. בנוסף, נעשה שימוש

ניהול סיסמאות וזהויות ברשתות מיקרוסופט

www.DigitalWhisper.co.il

בחבילת אבטחה של Web Digest, זהו רכיב אימות שמשמש להזדהות מול שרתי ווב בצורה מתוככמת יותר מאימות ווב בסיסי. העניין הוא שחבילה זו זקוקה למחרוזת המשתמש והסיסמא עצמם כדי ליצור את תהליך ההזדהות ולכן שומרת בזיכרון את הסיסמא של המשתמש בצורה גלויה לגמרי... הגישה לזיכרון של תהליך ה-LSASS מצריכה אמנם הרשאות ניהול מקומיות בזמן ריצה (בניגוד לגישה ישירה למערכת הקבצים), מצד שני רוב הפעמים נדרשת גישה בהרשאות ניהול מקומיות - או גישה פיזית כזו שמאפשרת השגת הרשאות ניהול מקומיות במהירות.

כלים מוכרים שמבצעים בשבילכם את התקיפה ומחלצים את המחרוזות הרלוונטיות: [WCE](#) ו-[Mimikatz](#). גרסה עדכנית של [Mimikatz](#) גם מאפשרת שליפה מתוך קובץ DUMP של תהליך ה-LSASS.



ויש לא מעט כלים נוספים שמאפשרים לבצע - לדוגמא בפוסט [הזה](#).

חלק מעניין ומשמעותי נוסף בשלב השימוש בזיכרון זו היכולת לשלוף נתוני קברוס שמאפשרים התחזות לכל משתמש ברשת - ניתן לראות טבלה מסכמת של בנג'מין "דלפי" [בעמוד הזה](#).

ניהול ססמאות וזהויות ברשתות מיקרוסופט

www.DigitalWhisper.co.il

שלב השימוש ברשת

בשלב הזה נתייחס לשני סוגי פרצות:

1. השגת מידע רגיש - גיבובים וסיסמאות
2. גישה ממשית למשאבי רשת

אין לכם עדיין גישה לאף תחנה ברשת ואתם סתם יושבים על הכבל. מה אפשר לעשות? שני הדברים הפשוטים ביותר להבנה וביצוע בשלב זה הם חיקוי של תהליך האימות ונסיונות פריצה של שיחות אימות שעוברות ברשת.

חיקוי תהליך האימות

העברת גיבובים - Pass The Hash

בחלק של "איך זה עובד" הסברנו שלמעשה בפרוטוקול NTLM לא נעשה שימוש מתמיד במשתמש ובסיסמא. בזמן האימות הראשוני המערכת מייצרת גיבוב מסוג NTLM ומשתמשת בו כדי לאמת את זהותה מול מערכות ושירותים שונים. לפי זה, כדי לבצע אימות מול שירותים שונים ברשת אנחנו צריכים רק את הגיבוב ולא את הסיסמא בכלל. בשילוב עם המתקפות משלב האחסון הסטטי והאחסון בזיכרון אנחנו בהחלט יכולים למצוא את גיבוב הסיסמא של המשתמש. בשלב הזה נוכל לבצע אימות מול כל משאב ברשת שהמשתמש שגילינו את הגיבוב שלו מורשה לגשת אליו.

נניח לדוגמא כי בוב מתחבר למחשב באמצעות הסיסמא שלו. הסיסמא מועברת לתהליך ה-LSASS. תהליך ה-LSASS מעביר את המשתמש והסיסמא לכל החבילות שהוא תומך בהן ביניהן MSV1_0 תהליך ה-MSV1_0 מייצר גיבוב מסוג LM ו-NTLM ומעביר אותו בחזרה ל-LSASS, LSASS מחזיק את הנתונים בזיכרון. בניסיון גישה לתיקיית שיתוף במחשב מרוחק המערכת פונה למחשב המרוחק ומקבל אתגר מסוג NTLM. LSASS שולף מהזיכרון את שם המשתמש והגיבוב ומשתמש בהם כדי לענות לאתגר ה-NTLM במידה וקיים במערכת המרוחקת משתמש עם גיבוב זהה, המערכת המרוחקת תאשר את השימוש

הערות צד: שימו לב ששיטה זו תעבוד רק במקרה שהמשתמש קיים במערכת המרוחקת או באחסון לטווח ארוך במערכת הקבצים, בתוך קובץ ה-SAM (משתמש מקומי), או באחסון בזיכרון (משתמש שהתחבר למערכת לאחרונה).

כפי שציינו כבר ברשתות ארגוניות התקשורת בין המחשבים מתנהלת בפרוטוקול קרברוס ולא בשיטת NTLM. בשיטת קרברוס נעשה שימוש בכרטיסים ולא בגיבוב של המשתמש ושיטה זו של העברת הגיבוב (PTH) לא תעבוד. מה שחשוב ומעניין לדעת פה הוא שניתן להגיד למחשב המרוחק כי איננו תומכים בקרברוס ולדרוש ממנה בעצם מעבר (ירידה) לשיטת NTLM הבטוחה פחות. אחת הדרכים הפשוטות



לעשות זאת היא פנייה למחשב המרוחק באמצעות כתובת ה-IP שלו. בצורה זו נעשה שימוש אוטומטי בשיטת NTLM.

דוגמאות למתקפות שניתן לבצע בשלב זה:

המשתף לחלק מן התקיפות הללו נמצא בתהליך התקשורת שלאחר ההזדהות. תהליך ההזדהות ברשת כולל תהליך של אימות "מבקש השירות", אבל לאחר האימות התקשורת מתבצעת באופן "פשוט" ללא שימוש בהצפנה שנקבעה בשלב ההזדהות (כמו בפרוטוקול SSL לדוגמא). בעקבות זאת, אם צלחנו את תהליך האימות בצורה כלשהי נוכל להתחבר אל השרת ללא קושי. לאלו מכם בעלי רקע במערכות ווב, הדבר דומה מאד לגניבת מפתח השיחה או העוגיה של המשתמש. (Cookie or Session ID).

מאפיין נוסף הוא היכולת שלנו כתוקפים לתפוס תקשורת שמכילה גם את האתגר (NONSE) וגם את התשובה המוצפנת של הלקוח (בפרוטוקול NTLM המוכר לנו) ואז אפשר לבצע מתקפת ראש בקיר ולנסות את כל האפשרויות כדי לגלות את מחרוזת הגיבוב ששמשה להצפנת האתגר (NONSE).

SMB Reflection

הבסיס לתקיפה זו הוא שהקרבתנו משמש בו זמנית גם כשרת וגם כלקוח במקביל. בפועל פונים לשרת ומבקשים לקבל גישה למשאבים, כאשר השרת מבקש להזדהות אנחנו **משקפים** לו את האתגר שלו עצמו. "השרת" עונה לנו כלקוח על האתגר שלו עצמו ועכשיו יש לנו תשובה בשבילו. אתם בטח שואלים את עצמכם למה שהשרת יסכים להיות הלקוח שלנו פתאום? אז ככה - שכתוקפים אנחנו מחכים לבקשה ברשת ממחשב כלשהו לגישה בתצורת NTLM. ברגע שאנחנו מזהים בקשה שלא נענית אנחנו מתחזים למחשב שאמור לתת את השירות. עכשיו מתבצע תהליך אימות כפול במקביל, שנינו משמשים בו זמנית גם כשרת וגם כלקוח. המחשב של הקרבן מנסה לקבל מאיתנו שירות (שלמעשה אנחנו רק מתחזים ולא באמת מסוגלים לספק לו) אנחנו כתוקפים מבצעים במקביל תהליך אימות אל המחשב המרוחק שביקש בעצמו לתקשר איתנו.

1. הקרבן (כלקוח) שולח בקשת שירות ואינו מקבל תשובה.
2. התוקף (כשרת) עונה סבבה, אבל רק רגע...
3. התוקף (כלקוח) שולח בקשת שירות לקרבן
4. הקרבן (כשרת) שולח אתגר הזדהות (NTLM).
5. התוקף (כשרת) שולח את אותו אתגר לקרבן (כלקוח).
6. הקרבן (כלקוח) עונה לאתגר ששלח התוקף.
7. התוקף (כלקוח) שולח את התשובה שקיבל מהקרבן (כלקוח) אל הקרבן (כשרת) בערוץ השני.



בלי שאנחנו יודעים את הסימא ובלי שאנחנו יודעים את הגיבוב. פשוט משקפים בחזרה לקרבן את אותו אתגר אימות ששלח לנו ונותנים לו לענות על דרישת ההזדהות של עצמו בשבילנו.

SMB Relay

בשנת 2001, אחד מחברי קבוצת ההאקינג Cult of The Dead Cow בשם Sir Dystic פרסם כלי ומסמך המפרט אודות שיטת תקיפה בשם SMB Relay. המתקפה נועדה לגשת למשאבי רשת על שרת מרוחק באמצעות תקיפת MITM. התוקף דואג לתווך מצב של התחברות בין לקוח לשרת ואז מתחזה ללקוח ו"ממשיך" השיחה בעצמו.

התוקף גורם למשתמש לגשת אל המחשב שלו בבקשת SMB (שיתוף קבצים) באמצעות מייל או דף WEB כלשהו. ברגע שהמשתמש הקרבן מנסה להתחבר אל המחשב של התוקף באמצעות SMB, התוקף מעביר (Relay) את הבקשה אל השרת הרצוי ומציב עצמו בתווך התקשורת בין SMB Client ל-SMB Server. הכלי יודע להעביר את התקשורת בין הצדדים - כמו בכל מתקפת MITM קלאסית - עד לקבלת הרשאה ואז להישאר מחובר לשרת תוך התחזות למשתמש המורשה (אותו עיקרון של בעיית "גניבת קוקי" שהזכרנו במקפת ה-Reflection SMB. כמו כן, הכלי מחלץ מנתוני התקשורת את המחרוזת המוצפנת מה-NONSE ומגיבוב ה-NTLM שנשלח על-ידי הלקוח. את המחרוזת המוצפנת (אל תבלבלו בינה לבין גיבוב רגיל) ניתן לנסות לפצח באופן לא מקוון באמצעות מספר כלים. למידע נוסף אודות מתקפה זו ניתן לקרוא:

<http://www.xfocus.net/articles/200305/smbrelay.html>

<https://en.wikipedia.org/wiki/SMBRelay>

Responder

אגב חילוץ נתוני תקשורת ופיצוח אתגרי NTLM לקבלת סימא מהסעיף הקודם, אי אפשר להתעלם מכלי התקיפה החמוד הזה. [Responder](#) יושב על הרשת בשקט יחסי ומנסה למשוך אליו תקשורת מסוג LLMNR, NS-NBT או MDNS שאינה מקבלת מענה, הוא מסוגל להתחזות לשירותים שונים כמו SMTP, FTP, SQL ועוד ולחכות שהקרבן פשוט ישלח אליו את הסימאאות. ברשתות מתחם כפי שכבר הסברנו האימות ברשת מתבצע באמצעות קרברוס שלא מאפשר פיצוח של תשובת האתגר בפשטות כמו בפרוטוקול ה-NTLM.

ריספונדר מאזין לרשת ולבקשות שונות, הוא מכריח את הלקוח/קרבן להתחבר אליו ישירות דרך כתובת ה-IP. מסיבה לא ברורה, מערכות מיקרוסופט משנמכות לאימות מסוג NTLM במקום קרברוס כאשר הן פונות לשירותי רשת באמצעות כתובת IP (כנראה מתוך הנחה שמחשבים במתחם מיקרוסופט יחזיקו שם מחשב נורמאלי בפרוטוקול NBNS או DNS). ברגע שהלקוח מוכן לעבוד ב-NTLM, שוב אנחנו שולחים לו אתגר ואת התשובה מנסים לפצח תוך שימוש בכל שיטות ניחוש הסימאאות המוכרות לנו.

SMB Replay

בשנת 2010 שני חוקרי אבטחה בשם Agustin Azubel ו-Hernan Ochoa מחברת AmpliaSecurity פרסמו במסגרת הכנס BlackHat הרצאה על מחקר שעשו אודות מתקפות שונות בפרוטוקול SMB של חברת מיקרוסופט וספציפית על כשלים במימוש ה-NTLM. במסגרת המחקר שלהם הציגו כשל ספציפי במנגנון הרנדומיזציה שנכתב לטובת ייצור ה-Nonce. הבעיה היא שתוך מספר בקשות עשוי לחזור שימוש באותו ה-NONSE.

במתקפה זו על התוקף לצותת לתקשורת ולאסוף ממנה כמה שיותר זוגות של ה-Challenge וה-Response שנשלחו בין הלקוח לבין השרת. לאחר מכן יוזם התוקף מספר תהליכי הזדהות בעצמו אל השרת עד אשר מתקבל אתגר זהה לזה שחילצנו בזמן ההסנפה. על Challenge כזה התוקף כבר יודע לענות, הוא מחזיק תשובה שלו שמורה משלב ההאזנה.

לקריאה מלאה של המאמר הנ"ל, המפרט מעבר למתקפה זו מתקפות רבות על הפרוטוקול, ניתן לקרוא בקישור הבא:

<http://www.ampliasecurity.com/research/NTLMWeakNonce-bh2010-usa-ampliasecurity.pdf>

שילוב של תקיפות אלו עם מתקפה כגון NBNS Spoofing יכול להיות קטלני. מאמר המפרט על אופן התקיפה הנ"ל פורסם בגיליון ה-32 של המגזין על-ידי אפיק קסטיאל וניתן לקרוא עליה בקישור הבא:

<http://www.digitalwhisper.co.il/files/Zines/0x20/DW32-1-NBNSspoofing.pdf>

קיימת מתקפה נוספת בשם Pass The Ticket עליה לא ארחיב במאמר זה, אך היא מעניינת ביותר ומומלץ לקרוא עליה בקישור זה:

<http://blog.gentilkiwi.com/secuirite/mimikatz/pass-the-ticket-kerberos>

מעין סיכום לנושא ה-SMB

רעיון האימות של NTLM מתבסס על ידיעת האתגר שנשלח מהמחשב המרוחק וידיעת הסיסמא שמשמשת ליצירת הגיבוב. בפועל, הגיבוב מספיק לנו ובהאזנה לרשת נוכל לראות אתגרי NTLM עוברים גלויים וחוזרים מוצפנים. ניתן לקחת את האתגר ולבצע ניסיונות פיצוח בשיטות ניחוש הסיסמאות השונות (ידע מוקדם, מילון ומתקפת ראש בקיר). חישוב הגיבוב של סיסמא אפשרות + הצפנה באמצעות האתגר ואז השוואת התוצאה לאתגר המוצפן שזיהינו ברשת.

אך גם בלי שנצטרך לבצע MITM ו/או האזנה מלאה לרשת ניתן לבצע ניסיונות לפיצוח סיסמאות. כמו שתיארנו בחלק הקודם של SMB Reflection - ניתן לחכות ל-"בקשה יתומה" לשירות NTLM ואז להתחזות לתחנה שמעניקה את השירות ולהגיש אתגר חוזר. המחשב שמבקש את השירות יחזיר לנו את האתגר המוצפן. עכשיו יש לנו את המחרוזת הרנדומאלית של האתגר וגם את האתגר שמוצפן בעזרת הגיבוב. שוב, ניתן לבצע ניסיונות פיצוח אופליין בלא "להרעיש" ברשת.

סיכום נושא תקיפה

עד כאן דיברנו על שיטות התקיפה השונות, נגענו בלא מעט שיטות, אך קיימות עוד הרבה. עם זאת, נעזור כאן על מנת להתחיל לדבר על החלק השלישי - אם וכיצד ניתן בכל זאת לאבטח את הרשת שלנו מפני מתקפות אלו?

שיטות הגנה

עד כה דיברנו על הנושא בכלליות והצגנו את הנושא מנקודת מבטו של התוקף, כעת נשנה את זווית הבחינה שלנו ונראה כיצד מנהל רשת וצוות ה-IT יכולים למנוע או להקשות על מתקפות כגון אלו להתממש. מלבד חילוץ הסיסמאות ClearText בעזרת כלים כגון Mimikatz אשר דורשים גישה ישירה לזיכרון של התהליך, או שימוש ישיר בשיטות PassTheHash / PassTheTicket, מלבד אלו, על מנת להשיג את הסיסמאות המקוריות של המשתמש עלינו לנסות לשבור אותם בעזרת ניחוש. גם אם יש וגם אם אין Salt, הסיכויים שלנו להצליח במשימה זו תלויים ישירות בחוזק של הסיסמה, ועל כן שימוש בסיסמאות חזקות (הגדרה לא פשוטה כל כך) בהחלט משפר את הסיכויים שלנו בתור משתמש. ישנן לא מעט המלצות על שימוש נכון בסיסמאות, לא ארחיב עליהן כאן, אולם יש מספר קונספטים שחשוב לזכור אותם. השתמשו תמיד בסיסמאות שקל לזכור אך קשה לנחש, דוגמא טובה לכך יש בקומיקס המוכר הבא:

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

[במקור: <https://xkcd.com/936>]

ניהול סיסמאות וזהויות ברשתות מיקרוסופט

www.DigitalWhisper.co.il



זו נקודה משמעותית מאד בעיניי שפותרת חלק נכבד מהבעיות עם סיסמאות בכלל. מעבר לכך, כדאי לחשוב גם על דרך לניהול הסיסמאות המרובות, אם זה בהטמעה של תוכנה לניהול סיסמאות ואם זה בהדרכה של משתמשים איך לנהל את הסיסמאות שלהם בצורה יעילה. נרחיב כעת על אמצעים טכנולוגיים לפתרון בעיות אלו.

רגע קצר לפני כן, עוד דבר שחשוב לזכור הוא שבסופו של דבר, הבעיה האמיתית בפן הטכנולוגי (ולא האנושי) היא בעיה מהותית. זו דוגמא קלאסית בעיניי לכשל ברמת התכנון והיעיצוב של המערכת. פרוטוקולי השיחה וההזדהות שמשתמשת בהם מיקרוסופט (לצרכי תמיכה לאחור) הם ישנים על גבול העתיקים שלא הושקעה בהם אותה מחשבה ותכנון מונחה הגנת מידע שיש היום בשוק. כל הפתרונות המיושמים הם טלאים על דלי מלא חורים, בסופו של דבר מיקרוסופט תצטרך לצאת עם חבילת אבטחה חדשה שתסיר כל תמיכה לאחור בפרוטוקולים בעייתיים כמו LM ו-NLTM.

הגנה על המידע השמור בכונן:

את הגיבובים יש לשמור לטווח הארוך, קשה להמנע מזה, כדי להפוך את האפשרות הזו למאובטחת יותר כדאי להתרכז בסיסמאות קשות לניחוש - (ארוכות, שימוש בתווים לא סטנדרטיים כמו עברית). בנוסף, ניתן לבטל את השימוש באלגוריתם LM, אם אנו לא צפויים להתקל במערכות הפעלה הישנות מ-2000, ממש אין צורך באלגוריתם ה"ל". על מנת לבטל אותו פשוט נוסף בעורך הרישום את המפתח:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa
```

ובו את:

```
NoLMHash
```

למידע נוסף בעניין:

<https://support.microsoft.com/en-us/kb/299656>

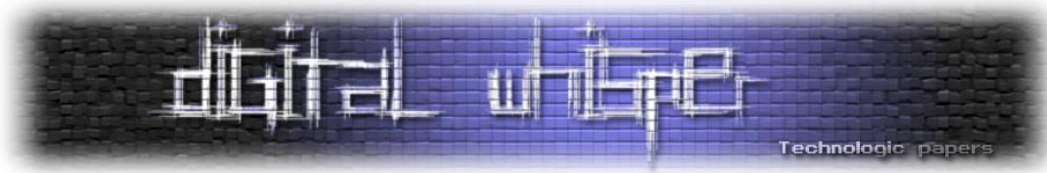
ביטול משתמשים מקומיים - גיבובים של MSCACHE עוברים "המלחה" באמצעות שם המשתמש. תקיפות על גיבובים מסוג MSCACHE מסתמכות לרוב על שמות משתמשים צפויים כמו USER או ADMINISTRATOR. החלפת שמות משתמשים מובנים כאלה ממזערת את החשיפה לשימוש בעיבוד מקדים כדי לחשוף את הסיסמאות.

כמו כן ברשתות דומיין של מיקרוסופט אין סיבה אמיתית להחזיק משתמשים מקומיים פעילים ומומלץ לאחר החלפת השם גם להכניס אותם למצב Disabled. ניתן גם לבטל שימוש ב-MSCACHE לטווח ארוך:

<http://support.microsoft.com/kb/172931>

<http://www.ampliasecurity.com/research/wcefaq.html#thisisnotcachedump>

ל עוד התקנה של [עדכון MS 2871997](#) כנגד PTH אשר מבטל הרשאות רבות למשתמשים מקומיים בסביבת דומיין.



הגנה על המידע אשר נמצא בזיכרון:

ביטול חבילות אבטחה לא נדרשות כמו NTLM בסביבת דומיין, WDIGEST בכלל וכו'. שדרוג מערכות הפעלה לגרסת 8.1.

בזמן שימוש במרחב הרשת:

במידה והרשת מאפשרת זאת, נוכל לאפשר הזדהות לתחנה אך ורק באמצעות NTLMv2, נעשה זאת ע"י שינוי הערך הבא בעורך הרישום ל-5:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Imcompatibilitylevel

ניתן לעשות זאת גם באמצעות GPO:

Computer Configuration -> Windows Settings -> Security Settings -> Local Policies -> Security

Options -> Network Security -> LAN Manager Authentication Level



סיכום

במאמר זה הצגתי את רוב שיטות האחסון וההזדהות הנהוגות היום במייקרוסופט. דיברנו על שלושה מחזורי חיים של הסיסמא, על הכוון, בזיכרון ובמרחב הרשת והצגתי דרכים שונות לתקוף את המבנים והפרוטוקולים השונים הללו כמו דרכים להגן מפני פרצות מסוג זה.

תודות

מעבר לתודות שנתנו בתחילת המאמר, ברצוני להעניק הקדשה מיוחדת אחרונה וחביבה לתלמידי שלב כלשהו אדר התשע"ה, שהם וגם אני רצינו מאד לעסוק יחד בנושאים אלו אך הזמן הניף את חרבו הקצרה כדי להכריע סופית את הקרב האלמותי בין סופרמן לבאטמן, בשבילכם: אילן, רועי, גל, אמיתי, עמרי, אסף, מיכל, הילה, אור, אלירן, טל, מור, אורן, אפי, יוסי ותמר. הייתם אחלה תלמידים.

עוד תודה מיוחדת שמורה לעורכי המגזין שבזכותם יש לכולנו תוכן איכותי באמת בצורה נגישה ונוחה כמו שלא ראיתי בשום מקום אחר. שאפו. והם גם עורכים מעולים ומקצועיים.

ביבלוגרפיה ומקורות להרחבה

[Wikipedia](#) - suit yourself.

בנושא LSA:

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa378326\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa378326(v=vs.85).aspx)

בנושא NTLM:

<https://www.youtube.com/watch?v=fyk-0rub6Kw>

<https://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-grutmacher.pdf>

בנושא PTH:

<https://media.blackhat.com/us-13/US-13-Duckwall-Pass-the-Hash-WP.pdf>

<http://www.harmj0y.net/blog/penetesting/pass-the-hash-is-dead-long-live-pass-the-hash/>

בנושא סיסמאות בכלל:

[https://technet.microsoft.com/library/hh994558\(v=ws.10\).aspx](https://technet.microsoft.com/library/hh994558(v=ws.10).aspx)

<https://technet.microsoft.com/en-us/library/hh994565.aspx>

בנושא מטמון סיסמאות מתחם:

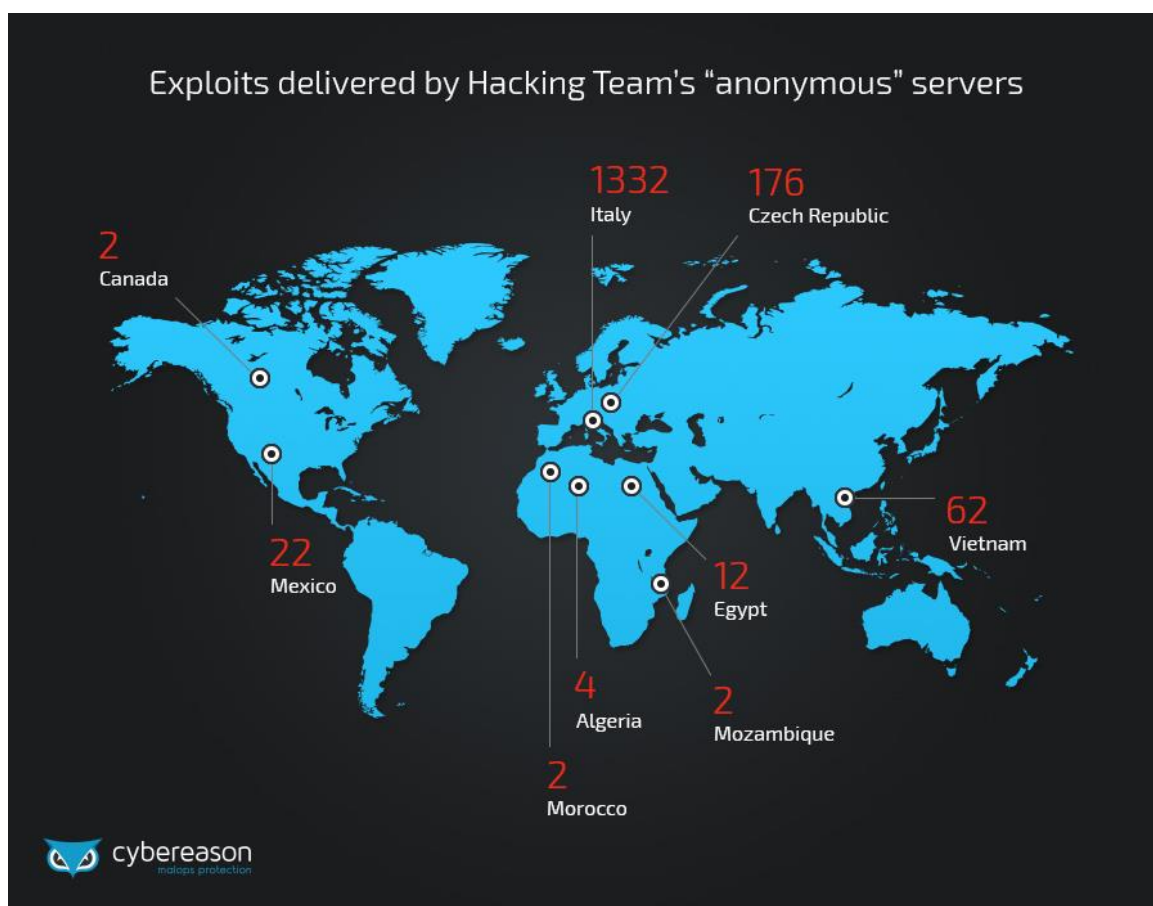
<http://webstersprodigy.net/2014/02/03/mscash-hash-primer-for-pentesters>

זליגת שיטות התקיפה של HackingTeam: שידרוג מיידני לכל האקר מתחיל

מאת איאן מילר, עמית סרפר ואלכס פרייזר

רקע

במאמר זה נציג ניתוח אשר בוצע על-ידי אנשי קבוצת המחקר של Cybereason על שיטות וכלי התקיפה של HackingTeam, המציע יכולות מיסוך, התחזות ותקיפה מתוחכמות, הזמינות לכל דורש.



לאור הפרסומים על תקיפת הסייבר על חברת HackingTeam וזליגת הידע של החברה לאינטרנט, קבוצת החוקרים שלנו החליטה לחקור לעומק ולגלות את שיטות התקיפה שעמדו לרשות אנשי החברה.

שידרוג מיידני לכל האקר מתחיל: HackingTeam זליגת שיטות התקיפה של קבוצת

www.DigitalWhisper.co.il



לחוקרי אבטחת מידע, מידע כזה הוא מכרה זהב, המעניק אשנב לשיטות פעולה של האקרים ולדרך המאפשרת להם לקיים את התקיפות לאורך זמן. שניים מבין חוקרי אבטחת המידע של חברת Cybereason גילו במאגר המידע שנפרץ פרטים על פעילויות הקבוצה ויעדי התקיפה שלה.

הזמינות הגבוהה של המידע עשויה להעצים את יכולותיהם של האקרים מרחבי העולם, ולשים בידיהם כלים ושיטות עבודה מתוחכמים יותר להוצאה לפועל של תקיפות הסייבר עליהם הם עומלים, כמו גם חולשות Zero-day חדשות ששוחררו מבלי שניתנה לחברות הרלוונטיות השהות לתקן אותן. על אף שהתקיפות החדשות שיצאו לפועל בחסות המידע שנלמד מ-HackingTeam יהיו בעלות חתימה שונה ממצעי HackingTeam, אנו מעריכים כי מכיוון שאנשי החברה השאירו על שרת ההדבקה שלהם קוד קל לקריאה והערות שימוש מפורטות, התוקפים יעקבו אחרי הוראות אלו בדיוק רב, דבר שעשוי לאפשר לאנשי אבטחת מידע לפתח יכולות זיהוי שלהם בעתיד.

ברצוננו לבחון מקרוב כיצד אנשי HackingTeam כיוונו את תקיפותיהם אל מטרותיהם ואת השיטות בהם השתמשו כדי לשמר את אחיזתם ביעד המטרה לאורך זמן ממושך.

חיקוי שיטת התקיפה של Flame בניסיון להסתיר את מקור התקיפה

קבוצת HackingTeam השתמשה באסטרטגיה חכמה על מנת לחדור למחשב היעד. ראשית, מבצעי החברה חיקו את פעילות התוכנה הזדונית Flame אשר נחשפה ב-2012. Flame התחבר לשרת ה-C&C (שרת פיקוד ובקרה) באמצעות ממשק משתמש אשר נראה כמו אתר חדשות או שירות של adwords, אשר הציע לכאורה ל"לקוחות" (אנשי HackingTeam השתמשו במושג זה ככינוי למטרות שלהם) לינק לשרת "איחסון פרסומות", אשר לחיצה עליו גרמה להתקנה של התוכנה הזדונית. רבות מהפקודות והפרוטוקולים בהם נעשה שימוש בתקיפות "Flame" השתמשו בז'רגון מעולם החדשות והפרסום על מנת להתל בכלי זיהוי ובאנליסטים, וקבוצת ה-HackingTeam השתמשה באותה אסטרטגיה.

```

#z5###A:[root@htcnc data]# ls -h
a_jax-loader.gif          content.swf_ie          index.html              privesc_filter.py
chrome_non_chrome_filter.py customerkey.js          news                    xp_filter.py
content.swf_chrome        empty.swf               platform.swf
[root@htcnc data]# ls -hl
total 1.6M
-rw-r--r-- 1 1000 1000 2.6K Jun 28 13:17 a_jax-loader.gif
-rwxr-xr-x 1 1000 1000 671 Jun 28 13:17 chrome_non_chrome_filter.py
-rw-r--r-- 1 1000 1000 40K Jun 28 13:17 content.swf_chrome
-rw-r--r-- 1 1000 1000 11K Jun 28 13:17 content.swf_ie
-rw-r--r-- 1 1000 1000 55 Jun 28 13:17 customerkey.js
-rw-r--r-- 1 1000 1000 562 Jun 28 13:17 empty.swf
-rw-r--r-- 1 1000 1000 924 Jun 28 13:17 index.html
-rw-r--r-- 1 1000 1000 1.5M Jun 28 13:17 news
-rw-r--r-- 1 1000 1000 26K Jun 28 13:17 platform.swf
-rwxr-xr-x 1 1000 1000 894 Jun 28 13:17 privesc_filter.py
-rwxr-xr-x 1 1000 1000 613 Jun 28 13:17 xp_filter.py
[root@htcnc data]# cat customerkey.js
affiliate=adwords&customerid=YmlzTmxPd0x2NFNWUlpBRQ==
[root@htcnc data]#

```

[בתמונה - שימו לב למילים "news" ו-"adwords" בקוד ובשמות הקבצים]

שידרוג מיידית לכל האקר מתחיל: HackingTeam זליגת שיטות התקיפה של קבוצת

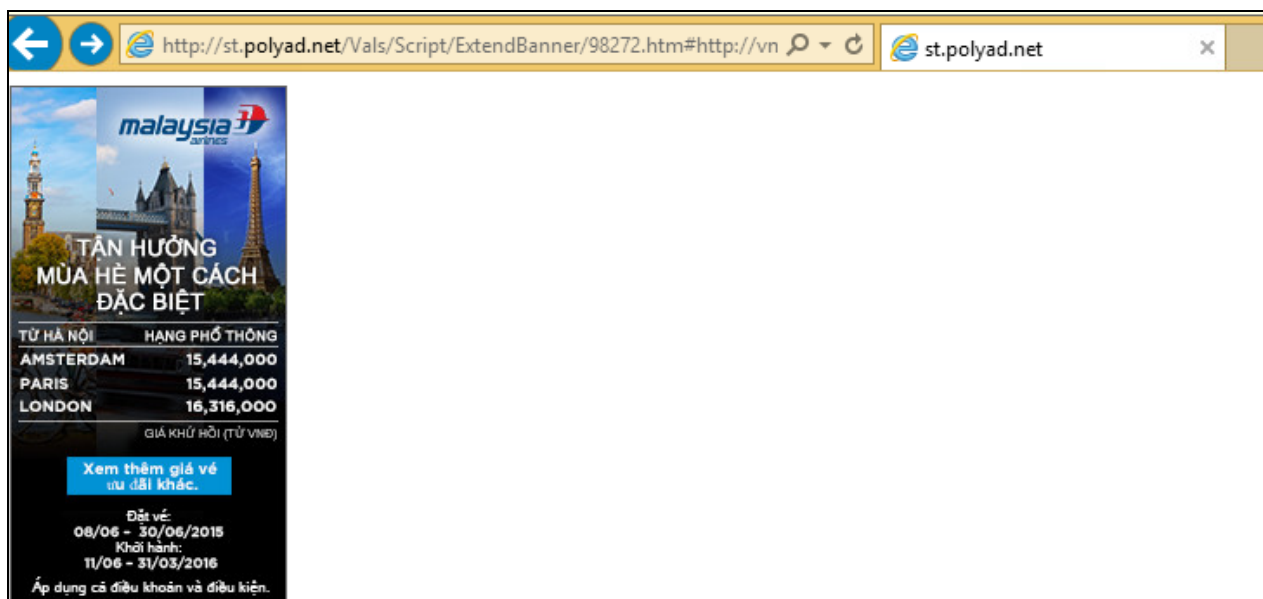
www.DigitalWhisper.co.il

שיטת ההדבקה

על שרת ההדבקה של ה-HackingTeam מצאנו קובץ בינארי מסקרומבל ב-base64 עם השם "חדשות" ("news"), שגילינו שהוא למעשה היה ה-Payload. כאשר פיענחנו את קובץ ה-base64 התגלה בפנינו קובץ מוצפן ב-AES-256 המכיל Oday שמבצע Privilege escalation ל-System באמצעות ניצול חולשה ב- Driver של Adobe.

על ידי שימוש במגוון שיטות מקובלות, ביניהן פשינג והנדסה חברתית קיבלו המטרות לינק. ברגע שמקבל הלינק לחץ עליו, שרת ההדבקה בדק האם זהות המותקף נכונה. באם לא - המקליק הופנה ישירות לעמוד שגיאה 404 או לעמוד בית כלשהו הקשור לחדשות (וניתן להתאמה לפי הלקוח) על מנת שלא לעורר חשד. לעומת זאת, אם המקליק היה אכן היעד הנכון לתקיפה - השרת המשיך לאבחן את המחשב ממנו התחבר על מנת לזהות את מערכת ההפעלה והדפדפן בהם נעשה שימוש. השרת זיהה האם המטרה עושה שימוש ב-Internet Explorer, Firefox או Chrome, ואיזו מערכת הפעלה רצה על המחשב, על מנת להתאים את השימוש בחולשת Adobe Flash מתאימה אשר תאפשר לתוקף להשתלט על מחשב היעד.

מנקודה זו מערכת השליטה מרחוק הפכה מותקנת ופעילה על המחשב ואיפשרה לתוקפים להתקדם לצעד הבא במבצע התקיפה שלהם.



[צילום המסך לעיל מראה דוגמא של תקיפה מוכוונת ליעד בויטנאם אשר שולחת לפרסומת משתמש אשר לא זוהה כיעד לתקיפה ומשתמש ב-

[Internet Explorer

הצלחנו לעקוב אחרי התהליך הנ"ל באמצעות קריאת קוד המקור (המתועד היטב!) של הקבצים על שרת ההדבקה ושל הלוגים של תקשורת ה"לקוח". כשהתעמקנו עוד יותר במידע, יכולנו לראות מתי חדרו אנשי HackingTeam למטרה (עד לכדי רמת דיוק של מיקרו שניות), איפה הם היו ממוקמים, באיזה ספק שירותי

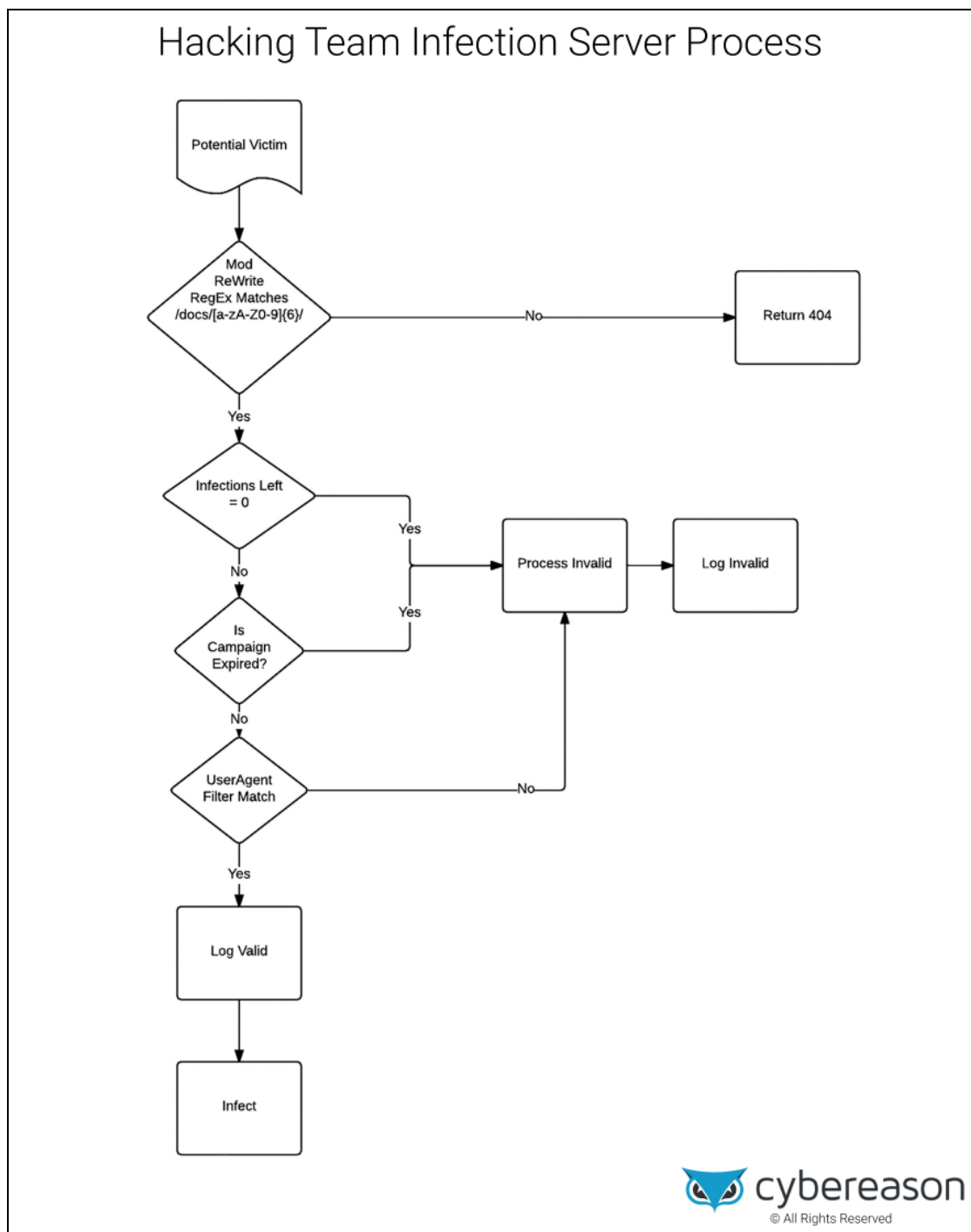
שידרוג מיידית לכל האקר מתחיל: HackingTeam זליגת שיטות התקיפה של קבוצת

www.DigitalWhisper.co.il

דומיין החדשות - ככל הנראה כתוצאה של מספר הקבוצות שכעת מורידות, מריצות ועורכות את הקוד בעצמן.

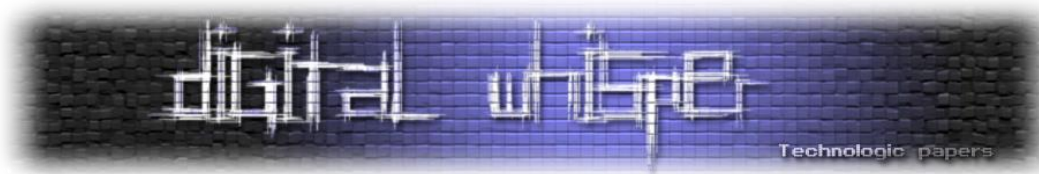
שרת ההדבקה: מותאם למטרה ובר תוקף

על מנת להבין את תהליך התקיפה של HackingTeam בחנו את פעילות שרת ההדבקה. להלן תרשים זרימה המתאר את התהליך:

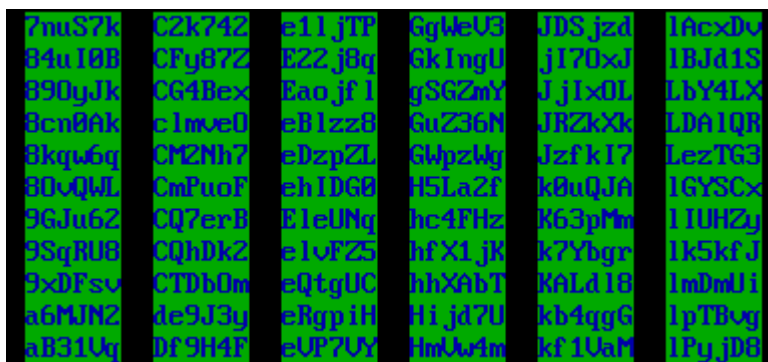


שידורג מידי לכל האקר מתחיל: HackingTeam זליגת שיטות התקיפה של קבוצת

www.DigitalWhisper.co.il

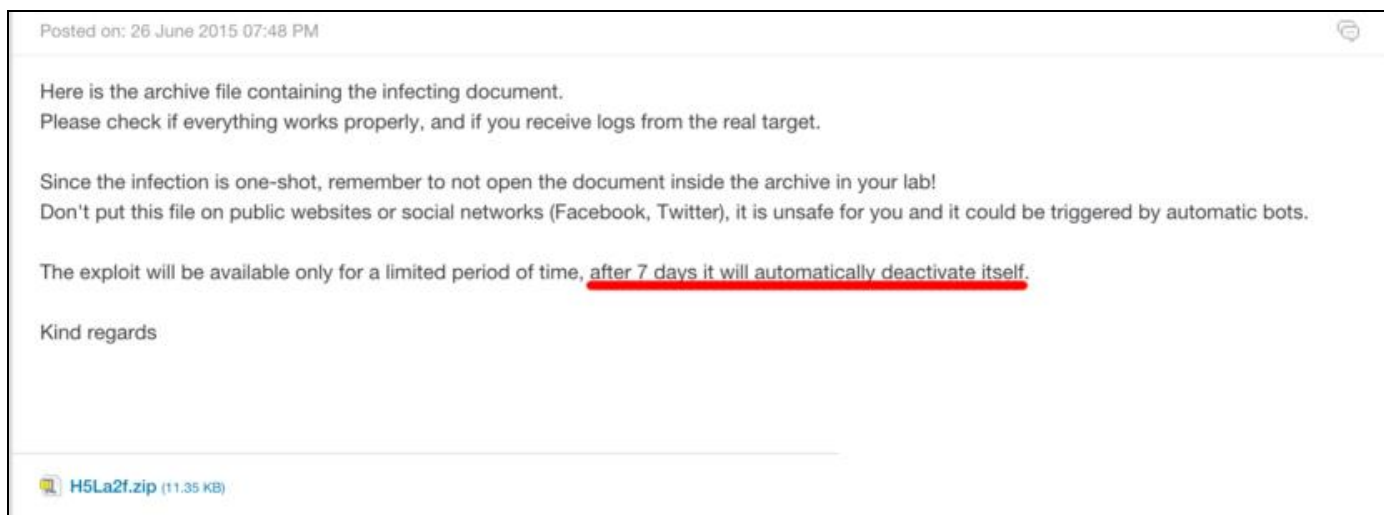


התרשים מתאר את התהליך הבא: בתחילה, השרת מעביר את המבקר לדומיין המודבק דרך Regular Expression של Mod_rewrite, על שרת ה-Apache על מנת לבדוק התאמה בין מזהה הקמפיין בן ששת התווים לבין ערכת ה-exploit ול-Payload בקובץ /var/www/files/campaignID. כאשר לא קיימת התאמה בין מזהה הקמפיין, השרת מוביל את המבקר ישירות לעמוד שגיאה 404. באם קיימת התאמה, התוכנה מתקדמת לשלב השני.



[דוגמה של אוסף מזהי קמפיין בין שישה תווים לתקיפות מבוססות מערכת הפעלה Windows]

בשלב השני, התוכנה בודקת את מונה הכניסות לקמפיין הספציפי על מנת לוודא שהוא עומד על אפס, דבר המעיד על כך שאיש עדיין לא הודבק על ידי הקמפיין הנוכחי. בנוסף, התוכנה בודקת את תאריך התפוגה של הקמפיין לוודא שהוא בר תוקף. עד כה כל הקמפיינים של הקבוצה אותם בחנו הכילו תאריך תפוגה אחיד בן שבוע מיום יצירת הקמפיין.



[צילום מסך של אימייל משירות לקוחות של HackingTeam המדגיש את תאריך התפוגה בן השבוע של שרת ההדבקה]

שידרוג מיידית לכל האקר מתחיל: HackingTeam זליגת שיטות התקיפה של קבוצת

www.DigitalWhisper.co.il

```
[general]
expiry=1435912656 Campaign Expiration Date - Fri, 03 Jul 2015 08:37:36 GMT
hits=0
pos=first

[valid]
type=data
headers[Content-Type]=text/html
headers[Cache-Control]=no-cache, no-store, must-revalidate
headers[Pragma]=no-cache
headers[Expires]=0
path= ./index.html
visitdate=1435488470054 Last Valid Target - Sun, 28 Jun 2015 10:47:50 GMT
visitaddress= Vietnamese IP Address
visitagent=Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.130 Safari/537.36

[invalid]
type=301
headers[Content-Type]=application/octet-stream
headers[Cache-Control]=no-cache, no-store, must-revalidate
headers[Pragma]=no-cache
headers[Expires]=0
headers[Location]=http://st.polyad.net/Vals/Script/ExtendBanner/98272.htm#http://vnexpress.net/&pos=BigLogo5&link=&otherlink=
visitdate=1435488501132
visitaddress= Vietnamese IP Address
visitagent=Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.130 Safari/537.36

[filters]
platform=/win/i
browser=/^(IE|Chrome|Firefox)$/

[related]
platform.swf+=2min
content.swf+=2min
customerkey.β+=2min
```

[דוגמא של קוד התיקוף של שרת ההדבקה - מתוך קמפיין תקיפה של יעד בוטנאם]

אם מונה הכניסות ותאריך התפוגה תקפים, התוכנה משווה את ה-user-agent בדפדפן של הנתקף בעזרת שימוש בספריית PHP בשם BrowseCap אשר מותקנת על שרת ההדבקה, על מנת להבטיח שהמחשב המותקף עומד בדרישות הקמפיין. לדוגמא, ראינו מקרה בו התוכנה בדקה האם מותקנים על מחשב המטרה מערכת הפעלה Windows7, ודפדפן כרום גרסה 43.0.2357.130.

עוד פריט מידע מעניין שגילינו הוא סקריפט Python בשם xp_filter.py. הסקריפט בודק את מערכת ההפעלה של הקורבן על מנת לקבוע באם היא מריצה Windows XP. במקרה והמערכת אינה ווינדוס XP, שרת ההדבקה יריץ חולשה שאינה מבוססת Windows XP. ובאם המערכת מבוססת Windows XP היא תגיש קובץ SWF מדומה: empty.swf.

```
#!/usr/bin/env python

import os
import sys
import struct

def main():
    platform = os.environ.get('_BROWSCAP_platform')
    sys.stderr.write(platform)

    target_dir = os.path.dirname(os.path.realpath(__file__))
    if platform.lower().find('xp') == -1:
        # not xp, serve the exploit
        sys.stderr.write('\nnot xp')
        sys.stdout.write(open(os.path.join(target_dir, 'platform.swf')).read())
    else:
        # xp, serve fake swf
        sys.stderr.write('\nxp')
        sys.stdout.write(open(os.path.join(target_dir, 'empty.swf')).read())

if __name__ == '__main__':
    main()
```

[סקריפט סינון XP : ההערות נכתבו ככל הנראה על ידי גורם חיצוני, ממנו קנו אנשי HackingTeam את ה-exploit]

שידרוג מיידי לכל האקר מתחיל: HackingTeam זליגת שיטות התקיפה של קבוצת

www.DigitalWhisper.co.il

המשך ההדבקה: השגת System Privelege ושליטה מרחוק

בשלב הבא הסקריפט מעתיק את התוכן של Payload ה"חדשות" אל STDOUT, על מנת לשלוח את ה-Payload דרך שרת הווב ומשם להעביר אותו למטרה. ה-Payload הוא למעשה אותו קוד base64 מוצפן עליו התייחסנו מוקדם יותר במאמר, המכיל את רכיב ה-RCS (רכיב השליטה מרחוק) ואת חולשה ה-privilege escalation.

עתה, יש לתוקפים יכולת הרצת Shellcode על מחשב הנתקף. ה-shellcode מריץ את חולשת ה-privilege escalation כדי לקבל הרשאות SYSTEM. לאחר מכן, מורד מהשרת הקובץ Agent.exe שהוא למעשה ה"רושעה" עצמה של HackingTeam - הלקוח של מערכת ה-RCS. חברת Trend Micro סקרה את חולשת ה-privilege escalation במאמר שבקישור הבא:

<http://blog.trendmicro.com/trendlabs-security-intelligence/a-look-at-the-open-type-font-manager-vulnerability-from-the-hacking-team-leak/>

בנוסף לשרת ההדבקה הנ"ל אשר תוקף מערכות מבוססות Windows, ל-HackingTeam היו גם שרתי הדבקה אשר יועדו לתקוף מערכות מבוססות אנדרואיד, אשר השתמשו בטכניקות דומות מבלי לעשות שימוש בחולשת Flash אלא בחולשות במערכת Android.

```
#!/usr/bin/env python
import os
import sys
import struct

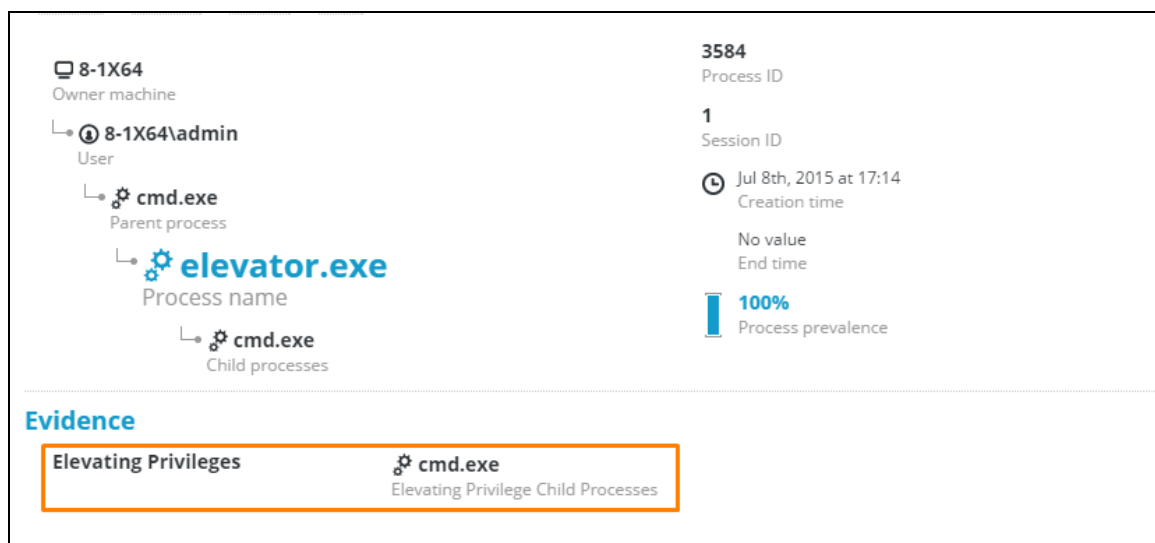
def main():
    browser = os.environ.get('_BROWSCAP_browser')
    sys.stderr.write('[*] Browser {}'.format(browser))
    target_dir = os.path.dirname(os.path.realpath(__file__))
    privesc = open(os.path.join(target_dir, 'news')).read()
    if 'IE' not in browser:
        article_number = os.environ.get('_REQUEST_article')
        if int(article_number) == 61441:
            sys.stdout.write(open(os.path.join(target_dir, 'news')).read())
            sys.stdout.write(privesc)
            sys.stdout.flush()
            sys.stderr.write('[*]..server')
        sys.stderr.write('[*] Chrome/FF News {}'.format(article_number))
    else:
        sys.stdout.write(privesc)
        sys.stdout.flush()
        sys.stderr.write('[*] IE len {}'.format(len(privesc)))
```

[סקריפט ה-privilege escalation והתקנת ה-Payload]

שידורג מיידית לכל האקר מתחיל: HackingTeam זליגת שיטות התקיפה של קבוצת

www.DigitalWhisper.co.il

כהערת שוליים ברצוננו לציין כי המערכת של Cybereason זיהתה באופן מיידי את השימוש בחולשת ה-
 privilege escalation כבר עם הניסוי הראשון שלנו של המערכת במעבדה שבחברה ☺



[מערכת Cybereason מזהה את חולשת ה-privilege escalation ב-elevator.exe]

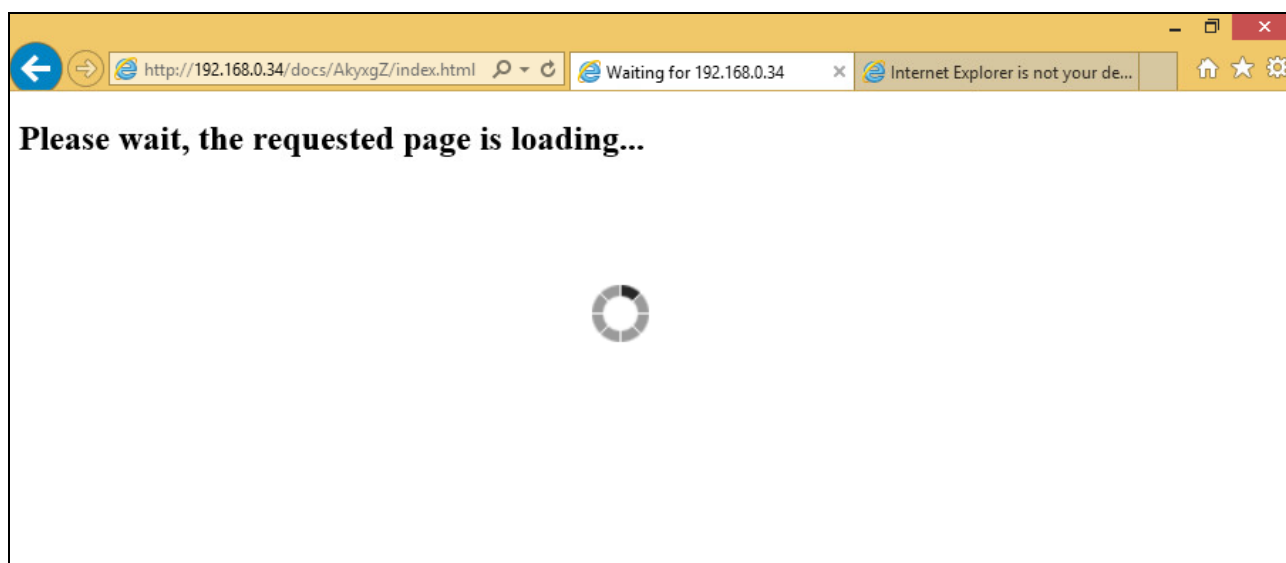
תהליך הטמעת ה-payload מרשים ברמת התחכום שלו. בעוד שרבים יטענו כי הכלים והחולשות שבהם השתמשו אנשי הקבוצה אינם מתוחכמים בפני עצמם, שיטות השימוש בהן והצירופים בהם עשו שימוש היו יצירתיים במיוחד. בנוסף, המגוון הרחב של שיטות פעולה איפשרו ללקוחות הקבוצה יכולת פעולת נרחבת כנגד יעדים שונים.

עם ביצוע ההדבקה של המטרה, רכיב ה-RCS נכנס לפעולה. ברשות HackingTeam עמד מגוון רחב של מודולים אותם יכלו להתקין, בהתאם לבקשת הלקוח, ביניהם: מודול צילום תמונות ממצלמת הרשת, מודול הקלטת שיחות סקיפ, מודול מעקב אחרי טקסט המוקלד במקלדת, מודול מעקב אחרי פעולות בנקאיות (הכוללות תשלום בביטקוין או במטבעות מקוונים אחרים), או מודול לזיהוי מיקומו הגאוגרפי של היעד.

בנוסף לכך, לקבוצה היו יכולות המותאמות לשימוש בתקיפות של טלפונים ניידים, ביניהן היכולת לשלוח הודעות SMS בלתי-נראות שעשו שימוש בחולשות במערכת ה-SMS של הטלפון ואיפשרו ל-HackingTeam להתקין את התוסף שלהם לטלפונים ניידים אשר ביצע פעולות כמו הפעלת המיקרופון של הנייד, ושידור בזמן אמת מהחדר בו נמצא הטלפון הנתקף.

HackingTeam, Network Injector וה-Anonymizer: כלים ייחודיים ל-HackingTeam

התהליך המתואר לעיל הוא רק דוגמא אחת להליך תקיפה של הקבוצה. HackingTeam העניקו מגוון פתרונות שהותאמו לצרכי הלקוח, ביניהם פתרונות שהותאמו לתקיפות של מדינות וצבאות. דוגמא אחת לכך היתה השימוש ב-network injector, כלי אכזרי במיוחד שהתחבר לתשתית שרת האינטרנט. עם הפעלתו, ה-network injector זיהה את המטרה/ות בהתאם לסט כללים שהוגדרו לו מראש על ידי הלקוח, וחיכה שהקורבן יבקר בכתובת אינטרנט מסויימת, לדוגמא - YouTube.com. כניסה לאתר המוגדר מראש על ידי הקורבן גרמה להעברתו לשרת ההדבקה במקום לאתר המבוקש. לקורבן הוצג מסך שבו נכתב "האתר בטעינה".



[זהו המסך אותו ראה היעד בזמן שה-exploit יותקן על המחשב שלו]

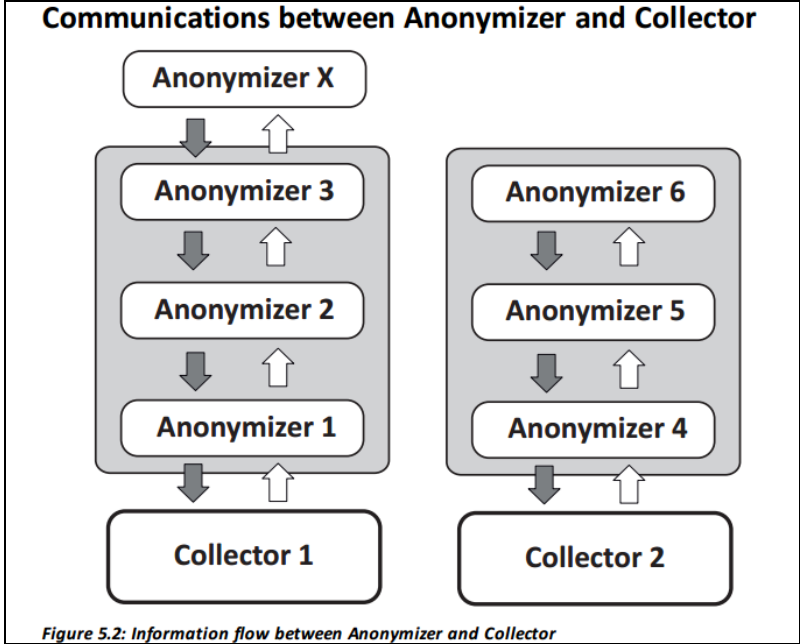
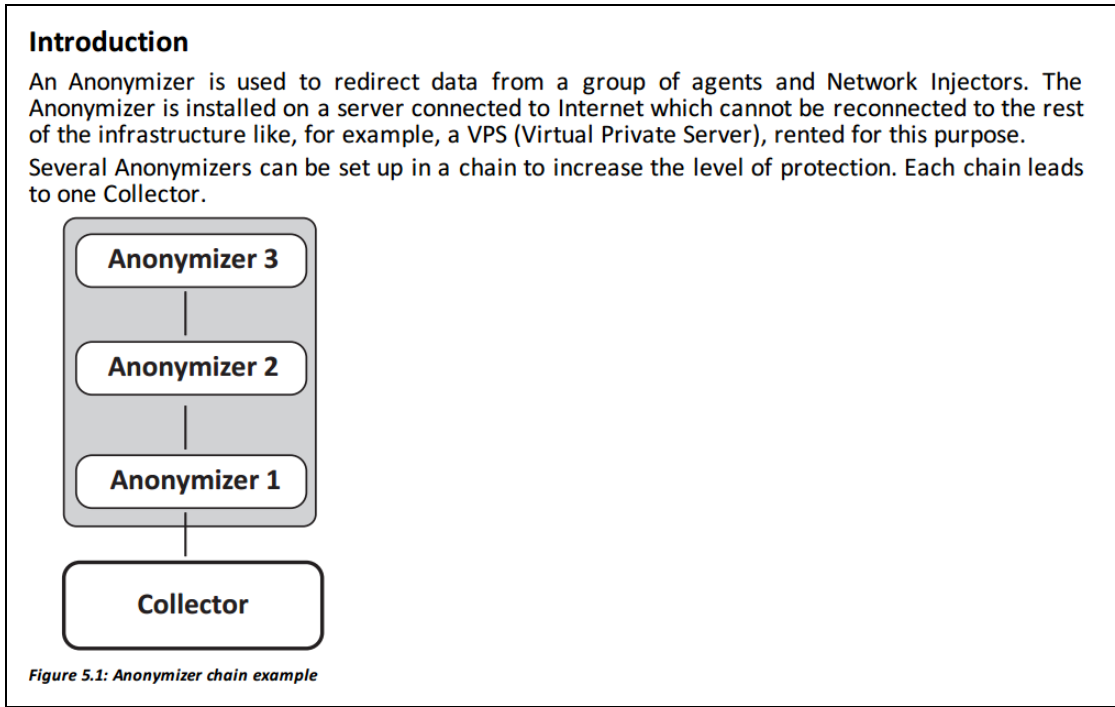
במקביל לשימוש ב-network injector עשו אנשי הקבוצה שימוש בכלי המכונה Melter. כלי זה איפשר ללקוחות "להתיר" את רכיב ה-RCS אל תוך הקוד של תוכנה תמימה כלשהי. על אף ששיטה זו אינה חדשה לכשעצמה, בשילוב עם ה-Network Injector היא מאפשרת לקמפיין לתקוף הורדות תוכנה ומוודאת למעשה שהמטרה/ות התקינה את רכיב ה-RCS יחד עם התוכנה התמימה אותה רצו להוריד מהרשת.

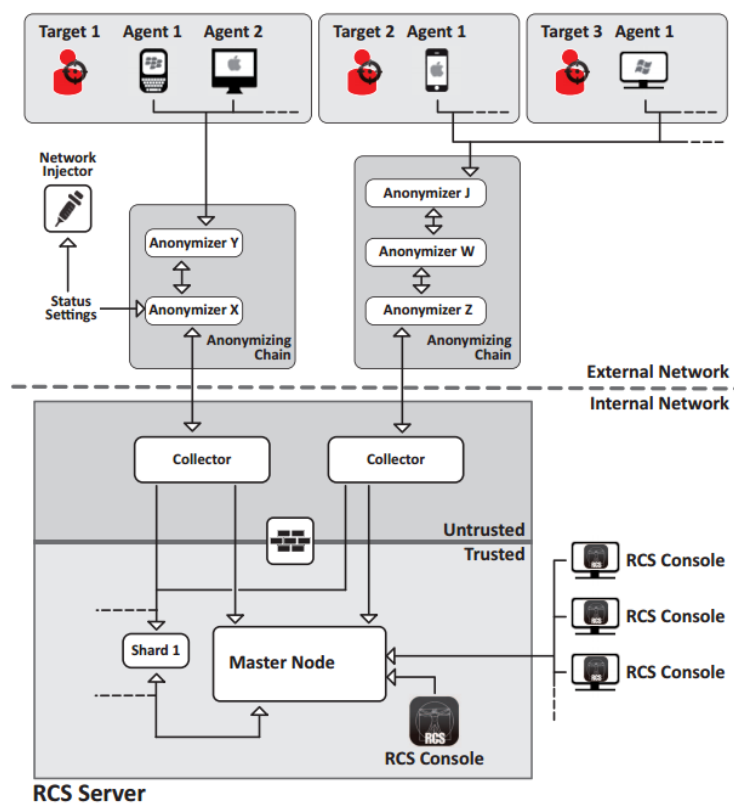
מובן שכל אחת מהשיטות שתוארו לעיל ניתנות לגילוי - ועל כן אנשי HackingTeam בנו גם תשתית להסתרת כתובות ה-IP של מערכות התקיפה: ה-Anonymizer. האנונימיזר היה פתרון מבוסס ענן שהוצע על ידי HackingTeam. הוא איפשר לכל לקוח להתקין שרת וירטואלי פרטי - VPS - virtual private server - אשר יכול להיות משורשר לפרוקסי אנונימי על מנת למנוע יכולת מעקב אחרי ה-collectors החיצוניים הרצים על ידי כל לקוח.

שידרוג מיידי לכל האקר מתחיל: HackingTeam זליגת שיטות התקיפה של קבוצת

www.DigitalWhisper.co.il

דבר זה הושג על ידי העברת המידע שנאסף מהקורבנות דרך מספר של מכונות אנונימיזציה עד ל- collector node אשר העביר את המידע חזרה ל-master node (שרת ה-C&C). להלן מספר דוגמאות של תיעוד כלי ה-Anonymizer, כפי שנאספו מתוך הוראות השימוש ב-RCS 9.6 של HackingTeam:





חשוב לציין כי קוד המקור של כל הכלים שתוארו לעיל זמין כעת להורדה ושימוש על ידי כל החפץ בכך. למעשה, היכולות שתוארו להלן שוחררו לאוויר העולם והן זמינות לשימוש חנים על ידי כל האקר מומחה או מתחיל. יכולות אלו, בשילוב עם הדיווחים על BGP hijacking attack (לקריאה: [כאן](#) ו[כאן](#)), איפשרו ל-HackingTeam לפחות באופן תאורטי (ובהנחת נגישות מתאימה לעורקי תעבורה) להעביר את כל משתמשי האינטרנט דרך המערכות שלהם ולהדביק אותם.

לסיכום

דליפת המידע של קבוצת HackingTeam בעקבות תקיפת הסייבר על החברה חשפה שיטות תקיפה חדשות, כלים המנצלים חולשות לא ידועות, ויכולות מיסוך, הסוואה והטעיה. יכולות תקיפה מתוחכמות אלו היו עד כה ברשות האקרים הפועלים בחסות מדינות וגופים גדולים, ומעתה הן חופשיות לשימוש לכל דורש. הדו"ח לעיל חושף מספר שיטות פעולה של הקבוצה על מנת לאפשר פיתוח אמצעי זיהוי והתגוננות מפניהם.

עמית סרפר ואלכס פרייזר הינם חוקרים בחברת Cybereason, חברת סטארט-אפ המייצרת פתרון מתקדם לאיתור מבצעי תקיפות רשת מורכבים. עקבו אחרינו בטוויטר:

[@0xAmit](#) and [@awfrazier](#)



דברי סיכום

בזאת אנחנו סוגרים את הגליון ה-63 של Digital Whisper, אנו מאוד מקווים כי נהנתם מהגליון והכי חשוב- למדתם ממנו. כמו בגליונות הקודמים, גם הפעם הושקעו הרבה מחשבה, יצירתיות, עבודה קשה ושעות שינה אבודות כדי להביא לכם את הגליון.

אנחנו מחפשים כתבים, מאיירים, עורכים ואנשים המעוניינים לעזור ולתרום לגליונות הבאים. אם אתם רוצים לעזור לנו ולהשתתף במגזין Digital Whisper - צרו קשר!

ניתן לשלוח כתבות וכל פניה אחרת דרך עמוד "צור קשר" באתר שלנו, או לשלוח אותן לדואר האלקטרוני שלנו, בכתובת editor@digitalwhisper.co.il.

על מנת לקרוא גליונות נוספים, ליצור עימנו קשר ולהצטרף לקהילה שלנו, אנא בקרו באתר המגזין:

www.DigitalWhisper.co.il

"Talkin' bout a revolution sounds like a whisper"

הגליון הבא ייצא ביום האחרון של חודש אוגוסט 2015.

אפיק קסטיאל,

ניר אדר,

28.07.2015