

# פרק 0: הקדמה

## פרטים

- שם הקובץ: bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3
- גודל: 180 קילו-בייט.
- סוג: בינארי

## מטרות

- להבין מה הקובץ החשוד יכול לעשות.
- האם הוא מבצע תקשורת החוצה.
- האם כותב / מוחק קבצים / ממערכת ההפעלה.
- כיצד לחסום את הקובץ.
- יצירת חוקים לחסימת הקובץ במערכת הגנה ארגונית.
- הקמת EDR בענן וקבלת התראות בזמן אמת על הקובץ החשוד.

## טכניקות

- בשביל להבין את הקובץ החשוד (קובץ בינארי בלתי ניתן לקריאה (Non-Human Readable)), אשתמש בשתי טכניקות:
- ניתוח סטטי** – מערב כלים שונים, אשר כל אחד נותן פיסות מידע שונות על הקובץ. ניתוח סטטי מתבצע ללא הרצה של הקובץ. הכלים השונים יקנו תמונה ברורה יותר לאופי הקובץ ומה הוא יכול לעשות מבלי להריץ אותו על מערכת ההפעלה.
  - ניתוח דינאמי** – הרצת הקובץ הבינארי עם כלי ניטור שונים בשביל לעקוב אחרי הפעילות של הקובץ על המערכת הן בתעבורה ברשת והן על קבצים, קבצי רג'יסטרי, תהליכים וכו'...
  - מיתון ודרכי הגנה** – לאחר שלבי הניתוח, צריך לייצר כללי הגנה בכדי למנוע מהקובץ לחדור לרשת ארגונית ו/או לתפוס הפעלת הקובץ אצל משתמשי קצה באם חלילה חדר לארגון. מפורט [בנספח א'](#) בסוף הקובץ.

## גילוי נאות

לא ידעתי כלום על ניתוח מאלווררים עד לזמן שהתחלתי עם האתגר הזה. כל מה שרשמתי במסמך זה הגיע מתוך לימוד עצמי וחקירה. מפאת זמן ועצם העובדה שהמאלוור לא פעיל יותר, לא חקרתי אותו עם דיבאגרים כמו IDA ו-OlllyDBG כחלק מהניתוח הדינאמי הרצוי. מה שמאוד התחברתי אליו הוא הקמת EDR ברשת הביתית וניתוח בזמן אמת כאילו אני יושב בארגון ומקבל התראות בזמן אמת ממשתמשי קצה.

# תוכן עניינים:

1.....	פרק 0: הקדמה
1.....	פרטים
1.....	מטרות
1.....	טכניקות
1.....	גילוי נאות
4.....	פרק 1: ניתוח סטטי
4.....	אילו פונקציות הקובץ משתמש
4	ADVAPI32.dll
5	USER32.dll
5	IMM32.dll
5	RASAPI.dll
5	KERNEL32.dll
6.....	זיהוי סוג הקובץ, (PE File Headers & Sections)
7	סיכום
8.....	חתימות מאלוור (Malware Hashing)
8	יצירת חתימות רלוונטיות עם HashMyFiles
8	PEStudio Hashes
8	Sections MD5 Hashes
9.....	ניתוח סטטי ב-VirusTotal
9	Relations Tab
10.....	ניתוח סטטי ב-Any.run ע"י חיפוש ה-MD5
11.....	חליצת סטרינגס (STRINGS)
12.....	PE Header
12.....	Resources
13.....	דרכים לחסימת המאלוור
13	Yara rules
14	זיהוי על פי Indicators of Compromise
15.....	פרק 2: ניתוח דינאמי
15.....	שימוש ב-Malware Sandbox
15	Any.Run

---

## פרק 0: הקדמה

---

19.....	הרצה על מכונה וירטואלית
19	ניטור עם Process Monitor
24	ניתוח עם Process Explorer
30	השוואת שינויים ברג'יסטרי עם RegShot
31	זיוף רשת
32	תפיסת קבצים שנחקקים אוטומטית
33	Microsoft Network Monitor
34.....	סיכום דברים / ממצאים
נספח א': דוגמה להקמת EDR בענן, קביעת חוקי D&R, הרצת המאלוור במכונה אחרת וזיהוי בזמן אמת	
36.....	
36	יצירת חוקי Detection & Response (D&R)
39	הרצת המאלוור במכונה ובדיקת אמינות החוקים שנוצרו

# פרק 1: ניתוח סטטי

## אילו פונקציות הקובץ משתמש

פתיחת הקובץ עם *PEStudio-i Dependency Walker*, כלים לניתוח קבצים והצגת ספריות שהוא מייבא. מתוך המודולים אפשר לראות את הפונקציות השונות שמיוצאות מהספרייה בתצורת עץ.

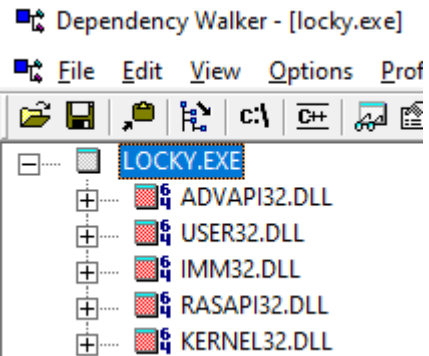
### ADVAPI32.dll

הפונקציות מהמודול הזה מספרות לנו שיש שימוש ברג'יסטרי. פונקציות כגון *RegCreateKeyExW*, *RegDeleteKeyA*, *RegSetValueA* וכדומה.

:MITRE-ATT&CK

[MITRE-Technique T1112](#) פונקציות אלה מזוהות ע"י MITRE כאלה שמשמשות בטכניקת [MITRE T1112](#). "תוקף אפשרי יכול לפעול עם מפתחות רג'יסטרי כדי להסתיר מידע על קונפיגורציה, למחוק מידע כחלק מניקוי, טשטוש עקבות, או כחלק מטכניקות אחרות כדי לסייע בעבודה עקבית והרצה של התוכנה."

[MITRE Technique T1106](#) פונקציית *CreateProcessAsUserA* מזוהה כשמשומשת כחלק מטכניקה T1106 ע"פ MITRE-ATT&CK. "תוקפים עשויים לקיים אינטראקציה ישירה עם ממשק תכנות היישומים המקורי של מערכת ההפעלה (API). ממשקי API מקומיים מספקים אמצעי מבוקר לקרוא לשירותי מערכת הפעלה ברמה נמוכה בתוך הליבה, כגון אלה הקשורים עם חומרה / מכשירים, זיכרון ותהליכים. ממשקי API מקוריים אלה ממונפים על ידי מערכת ההפעלה במהלך אתחול המערכת (כאשר רכיבי מערכת אחרים עדיין לא מאותחלים) וכן ביצוע משימות ובקשות במהלך פעולות שגרתיות."



<a href="#">RegQueryInfoKeyA</a>	registry	-	advapi32.dll
<a href="#">RegSetValueExA</a>	registry	x	advapi32.dll
<a href="#">RegDeleteKeyA</a>	registry	x	advapi32.dll
<a href="#">RegCloseKey</a>	registry	-	advapi32.dll
<a href="#">RegQueryValueA</a>	registry	-	advapi32.dll
<a href="#">RegLoadKeyA</a>	registry	x	advapi32.dll
<a href="#">RegConnectRegistryA</a>	registry	x	advapi32.dll
<a href="#">RegQueryValueW</a>	registry	-	advapi32.dll
<a href="#">RegSetValueW</a>	registry	x	advapi32.dll
<a href="#">RegOpenKeyExA</a>	registry	-	advapi32.dll
<a href="#">RegCreateKeyExW</a>	registry	-	advapi32.dll
<a href="#">RegFlushKey</a>	registry	x	advapi32.dll
<a href="#">RegSetValueA</a>	registry	x	advapi32.dll

Figure: מודולים בשימוש ע"י הקובץ החשוד

שימוש ב-*AddAce* מוסיפה רשומה ל-*Access Control List* אשר מכילה סטים של זכויות גישה המזהות למי הזכויות מוגשות, חסומות או משונות. [\[MSDN\]](#)

<a href="#">AddAce</a>	security	x	advapi32.dll
------------------------	----------	---	--------------

## פרק 1: ניתוח סטטי

עוד פונקציה מעניינת היא הפונקציה *EncryptFileW*, אשר מצפינה קבצים או תיקיות. אם תיקייה הוצפנה כל הקבצים תחתיה מוצפנים גם כן. פונקציה זו היא רמז ראשוני לכך שהקובץ החשוד הוא מאלוור מסוג **Ransomware**.

<a href="#">EncryptFileW</a>	cryptography	x	advapi32.dll
------------------------------	--------------	---	--------------

## USER32.dll

היבואים מ-*USER32.dll* אומרות שקיים GUI לתוכנה, אך לא בהכרח שהוא מוצג למשתמש. *ShowWindow*, *DrawText*, וכו'... כמו כן יש שימוש בפונקציית *mouse\_event* אשר עוקבת אחרי תנועות ולחיצות עכבר. רומז על כך שקיים *Keylogger* על התוכנה.

## IMM32.dll

ספרייה שמשומשת ע"י Microsoft Windows Input Method Manager (IMM), חייבת לפעול בשביל שווינדוס יפעל כראוי. הקובץ משתמש בספרייה כדי לדמות לחיצות קיצורי מקשים (אשר כביכול קיימים בקובץ) (ראה *Accelerators Objects*) כאילו אותו משתמש לחץ עליהם. דוגמה לפונקציה מסוג כזה היא *ImmSimulateHotKey*.

<a href="#">ImmSimulateHotKey</a>	-	-	imm32.dll
-----------------------------------	---	---	-----------

## RASAPI.dll

*RASAPI.dll* קשור לכל הנוגע בהתחברות RAS (*Remote Access Service*) עם סרבר וקליינט. המאלוור משתמש בפונקציית *RasDial* האחראית להקמת ערוץ תקשורת RAS בין קליינט לסרבר. המידע בחיבור יכול להיות בתוכו פרמטר CALLBACK ואמצעי זיהוי של המשתמש (קליינט).

<a href="#">RasDialA</a>	-	-	rasapi32.dll
<a href="#">RasGetProjectionInfoA</a>	-	-	rasapi32.dll

Figure 1: שתי הפונקציות מתוך הספרייה RASAPI

## KERNEL32.dll

שלושת הפונקציות המיובאות מתוך *KERNEL32* מספרות לנו שהקובץ אולי עושה מניפולציות תהליכים ועל קבצים במערכת.

- *GetLongPathNameA* הופכת את הנתיב שניתן לנתיב בפורמט הארוך שלו. (Converts specified path to its long form)
- *WriteFileGather* לוקחת מידע מתוך מערך וכותבת אותו לתוך קובץ.
- *PulseEvent* מסנכרנת תהליכים. מציבה את אובייקט האירוע שצוין למצב מסומן ואז מאפסת אותו למצב המקורי (?) לאחר ששוחררו כל מספר התרדים (Threads) הממתינים המתאימים.

<a href="#">PulseEvent</a>	synchronization	-	kernel32.dll
<a href="#">WriteFileGather</a>	file	-	kernel32.dll
<a href="#">GetLongPathNameA</a>	-	-	kernel32.dll

# זיהוי סוג הקובץ, (PE File Headers & Sections)

קובץ **Portable Exe**, מזוהה ע"י שני הבייטים הראשונים (MZ) 4D 5A + המחרוזת:

“This program cannot run in DOS mode”

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ .....ÿÿ..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	.....@.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	E0	00	00	00	.....à
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	..°...'.í!..Lí!Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS
00000070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode....\$......

קובץ PE מכיל מידע רב בתוך ההדרים. ניתן להשתמש ב-**PEView** כדי לסקור את המידע בתוך ההדרים.

	pFile	Data	Description	Value
bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3	000000E4	014C	Machine	2 IMAGE_FILE_MACHINE_I386
IMAGE_DOS_HEADER	000000E6	0004	Number of Sections	
MS-DOS Stub Program	000000E8	42B63E17	Time Date Stamp	3 2005/06/20 Mon 03:55:03 UTC
IMAGE_NT_HEADERS	000000EC	00000000	Pointer to Symbol Table	
Signature	000000F0	00000000	Number of Symbols	
IMAGE_FILE_HEADER	000000F4	00E0	Size of Optional Header	
IMAGE_OPTIONAL_HEADER				

Figure 1: פתיחת קובץ עם PEView

בתמונה הנ"ל מספר 1 מייצג את ה-Header הנבחר.

2 – מציג עם איזה סוגי מעבדים הקובץ רץ: i386

3 – מציג את תאריך הקימפול: 20 ליוני 2005 בשעה 03:55 UTC.

- תאריך הקימפול חשוב בשביל לדעת אם הקובץ ישן או חדש. אם התאריך ישן כנראה שכבר גורמים אחרים עשו עליו ניתוח, ואפשר יהיה להצליב נתונים עם חוקרים אחרים.

אזור ה-**IMAGE\_OPTIONAL\_HEADER**:

- **Subsystem** חושף אם הקובץ פועל עם **ממשק משתמש (GUI)** דרך מערכת ההפעלה או דרך **שורת פקודות (Command Line Interface)**.

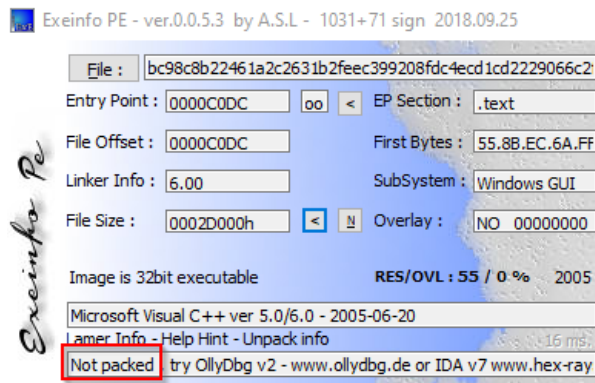
0000013C	0002	Subsystem	IMAGE_SUBSYSTEM_WINDOWS_GUI
----------	------	-----------	-----------------------------

הערך **IMAGE\_SUBSYSTEM\_WINDOWS\_GUI** מצביע על כך שלקובץ יש ממשק משתמש (ראוי לציון, שה-GUI לא תמיד נראה למשתמש, גם אם הוא קיים)

○ (אם הערך היה **IMAGE\_SUBSYSTEM\_WINDOWS\_CUI** משמע שהתוכנה משתמשת ב-CLI).

## פרק 1: ניתוח סטטי

פתיחת הקובץ עם *exeinfo*, מגלה שהקובץ לא ארוז:



פתיחה עם *pestudio*: זיהוי ארכיטקטורה:

file-type	executable
cpu	32-bit

## סיכום

- סוג הקובץ: Portable Executable (PE)
- ארכיטקטורה: 32-ביט.
- ארוז: שלילי.
- **Compiler-Stamp**: 0x42B63E17 (Mon Jun 20 06:55:03 2005)
- **מכונה**: i386.

# חתימות מאלוור (Malware Hashing)

## יצירת חתימות רלוונטיות עם HashMyFiles:

b06d9dd17c69ed2ae75d9e40b2631b42	MD5
b606aaa402bfe4a15ef80165e964d384f25564e4	SHA1
bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3	SHA256

## PEStudio Hashes

### ImpHash

imphash	0FCEA3AF550AD0A893E93808DCCF17F4
---------	----------------------------------

## Sections MD5 Hashes

property	value	value	value	value
name	.text	.rdata	.data	.rsrc
md5	D10A376572A1B107FA4157009223EDB1	F093C3F1C3A979DC5E0E622C993043A9	B765887F1CCA7B8EB7FC07FA8EB982B8	422ED60A1303FF02CDD0E1E518D33F01

```
hash_md5 = "b06d9dd17c69ed2ae75d9e40b2631b42"
hash_sha1 = "b606aaa402bfe4a15ef80165e964d384f25564e4"
hash_sha256 = "bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3"
imp_hash = "0FCEA3AF550AD0A893E93808DCCF17F4"
section_hash_text = "D10A376572A1B107FA4157009223EDB1"
section_hash_rdata = "F093C3F1C3A979DC5E0E622C993043A9"
section_hash_data = "B765887F1CCA7B8EB7FC07FA8EB982B8"
section_hash_rsrc = "422ED60A1303FF02CDD0E1E518D33F01"
```

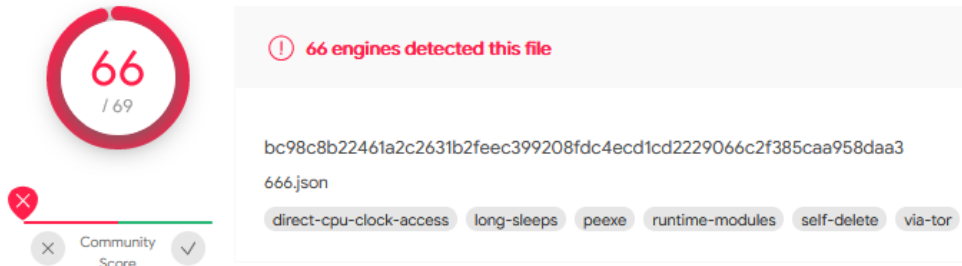
## שמירה לקובץ בפורמט הנכון לשימוש YARA מאוחר יותר:

```
1 hash_md5 = "b06d9dd17c69ed2ae75d9e40b2631b42"
2 hash_sha1 = "b606aaa402bfe4a15ef80165e964d384f25564e4"
3 hash_sha256 = "bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3"
4
5 imp_hash = "0FCEA3AF550AD0A893E93808DCCF17F4"
6
7 section_hash_text = "D10A376572A1B107FA4157009223EDB1"
8 section_hash_rdata = "F093C3F1C3A979DC5E0E622C993043A9"
9 section_hash_data = "B765887F1CCA7B8EB7FC07FA8EB982B8"
10 section_hash_rsrc = "422ED60A1303FF02CDD0E1E518D33F01"
```



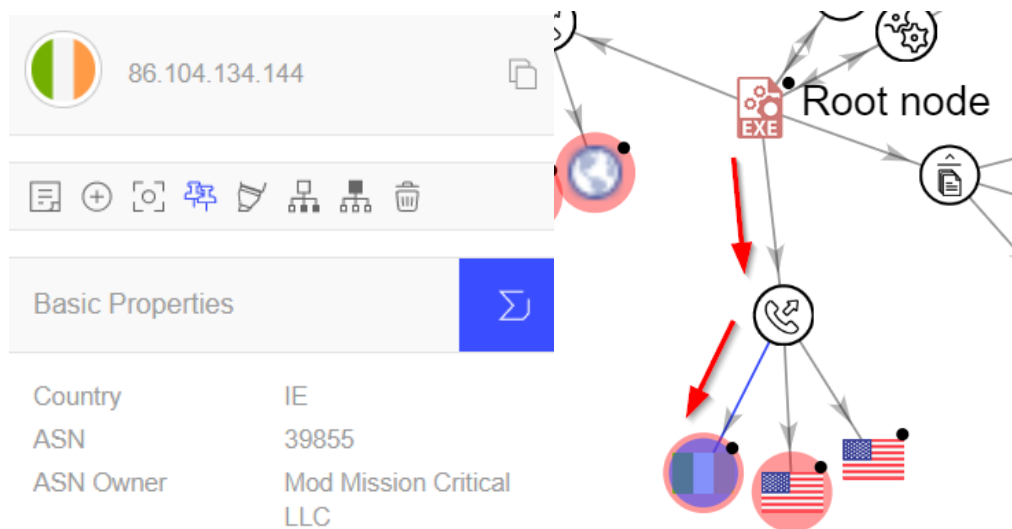
# ניתוח סטטי ב-VirusTotal:

- 66 מתוך 69 מנועי אנטי וירוס זיהו את הקובץ כמסוכן.
- סוג קובץ: Win32 EXE
- Magic: PE32 executable for MS Windows 32-bit
- מטרה: מעבדי Intel 386 ומעלה.



## Relations Tab

- מידע על האינטראקציות שיש לקובץ עם ארטיפאקטים אחרים כמו כתובות IP (כנראה C&C)
- תקשורת עם כתובת IP מאירלנד (מיוצגת ע"י החצים האדומים):



- מידע על התהליכים והקבצים שהקובץ משתמש בהם:

Execution Parents			
Scanned	Detections	Type	Name
2019-10-10	35 / 70	Win32 EXE	Testx.exe
2021-02-06	66 / 69	Win32 EXE	666.json
2016-06-05	48 / 57	Win32 EXE	d7458aafd2592610a765e9bb8db29077virus

Dropped Files			
Scanned	Detections	File type	Name
2021-02-06	66 / 69	Win32 EXE	666.json

PE Resource Children			
Scanned	Detections	File type	Name
2019-11-20	0 / 57	?	Icon125.bin
2020-11-15	0 / 61	?	151706
2019-11-20	0 / 55	?	151705.bin

# ניתוח סטטי ב-Any.Run ע"י חיפוש ה-MD5

חיפוש האש MD5 של הקובץ מקנה לי את המסקנות הבאות:

1. ההאש מזוהה כ-Malware מסוג Ransomware בשם Locky.

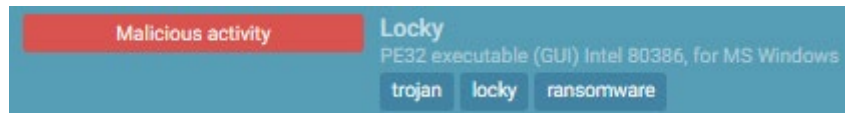


Figure III: תוצאות חיפוש ההאש ב-Any.Run

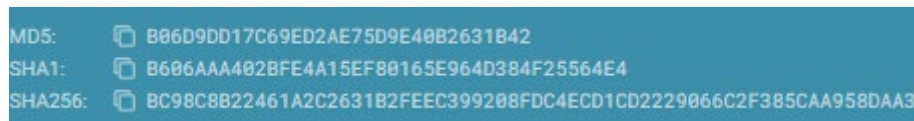


Figure IV: ההאשים שהוגשו עם הניתוח

שלושת ההאשים השונים תואמים את ההאשים שג'ונרטו מהקובץ הנחקר.

2. יוצר תקשורת החוצה בבקשת POST עם IP 86.104.134.144 בפורט 80. ספציפית לכתובת: <http://86.104.134.144/main.php>, שמקורה באירלנד. (קוד ארץ: IE)

# חליצת סטרינגס (STRINGS)

*floss Locky > flossLocky.txt*

**C:\Users\MalwareAnalysis\Desktop\Infected>floss Locky > flossLocky.txt**

```
$uni_string1 = "FileSee.com"
$uni_string2 = $uni_string = "0.37.213.27"
$uni_string3 = "Intend (C) 2013"
$uni_string4 = "Lipreading Fenced"
$uni_string5 = "uA2c861"
$uni_string6 = "&c516t5 JyTm73u7 kL6sBT j85Z074"
$uni_string7 = "&ht49y39 wJt5zXU"
$uni_string8 = "&Z3ZlW3 K6P638e yv5PTs"
$uni_string9 = "&kA88 f006q9"
$uni_string10 = "&K7065"
$uni_string11 = "&GHS"
$uni_string12 = "&CmQQK jF71w0m4 XD7"
$uni_string13 = "&Ujc5"
$uni_string14 = "&YU4 L2q1505 "
$uni_string15 = "32880x K0QB169"
$uni_string16 = "&Dgc VJ592op2"
$uni_string17 = "&v9SVCzg AHdhC1E"
$uni_string18 = "&o221 VR69953 McpL5U0E OREu22J"
$uni_string19 = "&Y91e1 d78Q4mcR fqX62"
$uni_string20 = "&e87h Y5MN0 B6801N C5B3K"
$uni_string21 = "d1Ypn3"
$uni_string22 = "wM124m F8Q uo4D1"
$uni_string24 = "HvEEL8a V782Jcv"
$uni_string25 = "XA045y1"
$uni_string26 = "NKw R872Q297"
$uni_string27 = "h0Wv"
$uni_string28 = "CJW9580z s2J0X"
$uni_string29 = "\X$2"
$uni_string30 = "yEUK2 DMUK gY47IG x5Ewn03"
$uni_string31 = "xNTT p2B5V"
$uni_string32 = "zq7y25 QVE24 WSfm w8MF6"
$uni_string33 = "ZQS01n"
$uni_string34 = "L6TDY2A4X"
$uni_string35 = "Hkh X18T5"
$uni_string37 = "ty3834 lew0 REY9j S7AKWo4"
$uni_string38 = "kR3113hA r16VKeI"
```

כשפותחים את קובץ הטקסט אפשר לראות את כל מחרוזות ה-*ASCII* וגם ה-*UTF-18*. רוב המחרוזות הם רק הוראות שפורשו כמחרוזות. כל המחרוזות המיוחדות יישמשו כסט חוקים ב-*YARA*. במצב הנוכחי, נראה שיש המון מחרוזות לכן בחרתי רק מחרוזות ארוכות ומיוחדות בתור מחרוזות שימשו ל-*YARA*. דבר זה נועד כדי למנוע עומס בתהליך איתור המחרוזות בחוק *YARA*.



hashes.txt



imports.txt



uni\_strings.txt

# PE Header

מציאת הפונקציות שמיובאות ע"י הקובץ. פתיחת הקובץ ב-**PESTUDIO** תיתן מידע אודות הפונקציות השונות וגם אינדיקציה לאם **PESTUDIO** מזהה את הפונקציה כפונקציה שמאלוורים אחרים משתמשים בה. פונקציות כאלה מסומנות ב-X תחת לשונית **BLACKLIST**. ניתן להשתמש בשמות של פונקציות אלה לחוקי YARA לצורך זיהוי המאלוור.

```
$import0 = "GetGuiThreadInfo"  
$import1 = "GetCapture"  
$import2 = "GetSecurityDescriptorDacl"  
$import3 = "GetSidSubAuthorityCount"  
$import4 = "GetKernelObjectSecurity"  
$import5 = "GetSidSubAuthority"  
$import6 = "LookupPrivilegeValueA"  
$import7 = "GetSidIdentifierAuthority"  
$import8 = "OpenThreadToken"  
$import9 = "LsaQueryInformationPolicy"  
$import10 = "MakeAbsoluteSD"  
$import11 = "AddAce"  
$import12 = "SetNamedSecurityInfoW"  
$import13 = "SetSecurityDescriptorSacl"
```

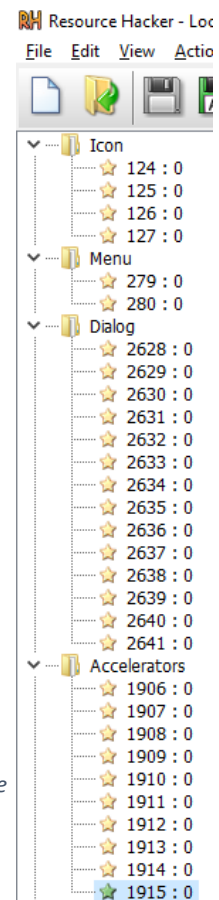
# Resources

עם **ResourceHacker** אפשר לבחון את רכיבי ה-GUI של המאלוור. רכיבי ה-GUI הם רכיבים שלא נראים בזמן הרצה דינאמית של המאלוור אלה קיימים בתוך הקובץ. רכיבים אלה יכולים לסייע בהבנה עמוקה יותר של פעילות המאלוור. כפי שניתן לראות מהתמונה, לקובץ יש 4 אייקונים, 2 אובייקטים שונים לתפריט, 14 פריטי דיאלוגים שמוצגים ע"י ביצוע פעולות שונות.

ראוי לציין שלמאלוור הזה יש 10 אובייקטים מאיצים (Accelerators). אלה קיצורי-מקשים לפעולות המקושרות לקובץ הבינארי ורומז לנו שאולי המאלוור עוקב אחרי הקלדות של משתמש.

```
1  
2 1914 ACCELERATORS  
3 LANGUAGE LANG_NEUTRAL, SUBLANG_NEUTRAL  
4 {  
5     "k", 22418  
6     "^M", 20324  
7     VK_NUMPAD3, 19770, ALT, CONTROL, SHIFT, VIRTKEY  
8     VK_F22, 20338, CONTROL, SHIFT, VIRTKEY  
9     "j", 17503  
10    "a", 20340  
11    "r", 18092  
12    VK_F15, 22922, SHIFT, VIRTKEY  
13 }
```

Figure V: דוגמא ל-Accelerator (מיוצג כ-1914)



# דרכים לחסימת המאלוור

## Yara rules

אחרי הניתוח הסטטי ואחרי שהוצאתי את כל המידע הרלוונטי, אשתמש בו כדי ליצור חתימות YARA עם סט חוקים. לקובץ חוקי YARA שלושה אזורים, **Meta**, **Strings** ו-**Condition**. כל אזור אחראי למשהו אחר כפי שמפורט:

## Meta Section

אזור זה הוא לשימוש קורא הקוד. הוא מכיל מידע על החוק, שם החוק, כותב החוק, האשים רלוונטיים, תאריך... לא משמש בבדיקת החוק עצמו או איתור התנאים.

```
1 rule LockyRansomware {
2   meta:
3     description = "rule to detect malware of ransomware type called Locky"
4     hash_md5 = "b06d9ddl7c69ed2ae75d9e40b2631b42"
5     hash_shal = "b606aaa402bfe4a15ef80165e964d384f25564e4"
6     hash_sha256 = "bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3"
7     imphash = "0FCEA3AF550AD0A893E93808DCCF17F4"
8     sector_text_hash = "D10A376572A1B107FA4157009223EDB1"
9     sector_rdata_hash = "F093C3F1C3A979DC5E0E622C993043A9"
10    sector_data_hash = "B765887F1CCA7B8EB7FC07FA8EB982B8"
11    sector_rsrc_hash = "422ED60A1303FF02CDD0E1E518D33F01"
12    date = "2005-06-25"
```

## Strings

← אזור זה מכיל את כל המחרוזות שהחוק ישתמש בהן לזיהוי המאלוור בעת התהליך. התנאים השונים נכתבים בצורה שיקללו את כל המחרוזות בקובץ, כל תנאי על פי אופיו:

```
14 strings:
15   $mz = {4D 5A}
16   $u0 = "FileSee.com" ascii
17   $u1 = "0.37.213.27"
18   $u2 = "2013" ascii
19   $u4 = "Lipreading Fenced" ascii
20   $ux5 = "uA2c861" wide
21   $ux6 = "c516t5 JyTm73u7 kL6sBT j85Z074" wide
22   $ux7 = "ght49y39 wJt5zXU" wide
23   $ux8 = "Z3ZW3 K6P638e yv5PTs" wide
24   $ux9 = "kA88 f006q9" wide
25   $ux10 = "K706S" wide
26   $uni_string11 = "GHS" wide
27   $uni_string12 = "CmQQK jF7lw0m4 XD7" wide
28   $uni_string13 = "Ujc5" wide
29   $uni_string14 = "YU4 L2ql505 " wide
30   $uni_string15 = "32880x K0QB169" wide
31   $uni_string16 = "Dgc VJ592op2" wide
32   $uni_string17 = "v95VCzg AHdhClE" wide
33   $uni_string18 = "o221 VR69953 McpL5U0E OREu22J" wide
34   $uni_string19 = "Y91el d78Q4mcR fqX62" wide
35   $uni_string20 = "e87h Y5MNO B6801N C5B3K" wide
36   $uni_string21 = "dlYpn3" wide
37   $uni_string22 = "wM124m F8Q uo4D1" wide
38   $uni_string23 = "Microsoft Sans Serif" wide
39   $uni_string24 = "HvEEL8a V782Jcv" wide
40   $uni_string25 = "XA045yl" wide
41   $uni_string26 = "NKw R872Q297" wide
42   $uni_string27 = "h0Wv" wide
43   $uni_string28 = "CJW9580z s2J0X" wide
44   $uni_string30 = "yEUK2 DMUK gY47IG x5Ewn03" wide
45   $uni_string31 = "xNTT p2B5V" wide
46   $uni_string32 = "zq7y25 QVE24 WSfm w8MF6" wide
47   $uni_string33 = "ZQ50ln" wide
48   $uni_string34 = "L6TDY2A4X" wide
49   $uni_string35 = "Hkh Xi8T5" wide
50   $uni_string36 = "Candara" wide
51   $uni_string37 = "ty3834 lew0 REY9j S7AKWo4" wide
52   $uni_string38 = "kR3l13hA r16VKeI" wide
53   $uni_string39 = "Yf5j0 FA2BQXE6 T6P733Z YL48" wide
54   $uni_string40 = "qW2a2j0 RTl Vi05" wide
55   $uni_string41 = "z2ad300" wide
56   $uni_string42 = "j3qBP9" wide
57   $uni_string43 = "a350 N3z77R9" wide
```

### Condition

אזור שבו נכתבים התנאים לבדיקה בהצלבת המאלוור עם החוק:

```
144 condition:
145     ($mz at 0)
146     20 of $uni_string1
147     $u0 and $u1 and $u3 and $u4
148     2 of ($a*) or 4 of ($ux*)
149     5 of ($import*)
150 }
```

`($mz at 0)` // Check for MZ '\x4D\x5A' Bytes at offset 0.  
`20 of $uni_string1` // Checks if 20 of the uni\_string strings are present.  
`$u0 and $u1 and $u3 and $u4` // Checks if all \$u are present.  
`2 of ($a*) or 4 of ($ux*)` // Checks if 2 of the text strings are present.  
Checks if 4 of \$ux strings are present.  
`5 of ($import*)` // Checks if 5 of the blacklisted \$import strings are present.

## זיהוי על פי Indicators of Compromise

### חתימות ההאש של הקובץ:

property	value
md5	<a href="#">B06D9DD17C69ED2AE75D9E40B2631B42</a>
sha1	<a href="#">B606AAA402BFE4A15EF80165E964D384F25564E4</a>
sha256	<a href="#">BC98C8B22461A2C2631B2FEEC399208FDC4ECD1CD2229066C2F385CAA958DAA3</a>

הדרך הפחות מאובטחת. מספיק שישתנה \ יוסף תו אחד (*Garbage Chars*) וכל ההאש של הקובץ יהיה אחר לגמרי אך הפונקציונליות של הנוזקה תישאר שלמה, מה שהופך את ההאש הנ"ל ללא יעיל. כל ה \$a לדוגמה שנוספו ע"י מפיץ הנוזקה "שיבשו" את חתימת ההאש.

### בקשות DNS שזוהו:

Domain	xwmaghtuu.pm
Domain	dyoqumjpbkr.pw
Domain	pjoaycpw.fr
Domain	ieehxmtcpwgdq.us
Domain	vtbqg.uk
Domain	goldrpbisbyly.in

### חיבורים

כתובת IP חשודה: 86.104.134.144

### בקשות HTTP

בבקשת *POST* <http://86.104.134.144/main.php>

שילוב של כל האינדיקטורים במערכת הגנה ארגונית תהיה הגישה היעילה ביותר

## פרק 2: ניתוח דינאמי

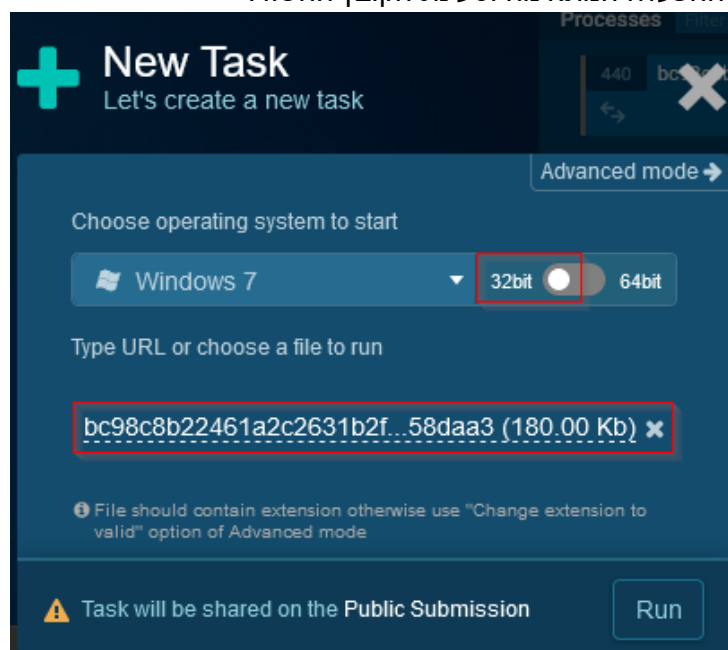
### שימוש ב-Malware Sandbox

#### Any.Run

[Any.Run](#) הוא שירות מצוין להעלאת קבצים חשודים ובחינתם בסביבה מאובטחת מבוססת ענן.

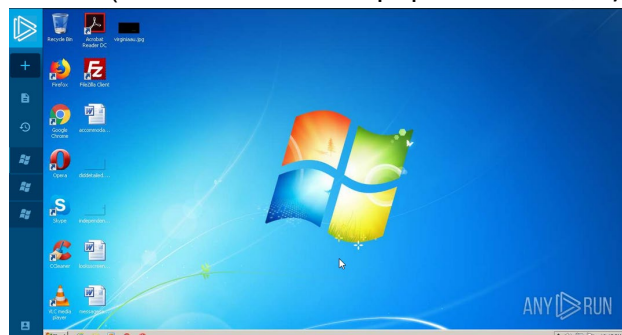
מתוך הניתוח הסטטי הבנתי שהקובץ הוא **Portable Executable** למערכות הפעלה מסוג **ווינדוס 32 ביט** בעלי מעבדים מסדרת **Intel i386** ומעלה.

1. בניית מערכת ההפעלה המתאימה וטעינת הקובץ החשוד:



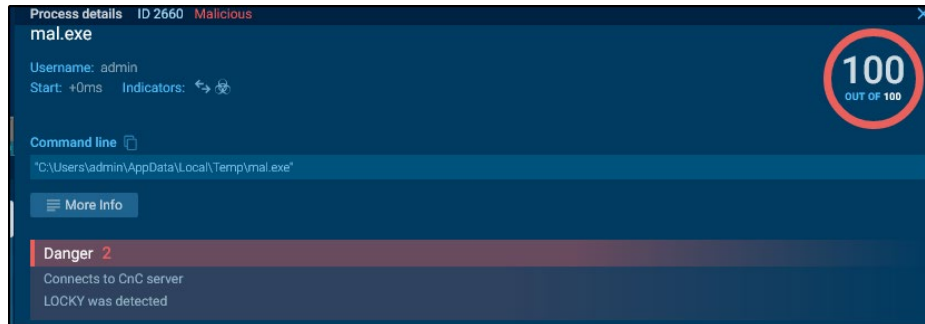
המערכת שתעלה תהיה **Windows 7 32-bit**. מיד אחרי שהמכונה עולה הקובץ יפעל ויוצג ניתוח על כל השינויים, בקשות HTTP, DNS, חיבורים ותהליכים.

2. אחרי לחיצה על **Run**, מערכת ההפעלה תעלה בסביבה מאובטחת ותריץ את הקובץ החשוד. (אם אין סיומת לקובץ Any.Run, תנתח את הקובץ כדי שיוכל לפעול כראוי)



## פרק 2: ניתוח דינאמי

**זיהוי הקובץ כמזיק:** מנסה להתחבר ל-C&C Server ומזוהה כ-LOCKY. כמו כן, 100 מתוך 100 מנועי אנטי וירוס מזיהים את הקובץ כמזיק:



## HTTP Requests

3. נראה שבוצעה בקשה אחת לכתובת 86.104.134.144 לדף בשם *main.php* בתצורת POST. הבקשה לא צלחה – אין תגובה מהשרת.

Request	
URL	/main.php
Method	POST
Host	86.104.134.144
Content-Length	100
Connection	Keep-Alive
Cache-Control	no-cache
Response	
No response	

## DNS Requests

4. ניסיון ל-6 בקשות DNS שונות:

I. xwmaghtuu.pm

II. dyoqumjpbrk.pw

III. pjoaycpw.fr

IV. ieehxmtcpwgdq.up

V. vtbqg.uk

VI. goldrpbisbyly.in

HTTP Requests		1	Connections		1	DNS Requests		6	Threats		1
NETWORK	Timeshift	Status	Rep	Domain		IP					
	48370 ms	Requested	?	xwmaghtuu.pm		IP Addresses not found					
	50417 ms	Requested	?	pjoaycpw.fr		IP Addresses not found					
	52461 ms	Requested	?	dyoqumjpbrk.pw		IP Addresses not found					
	55533 ms	Requested	?	qoldrpbisbyly.in		IP Addresses not found					
FILES	55534 ms	Requested	?	ieehxmtcpwgdq.us		IP Addresses not found					
	55534 ms	Requested	?	vtbqg.uk		IP Addresses not found					

לששת הבקשות **לא נמצאה** כתובת IP על כן לא התבצע חיבור והמאלוור נתקע בהרצה.



### הנחתת קבצים ניתנים להרצה (Dropped Exe Files)

המאלוור יצר קובץ EXE בתיקית %TEMP% בשם `svchost.exe`. לקובץ אותו האש SHA256 כמו למאלוור. כלומר, הקובץ הוא העתק מדויק של קובץ המאלוור עצמו המתחזה לתהליך `svchost.exe` על כן, זוהתה פעילות חשודה וזה נחשב כ-IOC:

Dropped executable file		
SHA256	C:\Users\admin\AppData\Local\Temp\svchost.exe	
	BC98C8B22461A2C2631B2FEEC399288FDC4ECD1CD2229866C2F385CAA958DAA3	

### Indicators of Compromise

Title	Type	IOC
• Main Object – “mal.exe”:		
	MD5	b06d9dd17c69ed2ae75d9e40b2631b42
• DNS Requests:		
	Domain	xwmaghtuu.pm
	Domain	dyoqumjpbrk.pw
	Domain	pjoaycpw.fr
	Domain	ieehxmtcpwgdq.us
	Domain	vtbqg.uk
	Domain	goldrpbisbyly.in
• Connections:		
	IP	86.104.134.144
• HTTP Requests:		
	URL	http://86.104.134.144/main.php

אינדיקטורים אלה יושמשו במערכת הגנה ארגונית להגנה מפני המאלוור באם אחד או יותר מהאינדיקטורים הנ"ל נקלטו ע"י המערכת (כגון EDR)

### איומים מרכזיים

- פעילות חשודה ומאיימת ע"י התחברות או ניסיון להתחברות לשרת Command & Control.
- בקשת DNS לדומיין עם פוטנציאל מזיק בתבנית \*.pw.

Threat details	Threat details
Here are the details of the threat	Here are the details of the threat
Suspicious activity	Malicious activity
Potentially Bad Traffic	A Network Trojan was detected
Timeshift ..... 71238 ms	Timeshift ..... 66646 ms
SID ..... 2016778	SID ..... 2022538
Message ..... ET DNS Query to a *.pw domain - Likely Hostile	Message ..... ET TROJAN Ransomware Locky CnC Beacon
Src / Dst ..... 192.168.100.62: 62228 ⇄ 192.168.100.2: 53	Src / Dst ..... 192.168.100.62: 49729 ⇄ 86.104.134.144: 80

### מסקנות ניתוח דינאמי עם Any.Run

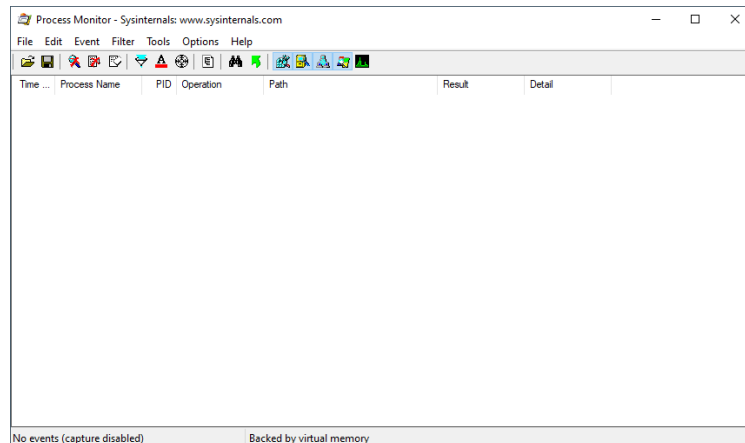
1. מאלוור שמנסה לתקשר עם שרת באירלנד המזוהה כמזיק בשרתי Any.Run (כנראה לצורך הורדת קבצים מזיקים אחרים המכילים מידע / הוראות / פקודות נוספות להפעלת המאלוור)
2. המאלוור שולח בקשות DNS הנ"ל בצורה אינסופית אך אין תגובה מבקשות אלו, ואין תגובה משרת ה-C&C, לכן המאלוור לא מממש את מטרותיו.

# הרצה על מכונה וירטואלית

## ניטור עם Process Monitor

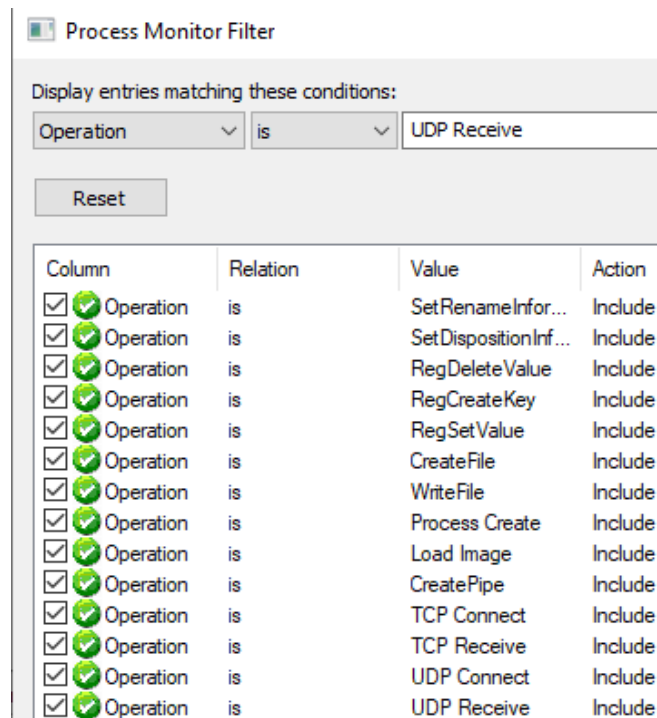
*Process Monitor* או *procmon*, חלק מ-Windows Sysinternals, הוא כלי ניטור מתקדם למערכות ווינדוס המקנה את היכולת לעקוב אחר פעילות רג'יסטרי, קבצי מערכת, תקשורת, ותהליכים.

### ניקוי התצוגה לפני הרצת המאלוור:



### הפעלת פילטרים

*Procmon* מציג מידע רב. לפעמים מידע רב מדי שיהיה קשה מדי לנסות ולנתח את כולו. לכן ל-*procmon* יש אפשרות להפעלת פילטרים כדי להציג רק מידע רלוונטי ושחשוב לנו.



VI: Operations included for procmon display Figure

## פרק 2: ניתוח דינאמי

### הפילטרים שנבחרו לתצוגה:

**הפעולות** שבוצעו: אלה **פעולות** הכי נפוצות לניתוח מאלווריס ובהם אתמקד:

- **SetRenameInformationFile**: Rename operation occurred.
- **SetDispositionInformationFile**: Deletion of file occurred.
- **RegDeleteKey**: Key gets deleted from the registry.
- **RegDeleteValue**: The value of a key is deleted from the registry.
- **RegCreateKey**: Registry key is created.
- **RegSetValue**: Data in the registry is set in the registry.
- **CreateFile**: Process wants to create a file.
- **WriteFile**: Process writes data to file.
- **Process Create**: Process creates another process.
- **Load Image**: Process loads any DLL's \ Executables.
- **CreatePipe**: Process creates a Pipe.
- **TCP / UDP Connect / Receive**: Process is sending \ receiving TCP connection.

### שימוש ברג'יסטרי

ברגע הרצה נוצרים מפתחות רג'יסטרי **בשם של הקובץ + \_RASAPI32**. מפתחות אלה נוצרים כאשר אפליקציה עובדת עם ה-API של **גישה מרחוק (Remote Access)**, והספרייה **rasapi32.dll**. מפני שהקובץ עובד עם הספרייה הזו, ניתן להסיק שיש ניסיון לתקשורת.

לאחר שנוצרו המפתחות, הערכים ב-**EnableFileTracing** ו-**EnableConsoleTracing** מוגדרים ל-0. כתוצאה מכך, לא ייווצרו לוגים בתיקייה המוגדרת **FileDirectory** לפעולות שהמאלוור מבצע.

mal.exe	RegCreateKey	HKLM\Software\WOW6432Node\Microsoft\Tracing	SUCCESS
mal.exe	RegCreateKey	HKLM\Software\WOW6432Node\Microsoft\Tracing	SUCCESS
mal.exe	RegCreateKey	HKLM\Software\WOW6432Node\Microsoft\Tracing\mal_RASAPI32	SUCCESS
mal.exe	RegSetValue	HKLM\SOFTWARE\WOW6432Node\Microsoft\Tracing\mal_RASAPI32\EnableFileTracing	SUCCESS
mal.exe	RegSetValue	HKLM\SOFTWARE\WOW6432Node\Microsoft\Tracing\mal_RASAPI32\EnableAutoFileTracing	SUCCESS
mal.exe	RegSetValue	HKLM\SOFTWARE\WOW6432Node\Microsoft\Tracing\mal_RASAPI32\EnableConsoleTracing	SUCCESS
mal.exe	RegSetValue	HKLM\SOFTWARE\WOW6432Node\Microsoft\Tracing\mal_RASAPI32\FileTracingMask	SUCCESS
mal.exe	RegSetValue	HKLM\SOFTWARE\WOW6432Node\Microsoft\Tracing\mal_RASAPI32\ConsoleTracingMask	SUCCESS
mal.exe	RegSetValue	HKLM\SOFTWARE\WOW6432Node\Microsoft\Tracing\mal_RASAPI32\MaxFileSize	SUCCESS
mal.exe	RegSetValue	HKLM\SOFTWARE\WOW6432Node\Microsoft\Tracing\mal_RASAPI32\FileDirectory	SUCCESS

#### Event Properties

Event	Process	Stack
Date:	11/02/2021 18:27:43.9592737	
Thread:	3952	
Class:	Registry	
Operation:	RegSetValue	
Result:	SUCCESS	
Path:	HKLM\SOFTWARE\WOW6432Node\Mic	
Duration:	0.0000929	
Type:	REG_DWORD	
Length:	4	
Data:	0	

### קבצים

המאלוור מנסה לגשת לקבצים ולפי *procmon* נראה שהוא רק מנסה לקרוא. מתבטא בטור ה-: **Detail** *Desired Access: Read Attributes\Generic Read*

Process Monitor - Sysinternals: [www.sysinternals.com](http://www.sysinternals.com)

Process Name	Operation	Path	Result	Detail
mal.exe	CreateFile	C:\Windows\Prefetch\MAL.EXE-25C363BD.pf	NAME NOT FOUND	Desired Access: Generic Rea...
mal.exe	CreateFile	C:\Windows	SUCCESS	Desired Access: Execute/Tra...
mal.exe	CreateFile	C:\Windows\System32\wow64log.dll	NAME NOT FOUND	Desired Access: Read Attribut...
mal.exe	CreateFile	C:\Windows	SUCCESS	Desired Access: Read Attribut...
mal.exe	CreateFile	C:\Users\MalwareAnalysis\Desktop\Infected	SUCCESS	Desired Access: Execute/Tra...
mal.exe	CreateFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS	Desired Access: Read Attribut...
mal.exe	CreateFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS	Desired Access: Read Data/Li...
mal.exe	CreateFile	C:\Users\MalwareAnalysis\Desktop\Infected\mal.exe	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\ntdll.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\kernel32.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\KernelBase.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	Desired Access: Generic Rea...
mal.exe	CreateFile	C:\Users\MalwareAnalysis\Desktop\Infected\mal.exe	SUCCESS	Desired Access: Generic Rea...
mal.exe	CreateFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	Desired Access: Generic Rea...
mal.exe	CreateFile	C:\Users\MalwareAnalysis\Desktop\Infected\mal.exe	SUCCESS	Desired Access: Generic Rea...
mal.exe	CreateFile	C:\Users\MalwareAnalysis\Desktop\Infected\mal.exe	SUCCESS	Desired Access: Generic Rea...
mal.exe	CreateFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	Desired Access: Generic Rea...
mal.exe	CreateFile	C:\Users\MalwareAnalysis\Desktop\Infected\RASAPI32.dll	NAME NOT FOUND	Desired Access: Read Attribut...
mal.exe	CreateFile	C:\Windows\SysWOW64\rasapi32.dll	SUCCESS	Desired Access: Read Attribut...
mal.exe	CreateFile	C:\Windows\SysWOW64\rasapi32.dll	SUCCESS	Desired Access: Read Data/Li...
mal.exe	CreateFile	C:\Users\MalwareAnalysis\Desktop\Infected\vasman.dll	NAME NOT FOUND	Desired Access: Read Attribut...
mal.exe	CreateFile	C:\Windows\SysWOW64\vasman.dll	SUCCESS	Desired Access: Read Attribut...
mal.exe	CreateFile	C:\Windows\SysWOW64\vasman.dll	SUCCESS	Desired Access: Read Data/Li...
mal.exe	CreateFile	C:\Windows\SysWOW64\msvcrt.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\vpqrt4.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\sechost.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\advapi32.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\win32u.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\ucrtbase.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\msvcrt_win.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\gdi32full.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\gdi32.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\user32.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\imm32.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\vasman.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\combase.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\rasapi32.dll	SUCCESS	Desired Access: Read Control...
mal.exe	CreateFile	C:\Windows\SysWOW64\edgegdi.dll	NAME NOT FOUND	Desired Access: Read Attribut...
mal.exe	CreateFile	C:\Users\MalwareAnalysis\Desktop\Infected\vtutils.dll	NAME NOT FOUND	Desired Access: Read Attribut...
mal.exe	CreateFile	C:\Windows\SysWOW64\vtutils.dll	SUCCESS	Desired Access: Read Attribut...
mal.exe	CreateFile	C:\Windows\SysWOW64\vtutils.dll	SUCCESS	Desired Access: Read Data/Li...
mal.exe	CreateFile	C:\Windows\SysWOW64\vtutils.dll	SUCCESS	Desired Access: Read Control...

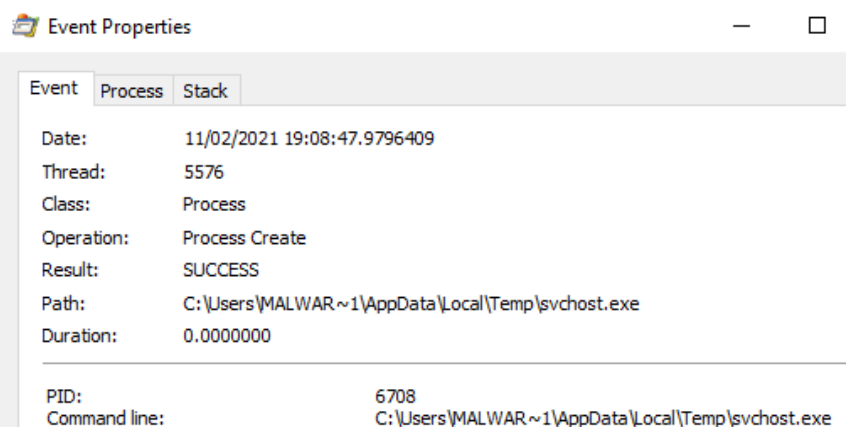
## פרק 2: ניתוח דינאמי

כמו כן, יש המון קבצים שהוא מנסה לגשת אליהם אך שמם לא נמצא לפי *procmon*.



## תהליכים

המאמלוג יצר שני תהליכים תחתיו: `svchost.exe` ו-`cmd.exe`.



*svchost.exe Event Properties :VII Figure*



## פרק 2: ניתוח דינאמי

בתהליך השני מתבצעת פקודה דרך ה-CMD למחיקת קובץ בשם `sys270E.tmp`, שנמצא בתיקיית Temp של המשתמש הנוכחי.

Command line:

`"cmd.exe /C del /Q /F "C:\Users\MALWAR~1\AppData\Local\Temp\sys270E.tmp"`

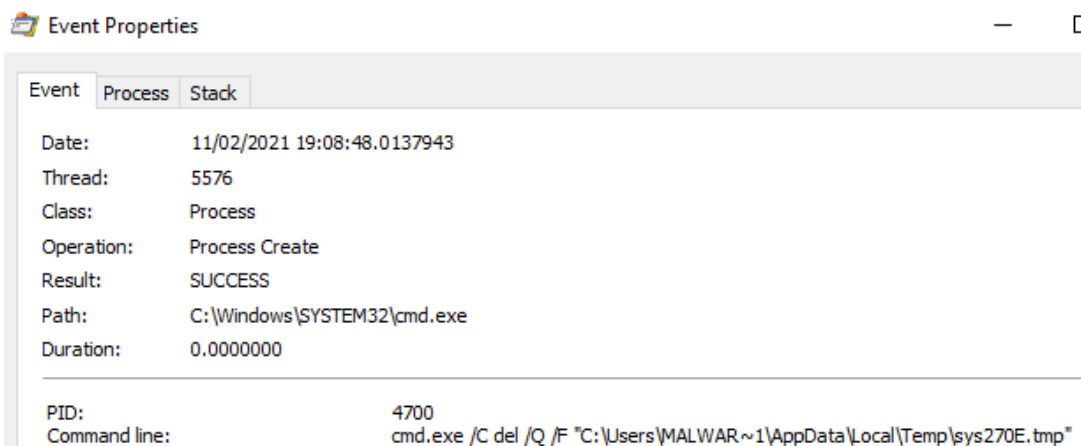
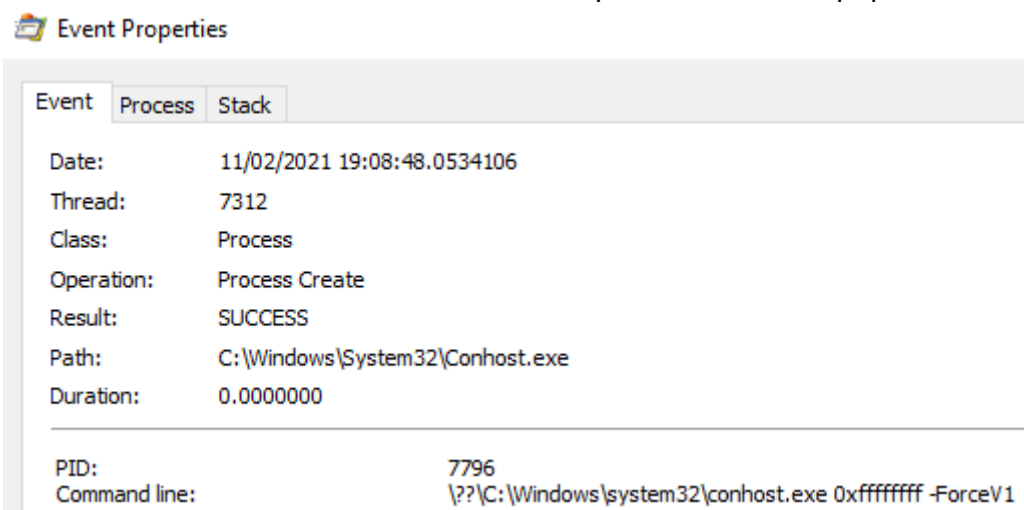


Figure VIII: cmd.exe Event Properties

דרך ה-CMD שנוצר מקובץ המאלוור, נוצר תהליך משני בשם `.Conhost.exe`.



cmd.exe Event Properties :IX Figure

Command Line:

`\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1`

mal.exe	Process Create	C:\Users\MALWAR~1\AppData\Local\Temp\svchost.exe	SUCCESS
mal.exe	Process Create	C:\Windows\SYSTEM32\cmd.exe	SUCCESS
C:\cmd.exe	Load Image	C:\Windows\System32\cmd.exe	SUCCESS
C:\cmd.exe	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS
C:\cmd.exe	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS
C:\cmd.exe	Load Image	C:\Windows\System32\KernelBase.dll	SUCCESS
C:\cmd.exe	Process Create	C:\Windows\System32\Conhost.exe	SUCCESS

## פרק 2: ניתוח דינאמי

תצורת עץ של התהליכים שנוצרו מתוך *procmon*:

mal.exe (6128)	C:\Users\MalwareAnalysis\Des...
svchost.exe (6708)	C:\Users\MALWAR~1\AppData...
cmd.exe (4700)	Windows Comma... C:\Windows\SYSTEM32\cmd.e...
Conhost.exe (7796)	Console Window ... C:\Windows\System32\Conhost...

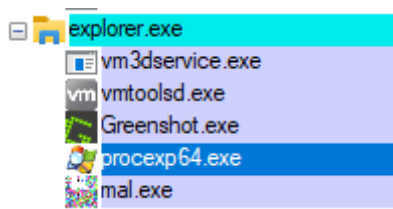
### תעבורת רשת

לא נתפסה תעבורה ברשת דרך *procmon* בפרוטוקולים TCP או UDP. ראוי לציין מהניתוח ב-Any.Run, כי המאלוור שולח בקשת HTTP בתצורת POST לכתובת IP מאירלנד. לכן כנראה הבקשה הזו לא נתפסה ב-*procmon*.

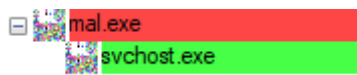
## ניתוח עם Process Explorer

*Process Explorer* מביא את האפשרות לנתח תהליכים שרצים על המערכת, לראות ספריות DLL שהוטענו מתהליך מסוים, תקשורת, ולנתח סטרינגים כפי שהם נטענו לזיכרון בעת הרצה.

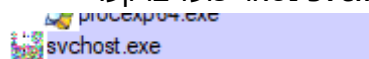
1. הרצת המאלוור:



2. המאלוור רץ, הורג את עצמו (מסומן באדום) ופותח תהליך חדש בשם *svchost.exe* בעצמו (מסומן בירוק):



3. בסופו של דבר רק *svchost.exe* נשאר פועל ברקע:



### חקירת svchost.exe

#### DLLים מעניינים בשימוש:

Name	Description	Verified Signer
advapi32.dll	Advanced Windows 32 Base API	(Verified) Microsoft Windows
apphelp.dll	Application Compatibility Client Library	(Verified) Microsoft Windows
bcrypt.dll	Windows Cryptographic Primitives Library (Wow64)	(Verified) Microsoft Windows
bcryptprimitives.dll	Windows Cryptographic Primitives Library	(Verified) Microsoft Windows
combase.dll	Microsoft COM for Windows	(Verified) Microsoft Windows
cryptbase.dll	Base cryptographic API DLL	(Verified) Microsoft Windows
cryptsp.dll	Cryptographic Service Provider API	(Verified) Microsoft Windows
dnsapi.dll	DNS Client API DLL	(Verified) Microsoft Windows
dnsapi.dll	DNS Setup Client DLL	(Verified) Microsoft Windows

1. *Advapi32* – שימוש ברג'יסטרים, פונקציות כגון: *AddAce*, *EncryptFileW*...

2. ספריות *crypto* של ווינדוס – מצאתי את ה-DLL הזה מעניין משום שהוא רומז שיש שימוש במפתחות קריפטוגרפיים מה שמחזק את ההשערה שהמאלוור מריץ Ransomware ע"י הצפנה של קבצים ותיקיות עם הפונקציה *EncryptFileW*.

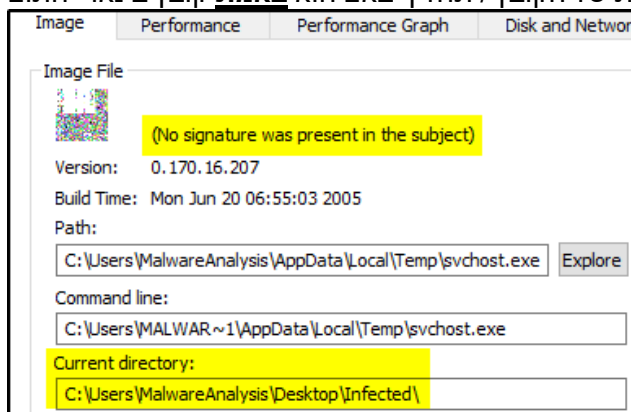


profapi.dll	User Profile Basic API	(Verified) Microsoft Windows
rasadhlp.dll	Remote Access AutoDial Helper	(Verified) Microsoft Windows
rasapi32.dll	Remote Access API	(Verified) Microsoft Windows
rasman.dll	Remote Access Connection Manager	(Verified) Microsoft Windows
rpcrt4.dll	Remote Procedure Call Runtime	(Verified) Microsoft Windows
rsaenh.dll	Microsoft Enhanced Cryptographic Provider	(Verified) Microsoft Windows
rtutils.dll	Routing Utilities	(Verified) Microsoft Windows
sechost.dll	Host for SCM/SDDL/LSA Lookup APIs	(Verified) Microsoft Windows
SHCore.dll	SHCORE	(Verified) Microsoft Windows
shell32.dll	Windows Shell Common Dll	(Verified) Microsoft Windows
shlwapi.dll	Shell Light-weight Utility Library	(Verified) Microsoft Windows
SortDefault.nls		(Verified) Microsoft Windows
sspicli.dll	Security Support Provider Interface	(Verified) Microsoft Windows
svchost.exe		(No signature was present in
ucrtbase.dll	Microsoft® C Runtime Library	(Verified) Microsoft Windows
user32.dll	Multi-User Windows USER API Client DLL	(Verified) Microsoft Windows
uxtheme.dll	Microsoft UxTheme Library	(Verified) Microsoft Windows
win32u.dll	Win32u	(Verified) Microsoft Windows
windows.storage.dll	Microsoft WinRT Storage API	(Verified) Microsoft Windows
winhttp.dll	Windows HTTP Services	(Verified) Microsoft Windows
wininet.dll	Internet Extensions for Win32	(Verified) Microsoft Windows
winnsi.dll	Network Store Information RPC interface	(Verified) Microsoft Windows
wldp.dll	Windows Lockdown Policy	(Verified) Microsoft Windows
wow64.dll	Win32 Emulation on NT64	(Verified) Microsoft Windows
wow64cpu.dll	AMD64 Wow64 CPU	(Verified) Microsoft Windows
wow64win.dll	Wow64 Console and Win32 API Logging	(Verified) Microsoft Windows
ws2_32.dll	Windows Socket 2.0 32-Bit DLL	(Verified) Microsoft Windows

1. **Rasapi32** ו-**rasman** – עוזרים לשימוש בשליטה מרחוק (Remote Access API & Manager)
2. **Svchost.exe** – התחזות ל-**svchost**, פותח תהליכים תחת תהליך אחד כדי לשמור על צריכת משאבים מהמחשב.
3. **Winhttp** – מנגישה את המערכת לממשק ברמה גבוהה עם תמיכה בשרתים לפרוטוקולים **HTTP2** ו-**HTTP1.1**. משומש בעיקר בסביבה של שרת ע"י אפליקציות שמתקשרות עם שרתי **HTTP**. רומז על כך שיש תקשורת או לפחות ניסיון לתקשורת בפרוטוקול **HTTP**.
4. **Wldp** – **Windows Lockdown Policy**. קורא לספריה בשביל לקבל את מצב האבטחה ביחס למערכת והסקריפט (תהליך ה-**svchost** בהקשר זה) או מנהל ההתקנות של ווינדוס (MSI) הרץ.
5. **Wow64** – מנגישה אפליקציות של 32-ביט למערכות 64-ביט.

### svchost: מאפיינים

אפשרות שימושית מאוד כשמנתחים מאלוור היא אפשרות ה-**Verify** בלשונית ה-**Image** במאפיינים של תהליך מסוים. לחיצה על **Verify** תוודא את האמינות של הקובץ / תהליך באם הוא **באמת** קובץ בינארי חתום



ממייקרוסופט. מייקרוסופט מטמיעה חתימות דיגיטליות לרוב הקבצים הניתנים להרצה מטעמה. כש-**Process Explorer** מוודאת את החתימה הדיגיטלית אפשר להיות בטוחים שהתהליך הוא באמת תוצר מבית מייקרוסופט ולא תהליך מתחזה בשם דומה.

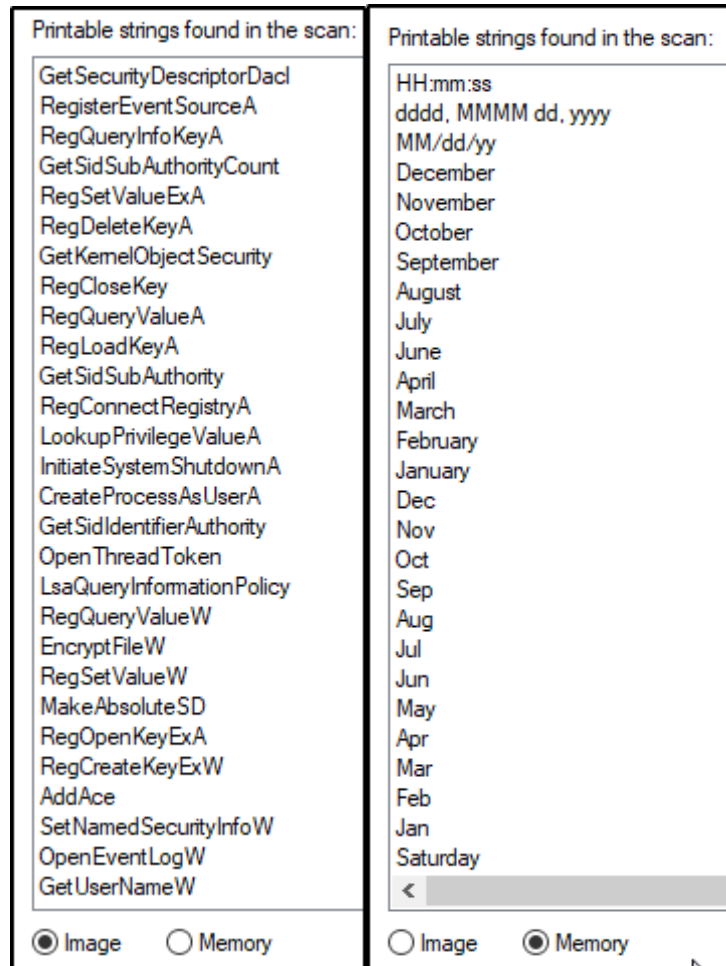
ל-**svchost.exe** שנפתח דרך התהליך של המאלוור אין חתימה דיגיטלית מה שאומר שהוא לא תהליך ממייקרוסופט ויכול להיות מזיק.

## פרק 2: ניתוח דינאמי

### השוואת סטרינגים

דרך לאמת החלפת תהליכים במאלווריס היא להשוות בין המחרוזות ב-Image לבין המחרוזות שבאמת נטענו לזיכרון. *Process Explorer* נותן לנו את האפשרות לבחון את זה בלשונית *Strings* בתוך המאפיינים של התהליך.

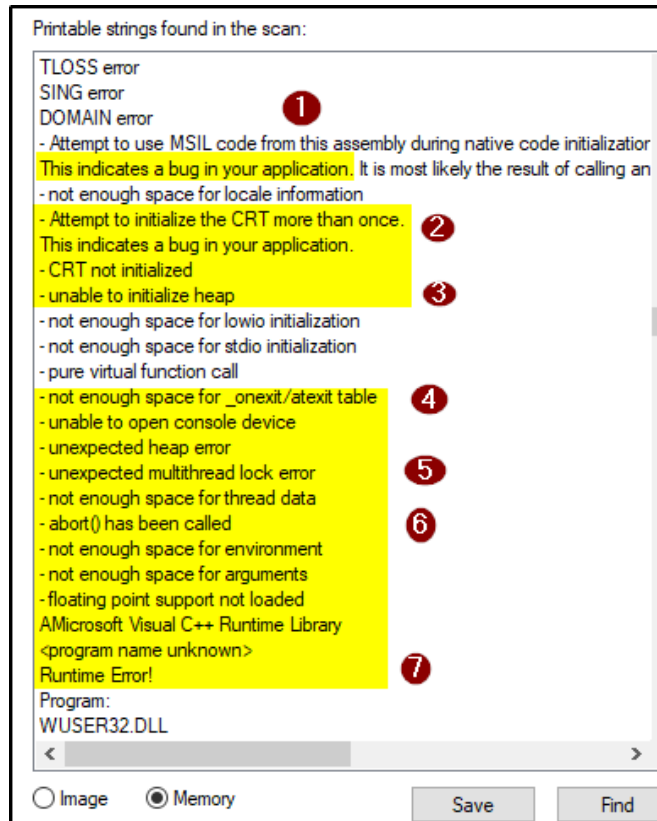
ב-*svchost.exe* המחרוזות שנטענו לזיכרון המערכת שונות מהמחרוזות ב-Image.



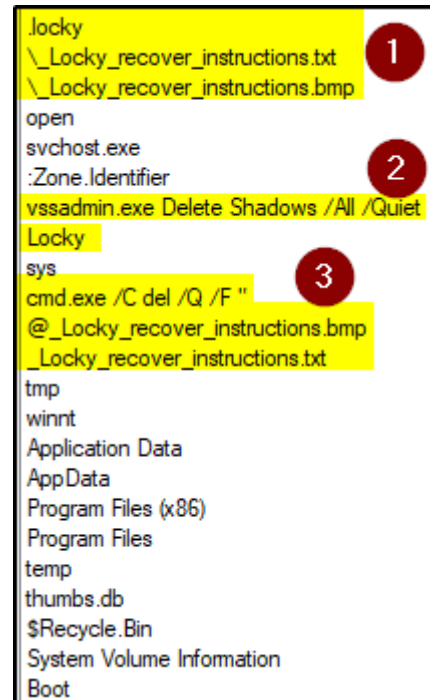
## פרק 2: ניתוח דינאמי

חלק מהמחרוזות שנטענו לזיכרון הם הודעות שגיאה על פעולות שהמאלוור מנסה לעשות.

1. ניסיון שימוש ב-Microsoft Intermediate Language לא צלח. יש באג בתוכנה. *"This indicates a bug in your program"*
2. אתחול CRT לא מוצלח.
3. הודעת שגיאה על HEAP שלא אותחל.
4. הודעות שגיאה על חוסר מיקום.
5. שגיאה לא צפויה ב-HEAP + שגיאה על נעילת טרדים.
6. פונקציה בשם *abort* נקראה. (ביטול המאלוור (??))
7. Runtime Error! שגיאה בזמן הרצה.



1. מחרוזות Locky מופיעות לכל אורך העמוד. Locky הוא שם של ה-Ransomware.
2. *Vssadmin.exe Delete Shadows /ALL /Quiet*: מוחק Shadow Copies מהמערכת.
  - */ALL* – כל הערכים (All Volumes)
  - */Quiet* – אל תוציא פלט, תעבוד ברקע.
3. *cmd.exe /C del /Q /F* – פקודת PowerShell להרצת פקודות עם CMD.
  - */C* – מריץ את הפקודה הבאה ויוצא מ-CMD.
  - *del* – הפקודה שתבצע. בהינתן נתיב, מוחק קובץ אחד או רשימה של קבצים.
  - */Q* – מצב שקט. לא שואל "כן/לא".
  - */F* – Force. אם קובץ מוגדר "לקריאה בלבד", מתעלם מההגדרה ומוחק בכל זאת.
4. *Locky\_recover\_instructions*: שם הקובץ בו כתובות הוראות לתשלום הכופר. הוראות אלה יוצגו למשתמש כשהמאלוור יפעל.



מספר 2 אומר שהמאלוור מוחק קבצי גיבוי מהמערכת כדי שלא יהיה ניתן להחזיר את המערכת לקדמותה טרם הרצת המאלוור.

מספר 3 מצביע על ניסיון למחיקת קבצים עם הפקודה בהינתן נתיב רלוונטי.

## פרק 2: ניתוח דינאמי

סיומות קבצים כגון אלה. המאלוור מתעסק עם עשרות, אם לא מאות סיומות קבצים.

Printable strings found in the scan:

.sldm  
.sldx  
.ppsm  
.ppsx  
.ppam  
.docb  
.mml  
.sxm  
.otg  
.odg  
.uop  
.potx  
.potm  
.ptx  
.ptm  
.std  
.sxd  
.pot  
.pps  
.sti  
.sxi  
.otp  
.odp  
.wks  
.xltx  
.xltm  
.xlsx

1. **קישורים (Reference) קבועים** למשתנים שונים, מסומנים עם הכרזת '&' בתחילת המשתנה. אלה יישמשו את התוכנה בשביל להגיע למשתנים בזיכרון אם נקראו. יש משתנה אחד מעניין בשם **encrypted** – כנראה משמש בתהליך הצפנת הקבצים ו/או בדיקה אם קובץ הוא מוצפן או לא.
2. **מערכות הפעלה:** החל מווינדוס 2000 עד ווינדוס 10, ווינדוס סרבר 2008 עד 2016.

```
GetActiveWindow  
MessageBoxW  
vector<T> too long  
string too long  
invalid string position  
&length=  
&failed=  
&encrypted=  
&act=stats&path=  
&act=report&data=  
Windows 2000  
Windows XP  
Windows 2003  
Windows 2003 R2  
Windows Vista  
Windows Server 2008  
Windows 7  
Windows Server 2008 R2  
Windows 8  
Windows Server 2012  
Windows 8.1  
Windows Server 2012 R2  
Windows 10  
Windows Server 2016 Technical Preview  
unknown  
&serv=  
&corp=
```

## פרק 2: ניתוח דינאמי

1. נתיב לרג'יסטרי בשם Locky תחת Software.
2. אולי קשור להוראות תשלום של הכופר המוצגות לקורבן.
3. נתיב רג'יסטרי בשימוש המאלוור – המאלוור כותב לנתיב זה מפתחות חדשים. כל מה שנמצא בתיקיית Run יפעל אוטומטית כל פעם שהמערכת עולה.
4. ביטול Redirection לקבצי מערכת. במערכות 64 ביט, נתיב `%windir%\System32` שמור לאפליקציות 64 ביט. רוב שמות קבצי DLL לא השתנו כשקבצי DLL ל-64 ביט נוצרו, אז קבצי DLL של 32 ביט אוחסנו בתיקייה נפרדת. **WOW64** מחביאה את השינוי עם *File System Redirector*, **ניתוב מחדש לקבצי מערכת**. כשאפליקציה מנסה לגשת אל תיקיות **System32** או **regedit.exe**, קורה ניתוב מחדש לנתיב הספציפי המתאים לארכיטקטורה הרצה. אפליקציות לא אמורות לגשת לתיקיות אלה, אלא להינתב ל-API כמפורט:

Original Path	Redirected Path for 32-bit x86 Processes
%windir%\System32	%windir%\SysWOW64
%windir%\lastgood\system32	%windir%\lastgood\SysWOW64
%windir%\regedit.exe	%windir%\SysWOW64\regedit.exe

```
Printable strings found in the scan:

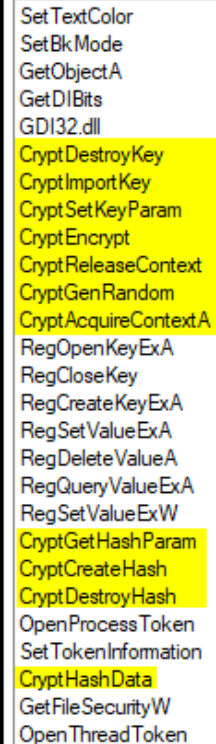
Control Panel\Desktop
WallpaperStyle
TileWallpaper
Software\Locky 1
pubkey
paytext 2
completed 3
&act=gettext&lang=
Software\Microsoft\Windows\CurrentVersion\Run
Locky
0123456789ABCDEF
Wow64DisableWow64FsRedirection 4
kernel32.dll
IsWow64Process
HTTP/1.1
rupweuiyntpmusfrdeitbeuknltf/main.php 5
http://
POST
bad exception
```

`Wow64DisableWow64FsRedirection` מבטל את הניתוב מחדש מה שמקנה לאפליקציה גישה ל-**System32** לכל הפעולות שאותו תהליך מנסה לעשות. כך לדוגמה, אפליקציית 32-ביט שרצה תחת **WOW64** יכולה לפתוח אפליקציות 64-ביט מתוך `%SystemRoot%\System32` במקום להינתב מחדש לגרסת ה-32-ביט בתוך `%SystemRoot%\SysWOW64`.

5. בקשת POST בפרוטוקול HTTP/1.1 – ניסיון לתקשורת עם כתובת: `rupweuiyntpmusfrdeitbeuknltf/main.php`

## פרק 2: ניתוח דינאמי

**פונקציות Crypt מתוך Wincrypt.h: CryptEncrypt** מצפינה מידע ע"פ הפרמטרים שניתנו לה מהפונקציות האחרות המתחילות ב-Crypt. לדוגמה **CryptGenRandom** ממלאת באפר בייטים קריפטוגרפים רנדומליים. **CryptCreateHash** יוצרת ומחזירה **Handle** לאובייקט האש של ספק שירות קריפטוגרפיה או בלעז, **(Cryptographic Service Provider (CSP)**. ה-**Handle** הזה משמש בקריאות פונקציות אחרות במשפחת Crypt.



SetTextColor  
SetBkMode  
GetObjectA  
GetDIBits  
GDI32.dll  
CryptDestroyKey  
CryptImportKey  
CryptSetKeyParam  
CryptEncrypt  
CryptReleaseContext  
CryptGenRandom  
CryptAcquireContextA  
RegOpenKeyExA  
RegCloseKey  
RegCreateKeyExA  
RegSetValueExA  
RegDeleteValueA  
RegQueryValueExA  
RegSetValueExW  
CryptGetHashParam  
CryptCreateHash  
CryptDestroyHash  
OpenProcessToken  
SetTokenInformation  
CryptHashData  
GetFileSecurityW  
OpenThreadToken

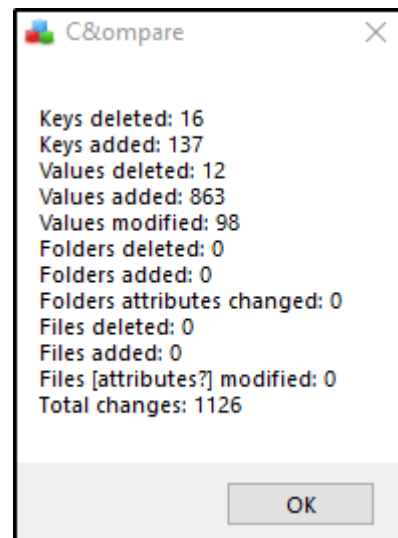
## השוואת שינויים ברג'יסטרי עם RegShot

### דרכי פעולה

- צילום (Snapshot) של הרג'יסטרי עם **RegShot** טרם הפעלת המאלוור.
- הרצת המאלוור, המתנה לסיום.
- צילום של הרג'יסטרי אחרי שהמאלוור פעל.
- השוואה בין שני הצילומים וחקירת שינויים שנעשו.

### תוצאות

סה"כ נעשו **1126** שינויים אחרי הרצת המאלוור. רוב השינויים לא קשורים להרצת המאלוור וידוע שיהיו רעשי רקע.



## פרק 2: ניתוח דינאמי

### מפתחות שנוספו

HKLM\SOFTWARE\WOW6432Node\Microsoft\Tracing\mal\_RASAPI32  
HKLM\SOFTWARE\WOW6432Node\Microsoft\Tracing\svchost\_RASAPI32  
HKU\S-1-5-21-2427678116-2504986317-361849973-1001\SOFTWARE\Locky

### שינויים במפתחות

כולם נמצאים באותו נתיב המתחיל:

**HKLM\SOFTWARE\WOW6432Node\Microsoft\Tracing**

Key	Value
\mal_RASAPI32\EnableFileTracing	0x00000000
\mal_RASAPI32\EnableAutoFileTracing	0x00000000
\mal_RASAPI32\EnableConsoleTracing	0x00000000
\mal_RASAPI32\FileTracingMask	0xFFFF0000
\mal_RASAPI32\ConsoleTracingMask	0xFFFF0000
\mal_RASAPI32\MaxFileSize	0x00100000
\mal_RASAPI32\FileDirectory	"%windir%\tracing"
\svchost_RASAPI32\EnableFileTracing	0x00000000
\svchost_RASAPI32\EnableAutoFileTracing	0x00000000
\svchost_RASAPI32\EnableConsoleTracing	0x00000000
\svchost_RASAPI32\FileTracingMask	0xFFFF0000
\svchost_RASAPI32\ConsoleTracingMask	0xFFFF0000
\svchost_RASAPI32\MaxFileSize	0x00100000
\svchost_RASAPI32\FileDirectory	"%windir%\tracing"

הגדרת הערכים הנ"ל ברג'יסטרי מבטלים יצירת לוגים על פעולות שהמאלוור מבצע כנראה לצורך לחשאיות.

### זיוף רשת

עם **ApateDNS**, אפשר לראות בקשות DNS שנעשו על ידי תוכנות זדוניות בדרך המהירה ביותר. **ApateDNS** מתחזה לתגובות DNS לכתובת IP שצוינה על-ידי המשתמש על-ידי האזנה לפורט 53 במחשב המקומי.

### דרכי פעולה

1. ניתוק המכונה מהרשת. (Host-only adapter)
2. הרמת שרת WEB במכונה אחרת. אשתמש במכונה שלי של Kali Linux.
3. הגדרת כתובת IP שתישלח בתגובות DNS: **83.130.134.238** – הכתובת של Kali Linux.
4. התחלת השרת.
5. הקשבה לפורט 53 עם ApateDNS.
6. בדיקת אילו בקשות נעשו.

### ממצאים

אחרי הרצת המאלוור, בזמן ההאזנה, ובזמן ששום תהליך אחר לא פועל והרשת מנותקת. ApateDNS מציג רשימה של בקשות DNS שנעשו:



## פרק 2: ניתוח דינאמי

Time	Domain Requested	14:51:23	xgnoecowowvpkod.tf	14:51:06	dlgjf.us
14:51:57	infvshob.yt	14:51:23	pvgshvtkuvqejx.be	14:51:06	uxfcf.eu
14:51:57	infvshob.yt	14:51:23	dlgjf.us	14:51:06	infvshob.yt
14:51:57	hiwnwhegpevq.uk	14:51:24	uxfcf.eu	14:51:06	xgnoecowowvpkod.tf
14:51:57	hiwnwhegpevq.uk	14:51:24	xgnoecowowvpkod.tf	14:51:06	xgnoecowowvpkod.tf
14:51:57	hiwnwhegpevq.uk	14:51:24	xgnoecowowvpkod.tf	14:51:06	dlgjf.us
14:51:57	hiwnwhegpevq.uk	14:51:24	xgnoecowowvpkod.tf	14:51:06	dlgjf.us
14:51:57	hiwnwhegpevq.uk	14:51:42	pvgshvtkuvqejx.be	14:51:06	uxfcf.eu
14:52:16	xgnoecowowvpkod.tf	14:51:42	dlgjf.us	14:51:06	infvshob.yt
14:52:16	pvgshvtkuvqejx.be	14:51:42	pvgshvtkuvqejx.be	14:51:06	infvshob.yt
14:52:16	xgnoecowowvpkod.tf	14:51:42	dlgjf.us	14:51:23	hiwnwhegpevq.uk

- המאלוור חוזר על עצמו ומגיש בקשות DNS בשביל כתובות אלה שוב ושוב.

בשילוב עם *Netcat* והאזנה לפורט 80, ניתן לראות בקשות HTTP שהמאלוור מבצע:

המאלוור מבצע בקשת פוסט ל:  
[infvshob.yr/main.php](http://infvshob.yr/main.php)  
המידע ב-POST Data לא קריא.

```
C:\Windows\system32>nc -l -p 80
POST /main.php HTTP/1.1
Host: infvshob.yt
Content-Length: 100
Connection: Keep-Alive
Cache-Control: no-cache

»Ω[ "5-ך%jÉ!!E&+N&σ(ε-ÀTÇKl110A↑..|-%ú-VΓγÜ&¹-M%xf|10ñ
```

## תפיסת קבצים שנמחקים אוטומטית

**CaptureBAT** כבר לא זמין ולא מצאתי הורדה קיימת, לכן אשתמש בכלי חלופי שמצאתי ב-GitHub הנקרא [CapturePy](#). [CapturePy](#) הוא כלי ניתוח שעושה עותק של כל הקבצים שנמחקו או שונו בתיקייה נתונה ותתי-תיקיות שלה אל תיקייה אחרת.

**שימוש:** `python capture-py.py <capture-directory> <save-directory>`

המאלוור יוצר ומוחק קבצים מתיקיית %TEMP% של המערכת. לכן הגדרתי לכלי לעקוב אחר התיקייה ולשמור קבצים שנמחקו בתיקיית `capture_files` על שולחן העבודה.

```
C:\Users\MalwareAnalysis\Desktop\Capture-Py-master> python capture-py.py
C:\Users\MALWAR~1\AppData\Local\Temp
C:\Users\MalwareAnalysis\Desktop\capture_files
```

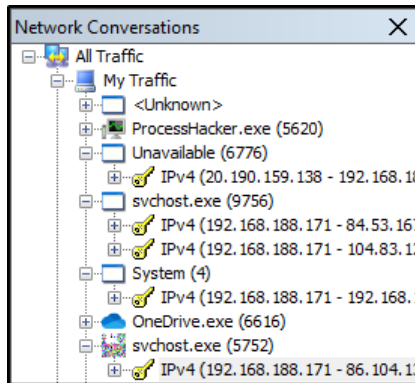
## תוצאות:

**5** קבצים נמחקו והועברו לתיקייה `Capture files`.

Name	Date modified	Type	Size
2021-02-14T16-16-37.824901-svchost.exe	14/02/2021 16:16	Application	180 KB
2021-02-14T16-16-37.831834-sysC4DE.tmp	14/02/2021 16:16	TMP File	180 KB
2021-02-14T16-16-52.597623-tmpaddon	14/02/2021 16:16	597623-TMPADDO...	11 KB
2021-02-14T16-18-23.519373-~DF6747B5763E6367F4.TMP	14/02/2021 16:18	TMP File	16 KB
2021-02-14T16-18-40.784714-~DFD47E783F10ED3C76.TMP	14/02/2021 16:18	TMP File	16 KB

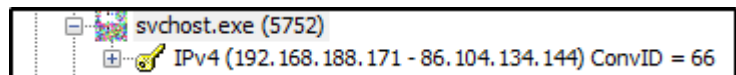


# Microsoft Network Monitor



Microsoft Network Monitor מאפשר לתפוס חבילות רשת רצות על המכונה. בחרתי בכלי על פני Wireshark, מפני שהוא יותר מסודר מבחינת תצוגת התעבורה. בשונה מ-Wireshark, MNM מציג את החבילות שנתפסו לפי התהליכים שיצרו את החבילות באופן מסודר בתצורת עץ. לכן, קל יותר לסנן חבילות ולבדוק חבילות שנתפסו מתהליכים שאנחנו רוצים לבחור.

לאחר הרצת המאלוור, אפשר לבחון את תעבורת הרשת של **svchost.exe**, תהליך הבן של המאלוור.



כל בין חצי דקה ל-40 שניות **svchost.exe** מנסה ליצור תקשורת TCP עם

ה-IP החשוד מאירלנד, **86.104.134.144**. התקשורת לא מצליחה והוא מנסה שוב ושוב ונתקע בלופ אינסופי.

Frame Number	Time Date Local Adjusted	Time Offset	Process Name	Source	Destination	Protocol	Description
194	12:54:03 15/02/2021	70.1452966	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:Flags=.....S., SrcPort=50072, DestPort=HTTP(80), PayloadLen=0, Seq=529121676, Ack=0, Win=65535 ( Negotiat...
197	12:54:04 15/02/2021	71.1464409	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #194]Flags=.....S., SrcPort=50072, DestPort=HTTP(80), PayloadLen=0, Seq=529121676, Ack=0,...
200	12:54:06 15/02/2021	73.1466457	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #194]Flags=.....S., SrcPort=50072, DestPort=HTTP(80), PayloadLen=0, Seq=529121676, Ack=0,...
251	12:54:18 15/02/2021	77.1615867	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #194]Flags=.....S., SrcPort=50072, DestPort=HTTP(80), PayloadLen=0, Seq=529121676, Ack=0,...
309	12:54:18 15/02/2021	85.1908868	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #194]Flags=.....S., SrcPort=50072, DestPort=HTTP(80), PayloadLen=0, Seq=529121676, Ack=0,...
312	12:54:24 15/02/2021	91.1595672	svchost.exe	86.104.134.144	192.168.188.171	TCP	TCP:Flags=...A.R., SrcPort=HTTP(80), DestPort=50072, PayloadLen=0, Seq=232744653, Ack=529121677, Win=64240
350	12:54:42 15/02/2021	109.4018735	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:Flags=.....S., SrcPort=50076, DestPort=HTTP(80), PayloadLen=0, Seq=2050645238, Ack=0, Win=65535 ( Negotia...
352	12:54:43 15/02/2021	110.4247452	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #350]Flags=.....S., SrcPort=50076, DestPort=HTTP(80), PayloadLen=0, Seq=2050645238, Ack=...
355	12:54:45 15/02/2021	112.4275635	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #350]Flags=.....S., SrcPort=50076, DestPort=HTTP(80), PayloadLen=0, Seq=2050645238, Ack=...
360	12:54:49 15/02/2021	116.4408092	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #350]Flags=.....S., SrcPort=50076, DestPort=HTTP(80), PayloadLen=0, Seq=2050645238, Ack=...
367	12:54:57 15/02/2021	124.4434320	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #350]Flags=.....S., SrcPort=50076, DestPort=HTTP(80), PayloadLen=0, Seq=2050645238, Ack=...
377	12:55:03 15/02/2021	130.4107519	svchost.exe	86.104.134.144	192.168.188.171	TCP	TCP:Flags=...A.R., SrcPort=HTTP(80), DestPort=50076, PayloadLen=0, Seq=684214923, Ack=2050645239, Win=64240
399	12:55:17 15/02/2021	144.1766400	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:Flags=.....S., SrcPort=50077, DestPort=HTTP(80), PayloadLen=0, Seq=3296713233, Ack=0, Win=65535 ( Negotia...
401	12:55:18 15/02/2021	145.1924749	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #399]Flags=.....S., SrcPort=50077, DestPort=HTTP(80), PayloadLen=0, Seq=3296713233, Ack=...
404	12:55:20 15/02/2021	147.2083348	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #399]Flags=.....S., SrcPort=50077, DestPort=HTTP(80), PayloadLen=0, Seq=3296713233, Ack=...
409	12:55:24 15/02/2021	151.2224917	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #399]Flags=.....S., SrcPort=50077, DestPort=HTTP(80), PayloadLen=0, Seq=3296713233, Ack=...
414	12:55:32 15/02/2021	159.2239447	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #399]Flags=.....S., SrcPort=50077, DestPort=HTTP(80), PayloadLen=0, Seq=3296713233, Ack=...
417	12:55:38 15/02/2021	165.1853153	svchost.exe	86.104.134.144	192.168.188.171	TCP	TCP:Flags=...A.R., SrcPort=HTTP(80), DestPort=50077, PayloadLen=0, Seq=1515214804, Ack=3296713234, Win=64240
430	12:55:38 15/02/2021	165.6201446	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:Flags=.....S., SrcPort=50078, DestPort=HTTP(80), PayloadLen=0, Seq=2321357085, Ack=0, Win=65535 ( Negotia...
432	12:55:39 15/02/2021	166.6292979	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #430]Flags=.....S., SrcPort=50078, DestPort=HTTP(80), PayloadLen=0, Seq=2321357085, Ack=...
435	12:55:41 15/02/2021	168.6603207	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #430]Flags=.....S., SrcPort=50078, DestPort=HTTP(80), PayloadLen=0, Seq=2321357085, Ack=...
441	12:55:45 15/02/2021	172.6612438	svchost.exe	192.168.188.171	86.104.134.144	TCP	TCP:[SynRetransmit #430]Flags=.....S., SrcPort=50078, DestPort=HTTP(80), PayloadLen=0, Seq=2321357085, Ack=...

Figure X: תצוגת MNM

## סיכום דברים / ממצאים

אחרי ניתוח המאלוור אלה הממצאים המסוכמים. המאלוור לא מממש את המטרות שלו. תקשורת בין המאלוור לבין מה שיכול להיות C&C Server לא מוצלחת, כנראה להורדת קבצים או משיכת פקודות שהמאלוור זקוק להם בשביל שיפעל כראוי. למרות שהמאלוור אינו פעיל ניתן היה להוציא מידע על מטרת המאלוור, ניסיון לתקשורת ובחינת תהליכים שהוא יוצר.

1. ברגע שהמכונה נגועה, המאלוור יכול להצפין את כל צורות הקבצים, ממסמכים, תמונות וסרטונים. הוא יכול להצפין נתונים, ואולי בעל אפשרות הפצה למחשבים אחרים באותה רשת. כדי לקבל את הקבצים בחזרה, התוקף מבקש תשלום בביטקוין. סימן היכר נוסף של Ransomware הוא שתינתן מגבלת זמן קצרה כדי לשלם את הכופר או שנאבד את הנתונים לתמיד. ראוי לציין, כי בזמן הרצה במכונה וירטואלית מקומית בהגדרות שונות וגם הרצה בארגז-חול מבוסס ענן כמו Any.Run, לא נראתה פעילות ולא הוצפנו קבצים, **נראה כי המאלוור לא פעיל יותר**.
2. מנסה ליצור תקשורת החוצה בבקשת POST עם IP 86.104.134.144 בפורט 80. ספציפית לכתובת: <http://86.104.134.144/main.php>, שמקורה באירלנד. (קוד ארץ: IE) כמו כן, קיימות בקשות DNS לכתובות שונות. בזמן הרצה על מכונה וירטואלית ובדיקה במנועים אחרים, לבקשה זו, אין תגובה מהשרת המבוקש, על כן המאלוור לא ממשיך בפעילותו.
3. יוצר קבצי TMP בתיקיית %TEMP% של מערכת ההפעלה ומוחק אותם לאחר שנעשה בהם שימוש. יוצר עותק של התהליך svchost.exe תחת אותה תיקייה, כנראה לתחזוקת המאלוור והרצה מחדש באם מערכת ההפעלה עשתה אתחול מחדש.
4. דרכי התמודדות במערכות הגנה ארגוניות:
  - א. שימוש ב-NAC: בקרת גישה לרשת (Network Access Control) מסייעת לארגונים ליישם מדיניות לשליטה במכשירים ובגישה של משתמשי קצה לרשת. **בהקשר למאלוור הנחקר** ניתן להגדיר מדיניות שלא תאפשר תקשורת עם ה-IP 86.104.134.144 וכן לכל בקשות ה-DNS שנמצאו בניתוח הנ"ל. דרך הגנה זו לבד כשלעצמה לא היעילה ביותר, לכן שילוב פתרון זה עם עוד דרכי הגנה יהיה יעיל ביותר.
  - ב. SIEM: פלטפורמת SIEM אוספת נתונים מפירוולים שעשויים להצביע על תקשורת מוצלחת עם דומיינים או כתובות IP. SIEM גם מזהה תוכנות זדוניות המשויות לדומיינים אלה וכולל תוכנת אנטי-ספאם המזהה קבצים שיכולים לגרום נזק לרשת הפנימית - והכל בזמן אמת ומסוכם בהתראת אבטחה אחת. על-ידי הצלבת מידע זה עם Indicators of Compromise (IoCs), אנליסטים יכולים לזהות ולהגיב לפעילות רשת זדונית - במיוחד תוכנות כופר (כמו התוכנה הנחקרת) - במהירות ובדייקנות רבה יותר.
  - ג. **בהקשר למאלוור הנחקר**, אנליסטים או האחראי יכולים לייצר כללים חדשים, שיתריעו כאשר תקשורת לדומיין החשוד באירלנד נעשתה החוצה ו/או פנימה. כמו כן, הוספת ההאשים של הקובץ לרשימה השחורה של מערכת ה-SIEM, תתריע אם הקובץ חדר לארגון.
  - ג. **ראה נספח א' בשביל דוגמה מעשית** לשימוש במערכת EDR. בהקשר לתוכנות זדוניות המועברות במוצר EDR או כל מוצר אבטחה ארגוני, אנליסטים יכולים להפעיל במהירות ובדייקנות דרכי תגובה. אם התוכנה הזדונית נמצאה כלא ידועה, כלומר חדשה או אין חתימת אנטי-וירוס זמינה, השלב הבא בתהליך התגובה צפוי להיות חקירה נוספת. אנליסט יכול להעביר את הקובץ החשוד בממשק המשתמש של ה-EDR לפלטפורמת ניתוח. משם ניתן לחקור ולהנדס לאחור את הקובץ ולבצע ניתוח סטטי מלא. ניתוח סטטי יציף לפני השטח את כל המבנים הקשורים למאלוור בקובץ, קשרים לפונקציונליות ידועות של תוכנות זדוניות, וכל טכניקות ההתחמקות שהמאלוור משתמש בהן. לאחר מכן ניתן לשלוח את הקובץ לניתוח נוסף בסביבה מאובטחת (Sandbox), אך הפעם עם סבירות גבוהה בהרבה להרצה מוצלחת מכיוון שטכניקות התחמקות התגלו בניתוח הסטטי. בסופו של דבר, אפילו עם גרסה זדונית לא ידועה, ניתן לנתח את האיום שהיא מציבה. עם מנוע חוקי

**YARA**, ניתן לבנות חוק YARA ולבדוק בפלטפורמות המתאימות אם החוק תואם. לאחר בדיקה מוצלחת, ויידוא שהחוק מזהה את האיום בקובץ החשוד, ניתן לייבא את חוק YARA לכלי הזיהוי, כולל מוצרי **EDR**, כך שבפעם הבאה שהתוכנה הזדונית הלא ידועה תכה, כלי הזיהוי יזהו אותה באופן מיידי.

ד. מניעת Open Mail Relay: קינפוג לא נכון של שרת אימיילים (**SMTP**), יכול להיות גן-עדן לספאמרים ותוקפים לארגון. ברמה בסיסית, **Open Mail Relay** הוא שרת **SMTP** המוגדר לאפשר לכל אחד באינטרנט לשלוח מיילים לארגון. שיטה זו מלאה חורי אבטחה ומהווה סיכון לארגון כך שמיילים מגורמים זדוניים יכולים להגיע לכל אחד בארגון. מספיק מייל אחד עם *תולעת / וירוס / טרואני / RAT* בשביל לגרום לארגון נזק רב בארגון בגלל מיס-קונפיגרציה של שרת **SMTP**.

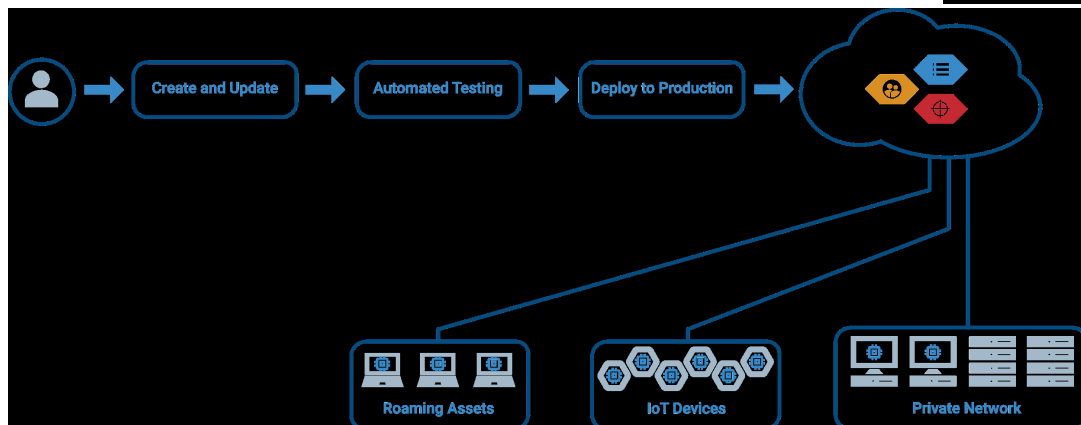
**בהקשר למאלוור הנחקר**: ניתן למתן סכנה זו בארגון ע"י קינפוג נכון של שרת **SMTP** בחברה. א. יצירת רשימה שחורה של כתובות IP שייחסמו באם התקבלה הודעה מהכתובת. שמירה על הרשימה מעודכנת, תמנע מיילים לא רצויים מכתובות זדוניות. ב. הפעלת **Reverse DNS Lookup**, בודק אם הכתובת IP של השולח מתאימה לשם ולדומיין שהוגשו ע"י ה-**SMTP**.

ג. הגבלת מספר חיבורים לשרת ה-**SMTP**, כולל מספר חיבורים המאופשר בו-זמנית, מספר חיבורים מקסימאלי, מספר חיבורים כולל. מיתון זה, יכול למנוע מתקפת ספאם ו-**DoS** על הארגון.

# נספח א': דוגמה להקמת EDR בענן, קביעת חוקי D&R, הרצת המאלוור במכונה אחרת וזיהוי בזמן אמת

לצורך דוגמה של שימוש במערכת הגנה ארגונית, אשתמש בשירות החינמי של [LimaCharlie](#), המאפשרת למשתמש ביתי לקבל ניסיון "Hands-On" על מערכת EDR ברמת Enterprise בחינם עד לשני משתמשי קצה.

הסבר על המבנה:



המערכת מבוססת ענן וכל ההתנהלות קורית דרך אפליקציית Web: [app.limacharlie.io](https://app.limacharlie.io)

- אתחיל ביצירת ארגון. הארגון הוא דמו ומקבל עד שני משתמשי קצה:

Organizations				
Name	Status	Errors	Version	Online / Quota
Robert's Incident Response	<span style="color: green;">●</span>	<span style="color: green;">✔</span>		1 / 2

- אחרי שהארגון הוקם, התקנתי "סוכן" על מכונת ווינדוס שאני רוצה לעקוב אחריה. המכונה נוספה במערכת וכעת מנוטרת ע"י LimaCharlie תחת הארגון הנ"ל:

Hostname	Tags	Status
desktop-v2h3e5o.localdomain	workstation endpoint windows10	<span style="color: green;">✔</span> <span style="color: green;">✔</span> <span style="color: green;">✔</span>

## יצירת חוקי Detection & Response (D&R)

### זיהוי ע"פ SHA256:

- נותן לחוק שם.
- בוחר סוג אינדיקטור (Hashes SHA256)
- בוחר מערכות הפעלה שהחוק יחול עליהן (Windows, Linux, MacOS)
- כותב את ה-IOC, במקרה זה, חתימת ההאש SHA256 של קובץ המאלוור -  
bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3

## סיכום דברים / ממצאים

New Rule

A detection and related response in YAML format. See the [documentation](#) for more details.

Rule Name	Expiration
Locky SHA256	YYYY-MM-DD HH:MM:SS

Basic Assisted

The simple rule builder will help you generate a complete detection and response rule. Once you have configured the rule, click Apply and the rule will be generated. Review the generated rule and click Create.

Indicator Type

Hashes (SHA256)

Platforms

✓ Windows ✓ MacOS ✓ Linux

Indicators

1	bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3
---	--

5. בחר את התגובה הרצויה: במקרה זה אני רוצה לבודד את המכונה מהרשת באם זוהה החוק ולקבל דיווח.

Response

☐ Kill ✓ Isolate ✓ Report

APPLY

### החוק שנוצר ופרטיו:

Name	Details	Status
Locky SHA256	Details	●

(שורות 1-6 בתבנית ה-Detection שנחתכו בצילום מגדירות את מערכות ההפעלה שהגדרנו קודם לכן.)

```
Detection:
- path: event/HASH
  case sensitive: false
  value: bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3
  op: is
- path: event/HASH
  case sensitive: false
  value: bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3
  op: is
op: or
events:
- CODE_IDENTITY

Response:
1 - action: task
2   command: segregate_network
3 - action: report
4   name: Locky SHA256
```

Figure XI: חוק D&R מלא

### זיהוי ע"פ בקשות DNS (Domains) וכתובות IP

אעבור על אותם שלבים, ואכניס את ה-loC's הנותרים בתור חוקי D&R. (התהליך דומה לכן אצמצם בצילומי מסך והסברים ואגיע לזיהוי בזמן אמת בעת הרצה).

← תמונה חלקית מחוק זיהוי בקשות DNS פועל ע"י זיהוי DOMAIN\_NAME בקטגוריית DNS\_REQUEST אשר הערכים שלה הם שמות הדומיין החשודים.

```
Detection:
33 | | | case sensitive: raise
34 | | | value: rupweuinytpmusfrdeitbeuknltf/main.php
35 | | | op: is
36 | | - path: event/DOMAIN_NAME
37 | | | case sensitive: false
38 | | | value: infvshob.yt
39 | | | op: is
40 | | op: or
41 | events:
42 | - DNS_REQUEST
43 | op: and
44 |

Response:
1 | action: report
2 | name: 'Locky: Malicious Domain Names'
```

← חלק מחוק זיהוי חיבורים לכתובות IP. פועל ע"י בדיקת אירוע NETWORK\_ACTIVITY בקטגוריית NETWORK\_CONNECTIONS אשר הערכים שווים לכתובות IP הזדוניים שסופקו.

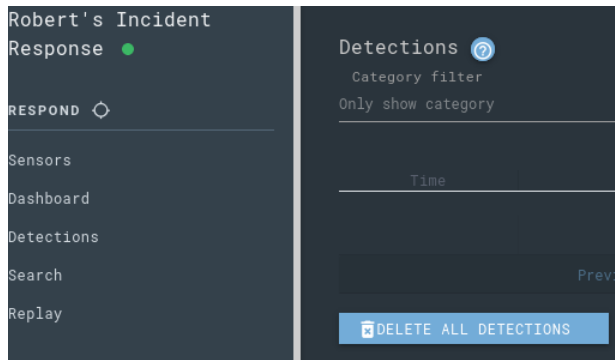
```
Detection:
8 | | | - path: event/NETWORK_ACTIVITY/?/IP_ADDRESS
9 | | | value: 86.104.134.144
10 | | | op: is
11 | | - path: event/NETWORK_ACTIVITY/?/IP_ADDRESS
12 | | | value: 86.104.134.144
13 | | | op: is
14 | | op: or
15 | events:
16 | - NETWORK_CONNECTIONS
17 | op: and
18 |

Response:
1 | action: report
2 | name: 'Malicious Locky IP (Country: IE)'
```

התגובות לאירועים אלה כרגע נשארות כ"דווח" בלבד, בלי בידוד מהרשת, ובלי הריגת התהליך בשביל קבלת כמה שיותר מידע ולתת למאלוור לרוץ על המכונה ללא הפרעות לצורך ניתוח מאוחר יותר.

#### כל החוקים שנוצרו מסודרים בלשונית ה-D&R Rules:

D&R Rules	
Name	
Locky SHA256	
Locky: Malicious Domain Names	
Malicious Locky IP (Country: IE)	



בשביל לבדוק את אמינות החוקים שנוצרו, התחלתי עם לשונית התראות ריקה ונקייה לחלוטין טרם הרצה:

Figure XII: לשונית התראות ריקה טרם הפעלה

## הרצת המאלוור במכונה ובדיקת אמינות החוקים שנוצרו

בנוסף לחוקי ה-D&R, ל-LimaCharlie יש אפשרות להתקין תוספים ממקורות שונים שתופסים פעילות חשודה בקלות כגון תוספי חוקי **Sigma** ו-**Soteria** נותנים חשיפה לחוקי **Sigma** ו-**Soteria** בקלות ובצורה מסודרת ועוד... התקנתי כמה מהם כדי לראות איך הם משפיעים / מזהים את המאלוור הנחקר. **כתוצאה מכך**, לשונית ה-**Detections** התמלאה בהתראות שלא אני יצרתי, אך ניתן ללמוד מהם המון על אופן הפעולה של המאלוור, כמו יצירת תהליך **svchost.exe** מתוך תהליך הורה חשוד, יצירת תהליכי מערכת מתיקיות חשודות, **svchost** ללא שורת פקודה (חריג), הרצת **svchost.exe** / **קובץ מערכת** מתיקייה חריגה (Temp).

**בהקשר לחוקים שיצרתי**, אלה עבדו והופיעו בלשונית ההתראות **בזמן אמת בעת הרצה**.

← תמונה שנלקחה מתוך מערכת **LimaCharlie**, בלשונית **Detections**. אלה התראות שהתקבלו בזמן אמת ומסודרות בסדר כרונולוגי ע"פ זמן ההתרחשות. מימין לעמודת הקטגוריה (חתוך בתמונה בגלל מקום וגודל תמונה) יש עמודה עם שם המכונה (**DESKTOP-V2H3E50.Localdomain**) ממנה ההתראה נוצרה. ניתן לראות את כל אחרי שהרצתי את המאלוור. אלה המסומנות בצהוב **הן התראות מחוקים שיצרתי קודם לכן**. התראות אחרות הן מתוספים אחרים ממקורות אחרים. בהמשך, אפתח את שתי ההתראות מהחוקים שיצרתי ואנתח אותם לעומק.

Time	Category
2021-02-17 14:02:06	Malicious Locky IP (Country: IE)
2021-02-17 14:01:35	Malicious Locky IP (Country: IE)
2021-02-17 14:01:04	Malicious Locky IP (Country: IE)
2021-02-17 14:00:34	Malicious Locky IP (Country: IE)
2021-02-17 14:00:02	Malicious Locky IP (Country: IE)
2021-02-17 13:59:31	Malicious Locky IP (Country: IE)
2021-02-17 13:59:00	Malicious Locky IP (Country: IE)
2021-02-17 13:58:29	Malicious Locky IP (Country: IE)
2021-02-17 13:57:47	Locky SHA256
2021-02-17 13:57:46	Suspect Svchost Activity
2021-02-17 13:57:46	Suspicious Svchost Process
2021-02-17 13:57:46	System File Execution Location Anomaly
2021-02-17 13:57:45	Locky SHA256
2021-02-17 13:57:45	00281-WIN-Svchost_Executing_from_Unusual_path
2021-02-17 13:57:45	00076-WIN-svchost_With_No_Commandline
2021-02-17 13:57:45	Windows Processes Suspicious Parent Directory
2021-02-17 13:57:45	00063-WIN-Suspect_Svchost_Parent_Process
2021-02-17 13:57:14	Locky SHA256

### Locky SHA256 D&R Rule

← לחיצה על אחת ההתראות תפתח חלון עם פרטים נוספים ועמוקים בנוגע לאופי ההתראה כגון, שם הקובץ, גודל, נתיב, זמן גישה לאירוע + זמן ההתראה שנוצרה בזמן UNIX, ההאש שהפעיל את ההתראה וגם MD5 ו-SHA1, בדיקה האם הקובץ חתום.

```
"Detect": {
  "author": "manalysis945@gmail.com"
  "cat": "Locky SHA256"
  "detect": {
    "event": {
      "ACCESS_TIME": 1613570039043
      "ATTRIBUTES": 0
      "CREATION_TIME": 1612896832570
      "ERROR": 0
      "FILE_INFO": "0.195.16.207"
      "FILE_PATH": "C:\\Users\\MalwareAnalysis\\Desktop\\Infected\\mal.exe"
      "FILE_SIZE": 184320
      "HASH": "bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3"
      "HASH_MD5": "b06d9dd17c69ed2ae75d9e40b2631b42"
      "HASH_SHA1": "b606aaa402bfe4a15ef80165e964d384f25564e4"
      "MODIFICATION_TIME": 1612896832683
      "SIGNATURE": {
        "FILE_CERT_IS_VERIFIED_LOCAL": 0
        "FILE_IS_SIGNED": 0
        "FILE_PATH": "C:\\Users\\MalwareAnalysis\\Desktop\\Infected\\mal.exe"
      }
    }
  }
}
```

← פרטים נוספים כגון מספר האירוע, זמן האירוע ב-UNIX, סוג האירוע (CODE\_IDENTITY), כתובת IP חיצונית ופנימית, Hostname.

```
"routing": {
  "arch": 2
  "did": ""
  "event_id": "5901794e-9b3f-4510-ad64-0962ca138e5c"
  "event_time": 1613570226269
  "event_type": "CODE_IDENTITY"
  "ext_ip": "83.130.134.238"
  "hostname": "DESKTOP-V2H3E50.localdomain"
  "iid": "37cb2377-4a92-4b66-802e-e472ee19dc49"
  "int_ip": "192.168.188.192"
  "moduleid": 2
  "oid": "02683e89-a309-48d1-ad98-d934cad990c1"
  "parent": "7bc544a26706c03a9c37423d602d20b2"
  "plat": 268435456
  "sid": "5272baf3-5de7-4883-ba8e-923278873328"
  "tags": [...]
  "this": "47f702c62e7792ddce497256602d20b2"
}
```



## Malicious Locky IP (Country: IE)

← התראה שנוצרה מחוק לתפיסת תקשורת עם IP המזוהה כזדוני מאירלנד בכתובת **86.104.134.144**. ניתוח מעמיק מגלה כי יוצר התקשורת הוא התהליך **svchost.exe** הזדוני. מנסה לתקשר עם ה-IP בפורט **80** בפרוטוקול **TCP4**.

```
"Detect" : {
  "author" : "manalysis945@gmail.com"
  "cat" : "Malicious Locky IP (Country: IE)"
  "detect" : {
    "event" : {
      "COMMAND_LINE" : "C:\Users\MALWAR~1\AppData\Local\Temp\svchost.exe"
      "FILE_IS_SIGNED" : 0
      "FILE_PATH" : "C:\Users\MALWAR~1\AppData\Local\Temp\svchost.exe"
      "HASH" : "bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3"
      "NETWORK_ACTIVITY" : [
        0 : {
          "DESTINATION" : {
            "IP_ADDRESS" : "86.104.134.144"
            "PORT" : 80
          }
          "IS_OUTGOING" : 1
          "PROTOCOL" : "tcp4"
          "SOURCE" : {
            "IP_ADDRESS" : "192.168.188.192"
            "PORT" : 49725
          }
          "TIMESTAMP" : 1613570288720
        }
      ]
    }
    "PARENT_PROCESS_ID" : 7708
    "PROCESS_ID" : 1944
    "USER_NAME" : "BUILTIN\Administrators"
  }
}
```

## WIN-Suspect Svchost Parent Process-00063

← לצורך הדגמה, אתעמק גם בהתראה זו שנוצרה ע"י טריגר מתוך תוסף הנותן גישה לכללי **Sigma** לחשיפת מאלווריס ואירועים זדוניים פוטנציאליים. קטגוריית **Suspect Svchost Parent Process** מתארת מצב, כמו השם שלה, בו תהליך **svchost.exe** נוצר מתהליך-אב **svchost.exe** לא צריך לפעול מתוך שום תהליך אחר חוץ מ-**sercives.exe** כשהתראה נוצרת מזיהוי תהליך-אב חשוד.

```
"Detect" : {
  "author" : "_soteria-rules[bulk][lock][secret][segment]"
  "cat" : "00063-WIN-Suspect_Svchost_Parent_Process"
  "detect" : {
    "event" : {
      "BASE_ADDRESS" : 4194304
      "COMMAND_LINE" : "C:\Users\MALWAR~1\AppData\Local\Temp\svchost.exe"
      "FILE_IS_SIGNED" : 0
      "FILE_PATH" : "C:\Users\MALWAR~1\AppData\Local\Temp\svchost.exe"
      "HASH" : "bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3"
      "MEMORY_USAGE" : 6430720
      "PARENT" : {
        "BASE_ADDRESS" : 4194304
        "COMMAND_LINE" : "C:\Users\MalwareAnalysis\Desktop\Infected\mal.exe"
        "FILE_IS_SIGNED" : 0
        "FILE_PATH" : "C:\Users\MalwareAnalysis\Desktop\Infected\mal.exe"
        "HASH" : "bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3"
        "MEMORY_USAGE" : 6381568
        "PARENT_ATOM" : "e621fb38115bf770501064dc602d20b2"
        "PARENT_PROCESS_ID" : 6792
        "PROCESS_ID" : 7708
        "THIS_ATOM" : "7bc544a26706c03a9c37423d602d20b2"
        "THREADS" : 3
        "TIMESTAMP" : 1613570226064
        "USER_NAME" : "BUILTIN\Administrators"
      }
    }
  }
}
```

תהליך-האב ממנו נוצר **svchost.exe** הוא **קובץ המאלוור הזדוני**. יותר למטה בחלון ההתראה מופיעים פרטים על המערכת ממנה הגיעה ההתראה וגם **לינק** לפרטים נוספים (Documentation) על הקטגוריה. בלינק יש הסבר מפורט על ההתראה שהתקבלה וקישורים לשימושים ב-**MITRE-ATT&CK**.