אתגר להב 433 סייבר

שלב ראשון – RIDDLE.WEBSITE

השלב הראשון של האתגר הוא אתר אינטרנט אשר אנחנו מתבקשים "לפרוץ" אליו. מבט חטוף בקוד ה-HTML של הדף מראה את קטע הקוד הבא:

```
$(function() {
    $("input[type=button]").on("click", function() {
        i = $("#l").val();
        j = $("#p").val();
        if (i == "PoliC3") {
            if (md5(i + "allwa7" + j) == "8e16941e6d51be408459221a1c905eda") {
                location.href = "/" + i + "allwa7" + j;
            }
        }
    });
});
```

שם המשתמש הוא בבירור "PoliC3". הסיסמה צריך לקיים את התנאי הבא: אם מפעילים את פונקציית ההאש MD5 על הביטוי ____PoliC3allwa7 (כאשר השטח הריק הוא הסיסמה) התוצאה צריכה להיות "8e16941e6d51be408459221a1c905eda".
על מנת לפתור זאת, יש לכתוב/להוריד ברוטפורס ל-MD5. לאחר זמן קצר של הרצה נמצאה הסיסמה: "s4u", ובכך הביטוי מושלם ל "Police always for you". התחברות עם הנתונים הללו מביא אותנו לדף:
http://riddle.website/Polic3alwa7s4u

Congratulations on passing the 1st step, your first code word is: Knowledge

Ar3 y00 r34dy f0r th3 n3xt st3p ?

Go to the next step

לחיצה על "Go to the next step" מורידה קובץ EXE שנקרא "okayletsplay.exe". אנטי וירוסים מסוימים כגון Windows defender מחליט שהוא וירוס, ולכן לפעמים יש צורך לכבות אותו לפני ההורדה.
הרצת strings (חילוץ כל המחרוזות מקובץ ההרצה) על הקובץ מביאה מספר מחרוזות חשודות:

```
HELLO
police
CSRF
aHR0cHM6Ly9pbWcubWVtZWNkbi5jb20vaGFja2VyLWhhbmtfb180OTQzMDEuanBn
R U GOOD ENOUGH?
0kayletsplay
U3RhbVN0cmluZw==
Y31iZXJ0ZWNo
bnVsbA==
SHA1ME
GET / HTTP/1.0 Host:
cm1kZGxlLg==
http://google.com
http://facebook.com
aHR0cDovL211ZG1hLmdpcGh5LmNvbS9tZWRpYS8zbzZadDd1bGJvdnY1NWN1aHEvZ21waHkuZ21m
aHR0cHM6Ly9tZWRpYS50ZW5vci5jby9pbWFnZXMvM2YwYTA0YzY2NzgwYWM2YTNjN2RmZTU4NmRmMDVhNjkvdGVub3IuZ21m
aHR0cHM6Ly9tZWRpYS5naXBoeS5jb20vbWVkaWEvd2VYODV4bU1SZTZSTy9naXBoeS1kb3duc2l6ZWQtbGFyZ2UuZ21m
d2Vic2l0ZQ==
L3R1cnRsM20zLw==
Q3JhY2ttZS51eGU
73 68 61 31 6d 33 66 30 34 6c 69 66 33
```

== מרמז לרוב על Base64, והנה התוצאות של הרצת Base64 decode על כל הביטוים החשודים:

https://img.memecdn.com/hacker-hank_o_494301.jpg
StamString
cybertech
null
riddle.
http://media.giphy.com/media/3o6Zt7elbovv55cehq/giphy.gif
https://media.tenor.co/images/3f0a04c66780ac6a3c7dfe586df05a69/tenor.gif
https://media.giphy.com/media/weX85xmMRe6RO/giphy-downsized-large.gif
website
turtl3m3/
Crackme.exe

אם נמיר את הביטוי האחרון מ-Ascii לטקסט נקבל "sha1m3f04lif3".

כפי שאפשר לראות, בין שלל ה-GIFים המשעשעים, יש כתובת אינטרנט: "riddle.website/turtl3m3/Crackme.exe" אשר מובילה לשלב הבא.

שלב 3 – CRACKME.EXE

בשונה מהקובץ הקודם, Crackme.exe תופס כ-4MB של זיכרון. הפעם strings לא נותן לנו את הפתרון אלא רמז. הוא כולל הפניות לפונקציות python ולקבצי python. כלומר, סביר להניח שמדובר בתוכנה אשר כוללת את python בתוכה ומריצה קוד.
נוסף על כך, בשונה מהקובץ הקודם, הפעם יש לקובץ אייקון. אם נשתמש בפיצ'ר "חפש לפי תמונה" בגוגל תמונות, נגלה שמדובר בספריית PyInstaller שממירה קבצי python ל-EXE.

```
win32con(
win32evtlogutil(
winerror(
mstruct
mpyimod01_os_path
mpyimod02_archive
mpyimod03_importers
spyiboot01_bootstrap
schallenge
bMicrosoft.VC90.CRT.manifest
b_ctypes.pyd
b_hashlib.pyd
b_socket.pyd
b_ssl.pyd
bbz2.pyd
bchallenge.exe.manifest
bmsvcm90.dll
bmsvcp90.dll
bmsvcr90.dll
bpython27.dll
bpywintypes27.dll
bselect.pyd
bunicodedata.pyd
bwin32api.pyd
bwin32evtlog.pyd
opyi-windows-manifest-filename challenge.exe.manifest
zout00-PYZ.pyz
python27.dll
```

עוד פרט מעניין בנוגע לקובץ הוא שאם נפתח אותו ב-Hex Editor נגלה מחרוזות מעניינות בתחילת הקובץ, במקום שבדרך כלל מופיעות המחיצות של הקובץ: UPX, UPX1, LOL2. חיפוש קצר בגוגל יגלה ש-UPX הוא Packer מפורסם, שנתמך דרך PyInstaller באופן אוטומטי. המשך שיטוט באינטרנט יגלה כי שמות המחיצות הן UPX0, UPX1, UPX2... ולכן מתגלה כי LOL2 הוא ניסיון למנוע מהפקודה "upx -d" (decompress) לעבוד.

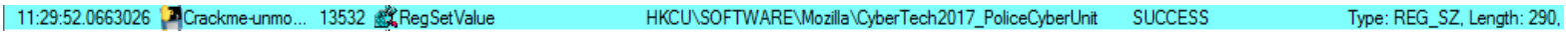
מכאן ישנם מספר פתרונות אפשרים. פתרון ראשון הוא לחלץ את קבצי ה-Python מהקובץ ולהבין איך הם עובדים.
פתרון שני הוא לנסות לנתר את התוכנה (באמצעות Procmon למשל) ולהבין מה היא עושה.

פתרון ראשון – PROC MONITOR

אם נריץ את Procmon.exe במקביל לתוכנה, נגלה 1125 אירועים שקרו במהלך הריצה. ובפרט:

- קריאה רבה מהקובץ עצמו. הדבר הגיוני מכיוון שגם מדובר ב-Packer שצריך לפענח את עצמו, וגם מדובר ב-PyInstaller שצריך לחלץ את Python.
- שמירת קבצי ההתקנה של Python לתקייה זמנית ב %appdata%/temp ומחיקתם בסיום הריצה. הדבר מעניין, אולם הוא לא מחלץ לשמה את הקוד עצמו שרץ, ולכן אין זה עוזר לנו.

אולם מה שאנחנו מחפשים, היא כתיבה תמימה לרגיסטרי אשר במבט ראשוני ואף שני אנו כנראה נחמיץ:



שם הערך שאליו כותבים לא מתיר ספקות בנוגע לכך שהכתיבה קשורה לאתגר. ואכן, הערך שנכתב הוא לא אחר מאשר Base64:

d2VsbCBkb25lIDtdLFN0ZXBDb2RlOmlzICxOZXh0TGluaztodHRwczovL3d3dy5kcm9wYm94LmNvbS9zaC9jcHJvdGl6aTAyNmc3MWYvQUFBLWxlbnN6dmlLQkJ5cW1tT2FtRVNPYT9kbD0w

שערכו הוא:

well done;], StepCode: is ,NextLink;https://www.dropbox.com/sh/cprotizi026g71f/AAA-lHnszviKBByqmmOamESOa?dl=0

פתרון שני – חילוץ קבצי ה-PY

חיפוש קצר באינטרנט ימצא את pyinstxtractor.py. כשנריץ אותו על הקובץ, נקבל את הקבצים הבאים:

Name	Date modified	Type	Size
out00-PYZ.pyz_extracted	2/3/2017 13:25	File folder	
_ctypes.pyd	2/3/2017 13:25	PYD File	46 KB
_hashlib.pyd	2/3/2017 13:25	PYD File	449 KB
_socket.pyd	2/3/2017 13:25	PYD File	24 KB
_ssl.pyd	2/3/2017 13:25	PYD File	654 KB
bz2.pyd	2/3/2017 13:25	PYD File	42 KB
challenge	2/3/2017 13:25	File	1 KB
challenge.exe.manifest	2/3/2017 13:25	MANIFEST File	1 KB
Microsoft.VC90.CRT.manifest	2/3/2017 13:25	MANIFEST File	2 KB
msvcm90.dll	2/3/2017 13:25	Application extens...	240 KB
msvcp90.dll	2/3/2017 13:25	Application extens...	383 KB
msvcr90.dll	2/3/2017 13:25	Application extens...	249 KB
out00-PYZ.pyz	2/3/2017 13:25	PYZ File	1,017 KB
pyiboot01_bootstrap	2/3/2017 13:25	File	7 KB
pyimod01_os_path	2/3/2017 13:25	File	3 KB
pyimod02_archive	2/3/2017 13:25	File	11 KB
pyimod03_importers	2/3/2017 13:25	File	19 KB
pyi-windows-manifest-filename challen...	2/3/2017 13:25	MANIFEST File	0 KB
python27.dll	2/3/2017 13:25	Application extens...	947 KB
pywintypes27.dll	2/3/2017 13:25	Application extens...	60 KB
select.pyd	2/3/2017 13:25	PYD File	11 KB
struct	2/3/2017 13:25	File	1 KB
unicodedata.pyd	2/3/2017 13:25	PYD File	181 KB
win32api.pyd	2/3/2017 13:25	PYD File	43 KB
win32evtlog.pyd	2/3/2017 13:25	PYD File	23 KB

מבין הקבצים הללו, challenge נשמע מעניין. ננסה לפתוח אותו, והפלא ופלא, זה קובץ Python.

```
from _winreg import *
import base64
def write2reg(dirname,keyname,data):
    try:
        xReg = ConnectRegistry(None,HKEY_CURRENT_USER)
        key = CreateKey(HKEY_CURRENT_USER, "Software\\" +dirname)
        bKey = OpenKey(xReg, r"Software\\" +dirname,0,KEY_WRITE)
        SetValueEx(bKey,keyname,0, REG_SZ, data)
    except Exception as e:
        print e
    CloseKey(bKey)

write2reg("Mozilla","CyberTech2017_PoliceCyberUnit",base64.b64encode('well done ;],StepCode:is ,NextLink;https://www.dropbox.com/sh/cprotizi026g71f/AAA-lHnszviKBByqmmOamESOa?dl=0'))
NUL
```

הגענו לאותה התוצאה.

שלב 4 – התמונה הארורה

בשלב הזה קרו המון פאשלות. לדוגמה, תמונה נוספת שנקראת Pickle.jpg, שכביכול אמורה לרמז על כך שיש צורך להשתמש ב-pickle. אולם, התמונה נמחקה מספר ימים אחרי. זאת ועוד, התמונה השתנתה לפחות פעמים במהלך החודש האחרון, ולכן השערתי היא שבמשך שבועות הופיעה התמונה המקורית במקום התמונה של האתגר. מביך.

הקישור מביא אותנו לתקיית Dropbox שנקראת בשם "Tr1Hard3r". נתעלם מקובץ ההבהרה (שנוצר כי אף אחד לא הצליח לעבור את התמונה בזמן שהופיעה התמונה המקורית לכאורה). בתקייה יש קובץ תמונה הנקרא "l0v3m3.jpg" ומופיע בו הסמל של היחידה. נוריד אותו.

יש מספר אפשרויות להחבאת מידע בתוך קובץ תמונה, והנה הנפוצות שבהן:

- עריכת נתוני הצבעים של התמונה. לדוגמה, לשנות את הביט האחרון של כל פיקסל.
- הכנסת מידע בתוך רווחים במבנה התמונה, בצורה כזאת שהפורמט עדיין תקין.
- הכנסת מידע לאחר הפקודה שאומרת למפרש להפסיק לקרוא.

לפי ויקיפדיה, רצף התווים שמסמן End of File עבור קובץ JPEG הוא FF D9. אם נחפש את התווים הללו בתמונה נגלה שיש עוד מידע אחריו, ובפרט הטקסט "Rar!", שהוא, כצפוי, החתימה של קובץ Rar.

אם נשנה את הסיומת ל-rar, נגלה שלקובץ יש סיסמה. אם נריץ ברוטפורס, עם סיסמאות נפוצות, נגלה שהסיסמה היא "pa$$word".
הערה: החילוץ לא יעבוד ב-7Zip אם לא נוריד את כל המעטפת של התמונה לפני. הקובץ צריך להתחיל בRar

בתוך ה-Rar יש שני קבצים. ה-QRCODE מוביל לשלב הבא - http://q-r.to/baibxk


QRLink2pB.png

StepCode.bmp

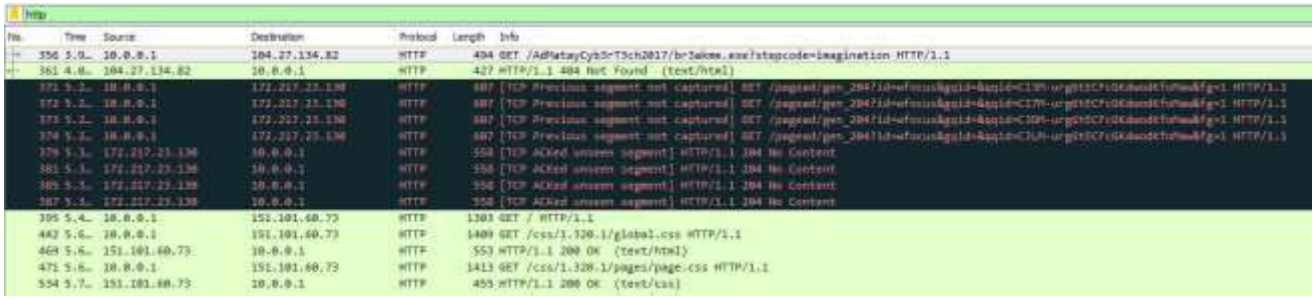
נכנסים ל-QRCODE, ומגיעים לקישור הבא: http://riddle.website/Tr7Hard3r/unrarme.rar . אבוי! אנו מקבלים דף 404.

הכתובת של הדף היא Tr7Hard3r, אולם הכותרת בשלב הקודם הייתה Tr1Hard3r. לאחר תיקון הכתובת, הקובץ uname.rar מורד. הקובץ כולל את challenge.cap, שהוא קובץ הכולל תעבורת רשת שהוסנפה. אולם כשמנסים לחלץ את הקובץ מתגלה כי יש לקובץ סיסמה. הפעם, ברוטפורס על סיסמאות נפוצות לא יעזור. זוכרים שבשלב 2 הצלחנו להוציא את הביטוי "sha1m3f04lif3" מ-Ascii? אז אם מריצים על זה SHA1 מקבלים את הסיסמה לקובץ.

ניתוח של הקובץ לפי סוג הפרוטוקול שהשתמשו בו מוביל למידע הבא:

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	5711	100.0	3142798	2243 k	0	0	0
Ethernet	100.0	5711	2.5	79954	57 k	0	0	0
Internet Protocol Version 4	100.0	5709	3.6	114180	81 k	0	0	0
User Datagram Protocol	12.0	686	0.2	5488	3917	0	0	0
Dropbox LAN sync Discovery Protocol	0.1	5	0.0	1010	720	5	1010	720
Domain Name System	11.7	666	1.8	56754	40 k	666	56754	40 k
Data	0.3	15	0.1	2977	2125	15	2977	2125
Transmission Control Protocol	88.0	5023	91.7	2881806	2057 k	4209	2142330	1529 k
Secure Sockets Layer	2.9	164	5.8	181596	129 k	153	159703	113 k
Malformed Packet	0.0	1	0.0	0	0	1	0	0
Hypertext Transfer Protocol	11.6	660	57.5	1807976	1290 k	531	627518	447 k
Portable Network Graphics	0.2	11	1.5	48068	34 k	11	50943	36 k
Media Type	0.3	15	29.9	940223	671 k	15	283485	202 k
Line-based text data	0.3	17	68.1	2140246	1527 k	17	342056	244 k
JPEG File Interchange Format	0.5	27	12.4	390841	278 k	27	398788	284 k
JavaScript Object Notation	0.4	22	4.3	136656	97 k	22	68341	48 k
Compuserve GIF	0.6	37	0.3	8714	6220	37	9715	6934
Data	0.0	2	0.0	432	308	2	432	308

נסנן לפי HTTP, וכבר בבקשה הראשונה נמצא את מבוקשינו:



הכתובת לשלב הבא היא: http://riddle.website/AdMatayCyb3rT3ch2017/br3akme.exe?stepcode=imagination
אשר מורידה קובץ EXE. שאר קובץ ההסנפה כולל תעבורה של גלישה סטנדרטית באינטרנט.

כרגיל, נתחיל בלהריץ strings על הקובץ.

```
aHR0cHM6Ly9yaWRkbGUud2Vic210Z59uMHQ3aGVyaWdodC9maW5hbC51eGU=
const int RL = 32;
const int StepCodeLen = 3;
char StepCode[3] = "dq";
char Rslt[32] = "crpm4.(mf^bgb)qb`obob+U.kji.q2-";
char key[] = "52436175362535632475344226353127 26";
int i;
for (i = 0; i < RL - 1; i++)
        Rslt[i] = Rslt[i] + ((key[i]) - '0');
        if (i < StepCodeLen - 1)
                StepCode[i] = StepCode[i] + ((key[i]) - '0');
                StepCode[i] = StepCode[i] * ((key[i]) - '0');
return 0;
```

ביטוי ה-Base64 שמצאנו מוביל לכתובת https://riddle.website/n0t7heright/final.exe
אם נריץ את קוד ה-C, נקבל שהמשתנה Rslt שווה בסוף ההרצה ל /http://riddle.website/Y0mpl3t3 . המשתנה Stepcode לעומת זאת הוא גיבריש. אולם, אם מורידים את השורה הלפני אחרונה, מקבלים שהוא שווה ל "is".

הקישור השני מוביל לדף סיום האתגר, ונותן לנו את ה-Stepcode הבא: "Infinity".

מבחינת Step codes שאספנו:

- בשלב 1 (אתר) קיבלנו את **Knowledge**.
- בשלב 2 (okayletsplay.exe) לא קיבלנו כלום.
- בשלב 3 (crackme.exe) קיבלנו את **is**.
- בשלב 4 (תמונה) קיבלנו את **but**.
- בשלב 5 (הסנפה) קיבלנו את **imagination.**
- בשלב 6 (breakme.exe) קיבלנו את **Infinite**.

היה אמור לצאת כנראה משפט באנגלית, אבל לא יצא.