סטגנוגרפיה בשני שקל.

מאת אפיק קסטיאל (cp77fk4r)

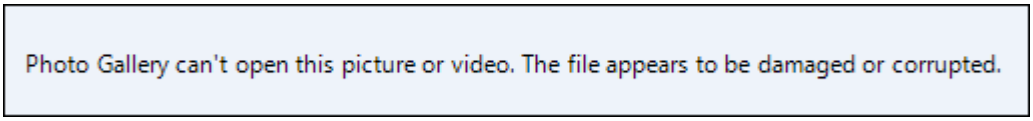
הקדמה

מאמר זה אינו מאמר המשך למאמר הקודם שכתבתי בנושא, במאמר הקודם הצגתי דרכים ושיטות לבצע פעולות סטגנוגרפציה בתמונות באופן חכם על-ידי ניצול תכונותיה של העין האנושית. במאמר זה אבצע סקירה נרחבת על מספר שיטות סטגנוגרפיה שונות להחבאת מידע (מידע יכול להיות מחרוזת קצרה ואף קובץ בר-הרצה שלם!) בתוך אובייקטי תמונה דיגיטלים. אבצע זאת תוך כדי הצגה של דוגמאות ואסביר את החסרונות והיתרונות בכל שיטה ושיטה.

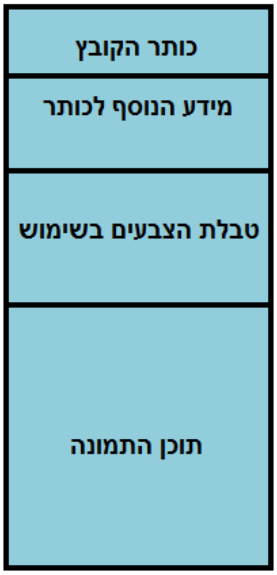
מבנה הקובץ

כאשר מדובר בסטגנוגרפיה מבוססת תמונה דיגיטלית הרעיון פשוט מאוד, אך עדיין, בכדי להבין אותו אנו צריכים להבין קודם כל איך מערכת ההפעלה שלנו מציגה ומתייחסת לתמונות תמונות.

כמו שאתם בוודאי יודעים, קיימים מספר רב מאוד סוגים ("פורמטים") של קבצי תמונה- או קבצים בכלל. פורמטים מוכרים הם BMP, JPG, PNG, GIF ועוד. אם תשימו לב, תראו שאותה התמונה, תשקול שונה בכל פורמט, ממה ההבדל נובע? מפני שכל פורמט מאחסן את פרטי התמונה באופן שונה. אני לא אכנס כאן על המבנה של כל פורמט ופורמט, אך חשוב שתבינו כי מבנה של תמונה- או בעצם, כמעט כל קובץ סטנדרטי שתפגשו יהיה בנוי ממספר חלקים מרכזיים. פורמט הקובץ אינו נקבע על פי הסיומת של אותו הקובץ- סיומת זאת נועדה להגדיר למערכת ההפעלה באיזה פעולה לנקוט כאשר מריצים את הקובץ. אז מה קובע את פורמט הקובץ? המבנה שלו כמובן. ניתן להבחין בכך- כאשר משנים סיומת של קובץ (למשל "ZIP") לפורמט שונה- לדוגמה: "JPG", ה-ICON שלו אומנם ישתנה ל-ICON של תמונה, ואף אם תלחצו כאת על הקובץ, מערכת ההפעלה תדע לשפוך את תוכנו לתוך העורך הגראפי שנקבע כברירת מחדל לקבצי "JPG", אך שימו לב, מיד לאחר טעינת העורך תופיע שגיאה מטעמו שתקבע כי אין מדובר בפורמט תקין של קובץ "JPG" - או כי הוא פגום:



אז מה זאת אומרת "מבנה הקובץ"? בכדי להבין זאת, נוכל לקחת כדוגמה את המבנה של קבצי התמונה הפשוטים ביותר, קבצי ה-Bitmap (קבצי "BMP"). מבנה קובץ ה-Bitmap הסטנדרטי במערכת ההפעלה Windows נראה כך:



כותר הקובץ (ה-"Header") - יגיד למערכת ההפעלה באיזה סוג קובץ מדובר, במערכות חלונאיות נפגוש כאן לרב את המחרוזת "424D" שאומרת "BM".

מידע הנוסף לכותר (ה-"Bitmap Information" או ה-"DIB header") - יכלול בתוכו מידע על התמונה, כגון אורך ורוחב, משקלו של התוכן, יחס ביטים/פיקסלים וכו'.

טבלת הצבעים בשימוש (ה-"Color palette") - תכלול את רשימת הצבעים הזמינים לשימוש בתמונה ועל-פיה יקבעו הצבעים בתמונה (לא מיקומיהם!)

תוכן התמונה (ה-"Bitmap data") - יכלול את תוכן התמונה, פיסקל אחר פיקסל מסודרים על פי תבנית המוכרת כ-"Raster scan".

למידע נוסף על מבנה קובץ ה-Bitmap תוכלו לקרוא בקישור הבא:

http://en.wikipedia.org/wiki/BMP_file_format#Color_palette

עד פה הסברנו מהו מבנה של קובץ וראינו את הדוגמה של קבצי BMP, חשוב לדעת שלכל קובץ סטנדרטי יש מבנה. מבנה זה נחוץ לנו בכדי שנוכל לבצע יישור קו על פיו יכתבו תוכניות שידעו לעבוד עם הקבצים האלו. בדיוק כמו בפרוטוקולי תקשורת.

החבאת מחרוזת בתמונה

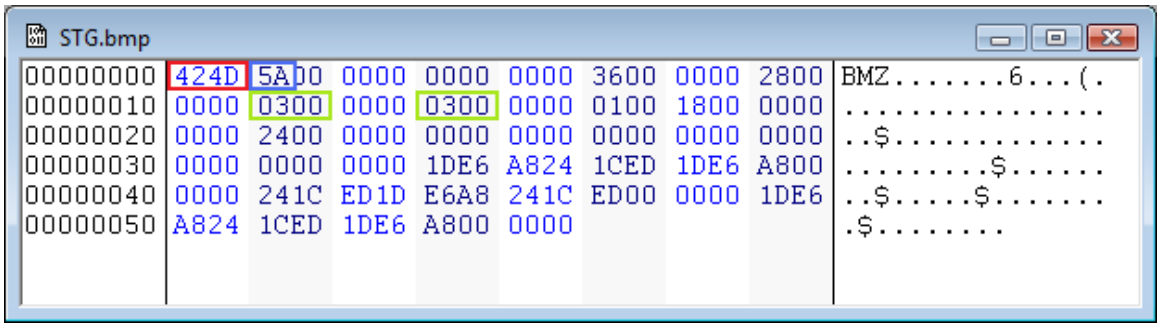
כמו שראינו- בכדי שהעורך הגראפי שלנו ידע לפענח את המידע השמור בתמונה, עליה לעמוד בסטנדרט מסויים, אם נבצע חריגה מאותו הסטנדרט- העורך הגראפי שלנו לא ידע להציג את המידע החורג. כמובן שבמידה ונבצע חריגה ב"קטעים קריטיים" סביר להניח כי העורך הגראפי שלנו לא ידע איך לטעון את תוכן התמונה.

מה אלו אותם "קטעים קריטיים"? - ברב המקרים מדובר בקטעים המגדירים את מבנה הקובץ- אם נקח את הדוגמה הקודמת- הרי שמדובר בקטעי ה-Header וה-DIB Header.

לצורך ההמחשה, יצרתי תמונה בת תשעה פיקסלים:

(כמובן שהתמונה מוגדלת)

במידה ונפתח אותה בעזרת עורך-הקסדצימאלי נוכל לראות את המידע הבא:

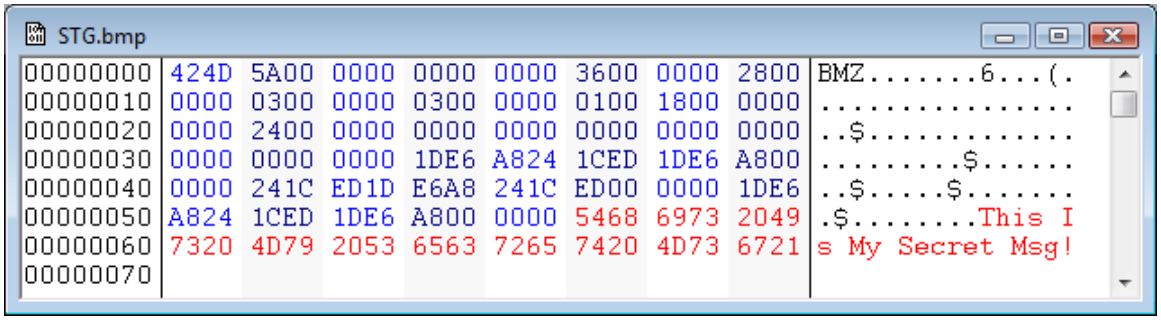


שימו לב לחלקים המסומנים:

- **הבלוק המסומן באדום** - כמו שכבר אמרנו, מסמן סוג הקובץ.
- **הבלוק המסומן בכחול** - אומר למערכת מהוא גודל תוכן הקובץ (5A = 90 בתים).
- **שני הבלוקים המסומנים בירוק** – מסמנים את האורך והרוחב של התמונה (3x3 = 9 פקסלים).

מכאן ניתן ללמוד כי כאשר העורך הגראפי שלנו טוען את התמונה, הוא לומד על פי הנתונים ב-Header שלה את גודלה, ולכן, במידה ונוסיף מידע לאחר הגודל שנקבע- הוא לא יוצג בתמונה.

בכדי לנסות זאת- פשוט מאוד, פיתחו את התמונה שוב בעזרת עורך-הקסדצימאלי ותוסיפו את המחרוזת "This is my Secert Msg!" בסופה, ותשמרו:



במידה ונפתח את הקובץ בעזרת העורך הגראפי שלנו, לא נוכל להבחין בשינוי שבוצע:

ולמה זה? מפני שכמו שראינו קודם לכן- העורך הגראפי שלנו לא באמת מוודא מהוא גודל תוכן הקובץ, אלה מקבל את המידע הזה מהנתונים שנקבעו לו ב-Headers של הקובץ. ולכן, כל מה שנכתב לאחר מכן- פשוט לא קיים מבחינתו.

החבאת תמונה בתמונה

הרעיון שהצגנו בפרק הקודם יכול לעבוד מאוד טוב, אך במידה ונפתח את התמונה בעזרת עורך טקסט פשוט נוכל להבחין מיד במידע המוחבא. בכדי לפתור זאת, נוכל בקלות מאוד להחביא את המידע הנ"ל בתוך תמונה, ואת התמונה להחביא בתוך תמונה תמימה נוספת.

לדוגמה, ניצור תמונה כזאת:

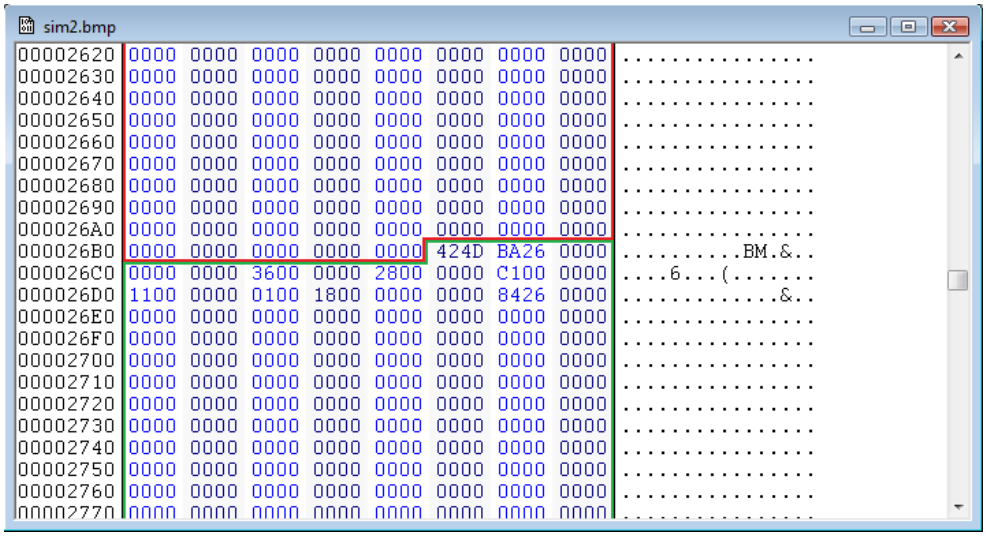
נוכל באותו אופן כמו שהכנסנו את הטקסט לתמונה- להכניס את התמונה הזאת.

מהלך הביצוע פשוט מאוד- נעתיק את תוכן התמונה ונעתיק אותו לתוך קובץ התמונה שנבחר. במידה ונרצה להחביא את התמונה בתמונה הבאה:

אופן מהלך ההחבאה יתבצע כך:
נפתח את התמונה בעת התוכן הסודי (לצורך העניין- תרשים של הכור הגרעיני באיראן) בעזרת עורך-הקסדצימאלי. נעתיק את כלל המידע ונדביק אותו לאחר תוכן הקובץ של התמונה התמימה.

כמובן ששוב- לא יהיה שום הבדל ויזואלי בין התמונה התמימה המקורית לבין התמונה התמימה שמכילה את התוכן הסודי שלנו. ומפני שהפעם הכנסנו את תוכן התמונה ולא טקסט מפורש ("Clear-text"), במידה וגורם עויין יפתח את התמונה בעזרת עורך-הקסדצימאלי, הוא לא יוכל להבחין במידע חריג בגוף הקובץ.

בכדי לשלוף את המידע, עלינו לפתוח את התמונה "התמימה" בעורך-הקסדצימאלי, לגשת לסופו של תוכן התמונה הראשונה, ולשלוף את תוכן התמונה השניה. בכדי למצוא את תחילת התמונה השניה נוכל- או פשוט לקרוא את גודל תוכן הקובץ המקורי (על-ידי התבוננות ב-Headers) או על-ידי חיפוש המחרוזת "424D" שנמצאת בראש קבצי ה-BMP:



- החלק המסומן באדום - סוף תוכן התמונה התמימה.
- החלק המסומן בירוק - תחילת מבנה התמונה הסודית.

ולאחר מכן- חילוץ התמונה על-ידי העתקת המידע לקובץ חדש ונקי- ופתיחת הקובץ החדש בעורך גראפי.

ע"י שימוש בפעולה זאת, נוכל בקלות מאוד להתגבר על המקרים בהם יפתחו את הקובץ בעורך-טקסט, אך עדיין, גם כאן לא תהיה בעיה לגלות כי אכן יש כאן מידע מוחבא. בקלות מאוד אפשר לראות (בנתוני מערכת ההפעלה על מאפייני נפח הקובץ) כי גודל הקובץ הרבה יותר גדול ממה שהוא אמור להיות (על ידי הסתכלות ב-Header של הקובץ עצמו) וכך להבין כי מוחבא בו מידע אשר לא מוצג.

מכאן אנו יכולים ללמוד כי פעולה סטגנוגרפית איכותית אינה אמורה לפגוע באחד ממאפייני הקובץ (כגון- גודלו)- או לפגוע בו כמה שפחות. מה שאומר, שבכדי להשאר "אמינים" כמה שיותר עלינו להחביא את המידע כך שתוכנו של הקובץ אכן ישקף את מה שכתוב ב-Header שלו.

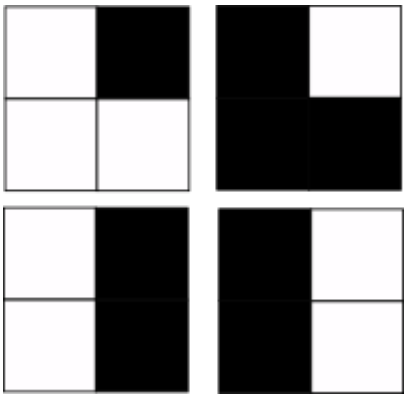
שלב אחד קדימה

מה שאנו יכולים ללמוד מכאן הוא שאסור לנו לפגום ב-Headers של קבצי התמונה, מפני שכך יהיה ניתן להבין כי מדובר בקובץ "חשוד", דוגמה מעניינת שפגשתי באחד הפוסטים בבלוג "לא מדוייק" של גדי אלכסנדרוביץ' יכולה לתת מענה מעניין לפעולה סטגנוגרפית שכזאת, פעולה סטגנוגרפית אשר לא פוגמת במבנה הקובץ- הפוסט מדבר על רעיון שהציגו נאור מוני ועדי שמיר. אני לא ארחיב יותר מדי מפני שגדי עשה זאת באופן יוצא מהכלל ואני ממליץ לעבור עליו (ובכלליות על כלל הבלוג) אלה רק אתן את התקציר:

אנו מעוניינים לחלק מידע מסויים הקיים ברשותנו לשני תמונות שונות, הרעיון הוא שאנו מעוניינים ליצור מצב כזה שמהסתכלות על כל תמונה בנפרד לא יהיה ניתן להסיק שום מידע לגבי התוכן בשלמותו, ואף יותר מכך- אנו מעוניינים להגיע למצב שמהסתכלות על כל תמונה בנפרד נוכל לראות תוכן לגיטימי כגון מידע הגיוני אחר (וכך לא ניצור "חשד" כי מדובר כאן בחומר סטגנוגרפי).

הרעיון שנאור ועדי הציגו במאמרם, הוא ליצור תמונה, אשר כל פיקסל בה נוצר מארבעה פיקסלים שונים שיפוזרו בין התמונות (מה שאומר שכל פיקסל בתמונת התוצר הסופי בגודל 2x2). הרעיון הוא שכשנניח את התמונות אחת על השניה (בהתחשב בכך שצבע לבן ייחשב כצבע שקוף) נחבר את הפקסלים שיצרו את התמונה השלמה, אז איך בכל זאת אנו לא יכולים ללמוד דבר מתמונה אחת על לפחות חצי מהתמונה השלמה? מפני שכאשר אנו מחזיקים בתמונה אחת (מתוך שניים) אנו לא מחזיקים בשום פקסל שלם- כי כמו שאמרנו, בכדי ליצור פקסל שלם אנו חייבים גם את חלקו השני.

ועכשיו לאופן המימוש:
דוגמה לפקסלים שיכולים לשמש אותנו בכל תמונה בכדי להרכיב את הפקסל הסופי בתמונה הסופית:

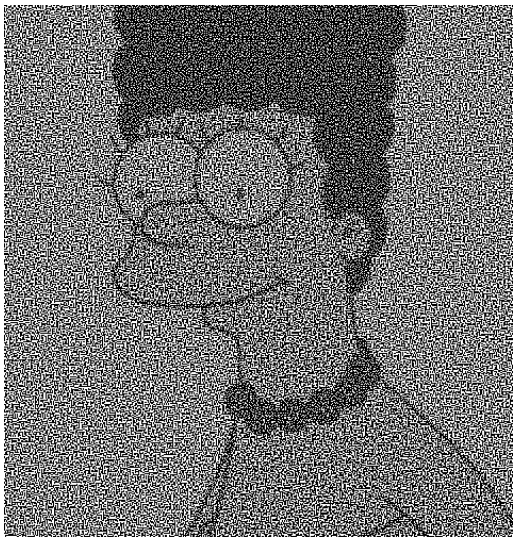


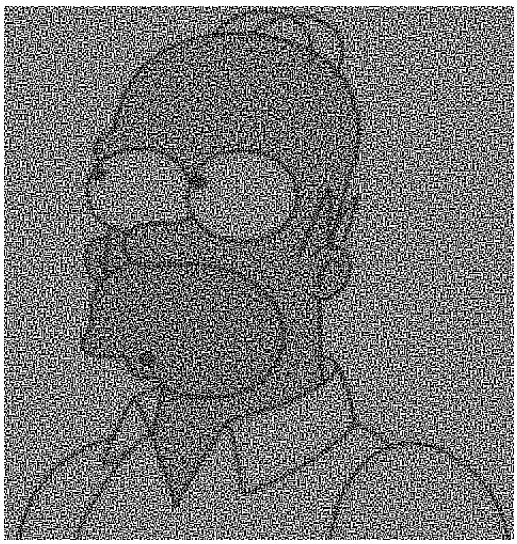
(במקור: http://gadial.blogli.co.il/archives/163)

"אם נשים שני פיקסלים כאלו אחד על השני (כשלבן משמש כשקוף, כמובן) אנחנו יכולים לקבל כמעט כל הרכב שנרצה. לדוגמה, תת הפיקסל התחתון משמאל הוא יחסית לבן (כי חצי ממנו לבן). אם נניח אותו על עצמו (כלומר - בשתי התמונות באותו פיקסל החצי הימני יהיה צבוע בשחור והשמאלי יהיה שקוף) נקבל את עצמו - משהו שעדיין נראה לבן יחסית. לעומת זאת, אם נניח אותו על שכנו מימין, הפיקסל שחלקו הימני שקוף והשמאלי שחור - נקבל פיקסל שהוא שחור לחלוטין."

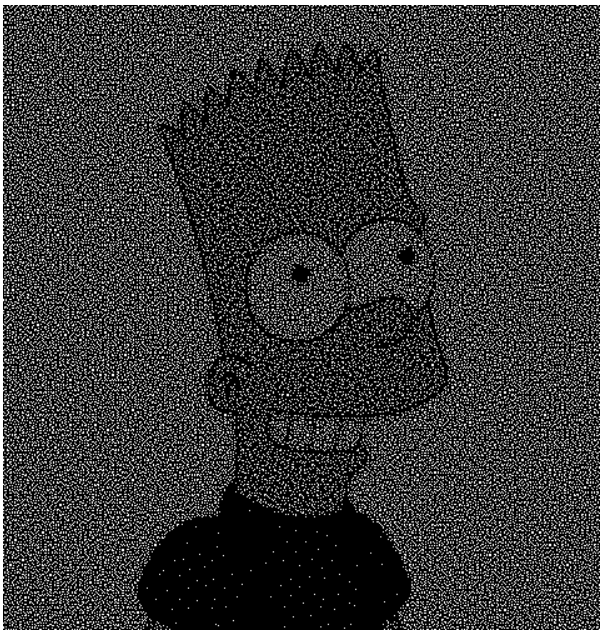
(צוטט במקור: **http://gadial.blogli.co.il/archives/163**)

כמימוש של זה, אפשר לראות את שני התמונות הבאות:

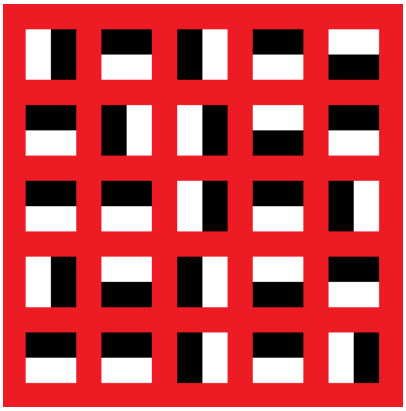




כאשר נציב אותן אחת על השניה (שוב, בהתחשב בכך שצבע לבן משמש כצבע שקוף) נקבל את התמונה הבאה:



ניתן לשים לב כי התמונה הסופית הרבה יותר כהה מהתמונות שהרכיבו אותה- וכאן בעצם טמון הרעיון, כאשר אנו מסתכלים על הפקסלים הבאים (נלקחו מהפינה השמאלית של התמנוה של הומר):



(הקווים האדומים משמשים להפרדה בין הפקסלים בלבד ואינם מופיעים בתמונה המקורית)

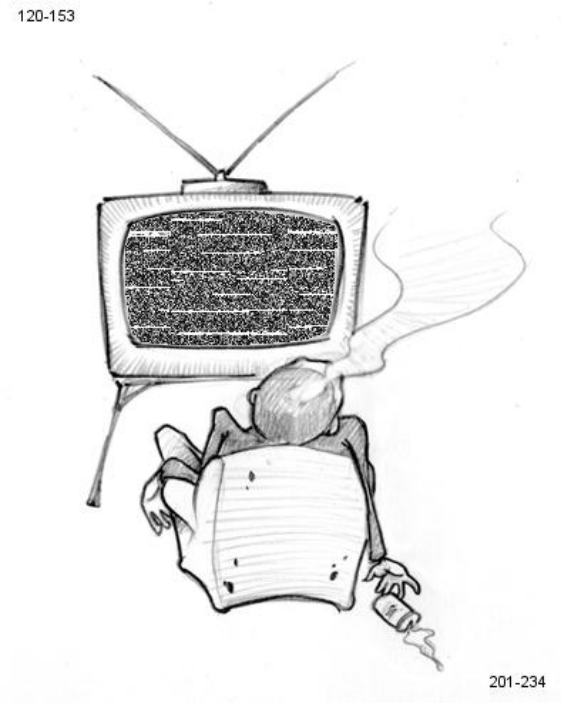
אננו יכולים בשום דרך לדעת איזה פיקסלים- הפקסלים הנ"ל ירכיבו בסופו של דבר. כך שגם בהינתן לנו חציו של המידע- אין לנו היכולת להסיק שום מידע לגבי התמונה הסופית.

החבאת קובץ הרצה בתוך תמונה.

לפי מה שראינו עד כה- אנו חייבים להחביא את המידע **בתוך** התמונה, מה שאומר- שעלינו להחביא את המידע כך שהוא יוצג למשתמש, אך באופן כזה שהמשתמש (אשר לא מודע לכך שבוצעה בתמונה פעולה סטגנוגרפית) לא יוכל להבין כי הוא אכן צופה במידע מוסתר.

בכדי להסביר את הרעיון, אתן את הדוגמה מהמקום בו פגשתי אותה לראשונה- אתר אתגרי Reversing ו-Stegano מומלץ ביותר בשם- Ma's Reversing.

באחד מהשלבים אנו נתקלים בתמונה הבאה:



האתגר הוא כמובן לגלות מה היא הסיסמה המוחבאת בתמונה. וזאת דוגמה מצויינת למה שאני מעוניין להסביר. ברור לי שבגלל ההקשר שבו אני מביא את התמונה ישר אתם תתמקדו על החלק המעניין ביותר בה- התוכן המוצג בטלוויזיה, אך, ברור לי שאם הייתם פוגשים את התמונה הנ"ל בהקשר אחר לחלוטין- אף אחד לא היה מתמקד בתוכן הנ"ל- הרי מדובר ב"רעש לבן" לגיטימי לחלוטין.

ובכל זאת, איך נגשים לאתגר שכזה?
צמדי המספרים בפינה השמאלית העליונה והימנית התחתונה מסמנים גבולות של ריבוע (שבמקרה נופל בדיוק על ה-"רעש הלבן" בטלוויזיה) – אותו יש לחתוך ולשמור כ-"Raw Files". לאחר מספר משחקים עם אותו הקובץ יש להמיר לקובץ "COM" ולהריץ. אתם אולי לא תאמינו- אבל מדובר בסט פקודות בינאריות שיציגו לכם פלט על המסך. זהו לא סוף האתגר- אבל לא האתגר הוא נושא המאמר. ובכוונה לא נכנסתי לפרטים, בכדי לא להרוס למי שכן מעוניין לפתור את האתגר לבד.
מה אנחנו יכולים ללמוד מהדוגמה הנ"ל?

שימו לב שאם נסתכל על פורמט התמונה- לא נבחין כי קיימות חריגות- מפני שאין שום חריגות בפורמט הקובץ, התוכן מוצג למשתמש בדיוק כמו כל פיקסל אחר.

בנוסף על כך, התוכן המוצג לא מהווה שום גורם מחשיד- מפני שהוא מוצג במיקום רלוונטי ("רעש לבן" בתוך מסגרת של טלוויזיה - לגיטימי לחלוטין) ועל כן מדובר בתמונה סטגנוגרפית טובה.

סיכום

כמו שהבנתם מקריאת המאמר, רבות השיטות בהן ניתן להחביא מידע בתוכן ויזואלי, מספר רב מהדוגמאות ניתן אף ליישם מחוץ לעולם הדיגיטלי, כמובן שישנן עוד שיטות רבות להחבאת מידע בתוכן חזותי באופן מוצלח אבל אני אעצור כאן.