**מחקר – GetOpenFileName**

מושגים:

Hooking הוא פעולה בה משנים את ההתנהגות של פונקציה או תוכנית מסויימת בדרך שהיא לא הייתה אמורה לפעול. hooking משמש למטרות רבות ובהן חיפוש [באגים](https://he.wikipedia.org/wiki/%D7%91%D7%90%D7%92) והרחבת אפשרויות השימוש בפונקציות קיימות. באמצעות טכניקה זו ניתן ללכוד את המידע העובר בין ה[מקלדת](https://he.wikipedia.org/wiki/%D7%9E%D7%A7%D7%9C%D7%93%D7%AA_%D7%9E%D7%97%D7%A9%D7%91) או ה[עכבר](https://he.wikipedia.org/wiki/%D7%A2%D7%9B%D7%91%D7%A8_(%D7%9E%D7%97%D7%A9%D7%91)) ל[מחשב](https://he.wikipedia.org/wiki/%D7%9E%D7%97%D7%A9%D7%91) או לזהות קריאה ל[פונקציות](https://he.wikipedia.org/wiki/%D7%A4%D7%95%D7%A0%D7%A7%D7%A6%D7%99%D7%94_(%D7%AA%D7%9B%D7%A0%D7%95%D7%AA)) [ספרייה](https://he.wikipedia.org/wiki/%D7%A1%D7%A4%D7%A8%D7%99%D7%99%D7%94_(%D7%AA%D7%9B%D7%A0%D7%95%D7%AA)) כדי לבחון את התנהגותן ועוד...

הHooking מתבצע באמצעות "טראמפולינה". הטראפולינה היא קטע קוד קצר שמוסף בכתובת הנוכחתי של אותה פעולה שאני רוצים להרחיב את אפשרויתיה. קטע הקוד הזה "קופץ" לפונקציה שלנו ומשם הפונקציה נראתה בחזרה. לכן השם "טראמפולינה".

Injection היא פעולה בה מריצים תוכנית משלנו בתוך תהליך של תוכנית אחרת על ידי טעינה של ספריה דינמית (DLL או Dynamic Link Library) בתוך אותו תהליך. על ידי ביצוע פעולה זו ניתן לשנות את התוכנית לטובה או לרעה.

כבר בהתחלת הפרויקט הבנתי שכדי למנוע מקבצים מסויימים לעזוב את המחשב יש למנוע מתוכניות לפתוח אותם. פתיחת קבצים ממומשת בעזרת הפעולה CreateFile. בפרויקט שלי אני מזריק DLL שעושה Hooking לפעולה זאת לתוכניות שאני מגדיר למוצר.

לאחר שניסיתי את המימוש שלי בדפדפן האינטרנט ושלחתי לעצמי אימייל גיליתי כי למרות שאני מונע מהדפדפן לפתוח את קובץ הוא עדיין שולח קובץ עם אותו שם אך הקובץ ריק. כאשר דפדפנים או תוכנות אחרות מעוניימנות לפתוח קובץ הם יוצרות חלון מיוחד שנותן למשתמש לעבור על כל הקבצים שבמחשב שלו ולבחור את הקובץ. התוכנית אז מקבלת מחלון זה מסלול מלא לקובץ וכך פותחת אותו. זו גם הסיבה שהדפדפן שלח קובץ ריק. הוא שלח את הקובץ אך לא היה לו את תוכנו מכוון שלא הצליח לפתוח אותו.

באחד משיעורי סייבר הייתה לנו משימה דומה ובא היינו צריכים לבצע Hooking לפעולה שיוצרת חלון זה בצייר. הפעולה נקראה GetOpenFileName. אז החלטתי לממש זאת שוב ולבדוק אם זה יעבוד. לאחר בבדיקות מוצלחות בבית במערכת ההפעלה Windows XP ובדיקות כושלות במערכות ההפעלה Windows 7 וWindows 10 הגעתי למקנה כי באחר ממערכות ההפעלה החדשות יותר של Microsoft (אחרי Windows XP), הם שינו את הדרך בה תוכנות יוצרו את החלון הזה.

לאחר מחקר גיליתי כי במערכת ההפעלה Windows Vista, Microsoft החליטו לשות את גישתם הישנה (שהייתה ממושת כתכנות מונחה פונקציות) הגישה חדשה, מתקדמת ובטוחה יותר בשם COM (שממומשת כתכנות מונחה עצמים) בכדי לפתוח את החלון לפתיחת קבצים. בכך הם הפכו את המשימה הפשוטה יחסית של Hooking לפונקציה למשימה קשה בהרבה.

המסקה היא: יש להתרכז ביצירת מוצר עובד קודם ורק אז לחזור ולנסות לפתור בעיה זו.

**מחקר – Media Devices Detection (USB)**

**Create a window and detect system messages with it about the devices inserted or removed:**

Windows sends all top-level windows a set of default [**WM\_DEVICECHANGE**](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363480(v=vs.85).aspx) messages when new devices or media (such as a CD or DVD) are added and become available, and when existing devices or media are removed. You do not need to register to receive these default messages. See the Remarks section in [**RegisterDeviceNotification**](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363431(v=vs.85).aspx) for details on which messages are sent by default. The messages in the code example below are among the default messages.

Each [**WM\_DEVICECHANGE**](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363480(v=vs.85).aspx) message has an associated event that describes the change, and a structure that provides detailed information about the change. The structure consists of an event-independent header, [**DEV\_BROADCAST\_HDR**](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363246(v=vs.85).aspx), followed by event-dependent members. The event-dependent members describe the device to which the event applies. To use this structure, applications must first determine the event type and the device type. Then, they can use the correct structure to take appropriate action.

When the user inserts a new CD or DVD into a drive, applications receive a [**WM\_DEVICECHANGE**](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363480(v=vs.85).aspx) message with a [DBT\_DEVICEARRIVAL](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363205(v=vs.85).aspx) event. The application must check the event to ensure that the type of device arriving is a volume (the **dbch\_devicetype** member is **DBT\_DEVTYP\_VOLUME**) and that the change affects the media (the **dbcv\_flags** member is **DBTF\_MEDIA**).

When the user removes a CD or DVD from a drive, applications receive a [**WM\_DEVICECHANGE**](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363480(v=vs.85).aspx) message with a [DBT\_DEVICEREMOVECOMPLETE](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363208(v=vs.85).aspx) event. Again, the application must check the event to ensure that the device being removed is a volume and that the change affects the media.

The following code demonstrates how to check for insertion or removal of a CD or DVD.

**Or Manually enumerate all devices with SetupDi API:**

With that API you can request the system to return you a list of devices matching the given device GUID class (something like an identifier for a type of devices like GUID for USBs, GUID for Audio devices, GUID for keyboards and mice etc).

The first method is far more convenient, since we don’t need to work hard and iterate a list of devices over and over and check for changes in it. In the first method we are also notified whether it's a device insertion or removal event. In addition in the second method after we detected a device we manually need to retrieve information about it unlike in the first method where it is given in the parameters passed to the function. However, the first method does not retrieve information about already connected devices. It retrieves information about devices that were inserted or removed after the launch of the program and that is why I will use both methods. The second method will be used to detected already inserted devices and afterwards I will detect devices with ease by using the first method.