דרייברים על קצה המזלג

מי נגד מי ולמה

מה זה בכלל דרייבר

- קוד שרץ בקרנל, כלומר איפה שהקוד של מערכת ההפעלה רץ
- יש לו הרשאות לעשות כמעט הכל
- כמו kernel module בלינוקס


WITH GREAT POWER COMES GREAT RESPONSIBILITY..

למה לא לעשות הכל בדרייבר

- בגלל שיש לנו הרשאות לעשות הכל – מערכת ההפעלה לא מגנה עלינו מלעשות טעויות ולהרוס לעצמנו את המחשב
- אם אנחנו עושים באגים בדרייבר זה גורם לקריסה של המחשב – BSOD
  - אם אנחנו מספיק מוכשרים אפשר לגרום לו לקרוס בכל פעם שהוא עולה וככה להרוס לעצמנו את המחשב

מה אפשר לעשות בדרייברים

1. לנטר על פעולות על קבצים
2. לנטר על פעולות registry
3. לנטר על יצירת processים
4. להסניף ולהוציא פקטות
5. הכל

עבודה עם קבצים

פעם היה ממשק ישן של FSFilter.
היה צריך לממש המון המון פונקציות (למעשה כל הפונקציות האפשריות שיש לקבצים) על מנת לייצר דרייבר כזה, ומצד שני היה אפשר להיות באמצע כל תהליך שקשור לפעולה על קובץ.

פעולות קיימות לדוגמה:
פתיחה, קריאה, כתיבה, סגירה, תשאול מידע על הקובץ ועוד

עבודה עם קבצים 2.0

מייקרוסופט חשפו ממשק שקוראים לו MiniFilter.

לא חייבים לממש את כל הפונקציות – רק את אלו שמעניינות אותנו.

פונקציות Cllback

זאת גם פונקציה, כמו כל הפונקציות שקיימות בעולם

נניח שיש לי תלות בevent מסויים ברכיב מסויים
אפשר שפעם ב-X זמן אני אשאל את אותו הרכיב "תגיד, הevent כבר קרה?"
וכאשר הוא יאמר כן ואני אריץ את הפונקציה

במקום זה – נבקש ממנו "שמע, כשהevent מתקיים – תוכל להריץ את הפונקצייה שלי?"

והוא אח על מלא, אז הוא יזרום איתנו.

אז איך minifilter עובד?

אני ארשום minifilter משל עצמי, ועבור כל פונקצייה שמעניינת אותי אני ארשום callback.

העבודה עם callbackים מאוד נפוצה בעולם הדרייברים של ווינדוס.

איך מפתחים דרייבר

- תמיד **אבל תמיד** לבדוק על מכונה וירטואלית
  - לא היינו רוצים לדפוק לעצמנו את המחשב ☹
- מומלץ להיות מחוברים עם kernel debugger
  - באג בדרייבר = קריסה של המחשב. להיות מחוברים עם kernel debugger עוזר לנו לתפוס את הבאג רגע לפני שהמחשב קורס.
- גם printים תמיד עוזרים
  - משתמשים בפונקצייה DbgPrint בדרייברים
- קוראים מלא באינטרנט

מה נעשה היום

נכתוב את הMinifilter הראשון שלנו!

ברגע שהוא יזהה שמישהו רוצה לפתוח קובץ שקוראים לו virus.exe הוא יחסום את הבקשה של המשתמש!