digital whisper
Technologic Papers

ניתוח נוזקות

מאת יובל (tsif) נתיב, להד לודר ו-5Finger

מבוא

במאמר הבא אנסה לסקור בקצרה את נושא ניתוח הנוזקות. ניתוח נוזקות נחשב בהרבה מקומות לנושא אשר שייך למקצוענים בלבד ולמקפצה הבאה בעולם הריברסינג. ניתוח נוזקות אומנם יכול להיות מורכב, מיוחד ומרתק, אך אני טוען שכל אחד עם ידע והבנה בסיסית בעולם המחשוב מסוגל לבצע ניתוח נוזקה בסיסי בכמה שעות בודדות ולצפות בנוזקה בפעולה.

ניתוח נוזקות

לפי הגדרת ויקיפדיה נוזקה הינה "תוכנה שמטרתה לחדור או להזיק למחשב ללא ידיעתו של המשתמש בו". אני חושב שההגדרה הזאת נכונה חלקית. לפני שניגש לניתוח הנוזקות עצמן ננסה לחלקן לקטגוריות (ראשיות בלבד!) בכדי שנוכל להבין טוב יותר כיצד עלינו לצפות מכל נוזקה להתנהג:

וירוסים

וירוסים הינם תוכנות אשר כל מטרתן הינה להזיק למחשב. לרוב תוכנות אלו יהרסו את הפונדקאי מהר מכדי שיוכלו להתפשט. דרכי השמדת המכונה הינן רבות ויכולות לכלול פגיעה במערכת ההפעלה עצמה, פגיעה בכונן הקשיח על ידי כתיבה לאזורים אשר אינה אמורה ועוד. וירוסים היום הינם דבר נדיר למראה ויש כאלה שיגרסו שזאת בגלל בעיית ההפצה, אך אני טוען שמאחורי וירוסים פשוט לא עומדים כוחות כלכליים כבדים כמו מאשר שאר סוגי הנוזקות שנראה בהמשך וזאת הסיבה העיקרית וכמעט הבלעדית שבגללה איננו רואים עוד וירוסים.

סוסים טרויאנים

סוסים טרויאנים הם כלים שמטרתם פשוטה - לאפשר שליטה מרחוק במכונה לשרת חיצוני. היום, מכיוון שאנחנו רגילים לעבוד מאחורי נתבים, מתגים ורכיבי תקשורת כאלה ואחרים, נוכל לראות שבדרך כלל התקשורת של הכלים הללו מתבצעת בדרך הפוכה. המשמעות הפרקטית של הדבר אומרת שהיום בדרך כלל הפניה נעשית מהמחשב הנגוע חזרה אל מרכז השליטה של הסוס.

רוגלות

רוגלות הינן תוכנות בעלות פוטנציאל כלכלי מרתק. בגלל מיקומן על הגבול האפור בין חוקי ללא-חוקי ניתן לראות הרבה מאוד סוגים שונים ומגוונים של רוגלות. חלקן מיוצרות על ידי האקרים שחורים ונשתלים במחשבי הקורבנות באמצעים בלתי חוקיים בעליל וחלקן מיוצרות על ידי חברות גדולות ומכובדות ומותקנות בשמחה וברצון על ידי המשתמש על מכשירו. דוגמא בולטת מאוד הינה תוכנת הפנס המפורסמת של אנדרואיד.

תוכנת ה-Tiny Flashlight קיבלה עד כה 637,344 המלצות חמות, 2,606,263 דירוגים ב-Google Play, דירוג של 4.6 (מתוך 5 כוכבים!) כוכבים (לצורך השוואה - פייסבווק קיבלה 3.8, דרופבוקס קיבלה 4.6, ולינקדאין 4.2) ובנוסף יש לתוכנה רק 100,000,000+ הורדות. למה אני מזכיר את התוכנה הזאת כאן? אם תנסו להתקין את התוכנה תשימו לב שהיא מבקשת הרשאה לחיבור אינטרנט, קריאת שיחות, הודעות ואנשי קשר. בדיקה מאוד מהירה תראה לכם שהאפליקציה חברה ברשת Millenial ושמרגע שהתקנתם את התוכנה אתם רשמית מנדבים מידע אישי אל מילניאל מדיה.

תוכנות-כופר

תוכנות כופר הן תופעה חדשה יחסית שהחלה לתפוס תאוצה לקראת סוף 2013 בעיקר הודות לנוזקה CryptoLocker אשר הצליחה להדביק מחשבים רבים (יחסית) ולגרום לנזק משמעותי בזמן קצר מאוד בעזרת שיטה חדשה. משמעותן של תוכנות-כופר (ransomeware) הוא שהן דורשות כופר מסיום בעבור משהו. בחלק מהמקרים יהיה זה מפתח עבור מידע מוצפן, בחלקם רכישה של 'אנטי וירוס' מיוחד כדי להפטר מהנוזקה וכן הלאה וכן הלאה.

מאפיינים

רוטקיטים

רוטקיט (Rootkit) הינה סוג של תוכנת אשר מחביאה קבצים, תעבורה , קבצי זכרון ועוד, לרוב הרוטקיט יגיע משולב כאשר המערכת נפרצה והפורץ משתמש בתוכנת הרוטקיט על מנת לשלוט בצורה גבוהה יותר ו"נקייה" יותר ובכך להחביא את הנוזקות או קבצים אחרים שהשתיל.

רוטקיט כשמה כן היא, נקראת כך בשל הסיבה בה הפורץ צריך להגיע להרשאות מערכת גבוהות (root במערכות יוניקס) ובכך להצליח להעלים את הנוזקות הנוספות אשר הפורץ השתיל, במצב זה לפורץ יש גישה מלאה למערכת ההפעלה ולמערכת הקבצים במחשב ולכן רוטקיט הולכת צעד רחוק יותר מהנוזקות הממוצעות, הן עלולות אפילו להדביק את BIOS ובכך להחביא היטב את הרוטקיט והנוזקות שהגיעו עימן. במקרים רבים הוספת מרכיב הרוטקיט לנוזקה תמנע גם ממערכות הגננה כגון אנטי ורוסים את יכולת הזיהוי והטיפול בנוזקה.

קילוגרים

קילוגר (Keylogger) הינו תוכנה או התקן חומרה העוקב אחר משתמשי מחשב (בעידן של היום גם סמארטפונים) ע"י מקשי המקלדת שהם לחצו. לא ניתן לזהות את נוכחותו של הקילוגר על המחשב שלך שכן הוא פועל ברקע ובמקרים רבים גם לא מופיע במנהל משימות או לוח בקרה (הסרה\התקנה של תוכניות) מכיוון שהוא מבצע חיבור (hooking) לרכיבים קיימים אשר אחראיים על עיבוד נתונים אשר מגיעים מהמקלדת.

על-אף העבודה שקילוגר יכול להיות כלי הרסני, קילוגר מספק מספר יתרונות במצבים שונים. במקום העבודה, קילוגר משמש לעתים קרובות כדי לפקח על העובדים כשהם משתמשים במחשבי חברה. אמנם זה אולי נראה פולשניות, אך זו גם דרך יעילה לוודא כי העובדים לא עושים שימוש לא ראוי במשאבי החברה. ניתן להשתמש בקילוגר ככלי אבטחה אשר יכול להיות בשימוש ע"י הורים על מנת להשגיח על ילדיהם כשהם גולשים ברחבי הרשת.

לצד ההיתרונות של הקילוגר, ניתן להשתמש שבו גם בצורה זדונית כגון חברה שמבצעת ריגול תעשייתי על חברה אחרת בתחום או מעקב אחר מחשבים פרטיים\ציבוריים כדי לגנוב סיסמאות, פרטי חשבון בנק, פרטי כרטיס אשראי וכו'. כתוקפים קילוגרים משמשים הרבה פעמים בכדי לאסוף מידע אשר בדרך אחרת נמצא מוגן. כך לדוגמא, כאשר מערכות הפעלה שומרות סיסמאות הן בדרך כלל שומרות אותן באופן מאובטח אשר קשה להגיע אליהם באופן ישיר. היצמדות לקלט המגיע מהמקלדת מאפשר "לאסוף" את אותן סיסמאות באופן נוח יותר.

קיילוגר כמזיק יכול לפעול בצורות שונות ולהתבסס על מאגרי שאיבה שונים , בעוד חלק מהקיילוגרים מתבססים על כתיבת המקלדת , חלקם מתבססים על שאיבת המידע מזכרוRAM בעוד חלק אחר מתבסס על סיסמאות השמורות בצורה נקייה ולא מוצפנת על המערכת, לרוב קיילוגרים מסוגים אלו ישלחו את המידע ל"האקר" בשיטות שונות כגון FTP,HTTP או שליחה ישירה לאיימיל של הפורץ, ישנם גרסאות שונות וחדשות גם בתחום המובייל אשר עובדות באותו המתכונת בדיוק, ועוקבות אחר ההודעות המדיה, פעילות טלפונית ועוד.

תולעים

תולעים הן אחד המאפיינים האחרונים אשר מתפתחים מאוד בשנים האחרונות ומעידות על שינוי מאוד משמעותי בתום הנוזקות. תולעת היא נוזקה (יכולה להיות כל סוג של נוזקה) אשר גם מפיצה את עצמה הלאה. ההפצה הזאת יכולה להיות על ידי שליחת מיילים, הדבקת מדיות חיצוניות, ועוד דרכים רבות ומעניינות. שתי הדרכים אשר תופסות תאוצה ומוכיחות את עצמן הן על ידי שיתופי SMB אשר קיימים בארגונים ומאפשרים דרך הדבקה מעולה בתשתיות. הדרך הנוספה הינה שימוש באקספלוייטים. היום נוזקות רבות מנצלות פרצות ואף פרצות יום 0 כדי להמשיך ולהדביק מכונות נוספות.

שיטות גישה לניתוח נוזקות

קיימות שיטות רבות לניתוח נוזקות. אין 'שיטה נכונה' או שיטה לא נכונה. ההתאמה של השיטה תהיה בהתאם למגבלות ולמטרת הניתוח. לפעמים אנו מעוניינים לייצר חתימה לנוזקה לטובת תוכנת אנטי וירוס כזאת או אחרת. לפעמים אנו נרצה להשתמש בנוזקה על מנת להבין בדיוק כיצד היא עובדת ומה היא מסוגלת לעשות - וחשוב יותר - כיצד. כל המטרות האלה הן שונות ולכן גם השיטות. במידה ואנחנו מעוניינים להחליף שמן אין צורך שנלמד כיצד כל בוכנה וגלגל שיניים במכונית שלנו עובד.

אנו עומדים להסתכל מעט בכמה שיטות ניתוח אך במבט עילי נוכל לחלק את שיטות הניתוח לשיטות הדינמיות ולשיטות הסטטיות. השיטות הדינמיות מחייבות טעינה של הנוזקה לזיכרון והרצה שלה. השיטות הסטטיות מבוססות על כך שאיננו רוצים להריץ את הנוזקה אלא לנתח את הקובץ שבו קיבלנו אותה כקובץ סטטי (hence the naming) על הדיסק.

יש לשים לב שאנו לא עומדים לדבר על כאן על נוזקות אשר מגיעות כחלק מקבצים אחרים. היום נוזקות רבות נישאות ומופצות על ידי קבצים 'תמימים' או אשר נתפסים כתמימים ואינן אמורים לאפשר הרצת קוד כגון קבצי PDF, JPEG, TIF, DOCX ועוד. על פורמטי קבצים אלה יש שיטות ניתוח שונות ומרובות אך המטען (payload) אותו נושאים אותם קבצים נשאר באותה תצורה של הנוזקה הראשונית אך מסופקת (injected) בשיטה אחרת.

סטטית - חילוץ מידע בסיסי

כשלב ראשוני בתהליך בו אנו ננסה להבין את הקובץ (יכול להיות שמדובר בנוזקה, חשוד לנוזקה או דברים אחרים רבים) ננסה להוציא מידע בסיסי על הקובץ עצמו בלי להכנס לעובי הקורה. כך לדוגמא נתחיל להשתמש בכמה כלים אשר יכולים לעזור לנו מאוד בתהליך ההכנה של קובץ טרם ההרצה שלו. התוכנה הראשונה אשר נשתמש בה כאן תהיה strings.exe. אחת מהתוכנות הנהדרות שהוא מארק רוסינוביץ' (SysInternals) והיא נמצאת להורדה כאן. לצורך הדוגמא כאן אנו נעיף מבט חטוף בתוכנה PwDump7.exe אשר משמשת למשיכת סיסמאות מקומיות ממערכות חלונות (קבצי SAM).

כאשר נריץ את strings.exe התוכנה תסרוק את הקובץ הניתן בארגומנט כדי להבין איזה מחרוזות מאוכסנות בקובץ באופן סטטי. התוצאה המלאה תופיע כאן. אך הנה נסתכל על אזורים מסוימים מתוך הפלט וננסה להבין מה נוכל לזהות.

```
SunMonTueWedThuFriSat
JanFebMarAprMayJunJulAugSepOctNovDec
GetLastActivePopup
GetActiveWindow
MessageBoxA
user32.dll
ReadFile
SetFilePointer
GetLastError
DosDateTimeToFileTime
SetLastError
FlushFileBuffers
WriteFile
GetFileAttributesW
CreateFileA
CloseHandle
GetWindowsDirectoryA
KERNEL32.dll
MessageBoxA
IsCharUpperW
IsCharAlphaW
USER32.dll
LIBEAY32.dll
HeapAlloc
MultiByteToWideChar
HeapFree
ExitProcess
TerminateProcess
GetCurrentProcess
HeapReAlloc
GetTimeZoneInformation
GetSystemTime
GetLocalTime
GetCommandLineA
GetVersion
HeapDestroy
```

```
HeapCreate
VirtualFree
VirtualAlloc
SetHandleCount
GetStdHandle
GetFileType
GetStartupInfoA
UnhandledExceptionFilter
GetModuleFileNameA
FreeEnvironmentStringsA
FreeEnvironmentStringsW
WideCharToMultiByte
GetEnvironmentStrings
GetEnvironmentStringsW
RtlUnwind
SetStdHandle
LCMapStringA
LCMapStringW
GetStringTypeA
GetStringTypeW
GetCPInfo
GetACP
GetOEMCP
GetProcAddress
LoadLibraryA
SetEndOfFile
CompareStringA
CompareStringW
```

אם כן, מה שאנחנו יכולים לראות בשורות 726 עד 789 אלה פונקציות אשר להן קוראת התוכנה. נכון, זה לא הרבה, אך אנחנו יכולים להתחיל להבין לאיזה פונקציות קוראת התוכנה ולפי כך מה הן הפעולות הבסיסיות שלה. אנחנו יכול להיות לב לפונקציות GetEnvironmentStrings, GetProcAddress, GetStartupInfoA, GetFileAttributesW ו-GetWindowsDirectoryA. השילוב של כל אלה מאפשר לנו להבין שכנראה שהתוכנה מתעסקת עם נושא ההרשאות לקבצים, הבנה לגבי תהליך ההתחלה של מערכת ההפעלה, משתנים סביבתיים, ותיקיות של מערכת ההפעלה.

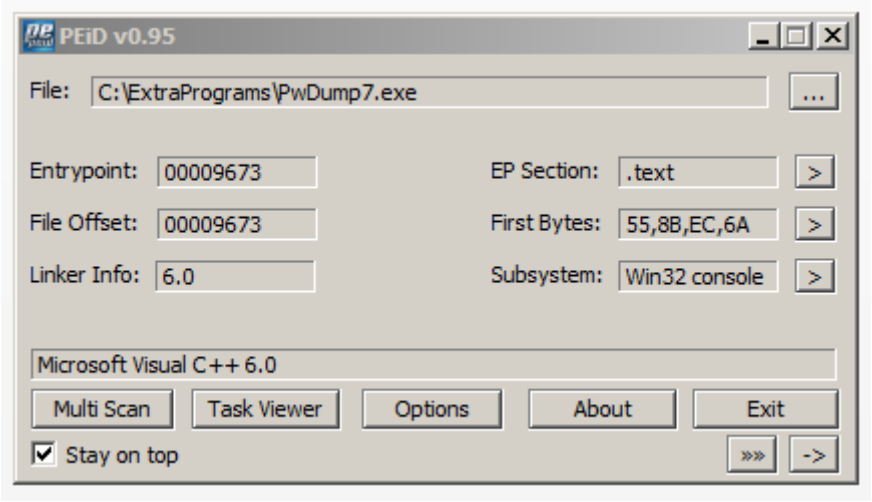
אחרי שראינו את החלק בקוד שלנו שקורא לקריאות מערכת ההפעלה בואו נסתכל על החלק בקובץ שמציג לנו מחרוזות נוספות אשר מקודדות פנימה:

```
Error reading FAT32 FS
UNABLE TO OPEN DEVICE?
\\.\%C:
Skew1
GBG
Data
Error reading system registry file %S
Error reading hive root key
%s\Select
Default
```

```
%s\ControlSet%03d\Control\Lsa\
%s%s
Error accessing key %s
Wrong/corrupted hive??
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es
usage:
 pwdump7.exe                                (Dump system passwords)
 pwdump7.exe -s <samfile> <systemfile>     (Dump passwords from files)
 pwdump7.exe -d <filename> [destionation] (Copy filename to
destionation)
 pwdump7.exe -h                             (Show this help)
!@#$%^&*()qwertyUIOPAzxcvbnmQQQQQQQQQQQQ)(*@&%
0123456789012345678901234567890123456789
NTPASSWORD
LMPASSWORD
savedump.dat
%s\SYSTEM32\CONFIG\SYSTEM
%s\SYSTEM32\CONFIG\SAM
saving %S as %s
 Unable to dump file %S
File %s saved
Error opening sam hive or not valid file("%S")
Error reading hive root key
%s\SAM\Domains\Account
%s\SAM\Domains\Account\Users
%s key!
No F!
No Users key!
Names
Asd -_- _RegEnumKey fail!
No V value!
%s:%d:
%.2X
%.2X
:::
%s:%d:
NO PASSWORD*********************:
%.2X
:::
%s:%d:
NO PASSWORD*********************:NO PASSWORD*********************:::
         (((((                   H
PST
PDT
\#A
```

מה שאנחנו יכולים לראות כאן כבר מרמז לנו משמעותית יותר על התוכנה. אנחנו יכולים לראות מחרוזות שמדברות על בעיה בגישה ל-HIVE, את המיקום של קבצי ה-SAM במערכת ההפעלה, שימוש והוראות הפעלה לתוכנה, ועוד ועוד ועוד. ניתוח כזה מאפשר לנו הבנה טובה מאוד על תוכנה שעדיין לא הרצנו ועכשיו אנו יודעים קצת יותר למה לצפות מהתוכנה. כמובן שבנוזקות בדרך כלל אנחנו מתמודדים עם דברים בעיתיים יותר.

ננסה להעיף מבט בתוכנה נוספת הנקראת PEID. התוכנה הזאת מנסה לזהות לפי אותן מחרוזות וסממנים נוספים באיזה קוד כתובה התוכנה, מה היא נקודת הכניסה, בעזרת איזה מהדר נבנה הקובץ הבינארי ועוד. לאחר שנריץ את PEID על אותו PwDump7.exe נקבל את התוצאה הבאה:



כמו שניתן לראות, התוכנה מיועדת לפלטפורמה של Win32, נכתבה והודרה ב-Visual C++ 6.0. בנוסף ניתן לראות שנקודת הכניסה לקובץ נמצאת בהסטה של 00009673 לתוך הקובץ. מידע נוסף שיכול לעזור לנו להבין באיזה תוכנה מדובר ומה היעוד שלה. אולי אפילו במקרים מסוימים למצוא מהדר לאחור (DeCompiler) מוכן במידה וקיים. כמובן שזאת רק ההתחלה ואלה רק חלק מהכלים, אך ניתוח סטטי ומהיר יחסית כגון זה יכול לספק מידע איכותי ומעניין לגבי הקובץ אותו מנתחים.

סטטית - הנדסה לאחור

הנדסה לאחור היא במצבים אופטימלים ה'שיטה האולטימטיבית' לניתוח נוזקות. ללא הרצת הקוד ניתוח ליצור פרויקט ב-IDA ולהתחיל לנתח כל שורה ושורה וכל קריאה וקריאה אשר מתבצעת בקוד. הבעיות העיקריות אשר עולות משיטה זאת הינן ההיכרות הטובה אשר נדרשת עם קוד גם בשפת המקור וגם ברמת האסמבלי יחד עם זמן רב לשרוף. לא נעמיק בשיטה הזאת מכיוון שבאמת שאינני יודע אפילו מהיכן להתחיל ורבים וטובים לפניי כתבו מאמרים איכותיים מאוד.

דינמית - ניתוח תבניות תעבורה

לפעמים איננו מעוניינים לנתח את הנוזקה באמת אלא רק את התנהגות הרשת שלה. כך לדוגמא נוכל להשתמש ב-CryptoLocker כמקרה מעניין. CryptoLocker משתמש במנגנון DNS כדי לבחור שרת שאליו תפנה הנוזקה. הנוזקה תשתמש בשרת המקור כשרת לחילול מפתחות אשר בעזרתו תצפין את המידע על הדיסק. אולם אם היה שרת אחד בלבד או כמה שרתים בודדים היה ניתן בקלות לאתר את אותם שרתים ולמנוע גישה אליהם. CryptoLocker מגיש מחולל שמות דומיין לפי אלגוריתם מסוים ולפי זמן מסוים ומנסה לעשות פניה אל עשרות (ולפעמים מאוד) שרתים אשר רובם נדחים. בסופו של דבר רק אחד מהם מחזיר תשובה והוא יהיה השרת אשר יספק את המפתחות וינהל את נושא ההצפנה של הקבצים הרגישים.

הבנה כזאת של הנוזקה (שנוכל לראות מאוחר יותר כיצד להגיע אליה) מספקת לנו אפשרות לשתק את CryptoLocker ללא השמדה של הנוזקה עצמה. יהיה לנו קל אפוא לזהות את תבנית הבקשות, הכמות והאופי ולחסום את התוכנה מלגשת אל השרת שלה - דבר שבמקרה הזה מוביל לאי הצפנת המידע מכיוון שלא מסופק מפתח.

דינמית - ניתוח התנהגות

ניתוח התנהגות דומה במקצת לניתוח תבניות רשת אך ההבדל כאן הוא היחס הכלל מערכתי ולא רק רשתי. בניתוח התנהגותי אנחנו נרצה לראות כיצד התוכנה עולה ומה היא עושה, נרצה לראות לאיזה תיקיות וערכים ברג'יסטרי היא פונה וכמובן גם איזה תעבורת רשת היא מחוללת. לא נרחיב על ניתוח התנהגותי כאן מכיוון שאנו מגיעים לזה בפרק הבא של המאמר.

קידום עצמי "מעודן"

לאחרונה הבנו שקיימות נוזקות שם בחוץ וניתן להתחיל לחקור אותן כדי לנסות להבין מה מבצעות. לאומת זאת, אותן נוזקות אינן זמינות להורדה ממוקד אחד אשר פתוח לציבור ובטח שאינן מסודרות ומאונדקסות לפי סדר מסויים ולכן החלטנו לייצר את malware-db. הרעיון של הפרויקט הוא לרכז נוזקות ולסדר אותן כך שכל אדם אשר מתעניין בנושא יוכל להוריד אותן ולהתחיל לחקור את אותן נוזקות ולצפות בהן בפעולה.

הורדה

כאן אין יותר מידי מה לאמר. להמשך המאמר יש לבצע git clone לריפוסיטורי בגיט האב אשר מכיל את הקוד ואת הנוזקות (בהמשך יופרד למאגר נוזקות ולתוכנת אינדוקס והרכשה). לאחר ההורדה אנו עוברים לסקירה מהירה של כלים בהם נשתמש וכיצד להקים את סביבת העבודה הראשונית.

כלים שימושיים בניתוח התנהגותי

בקטע הזה אנסה לסקור כמה תוכנות מומלצות להתחלת ניתוח התנהגותי של נוזקה.

- **Wireshark** - כלי נהדר, פתוח ויעיל מאוד. Wireshark יאפשר לכם לצפות בכל חבילה שנכנסת או יוצאת מהמכונה שלכם. היתרון הגדול ביותר של Wireshark או גם החיסרון שלו - Wireshark מאפשר צלילה עמוקה מאוד לפרטים לרמה של כל ביט בכל חבילה. מומלץ לשמור תיעוד מלא של האזנת Wireshark לאורך כל ההדבקה של המכונה.
- **DirWatch** - התוכנה הזאת מבקשת כפרמטר לדעת איזה תיקייה לצפות בה. לאחר מכן היא מתחילה לצפות בכל רכיב שנוצר או משתנה במערכת הקבצים. התוכנה פשוטה והייעוד שלה הוא פשוט אך עם עזרה ניתן להבין בקלות איזה קבצים נוצרים או נמחקים ומה קורה עם כל רכיב באותה תיקייה אשר בה צופים.
- **Process Explorer** - התוכנה הזאת מאפשרת לנו לצפות בעץ התוכנות הפועלות המחשב, באיזה היררכיה ואפילו לייצא זיכרון של תוכנית מסויימת לקובץ memory dump. מומלץ להעמיק במדריך של התוכנה הזאת מכיוון שהיא מכילה אפשרויות רבות שאל חלקן צריך להעמיק קצת יותר כדי להגיע.
- **ProcMon** - תוכנה מגניבה נוספת מבית SysInternals אשר מאפשרת לצפות בפעולות הנעשות במחשב כולל גישה למערכת הקבצים, קריאה ל-DLLים שונים, עבודה עם הרג'יסטרי ועוד. התוכנה תנטר את כל המתבצע במחשב ותאפשר לייצא קובץ מסודר. יש לשים לב שתיעוד ארוך טווח (עם באפר גדול של כמות אירועים) עלול לגרום לתוכנה לקרוס באמצע וכך לאבד את המידע שאגרה עד כה.
- **RamCap32/64** - RamCap היא אחת התוכנות היותר פשוטות שתתקלו בה בתהליך אך מצד שני שימושיות ביותר. התוכנה לוקחת את מצב הזכרון הנוכחי ושומרת אותו לקובץ IMG. מומלץ לייצר

קובץ תמונה של הזכרון לפני הדבקה ובשלבים שונים של הריצה. התוצאה תהיה יכולת לבצע השוואות מאוד מעניינות בין התהליכים שהשתנו. ניתן תמיד להעזר ב-Autopsy כדי לנתח את התמונות (למרות שזהו לא ייעוד הכלי המקורי).

- **CurrPorts** - התוכנה מאפשר צפייה בחיבורים הקיימים המערכת. כל הפורטים שנפתחים, נסגרים, במצב האזנה, נכנסים, יוצאים מכל הסוגים והדרכים. כך ניתן יהיה לראות במידה והנוזקה שאנו חוקרים פותחת חיבורים מסויימים או מנסה לפתוח פורטים מסויימים.
- **SniffHit** - סניפהיט תציג לנו את אותם הדברים שראינו מקודם אך בתצוגה קצת שונה. בעזרת סניף היא נוכל לראות בקשות שונות שנכנסות ויוצאות וחלוקה לאיזה מחשבים נגשה התוכנה ולאיזה שרתים. כך תהיה לנו דרך נוספת להצליב בין השרתים השונים אליהם נגשנו ושרתים מסויימים 'יקפצו' לנו לעין מהר יותר.
- **SandBoxie** - תוכנה אשר מיועדת להגנה על מחשב ומאפשרת לנו להריץ תוכנה מסויימת בתוך סביבה וירטואלית ולהפריד אותה מהמכונה עליה אנחנו רצים.
- **APILogger** - נוזקות רבות משתמשות בהוקינג לתהליכים רצים ולפעולות מסוימות. התוכנה הזאת מאפשרת מבט קצת יותר מעמיק אל תוך הקריאות הללו וכיצד ומתי הן מתבצעות.

יצירת סביבת עבודה ודגשים

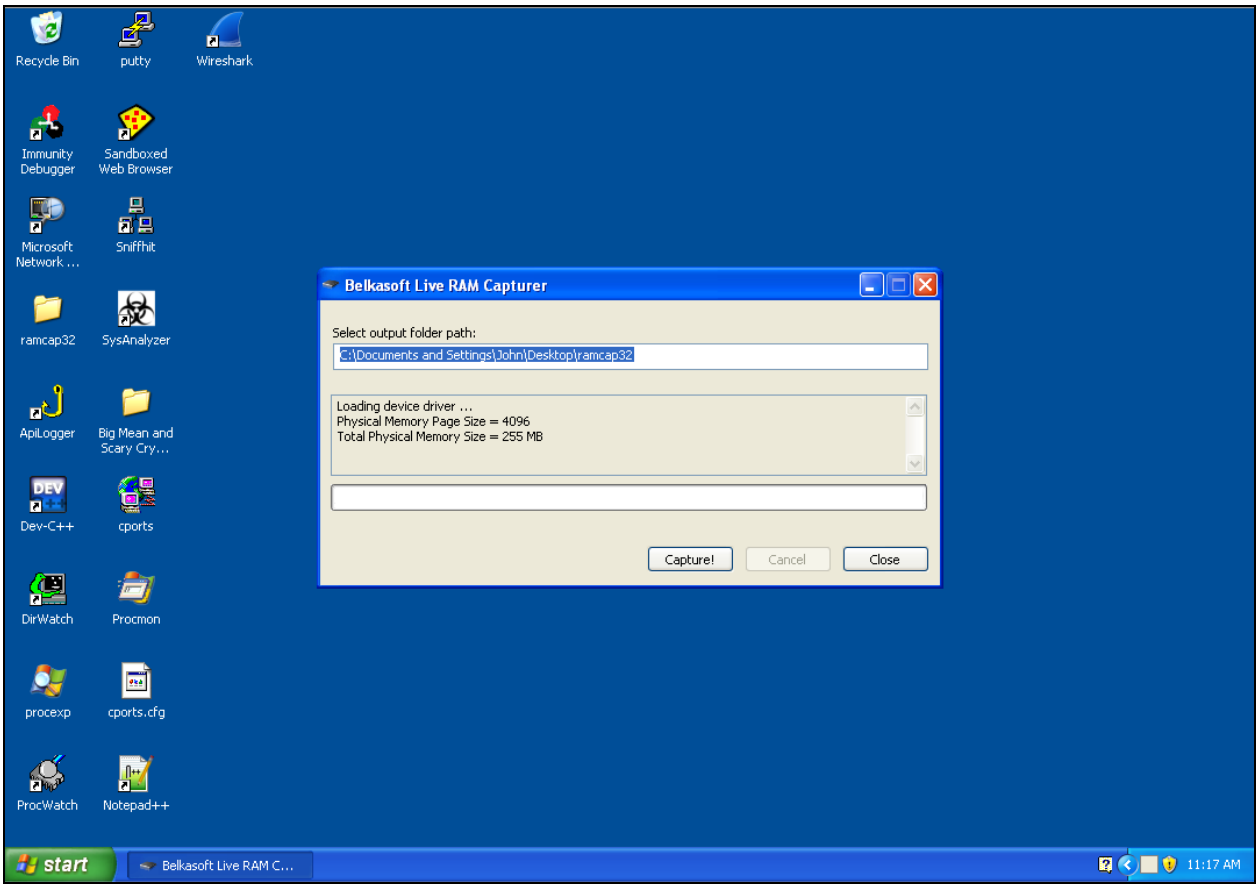
עכשיו שדיברנו על כלים אנחנו יכולים לעבור למבט עילי יותר לגבי כיצד נקים את סביבת העבודה שלנו. כל ניתוח לנוזקה שנבצע צריך לרוץ באופן נפרד ממערכת העבודה שלנו. בשאיפה על מחשב נפרד פיזית (נוזקות מסוימות מנסות לזהות האם הן רצות בתוך מכונה וירטואלית) אך לרוב אין הדבר אפשרי ולכן נשתמש בפתרונות וירטואליזציה. אני מעדיף לעבוד עם VirtualBox מכיוון שהיא פשוטה, יציבה ובמקרה הזה אנחנו לא זקוקים ליכולת קסטומיזציה גבוהה. לצורך ניתוח אני ממליץ לעבוד עם מערכת ההפעלה Windows XP מכיוון שבגרסה זאת מנגנוני הגנה רבים טרם התווספו, דבר אשר עלול להקשות על תהליך הניתוח שלנו. מרגע שמנגנוני הגנה מסוימים מופעלים במערכת נוזקות יבצעו תמרונים אי אלו ואחרים כדי להתחמק מהם ובמקרה של ניתוח התנהגותי אנו מעוניינים ב'שורה התחתונה' ולכן בדרך כלל האם הנוזקה מתחמקת מהגנות אלו או לא הוא פחות רלוונטי לשלב הראשוני בו אנו מנסים להבין מה הנוזקה בכלל עושה.

לאחר התקנת המערכת הוירטואלית קחו את קובץ ה-ISO שהורדתם כאשר שכפלתם את malware-db. קובץ ה-ISO הזה מכיל חבילת תוכנות ראשונית (וממש לא מתיימרת להיות מקיפה!) לניתוח התנהגותי. חלק מהתוכנות צריכות התקנה וחלק ניידות ולא דורשות. אני ממליץ להתקין עם את Immunity Debugger ו-++Notepad על המכונה כדי לאפשר כתיבה של סקריפטים במידת הצורך.

לאחר סיום ההתקנות וההגדרות, צרו Snap-Shot של תצורת המערכת הזאת כדי שתוכלו לחזור אליה כל פעם לאחר ניתוח נוזקה זאת או אחרת.

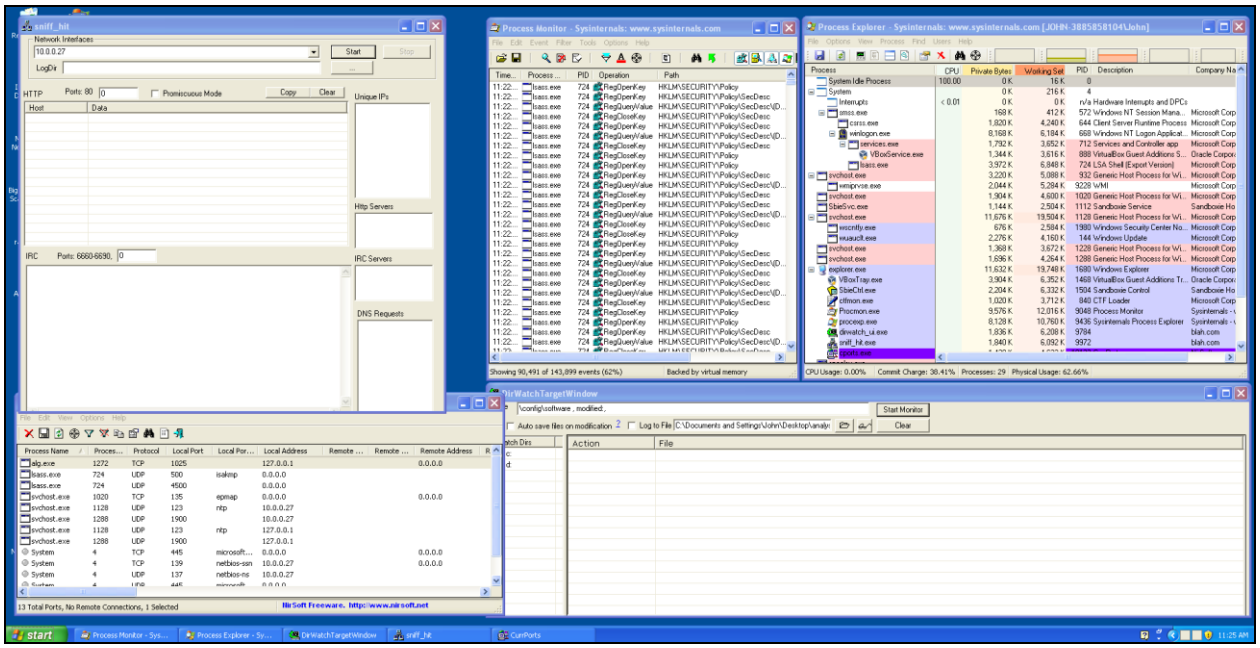
ביצוע התהליך שלב אחר שלב

אחרי שסימנו להעלות את המערכת, בואו נראה כיצד נתחיל בניתוח. עוד דיסקליימר קטן וחשוב - אנחנו לא עומדים לעבור על כל הטכניקות וכל הדרכים שמומלץ ליישם, אלא באופן שרירותי נעבור על הדברים הזריזים ביותר בלבד שמצריכים כמה שפחות ידע והבנה כדי לראות במה מדובר ואיך הנוזקה עובדת. אז לאחר שהעלינו את המכונה - הריצו את RamCap וקחו תמונה של מצב הזיכרון. שמרו אותו לקובץ וייצאו אותו מהמכונה הווירטואלית למכונה הראשית שלכם.



לאחר שעשיתם את תמונת הזכרון אני ממליץ על ארגון של שולחן העבודה באופן כזה:

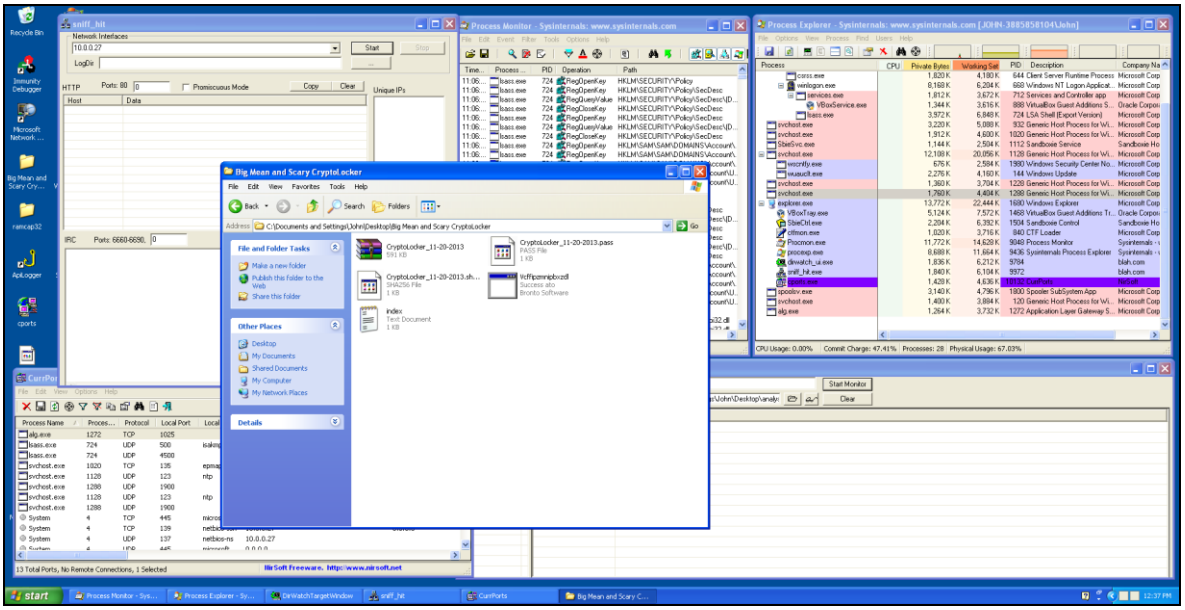
1. תעלו את כלי הניטור הבסיסיים שדיברנו עליהם. אני עובד עם תצורה כמו בתמונה מטה אשר מאפשרת לי לראות את הכלים ואת הנתונים שהם מציגים בזמן אמת כך שיהיה הרבה יותר נוח לזהות נתונים מסויימים שקופצים לעין.
2. לאחר שסידרתם את כל אלה, ואתם מרגישים מספיק מוכנים להתחיל את התהליך תוודאו שכל אפליקציה התחילה להקליט.
3. התחלת האזנה של WireShark לשמירת התעבורה אשר תתחולל. שימו לב להגדיר בתחילת ההאזנה שמירה לקובץ - בשאיפה עם חלוקה לקבצים.
4. הרצת הנוזקה.



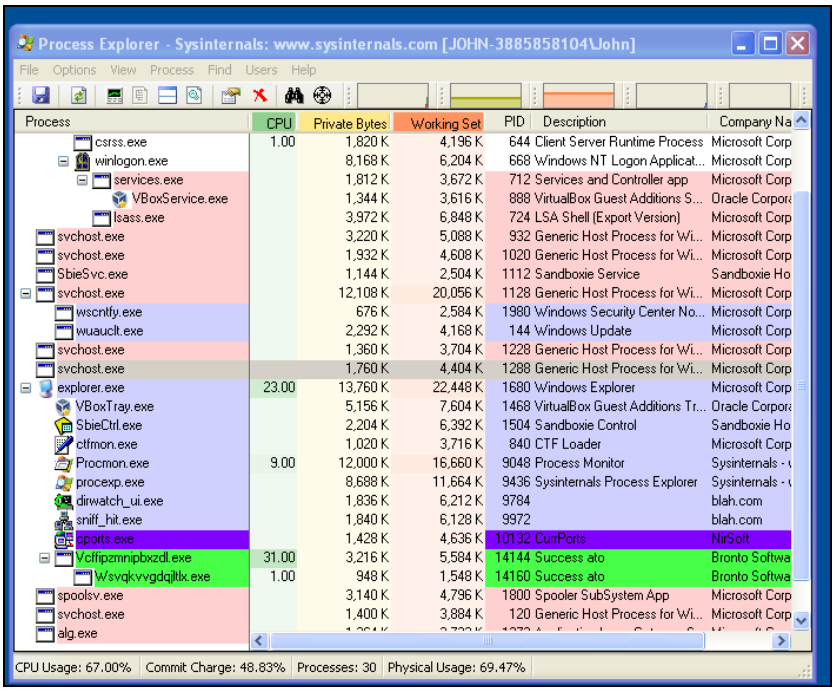
למשתמשי VirtualBox - שימו לב שבעזרת מקש ההוסט שלכם (בדרך כלל הקונטרול הימני) והלחצן P במקלדת אתם יכולים להקפיא את ההרצאה של כל המערכת הוירטואלית בכל שלב ולבחון מה קורה.

הרצת הנוזקה

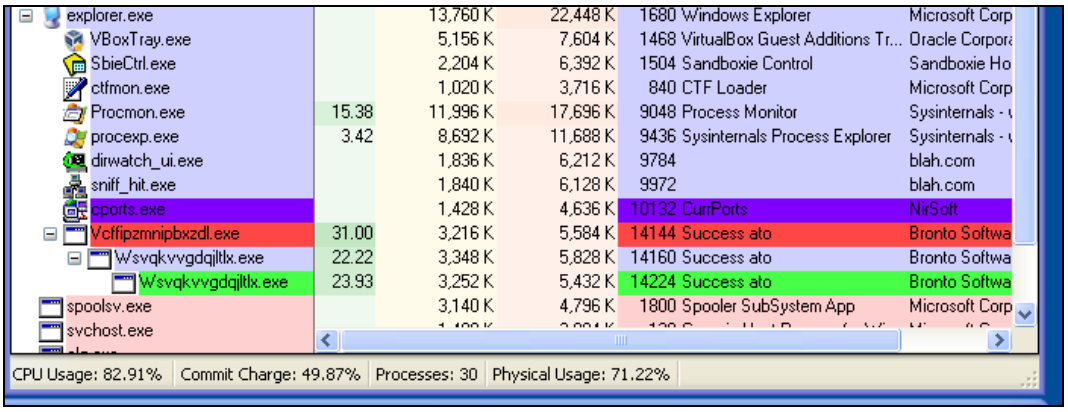
כאשר אתם מוכנים, גשו לתיקיה בה שמרתם את הנוזקה וטענו אותה לזכרון. במקרה הזה הנוזקה נשמרה תחת תיקיה עם השם ההולם Big Mean and Scary CryptoLocker.



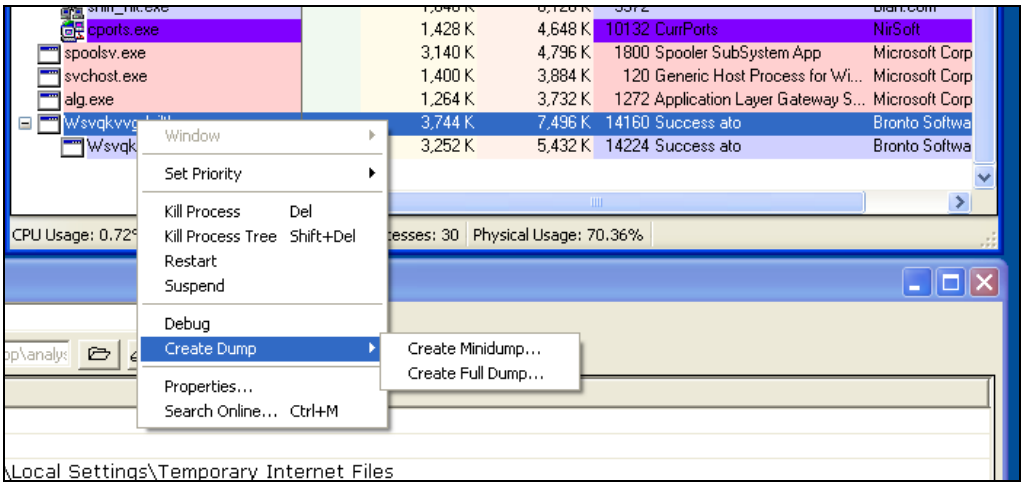
שימו לב שמיד לאחר ההרצה הקובץ 'נמס' (melted). זהו מנגנון הגנה נפוץ בקרב נוזקות אשר מוחק את הקובץ המקורי כך שיהיה קשה יותר להשיג דוגמא חיה של ההדבקה המקורית ולנתח אותה. לאחר ההרצה הראשונה של הנוזקה CryptoLocker, נסתכל מה קורה: אנחנו יכולים ישר לשים לב שהתהליך יוצר תהליך בן דרך Process Explorer.



התהליך הראשי נסגר ונמחק בשלב זה:

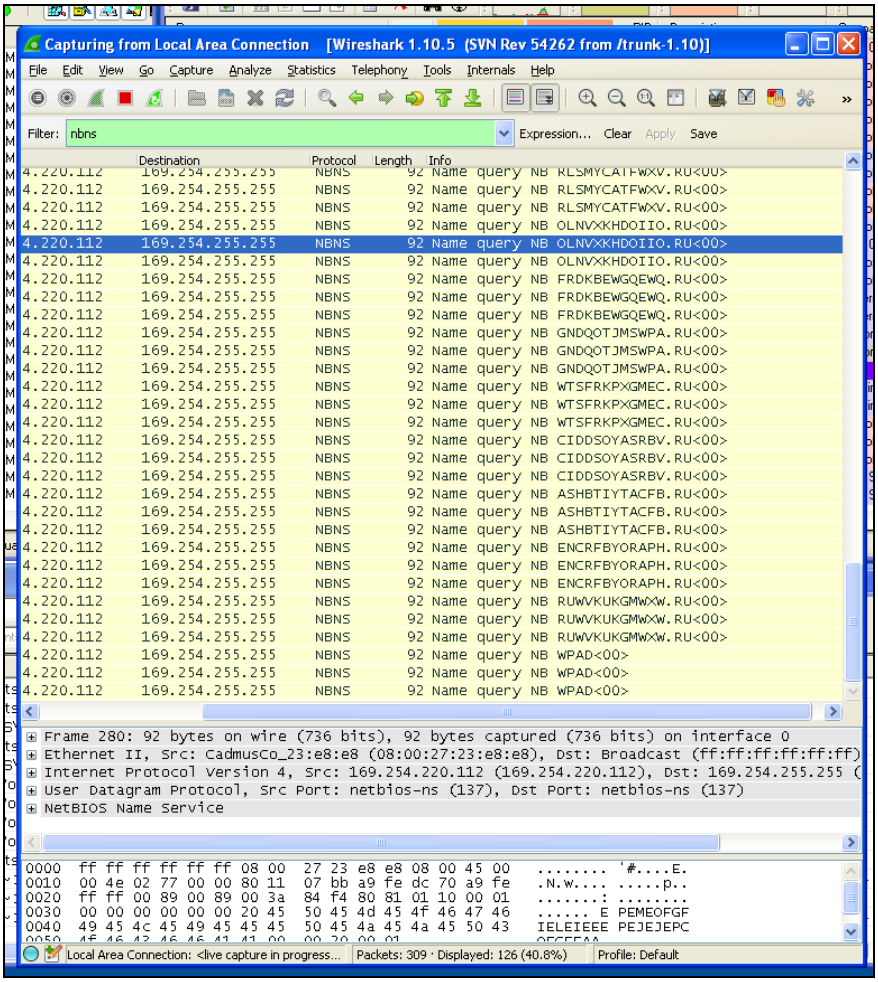


לאחר מכן מומלץ לקחת תמונה של זיכרון התהליך לניתוח מאוחר יותר:

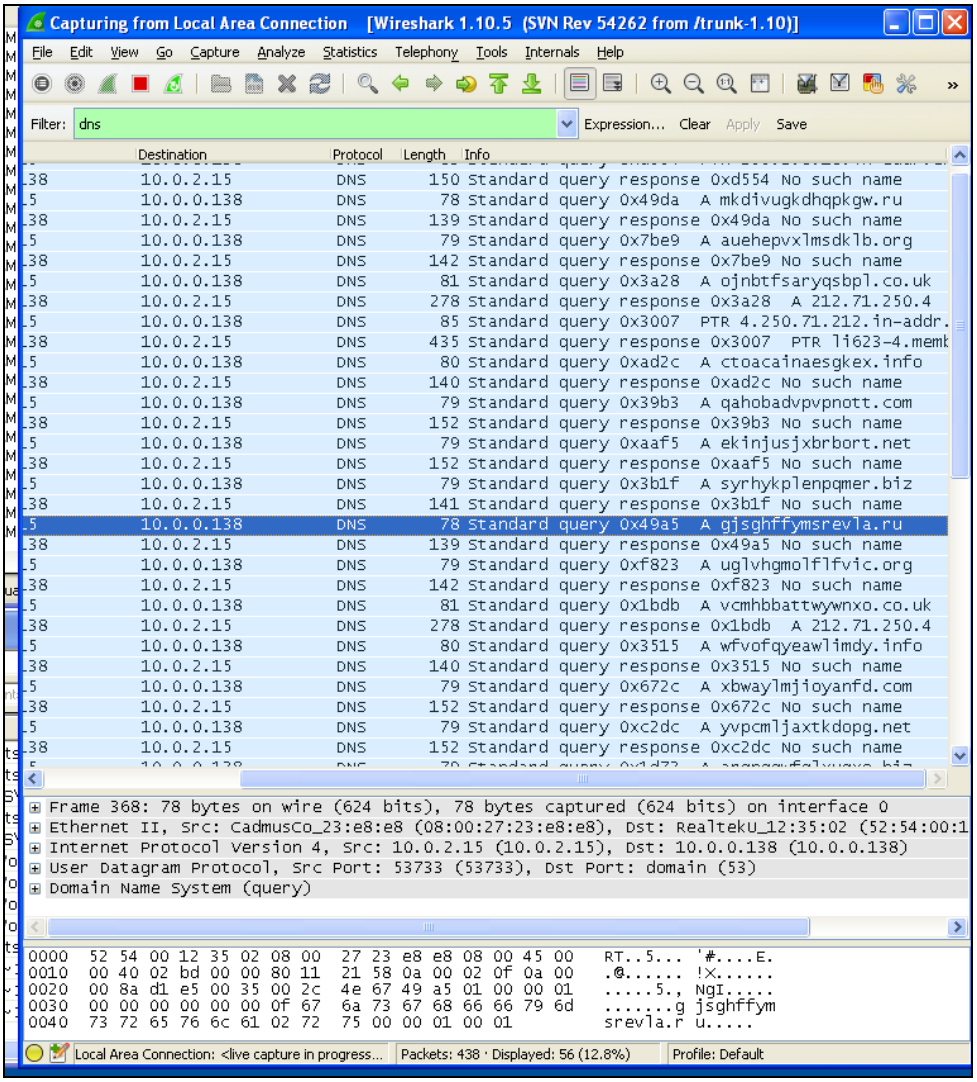


בשלב זה נוכל להתחיל לראות את קריפטולוקר ממפה את הדיסק לפי קבצים מסויימים ושומר אותם לערכים ברג'יסטרי של המערכת להצפנה מאוחרת יותר. מומלץ בנוסף לאורך כל כמה דקות לבצע SnapShot למכונה על מנת שתוכלו לחזור בכל נקודה אחורה ולראות מה קרה שם או מה השתנה. סוג של Rewind בנגני מדיה.

כאשר המכונה לא מחוברת לאינטרנט אנחנו יכולים לראות שבמקום בקשות DNS המערכת מגישה בקשות NBNS:



כאן אנחנו יכולים לראות את בקשות ה-DNS המשונות ואת האחת שבמקרה כן חזרה ונפתרה ל-212.71.260.4:



ניתוח הממצאים

מה הצלחנו להבין על CryptoLocker?

- הצלחנו לראות את התהליכים שיוצרת הנוזקה ולהיכן שומרת אותם.
- הצלחנו לזהות ערכים ברג'יסטרי אליהם ניגשת התוכנה.
- ראינו איזה סוג תעבורה (בעיקר בקשות DNS) יוצרת התוכנה (וכיצד נוכל לזהות אותה מבחוץ).
- הבנו איך היא מצפינה את הקבצים ובעזרת מה.

יחסית לא רע לכמה דקות עבודה.

סיכום

במאמר הזה ניסינו להסביר קצת מבוא לנושא ניתוח הנוזקות ברמה הנוחה והמהירה ביותר. הרעיון הוא חלק מפרויקט שמנסה לעודד אנשים (גם לא מתחום אבטחת מידע) להתנסות במשחקים האלה. להכיר את הכלים שתוקפים אותם. לא לחשוב עליהם כעל קסם מיוחד שקורה ברקע ושחברות האנטי-וירוסים צריכות להתמודד איתו אלא משהוא שכל אחד מאיתנו יכול לקחת, לשחק, לפרק, להרכיב מחדש, להבין איך עובד וכתוצאה מכך, גם אם לא לספק חיסון, לייצר תלאי לבינתיים. הדוגמא שבחרנו עם CryptoLocker היא דוגמא נהדרת מסיבה מאוד פשוטה - קל לזהות את התבנית של תקשורת ה-DNS היוצאת. מכיוון ש-CryptoLocker מבקש מפתח משרת האם היא אינה מצפינה את הקבצים ללא תקשורת מוקדמת עם השרת. המשמעות היא שעל ידי ניתוח התנהגותי פשוט והוספת חוק לגבי חיפוש שרתי DNS ניתן יחסית בקלות למנוע את הנזק העיקרי שיוצרת הנוזקה.

לסיכומו של דבר - בהצלחה ☺