

ניתוח התולעת Waledac

מאת מיתר קרן ויונתן גולדהירש

הקדמה להקדמה

מאמר זה הינו דו"ח סופי שהוגש כחלק מ-"236349: פרויקט באבטחת מידע" ע"י מיתר קרן ויונתן גולדהירש כחלק מלימודיהם בטכניון, המאמר עצמו הוגש ב-2008 ועוסק במחקר Botnet בשם Waledac. כיום (2013) ה-Botnet אינו פעיל, אך שיטות המחקר, הכלים והדרך שבה פעלו מיתר ויונתן על מנת לחקור את דרכי ההדבקה, התקשורת, המבנה וההתנהגות של התולעת עדיין רלוונטיות ונמצאות בשימוש גם כיום.

הקדמה

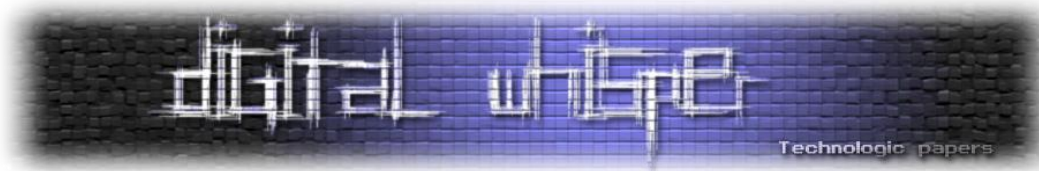
Botnets - רשתות בוט הוא מונח שמתאר קבוצה של מחשבים המריצים תוכנה (במקרה שלנו זדונית), המאפשרת לישות כלשהי להשתמש במחשבים הללו לצרכיה. מחשבים אלה נקראים מחשבי "זומבי". בדרך כלל בעליהם אינם יודעים שמחשביהם בתוך רשת-בוט ומבצעים את הוראות רשת-הבוט.

רשתות-בוט משמשות לרוב לשליחת SPAM, הפצת תוכנה זדונית, גניבת מידע, ביצוע התקפות רשת-הבוט. טקטיקות אלה כוללות הצפנת התקשורת, שימוש ברשת peer-2-peer, שימוש ב-rootkits על מנת להסתיר את הפעילות ממערכת ההפעלה, שימוש במנגנון fast-fluxing על מנת להסתיר זהות המחשבים, ועוד.

רשתות-בוט משתמשות במספר טקטיקות על מנת למנוע זיהוי של בעליהן ועל מנת להקשות על הורדת רשת-הבוט. טקטיקות אלה כוללות הצפנת התקשורת, שימוש ברשת peer-2-peer, שימוש ב-rootkits על מנת להסתיר את הפעילות ממערכת ההפעלה, שימוש במנגנון fast-fluxing על מנת להסתיר זהות המחשבים, ועוד.

לפי הערכות¹ קרוב לרבע ממחשבי האינטרנט חברים ברשתות-בוט. רשת הבוט Kraken² הוערכה בכ-400 אלף מחשבים באפריל 2008, והדביקה לפחות עשירית מהחברות ב-Fortune 500. Srizbi היתה אחראית³, בשיאה, ל-39% מהספאם בעולם, ול-21% מכל תעבורת הדוא"ל בעולם.

¹ Criminals "may overwhelm the web", BBC, 25 January 2007, <http://news.bbc.co.uk/1/hi/business/6298641.stm>



הערכות שמרניות על Storm⁴ מדברות על כ-160,000 מחשבים החברים ברשת והערכות אחרות מדברות על 50 מליון. Conficker, אחת התולעים המפורסמות ביותר בעולם, מוערכת בכ-10 מליון מחשבים. אנו נשתמש במונח "תולעת"⁵ על מנת לתאר את התוכנה הרצה על מחשב "זומבי" מסויים.

תולעת ה-WALEDAC - הופיעה לראשונה ברחבי האינטרנט בדצמבר 2008⁶, כאשר החלה להפיץ עצמה בעזרת הודעות אימייל פיקטיביות, המפנות לאתר אינטרנט בדוי ובו לינקים להורדת התולעת תחת כסות אחרת.⁸⁷ מאוחר יותר, החלה Waledac להפיץ הודעות שתוכנן מבוסס על מיקום הנתקף.⁹

לפי ניתוחים קיימים¹⁰, משמשת התולעת להפצת SPAM, גניבת כתובות דואר אלקטרוני, שימוש כ-Proxy לתקשורת, האזנה לתעבורת רשת, השתתפות בפעולות DDOS וכן מסוגלת לקבל ולבצע פקודות מרחוק.

עם ההדבקה, יוצרת התולעת מספר כניסות ב-Registry, מתחילה בסריקה של הקבצים במחשב, ויוצרת קשר עם שרתים מרוחקים.

רשת ה-Waledac משתמשת במנגנון Fast-Fluxing¹¹ (מנגנון המשנה במהירות את שרתי ה-Web עבור דומיין מסויים), דבר המקשה על איתור שרתי הפיקוד, ועל הורדת שרתי ה-Web.

קיימות הערכות בקרב החוקרים הקושרים בין Waledac לתולעת ההיסטורית Storm¹² אם כי דעה זו אינה מקובלת על הכל.

² Wikipedia, Kraken Botnet, http://en.wikipedia.org/wiki/Kraken_botnet

³ Srizbi Botnet, Wikipedia, http://en.wikipedia.org/wiki/Srizbi_botnet

⁴ Storm Botnet, Wikipedia, http://en.wikipedia.org/wiki/Storm_botnet

⁵ Wikipedia, Computer Worm, http://en.wikipedia.org/wiki/Computer_worm

⁶ Infiltrating WALEDAC Botnet's Covert Operations, TrendMicro, pg. 3

http://us.trendmicro.com/imperia/md/content/us/pdf/threats/securitylibrary/infiltrating_the_waledac_botnet_v2.pdf

⁷ Waledac Trojan Hosted by Fake Obama Website, Threat Research & Response Blog, Microsoft Malware Protection Center, <http://blogs.technet.com/mmpc/archive/2009/01/19/waledac-trojan-hosted-by-fake-obama-website.aspx>

⁸ W32.WaleDac Analysis, Bughira's Blog, <http://bughira.wordpress.com/2009/01/28/w32waledac-analysis/>

⁹ Waledac Localizes Social Engineering, TrendLabs Malware Blog, <http://blog.trendmicro.com/waledac-localizes-social-engineering/>

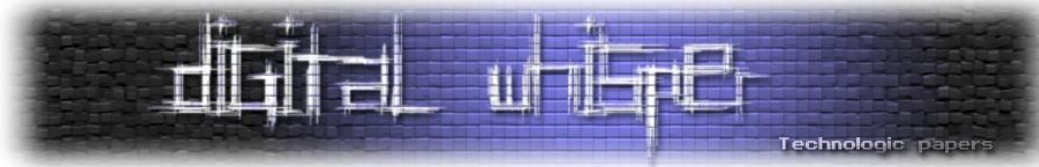
¹⁰ W32/Waledac, Threat Research & Response Blog, Microsoft Malware Protection Center, <http://www.microsoft.com/security/portal/Entry.aspx?Name=Win32%2fWaledac>

¹¹ W32.Waledac Threat Analysis, Symantec Security Response, pg. 5

http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/W32_Waledac.pdf

¹² Storm Worm Reincarnates as Waledac, SecurityProNews,

<http://www.securitypronews.com/insiderreports/insider/spn-49-20081231StormWormReincarnatesAsWaledac.html>



ערוץ הפצה נוסף של ה-Waledac נוצר כאשר וריאנטים מסויימים של תולעת ה-Conficker החלו להפיץ אותו גם כן¹³. יש גם דיווחים על קשר בין RBN (Russian Business Network) לבין Waledac¹⁴.

תולעת ה-Waledac מפיצה קמפיינים של "Canadian Pharmacy", אשר לפי SPAMhaus נכון ל-20.9.08 הוא תרמית הספאם הגדולה באינטרנט. לפי הערכות, "בית מרקחת" זה מכניס כ-150 מיליון דולר בשנה¹⁵.

לפי הערכות של TrendMicro¹⁶ נכון לאפריל 2009 התולעת מסוגלת לשלוח לפחות 924 מיליון הודעות דואר זבל ביום. בנוסף הם מעריכים שיש כ-600 תחנות "ממסר" ו-6,600 מחשבי "עבד". הערכות עדכניות יותר מדברות על כמה עשרות אלפי מחשבים המתפקדים כ"עבדים"¹⁷ - לפחות עשרים אלף, מה שיכפיל יכולת זו פי שלוש.

הדבקה

סביבת המעבדה

לצורך בקרה על מהלך ההדבקה, החלטנו לבצע את נסיונות ההדבקה בתוך מערכת הפעלה שתרוץ בסביבת אמולציה. מחשב הניסוי הריץ Ubuntu Linux 8.10 ועל גביו רצה מכונה וירטואלית על ידי QEMU עם מערכת הפעלה Windows XP SP1. על המכונה הווירטואלית הותקנו Office 2003 וכן כלים סטנדרטיים לניטור Registry, System Calls, File System, Strace, Filemon, Regmon. על המחשב המארח הותקנה תוכנת Wireshark לניטור התקשורת. המחשב המארח חובר לאינטרנט באמצעות נתב ביתי.

הדבקות פאסיבית

מחקרים בתחום honeynets טוענים¹⁸ שמחשב חסר עדכונים המחובר בחיבור חשוף לאינטרנט ידבק בתולעת תוך מספר דקות. עם זאת, נתקלנו בקושי לחבר את המכונה הווירטואלית באופן ישיר לאינטרנט, באופן שיהיה ניתן ליצור איתה קשר מבחוץ. הקושי נבע ראשית מכך שבתשתית האינטרנט שלנו אנו

¹³ Win32/Conficker teams up with Win32/Waledac, CA Security Advisor Research Blog, <http://community.ca.com/blogs/securityadvisor/archive/2009/04/15/win32-conficker-teams-up-with-win32-waledac.aspx>

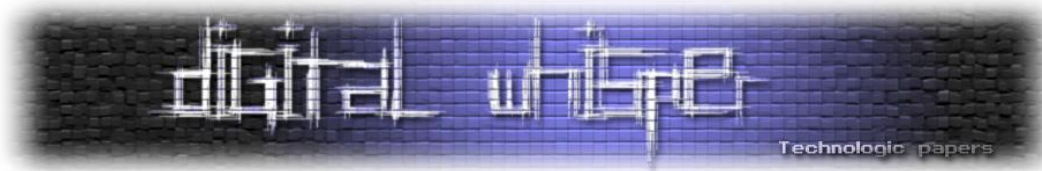
¹⁴ Lavasoft, Waledac questions answered, <http://www.lavasoft.com/mylavasoft/company/blog/waledac-questions-answered>

¹⁵ Dark Reading, <http://www.darkreading.com/security/management/showArticle.jhtml?articleID=211201114>

¹⁶ Infiltrating WALEDAC Botnet's Covert Operations, TrendMicro, pg. 33

¹⁷ Waledac botnet being prepared to send Independence Day-related spam, SC Magazine, <http://www.scmagazineuk.com/waledac-botnet-being-prepared-to-send-independence-day-related-spam/article/139504/>

¹⁸ Getting Information With The Help of Honeynets, The HoneyNet Project, <http://www.honeynet.org/node/59>



מחברים לאינטרנט מאחורי נתב ביתי, ולכן יש NAT המסתיר את המחשב. על זאת ניתן להתגבר על ידי הגדרת המחשב כ-DMZ, אבל אז היינו צריכים ליצור קשר ישיר בין המכונה הווירטואלית החוצה, ולאחר שהקדשנו זמן רב לנסיונות כושלים להקמת קשר כזה, החלטנו לנטוש כיוון הדבקה זה. לכן לא התבצע ניסוי הדבקה כזה.

הדבקות אקטיבית

על מנת לחקות הדבקות "טבעית" בתולעת החלטנו לנסות להדבק מדואר אלקטרוני. לשם כך נוצר חשבון Gmail אליו העברנו דואר מתיקית ה-SPAM של חשבונות הדוא"ל הרגילים שלנו. לאחר מכן, הפעלנו את חשבון הדואר במחשב הווירטואלי, הפעלנו את כלי הניטור, ולכל אחד מפרטי הדואר - קראנו אותו וביקרנו בקישורים. לאחר מספר הצעות לשיפור חיי המין, נתקלנו בדואר הבא:

From: Sarah <dan@sg.statschippac.com>
Date: Tue, Mar 31, 2009 at 5:25 PM
Subject: Damned terrorists!!!
To: meitark@gmail.com

Are you in the city now? <http://peulp.blogsitedirect.com/news.php>

עם הכניסה לקישור הגענו לאתר הבא:

Powerful explosion burst in Tel Aviv-yafo this morning.


At least 12 people have been killed and more than 40 wounded in a bomb blast near market in Tel Aviv-yafo. Authorities suggested that explosion was caused by "dirty" bomb. Police said the bomb was detonated from close by using electric cables. "It was awful" said the eyewitness about blast that he heard from his shop. "It made the floor shake. So many people were running"

Until now there has been no claim of responsibility.

Powerful explosion burst in Tel Aviv-yafo this morning.

At least 12 people have been killed and more than 40 wounded in a bomb blast near market in Tel Aviv-yafo. Authorities suggested that explosion was caused by "dirty" bomb. Police said the bomb was detonated from close by using electric cables. "It was awful" said the eyewitness about blast that he heard from his shop. "It made the floor shake. So many people were running"

Until now there has been no claim of responsibility.

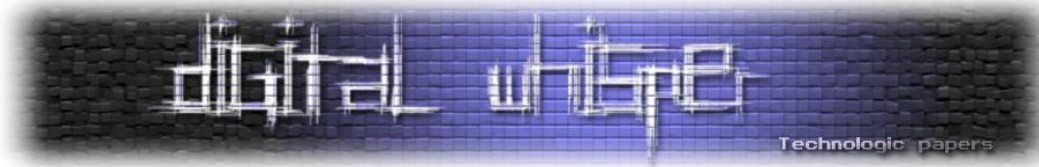


You need the latest Flash player to view video content. [Click here to download.](#)

Related Links:
http://en.wikipedia.org/wiki/Dirty_bomb
<http://www.google.com/search?q=Tel+Aviv-yafo+terror+attack>

נשים לב למספר פרטים - הדף (וכותרתו) מדווחים על פיגוע בתל אביב - חיפוש (מאוחר יותר) באינטרנט מעלה¹⁹ כי אתר דומה עולה עם פרטי מיקום שונים - לפי מיקום הקורבן הניגש אליו. נשים לב גם לקישורים הלגיטימיים שבסוף הדף שמוסיפים לאמינותו בעיני הנתקף. לחיצה על הקישור Click here או

¹⁹ Waledac: Reuters Video News Social Engineering , Countermeasures: Security, Privacy & Trust, Trendmicro Blogs, <http://countermeasures.trendmicro.eu/waledac-reuters-video-news-social-engineering>



על התמונה שנראית כסרטון דמוי Youtube היא קישור להורדת קובץ news.exe שהוא קובץ הרצה בינארי של התולעת WALEDAC (זאת גילינו מאוחר יותר. עוד על כך בהמשך).

יש לציין שתוקפו של הקישור מוגבל, וזמן קצר לאחר מכן הוא הוביל לאתר בעל לגיטימיות מפוקפקת למכירת תרופות²⁰. בדיקת פרטי WHOIS של הדומיין מראים כי הוא נרשם על ידי:

SHANGGUANMING GONGYUWUYEYOUXIANGONGSI
jongchangde@126.com
QIANJIN, 2005451

בתאריך 18.03.09 - זמן לא רב לפני שביקרנו בו.

זיהוי ההדבקה והתנהגות מקומית

לאחר הורדת והרצת התולעת, שמנו לב למספר פעילויות מצידה:

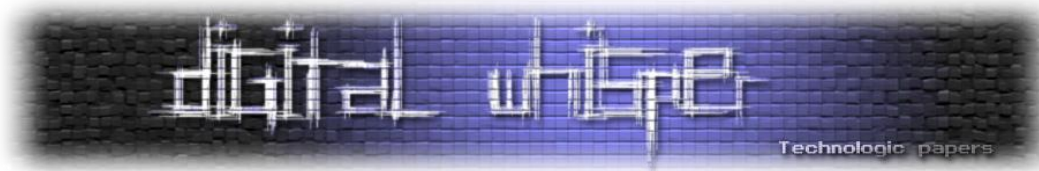
- סריקת תוכן הדיסק הקשיח.
- כתיבה לחמש כניסות ב-Registry.
- יצירת תקשורת החוצה.

ניטור התנהגות הקובץ news.exe גילה שתי פעילויות מרכזיות:

סריקת הדיסק הקשיח - קבצי ה-log של כלי ה-Filemon מלמדים כי התולעת עוברת על כל תיקיות הדיסק הקשיח באופן סדרתי, פותחת קבצים וקוראת את כל תוכנם. ניכר כי ישנה סלקטיביות מסוימת בבחירת הקבצים, לפי הסיומות לפחות. כפי שדווח²¹, גם אצלנו ניכר כי התולעת דילגה על קבצי exe, bmp ונוספים, אך לא החמיצה כלל קבצי lnk, txt, ini, tmp, pf. בניגוד למדווח, מצאנו גישות של התולעת לקבצי dll, אבל רק למספר קבצים נבחר ולא לכלל הקבצים, מה שכנראה מצביע על כך שהגישה הייתה לצורך שימוש בהם ולא כחלק מהסריקה הכללית, עדות נוספת לטובת הסברה הזאת היא כך שהייתה גישה חוזרת ונשנית לקבצים אלה.

²⁰ Canadian Pharmacy, EU Spam Trackers, http://www.spamtrackers.eu/wiki/index.php?title=Canadian_Pharmacy

²¹ Email-Worm:W32/Waledac.A, F-Secure Security Lab, http://www.f-secure.com/v-descs/email-worm_w32_waledac_a.shtml



קבצי ה-dll אליהם ניגשה התולעת:

```
ntdll.dll, kernel32.dll, user32.dll, gdi32.dll, advapi32.dll,
rpcrt4.dll, psapi.dll, dnsapi.dll, msvcrt.dll, ws2_32.dll, shlwapi.dll,
iphlpapi.dll, ole32.dll, oleaut32.dll, shell32.dll, shlwapi.dll,
comctl32.dll, crypt32.dll, msasn1.dll, wininet.dll, netapi32.dll,
rsaenh.dll, wpcap.dll, uxtheme.dll, secur32.dll, wsock32.dll,
rasapi32.dll, rasman.dll, tapi32.dll, rtutils.dll, winmm.dll,
sensapi.dll, urlmon.dll, version.dll, mswsock.dll, wshtcpip.dll,
winnr.dll, wldap32.dll, rasadhlp.dll, apphelp.dll
```

- **גישה ל-Registry** - קבצי ה-log של כלי ה-Regmon מלמדים כי התולעת מבצעת קריאה של ערכים הקשורים ב-Internet Settings, Winlogon, Winsock2, DNSCache, Tcpip parameters, Tracing, Sound drivers, Terminal Server - מה שתואם לזהות ה-dll-ים אליהם ניגשה התולעת - גישה לרשת, ניטור תהליכים, גישה מרחוק, ובמפתיע, שימושי מולטימדיה (מפתיע שכן אין סיבה לתולעת להשתמש בספריות לנגינת קול ווידאו, יש להניח שאין זה מכוון).
כמו כן, התולעת מבצעת מספר כתיבות ל-Registry הנתיב לתולעת נשתל ב:

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\PromoReg
```

לאחר מכן, התולעת מעדכנת מספר רב של פעמים (כמה מאות) את:

```
HKCU\Software\Microsoft\Windows\CurrentVersion\RList
```

עם ערך בינארי כלשהו, וכן כותבת פעם אחת ערך בינארי כלשהו ל:

```
HKCU\Software\Microsoft\Windows\CurrentVersion\MyID
HKCU\Software\Microsoft\Windows\CurrentVersion\FWDone
HKCU\Software\Microsoft\Windows\CurrentVersion\LastCommandId
```

נוסף לכך, התולעת כותבת לכמה כניסות הקשורות ב-Cache ו-Temporary Internet Files:

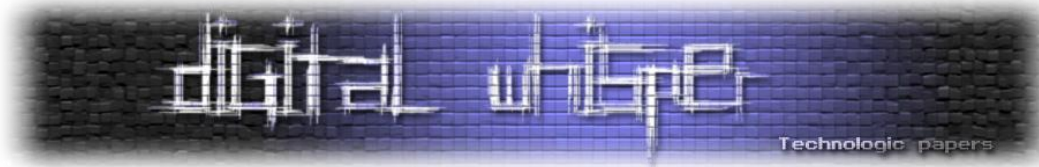
```
HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed
```

אך נראה שהדבר הוא באופן אוטומטי כחלק משימוש שלה בספריות Windows סטנדרטיות, ולא כחלק מפעילות עויינת.

מהפעילות של הקובץ, וכן מהעובדה שלא קיבלנו שום הודעה על המסך, ניתן היה להבין שזו תולעת. הקובץ היה מלכתחילה "חשוד", שכן הגיע מדואר שסווג כ-SPAN, וכן כי כל ה"נגן" שבדף היה קישור - וזה אינו תואם לאופן שבו מתקנים נגן Flash. כמו כן, סריקת הדיסק הקשיח תואמת להתנהגות סבירה של חיפוש אחר איזו תבנית טקסטואלית (ולכן ההתעלמות מקבצים בעלי אופי לא-טקסטואלי).

ניתוח התולעת Waledac

www.DigitalWhisper.co.il



והגישה ל-Registry - חלקה "שתילת" התולעת (הכנסת ערך ל-Run שיגרום לכך שתופעל עם הפעלת המחשב), חלקה כנראה קשור לספריות השונות שהיא משתמשת בהן (הצפנה ותקשורת), וערכים נוספים כנראה קשורים לפעילות התולעת עצמה - rlist, myid, lastcommandid. יכול להיות שחלק מהפעילות ב-Registry נועד גם לחשיפת המחשב (הכתיבה ל-internet settings מבטלת שימוש בשרתי Proxy).

זיהוי התולעת

ניתוח הכתובות ל-Registry הבליט כתיבות מרובות לערך RList מה שגרם לנו להבין שניתן לראות בזה מאפיין ברור של התנהגותה. חיפוש אחר שם ערך זה באינטרנט על מנת להבין את משמעותה, העלה כי הוא מאפיין של תולעת ה-Waledac²². על מנת לאושש אבחנה זו, בדקנו וראינו כי גם כניסות ה-Registry האחרות שאליהן כתב הקובץ מדווחות כמאפיינות את התנהגות תולעת זו. כמו כן, ראינו כי סריקת הקבצים (וכן ההתמקדות בקבצים בעלי תוכן טקסטואלי) מאפיינים אותה, וכן האתר הספציפי דרכו נדבקנו²³.

הסברה הנפוצה ברוב האתרים (לדוגמא²⁴) היא כי התוכנה סורקת אחר כתובות אימייל, וכותבת ל-RList כתובות של שרתים מרוחקים איתן היא מתקשרת. לא הצלחנו לאשר סברה זו בעצמנו, אם כי מצאנו קשר בין גישה לערך Registry זה לבין כתובות מחשבים מרוחקים הנמצאים בזיכרון הריצה (מפורט בפרק הדיסאסמבלי).

בזמן ההדבקה לא הייתה לנו תוכנת אנטי-וירוס כלשהי על העמדה, כדי שלא תשבש את התנהגות התולעת. מאוחר יותר ניסינו לבדוק האם תוכנת אנטי-וירוס סטנדרטית מזהה את התולעת, ואכן AVG מתריע עליו כ-"Virus Identified Win32/Cryptor".

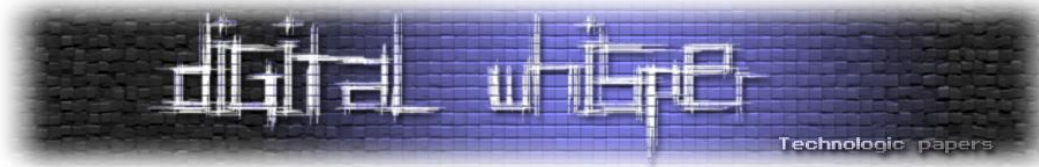
מצאנו מספר גדול של תוכנות זדוניות שמזהות עם שם זה ודומים לו, מה שגורם לנו לחשוב שזו עשויה להיות איזו אבחנה גנרית שמבוססת על כך שהקובץ ארוז. בכל מקרה, מסתבר שידוע²⁵ כי AVG מספק אבחנה זו לתולעת ה-Waledac.

²² Threat Encyclopedia, Microsoft Malware Protection Center, <http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Trojan%3aWin32%2fWaledac.gen%21A>

²³ Waledac Reuters Theme – Security Labs Alert, <http://securitylabs.websense.com/content/Alerts/3321.aspx>

²⁴ W32.Waledac, Symantec, http://www.symantec.com/security_response/writeup.jsp?docid=2008-122308-1429-99&tabid=2

²⁵ W32/Waledac, McAfee, http://vil.nai.com/vil/content/v_207110.htm

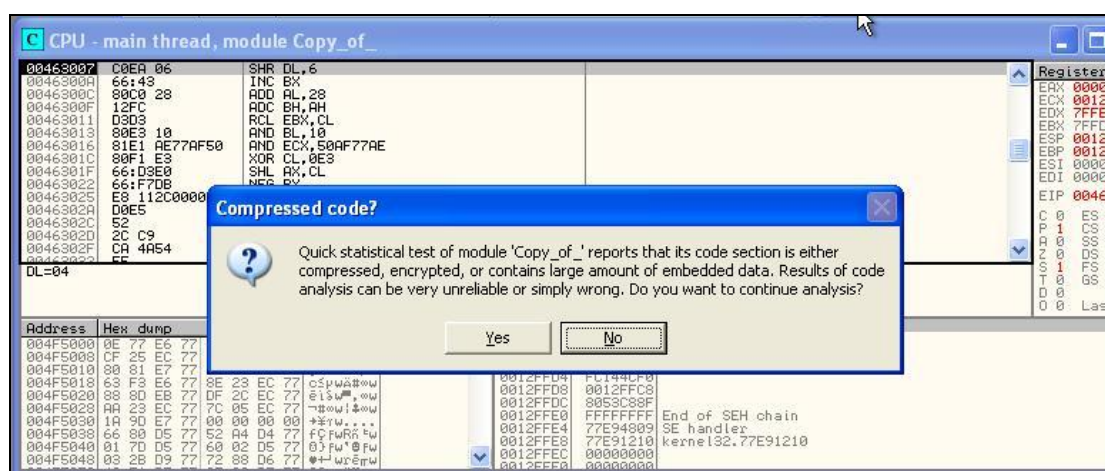


דיסאמבלי של התולעת

טכניקת הביצוע

לפי הידוע²⁶ הקובץ הבינארי של Waledac עטוף ומוצפן במספר דרכים. ב²⁶ מצוין כי בגרסא F של התולעת יש שימוש ב-code obfuscation, וכן בעטיפה באמצעות UPX ובאמצעות כלי ייחודי לתולעת. כמו כן, לפי²⁶, מפעיל Waledac מנגנונים לגילוי הרצתו תחת debugger.

הכלי ששימש אותנו לביצוע הניתוח היה OllyDbg. הפתיחה הראשונית ב-OllyDbg גם כן תומכת בסברה שהקובץ ארוז:



כן, האתגר המרכזי שעומד בפנינו הוא השגת גישה לקובץ מפוענח, והאתגר הבא הוא הימנעות מגילוי פעילותנו על ידי התולעת. יצוין כי לא מצאנו התנהגות של התולעת בתגובה להרצה ב-debugger, אבל כן נמצאו עדויות לכך שהיא אכן מחפשת אחר כזה (על כך בתת הפרק "פרמטרים להתנהגות").

נסיון הפרישה הראשון התבסס על הנחה (או שמא - משאלת לב) כי הקובץ ארוז באופן פשוט. הרצנו את הקובץ ב-debugger צעד אחר צעד בנסיון להתחקות אחר חוקיות בהתנהגותו. שמנו לב לרצף ארוך של קפיצות ללא חזרה, וכך לא הצלחנו להבין כל דבר בנוגע להתנהגותו.

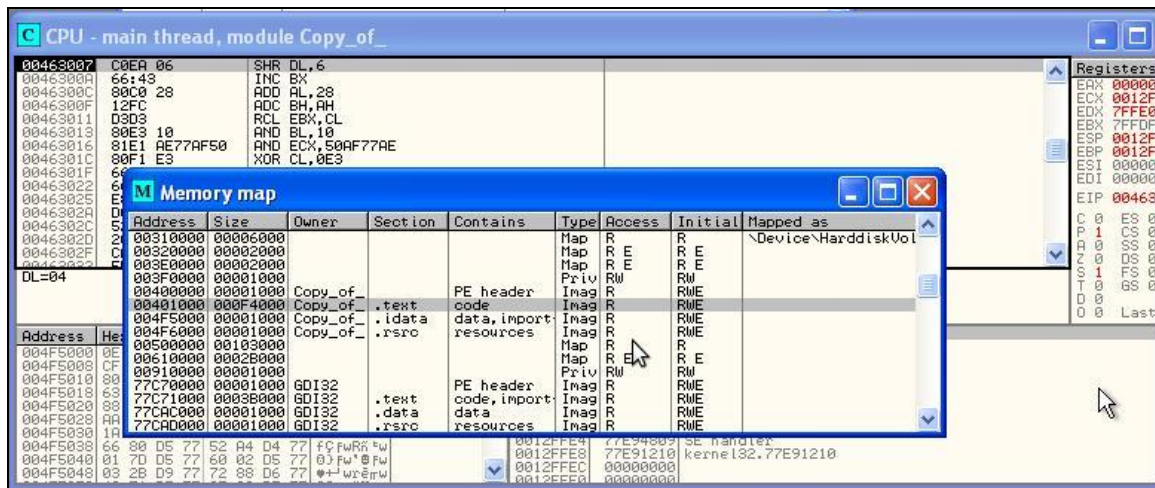
ביצענו מספר נסיונות לפרישת הקובץ בעזרת הכלים WinUPack, UPX ו-NSPack המיועדים לפרישת עטיפות מסוגים אלה, אבל אלה נכשלו. בדיעבד ניתן לייחס את כשלונם לעטיפה החיצונית שייחודית לתולעת.

²⁶ W32.Waledac Threat Analysis, Symantec Security Response, pg. 2

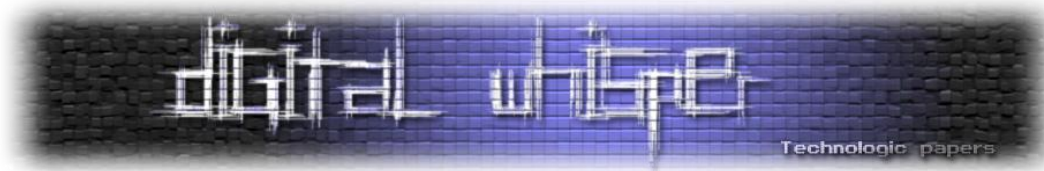
http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/W32_Waledac.pdf

בסופו של דבר, הטכניקה לחשיפת הקובץ הפרוש, שנתגלתה על ידי ניסוי-וטעיה היא זו:

1. הרץ את הקובץ בתוך OllyDbg, כשבשורת הסטטוס מדווח על יצירת חוטמים בצע pause.
 2. העלה את מפת הזכרון (Alt-M).
 3. קבע Breakpoint על Code (F2).
 4. המשיך את הרצת הקובץ.
 5. כאשר יגיע ל-Breakpoint, בצע אנליזה מחדש (Ctrl-A).
- שיטה זו גורמת לכך שהקובץ "מקלף" את האריזה בעצמו, ואז ניתן לבצע אנליזה לקוד החשוף.



[שימו לב לטכניקה הדו-עכברית שבשימוש בתמונה זו. על מנת לאתר פריטים מסוימים, חיפשנו אחר מחרוזות מתאימות או אחר קריאות מערכת רלוונטיות, ועקבנו אחר הפניות אליהם.]



ממצאים

תעודות ומפתחות

בספרות מקובלת הסברה²⁷ כי נמצאים בקוד שני מפתחות AES מקודדים בו, וכן תעודה ציבורית. מפתחות ה-AES משמשים להצפנת הערכים ב-Registry.²⁸ כמו כן, ישנו מפתח hard-coded לפענוח עדכוני תוכנה²⁹. בתעודה הדיגיטלית נתקלנו על ידי מעבר ידני על זיכרון הריצה (לא ידענו לחפש אחריה בזמן), אם כי פשוט למצוא אותה על ידי חיפוש טקסטואלי אחר המחרוזת "CERTIFICATE":



בקטע זה, הנשלף מזיכרון הריצה של התוכנית, ניתן לראות את התעודה הדיגיטלית שבשימוש התוכנית (ההתחלה והסוף מסומנים בכחול). כאמור, לפי²⁷ אלה מפתחות AES המיועדים להצפנת הערך ב: HKCU\Software\Microsoft\Windows\CurrentVersion\RList (להלן יקרא ה-RList).

בפענוח התעודה נתקלנו במספר קשיים - יצאנו את התעודה מהזיכרון במספר פורמטים שונים, וניסינו להשתמש בכלי של openssl כדי לקרוא את תוכנה. בתחילה נתקלנו בהודעת השגיאה unable to load certificate: no start line, ואחרי חלוקת begin/end certificate לשורות משל עצמם, קיבלנו את ההודעה bad base64 decode. ניסינו גם לייצא בפורמט בינארי, ולהמיר אותו באמצעות תוכנה פשוטה, אבל גם זה נכשל.

בהשוואה לתעודה ב-²⁷ ראינו כי בתעודה אצלנו מופיעות נקודות, בעוד שם לא. בחינת הזיכרון ב-OllyDbg הראתה שערכן של הנקודות הוא 0A - כלומר line feed. כך, החלפנו את הנקודות בירידת שורה ואז הצלחנו לקרוא את התעודה בעזרת שורת הפקודה:

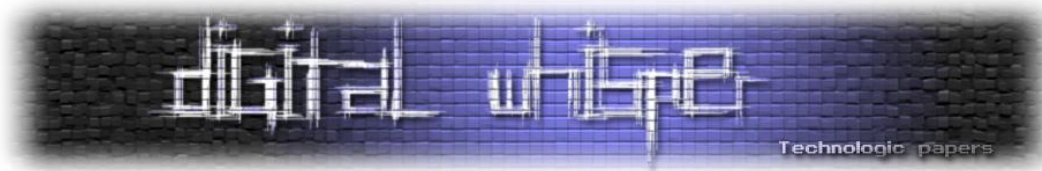
```
openssl x509 -text -noout -in filename
```

²⁷ Infiltrating WALEDAC Botnet's Covert Operations. TrendMicro. pg. 7

28 Decoding Waledac's Registry,

<http://www.nnl-labs.com/cblog/index.php?/archives/9-Decoding-Waledacs-Registry.html>

²⁹ W32.Waledac Threat Analysis, Symantec Security Response, pg. 12



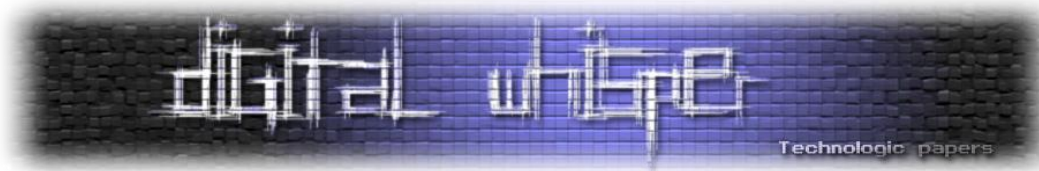
כך קיבלנו את המידע:

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      bb:c5:91:63:0b:ff:54:79
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=GB, ST=Berkshire, L=Newbury, O=My Company Ltd
    Validity
      Not Before: Oct 21 20:11:48 2007 GMT
      Not After : Nov 20 20:11:48 2007 GMT
    Subject: C=GB, ST=Berkshire, L=Newbury, O=My Company Ltd
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (1024 bit)
        Modulus (1024 bit):
          00:9f:74:fa:f0:bb:8a:c5:21:28:1f:28:03:33:01:
          ff:09:84:ff:2a:48:08:b5:36:a3:59:eb:f2:05:65:
          48:90:bc:65:76:01:20:4d:4e:03:38:80:49:86:9d:
          00:9b:4d:d0:0b:fa:29:6d:2c:bb:70:e1:f0:62:09:
          cb:bc:c9:04:ff:a2:d3:de:30:e1:8c:b6:07:4a:63:
          b4:ba:fd:83:63:60:9d:6c:05:1a:df:f4:1a:31:1a:
          81:e9:8c:6b:27:fa:00:35:2d:2a:21:37:a4:61:bd:
          26:b4:62:28:2f:7d:4d:7d:f5:00:9b:23:61:23:37:
          aa:c2:f8:43:c9:53:21:32:c9
        Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Subject Key Identifier:
        2F:5D:F6:2B:10:75:38:E7:E9:49:EC:7D:8D:23:CE:7D:46:33:5E:10
      X509v3 Authority Key Identifier:
        keyid:2F:5D:F6:2B:10:75:38:E7:E9:49:EC:7D:8D:23:CE:7D:46:33:5E:10
        DirName:/C=GB/ST=Berkshire/L=Newbury/O=My Company Ltd
        serial:BB:C5:91:63:0B:FF:54:79

      X509v3 Basic Constraints:
        CA:TRUE
```

ניתוח התולעת Waledac

www.DigitalWhisper.co.il



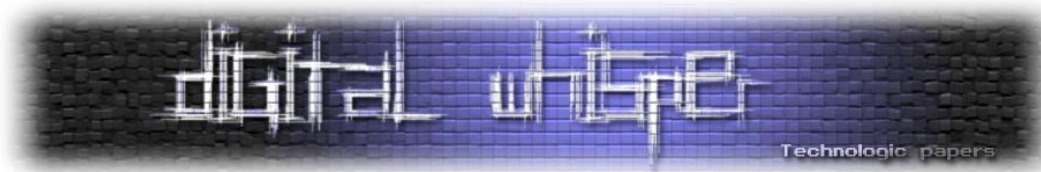
Signature Algorithm: sha1WithRSAEncryption

```
59:8a:61:16:6f:db:8b:91:cf:ee:19:8f:10:6b:7c:8f:42:5f:
c5:cb:d6:f0:fd:56:b7:65:c2:a2:93:bc:1a:2c:12:39:49:d1:
14:20:9a:9b:e3:c8:61:99:ee:4d:24:0c:1c:e7:d0:0a:3a:02:
0f:62:21:fa:31:06:bb:e6:ce:a5:c1:c2:97:2f:c4:ad:de:ec:
c0:7a:39:59:c1:a1:16:aa:72:ca:24:d0:b7:52:63:6d:b0:dd:
29:1a:5b:ce:e6:35:a6:9d:4b:c5:fc:2c:a0:46:9d:52:2f:30:
67:c1:ed:22:b8:39:b6:67:7a:27:52:01:91:78:7d:7b:8c:f4:
ae:f9
```

נשים לב - התעודה כנראה משמשת שימוש פנימי בלבד, אין כל נסיון לשוואת לה אותנטיות - היא חתומה על ידי המשתמש בה (self-signed), התוקף קצר ולא עדכני (שלהי 2007) והיא הוצאה ושימוש על ידי My Company Ltd. יש לתהות בנוגע לתאריך, והאם הוא רומז על כך שהתולעת הייתה בפיתוח בזמן זה. פרט לכך ניתן לראות כאן שימוש במפתח RSA פומבי של 1024 ביט.

את המחרוזות שאמורות לשמש מפתחות AES הצלחנו לחלץ, אבל לא הצלחנו לעשות בהם שימוש. מכון שהצלחנו לחלץ את תוכן ה-RList בשיטה אחרת (על כך בתת פרק ה"פרמטרים להתנהגות") זנחנו את המאמצים בכוון הזה.

בתת פרק ה"הורדות" בפרק ה"תקשורת" נדון ביכולות הורדת תוכנות זדוניות של Waledac. תוכנות אלה מגיעות מצורפות לסופו של קובץ jpeg באופן מוצפן. הצלחנו לאתר את המפתח הרלוונטי בקוד ולהשתמש בו. השיטה והתובנות ממנה יפורטו בתת פרק ה"הורדות", שכן הן כרוכות גם בהבנת תקשורת ה-Waledac.



פרמטרים להתנהגות

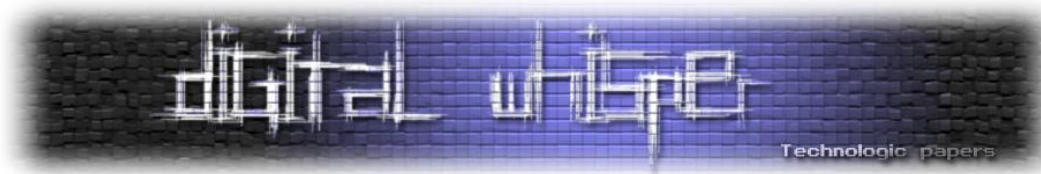
בבדיקת זיכרון הריצה של התוכנה, מצאנו מספר מחרוזות שאפשרו לנו להבין רבות אודות התנהגות התולעת. כאמור ב²², בעת סריקת הדיסק הקשיח, מתעלמת התולעת מקבצים בעלי סיומות מסויימות.

עדות לכך ניתן למצוא בזיכרון התוכנית:

004420B6	PUSH Copy_of_.004D2B2C	ASCII ".avi"
004420C8	PUSH Copy_of_.004D2B34	ASCII ".mov"
004420E0	PUSH Copy_of_.004D2B3C	ASCII ".wmv"
004420F5	PUSH Copy_of_.004D2B44	ASCII ".mp3"
0044210A	PUSH Copy_of_.004D2B4C	ASCII ".wave"
0044211F	PUSH Copy_of_.004D2B54	ASCII ".wav"
00442134	PUSH Copy_of_.004D2B5C	ASCII ".wma"
00442149	PUSH Copy_of_.004D2B64	ASCII ".ogg"
0044215E	PUSH Copy_of_.004D2B6C	ASCII ".vob"
00442173	PUSH Copy_of_.004D21D4	ASCII ".png"
00442188	PUSH Copy_of_.004D2B74	ASCII ".jpg"
0044219D	PUSH Copy_of_.004D2B7C	ASCII ".jpeg"
004421B2	PUSH Copy_of_.004D2B84	ASCII ".gif"
004421C7	PUSH Copy_of_.004D2B8C	ASCII ".bmp"
004421DC	PUSH Copy_of_.004D2B94	ASCII ".exe"
004421F1	PUSH Copy_of_.004D2B9C	ASCII ".dll"
00442206	PUSH Copy_of_.004D2BA4	ASCII ".ocx"
0044221B	PUSH Copy_of_.004D2BAC	ASCII ".class"
00442230	PUSH Copy_of_.004D2BB4	ASCII ".msi"
00442245	PUSH Copy_of_.004D2BBC	ASCII ".zip"
0044225A	PUSH Copy_of_.004D2BC4	ASCII ".7z"
0044226F	PUSH Copy_of_.004D2BC8	ASCII ".rar"
00442284	PUSH Copy_of_.004D2BD0	ASCII ".jar"
00442299	PUSH Copy_of_.004D2BD8	ASCII ".gz"
004422AE	PUSH Copy_of_.004D2BDC	ASCII ".hxx"
004422BF	PUSH Copy_of_.004D2BE4	ASCII ".hxx"
004422D0	PUSH Copy_of_.004D2BEC	ASCII ".hxx"
004422E1	PUSH Copy_of_.004D2BF4	ASCII ".hxd"

נשים לב שהתוכנית מתעלמת מקבצים שלא סביר שיכילו מידע טקסטואלי (שאינו קוד). ניתן למצוא התייחסות סדרתית לסיומות בתוך הקוד, כנראה השוואה למולן במהלך סריקת הדיסק הקשיח:

004420F5	. 68 442B4D00	PUSH Copy_of_.004D2B44	ASCII ".mp3"
004420FA	. 56	PUSH ESI	
004420FB	. E8 1AD60400	CALL Copy_of_.0048F71A	
00442100	. 85C0	TEST EAX,EAX	
00442102	. 59	POP ECX	
00442103	. 59	POP ECX	
00442104	. 0F84 28020000	JE Copy_of_.00442332	
0044210A	. 68 4C2B4D00	PUSH Copy_of_.004D2B4C	ASCII ".wave"
0044210F	. 56	PUSH ESI	
00442110	. E8 05D60400	CALL Copy_of_.0048F71A	
00442115	. 85C0	TEST EAX,EAX	
00442117	. 59	POP ECX	
00442118	. 59	POP ECX	
00442119	. 0F84 13020000	JE Copy_of_.00442332	
0044211F	. 68 542B4D00	PUSH Copy_of_.004D2B54	ASCII ".wav"
00442124	. 56	PUSH ESI	
00442125	. E8 F0D50400	CALL Copy_of_.0048F71A	
0044212A	. 85C0	TEST EAX,EAX	
0044212C	. 59	POP ECX	
0044212D	. 59	POP ECX	
0044212E	. 0F84 FE010000	JE Copy_of_.00442332	
00442134	. 68 5C2B4D00	PUSH Copy_of_.004D2B5C	ASCII ".wma"
00442139	. 56	PUSH ESI	
0044213A	. E8 DBD50400	CALL Copy_of_.0048F71A	
0044213F	. 85C0	TEST EAX,EAX	
00442141	. 59	POP ECX	
00442142	. 59	POP ECX	
00442143	. 0F84 E9010000	JE Copy_of_.00442332	
00442149	. 68 642B4D00	PUSH Copy_of_.004D2B64	ASCII ".ogg"
0044214E	. 56	PUSH ESI	



במאמר NNL-Labs ב²⁸ מפורט תוכן מפתח ה-RList. חיפוש אחר מופעי המחזורת "RList" מאפשר למצוא שימוש בקריאת המערכת RegQueryValueExA על מנת לכתוב למפתח זה:

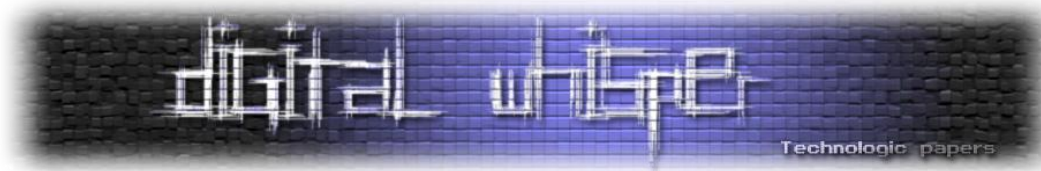
004354A3	. 50	PUSH EAX	pBufSize
004354A4	. 53	PUSH EBX	Buffer => NULL
004354A5	. 8D45 E4	LEA EAX, DWORD PTR SS:[EBP-1C]	
004354A8	. 50	PUSH EAX	pValueType
004354A9	. 53	PUSH EBX	Reserved => NULL
004354AA	. BF EC214D00	MOV EDI, Copy_of_.004D21EC	ASCII "RList"
004354AF	. 57	PUSH EDI	ValueName => "RList"
004354B0	. FF75 EC	PUSH DWORD PTR SS:[EBP-14]	hKey
004354B3	. 895D E8	MOV DWORD PTR SS:[EBP-18], EBX	
004354B6	. C745 E4 030000	MOV DWORD PTR SS:[EBP-1C], 3	
004354B0	. FFD6	CALL ESI	RegQueryValueExA

כעת ניתן למצוא את ערכי ה-RList מפוענחים בזיכרון על ידי קביעת breakpoint בקטע קוד זה, ומעקב אחר המצביעים. כך מצאנו:

00430088	PUSH Copy_of_.004D103C	ASCII "24.119.84.190"
004300C6	PUSH Copy_of_.004D104C	ASCII "96755a2a34252c79e03f5d33cb13be4c"
004300FF	PUSH Copy_of_.004D1070	ASCII "217.23.16.222"
0043013D	PUSH Copy_of_.004D1080	ASCII "485b1f7a7764491cf26d8f341b79ba40"
00430176	PUSH Copy_of_.004D10A4	ASCII "72.129.22.92"
004301B4	PUSH Copy_of_.004D10B4	ASCII "457b8c760d5db91f9b37f148e95f4478"
004301ED	PUSH Copy_of_.004D10D8	ASCII "81.190.159.123"
00430228	PUSH Copy_of_.004D10E8	ASCII "fc3bea2600500e682133a6708c126739"
00430264	PUSH Copy_of_.004D110C	ASCII "70.241.124.121"
004302A2	PUSH Copy_of_.004D111C	ASCII "7d056872a1467c05073102666710a924"
004302D0	PUSH Copy_of_.004D1140	ASCII "81.105.248.214"
00430319	PUSH Copy_of_.004D1150	ASCII "fb024850f111b53cde67f9331437ec46"
00430352	PUSH Copy_of_.004D1174	ASCII "77.124.149.230"
00430390	PUSH Copy_of_.004D1184	ASCII "e87ada36494b5a017760e16c51012932"
004303C9	PUSH Copy_of_.004D11A8	ASCII "61.46.242.69"
00430407	PUSH Copy_of_.004D11B8	ASCII "6d344b0bdd161220652b8e54db770a71"
00430440	PUSH Copy_of_.004D11DC	ASCII "24.98.127.140"
0043047E	PUSH Copy_of_.004D11EC	ASCII "6773a9375f3f3e61536dee5dbb4fe768"
004304B7	PUSH Copy_of_.004D1210	ASCII "12.237.34.248"
004304F5	PUSH Copy_of_.004D1220	ASCII "c42b48369921bf60ee7aac540301f777"
0043052E	PUSH Copy_of_.004D1244	ASCII "88.169.207.221"
0043056C	PUSH Copy_of_.004D1254	ASCII "1e69e11eb32d1c73b30629340278f778"
004305A5	PUSH Copy_of_.004D1278	ASCII "72.190.38.46"
004305E3	PUSH Copy_of_.004D1288	ASCII "cc75290e445811613f6f29717c33fe6b"
0043061C	PUSH Copy_of_.004D12AC	ASCII "98.197.106.184"
0043065A	PUSH Copy_of_.004D12BC	ASCII "a43aa41e160adc60e632ab14e516094c"
00430693	PUSH Copy_of_.004D12E0	ASCII "98.227.164.0"
004306D1	PUSH Copy_of_.004D12F0	ASCII "e10a4a32164c0b06ce5b9342453da04a"
0043070A	PUSH Copy_of_.004D1314	ASCII "72.132.156.122"
00430748	PUSH Copy_of_.004D1324	ASCII "12609a2d685f7f1aa1583b6dde2a5006"
00430781	PUSH Copy_of_.004D1348	ASCII "82.238.116.137"
004307BF	PUSH Copy_of_.004D1358	ASCII "d514677ef009834573131916773d5c1b"
004307F2	PUSH Copy_of_.004D137C	ASCII "88.172.44.197"
0043082D	PUSH Copy_of_.004D138C	ASCII "89172e25b57eaa4a5d7d02018d446e69"
00430866	PUSH Copy_of_.004D13B0	ASCII "66.177.209.68"
004308A4	PUSH Copy_of_.004D13C0	ASCII "ce7f0757e90f8e6ef0274a686d5c5b30"
004308D0	PUSH Copy_of_.004D13E4	ASCII "88.165.250.153"
0043091B	PUSH Copy_of_.004D13F4	ASCII "9d0b95452569b6233577565a8522f615"
00430954	PUSH Copy_of_.004D1418	ASCII "24.1.139.157"
00430992	PUSH Copy_of_.004D1428	ASCII "c85e3765b3271655d54b95124d066d38"
004309CB	PUSH Copy_of_.004D144C	ASCII "81.104.221.69"
00430A09	PUSH Copy_of_.004D145C	ASCII "1e3fea3a5d782e4c3010f619b57ef27f"
00430A42	PUSH Copy_of_.004D1480	ASCII "99.153.5.12"
00430A80	PUSH Copy_of_.004D148C	ASCII "3d1e6478a82161549d6fe206953cad45"
00430AB9	PUSH Copy_of_.004D14B0	ASCII "76.30.215.32"
00430AF7	PUSH Copy_of_.004D14C0	ASCII "5416bc7f2b4c16651f175664b005d12d"
00430B30	PUSH Copy_of_.004D14E4	ASCII "76.108.2.193"
00430B6E	PUSH Copy_of_.004D14F4	ASCII "6854096ad87bc335af43d709cb3e1a44"
00430BA7	PUSH Copy_of_.004D1518	ASCII "68.44.20.169"
00430BE5	PUSH Copy_of_.004D1528	ASCII "b64a5e5a522e5e5c707cce656a342032"

איננו בטוחים בנוגע למשמעות המחזורת העוקבת אחר כל כתובת, אם כי הסברה המקובלת³⁰ היא כי מדובר באיזה מזהה של המחשבים המדוברים. עוד על זהותם בתת הפרק "זהות האחרים" בפרק "תקשורת".

³⁰ W32.Waledac Threat Analysis, Symantec Security Response, pg. 4



בדו"ח של Symantec מדווח³¹ כי מנגנון העטיפה של התולעת מחפש אחר הרצה צעד-אחר-צעד ב-Debugger, ואם מאתר אחד, שולח את התוכנה למבוי סתום. מצאנו עדות לכך:

0048F192	. FF15 84414B00	CALL DWORD PTR DS:[4B4184]	IsDebuggerPresent
0048F198	. 6A 00	PUSH 0	pTopLevelFilter = NULL
0048F19A	. 8BF0	MOV ESI,EAX	SetUnhandledExceptionFilter
0048F19C	. FF15 88414B00	CALL DWORD PTR DS:[4B4188]	
0048F1A2	. 8D45 D0	LEA EAX,DWORD PTR SS:[EBP-30]	pExceptionInfo
0048F1A5	. 50	PUSH EAX	UnhandledExceptionFilter
0048F1A6	. FF15 8C414B00	CALL DWORD PTR DS:[4B418C]	

איתרנו קריאה זו במהלך עיון ב-intermodular calls - היינו, קריאות בין מודולים, וזאת קריאה למודול ה-kernel32 כלומר קריאת מערכת של windows. תפקיד קריאת מערכת זו, כאמור, הוא זיהוי האם התוכנה מורצת בתוך debugger³².

קריאת המערכת הנ"ל בודקת האם מתבצעת הרצה צעד-אחר-צעד של התוכנה. בארכיטקטורת x86 משמעות הדבר האם דולק ה-Trip Flag המאפשר לתוכנה חיצונית (במקרה זה, ה-debugger) לקבל שליטה בחזרה לאחר כל הוראת מכונה.

יש לציין, שטכניקת פתיחת העטיפה שפירטנו בתת פרק "שיטת הביצוע" עוקפת מגבלה זו, שכן אינה מבצעת כל הרצת צעד-אחר-צעד במהלך ריצת פותח העטיפה. עוד מחרוזת חריגה ומפתיעה שמצאנו בקוד:

```
004358F6 | PUSH Copy of .004D2230 | ASCII "http://easyworldnews.com/index.php"
```

ביקור באתר הביא מחרוזת בינארית שלא ידענו לייחס לה משמעות בתחילה. מאוחר יותר, פרסום הדו"ח של Symantec הסביר את משמעותה³³ - לכתובות זו יכולה התולעת לגשת על מנת להשיג רשימת עמיתים חדשה ל-RList. ניסיונונו תומך בסברה זו - לאחר זמן רב שהתולעת לא יכלה להתחבר לאינטרנט, בחנו את התנהגותה. כל הכתובות שפנתה אליהן (הסבר על אופן הפניה והפרוטוקול בפרק "תקשורת") סירבו לפניה, או לא היו קיימות כלל. לאחר מספר ניסיונות כאלה, פנתה התולעת לאתר זה:

391	620.121568	192.168.1.31	192.168.1.1	DNS	Standard query A easyworldnews.com
392	621.133149	192.168.1.31	192.168.1.1	DNS	Standard query A easyworldnews.com
393	622.133292	192.168.1.31	192.168.1.1	DNS	Standard query A easyworldnews.com
394	624.133347	192.168.1.31	192.168.1.1	DNS	Standard query A easyworldnews.com
396	628.133358	192.168.1.31	192.168.1.1	DNS	Standard query A easyworldnews.com

במקרה שלנו, הדבר לא עזר לתולעת - שכן ה-domain כבר לא היה פעיל בשלב זה. מעניין גם לציין כי אין פניות כאלה בלוגים של תקשורת לפני הניתוק לזמן ארוך. כאשר מחפשים אחר פרטי ה-WHOIS של domain זה מוצאים, בדומה ל-domain ממנו נדבקנו, כי נרשם בסין.

³¹ W32.Waledac Threat Analysis, Symantec Security Response, pg. 2

³² Windows Media Developer Center, MSDN,

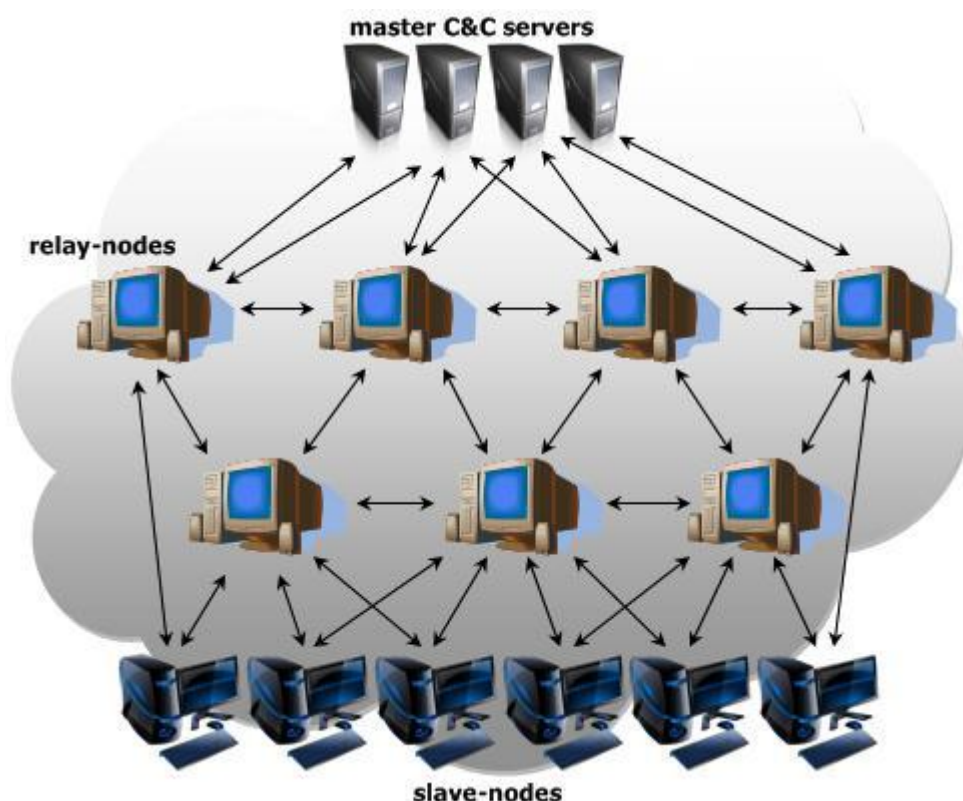
[http://msdn.microsoft.com/en-us/library/ms680345\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms680345(VS.85).aspx)

³³ W32.Waledac Threat Analysis, Symantec Security Response, pg. 5

תקשורת

מבנה פנים הרשת

לפי המדווח בספרות³³ לרשת ה-Waledac מבנה היררכי תלת שכבתי, המתואר באיור להלן, אותו לקחנו מהדו"ח של Symantec המצוטט בהרחבה בעבודה זו:



השכבה העליונה היא שרתי השליטה המרכזיים של הרשת, שתי השכבות מתחת להם הן מחשבי זומבי שהודבקו בתולעת. שכבת ה"עבדים" היא זו שמבצעת את הפעילויות הזדוניות (דואר זבל, הורדות, חיפוש כתובות אימייל), ושכבת ה"ממסרים" היא שמעבירה לה את הפקודות ומקבלת ממנה דיווחים. מבנה רשת זה הוא בחלקו peer-to-peer ובחלקו מרכזי, שכן רוב תקשורת הסוכנים אינה אל שרתי השליטה ובקרה המרכזיים, אבל בכל זאת ישנם כאלה. בהמשך נזהה כמה תפקודי שרתים מרכזיים אחרים נוספים. בחירת התפקיד לזומבי חדש מתבססת על בדיקות שנעשות עם ההדבקה - הזומבים שיש להם רוחב פס גדול יותר וניתנים לגישה מרחוק הם אלה שיבחרו להיות ממסרים, האחרים יהיו עבדים³⁰.

האתר sudosecure מפעיל מנגנון למעקב אחרי רשת ה-Waledac וההדבקות בה:

<http://www.sudosecure.net/waledac>

התקשורת בין הזומבים מתבצעת על בסיס פרוטוקול שמוכנה מעל HTTP³⁴, מה שגורם לה לא להראות חשודה בקרב תעבורת הרשת. נפרט יותר על פרוטוקול זה בתת הפרק הבא. ניתן לראות כי ההדבקה שלנו היא זומבי מסוג עבד. ראשית, לפי הדרך בה בנינו את המעבדה - אין גישה מבחוץ. פרט לכך, כל התקשורת של התולעת שלנו יזומה על ידיה, ופונה לעמיתים אחרים. לו הייתה התולעת שלנו זומבי ממסר - היינו מצפים גם לפניות אליה. בתת הפרק הבא נציג דו-שיח לדוגמא בין הזומבי שלנו לזומבי ממסר.

יש לציין כי מצאנו חלק מזומבי הממסר איתן תקשורה התולעת שלנו במנגנון המעקב של sudosecure, אבל זו שלנו לא הופיעה שם, גם לאחר זמן-מה של פעילות. מכך אולי יש להבין שהמנגנון עוקב בעיקר אחר ממסרים, אבל אין הסבר על הנושא באתר, שנוטה להיות די לקוני בכל הנוגע לשיטותיו. נוסף לתשתית תקשורת השליטה המרכזית, זיהינו שני תפקודי שרתים נוספים:

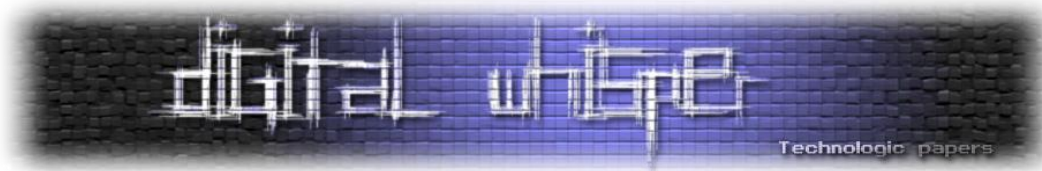
- **שרתי "גיבוי" לרשימת עמיתים** - אלה הם השרתים מסוגם של השרתים הפועלים תחת ה-domain: easyworldnews.com. כפי שפירטנו בתת הפרק "פרמטרים להתנהגות". שרתים אלה, כאמור, מאפשרים לזומבי עבד לקבל רשימת עמיתים מעודכנת כדי לחבור לרשת.
- **שרתי הפצת תוכנה** - אלה הם השרתים המפיצים תוכנות נוספות לזומבים העבדים. דוגמא לכך הם השרתים הפועלים תחת ה-domain: usabreakingnews.com שעל תפקודם נפרט בתת הפרק על "הורדות". בתמצית, מטרתם היא לאפשר הפצת תוכנה נוספת ברשת.

יש לציין מספר פרטים בנוגע לתפקודי שרתים אלה - ראשית, נשים לב לבחירת ה-domains בהתאם לקמפיין ה-social engineering שבו נדבקו - גם הם "אתרי חדשות". שני domains אלה נרשמו בסין, ושניהם סיפקו את אותם שרתי DNS - NS1.LOIDEVE.COM (כאשר הספרה 1 מוחלפת בכל הספרות מ-1 ועד 6).

בעת כתיבת שורות אלו, domain-ים אלה נסגרו והם אינם פעילים יותר. נקבע להם סטטוס של: clientDeleteProhibited/clientTransferProhibited - כלומר אין אפשרות למחוק את ה-domain או להעביר עליו בעלות, ולמעשה "נסגר". איננו יודעים, ולא מצאנו דיווח בספרות, מי הם המחשבים המשמשים בתפקידים אלה, והאם הם מחשבי הממסר או מחשבים אחרים. ניתן למצוא מאגר של domain-ים הקשורים בפעילות waledac, המחולקים לפי קמפיינים, בכתובת:

http://www.shadowserver.org/wiki/uploads/Calendar/waledac_domains.txt

³⁴ Speaking Waledac, The Honeynet Project,
<http://www.honeynet.org/node/348>



פרוטוקול

פרוטוקול ה-Waledac פוענח לחלוטין למעשה בספרות³⁵. המבנה הכללי הוא של בקשות HTTP - זומבי העבד פונה בבקשת POST לזומבי הממסר, ומקבל ממנו מידע בהודעות ה-OK. כמפורט ב-³⁴ כל שיחה מתחילה בהחלפת תעודות פומביות, ועל סמכן החלפת AES Session Keys. לאחר מכן, ישנו פרוטוקול (מוצפן) המבוסס xml המעביר הוראות מהממסרים, ודווחים מהעבדים. דוגמא לשיחה בין העבד שלנו לממסר (ה payload הבינארי מקוצץ):

```
POST /lsxq.png HTTP/1.1
Referer: Mozilla
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla
Host: 60.244.196.128
Content-Length: 210
Cache-Control: no-cache
a=BAAAAIay...

HTTP/1.1 200 OK
Server: nginx/0.6.34
Date: Thu, 30 Apr 2009 14:49:06 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.2.8
f6dBAAEQY...
```

כאמור, ניתן כאן לראות שמבוצע POST לקובץ png, הקלטנו בקשות דומות גם עם קבצי .htm. נשים לב למספר מאפיינים חריגים נוספים ששימשו אותנו בהמשך לזיהוי התולעת - שדה ה-Referer מכיל Mozilla בעוד הוא מיועד להכיל את ה-URL ממנו הגיע הגולש³⁶, וה-payload מתחיל ב-"a=" (לא נוכח בכל ההודעות). כפי שהזכרנו בעבר, העובדה שהוקלטה רק תקשורת בתבנית כזאת בין התולעת המקומית שלנו לאחרות מביאה אותנו למסקנה שהתולעת אצלנו תפקדה בתפקיד "עבד".

לצערנו לא הצלחנו לפענח את התקשורת כפי שהצליחו אחרים, אבל כן ניתן לצפות בתעבורה על ידי חקירת המחרוזות שבזיכרון העבודה של התוכנית. כך ראינו שהתולעת אכן מתחילה את ההתקשרות בשליחת התעודה, כאשר ראינו את ההודעה הבאה:

```
004EE998 . 307CAE00 DD 0>; ASCII
"<lm><t>getkey</t><v>34</v><i>ac690c4e9536b11a2339a851fa6f1053</i><r>1</r><props><p n="cert">-----BEGIN CERTIFICATE-----
MIIBVjCCASegAwIBAgIBADANBgkqhkiG9w0BAQQFADAlMQswCQYDVQQGEwJVSzEW
MBQGA1UEAxMNT3B1b1NTTCBHcm91cDAeFw0wOTA5MTAyMzU5MTRa"...
```

³⁵ Peer-to-peer botnets: A case study on Waledac, Lasse Trolle Borup, pg. 28-38

³⁶ Request Headers in the HTTP protocol, W3,
<http://www.w3.org/Protocols/HTTP/HTTRQ-Headers.html#z14>

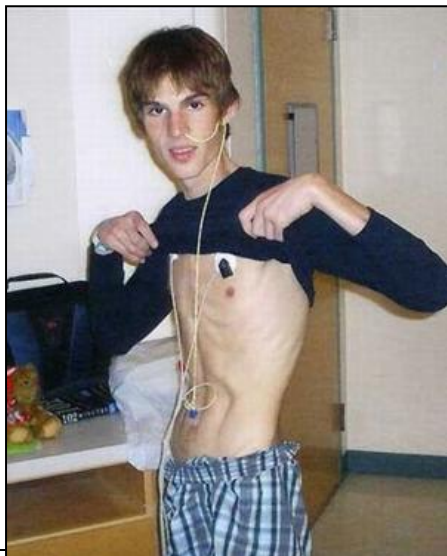
רואים כי נשלחת הודעה מסוג getkey, שמספר הגרסא הוא 34 ושמצורפת התעודה בסוף ההודעה. בשדה i מופיע מזהה המכונה - השווינו אותו לזה הנמצא ב-Registry בערך Myld והם זהים. ניתן למצוא פירוט מלא ונוח של פקודות הפרוטוקול בדו"ח של טרנדמיקר³⁷ שם מוסברים סוגי הפקודות השונים ומטרותיהם. לצערנו, האינדיקציה היחידה שיש לנו למבנה הפקודות היא מניתוח הקוד:

0042955C	• 83F8 08	CMP EAX,8	Switch (cases 0..8)
0042955F	• 77 46	JA SHORT Copy_of_.004295A7	ASCII "getkey"; Case 0 of switch 0042955C
00429561	• FF2485 B79542	JMP DWORD PTR DS:[EAX*4+4295B71]	ASCII "first"; Case 1 of switch 0042955C
00429568	> 68 28FD4C00	PUSH Copy_of_.004CFD28	ASCII "notify"; Case 2 of switch 0042955C
0042956B	• EB 3D	JMP SHORT Copy_of_.004295AC	ASCII "taskreq"; Case 3 of switch 0042955C
0042956F	> 68 30FD4C00	PUSH Copy_of_.004CFD30	ASCII "words"; Case 4 of switch 0042955C
00429574	• EB 36	JMP SHORT Copy_of_.004295AC	ASCII "taskrep"; Case 5 of switch 0042955C
00429576	> 68 38FD4C00	PUSH Copy_of_.004CFD38	ASCII "httpstats"; Case 6 of switch 0042955C
0042957B	• EB 2F	JMP SHORT Copy_of_.004295AC	ASCII "emails"; Case 7 of switch 0042955C
0042957D	> 68 40FD4C00	PUSH Copy_of_.004CFD40	ASCII "creds"; Case 8 of switch 0042955C
00429582	• EB 28	JMP SHORT Copy_of_.004295AC	ASCII "unknown command"; Default case of switch 0042955C
00429584	> 68 48FD4C00	PUSH Copy_of_.004CFD48	
00429589	• EB 21	JMP SHORT Copy_of_.004295AC	
0042958B	> 68 50FD4C00	PUSH Copy_of_.004CFD50	
00429590	• EB 1A	JMP SHORT Copy_of_.004295AC	
00429592	> 68 58FD4C00	PUSH Copy_of_.004CFD58	
00429597	• EB 13	JMP SHORT Copy_of_.004295AC	
00429599	> 68 64FD4C00	PUSH Copy_of_.004CFD64	
0042959E	• EB 0C	JMP SHORT Copy_of_.004295AC	
004295A0	> 68 6CFD4C00	PUSH Copy_of_.004CFD6C	
004295A5	• EB 05	JMP SHORT Copy_of_.004295AC	
004295A7	> 68 74FD4C00	PUSH Copy_of_.004CFD74	
004295AC	> 8BC6	MOV ECX,ESI	
004295AE	• E8 A87FFDFF	CALL Copy_of_.0040155B	
004295B3	• 8BC6	MOV EAX,ESI	
004295B5	• 59	POP ECX	
004295B6	• C3	RETN	
004295B7	• 68954200	DD Copy_of_.00429568	Switch table used at 00429561

כאן ניתן למצוא מבנה של switch-case שניתן לשער שמופיע בחלק הקוד המקבל הודעה ומפענח אותה. ניסינו לעקוב אחר הפרוצדורות השונות, אבל בהעדר מחרוזות משמעותיות לא הצלחנו להבין את התנהגותן.

מנגנון ההורדות (downloader)

בספרות מתוארות³⁸ יכולות הורדת עדכוני תוכנה של Waledac. התולעת מנצלת יכולות אלה על מנת להפיץ עדכונים לקובץ ההרצה, וכן להפצת winpcap ותרמיות אנטי-וירוס. הפרוטוקול מבוסס על פקודה מתאימה שנשלחת למחשב העבד ממחשב ממסר, שגורמת לו לגשת ולהוריד קובץ JPEG מאתר מרוחק.



קובץ זה יפתח כקובץ תמונה תקין, אבל בסופו נוסף עדכון התוכנה מוצפן על ידי XOR עם מפתח הנמצא בקוד. ביצענו חיפוש ברישומי התקשורת ב-wireshark לפי בקשות HTTP GET על מנת לזהות חריגות מהפרוטוקול. זה הביא אותנו להורדת usabreakingnews.com/win.jpeg שכתמונה נראה כך (אל חשש, מתמונה זו כבר הוסר התוכן ה-Waledac-י):

³⁷ Infiltrating WALEDAC Botnet's Covert Operations, TrendMicro, pg. 18-25

³⁸ W32.Waledac Threat Analysis, Symantec Security Response, pg. 12-14

בדקנו diff בין קבצי ההרצה של waledac מתאריך שקודם ששמרנו לחוד, ולא נמצא כל הבדל. זה הוביל אותנו לתהות האם אכן התרחש עדכון, מה שנתמך על ידי דיווח של טרנדמיקרו³⁹ כי חלק מהקבצים אינם מכילים כל עדכון. חשדנו שב ועלה כאשר שמנו לב כי התמונה, שנראית באיכות נמוכה יחסית, היא בנפח 243KB שהערכנו כגדול מדי. ניסוי ראשון היה לקצץ חלק מסוף הקובץ, וכך ראינו שאכן ניתן להוריד חלק ניכר מהקובץ והתמונה תשאר תקינה. בדו"ח של סינמטק מתוארת מחרוזת המפרידה בין התמונה לעדכון התוכנה⁴⁰. חיפשנו אחר המחרוזת בקובץ ההרצה, וסמוך לשם מצאנו את פרוצדורת הפענוח שלו:

00440032	> 8B45 E8	MOV EAX,DWORD PTR SS:[EBP-18]
00440035	. 800488	LEA EAX,DWORD PTR DS:[EAX+ECX*4]
00440038	. 8130 EDEDED	XOR DWORD PTR DS:[EAX],EDEDED
0044003E	. 41	INC ECX
0044003F	. 3BCE	CMP ECX,ESI
00440041	. ^72 EF	JB SHORT Copy_of_.00440032

ניתן לראות כי הפענוח מבוצע על ידי XOR עם המחרוזת "ED". ביצענו זאת בעצמנו וקיבלנו קובץ התקנה של winpcap שנוצר על ידי nullsoft scriptable install system - כלי יצירת התקנות מבוסס קוד פתוח. כלי זה (winpcap) הוא ספריה המאפשרת יכולות sniffing לתקשורת המחשב. לפי הספרות Waledac משתמשת בה לצורך גניבת סיסמאות, ומידע של המשתמשים דרך התקשורת³⁸ ובנוסף להסתרה של התקשורת הזדונית מפני Wireshark⁴¹.

מעניין לציין שלמרות שהורדה ספריה, ולא יכולת חיצונית, לא היה שינוי בקובץ ההרצה של התוכנית. לאור כך, חיפשנו ומצאנו התייחסות לכך בקוד:

00453791	. 68 5C364D00	PUSH Copy_of_.004D365C	FileName = "wpcap.dll"
0045379c	. 32DB	XOR BL,BL	LoadLibraryA
0045379e	. FF15 10424B00	CALL DWORD PTR DS:[4B4210]	
0045379f	. 95C0	TEST EAX,EAX	
004537a0	. 8906	MOV DWORD PTR DS:[ESI],EAX	
004537a2	~0F84 B7000000	JE Copy_of_.0045385F	
004537a8	. 57	PUSH EDI	
004537a9	. 8B3D F8404B00	MOV EDI,DWORD PTR DS:[4B40F8]	kernel32.GetProcAddress
004537af	. 68 68364D00	PUSH Copy_of_.004D3668	ProcNameOrOrdinal = "pcap_findalldevs"
004537b4	. 50	PUSH EAX	hModule
004537b5	. FFD7	CALL EDI	GetProcAddress
004537b7	. 8946 04	MOV DWORD PTR DS:[ESI+4],EAX	
004537ba	. 8B06	MOV EAX,DWORD PTR DS:[ESI]	
004537bc	. 68 7C364D00	PUSH Copy_of_.004D367C	ProcNameOrOrdinal = "pcap_freealldevs"
004537c1	. 50	PUSH EAX	hModule
004537c2	. FFD7	CALL EDI	GetProcAddress
004537c4	. 8B0E	MOV ECX,DWORD PTR DS:[ESI]	
004537c6	. 68 90364D00	PUSH Copy_of_.004D3690	ProcNameOrOrdinal = "pcap_open"
004537cb	. 51	PUSH ECX	hModule
004537cc	. 8946 08	MOV DWORD PTR DS:[ESI+8],EAX	GetProcAddress
004537cf	. FFD7	CALL EDI	
004537d1	. 8B16	MOV EDX,DWORD PTR DS:[ESI]	
004537d3	. 68 9C364D00	PUSH Copy_of_.004D369C	ProcNameOrOrdinal = "pcap_loop"
004537d8	. 52	PUSH EDX	hModule
004537d9	. 8946 0C	MOV DWORD PTR DS:[ESI+C],EAX	GetProcAddress
004537dc	. FFD7	CALL EDI	
004537de	. 8946 10	MOV DWORD PTR DS:[ESI+10],EAX	
004537e1	. 8B06	MOV EAX,DWORD PTR DS:[ESI]	
004537e3	. 68 A8364D00	PUSH Copy_of_.004D36A8	ProcNameOrOrdinal = "pcap_compile"
004537e8	. 50	PUSH EAX	hModule
004537e9	. FFD7	CALL EDI	GetProcAddress

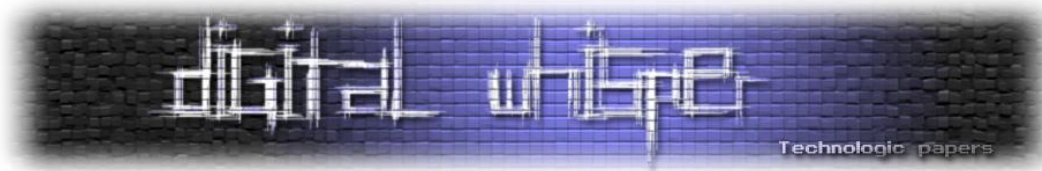
כפי שניתן לראות, כבר קיימת בקוד התשתית לשימוש בספריה, וכנראה יש דגל כזה או אחר שמציין את האפשרות להשתמש בה. שיטה זו תמוהה מעט - אם מלכתחילה יש שימוש בספריה בקוד, למה לא לצרף

³⁹ Infiltrating WALEDAC Botnet's Covert Operations, TrendMicro, pg. 20

⁴⁰ W32.Waledac Threat Analysis, Symantec Security Response, pg. 13

⁴¹ Hello Waledac, My Old Friend, Cisco Blog,

http://blogs.cisco.com/security/comments/hello_waledac_my_old_friend/



אותה לקובץ המופץ? ומצד שני, אם כבר טורחים להפיץ עדכון עם תוכנה נוספת, למה לא להוסיף בו את שינויי הקוד? ניתן לתהות גם האם יש יכולות מודולריות נוספות של Waledac המסתתרות בקוד. נוסף על כך, בולטת ההצפנה החלשה של קובץ זה, בהשוואה להצפנה החזקה של התקשורת וקובץ ההרצה. לבסוף, יש לציין כי לא מצאנו את כתובת העדכון בקובץ ההרצה של Waledac, ולכן סביר להניח שהועברה לתולעת בפקודה להוריד את העדכון.

שליחת דואר זבל

השליחה

בספרות מתואר ש-Waledac מפיצה קמפיינים של ספאם ושל הפצת התולעת⁴². במהלך אחד מהניסויים המקוונים שלנו, תוך כדי ניטור התקשורת ע"י Wireshark, נתקלנו בהפצה של ספאם מהמכונה שלנו, לאחר זמן לא רב מהרגע שראינו זאת החלטנו לנתק את המכונה מהרשת ולעבור לניסויים בלתי-מקוונים על מנת למזער נזקים לאחרים ומכיוון שחששנו כיצד יגיב לכך ה-ISP שלנו. להלן דוגמא לשליחת SPAM:

```
220 mx6.dhs.gov ESMTP Postfix
HELO bzq-84-110-247-186.red.bezeqint.net
250 mx6.dhs.gov
MAIL FROM:<>
250 2.1.0 Ok
RCPT TO:<cklin@dhs.gov>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Received: from bqz ([159.177.134.104])
.by bzq-84-110-247-186.red.bezeqint.net (8.13.1/8.13.1) with SMTP id
200904301750055542;
.Thu, 30 Apr 2009 17:50:49 -0800
Message-ID: <000501c9c9f6$be002c70$9fb18668@johnnybqz>
From: "Sadie Ortiz" <cm.846dt.h415ld4.r@fisco.it>
To: <cklin@dhs.gov>
Subject: Successful formula, for men, successful in love.
Date: Thu, 30 Apr 2009 17:49:54 -0800
MIME-Version: 1.0
Content-Type: text/plain;
.format=flowed;
```

⁴² Infiltrating WALEDAC Botnet's Covert Operations, TrendMicro, pg. 29

```
.charset="iso-8859-1";
.reply-type=original
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 6.00.2800.1158
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2800.1158

Harder banging is real! http://ym.ugcacopce.com/

.
250 2.0.0 Ok: queued as E7AD82F788E3
QUIT
221 2.0.0 Bye
QUIT
```

נא לשים לב לכך שההודעה נשלחה למשתמש ב-Department of Homeland Security, שמאכזבים בכך שלא הפעילו אמצעים לסנן הודעות מסוג זה. יש לציין שחלק ניכר מהודעות ה-SPAM שניסטה לשלוח התולעת נחסמו על ידי שרתי ה-smtp אליהם התחברה על ידי כלים אוטומטיים.

מאפיינים חריגים שהתגלו מניתוח של הספאם הוא ששדה ה-"MAIL FROM" נותר ריק, לאחר קריאה בתקן ה-smtp⁴³ גילינו ששדה זה אמור להישאר ריק אך ורק כשמעוניינים שלא תהיה כתובת למשלוח חזרה - דבר שמשמש שרתי smtp למניעת לולאות אינסופיות של הודעות על שגיאות בהפצת דוא"ל, ובהחלט דבר חריג בדוא"ל רגיל.

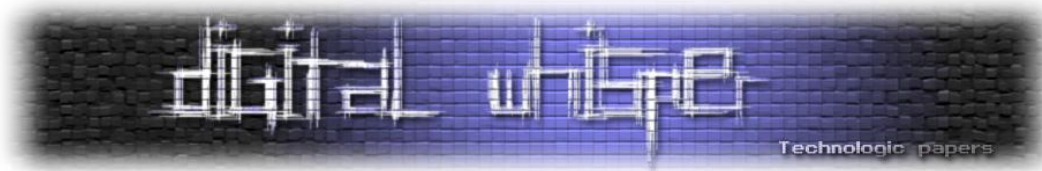
תוכן

מבין מעט ההודעות ששלחנו, נמצא מגוון גדול של כותרות וניסוחים, זאת בהתאמה לדיווחים בספרות על מנגנונים המיועדים ליצירת גיוון בהודעות הנשלחות⁴⁴. עם זאת, כל הקישורים שעקבנו אחריהם הובילו לאתרים בעלי אופי דומה - כולם מציעים תרופות בסגנון ויאגרה וציאליס ומזהים עם תרמית ה-Canadian Pharmacy. לפי SPAMhaus Project, זאת אחת מתרמיות דואר הזבל הפעילות בעולם⁴⁵, ולפי בדיקתנו ברשימתם ב-16 בספטמבר 2009 - הפעילה ביותר בעולם. כאמור, ניתקנו את העמדה לאחר הבחנה

⁴³ RFC821 - Simple Mail Transfer Protocol, <http://www.faqs.org/rfcs/rfc821.html>

⁴⁴ Infiltrating WALEDAC Botnet's Covert Operations, TrendMicro, pg. 27

⁴⁵ The 10 Worst ROKSO Spammers, <http://www.spamhaus.org/statistics/spammers.lasso>



בפעילות ה-SPAM, וכך אין לנו יכולת להסיק מסקנות משמעותיות על תפוקת ה-SPAM של העמדה - לאחר 16 דקות פעילות, היו שתיים וחצי דקות של שליחת SPAM עד שניתקנו, ובזמן זה נשלחו 194 הודעות דואר זבל. מכון שאיננו יודעים כמה זמן הייתה נמשכת שליחת ה-SPAM הזו, אי אפשר להסיק מכך דבר.

SNORT - שימוש בחתימות תקשורת לזיהוי התולעת

חתימות

על מנת לזהות פעילות Waledac ברשת, יצרנו חתימות משני סוגים - אחת המבוססת על זיהוי מאפייני הפרוטוקול הפנים רשתי, והשניה מבוססת על זיהוי מאפייני שליחת ה-SPAM.

החתימה מבוססת הפרוטוקול הפנימי:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 80 (msg: "Waledac messages detected: POSTing png files."; flow:to_server,established; content:"POST"; content:"Referer\ : Mozilla"; pcre:"/\.(htm|png)/"; content:"a="; sid:6666;)
```

תחילת החתימה מכתובה כי התקשורת אליה היא מתייחס היא תקשורת יוצאת בפורט 80, בהתאמה לתקשורת באמצעות HTTP. לאחר מכן, אנחנו מגבילים את התקשורת לכזאת שמכונן אל השרת, בקשר שכבר הוקם. יתר המאפיינים מטרם לסנן את התוכן לפי המאפיינים שמצאנו - בקשות POST, שדה ה-Referer החריג, סיומות הקבצים, ופתיחת הקובץ ב-"a=".

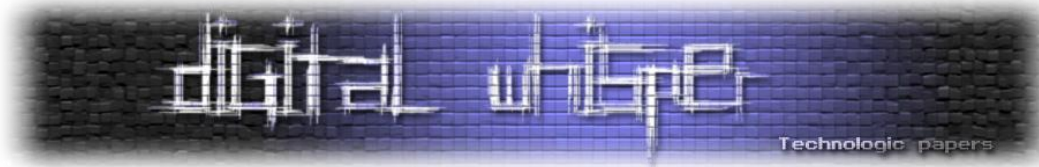
החתימה מבוססת שליחת SPAM:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 25 (msg: "Waledac messages detected: sending SPAM."; flow:to_server,established; content:"MAIL FROM:<>"; sid:6666;)
```

כאן אנחנו מזוהים שוב תקשורת יוצאת, בכונן השרת, בקשר שכבר הוקם. אנחנו מסננים את התקשורת לפי פורט, כדי לוודא שזאת תקשורת SMTP. לב החתימה הוא בזיהוי ששדה ה-MAIL FROM ריק. יש לציין שחתימה זו מוגבלת יותר, שכן התנהגות זו, כאמור, לגיטימית בתקשורת בין שרתי SMTP. יש מספר דוגמאות של חתימות Snort קיימות המזהות את פרוטוקול הרשת של Waledac: בתזת המאסטר שלו, מציע Lasse Trolle Borup חתימה⁴⁶ המבוססת על זיהוי המחרוזת:

```
X-Request-Kind-Code: nodes
```

⁴⁶ Peer-to-peer botnets: A case study on Waledac, Lasse Trolle Borup, pg. 52, Fig 5.27



מחרוזת זו מאפיינת, לפי טענתו⁴⁷, הודעות בהן מופיעה מחרוזת זו בכותרת הן בקשה של הלקוח לקבלת רשימת עמיתים חדשה. גם אנחנו מצאנו הודעות מסוג זה, שיש לציין שהקובץ המצורף להן לא מתחיל ב-"a=". אין מידע בעבודתו של Borup בנוגע לאפשרות להיווצרות מחרוזת זו בתקשורת חוקית. דוגמא נוספת מסופקת⁴⁸ על ידי Shadowserver Foundation ודומה למדי לשלנו:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"ET TROJAN
Waledac Beacon Traffic Detected"; flow:to_server,established;
content:"POST /"; depth:6; content:"|0d 0a|Referer\: Mozilla|0d 0a|";
nocase; within:50; content:"|0d 0a|User-Agent\: Mozilla|0d 0a|";
within:120; content:"a="; nocase; within: 100; classtype:trojan-
activity;reference:url,www.shadowserver.org/wiki/pmwiki.php?n=Calendar.2
0081231; sid:2008958; rev:1;)
```

תוספת חשובה של חתימה זו ביחס לשלנו הם שדות ה-depth ו-the within המגבילים את טווח החיפוש וישפרו את ביצועי Snort בעת הפעלת החתימה. חתימה נוספת שנוצרה⁴⁹ היא זו:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"2009 Holiday
Greeting Spam - Unusual Referer String (Mozilla)";
flow:to_server,established; content:"Referer\: Mozilla"; nocase;
classtype:trojan-activity; sid:999999;)
```

חתימה זו מבוססת אך ורק על זיהוי שדה ה-Referer החרגי.

ניסויים

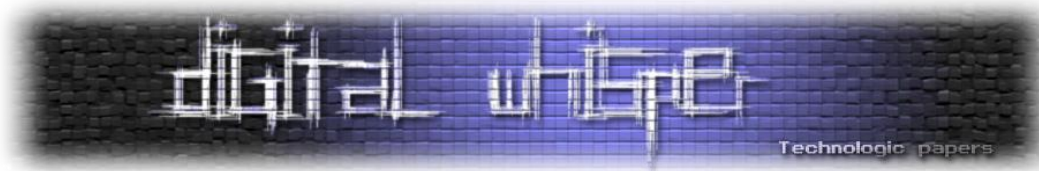
ביצענו מספר ניסויים בלתי-מקוונים על שתי החתימות שלנו ע"י הזנת התקשורת שהקלטנו כאשר התולעת הייתה מחוברת לרשת. החתימה הראשונה התריעה על Waledac בכל הלוגים ששמרנו. החתימה השנייה התריעה רק בלוגים שבהם התולעת שלנו שלחה ספאם.

בנוסף הרצנו את החתימות הללו על תקשורת רגילה על מנת לבדוק האם החתימות יניבו התרעות מסוג "false positive", ולשמחתנו הן לא. נוסף על כך, בחינת תקשורת HTTP תקינה הראתה ש-Mozilla לא הופיע בשדה ה-Referer. יש לציין שמכיוון שעברנו לעבודה בלתי-מקוונת ואיבדנו קשר עם רשת הבוט, לא יכלנו לבדוק את החתימות הללו על תקשורת "חיה".

⁴⁷ Peer-to-peer botnets: A case study on Waledac, Lasse Trolle Borup, pg. 32, Fig 5.5

⁴⁸ Waledac is Storm is Waledac? Peer-to-Peer over HTTP.. HTTP2p?, Shadowserver Foundation, <http://www.shadowserver.org/wiki/pmwiki.php/Calendar/20081231>

⁴⁹ Fast Flux Greeting Card Spam, DISOG, <http://www.disog.org/2008/12/fast-flux-greeting-card-spam.html>



סיכום

רשמים כלליים ואישיים

במהלך העבודה על הפרויקט, למדנו כיצד ניתן לתפוס ב"טבע" קובץ זדוני, ולנתח אותו בסביבת מעבדה בטוחה. לשם כך היה עלינו ללמוד ולפתח טכניקות מתאימות לניטור הסביבה, פענוח קובץ ההרצה, והבנת התקשורת. לצורך כך נעזרנו גם בפרוייקטים קודמים שנעשו, ותצוין במיוחד עבודתו של ברק נירנברג על תולעת ה-Storm⁵⁰, שהוותה את הבסיס לתכנון המעבדה שלנו.

חלק מרכזי נוסף בעבודתנו היה מציאת והטמעת הידע הקיים על התולעת שמצאנו. כשהתחלנו בעבודה, מקורות המידע המרכזיים שלנו היו מספר בלוגים:

- SudoSecure - www.sudosecure.net - מנסה לבצע מעקב אחר התפשטות והתנהגות התולעת, וכתוצאה מכך מספק מידע רב על תפוצת התולעת ופעילותה. נוסף לכך, מתחזק בלוג ובו יש מידע על רוב קמפייני ה-Waledac ומבחר סטטיסטיקות מהמעקב אחר התולעת. הבלוג נמנע לגמרי מתיאור השיטות בהן הוא משתמש או העקרונות בבסיס מנגנון העקיבה שלו.
- HoneyNet Project - www.honeynet.org - אתר זה מרכז מספר מאמצי מחקר בתחומי אבטחת המידע. יש בו מידע רב על הידבקות ב-botnets באופן כללי. יש באתר גם דווחים בנוגע לתולעת ה-Waledac. בין השאר יש שם הסבר כללי על פרוטוקול התקשורת ותחילת פענוחו - מבלי להסביר את השיטה, וכן מידע ראשוני מתוך הבינארי - התעודה הדיגיטאלית, מבלי להסביר איך מפענחים אותו (פרט להערה המעודדת "it's fairly easy").
- NNL Labs - www.nnl-labs.com - בעת תחילת העבודה, שם נמצא ההסבר המפורט ביותר על פרוטוקול התקשורת של Waledac (והפעם כתוב במפורש שאין לכותב הבלוג כוונה לחלוק את הטכניקות בהן השתמש לצורך פענוחה). יש בו גם דוגמא לערך RList מפוענח, שמול המבנה שלו השווינו את התוכן המפוענח שהצלחנו למצוא בזיכרון הריצה לראשונה, שוב, בלי הסברים איך לעשות זאת (פרט להצהרה כי ניתן למצוא את המפתחות הרלוונטיים בקוד).
- Shadowserver Foundation - www.shadowserver.org - עוד בלוג שמדווח על Waledac בין השאר. מכיל בעיקר מידע על הקמפיינים השונים, ופעילות זדונית חריגה של התולעת (הורדת rogue antispyware, ביצוע joe-jobbing).

⁵⁰ Storm Bot-Net, Barak Nirenberg,

<http://webcourse.cs.technion.ac.il/236349/Winter2009-2010/ho/WCFiles/project5-final-report.pdf>

על אף השימושיות הרבה שבמידע בבלוגים אלה בתחילת עבודתנו, קשה שלא להתרגז מהנחישות וההקפדה של כותביהם על "שמירת הקלפים קרוב לחזה" בכל הנוגע לשיטות העבודה. לטעמנו, חלק גדול מהתרומה האפשרית של דו"ח זה הוא בהדגשתנו את השיטות להשגת המידע.

על מנת לעקוב אחר התפתחויות בנוגע לתולעת, במיוחד כשהיא "חמה" ועדכנית, השתמשנו בשירות Google Alerts. במהלך עבודתנו (יוני) פורסם דו"ח של חברת TrendMicro בנוגע ל-Waledac, שלפני כן רק דווחה בבלוג שלה על קמפייני ההדבקה השונים. מעניין לציין שתחילת פעילות התולעת בדצמבר (ותחילת עבודתנו באפריל) - כך שעברה חצי שנה מתחילת הפעילות הידועה של התולעת עד פרסום הדו"ח המרכזי הראשון. בדו"ח זה כבר ניתן למצוא מידע רב על שיטות הפעולה של התולעת, וגם, לראשונה, כיצד למצוא את מפתחות ה-AES המשמשים אותה להצפנת ה-Registry. דו"ח זה מפורט מאוד מבחינת דוגמאות, וכן מספר מראי מקום על מציאת מידע באמצעות disassembly (אם כי גם בו אין הסבר כיצד לבצעו), וקידם אותנו מאוד בהבנת הממצאים שלנו ומיקומם ב"תמונה הגדולה".

לאחרונה (אוגוסט) פורסם דו"ח מקיף מאוד של חברת Symantec על Waledac. דו"ח זה מפורט יותר מזה של TrendMicro והשלים בו פרטים חסרים, כמו הורדת העדכונים. נוסף על כך, דו"ח זה הוא הידיוטי ביותר לשימוש מכלול. לצערנו, כשפורסם כבר היינו לקראת סוף העבודה, ולכן השימוש המרכזי בו היה להשלמת חורים בהבנתנו חלק מהממצאים.

ישנו מקור מידע נוסף, שלא שימש אותנו כל כך משום שלא מצאנו בו מידע שלא נמצא במקורות אחרים, וזה תזת המאסטר של Lasse Trolle Borup מהאוניברסיטה הטכנית של דנמרק. גם שם יש מדיניות כללית של הסתרת הטכניקות, וביטויים לקוניים כגון "יש להשתמש באחד משני מפתחות הנמצאים בקוד". ישנה חשיבות גדולה לקהילה (שחלקה תעשייתית וחלקה שלא למטרות רווח) החוקרת פעילויות מסוג זו, אך לדעתנו שקיפות רבה יותר בנוגע לטכניקות תאפשר התקדמות מהירה יותר של הידע. ממילא, יוצרי התולעת ידעו שפוצחה אם מתפרסמים עליה מחקרים מקיפים, בין אם יפרטו את הטכניקות ובין אם לא.

רכיב מרכזי בעבודה היה הגילוי. העבודה תוך כדי שאין לנו תמיד כוון מוגדר לחיפוש, וכשהליכה בכוון מוגדר מביאה פעמים רבות לממצאים אחרים. ביצענו תהליך חוזר של ניסוי וטעיה, נסיון לעקוב אחר הידע הקיים ומציאת פרטים אחרים (או מילוי פרטים חסרים), והערכה והבנה מחודשת של ידע קודם שהיה לנו. בנימה אישית, אנו מרגישים שהאספקט המרכזי בו אנחנו נרתמנו מהעבודה הוא זה - התקדמות בנתיב לא מוגדר שהידע בו לא שלם, ודורש מאתנו פיתוח תמונת מצב וטכניקות עצמאיים, תוך כדי אינטגרציה של מעט הידע הקיים.

על חשיבות התיעוד

חלק מהקושי בעבודה על פרויקט מסוג זה הוא העדר נקודות ציון להתקדמות. כך, מוצבות כאלה באופן מלאכותי - דו"ח האמצע והדו"ח הסופי. פרט לחיבורם, הדבר העיקרי שנצבר בעבודה על הפרויקט הוא ידע, אותו מאוחז יותר מעבדים לדו"חות אלה. לכן יש חשיבות רבה לתיעוד הידע הזה, לפני, במהלך, ולאחר יצירתו. לצערנו, לא הייתה לנו שיטת תיעוד מסודרת, וכך היו דברים שתועדו יותר, והיו שלא תועדו כלל - ונאלצנו לחזור על ניסויים ובדיקות לצורך כתיבת הדו"חות. כתיבת תיעוד מסודר במהלך העבודה תחסוך מאמץ כפול זה, וכן תאפשר להבחין בתבניות גדולות ובנתיבים להתקדמות בקלות רבה יותר.

נוסף לכך, מכון שהמחקר עוסק בתופעה שחיה ומתפתחת במהלך העבודה, כך גם הידע עליה מתפתח, ויש תועלת רבה במעקב אחריו גם לאחר סקר הספרות הראשוני. שימוש פשוט שלנו בשירות Google Alerts (מעקב אחר המילה "Waledac") אפשר לנו להיות מעודכנים בתופעות והפרסומים בנושא.

מכונה וירטואלית

בפני מי שמתכוון ליצור פרויקט כזה עומדות באופן טבעי שתי אפשרויות - לבצע את ההדבקה ישירות על מחשב כלשהו, המוקדש למטרה, תוך ניטור במחשב ומחוצה לו, או לבצע את ההדבקה במכונה וירטואלית.

לטעמינו, הגישה הראשונה מערימה קשיים שלא לצורך - קשה יותר לתחזק גרסאות שונות של מצב המחשב, הניטור מסובך יותר שכן תוכנות זדוניות עשויות להתערב בפעולתן של כלי ניטור קיימים, ומצב בו המחשב כושל כתוצאה מפעילותן הזדונית גם עלול לעקב את התקדמות הפרויקט. כך העדפנו להשתמש במכונה וירטואלית, שתאפשר לנו יכולות חזקות יותר של שחזור וניהול גרסאות, חזרה על ניסויים, וניטור.

בחרנו בכלי qemu בשל היותו כלי חופשי, וכזה שמאפשר יכולות ניידות טובות של תמונות המכונה - ניתן להעביר את הקבצים בין מחשבים ומערכות הפעלה בלי כל קושי. נוסף על כך, ניתן ליצור תמונה כזאת שמאפשרת גישה חיצונית לכוני המחשב הוירטואלי, מבלי להפעילו. כך ניתן להוציא ממנה מידע במקרה של כשל, וגם להוציא מידע באופן שלא מנוהל על ידי מערכת ההפעלה הנגועה. היכולת לנתר את פעילות המכונה "מבחוץ" מונעת מהתולעת להתערב בניטור ולהסוות את פעילותה. יש לציין שגם בבחירה זו יש חסרונות, ואולי הבולט בהם הוא העובדה שיש תוכנות זדוניות שבדקות האם רצות מעל מכונה וירטואלית בכל מיני שיטות, וכך יבחרו שלא לרוץ. מדריך קצר אך ממצה לשימוש ב-qemu ניתן למצוא⁵¹ בעבודתו של ברק נירנברג על תולעת ה-Storm.

⁵¹ Storm Bot-Net, Barak Nirenberg, pg.8

ניתוח הבינארי וההדבקה

על מנת לנתר את מערכת הקבצים, ה-Registry, קריאות מערכת, השתמשנו בחבילת sysinternals⁵². כל שינוי במערכת אמור להירשם בנתיבים אלה ולכן זו דרך טובה לעקוב אחר הפעולה של התוכנה הזדונית על המחשב.

בנוסף על מנת לפענח את הקובץ הבינארי השתמשנו בכלי Disassembling הנקרא Ollydbg. Ollydbg הוא כלי חינמי וחזק ולמעשה רוב התוצאות שקיבלנו התבססו על עבודה עם כלי זה. מספר נקודות התחלה לשימוש בכלי זה:

- חלק גדול מהתוכנות הזדוניות מגיע בצורה ארוזה ומוצפנת, ישנם מספר טכניקות שבהן הן מגלות שמנסים לפענח אותן ומונעות את זה. התגברות על מכשולים כאלו זו תורה בפני עצמה, אך אם יתמזל מזלכם והן לא משתמשות בטכניקות יותר מדי מתוחכמות, תוכלו להריץ את התוכנה הזדונית, לחכות שתפענח את עצמה, ולאחר מכן לעשות אנליזה מחדש של הקוד ctrl+a.

- בנוסף יש לכלי יכולת מעולה לשלוף את כל המחרוזות אליהן קיימת הפניה בקוד - כפתור ימני על הקוד, search for, all referenced text strings. בשיטה זו ניתן להבין הרבה על פעילות הקוד.

- שימוש חשוב אחר הוא מציאת כל ההפניות למודולים אחרים, לדוגמא שימוש ב-windows api. ניתן לבצע זאת ע"י כפתור ימני על הקוד, search for, all intermodular calls.

- alt-m, נותן את מפת אזורי הזיכרון של התכנית - לדוגמא אזור הזיכרון של הקוד, איזור המחסנית (אותו גם ניתן לראות דרך אחד המסכים תוך כדי המעבר על הקוד), אזור ה-data.

- F2 - קביעת breakpoint על שורה בקוד או על אזור זיכרון.

- בנוסף ניתן גם לקבוע hardware breakpoints אשר ניתן לשים על אזור בזיכרון ולקבוע לדוגמא שיעצור ברגע שיפנו לאזור זה בזיכרון.

מקשי קיצור נוספים הן: F9- הרצה, Step Over F8, Step Into F7 - יש לציין שחלק מהקבצים מכילים מנגנון המזהה אם מבצעים stepping על הקוד וברגע שהוא מאתר זאת הוא עלול להוביל את המפענח למרדף שווא.

⁵² Sysinternals, Microsoft, <http://technet.microsoft.com/en-us/sysinternals/default.aspx>

הקלטת וניתוח תקשורת

הרכיב המרכזי השני בניתוח שלנו את פעילות התולעת היה ניתוח התקשורת. למעשה, עד שהצלחנו לפצח את הבינארי, זה הכלי היחיד שעמד לרשותנו. יש לציין שחשוב להציב את כלי ה-sniffing במקום כזה שבו התולעת לא תשבש את פעילותו - במקרה שלנו זה במחשב המארח את המכונה הווירטואלית. נקודה נוספת, ממנה התעלמנו, היא מניעת פעילות זדונית של התולעת או הגבלתה - לכל הפחות כדאי להציב Firewall בדרכה של התולעת אשר ימנע תקשורת smtp יוצאת וכך יגביל שליחת SPAM. שיטה מתוככמת יותר תהיה להציב איזה Proxy שיאט או יגביל את הפעילות הזדונית.

כלי ה-sniffing שבו השתמשנו הוא Wireshark החופשי. כלי זה מספק אפשרויות לתיעוד וניתוח התקשורת. כלי זה הוא פשוט למדי לשימוש, אם כי יש להקפיד על ה-format בו נשמר המידע - tcp dump מאפשר לשמר את כל המידע, בעוד אחרים עלולים לאבד חלק ממנו. כמו כן, כדאי להשתמש ביכולות הסינון הטובות וביכולת של follow tcp stream למעקב אחר שיחת TCP מסוימת.

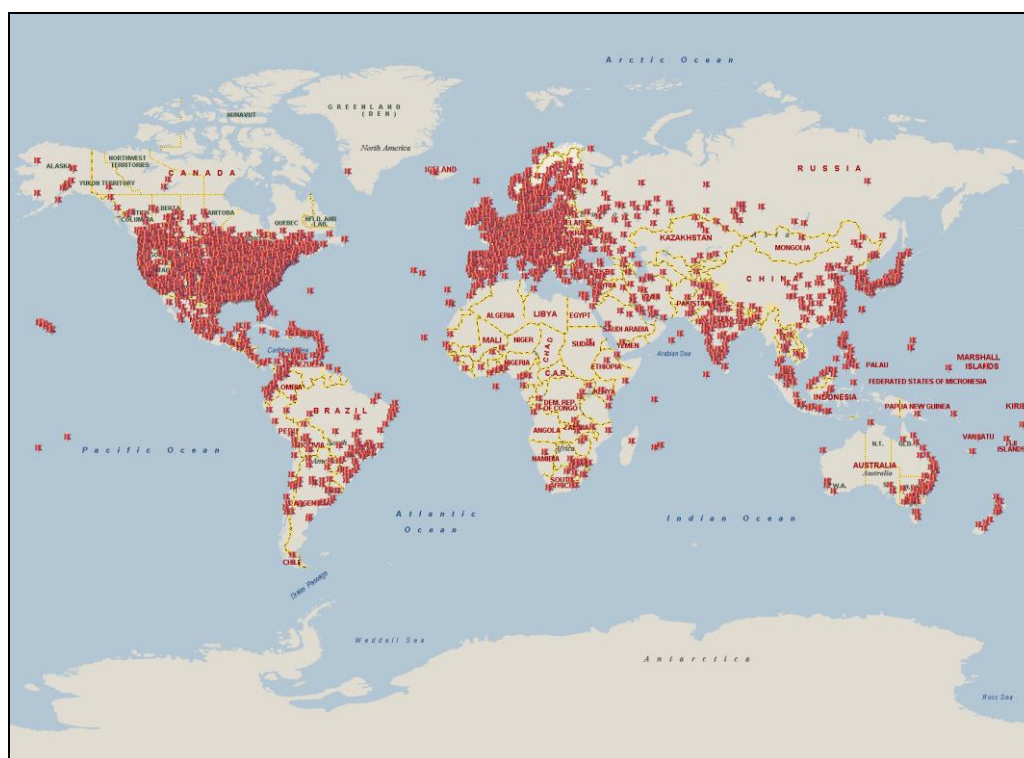
נוסף ל-Wireshark, השתמשנו גם ב-Snort, כלי לגילוי ומניעת חדירות. ניתן להשתמש ב-Snort גם על תקשורת "חיה", וגם על תקשורת שהוקלטה בעבר (למשל, בעזרת Wireshark) ב-format של tcp dump. לצורך השימוש בו יש ליצור חתימות, קיימים ברשת מספר מדריכים סבירים לכך - למשל⁵³.

⁵³ Writing Snort Rules: A Short Guide, The Shmoo Group,
<http://www.shmoo.com/~bmc/presentations/2004/honeynet/caswell-writing-snort-rules.ppt>

מאז ועד היום

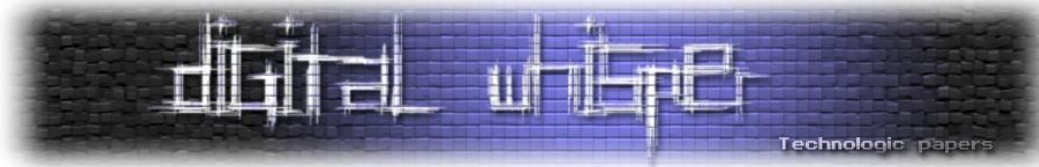
בתחילת שנת 2010, חברת Microsoft החליטה להרים את הכפפה ולעצור את התולעת. לפי חוקרי אבטחת מידע שונים שחקרו את התולעת באותו הזמן, גודל ה-Botnet כלל בין 70,000 ל-90,000 זומבים ומשלוח הספאם שנשלח באמצעותם נערך כ-1.5 מיליארד הודעות ביום (1% מכלל משלוח הספאם באותה התקופה). מטרת המבצע, שכונה "b49", הייתה לעצור את התולעת ולהוריד את התשתית עליה היא מתבססת אחת ולתמיד. במסגרת המבצע, נסגרו (בעזרת הליך משפטי) מעל 270 דומיינים אשר שימשו את תשתית ה-fast-fluxing שהגנה על שרתי ה-Command & Control של התולעת.

על פי מפת החום שפרסמה [Microsoft](#), נראה שרב המחשבים הנגועים היו באירופה ובארצות הברית:



סגירת אותם הדומיינים, חקר התולעת והפצת החתימות שלה בקרב חברות האנטי-וירוסים השביתו את התולעת לגמרי, ומאז מבצע "b49" כמעט ולא שמעו אודות התולעת.

עם זאת, ממש לפני ימים ספורים (ב-16/01/2013), חברת Symantec [פרסמה בבלוג שלה](#), עדכון על כך שנראה כי Botnet מוכר, המכונה "Virut", שכיום מורכב מכ-308,000 זומבים, החל להפיץ את גרסת D של Waledac, ולטענתם קיימת מגמת התרחבות מהירה מאוד.



לפי הנתונים ש-Symantec מפרסמים, ברגע ש-Virut מתקין את Waledac על המחשב, הוא מתחיל להפיץ כ-2,000 הודות ספאם בשעה. עד כה נראה שאחד מתוך כל ארבעה מחשבים הנגועים ב-Virut הספיק להתקין את Waledac (מה שאומר 77,000 מחשבים). אם נצליב את הנתונים, נראה שביממה אחת, הרשת החדשה של Waledac מסוגלת לשלוח כמעט 3.7 מיליארד הודעות ספאם.

האם נראה שהיוצרים של Waledac מתכננים גל שני? נכון לעכשיו - אין לדעת, אבל הנתונים בשטח בהחלט מראים מגמה כזאת.

על המחברים

מיתר קרן, בוגר תואר ראשון בהנדסת תוכנה מהטכניון, מתכנת מילדות ומתעניין בצד האפל של האינטרנט. ליצירת קשר:

me@meitarkeren.com

יונתן גולדהירש, סטודנט לדוקטורט במדעי המחשב בטכניון, ועוסק במחקר בתחום האלגוריתמים למידע גדול. נשוי באושר ומתגורר בחיפה. ליצירת קשר:

jongold@cs.technion.ac.il

לקריאה נוספת

- http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/W32_Waledac.pdf
- http://blogs.technet.com/b/microsoft_on_the_issues/archive/2010/02/24/cracking-down-on-botnets.aspx
- <http://www.darkreading.com/security/news/211201114>
- http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp_infiltrating_the_waledac_botnet_v2.pdf
- http://en.wikipedia.org/wiki/Fast_flux
- www.shadowserver.org
- www.sudosecure.net
- www.nnl-labs.com