

# Digital Whisper

גלאיון 24, ספטמבר 2011

מערכת המגזין:

מייסדים: אפיק קסטיאל, ניר אדר

móvel הפרויקט: אפיק קסטיאל

עורכים: ניר אדר, אפיק קסטיאל

כותבים: אפיק קסטיאל (cp77fk4r), ד"ר אריק פרידמן, עמי ח' פרץ קלופשטיוק, ששה גולדשטיין, שלמה יונה, לירן בנודיס ואלעד גבאי.

יש לראות בכל האמור במגזין Digital Whisper מידע כללי בלבד. כל פעולה שנעשית על פי המידע והפרטים האמורים במגזין Digital Whisper הינה על אחריות הקורא בלבד. בשום מקרה בעלי Digital Whisper ו/או הכותבים השונים אינם אחראים בשום צורה ואופן לתוצאות השימוש במידע המובא במגזין. עשיית שימוש במידע המובא במגזין הינה על אחריותו של הקורא בלבד.

פניות, תשובות, כתבות וכל הערה אחרת - נא לשלוח אל [editor@digitalwhisper.co.il](mailto:editor@digitalwhisper.co.il)

---

## דבר העורכים

---

ברוכים הבאים לגליון ה-24 של Digital Whisper! אני אשתדל להיות כמה שיותר מואפק בדקדות הקróבות, אבל... הגליון העשרים וארבע של Digital Whisper בחוץ! (OMG OMG!!! OMG OMG) וכל זה כמובן בזכותום-הקהליה (כמובן שהייתם זקנים לדחיפה קלה, אבל מה זה משנה...), במעטם זהה אני מעוניין להגיד תודה לכל מי שעוזר לנו, כתוב לנו מאמרים, עורך מאמרים של אחרים, הציע הצעות ופועל לטובת המגזין בשנתיים האחרונות:

הלו חימובי'ץ' (HLL), הרצל לוי, Zerith, סולימני יגאל, Hyp3rInj3cT10n, HMS, גדי אלכסנדרוביץ', LaBBa, Crossbow, עידן קנר, צבי קופר, עוזי יונתן קלינגר, אלכס רוכמן, בנימין כהן, ליאור ברש, ד"ר אריק פרידמן, רועי חורב (AGNil), יוסף ריסין, אורית עידן, נתנאל שיין, אביעד אביב ברזילי (NiGhT), ניר ולטמן, אייל גל, שלומי נרקוליב, עומר כהן, שי רוד (greenblast), יצחק (Zuk) אברהם, TheLeader, אוראל ארד, ליאור קפלן, עמנואל ברונשטיין (NightRanger), emanuel1234), ארץ מטולה, שלמה יונה, דנור כהן (7iAn), אורית להב (vbCrLf), אוריאל מלין (Ratinho), קיריל לשצ'יבר, אמיתי דן, יהודה גרטטל (Mo5), ד"ר אור דונקלמן, אDIR אברהם, עידן נאור (Ce@ser), סשה גולדשטיין, בר, לירן בנודיס, אלעד גבאי, סילאן דלאל, ליזה גלור-NSBS.

בשנתיים האחרונות, פורסמו במסגרת Digital Whisper 129 מאמרים שנכתבו על ידי 53 כתבים שונים, אותם מאמרים נפרסו על 1728 (!) עמודים.

וכמובן, איך לא- תודה רבה לכל מי שנרתם וכותב מאמרים לגליון החודש: תודה רבה לד"ר אריק פרידמן, תודה רבה לשלה יונה, תודה רבה לעמיחי פרץ קלופשטיין, תודה רבה לסשה גולדשטיין ותודה רבה לירן בנודיס ותודה רבה לאלעד גבאי.

### קריאה נזימה!

אפיק כסטייאל וניר אדר.

---

## תוכן עניינים

---

2	<b>דבר העורכים</b>
3	<b>תוכן עניינים</b>
4	<b>למה מומלץ לבדוק לשווא את השינויים</b>
16	<b>NOICOIN - כסף דיגיטלי ב-P2P</b>
29	<b>תזמן חוטים ב-SWINDOWS</b>
41	<b>AIR לעקב אחר גולשים בעזרת עוגיות חסינות מחיקה</b>
47	<b>תשתיות מפותחות ציבוריים</b>
53	<b>טכניקות התרבות בקרב תולעים חברותיות</b>
66	<b>HTML5 - מנוקדת מבט אחרת</b>
91	<b>דברי סיום</b>

## למה מומלץ לבדוק לסוא את השינויים

מאת: אפיק קסטיאל (cp77fk4r)

### הקדמה

במאמר זהה אני הולך להציג מקרה שקרה בחודש האחרון. מסיבות שאציג בהמשך, אשתדל (עד כמה שאפשר) לתת כמה שפחות נתונים מזהים על המקרה, למטרות שידוע לי שבאזור גוגל תוכל למצא בדיקן את המקרה שאני מדבר עליו. ובכל זאת, אני מקווה שהדבר לא יפגע בחווית הקרייה.

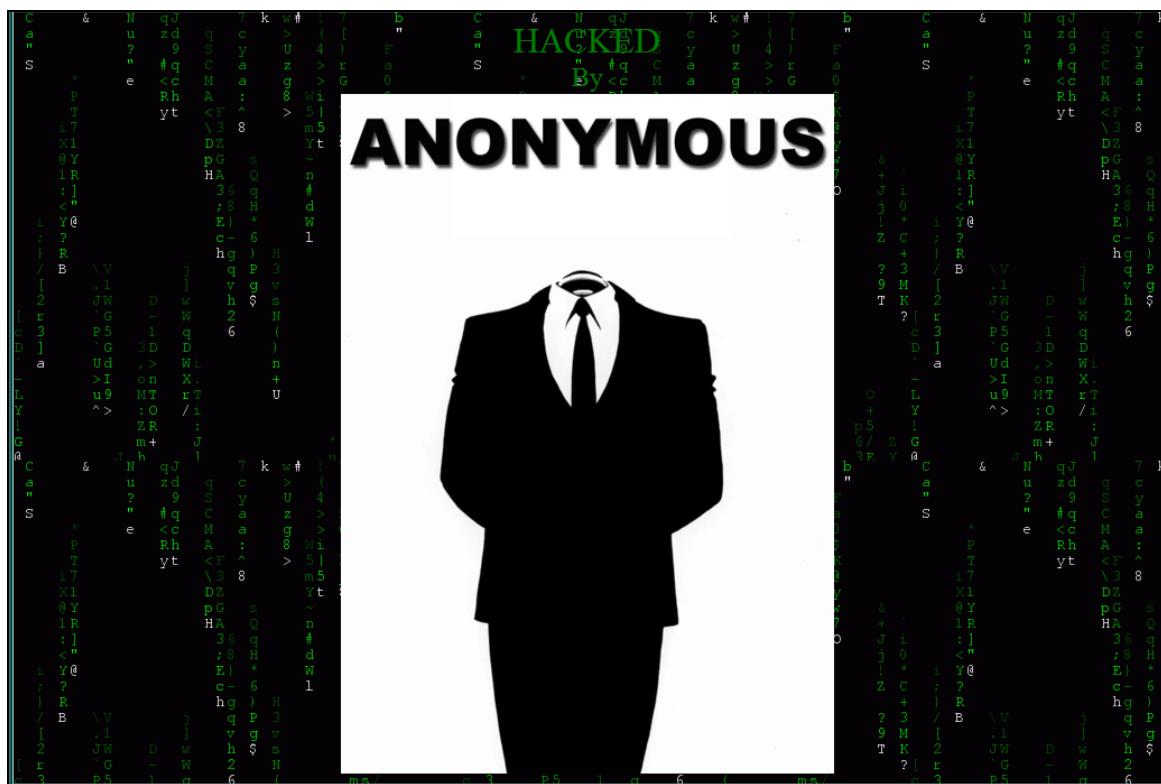
הכל התחיל מכניסה שגרתית למערכת אטג'ר ההאקיןグ שאני מנהל: [TryThisOne.com](http://TryThisOne.com) באחד מסופי השבוע. אחד הפיצרים במערכת הוא שכל משתמש יכול לפתחו לעצמו עמוד פרופיל, לנכון בו מידע על עצמו ולעצב אותו כמו שהוא מעוניין. בסופו של דבר על העמוד לעמוד לubsור אישור של צוות האתר (גם במקרה שנוכל לשולט בחומר שמתפרסם וגם במקרה שנוכל לוודא שלא מנוטים להכניס שום קוד זדוני שיפגע בשאר המשתמשים במערכת).

כאשר נכנסתי למערכת קפזה לי הטראה כי יש משתמש שערק את הפרופיל שלו והוא מעוניין שאirror את השינוי. מכניסה לעמוד העריכה, התברר לי כי השינוי הוא הוספה הקוד הבא:

```
<meta http-equiv="refresh" content="0;URL=http://trythisone.com/index.php?..."/>  
  
active:0
```

בתחילתה נראה כי מדובר בניסיון של המשתמש לעשות "Deface" טפשי בעזרת ניצול של חולשת Cross Site Scripting (שכמובן לא קיימת) ולאחר מכן עם המשתמש עלתה האפשרות שהחשבון שלו נפרץ.

הlienק מפנה לעמוד ש(בעבר היה)נראה כך:



כן, בטע, בהחלט נשמע אמי.

אישית, בלי להעליב כמובן, אני לא תופס יותר מדי מחברי קבוצת *Anonymous*, ואף פעם לא אמרתי شيئا' להאשים אותם בitter בගרות, אבל אני חושב שאם שמענו עליהם כל כך הרבה, סביר להניח שהם לא טפשים עד עדי כך, כך שברור לי שאין כל קשר בין העמוד הנ"ל לבינם.

באוטו עמוד, לחיצה על התמונה של הלוגו של *Anonymous*, הובילה לעמוד באתר שיתוף הקבצים "Mediafire.com", לקובץ בינארי השוקל קצת פחות מ-MB1 המכיל בשם, בין היתר, את המחרוזת  
."Web Hack Pack"

פה חמדתי. ☺

כאן נשאלת השאלה: למה שבן אדם שפרץ לשרת / מערכת מסויימת יפרסם את הכלים שבעזרתם הוא עשה זאת? והתשובה לכך - לא באמת מדובר בכלים לפריצה, אלא מדובר בשיטת הפצה מעניינת של סוס-טרוייאני.

למה מומלץ לבדוק לסוס את השינויים  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

הרעין הוא שבדי להפיץ את אותו הסօס הבחירה היה פורץ לאתרים, מפנה את הגולשים לאוטו עמוד, וכך לנסות לגרום להם לחשב שהאתר נפרץ על ידי אותה קבוצה ולגרום לגולשים הסקרים לנסות להריץ את אותו הכלי על ידי זה שהם יחשבו שמדובר בכליה האקדמי מתקדמים.

מבחן בראשת, התברר שלאotto עמוד הפנו מספר רב ביותר של סרטוני YouTube אשר מתגיירים להסביר לגולש כיצד פורצים לשרתים ולמערכות Web אפליקטיביות. המשותף לכל הסרטים (חוץ מהיוצר שלהם) הוא שבסופם היוצר היה מראה איך הוא מפנה את כל הגולשים לאוטו עמוד וכך בעצם, ע"י פרסום הלינק בהקשר "תמים" היה מפייצ את הכלים הדזוניים שלו.

از הרצתי את אותה ערכה לפיצת אתרים תחת WM. התוצאות משומם מה לא היו מפתיעות במיוחד... מבחינה טכנית אותו קובץ התקין Keylogger על המחשב בשם "[Ardamax Keylogger](#)". מדובר ב-Ardamax Keylogger בשם "Ardamax Keylogger" שמסוגל לתקשר עם היוזם שלו במספר דרכי כגון העלאת המידע שלו לשרת FTP שנקבע מראש או דרך התחברות לティבת אימייל ושליחת אימיילים לעצמו עם המידע מצורף attachments.

שימוש לב שתי הדרכים שהזכיר פה מאוד בעיות, בשני המקרים פרטן ההזדהות לשרת הביניים נמצאים hardcoded בקובץ שרצה בسطح. מה שאומר שאם מישו ימצא את הכליה ויחזור אותו - בקלות הרבה תהיה לו גישה לשטח האחסון שלו מעלה הסօס הטוריани את המידע אותו הוא אוגר.

דרך נconaה לעשות היא יצירת שני ערוצים שונים שבהם ניתן להתחבר לשרת הביניים: דרך אחת היא רק כתיבה **مبלי הצורך** ודרך שנייה היא גם כתיבה וגם קריאה, דרך אשר דורשת ההזדהות. הסօס הטוריани מתקשר עם שרת הביניים בעזרת הדרך הראשונה. ככה אין צורך שהוא יחזיק בפרטן ההזדהות אותו מاجر. אגב, למי שעוקב אחר סדרת המאמרים הזאת, יכול להזכיר במאמר על [Koobface](#), שפורסם בגלוון ה-14 של Digital Whisper, שבו בדיק נתקלנו במקרה שתולעת היי רק הרשות כתיבה ולא הרשות קריאה.

או.. למה מומלץ לבדוק לסוס את השינויים?

על פני השטח הריצה של הקובץ לא עושה משהו מועיל, אבל בעזרת Wireshark-ו Process Monitor אפשר לראות מהו אוטומטית ביחס למשרדים.

בלוגים של Wireshark ניתן לראות את ה-TCP Stream הבא:

Follow TCP Stream

Stream Content

Content-Transfer-Encoding: 7bit

You will find log file attached to this letter.

-----=01254C0B

Content-Type: text/html;  
.charset="iso-8859-1";  
.name="Keys\_Jul\_29\_2011\_18\_28.html"

Content-Transfer-Encoding: 7bit

Content-Disposition: attachment;  
.filename="Keys\_Jul\_29\_2011\_18\_28.html"

<HTML><HEAD><STYLE>BODY{ BACKGROUND-COLOR: #FFFFFF; FONT-SIZE: 12pt; COLOR: black; FONT-FAMILY:Courier New; }H1{ FONT-FAMILY:Arial; FONT-SIZE: 10pt; FONT-WEIGHT: normal; }.MARGIN-BOTTOM: 11px; BORDER-STYLE: solid; BORDER-COLOR: #DFDFE5; BORDER-WIDTH: 2px; BACKGROUND-COLOR: #DFDFE5; }H2 { COLOR: black; BACKGROUND-COLOR: #FFFFFF; FONT-SIZE: 12pt; FONT-WEIGHT: normal; }.MARGIN-BOTTOM: 0px; MARGIN-TOP: 10px; </STYLE></HEAD><META http-equiv="Content-Type" content="text/html; charset=utf-8"><BODY><H1>6:27 PM  
[Friday, July 29, 2011] wireshark.exe: WAN (PPP/SLIP) Interface: Capturing - wireshark</H1>http://www.mediafire.com/ <H1>6:27 PM [Friday, July 29, 2011] procmon.exe: Process Monitor - Sysinternals: www.sysinternals.com</H1>http://www.mediafire.com/ </BODY></HTML>

-----=01254C0B

Content-Type: text/html;  
.charset="iso-8859-1";  
.name="Web\_Jul\_29\_2011\_18\_28.html"

Content-Transfer-Encoding: 7bit

Content-Disposition: attachment;  
.filename="Web\_Jul\_29\_2011\_18\_28.html"

<HTML><HEAD><STYLE>BODY{ BACKGROUND-COLOR: #FFFFFF; FONT-SIZE: 12pt; COLOR: black; FONT-FAMILY:Courier New; }H1{ FONT-FAMILY:Arial; FONT-SIZE: 10pt; FONT-WEIGHT: normal; }.MARGIN-BOTTOM: 11px; BORDER-STYLE: solid; BORDER-COLOR: #DFDFE5; BORDER-WIDTH: 2px; BACKGROUND-COLOR: #DFDFE5; }H2 { COLOR: black; BACKGROUND-COLOR: #FFFFFF; FONT-SIZE: 12pt; FONT-WEIGHT: normal; }.MARGIN-BOTTOM: 0px; MARGIN-TOP: 10px; </STYLE></HEAD><META http-equiv="Content-Type" content="text/html; charset=utf-8"><BODY><H1>6:27 PM  
[Friday, July 29, 2011] : <H1><a href="http://www.mediafire.com/">http://www.mediafire.com/ </a><H1>6:27 PM  
[Friday, July 29, 2011] : <H1><a href="http://www.mediafire.com/?">http://www.mediafire.com/ </a></BODY></HTML>

-----=01254C0B--

Find Save As Print Entire conversation (3673 bytes) ASCII EBCDIC Hex Dump C Arrays Raw Help Close Filter Out This Stream

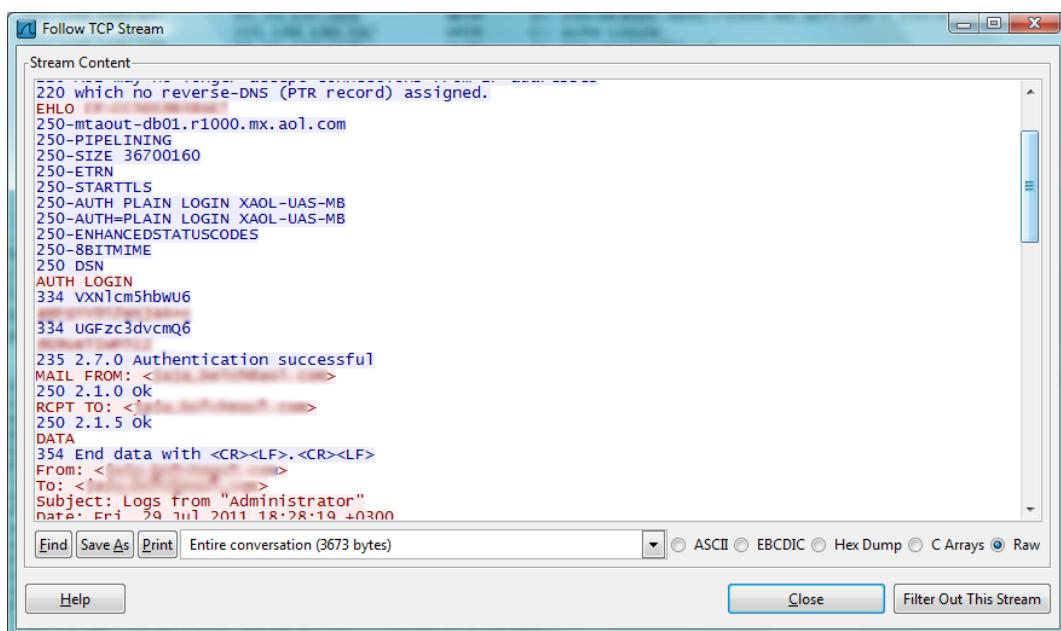
נראה שהקובץ שולח לשרת מסוים מספר קבצי HTML, אחד מהם מכיל את רשימת החולנות שלו:  
פתרונות בעמ' הרצה:

```
<HTML><HEAD><STYLE>BODY{ BACKGROUND-COLOR: #FFFFFF; FONT-SIZE: 12pt; COLOR: black; FONT-FAMILY:Courier New; }H1{ FONT-FAMILY:Arial; FONT-SIZE: 10pt; FONT-WEIGHT: normal; }MARGIN-BOTTOM: 11px; BORDER-STYLE: solid; BORDER-COLOR: #DFDFE5; BORDER-WIDTH: 2px; BACKGROUND-COLOR: #DFDFE5; }H2 { COLOR: black; BACKGROUND-COLOR: #FFFFFF; FONT-SIZE: 12pt; FONT-WEIGHT: normal; MARGIN-BOTTOM: 0px; MARGIN-TOP: 10px; }</STYLE></HEAD><META http-equiv=Content-Type content="text/html; charset=utf-8"><BODY><H1>6:27 PM [Friday, July 29, 2011] wireshark.exe: WAN (PPP/SLIP) Interface: Capturing - wireshark</H1>http://www.mediadfire.com/ <H1>6:27 PM [Friday, July 29, 2011] Procmon.exe: Process Monitor - sysinternals: www.sysinternals.com</H1>http://www.mediadfire.com/ </BODY></HTML>
```

והקובץ השני מכיל את רשימת אתרי האינטרנט שהו פתוחים בעת ההרצה של הכלל:

```
<HTML><HEAD><STYLE>BODY{ BACKGROUND-COLOR: #FFFFFF; FONT-SIZE: 12pt; COLOR: black; FONT-FAMILY:Courier New; }H1{ FONT-FAMILY:Arial; FONT-SIZE: 10pt; FONT-WEIGHT: normal; }MARGIN-BOTTOM: 11px; BORDER-STYLE: solid; BORDER-COLOR: #DFDFE5; BORDER-WIDTH: 2px; BACKGROUND-COLOR: #DFDFE5; }H2 { COLOR: black; BACKGROUND-COLOR: #FFFFFF; FONT-SIZE: 12pt; FONT-WEIGHT: normal; MARGIN-BOTTOM: 0px; MARGIN-TOP: 10px; }</STYLE></HEAD><META http-equiv=Content-Type content="text/html; charset=utf-8"><BODY><H1>6:27 PM [Friday, July 29, 2011] : </H1><a href="http://www.mediafire.com/?" style="color: red; text-decoration: underline;">http://www.mediafire.com/</a><H1>6:27 PM [Friday, July 29, 2011] : </H1><a href="http://www.mediafire.com/?" style="color: red; text-decoration: underline;">http://www.mediafire.com/</a><H1>6:27 PM [Friday, July 29, 2011] : </H1><a href="http://www.mediafire.com/?" style="color: red; text-decoration: underline;">http://www.mediafire.com/</a></BODY></HTML>
```

אך לאיפה המידע הזה נשלח? אם נסתכל קצת לפני בלוגים של Wireshark, נוכל לראות את ה-stream הבא:



از הבינארי שלנו מבצע הזרהות אל מול שירות SMTP על השרת mtaout-db01-r1000.mx.aol.com מדובר באחד משרתי המail של AOL. אישית אף פעם לא הייתה לי תיבת mail ב-AOL, אבל שירות SMTP זה דבר שאינו מכיר.

בשניה הראשונה ניתן לחשב שפרטיה הזרהות מוצפנים או מההו- אבל ברגע השני ניתן ישר להבין שפ hotspot מדבר ב- "Base64" סטנדרט.

ניתן לראות בלוגים של Process Monitor בדיק מה אותו בינארי עווה, אבל עצרתי כאן. הנחתי (מה שהסתבר לאחר מכן כמובן) שגם קובץ בינארי מעלה את המידע שלו לאותה תיבתaimail, כך שנבדוק מה עווה אותו הקובץ- פשוט נגש לתיבה ונשם נוכל לראות הכל באופן מסודר...

---

למה מומלץ לבדוק לסוא את השינויים  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

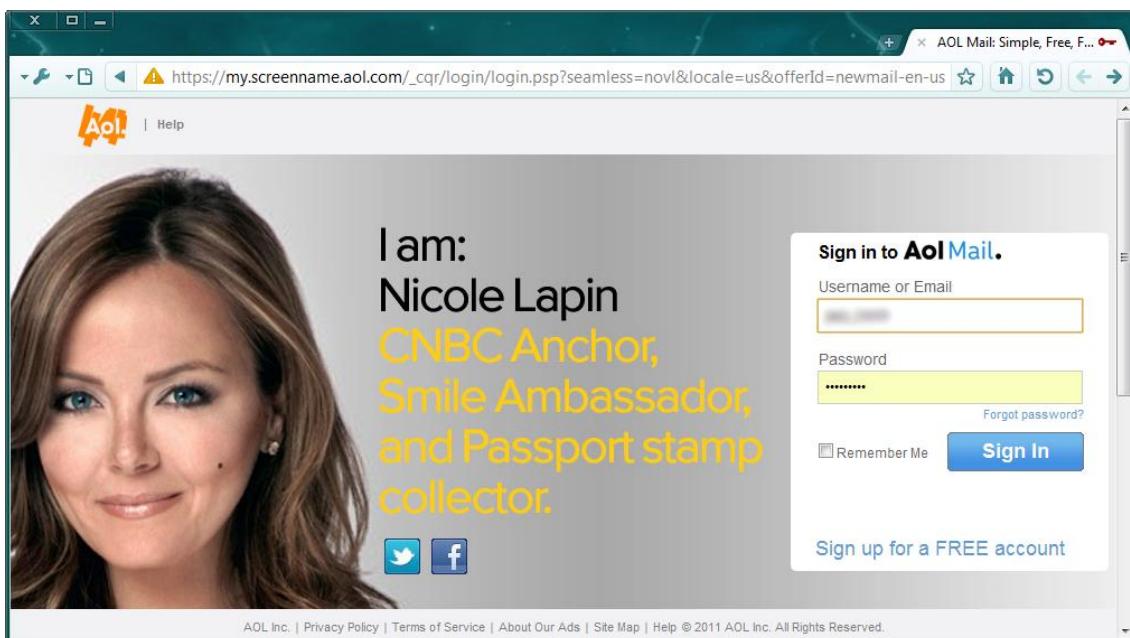
נותכל שוב על מהלך ההזדהות לתיבת:

```
AUTH LOGIN  
334 VXNlcjcm5hbwu6  
334 UGFzc3dvcmQ6  
235 2.7.0 Authentication successful
```

המרה פשוטה ל-Base64 ואפשר בקלות להחזיר את פרטי ההזדהות:

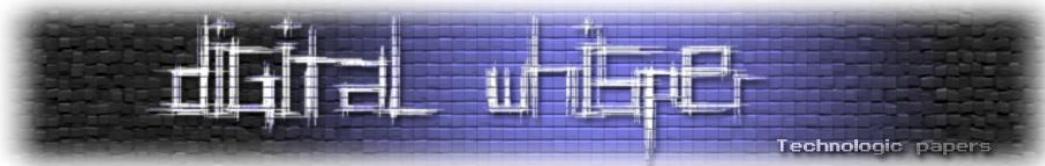
```
AUTH LOGIN  
334 Username:  
334 Password:  
235 2.7.0 Authentication successful
```

יש לנו את שם המשתמש ואת הסיסמה. כל מה שאנו צריכים לכך להכנס לתיבת:



לאחר ההזדהות מוצלחת ניתן להבין בדיקת מה שלוח אותו סויס-טרוייאני, חוץ ממה שכבר ראיינו, ניתן למצוא בטיבת 1423 אימיילים שונים (והמספר עלה כל כמה דקות), שכלי אחד מהם הכיל נתונים כגון:

- תצלומי מסך (Print Screens)
- לוג הקלדות (Key Strokes),
- תהליכי וחלונות פעילים.
- אתרים שבהם ביקר המשתמש וכו'



מספר תמונות להמחשה (פרטים רבים יוצגו באופן מוטשטש בכך לשומר על פרטיותם של הקורבנות).  
כאן ניתן לראות את כמהו המילימ' הנקנכים:

The AOL Mail interface shows the inbox with 1423 new messages. One message is highlighted with the subject "Ex-MTV Producer Faces Charges". Another message is from "AP Reagan County Sheriff's Dept." with the subject "Rocky Start to Sex Assault Case". A third message is from "AP" with the subject "Jeffs raised eyebrows with some surprising behavior on the first day of his trial". A fourth message is from "Why family says US gov. failed him" with the subject "Bizarre legal move he made in court". The interface includes a search bar, a weather widget for Orlando, FL (85°F), and various AOL services like AIM and My Maps.

כאן ניתן לראות אימיילים שהגיעו מסօס-טרויאני שנמצא על מחשב בשם "E\*\*\* \*\*\*\*\*e":

The AOL Mail interface shows the inbox with 1423 new messages. A message from "Logs from 'E'" is listed multiple times with different dates and times. The interface includes a search bar, a weather widget for Orlando, FL (85°F), and various AOL services like AIM and My Maps.

(מספר רב של קורבות נניתן היה לאתר- אותו והסביר להם כיצד ניתן להסיר את המזיק)

כאן ניתן לראות את תוכן של אימיילים שהגיעו לאחר ההרצתה של הכלי על ה-VM שלנו!

The AOL Mail interface shows the inbox with 1420 new messages. A message from "Logs from 'Administrator'" is listed twice with the date and time 7:28 pm and 2:29 pm respectively. The interface includes a search bar, a weather widget for Orlando, FL (85°F), and various AOL services like AIM and My Maps.

(הימיילים הספציפיים שנשלחו מה-VM שלנו נמחקו מהתיבת)

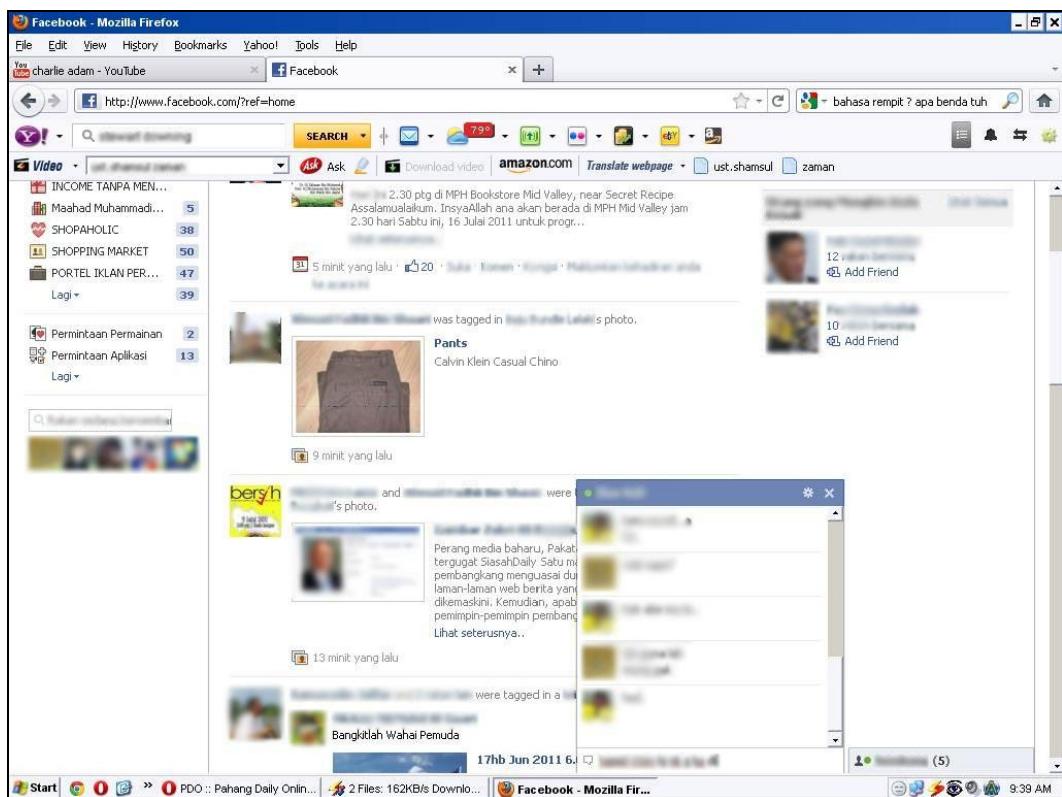
למה מומלץ לבדוק לsono את השינויים

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

התמונות הבאות הן מספר תמונות מסך שהגינו ממחשבים שונים:



למה מומלץ לבדוק לסוי את השינויים  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



למה מומלץ לבדוק לסוא את השינויים  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

ועוד ועוד... אותה תיבת הייתה מלאה בדברים בסיגנון זהה. במעבר על המיילים שהגיעו לתיבת ניתן לראות שבעל החשבון סימן בדגל-אדום (אופציה מובנת ב-AOL לשימון מיילים מעוניינים/חשובים) כלإيمיל שהגיע מהקוורבנות שלו שבהם היה מידע עם סיסמאות לחשבונות שונים.

### השחתת הטרויאן

אחרי הבנתה של איך הטרויאן המטרה שלי הייתה להשבית את הכלי כדי שלא יפגע על משתמשים נוספים.

מחיפוש באימיילים הראשונים של התיבה, ניתן היה למצאו מספר נסיבות שאומרו בחור ניסה על עצמו. החיפוש עלה יפה ונמצאו מספר ממצאים יפים. **ראשית**- תמונה מסך מכל' שהוורץ על המחשב של בעל הכללי:

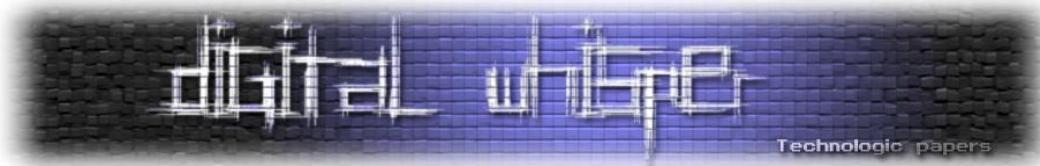


**שנית**- אוורר לוג הקשות מקלדת וボ נצפה בעל החשבון מתחבר לחשבון המיל המקורי שלו ב-"!oooh.Yahoo".

מה עוד אפשר לבקש? ☺ עם גישה לחשבון המקורי של המשתמש הושגה נגישות לחשבון שבו אותו משתמש אחסן את עמוד ה-Deface (שהוזג בתחילת המאמר) והוחלף בעמוד אחר, שומו לו פרט' ההזהדות ואיתה ספקית אחסן קיבלה מייל עם המידע על הפעולות הנעשות מאותו בחשבון.

---

למה מומלץ לבדוק לסוא את השינויים  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



בנוסף הושגו פרטי ההזדהות לחשבון youtube שלו (החשבון שבו היי סרטוניים שהפנו לדף עם הטרויין) וגם לשם שונו פרטי ההזדהות. שבוטטו של דבר הכל (לפחות הנוכחי) נוטREL. כמובן שאף אחד לא מבטיח לנו שמחר אותו יلد לא יתעורר ב�отREL ויקים הכל מחדש תחת חשבון חדשים...

האימיל עם פרטי ההזדהות לשטח האחסון שבו אוחסן עמוד ה-Deface:

**AWARDSPACE.COM**

**Web Hosting, Domain Names & Online Services**

**Successful Signup!**

PLEASE PRINT THIS EMAIL FOR YOUR RECORDS AND READ IT THOROUGHLY!

Dear Valued Customer,

Thank you for purchasing web hosting with awardspace.com.

**Hosting Information**  
Hosting Package: FREE Web  
Client ID:  
Login email:  
Password: [REDACTED]

**Domain Names**  
If you have purchased domains registration/transfer or you have existing domains, please add them inside your Domain Manager section. Additionally you should set the following name servers for all domains, except for the domains that has been registered with us.  
Nameserver 1: ns5.awardspace.com  
Nameserver 2: ns6.awardspace.com

**Website Upload**  
Make sure you upload your files to the domain/subdomain directory on the server; otherwise they will not be visible on the Internet. Also, please be sure that your homepage is saved as an "index" file e.g., index.php, index.html, index.htm, etc. We suggest you download some advanced ftp client to manage your files quickly, or use the File Manager inside the Hosting Control Panel.

**FTP Account Information**  
Your default FTP account information:  
  
FTP Hostname: You should first add a domain/subdomain in your Hosting Control Panel.  
FTP Username: [REDACTED]  
FTP Password: [REDACTED]  
You can manage your FTP accounts from the FTP Manager section.

**E-mail Account Information**

#### התחברות לחשבון עם התוכן הפוגע:

TIP: Connecting to the FTP server and upload your website, you should have FTP hostname (domain/subdomain).

File Manager Help!

Current folder: /home/www/

Warning: Directory protection for your hostnames is **disabled** - you can delete them by accident.  
(You CAN delete hostname directories and upload files outside them) Directory protection:

Options	Name	Size	Type	Date Modified	Permissions
	..	-	Folder	Jan 31 19:15	drwxr-xr-x
	[REDACTED]	-	Folder	Jan 31 19:15	drwxr-xr-x
	[REDACTED]	-	Folder	Jul 8 22:02	drwxr-xr-x
<input type="checkbox"/> Select All	Total: 1 folders				

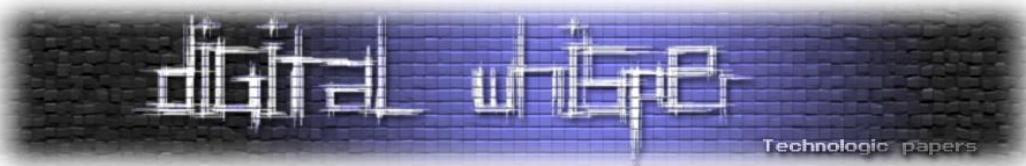
Current folder: /home/www/

Warning: Directory protection for your hostnames is **disabled** - you can delete them by accident.  
(You CAN delete hostname directories and upload files outside them) Directory protection:

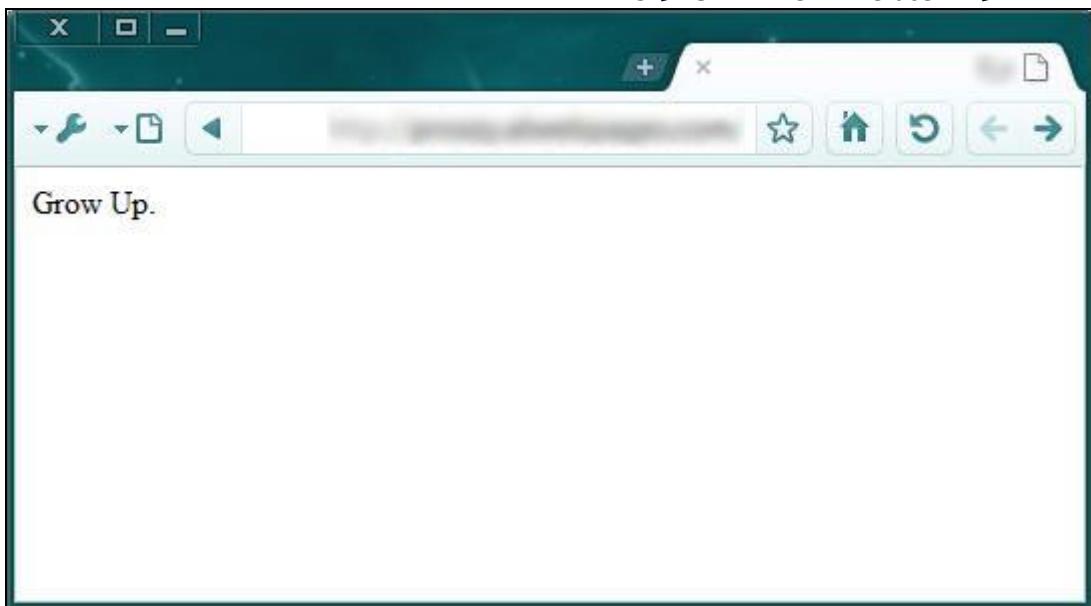
Options	Name	Size	Type	Date Modified	Permissions
<input type="checkbox"/> Select All	Total: 1 folders				

למה מומלץ לבדוק לסייע את השינויים

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



והתוצאה... עמוד Deface חדש ובלתי פוגע ☺



## סיכום

כآن פחות או יותר הסיפור נגמר. את המטרה שהצבתי לעצמי- השגתי. אומנם אני לא בטוח שכתה האינטרנט הוא מקום בטוח לגלוש בו, אבל לפחות אני יודע שלא העלמתי עין, ואם בזכות הפעולה הזאת הצלחנו להציג עוד כמה משתמשים תמים- את שלנו עשינו.

בקשר לחוקיות הדברים, ברור לי שמדובר בפעולות אפורות, וכששאלתי עורך דין המתעסק בנושא האם פרסום הדברים כדאי- קיבلت תשובה שלילית. ובכל זאת פרסמתי את הדברים, והנה הם לפניכם. הדבר געשה אך ורק במקרה לכם, הקוראים, שניתן להשיב לאותם סקריפט-קידיז שימושים לפוגע במסתמשים תמים ברחבי האינטרנט. מאמר זה אינו מהו המליצה לביצוע פעולות דומות מצדכם - כאמור מדובר בתחום מאד אפור.

---

למה מומלץ לבדוק לסוא את השינויים  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

## cosa-Digitali-B-P2P - Bitcoin

מאת: ד"ר אריך פרידמן

"The chief value of money lies in the fact that one lives in a world in which it is overestimated."

H.L. Mencken



### הקדמה

בשנתים האחרונים הגיע סוג חדש של מטבע. מטבע שמאים להפוך על-פיהם את סדרי הכלכלה העולמית ולשבור את התלות במדינות ובנקים בכל הנוגע להעברת כספים בין פרטים. הביטקוין.

הביטקוין הואcosa-Digitali. הרעיון שלcosa-Digitali כשלעצמו אינו חדש, ונחקר כבר למעלה מ-20 שנה ("יצוג דיגיטלי שלcosa-Digitali הוא אמיתי כמובן ותיק בהרבה"). יתר על כן, מטבעות דיגיטליים נמצאים כבר בשימוש נרחב, למשל נקודות מיקרוסופט לרכישת משחקי Xbox, או Facebook Credits לביצוע עסקאות בפייסבוק. החידוש והיחוד שלביטקוין הוא שמדובר במטבע שאין אף גוף מרכזי שלול או מפקח עליון, אלא הוא מנוהל על-ידי מערכת Peer to Peer. לאי-תלות זו יש מספר השלכות מעניות. ראשית, הצורך במתווכים (בנקים) לביצוע עסקאות במערכות המסורתיות הכרך בעמלות, שמייקרות את עלות העסקאות. עלות זו הופכת להיות בעייתית במיוחד כשמדבר בעסקאות קטנות (מיקרו-תשולם). שנית, במערכות המסורתיות אין יכולת לבצע תשלים לא הפיכים תמורה שירותים שאינם הפיכים, אלא אם נעשה שימוש בסוף פיסי. הונאותן הן חלק מהמשחק, ותמיד יש סיכון שהעסקה לא תucceed (צ'ק שחווץ, עסקת אשראי שבוטלה). מערכת הביטקוין מבטיחה לתת מענה לבעיות אלה, תוך התבוססות על כלים קרייפטוגרפיים ולא צורך באמון בגורם צד ג' כגון בנק או חברת אשראי. אחד האתגרים הגדולים במערכת כזו הוא למנוע שימוש כפול במטבע (כלומר תשלום עם אותו מטבע ביותר עסקה אחת בו זמן). במאמר זה נבחן כיצד עובדת מערכת ביטקוין, וכייזה היא מתמודדת עם הונאות.

המאמר מניח כי הקורא מכיר את המושגים של מפתחות פרטיים וציבוריים, חתימות דיגיטליות, ופונקציות תמצאות קרייפטוגרפיות. המעניינים מייד על מפתחות ציבוריים וחתימות יכולים לקרוא את המאמר [של הל חימוביץ בנושא](#) בgelion 3, והנושא של פונקציות תמצאות קרייפטוגרפיות תואר [במאמר של ד"ר או דונל קלמן](#) בgelion 21.

## מיאיפה הגיע ?BitCoin

המערכת של ביטקון הינה לעובד ב-2009. לא ידוע מי מקיים המערכת - הוא השתמש בשם הבדוי סטושי נקמוטו (Satoshi Nakamoto) וטען שהוא מילן (אם כי נראה שיש לו שליטה מעולה באנגלית), ולא נראה שימוש ביפנית באתר או בקוד של ביטקון). סטושי פרסם את עקרונות המערכת במסמך טכני, ובמקביל האתר של ביטקון ספק את התוכנה עצמה ופרטים נוספים. עם זאת, אין צורך להסתמך על מהימנותו של סטושי כדי לבתו בביטקון. התוכנה של ביטקון מנוהלת תוכנת קוד פתוח, וכל אחד יכול לגשת אתה של ביטקון ב-Sourceforge, להוריד את הקוד, לבחון אותו, ולהדר אותו בעצמו.

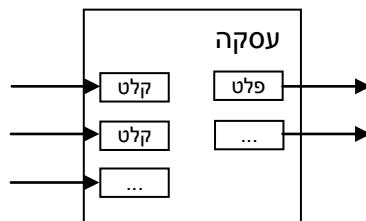
## מי יכול להשתמש ב-BitCoin ?

כל אחד יכול להשתמש בביטקון. כל מה שנדרש הוא להוריד את התוכנה של ביטקון האתר של ביטקון, ואפשר להתחילה לקבל ולבצע תשלומים בביטקון. מאחר ואין גוף מרכזי שמנהל את המערכת, אין צורך לפתיחת חשבון או להזדהות. עם התקנת התוכנה, מוקצתה למשתמש זוג מפתחות ציבורי-פרטי חדש, "כתובתו" של המשתמש נגזרת ישירות מהמפתח הציבורי שלו, וניתן להשתמש בה כדי לתשלום. המשתמש יכול לייצר לעצמו מפתחות נוספים במקרה נפשו. למעשה, התוכנה של ביטקון מעודדת זאת, וכך אשר מקבל תשלום בכתבתו של המשתמש, התוכנה תייצר מיד מפתחות חדשים (וכתובה חדשה) לשימוש עסקה הבאה. באופן זה, יהיה קשה לקשר עסקים שונים שביצעו אותו המשתמש, אם נעשו באמצעות כתובות ביטקון שונות.



תוכנת ביטקון היא תוכנת P2P, והיא מוצאת משתתפים אחרים ברשת באמצעות חיבור לשרת IRC (ערוץ #bitcoin בשרת irc.Ifnet.org). במידה ולא הצלחה לייצר קשר צזה, נעשה שימוש ברשימת משתתפים הכלולה בקוד התוכנית, ודרכם ניתן לייצר קשר עם משתתפים נוספים.

**ביטקון** היא למשה מערכות לעיבוד עסקאות, והן מוחאות את לב המערכת. לכל עסקה יהיו קלטים ופלטטים:



הקליטים הם עסקאות קודמות, המגדירות סכומי מטבעות המשמשים בעסקה החדשה, והפלטיהם הם סכומי המטבעות המוקצחים למשתתפים בעסקה (ומהווים בסיס לעסקאות עתידיות). בדרך כלל יהיה שני פלטיטים: אחד עבור הקצתת סכום למוטב בעסקה, ואחד כדי לייצג את העודף למבצע העסקה. ברוב העסקאות סכום המטבעות בקלט וסכום המטבעות בפלט יהיו שווים, אולם יש גם סוג מיוחד של עסקאות המשמשות להטבעת" מטבעות חדשים. וזאת עסקת הטבעת מפתחות:

כעסקה להטבעת מטבח חדש אין לה קלטים (prev\_out מאופס), ויש לה פלט אחד, המקצתה 50 ביטוקין לבעל המפתח הפומבי שמתחליל ב-...04283. העסקה מייצגת באמצעות התמצאות שלה, המתחליל ב-8ee...f5f5fp ומודיעה בתחילה.

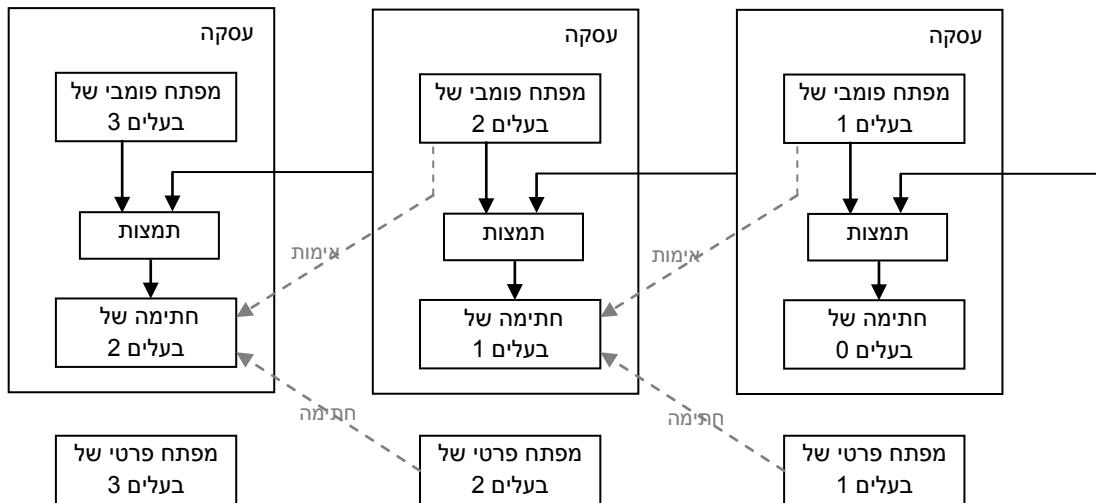
להלן עסקת העברת COPIM:

```
{  
    "hash": "5a4ebf66822b0b2d56bd9dc64ece0bc38ee7844a23ff1d7320a88c5fdb2ad3e2",  
    "ver": 1,  
    "vin_sz": 1,  
    "vout_sz": 1,  
    "lock_time": 0,  
    "size": 158,  
    "in": [  
        {  
            "prev_out": {  
                "hash": "f5d8ee39a430901c91a5917b9f2dc19d6d1a0e9cea205b009ca73dd04470b9a6",  
                "n": 0  
            },  
            "scriptSig": "304502206e21798a42fae0e854281abd38bacd1aeed3ee3738d9e1446618c4571d10  
90db022100e2ac980643b0b82c0e88ffdfecc6b64e3e6ba35e7ba5fdd7d5d6cc8d25c6b241501"  
        }  
    ],  
    "out": [  
        {  
            "value": "50.00000000",  
            "scriptPubKey": "OP_DUP OP_HASH160 404371705fa9bd789a2fcfd52d2c580b65d35549d  
OP_EQUALVERIFY OP_CHECKSIG"  
        }  
    ]  
}
```

ניתן לראות כי הקלט לעסקה חז' הוא העסקה הקודמת (שערכה 50 ביטקווין), ויש לה פלט יחיד המקצת את כל 50 הביטקווין למשתמש שהמצאות המפתח הפומבי שלו מתחילה ב-40437...

למעשה, עסקאות אלה מגדירות את המטבעות ואת בעלייהם. באופן כללי, כל מטבע מוגדר על-ידי סדרת עסקאות, הראשונה בהן היא עסקת הטבעת המטבע, ואחריה יש שרשרת של עסקאות המתארות את שרשרת הבעלות על המטבע. כדי להבהיר מטבע, בעל המטבע מייצר עסקה חדשה. עסקה זו תכלול את התמצאות של העסקה הקודמת (כלומר, ממשיכים את שרשרת הבעלות) ואת המפתחות הפומביים של נווטן המטבע ומקבל המטבע, והיא תהייה חתומה על-ידי הבעלים של המטבע. קיימת אפשרות גם להציג עמליה עבור עיבוד העסקה (פרטים בהמשך). התהילה מתואר בתרשימים בעמוד הבא. כיוון שרק לבעליים החוקיים יש את המפתח הפרטי המתאים למפתח הפומבי בעסקה האחורונה בשרשרת, הוא היחיד שמסוגל לייצר חתימה תקינה ובכך להכריז על העסקה. עם זאת, העסקה החתוםה מהוות רק ראייה ראשונית (אם כי עדין לא מספקת, כפי שנראה) לכך שהועברו המטבעות.

אם כך, עסקאות ביטקווין מאפשרות לייצר מטבעות ולהעביר אותם מיד ליד. אבל מה בעצם מונע מכל אחד לייצר מטבעות כאלה نفسه? או להשתמש באותו מטבע פעמיים באמצעות יצירת שתי עסקאות שונות על-בסיס אותה עסקה קיימת? התשובה טמונה בתהילך עיבוד העסקאות של ביטקווין, המתבסס על יצירת בלוקים של עסקאות.



## שימוש עסקאות BitCoin - בלוקים והוכחת עבודה

כאשר מישהו מייצר עסקת ביטקון וחותם עליה עם המפתח הפרטוי שלו, הוא מכירז על נכונותו לבצע את התשלום, אולם העסקה לא תקפה עד שמשה קרייטית של משתתפים במערכת ה-2P2C של ביטקון מכירים בעסקה ומקבלים אותה. הדרך שבה עסקאות נבחנות ומתתקבלות על-ידי המשתתפים במערכת היא באמצעות תהליך עיבוד בלוקים של עסקאות. היסטוריות העסקאות של ביטקון מתוחזקת על-בסיס שרשרת ייחודית המורכבת מבlokים, כאשר כל בלוק מכיל עסקה אחת או יותר. נכון בזמן כתיבת שורת אלה, השרשרת מכילה 142,620 בלוקים. הבלוק הראשון בשרשרת (בלוק 0) נקרא בלוק בראשית (The Genesis block), והוא נראה כך:

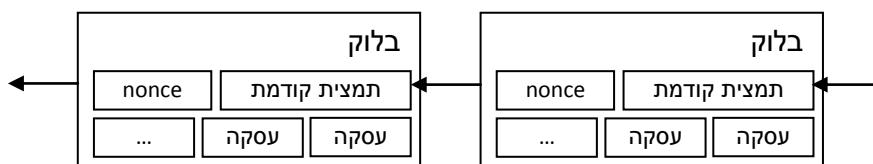
P2P - Bitcoin - סוף דיגיטלי ב-  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

```

    "n": 4294967295
  },
  "coinbase": "04ffff001d0104455468652054696d65732030332f4a616e2f32303039204368616e63656c6c6f
72206f6e206272696e6b206f66207365636f6e64206261696c6f757420666f722062616e6b73"
  ],
  "out": [
    {
      "value": "50.00000000",
      "scriptPubKey": "04678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0e1f61deb649f6bc3f
4cef38c4f35504e51ec112de5c384df7ba0b8d578a4c702b6bf11d5f OP_CHECKSIG"
    }
  ]
},
"mrkl tree": [
  "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b"
]
}

```

הבלוק הזה כולל **עסקה אחת** (50 הביטוקין הראשונים שנוצרו). הקוד של ביטוקין כולל את הבלוק זהה, כך שכל לקוח ביטוקין מסוגל לוודא שרשרת הבלוקים מתחילה בבלוק הבראשית. כל בלוק נוסף בשרשרת כולל את התמצית של הבלוק שלפניו, עסקה אחת או יותר, ערך מסווני בשם nonce (פרטים בהמשך), ותמצית של ערכי הבלוק. כך נוצרת שרשרת הbabah:



שרשרת בלוקים חוקית חייבת להתחילה בבלוק הבראשית, אולם ניתן ליצור יותר שרשרת אחת עצמאית. במקרה ומשתתפים ברשות מכירים על יותר שרשרת חוקית אחת המערכת, מערכת ביטוקין (כלומר האוסף של המשתתפים החברים בה) תמיד תיקח את השרשרת הארוכה ביותר. כפי שנראה בהמשך, יש צורך בהשquette משבבים חישוביים ניכרים לצורך ייצור שרשרת חוקית. במידה וגורם זדוני כלשהו ירצה לשכטב את היסטוריה העסקאות ולספק שרשרת חלופית, הוא יצטרך לספק שרשרת ארוכה יותר מהשרשרת הקיימת. משימה זו תדרוש שהיא ברשותו כוח חישובי העולה על זה של כל שאר המשתתפים במערכת ביחיד. במידה ואין לו כוחות חישוביים כאלה, זיווג שרשרות הופך להיות משימה בלתי אפשרית מכל בחינה מעשית. במידה ונוצר מצב ואכן לגורם כלשהו יש שליטה על רוב הכוחות החישוביים במערכת, הרי שככל הנראה יוכל להרוויח יותר על-ידי עיבוד עסקאות וייצור בלוקים "לפי הספר", אשר באמצעות רמות שתגרום לאובדן אמון במערכת ובריחת המשתתפים. באופן זה המערכת מתוכננת לתמוך בביצוע מהימן של עסקאות המשתתפים.

## כף לא צומח על העצים

"How can you have money," demanded Ford, "if none of you actually produce anything? It doesn't grow on trees you know."

"If you would allow me to continue..."

Ford nodded dejectedly.

"Thank you. Since we decided a few weeks ago to adopt leaves as legal tender, we have, of course, all become immensely rich."

Ford stared in disbelief at the crowd who were murmuring appreciatively at this and greedily fingering the wads of leaves with which their track suits were stuffed.

"But we have also," continued the Management Consultant, "run into a small inflation problem on account of the high level of leaf availability, which means that, I gather, the current going rate has something like three major deciduous forests buying one ship's peanut."

Murmurs of alarm came from the crowd. The Management Consultant waved them down.

"So in order to obviate this problem," he continued, "and effectively revalueate the leaf, we are about to embark on a massive defoliation campaign, and... er, burn down all the forests. I think you'll all agree that's a sensible move under the circumstances."

(Life, the Universe and Everything, Douglas Adams)

כפי שיתואר בהמשך, יצור בלוק כרוך בעבודה חישובית, והמערכת מכונה לייצר בלוק חדש בערך כל 10 דקות. העסקה הראשונה (ולעתים היחידה) בכל בלוק תהיה תמיד עסקה המוניקה מטבעות ביטקון לייצר הבלוק כשר על העמל החישובי. חלק משכר זה מגיע למטבעות חדשים שנוצרו (כלומר העסקה הראשונה היא עסקת הטבעת מטבעות), וחלקו מגיעים מעמולות שמציעים מייצרי העסקאות. הקוד של ביטקון נכתב כך שב-4 השנים הראשונות, כל בלוק חדש ייצור 50 ביטקון (כך שיוצרים 0 10,500,000 מטבעות בתקופה זו). כמות זו תחולק בחצי כל 4 שנים, וכך שבשנים 8-4 יתקבלו 25 ביטקון על כל בלוק (עם 5,250,000 מטבעות חדשים בתקופה), בשנים 12-8 יתקבלו 12.5 ביטקון וכן הלאה. לאורך זמן, כמות המטבעות הכוללת תתקרב ל-21,00,000, ואף פעמי לא תעבור את רף זה - בשנת 2140 התשלום במטבעות חדשים על יצירת בלוק יאופי. תכנון זה של המערכת צופה שבשלבי החיים המאוחרים יותר של המערכת העלות של עיבוד העסקה תcosa בעיקרה על-ידי העמלות שמציעים מייצרי העסקאות, והצורך בתמരיך של מטבעות חדשים יפחת. כמו כן, המערכת צוז לעולם לא תהיה אינפלציה - יש חסם עליון וקשה לכמות הכספי שיווצר.

אחד המרכיבים המרכזיים במערכת ביטקון הוא הוכחת עבודה (Proof of work). עקרון זה הוצע לראשונה ב-1997 על ידי אדם בק, בהקשר של מערכת [hashcash](#) למניעת ספאם. הרעיון היה שכל משלוח דואר אלקטרוני יצריך את השולח לכך שהשקייע כמות מסוימת של משאבי עיבוד בשילוח ההודעה, למשל, חישוב שיצירך זמן עיבוד של שתי שניות. עבור משתמשים רגילים, זה לא היה משפיע באופן מהותי, שכן עיכוב של שתי שניות יהיה כמעט ונבלתי מורגש מבחינכם. לעומת זאת, עבור

ספאמרים השולחים מיליוןים של תכנתות אלקטרוניות תידרש השקעה משמעותית של זמן וכסף, דבר שיהפוך את עסקי הספאם ללא כדאים. בධיבד, שיטה זו התרבורה לא מוצלחת למניעת ספאם, בעיקר מפני ששימוש בה היה פוגע גם בגופים השולחים כמויות גדולות של דואר לגיטימי. אולם הרעיון יושם בהצלחה במסגרת המערכת של ביטוקין.

על מנת שבлок יהיה חוקי, עליו לקיים את התנאי הבא: התמצית של הבלוק צריכה להיות קטנה מערך מטרה מסוים, המוסכם על-ידי המערכת. במלים אחרות, התמצית של הבלוק צריכה להתחילה בשרשרת של אפסים. המפתח לייצור תמצית חדשה טמון במחוזת nonce הנמצאת בבלוק. תהליך ייצור בלוק חדש מתבצע באופן הבא:

1. מצא את שרשרת הבלוקים החוקית הארוכה ביותר במערכת.
2. הגרל מחוזות nonce. יצר בלוק חדש המורכב מהתמצית הבלוק האחרון בשרשרת, מהעסקאות הממתינות לעיבוד (כולל עסקת ייצור מטבעות המעניינה תשולם למיצר הבלוק) ומה-nonce.
3. אם ערך התמצית של הבלוק החדש מתחילה במספר האפסים החדש, פרסם את הבלוק.
4. אחרת חזר ל-2 עם nonce אחר.

אם בשלב כלשהו בתהליך תפורסם שרשרת בלוקים חוקית נוספת (כלומר מישהו הצליח לחשב את הבלוק הבא), אז תהליך ייצור הבלוק יעבור להتابסס על שרשרת זו. במידה ומישהו מנסה לייצר בלוק שאינו עומד בכללים של המערכת (למשל, אי-התאמה בחישובי תמצות, חריגה מסכם הכסף שמאյית למיצר הבלוק), הבלוק ידחה על-ידי שאר המשתתפים בראשת, וכך ימנעו רמאיות.

ערך המטרה של התמצית, או מספר האפסים החדש, נקבע על-פי הזמן שהPLIC ייצור את 2016 הבלוקים הקרובים. המטרה היא להتاימים את רמת הקושי של הביעיה החישובית לכמות המשאים החישוביים הקיימים בראשת, כך שבממוצע יוצר בלוק חדש כל 10 דקות.

באטר <http://blockexplorer.com> ניתן לעקוב אחר ייצור הבלוקים, ולראות את כל ההיסטוריה הבלוקים (והעסקאות) עד כה.

בימים המוקדמים יותר של המערכת, תוכנת ביטוקין כללה את האפשרות להשתתף בתהליך עיבוד הבלוקים, ומכאן גם את יכולת לזכות במטבעות ביטוקין. אולם עם התפשטות המערכת ועלייה הערך של ביטוקין, חל תהליך של התמצחות ביצור מטבעות. למשל, קרטיים גרפיים יכולים ליצור מטבעות בקצב מהיר פי 100 ויותר מאשר מעבד מחשב רגיל. התמരיץ ליצור מטבעות עלה, והחלו להופיע חוות לכריית ביטוקין. למעשה, צריכת החשמל של חוות כריית ביטוקין מזקירה את זו של חוות לגידול מריחואנה, ובכבר דוח על מקרים שמשטרת פרצה לבתים במטרה למצוא מגדי סמים ומצאה גיקים עם מחשבים.

בנוסף, משתתפים החלו להתאגד לכרייה משותפת של מטבעות (soloos), ויצרו קבוצות כדוגמת [deepbit](#) ו-[btcguild-slush](#).

עם גידילת משאבי החישוב בראשת, רמת הקושי של הבעייה החישובית עלה (నכוון לכתיבת שורות אלה דרישה במתכנית תחילה של 52 אפסים). לאחר ובמצב הנוכחי, עלות כריית הביטקון תשתלם רק עבור אלה הבונים מערכות ייעודית שתבצע את הכרייה ביעילות, האפשרות לכרייה הוסרה מהתוכנה, ויש צורך להוריד תוכנה ייעודית לכריית ביטקוין.

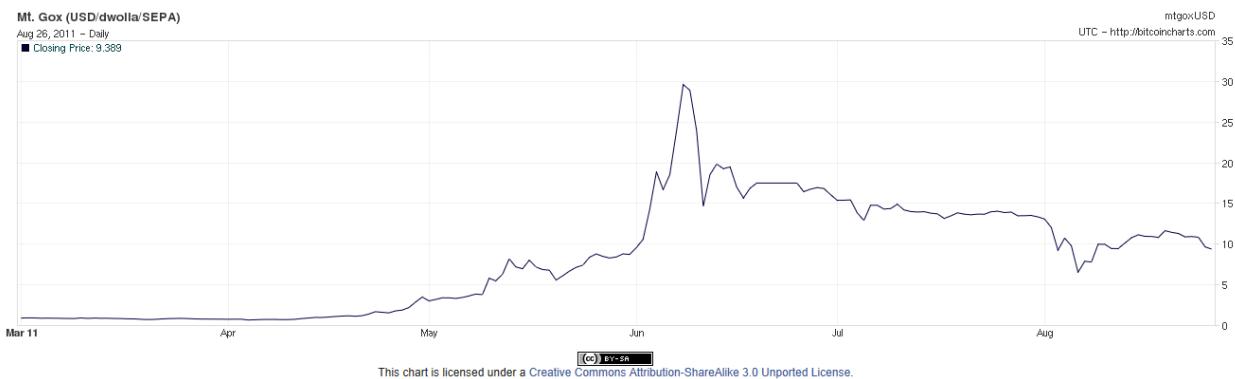


(מכרה ביטקוין ברוסיה. התמונה נלקחה מ-<http://bitcoinalitalia.com/forum/viewtopic.php?f=11&t=26>)

## שווייה משווה, ה-*how?* BitCoin?

העסקה המתועדת הראשונה בה נרכש פריט פיסי תמורת ביטקון התקיימה ב-22 במאי 2010. בעסקה זו המשתמש szasz שילם 10,000 ביטקון עבור פיצה. באותו זמן ערך של 10,000 ביטקון היה בערך \$41. כיום, כעבור שנה ושלושה חודשים, סכום דומה של ביטקון שווה בערך \$100,000. שעור קטן על חסכו.

עם התפתחות מערכת ביטקון, החלו לנצח בורסות (Bitcoin markets) המאפשרות לophobic בביטקון תמורה מטבעות אחרים. למשל, אחד מהשווקים הבולטים הוא xMtgox, וניתן לראות להלן את התנדות בשער מטבע הביטקון ביחס לדולר האמריקאי לאורך ששת החודשים האחרונים:



(שער הביטקון ביחס לדולר האמריקאי. התרשים נלקח מ- <http://bitcoincharts.com/charts>)

מערכת ביטקון הייתה על רכابتם בתקופה זו, במיוחד לאור החשיפה הגוברת שקיבלה מאז מאי יוני השנה. בשלב מסוים מטבע הביטקון נסחר תמורה כ-\$30, אלף מז רמתו ירדה לאזור ה-\$10, עדין הרבה מעל ערכו רק לפני חצי שנה. קיימים גם שווקי מסחר נוספים לביטקון, וניתן לעקוב אחריהם באתר <http://bitcoincharts.com/markets>. אתרים נוספים בהם ניתן לעקוב אחר קורותיו של הביטקון הם <http://bitcoinwatch.com> ו- <http://www.bitcoinmonitor.com>. נכון לזמן כתיבת שורות אלה, קיימים כ- 7 מיליון מטבעות ביטקון, שווים הכול מוערך בכ-68 מיליון דולר אמריקאי.

## עתידו של BitCoin

לאחר דהירה בערכו של הביטקון במהלך החודשים מאי ויוני, נראה כי המגמה התהפקה. בעקבות צניחה נוספת בערכו של הביטקון בתחילת אוגוסט, כתוב של הפורבס [מיהר להספיד](#) את המطبع, אם כי הירידות נבלמו מכך. כך או אחרת, עתידו של המطبع הוא נשא לספקולציות.

מצד אחד, מערכת ביטקון מתמודדת בהצלחה מרשימה עם האתגרים שדורשת מערכת פיננסית בהיקף גדול, ובפרט יכולת להבטיח את מהימנות העסקאות שבוצעות ומונעת שימוש כפוף במטבעות. בנוסף, ההבטחה לעמלות עסקה נמוכות הופכת את ביטקון למערכת מבטיחה מאוד לאור התגברות הצורך בדרך זולה לביצוע עסקאות קטנות ומיקרו-תשומות באינטרנט. הצדדים הטכנולוגיים והкриpto-וגראפיים של המערכת הופכים אותה לחרבת הגיקים; האנונימיות שנותנת יכולת להתחבא מאחורי מפתח פומבי חסר זהות קורצת לארגונים הפעילים מחוץ לחוק ולאקטיביסטים. שוק הביטקון עדין צער יחסית, יוכל לספק הזדמנויות מעניינות ליזמים. למשל, השוק עדין לא למגרי משוכל, וניתן לנצל [פער שעירים בין שוקי מסחר שונים לארביטרז](#).

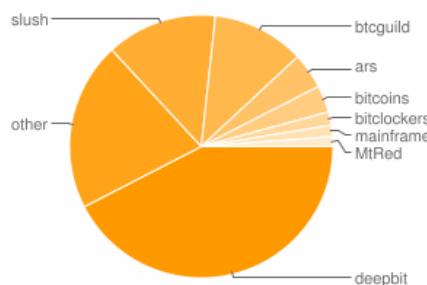
מצד שני, היתרונות של ביטקון הם גם החסרונות שלו. גופי אכיפת חוק בוחנים את המערכת בבדיקות כ芒גןון עשוי לשמש להלבנת כספים, ואין להוציא מכלל אפשרות שהשימוש במערכת יצא מחוץ לחוק. למשל, ארגון (EFF) Electronic Frontier Foundation, קבוצה העוסקת בשימור זכויות דיגיטליות בעולם הדיגיטלי, [אימצה בתחילתzeit ביטקון](#) ואיפשרה תרומות ביטקון לארגון. אולם עם התגברות העניין בביטקון, הארגון [נסוג מהמחלטה](#), בין השאר בשל חוסר הביריות סביבה השלכות החוקיות הכרוכות בהקמת מערכת מطبع חדשה, ורצונו של הארגון להתרחק ממעורבות מצד ישיר בעימותי אפרטי עתידי סביבה המערכת.

מכשול נוסף העומד בדרכה של המערכת הוא רמת הידע והכישורים הגבוהה הנדרשת לשימוש בטוח במערכת. בעלות על מטבעות ביטקון נשענת כולה על אבטחת המפתחות הפרטיים המתאיםים. אובדן של מפתחות אלה (למשל בעקבות קriseת כונן קשיח) או גניבתם על-ידי קוד זדוני, משמעותם אובדן בלתי הפיך של הכספיים. למשל, ב-13 ביוני משתמש בשם [allinvain allinvain](#) [דיוח על-can](#) שהמחשב שלו נפרץ ונגנבו ממנו 25,000 מטבעות ביטקון (שווי-ערך ל-\$375,000 בזמן הganiba). על-פי [allinvain](#), הפורץ הצליח להציג את המפתח הפרטיא שלו, והשתמש בו כדי להעביר אליו את הבעלות על המטבעות. העסקה, כמובן, בלתי הפיכה, ואין אמצעי פשוט למציאת הפשע על-פי המפתח הפומבי בו השתמש כדי לעסוקה. מקרה זה מחייב כי מערכת ביטקון, לפחות בצורתה הנוכחיית, לא מתאימה לתפוצה מעבר לאוכלוסייה הגדילים, שהיא גם מתחום מספיק כדי להגן מפני עלי קובץ ה"ארנק" בו מאוחסנים המפתחות הפרטיאים שלהם; האדם מהרחוב יהווה טרף קל לפושעים מזווינים. בנקים וחברות אשראי מספקים בטחון והגנה

מן הונאות, ורכיב של בטיחות אלה עברו רוב האוכלוסייה משמעותי ומהותי יותר מהיתרונות שמציעה ביטקוין.

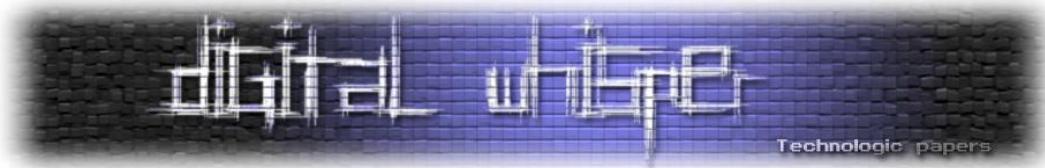
## סיכום

המגמות של החודשים האחרונים במערכות ביטקוין מעולות תהיה נוספת לגבי עתידה של מערכת כזו. התמראץ הכלכלי הביא למיזוג של משבבי מחשב, וכןן להיום מספר מצומצם יחסית של גופים שלוט ברוב המכריע של משבבי החישוב במערכות. אם מערכת ביטקוין תצליח לשרוד לאורך זמן ולהציג תמייה מספיק רחבה כדי לשמר על ערכו של המטבע, גופים אלה יהיו בעמדה לקבוע כרצונם תעריפים לביצוע עסקאות. באופן מעשי, הביטקוין עשוי להפוך ממטבע מתוחזק על-ידי קהילת המשתמשים בו, למטבע המנוהל על-ידי מספר מצומצם של גופים פרטיים. באופן אירוני, יתרון כי דוקא הצלחת המערכת תביא לעלייה הדרגתית בעמלות הנדרשות (ביכולתם של מייצרי הבלוקים להעתלם מעסקאות המציעות עמלות נמוכות), ובסיומו של דבר תחתור נגד אחד הצרכים שהביאו לייצור המערכת מלכתחילה.



(חלוקת משבבי החישוב ברשת ביטקוין נכון ל-11.8.26 - נלקח מתוך <http://bitcoinwatch.com>)

בין אם מטבע הביטקוין ימשיך ויתפוא תאוצה, ובין אם העניין בו יתמוסס והוא יגוע, מערכת הביטקוין פרצה דרך חדשה, וממחישה את הצורך והעוני בחלופות לשיטות המסחר המסורתיות. קרוב לוודאי שיש עוד דרך מרתקת בפניה של טכנולוגיה זו.



## מקורות ומידע נוספת

המאמר מבוסס בעיקרו על המידע באתר של ביטקוין (<http://www.bitcoin.org>) ובעיקר [עמוד השאלה הנפוצות](#), ועל המאמר של סטושי נקמוטו, [Bitcoin: A Peer-to-Peer Electronic Cash System](#). אין לראות בשום מידע המובא כאן בוגדר המלצה להשתתף או להימנע מהשתתף במערכת ביטקוין. בעלי ביטקוין שנהנו מקריאת המאמר, יכולים להביע את הערכתם באמצעות תרומה בביטקוין לכתבות [©MAJzaZP3sGJQpeGUrzWmqLg8tLiNtZ1Px1](#).

למאזיני פודקאסטים, להלן מספר המלצות חממות נוספות:

1. [פרק 98](#) בפודקאסט [עשיהם היסטוריה](#) של רן לוי (ברכות לרגל הפרק ה-100!!!) עוסק בהיסטוריה של האינפלציה, ודן גם בביטקוין.
2. [פרק 287](#) בפודקאסט [Security Now](#) של סטיב גיבסון מבית Twinkie, כולל תיאור נרחב של ביטקוין וכי怎 המ מערכת פועלת.
3. [פרק 423](#) בפודקאסט [This American Life](#) עוסק בהמצאת הכסף, וממחיש את המסקנה שהכסף הוא אשליה.

## על המחבר

ד"ר אריק פרידמן הוא חוקר בתחום של פרטיות ובטחת מידע, ובעיקר שיובם במסגרת אלגוריתמים ללמידה ממוחשבת וкриית נתונים. אריק סיים את לימודי הדוקטורט בפקולטה למדעי המחשב בטכניון בשנת 2011, והוא מחזיק גם בתואר MBA מאוניברסיטת תל-אביב. עד לאחרונה אריק שילב את המחקר עם תפקיד של Program Manager במיקרוסופט, שם עבד על מוצרים בתחום בטחת המידע והפרטיות.

## תזמן חוטים ב-Windows

מאת: משה גולדשטיין

### הקדמה

בפעם הקודמת הסתכלנו על המימוש של מנגנוני סנכרון ב-Windows. הצורך במנגנוני סנכרון עולה משימוש שמערכת הפעלה עשויה במספר מעבדים, ומהרצת מספר חוטים (Threads) על מעבד אחד או כמה מעבדים.

במאמר זה נסקרו כיצד Windows מחליטה איזה חוט יש להריץ, על איזה מעבד יש להריץ אותו, כיצד ניתן להשפיע על החלטות אלה, ואילו אופטימיזציות ושיקולים משפיעים על ההחלטה כתוצאה משינוי החומרה של השנים האחרונות.

### מדוע בכלל להשתמש בחוטים?

ambil' להיכנס לפרטים שחרוגים מגבולות המאמר, ניתן לומר שיש שלוש סיבות עיקריות לשימוש בחוטים הרלוונטיות הן בכתיבת תוכניות משתמש והן בפיתוח מערכת הפעלה עצמה:

1. **ניתול מספר מעבדים** - חוט אחד יכול לרווח על מעבד אחד בלבד בכל רגע נתון, ולכן על מנת להשתמש בצורה מיטבית במספר גדול של מעבדים יש ליצור מספר גדול של חוטים ולדואג לספק להם עבודה בכל רגע. (כמובן, מטרתנו ברכישת מחשב עם מספר רב של מעבדים היא להביא אותם לניצול של 100%, ולא להניח להם להתבטל).

2. **שילוב של פעולות קלט/פלט ופעולות חישוב** - בזמן שרכיב חומרה מסוימת (למשל, דיסק קשיח) מטפל בבקשת קלט/פלט, המעבד פניו יכול להמשיך לבצע חישובים. תוכנית שבוצעת הן קלט/פלט והן חישובים יכולה להשתמש במספר חוטים. למשל, בעת שחווט מכהה לתוצאות של פעולות קלט/פלט, חוט אחר יכול לבצע חישובים ולהמשיך לנצל בצורה מיטבית את המשאבים.

3. **תגובהיות למשתמש** - שימוש במספר חוטים (אפילו במערכת בעלת מעבד אחד) מאפשר לתוכנית משתמש להישאר תגובתי, ע"י כך שתות הטיפול בפעולות המשתמש מבצעים בחוט אחד ואת הפעולות הדורשות עיבוד כבד מבצעים "ברקע", בחוט אחר.

ובן שהשימוש בחוטים רבים הכרוך גם בבעיות רבותות, שאת מקצתן ראיינו במאמר הקודם - בעיות כגון **טבק** (deadlock), גישה לא מסונכנת לזכרון משותף, הפרות סדר פעולות הגישה לזכרון ועוד. בכלל זאת, היתרונות עולים על החסרונות, וכמעט כל תוכנית משתמש היום במספר חוטים - לעיתים באמצעות תשתיות תוכנה מוחכמת המנהלות את יצירת החוטים והשנת עבודה לחוטים בעצמן.

### כיצד מחליפים בין חוטים?

Windows מחליף בין חוטים לעיתים קרובות מאוד. כך נשמרת האשלה, אפילו במחשב בעל מעבד אחד, כדי כמה תוכניות יכולות לרוץ במקביל. בהמשך נסקרו את הזמן שנitin לכל חוט כאשר הוא מקבל את ההזדמנויות לרוץ, אבל קודם יש לתת את הדעת לסוגיה הבאה: בכל פעם שמערכת הפעלה בוחרת להחליף בין חוטים, עליה "להסיר" מן המעבד את החוט שמתבצע כתע, ו"להשים" על המעבד את החוט החדש. מה המשמעות של "להסיר" ו"להשים"?

מערכת הפעלה מתחזקת מבנה נתונים הנקרא Context, שמתאר חוט שמתבצע כרגע על המעבד. זה- Context מכיל את כל האוגרים (Registers) של המעבד. למשל, במעבד x86, ה-EIP המכיל את האוגר EIP המתאר את הפקודה הבאה שהמעבד יבצע, ואת האוגר ESP המתאר את ראש המחסנית. להלן

הагדרה של מבנה הנתונים CONTEXT עבור מעבדים משפחתיים<sup>1</sup>:

```
typedef struct _CONTEXT {
    DWORD ContextFlags;
    DWORD Dr0;
    DWORD Dr1;
    DWORD Dr2;
    DWORD Dr3;
    DWORD Dr6;
    DWORD Dr7;
    FLOATING_SAVE_AREA FloatSave;
    DWORD SegGs;
    DWORD SegFs;
    DWORD SegEs;
    DWORD SegDs;
```

<sup>1</sup> מבנה הנתונים CONTEXT עבור חוטים שאין בהם פעילים כרגע נשמר על המחסנית של החוט. ניתן גם להשפיע על תוכנו ישירות באמצעות הפונקציות SetThreadContext ו-GetThreadContext.

```
DWORD    Edi;
DWORD    Esi;
DWORD    Ebx;
DWORD    Edx;
DWORD    ECX;
DWORD    Eax;
DWORD    Ebp;
DWORD    Eip;
DWORD    SegCs;
DWORD    EFlags;
DWORD    Esp;
DWORD    SegSs;
BYTE     ExtendedRegisters[MAXIMUM_SUPPORTED_EXTENSION];
} CONTEXT;
```

(מבנה הנתונים CONTEXT עבור מעבדי x86, לקו מຕוך.h WinNT.h)

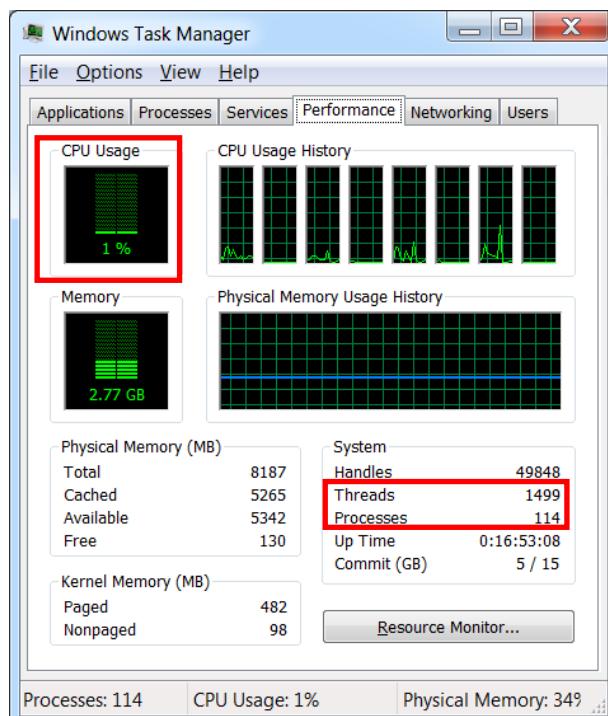
לכן, הchèלה בין חוטים משמשותה שמיירת האוגרים מהמעבד בתוך ה-Context של החוט המתבצע כרגע, ואחזרו האוגרים של המעבד מתוך ה-Context של החוט החדש. זה מספק בדיקת כדי להחליף בין החוטים - למשל, החלפת האוגר EIP תגרום לכך שהחוט יירץ קוד שונה מהחוט הקודם, והחלפת האוגר ESP תגרום לכך שפעולות של החוט החדש על המחסנית לא ישפיעו על המחסנית של החוט הקודם.

### ההבטחה הגדולה: החוט האחד בעל העדיפות הגבוהה ביותר שMOVED ליריצה יdroz תמיד

סכמת התזמון של מערכת הפעלה מבוססת על רעיון אחד: בכל רגע נתון בוחרים מבין החוטים המוכנים ליריצה את החוט בעל העדיפות הגבוהה ביותר, ומריצים אותו. אין זה משנה האם החוט הזה רץ כבר בעבר או שהוא נוצר ברגע זה, והאם העדיפות שלו השתנתה או מאז ומעולם הייתה הגבוהה ביותר. ההחלטה מבוססת אך ורק על המצב הנוכחי, מה שהופך את המימוש של המתזמן (Scheduler) לפשטוטית.<sup>2</sup>.

יש לציין שבמערכות טיפוסית, ישנו מאות חוטים בכל רגע, אפילו אם המשמש לא הפעיל עדין אף תוכנית מסוימת. אם כל החוטים האלה היו מוכנים ליריצה, לא ניתן היה לעבוד בצורה סבירה על המערכת. מכאן, רוב החוטים אינם מוכנים ליריצה בכל רגע נתון (כי הם ישנים, ממתינים על מנגןון סנכרון, ממתינים לתוצאות של פעולה קלט/פלט, ועוד) - מה שהופך את מלאכתו של המתזמן פשוטה עוד יותר, שכן הבחירה היא בין מספר מצומצם יחסית של חוטים.

<sup>2</sup> יש לציין שרעיון "פשוט" זה עובד היטב במערכת בעלת מעבד אחד. ישנה הרחבה מתבקשת למספר מעבדים, אבל נראה בהמשך שמסיבות של ביצועים וסקלאbilיות לא משתמשים בה.



(תצלום מסך מערכת בעלת 8 מעבדים, שעלה כרגע 1499 חוטים. בכל זאת, צריכת המעבד היא 1% - מרבית החוטים אם לא כולם אינם מוכנים ליריצה (כרען))

למעשה, בעיקר במערכות קצה, במשר מרבית הזמן אין אפילו חוט אחד המוכן ליריצה - ובמקרה זה המתזמן מרייצ' קוד המכונה Loop Idla, שבין היתר יכול להוריד את צריכת החשמל של המערכת או אפילו לכבות מעבדים שאינם בשימוש.

מطبع הדברים ישנן גם מערכות שבהן עשרות חוטים המוכנים ליריצה בכל רגע נתון. יכול להיות אף אילו שיש חוט אחד או שניים שככל הזמן מוכנים ליריצה, לאחר שכל העבודה היא חישובית. במקרה זה, יתכן שההרעין של הריצת החוט החשוב ביותר נראה קצת לא הוגן - יתכן שקבוצה קטנה של חוטים מסגלאת עצמה את כל משאבי החישוב של המערכת, ולא מאפשרת לחוטים אחרים לרוץ בכלל. מסיבה זו, בין היתר, Windows מממשת מגנון של שינוי עדיפות דינמיים, כפי שנראה בהמשך.

## סיכום העדיפויות של Windows

עדיפות של חוט ב-Windows מורכבת משני חלקים: **עדיפות הבסיס** (Process Priority Class) של התהיליך שלו, וה**עדיפות היחסית** (Relative Priority, Thread Priority) של החוט. כל אחד מהחלקים קבוע בנפרד באמצעות פונקציות שונות - עבור עדיפות של תהיליך, ו- SetThreadPriority עבור עדיפות יחסית של חוט. אולם בסופו של דבר, המערכת לא מתעניינת בעדיפות היחסית של החוט או בעדיפות הבסיס של התהיליך, אלא רק בעדיפות האבסולוטית של החוט - מספר בין 0 ל-31. העדיפות 0 היא העדיפות הנמוכה ביותר, והעדיפות 31 היא העדיפות הגבוהה ביותר.

הטבלה הבאה מסכמת כיצד עדיפות הבסיס של התהיליך והעדיפות היחסית של החוט משלבות יחד ליצור העדיפות האבסולוטית:

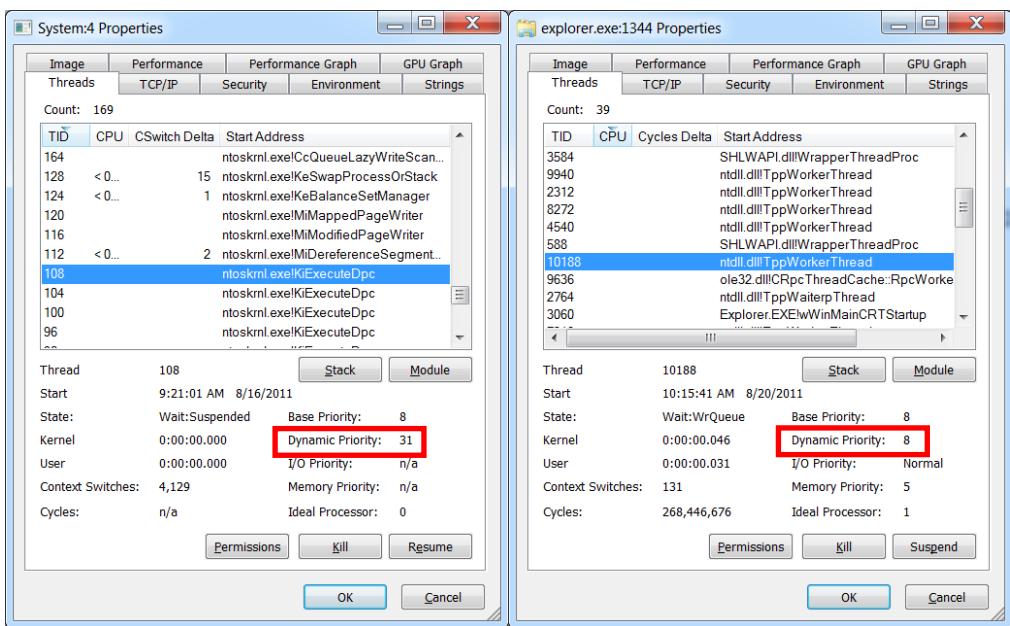
Process Priority Class	Thread Priority Level	Absolute Priority
Idle	Idle	1
	Lowest	2
	Below Normal	3
	Normal	4
	Above Normal	5
	Highest	6
	Time Critical	15
Below Normal	Idle	1
	Lowest	4
	Below Normal	5
	Normal	6
	Above Normal	7
	Highest	8
	Time Critical	15
Normal	Idle	1
	Lowest	6
	Below Normal	7
	Normal	8
	Above Normal	9
	Highest	10
	Time Critical	15

Above Normal	Idle	1
	Lowest	8
	Below Normal	9
	Normal	10
	Above Normal	11
	Highest	12
	Time Critical	15
High	Idle	1
	Lowest	11
	Below Normal	12
	Normal	13
	Above Normal	14
	Highest	15
	Time Critical	15
Realtime	Idle	16
	Lowest	22
	Below Normal	23
	Normal	24
	Above Normal	25
	Highest	26
	Time Critical	31

(רשימת העדיפויות האבסולוטיות האפשריות כתלות בעדיפות היחסית של החוט ועדיפות הבסיס של התהיליך)

יש לציין שקביעת עדיפות הבסיס Realtime דורשת מהמשתמש הרשות מיוחדת, שלרוב נתונות רק למנהל המערכת (Administrator). שאר העדיפויות זמינים לכל התוכניות, אולם רוב התוכניות אין טורחות לשנותן את עדיפות הבסיס של התהיליך והן את העדיפויות היחסית של החוטים. האם זה כדאי? על פניו נראה שכדי לשנות את רמת העדיפות בהתאם למשימה, או להעלות את העדיפות של התוכנית שלנו על חשבון תוכניות של משתמשים אחרים, או של יצרני תוכנה אחרים. אבל אם כל התוכניות היו משנות את העדיפות שלהן, למשל לעדיפות High, כי אז רמת העדיפות הزادה הייתה הופכת למעשה לרמת ברירת המחדל.

בailo רמות עדיפות משתמשת מערכת ההפעלה עבור החוטים שלה? ובכן, התשובה תלוי בהגדרה של "מערכת ההפעלה". למשל, החוטים של התהיליך Explorer.exe רצים בעדיפויות רגילות, הגם שעבור משתמשים רבים הוא מערכת ההפעלה, כיוון שהוא מציג את שולחן העבודה ואת שורת המשימות. לעומת זאת, ישנו חוטים של המערכת שרצים בעדיפויות גבוהות מאוד, למשל חוטים של מנהל הזיכרון הווירטואלי.



(שני חוטים לדוגמה מתוך exe והתהילך System. העדיפויות הן 8 ו-31 בהתאם)

מה ההבדל בין Base Priority ל-Dynamic Priority? Windows מבצעת שינויים דינמיים של העדיפות לפי פרמטרים שונים. בתצלומי המסך מוצגת העדיפות של החוט כפי שנקבעה לעומת העדיפות של החוטים כפויים להפעלה.

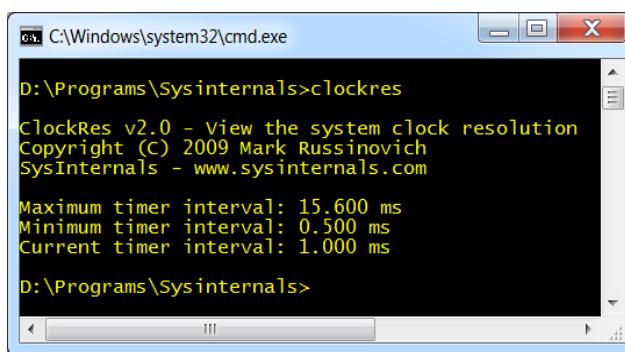
אם אתם מכירים את המושג IRQ (רמת עדיפות של פסיקות), אין קשר בין עדיפויות של פסיקות לבין עדיפויות של חוטים. למעשה, ניתן לומר שכל פסיקה היא יותר חשובה מאשר כל חוט - ולראיה, אפילו אם כרגע במערכת מתבצע חוט בעדיפות 31, ומתרחשת פסיקה, המערכת תפסיק את ביצוע החוט ובכצע את הפסיקה. מצד שני, במהלך טיפול בפסיקה לא מתרחשת (בדרך כלל) החלפת Context של חוטים, כך שני המושגים - עדיפות של פסיקות ועדיפות של חוטים - הם אורתוגונליים לחלוטין.

### יחידת הזמן הדינית לחוט (Quantum)

לפעמים המתזמן של המערכת מופעל כאשר מתרחשים שינויים בחוטים - למשל, העדיפות של חוט מסוים משתנה, חוט נכנס להמתנה, חוט "וחזר" מהמתנה לתוצאות של פעולה קלט/פלט. לעומת זאת, לעיתים קרובות לא מתרחשים שינויים בקבוצת החוטים שמוכנה לריצה. במקרה זה, דרוש מגנון חיצוני שיגרום למתזמן לזרוץ כדי שלא לאפשר לחוט אחד (שאלוי תקע בלולאה אינסופית) להשתלט על כל משאבי החישוב של המערכת.

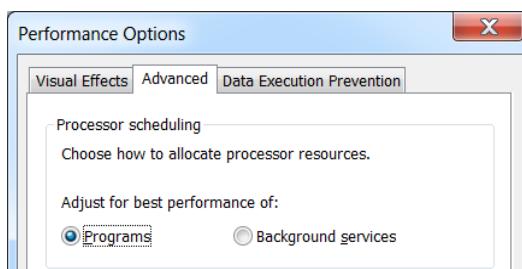
כאשר חוט נבחר לריצה והמערכת מאפשרת לו לrhoץ על מעבד מסוים, הוא מוגבל בזמן. זמן של החוט קצוב אפילו אם החוט מוכן להמשיך לרוץ לנצח ולא מתרחשים אירועים מעוניינים אחרים במערכת הגורמים לתזמון. לאחר שזמן זה פג, המתזמן מקבל מחדש את ההחלטה על החוט שירוץ ביחידת הזמן הבאה.

משך הזמן שניtin לחוט (הנקרא Quantum) קשור בשני גורמים. הגורם הראשון הוא קצב השעון המחבר למערכת (אין מדובר על השעון של המעבד, אלא על רכיב חומרה חיצוני היושב על לוח האם). השעון מייצר פסיקות במרווחי זמן קבועים, והמתזמן יכול להשתמש בפוסיקות האלה כדי להחליט متى פגה יחידת הזמן שהוקצתה לחוט מסוים. קצב השעון נמדד בדרך כלל במילישניות - 15 מילישניות הן ערך נפוץ.<sup>3</sup>



(בדיקות קצב השעון של המערכת באמצעות הכלי Clockres.exe של Sysinternals)

הגורם השני המשפיע על ה-Quantum הוא הגדרה שנמצאת בידי המשתמש. Quantum קצר גורם לכך שהחלפה בין חוטים מתרחשת לעיתים קרובות יותר, מה שיכל לשפר את התגובהות של המערכת - כל חוט מקבל הזדמנויות לרוץ לעיתים קרובות יותר, מה שמתאים לעיתים קרובות לתוכנות המבצעות אינטראקציה עם המשתמש. לעומת זאת, Quantum ארוך מגדיל את הסיכוי שחוט יסיים את עבודתו (או לפחות חלק ניכר ממנו) לפני שייקשו ממנו לפנות את המעבד, מה שמתאים לעיתים קרובות לתוכנות המבצעות עבודה ברקע. לא תמיד ברור מה האפשרות העדיפה, ולכן ניתן לשנות עליה באמצעות ה-Registry או הגדרות "המחשב שלי":



(הגדרה המשפיעה על ה-Quantum. האפשרות המשומנת משמעותה Quantum קצר יותר)

<sup>3</sup> בהרבה מקרים, יש אפשרות להשפיע על קצב השעון. למשל, בתצלום המפק להלן, קצב השעון הנוכחי הוא מילישניה אחת, אולם הקצב המקסימלי הוא 15.6 מילישניות והקצב המינימלי הוא 0.5 מילישניות (500 מיקרוישניות). שינוי קצב השעון בזמן ריצה גורם לכך שמנגנוני זמן כמו Timers יכולים להיות מדויקים יותר, אבל הוא לא להשפיע על תדריות הריצה של מתזמן החוטים, שמסתמך על נתון ה"מקסימום" בתצלום המפק לעיל.

כיצד ההגדרות הנ"ל מתרגמות ליחידות זמן? הבחירה ב-“Programs” גורמת ל-Quantum להיות באורך כפול מתדיות השעון, למשל 31.2 מילישניות אם תדיות השעון היא 15.6 מילישניות. לעומת זאת, הבחירה ב-“Background services” גורמת ל-Quantum להיות כפולת-12 של תדיות השעון, למשל 187.2 מילישניות אם תדיות השעון היא 15.6 מילישניות.

גם כאן אין מנוסה מדקויות: ראשית, לא כל חוט יכול לבדוק את משך הזמן שהוקצב לו מלכתחילה, שכן במהלך ריצתו יכול להיווצר (או להתעורר) לפטע חוט חדש, חשוב יותר, והמתזמן יחליף אותו מיד. שנית, יתכן שבסוף ה-Quantum, המתזמן יחליט להמשיך להריץ את החוט, לאחר שהוא עדין החוט החשוב ביותר המוכן לריצה. שלישי, יתכן שבמהלך ריצת החוט מתרחשות פסיקות רבות הגוזלות משך הביצוע של החוט, ובמקרה כזה מערכת הפעלה תתחשב בכך ותאריך את משך הזמן המוקצב לו.<sup>4</sup>.

### שינויי עדיפויות דינמיים

מהד, השימוש בעדיפויות מאפשר לתוכניות לטעוף משימות חשובות יותר ופחות, ולמערכת הפעלה להחליט בקלות איזה חוט להריץ בכל רגע. מאידך, שימוש עיור בעדיפויות מוביל לביעות של אי-הוגנות, כאשר הנפוצות ביותר מביניהן הן הרעה והיפורعدיפויות.

הרעה היא מצב שבו חוט לא מקבל אפשרות לרוץ מכיוון שכל זמן שהוא מוכן לריצה, יש חוט חשוב יותר שМОוכן גם הוא לריצה. היפור בעדיפויות הוא מצב עדין יותר, שבו משתתפים לפחות שלושה חוטים. נניח ש-C, B, A הם חוטים בעלי עדיפויות 3, 2, 1 בהתאם. נניח גם שהחוטים C, A, B משתפים ביניהם משאב, ולכן משתמשים במנגנון סנכרון M למניעה הדנית. כתע “יתכן המצב הבא: החוט A מתחילה לרוץ, ותוסס את המשתתפים במנגנון סנכרון M למניעה הדנית. כתע “יתכן המצב הבא: החוט A מתחילה לרוץ, ותוסס את ה-”מנעל” M. מיד לאחר מכן החוט C מתעורר, מה שגורם למતזמן החוטים לסלק את A מהמעבד ולהחליפו ב-C. החוט C זוקק גם הוא ל-M, וכך הוא נכנס למצב המתנה. כתע מתעורר החוט B ומתחילה לרוץ - הוא חשוב יותר מ-A, ואם כתע הוא מוכן לריצה במשך זמן רב, החוט A מורעב. אבל לא זו בלבד ש-A מורעב ולא מקבל הזדמנות לרוץ, אלא שבגלל M גם C לא מקבל הזדמנות לרוץ (כיון שהוא זוקק ל-M ש-A מחזיק). נוצר היפור בעדיפויות - חוט בעדיפות 2 רץ בעוד שיש חוט בעדיפות 3 שהוא יכול לרוץ, אבל אינו מוכן לריצה באותו רגע.

על מנת לטפל במצבים מעין אלה Windows מבצעת שינוי עדיפות דינמיים של חוטים במהלך ריצתם. רק חוטים בעלי עדיפויות 1-15 ננים שינוי עדיפות דינמיים אלה, וכל שינוי העדיפות הם זמינים,

<sup>4</sup> התהשבות כזו קיימת רק בגרסת 6.0 NT ומעלה, ככלומר החל מ-Windows Vista.

-זמן חוטים בסוウס-Windows

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

ובדרך כלל דועכים ברמת עדיפות אחת מד' Quantum עד חזרתם לrama המקורית. כמו כן, כל השינויים הם כלליים אצבע - כלומר, המערכת לא יכולה לדעת בוודאות שינוי העדיפות ישר את התగובתיות, ההווגנות, או פרמטרים חיבויים אחרים. התקווה היא שב מרבית המקרים, שינוי העדיפות לא מזיקי, ולפעמים יכולים גם להועיל.

קצתה הירעה מכדי למן את כל שינוי העדיפות הדינמיים ש-Windows מבצעת, ولكن נתבונן רק בשלושה מהם:

הסבר	גודל השינוי	מהות השינוי
התיחסות לחוט SMB3 המנתנה בתוך חוט ה"מווטר" בחביבות על-h-Quantum שלו	1 או 2 רמות עדיפות	העלאת עדיפות לאחר חרזה מהמתנה על אובייקט סנכרון
לאפשר לחוט שჩיכה לסיום פעולה קלט/פלט לכבר הוא קורא לפונקציה IoCompleteRequest	כ整洁 במנעל ההתקן (Driver)	העלאת עדיפות לאחר סיום פעולה קלט/פלט
התמודדות עם הרעבה והיפוך עדיפות	העלאה לרמת עדיפות 15 וריצה במשר שני Quantum-ים	העלאת עדיפות והארכת המעבד Quantum לחוט שלא רץ במשך מספר שניות, ומוקן לריצה

(שלוש דוגמאות לשינוי עדיפות דינמיים ש-Windows מבצעת במקרים שונים)

### בחירת מעבד פניו להרצת חוט ובחרת חוט להרצאה על מעבד פניו

כאשר מערכת הפעלה צריכה לבחור מעבד להריץ עליו חוט מסוים (למשל, חוט שנוצר כרגע או שהתעורר מהמתנה), היא צריכה להתחשב לא רק בעדיפות של החוט, אלא גם באיכות היחסיות של המעבד ביחס לחוט. למשל, המערכת צריכה להעדיף את המעבד שעליו החוט רץ לאחרונה, שהרי שם יש עוד סיכוי למצאו במתמון את המידע שהחוט יctrar בריצה העתידית. אולם גם החוט יכול לקבוע את העדרותיו - באמצעות פונקציות כגון SetThreadAffinity חוטים יכולים לקבוע מעבד או קבוצת מעבדים שרק עליהם יוכל לרוץ<sup>5</sup>. באופן דומה, כאשר מתפנה מעבד (למשל, כיוון שחוט מסוים או נכנס למתנה), יש לבחור חוט חדש להריץ עליו במקום החוט הקודם ממנו, מערכת הפעלה תעדיף חוטים שרצו על המעבד לאחרונה.

<sup>5</sup> מבלי להזכיר פרטים שאין מקום עבורה במאמר זה, ישנה בעיות גדולה בקביעת העדרות מעבדים בצוותה נוקשה. למשל, ניתן מצבים שבהם קביעת העדרה כוותת גורם לחוט מסוים להיות מושעב, בעוד שהמעבדים המועדפים עליו תופסים כל העת על ידי חוטים חשובים אפילו יותר. הדבר נכון עוד יותר כאשר התוכנית צריכה לרוץ על מחשבים בעלי מספר שונה של מעבדים ובעל ארQUITקטורה חומרה שונה – כי אז קשה מאוד לנחש מראש מהו תעדוף המעבדים המיטבי עבור התוכנית, ולעתים קרובות עדיף להשאיר את החלטות למערכת הפעלה.

## כיצד המערכת מתאימה את עצמה לשינויי חומרה

Windows נאלצת להתאים את עצמה לשינויי חומרה המתרחשים בשנים האחרונות. שינויים במעבדים ובמבנה הזיכרון גורמים למתzman החוטים לקבל החלטות בצורה מורכבת יותר. נביא שלוש דוגמאות מייצגות למצבים אלה.

זהה כעשור, חברות שונות ובראשן אינטל מיצירות מעבדים בעלי יכולת HyperThreading, כלומר סימולציה של מספר מעבדים לוגיים (בדרכן כל 2) על מעבד פיזי אחד. הדבר נעשה על ידי שכפול האוגרים בمعالד השומרם את מצב הביצוע, כגון EIP שראינו קודם, אך ללא שכפול משאבי החישוב, כמו היחידה האריתמטית-לוגית (ALU). כאשר המעבד אינו משתמש במשאבי החישוב וממתין - למשל כתוצאה מגישה לזיכרון - משתמשים במשאבי החישוב להרצה של חוט אחר. שיפור הביצועים המתאפשר בעומת מעבד לוגי אחד על מעבד פיזי אחד הוא בכך כל קטן יחסית, ואפיו בתחום האופטימיות ביותר מדבר על שיפור של 30% בלבד.

מערכת הפעלה צריכה להתחשב ב-HyperThreading במהלך תזמון חוטים. למשל, כיוון שהמת�ונן משופר במידה לכל המעבדים הלוגיים על אותו מעבד פיזי, אין יתרון להעדפת מעבד לוגי מסויים כיוון שהחוט רץ עליו לאחרונה. לעומת זאת, כאשר בוחרים חוט עבור מעבד לוגי פנוי, יש להסתכל גם על חוטים שרצו לאחרונה על מעבדים לוגיים אחרים על אותו המעבד הפיזי. באופן דומה, המערכת תעדייף לשבץ חוט על מעבד לוגי שהמעבד הפיזי שלו פנוי למגררי על פני מעבד לוגי שהמעבד הפיזי שלו כבר מריץ חוט אחר (בעוד לוגי אחר).

דוגמה אחרת לשינויי ארכיטקטורה שמקטיבים שינויים במתzman החוטים היא NUMA, ארכיטקטורת חומרה שבה המעבדים והזיכרון מסודרים בקבוצות הנקראות Nodes. בכל קבוצה נמצאים חלק מהמעבדים וחלק מהזיכרון. למשל, מערכת בעלת 8 מעבדים ו-24 ג'יגה-בתים של זיכרון, יכולות להיות ארבע קבוצות, כאשר בכל קבוצה 2 מעבדים ו-6 ג'יגה-בתים של זיכרון. הסיבה לחלוקת היא הנדסית בעיקר: ככל שימושים יותר מעבדים יותר זיכרון למערכת, כך קשה יותר להחזיק את כל המעבדים במרחב זהה (וקטן) מהזיכרון. מכאן שמעבד הניגש לזכרון בתוך הקבוצה שלו עשויה זאת במהירות מירבית, ואילו גישה לזכרון בקבוצה אחרת היא איטית בהרבה (לעתים פי שניים או פי שלושה).

כמו כן, גם במקרה זה מערכת הפעלה מוכרכה להתחשב ב-NUMA כאשר מתzman החוטים עושה את עבודתו. למשל, המתzman י יצא מגדרו על מנת ש热点 לא ינדוד בין מעבדים הנמצאים בקבוצות שונות. כמו

כן, המערכת צריכה לדאוג שהקצאות זיכרון המבוצעות על ידי חוט מסווקות מהזיכרון הפיזי הנמצא בתוך הקבוצה של המעבד עליו הוא רץ<sup>6</sup>.

לבסוף, בשרתים מודרניים ישנה היכולת לכבות חלק מהמעבדים לחולוטן בזמן שלא נעשה בהם שימוש. יכולת זו נקראת Core Parking והוא מאפשר חיסכון משמעותי באנרגיה. גם כן, מערכת הפעלה נדרשת להתחשב במעבדים כבויים - היא תשבץ חוטים קודם למעבדים פנויים שאינם כבויים, לפני שהיא מדילקה מעבד כבוי.

## סיכום

זמן חוטים הוא משימה מורכבת בהרבה מכפי שנראה أولית, במערכות בעלות מעבד אחד ומספר מצומצם של "טריקים" לשינוי עדיפות ומשך זמן עבור ריצת החוטים. בימינו, מערכות הפעלה השונות מתחרות הן במישור הביצועים והן במישור צדירת החישול, ו{return}היגיון המעריצי במספר המעבדים אינו מקל את עבודת מתזמן החוטים.

ימים יגידו האם Windows ומערכות הפעלה אחרות הבנוויות סביב מודל זמן החוטים הקלסטי שהוצע כאן יוכל להתמודד, כפי שהן, עם אלפי או עשרות אלפי מעבדים. בinternals, Windows מסוגלת לתזמן عشرות אלפי חוטים על 256 מעבדים בהצלחה - ובמורכבות - גודלה.

## על המחבר

סשה גולדשטיין הוא ה-CTO של [קבוצת סלע](#), חברת ייעוץ, הדריכה ומיקור חז' בינהומית עם מטה בישראל. סשה אוהב לנבור בקרביים של Windows ו-H-CLR, ומתמחה בניפוי שגיאות ומערכות בעלות ביצועים גבוהים. סשה הוא ממחבר הספר Windows 7 for Developers, ובין היתר מלמד במכלאת סלע קורסים בנושאים Windows Internals. בזמן הפנוי, סשה כותב [בלוג](#) על נושאי פיתוח שונים.



<sup>6</sup> התמיכה המפורשת של Windows ב-NUMA הוצאה בתקופת Windows Vista והשתפרה עוד ב-7 Windows.

-זמן חוטים בwindows  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

## איך ליעקוב אחורי גולשים בעזרת עוגיות חסינות מחיקה

מאת: שלמה יונה ([shlomo.yona@gmail.com](mailto:shlomo.yona@gmail.com) | <http://shlomoyona.blogspot.com>)

### הקדמה

אתרי אינטרנט יכולים לספק שירות טוב יותר ולהרוויח יותר באמצעות איסוף מידע על הגלישה בהם. מידע זה שימושי במיוחד כאשר הוא מעובד באמצעות נתונים שונים, שמאפשרים תצוגה חזותית נוחה של המידע, בוחינת השירות והסקת מסקנות. דוגמה לשירותים זהה הוא [Google Analytics](#). מתברר שישנו עוד מידע רב הרבה להבין יותר ולהעלות יותר את השירות ואת הרוחחים. מידע זה הוא מידע על אודות הגולשים גם מחוץ לאתר עצמו. אם יודעים מה גולשים עושים גם באתר וגם מחוץ לאתר, אפשר להסיק מסקנות נרחבות יותר, ולהשתמש במסקנות אלה כדי להציג הצעות שתהינה מושכות יותר לגולשים, ושתעלינה את ההסתברות לרכישה, וכך יגדלו את הסכום שבפועל מקבלים בממוצע מרכישות באתר.

היא חברת [KissMetrics](#) מסווג כזה שמוכרת מידע על גלישה לאתרם. את המידע היא אוספת באופןים שונים. בנוסף, החברה מוכרת גם כלים לניטוחו ולהסקת מסקנות מהמידע. הדרכים להשיג מידע על גולשים באתר ידועות בדרך כלל, וביניהן, מידע שנאסף מתוך בקשות [HTTP](#) (למשל, על ידי רישומי [logs](#) של שרת האינטרנט), מידע מסדי הנתונים וגם עוגיות בצד הלוקוט. העוגיות הן מנגןן מובנה ב-[HTTP](#), ודףנים תומכים בו. זאת שיטה לאפשר שבירת מידע בצד הלוקוט. זה אחד המנגנונים לשימרת מצב הלקוח מול אתר בעת גלישה. מנגןן לשימרת מצב הלקוח, מפני שפרוטוקול [HTTP](#) הוא פרוטוקול שאינו שומר מצב. אפשר גם לשמר את מצב הלקוח בצד השרת. ישנה נוחות בשימרת מצב בצד הלוקוט, משום שהמידע מגיע יחד עם הבקשות מהלוקוט, וכך ניתן בקשות אלו, ואין הכרח לאחסן אותן. יחד עם זאת, העוגיות מאפשרות מעקב אחרי הלקוח, שבירת ההיסטוריה שלו באתר (מה עשה, מתי עשה וכו'). לכאורה, מנגןן [same origin policy](#) אמרור למונע מעוגיות שמנוהלות בשם מתחם מסוים להיות בשימוש בשם מתחם אחר. את זה עוקפות חברות באמצעות חבורות צד ג' שיש להן קוד שМОבנה בדף באתר. הקוד מאפשר לשלוח מידע על גולש בשם מתחם מסוים לשם מתחם אחר. למשל, חברת פרסום שיש לה קוד באתרים רבים מקבלת מידע על גולשים, אותו היא שומרת בעוגיות תחת שם המתחם שלה, וכך יכולה לאסוף מידע על גולש בין אתרים שונים וגם בין שמות מתחם שונים.

איך ליעקוב אחורי גולשים בעזרת עוגיות חסינות מחיקה

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

בימינו גולשים רבים מודעים לכך, וכך מוחקים עוגיות. בנוסף, קיימים כלים נחדים שמאפשרים לדעת בקלות אילו עוגיות צד ג' מותקנות בכל עמוד שגולשים אליו, ולקבל מידע על החברות ועל השירותים שמאחוריו העוגיות, ואפילו להסום אותן. כלי אחד זה הוא [Ghostery](#). כל'i אחר, שמאפשר גם לראות גرافים בין חברות צד ג' שמרגלות אחרים, הוא [Collusion](#).

ישן יוזמת חדשת שעוסקות ב-[do no track opt](#) של משתמשים לפי רצונם, למשל, [HTTP header](#) שמנסם לתמוך בו באופן אחיד בדףנים כדי למנוע לחברות ומשרדי צד ג' לעקוב אחר הפעולות.

عقب המודעות אליהן, חברות שמתפרנסות ממעקב אחר משתמשים הבינו שעליהן להשתככל. אחד השכלולים של חברות צד ג' להמשיך ולעקוב אחרי משתמשים שמוחקים את קבצי העוגיות שלהם היה לאחסן גיבוי של העוגיות בתור עוגיות פלאש, [Flash Cookies](#), אשר עד לא מכבר (ובחלק מהדףנים גם היום) יש צורך בפעולה נפרדת כדי למחוק אותן. מה שקרה הוא שימוש שmagui לאתר ואין לו עוגיה מתאימה, זוכה לחיפוש עוגית הפלאש המתאימה, ואם המידע נמצא בעוגיית הפלאש הוא משוחזר לעוגייה. כאשר שומרים מחדש מידע בעוגיה אז מגבים אותה ליתר בטחון גם בעוגיית הפלאש. המחקר שחשף את השימוש בעוגיות פלאש לצורך שרידות העקבה זמין ב:

<http://ssrn.com/abstract=1446862>

דרכים נוספות להמשיך ולשמור מצב (ולאפשר מעקב...) בצד הלוקה הן באמצעות שיטות אחסון אחרות ומודרניות יותר בצד הלוקה, כמו למשל [HTML5 WebStorage](#) [localStorage](#) ו-[sessionStorage](#) שמאפשר באופן מוסדר (עדין לא כל הדףנים ממשיכים את זה במלואו) לאחסן בדף זוגות של מפתח-ערך, ואפילו מסד נתונים שלו. בגרסאות ישנות יותר של דףנים יש גם [userData](#) (בפיירפוקס) ו-[localStorage](#) ו-[sessionStorage](#) (באקספלורר). לאור עליית המודעות אצל הגולשים, גרסאות חדשות של דףנים מאפשרות בקליק יחיד למחוק, אם רוצים, את כל המידע שנאסף על המשתמש מכל התקנים הללו לפי דומיין, ולפי קרייטריונים נוספים.

ה מצב הזה נזהה מראש וחברות המערכת שרצו להתרננו לא הפסיקו לחפש דרכים נוספות כדי לגבות את המידע ולאפשר המשך המעקב.

**בסוף יולי 2011 פורסם מחקר:** [מחיקת עוגיות, מחיקת עוגיות פלאש, אפילו נטרול הפלאש לחלוטין ונטרול אמצעי אחסון בצד הלוקה לא יועילו](#) - גם לא ביטול השימוש ב-[JavaScript](#) [\(incognito\)](#) (בדףן).

## או איך זה עובד?

בפרוטוקול HTTP החל מגרסתה 1.1 יש מגנון שמאפשר חסוך בהעברת הودעות באמצעות מטמון, [cache](#). מגנון זה מכונה [ETag](#). הדריך המקובל לשימוש ב-ETag היא לעדכן בשדה הערך של כותרת ה-ETag בבקשת HTTP את גרסת המשאב (תוכן, תמונה וCORDOMA) כדי לאפשר למכוונות ולשירותים בדרך (לרבבות אפילו השירות של השירות שאלוי גולשים והדף של הגולש, בפרשנות רחבה של שימושות השדה זהה) להחזיר גרסה קיימת של המשאב אם זו קיימת אצלם (ולחסוך את הזמן ואת התקשרות כדי לבקש מהשרת שוב ולקבל ולהחזיר אותו ללקוח).

מתי זה שימושי? זה שימושי כאשר יש תוכן ותמונות שחוזרים על עצם בין דפים, ואז אפשר לקבל אותם מהמטמון ולא לשוב ולבקש ולשנע אותם באינטרנט בכל פעם מחדש.

עדכון של גרסת המשאב בכותרות מאפשר לכל הצדדים שמעוניינים לתמוך במטמון לשתף פעולה ולחסוך. המשאב יובא מחדש רק כאשר השירות ידוח שהגרסה השתנתה או אם יש גורם בדרך שמחזק או משנה באופן שאיןנו נתמך את המידע על הגרסה. ה-ETag מאפשר לא רק לשלוט בגרסה אלא גם לזהות באופן ייחודי את המשאב. אם למשל שומרים על ערך ערבי, [hash](#), שמייצג את המשאב, למשל באמצעות [md5](#) או [CRC](#), אז אפשר בהסתברות גבוההלי ליאץ משאב ללא התנגדויות ולנהל גרסאות מטמון עבורי.

חברת KissMetrics השתמשה בשדה זהה לשימוש אחר. היא אחסנה בו העתק של ייצוג המידע שמאפשר לייצר מחדש את המידע בדיוק בצד הליכון (למשל עוגיות). בפרט, אחסון של ערך העוגיה בשדה ה-ETag. פרסום המאמר שהזכיר שני אתרים ידועים, [hulu](#) ו-[news](#) גירר דיון ציבוררי, תביעות בנושא פרטויות אלה גיררו את פרסום המאמר הזה.

נפרט על מגנון ה-ETag. ראשית נראה כיצד משתמשים בו כראוי, ואז נראה מהו השימוש הלא שגרתי לצורכי גיבוי העוגיות. מגנון ה-ETag, כאמור, מאפשר לבדוק האם נדרש לשחק משאב מסוים מטמון קיים או שיש צורך להביא אותו מחדש מהמקור. רעיון המטמון פשוט. חשבו לעצמכם שאתם יושבים לכם במשרד ובמטבח יש חטייף שמתחשך לכם לנשנש ממנה. אתם יכולים לגשת למטבח, לקחת יחידה אחת, לנשנש ולשוב למקוםכם. אם תרצו עוד, תצטרכו לקום, לגשת למטבח לקחת עוד לנשנש וחזר חיללה. מה עושים ילדים קטנים במסיבות יום הולדת כדי לפתור לעצם את הבעיה? הם יוצרים לעצם מטמון מקומי: לוקחים כוס חד פעמי, מלאים אותה בחטיפים ואז ניגשים למקום שם הם זוללים. כך הם יכולים לחסוך לעצם את הגישה שוב ושוב לשולחן החטיפים, אולי אפילו חוסכים את העמידה בתור ומרוויחים זמן וצריכה נוספת מהמקום בנוחות. למען האמת, مثل החטיפים מתאר מגנון אגירה, [buffering](#), או ח齊יה, [aggregation](#).

---

AIR ליעקב אחרי גולשים בעדרת עוגיות חסינות מחיקה

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

צריכים להעביר מקום למקום. באופן זה, אפשר לחסוך מסע הלוך ושוב כמספר הפריטים. האילוץ הוא הזמן שיש לנו בכל פעם לאסוף פריטים, וכמובן מגבלה כמוות המטען שהוא יכול לשות בכל מסע. מטעון במערכות מחשב הוא אולי מתחכם יותר. הוא משלב אגירה שכזאת ביחד עם ייצור של עותק מקומי: להבדיל מחטיף שנוצר ואוזל, כאשר מדובר במסaab כמו טקסט גדול, תמונה, סרטון או מבנה נתונים גדול, שמשירה על העותק זמין במתוון מקומי טוב לשימוש חוזר לניצח. או עד אשר המשאב מתעדכן או משתנה. חשבו למשל על רשומה ענקית של נתונים. כל עוד העותק המקומי שלנו נכון אז אין בעיה להשתמש בו שוב ושוב. אולם משעה שהרשומה מתעדכנת המקורי, היינו רוצים להיות מסוגלים למשורר העותק חדש, או רצוי רק עדכון של השינויים.

מגנון ה-ETag מאפשר לעשות את הדברים האלה בכך שהוא מעביר מידע על העדכניות של המשאב שבמטוון הדפסן. כך הדפסן מביא משאים רק אם אין עותק עדכני שלהם ברשותו. מי מחליט האם העותק עדכני? מה רושמים ב-ETag? השרת רושם מזהה ייחודי לכל משאב שאפשר למצוא באמצעות URL. בכל פעם שישתנה המשאב בכתובת השרת יעדכן את המזהה. דפסן שציריך תמונה, למשל, כדי להציגה בעמוד, בודק אם יש ETag. אם יש ETag, האם הערך שמתאים ל-URL זהה לערך נשמר בפעם הקודמת? אם כן, אין צורך להביא את המשאב מחדש. אחרת, הדפסן מביא את המשאב מחדש. הערך שימושים בו כחתיימה או כתביעת מצב **לוויי** הגרסה נשמר כמחוזת טקסט בתווים המותרים בערכיהם של כותרות-ב-HTTP.

לפניכם דוגמה טיפוסית: בקשה GET ראשונה להצגת התמונה הלוגו של DigitalWhisper תראה כך:

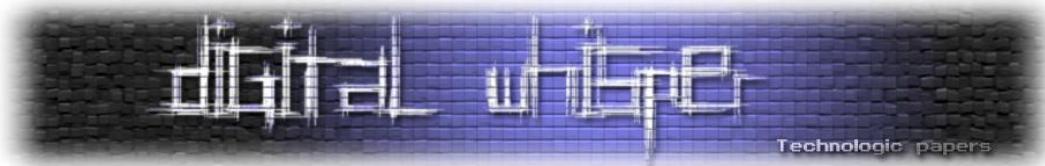
```
GET /logo.png HTTP/1.1
Host: digitalwhisper.co.il
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US)
AppleWebKit/533.4 (KHTML, like Gecko) Chrome/5.0.375.70 Safari/533.4
Cache-Control: max-age=0
Accept:
application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8,vi;q=0.6
Accept-Charset: windows-1255,utf-8;q=0.7,*;q=0.3
```

התגובה שנתקבל מהשרת בתמורה תראה כך:

```
HTTP/1.1 200 OK
Date: Thu, 25 Aug 2011 21:32:18 GMT
Server: Apache/2.2.9 (Unix) mod_ssl/2.2.9 OpenSSL/0.9.8e-fips-rhel5
PHP/5.2.6
Last-Modified: Wed, 06 Jul 2011 20:59:07 GMT
ETag: "25cd3c-33ab5-4a76cdf081cc0"
Accept-Ranges: bytes
Content-Length: 211637
```

איך לעקוב אחר גולשים בעדרת עוגיות חסינות מחיקה

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



Content-Type: image/png

[PNG DATA]...

תוכן התמונה חוזר ואיתו ה-ETag:

ETag: "25cd3c-33ab5-4a76cdf081cc0"

במידה ונבצע "ריענון" לעמוד / או שנגייע לעמוד אשר טוען את אותה התמונה, הדף ישלח את בקשה ה-GET הבא:

```
GET /logo.png HTTP/1.1
Host: digitalwhisper.co.il
Proxy-Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US)
AppleWebKit/533.4 (KHTML, like Gecko) Chrome/5.0.375.70 Safari/533.4
Cache-Control: max-age=0
Accept:
application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8,vi;q=0.6
Accept-Charset: windows-1255,utf-8;q=0.7,*;q=0.3
If-None-Match: "25cd3c-33ab5-4a76cdf081cc0"
If-Modified-Since: Wed, 06 Jul 2011 20:59:07 GMT
```

והתגובה שנקלט בתמורה תראה כך:

```
HTTP/1.1 304 Not Modified
Date: Thu, 25 Aug 2011 21:39:02 GMT
Server: Apache/2.2.9 (Unix) mod_ssl/2.2.9 OpenSSL/0.9.8e-fips-rhel5
PHP/5.2.6
ETag: "25cd3c-33ab5-4a76cdf081cc0"
```

כך הדף יודע שהוא המשאב (תמונה הלוגו) השומרה אצל ב-*Cache* עדין רלוונטי וניתן להציג אותה ללא חשש שהיא שונתה.

איך מtabצע גיבוי העוגיות באמצעות ETag? השרת מניח שהדף יכבד כותרות ETag, וישתמש בערך שלhn בבקשתות עתידות באמצעות If-None-Match. המשמעות היא שהדף שומר על הערך הזה אצל וחווזר ומשדר אותו לשרת שוב ושוב. זה בדיק המנגנון הרצוי. עכשו השרת לא רק מבדיק עוגיה עם המידע שנცבר על המשתמש, אלא גם שומר על עותק שלה בשדה ה-ETag. אם השיטה הזאת חוזרת על עצמה בכל עמוד ועמוד באתר, אז גם הדף מוחק את העוגיה מטעמי פרטיות, אין לו סיבה למחוק את

---

איך לעקוב אחריו גולשים בעזרת עוגיות חסינות מחיקה

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

ערכי ה-ETag נשמרו מסיבות של חיסכון בתקשורת. כך אם השרת אינו מוצא עוגיה בבקשת מדף, אבל מוצא ערך שמתאים לפורמט העוגייה שלו בשדה If-None-Match, אז הוא משחרר מתוך כך את העוגייה, מעדכן אותה, מבדיק אותה מחדש (או שלא...) בתגובה הבאה, ובכל מקרה מעדכן עותק שלה ב-ETag.

## התגוננות

איך אפשר לבדוק אם יש אצלנו דבר זהה? בודקים עוגיות למיניהן ואמצעי אחסון צד לקוח - או שימושים - מעטה גם מנסים להסיר שדות של כותרי HTTP שאיננו מכירים ושאינם יודעים שהם משרות מטריה מועילה בתקשורת שלנו מול אתרים, למשל ה-ETag.

זה סיפור לא פשוט למשתמש רגיל אבל ממש לא מסובכת לתווסף לדף (ה��ition) כבר לכואורהאפשר חסימה של הטכנולוגיה הזאת של KissMetrics - אבל אל תאמינו עד שתראו זהה באמת עובד) או לפרטן אבטחה צד לקוח אחר (כמו מכונות קצה שמבצעות אבטחת מידע אפליקטיבית).

בintérm, החברה נתבעה, וטענה שהיא הפסיקה את השימוש בטכנולוגיה. אולם חברות נוספות נמצאו משתמשות בתרגיל הזה, חלון נתבעו ולא ברור מי או מה ימנע מהן או מכל גוף אחר להמשיך להשתמש בה לצורכי מעקב. נראה שתווצהה ברורה היא חברות מהוונוט יפסיקו להשתמש בטכנולוגיה זאת.

נראה ש גופים אחרים לא בהכרח ירגישו מחוייבים גם כן לעשות זאת. נראה שהMRI צד בין נסיבות שונות להשתמש בטכנולוגיה כדי למשר פרטיות לבין שימוש בטכנולוגיה כדי למשר מעקב ימשייך בדיק כמו שהMRI והMRI בין הפורצים למיניהם ופתרונות האבטחה ממשיר. יש אולי שחוobsרים שחקיקה מסיע. אני סבור שהיא לא תסייע לנו נראה יותר מאשר היא מסייעת נגד פשעים אחרים, במחשב ושלא במכשיר.

למה עניינים, אפשר לקרוא תייר שווה לכל נפש של אחד מהחושפים של הטכנולוגיה כאן:  
[http://ashkansoltani.org/docs/respawn\\_redux.html](http://ashkansoltani.org/docs/respawn_redux.html)

## תשתיות מפותחות ציבוריים

מאת: עמיחי פרץ קלופשטיין

### הקדמה

פעמים רבות בעולם המחשבים וابتחת המידע, נוצר צורך בראש תקשורת מאובטחת ומהימנה, אשר העברת נתונים דרכה תהיה סודית ובטוחה מפני שימי - זדוני או מקרי כאחד. ישנו פתרונות רבים לבעה זו, כשהבולטים שבהם מגיעים בתחום הкриיפטוגרפיה. במאמר זה אסקור את שיטת המפותחות הפומביים, כיצד היא עובדת, מהם היתרונות והחולשות שלה, ו דרכים שונות למשוך אותה כתשתית תקשורת.

### הצפנה סימטרית ואסימטרית

בהתורת הкриיפטוגרפיה, קיימות שתי צורות הצפנה עיקריות: הצפנה סימטרית, בה מפתח ההצפנה זהה למפתח הפענוח, והצפנה אסימטרית, בה קיימים זוגות של מפתחות הצפנה תואמים זה לזה. עקב התכונות המתמטיות של המפתחות האסימטריים, הودעה שתואפען באמצעות מפתח אחד יוכל להיות מפענחת רק באמצעות המפתח השני, ולהפך.

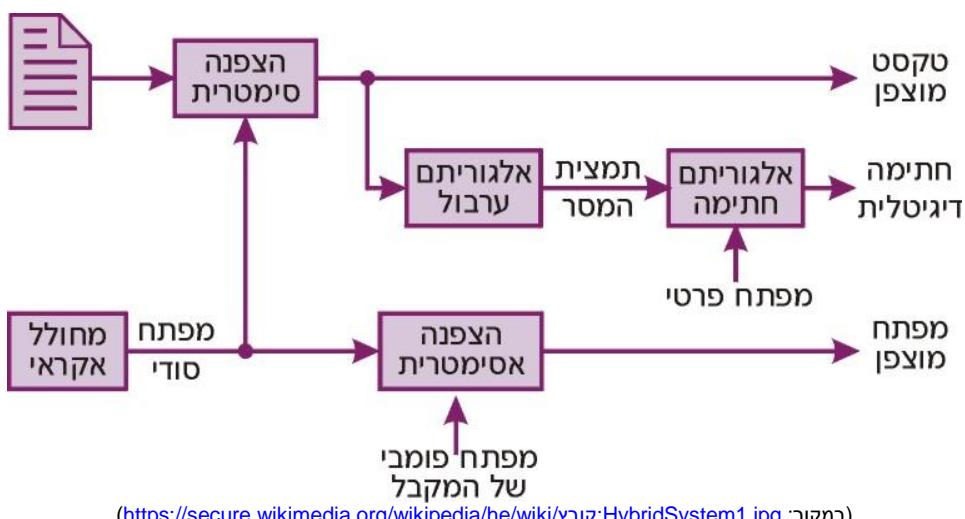
לכל אחת מן השיטות הנ"ל יתרונות וחסרונות, אך הסיבה העיקרית שבזקотה הן מעדיפים להשתמש בהצפנה אסימטרית לתקשורת מסווג זה, היא שבשונה מהצפנה סימטרית, בה יש צורך להעביר את מפתח ההצפנה בצורה סודית לשני הצדדים המשתתפים בתקשורת, בהצפנה אסימטרית אין צורך לשומר על סודיות המפתחות - למעשה, ניתן לפרסם אותם באופן פומבי באינטרנט.

כאשר משתמשים בהצפנה מפתח פומבי, לכל גורם בקשר התקשרות ישנו שני מפתחות. באמצעות מפתח אחד, שנוהג לכנותו מפתח ציבורי או פומבי, (Public Key) ניתן להצפין הודעות אל האדם, אשר רק הוא יכול לפענה. מפתח זה מופץ באופן חופשי לכל דושר, על גבי שרתטי מפתחות ייעודיים או בדרךים אחרות.

המפתח השני, הינו מפתח סודי, אשר רק לבליו יש גישה אליו, ונקרא מפתח פרטי. (Private Key) מפתח זה משמש את בעליו כדי לפענה הודעות שהוצפנו באמצעות המפתח הפומבי שלו, ולהתמוד על הודעות יוצאות.

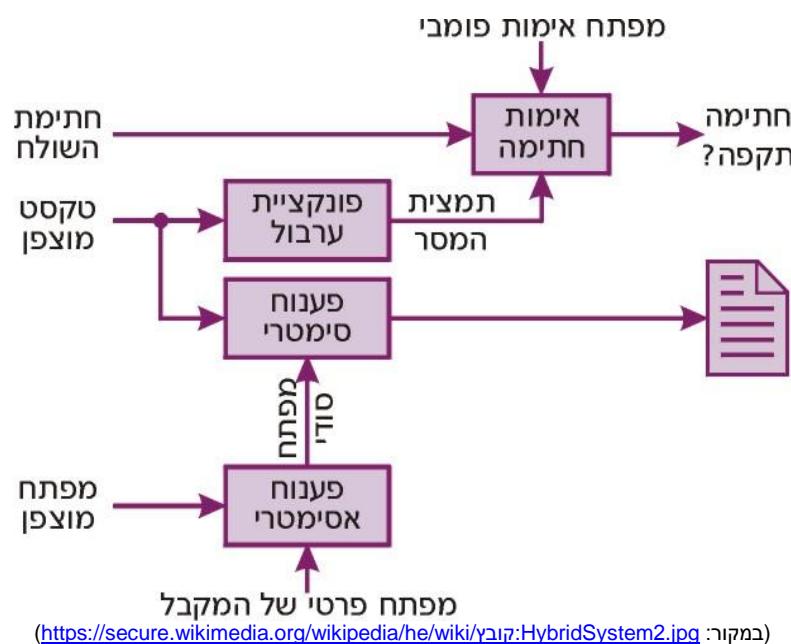
לדוגמא, אברהם מעוניין להעביר הודעה סודית אל יצחק. ראשית אברהם יחתום על תמצית (Hash) של ההודעה המקורי, ייצורו אותו אל ההודעה. בעת אברהם יצפין את כל המסר בצוון סימטרי, ואת המפתח יצפין באמצעות המפתח הציבורי של יצחק, וישלח את ההודעה המוצפנת+המפתח המוצפן אל יצחק דרך ערוץ תקשורת לא מאובטח.

ניתן לראות את התהליך המלא בדוגמה הבאה:



כאשר יצחק קיבל את ההודעה, ראשית הוא יפענה אותה בעזרת המפתח הפרטי שלו. לאחר מכן, הוא יודא שakan ההודעה נשלחה על ידי אברהם, בכך שיפענה את התמצית המוצפנת שצורפה להודעה,

וישווה אותה מול תמצית שנוצרה באותו רגע מההודעה עצמה. כל שינוי קטן בהודעה יתגלה בצורה חזותית מייד. תהליך זה מוצג באירור הבא:



כך, אברהם יודע שרק יצחק יהיה מסוגל לקרוא את ההודעה, משום שרק לו יש את המפתח הפרטי הנכון, יצחק ידע שרק אברהם יוכל לשולח לו את ההודעה, כיוון שהיא חתומה עם המפתח הפרטי שלו.

## חולשה פוטנציאלית

נתאר לעצמנו מצב כזה: יצחק חי בישראל, בזמן שאברהם חי בארצות הברית. יש להניח, שהם לא יפגשו לעיתים קרובות, ויתכן שהתקשרות שלהם מבוססת על מפתחות פומביים שנשלחו בדוא"ל או הופצו דרך שרת מפתחות.

מצב כזה עשוי לפתח דלת למתפקיד האדם שבאמצע (Man in the middle). בפשטות, מדובר על מצב בו גורם זדוני כלשהו יתחזות ליצחק כלפי אברהם, ולאברהם כלפי יצחק. אם הופיע גורם ליצחק לחשב שהמפתח הציבורי שלו הוא המפתח הציבורי של אברהם, ולאברהם לחשב שהמפתח הציבורי שלו הוא של יצחק, כל התקשרות ביניהם תהיה נתונה להסדיו - הוא יכול להאזין, לסנן, לזייף ולשנות הודעות, אשר יראו אמינותן לחלווטין. למעשה, חוויה של מתקפה זו הוא בכך שאין לקרבן כל סיבה להחשוד בפעולות זدونיות, ובכך שההתוקף מקבל אמינות גבוהה יחסית.

## חתימה על מפתחות

ישנן שיטות רבות להתגבר על בעיה זו, וכולן מבוססות על זיהוי פיזי של האדם, שיקשור למפתח ציבורי מסוים. בקרה כזו, לגורם זמני יהיה קשה בהרבה להתחזות לאדם אחר.

הבעיה, כמובן, היא הצורך בזיהוי אמין - זיהוי פנים אל פנים, באופן אישי או על ידי גוף שלישי אמין. כאשר מוחאים את הגורם המבוקש, חותמים על המפתח הציבורי שלו (למעשה, על תמציתו של המפתח), באמצעות המפתח הפרטי של הגורם המאושר.

קיימות שתי שיטות עיקריות לזיהוי כזה:

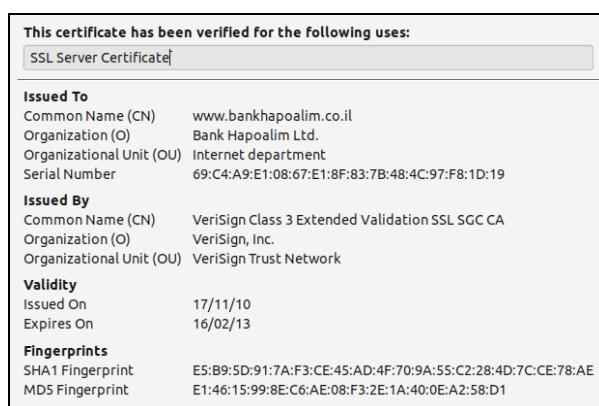
- שיטת הרשות המאשרת (Certificate Authority)
- שיטת רשת האמון (Web of trust)

### רשויות מאשרת

בשיטתה זו, ישנן חברות אשר עוסקות בוידוא זהותם של גורמים, ובחתימה על מפתחותיהם. גורם אשר מבקש לקבל אישור כזה, צריך לאמת את זהותו על פי דרישות ונהלי החברה. רשותות מאשרות נפוצות במיוחד במגזר העסקי, בו יש צורך בזיהוי אמין מעלה לכל ספק, ובתחום האינטרנט, לצורך אינטראקטיביות של אתרים שאין אפשרות הגולשים לאמת בעצמם.

שיטתה זו נפוצה לרוב אתרים אינטרנט גדולים, לצורך שימוש בפרוטוקול SSL. במקרים כאלה, האתר מאמת את זהותו באמצעות אישור אבטחה החתום על ידי רשות מאשרת, אשר המפתח הציבורי שלו מוטמע בדף או מתקבל משרת מיוחד. בזכות האמון ברשות מאשרת (שנובע מגורמים רבים), ניתן לסמוך על זהותו של אתר האינטרנט.

האמון של המשתמשים הינו בתחום הלו בלבד, ואישוריהם הדדיים בין "משתמשים" אינם נפוצים, ואין בהם לבתוחים. חברות מוכרות בתחום זה כוללות את Comodo, RSA, VeriSign, ועוד... ניתן לצפות באישורי האבטחה באמצעות הדף – הנה, לדוגמה, אישור האבטחה של בנק הפועלים:



### רשות אמון

השיטה השנייה לבניית תשתיות מפותחות ציבוריים מסתמכת על האמון שקיים בין אנשים באופן טבעי. בשיטה זו כל גורם ברשת משתמש הן כרשות מאשרה והן כמשתמש. הרשות החברתית שנוצרת ב;zורה כזאת מבטיחה את אמינותם של המפתחות.

לדוגמה, אברהם רוצה להעביר ליצחק הودעה מוצפנת. אברהם, שמכיר את מתפקיד האדם שבאמצע, מעוניין לוודא את אמינותו של המפתח שיצחק שלח לו. במקרה, חבר משותף של השניים, יעקב, נפגש עם יצחק, ומהזיק בעותק של המפתח המקורי. כאשר הוא נפגש עם אברהם, הוא נותן לו את המפתח המקורי, וכך לאברהם יש ערך תקשורת בטוח אל יצחק.

שיטת זו נקראת רשות אמון. למעשה, אברהם נתן אמון יעקב, יעקב ידידותם, וסמרק עליו כגורם מאשר למפתח של יצחק. שיטת רשות האמון מבוססת על עיקנון זה, אך בקנה מידה גדול בהרבה. אכן, במקרים רבים להעביר את המפתח עצמו, המפתחות נמצאים על שרת מפתחות מרכזי, וכל אדם שווידיא את זהותו והמפתח של אדם אחר, חותם על המפתח ומעלה את החתימה לשרת. במקרה, אם אברהם ירצה לשולח ליצחק מכתב, הוא יוכל להוריד את המפתח הציבורי שלו, ולראות את כל החתימות שבוצעו עליו. במקרה זה, כאשר יראה שאנשים אמינים (כמו יעקב, שרה, רחל ולאה) חתמו עליו, הוא יוכל לתת בו אמון מסוים.

בתورو, אברהם יחתום על מפתחות של אנשים אותם הוא מכיר, ואנשים ש מכירים אותו יחתמו על המפתח שלו. ברשותו, אנו נתונים אמון במפתח גם יעקב שרשרת של קשיים אישיים: אם אברהם חתום על המפתח של יצחק, יצחק חתום על המפתח של יעקב, אברהם יוכל לסמן על המפתח של יעקב.

צורה זו פוחת בטוחה, כיון שאין לנו אפשרות לדעת האם יצחק אכן בדק richtig את זהותו של יעקב. לעומת זאת, אם יצחק, שרה ורחל חתמו על המפתח של יעקב, אברהם סומך על כולן, ניתן יהיה לסמן על המפתח של יצחק.

ובנוסף זה רצוי להבחן בין שני סוגים אמון - אמון באדם כמשתמש ברשת, הכולמר אמון שהוא אכן מי שהוא טוען שהוא, והמפתח אכן שלו, לבין אמון באדם כרשות מאשרת, שבסביבתו צריך לבדוק האם האדם ראוי לאמון, ובבדיקה היבט כל משתמש ומפתח ברשות לפני שייחתום עליו.

### מסיבת חתימת מפתחות

כיון שבחיי היום-יום רובנו לא משתמשים עם עותקים של טביעת האצבע של המפתח הפומבי שלנו, ולא מכירים הרבה אנשים שימושתתפים ברשות אמון כזו או אחרת, אין לנו הרבה הזדמנויות לחתום על מפתחות ולשפר ולהרחיב את הרשות.

לשם כך נוצרו מסיבות חתימת מפתחות. מסיבת חתימת מפתחות הנה בעצם מפגש חברותי, בו מתרכזים אנשים (ובעיקר גאים) לצורך חתימה על מפתחות ציבוריים. כל משתף מקבל עוד לפני (או בתחילת) המסיבה רישמה של כל המשתתפים, וטביעות האצבע (Key fingerprint) של המפתחות שלהם.

במהלך המסיבה, כל משתף אמר לזוות את עצמו בפני כל האחרים בעזרת תעודה זהה, רישון נהג או דרכן (ולעתים אף באמצעות תעודה חוגר), ולהקريا את טביעת האצבע של המפתח שלו. שאר המשתתפים מודאים את זהות האדם, ואת זהות המפתח שלו, ומסמנים זאת בטופס.

לאחר המסיבה, כל משתף מגיע למחשב בטוח (בעיקר נטול רוגלוֹת), ומודא לפיה הרשימה את טביעות האצבע של המפתחות שברשותו. אם טביעות האצבע תואמות, הוא חותם על המפתח ובכך מאשר את אמינותו. לאחר מכן, ניתן להעלות את המפתחות החתוםים לשרת ציבורי.

## סיכום

במאמר זה הצגתني את שיטת רשות האמון, ואת רעיון מסיבת חתימת המפתחות. למעשה, שימוש נכון בراتת אמון (ובתנאי שהוא גדול מספיק) עשוי להיות צורת תקשורת מהימנה למრבית השימושים.

יתכן שלאחר קריית המאמר התעורר לכם הشك ל问我 אם או להצטרף לרשות אמון. הנה! בישראל ישנה רשות אמון ותיקה וモורכנת של קהילת הלינוקס והקווד הפתוח, אשר עורכת מסיבת חתימת מפתחות מדי שנה בכנס הקוד הפתוח "אוגוסט פינגוין", והחבריה ישמחו לחותם הדדי על מפתחות כמעט בכל עת, וכמוון שישן (וניתן להקים) רשותות נוספות לאנשים עםם אתם בקשר.

שימו לב, שככל שייתר אנשים יהיו קשורים לרשות אמון כזו או אחרת, כך תגדל יעלותה של הרשות הגלובלית. רצוי מאוד ליצור קשרים בין רשותות קטנות, וביחוד מול גורמים בחו"ל. כל קשר כזה מספק תקשורת מהימנה לאנשים רבים.

## על המחבר

עמיחי פרץ קלופשטיין, בן 18, עוסק באבטחת מידע מזה כונה וחצי, חובב לינוקס ותוכנה חופשית, ומתנדב ב��בוצת [dc9723](#).

## טכניקות התרבות בקרבת תולעים חברתיות

מאת: אפיק קוסטיאל (cp77fk4r)

### הקדמה

במסגרת מאמר זה אני מעוניין להציג סקירה על מספר מנוגנוני התפשנות של תולעי-אינטרנט שונות שפקדו ופקודות אותנו בעבר ובזמן האחרון. מטרת המאמר הינה להציג לכם, הקוראים, איך עולם תולעי-האינטרנט בכלל, ומנגנוני התפשנות שלהם בפרט הולך וצובר תאוצה בשנים האחרונות.

קיימיםים מספר רב של וקטורי הפצתם בהם תולעי-אינטרנט מבצעות שימוש בכדי להגיא ל扩散 נרחבת, וקטורי הפצתם כגון:

- ניצול חולשות במערכת הפעלה.
- הדבקת התקני אחסון ניידים (USB).
- שליחת אימיילים (Mass-Mailing).
- הדבקת קבצי ברזי-הרצה.
- הנדסה חברתית.
- רשתות Peer 2 Peer.
- ועוד ...

אך במאמר זה אני מעוניין לגעת רק בקטורי הפצתם בתולעים מבצעות שימוש בהנדסה חברתית. מצד אחד, הגיוני מאד להניח כי מנגנוני הפצתם אשר אין דרישים אינטראקטיביים של המשתמש ומנצלים חולשות במערכת הפעלה / דף-און הם מנגנוני הפצתם הטובים ביותר. אך מצד שני - ברגע שההתולעת נתפסה, נחקרה והובן מהו כשל האבטחה בו היא עשו שימוש בכדי להתפשט, ח"י התולעת יהיה קצרים ביותר. אותה החברה שאחרהית על המנגנון בו נמצא כשל האבטחה תשחרר טלאי שיסגור אותו, וכן בערך תמו חייה של התולעת.

מהסתכלות אחרת בזמן, ניתן לראות כי דווקא התולעים שלא נצלו כשל אבטחה אלא עשו שימוש בקטורי התפשנות אשר כן דרישים אינטראקטיביים עם המשתמש ובמציאות שימוש בהנדסה חברתית- הן התולעים עם אורך החיים הגדל ביותר.

מדובר בשאלת לא פשוטה, והשאלה האם וקטורים אשר דורשים אינטראקציה עם המשתמש ("user") הם וקטורי ההפעזה הטובים ביותר להפצת תולעים היא שאלת מעניינת בפני עצמה, אך היא אינה تعالיה על הפרק במאמר זהה.

ולענינו, כותב התולעת יכול להשתמש בהנדסה חברתית במספר פלטפורמות שונות, כגון:

- תוכנות מסרונים מיידיים (ICQ, MSN, Yahoo! Messenger, Skype, IRC ועוד).
- משלוח אימיילים (דרך Outlook, שרת SMTP שונים ו-Webmails וכו').
- רשתות חברתיות (Facebook, Myspace וכו').
- ועוד ...

במסגרת המאמר אציג מספר תולעים אשר מבצעות שימוש בפלטפורמות השונות בכך שנוכל לבצע השוואה.

### אני אוהב אותך אנה קורניקובה!

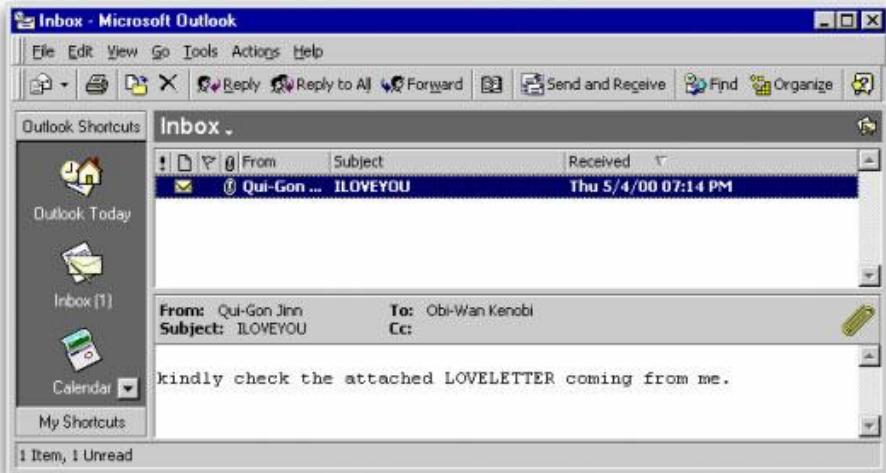
מדובר בשני תולעי-אינטרנט אגדיות (משנת 2000 ו-2001), הראשונה היא ה-"Love Bug" (מוכרת גם כ-[ILOVEYOU](#) ו-"[Love Letter](#)") והשנייה היא כМОבן התולעת: "[VBS.SST@mm](#)", או כמו שכל המדיה באותה התקופה כינה אותה: "[The Anna Kournikova Worm](#)".

שתי התולעים הללו התפשטו בעזרת משלוח הודעות דוא"ל. שתי התולעים הפיצו את עצמן בעזרת התממשקות לתוכנת Outlook של המחשב בו הונפתחו ושלחו את עצמן כקובץ מצורף לאימייל אשר נשלח לכל רשימת אנשי הקשר של הקורבן.

בשתי התולעים לא נצלה אף חולשה בקוד של אף אחד מהמנגנים שלוו את השילחה (כגון חולשות לוגיות או חולשות מבוססות זכרון כמו שנמצאה בתולעים אחרות), אלא פשוט התממשקות לתוכנת ה-Outlook בעזרת שירות סקריפט לפקדי מאקרו של התוכנה (אובייקטיבים כגון `Outlook.Application` ו-`WSH` ו-`VBS` עושים את העבודה יופי).

התולעת "ILOVEYOU" הפיצה אימייל עם הכותרת "ILOVEYOU" ובתוכנו היה רשום המשפט:

Kindly check the attached LOVELETTER coming from me"



(במקור: <http://iamjenessa.wordpress.com/2011/06/28/memories-of-the-love-bug-worm-by-graham-cluley-on-may-4-2009/>)

לאוטו האימייל היה מצורף קובץ- "מכתב אהבה" בשם: "LOVELETTER.TXT.VBS". הרעיון הוא שاز, במערכת הפעלה Windows, ברירת המחדל הייתה לא להציג את סימנות הקבצים, כך שהקובץ הוצג כ-"LOVELETTER.TXT". כמובן שכאשר הקורבן היה לוחץ על הקובץ במחשבה שהוא הולך לקרוא מכתב אהבה- הוא בעצם הריץ את אותו קובץ VBS שהכיל את התולעת שהייתה מדביקה לו את המחשב ומפיצה עצמה לכל רשות אנשי הקשר שלו.

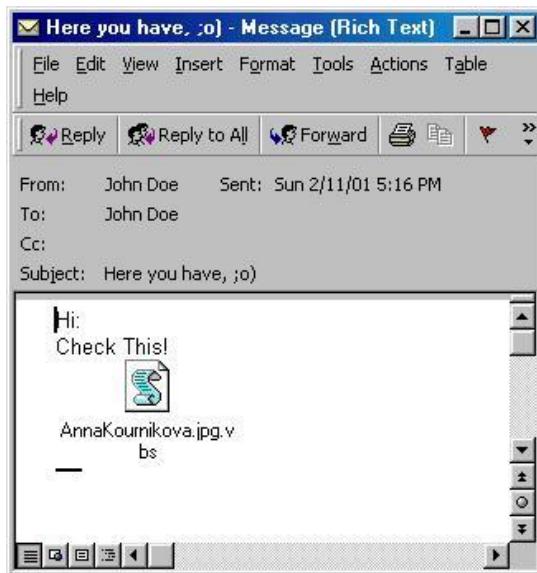
התולעת השנייה, על שם שחקנית הטניס הרוסיה [אננה קוירניקובה](#), הרעיון הוא אותו רעיון. המימוש זהה כמעט לחלוטין, אך במקום לקבל מכתב אהבה, התולעת הפיצה את עצמה באימיילים שהתיימרו להיכיל תמונה של השחקנית. נושא ההודעה היה: "(o;) Here you have, and the content was "Check This!". Shmo של הקובץ המצורף היה "AnnaKournikova.jpg.vbs", ושוב- עקב הגדרות ברירת המחדל של מערכת הפעלה Windows, הסימת ".vbs" הייתה נשמטת.

בכדי שהtolעת לא תרוץ על אותו משתמש מספר פעמים, לאחר הריצה ראשונה, היא הייתה כתבתת ערך לעורך הרישום של מערכת הפעלה, במיקום:

HKEY\_CURRENT\_USER\Software\OnTheFly\mailed

וכך, בכדי לא לשלוח אותה רשות אנשי קשר את עצמה שנית- הייתה מתבצעת בדיקה לפני כל שליחה- האם אותו מפתח קיימ. במידה והיא הייתה מוצאת אותו- לא הייתה מתבצעת השילחה.

ושוב, כאשר הקורבן היה לוחץ על ה"תמונה", התולעת הייתה מורצת ושולחת את עצמה לכל רשימת אנשי הקשר שלו.



(במקור: <http://www.f-secure.com/v-descs/onthefly.shtml>)

כשחובבים על שתי התולעים כיום, מפתיע לחשב שהן הצלicho להתפשט לכל קר הרבה מחשבים. הרי לא עשה כאן כמעט שימוש בהנדסה חברתית, ואפשר להצביע כאן על מספר נקודות די בעיות, כגון זה שהסימנת של הקובץ כאשר הוא מצורף למיל מופיע כ-"S5.TXT" או "sbs.jpg.vbs", או זה שהאייקון של הקובץ נשאר אייקון של קובץ סקריפט ולא מופיע האייקון של הקובץ שהוא מתאים להיות.

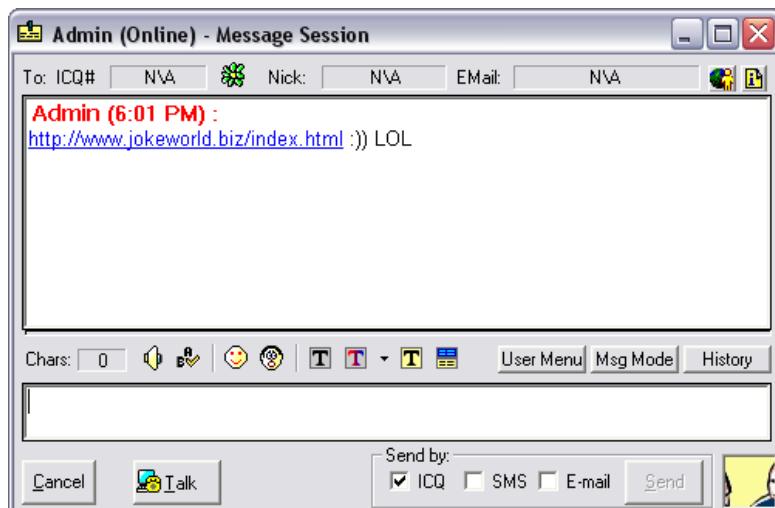
בדדי להבין למה אותן התולעים הצלicho כל קר, אנו צריכים לזכור כי מדובר בתחילת שנות ה-2000, וכמו שכולנו יודעים, קשה להאשים את אותה התקופה במידע מודעoted יתר לסייעים הכרוכים בגלישה באינטרנט. כמובן, המידע לאבטח מידע ולסינונים כגון אלו גבואה יותר. וכך לתולעים כגון אלו, כיום, כמעט ואין סיכוי ליצור את הדר אוטו הן הצלicho ליצור בתחילת העשור הקודם.

## דברי אלǐ בפרחים

סוג התולעים השני עליו נדבר היום הוא ה-"Instant Messenger Worms". מדובר בתולעים שודיעו להתmeshק לפרטוקול של תוכנות המסרים המיידיים השונות (כגון ICQ או MSN Messenger) באופן כזה או אחר, ובעזרת גיבת פרטי ההזדהות של בעלי החשבונות השונים שלהם הודיעו לחבריהם עם קישורים לקבצים / אתרים זדוניים אשר הכילו את התולעים עצמן.

לדוגמא, כאשר תולעים כגון "[Stration](#)" או "[Bizex](#)" היו מוצrcות במחשב אשר עליו מותקנת תוכנת המסרים המיידיים "ICQ" הן היו שולפות את פרטי ההזדהות של המשתמש מהclid מקבץ ה-DAT של התוכנה (הקובץ בו נשמרים פרטי ההזדהות כאשר המשתמש מסמן "Remember Me" בעת ההזדהות לצורכי הזדהות עתידות), מפענחות אותם, ומשתמשות בהם כדי להזדהות בשם של המשתמש לשרת ה-ICQ ולהוריד את רשימת אנשי הקשר שלו. זאת כמובן בכדי ולשלוח בשםו את עצמו לשאר לחבריו. בדרך כלל הן היו שלוחות לחבריו הקורבן (בשם כמובן) קישור לאתר אשר דרש מהקורבן להתקין קובץ EXE במסווה של משחק קטן או שומר-מסמך נחמד.

דוגמא להודעה שנשלחה על ידי התולעת ":"Bizex"



(במקור: <http://www.kaspersky.co.in/news?id=4277566>)

בגרסאות מתקדמות של התולעת, אף נעשה שימוש בחולשה שנמצאה במנוע לפענוח DHTML בדף Internet Explorer (MS03-040) כך שמספיק שהמשתמש היה לווח על הLINK, ומוביל להוריד את קובץ Exe הוא היה נדבק.

QICQ אינה תוכנת המסרים המיידים שכותבי וירוסים ותולעים השתמשו ומשתמשים בה גם היום בתור פלטפורמה להפצת המזקרים שלהם. תולעים אחרות, כגון "Bropia" ו-"Yimfoca" (شمוטציות שונות שלה עדין רצות בשטח) בחרו להפיץ עצמן דרך תוכנות כגון Messenger I-Msn Yahoo! Messenger ו-Yahoo! (בהתאם). הפלטפורמה אולי שונה- אך הקונספט זהה לחלוון. המשתמש מקבל הודעה ממכר / אדם רנדומלי ומתקבש להחזיר על קישור באינטרנט. קישור זה בדרך כלל מפנה לקובץ Exe או במקרים אחרים - עמוד המנצל חולשת 0day בדף אשר מרייצ' קוד וմבדיק את המשתמש התמים בתולעת. כאן אציין כי במקרים מסוימים, במידה ולא נשמרו פרטי ההזהדות בקבצים המיעדים לכך, נעשה שימוש ב-Keyboard Sniffing בכך לדלות את סיסמת החשבון.

עד היום פורסמו לא מעט חולשות בתוכנות מסרים מיידים או בגורםים שונים שניים שנitin היה לנצל באופן מרוחק בכך לגרום להריצת קוד דרך תוכנות מסווג זה, כגון המקרה [הבא](#), אך בסופו של דבר נצפו מספר קטן מאוד של תולעים שניצלו חולשות אלו. מספר התולעים שניצלו חולשות בפרוטוקול התקשרות, או במנגנון שאחראי על פרסור ההודעות של תוכנת המסרים המיידים בכך להריץ את עצמו- מאוד נדירות. במקרים שונים, כמו במקרה של התולעת [Witty](#), נעשה שימוש בחולשת Buffer Overflow שנמצאה במנגנון פענוח פרוטוקול ה-QICQ בתוכנות אבטחה שונות, כגון: BlackICE ו-RealSecure (שתייה של ISS) בכך להריץ קוד על הקורבן.

## תולעי IRC

IRC הינו פרוטוקול המשמש לביצוע שיחות צ'אט בין הרבה משתמשים בזמן אמת. שרתים וערוצי IRC קיימים כבר מסוף שנות התשעים, וכותבי הווירוסים לא פסקו עליהם. חשוב להבין את ההבדל בין Botnets לבין מספר רב מהמקרים **ארכיטקטורת השליטה עליהם מתבצעת על-גבי שרתים IRC** לבין IRC שמנצלות את פרוטוקול ה-IRC **لتותבת הפצתן**. למידע בנוגע ל-[Botnets](#) ניתן לקרוא במאמר: "[מה זאת](#) - [מה זאת?](#)" שפורסם ב[גלוון השלישי של Digital Whisper](#).

תולעי IRC מתחזות למשתמשי IRC אמיתיים ובעזרת הנדסה חברתית / ניצול חולשות בתוכנות לקוח לפרוטוקול ה-IRC (IRC Clients) מפייצות עצמן. גם בפלטפורמה זאת, המקרים בהם תולעי IRC משתמשות בהנדסה חברתית רבים יותר מתולעים אשר ניצלו חולשות שונות ב-[IRC Clients](#) לצרכי הפצתה.

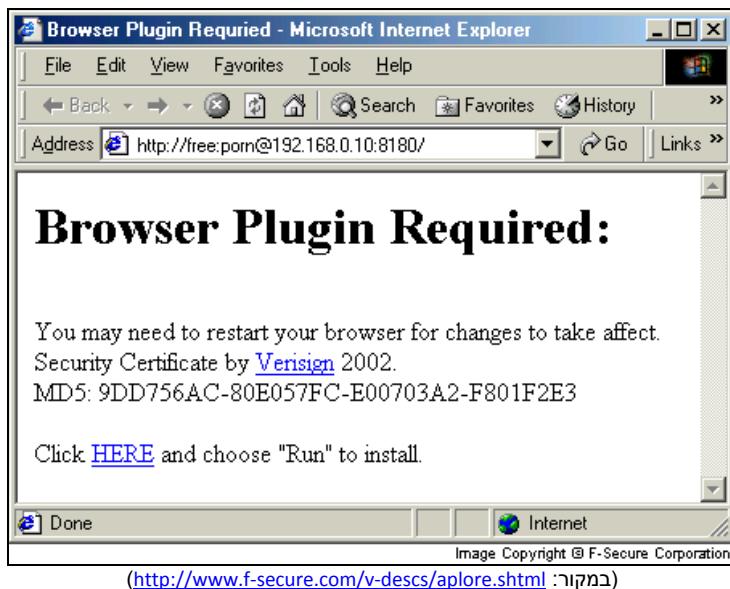
אפשר לחלק תולעים מסוג זה לשתי קבוצות:

- תולעים בעלות מנוע IRC מובנה שמאפשר להן להתmeshק לערוצי IRC מבי' תלות בתוכנת ליקוי ה-IRC שמותקנת על המחשב בו הן רצוט.
- תולעים חסירות מנוע IRC, אשר מתmeshקות לתוכנת הליקוי שמותקנת על המחשב עליו הן רצוט ובעזרתן מתחשרות עם שאר משתמשי הערוץ.

נקח לדוגמה את התולעת "Aphore" (מכירת גם כ- "Aphex") שניצפה לראשונה בשנת 2004. התולעת הפיצה עצמה ע"י מנוע IRC מובנה, שבuzzרטה היא הייתה מתחברת לערוצי IRC שונים, ומפיצה הודעה פרטית לכל המשתמשים הנוכחיים בערוץ. תוכן ההודעה הכליל בדרך כלל הודעות בסיגנון "Free Porn" ו קישור לכתובות אינטרנט של אתר פורנוגראפי בביבול:



משתמש שהיה נכנס לאתר, היה מתקש להתקין תוסף לדף הדרש לצורך צפייה בתוכן האתר:



וכמובן- כל מי שהיה מתקין את התוסף לדף הדרש, היה נדבק גם הוא.

---

טכניקות התרבות בקרוב תולעים חברותיות

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

לעומת "Aplore", שהכילה מנוע IRC פנימי משלה, תולעים כגון "Momma" ו-"Hellfire" היו תולעים שגם הפיצו עצמן דרך ערוצי IRC אך בשונה מ-"Aplore", הן עשו זאת בעזרתו שימוש במנוע הסקריפטינג של התוכנה MiRC (תוכנת לkop IRC נפוצה). במקרים כאלה יוצר התולעת יכול לבחור בשתי דרכי נספנות להדבקת הקורבן, חוץ מהפניה הקורבן לעמוד עם תוכן זמני, ניתן לנסות לשכנע את הקורבן להריץ סקריפט (בטענה שמדובר בפקודות הזרחות לערוֹץ וכו') שיתפרק כ-Backdoor ויאפשר לתקוף שליטה Peer to Peer על תוכנת הצ'אט או שליחת קובץ מדבק על גבי DCC (פרוטוקול נפוץ להעברת מידע בין המחשב הנתמך במספר רב של לkopות IRC) שיאפשר גישה למערכת המותקפת. כמובן שלאחר הדבקת המשמש התולעת אינה נשארת במסגרת ה-MiRC, היא מורידה ומריצה Payloads שונים על המחשב ומפנה לתקוף שליטה מלאה גם על המערכת.

מנוע הסקריפטינג של MiRC (ושאר תוכנות הצ'אט אשר תומכות באוטומציה) מאפשר למשתמש, בין היתר, להגיד מקרו- רצף פעולות שיבוצע בכל פעם שאירוע מסוים מתרחש בערוֹץ IRC בו המשתמש פועל.

אירועים לדוגמה:

- הרצת ישירה של המקרו
- הצליפות לערוֹץ / שרת
- התנטקות מערוֹץ / שרת
- הצליפות משתמש חדש לערוֹץ
- הופעה של מילה או משפט מסוים בערוֹץ
- ועוד אירועים רבים...

הפעולות שניתן לבצע הן אינטנסיביות: מפעולות שנitinן לבצע במסגרת ה-IRC (כתיבת הודעה בערוֹץ, שליחת הודעה פרטית למשתמש, שליחת קובץ למשתמש על גבי DCC, נתינת או ליקיחת הרשות למשתמשים וכו') ועד פעולה על המחשב המקומי (עבודה עם קבצים, הרצת פקודות מערכת, הורדת קבצים מהאינטרנט והרכבתם וכו').

שכתבו של קובץ ייחיד אשר אחראי על מימוש המקרו הקשור לאירועים מסוימים, המשתמש מודבק, אין צורך לבצע Hook לשום פונקציה, ואין צורך לפתח סוקטים ולפרסר את הצ'אט. תוכנות צ'אט המאפשרות אוטומציה הן מכורה זהב, גם מפני שהן נפוצות וגם מפני שבקלות מאוד ניתן להתmeshק אליהן. מסיבות אלו ודוגמתן כמהות התולעים מסווג זה היא/הייתה הרבה כל משתמש ממוצע עם כוונות זדוניות יכול למסח תולעת זאת גם מבלי להכיר יותר מדי את עולם התוכנות.

דוגמאות מעולות פשוטות שבדבר ניתן לראות במאמרים הבאים:

<http://users.telenet.be/ahmadi/mircworm.htm>

<http://vxheavens.com/lib/vsp21.html>

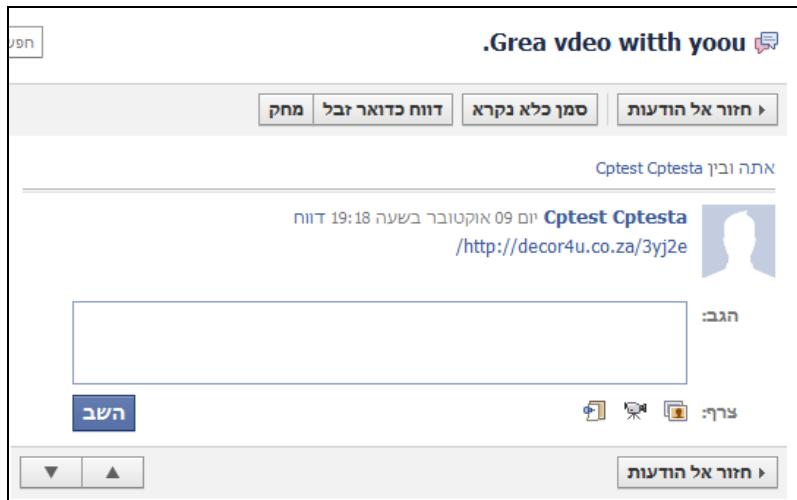
## תולעים חברתיות

רשתות חברותיות הן אחת הפלטפורמות הפופולריות כיום בקרוב כתבי התולעים, דוגמה טובה לכך אפשר למצוא בפרק הראשון של סדרת המאמרים "Chasing Worms" שפורסם בגלוון ה-14 של המגזין, בו הוצג יתוח של התולעת "Koobface".

כיום יש מספר רב של רשתות חברותיות, ישן רשתות בעלות נושאים ספציפיים (InIn, Digg, LinkedIn וכו') וישן רשתות כלליות יותר (Google+, Facebook וכו'), בשני המקרים, עובר מידע רב בין המשתמשים, וכותבי תולעים הבינו את הפוטנציאל שבדבר.

במקרים הקלאסיים, תולעת שמצילה לגנוב את פרטי ההזדהות של המשתמש פשוט שולחת מהחשבון שנפרץ הודעה פרטית לחבריו ובו קישור לעמוד בעל תוכן זמני, אך במקרים קצת יותר נDIRMS נעשה שימוש ייפה בהנדסה חברתית.

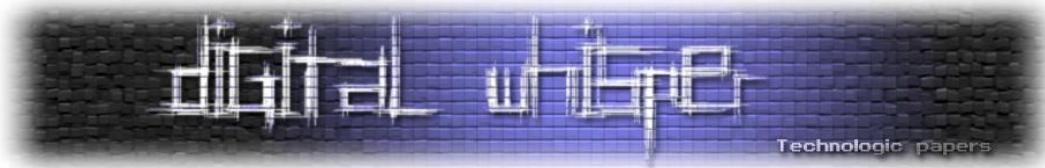
דוגמה למקירה קלאסי הינה Koobface - באופן פשוט יחסית, לאחר השגת גישה לחשבון המשתמש מתבצעת שליחת הודעה פרטית + כתיבה על עמוד הפרופיל של כל חברי המשתמש עם קישור:



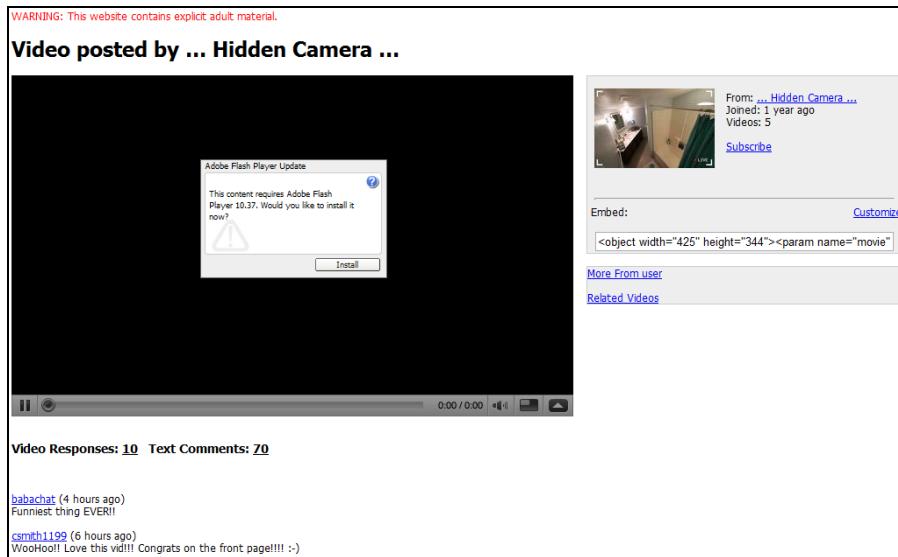
---

טכניקות התרבות בקרב תולעים חברותיות

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



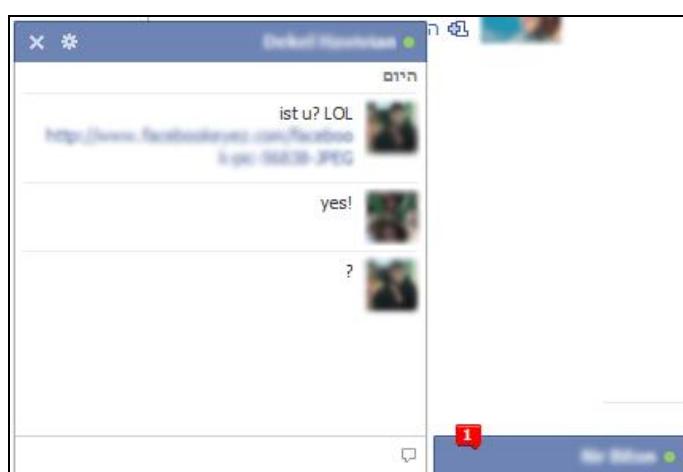
הקישור מפנה לעמוד Youtube פקטיי אשר דרש מהמשתמש להתקין עדכון לתוכנת הפלаш שלהם ב כדי לצפות בסרטון:



עמוד youtube סטטי, ומתחתי יש תגיות של "משתמש" youtube פקטיי. ולמרות זאת, הדבר הספיק ב כדי להפיל אנשים רבים בפח.

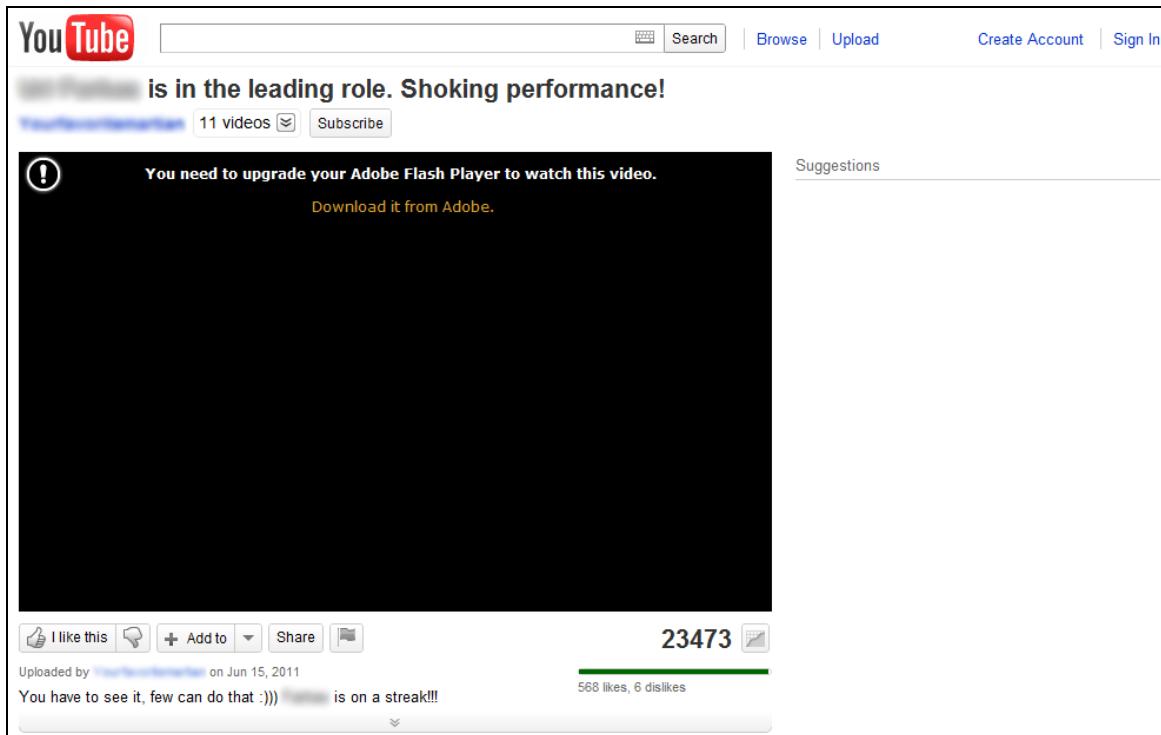
דוגמה הרבה פחות קלאסית, והרבה יותר מרשיימה, אפשר לזרוף לתולעת חדשה יחסית, עדין חסרת שם, אך לפי מחקר קצר שעשיתי (ותודה ל-Zerith) שעורך כאן, נראה שהיא עוזרת להפיץ כלי לריקון חשבוןות BitCoin.

ברגע שהtolעלת מורצת על מחשב ועליו יש חשבון Facebook, היא מתממשקת לחשבון ושולחת הודעה צ'אט לחבר שנמצא כרגע On-Line (גם עם בעל החשבון הפרוץ כרגע מחובר לפיסבוק הוא אינן יראה את שיחת הצ'אט אלא עם החשבון המותקף כתוב לתולעת בחזרה) עם הודעה ו קישור:



טכניקות התרבות בקרב תולעים חברתיות  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

הקישור מפנה גם הוא לעמוד YouTube שcamatan דורש מהקורבן להתקין עדכון "תמים" לדפדף:

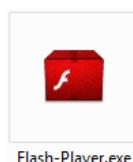


מה שכל כך יפה בעמוד הזה הוא שהוא דינامي. מה זאת אומרת? זאת אומרת שהוא נוצר Fly On The Fly על ידי התולעת, שניות מספר לפניות הودעת הצ'אט.

מה דינامي בעמוד?

- הכוורת של הסרטון מرمצת על כר שמדובר הסרטון על הקורבן
- התגובה על הסרטון מדברות על הקורבן עצמו.
- שמות המגיבים התגובה נלקחו מרשימה החברים של הקורבן.
- תמונות המגיבים לקוחות מתונות פרופיל הפיסבוק האמיתיות של חברי הקורבן.

שכלול של כלל הנתונים הנ"ל לא רק גורמות לקורבן להאמין כי מדובר אמיתי, אלא גם גורמות לו להיריד את העדכון, שאגב, נראה כך:



בוחלת עושה רושם שבוצעה כאן עבודה מקצועית ביותר.

## מספר תגובות על הסרטון הפקטיבי:

He must have been shamed to do that :)))  
5 minutes ago

I had to update Flash Player, but it was worth it :)) this video is the very best!  
6 minutes ago

one word for it – TERRIBLE!  
7 minutes ago

He's the new TV star! Put him on the tonight show! :))))  
10 minutes ago

no comments....  
10 minutes ago

SUPER !!!!  
12 minutes ago

a living person cannot do that, this is fake!  
15 minutes ago

Breathtaking...  
19 minutes ago

Are they high?  
22 minutes ago

Cool vid!  
30 minutes ago

I like it!  
30 minutes ago

Ha-ha!  
30 minutes ago

wow! 23125 views already!!!  
30 minutes ago

You do not get much people to do that...  
42 minutes ago

they are drunk  
45 minutes ago

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [Next](#) [View all Comments »](#)

[Help](#) [About](#) [Press & Blogs](#) [Copyright](#) [Creators & Partners](#) [Advertising](#) [Developers](#) [Safety](#) [Privacy](#) [Terms](#)  
[Report a bug](#) Language: English Location: Worldwide Safety mode: Off

לאחר הרצת ההעדרון, המשתמש מקבל הודעה שגיה שאין לו הרשאה להתקין את הקובץ, אך בשלב זה התולעת כבר הדביקה את המחשב...

## סיכום

לא נגעתי בכלל הפלטפורמות בהן כתבי התולעים משתמשים, אבל עדיין, מהמیدע שהוזג במאמר זה בהחלט ניתן לראות את התפתחות הנושא בשנים האחרונות. אם בעבר כתבי התולעים הרשו לעצםם "להחליק פינוט", היום, כאשר המודעות לאבטחת מייד גבואה יותר- נראה כי הם עובדים קשה יותר ויתר בצד לknoot את אמון הקורבנות. בדרך כלל, משתמש חדש יכול לזהות את התרמית, אך במקרים רבים (כגון הדוגמה האחרונה במאמר) נראה כי הסיכויים שגם משתמשים אלו יפלו לפחות גבואהים.

נראה כי תולעים אשר מבצעות שימוש בחולשות 0day שונות בעלות עיקומת תפוצה נרחבת יותר, אך כאשר מתגלה התולעת ונסגרת אותה החולשה, ח"י התולעת קצרים ביתר. לעומת זאת, וקטור הפצה המשתמש בהנדסה חברתית אומנם מאפשר לתולעת להפיץ עצמה בקצב איטי יותר, אך כאן, אין באג שzier לסגור, והדרך היחידה לעזור את התולעת מלהתפשט היא לעדכן את תוכנת האנטי-וירוס ולהגבר את המודעות הסביבתית לאבטחת מייד.

## מקורות

- [http://www.wormblog.com/im\\_worms/](http://www.wormblog.com/im_worms/)
- <http://ftp.erm.tu-cottbus.de/security/witty-analysis.html>
- <http://digitalwhisper.co.il/files/Zines/0x0E/DW14-1-KoobfacePwning.pdf>
- <http://vx.netlux.org/29a>
- [http://www.symantec.com/security\\_response/writeup.jsp?docid=2001-021219-1830-99](http://www.symantec.com/security_response/writeup.jsp?docid=2001-021219-1830-99)
- [http://en.wikipedia.org/wiki/Anna\\_Kournikova\\_\(computer\\_virus\)](http://en.wikipedia.org/wiki/Anna_Kournikova_(computer_virus))
- <http://en.wikipedia.org/wiki/ILOVEYOU>
- <http://iamjenessa.wordpress.com/2011/06/28/memories-of-the-love-bug-worm-by-graham-cluley-on-may-4-2009>
- <http://www.f-secure.com/v-descs/onthefly.shtml>
- [http://www.spywareguide.com/product\\_show.php?id=3108](http://www.spywareguide.com/product_show.php?id=3108)
- <http://www.kaspersky.co.in/news?id=4277566>

## HTML5 - מנוקודת מבט אחרת

מאת: לירן בנודיס ואלעד גבאי

### הקדמה

בזמן האחרון אנו שומעים עוד ועוד על HTML5 - יש הגורסים כי זהה הטכנולוגיה שתנסה את פניה הרשות (ואף תהווה תחליף לטכנולוגיות פיתוח רבות), אחרים מתעסקים בפתרונות העיצוב החדשנות טכנולוגיה זו מאפשרת ואחרים מתעסקים בשאלת האם HTML5 תחליף את Flash של Adobe או Silverlight של Microsoft.

על אף השפע העיצובי והטכנולוגי של HTML5 מביאה אליה, ישנו לא מעט אנשים שלא מדברים על אפשרויות העיצוב החדשנות שטופנת בתוכה HTML5, לא מתעסקים בדיונים על אפשרות לדרג אתרים קיימים בעזרתה ואףילו לא על אפשרות השיווק החדשנות. אנשים אלו, שהתעמקו ב-HTML5 והטכנולוגיות הנלוות, רואים סיבות לדאגה - HTML5 טומנת בתוכה, בין היתר הירות, הרבה בעיות פרטיות וابتחתה מידע.

בסדרת מאמרים הבאים, נסקור את הנוקודות הביעתיות בHTML5 ובטכנולוגיות הנלוות. מאמר זה יסקור את התגים והתכונות החדשנות ב-HTML5, את מנגנון-XDR/CORS וUMBRELLA ידוע בהשלכות של אלו על מידת האבטחה של התקן החדש.

**יש לזכור כי כרגע HTML5 הינו תקן בפיתוח ולכן חלק מדוגמאות הקוד המובאות במאמר זה לא עבדו בכל הדפדפניים.**

## HTML (HyperText Markup Language)

HTML היא שפת תגיות המיועדת להציג תוכן בעמודי אינטרנט. זו שפה קלה יחסית ללמידה וניתן להציג להציג בערתה אינסוף תכנים באינסוף צורות שונות. את תקן השפה קובע ארגון ה-W3C (ארגון תקינות הרשות העולמי).



גרסת HTML המוכרת והיציבה, אשר נקבעה ב-1999 ולא שונתה מאז, הינה HTML, אליה מתייחסים CSS, JavaScript ו-DOM HTML, ויצרים את השפה הדינמית XHTML. מאז, עולם המחשב כמו גם האינטרנט השתנו רבות, ויש צורך לענות על צרכי מפתחי האתרים שתמיד דורשים יותר ויותר, וכך גם על צרכי הקהל אשר צריך את המידע ותמיד דורש אפשרויות נוספות, ובכל פעם צריך את המידע בצורה חדשה ובפלטפורמות שונות (פלאפונים, מכשירי Tab, טלויזיות, windows, טוסטר משולשים, ועוד...). לפיכך, גבר הצורך בעדכון שפת HTML.

הדור הבא של התקן הינו HTML5, שגרסת התקן הראשונה שלו יוצאה ב-2008 ועדין נמצא בפיתוח.

כיום, כאשר דרישת הקהל, ובעקבותיו דרישת המפתחים, לפיצ'רים חדשים ופונקציות נוספות היא מיידית, לא תמיד ניתן לספק דרישת זו בזמן. דוגמה חייה היא HTML5. לאחרונה ישנים הרבה אתרים אינטרנט המשתמשים בתקן HTML5. תקן זה עד נמצאו בפיתוח ועתיד לצאת רשמית רק ב-2014!

העניין שלם ב-HTML5 נובע בבדיקה מסיבה זו. כיום נוצר מצב שהתקן של HTML5 לא קבוע ומשתנה לעיתים תכופות.

רוב הדפדפניים הפופולריים כבר תומכים בחלק מן האלמנטים והפיצ'רים החדשניים שמציע התקן, כל דףדף על פי בחירתו ובצורתימוש שונה. בנוסף על כל זה, יש את עניין הפיצ'רים הנוספים של HTML5, שעצם הוסיףם של חלק גדול מהם יוצר בעיות אבטחה.



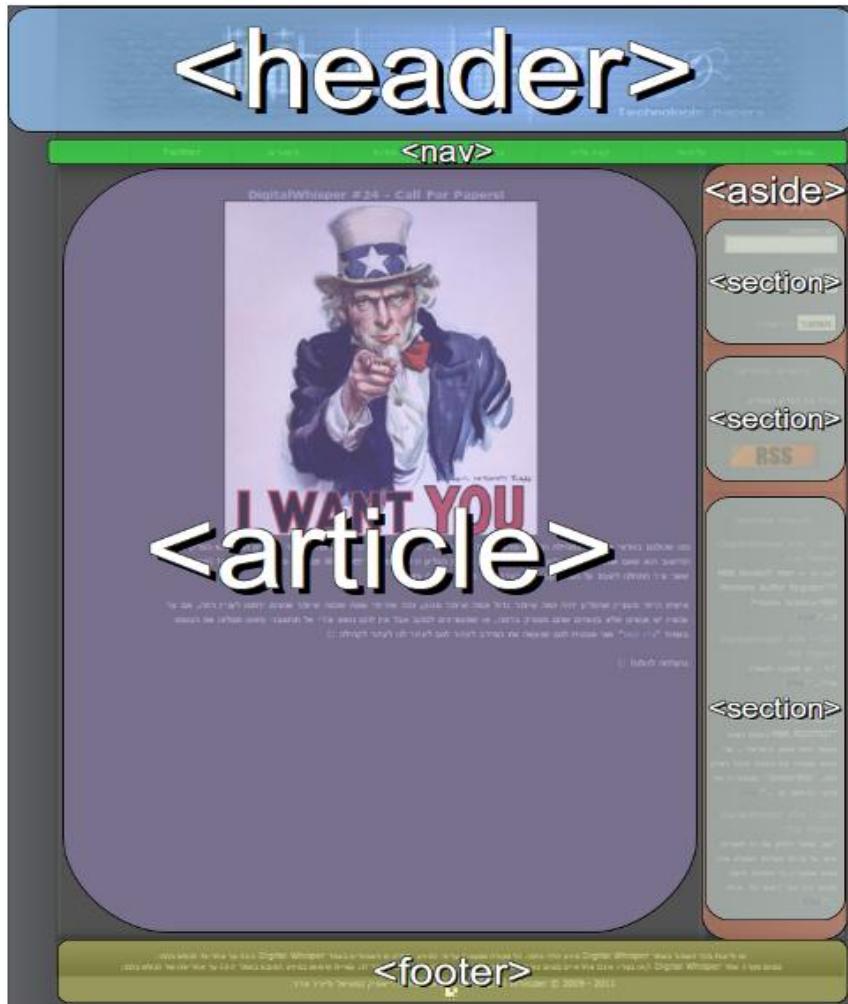
## הנתיגות החדשניות

ב-HTML5 ישן-28 תגיות חדשות, ניתן לסתוגן לסוגים:

### תגיות חלוקת הדף וטקסט

התגיות הללו לא משנות או מבצעות שום דבר בנוגע לעיצוב, אלא רק נתונים סמנטיקה עבור התוכן אותו הן מייצגות. סמנטיקה זו עוזרת מאוד למגוון חיפוש ורוביוטים.

להלן דוגמה של הפוסט האחרון ב-DigitalWhisper בעזרת התגים החדשניים:



HTML5 - מנקודות מבט אחרית  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

כמו שראים בתמונה, תג header מסמן את ראש העמוד, בעוד שtag footer מסמן את סופו (לא עוד tag div עם id המתאר את משמעות התוכן). בנוסף, ניתן לראות כי קיים tag article, המכיל את תוכן המאמר, tag nav המכיל את תפריט הניווט של האתר, וtag aside אשר מכיל את התפריט הצדדי, כאשר בתוכו תגי section המגדירים את החלקים השונים בתפריט.

לא נפרט יותר מיד על תגים אלו במאמר זה, אך אלו מכמם המתעניינים יכולים למצוא יותר מידע בקישורים בסוף הפרק.

#### תגיות מולטימדיה:

HTML5 מאפשר מספר תגי מולטימדיה המשמשים להטמעת קול, סרטונים ופלטים בעמוד.



תג הוויידאו והאודיו מאפשרים לנו להציג תוכן מקור (origin) אחר, וזה אפשר לנו להציג סרטונים אהובים מאתרים אחרים ללא כל צורך לאחסן את אותם סרטים בשרתים שלנו, או להשמע רשימות מוזיקה שלמות ומתחדשות מאתר מוזיקה מובייל. אפשרות זאת יוצרת עיית אבטחה משני כיוונים:

כאשר אתר תמים מציג תוכן זמני, וכך אשר אתר זמני מציג תוכן תמים.

מצד אחד, כאשר לעמוד תמים מציג תוכן זמני, החשש העיקרי הוא שמקור המדיה הזמני עלול להכיל סקירה כתשה אשר ינסה לבצע אינטראקציה עם תוכן העמוד ולשנות אותו. הדפננים יכולים לבצע הפרדה בין context של התוכן המוצג לבין זה של העמוד המוצג אותו, לדוגמה, אם בעל העמוד חיפז להציג אנימציית gifs בתחום tag video, על הדפן להפריד בין השתיים כך שהטג יראה את עצמו כעמוד נפרד ולא אב ולא יוכל לשולוט על ה-DOM בעמוד שהטמע את הוידיאו.

מצד שני, כאשר לעמוד זמני מציג תוכן תמים, החשש הוא שהעמוד יוכל לשולוף מידע על אותו התוכן שבכורה אחרת לא יכול להשיג. דוגמאות למידע שהעמוד הזמני יכול להציג הן: משך הסרטון\אודיו, משקל הקובץ, עצם קיומו של הקובץ, מידע נוסף על השרת המ אחסן את הקובץ ועוד.

חשיפה של מידע מסווג זה לאתר זמני היא כבר עייתי, אך קיימת בעיה חמורה עוד יותר: אם הדפן חשוף מידע נוסף על התוכן (כמו שהרבה גופים אוהבים לעשות), כמו כתוביות, רשימת כתורים או פרקים או כל מידע אחר. מידע שכזה אולי לא נראה חשוב כל כך, אבל תחשבו על התרחש הבא:

נניח כי קיימת חברת סטארט-אפ אשר עובדת על פיתוח של רעיון אשר עתיד להניב לחברה המונע כסף. במחשב החברה קיימים סרטונים אשר מציגים את שלבי הפיתוח עד כה. בכך לעדכן משקיעים ולהליב משקיעים פוטנציאליים, סרטונים יש כתוביות על מנת שגם משקיעים אשר לא דוברים את שפת הקודש

יכולו להבין שהחברה בדרך לכסף הגדול. עובד תמים מן החברה גולש לעמוד זמני מתוך אחד מחשביו החברה. בעת הגלישה לעמוד, מורץ ברקע על הדפסן סקריפט זמני אשר מציג (בגלו או בסתר) לעובד את הסרטונים הרגשיים הללו. כמובן שכעת הסקריפט יכול לראות מה אורך הסרטונים. ובאם דפסן הגולש החליט לספק קצת יותר מידע, יוכל העמוד/zמני גם לראות את הכתוביות עבור הסרטים, ואלו כבר יספיקו לו בכדי לדעת מהו מוצר הקסם עליו החברה עובדת. בנוסף לביעות אלו יש לקחת בחשבון כי מימוש המנגנון הנ"ל הינו מטלת הדפסן, בניגוד למצב ביום שטלה זו מובוצעת על ידי פלאגינים חיצוניים, בעיקר על ידי הנפוצים כמו Microsoft Silverlight ו-Adobe Flash אשר ידועים כמקורות טובים לביעות אבטחה. בעת הדפסנים יצטרכו למשתמשים עצמם, וכך נראה שנראה שובי חורי אבטחה. הדפסן יצטרך להתמודד עם מספר גדול יותר של פורמטים מאשר היה צריך בעבר (למשל mp3, mp4, avi, ...) ובין השאר יצטרך להתמודד מול התקפות עתידיות אשר ישמשו בחולשות בפורמטים של קבצי וידאו ו奥迪ו, ובגנים אשר ימשיכו על מנת להציגם.

יש לשים לב כי למרות שרוב מה שכיוםתוספים כמו Microsoft Silverlight ו-Adobe Flash עשויים בעמוד אינטרנט,icut יכול להתבצע בעזרת HTML5 ו-JS, עדין HTML5 מאפשר לנו להטמעתוספים (plugins)בעמוד באמצעות Tag ה-`embed`, בתוספים אלו יכולים להיות פרצות אבטחה בעצמם ובמקרה שכזה אלו יכולים להיות סיכון עבורנו ועבור המשתמשים.

#### tag ה-`canvas`:

tag ה-`canvas` מאפשר רינדור דינמי של תמונות בעזרת סקריפט, או במילאים אחרות, ניתן לצייר על tag ה-`canvas` בעזרת javascript.

בכדי ליצור `canvas` נכתב קוד HTML הבא:

```
<canvas id="example" width="200" height="200">
This text is displayed if your browser does not support HTML5 Canvas.
</canvas>
```

וכעת נצייר עליו בעזרת JS מלבן אדום:

```
var example = document.getElementById('example');
var context = example.getContext('2d');
context.fillStyle = "rgb(255,0,0)";
context.fillRect(30, 30, 50, 50);
```

עד כאן הכל טוב ויפה. מה הבעיה?



אם קיימים מצב בו סקריפט ממוקור (origin) אחד יכול לגשת למידע אשר נמצא במקור שונה, הסקריפט יוכל לקרוא את הפיקסלים מן ה-canvas ולדעת מהי התמונה המוצגת.

בכדי למנוע את הבעיה, החליטו ב-C3W להוסיף דגל (flag) בשם origin-clean. הדגל מתקבל ערך בוליאני, true אם כל האובייקטים המצוירים ב-canvas (או המשמשים לציר ה-canvas) נמצאים באותו origin כמו העמוד המכיל את אלמנט ה-canvas ו-false אחרת.

התנאים היותר מדויקים נמצאים כאן:

<http://dev.w3.org/html5/spec/Overview.html#security-with-canvas-elements>

גם עם הפתרון הזה, הבעיה עדין קיימת ובכל פעם שה-canvas משתמש באובייקט ממוקור אחר, סקריפט כלשהו שרצה ב-context של העמוד יוכל לדעת בדיק מה מוצג על ה-canvas. להרחבה:

<http://html5example.net/tag/WebGL%20Browser%20Exploitation>

### תגיות טפסים

ב-HTML5 נוספו כמה תגי טפסים נוספים כמו, tag ה-output אשר אמור בתחום פלט מסוים הנוצר על ידי סקריפט, וtag ה-datalist אשר מחזיקה אפשרות לבחירה והשלמה אוטומטית בתיבת טקסט. אך התג ה-*keygen* מעניין במיוחד הוא tag ה-*keygen* אשר מייצר זוג מפתחות רנדומליים.

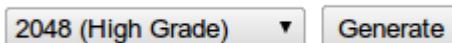
### **תג ה-keygen**

כמו שנאמר, tag זה מייצר זוג מפתחות רנדומליים המתאימים להצפנה א-סימטרית המשמשים לאונטיקציה. כאשר הטופס (שבו נמצא התג) ישלח, המפתח הפרטי ישמר ב-local keystore והמפתח הպומבי נארץ ונשלח אל השרת.

לדוגמה, נוכל ליצור את הטופס הבא:

```
<form method="POST" action="http://www.server.com">
<keygen name="pubkey">
<input type="submit" name="createcert" value="Generate">
</form>
```

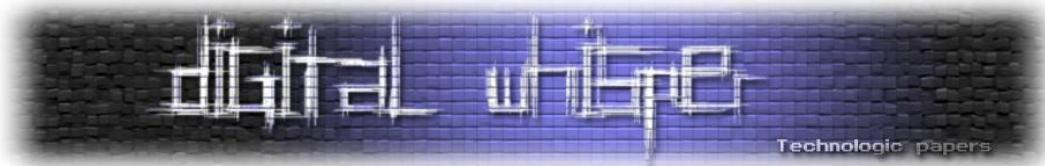
אשר הדפסן (כרום) יрендר בצורה הבאה:

 2048 (High Grade) ▾ Generate

וב-List Drop Down תהיה רשימה של keylength הנתמכים על ידי הדפסן.

HTML5 - מנוקדת מברח

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



כאשר נלחץ על **Generate**, הדף ייצור שני מפתחות (פרטי וציבורי) ואת המפתח הפומבי ישלח לשרת <http://www.server.com>.

לtag זה יש מספר תכונות:

- תכונת **challenge** אשר מקבלת מחרוזת (אם לא מוגדר, מקבל ערך של מחרוזת ריקה).

- תכונת **keytype** אשר מדגירה את האלגוריתם ששימוש ליצירת את המפתח (כרגע אין רשימה של סוגי הצפנות שלו) דפינים למשם, אך שדה זה יכול להכיל לדוגמה את הערך "RSA"

המפתח הפומבי ומחרוזת **challenge** מキודים בעזרת DER ונקראים **PublicKeyAndChallenge**. לאחר מכן, ערך זה נחתם בעזרת המפתח הפרטי בצד- ן ליצור ערך הנקרא **SignedPublicKeyAndChallenge**.

משהו נוסף שייהי מעוניין לראות הוא כמה רנדומלי יהיה רצף הבטים הנוצר. גם בעבר נראה אלגוריתמים אשר כביכול נתנו רצף בתים רנדומלי, אך לבסוף התגלה כי הרנדומליות לא כל כך רנדומלית.

בנוסף לתגים, ישנו כמה ערכים נוספים אשר ניתן לשים בתכונת **type** של tag ה-**input**, למשל: **search**, **url** ועוד. אלו משפיעים בעיקר על תצוגת שדה הקלט ועל אימות תאימות הקלט לסוג השדה על ידי הדף.

כל התוספות הללו לא מהוות בעייה אבטחה מבחינה עקרונית, אך ברגע שכולן ימומשו בדפינים יכול באווד להיות שצורת המימוש תהייה פגומה. בדיקות תאימות קלט הן דבר מסוובר, ועבור זמן עד שנוכל לחתם לדף לבצע זאת עבורינו בעניינים עצומות. גם אז נדרש לחשב על כל הדפינים ולא רק על אחד שהשימוש שלו נכון. כמוון זהה לא חוסך לנו את הבדיקות מצד השירות. משתמש זדוני יכול בקלות לעקוף בדיקות מצד הקליינט, וגם משתמש שאינו זדוני ומשתמש בדף ישן אשר אינו מבצע בדיקות אלו בצוואה טובה או בכלל לא, ואף כלים אוטומטיים למילוי טפסים יכולים לשולח לשרת מידע אשר לא אומת.

אם כן, אסור להסתמך על בדיקות אלו לא מצד הקליינט (אם כי זה פחות קרייטי), ובטע שלא מצד השירות!  
**מנגנוני הבדיקה הללו נועד אך ורק כדי להסיק לחוויות המשתמש ולא מנגן אבטחה.**

את רישימת התגים החדשים של HTML5 והסביר עליהם ניתן לראות בلينקים הבאים:

- [http://www.w3schools.com/html5/html5\\_new\\_elements.asp](http://www.w3schools.com/html5/html5_new_elements.asp)
- <http://www.htm.co.il/2009/10/26/html-5-%D7%9E%D7%94-%D7%A0%D7%9B%D7%A0%D7%A1-%D7%95%D7%9E%D7%94-%D7%99%D7%95%D7%A6%D7%90-%D7%97%D7%9C%D7%A7-%D7%90/>

---

HTML5 -

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

### **התכונות החדשנות:**

ב-HTML5 ישן תכונות גלובליות אשר חלות על כל התגים, כמו תכונת ה-`contenteditable` אשר קובעת כי המשמש יכול לעורר את תוכן התג.

כמו כן ישן גם תכונות חדשות עבור Tags מוכרים כמו תכונת `autofocus` אשר בעת מיידי טופס עשו פוקואוטומטי על השדה הרצוי, תכונת `placeholder` אשר בעת מיידי טופס תראה לנו דוגמה למה-Amor להיות באותו השדה, תכונת `required` אשר מגדירה כי שדה מסוים בטופס הינו שדה חובה, תכונת `pattern` תגרום לדפדן לבדוק שהערך בשדה ה-`input` שעבورو ישמו את התכונה יתאים לביטוי הרגולרי שקיבלה התכונה כערך.

בנוסף, ב-HTML5 לא חייבים לתת לכל תכונה ערך, וכאשר כתבים ערך שהוא ללא רווחים, לא חייבים לשים יותר גרשימים (למרות שהוא נראה טוב יותר עם גרשימים). ניתן למשל לכתוב את התג הבא:

```
<input type="email" placeholder="hello@mail.com" autofocus required />
```

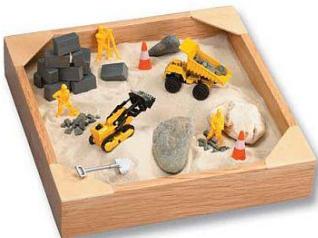
דבר נוסף שחלקכם ימצאו שימושי הוא תכונות מותאמות אישיות (Custom Attributes), זאת אומרת שאם יש תכונה שהייתם רוצים להוסיף והיא לא קיימת ב-HTML5, אתם יכולים לכתוב אותה ולא תיגרם כל שגיאה. ניתן לאחר מכן להשתמש בה ב-JS ו-CSS. זהה דרך מצוינת להכניס עוד מידע הרלוונטי לעמוד אך לא נמצא לו מקום מתאים. דוגמה טובה לשימוש בה היא ליצור תכונת `explanation` המכלילה הסבר על המושג שהtag שהוא נמצא בו תוחם, ובעזרת JS להציג את אותו הסבר כאשר עוברים על המילה. כך ההסבר יוכל ב-HTML5 אשר-Amor להכיל את התוכן.

כמובן שגם את התכונות המותאמות ניתן לנצל אם קיים מצב כמו בדוגמה. המנגנון אשר מדפיס את ההגדרות של המושגים משתמש ב-`document.write`, ומעביר לו כפרמטר את ערך ההגדירה.icut נניח אפילו שיש IPS, IDs או כל מנגנון אחר אשר מסנן כל תכונה קיימת ב-HTML5 וגם ב-HTML5 כדי שלא נציג להחדר שום תכונה או מאורע, אשר ישמשו להרצת סקריפט, לתוך התג הנ"ל. אם המנגנון לא עובר התאמה לתכונות חדשות שבעל האתר הוסיף (זה משווה שקל יחסית לפוסוף), נוכל להזיר את אותה התכונה למקומות החשופים ל-SAJ (כמובן שהם נבדקים על ידי המנגנון ולכן בעל האתר חושב שהם בטוחים), ולגרום להרצת סקריפט.

### **tag ה-`iframe`:**

tag ה-`Iframe` הינו tag המאפשר להטמע עמוד אינטרנט אחר ישירות בתוך האתר. קיימות הרבה בעיות אבטחה בדבר זה. חלקן מן הניסיון לפתור בעיות אלו, וגם בכך להוסיף לפונקציונליות של ה-`iframe`, הוסיףו לתג ה-`iframe` כמה תכונות מעניינות.

## תכונת ה-sandbox



תכונת ה-sandbox היא תכונה שימושית ביותר. כאשר נוסיף תכונה זו לTAG iframe, הדפדן בעצם יוריד את הרשות של הדף זהה וימנע ממנו לבצע פעולות מסוימות שעשויה לסקן את המשתמש ואת האתר עצמו. תכונה זו יכולה לקבל ארבעה ערכים:

- allow-forms - מאפשר שליחת טפסים מן הדף המוטמע.
- allow-same-origin - מתייחס לתוכן הדף אליו הגיע מן הדומיין של האתר המתמיע (בין השאר מאפשר לסקרייפטים בדף המוטמע לגשת לאלמנטי DOM, לעוגיות ו-localStorage הנמצאים בדומיין של הדף המתמיע).
- allow-scripts - מאפשר הרצת סקריפטים בדף המוטמע (למרות שעדין הסקריפטים אינם יכולים להקפיז והודעות pop-ups).
- allow-top-navigation - מאפשר לינקים ולסקרייפטים לנוט את העמוד המתמיע (למשל, בזמן טעינת הדף לקשר את העמוד העליון לעמוד פישינג).

בנוסף, במקרה שלא מכנים אף אחד מן הערכים הללו, אלא נתונים ערך ריק, הדפדן יחסום את כל הפעולות הנ"ל. זאת אומרת שכאשר הערך הוא ריק, הדפדן יראה את העמוד המוטמע בתור עמוד מדומיין אחר, שליחת טפסים והרצת סקריפטים מן הדף המוטמע לא תתאפשר, לינקים לא יוכל לנוט את העמוד המתמיע ותוספים אחרים (plugins) לא יכולו לזרוץ.

חדי העין (וחדי הקרון) בודאי שמו לב שבמקרה ובאתר המוטמע מוגדרים גם allow-same-origin, וגם allow-scripts וגם העמוד המוטמע ב-frame הוא מאותו מקו, האתר המוטמע יכול בקלות לגשת לדף המוטמע ולבטל בעזרה סקריפט את תכונת sandbox ולבטל כל הגנה שזו סיפה, קוד לדוגמה:

### inner.html:

```
<script>
top.document.getElementsByTagName("iframe")[0].attributes.removeNamedItem("sandbox");
</script>
```

### outer.html:

```
<iframe src="http://127.0.0.1:80/inner.html" sandbox="allow-same-origin
allow-scripts">
</iframe>
```

### תכונת ה-srcdoc

תכונת ה-srcdoc מאפשרת למפתח להכניס קוד HTML של האתר המוטמע. זאת אומרת, במקום לחת את הילינק לדף שברצוננו להטמייע ולגרום לדפדף לבצע בקשה עבור התוכן של דף זה, ניתן לבצע את הבקשתה בעצמינו ולהכניס ישירות את קוד ה-HTML.

במחשבה ראשונה הרעיון נשמע קצת מוזר, כי הרי באותו אופן יכולנו להעתיק את קוד המקור של העמוד לתוך העמוד שלנו. אך יש לזכור כי חלק מהתקיד של תג ה-*Iframe* הוא למנוע גישת תוכן לא מורשה לתוך העמוד שלנו (בין השאר בעזרה תכונת ה-sandbox).

במידה ולtag *Iframe* קיימות גם תכונת ה-src וגם תכונת ה-srcdoc תמיד תהיה עדיפה לתכונת ה-src. רק במצב של דפדף מיושן אשר לא מכיר את תcona זו יהיה שימוש בתכונת ה-srcdoc.

### תכונת ה-seamless

פתחים רבים משתמשים בתagi HTML עם סמנטייקה מאוד ספציפית עבור תוכן לא מתאים בטענה ש'זה נראה יפה'. תכונת ה-seamless היא תcona בוליאנית שמעלימה את הגבולות של תג ה-*Iframe* וגורמת לו להיראות כחלק מן העמוד.

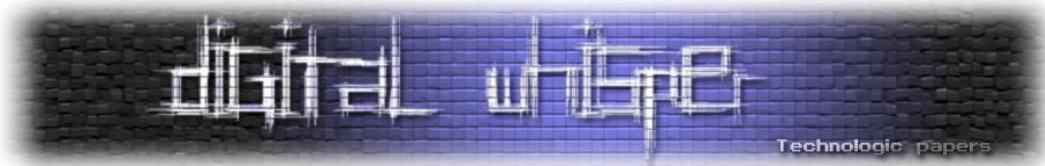
ניתן להשתמש בה بصورة הבאה:

```
<iframe seamless>
```

אך זה לא כל מה שתcona זו עשויה. התcona לא רק גורמת לדף המוטמע להיראות כחלק מהעמוד, היא גם גורמת לדפדף לרendar את הדף המוטמע כחלק מהעמוד!

כמו דוגמאות להתנהגות הדפדף במקרה שהתcona מופיעה בתag *Iframe* מסוים:

- לינקים בתחום ה-*Iframe* ינוטו מחדש את הדף כולו ולא רק את ה-*Iframe* (אלא אם ב-*target* צין *self*\_).
- ה-*css* של הדף המוטמע ישפייע לא רק על ה-*Iframe* אלא יצורף גם אל ה-*css* של העמוד המטמייע וישפייע על העמוד כולו.
- רוחבו של ה-*Iframe* יהיה זהה לזה של כל אלמנט אחר באותה רמה ובעל רוחב אוטומטי.
- גובהו של ה-*Iframe* יהיה גובה של מסגרת התוכן המוטמע בתוכו (Bounding Box) בהינתן הרוחב שהוגדר בנקודת הקודמת.
- כאשר תוכנת נגישות כמו קורא קולי יקרא את העמוד, הוא יקרא גם את תוכן ה-*Iframe* בלבד להכריז על כך שמדובר בדף נפרד.



- הדף מתייחס אל הדף המוטמע אליו הוא חלק מן הדף המטמייע, גם כאשר גריץ סקריפט. לדוגמה, אם בדף המטמייע יהיה סקריפט אשר סופר את מספר הلينקים בעמוד, גם הلينקים אשר נמצאים בתוכן של הדף המוטמע יספרו.

### **בקשות בין מקורות שונים (Cross-Origin Requests - XDR) (CORS)**

בימים שלפני HTML5, בקשות AJAX הוגבלו על ידי "Same Origin Policy", אשר מאפשרת בקשות עبور משאב מסוים להתבצע רק בתחום אותו דומין, זאת אומרת שהעמוד `http://www.site.com/index.html` אשר אמרת שהוא תחת `http://www.site.com`. ב-HTML5 ניתן לבטל הגבלה זו, וכך יוכל לבצע בקשות רק עבור משאים הנמצאים תחת `http://www.site.com`.

למה לבטל את הגבלה? זהו לא משהו חדש, כבר היום מפותחים אתרים יוצרים מנגןונים משלהם על מנת לאפשר בקשות בין מקורות, על מנת לספק למשל APIs ולנצל יותר מן הפוטנציאל של הרשת (תחשבו על מערכות התגובה של פייסבוק המאפשרת הטעמה אתרים שונים).

#### מה חדשי?

בקשות למשאים מקורות אחרים יכולים להתבצע גם עוד לפני HTML5, הרי עוד הרבה לפני-XAJAX וה-HTML5 דפים היו מבצעים בקשות עبور משאב הנמצא בדומין אחר, אם על ידי `img`, `script` או `iframe`.

אך בבקשות מסווג זה לעמוד לא הייתה גישה לתוך המשאב. הדף רק הרץ את המשאב או רינדר אותו והציג אותו בעמוד.

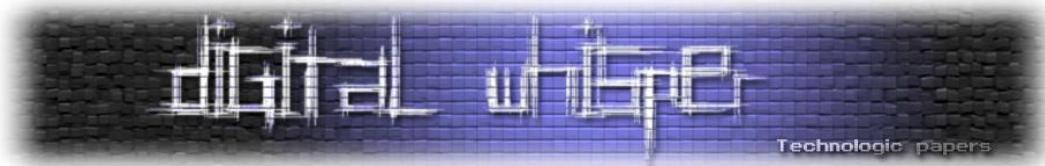
הבדל הוא ש-XDR מאפשר ל-`JavaScript` לקרוא את תוכן המשאב על ידי תכונת `'responseText'`.

#### איך זה פועל?

XDR הוא מעין מפרט על שmagdir איך לאפשר בקשות Cross-Origin אל לעמוד מסוים. מפרט זה מגדיר מספר כותרים (headers) שנitin להוציא לקשות ותשובות של HTTP.

#### **הcottrim shamogdrut b-XDR**

XDR מגדיר מספר כותרים, חלקם יכולים בבקשתות מן השרת וחלקם בתחום תשובות הרשת בקשות אלו. אנו נסביר כאן רק על הכותרים העיקריים:



### Access-Control-Allow-Origin

Access-Control-Allow-Origin נמצאת בכותר של תגובה מן השרת, ומגדירה לאילו דומיינים מותר לבקש בבקשת Cross-Origin את המשאב.

התחבר הוא:

Access-Control-Allow-Origin = \* | רשות דומיינים =  
(כאשר התו '\*' יגרום לכך שכל דומיין יוכל לבצע את הבקשה.)

זה נראה היה header הביעתי ביותר. הרבה מפתחים ישימו כערך '\*' ובכך בעצם יצרו דלת פתוחה לכל מי שרצה לבצע XDR.

### Access-Control-Max-Age

Access-Control-Max-Age נמצאת בכותר של תגובה מן השרת, ומגדירה לכמה זמן התגובה לבקשת preflight (IOSVER בהמשך) תישמר ב-cache.

התחבר הוא:

Access-Control-Max-Age = מספר שנים

### Access-Control-Allow-Methods

Access-Control-Allow-Methods נמצאת בכותר של תגובה מן השרת, ומגדירה באילו שיטות מותר לבקש בבקשת Cross-Origin את המשאב.

התחבר הוא:

Access-Control-Allow-Methods: GET/POST/OPTIONS/DELETE/...

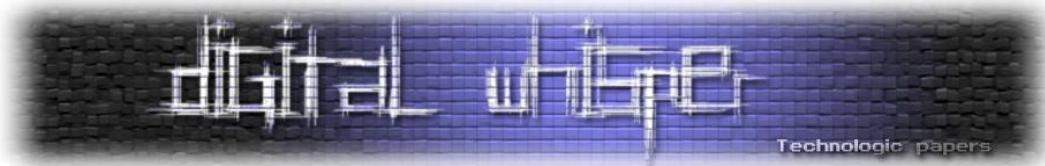
ניתן להגדיר גם כמה שיטות על ידי רשות שמן עם רווח ביניהן.

### Access-Control-Allow-Headers

Access-Control-Allow-Headers נמצאת בכותר של תגובה מן השרת, ומגדירה עם אילו custom headers מותר לבקש בבקשת Cross-Origin את המשאב.

התחבר הוא:

Access-Control-Allow-Headers: custom headers names



### Access-Control-Allow-Credentials

התשובה לבקשת יסודה **Access-Control-Allow-Credentials** נמצאת בcotor של תגובה מנשותה, ומגדירה האם האם אישר דגל ה-Credentials (ויסבר בהמשך) דלוק.

כאשרcotr זה חוזר עם תשובה לבקשת preflight, הוא מגדיר האם הבקשת האמיתית יכולה להישוט אישר דגל ה-Credentials דלוק.

התחבר הוא:

```
Access-Control-Allow-Credentials: true | true: %x74.72.75.65
```

### Origin

נמצאת בcotor של בקשה לשרת, ומגדירה מייזה דומיין נעשית הבקשת.

### Access-Control-Request-Method

נמצאת בcotor של בקשה לשרת, וכאשר הדפדן מבצע בקשה preflight, הוא משתמש בcotor זה על מנת להגיד באיזו שיטה הוא ירצה לבצע את הבקשת האמיתית.

### Access-Control-Request-Headers

נמצאת בcotor של בקשה לשרת, וכאשר הדפדן מבצע בקשה preflight הוא משתמש בcotor זה על מנת להגיד עם אילו custom headers הוא ירצה לבצע את הבקשת האמיתית.

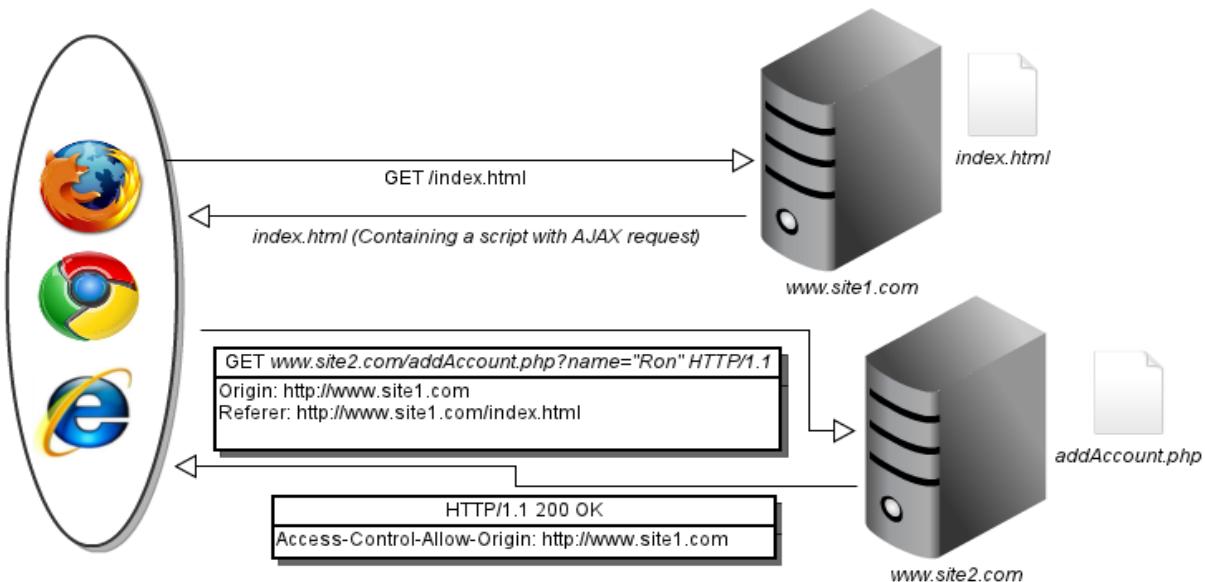
### שימוש ב-XDR

ניתן להשתמש במפרט ה-XDR בעמוד http://site2.com/addAccount.php בכך לאפשר לעמוד <http://site1.com/index.html> ולעומודים נוספים לבצע בקשה Cross-Origin אליו בצורה הבאה:

בכך לעשות זאת נוסיף את header:

```
Access-Control-Allow-Origin: http://site1.com
```

لتשובה שהשרת של <http://site2.com> מוחזיר בעת בקשה הדף addAccount.php.



בעת ביצוע בקשה מן העמוד: `http://site1.com/index.html` לעמוד: `http://site2.com/addAccount.php` יבצע את הצעדים הבאים:

1. יבצע בקשה לעמוד `http://site2.com/addAccount.php`
2. יבדוק את ה-headers של התשובה שהוא קיבל חזרה, ויחפש אישור לבקשת Cross-Origin מן הדומיין `http://site1.com`
  - a. אם הדפדפן מצא header אשר מגדיר שלדומיין זהר מותר לבצע את הבקשה זהו, העמוד יוכל לקרוא את תוכן התגובה.
  - b. אחרת, העמוד לא יוכל לקרוא את תוכן התגובה.

ניתן לחשב שסדר זה בעיתוי, שהרי הדפדפן קודם מבצע את הבקשה ורק אז לפי תגובת השרת מודוד אם מותר לו לבצע את הבקשה. אבל לעיתים עצם בקשת עמוד מסוים עלולה לגרום לפעולות לא מאושرات מצד השרת.

אך זהו לא באמת סיכון מכון שגם בצד שרת ניתן לבדוק מיهو הדומיין אשר מבצע את הבקשה על ידי קרייה של ה-`origin` header שם, בכל בקשה שנעשית, הדפדפן מכניס כערך את הדומיין אשר ממנו הzbתעה הבקשה.

כך, במקרה שהשרת זיהה שדומיין לא מורשה ביצוע את הבקשה, הוא יכול להחזיר עמוד ריק במקום התוכן המקורי ולא לבצע שום פעולה.

### בקשת preflight

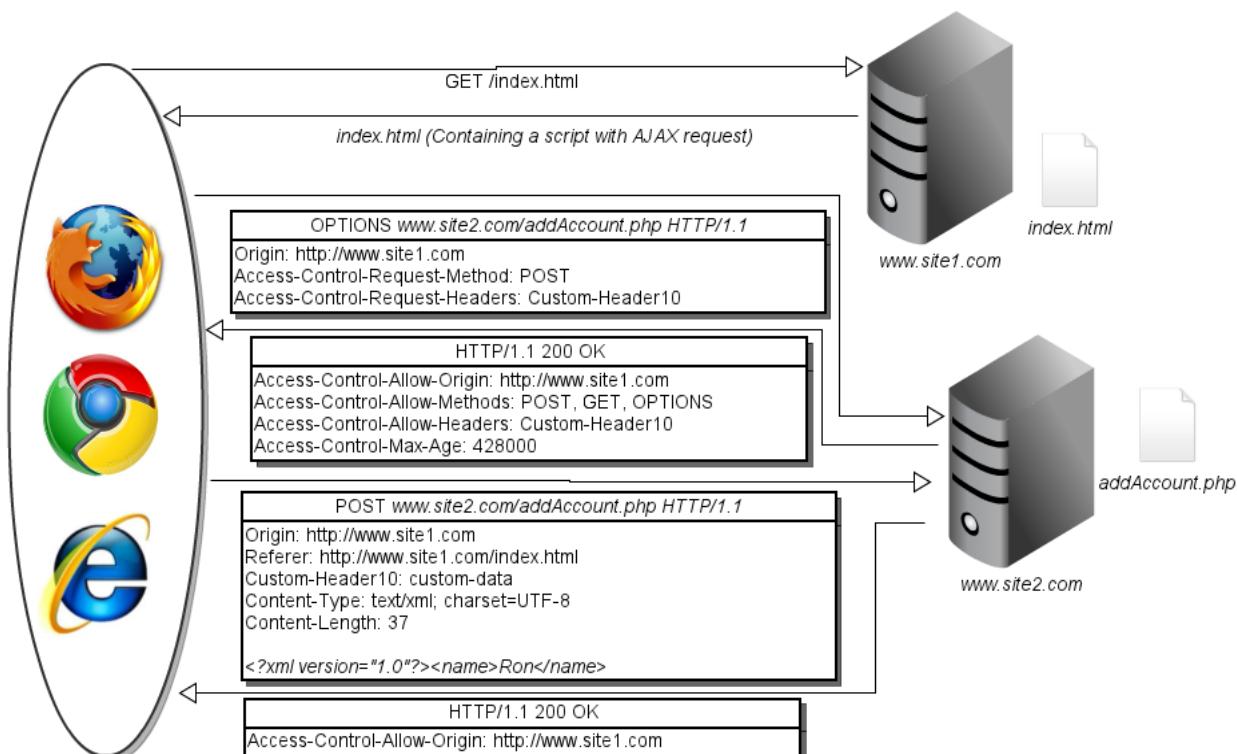
בנוסף לבקשתות הרגילות, קיימת גם preflight request, אשר קודם שלוחת בקשה OPTIONS בהתחלה כדי לבדוק האם לדומיין מותר לבצע בקשה Cross-Origin, ורק אחר כך תבצע הבקשה (בתנאי של-domain מותר לבצע אותה).

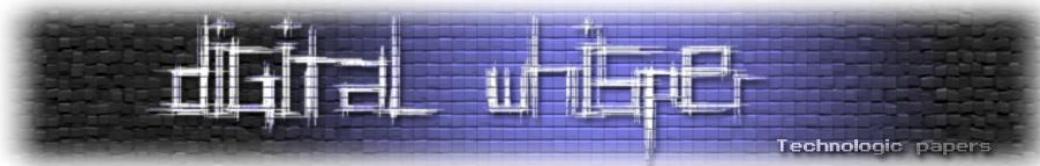
מגנון זה קיים מכיוון שלפעמים עצם שליחת הבקשה אינה בטוחה.

הדף מבצע את בקשה preflight אוטומטית, ללא צורך בקוד נוסף אם מתקיים:

- הבקשה נעשית בשיטה אשר שונה מ-GET או POST, וגם אם הבקשה נעשית בשיטה אשר שונה מ-POST כאשר ה-Type הנשלח יהיה שונה מ: application/x-www-form-urlencoded, או בעברית: text/plain, או multipart/form-data עם Content-Type של xml או application/xml יבצע את הבקשה בעזרתו.(preflight

- ישנו custom header בבקשתה (לדוגמה, ב-header יש שדה בשם "Custom-Header10")





בעת ביצוע בקשה מן העמוד <http://site1.com/index.html> לעמוד <http://site2.com/addAccount.php> הדף יבצע את הצעדים הבאים:

1. הדף ישלח בבקשת Option לעמוד <http://site2.com/addAccount.php> כאשר בנוסף הוא מצורף את הכותרות `Access-Control-Request-Headers`-`I` `Access-Control-Request-Method`
2. כעת, אם השירות מאשר לדמיין הזר לבצע את הבקשה עם הפרמטרים הללו הוא יענה בתגובה `OK`.
3. אם השירות אישר לדף ל�行 את הפעולה, הוא ישלח כעת את הבקשה המקורי.
4. השירות יקבל אותה וילח בחרה את תוכן העמוד.

### דגל ה-Credentials

דגל זה מסמן האם ה-Credentials של המשתמש יצורפו לבקשתה.

המ: עוגיות, אוטנטיקציות HTTP שנעשה, I-Certificates של הקליינט עבור SSL.

בברירת המחדל, דגל ה-Credentials נקבע כבוי. כלומר, הדף בברירת המחדל אינו מצורף עוגיות לביקשות שנעשו ב-XDR אלא אם הוגדר אחרת, ובמקרה שכזה הוא שולח את הכותרות המתאימות אשר שואלוות את השירות האם זה בסדר מבחינתו. במקרה שכן, ה-Credentials נשלחים עם הבקשה. אחרת הבקשה לא מתבצעת.

כותרת ה-`Access-Control-Allow-Origin` לא יכולה להיות '\*' כאשר דגל ה-Credentials דלוק.

### בעיות אבטחה של XDR

#### אפשרות כללי

אפשרות כללי זו היא הטעות כי ברורה וכי קלה שפתחים יכולים לעשות. כמו שכבר אמרנו, הכותרת 'Access-Control-Allow-Origin' אמורה להכיל רשימה של הדומיינים אשר יכולים לבצע בקשה Cross-Origin לעמוד, אך היא יכולה גם לקבל ערך wildcard ('\*') אשר מאפשר לכל אתר לבצע בקשות Cross-Origin. ניתן לנצל זאת במספר דרכים:

- אם קיים איזשהו דף השיר לרשות פנימית (intranet) של ארגון כלשהו ואין אליו גישה מבוחוץ. עובד בארגון אשר גולש בעთות לאתר זドוני מאחד מחכבי החברה המחברים לרשות הפנימית (וגם לחיצונית) עלול לחשוף את המידע ברשות הפנימית לאתר הזדוני, אשר האתר הזדוני יבצע

בקשת XDR לעמוד ברשות הפנימית, וכך יוכל לקרוא את המידע בו ולשלוח אותו לבעל האתר הצדוני.

- מקרה דומה הוא למשל אם גוגל מאפשרת אופציה נוספת המכפשים בשירות החיפוש שלה (למשל מחיקת תוצאות מן החיפוש). אך אופציה זו קיימת רק כאשר מכפים מתוך הרשות הפנימית. נניח בנוספ שגוגל מאפשרת לכל אתר לעשות לה XDR. כאשר עובד גוגל יגלוש לאתר הצדוני, האתר הצדוני יוכל לבצע בקשת XDR ולמחוק רשותות מן תוצאות החיפוש של גוגל.
- נניח שמצאנו באתר מסוים SQL Injection בעמוד שניtan לבצע אליו XDR מכל דומיין. אם נשלוף את כל המידע בעמוד עצמו, הלוגים של הרשות יצבעו עליינו באופן ברור וניתפס. אך נוכל ליצור קוד JS אשר מבצע את השליפה ושולח את התוצאות אליהם באמצעות XDR. את הקוד JS נשים באתר שלנו או שנעביר אותו דרך האתר עם XSS אל משתמש כלשהו. כאשר קורבן יבקר באתר בו שמננו את ה-SL, הדף殿ן שלו ישולף את הנתונים מן האתר וישלח אותם אלינו. בדיקות בלוגים של האתר יראו כי המשתמש הוא זה שביצע את השליפה, מכיוון שכותרות של בקשות בדרך כלל לא נשמרות בלוגים. הקורבן לא יוכל לטען שהמחשב שלו נפרץ, מכיוון שבדיקה על מחשב הקורבן לא תראה עקבות של malware או כל צורת השתלטות אחרת.

#### Include שגוי

נניח כי קיים מצב שבו לשרת יש קובץ common.php.kbh לדוגמה, אליו הוא אפשר בקשות XDR, והמן קבצים בשרת עושים לקובץ זה include. עשוי להיווצר מצב שעמוד אשר לא התכוון לתת הרשות לבצע אליו בקשות XDR עשו include לקובץ זה, אשר בתורו מוסיף את הכותרים שמאפשרים בקשות XDR, וכעת יהיה ניתן לבצע בקשות XDR אל העמוד וגם אל כל עמוד אחר אשר יבצע include לקובץ זה.

פתרון אפשרי לבעה זו הוא לבצע בדף common.php בדיקה מול "מילון" המכיל מידע על דפים אליהם אנואפשרים בקשות XDR, ועל פי "מילון" זה לשלוח את הבקשות המתאימות, ולא להסתמך על כך שעשו include לקובץ זה.

#### הסתמכות יתר על כותרת ה-Origin

יש לציין שכותרת ה-Origin נוספת על ידי הדף殿ן, וכל אחד יכול להנדס בקשות-HTTP משלו עם כותרת Origin שיכיל כל ערך שירצה.

לדוגמה, הקוד הבא מכיל פרצה:

```
<?php
    if($_SERVER['HTTP_ORIGIN'] == "http://intranet.andlabs.org"){
        header('Access-Control-Allow-Origin: http://intranet.andlabs.org');
        //perform some important action
    }
}
```

HTML5 -  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

```
        print >>> sensitive internal information <<< ;
}
else{
    print >>> normal page <<< ;
}
?>
```

תוקף יכול בקלות לזייף את כותרת Origin שלו, ובעזרת החולשה לבצע פעולות חשובות ולקבל גישה למידע רגיש.

#### שמירת preflight cache לתקופות ארוכות

לביצוע בקשת prefight יש overhead, ולכן ביצוע הבקשה לוקח יותר זמן מאשר בקשת RDRX רגילה. לכן יש אפשרות לשמור את תוצאת ה-preflight ב-cache, וכך שבעם הבאה שנרצה לבצע את אותה בקשה לא נדרש לבצע את בקשת-preflight שוב ונוכל לפעול על פי הנתונים שיש ב-cache. שמירת נתוני ה-preflight ב-cache נעשית על ידי הוספה הcotract 'Access-Control-Max-Age' לתשובה השרת לביקורת ה-preflight.

שמירה של cache לזמן ארוך יכולה לגרום לביעית אבטחה. נניח שהשרת עדכן את מדיניות הגישה אליו. דפדףים ישתמשו ב-cache שלהם המכיל את המדיניות הישנה ולא במדיניות החדשה. הזמן המומלץ הוא 30 דקות.

#### מתן אמון יתר בשותף

ישנו שני צדדים ב-RDRX. הצד ש牒קש את המידע והצד שמוסר את המידע. נדרש להיות כמהות מסוימת של אמון בין הצדדים. האתר המ牒ק שומר עלacr שהמידע שיקבל זהה הוא אמין ובטוח, בעודו השומר את המידע שומר עלacr שהצד השני מוסמך לבצע את הבקשה ולקבל את המידע.

גם אם שני הצדדים הם לגיטימיים (וגם אם אותו אדם כתב את שנייהם), יש להפקיד את כמהות האמון המינימלית במידע שמספק הצד השני. זאת אומרת שעדין علينا לבדוק שהמידע אינו מכיל סקרים פוטיים זדוניים, מידע מטענה (כאשר אפשר) וכו'. זאת מכיוון שגם הצד השני לגיטימי, יכול להיות שתוקף הצליח לקבל אליו גישה וכעת משתמש בו בכדי לתקוף אותנו.

כעת נביא דוגמה דמיונית למקרים כאלה: נניח ש-DigitalWhisper היה מציג את עדכוני הטוויטר של תוכנית החדשנות אהובה עליו בעמוד הראשי.

אתר DigitalWhisper יבצע בקשות RDRX אל טוויטר בכדי לדעת מה היו העדכנים האחרונים, וכמו כן שטוויטר מאפשר ל-DigitalWhisper לבצע בקשות אלו, שהרי אלו בקשות לגיטימות. טוויטר בחזרה ישלח את העדכנים האחרונים שתוכנית החדשנות פרסמה בקידוד HTML. בין הצדדים יש אמון, ולכן

DigitalWhisper לא מבצע בדיקות קלט ופושט מדפס את העדכנים לעמוד וטוויטר מאפשר לך לבצע מגוון רחב של פעולות.

#### ההריש מס' 1 - לתוכף יש שליטה על טוויטר

כמו שאמרנו, מכיוון ש-DigitalWhisper סומר על טוויטר יש לה מידע מקודד HTML הוא מכנים את המידע לעמוד ללא בדיקות קלט.Cut, מכיוון שטוויטר בשליטת התוקף, הוא יכול לשלו מידע זדוני המכיל תגיוט HTML, וסקרים אחרים אשר DigitalWhisper יציג למשתמשים ובכך יסכן אותם.

#### ההריש מס' 2 - לתוכף יש שליטה על DigitalWhisper

נניח שטוויטר סומר על DigitalWhisper ולכז נתן לו גישה למגוון רחב של אופציות. DigitalWhisper יכול לקרוא ציוצים, לשנות ציוצים, להוסף משתמשים, למחוק משתמשים וכו'. Cut, מכיוון ש-DigitalWhisper בשליטת התוקף, הוא יכול לבצע בקשות זדוניות על שירות טוויטר בשם DigitalWhisper.

חשוב מאד שהצד המבקש יודא את תקינות התגובה שהוא קיבל, והצד המוסר יחשוף רק את האפשרויות הכרחיות עבור הצד השני. גם אם תוכף קיבל שליטה על צד אחד, אין זה אומר שהוא צריך לקבל שליטה על הצד השני באופן אוטומטי.

#### DDoS בעזרת XDR

יש לציין כי קיימים שני סוגי בקשות XDR: בקשה פשוטה ובקשה עם preflight. ישנו רק כמה מקרים בהם משתמשים ב-preflight, ורוב הבקשות משתמשים במנגנון הפשוט.

חכیر, במנגנון הפשוט אנו שולחים בקשה לשרת, מקבלים את המידע ואיתו את הכותרים, אנו בודקים על פי הכותרים האם לדמיין אשר יצר את הבקשה מותר לראות את המידע. אם כן, ניתן לו גישה, אחרת לא.

לעומת זאת, במנגנון ה-preflight ישלח בקשה Options עם הכותרים המתאים, כך שהשרת ידע באיזו שיטה ועם אילו כותרים הבקשה האמיתית תבוצע, מקבל תשובה מן השרת עם הכותרים המתאים ובודק האם הבקשה המקורי יכולה להתבצע. נזכיר גם שבעת בקשת XDR פשוטה (לא preflight) השרת מבצע את כל החישובים הנחוצים בכך ליצור את התגובה (ומתייחס לבקשת בקשה לגיטימית). זהו רק הדיפден אשר מחליט האם הבקשה היא באמת לגיטימית או לא.

ניתן להשתמש בזה בכך לבצע התקפות DDoS בשכבות האפליקציה.

נניח לדוגמה שקיים עמוד המבצע פעולות חישוביות רבות ולבסוף שלוח בחרזה את תוצאות חישובי. בעת ביצוע XDR לעמוד, העמוד יבצע את החישובים וישלח את התוצאות, רק לאחר מכן הדפסן יחליט האם לאפשר לעמוד שיבצע את הבקשה לראות את תוכן התשובה או לא. אך זה כבר לא משנה, מכיוון וגרמננו לביצוע הפעולות על מחשבי השרת.

כעת נוכל לחפש עמודי אינטרנט המכילים XSS Persistent ולהזrik לתוכם קוד JS אשר מבצע את בקשת ה-XDR הזו.

בעוד אנשים יכנסו לאתרם אלו, הדפסנים שלהם יבצעו את בקשות ה-XDR, ובכך בעצם יעמיסו על מחשבי האתר.

הינו יכולים להשיג את אותה התוצאה בעזרת החדרת Tag של תמונה בעל תכונת src מתאימה לעמודי האינטרנט המכילים XSS Persistent, אך ב-XDR נוכל להשתמש גם במקרה של בקשת POST/DELETE/PUT iframe. הינו יכולים גם להטמייע עם src מתאים או לשולח טופס (כאשר צרכים לשלוח את הבקשה ב-POST לדוגמה) מתוך iframe, אך XDR נותן ביצועים טובים יותר, בעזרת XDR ניתן לשלוח כ-100,000 בקשות בדקה (בכרום וספארי).

פתרון חלקו לבעה זו הוא לבצע בדיקות על כוורתת ה-Origin בצד השרת, כאשר ה-Origin אינו אתר שאנו רוצים שיבצע לנו XDR לא נבצע שום עיבוד בצד השרת. יש לזכור כי למוראות שהתקוף יכול לד"ר בעצמו את כוורתת ה-Origin, הוא לא יכול לעשות זאת אצל משתמשים תמיימים הנכנסים לאתר הזרוני, וכן, מכיוון שב-DDOS עסקנו, כל עוד אם התקוף בעצמו ישלח את הבקשות עם ה-Origin המזמין זו כבר לא תהיה מתקפת DOS, לפחות לא מסוימת (יכול להיות שהתקוף יש כמה מחשבים) אלא מתקפת DOS רגילה (או DDOS קטנה) שספק שתשפייע על השירותים שלנו.

בנוסף, פתרון זה רק מנסה علينا לנצל את הבעיה, שכן אם נשלחות בקשות רבות מיד' גם רכיבי הרשת בדרך אלינו יהיו עומסים (כמו Switches, Routers, וכו').

### אפשרויות ניהול חדשות למתkopft ישנות

למרות שחלק מהתוספות החדשנות ב-HTML5 אין גוררות בעיות אבטחה חדשות, הן יוצרות וריאציות חדשות לביעיות ידועות.

### באמצעות תגים חדשים

מנגןון הגנה ייעיל על מנת למנוע (או לפחות לצמצם) מתקפות XSS, כ-sh-encoding לא אפשרי (כאשר מקבלים תוכן עשיר מה-user) זהו מנגןון של פילטרים.

כמו בכל מנגןון פילטרים, יכולה להיות רשיימה לבנה ורשיימה שחורה. לצערנו (או שלא), רוב המנגנונים משתמשים בגישה של רשיימה שחורה.

פילטרים שחסומים למשל תגים כמו '`script`' או '`img`' וכו', על מנת למנוע מאייתנו להכניס סקריפט שלנו שירוץ. אך כבר דיברנו על התגים החדשניים של HTML5, ובעזרתם נוכל לעקוף מנגןוני פילטור אלו, ולה裏ץ את הסקריפט הנחיש.

למשל, תגי המולטימדיה החדשניים עבור `video` ו-`audio`, להם יש מאורע של מה לעשות בזמן שגיאה (onerror).

וכמו תמיד, שגיאות אנחנו יכולים ליצור גם בעצמנו. כאשר נקבע `src` לא לגיטימי, מאורע ה-`onerror` של תג ה-`audio\video` יתרחש. לדוגמה:

```
<video src="0" onerror="javascript:alert(1)"></video>
```

```
<audio src="0" onerror="javascript:alert(1)"></audio>
```

כאשר שמים כמה תגי `source` מוקפים בתג `video/audio`, הדף יעבור על כל רשיימת ה-`source`-`src` עד שיימצא `source` בו הוא תומך והוא יפעיל. גם לתג ה-`source` קיימים מאורע `onerror` אשר יקרה כאשר ה-`source` הגרפי לא נתמך, וכן עוד דרך למאורע ה-`onerror` היא לגרום לכך שככל הפורמטים של תגי ה-`sources` ששמננו יהיו לא נתמכים, לדוגמה:

```
<video><source onerror="javascript:alert(1)">
```

```
<audio><source onerror="javascript:alert(1)">
```

כעת נראה רמה אחת למעלה, פילטרים אשר חוסמים את התווים '`<`'-'`>`'. לא נוכל להזrik תגיוט, אך נניח (הנחה סבירה יחסית) כי הטקסט שנכנס כקלט נכנס כערך של תוכנה של תג, וכן נעבור לאפשרויות הבאות.

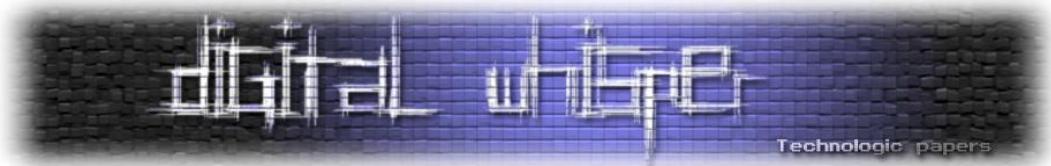
### באמצעות תכונות חדשות

נסתכל לדוגמה על הקוד הבא:

```
<input type='text' value='someUserInput' />
```

- HTML5 -

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



לפני HTML5, על מנת לנצל חולשת CSS זו, היינו משתמשים למשל במאורעmouseover במאומו, וכך שמשתמש היה עובר עם הסמן על מקום זה, היה מושך הסקריפט:

```
someUserInput = ' onmouseover='alert("XSS HTML4 version")'
```

באמצעות HTML5, המספק לנו תכונה חדשה לתג ה, תכונת autofocus שמטרתה לשים את שדה הטקסט בפוקו עם טיענת הדף אוטומטית, אנו מקבלים כוח רב יותר, ונוכל בקלות לגרום לסקרייפט לירוץ ישר עם טיענת הדף ללא צורך במעבר ידני של הגולש עם הסמן, כלומר ללא אינטראקציה עם הקורבן:

```
someUserInput = ' onfocus='alert("XSS HTML5 version")' autofocus='autofocus'
```

צינו קודם כי תכונת ה-autofocus לא חייבת לקבל ערך. כלומר, השורה הבאה שකולה:

```
someUserInput = ' onfocus='alert("XSS HTML5 version")' autofocus custom=''
```

שים לב ששמננו תכונת custom, שהיא custom attribute סתם בכך להיפטר מההגדר האחרון שסוגר את הערך של value בלי שגיאות.

כמו כן:

```
someUserInput = ' onfocus=alert("XSS HTML5 ++ version")' autofocus custom=''
```

(לא חייבים להקיף את ערכי התכונות בגרשיים)

מה שיקרה, זה ששמנו את השדה בפוקו אוטומטי, וכל עוד הוא בפוקו, יירוץ ה-alert.

משמעותנו לא רוצים ליצור אינסוף דיאלוגים נוספים, כלומר שהסקרייפט יירוץ כל עוד השדה בפוקו, ננצל את העובדה שרק אחד יכול להיות בפוקו בזמן נתון. כאשר שני תagi input מכילים autofocus, הדף בסופו של דבר יעביר את הפוקו מהראשון שהוגדר לשני. כלומר, הראשון בשלב זהה יפסיק אותו. מזכיר לנו event כלשהו? (האם-onblur?)

```
someUserInput = '/><input onblur='alert("I love Israel")' autofocus custom=''
```

ונקבל הריצה פעם אחת ללא אינטראקציית המשתמש. כמובן שגם הוא יחזיר לפוקו על השדה ויצא ממנו שוב אז הסקריפטשוב יירוץ. אך נוכל בהרצת הסקריפט בפעם הראשונה לדאוג לכך ש מצב זה לא יקרה.

מה קורה אם יש פילטור ספציפי על EVENTS (events), כמו למשל onload, onerror או events נפרעים אחרים, וכעת לא נוכל להשתמש בהם?

לשם כך נשתמש באירועים החדשניים שבאים יחד עם HTML5 כמו למשל `onformchange`-`onforminout`.

לדוגמה:

```
<form id=formid onforminout=alert(1)><input></form>
```

```
<button form=formid onformchange=alert(2)>
```

ואפיו אם נעה עוד שלב, פילטור חזק יותר באמצעות ביטוי רגולרי אשר מזהה תחילת של 'on', לאחריו מילה ועוד סימן =. כלומר, הביטוי הרגולרי:

```
on\w+=/i
```

פילטר חזק, אך גם זה לא ימנע מאייתנו להשתמש בתכונות החדשניות של HTML5 אשר עברו את הפילטר זהה בקלות, כמו למשל:

```
<form id="formid" /><button form="formid"  
formaction="javascript:alert(1)">
```

עוד וקטורי תקיפה חדשנית תוכל למצוא ב- <http://www.html5sec.org>

### ביצוע התקפות CSRF בעזרת XDR

מתתקפת CSRF, מתבססת על העובדה שהמשתמש סומך על הדף (וגם על האתר הנתקף). זו נגרמת מכיוון שבכל פעם שהדףשלח בקשה לעמוד מסוים הוא מצרף את ה-Credentials (עוגיות, SSL Certificate, HTTP Authentication Credentials) אשר מזוהות עם אותו מקור לבקשתה.

כך לדוגמה אם היינו מחוברים לאתר של הבנק שלנו ובאותו הזמן היינו נכנסים לאתר זמני המכיל את תג התמונה הבא:

```

```

הדף שלנו היה מבצע בקשה GET כדי לקבל את התמונה, ובקשה זו הייתה גורמת להעברה כספית אל חשבונו (שכן באותו הזמן הבנק זיהה אותו על פי ה-Credentials שהדף צירף).

פתרונות לבעה זו הגיעו ממעבר לשיטת POST (שלא פותרת את הבעיה לגמרי - בעזרת יצירת טופס מוחבא ב-iframe ושליחתו אל אתר הבנק ניתן לשלוח בקשה ב-POST, אך זה טיפה יותר מסורבל) ועד שליחת Challange (או Token רנדומלי) בחזרה עבור כל פעולה שכזו.

בעזרת XDR, ניתן לבצע כתף CSRF גם בבקשתות DELETE ו-PUT!

לכן שרתים אשר לא מוגדרים טוב ומאפשרים בקשות שכאלו ללא אונטנטיקציה (ואפילו עם אונטנטיקציה) עלולים להיות פגיעים, תוקף יהיה מסוגל לנצל את ה-Credentials של משתמש בעל הרשות כתיבה ומחיקה ולבצע פעולות אלו במקומו, וכך להשתיל, לשנות ולמחוק עמודים מן האתר. כדי שהזה יקרה, על הרשות לאפשר בקשות XDR עבר שיטות אלו באופן ידני (זה לא געשה אוטומטי), ובברירת המחדל זה לא מתאפשר. אך נזכר ש-XDR יכול לבצע בקשות גם לרשות הפנימית, ושם יש סיכוי גדול יותר למצאו פרצות אבטחה שכאלו, וקיים איום ממש>.

## לסיכום

במאמר זה סקרו והציגו חלק קטן משלל הפיצ'רים והתוספות ש-HTML5 מאפשר לנו.

ראינו כיצד פיצ'רים חדשים עלולים ליצור בעיות אבטחה חדשות, לחדש אפשרות ניצול בעיות ישנות וגם כיצד פיצ'רים חדשים באים לסייעו בניסיון לפתור בעיות אבטחה ישנות (sandbox), על אף שהן פתרונות אלו עדין לא הגיעו לכדי שלמות.

ישן הרבה תוספות ושינויים שהצטברו ויצרו את מצב האינטרנט כיום (וכנראה מצב האינטרנט בעתיד), ברובם לא נגענו, ובמאמרים הבאים נמשיך ונסקור את שאר התוספות שנעשו ל-HTML5, ונתעמק בשאלות חשובות כמו: האם מנגנון Drag&Drop יוצר גל מתקפות חדשות מסווג Clickjacking, כיצד ניתן להשתמש ב-API History בכך לבצע מתקפות פישינג.

נמשיך ונסקור גם את התוספות השונות שנעשו מסביב ל-HTML5, וגם הן מהוות חלק בלתי נפרד מהשינוי המסיבי שה האינטרנט עבר כיום, כמו Websockets, Geolocation, Client Storage וכן הלאה, שהן אמורים חלק מהחדשניים שבאים יחד עם HTML5, אך בנגדם למה שנוהג לחשב, הן אינן חלק מ-HTML5. גם פה נעה על שאלות חשובות כמו: האם מנגנון DatabaseStorage יהיה חשוב למתקפות מסווג SQL Injection? מה ההבדל בין ובין Session Storage, Database Storage או Global Storage ולמה אנחנו צריכים כל כך הרבה שיטות לשומר מידע לצד הקוקיז?

נראה גם כיצד ניתן לבצע Port Scanning בעזרת בקשות XDR ו-WebSockets, ונציג דרכי חדשים לנצל משתמשים תמיימים לצרכינו הזרים, כל אלו ועוד במאמרים הבאים.

### קצת קישוריהם

- <http://www.andlabs.org>
- <http://code.google.com/p/html5security>
- <https://developer.mozilla.org/en/HTML/HTML5>
- <http://mashable.com/2011/04/29/html5-web-security>
- <http://www.softwaremag.com/focus-areas/security/featured-articles/what-does-html5-mean-for-security>



### על הכותבים

ליין בנודיס ואלעד גבאי, בני 19, סטודנטים בשנה ד' בהנדסת מחשבים באוניברסיטה העברית. סטודנטים במעבדת האינטרנט של האוניברסיטה העברית:

<http://internetlabhuji.wordpress.com>

### תודות

תודה ל:

- ישראל חור'בסק (SRO), אפיק קוסטיאל (cp77fk4r) ואוהד אסולין.

קואבאנגהההה!!!!

---

## דברי סיום

---

בזאת אנחנו סוגרים את הגלילון ה-24 של Digital Whisper. אנו מארד מקווים כי נהנתם מהגלילון והכי חשוב - למדתם ממנו. כמו בಗליונות הקודמים, גם הפעם הושקעו הרבה מחשבה, יצירתיות, עבודה קשה ושעות שינה אבודות כדי להביא לכם את הגלילון.

אנחנו מוחפשים כתבים, מאירים, עורכים (או בעצם - כל יוצר חי עם טמפרטורת גוף בסביבת ה-37 שיש לו קצת זמן פניו [אנו מוכנים להתאפשר גם על חום גוף 36.5]) ואנשים המעורניים לעזר ולתרום **לגלילונות הבאים.** אם אתם רוצים לעזר לנו ולהשתתף במגזין Digital Whisper - צרו קשר!

ניתן לשלוח כתבות וכל פניה אחרת דרך עמוד "צור קשר" באתר שלנו, או לשלוח אותן לדואר האלקטרוני שלנו, בכתבوبة [editor@digitalwhisper.co.il](mailto:editor@digitalwhisper.co.il)

ועכשיו בדיחה (נראה אם ניר ישם לב אליה בעריכה):

מתמטיקאי פיזיקאי וביוולוג יושבים במסעדת מסתכלים מהחלון, פתאום עוברת אישת ונכנסת לבניין ממול, לאחר עשר שניות יוצאים מהבניין שני גברים. הפיזיקאי אומר: "לא הגינו... ננראה שטעינו במדידה", אומר הביוולוג: "מה זאת אומרת לא הגינו? הם התרבו!", המתמטיקאי אומר: "מספיק שאדם נוסף יכנס לבניין והוא יהיה ריק!". ☺

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

**[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)**

"Talkin' bout a revolution sounds like a whisper"

הגלילון הבא י יצא ביום האחרון של חודש ספטמבר 2011.

אפיק קוסטיאל,

ניר אדר,

31.08.2011