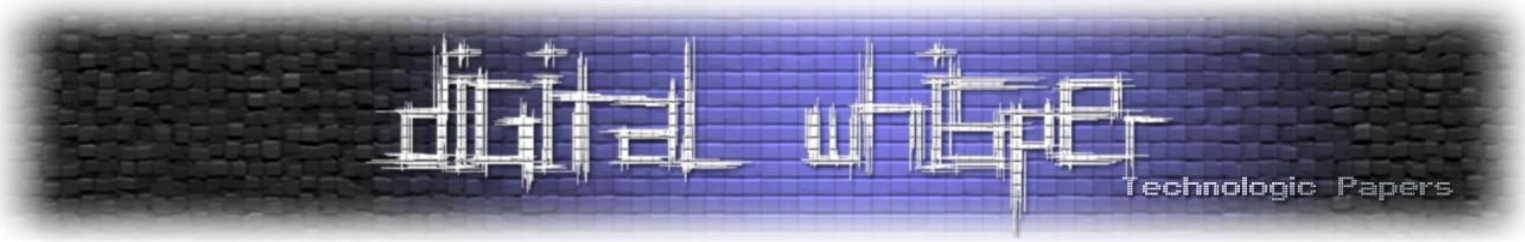


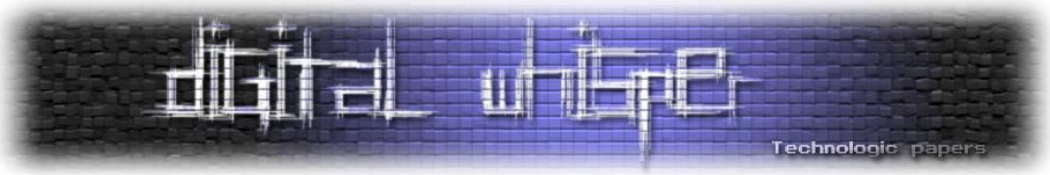
Digital Whisper

גליון 5, פברואר 2010

מערכת המגזין:

מייסדים:	אפיק קסטיאל, ניר אדר
מוביל הפרוייקט:	אפיק קסטיאל
עורכים:	סילאן דלאל, ניר אדר
כתבים:	אפיק קסטיאל, יהונתן קלינגר, עידו קנר, LaBBa, Crossbow, צבי קופר, ניר אדר

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



דבר העורכים

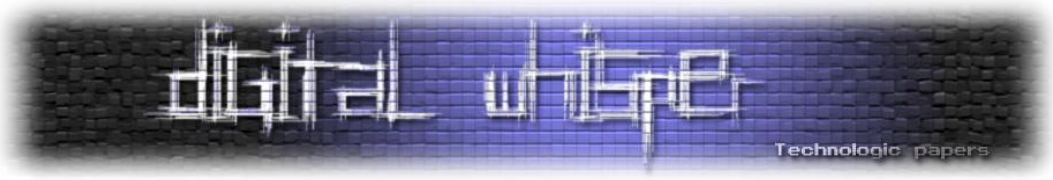
סוף ינואר הגיע, ואיתו הגליון החמישי של Digital Whisper. החודש היה חודש צפוף מאוד יחסית, לפחות מבחינתי. אך בכל זאת, אתם יכולים לראות-הגליון החמישי של Digital Whisper שוחרר בזמן ואנחנו כבר בחצי הדרך לקראת הגליון השישי.

לאחרונה, כמו שבטח כבר שמתם לב, אנחנו מנסים להוסיף עוד אלמנטים ל-Digital Whisper ולהרחיב את הפלטפורמה שיצרנו, התחלנו להריץ במסגרת האתר בלוג בנושא חדשות אבטחת מידע ואף לדבר על פיתוח קהילה. בגליון הנוכחי יש שבעה מאמרים במגוון נושאים כגון אבטחת מידע ו-Hacking, הינדוס לאחור (Reversing), תקשורת וטלפוניה (אבטחת מידע בשרתי Asterisk), פיתוח תקני ונכון, Computer forensics והמאגר הביומטרי. אם בגליונות הראשונים היו מקסימום שלושה כותבים לכל גליון וכל העומס היה נופל עלי ועל ניר, הרי שכיום מגוון הכותבים עלה וכך גם תחומי העיסוק. בגליון הנוכחי משתתפים חמישה כותבים חדשים:

- **Crossbow**, בחור מגניב שרק הוותיקים בסצינה בארץ יכולים בקושי לזכור, הביא לנו (ביחד עם cp77fk4r) מאמר בנושא ביצוע התקפות על כלי האנטי-וירוס בעזרת Zip Bombs.
- **יהונתן קלינגר**, עורך דין העוסק בתחום דיני המידע ובעל תואר שני במשפטים ותארים ראשונים במשפטים וממשל, בעל הבלוג Intellect or Insanity המתייחס במאמר למאגר הביומטרי ולגישה שתתאפשר אליו.
- **LaBBa**, מתכנת ישראלי שהצליח לפרוץ את מנגנון ה-DRM ב-Kindle של חברת Amazon וכותב על פעילותה של מערכת ה-DRM, חולשותיה והדרך לנצלן.
- **צבי קופר**, יועץ לענייני אבטחה בחברת הייטק גדולה בארץ, בעל הבלוג Sec-See (איזה משחק מילים) כותב על Computer Online Forensic Evidence Extractor, כלי הנמצא בשימושן של רשויות חוק.
- **עידו קנר**, הבעלים והמפתח של LINESIP המתמחה בתפירת צרכי הלקוח בשרתי לינוקס בדגש על מרכזיית ה-Asterisk באמצעות יעוץ כותב מאמר על אבטחת מידע בעולם ה-VoIP.

קריאה מהנה!

אפיק קסטיאל ניר אדר



תוכן עניינים

דבר העורכים	2
תוכן עניינים	3
אבטחת SIP עם Asterisk	4
Hacking Kindle DRM	12
Meta Data - פירורי מידע לאויב שלך	20
הפסקת קפה	33
בניית אתרים וסמנטיקה (Semantic HTML)	48
המאגר הביומטרי	58
ZIP Bombs	63
דברי סיום	75

אבטחת SIP עם Asterisk

מאת עידו קנר

אבטחת מידע מעלה אצל לא מעט אנשים מחשבות אודות Buffer Overflow, Format String, XSS, SQL Injection ובעיות רבות הנוגעות למחשב ובעיקר לתוכנה, אך אבטחת המידע המודרנית של ימנו לא החלה עם הופעת המחשב האישי, אלא עם הגעת המכשור הטלפוני למרבית הבתים, שלא לדבר על הטכנולוגיה הסלולרית. חיובי חברות הטלפוניה עבור השימוש במכשירי הטלפון השונים הובילו אנשים לחיפוש ולייצור מערכות המזייפות את אמצעי התשלום (billing) ועוזרות להימנע מתשלום, או סתם להתלבש על שיחה קיימת (man in the middle) וניצולה לצורכי התוקף. התקפות אלו בוצעו על ידי קופסאות כחולות, שחורות ואדומות. בימים ההם פעלו אנשים כדוגמת Captain Crunch אשר התפרסם כחוקר אבטחת מידע, וייצר את הקופסה הכחולה הראשונה. עם מעבר עולם הטלפוניה לתקשורת דיגיטלית באופן כמעט מוחלט, הקופסאות הצבעוניות נמוגו מהעולם כמעט לגמרי, אך אנשים כדוגמת הקפטן עדיין מספקים לנו השראה בעולם אבטחת המידע, וכל חוקר אבטחת מידע (או סתם חובב) המכבד את עצמו מכיר אותו ואחרים מאותה התקופה.

במאמר זה אדון במרכזיית הטלפוניה Asterisk, מרכזיית תוכנה המשוחררת כקוד פתוח. המרכזייה עצמה שינתה את פני הטלפוניה בעולם בכלל ובישראל בפרט אך לדעתי נושא אבטחת המידע בתחום הטלפוניה אינו מקבל חשיפה ראויה.

מה היא מרכזיה ומה זו טלפוניה?

טלפוניה היא תחום רחב, עוד הרבה לפני שהמילה VoIP נכנסה לתמונה, לכן חשוב להבין ולהכיר כמה מונחים בתחום. טלפוניה היא תת נושא בתוך נושא הנקרא טלקומיוניקציה -היכולת לבצע תקשורת בצורה מרוחקת תוך שימוש באותות (signaling) שונים. אותות מוכרים הם: תופים, סימני עשן, סימני דגלים, מורס ועוד.

בהכללה גסה זהו כלי או מכשיר המאפשר להעביר קול בצורה מרוחקת באמצעות אותות וסימנים מוסכמים. כיום אלו סימנים שהם בעיקר דיגיטליים, כאשר בחומרה כדוגמת ISDN אנחנו נעביר את המידע

של האותות על ערוץ המיועד לכך הנקרא d-channel בעוד שהמידע הקולי ו/או וויזואלי יעבור תחת ערוצי הקול ו/או הווידאו -b-channel. השימוש של ISDN נעשה על ידי "פרוטוקולים" המתארים את ספקי הטלפוניה, כאשר יש פרוטוקול בשם E1 (בארץ ובאירופה, T1 ארה"ב ו-J1 במזרח הרחוק).

מונח חשוב נוסף הוא מרכזייה, אשר כשמה כן היא: מכשיר המרכז אליה דבר מה. כאשר מדברים על מרכזיית טלפון, מתכוונים לכלי אשר מרכז בתוכו את כל התקשורת הקשורה לטלפוניה ומחליט על ניתובה ליעד. בהקשר של Asterisk מדובר במרכזייה פרטית לעסק או ארגון (בניגוד למרכזייה של ספק טלפוניה). מרכזיית טלפון כזו נקראת Private Branch Exchange או PBX בקיצור.

צורת טלפוניה נפוצה כיום נקראת Voice Over IP או VoIP בקיצור. כמו כן קיים מונח מקביל בשם VoB או VOBB אשר מדבר על Voice Over Broadband או Video Over Broadband המאפשרים בעצם לתקשר בקול ובווידאו דרך רשת האינטרנט בפס רחב. 2 אמצעים אלו, שהם למעשה אותו הדבר ממומשים דרך תוכנה ודורשים תקשורת בפרוטוקולים שונים המבוססים בדרך כלל על udp.

Asterisk על קצה המזלג:

Asterisk הינה מרכזיית טלפוניה פרטית לעסק המשוחררת בקוד פתוח ונוצרה על ידי אדם בשם מארק ספנסר מחברת Digium שבבעלותו. Asterisk רצה בתור שרת במערכות ההפעלה לינוקס ויוניקס בעיקר, אך ניתנת להרצה כיום גם תחת MS Windows.

המרכזייה תומכת ב:

- טלפוניה פשוטה (POTS) או תקשורת עם מרכזיות ציבוריות, שבארצנו נקראת גם קו בזק
- תמיכה בתקשורת PRI ו-BRI
- תקשורת VoIP הכוללת בתוכה גם תמיכה בווידאו, בפרו\טוקולים שונים כדוגמת SIP, IAX2, H232, Jabber, Skype ואחרים
- מיפוי מספר מול אדם בין מערכות כדוגמת DUNDI ו-ENUM עבור ניוד מספרים
- תמיכה בתוספים שונים כולל צד שלישי

את המרכזיה ניתן לתכנת באופן מלא בין אם עבור שיחה נכנסת, שיחה יוצאת, שיחה בתוך המרכזיה או מעקב אחר משתמשים שונים במערכת. המרכזיה עצמה, כאמור, היא שרת תוכנה לכל דבר ועניין אשר מאזין לפרוטוקולי VoIP, מספקת API שחלקו תחת תקשורת TCP לתוכנות והאזנה למרכזייה עצמה כדוגמת FastAGI אשר מתממשקת לשפת תכנות חיצונית באמצעות TCP או Manager אשר מספק יכולות לדעת ולשלוט מה קורה במרכזייה עצמה, על ידי שימוש בכמה פרוטוקולים כדוגמת HTTP, פרוטוקול פנימי שלו, ועוד כמה צורות עבודה כדוגמת צינורות (pipe) לקלט ופלט. בנוסף המרכזייה יודעת לדבר עם מסדי נתונים שונים כדוגמת MySQL, SQLite, ODBC ו-Berkly DB.

Asterisk תלויה במודולי קרנל אשר מאפשרים לשרת לדבר עם כרטיסי טלפוניה שונים כדוגמת ISDN, כרטיסי טלפוניה פשוטה, כרטיסים שאליהם מתחברים הטלפונים עצמם, או חומרה נלווית המתחברת בדרך כלל דרך USB אל השרתים. ניתן להבין כי אבטחת Asterisk כוללת הרבה מאוד אלמנטים וגישות. בנוסף לכך כל בעיה רגילה שיכולה להיות לשרת, אפשרית ואף לפעמים מתרחשת במרכזיה עצמה. היתרון העיקרי של Asterisk נעוץ בכך שהמרכזיה משוחררת בקוד פתוח ולכן ההתנהלות בבעיות האבטחה מתבצע בד"כ בצורת Full Disclosure בעת שחרור התיקון.

אבטחת המידע ב-Asterisk

בטלפוניה קיימות כיום מספר בעיות מרכזיות הנוגעות לתחום אבטחת מידע:

- גניבת שיחות על ידי שירות "עקוב אחרי"-לאחר חיוג ליעד מופעל קוד המפנה את הקו להיות קו שממנו יבוצעו שיחות יוצאות ומאפשר גניבת שיחות.
- השתלטות על שרתי VoIP-שרתים שאינם מאובטחים כהלכה נפרצים וכך התוקפים יכולים להוציא שיחות ללא עלות מצדם.
- התחזות-האדם שמעבר לקו מאמץ זהות שאינה שלו, בארץ לא ניתן לעשות שזאת למעט בשיחות שאינן מזוהות אלא אם מדובר ב-VoIP נקי. סוג זה של הונאה מאפשר לתוקפים להזדהות כבנקאים בבנק שלכם או כבעלי סמכות כלשהי ולקבל מידע שהוא פרטי.
- האזנות והקלטות שאינן חוקיות, בהן נוכחות המבצע אינה מורגשת (על ידי תפיסה של קו VoIP או גישה פיזית לאחד מצדי השיחה)
- חייגני מלחמה-צורת התקשרות אוטומטית לרוב שמטרידה אנשים ומסייעת בגניבת מידע, התקפות אלו דומות להתקפות עקוב אחריי אך ללא מענה אנושי. צורה זו נפוצה להפצת ספאם טלפוני.

- שידור מידע על Early media (חוסר יכולת לגבות כסף על השיחה). הגופים היחידים בארץ שמורשים לשדר ב-Early Media (המוכרים לי) הם ספקי הטלפוניה שמשמיעים מוזיקה מעל צליל החיוג, או שירות 144 אשר מתבצע מעל מהלך זה.

עוד מידע בנושא אפשר למצוא ב-Wikipedia במאמר בנושא אבטחת טלפוניה.

אבטחת VoIP

לא ניתן להתייחס באופן כללי ל-VoIP כאשר מדובר באבטחת מידע, היות וכל פרוטוקול וצורת עבודה דורשים התייחסות שונה לגמרי מהאנשים האמונים על התשתיות ואבטחת המידע. בעוד שחשוב להבין כי ישנם פרוטוקולים רבים (חלקם בשימוש נרחב יותר מאחרים וחלקם פחות) יש לזכור כי אי אפשר להתייחס במאמר אחד לכולם, לכן נסתפק לעת עתה בפרוטוקול בשם SIP הנפוץ מאוד בעולם התקשורת.

היכרות עם SIP

ראשי התיבות SIP הינם Session Initiation Protocol, שם המרמז בעצם על פעולת הפרוטוקול: יצירת סשן המאפשר לנהל תקשורת בין צדדים שונים. הכוונה ביצירת סשן מזכירה לנו מאוד את פעולת ה-SYN בתקשורת TCP, שבסופו של דבר אחראית על תקשורת בין הצדדים להעברת פקטות ה-TCP. רבים מתייחסים, שלא בצדק, אל SIP כאל מנהרה. בתוך SIP לא עובר מידע נוסף, כדוגמת מנהרת SSL, הפרוטוקול אחראי רק על יצירת הקשר ודואג שהמידע יגיע מנקודת ההתחלה לנקודת הסיום. אפשר להגדיר את SIP כגשר יותר מאשר מנהרה. פרוטוקול ה-SIP נחשב לאחד הפרוטוקולים המסובכים ביותר שיש ומכיל מספר RFC שמוסיפים אחד לשני עוד מידע והסברים. לפי אחד מאותם RFC, פרוטוקול התקשורת ש-SIP יכול להשתמש בו הוא UDP אבל גם ניתן לממש אותו מעל TCP (פחות נפוץ תאורטית). לאחר ש-SIP נמתח מנקודת התחלה לסיום, מתחילים פרוטוקולים אחרים להכנס לשימוש. הפרוטוקול המוכר כמעט לכל אדם המשתמש ב-SIP הוא ה-RTP, שתפקידו להעביר את שיטות הקול והווידאו (codec) של השיחה עצמה. בנוסף ישנו פרוטוקול חשוב בשם SDP שתפקידו לתאם את סוג המדיה שתעבור בין הצד המבקש ליעד הסופי. בנוסף לפרוטוקולים אלו, ישנם עוד פרוטוקולים שונים אשר

עוברים, חלקם אינם נדרשים, אך מוסיפים תכונות כדוגמת MSRP אשר מספק יכולת משלוח טקסט ולגרום ל-SIP לשמש גם כסוג של פרוטוקול Instant Messages.

אופן השימוש ב-SIP

השימוש ב SIP יכול להתבצע ב-3 צורות עיקריות:

1. שרת אל לקוחות
2. שרת אל שרת
3. קישור מחשב לרשת טלפוניה אחרת, קווית לרוב

כאשר מדובר על **שרת מול לקוחות**, הכוונה היא שיש שרת המדבר ב-SIP ויש לקוחות שונים (טלפונים שהם חומרה -> hard phone, טלפונים שהם תוכנה -> soft phone ו-Instant Messages).

בשרת אל שרת, אנחנו משתמשים ב-SIP כ-Proxy בין נקודות, אשר בעצם עוזר להעביר שיחות משרת אחד למשנהו. פעולה זו נקראת SIP Proxy ניתן למצוא על זה מידע ב-RFC מספר 3261. בקישור לרשת טלפוניה אחרת ניתקל בשם SIP Trunk, בו אנחנו מדברים ב-SIP בצד הלקוח ומתחברים לשרת, שהיציאה שלו היא בדרך כלל לתקשורת קווית.

בעיות הקיימות עם SIP

ל-SIP יש מספר בעיות מורכבות שיש לתת עליהן את הדעת לפני שניתן בכלל לדון בהגנה על הפרוטוקולים השונים.

- הבעיה הראשונה והחשובה ביותר היא בכך שהפרוטוקול דורש טווח של 10,000 פורטים פתוחים. בעוד ש-SIP עצמו דורש רק פורט אחד, 5060, שאר הפרוטוקולים דורשים את הטווח העצום הזה של הפורטים. זה לא הכל, היות ופרוטוקולים כדוגמת RTP ו-SDP בוחרים בצורה רנדומלית פורט אחד מתוך הטווח, אי אפשר לדעת מה יבחר מראש. בנוסף, Asterisk למשל, אינו מאפשר להגביל את

הטווח הזה, למרות שיש מספר שרתים ולקוחות המאפשרים להגבילו. בעיית הפורטים יוצרת מצב בו מאוד לא פשוט לעבוד עם NAT ובד"כ במקרה של מחסור בכלי הגנה שיודעים לעבוד עם SIP יש לבטל צורות הגנה כלליות כדי לאפשר את הטווח.

- הואיל ומדובר בפרוטוקול טקסט נקי ונדרשת הזדהות, הסיסמאות גלויות לכל, גם אם משתמשים ב-Digest להצפנתן.
- בטלפוניה בכלל וב-SIP בפרט יש רגישות יתר להתקפת man in the middle. במקרה של SIP קל מאוד לגנוב מאחד הצדדים את התקשורת, להאזין לשיחה או סתם לזהות את הצדדים המעורבים בה. כל מה שצריך זה לשתול עוד שורה בתחילת השיחה של Contact בשביל לשנות את הכתובת לכתובת החדשה והתקשורת הועברה לכתובת אחרת.
- SIP מאוד פגיע להתקפות DoS בכל נקודה ברשת, ואפילו שינוי ברוחב הפס לרעת ה-SIP יכולים להיחשב ככזו.
- ניתן להתערב ל-SIP באלגוריתמים, בפרוטוקולים והראשים השונים, דבר שיכול לגרור לבעיות אבטחה שלא תמיד ניתנות לחיזוי מראש.

עבודה בטוחה עם SIP

כדי להעביר בצורה בטוחה את כל הפורטים השונים, ניתן כאמור להשתמש בציוד היודע לעבוד עם SIP, אך מדובר בדרך כלל בציוד יקר מאוד. לכן ניתן לעבוד עם מנהרה בשם stun המאפשרת להעביר את כל פרוטוקולי התקשורת כדוגמת RTP ו-SDP תחת אותה מנהרה. המנהרה עובדת ב-UDP ומאפשרת לעבור NAT. קל יותר לאפשר פורט בודד מאשר טווח של 10,000 פורטים.

בנוסף, יש לזכור שיש בעולם גם את IPv6, אשר תאורטית מעלה את הצורך ב-NAT היות ואין הבדל בין כתובות פנימיות וחיצוניות, דבר שמשנה את כל הגישה לכמות הפורטים שצריך להעביר "פרטית" לרשת "חיצונית".

בעבר היה שימוש בשיטת הזיהוי של PGP בה לכל צד יש מפתח ציבורי ומפתח פרטי ובהתאם למפתח הציבורי של הצד השני כפול המפתח הפרטי של הצד המדבר היה זיהוי חד ערכי בין כל צד- דבר שניסה למנוע Man in the middle, אך שיטה זו נגנזה מחוסר הגדרה ברור בתקן.

RTP ו-פרוטוקול בשם RTCP (עוד פרוטוקול בשימוש SIP) מכילים הגנות שונות:

- פרוטוקול בשם ZRTP שנועד לבצע את אימות התעודות במצב ה-RTP ובכך כאמור לעזור במניעת Man In the Middle.
- יש פרוטוקול הצפנה בשם SRTP שביחד עם אחיו SRTCP מאפשרים לקודד תקשורת קולית וויזואלית.

שיטות הגנה אלו דורשות ששני הצדדים יתמכו בשיטות העבודה האלו, לצערי ישנם לא מעט לקוחות SIP אשר אינם תומכים בהן.

כאשר מדובר בארגון מסודר ישנן הגנות נוספות

הגנות ברשת מקומית: ברשת הפנימית בארגון מומלץ בראש ובראשונה ליצור באמצעות VLAN רשת המיועדת לתחום ה-SIP ושום דבר אחר אינו מאושר להשתמש בה. היתרון של VLAN הוא בכך שהוא לוקח רשת ומחלק אותה בצורה וירטואלית לרשת חדשה, פעולה שמפרידה בעצם את התקשורת בין הרשת במשרד המיועדת למידע כללי לבין רשת המיועדת רק ל-SIP. פעולה זו גם מאפשרת ליצור מסננים שונים עבור פעולות שונות כדוגמת Quality Of Service (מוכר בשם QoS) אשר יכול לספק תעדוף בתקשורת עבור תעבורת SIP. ניתן לאפשר רק לכתובות IP עם כתובות MAC ספציפיות. באותה רשת תקשורת הווירטואלית יתחברו רק הלקוחות SIP בעוד שכל סוג תעבורה אחר לא יחובר אליו.

כל הזדהות תתבצע באמצעות Message Digest אשר מצפין סמאות (אבל עדיין מאפשר הזדהות עם ההצפנה עצמה גם למי שרק "מאזין" לתקשורת) בצורה שאי אפשר לשחזר, ובנוסף גם ההזדהות יכולה להיות מבוססת כתובת IP ספציפית וכך כל ניסיון ניתוב מחודש של בקשת SIP לא תעבור הלאה גם אם מדובר בלקוח בתוך הרשת הייעודית.

במידה ובארגונים שונים יש גם כמה אתרים שונים שאינם יכולים להתחבר ל-VLAN אחד בצורה מאובטחת, ניתן לעבוד עם VPN מאובטח, אשר משם אפשר להכניס את 2 הרשתות לתוך VLAN בצורה מאובטחת יותר.

הגנות ברשת האינטרנט: במידה ואי אפשר לדבר ב-VPN מאובטח, יש להבטיח שרק כתובות IP מוגדרות יוכלו לדבר עם שרת ה-SIP, בנוסף להצפנות שהוזכרו למעלה. כמו כן, אפשר להגדיר לשרתים שונים

(בד"כ באמצעות כלים כמו OpenSips) להתעלם מבקשות Contact אשר מבקשות שכל מענה יתבצע בכתובת שונה.

כמו בכל שרת, החשיפה לאינטרנט צריכה להיות מינימלית ביותר, ומוגבלת לחיונית ביותר. כמו כן, צריך לזכור להוריד הרשאות למינימום ההכרחי לכל תהליך שרץ, כך שגם אם יש בעיית אבטחה, סביר להניח שהיא לא תספק אפשרות להסלמת הרשאות.

סיכום

עולם הטלפוניה גרם לכך שעידן המידע המודרני היה צריך לתת מענה לאבטחת מידע הרבה לפני התפתחות המחשוב האישי ונגישות המידע שקיימת כיום. הכרנו בעולם הטלפוניה את מרכזיית Asterisk המשוחררת בקוד פתוח ומספקת תמיכה בהרבה מאוד יכולות. באמצעות Asterisk גילינו כי עולם הטלפוניה מכיל המון תתי נושאים על גבי תתי נושאים, וכל אחד מהם מהווה בעיית אבטחת מידע משל עצמו שצריך לתת עליו את הדעת.

תחום ה-VoIP הוא מאוד כללי וצריך לרדת לרמת הפרוטוקולים שבשימוש על מנת לדעת איך לאבטח את הנושא. ראינו כי פרוטוקול ה-SIP מורכב מאוד ודורש התייחסות נקודתית, והבנו חלק קטן מהבעיות שקיימות אצלו.

כדי לדעת עוד על אבטחת פרוטוקול SIP מומלץ מאוד לקרוא את ה-RFC המטפל בנושא ולהבין שגם הוא לא מספיק בשביל לספק את מלוא ההגנה הנדרשת. הכי חשוב לזכור כי באבטחת מידע, הצרכים הם אלו שמכתיבים את צורת ההגנות ולכן לא באמת ניתן לכתוב מאמר שיכסה את כל האפשרויות לאבטחת פרוטוקולים כדוגמת SIP.

Hacking Kindle DRM

מאת LaBBa

את המאמר נפתח בפירוש מושג ה-DRM או בעברית-נז"ק. פירוש המושג Digital Rights Management, או "ניהול זכויות דיגיטלי", הוא פשוט למדי - החברה המשתמשת במושג בוחרת להגביל את המשתמשים באמצעות טכניקות שונות. טכניקות אלו יכולות להיות קשורות בתוכנה או בחומרה ומטרתן להבטיח כי הצרכן של אותו המוצר לא יוכל להעביר את תוכן המוצר לאדם אחר.

עצם העיקרון הכללי, המגביל את לקוחות החברה בכל הנוגע להעברת תוכן, גורר אחריו תוצאות ותופעות לוואי בלתי נמנעות. כלומר, אדם הרוכש מוצר כלשהו יתקשה, או לא יוכל להעביר את התוכן למקום אחר או לשימוש בתוכנה אחרת. לדוגמא, נניח כי חברה משתמשת ב-DRM על מנת להגן על שיר בצורה מסוימת, הגנה זו תמנע האזנה לשיר בכל נגן שאינו מסופק על ידי אותה החברה. אם ברשות הלקוח נגן אחר, הוא לא יוכל להאזין לשום שיר שרכש מפני שמדובר בקובץ יוצא דופן ומוגן. בעשור האחרון, חוקקה ממשלת ארצות הברית את חוק המילניום, האוסר על לקוחות ומשתמשים להסיר את הגנות ה-DRM, משום שעצם ההסרה היא פגיעה בזכויות יוצרים. ואכן, גם פה בארץ היו תיקונים לחוק זכויות היוצרים. תיקונים אלו נקבעו בשנת 2007 ומציגים אפשרות מעט שפויה יותר- החוק מתיר לאדם אשר רכש מוצר להסיר את ההגנות של זכויות היוצרים על מנת להתאימו לתוכנה אחרת, כמו גם יצירת גיבוי של המוצר והקלות שונות שמותרות אך ורק לבעל המוצר, אך בשום פנים ואופן אין למכור או להעביר את התוכן לאחר הסרת ההגנה.

Kindle

הקינדל יצא לראשונה לשוק כחומרה המאפשרת למשתמשים לרכוש ספרים מ-Amazon ולקרוא בהם על גבי אותו המוצר. כצפוי, חברת Amazon בחרה להגן על הספרים שלה בשיטות הצפנה ייחודיות כך שלא יהיה ניתן להעביר ספרים מאדם לאדם וכן שלא יתאפשר לקרוא את אותם הספרים על אף מוצר מלבד אותו מוצר יחיד שנרכש עבור הספר. הגנות אלו הקשו על המשתמשים, שכן לאחר רכישת הספרים לא יכלו לקרוא בהם למעט על גבי אותה חומרה ספציפית שיועדה לכך. כמו כן, לא התאפשר למשתמשים להמיר את הקבצים לפורמט PDF או פורמטים סטנדרטיים אחרים המיועדים לקריאה במחשבים ביתיים.

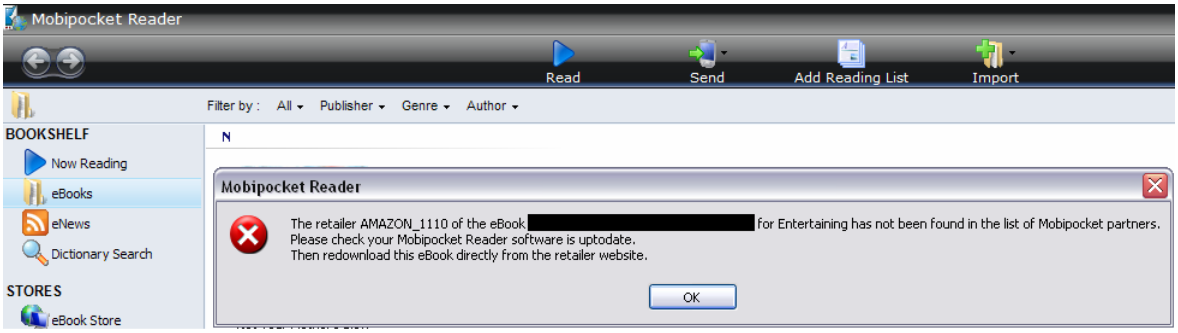
Amazon בחרה לתמוך אך ורק בפורמטים AZW, MOBI, ו-TPZ ואף לא תמכה בקריאת ספרים שאינם מוגנים בהתאם לשיטה שקבעה, כך שלא ניתן היה להשתמש במכשיר זה לכל ספר אחר.

בתגובה למחדל, קהילת ההאקרים החלה לנסות וללמוד את הטכנולוגיה העומדת מאחורי Kindle על מנת להבין כיצד ניתן יהיה להסיר את ההגנה ולאפשר קריאה של הספרים לא רק בעזרת החומרה, כמו גם לאפשר ל-Kindle לקרוא תוכן נוסף שלא בהכרח נרכש דרכם ולתמוך בפורמטים נוספים מלבד אלו שהוגדרו כתקניים על ידי Amazon. ההאקר הראשון שהצליח במשימה היה איגור סקוצ'ינסקי (Igor Skochinsky), שמצא את המספר הייחודי של כל מוצר (UUID), את האלגוריתם אשר מספק מהמספר הייחודי (ופרמטרים נוספים על המשתמש של המוצר) מפתח ייחוד שניקרא בשם PID (platform ID). לאחר מכן הוא קודד תוכנה המאפשרת למשתמש ליישם את החתימה של Amazon על גבי קבצים שאינם מוגנים, כך שניתן יהיה להשתמש בחומרה על מנת לקרוא ספרים שלא נרכשו דרך החברה. בנוסף, מצא סקוצ'ינסקי כי Amazon משתמשת בשיטת הצפנה הנקראת PC1 או בשמה המלא Pukall (הצפנה זו נקראת על שם אלכסנדר פוקאול (Alexander Pukall)).

זמן קצר לאחר תגליות אלו, האקר בשם DarkReverser עבד על הסרת ההגנה של ה-DRM כך שניתן יהיה להסיר את ההגנות ולקרוא את הספרים על ידי תוכנות אחרות מלבד התקן החומרה Kindle ואף ביצע עדכונים ותחזוקה בקוד של איגור סקוצ'ינסקי. הקוד של DarkReverser אפשר למשתמש, בהנתן ה-PID של המוצר, להסיר את ההגנות החלות עליו ולקבל קובץ סופי של ספר שאינו מוגן.

Kindle For PC

שלוש שנים לאחר מכן, החליטה Amazon כי היא רוצה להתפתח והחלה להציע ללקוחותיה מוצר חדש המאפשר קריאת ספרים מוגנים גם על גבי מחשבים ביתיים. הבעיה העיקרית בתוכנה זו היא שמדובר בגרסאת Beta, גרסא שנכון לעכשיו לא מכילה מאפיינים מתקדמים. בנוסף, כיום קיימות תוכנות המסוגלות לקרוא ספרים בפורמט MOBI ש-Amazon תומכת בהן ויש להן יכולות מיוחדות, כגון הדגשות בתוך הטקסט, הערות של המשתמש וכן הלאה. אך מה יקרה אם נרצה לפתוח ספר שרכשנו המכיל הגנות DRM של Amazon?



ובכן, כאן אני נכנס לסיפור. ההנחה הראשית שיצאתי ממנה היא Amazon לא תשבור את ממשק האבטחה שבנתה ל-Kindle ותמשיך עם אותה הטכנולוגיה. בהתבסס על כל העובדות שנצברו, התחלתי לחשוב כיצד אוכל למצוא את ה-PID שנוצר על המחשב האישי לעומת מה שקודמי עשו עם מוצר החומרה ה-Kindle. המחשבה הראשונה שעולה לכל אחד, הניגש למשימה דומה, היא כנראה "איפה לעזאזל מתחילים?" והתשובה במקרה זה היא פשוטה מכפי שתוכלו לדמיין. לקחתי את קוד המקור של DarkReverser והתמקדתי בחלק האחראי להסרת ההגנה, לאחר שכבר גילינו את ה-PID.

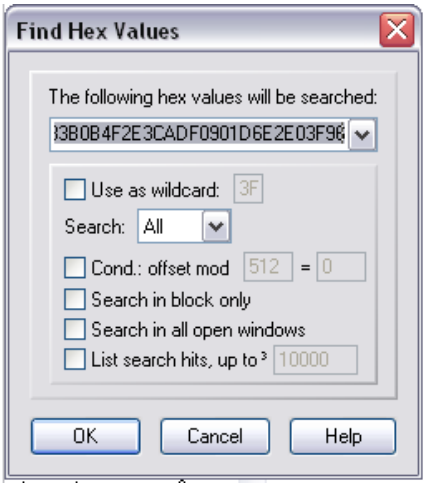
הנה חלק מהקוד של DarkReverser:

```
def parseDRM(self, data, count, pid):
    pid = pid.ljust(16,'\0')
    keyvec1 = "\x72\x38\x33\xB0\xB4\xF2\xE3\xCA\xDF\x09\x01\xD6\xE2\xE0\x3F\x96"
    temp_key = PC1(keyvec1, pid, False)
```

כפי שניתן לראות, הוא קורא לפונקצית ההצפנה PC1 עם 3 ארגומנטים, כאשר המעניין מביניהם הוא דווקא לא ה-PID אלא ה-keyvec1. משתנה זה הוא מעניין יותר עבורנו מפני שהוא ניראה כרצף בתים קבוע הנדרש על מנת להצליח לפרוק הצפנה של קבצים, בנוסף ל-PID.

ברגע שראיתי את הווקטור הקבוע המדובר, פתחתי HexEditor לחפש את רצף הבתים הזה:

```
007FB130  74 39 C8 00 05 00 00 00  03 00 00 00 E8 D9 C7 00
007FB140  06 00 00 00 04 00 00 00  D4 6F C8 00 07 00 00 00
007FB150  05 00 00 00 C8 6F C8 00  05 00 00 00 06 00 00 00
007FB160  74 00 72 00 75 00 65 00  66 00 61 00 6C 00 73 00
007FB170  65 00 75 00 6E 00 64 00  65 00 66 00 69 00 6E 00
007FB180  65 00 64 00 6E 00 75 00  6C 00 6C 00 30 00 4E 00
007FB190  61 00 4E 00 49 00 6E 00  66 00 69 00 6E 00 69 00
007FB1A0  74 00 79 00 00 00 00 00  72 38 33 B0 B4 F2 E3 CA
007FB1B0  DF 09 01 D6 E2 E0 3F 96  FF FF FF FF FF FF FF FF
007FB1C0  00 00 00 00 00 00 00 00  00 00 00 00 22 00 01 00
007FB1D0  22 00 00 00 00 00 00 00  26 00 01 00 26 00 00 00
007FB1E0  00 00 00 00 27 00 01 00  27 00 00 00 00 00 00 00
007FB1F0  3C 00 01 00 3C 00 00 00  00 00 00 00 3E 00 01 00
007FB200  3E 00 00 00 00 00 00 00  A0 00 01 00 A0 00 00 00
007FB210  00 00 00 00 A1 00 01 00  A1 00 00 00 00 00 00 00
```



כפי שניתן להבין מתרשים זה, רצף הבתים הזה קיים בשלמותו בתוך קובץ ההרצה של תוכנת ה- Kindle For PC. די נדהמתי לגלות כי הרצף קיים גם בתוכנת המחשב ורציתי לדעת היכן בקוד של התוכנה נעשה שימוש בערך זה, מפני שאם אמצא את המקום שמשתמש בערך הזה, שם אמור להמצא גם ה-PID (על פי דוגמת הקוד של DarkReverser).

על מנת למצוא את המקום שבו התוכנה תשתמש בערך זה, נעזרתי ב-IDA:

```
:00BFC5A8 dword_BFC5A8    dd 0B0333872h
:00BFC5A8
:00BFC5AC dword_BFC5AC    dd 0CAE3F2B4h
:00BFC5B0 dword_BFC5B0    dd 0D60109DFh
:00BFC5B4 dword_BFC5B4    dd 963FE0E2h
:00BFC5B8                 db 0FFh
:00BFC5B9                 db 0FFh
```

```
; DATA XREF: sub_5708C0+EA↑r
; sub_5708C0+1B8↑o
; DATA XREF: sub_5708C0+F0↑r
; DATA XREF: sub_5708C0+F6↑r
; DATA XREF: sub_5708C0+FF↑r
```

References

כפי שאתם רואים, IDA מספקת את ה-references להיכן בקוד משתמשים בערך זה:

```
:005709AA                mov     ecx, ds:dword_BFC5A8
:005709B0                mov     edx, ds:dword_BFC5AC
:005709B6                mov     eax, ds:dword_BFC5B0
:005709BB                mov     [esp+0C4h+var_AC], ecx
:005709BF                mov     ecx, ds:dword_BFC5B4
```

וכפי שניתן להבחין, בכתובת: 5709AA מתחילה השמת הערך של הווקטור למשתנים לוקאליים של הפונקציה. לאחר מכן, מה שנותר רק לעשות הוא להפעיל את ה-Debugger העדיף עלינו, לשים שם נקודת עצירה ולנסות לפתוח את הספר המוגן שקנינו. למרות שללא ספק ניתן לבצע Debug עם IDA, אני מעדיף לבצע תהליך זה עם OllyDbg.

וכך, אם נשים נקודת עצירה בנקודה הזו OllyDbg יעצור:

```
005709A6  . 837D 04 02      CMP DWORD PTR SS:[EBP+4],2
005709AA  . 8B0D A8C5BF00   MOV ECX,DWORD PTR DS:[BFC5A8]
005709B0  . 8B15 ACC5BF00   MOV EDX,DWORD PTR DS:[BFC5AC]
005709B6  . A1 B0C5BF00     MOV EAX,DWORD PTR DS:[BFC5B0]
005709BB  . 894C24 18       MOV DWORD PTR SS:[ESP+18],ECX
005709BF  . 8B0D B4C5BF00   MOV ECX,DWORD PTR DS:[BFC5B4]
```

אם תשימו לב, בצד תוכלו לראות כי יש כאן לולאות מאוד גדולות בקוד (כיאה לקוד של אלגוריתמים) ולכן לא ממש התעניינתי בניתוח הקוד, אלא רק רציתי לראות האם משהו יצוץ לי מול העיניים כאשר אלחץ על F8 (step over). לאחר כ-200 לחיצות אני מגיע לשורת הקוד בכתובת: 570A94

```
00570A75  . 83C4 0C         ADD ESP,0C
00570A78  . B9 A8C5BF00     MOV ECX,Patched.00BFC5A8
00570A7D  . 8D9424 A400000( LEA EDX,DWORD PTR SS:[ESP+A4]
00570A84  . C78424 A000000( MOV DWORD PTR SS:[ESP+A0],Patched.00C
00570A8F  . E8 FCECFDFF     CALL Patched.0054F790
00570A94  . 8D4424 18       LEA EAX,DWORD PTR SS:[ESP+18]
00570A98  . 50              PUSH EAX
00570A99  . B8 10000000     MOV EAX,10
00570A9E  . 8D4C24 1C       LEA ECX,DWORD PTR SS:[ESP+1C]
00570AA2  . 8BF2            MOV ESI,EDX
00570AA4  . E8 37EDFDFF     CALL Patched.0054F7E0
Stack address=0013F538, (ASCII "-31[redacted]')
EAX=0013F538, (ASCII "-31[redacted])
```

כאן תוכלו להבחין כי זו אכן כן מחרוזת. בהתאם, ביטלתי את נקודת העצירה הקודמת, שמתי נקודת עצירה חדשה על השורה הזאת (כפי שציינתי יש כאן הרבה לולאות) ולאחר מכן לחצתי על F9 (run).

לאחר מכן קיבלתי מחרוזת חדשה, הדבר חזר על עצמו פעמיים ולאחר מכן התוכנה המשיכה לרוץ ללא מפריע והציגה את הספר שרכשתי.

המחרוזת האחרונה שהוצגה הייתה בת 8 תווים, אם זה אכן ה-PID אז כל מה שנותר לעשות הוא לקחת את התוכנה ש-DarkReverser כתב ולנסות להסיר את ההגנה על ידי שימוש ב-PID שמצאתי.

מהלך זה הביא להפתעה לא נעימה:

```
C:\MobiDeDRM>mobidedrm.py enc.prc out.prc 2      E
MobiDeDrm v0.07. Copyright (c) 2008 The Dark Reverser
Error: invalid PID checksum
```

לאחר עיון ובדיקה שניה של הדוגמאות שהובאו על ידי חוקרים אחרים, גיליתי באחד ההסברים פיסה של מידע שימושי, גודל ה-PID צריך להיות 10 תווים! משמעות הדבר היא שחסרים לנו שני תווים וכי תווים אלו הם רק עבור בדיקת חוקיות. נקרא את הקוד שמזהה לנו מהו גורם ל-Bad CheckSum:

```
if checksumPid(pid[0:-2]) != pid:
    raise DrmException("invalid PID checksum")
```

כלומר, אם אשלח לפונקציה רק 8 תווים, במקום 10 (זו המשמעות של [pid[0:-2), אני אמור לקבל אותה מחרוזת בעלת 10 תווים שאליה אני צריך להשוות את המחרוזת שלי. משמעות הדבר היא שפונקצית ה-checksumPid ש-DarkReverser מימש מקבלת 8 תווים ומחזירה לי 10 תווים -את התווים החסרים. בהתאם, לקחתי מהקוד של DarkReverser רק את הפונקציה של ה-cehcksumPid ויצרתי קובץ חדש כך:

```
import sys,struct,binascii

def checksumPid(s):
    letters = "ABCDEFGHIJKLMNPQRSTUVWXYZ123456789"
    crc = (~binascii.crc32(s,-1))&0xFFFFFFFF
    crc = crc ^ (crc >> 16)
    res = s
    l = len(letters)
    for i in (0,1):
        b = crc & 0xff
        pos = (b // l) ^ (b % l)
        res += letters[pos%l]
        crc >>= 8
    return res

print "PID Fixer"
pid = sys.argv[1]
print "the fixed PID is: " + checksumPid(pid)
```

אם ננסה בשלב זה לשלוח את 8 התווים שלנו לסקריפט החדש , זאת תהיה התוצאה:

```
C:\MobiDeDRM>CalcFixed.py 2[redacted]E
PID Fixer
the fixed PID is: 2[redacted]EHR
```

שלחנו 8 תווים וקיבלנו 10 תווים! עכשיו ננסה שוב להשתמש בסקריפט של DarkReverser, הפעם עם 10 תווים.

```
C:\MobiDeDRM>mobidedrm.py enc.prc out.prc 2[redacted]HR
MobiDeDrm v0.07. Copyright (c) 2008 The Dark Reverser
Decrypting. Please wait... done
```

אם ננסה לפתוח את הספר עם תוכנה אחרת כגון Mobipocket Reader לא נקבל יותר את הודעת השגיאה ונוכל להשתמש בפונקציות היותר מתקדמות מאשר בגירסת הבטא.

סיכום

לסיכום, עלינו לזכור כי בסופו של דבר ישנו חוק מנחה אחד- אם זה מסוגל לרוץ בצורה מלאה פעם אחת, תמיד יהיה אפשר לפרוץ את המוצר. כל עוד כל קטעי הקוד הורצו בצורה לוקאלית על המעבד שלנו אין ל-Amazon הרבה ברירות והם יודעים את זה. מאחר והם חייבים לספק פיתרון אשר יאפשר למשתמש לקרוא את הספרים גם בצורה שאינה מחייבת קישור אינטרנט, הם חייבים להריץ את כל הקוד על המעבד שבמחשב שלנו. מסיבה זו בדיוק, Amazon תמיד יפסידו במשחק הזה היות והארכיטקטורה של x86 ביסודה מאפשרת לעשות הכל ללא הגבלה- אם החומרה לא מגבילה אותנו אז התוכנה פרוצה.

הערה: כיום עדיין לא אפשרי לפתוח ספרים מסוג TPZ בעזרת שום תוכנה אחרת מאחר וזהו פורמט לא סטנדרטי כמו MOBI אלא פורמט פרטי של Amazon.

יש צורך להשתמש בתוכנות אחרות אשר ימירו את הפורמט הפרטי של Amazon לפורמט HTML משם אפשר להמיר לכל פורמט שנרצה כדי לפתוח בכל תוכנה שנרצה. לאחרונה, פורסם בפורום של DarkReverser כלים שיודעים להסיר את ה-DRM מקבצי TPZ ולהמיר אותם לקבצי HTML להמרה עתידית לכל סוג של פורמט שנרצה.

בזמן כתיבת שורות אלו, הגירסא האחרונה היא: 1.6: http://www.box.net/shared/gqukrbp0js

כמובן- השימוש במידע זה הוא לצורכי לימוד בלבד ולא למטרות פיראטיות!

מקורות וקישורים

- DarkReverser home page -http://darkreverser.wordpress.com/
- DarkReverser script -http://pastebin.com/f7be270a9
- Igor Skochinsky -http://igorsk.blogspot.com/

Meta Data - פירורי מידע לאויב שלך

מאת אפיק קסטיאל (cp77fk4r)

"המונח Meta Data הוא אחד מהמונחים החשובים ביותר בעולם ניהול המידע. שימוש נכון ב-Meta Data יכול לשפר משמעותית את יכולת הנגישות למידע, ואת אפקטיביות השימוש באותו מידע. המשמעות המילולית של המונח היא "מידע על מידע", כלומר שדות מידע שמוסיפים (אוטומטית או ידנית) למידע. יש שתי מטרות עיקריות ל-Meta Data:

- להוסיף אינפורמציה שאינה כתובה בתוכן עצמו, ואשר עשויה לאפשר להבין את הכתוב טוב יותר, או לעזור לנו להתייחס לכתוב בהתאם. לדוגמא - שדה Meta Data סטנדרטי הוא התאריך. במקרים רבים התאריך עוזר לנו להבין אם הכתוב רלוונטי עבורנו או לא. שדה אחר הוא שם הכותב, שמאפשר לנו לשפוט לגבי מהימנות התוכן (אם אנחנו מכירים את הכותב). גם שדות נוספים כגון - אירוע שבמסגרתו התוכן נכתב, או יחידה ארגונית - עוזרים לנו להבין מניעים/זוויות מבט והיבטים נוספים, שאינם באים בהכרח לידי ביטוי בתוכן עצמו.

- לסייע באחזור המידע. שדות Meta Data רבים מהווים כלי מרכזי באחזור נוח ומוצלח של תכנים. היכולת לתייג תוכן עפ"י מספר מאפיינים, מאפשר לנו לאחזר אותו ע"י שימוש באחד או יותר מאותם מאפיינים. חיפוש עפ"י מספר מאפיינים מאפשר לנו לצמצם את החיפוש ולמקד אותו בדיוק בתכנים אותם אנו מחפשים. כאשר בונים פתרון ניהול תכנים, במקרים רבים ההצלחה שלו תלויה בבחירה נכונה של שדות ה-Meta Data ובהטמעה מוצלחת של השימוש בהם."

(נכתב במקור ע"י יאיר דמבינסקי, סמנכ"ל פרוייקטים Byon IT Solutions)

כמו שאפשר לראות Meta Data משתמש בעיקר להגדרת/איחזור יעיל של המידע-שזה דבר מצוין, אך כמו לכל דבר-גם לנושא הזה יש חסרונות.

הסיכון

מספר רב של תוכנות מוסיפות Meta Data לקבצי הפלט שלהם, אם מדובר במסמכי Doc של Word או במסמכי PDF של Adobe, ואם מדובר בקבצי תמונות של Photoshop או GIMP וגם אם מדובר בקבצים בינאריים שקומפלו ע"י מהדר כזה או אחר.

כאשר כותב מאמר מעוניין להוסיף Meta Data למאמר שלו בכדי לשפר את יכולות האיחזור של מנוע חיפוש כזה או אחר לגבי אותו המאמר זה מצוין, אבל בהרבה מאוד מקרים תוכנות שונות מצרפות למאמר Meta Data שיכול לעזור לגורמים עויינים ללמוד נתונים "אישיים" שיכולים לעזור להם או למקד אותם בעת פריצה לאותו מחשב או לרשת האירגון שבה נכתב המאמר.

הרבה אירגונים גדולים לא מפנים מספיק תשומת-לב לנושא, ולמרות שלרוב לא נחשפות ססמאות או פרטים אישיים של משתמשים, עדיין מדובר במידע יעיל מאוד כאשר מדובר בניתוח מבנה האירגון או הרשת הפנימית של אותו האריגון.

תוכנות שונות ודרכי ייצוא שונות משנות את ה-Meta Data שנכנס לקובץ המיוצא, אך אפשר למצוא שקבצים שונים מכילים Meta Data כגון:

- שם המחשב (ולפעמים גם ה-IP הפנימי של אותו המחשב) עליו נכתב הקובץ.
- שם המשתמש (ולפעמים גם שם הדומיין) שבחשבונו נכתב הקובץ.
- שם האפליקציה (ולרב גם גרסתה) שייצאה את הקובץ.
- סוג מערכת ההפעלה וגירסתה/הפצתה שעליה כתבו את הקובץ.
- מיקום הקובץ שכותב הקובץ בחר בעת שמירת הקובץ.
- כתובת דוא"ל של המשתמש שכתב את הקובץ.
- שמות/כתובות שרתים ברשת הפנימית המבצעים ניהול של הקבצים.
- שם/סוג/גרסאת המדפסת שבה השתמשו בכדי להדפיס את הקובץ.
- כמות הזמן שבו נכתב הקובץ.

כאשר מדובר ב-Meta Data שאנו בחרנו להכניס לקובץ, כגון שמו של הכותב, התאריך או המקום בו נכתב המאמר, אין מדובר בסיכון. אך במידה מדובר ב-Meta Data שמוזרק לקובץ באופן אוטומטי ע"י האפליקציה שבעזרתה כתבנו את הקובץ ואין לנו שליטה עליו (ולרב אנו בכלל לא מודעים שהמידע הוסף), מדובר פה בסיכון גדול מאוד שיכול לא רק לסכן את עמדת הקצה שממנה ביצענו את כתיבת הקובץ, אלא את כלל האירגון.

Meta Data in Microsoft Office

אחת מחבילות התוכנה היותר ידועות לשמצה בנושא השימוש ב-Meta Data "רגיש" היא חבילת ה-Office של חברת Microsoft.

לפי מיקרוסופט, ה-Meta Data העלול להשמר עם כל יצירת/שמירת קובץ הוא:

- Your name
- Your initials
- Your company or organization name
- The name of your computer
- The name of the network server or hard disk where you saved the document
- Other file properties and summary information
- Non-visible portions of embedded OLE objects
- The names of previous document authors
- Document revisions
- Document versions
- Template information
- Hidden text or cells
- Personalized views
- Comments

לצורך ההמחשה, במסגרת כתיבת המאמר ביצעתי חיפוש לקבצי Doc באתר Microsoft.com בעזרת גוגל:

Site:www.Microsoft.com FileType:doc

הגעתי לקובץ הבא: http://www.microsoft.com/rus/docs/mpa/eng/mpa.doc

בקובץ הנ"ל ישנו מאמר מעניין מאוד (כן בטח) של Microsoft שכותרתו הוא:

"Microsoft Product Activation"

במידה ואנתח את תוכנו של הקובץ באמצעות התוכנות הנכונות (לצורך המאמר אשתמש בתוכנה FOCA 3 RC שלאחרונה עברה שיפור משמעותי) אוכל לדלות נתונים מעניינים כגון:

- משתמשים שונים שהיו חלק מעריכת הקובץ:

Users	
Username	Leo
Username	Microsoft
Username	SergeyA
Username	Administrator

- מיקומים שונים שבוצע שימוש בהם על המחשב שממנו כתבו את הקובץ:

History	
Author	Leo
Path	G:\All Documents\Work\Design.ru\4 Ginsburg\Microsoft MPA\ENG_MPA.doc
Author	Leo
Path	G:\All Documents\Work\Design.ru\4 Ginsburg\Microsoft MPA\ENG_MPA.doc
Author	Leo
Path	G:\Documents and Settings\Administrator\Application Data\Microsoft\Word\AutoRecovery save of ENG_MPA.asd
Author	Leo
Path	G:\Documents and Settings\Administrator\Application Data\Microsoft\Word\AutoRecovery save of ENG_MPA.asd
Author	Leo
Path	G:\Documents and Settings\Administrator\Application Data\Microsoft\Word\AutoRecovery save of ENG_MPA.asd
Author	Leo
Path	G:\Documents and Settings\Administrator\Application Data\Microsoft\Word\AutoRecovery save of ENG_MPA.asd
Author	Leo
Path	G:\Documents and Settings\Administrator\Application Data\Microsoft\Word\AutoRecovery save of ENG_MPA.asd
Author	Leo
Path	G:\Documents and Settings\Administrator\Application Data\Microsoft\Word\AutoRecovery save of ENG_MPA.asd
Author	Leo
Path	G:\All Documents\Work\Design.ru\4 Ginsburg\Microsoft MPA\ENG_MPA.doc
Author	Leo
Path	G:\Documents and Settings\Administrator\Application Data\Microsoft\Word\AutoRecovery save of ENG_MPA.asd

- כתובת המייל של אחד המשתמשים:

Emails	
Email	sergeya@MICROSOFT.com

- האפליקציה שבאמצעותה נכתב הקובץ, החברה הרשומה ובנוסף-מערכת ההפעלה עליה רצה אותה אפליקציה בעת כתיבת הקובץ:

Other Metadata	
Application	Microsoft Office 2000
Operating system	Windows Server 2000
Company	Microsoft Corp.

איך הנתונים האלה יכולים לעזור לנו בעת פריצה לאירגון? לדוגמא, אני יכול לדעת שקיים סיכוי רב שאם אני אשלח מייל לכתובת: sergeya@MICROSOFT.com

המשתמש שיפתח את המאמר יהיה משתמש עם הרשאות Administrator.

דוגמא נוספת היא הקובץ הבא: (שגם אותו מצאתי באותה הדרך)

http://www.microsoft.com/korea/windows2000/docs/appsvcs_ko.doc

מאמר מאוד מעניין ומומלץ בקוראנית. ניתוח של הקובץ הנ"ל באותה התוכנה מאפשר לנו לדלות מידע נוסף על האירגון:

- משתמשים:

Users	
Username	KwonGee Kwak
Username	BED Web Team
Username	Noelle G. Knapp
Username	Microsoft
Username	박양우
Username	a-noelk

- מיקומים:

Folders	
Folder	C:\WINNT\Profiles\a-noelk\Application Data\Microsoft\Word\
Folder	C:\WINNT\Profiles\a-noelk\Desktop\
Folder	E:\Serapis_Documents\Wtc\Ms\0127\작업 파일\target\appsvcs\
Folder	C:\WINDOWS\바탕 화면\
Folder	\\SERVER3\WORK8\InsungInfo\Projects\WLIS907(MSCOM_W2K RTM)\Work\000121_RTM\DOC\target\appsvcs\
Folder	C:\WINDOWS\TEMP\

(אפשר לשים לב שבמקרה הנ"ל ה-Meta Data כולל כתובת המשויכת למשאב חיצוני ברשת הפנימית באירגון בשם: \\SERVER3)

- אפליקציה ומערכת הפעלה:

Other Metadata	
Application	Microsoft Office 97
Operating system	Windows 98

איך המידע הזה עוזר לנו? אנחנו יכולים לדעת כי לאחד המשתמשים שהשתתף בכתיבת המאמר יש גישה למשאב רשת נוסף: SERVER3. יש סיכוי שבמידה ונצליח להשיג גישה לחשבונו של המשתמש הנכון-נקבל גם גישה לאותו משאב חיצוני. המשאב הזה אולי לא מעניין אותנו כל כך, לעומת זאת מניתוח של הקובץ הבא:

http://www.microsoft.com/colombia/ftpfiles/premier.doc

אנחנו יכולים להבין כי למשתמש הבא:

Users	
Username	IAB

יש גישה לשרת הבא:

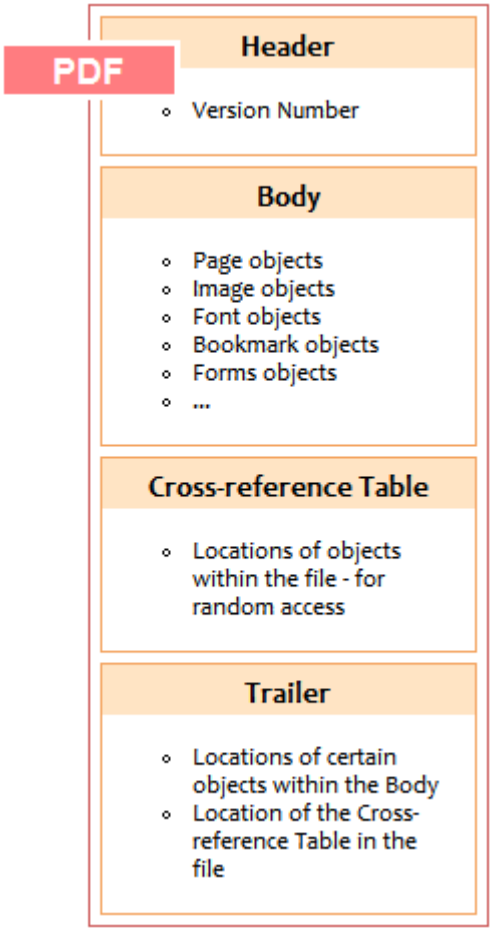
Folders	
Folder	\\MS_PRODUCCION\MSINETPUB\MS-Structure\Colombia\ftpfiles\

ועל-פי שמו אפשר להניח כי מדובר במשאב מעניין יותר.

שימו לב שאם נפתח את קבצי ה-Doc הללו בעזרת Word לא נוכל לראות שום חלק מה-Meta Data שראינו בקובץ עצמו, מפני שהוא מאוחסן מחוץ **לתחום** שבו קוראת Word.

גם בקבצי ה-PDF ישנו Meta Data רב המתווסף באופן אוטומטי וכברירת מחדל לקובץ בעת יצירתו, וגם פה מדובר במידע שיכול לחשוף פרטים רגישים על יוצרו של הקובץ.

קבצי PDF קצת יותר מסודרים בכל הנושא של ה-Meta Data אך עדיין חושפים פרטים היכולים לעזור לתוקף להבין דבר או שניים על האירגון שלנו. קובץ PDF סטנדרטי בנוי באופן הבא:

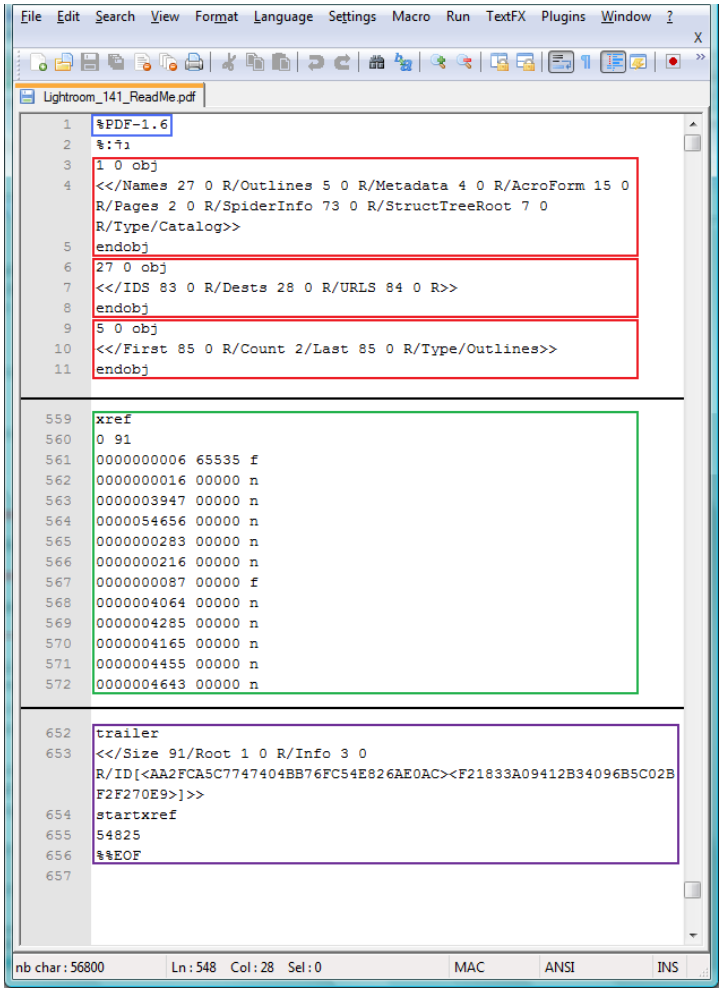


(התמונה נלקחה מאתר הבית של gnostice.com)

- בתחילת הקובץ נשמרת גרסתו (בכדי שהמפענחים ידעו באיזה אופן להציג אותו במקרים של שינוי עתידי במבנה הקובץ)

- לאחר מכן-מגיע גוף הקובץ, בגוף נשמרת רשימת האובייקטים המגדירים את הקובץ ומאפייניו.
- בחלק השלישי של הקובץ מופיעה טבלת ה-Cross-Reference (במבנה הקובץ היא מצויינת כ-xref) הטבלה תכלול את הכתובות של כל אובייקט.
- בחלק האחרון יופיעו נתוני מבנה הקובץ, כמו למשל-כתובתה של טבלת ה-xref או איזה אובייקט הוא האובייקט הראשי.

בכדי לפשט את הנושא, לקחתי קובץ PDF וחילקתי אותו לפני מבנה הנתונים שלו, שימו לב:



- **כחול** - Header.
- **אדום** - Objects List.
- **ירוק** - Cross Reference Table (xref).
- **סגול** - Trailer.

אז איפה בעצם נשמר ה-Meta Data בקובץ ה-PDF?

ה-Meta Data נשמר ב-Body, כחלק מנתוני האובייקטים הבלתי נראים של הקובץ (לא האובייקטים האחראים על מאפייניו-כגון סוג הפונט, צבע, גודל וכו', אלה האובייקטים האחראים על מאפייני הקובץ עצמו).

לאחרונה הנושא צבר תאוצה בעקבות הוויכוח/דיון שהתעורר ב-BugTraq בכל נושא ה-Meta Data בקבצי PDF לאחר ש-Inferno (הבחור מ-SecureThoughts.com) פרסם פוסט בשם:
"Millions of PDF invisibly embedded with your internal disk paths" בפוסט הוא מסביר על בעיה שהוא גילה במנגנון ייצוא הקבצים בפורמט PDF הקיים בדפדפן Internet Explorer. החשיפה היא, שבתהליך הייצוא, הדפדפן מוסיף כ-Meta Data לקובץ ה-PDF את המיקום (Path) של הקובץ על המחשב המקומי, והוא מסביר את התהליך:

1. Pick a .HTM or .HTML or .MHT file on your local computer.

2. Open this file in IE and click Ctrl-P.

(OR Right-click the file in explorer and select PRINT from context menu.)

4. Select any PDF writer as Printer such as Adobe PDF / CutePDF / PrimoPDF / etc.

5. Click Print. When the PDF writer asks for a filename, provide any name.

6. Open the generated pdf in notepad, and search for “file://” without quotes

בכדי להראות עד כמה נרחב השימוש בפונקציה, הוא מציע לבצע את החיפוש:

filetype:pdf file c (htm OR html OR mhtml)

במנוע החיפוש גוגל או במנוע החיפוש בינג.

לדוגמא ל- Meta Data "עוין" בקבצי PDF, ביצעתי את החיפוש הבא:

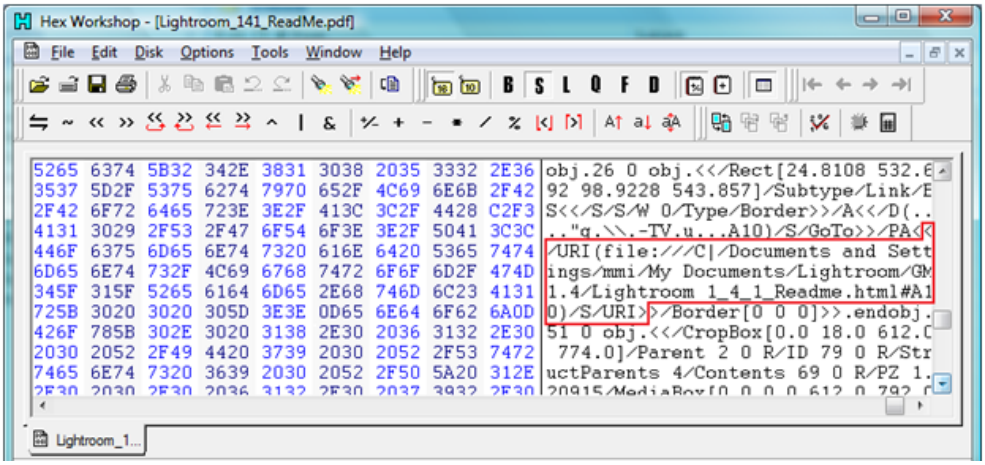
site:www.adobe.com filetype:pdf file c (htm OR html OR mhtml)

התוצאה הראשונה שיצאה היא:

http://www.adobe.com/special/photoshop/Lightroom_141_ReadMe.pdf

(קובץ ה-Read-Me של Photoshop Lightroom)

אם נפתח את הקובץ בעזרת Hex Editor כגון Hex Workshop ולא בעזרת תוכנת PDF Reader כגון Acrobat Reader, נוכל לראות את הדבר הבא:



שימו לב שאפשר לראות כי אכן התווסף ה-Path שבו נשמר הקובץ בעת שמירתו כ-Meta Data, בנוסף שימו לב שלפי ה-Path נוכל לראות את שמו של המשתמש שבו השתמש כותב המאמר.

שימו לב כי לא משנה באיזו תוכנת PDF Writer השתמשו בכדי ליצור את הקובץ, הבעיה אינה נמצאת שם אלה נמצאת במנגנון הייצוא של Internet Explorer.

האם הסיכון ממשי?

ניתוח של קובץ יחיד אומנם לא ייתן לנו מידע יעיל אשר יכול לעזור בפריצה לאירגון מסויים, אך במידה וננתח כמות גדולה של קבצים מאותו אירגון, נוכל לבצע מיפוי (חלקי אך משמעותי) של משתמשיו, כתובות המיילים שלהם, ונגישותם למשאבי הרשת השונים.

לדוגמא, חיפוש בגוגל תחת המחרוזות הבאות:

Site:www.Microsoft.com FileType:doc

Site:www.Microsoft.com FileType:docx

Site:www.Microsoft.com FileType:ppt

Site:www.Microsoft.com FileType:pptx

Site:www.Microsoft.com FileType:xls

Site:www.Microsoft.com FileType:xlsx

Site:www.Microsoft.com FileType:pdf

הביאו ביחד יותר מ-2600 קבצים שניתן לדלות מהם מידע על הרשת הפנימית של האירגון היושב תחת הדומיין: www.Microsoft.com

לנתח כל קובץ באופן ידני אכן בלתי אפשרי כשמדובר במספר קבצים מספיק גדול בכדי לתת לנו תמונה ולו קטנה יחסית לגבי אירגון מסויים. חבר'ה רציניים מ-informatica64 כתבו כלי בשם FOCA RC3 (בקרוב יוצאת הגרסא FOCA 2) שמקבל שאילתה כגון:

Site:www.Domain.com FileType:doc

ויודע לאתר את כל התוצאות המופיעות במנועי החיפוש גוגל ו-בינג, להוריד אותן ולבצע מהן ניתוח של הרשת הפנימית של אותו אירגון הכולל בין השאר:

- שמות משתמשים באירגון.
- כתובות מייל המשמשים את המשתמשים באירגון.
- הרשאות למשאבי רשת מרוחקים (Remote Users/Folders)
- מיקומים של תיקיות על מחשבים/שרתים ספציפיים באירגון.

- מיקומי מדפסות מקומיות/מרוחקות באירגון.
- גרסאות של תוכנות המותקנות על מחשבים ספציפיים באירגון.
- שמות שרתים באירגון.

התוצאות מדהימות-שווה לנסות.

דרכי התגוננות

כאשר מדובר באירגון גדול קשה להתגונן מפני הסכנה הזאת, אך במידה ומדובר בקובץ ספציפי עליו רוצים להגן קיימות שתי אפשרויות:

- להשתמש בכלי ייעודי לנושא.
- לדרוס את המידע באופן ידני.

הדרך הראשונה פשוטה ונוחה יותר בדרך כלל, אך אי-אפשר להשתמש בה בכל סוגי הקבצים מפני שלא לכל סוגי הקבצים קיימים כלי "Metadata scrubber". במידה ומדובר בקבצי Office כגון Doc, Ppt, Xls וכו' ניתן להשתמש בתוכנה Remove Hidden Data מבית מיקרוסופט, או בתוכנה Doc Scrubber למסמכי Word.

במידה ואנו משתמשים בקבצים שאין להם תוכנה ייעודית- או שנעדיף לבצע זאת ידנית בכדי לחסל כל סיכוי ל-Meta Data פשוט נפתח את הקובץ באמצעות כל כלי Hex Editor וכך נוכל לערוך את כל ה-Meta Data הקיים בקובץ למה שנרצה.

כאשר מדובר בארגון, נהלי חברה מסודרים יכולים לצמצם את הבעיה. למשל - נוהל עבור אחראי האתר המחייב לבצע דריסת meta data בכל קובץ Word העולה לאתר התוכן של האירגון.

סיכום

השימוש ב-Meta Data חשוב מאוד ובמקרים מסויימים הוא יכול ליעל את אירגון המידע והנגישות אליו באופן משמעותי, אך חשוב מאוד לזכור כיצד נכון להשתמש בו, לא בכל המקרים ולא בכל סוג של Meta Data הדבר נכון. אנו חייבים תמיד לדעת איזה מידע אנחנו משתפים עם שאר האינטרנט והאם המידע הזה יוכל לשמש כנגדנו באחד הימים.

הפסקת קפה

מאת צבי קופר

אם אתה לא איש רשויות החוק או סוכן FBI לצורך העניין, סביר להניח כי לא תוכל לקבל קפה מחברת מיקרוסופט לצורך עבודתך...

רקע

Cofee, או בשמו המלא- **Computer Online Forensic Evidence Extractor**, הוא למעשה כלי למטרות Forensics או יותר נכון טרום Forensics למערכות מבוססות Windows שפותח ע"י מיקרוסופט ומופץ דרך NW3C (National White Collar Crime Center). הכלי מיועד לסייע לאנשי רשויות החוק לבצע איסוף נתונים מהיר ONLINE בזירת הפשע. התהליך מתוכנן לרוץ על גבי מדיה נתיקה ובמינימום מעורבות של החוקר. כלומר, החוקר מגיע לזירת הפשע, מחבר את ה-DOK שהוכן מראש. הכלי רץ על התחנה החשודה בצורה אוטומטית, אוסף את המידע הרלוונטי ושומר אותו על גבי ה-DOK. צורת העבודה יעילה ופשוטה ואמורה לספק יכולת איסוף נתונים קריטיים גם לאנשי חוק ללא רקע טכנולוגי מעמיק.

בשנה האחרונה הסתובבו ברשת שמועות רבות סביב הכלי החסוי והסודי הזה שדלף לרשת בחודשים האחרונים ואף הספיק להכתב לו כלי אנטי-פורנזי בשם **DECAF** שיוצרו קיבל "המלצה" מהאינטרפול להורידו מהרשת ואכן האתר ירד לכמה חדשים אך בדצמבר 2009 הוא חזר לפעילות.

במאמר זה אתאר את יכולתיו של הכלי וכיצד הם מיושמים בו.

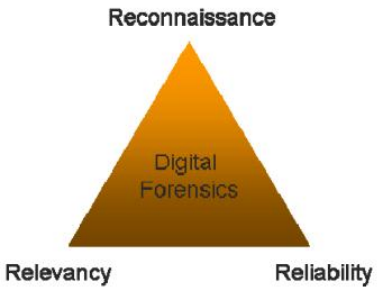
"הצהרת כוונות"

המאמר הוא לצורכי לימוד בלבד ונועד לאתגר את קהילת ההאקרים בכובע לבן, לשכלל, לתקן או ליצור טכנולוגיות טובות יותר הן במישור האבטחתי והן במישור ההתגוננות (anti forensic) .

ONLINE FORENSICS - על קצה קצה המזלג.

אחת הבעיות הגדולות בחקירה פורנזית היא איבוד הנתונים הפוטנציאלי שעלול להתרחש כתוצאה מכיבוי המחשב וניתוקו. נתונים כגון: **תהליכים המוטענים לזיכרון, נתוני רשת, קבצים פתוחים, שיתופים וגישות למחשבים מרוחקים, תוכנות התקשרות הפועלות ברקע, הצפנות, שמות משתמשים, שירותים מופעלים, נתוני גלישה, DNS, טבלאות ניתוב, מידע הנשמר בקבצים זמניים**, כל אלה הם "הלחם והחמאה" של החוקר הפורנזי והסיכון באיבודם גבוה עד וודאי בניתוק המחשב מהחשמל והרשת.

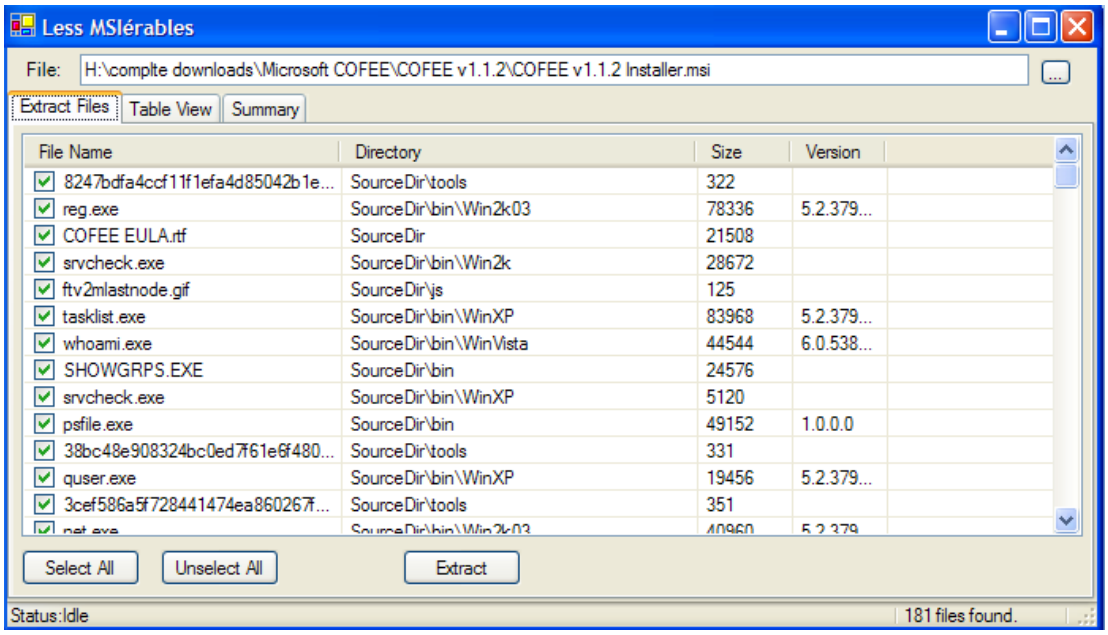
נושא חשוב נוסף הינו המשולש הפורנזי.



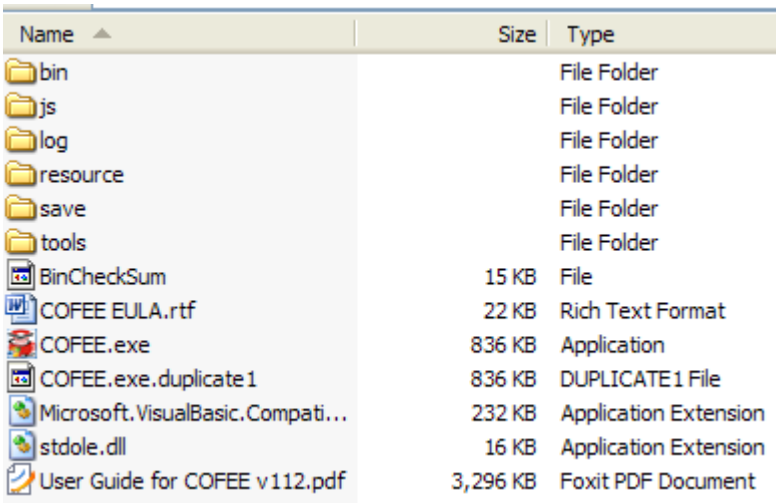
המטרה היא לשמור על איזון בין איסוף כמות רבה של נתונים, מידת הרלוונטיות של הממצאים לחקירה ומידת האמינות של החומר הנאסף. בנוסף תהליך איסוף הנתונים אמור לשאוף להשארת מינימום עקבות (FOOTPRINT) על האובייקט הנבדק וכן לאפשר מנגנון בקרה וחתימה המאשר כי הנתונים מהימנים ולא עברו על שינוי במהלך הבדיקה. העובדה שכלים כמו coffee משמשים לאיסוף נתונים ONLINE בסביבה לא מבוקרת ולעתים בידי ידיים לא מקצועיות, מחייבת את המפתחים לשמור על כללי המשולש ביתר שאת.

בואו נצלול לפרטים הטכניים:

ההתקנה המתגלגלת ברשתות שיתוף הקבצים מגיעה כקובץ MSI אך לא ניתן להתקין אותו ישירות מסיבה לא ברורה ולכן נשתמש באחת מתוכנות ה-MSI EXTRACT ונחלץ את הקבצים לספריות בהתאמה.

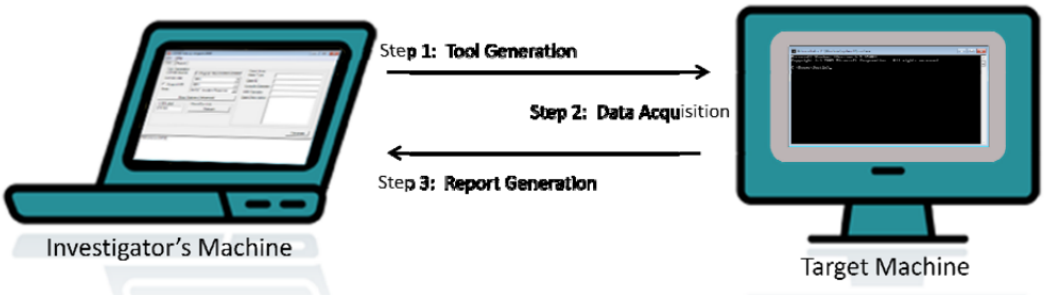


התוצאה שאמורה להתקבל היא מערך הספריות הבא:



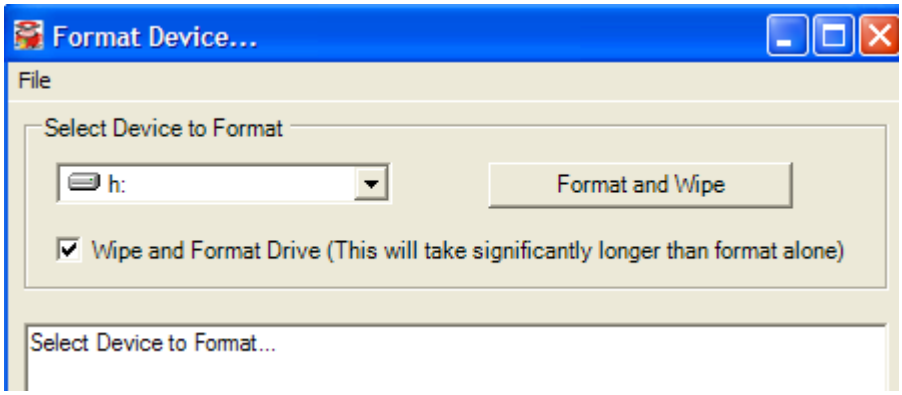
השימוש ב-COFEE מחולק לשלושה שלבים:

1. יצירת הכלי
2. איסוף המידע
3. יצירת הדו"ח



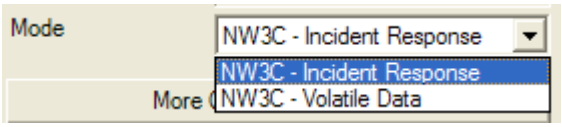
1. יצירת הכלי

בשלב הראשון נכין מדיה נתיקה נקייה ללא חשש משיירי מידע קודמים - התוכנה מאפשרת מחיקה עמוקה (SDELETE-wipe) ופירמוט למדיה שנבחר. גודל המדיה המומלץ הוא לפחות 2GB והפירמוט הוא ל-fat32.

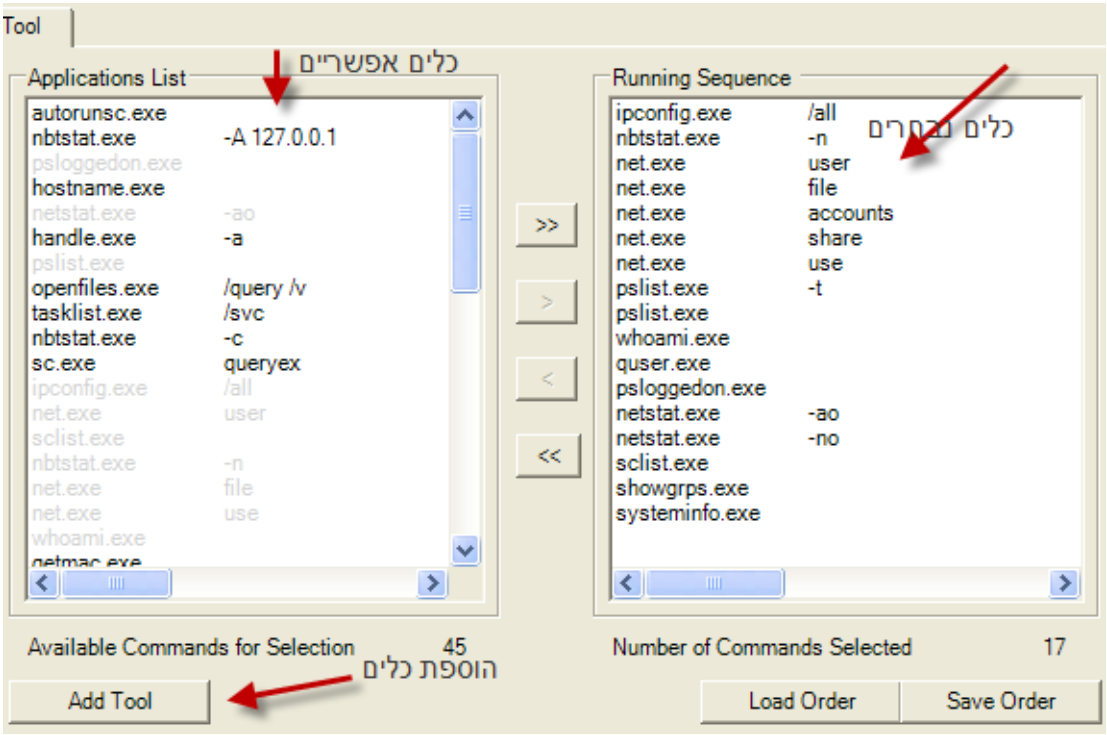


לאחר שהכנו את ה-DOK, ניגש לבניית הכלי עצמו:

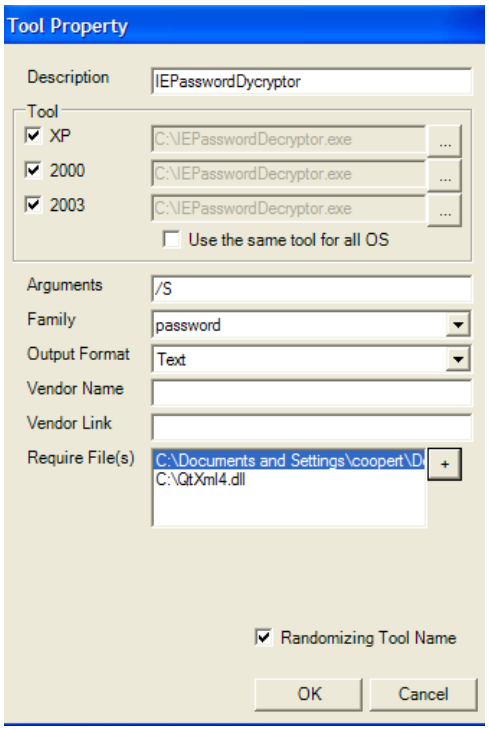
COFEE מגיע עם אוסף כלים המיועדים לאיסוף מידע ONLINE ומחולקים ל-2 פרופילים: incident response ו-volatile data. ההבדל בין הפרופילים הוא בכמות הפעולות שיופעלו על המחשב הנחקר כאשר incident response היא המעמיקה מביניהם .



בנוסף הכלי מאפשר הוספה והתאמה של כלים נוספים כולל הוספת פרמטרים וקבצי DLL הדרושים עבור ההרצה. כאן באמת אפשר "לחגוג" ולהוסיף כלים פיקנטיים. לשליפת ססמאות, צילומי מסך ועוד. אך יש כמובן לזכור כי הכלי מיועד לפעולה מהירה...



דוגמא לכלי שהוספתי לתיעוד סיסמאות מה-internet explorer:



התוכנה טוענת את הקבצים ומוסיפה אותם לרשימה האפשרית להרצה.

אוסף הכלים המגיעים עם COFEE מבוססים על כלי מיקרוסופט ידועים ופקודות מוכרות. הדבר עלול לגרום לחלק מהקוראים להרמת גבה אך יש לזכור כי הכוח כאן הוא בקיבוץ התוכניות תחת מנגנון אחד, יכולות ההרחבה שראינו והאוטומציה המלאה ONLINE.

כמו כן קיימות פקודות שונות המבצעות את אותה עבודה וזאת לצורך בדיקה השוואתית ואימות התוצאות המוצגות בד"ח הסופי.

היבט נוסף שנלקח בחשבון הינו האמינות המשפטית. כלומר, מהי הבדיקה שמתבצעת והאם הכלי עושה את מה שהוא אמור לעשות ולא דבר אחר? האם הפקודה משנה ערכים במחשב הניבדק? לצורך כך מיקרוסופט מספקת עם ההתקנה 200MB של מידע מפורט בקבצי EXCEL על פעילות הפקודות.

לדוגמא חלק מתיעוד הרצת קובץ at.exe:

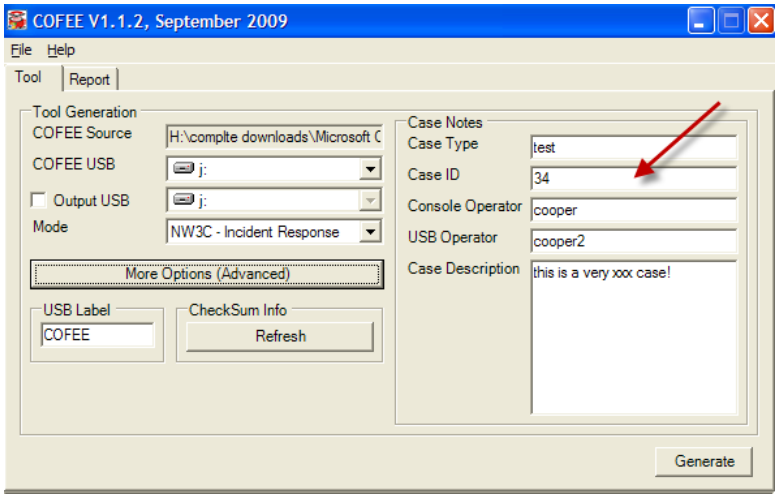
	A	B	C	D	E	F	
1	Time of Day	Process Name	PID	Operation	Path	Result	Detail
2	52:26.0	at.exe	3336	Load Image	C:\WINDOWS\system32\ntdll.dll	SUCCESS	Image Base: 0x7c900000, Image Size: 0xb0000
3	52:26.0	at.exe	3336	CreateFile	C:\WINDOWS\Prefetch\AT.EXE-1891128A.pf	SUCCESS	Desired Access: Generic Read, Disposition: Ope
4	52:26.0	at.exe	3336	QueryStandardInformationFile	C:\WINDOWS\Prefetch\AT.EXE-1891128A.pf	SUCCESS	AllocationSize: 16,384, EndOfFile: 13,062, Numb
5	52:26.0	at.exe	3336	ReadFile	C:\WINDOWS\Prefetch\AT.EXE-1891128A.pf	SUCCESS	Offset: 0, Length: 13,062
6	52:26.0	at.exe	3336	CloseFile	C:\WINDOWS\Prefetch\AT.EXE-1891128A.pf	SUCCESS	
7	52:26.0	at.exe	3336	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\at.exe	NAME NOT FOUND	Desired Access: Read
8	52:26.0	at.exe	3336	Load Image	C:\WINDOWS\system32\kernel32.dll	SUCCESS	Image Base: 0x7c800000, Image Size: 0xf4000
9	52:26.0	at.exe	3336	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Access: Read
10	52:26.0	at.exe	3336	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11	52:26.0	at.exe	3336	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	
12	52:26.0	at.exe	3336	Load Image	C:\WINDOWS\system32\msvcrt.dll	SUCCESS	Image Base: 0x77c10000, Image Size: 0x58000
13	52:26.0	at.exe	3336	Load Image	C:\WINDOWS\system32\advapi32.dll	SUCCESS	Image Base: 0x77dd0000, Image Size: 0x90000
14	52:26.0	at.exe	3336	Load Image	C:\WINDOWS\system32\rpcrt4.dll	SUCCESS	Image Base: 0x77e70000, Image Size: 0x91000
15	52:26.0	at.exe	3336	Load Image	C:\WINDOWS\system32\netapi32.dll	SUCCESS	Image Base: 0x5b860000, Image Size: 0x54000
16	52:26.0	at.exe	3336	Load Image	C:\WINDOWS\system32\shell32.dll	SUCCESS	Image Base: 0x7c9c0000, Image Size: 0x814000
17	52:26.0	at.exe	3336	Load Image	C:\WINDOWS\system32\cqdi32.dll	SUCCESS	Image Base: 0x77f10000, Image Size: 0x46000
18	52:26.0	at.exe	3336	Load Image	C:\WINDOWS\system32\user32.dll	SUCCESS	Image Base: 0x77d40000, Image Size: 0x90000
19	52:26.0	at.exe	3336	Load Image	C:\WINDOWS\system32\shlwapi.dll	SUCCESS	Image Base: 0x77f60000, Image Size: 0x76000
20	52:26.0	at.exe	3336	QueryOpen	C:\WINDOWS\system32\shlmenq.dll	SUCCESS	CreationTime: 8/4/2004 12:56:46 AM, LastAccess
21	52:26.0	at.exe	3336	CreateFile	C:\WINDOWS\system32\shlmenq.dll	SUCCESS	Desired Access: Execute/Traverse, Synchronize,
22	52:26.0	at.exe	3336	CreateFileMapping	C:\WINDOWS\system32\shlmenq.dll	SUCCESS	SyncType: SyncTypeCreateSection, PageProtect
23	52:26.0	at.exe	3336	CreateFileMapping	C:\WINDOWS\system32\shlmenq.dll	SUCCESS	SyncType: SyncTypeOther
24	52:26.0	at.exe	3336	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	NAME NOT FOUND	Desired Access: Query Value, Set Value
25	52:26.0	at.exe	3336	RegOpenKey	HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCESS	Desired Access: Query Value
26	52:26.0	at.exe	3336	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers\TransparentEnabled	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
27	52:26.0	at.exe	3336	RegCloseKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCESS	
28	52:26.0	at.exe	3336	RegOpenKey	HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	NAME NOT FOUND	Desired Access: Query Value
29	52:26.0	at.exe	3336	CloseFile	C:\WINDOWS\system32\shlmenq.dll	SUCCESS	
30	52:26.0	at.exe	3336	Load Image	C:\WINDOWS\system32\shlmenq.dll	SUCCESS	Image Base: 0x5cb70000, Image Size: 0x26000
31	52:26.0	at.exe	3336	CreateFile	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS	Desired Access: Generic Read, Disposition: Ope
32	52:26.0	at.exe	3336	QueryStandardInformationFile	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS	AllocationSize: 1,191,936, EndOfFile: 1,190,796,
33	52:26.0	at.exe	3336	CreateFileMapping	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS	SyncType: SyncTypeCreateSection, PageProtect
34	52:26.0	at.exe	3336	QueryStandardInformationFile	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS	AllocationSize: 1,191,936, EndOfFile: 1,190,796,
35	52:26.0	at.exe	3336	CreateFileMapping	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS	SyncType: SyncTypeOther

להלן כמה מקבצי ההרצה והפקודות המגיעות עם coffee:

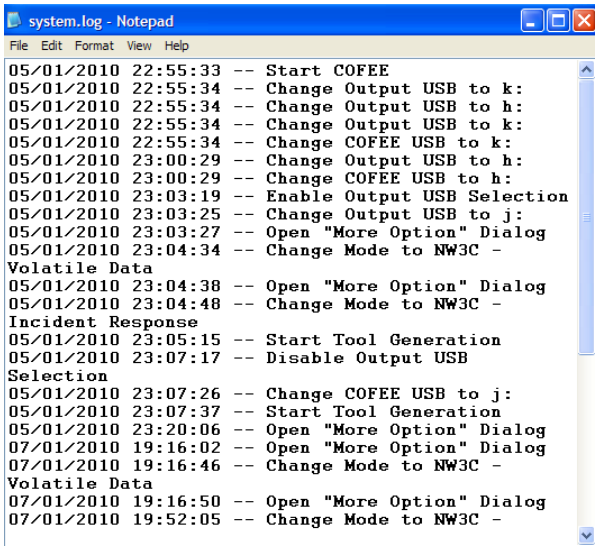
1. Arp -a תיעוד כתובת ה-MAC של התחנה.
2. autorunsc.exe - תיעוד התוכנות העולות אוטומטית באתחול התחנה.
3. handle.exe -a - תיעוד הגישה של תהליכים פתוחים לקבצים, רשומות PORTS REG ועוד.
4. getmac.exe - תיעוד נוסף של כתובות ה-MAC בתחנה (מפורט יותר מ-arp ומקיף את כל התקני הרשת).
5. at.exe - תיעוד משימות מתוזמנות במערכת.
6. Ipconfig /all - תיעוד כתובות ה-GATEWAY IP DHCP וכו'.
7. Hostname - תיעוד שם התחנה.
8. msinfo32.exe - תיעוד פרטי מערכת רבים.
9. psloggedon.exe - תיעוד שמות משתמשים ב-LOGON מקומי ומרוחק דרך שיתופים.
10. netstat -ao - תיעוד פורטים והתקשרויות ברשת + קישור התחברות לתוכנה ספציפית בתחנה (PID).
11. pslist.exe - תיעוד תהליכים (processes) המוטענים לזיכרון.
12. openfiles.exe /query /v - תיעוד קבצים פתוחים מקומית ועל גבי שיתופים מרוחקים.
13. Taskkist.exe /svc - תיעוד נוסף של processes אך עם יכולת לראות תת פרוססים.

14. Nbtstat.exe -c - תיעוד טבלאות NetBIOS בתחנה הכוללים שמות תחנות מרוחקות וכתובת ה-IP שלהן.
15. Sc.exe queryex - תיעוד השירותים הרצים על התחנה (במצב running).
16. net user - תיעוד שמות משתמש בתחנה.
17. Net file - תיעוד נוסף של קבצים פתוחים.
18. Net use - תיעוד מיפויי כוננים מקומיים ומרוחקים.
19. Net view - שיתופים פתוחים עם אפשרות לראות את כל השיתופים הפתוחים ב-DOMAIN.
20. Net localgroup - תיעוד קבוצות security מקומיות על התחנה.
21. Net localgroup administrators - תיעוד שמות המשתמשים החברים בקבוצת האדמין המקומי.
22. net session - תיעוד ה-sessions הפתוחים הקיימים כרגע על התחנה.
23. SHOWGRPS.EXE - תיעוד נוסף של חברות המשתמש הפעיל בקבוצות.
24. SCLIST.EXE - ע"ע 15.
25. Whoami.exe - זיהוי נוסף של שם המשתמש הנמצא ב-LOGIN.
26. Pstat.exe - תיעוד מידע נוסף ומפורט על PROCCESS הרצים במערכת, הדגש כאן הוא על ה-DRIVERS הפועלים במערכת (מצויין בסוף שורת הפלט)
27. uptime.exe - תיעוד זמן המערכת מה-RESTART האחרון.
28. route print - תיעוד טבלאות הניתוב.

לאחר שבחרנו כלים רצויים והוספנו כלים משלנו, כל שנותר לעשות הוא למלא את פרטי המקרה (CASE) שם החוקר, הערות וכו', ולייצר את ה-DOK הסופי שלנו.

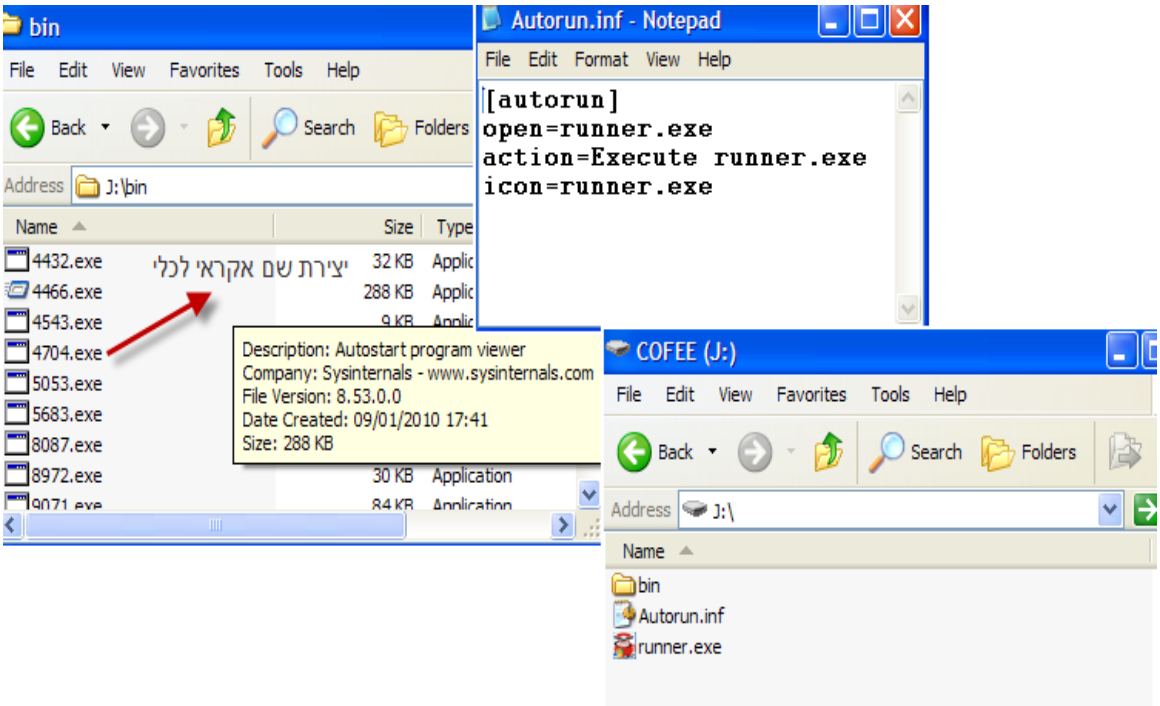


רגע לפני יצירת ה-DOK אציין כי קיים קובץ LOG לכל הפעולות הנעשות בכלי (כראוי).

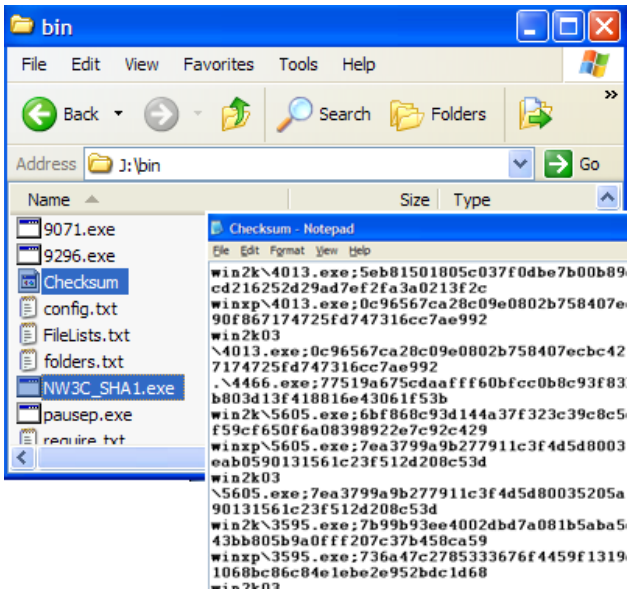


זהו. אם עשינו הכל נכון, נקבל DOK מוכן המשתמש בפונקצית ה-AUTORUN לצורך הרצה אוטומטית.

התוצאה נראית ככה:

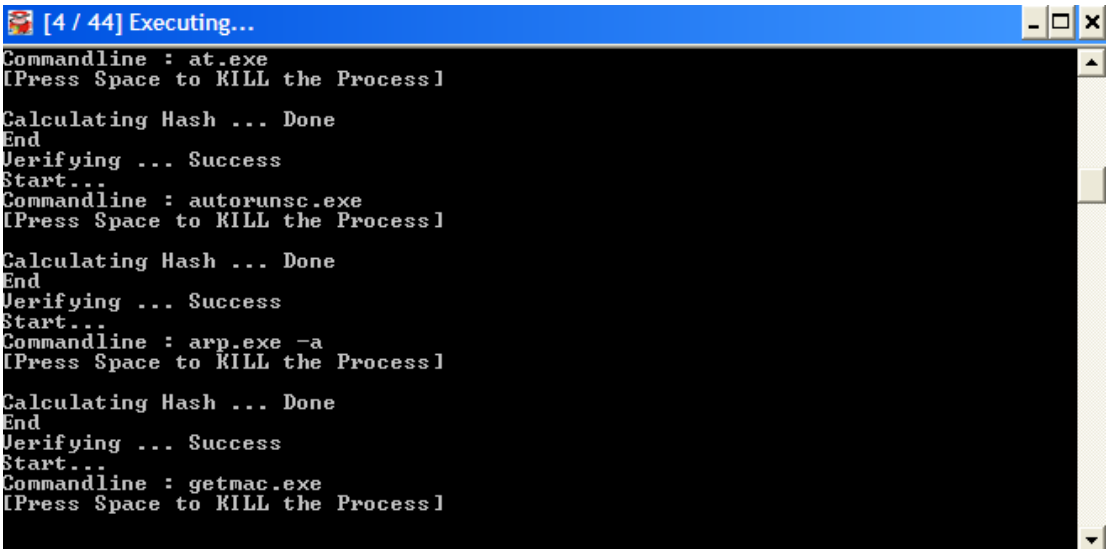


כמובן לא נתעלם מה-checksum המוודא כי תכולת ה DOK לא תשתנה:

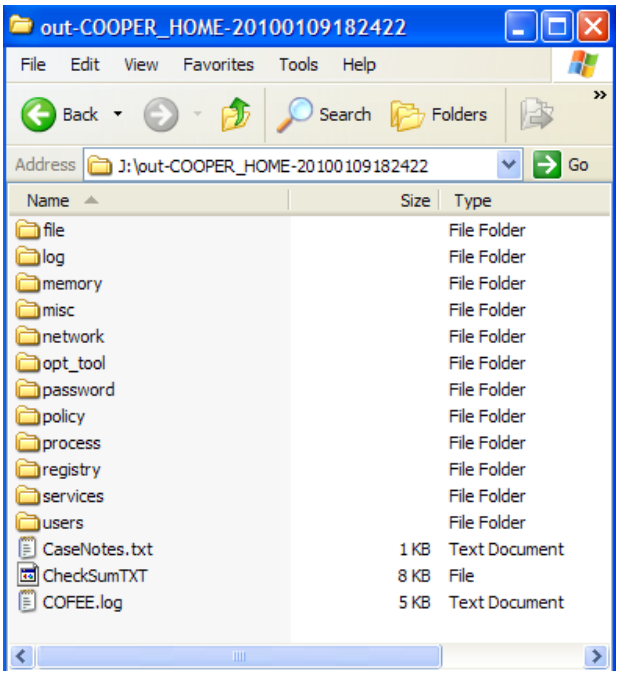


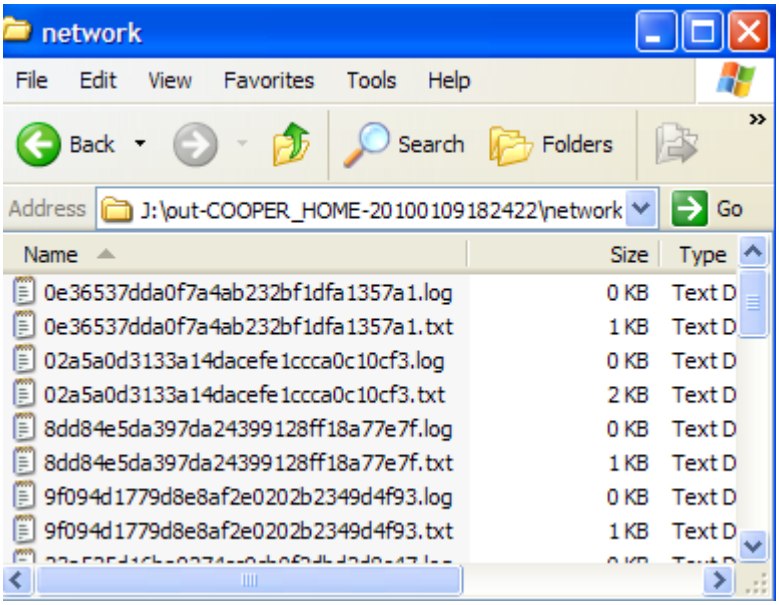
2. איסוף המידע

כאשר נחבר את ה-DOK לתחנה החשודה, הכלי יטען דרך ה-AUTORUN. במידה ופונקציה זו מבוטלת בתחנה, נאלץ להריץ את ה-RUNNER ידנית. הכלים ירוצו אחד אחרי השני כאשר קיימת אפשרות לבצע KILL לכלי ולעבור לכלי הבא. כמו כן מתבצע HASH על כל תוצאה שמתקבלת.



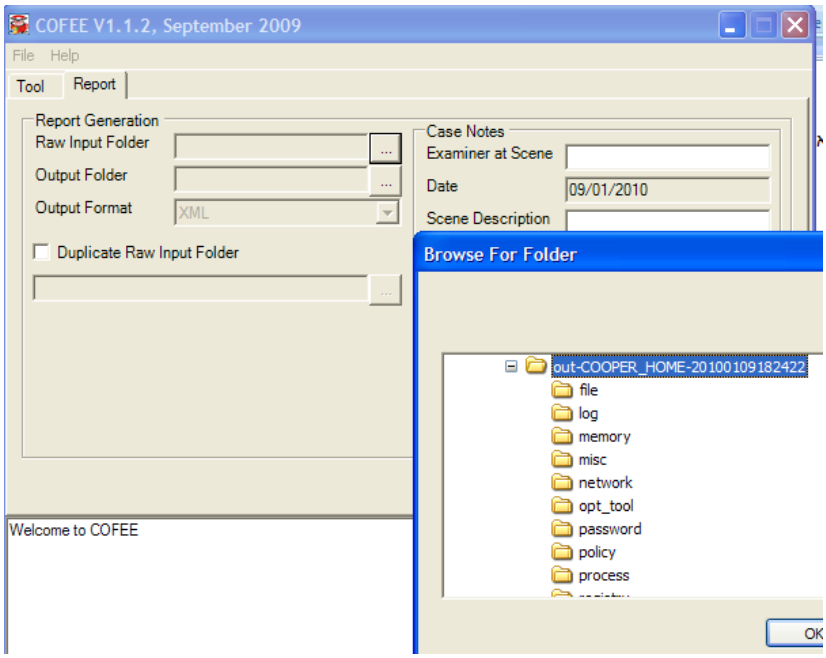
התוצאות נשמרות על ה-DOK ומחולקות לתיקיות לפי נושא הבדיקה כאשר שמות הקבצים הם ה-HASH המחושב בסוף הבדיקה. קיים לוג נוסף המתעד את פעילות הכלים.



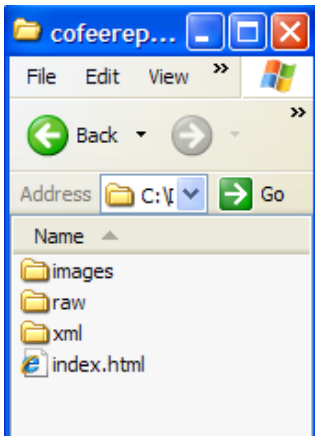


3. יצירת הדוח

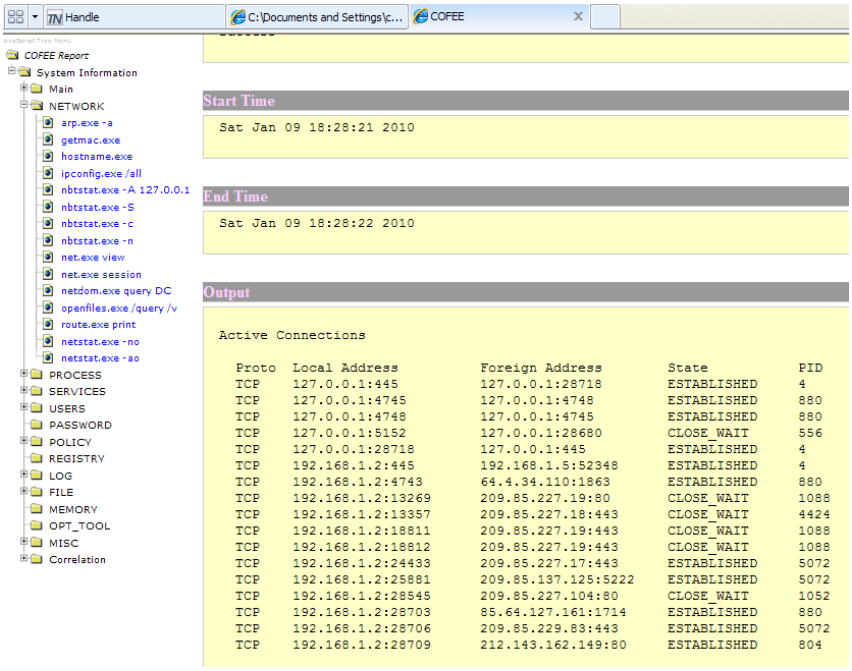
סיימנו עם איסוף הנתונים, כעת ניתן לנתק את ה-DOK ולטעון את הקבצים בעמדת העבודה :



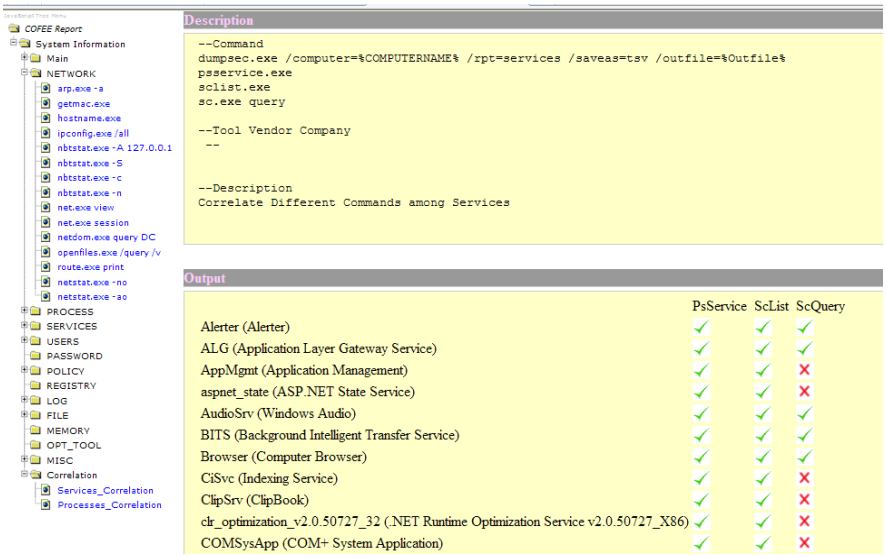
נבחר ספריה מקומית לשמירת הדו"ח, נבצע GENERATE - ה-checksum יבדק ותחל המרת הקבצים ל-XML. בסיום התהליך יווצרו הקבצים והתיקיות הבאים:



התוצאה הסופית שמתקבלת היא דו"ח HTML מפורט ומסודר לפי קטגוריות :



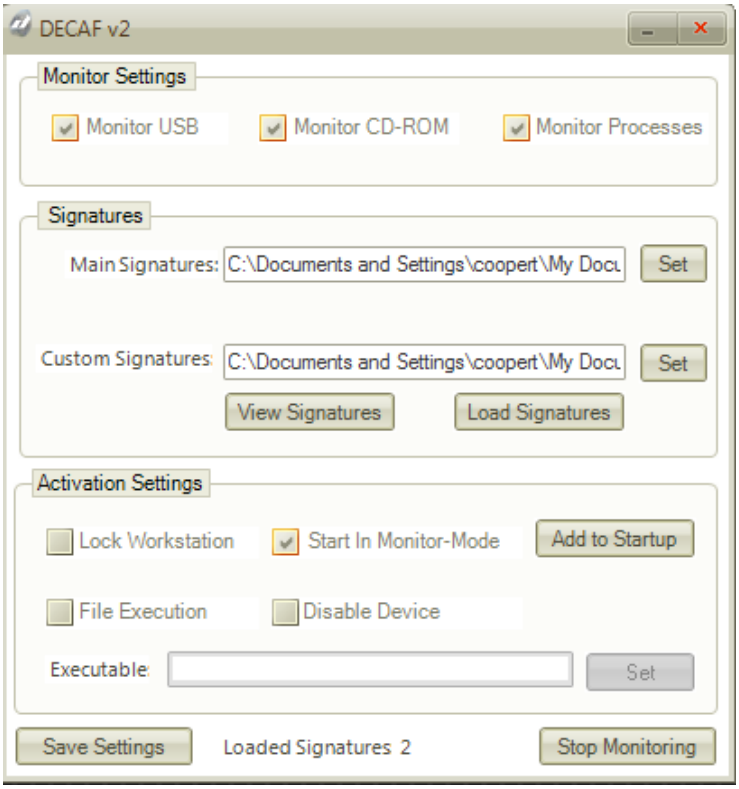
שימו לב לרובריקת ה-correlation אשר מציגה נתונים השוואתיים בין כלים שונים המבצעים אותה עבודה וזאת לצורך אימות נתונים נוסף.



לא רע ל-10 דקות עבודה, לא?

Coffee and more anti forensic tool-DECAF

נכון לזמן פרסום מאמר זה, פורסמה גרסה שניה לכלי שאמור לזהות חיבור coffee לתחנה ולנטרלו. נכון לעת כתיבת שורות אלו, לא הצלחתי לגרום ל-decaf לזהות את coffee למרות כל הנסיונות, המתכנת אכן רשם באתר כי קיימות שגיאות בקוד והוא עובד עליהן לגרסה הבאה... כנראה שבשלב זה נמשיך לקבל את הגרסה רבת הקפאין...:)



דיברנו קצת על ONLINE FORENSICS ועל הצרכים המיוחדים הדרושים לעבודה זו. ניתחנו לעומק כלי פשוט ויעיל, ובעל יכולות הרחבה השומר ככל שניתן על המשולש הפורנזי ועובד by the book בכל שלב ושלב על מנת לשמר את האיזון בין הכמות, הרלוונטיות והאמינות של הנתונים.

"The debate isn't security versus privacy. It's liberty versus control." -ברוס שנייר

לקריאה נוספת:

- http://news.cnet.com/8301-10789_3-9932600-57.html
- http://www.interpol.int/public/ICPO/PressReleases/PR2009/PR200937.asp
- https://cofee.nw3c.org/
- DECAF - http://www.decafme.org/

כלים ללינוקס:

- http://www.securityfocus.com/infocus/1503

בניית אתרים וסמנטיקה (Semantic HTML)

מאת ניר אדר

פתיחה

Semantic HTML פירושו להשתמש ב-HTML כדי להגדיר משמעות (סמנטיקה) לחלקים בטקסט שלנו, בניגוד להגדרה בה משתמשים מתכנתים רבים, המגדירה רק את עיצוב הטקסט שלנו. נציג דוגמא כדי להעביר את הכוונה. נניח שאנחנו רוצים לעשות כותרת לדף הבית שלנו. דרך אחת יכולה להיות:

```
<font size="14px" color="#000"><b>Welcome to the site</b></font>
```

במקרה הזה קשה למי שקורא את הקוד שלנו להבין שמדובר בכותרת. לעומת זאת, אנחנו יכולים לכתוב:

```
<h1>Welcome to the site</h1>
```

במקרה הזה ברור לחלוטין שמדובר בכותרת של האתר. לתג יש משמעות של כותרת. יתכן שמבחינת התצוגה על המסך שני קטעי הקוד יתנו תוצאה זהה, אבל קטע הקוד השני נותן לקורא מידע נוסף על הטקסט שהוא מעצב - הוא אומר לו "זו הכותרת של הדף".

למה חשוב לנו ליצור קוד עם משמעות?

1. מנועי חיפוש יבינו יותר טוב את הדף שלנו ואת התכנים שיש בו.
2. קוד הדף יהיה ברור יותר עבורנו, ועבור מתכנתים נוספים שיצפו בקוד הדף. המשמעות של חלקי הקוד השונים והחלקים השונים בדף תהיה מהירה יותר לקורא.
3. הפרדת העיצוב מהתוכן!
4. קוד נוח לתחזוקה ולשינוי:
   a. אין טבלאות לצורך עיצוב (כי העיצוב נעשה ב-CSS) ומכאן -אין עומס בדף עצמו - הרבה יותר קל לקרוא את הקוד.
   b. הקוד נקי יותר וקצר יותר, ומכאן - ברור יותר להבנה.
   c. הרבה יותר קל לבצע שינויים בעיצוב האתר.

5. אין לכם ברירה - תגים המיועדים לעיצוב בלבד, ללא משמעות, צפויים להעלם בגרסאות הבאות של HTML.
6. סיבות רבות נוספות.

איך נושא זה מתקשר לגליון שלנו? המאמר המצויין של אפיק דיבר על Meta Data. שימוש ב- Semantic HTML הוא בעצם הוספת Meta Data לדפי האינטרנט שאנחנו כותבים. אחרי המאמר שהציג את הסכנות ב-Meta Data, במאמר זה אני רוצה להראות לכם כמה מהיתרונות, ובדרך להציג את הנושא החשוב של Semantic HTML, שלטעמי חייב להיות הרבה יותר מוכר ונפוץ ממה שהוא היום.

המטרה של שפת ה-HTML לא היתה ואינה גם כיום לעצב את הדפים שלנו. המטרה של HTML היתה להוסיף לטקסט מידע שיגדיר את חלוקת הטקסט, ושיתן לו משמעות (סמנטיקה) נוספת. האתרים הראשונים בשפת HTML היו אתרים נכונים מאוד מבחינה סמנטית, אבל מכוערים - מתכנתים כתבו אותם, והשתמשו בתגיות השונות עם עיצוב ברירת המחדל. תגי העיצוב הוכנסו לשפת HTML במהלך שנות התשעים של המאה ה-20. הם גרמו לרווח מצד אחד מבחינת מראה האתרים, אך גרמו מהצד השני לקוד שערבב עיצוב ותוכן, והפך פחות קריא ופחות ברור.

CSS בא כדי לעזור למתכנתים להפריד את העיצוב מהתוכן. עם זאת, באתרים רבים ברשת עד היום הנושא לא מבוצע כראוי.

מעט רקע בנושא HTML

אני רוצה להתחיל את הדיון על סמנטיקה במעט רקע על שפת HTML, ועל המרכיב הבסיסי בה - HTML Element. באופן בסיסי, מסמכי HTML מורכבים מתחילתם ועד סופם מ-"רכיבי HTML". מבנה של רכיב HTML טיפוסי:

```
<tag>content to render</tag>
```

מבנה כללי יותר:

```
<tag attribute1="value1" attribute2="value2">content to render</tag>
```

אני מניח שכולכם מכירים את השימוש ב-HTML ושאין צורך להסביר מהם רכיבי HTML מעבר לכך. מה שאולי אינכם יודעים היא העובדה שהתגים עצמם מחולקים ל-3 קבוצות:

1. Block Elements - תופסים את כל הרוחב האפשרי בדף. לפניהם ואחריהם מתחילה שורה חדשה. (לדוגמא: <p>, <h1>, <ul>,)

2. Inline Elements - לוקחים בדיוק את המקום שהם צריכים, ולא מתחילים שורה חדשה. (לדוגמא: <a>, ,)
3. בלתי נראים - לא מופיעים בדף עצמו. (לדוגמא: <meta>)

מבחינת משמעות התגים השונים, מוגדר כי:

1. Block Elements יכולים להכיל Block Elements אחרים, או Inline Elements.
2. Inline Elements יכולים להכיל בתוכם Inline Elements אחרים וטקסט בלבד.

תגי ה-Block הם הנותנים את "התמונה הגדולה" של הדף. הם מחלקים את המסמך לחלקים שונים. תגי ה-Inline הם המרכיבים של החלקים השונים.

בתיאוריה זה נשמע ברור והגיוני. איפה הרבה אנשים טועים? יתכן שבעבר כתבתם קוד כזה:

```
<b><h1> Welcome to my site </h1></b>
```

יש כאן כותרת, ואנחנו רוצים גם שהיא תהיה מודגשת. זו טעות! כותרת היא תג Block ואילו <b> הוא תג inline. למרות שרוב הדפדפנים היום יציגו את הטקסט בכותרת מודגש, כמו שביקשנו, מבחינת המבנה הנכון של מסמך HTML זו פשוט טעות. הצורה היותר הנכונה היא:

```
<h1><b> Welcome to my site </b></h1>
```

אני אדגיש "היותר נכונה" מכיוון שהשימוש בתג <b> בעייתי באופן כללי - זהו תג של עיצוב ולא של משמעות. התג הנכון יותר לשימוש, במידה ואנחנו רוצים להבליט את הכותרת, הוא .

התמונה הגדולה

דיברנו על סמנטיקה, הצגנו מעט טעימות מה זה לבנות אתר "בצורה נכונה סמנטית", עכשיו בואו נדבר על התמונה הגדולה. כשאנחנו מדברים על בניה נכונה מבחינה סמנטית, על מה אנחנו מדברים?

מבנה מסמך הגיוני, זרימת מסמך הגיונית:

- מבנה קוד נכון - חלוקה גדולה לתגיות בלוקים. בתוכם חלוקה לבלוקים משניים.
- כחלק ממבנה הקוד - תגיות Hx במבנה היררכי נכון. תגיות הכותרת מסבירות את המבנה הלוגי של העמוד ואפשר להבין את נושאי העמוד מקריאת הכותרות בלבד.
- כשמסתכלים על קוד המסמך, ללא עיצוב - הסדר של הדברים במסמך צריך להיות הגיוני. למשל: ה-footer לא צריך להיות החלק הראשון מבחינת הקוד, המופיע בדף שלנו.

תיוג מושכל:

- שימוש בתגיות שיסבירו את התוכן. הדוגמא הכי בסיסית: תגית הכותרת, שהדגמנו בפרק הפתיחה. דוגמא נוספת לתגית בעלת משמעות היא תגית הפיסקה <p>. שימוש בתגית <p> במקום ב-
 כדי להפריד בין פסקאות מבהירה לקורא שמדובר בפיסקה חדשה בדף שלנו, לעומת
 שאין לו משמעות סמנטית. יש להשתמש בתגיות המסבירות את התוכן בכל מקום שרק ניתן להשתמש בו בהן.

- לא משתמשים בתגיות במשמעות שונה מהמשמעות הסמנטית שלהם! טעות זו חוזרת במקרים רבים כשמתכנתים משתמשים בתגית עם משמעות סמנטית כדי לקבל אפקט עיצובי כלשהו.

הפרדת עיצוב מתוכן:

- במצב הטוב ביותר - אין שום סימן של עיצוב כשמסירים את ה-CSS - הדף נראה כמו דף של שנות ה-90 ללא ה-CSS.

לעתים קיים בלבול בין קוד תקני לקוד נכון סמנטית. שימו לב כי קוד תקני הוא קוד ללא שגיאות syntax, אבל הוא לא מציין בהכרח שמוקמו בו התגיות הסמנטיות הנכונות. אתר המעוצב באמצעות טבלאות יכול להכיל קוד תקני, אבל הוא אינו נכון סמנטית.

מבנה מסמך נכון סמנטית

שלבים ליצירת מבנה דף נכון סמנטית:

1. המסמך שלנו צריך להתחיל בתגית DOCTYPE המסבירה לדפדפנים איך רצוי להתייחס למסמך שלנו. ביחרו את התג המתאים.
2. מבנה דף ה-html הבסיסי:

```
<html>
<head>
      <title> </title>
</head>
<body>
      Content here
</body>
</html>
```

חשוב לשמור על מבנה זה - כמה שהוא בסיסי, אפילו הוא מפוספס באתרים רבים.

3. ה-body צריך להיות מחולק ל-div שונים, המציינים את חלקיו השונים של הדף שלכם. ה-div השונים צריכים להיות בסדר הגיוני. ה-div בו נמצא ה-footer של הדף, למשל, צריך להופיע בסוף הקוד.
4. מבנה הכותרות וסדר הכותרות צריך להיות הגיוני. למשל, אסור שתהיה קפיצה מ-h1 ל-h4. דוגמא למבנה:

```
h1 -Page heading
 h2 -Section heading
  h3 -Section subheading level 1
  h3 -Section subheading level 1
 h2 -Section heading
  h3 -Section subheading level 1
  h3 -Section subheading level 1
   h4 -Section subheading level 2
    h5 -Section subheading level 3
  h3 -Section subheading level 1
   h4 -Subsection subheading level 2
 h2 -Section heading
```

משמעות סמנטית של תגיות HTML

תגיות בלתי נראות

<title>

- תגית ה-title לא מופיעה בדף עצמו, אך היא אחת התגיות החשובות ביותר הקיימות.
- תקן W3 מציין לגבי ה-title שעליה לציין את תוכן הדף. בנוסף הוא מציין שיש לתת תיאור משמעותי ולא תיאור קצר מכיוון שהרבה פעמים הגולש מגיע לדף ישירות ולא יודע את ההקשר. הכותרת "Introduction", למשל, היא כותרת גרועה לדף. כותרת מוצלחת יותר תהיה " Introduction to HTML" - כעת ברור מה יש בדף שלנו.
- שימוש ב-title טוב מועיל הרבה כאשר מדובר במנועי חיפוש: רוב מנועי החיפוש ישתמשו ב-title כדי להציג את הדף שלנו בתוצאות החיפוש. בנוסף מנועי החיפוש משתמשים בתג זה רבות כדי להחליט עבור אילו מילים הדף שלנו יופיע גבוה בתוצאות החיפוש.

תגיות Block

** **

- divider - חלוקת הדף לחלקים. התוכן ב-div מייצג חלק בדיד משאר חלקי הדף. לרוב ישמש כדי לציין חלוקת הדף ל-header, footer, איזור התפריט ואיזור התוכן. נעשה בו גם שימוש לחלק איזור לאיזור משנה - למשל - חלוקת כתבה לאיזור הטקסט המרכזי ולאיזור התגובות שלה.
- אם נכתוב את הדף עם המון div, כאילו עיצבנו את הדף בעזרת טבלאות. חשוב להשתמש בתג זה לפי המשמעות הסמנטיות שלו.

h1-h6

- כל הכותרות באות לייצג חלוקה של התוכן לחלקים שונים (המופרדים על ידי <p> ו-). הכותרות מגדירות למעשה את מבנה הדף.
- h1 היא הכותרת הראשית של הדף. היא מופיעה רק פעם אחת. כותרות אחרות יכולות להופיע מספר פעמים, בהתאם למבנה ההיררכי הנכון שהוצג בפרק הקודם.

<p>

- אלמנט הפסקה הוא האלמנט הבסיסי המחלק טקסט. מבחינה סמנטית, היא מייצגת קטע בדיד של טקסט.
- בניגוד ל- המגדיר חלוקה כללית בדף, <p> משמשת לחלוקת טקסט לפסקאות.

<blockquote>

- תג הבא לציין שקטע הטקסט שבתוכו מצוטט ממקור אחר.
- תג זה תמיד צריך לבוא עם תג cite המציין את המקור ממנו הציטוט מגיע, ולעתים גם מכיל מאפיין cite המציין URI של הטקסט המצוטט.

תגיות Inline

<a>

- תג זה מייצג קישור למשאב חיצוני, או לחילופין קישור לקטע פנימי בדף הנוכחי.
- חשוב לשים לב שבעזרת קוד סקריפט אפשר להקנות פונקציונליות מגוונת לתג <a>. מבחינה סמנטית חשוב לנו לשמור את המשמעות שלו כקישור.

****

- תג זה מייצג דגש קל. מטרתו הסמנטית היא להדגיש שהטקסט שהוא מקיף חשוב מעט מהטקסט הסובב אותו.
- הבחירה האם טקסט כלשהו דורש דגש קל היא בידי כותב הטקסט.
- כברירת מחדל, דגש קל מופיע בטקסט כטקסט מוטה, אך בעזרת CSS אפשר לשנות זאת לכל צורה אחרת.

****

- תג זה מייצג דגש חזק. מטרתו הסמנטית היא להדגיש שהטקסט שהוא מקיף חשוב מהטקסט הסובב אותו. זה בעצם גרסה חזקה יותר של
- הבחירה האם טקסט כלשהו דורש דגש היא בידי כותב הטקסט.

- כברירת מחדל, דגש חזק מופיע בטקסט כטקסט מודגש, אך בעזרת CSS אפשר לשנות זאת לכל צורה אחרת.

<address>

- תג המציין קטע בו מופיעים פרטי יצירת הקשר של כותב הדף.
- כברירת מחדל הטקסט בו מופיע מוטה.
- מקובל שתג זה מופיע ב-header או ב-footer של העמוד.
- תג זה מכיל לרוב את הכתובת הפיזית של העסק/כותב הדפים.

<code>

- מציין קטע קוד (של שפת תכנות) המוצג בתוך המסמך.

<samp>

- מציין פלט של תוכנית מחשב.
- נניח התוכנית שלכם מדפיסה "hello world", אז קוד התוכנית יוקף בתג <code> והפלט שלה, "hello world", אם תציגו אותו בדף, יוקף בתג <samp>.

<cite>

- מקור ציטוט - משמש כדי לציין מהו המקור של המידע.

<q>

- טקסט מצוטט קצר. לדוגמא:

```
<p>As <cite>Bill Gates</cite> said, <q>640K should be enough for
anybody.</q></p>
```

- הטקסט המצוטט יופיע באופן אוטומטי בין מרכאות עוטפות.
- חשוב לציין ש-Internet Explorer מפגין יישום גרוע של תג זה, ולפיכך אתרים רבים לא משתמשים בו.

****

- ל-span יש תפקיד סמנטי - מפריד פנימי כללי של בלוק. תג זה הוא מפריד כללי - אין לו משמעות ספציפית (כמו תגים אחרים שראינו בפרק זה).
- נשתמש בו כאשר אין תג ספציפי בעל משמעות מתאימה יותר.

תגי רשימות

**<ul>, <ol>, **

- תגים אלו מייצגים *רשימות* של פריטים. מייצג רשימה חסרת סדר ו- רשימה בעלת סדר. הוא פריט ברשימה.
- תפריטים, לדוגמא, הם דוגמא קלאסית של נתונים המתאימים להצגה כרשימה - תפריטים הם הרי רשימה של קישורים. הערות בבלוג מתאימות להיות רשימה בעלת סדר (לרוב הן ממוספרות בסדר לפי זמן שליחתן).

<dl>, <dd>, <dt>

- תגים אלו מייצגים "רשימת הגדרות" - ובאופן פרקטי - זוגות של מונחים (<dd>) ושל ההגדרות שלהם (<dt>).
- תג זה יחסית פחות מוכר מאשר הרשימה הרגילה וחשוב להכירו.
- שימוש בתג זה נכון מבחינה סמנטית בכל פעם שיש לך רשימה המורכבת מזוגות בעלי אופי של שם+ערך. ברוב שאר המקרים, <ul> הוא הפתרון הנכון.

תגיות בהן לא משתמשים

התגיות הבאות הוכרזו כמיושנות (Deprecated) ואין להשתמש בהן כלל: applet, center, font, dir, isindex, menu, s, strike, u. בגרסאות עתידיות של html כלל לא מובטחת שתהיה תמיכה בתגיות אלו.

תגיות בהן כמעט אף פעם לא נשתמש: big, small, b, i, tt, pre. תגיות אלו הינן חלק מהשפה, אבל אין להן משמעות סמנטית. לפיכך, נשתמש בהם רק במקרים נדירים בהם אחרי מחשבה נחליט שהשימוש הטוב ביותר הוא בהן. למשל <b> הוא הדגשה, אך ללא מתן חשיבות לטקסט המודגש.

תגיות נוספות

חשוב לדעת שיש עוד תגיות רבות ב-HTML עם משמעות סמנטית. אמליץ לכל הקוראים להעמיק ברשת בנושא זה כדי לקבל בו שליטה מלאה.

הפרדת העיצוב מהתוכן

הכלי המרכזי שיש לנו כדי לבנות מסמך סמנטי ולהפריד את העיצוב מהתוכן הוא CSS Positioning. הוא אינו קשה מאוד ללמידה ומומלץ להבנה לעומק.

העיצוב של האתר יבוצע תמיד על ידי קובץ CSS חיצוני. כאשר המסמך מחולק נכון - לא נדרשים אלמנטי עיצוב בתוך המסמך.

סיכום

המשמעות של התוכן מובנת על ידי הקורא של התוכן. חשוב לשים לב שהקורא הוא לא תמיד אנושי. רובוטים ותוכנות שונות יקראו את התוכן שלכם פעמים רבות.

כתיבה נכונה סמנטית נכונה גם במקומות נוספים, שאינם רק HTML. גם ב-Word, למשל, ניתן לקבוע כי שורה מסויימת היא מודגשת וגדולה, או לחילופין ניתן לסמן אותה ככותרת במסמך. אם תסמנו את השורות ככותרות, Word יוכל, למשל, לייצר לכם תוכן עניינים למסמך באופן אוטומטי. עניין הכתיבה הסמנטית צריך להיות בראש בכל פעם שכותבים מסמך, ובכל פלטפורמה.

ברצוני להמליץ גם על הבלוג המצויין של מר יובל רז, www.yuvalraz.com, חבר ואחד האנשים המבריקים בתחום ה-UX בארץ. במיוחד אפנה אתכם להרצאת וידאו המדברת על הנושאים במסמך זה, ומציגה דגשים שבחלקם שונים מאלו שהצגתי לכם היום. הרצאת הוידאו במפגש איגוד האינטרנט.

חשוב לי להדגיש כי לא לכל דבר יש ל-HTML תג יעודי, אך עדיין קיימים ב-HTML הרבה תגים ייעודיים בהם אתרים לרוב לא מבצעים שימוש. כשאתם חושבים על קוד - נסו לתת לו כל העת משמעות רבה ככל שניתן. כשאתם כותבים קוד HTML חישבו על מה התוכן ומה המשמעות שלו, ולא על איך אתם רוצים שהוא יעוצב, כאשר אתם בוחרים את התגים.

המאגר הביומטרי

מאת יהונתן קלינגר

חברי הכנסת עוסקים מדי בכניסה והתערבות בחייהם של בני אדם: מחוקקים חוקים בנושא בטיחות בדרכים מבלי שהם מומחים לתעבורה, מחוקקים חוקים בנושאי בריאות מבלי שיש להם את ההבנה בבריאות הציבור ואפילו מתערבים בנושאי אבטחת מידע מבלי שקיבלו הכשרה מתאימה. **האם רק משום שהם מייצגים את רצון העם עליהם להחליט?**

על פי החוק, ידרשו בקרוב כל אזרחי ישראל לתת למדינה לאחסן בצורה דיגיטלית עותק מצילום הפנים שלהם וסריקה של שתי אצבעותיהם המורות. מידע זה ישמר במאגר ביומטרי, אליו תהיה גישה למטרות שיפורטו ועל ידי אנשים שיפורטו. בעוד שלשיטתי המאגר עצמו מהווה פגיעה בפרטיות, הרי שגם אם המאגר לא מהווה פגיעה בפרטיות, האפשרויות לניצול לרעה ופגיעה באזרח הנובעות ממהלך זה הן מהותיות.

חוק שרשרת של הסכמות בין גופים שלטוניים בנוגע לארכיטקטורה שתפעיל את המאגר הביומטרי מתאר את התהליכים שעובר האדם ומנוסח בצורה רשלנית מעט. אולם הבעיה העיקרית היא בכמות האנשים להם תהיה הגישה למאגר. למרות שלל ההצהרות של מחוקקים כי המאגר יהיה מאובטח ברמה 11, סביר להניח כי לא היינו מסכימים לכך אם היינו מודעים לעובדה שלכמה עשרות אלפי אנשים תהיה גישה למאגר המדובר. כמו כן, סביר להניח כי אף אחד לא היה מאמין שמאגר המאפשר גישה לעשרות אלפי אנשים לא יפרץ לעולם.

מי הם האנשים שיכולים לגשת למאגר הביומטרי? ראשית, עובדי משרד הפנים. החוק מזכיר נטילה של אמצעי זיהוי; הוא מסמיך את עובדי משרד הפנים **לטול** מידע ביומטרי מאזרחי המדינה, וקובע כי האמצעים יועברו לרשות להכללתם במאגר ולמרכז ההנפקה כדי להנפיק לאדם תעודת זהות ביומטרית. מעבר לעובדי משרד הפנים, גם למרכז ההנפקה, גוף שמוסדר על ידי סעיף 4 ואינו מוגדר בחוק כרשות ממשלתית, ישנה גישה למאגר. מעבר לכך, ואולי הצעד האירוני ביותר הוא שנהלי אבטחת המידע שמופיעים בחוק לא מסדירים יתר על המידה את היכולת לפקח עליו, כי מרכז ההנפקה הופיע לראשונה רק בדיון מיום 12.07.2009. באותו הדיון התברר כי משרד החוץ, שעד כה הנפיק מסמכי נסיעה רשמיים (דרכון דיפלומטי) לא יוכל לעשות זאת, אלא שהכל יהיה חייב לעבור דרך אותו מרכז הנפקה. באותו הדיון עלתה שאלת ה-"גישה" למאגר בפעם הראשונה, כאשר אנשי משרד הפנים סתרו את עצמם והסבירו כי אין גישה למאגר, אך יש גישה למאגר.

הנה קטע מן הדיון:

היו"ר **מאיר שטרית**: אנחנו מדברים על הרכשה ראשונה. בהרכשה ראשונה כשאדם בא, נותן טביעת אצבעות שלו, מצלמים את הפנים שלו.

יהונתן קלינגר: והרכשה ראשונה תתבצע בקונסוליות, ולכן כן תצטרך להיות איזשהי גישה למאגר.

היו"ר **מאיר שטרית**: **אין לאף אחד גישה למאגר**.

ניסים אליאסף: גם למשרד הפנים אין גישה למאגר. אם הכוונה של גישה למאגר זה באון ליין, גם למשרד הפנים אין גישה באון ליין.

יהונתן קלינגר: למשרד הפנים יש, לפי סעיף 13.

ניסים אליאסף: **אין גישה באון ליין**.

נירה לאמעי: אז איך מעבירים את ההרכשות מהקונסוליות?

ניסים אליאסף: אנחנו אמרנו שנבוא ונציג את זה אחרי שיש לנו---

(...)

יהונתן קלינגר: אז אני שואל, איך אתה מונע הרכשה כפולה בקונסוליות בחוץ לארץ?

ניסים אליאסף: מביאים את הביומטריה לארץ בדרך של דיפ, כמו שהוא אמר ,ומעבירים את זה למאגר ובודקים ש---

יהונתן קלינגר: כלומר הכל יבוצע בדואר, דואר דיפלומטי אבל דואר.

עופר ישי: **לא בהכרח בדואר. יש היום מערכת תקשורת מסוימת בין משרד החוץ לבין משרד הפנים, ופה אנחנו נכנסים לשיטת המימוש** ,אז כמו שנאמר פה, זה יכול להיות בצורה של מדיה מגנטית ואופ ליין, זה יכול להיות באיזה שהיא העברה מוצפנת ומוסדרת כפי שקיים היום לנושאים אחרים. בסופו של דבר זה יגיע וזה יגיע למשרד ה... היום התקשורת היא בין משרד הפנים למשרד החוץ ומשם זה יעבור למאגר, ואז תיעשה הבדיקה, ואם יתגלה שיש כפילות, אז לא ינפיקו לו את הדרכון הנוסף. זו הכוונה פה.

יהונתן קלינגר: **אתה אומר שתהיה תקשורת מול משרד הפנים ומשם מול המאגר?**

עופר ישי: **כן** .אני מסייג את דבריי, כמו שנאמר פה ממשרד הפנים, אז הם רוצים לבדוק את זה, אז יש פה מספר דרכים של מימוש, זה לא משנה את התהליך העקרוני, זה יגיע בין אם זה דרך משרד הפנים או ישירות, זה לא משנה, זה יגיע ותתבצע הבדיקה, זה מה שנאמר פה. ולא באון ליין.

מעבר לכך, ההגדרה של מערך ההנפקה הוספה רק בדיון של 19.07.2009, לאחר שהתברר כי הצעת החוק המקורית כלל לא כללה יכולת להקים מערך כזה. השאלה של כיצד מערך ההנפקה "ניגש" למאגר

ולמידע הביומטרי על מנת להנפיק את תעודת הזהות היא גם שאלה שטרם נפתרה, אך טלאי החוק אמורים לכסות עליה. בדיון מיום 09.07.2009 עלתה הסוגיה לראשונה, כאשר בקריאה ראשונה של החוק, הבינו אנשי משרד הפנים כי החוק כלל לא מייצג את הארכיטקטורה שהם רצו ליישם. המידע שנלקח מהאזרח נאגר בשני מקומות: על התעודה ובמאגר הביומטרי; הן בתעודה והן במאגר נשמרים הן האמצעים הביומטריים (תמונות מקור) והן הנתונים הביומטריים (חתימות או HASH). המידע במאגר ובתעודה אמור להיות זהה, לפחות לכאורה. מערכת הגישה למאגר מסובכת, וכוללת מספר גישות למאגר, בצו בית משפט וללא צו בית משפט. ראשית, ללא כל צו וללא הגבלה על גישה הינה הגישה לפי סעיף 21. סעיף 21 מסביר כי הרשות "תאפשר להן [לרשויות הבטחון -י.ק] גישה למאגר הביומטרי". השאלה מהי גישה היא לא שאלה פשוטה; וככל הנראה חשאיותה של הגישה (שגם לא מתקיימת למשרד החוץ, כאמור)

בדיון שנערך בועדה ביום 20.07.2009 אלה היו ההסברים:

היו"ר **מאיר שטרית**: בסדר, אור השמש לא מתאים לדברים סודיים.

(...)**יהונתן קלינגר**: מה זה גם 'תעביר מידע מתוך המאגר'? זה יכול להיות מכל המאגר.

איתן כבל: מר גבע עדיין באמצע הדברים שלו ואחרי זה נתייחס, כי אני עדיין לא נחה דעתי.

דני גבע: הסעיף הזה בעצם מנוסח אחרי שנבדקו כל האפשרויות האחרות ועל מנת לאפשר לנו לסבול לפי הצרכים שלנו. מה שאני רוצה לומר זה שמה שאנחנו יוצרים פה, עם הנפקת התעודות הביומטריות והקמת המאגר, זה משהו חדש שלא היה קיים קודם. המצב החדש שנוצר, במצב הזה אנחנו חייבים להמשיך ולפעול למילוי תפקידנו וייעודנו.

(...)

נירה לאמעי: כשהם אומרים 'תאפשר להם גישה למאגר', הכוונה היא שיוכלו פשוט להיכנס למקום שבו נמצא... יהיו להם הרשאות גישה למאגר? כשאומרים 'תאפשר להם גישה', הרי זה לא רק להעביר להם---

היו"ר **מאיר שטרית**: לא בתקשורת.

נירה לאמעי: אז מה זה תאפשר להם גישה למאגר?

ניסים אליאסף: יכולים לבוא למאגר, לקבל מידע.

(...)

ניסים אליאסף: למאגר לא יהיה תקשורת.

היו"ר **מאיר שטרית**: אז אולי תשנו את המילה 'גישה'.

נירה לאמעי: אז מה זה גישה?

דני גבע: **לא משנה מה זה גישה**, המילה 'גישה' חייבת להישאר, כי בחנו את כל האפשרויות---

היו"ר **מאיר שטרית**: תסביר.

דני גבע: אדוני, יש דברים שאני לא יכול לפרט.

עומדת כאן גישה ששירות הבטחון אינו מסוגל לפרט ולפיה הוא דורש גישה למאגר הביומטרי. מעבר לסעיף 21, שמאפשר גישה, החוק מאפשר לשוטרים לטול אמצעי זיהוי מאדם הנמצא לפניהם ולהשוותם מול מידע במאגר. גם כאן, הדרך בה ההשוואה תבוצע אינה ודאית; גם כאן, הורגע הציבור כאילו אין גישה למאגר על ידי המשטרה. בדיון מיום 07.07.2009 נאמר:

היו"ר **מאיר שטרית**: איך זה עובד? מה עושה השוטר שצריך לקבל זהות?

נסים אליאסף: **הוא מעביר את טביעת האצבע אלינו, למאגר. המאגר לא חשוף לאינטרנט, כלומר, לא חשוף לתקשורת בכלל אלא הוא בפני עצמו**. לוקחים את טביעת האצבע ובודקים אותה מול המאגר. במידה שיש זיהוי כזה, יש איזשהו קוד שאנחנו יודעים אותו ודרך הקוד הזה פונים למערכת אחרת. המערכת האחרת בטלפון תיתן את השם שלו.

היו"ר **מאיר שטרית**: זאת אומרת, זה מה שאני אומר מהתחלה ואתם מנסים לתקן אותי שלא לצורך. כאשר המשטרה צריכה לפנות לזיהוי אדם, זה השימוש היחידי שהמשטרה עושה בקשר לאדם בלתי מזוהה, היא פונה לרשות, לגוף אחר ברשות, הרשות בודקת אם הטביעה הזאת בכלל מופיעה במאגר, אם יש אדם כזה במאגר. אם זה אדם שבא מחוץ לארץ, הסתנן לארץ ולא יודעים מי הוא.

(...)

נירה לאמעי-רכלבסקי: מה זה מעביר את הבקשה לטביעה? כל טביעה? את הקוד הדיגיטלי של הטביעה?

נסים אליאסף: הוא מעביר תמונה של טביעת האצבע.

נירה לאמעי-רכלבסקי: הוא מעביר את זה בתקשורת או ברשת?

יורם אורן: **הוא מעביר את זה בתקשורת**.

מעבר לגישה זו הקיימת למשטרה, שהיא לא און-ליין אך באורח פלא מאפשרת קבלת טביעות אצבע באמצעות הטלפון, מוסמך בית המשפט לאשר העברה של "נתונים או אמצעים ביומטריים הכלולים במאגר הביומטרי", לצורך חקירת עבירות, מניעתן או לצורך העברת המידע לרשויות אכיפה מחוץ לישראל (סעיף 17). כלומר, מקרים בהם רשויות חקירה כמו ה-FBI ידרשו מישראל את מלוא המאגר (שנכנס תחת ההגדרה של "נתונים או אמצעים ביומטריים הכלולים במאגר הביומטרי"), ישראל תתן או תהיה חשופה לסנקציות דיפלומטיות.

כפי שניתן להבין, המאגר הביומטרי מיישם בצורה ייחודית את המילה "גישה"; במיוחד בשים דגש על מדיניות אבטחת מידע סבירה. כמות האנשים להם תהיה *גישה* כלשהי למאגר (בין אם כתיבה בלבד או קריאה בלבד) היא עצומה את מלוא פקידי משרד הפנים, כל שוטר במשטרת ישראל, כל עובדי שירות הבטחון וכל שוטר במשטרה הצבאית. מעבר לכך, יכולים להווצר מאגרים נוספים אשר לא יהיו כפופים לאותן סנקציות הקיימות בחוק על מעבירי המידע מהמאגר.

השאלה **"מהי גישה"** היא שאלה משפטית שצריכה התייחסות נקייה. המילה *Access* מוגדרת על ידי מיליון מירים-וובסטר כחירות או יכולת להשיג או לעשות שימוש במשהו. גישה, תחת ההגדרה הזו, אינה מסוג הדברים שראוי לאפשר לעובדי מדינה, ובמיוחד לא לעשרות אלפים מהם. ברור שאותה אווירה, בה גורמים אזרחיים וממשלתיים יהיו בעלי גישה, נקודות הכשל והיכולות לאבטח את המאגר יהיו אפסיות. בעוד ש**מאיר שטרית** טוען כי המאגר עשוי להפרץ, אך אין לאדם אינטרס לעשות זאת, כל בר-דעת מבין שמערכת אליה יש גישה לעשרות אלפי אנשים אינה יכולה להיות מוגנת.

עכשיו, מה נותר לעשות? בשנתיים הקרובות החוק לא יבוא למימוש סופי אלא יוחל על בסיס בחירה בלבד. במידה והניסוי הביומטרי לא יצליח, כי לא מספיק ירשמו, כי המידע ידלוף, כי כל דבר אחר יביא אותם לכדי הגיון, אז נוכל עוד לשמור על מידע בטוח, על פרטיות ועל העתיד שלנו.

לפרטים נוספים על המאגר הביומטרי, על מטרותיו והשלכותיו על פרטיות האזרח במדינת ישראל, מומלץ לעיין באתרים של המרכז לעצירת חוק המאגר הביומטרי וכן באתר של האגודה לזכויות האזרח.

Zip Bombs

מאת Crossbow ו-cp77fk4r

הקדמה ורקע כללי

במאמר זה נסביר את מושג הנקרא "פצצות Zip" (Zip Bombs או Zip of Death). כמו כן נסביר איך ליצור אותה ואילו שימושים מעניינים ניתן לעשות איתה.

פצצות Zip אלו קבצי Zip אשר נוצרו בעבר על מנת לגרום לקריסת מערכת ההפעלה. כיום, מערכות ההפעלה יודעות להתמודד איתן ולכן הן פחות מסוכנות למערכת ההפעלה- אך לאפליקציות אנטי-וירוס אשר לא נכתבו כך שיכולו "לטפל בהן" - הן מסוכנות מאוד.

התיעוד הראשון למתקפה שכזאת הוא מאמצע שנת 2001 ע"י בחור בשם Michel Arboi, ועוד אז היא סוייגה כ-"Failure to Handle Exceptional Conditions" (שכמובן הובילה ל- Local denial of service attack). הרעיון המרכזי בהתקפה הוא לנצל עקרון בסיסי מאוד במנגנוני הדחיסה של אלגוריתמי הכיווץ וליצור קובץ Zip בעל יחס דחיסה (compression ratio) גבוה מאוד. כך-בזמן פתיחת הקובץ בעזרת מנגנוני פתיחת הכיווץ של תוכנות האנטי-וירוס בעת סריקתו-לגרום לו לקריסה. חשוב לשים לב כי מדובר ביחס דחיסה אדיר. לדוגמא, ה-Zip Bomb המפורסמת ביותר היא "פצצה" בשם "42.zip", מדובר בקובץ Zip בגודל 42kilobytes (ומכאן שמה) שלאחר פתיחת דחיסתו הוא נהפך לקובץ המכיל 4.5peta-bytes (כל peta-bytes אחד מכיל 1024 tera-bytes, וכל terabytes אחד מכיל 1024giga-bytes) משמע יחס דחיסה של יותר **ממאה ביליון** bytes! הזוי משהו.

קצת תיאוריה

בכדי להבין איך זה אפשרי ליצור פצצה שכזאת עלינו להכיר מקרוב את עולם דחיסת המידע, איך בכלל אפשר לכווץ קובץ? איך אנחנו יכולים לאחסן את אותו המידע כך שיתפוס פחות נפח בכונן שלנו? השאלות האלה קצת מפוצצות, אך התשובות להן בעצם לא כל כך קשות להבנה. כל הרעיון בדחיסת מידע הוא שינוי האופן בו המידע נשמר. לדוגמא, קיימים היום הרבה מאוד סוגים של פורמטים לקבצי

גרפיקה, מהמוכרים בינהם אפשר למנות את ה-BMP, JPG, PNG ועוד. שימו לב שאם תציירו תמונה בצייר ותשמרו אותה ב-BMP היא תתפוס נפח מסויים (יחסית גדול), ואם תשמרו את אותה התמונה בפורמט שונה, למשל-JPG, תראו שהיא תתפוס נפח שונה לחלוטין (וקטן בהרבה) מאותה התמונה שנשמרה בפורמט BMP.

איך זה קורה?

הסיבה הראשונה-התמונה היא לא אותה התמונה. כאשר שומרים תמונה ב-JPG האיכות שלה יורדת. מה זאת אומרת "האיכות יורדת"? זאת אומרת שאם תשימו לב טוב טוב מספר הגוונים בתמונה ירד, התמונה הרבה פחות חדה ואפשר לזהות שהיא קצת "מרוחה".

הסיבה השניה (והחשובה לנו)-אופן שמירת מידע התמונה בזכרון המחשב שונָה.

מה זאת אומרת?

במדעי המחשב, ישנו ענף שלם העוסק בתחום זה ונקרא "דחיסת נתונים". לא נסביר את כולו כי זה לא נושא המאמר, ולכן רק נצטט מויקיפדיה תחת הערך "דחיסת נתונים":

> "דחיסת נתונים אפשרית משום שבנתונים בצורתם הגולמית קיימת פעמים רבות יתירות גבוהה, כלומר מידע החוזר על עצמו או מידע שניתן לייצג בצורה חסכונית יותר. העיקרון הבסיסי בדחיסה הוא כי מידע שאנו עושים בו שימוש יש בו "סדר" מסוים. מצאת הסדר הזה מאפשרת לייצג את המידע בדרך יעילה יותר. מידע שהוא מקרי (רעש) לא יכול להידחס. מידת הסדר מכונה 'אנטרופיה.'"

(צוטט מויקיפדיה תחת הערך: דחיסת נתונים)

ישנם מספר רב מאוד של אלגוריתמי דחיסת נתונים שונים, המוכרים שבהם הם קוד הופמן, קידוד אורך חזרה, ועוד מספר רב של אלגוריתמים שאנו משתמשים בהם בדרך קבע במהלך היום-יום בעת הפעלה של קבצי תמונה, קבצי שמע או וידאו. אנחנו מדברים כמובן על קבצים כגון mp3, mpeg, jpg, png, avi ורבים אחרים.

העקרון אותו מנצלים בעת יצירת Zip Bombs הוא אפקט יחס הכיווץ הגדול הנוצר בעת דחיסת מידע בעל מידת אנטרופיה גבוהה ביותר. וכמו שראינו, ככל שהאנטרופיה בנתוני הקובץ גבוהה, כך יחס הכיווץ יגדל. איפה נקבל את מידת האנטרופיה הגבוהה ביותר? ברב המקרים-יהיה זה כאשר כל נתוני הקובץ זהים לחלוטין.

כך למשל, את המידע הבא:

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

נוכל לציין על פי RLE באופן כל כך פשוט בדרך הבאה: A187

קידוד Huffman

אם נקח את WinZip כדוגמא, נראה שבכדי לדחוס את הנתונים שבה היא משתמשת במעין שילוב של אלגוריתם הדחיסה LZ77 וקידוד Huffman.

שיטת קידוד Huffman נחשבת לאופטימלית במובנים רבים, האופטימליות מתבטאת בעיקר בכך שהיא דוחסת נתונים באופן מירבי, כלומר, משתמשת בהכי מעט ביטים שאפשר כדי לייצג כל תו. הרעיון הוא שלתווים שיחזרו על עצמם הרבה יוקצו מעט ביטים, בעוד שתווים 'נדירים' ישתמשו בהרבה ביטים.

כמה מונחים לא פורמליים בתורת הגרפים:

- גרף הוא צומת של צמתים וקשתות. ניתן לדמיין את הצמתים כ-"איים", בעוד הקשתות הם חיבורים בין הצמתים.
- גרף יכול להיות מכוון או לא מכוון. גרף מכוון הוא גרף שבו יש כיוון לקשתות.
- עץ הוא גרף ללא מעגלים, בו קיים צומת (לו קוראים 'שורש העץ') שממנו קיים מסלול אל כל צומת אחר.
- בעץ, לכל צומת יש צומת אב (מלבד לשורש). צומת שלו אין בנים נקרא עלה.

- עץ בינארי הוא עץ בו לכל צומת שאינו עלה יש לכל היותר שני בנים.

פעולת האלגוריתם:

האלגוריתם בונה עץ בינארי הפוך, כך שבעלים (צמתי הקצה) יש תווים. ניתן לתאר אותו כך:

- בנה טבלת תדירויות עבור כל תו, כלומר, בנה עלים כך שבכל עלה כתוב התו וכמה פעמים הוא חוזר על עצמו.
- מצא את שני הצמתים היתומים (צמתים ללא אב) המינימליים במספרם.
- צור צומת חדש שיהווה את אב שני הצמתים הללו, וערכו יהיה חיבור ערכם של בניו.
- אם הצומת החדש הוא הצומת היתום היחיד, סיים.

כעת, נקבע שכל בן ימני הוא ביט 0, וכל בן שמאלי הוא ביט 1. הקידוד של תו מסויים הוא שרשור הביטים על הקשתות המובילות מהשורש אל התו עצמו (שהוא עלה). לכן, ניתן לקודד כעת את הטקסט המקורי, ובמקום כל תו לשים את הקידוד שלו. הדבר היחיד שצריך להוסיף הוא טבלת התדירויות המקורית, או לחילופין את הקידוד של כל תו.

ניתן למצוא מידע נוסף ומורחב על האלגוריתם במאמר של גיל כהן בפרוייקט UnderWarrior בקישור הבא: http://www.underwar.co.il/document-details.asp?id=134

פעולות ושימושים

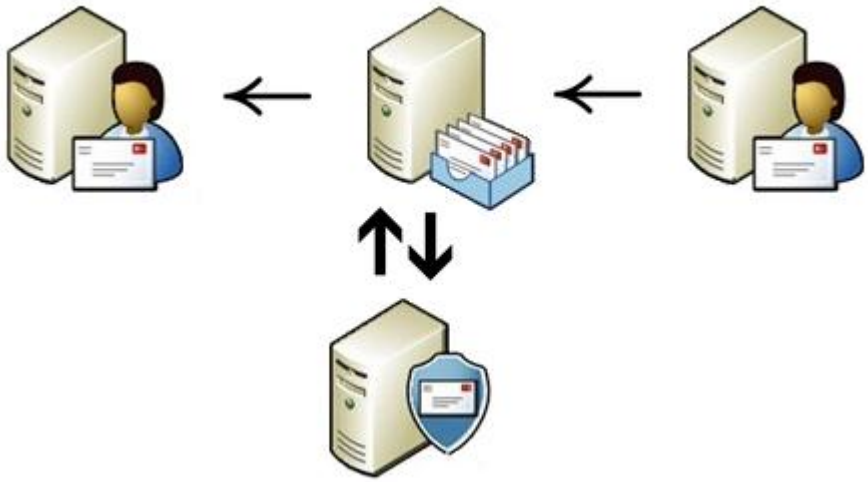
לאחר שהבנו איך העניין עובד, נשאלת השאלה-למה אנחנו צריכים את זה? חוץ מלהקריס לנו את תוכנת WinZip-על ימין ועל שמאל (מה שכיום גם כבר לא יקרה), איזה שימוש יעיל אפשר למצוא בחיה שכזאת? אחד השימושים היעילים שאפשר למצוא לפצצת Zip הוא ניטרול תוכנת ה-Anti Virus על מחשב מרוחק, לדוגמא-שרת ה-Exchange של אירגון מסויים. באירגונים גדולים המאפשרים לעובדיהם שליחת מיילים אל כתובות אינטרנט הנמצאות מחוץ לרשת הפנימית של האירגון, קרי: לתקשר עם מיילים כגון ג'ימייל, יאהו, ומיילים של עובדים מאירגונים אחרים- ישנו "Mail Delivery Server" שאחראי על שליחת המיילים. השרת מתפקד מעין "Default Gateway" לתכנות Outlook ודומיהן.

- דוגמא נפוצה לשרתים כאלה ברשתות חלונאיות היא שרתי ה-Exchange של מיקרוסופט.
- דוגמא נפוצה לשרתים כאלה ברשתות לינוקסאיות/יוניקסאיות היא שרתי Cyrus IMAP עם Sendmail.

מכאן תצורת המערכת יכולה להיות (בדרך כלל) אחת מהשתיים הבאות:

- על השרתים הללו מותקנת תוכנת (או מספר) אנטי-וירוס אשר מבצעת סריקת לכלל הקבצים המצורפים לתכתובות המייל (Attachments) - ומאשר האם להעביר הלאה או האם מדובר בקובץ המכיל וירוס:

- שרת המייל מעביר את הקובץ המצורף לשרת אנטי-וירוס ייעודי אשר מותקנת עליו סביבת אנטי-וירוס ייעודית:



ישנה תצורה נוספת שבה שרת האנטי-וירוס מתפקד גם כ-"Mail Relay" והוא זה השולח את המייל לאחר בדיקתו.

חבילת אנטי-וירוס מוכרת היא למשל GFI MailSecurity הכוללת בתוכה את חמשת מנועי האנטי-וירוס הבאים היודעים לבצע עבודה במקביל:

- McAfee
- Norman
- AVG
- BitDefender
- Kaspersky

לאחר הסריקה-השרתים מחזירים הודעה לשרת המייל האם יש אישור להעביר את הדוא"ל לנמען או האם יש למנוע מהוירוס להגיע ליעדו. חבילה נוספת המוכרת גם היא, היא-McAfee Email Gateway של McAfee. כל מתקפה מוצלחת על אחד מהתצורות שהצגנו תוביל להתקדמות קריטית מבחינת התוקף.

- במידה ומדובר בתצורה הראשונה-קיים סיכוי שתוכנת האנטי-וירוס תקרוס ומעתה-כל אימייל אשר ישלח דרך השרת ישר יעבור לנמען מבלי להיסרק בדרך.
- במידה ומדובר בתצורה השניה/שלישית-קיים סיכוי ששרת האנטי-וירוס יתקע או יקרוס והדבר יוביל ל-Denial of Service על מנגנון שליחת המיילים של האירגון-מה שימנע מכלל המשתמשים לקבל/לשלוח אימיילים על גבי רשת האירגון.

איך אפשר לדעת איזו תוכנת אנטי-וירוס יושבת על השרת של האירגון? פשוט מאוד-תוכנת האנטי-וירוס לרב מוסיפה חתימה לסוף המייל אשר מעידה על כך שהדואר והקבצים המצורפים אליו-נבדקו על-ידייה. לדוגמא, זאת חתימה שנוספה לקובץ שנסרק ע"י תוכנת האנטי-וירוס MailMarshal של m86security:

This e-mail message has been scanned for Viruses and Content and cleared by **MailMarshal**

זאת חתימה שנוספה לקובץ שנסרק ע"י תוכנת האנטי-וירוס MailScanner:

This message has been scanned for viruses and dangerous content by **MailScanner**, and is believed to be clean.

(תודה רבה ל-**spdr** ול-**execute** על החתימות)

גם במקרים בהם שם תוכנת האנטי-וירוס לא מופיעה בחתימת האנטי-וירוס, מספיקה הרצה קצרה של החתימה בגוגל בכדי להבין לאיזו תוכנת אנטי-וירוס שייכת החתימה.

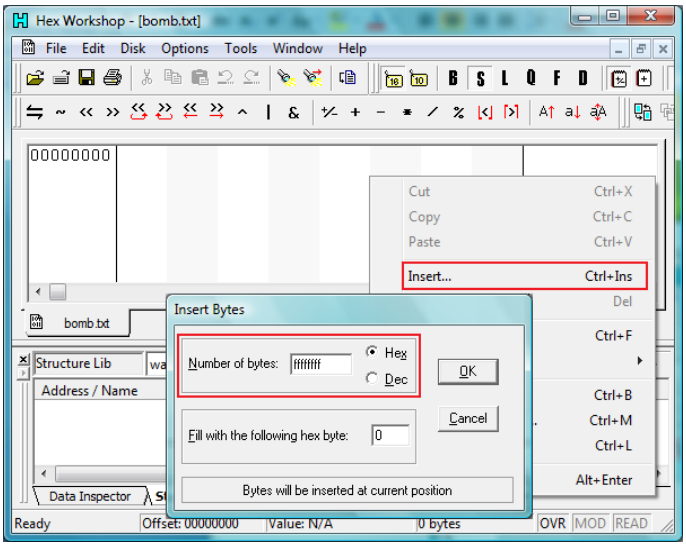
אופן ביצוע המתקפה

שלבי המתקפה ברורים ביותר- הרעיון הוא לשלוח קודם כל את ה-Zip Bombs אל שרתי האירגון ולאחר מכן לשלוח את המייל עם הוירוס/סוס טרויאני. במידה והצלחנו לבצע את המתקפה בהצלחה, או שהוירוס יעבור באופן חלק ללא כל בדיקת אנטי-וירוס, או ששום מייל לא יוכל להכנס לאותו אירגון - או שכלום לא יקרה ☺

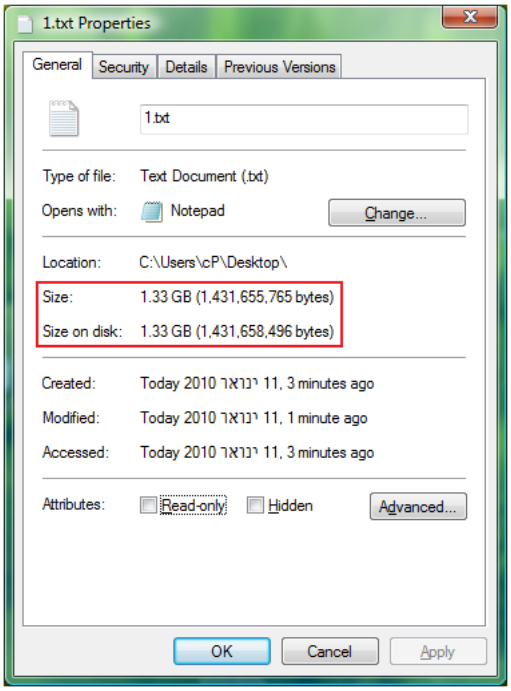
אחרי שהבנו את התיאוריה והרעיון הכללי- בואו נראה איך אפשר להכין Zip Bomb משלנו.

צרו קובץ טקסט בשם: Bomb.txt

פיתחו אותו בעזרת תוכנת Hex Editor והכניסו לו מספר רב של בייטים ריקים:



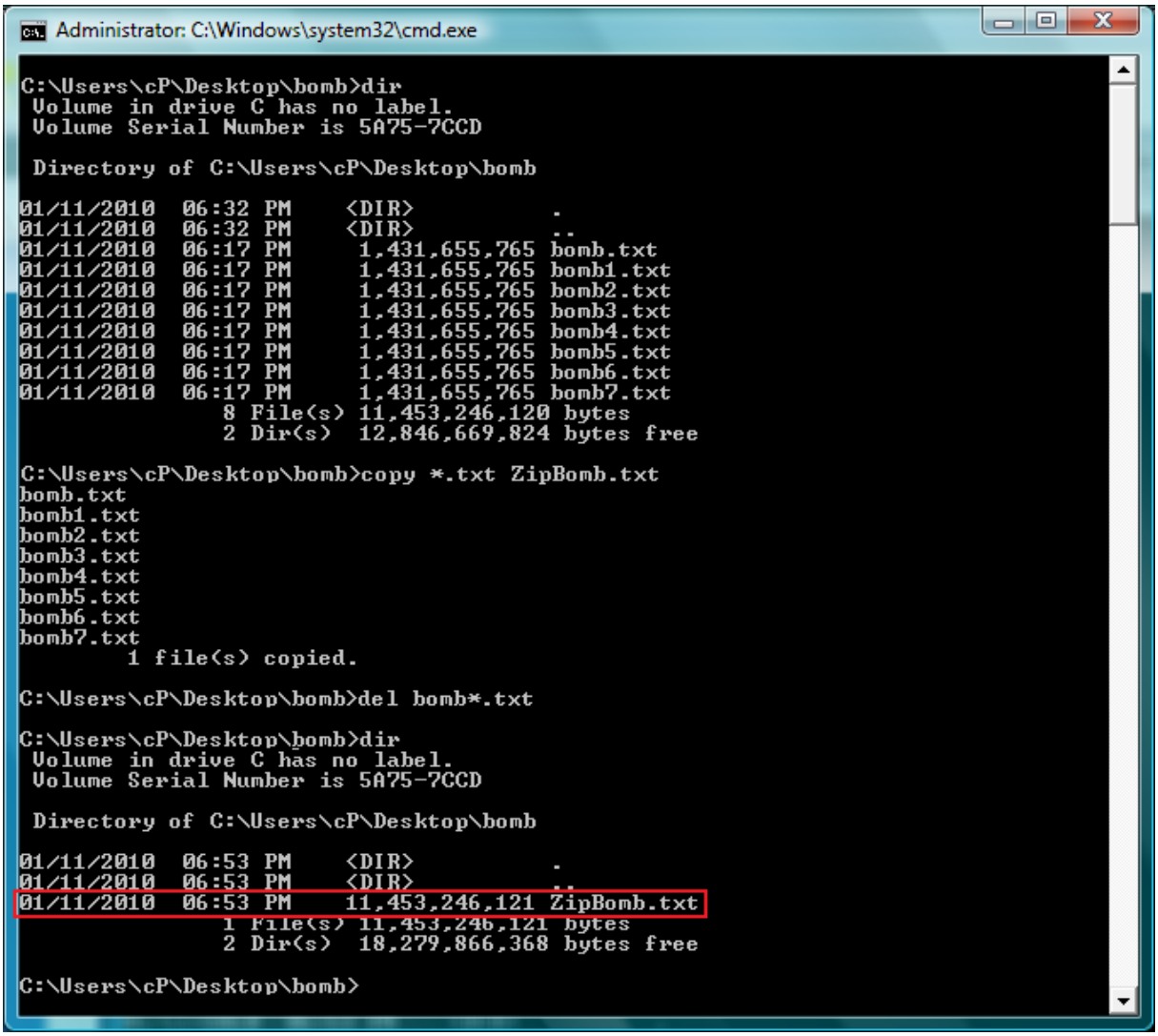
פעולת ה-Filling תמשך קצת-ויש סיכוי שתקבלו שגיאת Access Denied עקב נסיון יצירת קובץ גדול ממה שהמערכת מאפשרת ולכן יש סיכוי שתאלצו להוריד את מספר הבייטים שתרצו להוסיף בתוכנת ה-Hex Editor, לאחר שתוסיפו את הבייטים-שמרו את הקובץ. בסוף הפעולה תקבלו קובץ גדול יחסית מלא בבייטים ריקים:



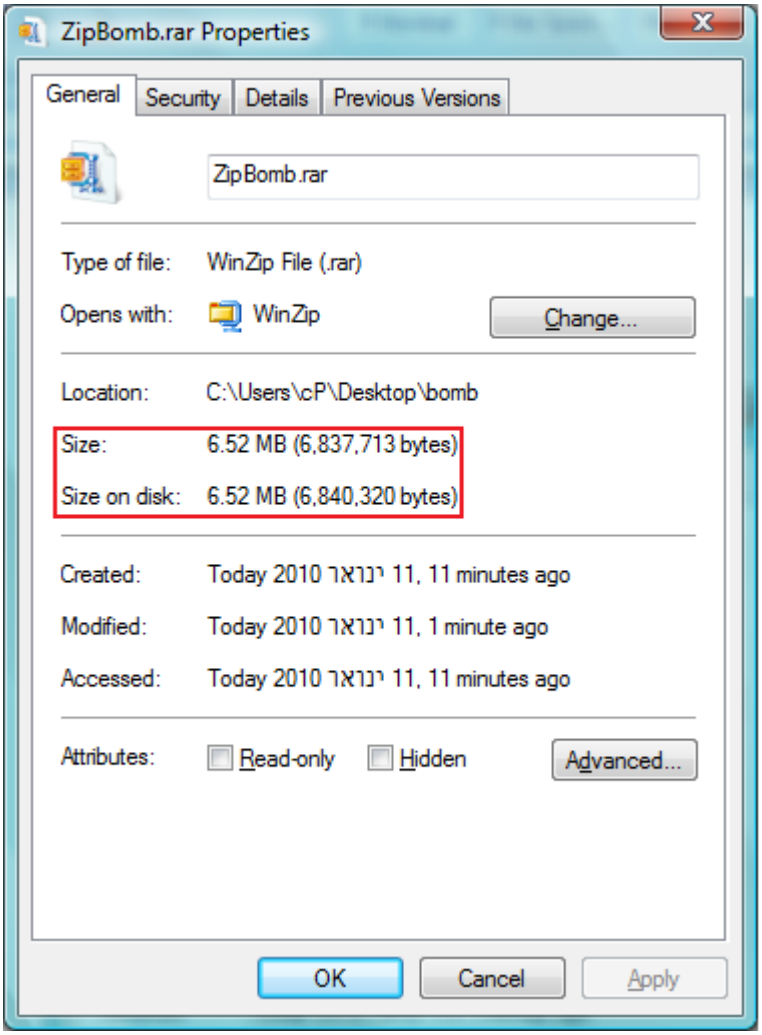
הקובץ אומנם ענק, אך מפני שכלל ערכי הבייטים שבו זהים האנטרופיה שלו גבוהה מאוד. צרו העתקים רבים של הקובץ והכניסו אותם לתיקיה חדשה. ב-CMD נווטו לתיקיה והקלידו את הפקודה:

```
Copy *.txt Zipbomb.txt
```

למעבד יקח פרק זמן מכובד לבצע את הפעולה ולאחר מכן יהיה לכם קובץ אשר יכיל את כלל הקבצים. כעת מיחקו את יתר הקבצים. אנו יצרנו שבעה העתקים חדשים לקובץ הקיים, ואת כל שמונת הקבצים המרנו לקובץ בודד השוקל כמעט 11.5 gigabytes. כאן אנו רק ממחישים לכם את הדבר, אך לצורך בניית הפצצה אנו צריכים נפח גדול הרבה יותר.



כעת, דחסו את הקובץ הנותר בעזרת תוכנת דחיסה כגון WinZip או WinRAR. גם הפעולה הזאת תמשך פרק זמן לא קצר, אך בסיומה תווצר לכם הפצצה. אצלנו הקובץ יצא בגודל של כ-6.5megabytes. אנשים לפנינו הצליחו להגיע לייחס הרבה יותר רציני ויעיל-אך בשביל כתיבת המאמר היחס הזה מספק.



בכדי להגיע ליחס איכותי יותר משלנו אפשר לבצע למשל:

- הכפלה של נתונים הדחיסה בעזרת עורך HexEditor.
- ייעוד סידור תוכן הקובץ באופן ספציפי כלפי אלגוריתם דחיסת הנתונים הספציפי.

הפצצה מוכנה, כעת נותר לנו רק לשגר אותה.

דרכי התגוננות

כיום כבר רב האנטי-וירוסים מצויידים במנגנון כנגד Zip Bombs. ישנם מספר דרכים ליישם מנגנון שכזה:

- ניתוח הקובץ הדחוס בטרם פריסתו והשוואה בינו לבין גודלו העתידי.
- פריסת הקובץ בארגז חול יעודי בלתי-תלוי במשאבי השרת ואתחולו בכל פריסה חדשה.
- קביעת גודל מקסימלי והשוואתו לגודל הקובץ **בזמן פריסת הקובץ** ולא בסופה.
- קביעת פרק זמן מקסימלי לפעולת פריסת הקובץ.

בנוסף, ההמלצה של כל החברות המספקות שרתי אנטי-וירוס היא שהשרת יהיה חזק מאוד, יציב ומתוקצב במשאבים רבים.

סיכום

מתקפות בעזרת פצצות ZIP הם לא דבר חדש, אך עדיין, גם כיום אפשר למצוא תוכנות אנטי-וירוס אשר עדיין רגישות למתקפות מסוג זה, דוגמא לתוכנות אנטי-וירוס שגם כיום פגיעות למתקפות אלה הן:

- Dr. Web cureit - כל הגרסאות (נכון לכתיבת שורות אלה הגרסא החדשה ביותר היא 5.00.9)
- Clam AntiVirus - כל הגרסאות (נכון לכתיבת שורות אלה הגרסא החדשה ביותר היא 0.95.3)

מומלץ בחום לבדוק את גרסאת האנטי וירוס שאתם משתמשים בה במחשבכם.

דברי סיום

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

אנחנו מחפשים כתבים, מאיירים, עורכים ואנשים בעלי כל כשרון לכל דבר (כן, נמצא לכם משהו לעזור בו) המעוניינים לתרום לגליונות הבאים. אם אתם רוצים לעזור לנו ולהשתתף במגזין Digital Whisper - צרו קשר!

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

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

www.DigitalWhisper.co.il

הגליון הבא ייצא ביום האחרון של פברואר 2010.

אפיק קסטיאל,

ניר אדר,

31/1/2010