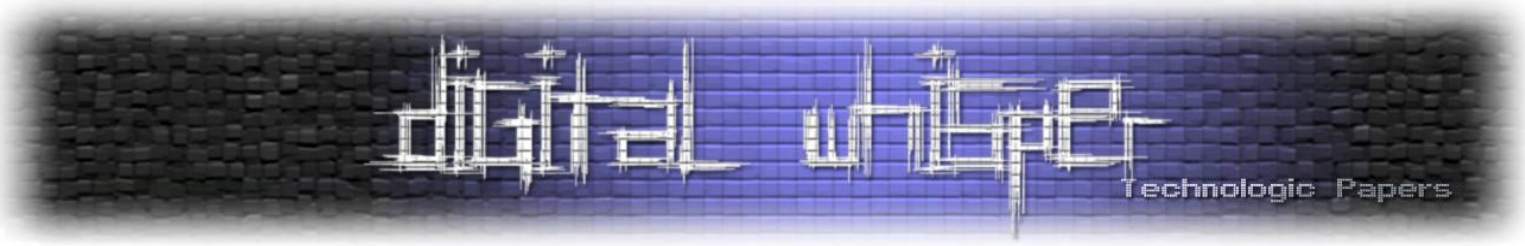


Digital Whisper

גליון 25, אוקטובר 2011

מערכת המגזין:

מייסדים:	אפיק קסטיאל, ניר אדר
מוביל הפרוייקט:	אפיק קסטיאל
עורכים:	ניר אדר, אפיק קסטיאל
כתבים:	גדי אלכסנדרוביץ', אדיר אברהם, עו"ד יהונתן קלינגר, בעז (tsabar), עוז אליסיאן ודור זוסמן.

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

דבר העורכים

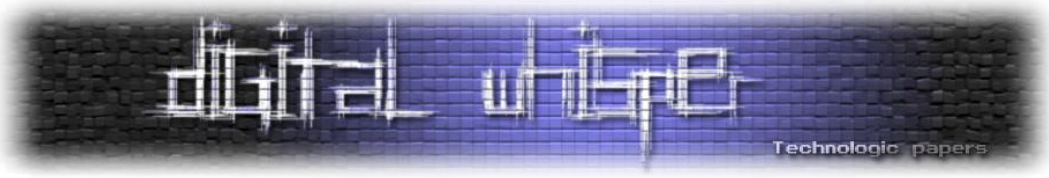
ברוכים הבאים לגליון ה-25 של Digital Whisper!

ראש השנה חלף לו ואיתו באה לה שנה חדשה, אני לא אתחיל לסכם את השנה כמו שהרבה חבר'ה עושים בדרך כלל, גם כי אין לי יותר מדי זמן לזה, וגם כי אין לי משהו חכם כל כך להגיד :) רק אגיד תודה רבה לכל מי שעזר לנו, ולמרות חוסר הזמן שהיה לנו החודש הצלחנו להגיש לכם את המגזין יחסית בזמן...

תודה רבה ל**גדי אלכסנדרוביץ'**, תודה רבה ל**אדיר אברהם**, תודה רבה ל**עו"ד יהונתן קלינגר**, תודה רבה ל**בעז (tsabar)**, תודה רבה ל**עוז אליסיאן** ותודה רבה ל**דור זוסמן**.

קריאה נעימה!

אפיק קסטיאל וניר אדר.



תוכן עניינים

הצפנה מבוססת עקומים אליפטיים

מאת: גדי אלכסנדרוביץ'

הקדמה

בשנת 1993 רעדה האדמה בעולם המתמטי כשאנדרו ווילס הכריז כי הוכיח את המשפט האחרון של פרמה - בעיה פתוחה בת 350 שנים שהייתה אחת מהבעיות הפתוחות המרכזיות במתמטיקה. ווילס הוכיח השערה מודרנית יחסית, שממנה (בזכות עבודתם של כמה מתמטיקאים שקדמו לו) נבעה ההוכחה למשפט האחרון של פרמה, שעסקה בשני אובייקטים מתמטיים מודרניים - עקומים אליפטיים ותבניות מודולריות. בעקבות ההוכחה, והרבה יותר ממנה בזכות הספר שכתב סיימון סינג על הנושא, הפכו שני המושגים הללו למילות קסם עבור העולם הלא-מתמטי; למרות שגם סטודנטים למתמטיקה עשויים שלא להיתקל בהם כלל בלימודיהם, יוצא לי לראות אותם מוזכרים שוב ושוב בידי הדיוטות.

אבל לעקומים אליפטיים ותבניות מודולריות יש חיים בפני עצמם, כל אחד בתחומו שלו. עקומים אליפטיים נחקרים כבר במשך עשרות שנים ועוסקת בהן אחת מההשערות המפורסמות ביותר במתמטיקה כיום - השערת בירץ' וסווינרטון-דייר (שהיא אחד מ"שבע בעיות המילניום", לצד שאלת P=NP למשל), אבל כל התורה העשירה שלהם לא ממש זמינה עבור ההדיוט המתמטי. למרבה המזל, עקומים אליפטיים הצליחו גם להסתנן לתחום הרבה יותר ידידותי למתחילים - קריפטוגרפיה. במאמר הזה אני מקווה לתת טעימה כלשהי מהנושא - לסייע לקורא להבין "מה זה" - גם מה זה עקום אליפטי, וגם מה זו הצפנה שמבוססת על עקומים אליפטיים. דרך ההצגה שלי תהיה שטחית - כאמור, להציג את העקומים האליפטיים במלוא יופיים זו משימה קשה למדי; אבל אני מקווה שהיא תהיה מעניינת גם כך.

נתחיל בהצפנה. אני מניח שהקוראים מכירים את המושג של הצפנת מפתח ציבורי; אם לא, מומלץ ללכת ולקרוא על הנושא כעת. למרות שהצפנה היא בת אלפי שנים, מפתח ציבורי הוא המצאה חדשה יחסית, בת פחות מ-40. המאמר המוקדם ביותר שעסק בה היה זה של דיפי והלמן מ-1976; מאמר שבו הם לא הציגו מערכת הצפנה פומבית (היו אלו ריבסט, שמיר ואדלמן שעשו את זה ב-1977 עם פרסום RSA) אבל הם הציעו שיטה ל**שיתוף מפתחות** באופן פומבי: שיטה שבה שני צדדים מסוגלים ליצור יש מאין מפתח סודי שישמש את שניהם באופן כזה שגם מי שמצותת לכל שיחתם לא מסוגל לדעת מה יהיה המפתח. מכיוון שזה יהיה חשוב מאוד להמשך, אתאר פורמלית את השיטה.

פרוטוקול דיפי-הלמן מתרחש ב"עולם" של המספרים השלמים מודולו ראשוני p; מודולו פירושו שאחרי כל ביצוע פעולת חיבור או כפל, מחלקים ב-p ולוקחים את השארית (אם p=7 אז 1= 4 x 2 כשהחשבון הוא מודולו p). העולם הזה הוא בעל תכונות מתמטיות יפות מאוד - למשל, לכל p קיים מספר $1 \leq g < p$ כלשהו כך ש**כל** מספר בין 1 ו- p-1 מתקבל כחזקה של g (מודולו p). ל-g כזה קוראים **יוצר**. לאוסף המספרים מ-1 ועד p-1, עם פעולת הכפל מודולו p, קוראים "החבורה הכפלית מודולו p" ומסמנים אותה ב-$\mathbb{Z}_p^*$. אני לא סתם זורק עליכם סימונים מפחידים - בהמשך יהיה ברור למה כדאי להכיר את השמות הללו.

פרוטוקול דיפי הלמן פועל כך: ראשית כל דיפי והלמן מסכימים איכשהו באופן פומבי על מספר ראשוני גדול p ועל g שיוצר את $\mathbb{Z}_p^*$. לאחר מכן דיפי מגריל לעצמו מספר a והלמן מגריל לעצמו מספר b והם שומרים את המספרים הללו בסוד לעצמם, אבל דיפי מחשב את g^a ושולח להלמן; והלמן מחשב את g^b ושולח לדיפי. כעת, דיפי מקבל מהלמן את ה-g^b שלו, ואת כל המספר הזה הוא מעלה בחזקת a ומקבל $(g^a)^b = g^{ab}$; והלמן מקבל מדיפי את ה-g^a שלו ומעלה בחזקת b ומקבל $(g^a)^b = g^{ab}$. כעת יש להם מפתח משותף - g^{ab}; ואילו כל מי שצותת להם לא יודע מהו g^{ab} שכן הוא ראה רק את g^a ואת g^b.

דיפי והלמן מסתמכים כאן על קושי של בעיה מתמטית מוכרת - בעיית **הלוגריתם הדיסקרטי**. האם, אם אנחנו יודעים מהו g ומהו g^a, אנחנו מסוגלים לגלות מהו a? התשובה היא כן, ולא. כן, כי יש דרכים לגלות את זה (אפשר למשל לעבור על כל ה-a-ים האפשריים ב-$\mathbb{Z}_p^*$ ולנסות אחד אחד), אבל לא, כי השיטות שאנו מכירים דורשות יותר מדי זמן (כלומר, יותר זמן מאשר חלף מאז תחילת היקום). תיאורטית, ייתכן שמחר בבוקר יגלו דרך לפתור ביעילות בעיות לוגריתם דיסקרטי; בפרט מחשבים קוונטיים, אם יצליחו לבנות אותם אי פעם, יפתרו את הבעיה הזו. בינתיים, בעולם האמיתי, זוהי בעיה קשה ודיפי-הלמן מיושם בהצלחה במקומות רבים (חשוב לציין שדיפי-הלמן כפי שהוצג כאן פגיע מאוד להתקפות מסויימות, ובפרט להתקפת Men-in-the-middle; זה ממש לא סוף הסיפור).

ההצפנה הפומבית של RSA לא מסתמכת על הקושי של לוגריתם דיסקרטי אלא של פירוק לגורמים, אבל יש שיטות אחרות להצפנה פומבית שכן מתבססות על לוגריתם דיסקרטי - אולי המוכרת שבהן היא השיטה של אל-גאמל שגם אותה אני רוצה להציג במהירות. כמקודם, גם כאן ה"עולם" שבו ההצפנה מתרחשת הוא $\mathbb{Z}_p^*$ כאשר p ויוצר כלשהו g ידועים לכולם. בנוסף, דיפי בונה לעצמו זוג של מפתח פרטי d ומפתח פומבי e כך ש-$e = g^d$ ומפרסם לעולם את e, אך d נשאר סודי (ואנו מאמינים שהוא אכן נשאר סודי גם למי שיודע את g^d ואת g בגלל שזוהי בדיוק בעיית הלוגריתם הדיסקרטי).

מה שקורה באל-גאמל הוא הדבר הבא: בהינתן הודעה m, המצפין שרוצה לשלוח לדיפי את ההודעה מגריל לעצמו איזה שהוא מספר k, שולח לדיפי את g^k, ובנוסף לכך שולח לו את $m \cdot y^k$ (כלומר - הוא שולח את m) כשהיא "ממוסכת" על ידי e^k, שהיא ערבוב של המידע הפומבי של דיפי עם המספר k שהמצפין הגריל; ובנוסף לכך הוא שולח לדיפי את g^k כדי שלדיפי יהיה מושג כלשהו על k.

כעת כדי לשחזר את ההודעה המקורית דיפי מחשב את $(g^k)^{-d}(m \cdot e^k) = g^{-kd} \cdot m \cdot g^{kd} = m$ זה חישוב שרק דיפי יכול לעשות כי הוא כולל העלאה של משהו בחזקת d- הסודי.

מה שאני רוצה שתזכרו מדיפי-הלמן ומאל-גאמל הוא שבשניהם התבססנו על פעולות חשבוניות פשוטות בעולם של $\mathbb{Z}_p^*$. השתמשנו רק בכפל ובהעלאה בחזקה, שהיא בעצם קיצור להרבה פעולות כפל; לא השתמשנו כלל בחיבור. זה אומר ששתי השיטות שהצגנו ניתנות, באופן תיאורטי, לשימוש בכל עולם מתמטי שבו קיימת פעולה חשבונית שמתנהגת נחמד כמו כפל. פורמלית, כל מה שצריך הוא שהעולם שלנו יהיה חבורה; למי שלא מכיר את המושג, לא נורא - לא אציג אותו במפורש כי אין זה הכרחי.

העקומים האליפטיים נכנסים למשחק

כעת אפשר סוף סוף להסביר איך כל זה קשור לעקומים אליפטיים. כל עקום אליפטי (ואסביר אוטוטו מה זה בכלל) מהווה **חבורה** בעצמו - מוגדרת על איבריו פעולת "כפל" (שהיא שונה מאוד בהגדרתה מכפל של מספרים) שמתנהגת יפה. זה אומר שאפשר לקחת את דיפי-הלמן ואת אל-גאמל ועוד שלל שיטות דומות ולהשתמש בהן כמעט ללא שינוי גם בתוך היקום של העקום האליפטי. ההבדל הוא שהעולם של העקומים האליפטיים הוא יותר מורכב ועשיר וכתוצאה מכך החבורות שצצות בו הן יותר "קשות לפיצוח" מאשר $\mathbb{Z}_p^*$.

מבלי להיכנס לכל הפרטים, השיטה החזקה ביותר הידועה כיום לפתרון (לא יעיל אבל גם לא גרוע עד כדי כך) של בעיית הלוגריתם הדיסקרטי - תחשיב אינדקסים - לא עובדת על עקומים אליפטיים (בהערת אגב למתקדמים אציין שלעתים קרובות החבורות שצצות בעקומים אליפטיים **איזומורפיות** לחבורות כמו $\mathbb{Z}_p^*$; רק שאת האיזומורפיזם הזה קשה לחשב ובכך נעוץ הקושי. זה ממחיש יפה עד כמה ייצוגים שונים לאותו דבר הם בעלי חשיבות במתמטיקה).

אם כן, במאמר הזה לא אציג לכם כלל שיטות להצפנה מבוססת עקומים אליפטיים כי מה שראינו עד כה - דיפי-הלמן, אל-גאמל - הן **בעצמן** שיטות להצפנה מבוססת על עקומים אליפטיים! מה שבאמת חשוב להבין הוא מהם עקומים אליפטיים וכיצד הם מגדירים חבורה; וכשיש לנו בראש את המוטיבציה המעשית לכך, אני מקווה שיהיה יותר קל לעכל את ההגדרות.

עקומים אליפטיים שייכים לתחום מתקדם למדי במתמטיקה - גאומטריה אלגברית - וככאלו יש להם הגדרה מורכבת ומחוכמת שלא אציג פה בכלל. במקום זאת אציג פה הגדרה פשוטה שגם תלמידי בית ספר יכולים להבין, במחיר אי דיוקים קטנים (שאמנם, חלקם רלוונטיים למי שבאמת רוצה לממש מערכת הצפנה עם עקומים אליפטיים אבל אלוהי המתמטיקה יסלח לי).

גאומטריה אלגברית היא אמנם תחום עמוק מאוד ומורכב מאוד, אבל את נקודת המוצא שלה רואים כבר בשיעורי גאומטריה אנליטית - הרעיון שאובייקטים גאומטריים ניתנים לתיאור באמצעות משוואות. כך למשל קו ישר בזווית של 45 מעלות מתואר על ידי המשוואה הפשוטה $y = x$. יותר במפורש, זה אומר שאוסף הנקודות (x, y) במישור שמקיימות את היחס $y = x$ בין הקואורדינטות שלהן מהוות את הישר המדובר.

אובייקט יותר מחוכם שמוגדר באמצעות משוואה הוא מעגל: אוסף כל הנקודות (x, y) במישור שמקיימות את המשוואה $x^2 + y^2 = 1$ הוא מעגל שרדיוסו 1 ומרכזו בראשית הצירים. אם רוצים לתאר מעגל מחוכם יותר, שמרכזו בנקודה (x, y) ורדיוסו R מקבלים את המשוואה $(x - a)^2 + (y - b)^2 = R^2$ בדומה לומדים בתיכון (לפעמים) לייצג גם כמה אובייקטים גאומטריים מורכבים יותר - אליפסה, היפרבולה, פרבולה... אבל כאן זה בערך נגמר.

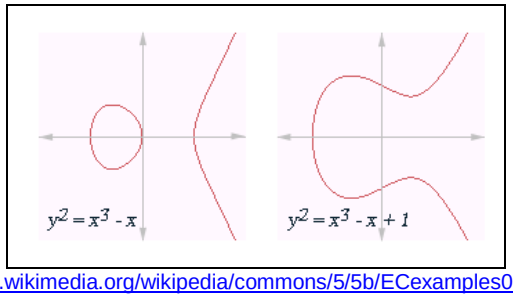
בגאומטריה אלגברית, תחת שנחפש משוואות שמתארות אובייקט גאומטרי שאנחנו כבר מכירים, אנחנו מנסים להבין פתרונות למשוואות **כלשהן** באמצעות חשיבה עליהם בתור אובייקטים גאומטריים ותקיפה שלהם מכל כיוון אפשרי - גם אלגברי וגם גאומטרי וגם שניהם יחד. עקומים אליפטיים הם מאותם יצורים שצצים כך: אוספי פתרונות למשוואה מסויימת שתכונותיהם הגאומטריות היפות מלמדות אותנו משהו עליהם.

המשוואה שמגדירה עקום אליפטי נראית בדרך כלל כך:

$$y^2 = x^3 + ax + b$$

במקרים פרטיים מסויימים המשוואה מסובכת יותר אבל אמרתי שלא אכנס לכך כאן. במובן מסויים עקום אליפטי הוא אך במרחק צעד אחד מרמת הסיבוך של מעגל או אליפסה - שם היו לנו x^2 ו-y^2 ומקדמים ממעלות נמוכות יותר, וכאן הגדלנו רק קצת את המעריך של x כך שגם x^3 ישתתף במשחק. בפועל, השינוי הקטן הזה כבר משנה לחלוטין את המשחק כולו.

הנה תמונה של איך נראה עקום אליפטי טיפוסי:



(במקור: http://upload.wikimedia.org/wikipedia/commons/5/5b/ECexamples01.png)

כפי שאפשר לראות, היצור הזה סימטרי סביב ציר x (זה נובע מכך שיש לנו y^2 באגף שמאל של המשוואה) ופרט לכך הוא לא נראה מרשים או יפה במיוחד. קרוב לודאי שאתם תוהים למה להתחיל להסתכל מלכתחילה על יצורים שכאלו. התשובה היא שבהקשרים מתמטיים מסויימים הם צצים באופן טבעי יחסית, אבל קשה להביא דוגמאות פשוטות שאפשר להציג בשורה אחת.

דוגמה אחת שאני כן רוצה להזכיר כאן היא **בעיית המספרים הקונגרואנטיים**. מספר קונגרואנטי (שם איום ונורא שאני לא מבין מהיכן צץ) הוא מספר טבעי n כך שקיים משולש ישר זווית שאורכי צלעותיו כולם רציונליים, ושטחו הוא בדיוק n. השאלה אילו מספרים הם קונגרואנטיים העסיקה כבר את היוונים הקדמונים - מסתבר ש-2, למשל, אינו קונגרואנטי, 5, 6 ו-7 כן, 11 לא, 13 כן, ועוד ועוד. בקיצור, זו לא שאלה טריוויאלית. למעשה, עד היום לא ידועה שום שיטה להכריע בודאות, בהינתן n, האם הוא קונגרואנטי או לא (אם הוא קונגרואנטי אפשר בתיאוריה על ידי חיפוש סדרתי למצוא משולש מתאים עבורו; אבל אם הוא אינו קונגרואנטי לא ידועה כיום דרך להכריע זאת).

והנה, בעזרת להטוטים אלגבריים מסתבר שקריטריון שקול לכך שמספר n יהיה קונגרואנטי הוא שלמשוואה של העקום האליפטי $y^2 = x^3 - n^2x$ יהיה פתרון רציונלי שאיננו $(0,0)$ כאן נשלפים כלים כבדים שעוסקים בעקומים אליפטיים ומניבים לבסוף קריטריון (שלא אתאר כאן) שמאפשר להכריע חד משמעית האם לעקום הזה יש פתרון רציונלי שכזה או לא; אבל הנכונות של הכלים הללו מסתמכת על אותה השערת בירץ' וסווינרטון-דייר שהזכרתי בתחילת הפוסט ולכן הבעיה טרם נחשבת פתורה.

אבל מהי הפעולה?

נחזור לענייננו. כדי שאפשר יהיה להשתמש בעקומים אליפטיים להצפנה, צריך להבין מהי פעולת ה"כפל" שאפשר לבצע עליהם. מכיוון שההגדרה מוזרה, אני רוצה להסביר מהיכן היא מגיעה. אם E מייצגת את המשוואה של עקום אליפטי, זה עדיין לא אומר לנו מיהו העקום; יש חשיבות אדירה לשאלה מהיכן נלקחים הפתרונות של המשוואה. אם אנו דורשים שהם יהיו מספרים רציונליים, זה עקום מעל המספרים הרציונליים $\mathbb{Q}$, ומסמנים זאת $E/\mathbb{Q}$. אם לעומת זאת אנחנו מרשים לפתרונות להיות מספרים מרוכבים, זה מסומן $E/\mathbb{Q}$ ואנחנו מקבלים אובייקט שונה למדי באופיו. מעל המרוכבים, ניתן לחשוב על אוסף הפתרונות של עקום אליפטי כעל **טורוס** ,או בשמו הפחות פורמלי - בייגלה. להסביר איך בדיוק קורה הדבר המוזר הזה אי אפשר כרגע, אבל חשוב מה שנובע מכך: במרוכבים אפשר להגדיר פעולת חיבור בקלות רבות על איברים מתוך הטורוס - זהו החיבור הרגיל במספרים מרוכבים ועוד פעולת מודולו (רק שבניגוד למודולו p שדיברנו עליו בהתחלה, כאן המודולו הוא של שני איברים שמייצגים "כיוונים" שונים). פעולת החיבור הכמעט-טבעית הזו בטורוס משרה על העקום האליפטי פעולת "כפל" כלשהי, שהיא מה שאני עומד לתאר כאן. אותה הגדרה של פעולת הכפל עובדת גם עבור עקומים אליפטיים מעל שדות אחרים, ובפרט אלו שמשמשים אותנו בהצפנה ואתאר בקרוב. כדי לבלבל אתכם עוד יותר אני אפסיק לקרוא לפעולה הזו "כפל" ואסמן אותה דווקא ב-$+$(למרות שהיא ממש לא פעולת חיבור) כי זה הסימון הנהוג בספרות.

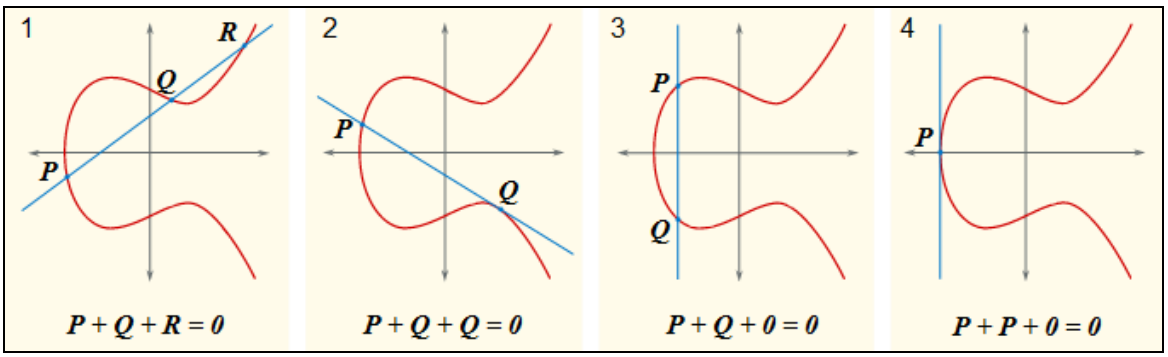
הדרך הפשוטה ביותר לחשוב על הפעולה היא באופן גאומטרי, וכדאי להסתכל שוב על התמונה של עקום אליפטי מעל הממשיים כדי להבין מה בדיוק קורה. בהינתן שתי נקודות על העקום, אנו מעבירים קו שמחבר אותן וממשיכים אותו עד שיחתוך את העקום בנקודה שלישית. את אותה נקודה שלישית אנו משקפים ביחס לציר x, והתוצאה היא תוצאת פעולת החיבור. כלומר, אם A, B הם שתי נקודות על העקום$A + B$, היא מה שמקבלים אחרי העברת קו, חיתוך עם העקום בנקודה שלישית, ואז שיקוף.

אולי השאלה הראשונה שאפשר לשאול היא מה קורה אם כשמעבירים קו בין A, B הוא כלל לא חותך את העקום בנקודה שלישית. זה קורה בדיוק כאשר A, B מחוברים על ידי קו אנכי. במקרה הזה אומרים -

לנשום עמוק - שהקו חותך את העקום ב"אינסוף". מה שבאמת מדהים פה הוא שיש פורמליזם מתמטי מדויק לחלוטין שמתאר את זה (משהו שמכונה "מישור פרוייקטיבי") אבל אין צורך להבין אותו כאן. מה שחשוב הוא שכל עקום אליפטי כולל נקודה נוספת $\mathcal{O}$ שהיא התוצר של חיבור שתי נקודות שמחוברת בידי קו אנכי. בנוסף, ל-$\mathcal{O}$ התכונה ש-$A+\mathcal{O}=A$ לכל נקודה על העקום: $\mathcal{O}$ הוא "אדיש חיבורי" כמו 0 במספרים השלמים הרגילים. בנוסף, אם $A+B=\mathcal{O}$ אז מסמנים זאת בתור $B=-A$. פורמלית, אם A היא הנקודה (x,y), אז $-A$ היא הנקודה $(x,-y)$, **אותה** קוארדינטת x, אבל קוארדינטת y בעלת סימן הפוך (זהו שיקוף ביחס לציר x).

כעת אפשר לתת ניסוח אחר לפעולת החיבור: אם A,B,C הן שלוש נקודות של העקום האליפטי שנמצאות על אותו קו ישר, אז $A+B+C=\mathcal{O}$. חשבו רגע מדוע ההגדרה הזו זהה להגדרה שנתתי קודם, שבה גם מבצעים שיקוף כלשהו.

כאן רואים דוגמאות לאופן שבו מוגדרת הפעולה:



עוד משהו שלא ברור עדיין הוא מה קורה כאשר מחברים נקודה עם עצמה. כלומר, מהי $A+A$? במקרה הזה מה שעושים הוא להעביר קו שמשיק לעקום האליפטי בנקודה A, ואז כרגיל מוצאים את נקודת החיתוך הנוספת של המשיק הזה עם העקום, משקפים ביחס לציר x וקיבלנו את $A+A$. כדי שיהיה לעקום משיק יחיד בנקודה A נדרש שהוא יהיה "חלק" - מבחינה מתמטית זוהי הדרישה שהנגזרת על פי x והנגזרת על פי y של המשוואה שמגדירה את העקום לא תתאפסנה בו זמנית.

טוב, הזהרתי אתכם שהפעולה הזו היא מוזרה למראה במבט ראשון. התחושה העיקרית שלי כשראיתי אותה הייתה "אוקיי, אבל איך לעזאזל עושים את החישובים הללו בפועל?" והתשובה לכך, למרבה המזל, היא "בקלות". לא עד כדי כך קשה להגיע ישירות למשוואות שמגדירות את הפעולה. אם $(x_1,y_1)+(x_2,y_2)=(x_3,y_3)$ ואנו מניחים ש-$x_1\neq x_2$ (כי אם $x_1=x_2$ הנקודות על אותו קו אנכי ואז סכומן הוא פשוט $\mathcal{O}$) אז הנקודה (x_3,y_3) מתוארת על ידי הנוסחאות:

$$x_3 = \left(\frac{y_2-y_1}{x_2-x_1}\right)^2 - x_1 - x_2$$
$$y_3 = -y_1 + \left(\frac{y_2-y_1}{x_2-x_1}\right)(x_1 - x_3)$$

ואילו אם $(x_1, y_1) = (x_2, y_2)$, כלומר הסכום מוגדר בעזרת המשיק, אז הנוסחה היא:

$$x_3 = \left(\frac{3x_1^2+a}{2y_1}\right) - 2x_1$$
$$y_3 = -y_1 + \left(\frac{3x_1^2+a}{2y_1}\right)(x_1 - x_3)$$

כאשר a הוא חלק מהמשוואה שמגדירה את העקום $y^2 = x^3 + ax + b$.

בעזרת הנוסחאות הללו, אפילו אם לא מבינים מהיכן הן באות (וכאמור - זה לא מורכב כל כך והמתמטיים שבכם אולי ירצו לנסות ולפתח אותן בעצמם) אפשר לבצע את פעולות החשבון בעקום אליפטי באופן קונקרטי לחלוטין. אנחנו רואים שביצוע פעולת חיבור הוא תובעני יותר מאשר חיבור של שני מספרים רגילים - יש כאן הרבה חיבורים, וכפל, וחילוק וכו'; זו הסיבה שעקומים אליפטיים הם תובעניים יותר מבחינת זמן ריצה מאשר חשבון ב-$\mathbb{Z}_p^*$ למשל. הרבה מהמחקר על הצפנה מבוססת עקומים אליפטיים עוסק באופן שבו ניתן לשפר את הביצועים הללו.

סיכום

לסיום, אעיר שלא התייחסתי כלל לדבר המעניין ביותר - איך בכלל מייצרים עקומים אליפטיים? לכאורה אפשר פשוט להגריל a, b ולעבוד עם העקום $y^2 = x^3 + ax + b$, אבל המציאות אף פעם אינה כה פשוטה. ראשית, לא בטוח שלכל a, b אכן נקבל עקום (זכרו את הדרישה שהעקום יהיה "חלק" שהזכרתי קודם), אבל שנית - לא כל עקום שמגרילים ברחוב יהיה בטוח. דרישה חשובה אחת היא שכמות הנקודות שעל העקום תהיה מספר גדול בלי יותר גורמים ראשוניים קטנים, וזה מעלה את השאלה - בהינתן עקום, איך יודעים כמה נקודות יש עליו? זו בעיה מעניינת מאוד בזכות עצמה, שמעידה אולי על כך שגירדתי לבינתיים רק את קצה הקרחון.

על המחבר

גדי אלכסנדרוביץ, בעל תואר ראשון במתמטיקה ותואר שלישי במדעי המחשב מהטכניון, כותב את הבלוג "לא מדויק" העוסק במתמטיקה ובמדעי המחשב:

http://www.gadial.net

זיהוי ומניעת חיבור שרותי פרוקסי אנונימיים

מאת: אדיר אברהם (Adir@computer.org)

הקדמה

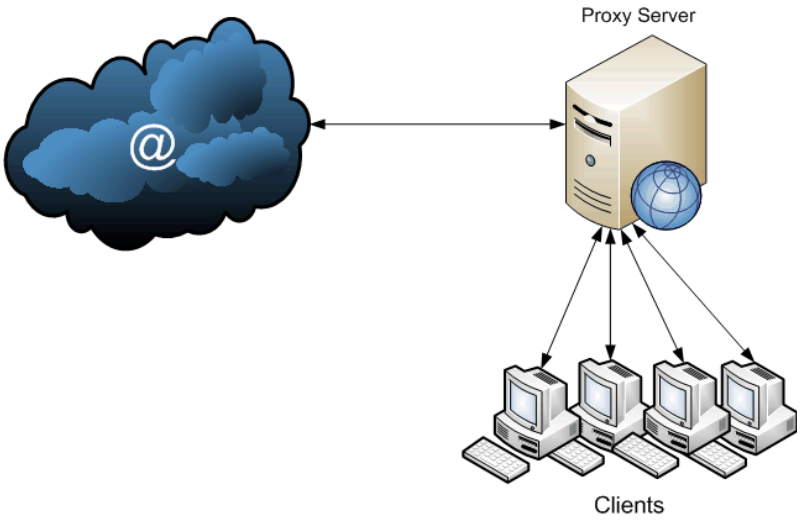
במהלך השימוש של משתמשים שונים במערכת מתרחשים אירועים שונים אשר מחייבים את מעורבות מנהל המערכת או מומחה אבטחת המידע המקומי. במהלך הפעלת השרת וניטורו מתגלים נסיונות חדירה למערכת אשר מגיעים ממקומות שונים ברשת, כאשר אחד המקומות הבעייתיים ביותר הוא משרתי פרוקסי. נתחיל בשאלה למה מיועדים שרתי פרוקסי?

- לאכוף איסורי-גישה פנים-ארגוניים שונים.
- להגדיל את המהירות למשאבי הרשת השונים.
- לבצע caching בין אתרים או דפים פופולריים לבין מחשבים ספציפיים שניגשים אליהם תכופות.
- להתיר גישה לתוכן מסויים או לאסור גישה אליו.
- לבצע גישה פנים-אירגונית בעזרת login מסודר דרך שרת ספציפי.
- לסרוק מעבר מידע לפני הגעתו לתוך האירגון או מחוצה לו מוירוסים ומזיקים אחרים.
- למנוע דליפת מידע רגיש החוצה (DLP).
- לשמור על המחשבים שנמצאים בתוך הארגון כאנונימיים.

אם נסכם את מטרת שרתי הפרוקסי - שרת פרוקסי משמש כמתווך בין בקשות של לקוחות לבין שירותים או שרתים אחרים אשר מהם הם מיועדים לקבל את המידע. שרת הפרוקסי מתוקף תפקידו יכול "להחליט" איזה מידע להעביר הלאה מצד הלקוח ואיזה להשאיר. באופן כזה, שרת הפרוקסי עשוי לשנות את המידע המגיע המזהה את הלקוח ולסננו כך שהלקוח לא יזוהה.

שימוש נוסף עליו לא נדבר כאן, הוא ששרת הפרוקסי יכול להשאיר אצלו את המידע מהשרת ממנו אנו מייעדים לקבל שירות רלוונטי, כך שנקבל משרת הפרוקסי את המידע ישירות ולא נצטרך להתחבר אל השרת המרוחק. האופציה האחרונה מאפשרת מתן שירות יעיל יותר ע"י שיפור המהירות (תהליך הנקרה "Caching").

בגדול, זה נראה כך:



(התמונה פורסמה במקור במאמר "Java Java Proxy Proxy", שנכתב על-ידי רועי חורב (AGNil) ופורסם בגליון ה-11 של המגזין. - שווה לקרוא בכדי להבין יותר על תפקידיהם של שרתים אלו)

זן "מיוחד" של שרתי פרוקסי הוא שרת פרוקסי אנונימי, המאפשר למעשה סינון מוחלט של זיהוי הלקוח המתחבר (למשל, את כתובת ה-IP שלו, פרטי הדפדפן שלו וכו'), ואם לקוח זה יתחבר דרך שרת הפרוקסי אל השרת שלך, הוא למעשה יוכל לדלג על איסורים שונים שמנהל הרשת קבע במערכת שלו, למשל חיבור מתחום כתובות IP מסויים (כגון מדינה).

כיום השימוש בשרת פרוקסי אנונימי נעשה נפוץ יותר לנסיונות החדירה הרבים לשרתים שונים. במאמר זה נדבר על זיהוי ומניעת שימוש בשרתי פרוקסי שונים.

זיהוי פרוקסי על-ידי כותרים

לא תמיד, אך במספר לא מבוטל של מקרים, שרתי HTTP Proxy מוגדרים להוסיף מספר כותרים (Headers) לבקשת ה-HTTP שהמשתמש שולח. דוגמא לכותר כזה הוא כותר ה-"X-Forwarded-For" (או מוכר גם כ-"XFF"), הערך שיגיע לאחר הכותר הנ"ל הוא כתובת ה-IP המקורית של לקוח הפרוקסי, בכדי למנוע את האפשרות להתחבר לשרת שלנו בעזרת שרותי פרוקסי המוסיפים את הכותר הנ"ל פשוט מאוד נוכל לבדוק האם בבקשת ה-HTTP שקיבלנו- מופיע הכותר הנ"ל. במידה הוא אכן מופיע אז כמעט ואין ספק כי מדובר בלקוח הגולש דרך שרת פרוקסי.

חשוב לציין כי הכותר הנ"ל אינו מופיע ב-RFC המקורי של הפרוטוקול והוא נוסף על-ידי המפתחים של שרת הפרוקסי Squid.

כותרים עם תפקידים דומים שהמצאותם בבקשת ה-HTTP שהגיע לשרת שלנו תוכל להדליק אצלנו נורה אדומה הם: (ותודה רבה לרועי / Hyp3rInj3ct10n שאסף את הכותרים הללו והציג אותם במאמר "Playing With HTTP" שפורסם בגליון השמיני של Digital Whisper!)

- Client-IP
- Proxy-User
- Forwarded
- Useragent-Via
- Proxy-Connection
- Xproxy-Connection
- Pc-Remote-Addr
- Via

דוגמא לחסימת בקשות HTTP הכוללות את אחד מהכותרים הנ"ל בעזרת שימוש בקבצי htaccess. ניתן לראות בקוד הבא:

```
RewriteEngine on
RewriteCond %{HTTP:VIA}                  !^$ [OR]
RewriteCond %{HTTP:FORWARDED}            !^$ [OR]
RewriteCond %{HTTP:USERAGENT_VIA}        !^$ [OR]
RewriteCond %{HTTP:X_FORWARDED_FOR}      !^$ [OR]
RewriteCond %{HTTP:PROXY_CONNECTION}     !^$ [OR]
RewriteCond %{HTTP:XPROXY_CONNECTION}    !^$ [OR]
RewriteCond %{HTTP:HTTP_PC_REMOTE_ADDR} !^$ [OR]
RewriteCond %{HTTP:HTTP_CLIENT_IP}       !^$
RewriteRule ^(.*)$ - [F]
```

הקוד מופיע במקור בבלוג "perishablepress.com", בקישור:

http://perishablepress.com/press/2008/04/20/how-to-block-proxy-servers-via-htaccess/

במידה ואתם מכירים שרת המשמש כשרת פרוקסי ציבורי, ניתן לשמור את כתובתו בתוך רשימה שחורה ולהפיץ אותה באינטרנט. באופן זה בעלי שרתים אחרים יוכלו להשתמש ברשימה זו גם כן ובמקום לעדכן בכל פעם מחדש את שרתי הפרוקסי המתגלים, תוכלו להשתמש ברשימה שמכילה מקרים קודמים כדי לסנן חיבורים בלתי רצויים.

אמנם, רשימה כזו מתעדכנת וגדלה כל הזמן, אך השימוש בה לטווח הרחוק שווה את ההשקעה, מכיוון שכך אנו גורמים לשרתי פרוקסי אנונימיים זדוניים לא להיות בשימוש. בנוסף, לעתים אין דרך מסודרת לגלות שרתי פרוקסי אנונימיים בעייתיים, ואז בידיעה על אחד מהם נוכל להוסיף אותם לרשימה בעצמנו ולמנוע את השימוש בהם אצלנו.

לדוגמא, אחד האתרים שמכילים רשימה כזו הוא האתר http://proxy4free.com. נוכל לדלות את הרשימה ממנו, למשל ע"פ הדומיין באמצעות הפקודה הבאה:

```
curl http://proxy4free.com/list/webproxy_domain1.html > proxy_list1.html
curl http://proxy4free.com/list/webproxy_domain2.html > proxy_list2.html
curl http://proxy4free.com/list/webproxy_domain3.html > proxy_list3.html

…
```

לאחר שהורדנו את דפי ה- HTML אלינו, נהפוך את רשימת הדומיינים לקובץ טקסט אחד גדול ונשתמש בו:

```
grep whois\.cgi\?domain\=proxy_list1.html | cut -d \= -f 3 | cut -d \" -
f 1 | sort | uniq > proxy_list.txt
```

```
grep whois\.cgi\?domain\=proxy_list2.html | cut -d \= -f 3 | cut -d \" -
f 1 | sort | uniq >> proxy_list.txt
```

```
grep whois\.cgi\?domain\=proxy_list3.html | cut -d \= -f 3 | cut -d \" -
f 1 | sort | uniq >> proxy_list.txt
```

נוכל לעדכן את הרשימה באופן יום-יומי ואוטומטי (בעזרת cron/schtasks) מתוך הרשימה הראשית הנ"ל, כך שלא נצטרך לעדכן אותה בעצמנו בפועל.

זיהוי שרתי TOR

TOR הוא פרויקט המאפשר גישה אנונימית ברשת ע"י שימוש ברשת מבוזרת של שרתים ברחבי העולם. (ניתן לקרוא עוד על הפרוייקט במאמר שפורסם בגליון השביעי של Digital Whisper על-ידי ליאור ברש) שרתי TOR עובדים על פורטים 9001-9004, 9030-9033 ו-9100 בנוסף לפורט 80 ו- 443 הסטנדרטיים.

נוכל להשתמש ב-Snort כדי לזהות שימוש בשרת TOR:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 80,443,9001,9030 (msg: "TOR
client access
detected"; pcre:"/.*(Tor).+(client <identity>).*/i"; classtype:policy-
violation; sid:50009;
```

שורה זו תגלה שימוש בפורטים הנ"ל ותתריע על גישת TOR, כמובן שבמידה ותוכנה אחרת תבצע שימוש בפורט הנ"ל בכדי להעביר את המידע לשרתים שלנו- נקלע ל-False Positive.

בנוסף, בעזרת אינטגרציה קלה, ניתן לבצע שאילתות מול אתר כגון torstatus.blutmagie.de כדי לדעת האם כתובת ה-IP שמנסה להתחבר משמשת כתחנת TOR.

איתור תבניות פרוקסי ידועות

PHPProxy

לשרתי פרוקסי ישנן "חתימות" ידועות הבנויות ממנועים עיקריים. אחד מהם הוא PHPProxy שנמצא באתר:

http://sourceforge.net/projects/phpproxy

למשל, אם ננסה להתחבר לאתר ynet דרך שרת הפרוקסי שנמצא בכתובת proxy.arcticgames.net, אנחנו נקבל את הכתובת הבאה:

http://proxy.arcticgames.net/proxy/index.php?q=aHR0cDovL3d3dy55bmV0LmNvLmlsL2hvbWUvMCw3MzQwLEwtOCwwMC5odG1s

כלומר, התבנית היא:

```
{hostname}/index.php?q={encoded_URL}
```

במצב כזה, שרת הפרוקסי יפתח מסגרת בדף כך שבתוך המסגרת יש את האתר אליו אנו רוצים לגשת. אתר האינטרנט אליו ניגשנו יראה את כתובת ה-IP וכן את החתימה של proxy.arcticgames.net ולא שלנו.

הכתובת הנ"ל מקודדת בעזרת Base64, אותו ניתן לפרש בעזרת כלים / אתרים ייעודיים כגון:
www.opinionatedgeek.com/dotnet/tools/base64decode

מספיק שנכניס את המחרוזת לאחר ה-"?q=" ונראה את הכתובת המקורית. באופן זה נוכל לדעת לאיזה דף ניסו לגשת בפועל. בנוסף לכך נוכל לראות לעתים מקרים דומים, כגון "Rotate13" אשר מקדם את האותיות 13 פעמים קדימה, כך ש-"www" יופיע בתור "jjj" וכן הלאה.

לסיכום, כדי לגלות שימוש בשרתי פרוקסי מסוג PHPProxy, נוכל לזהות ולסנן בקשות מהסוג הנ"ל ע"י שימוש בשורה הבאה:

```
grep -Ei '(index\.php\?q=).+(&hl).*' proxy_list.txt
```

וכך נכניס עוד שרתי פרוקסי לרשימה.

CGIProxy

CGIProxy הוא מנוע פרוקסי נוסף אשר בנוי בשפת Perl. בתור ברירת מחדל אין ערבול של ה-URL אליו אנו רוצים להתחבר, אך יש שימוש אופציונלי ב-Rotate13 וב-Base64.

למשל, עבור אתר הפרוקסי הבא:

http://rosinstrument.com/cgi-proxy.htm

נוכל להתחבר לאתר ynet ונקבל את הכתובת הבאה:

http://antt.tk/u.php/Oi8vd3d3/LnluZXQu/Y28uaWwv/aG9tZS8w/LDczNDAs/TC04LDAw/Lmh0bWw_/3D/b5/

לאחר שנוריד את כל ה-"/" לאחר "u.php" עד הסימן "_", נקבל את המחרוזת הבאה:

```
Oi8vd3d3LnluZXQuY28uaWwvaG9tZS8wLDczNDAsTC04LDAwLmh0bWw
```

כש"נפענח" את הכתובת הנ"ל, נגלה שהיא נותנת את ynet.

לכן, כדי לזהות פרוקסי מסוג CGIProxy, אנו נרצה להכניס פילטר מהסוג הבא (דרך grep או Snort):

```
/(u\.php/).+/.+/.+(/b).+/
```

Glype

באופן דומה ל-PHPProxy, Glype מכיל מחרוזת שמשתמשת ב-browse.php. למשל, עבור האתר http://glype-proxy.info ואתר האינטרנט ynet, אנחנו נקבל את הכתובת הבאה:

```
http://www.glype-
proxy.info/browse.php?u=Oi8vd3d3LnluZXQuY28uaWwvaG9tZS8wLDczNDAsTC04LDAw
Lmh0bWw%3D&b=5
```

ונוכל לסנן אותה בעזרת המחרוזת הבאה:

```
(browse\.php\?u=).+(&b).*
```

ישנם עוד מספר רב של שרתי פרוקסי מבוססי Web בסיגנון זה, אך אני מאמין כי הבנתם את העקרון.

SSL Proxy

ישנם שרתים המאפשרים להצפין את העברת המידע בפועל, וכך הגילוי שלהם קשה עוד יותר. אלו שרתים שהונפקו עבורם רשיון SSL. חסרונם הוא ששרתים אלו בדר"כ משלמים על הנפקת הסרטיפיקטים ולכן יהיה קל לחסום אותם בעזרת רשימה שחורה במידה ונזהה חיבור דרכם.

מתרגמים

שימוש לא שגרתי כפרוקסי הוא המתרגמים (translators) למיניהם שנמצאים ברשת. המפורסם שבהם הוא http://translate.google.com המתרגם מילים, משפטים ואתרים שלמים משפה אחת לשפה אחרת ע"פ בחירה. מתרגמים נוספים נמצאים גם אצל Yahoo וכן אצל Microsoft.

לעתים, נרצה להשתמש במתרגמים לא רק בשביל לתרגם, אלא גם בשביל להסתיר את כתובת ה- IP שלנו, שכן נקבל את המידע המתורגם ישירות מהמתרגם, והאתר יופיע בתוך מסגרת בדומה לתבנית של PHPProxy.

נניח שנרצה לתרגם את ynet לאנגלית, אז ניגש לאתר ונכניס לתוך המסגרת את שם האתר, ונקבל את הכתובת הבאה:

http://translate.google.com/translate?sl=auto&tl=en&js=n&prev=_t&hl=en&ie=UTF-8&layout=2&eotf=1&u=ynet.co.il

נשים לב כי ynet.co.il מופיע בתור מחרוזת פשוטה. כמו-כן, "&sl" היא שפת המקור, "&tl" היא שפת היעד ו-"&ie" הוא ה-character set שלנו.

נוכל לחסום מחרוזת מן הצורה הזו אם נרצה לחסום את גישת המתרגם אלינו. במקרה כזו אנו עלולים לפגוע בלא מעט משתמשים תמימים שירצו להשתמש באתר שלנו, אך במידה ויש גישה נרחבת דרך המתרגם ויש יותר מדי נסיונות לגשת לדפים שלא קיימים, כדאי לשקול את אופציית החסימה.

זיהוי גישה דרך פרוקסי אל השרת שלנו

כשדפדפן מתחבר אל האתר הוא מזדהה באמצעות מחרוזת מסוג User Agent, המכילה את המידע על הדפדפן, גרסת הדפדפן, מערכת ההפעלה ועוד.

כדי לאפשר חיבור דרך פרוקסי, הפרוקסי צריך לתת מחרוזת כזו ייחודית השונה ממחרוזת ה- User Agent המקורית. דרך לנסות לגלות מי התחבר אל השרת שלך היא ע"י סריקה אחורה. למשל, אם התחברו אליך דרך translate.google.com (כדי לבצע תרגום של דף שנמצא באתר שלך), תוכל לתשאל את google.com איזו כתובת IP עשתה זאת.

סיכום

גילוי גישה משרתי פרוקסי היא משימה חשובה שתעזור לך לשמור על גישה "נקיה" יותר לשרת שלך ע"י סינון גישה אנונימית (שלא באמצעות ספק האינטרנט הישיר) שעלולה לפגוע בשרת. מניעת הגישה אפשרית בעזרת הכנסת מסננים מתאימים בעזרת grep או snort.

למרות שאין דרך אחת קלה לזהות שימוש שוטף בשרתי פרוקסי אנונימיים - ניתן לזהות דפוס פעילות שאותו ניתן לנטר ולסנן.

בתור התחלה ניתן להשתמש ברשימות שחורות המכילות מידע מוקדם על שרתי פרוקסי שכבר "הוכיחו" את עצמם כבעייתיים, וכן ניתן ואף רצוי להוסיף לרשימה זו את שרתי הפרוקסי שהיו אצלך. רשימה זו מתעדכנת כל הזמן.

בעזרת שימוש ב-grep נוכל לזהות מקרים כגון שימוש ב-Base64 ו/או Rotate13. זיהוי כזה יבטיח לנו גילוי השימוש בשרת פרוקסי אנונימי.

דוגמאות רבות המופיעות במאמר זה נלקחו מהמאמר שנכתב על ידי John Brozycki ופורסם במקור בשנת 2008 בחדר הקריאה של Sans.org - שווה לקרוא!
http://www.sans.org/reading_room/whitepapers/detection/detecting-preventing-anonymous-proxy-usage_32943

פישינג והתחזות לאחר: מתי מותר להחזיק זהות פיקטיבית?

מאת: עו"ד יהונתן קלינגר

הקדמה, הבעיות בעבודה עם תחום אבטחת מידע

אנשים העובדים בתחום אבטחת המידע, לעיתים רבות מדי, מוצאים עצמם בשאלות משפטיות קשות במיוחד כמו האם מותר להם לבצע בדיקות אבטחה (pentest) של אתרים פלונים או האם מותר להם לפנות ולבדוק אבטחה על ידי שימוש במידע שהשיגו בדרך לא דרך. אלא, שבטרם עונים לשאלות הקשות באמת, יש שאלות קלות יותר שחוקרי אבטחה לא נוטים לבדוק והן משמעותיות יותר כיוון שהן מטילות אחריות פלילית על מי שעובר עליהן.

ההאקר בעבודתו עוסק בפעולות רבות, אך אנו לא נתייחס לכולן אלא רק נבדוק שני מקרי שטח רלוונטיים: הראשון הוא של הנדסה חברתית, כאשר איש האבטחה נשכר על מנת לדלות לפרטים או לבדוק את מידת האבטחה של ארגון מסוים. המקרה השני הוא המקרה בו איש האבטחה מייצר לעצמו זהויות פיקטיביות על מנת לדלות מידע מאחרים או לבדוק את האבטחה.

המקרה הראשון קשור לענייני חוקרים פרטיים; חוק חוקרים פרטיים ושירותי שמירה בישראל אוסר על אדם לבצע חקירות עבור אחר ללא רישוי; המקרה השני קשור להגדרות עבירת ההתחזות בחוק העונשין. במאמר זה אדון בהוראות החוק ובשאלה כיצד ניתן לבצע חקירות אבטחת מידע מבלי לעבור על הוראות החוק.

במאמר קצר זה אסקור את החקיקה על ביצוע חקירות עבור אחר ואת הבעיות בה, ואתמקד בנושאים הקשורים לאנשי אבטחת המידע. אבל: חשוב לזכור שהתחום רגיש במיוחד ושלא מומלץ לעבוד בנושא מבלי לקבל ייעוץ ספציפי. אני מציע שכל אדם שמגדיר עצמו "חוקר אבטחת מידע" ומספק שירותי חקירות לאחרים יעצור לרגע ויתייעץ עם עורך דין לפני שהוא ממשיך.

הבה נתחיל עם הוראות חוק חוקרים פרטיים ושירותי שמירה; החוק, בגדול, קובע כי חוקר פרטי הוא "מי שעוסק בהשגת ידיעות על הזולת או באיסופן, לצרכי אחרים ודרך שירות לכל, ושלא לצרכי מחקר מדעי, סקר דעת קהל או פרסום ברבים או לצורך מסירת ידיעות לבעל רישיון לפי חוק שירות נתוני אשראי, התשס"ב-2002"; כלומר, כל אדם שמטרתו איסוף ידיעות על הזולת לצרכי אחר הוא חוקר פרטי (בכפוף לחריגים). בתי המשפט לא ממש פרשו את ההגדרה ולא נכנסו לעומק השאלה, אבל ההגדרה שלה היא רחבה מדי ומסוכנת, ועשויה לחול על כל מיני מצבים שאנשי אבטחת מידע עוסקים בהם מיום ליום.

לצורך העניין, נקח לדוגמא מצב בו נמצא מחשב עם סוס טרויאני; בעל המחשב מעוניין לדעת מיהו האדם אשר פרץ למחשב והחדיר את אותו סוס טרויאני. לצורך כך הוא פונה לאיש אבטחת מידע (ולא לחוקר פרטי) אשר מבצע ניתוח של המידע והתקשורת מהמחשב החוצה. במצב כזה, הוא "משיג ידיעה על הזולת". עכשיו, ברור לנו שלא לכל חוקר פרטי יש את הידיעה או היכולת להשיג את המידע הזה כיוון שאין לו הכשרה פורנזית, ומנגד לאיש אבטחת המידע (בדרך כלל) אין רשיון חוקר פרטי (והיו מקרים בהם כבר הוגש כתב אישום נגד מי שאסף ידיעות על אחר בלי לקבל רשיון חוקר פרטי).

מנגד, כאשר אדם מגיע ומבקש לבדוק את מערכותיו שלו עצמו, אז לא מדובר, ככל הנראה, באיסוף ידיעות על הזולת, ולכן יכול אדם לומר שאין מדובר בחקירה פרטית. כלומר, Pentest, לכשעצמו, יכול שיהיה דווקא לגיטימי ולא לדרוש רישוי. אלא, שבמהלך אותה בדיקה, נבדקים כל מיני גורמים אנושיים בשרשרת: אותם גורמים הם "זולת" כהגדרתה בחוק ולכן חייבים לקבל גם התייחסות.

אוקי, אז הבנו את הבעיה של חקירות פרטיות, אבל מה קורה כאשר אנחנו צריכים לספק מידע עסקי? האם כאשר נשכר אדם כדי לזהות מי מחזיק אתר אינטרנט מעוול או לזהות גולש אנונימי על סמך מידע שהגולש השאיר ברקע, הוא מבצע פעולה אסורה? חוק הגנת הפרטיות אוסר על "בילוש או התחקות אחר אדם העשויות להטרידו"; איסוף של פרטים על אדם ללא פניה אליו אכן לא אמורה "להטרידו", אבל אם התוצאה של אותה פעולה היא תוצאה מטרידה, אז יכול להיות שהפעילות עצמה גם מטרידה. בעניין זה, ובצורה אחרת לגמרי, בית המשפט העליון פסק כי חוקרי משטרה צבאית ששפכו מי מלח לגרונו של אדם על מנת שיקיא סמים היא הטרדה אחרת (ד"נ 9/83 ועקנין נ' בית הדין הצבאי לערעורים).

שני חריגים מהותיים קיימים בחוק חוקרים פרטיים שיכולים לאפשר את עבודת חוקר אבטחת המידע: הראשונה היא מחקר אקדמי והשניה היא עבודה עבור עצמו. כלומר, לאדם מותר לבצע חקירות עבור עצמו (איסוף מידע על הזולת). כלומר: אם חדרו למחשב שלך או לאתר שלך, מותר לך לבצע עבור עצמך חקירה ולאסוף את המידע. זה לא אומר שמותר לחדור לחומרי מחשב או לפרוץ הגנות (וראו, לעניין זה, את סעיף 4 לחוק המחשבים), אלא שמותר לאסוף מידע על אחרים, לחקור ולשאול (שוב, חשוב להסתכל על המשך המאמר לגבי מה מותר ומה אסור לחקור).

הכיוון השני הוא מחקר מדעי. כאן, החוק מאפשר לך איסוף מידע על הזולת במסגרת מחקר מדעי. החוק עצמו לא מגדיר מהו מחקר מדעי; אבל סביר להניח שעבודת מחקר שהוזמנה על ידי אדם וכוללת מידע הנוגע לו אישית לא תכלל בהגדרה. לעומת זאת, עבודת מחקר שבודקת באלו מדינות שרתים רבים יותר נפרצים או מיהן הקבוצות המשמעותיות הפועלות בתחום החדירה למחשבים ישראלים. מחקר מדעי יכול גם לכלול מתודות כלליות יותר, כמו "מהן התוכנות בעלות פרצות האבטחה הרבות יותר", אך בכל מקרה לא שאלה כמו "מי העומדים מאחורי מתקפת הוירוסים מיום כך וכך נגד רשת המחשבים של פלוני".

כלומר, על מנת להכנס לחריג המחקר המדעי, ראוי שהמחקר יוגדר בצורה כללית שתאפשר את מכירתו לציבור הכללי, ולא כעבודה בהזמנה עבור אדם ספציפי.

התחזות

החוק אוסר על מספר מקרי התחזות, החל מהתחזות לאחר (סעיף 441 לחוק העונשין), דרך התחזות לבעלי מרות או סמכות כלשהיא (סעיף 283, התחזות לעובד ציבור, לדוגמא). ככלל, סעיף החוק עצמו הוא כזה: "המתייצג בכזב כאדם אחר, חי או מת, בכוונה להונות, דינו - מאסר שלוש שנים; התייצג כאדם הזכאי על פי צוואה או על פי דין לנכס פלוני והוא עושה זאת כדי להשיג את הנכס או את החזקתו, דינו - מאסר חמש שנים"; כלומר, יש שני תנאים לעבירת ההתחזות: הראשונה היא כי ההתחזות היא לאדם אחר, והשניה היא כי כוונת ההתחזות היא להונות. את יסוד הכוונה קשה יותר להוכיח, אך כלל האצבע צריך להיות כזה: אין בעיה לייצר לעצמך דמות פיקטיבית (אך לא כזור שפועלת בשליחותו של אחר), אבל אסור בתכלית האיסור להתחזות לאדם בשם שקיים או להתחזות לאדם העובד בגוף מסוים או ארגון עם סמכות כלשהיא.

בתי המשפט לא התלהבו, בלשון המעטה, ממקרים בהם חוקרים פרטיים יצרו סיפורי כיסוי שנועדו רק להלהיב את הנחקר כדי לגרום לו לשתף פעולה. לדוגמא, בתא (חד') 2293/08 מייס אלרים אבו שקרה בע"מ נ' יניר תקשורת בע"מ פסל בית המשפט דו"ח חקירה אשר נוצר לאחר שהחוקר הבטיח לנחקרת הזדמנות עיסקית לשיתוף פעולה בתחום הביגוד, ההנעלה ושמלות-כלה. כלומר, גם סיפור הכיסוי חייב להיות כזה אשר אינו מטעה פוגע. לגבי חוקרים פרטיים יש כללים ספציפיים (שכנראה לא חלים על חוקרי אבטחת מידע שאינם חוקרים פרטיים) אבל חשוב לזכרם: "חוקר פרטי לא יציג את עצמו בין במילים ובין בהתנהגות כשוטר, כפקח או כעובד ציבורי אחר כמשמעותו בחוק לתיקו דיני העונשין (עובדי הציבור) תשי"ז , חוקר פרטי לא יציג עצמו כבעל מקצוע הטעון רשיון על פי חוק, אלא אם היה לו אותה שעה רשיון כאמור. חוקר פרטי לא יתחזה כשליחו של אדם פלוני או כמי שפועל מטעמו" (תקנות חוקרים פרטיים ושירותי שמירה (אתיקה מקצועית)).

אלא, שהגדרת ההתחזות היא מעט חמקמקה. לאחרונה ניתן פסק דין בשאלת ההתחזות (תפ (ת"א) 3445/07 מדינת ישראל נ' רות לייטנר) בו זוכתה נאשמת מעבירת התחזות לאחר שהתחזתה לרופאת שיניים. בית המשפט זיכה את הנאשמת מעבירת התחזות לאחר ופסק כי "נראה כי לשם הרשעה בעבירה של התחזות לאחר, נדרש כי ההתחזות תהא לאדם קיים"; כלומר, לשיטת בית המשפט אין בעיה להתחזות לאדם שאינו קיים (כל עוד הוא אינו בר-סמכות בדין) ולייצר לעצמך זהויות רבות (וראו גם עפ 71377/06 טלי יחזקאל נ' מדינת ישראל).

אז מה אפשר לומר שמותר מבחינת הנדסה חברתית? אם נצא מנקודת הנחה שלחוקר אבטחת המידע אין רשיון חוקר פרטי אך הוא מנסה, לדוגמא, לחקור מי חדר למחשב שלו, אז מותר לו לאסוף את המידע האישי ועל סמך זה לבנות סיפור כיסוי אמין שלא יכיל מידע על גורמים קיימים (כלומר, הוא לא חבר בקבוצה קיימת או עובד עבור אדם קיים) אך יהיה מותר לו להשתמש בשם בדוי. הדבר מגביל ביותר, אך זהו החוק. אסור לאדם להציג עצמו כאילו הוא עובד בחברה קיימת אם אינו כזה, אסור לו לומר כי הוא חלק מארגון האקרים עולמי או אסור לו לומר כי הוא עובד בשליחות של אדם מסוים שמעוניין לקנות זהויות.

יש לא מעט סכנות בעבודה בתחום אבטחת מידע מההיבט המשפטי. חוק המחשבים אינו חוק קל לאדם שעוסק במחשבים. לדוגמא, סעיף 3 לחוק אוסר על אחסנה של מידע כוזב; מידע כוזב הוא "מידע או פלט שיש בהם כדי להטעות, בהתאם למטרות השימוש בהם". כלומר, די שתזינו בטופס הרשמה לאתר כלשהוא פרט שאינו נכון כדי שלשון החוק היבשה תחייב אתכם בעבירה פלילית (לדוגמא, בת 8852/98 דניאל כהן נ' מדינת ישראל נמחק כתב האישום בעבירות אלה מחמק אי הבנת לשון החוק).

מנגד, בתי המשפט פסקו כי עבירת ה"זיוף" בנוגע לחומרי מחשב לא תתקיים כאשר המידע שמאוחסן מזויף (עפ 148-09-10 פפילומוב נ' מדינת ישראל); באותו המקרה הואשם אדם על כך שהחזיק במחשב שלו מידע שמאפשר זיוף של מסמכים רשמיים. בית המשפט זיכה אותו מעבירת הזיוף ופסק כי "אי-אפשר לצאת מהמדינה ע"י הצגת "דרכון במחשב" למשטרת הגבולות, כשם שאי-אפשר להציג לשוטר תנועה מחשב נייד ולומר כי "הרשיון בפנים". דומה הדבר לכך, שלא ניתן לשלם לפלוני ע"י משלוח שיק לפקודתו בפקסימיליה או בדואר אלקטרוני".

כשזיכה בית המשפט את פפילומוב מעבירת הזיוף, זיכה אותו גם מאחסנת מידע כוזב, באמרו כי "כתב האישום מחזיק אמנם ארבעה אישומים, וכן צורפו תיקי משטרה בעבירות של החזקת נכס החשוד כגנוב - אך עיקר החומרה היה באישומים הראשון והרביעי, קרי בעבירות הזיוף, כאשר מבחינה כמותית העיקר הוא באישום הראשון".

כלומר, זיוף של תמונה באמצעות פוטושופ, לדוגמא, כדי להטעות אחרים, יכולה לחייב אתכם בעבירה על פי חוק המחשבים אם אתם טוענים שתמונה זו היא תמונתכם, אבל היא לא יכולה להיות זיוף על פי חוק העונשין או התחזות.

כרגע אתם אומרים לעצמכם: חבל שהתחלתי לקרוא את המאמר, הרי אם לא הייתי קורא אולי הייתי יכול להמשיך לגלוש באינטרנט בבטחה, לעבוד, להתפרנס ולחיות בבורות משוועת מכך שאני (ועשרות מחברי הטובים) עוד עוברים על החוק. אכן, החוק היבש אינו מטיב עם אנשי אבטחת מידע. אבל, הפסיקה דווקא הולכת לא מעט לכיוונם; החל מתיק בו נקבע כי ביצוע port scanning לאתר המוסד אינו בדיוק עבירה על החוק (עפ 8333/04 מדינת ישראל נ' אבי מזרחי): "המשיב רק עשה פעולה הנקראת port scanning, דהיינו בחן אילו פורטים פתוחים ואילו פורטים סגורים. המדובר הוא בתוכנה ראשונית בלבד, הבודקת כאמור, אילו פורטים פתוחים והאם האתר מאובטח. הא ותו לא. בנוסף, לא הוכח כי המשיב ניסה לפרוץ תוכנה זו או אחרת. לאור האמור, קבע בית המשפט קמא ש**לא הוכח יסוד נפשי של החפץ לפרוץ לאתר,**

זולת הרצון לבחון האם האתר מאובטח, ובקביעה זו אין מקום להתערב. על סמך נתונים אלה, קבע בית המשפט כי לא הייתה למשיב מחשבה פלילית - חפץ להשיג את התוצאה של פריצה למחשב" ועד קביעה כי בדיקת אבטחה לאתר בתי המשפט כדי להוכיח קיומה של פרצת אבטחה שמאפשרת איפוס קנסות אינה פריצה (פ 9497/08 מדינת ישראל נ' משה הלוי).

בתי המשפט עשו משהו שרשויות אחרות בישראל נוטות לא לעשות, וזה להפעיל את שיקול הדעת של השופט. אבל, החוק היבש חייב להשתנות והוא רע לכל מי שעוסק בתחום אבטחת המידע.

הפתרונות: רשיון חוקר פרטי

יש פתרון אחד שחייב להשקל על ידי מי מכם שעוסקים באבטחת מידע לפרנסה, והוא הצטרפות לקהילת החוקרים הפרטיים בישראל. החוקרים הפרטיים כיום נמצאים בבעיה בכל הנוגע לטכנולוגיה: הם היחידים המורשים לחקור עבור הזולת (למעט המשטרה) גם בנושאים טכנולוגיים, והם זקוקים נואשות לאנשים העוסקים בתחום ומבינים בו. אכן, מדובר בהכשרה ארוכה ומייגעת, שבסופה מבחנים, אבל מבחנים אלה יכולים לספק לכם בסופו של דבר את ההכשרה הרלוונטית והדרושה לצורך עבודה כחוק.

כן, חוק חוקרים פרטיים יאפשר לכם לערוך חקירות עבור אחר אבל לא ממש יתן לכם היתר לפרוץ למחשבים של זרים בשמו או להתקין מערכות מעקב מתוחכמות; החוק וההכשרה נועדו על מנת לתת לכם את הכלים להבדיל בין המותר והאסור: הכלים שאמורים לגרום לכך שתעבדו לפי החוק. לכן, כה חשוב שתקבלו את ההכשרה.

ההכשרה גם תספק לכם את המגע עם העולם הממשי, זה שעוסק בחקירות ממשיות ואת הצרכים של השוק. לעיתים רבות חוקרים פרטיים (בעיקר בתחום הגירושין) נדרשים לחקירות פורנזיות מסובכות (לדוגמא: האם בעלי קורא לי את המייל?) ואינם יודעים מה לעשות. כאן אתם תוכלו לסייע; אבל חשוב שתוכלו לסייע כחוק.

הרקע המתימטי של RSA, או: איך הצפנת RSA עובדת?

מאת: בעז (tsabar)

הקדמה

הצפנת RSA תופסת מקום מכובד ביותר בחיינו במאה ה-21. אם ננסה לדמיין את עולמנו כיום בלי היכולת של הצפנה ע"י מפתח ציבורי, כנראה שהעולם היה אחר בתכלית. כל נקודה פיזית על הקו, כל שרת או נתב בדרך, היה יכול להפוך מאוד בקלות לנקודת האזנה לכל פרט מידע אשר יוצא ונכנס בשער. בין אם זה אימיילים, סיסמאות, מצב חשבון הבנק או מספר כרטיס האשראי ששלחנו בטופס המקוון ל-Ebay.

סקירה היסטורית קצרה

הצפנה סימטרית היא הצפנה מאוד קלה: שני הצדדים נפגשים מעל תווך מוצפן (בד"כ פגישה פיזית), ומתאמים מפתח ואלגוריתם הצפנה. הבעיה ה"רקורסיבית" היא שבשביל ליצור תווך מוצפן, צריך תווך מוצפן. הצפנות סימטריות קיימות כבר אלפי שנים, בין אם באלגוריתמים פשוטים כמו של יוליוס קיסר (צופן ההזזה), או הצפנת ויז'נר שנפוצה במאה ה-16. המחשבים ויכולת החישוב האלקטרוני הקפיצו לגבהים חדשים את המודעות להצפנה חזקה, "אמיתית" מבחינה מתימטית, ולא כזו שהסוד שלה הוא צורת ההצפנה (האלגוריתם עצמו).

עד לתחילת שנות השבעים היה עוד אגוז קשה לפיצוח: כל ההצפנות, גם הטובות שבהם, היו סימטריות. בזמנו כבר היה מדובר באלגוריתמים קצת יותר מתוחכמים מהצפנים שפוענחו ידנית, אבל המחשבה על כך שצריך תווך מוצפן (או שליח שניתן לסמוך עליו ב-100%) היא מחשבה מטרידה בהתחשב בעובדה שלא תמיד יכולים שני הצדדים לתאם מפתח מוסכם מראש, כזה שאף אחד אחר לא ידע מהו.

הצפנה א-סימטרית, כזו שהמפתח לפענוח שונה ממפתח ההצפנה, היא לא דבר מובן מאליו. לפני שאנחנו מדברים על הצפנה א-סימטרית, צריך להוכיח שזה בכלל אפשרי - שניתן לעשות תרגילי לוליינות מתימטית על מערכת מספרים ומשוואות, כך שניתן להצפין בצורה אחת ולפענח בצורה אחרת, ושאי

אפשר לפענח בעזרת מידע המופק מתוך המפתח שאיתו מצפינים. את ההוכחה שאפשר ליצור תווך מוצפן בפרוטוקול שעובר כולו בתווך בלתי מוצפן (במקרה של RSA זה נוצר ע"י מפתח פומבי להצפנת הודעות ומפתח פרטי לפענוח, ושליחת המפתח הפומבי בלבד), נתנו החוקרים ויטפילד דיפי (Whitfield Diffie) ומרטין הלמן (Martin Hellman) במאמרם המהפכני "New Directions in Cryptography" משנת 1976. במאמר זה הם הוכיחו שניתן ליצור מפתח פרטי להצפנה סימטרית מעל תווך לא מוצפן, כך שמי שמאזין להודעות העוברות ברשת לא יוכל לגלות את המפתח. המאמר לא דיבר על RSA, ואני לא אדבר על המאמר.

בסיס מתימטי: הצפנה למתחילים

למען הדוגמא ניקח את פונקצית ההעלאה בריבוע:

$$f(x) = x^2$$

אם זוהי ההצפנה שלי, הרי שפונקצית הפענוח תהיה הוצאת שורש ריבועי:

$$f^{-1}(x) = \sqrt{x}$$

זוהי דוגמא פשוטה ומצוינת ל"הצפנה" שבה דרך הפענוח שונה מההצפנה. זו כמובן לא באמת הצפנה, ולא רק משום שאפשר בקלי-קלות "לפענח" אותה, אלא שגם אין לה מפתח הצפנה (לחילופין, ניתן לומר שמרחב המפתחות מכיל מפתח יחיד).

באלגוריתמים של הצפנה, מקובל לכתוב את פונקצית ההצפנה ב-E (קיצור של Encrypt) ואת פונקציית הפענוח ב-D (קיצור של Decrypt). בנוסף, הודעות להצפנה מסמנים בד"כ ב-m (קיצור של message), והודעות מוצפנות מסמנים ב-c (קיצור של cipher), וכך זה יסומן בהמשך. על פי הדוגמא הקודמת, נסמן את פונקצית ה"הצפנה" שלנו כך:

$$E(m) = m^2 = c$$

ואת פונקצית הפענוח המתאימה:

$$D(c) = \sqrt{c} = m$$

כמו שבוודאי שמתם לב, הפונקציה D היא ההופכית של E, ולא בכדי. התנאי הראשון והבסיסי שצריך להתקיים, הוא שפענוח הודעה מוצפנת יהיה זהה להודעה המקורית לפני ההצפנה. בסימנים:

$$D(E(m)) = m$$

אם התנאי הזה לא מתקיים, אי אפשר לפענח הודעה מוצפנת.

> לפעמים לא צריך לפענח הודעה מוצפנת. לדוגמא: שמירת סיסמאות של משתמשים על שרת נעשית עם פונקצית hash, שדומה במאפייניה לפונקצית הצפנה שאין לה פונקצית פענוח מתאימה. בכל אימות סיסמא, מצפינים את הסיסמא שנשלחה ומשווים את הערך המוצפן לערך השמור בשרת. ככה גם אם פורצים לשרת, אי אפשר לגנוב סיסמאות.

על הודעות מוצפנות ישנן הגבלות כאלו ואחרות מפאת הפרקטיקה היישומית של אמצעי המיחשוב העומדים לרשותינו: פרוטוקול התקשורת, כמות הביטים במנת נתונים, כמות הביטים של המפתחות, יכולת חישובית מסוימת וכו'. לא תמיד אנחנו יכולים להשאיר את הנתונים האלו "לגדול כרצוננו". לדוגמא: אם ההצפנה שלי היא העלאה בריבוע, ואני מוגבל ב-4 ספרות, אז לא תמיד אני באמת יכול להעלות את המספר "הגולמי" בריבוע, ואני נאלץ לקצר אותו לרוחב של 4 ספרות בשיטת המודולו. לדוגמא:

$$E(12) = 144$$
$$E(200) = 200^2 = 40000 \equiv 0(mod 10000)$$

ננתח את מה שקיבלנו: ההצפנה של הערך 12 עובדת מצוין ושווה ל-144. ההצפנה של הערך 200 תהיה שווה ל-0. ננסה כעת לפענח את ההצפנה:

$$D(0) = \sqrt{0} = 0$$

אופס...

למרות שפונקצית הפענוח היא הופכית לפונקצית ההצפנה, קיבלנו שהפענוח, בגלל אילוצים מסוימים, לא נותן תוצאה נכונה. זה מוביל אותנו לתנאי החשוב הבא בתורת ההצפנה: פונקצית ההצפנה צריכה להיות חד-חד-ערכית. בדוגמא לעיל, הבעיה בפענוח נוצרה כי:

$$E(200) = E(0) = 0$$

זה הזמן לעבור לשלב מתקדם יותר - השלב ה"בעייתי" מבחינה מתימטית - חיפוש אחר פונקציה שתענה לדרישות מאוד מסוימות, שפונקצית הצפנה במפתח פומבי צריכה לענות עליהן:

- שלא קשה לחשב את ערכה (כי אנחנו לא רוצים שזה יקח לנו חודשים של חישוב),

- שתפעל נכון על הודעות בתחום נתון כלשהו (לדוגמא: תחום רצוי של מספרים שלמים אי-שליליים).
- שתהיה חד-חד-ערכית (ולכן יש לה פונקציה הפיכה, או פונקצית פענוח),
- שיהיה קשה מאוד לחשב את הערך המקורי לפני שעבר בפונקציה (לפי הדוגמא לעיל: בהינתן מספר כלשהו, נניח 9, יהיה קשה לחשב את השורש הריבועי שלו),
- ובנוסף להכל, עם מעט מידע נוסף על המספר (על המספר עצמו!!! לא על הפונקציה), ניתן בקלות לחשב את ערכו לפני שעבר דרך הפונקציה (בהמשך לדוגמא שלנו: בהינתן ש-9 הוא כפולה של 3, יהיה ניתן בקלות לחשב את השורש הריבועי שלו).

פונקציה כזו נקראת "פונקציה חד כיוונית עם דלת סתרים". חד כיוונית - כי חישוב ההופכי הוא משימה מורכבת ביותר (פונקציות hash הן במובן מסוים פונקציות חד כיווניות, למרות שאינן חד-חד-ערכיות וכלל אין להן פונקציה הופכית). דלת סתרים - אפשרות למידע נוסף, שבזכותו ניתן לחשב בקלות את הכיוון ההופכי של הפונקציה.

מידע מוצפן מעל מספרים ראשוניים

מהו פירוק לגורמים כולנו יודעים מכיתה ב': בהינתן מספר n, צריך למצוא את הגורמים הראשוניים שלו. ע"פ המשפט היסודי של האריתמטיקה, יש רק מכפלה אחת של מספרים ראשוניים שתתן את n כתוצאה. נרצה גם איכשהו לשלב "עליהם" מידע להצפנה. בשלב זה אנחנו אמורים להתחשב בתנאים המקדימים לכל הצפנה: פענוח נכון וחד-חד-ערכיות. הכפלה של שני מספרים ראשוניים גדולים מאוד היא במובן מסוים פונקציה חד כיוונית, ודלת הסתרים האפשרית כאן היא, באופן טבעי, אחד הגורמים של המכפלה (ראוי לציין שדלת הסתרים של RSA היא דווקא ההעלאה בחזקה מודולרית ופונקצית אוילר שמחושבת בקלות בעזרת גורמי המכפלה. תודה לגדי אלכסנדרוביץ' על התיקון).

ובכן - רונלד ריבסט (Ronald Rivest), עדי שמיר (Adi Shamir) ולאונרד אדלמן (Leonard Adelman) עשו זאת, והצליחו להלביש מידע להצפנה בצורה מחוכמת על בעית הפירוק לגורמים. הם הנציחו את שמותיהם בשם האלגוריתם - RSA - ששמו מורכב מהאותיות הראשונות של שמות המשפחה שלהם.באלגוריתם ההצפנה משתמשים בחזקות שמחושבים מתוך המספרים הראשוניים, ומודולו n, כאשר n הוא מכפלת שני מספרים ראשוניים, וממנו קשה לדעת מהם אותם מספרים ראשוניים.

המתמטיקה

האלגוריתם מתבסס על האבחנה שאם n הוא מספר כלשהו, אז עבור מספר m זר ל-n וקטן ממנו (כזה שהמחלק המשותף המקסימלי שלו עם n הוא 1- $GCD(n,m)=1$), מתקיים $m^{\varphi(n)} \equiv 1(modn)$ [1]. בעברית: m בחזקת $\varphi(n)$ (פונקצית אוילר על n), לחלק ב-n תניב שארית 1. פונקצית אוילר של n היא כמות המספרים הזרים ל-n וקטנים ממנו.

כעת, אנחנו רוצים לחשב את פונקצית אוילר על מספרים מסוימים. פנקצית אוילר אינה קשה לחישוב אם יודעים מהם הגורמים הראשוניים של המספר עליו אנו רוצים לחשב את הפונקציה. דוגמא אחת היא מספר ראשוני: מכיוון שלכל מספר p ראשוני, כל מספר קטן ממנו (ואינו 1) הוא זר לו, תוצאת פונקצית אוילר היא p-1. דוגמא נוספת היא מכפלת שני מספרים ראשוניים: אם n הוא מכפלת שני מספרים ראשוניים שונים זה מזה, p ו-q, אז ישנם q-1 מספרים קטנים ממנו שמתחלקים ב-p, וישנם p-1 מספרים קטנים ממנו שמתחלקים ב-q.

דוגמא מספרית להמחשה:

$$n = 15, p = 3, q = 5$$

נבחן מהם המספרים **שאינם** זרים ל- 15: 3,6,9,12,5,10. יש כאן 5-1=4 מספרים שמתחלקים ב-3, ו-3-1=2 מספרים שמתחלקים ב-5.

נפתח את הנוסחא: [2]

$$\varphi(n) = (n-1) - (p-1) - (q-1) = n - 1 - p + 1 - q + 1 = \\ = n - p - q + 1 = pq - p - q + 1 = (p-1)(q-1)$$

אסביר מה הצבתי בהתחלת המשוואה:

- n-1 - כמות המספרים הקטנים מ-n (המועמדים הפוטנציאליים).
- p-1 - כמות המספרים הקטנים מ-n שהם כפולה של q (ולכן לא זרים ל-n).
- q-1 - כמות המספרים הקטנים מ-n שהם כפולה של p (ולכן לא זרים ל-n).

בסה"כ, הנוסחא הזו היא מקרה פרטי של נוסחת אוילר למציאת ערך הפונקציה.

מ-[1] ומ-[2], אפשר להסיק שעבור n שהוא מכפלה של 2 מספרים ראשוניים, p ו-q, ועבור m שאינו p או q (ולכן זר ל-n. זה אפשרי: פשוט בוחרים p ו-q שונים מ-m), מתקיים:

$$m^{\varphi(n)} \equiv 1(modn)$$

ואם נכפול את שני צידי המשוואה ב-m, נקבל כי:

$$m^{\varphi(n)+1} \equiv m(modn)$$

וזה תקף לכל מכפלה של $\varphi(n) = (p-1)(q-1)$, כי:

$$m^{\varphi(n)} \equiv 1(modn) \Rightarrow (m^{\varphi(n)})^x = m^{\varphi(n)\cdot x} \equiv 1^x(modn) \equiv 1(modn) \quad [3]$$

במילים אחרות, אפשר לרשום זאת כך: יש לנו מספר כלשהו, $\varphi(n)$, שאם מעלים את m בחזקה שהיא כפולה כלשהי שלו פלוס 1, ולוקחים את שארית החלוקה ב-n, מקבלים את m עצמו. זה אכן מקיים את דרישת החד-חד-ערכיות שקבענו בתור תנאי התחלתי: אם m זה המסר שאנו רוצים להצפין, זה מבטיח שיש לנו איך לפענח אותו; אין שתי הודעות שונות שלאחר הצפנה ופענוח יתנו את אותו ערך.

הואיל ואנו רוצים להגיע למצב של העלאה בחזקה שהיא מכפלה (כלשהי) של $\varphi(n)$ ועוד 1, אנחנו צריכים לחפש פתרון לנוסחא:

$$x \cdot y = \varphi(n) \cdot z + 1 \quad [4]$$

נשמע קשה? ובכן - לא ממש. אוקלידס היווני כתב לזה אלגוריתם לפני 2,300 שנים, ובגירסה המורחבת של האלגוריתם למציאת מחלק משותף מקסימלי, GCD, שעליו התבססנו מקודם בשביל לוודא ששני מספרים הם זרים (המחלק המשותף המקסימלי שלהם הוא 1), נוכל למצוא בקלות ערכי y ו-z שיפתרו את המשוואה עבור ערך x נתון.

אלגוריתם אוקלידס המורחב מוצא לא רק את המחלק המשותף המקסימלי של שני מספרים a ו-b, אלא גם את הערכים שבהכפלתם ב-a ו-b יתנו את המחלק הזה. במילים אחרות, הוא מוצא גם את הערכים c ו-d במשוואה:

$$a \cdot c + b \cdot d = GCD(a, b)$$

אז קודם כל נגריל מספר כלשהו x, ונוודא שהוא זר ל-$\varphi(n) = (p-1)(q-1)$, כלומר שמתקיים $GCD(x, \varphi(n))$. אחרי שינוי קל למשוואה [4] נקבל:

$$x \cdot y - \varphi(n) \cdot z = 1$$

כעת נמשיך ונמצא בעזרת אלגוריתם אוקלידס המורחב את שאר המספרים החסרים לנו: y ו-z (ניתן להחליף ב- z-, זה לא משנה את הערך המספרי, רק את הסימן).

בשלב זה הכל מוכן: המפתח הפומבי יהיה n ו-x, ואילו המפתח הפרטי יהיה p, q ו-y. שווה לשים לב: מעכשיו אין עוד צורך בגורמים הראשוניים p ו-q בשביל הצפנה או פענוח של הודעות. המספרים האלו היו מספרי עזר לתשתית - יצירת המפתחות, אבל לא משמשים להצפנה עצמה.

ההצפנה תהיה:

$$E(m) = m^x (mod\ n)$$

הפענוח יהיה בדומה:

$$D(c) = c^y (mod\ n)$$

כעת, בהתבסס על משוואה [4], והצבתה אל תוך משוואה [3], נקבל את המשוואה שמסבירה למה פענוח של הודעה מוצפנת בשיטת RSA אכן יניב את ההודעה המקורית:

$$D(c) = D(E(m)) = D(m^x) = (m^x)^y = m^{x \cdot y} = m^{\varphi(n) \cdot z + 1} \equiv m^1 (mod n) = m$$

עד עכשיו ראינו למה זה נכון מתימטית - כלומר, למה פענוח של הודעה מוצפנת אכן מניב את ההודעה המקורית, ולמה אין שתי הודעות שונות שהפענוח שלהן יהיה זהה (תחת אותו מפתח הצפנה).

כעת נעבור לחלק האחרון, והוא החלק המעניין של הקשר המתימטי בין פירוק לגורמים לשבירת צופן RSA.

פירוק לגורמים

לפני שבכלל אמשיך, חשוב להזכיר כאן את השאלה הידועה והנושנה, שיתכן שמתקרבת לידי פתרון: האם P=NP? הפתרון המסתמן הוא "לא, הם שונים זה מזה". אם הם היו שווים, הסיפור היה נגמר כאן, והכל היה מתמוטט. אבל... זה בכלל לא חשוב לענייננו, למרות הסיכוי שאולי לא נמצא לעולם אלגוריתם יעיל לפירוק לגורמים (פולינומיאלי בכמות הביטים למפתח). אלגוריתם RSA לא הוכח כקשה כמו בעית הפירוק לגורמים, אלא זו השערה בלבד. ייתכן - והסיכוי הוא לא מאוד גדול - שבעית הפירוק לגורמים היא לא פולינומיאלית ופענוח RSA הוא כן (בעית הפירוק לגורמים היא בעיה ב-NP, וככל הנראה אינה ב--NP

Complete. לפיכך, אם P≠NP, ייתכן שהיא ב-P וייתכן שלא). אסור לשכוח שאולי קיימת שיטה אחרת (שעוד לא מצאנו) לשבור את צופן RSA בלי שימוש בגורמים הראשוניים p ו-q.

ועכשיו נחזור לענייננו: למה צופן RSA נחשב לצופן בטוח. הנחת העבודה היא, כמובן, שבשביל לשבור את צופן RSA צריך לפרק את n לגורמיו הראשוניים, ושהפירוק הזה הוא סיפור קשה.

נסתכל על המפתח הפומבי ונראה מה אפשר ללמוד ממנו: המפתח הפומבי הוא n ו-x. נשים לב ש-x הוא מספר שבחרנו בצורה רנדומלית, והקשר (העקיף) שלו ל-n הוא דרך הכפלתו ב-y. אבל - y הוא חלק מהמפתח הפרטי, ואין לנו אותו. לכן x לא מסגיר איתו שום מידע על איך לפענח.

אם אנחנו מצליחים לפרק את n לגורמיו - p ו-q, אז נוכל למצוא בקלות את $\varphi(n) = (p-1)(q-1)$, להציב בנוסחא של אוקלידס, למצוא את y, ולפענח את ההודעה המוצפנת. אבל אם אין לנו דרך לפרק את n לגורמיו (ושוב - נניח שזו הדרך היחידה לפענח את ההודעה המוצפנת), אז לא נוכל למצוא את$\varphi(n)$, ולכן לא נוכל למצוא (בזמן סביר) את המספר y, שהוא תוצאת חישוב פשוט, אם אנחנו יודעים את x ואת $\varphi(n)$. כאן, $\varphi(n)$ משמש גם כדלת הסתרים של הפונקציה.

סיכום

במאמר זה הסברתי על קצה המזלג את הפן המתימטי של RSA, והשתדלתי ככל האפשר להשאיר את הנוסחאות בחוץ, כדי לא להבריח קוראים פוטנציאליים (למי שנזכר עכשיו בסטיבן הוקינג: כן...).

המאמר תיאר את הדרישות הבסיסיות מכל הצפנה, ובפרט מהצפנה א-סימטרית. בנוסף, המאמר תמצת את הנכונות המתימטית שמאחורי RSA, מהם הרעיונות המתימטיים העומדים בבסיסה של הצפנה זו, ואיך בתוך כל הבלגן הזה, אלגוריתם בן 2,300 שנים חובר לפונקציה מהמאה ה-18 כדי לשמש תשתית להצפנה הפופולרית של תחילת המאה ה-21.

המאמר נכתב בעזרת הספר "Cryptography - Theory and Practice" של דאגלס ר. סטיבנסון.

לקריאה נוספת (ויקיפדיה)

- RSA
- פונקצית אוילר
- מחלק משותף מקסימלי
- פונקציה חד כיוונית
- פירוק לגורמים
- שאלת P=NP

על המחבר

בעז (tsabar) סיים לאחרונה בהצטיינות תואר ראשון במדעי המחשב באוניברסיטה הפתוחה. הוא נכנס לבלוגוספירה די במקרה, וכותב בלוג כבר כמעט 3 שנים. בבלוג שלו, "צבר - בלוג עם קוצים", הוא כותב על נושאים שונים ומגוונים.

שנה טובה.

DNS Cache Snooping

מאת: עוז אליסיאן

הקדמה

DNS Cache snooping מתאר מצב שבו שרת DNS מתושאל על ידי גורם מסויים (בדרך כלל זדוני) בכדי "לחטט" בו, ולדעת האם השרת מחזיק ברשומה מסויימת בתוך המטמון (Cache). על ידי כך, אותו גורם יוכל להסיק האם בעל השרת או המשתמשים בו ביקרו באותו אתר (אשר מאוחסן כרשומה) ואף להסיק מתי.

במאמר זה נתאר איך נושא זה מתבצע. ידע מומלץ קודם הוא הכרות עם נושא ה-DNS - השרת והפרוטוקול. מאמר נוסף שמומלץ לקריאה לפני מאמר זה הוא המאמר המעולה של אפיק קסטיאל (cp77fk4r): DNS Cache Poisoning המסביר, בין היתר, באופן תיאורתי על הפרוטוקול.

איך עובד DNS Cache snooping

מנגנון ה-Cache בשרתי DNS

מנגון ה-Cache אשר קיים ברב שרתי ה-DNS הוא דבר יחסית סטנדרטי: אם נעשתה בקשה לגבי רשומה מסויימת, ושרת ה-DNS בסופו של דבר ענה לאותו גורם אשר ביצע את השאילתה (על ידי קבלת הרשומה משרת DNS הממוקם מעליו), השרת ישמור את הרשומה במנגון ה-Cache שלו, זאת בכדי שהוא לא יצטרך שוב לשאול את השרת שמעליו לגבי אותה רשומה, ויוכל לשרת את אותו גורם או גורם נוסף במהירות רבה יותר ובפחות עבודה אם יבקש שוב את אותו המידע.

שדה שחשוב להכיר הוא **שדה ה-TTL (בקיצור: Time To Live)** בשרת DNS - (חשוב לא להתבלבל עם TTL של חבילות IP). TTL עבור DNS, נועד בשביל לדעת כמה זמן יש להחזיק רשומה מסויימת במנגון ה-Cache. כאשר כל שרת DNS, האחראי על "Authoritative Nameserver", מחזיר תשובה, הוא יכול להגדיר TTL שונה אשר ינתן יחד עם התשובה לאותה בקשה. ה-TTL הנפוץ ביותר הוא בדרך

כלל 86,400 שניות, שהן 24 שעות. חשוב לציין בנוסף שגם המחשב עצמו של המשתמש הסופי מחזיק לרוב ברשומה בכדי למנוע מעצמו אפילו לתשאל את שרת ה- DNS מלכתחילה ולהקל עליו.

תשאול שרתי DNS

לנו, כמשתמשים של שרת DNS, עומדות בפנינו 2 סוגי שאילתות:

- Recursive Query
- Iterative Query

Recursive Query - או בעברית, שאילתה רקורסיבית. נתמכת על ידי מרבית שרתי ה-DNS. ובעצם מהווה את ההתנהגות הרגילה, הסטנדרטית והמוכרת לנו. לדוגמה, נניח כי משתמש מסויים מעוניין לגלוש באתר www.digitalwhisper.co.il. המחשב מתשאל את שרת ה-DNS שלו בשאילתה רקורסיבית מהו ה-A Record של האתר www.digitalwhisper.co.il.

A Record מהווה את התרגום מ-"Host Name" ל-IPv4. שרת DNS אמור להחזיר את כתובת ה-IP של האתר המבוקש בכדי שהמחשב יוכל לתקשר עימו.

פה בעצם נכנס עניין השאילתה הרקורסיבית: אם שרת ה-DNS, מחזיק כבר ברשומה (עקב שמירתה ב-Cache - משאילתה שנעשתה לפני כן) הוא יענה ישר "הנה קח את כתובת ה-IP של האתר www.digitalwhisper.co.il".

אם שרת ה-DNS אינו מחזיק ברשומה, (עקב זה ששאילתה לגבי האתר לא נעשתה לפני כן, או שזמן ה-TTL נגמר וכתוצאה מכך הרשומה נמחקה), והוא תומך בשאילתה רקורסיבית, יתחיל לבצע את התהליך הרקורסיבי הבא:

הוא יפנה אל שרת ה-DNS אשר מוגדר לו לקבלת המידע המבוקש. אם השרת הוא שרת DNS פנימי, אז ייתכן מאוד שברירת המחדל תהיה שרת ה-DNS של הספקית (ISP). אם זה שרת ציבורי, אז הוא ישאל את שרת ה-DNS המוגדר לו או אולי אפילו יתשאל את ה-Root Name Servers. אשר נמצאים ברמה הגבוהה בהיררכיה של שרתי ה-DNS ואמורים לספק את התשובות של שרתי "מפתח" (למשל מיהו שרת ה-DNS אשר מחזיק במידע לגבי הדומיין הישראלי co.il) ומשם הלאה.

אם השרת ה-DNS המתושאל אינו מחזיק ברשומה, הוא ילך וישאל את השרת ה-DNS המוגדר לו באותו האופן, יקבל את התשובה, ויחזיר את התשובה אל המחשב\השרת הקודת אשר ביצע את השאילתה.

Iterative Query - שאילתה איטרטיבית זהו סוג שאילתה המגדיר את הביט של הדגל הרקורסיבי כ-0.

נחזור שוב לדוגמא: נניח כי משתמש מסויים מעוניין לגלוש ל-DigitalWhisper, הוא שואל את שרת ה-DNS שלו, לגבי האתר www.digitalwhisper.co.il, והוא אינו מחזיק ברשומה במנגון ה-Cache. הוא יענה תשובה פשוטה: "אינני יודע, הנה שרת או כמה שרתים שמורשים/יכולים לענות לך על כך" כאשר בדרך כלל יצביע על השרת אשר מחזיק במידע לגבי הדומיין co.il.

היכן הבעיה?

כשנראה את הבעיה נוכל לתת דוגמא איפה הבעיה הזאת יכולה לשמש אותנו כנגד שרתי **DNS ציבוריים**, אבל הדבר המעניין באמת הוא שרתי ה-**DNS פנימיים בארגון**.

במקרים רבים, שרתים פרטיים (פנים-ארגוניים), אשר אמורים לשרת רק את משתמשי הארגון ולעזור בתשובות מהירות במקום שמחשבים פנים ארגוניים ילכו וישאלו שרתים מרוחקים, **עונים גם הם לשאילתות אשר מגיעות/מקורן ממחשבים חיצוניים לארגון.**

שרתים פנים-ארגוניים לא נועדו בכדי לענות לגורמיים חיצוניים. מטרתם אינה להוות שרתי DNS ציבוריים, אלא לשרת רק את הארגון עצמו ולתת לו את המהירות המקסימלית. המון מוצרי "הכל-כלול" (מוצר הכולל בתוכו IPS/NAT/AV/IDS וכו') מכילים שירות DNS, ובמקום לשרת רק את המחשבים הפניימים גם משרתים גורמים חיצוניים בלי ידיעה על כך – וכך, על-ידי תשאול אותם שרתים, גורמים חיצוניים לארגון, יוכלו להסיק האם בעל השרת או משתמשים בו, ביקרו באותו ואפילו להסיק מתי. מידע זה יכול להוביל לגילוי מידע וסטטיסטיקה לגבי יצרן / בנק / ספק אינטרנט / שותפים עסקיים ואפילו דברים מביכים או פרטיים.

הדגמה

בכדי לבצע את המתקפה הנ"ל ניתן להשתמש במספר רב של כלים, ישנם אפילו שירותי אינטרנט המאפשרים לתשאל דרכם את היעדים שלנו, במאמר הנ"ל, אשתמש בכלי "dig" על מנת לבצע שאילתה לשרת ה-DNS ממחשב מרוחק אשר נמצא ב-"External Network" ונראה גם איך מחשב פנימי (מאחורי שרת ה-DNS) גולש לאתר www.linkedin.com ומהי ההשפעה על תוצאת השאילתה שנבצע באמצעות הכלי "dig".

כמה מילים על הכלי הנ"ל: אפשר לומר שכלי זה הוא המחליף של "nslookup" ומאפשר יכולות רבות יותר, הכלי קיים גם ל-Windows וגם ל-Liunx.

אופן השימוש בו:

המחשב ה-"תוקף" אשר נמצא ברשת האינטרנט (וכמובן, מחוץ לרשת האירגונית), מבצע שאילתה לא רקורסיבית (= איטרטיבית), כאשר היעד הוא שרת ה-DNS של הארגון, הוא שואל מהי כתובת ה-IP, של האתר www.linkedin.com כמו שאמרנו בקשת - A Record:

```
C:\dig>dig @[redacted] www.linkedin.com A +norecurse

; <<>> DiG 9.3.2 <<>> @[redacted] www.linkedin.com A +norecurse
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1352
;; flags: qr ra; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 8

;; QUESTION SECTION:
;www.linkedin.com.              IN      A

;; AUTHORITY SECTION:
linkedin.com.           50091   IN      NS      pdns4.ultradns.org.
linkedin.com.           50091   IN      NS      pdns3.ultradns.org.
linkedin.com.           50091   IN      NS      pdns5.ultradns.info.
linkedin.com.           50091   IN      NS      pdns2.ultradns.net.
linkedin.com.           50091   IN      NS      pdns1.ultradns.net.
linkedin.com.           50091   IN      NS      pdns6.ultradns.co.uk.

;; ADDITIONAL SECTION:
pdns1.ultradns.net.     50028   IN      A       204.74.108.1
pdns1.ultradns.net.     52232   IN      AAAA    2001:502:f3ff::1
pdns2.ultradns.net.     50130   IN      A       204.74.109.1
pdns3.ultradns.org.     49996   IN      A       199.7.68.1
pdns4.ultradns.org.     49996   IN      A       199.7.69.1
pdns4.ultradns.org.     51918   IN      AAAA    2001:502:4612::1
pdns5.ultradns.info.    49996   IN      A       204.74.114.1
pdns6.ultradns.co.uk.   49996   IN      A       204.74.115.1

;; Query time: 12 msec
;; SERVER: [redacted]#53([redacted])
;; WHEN: Tue Sep 27 02:44:02 2011
;; MSG SIZE  rcvd: 357
```

תחביר הקלט הולך כך:

```
dig @<DNS_IP> <Request_Site> A +norecurse
```

- A בשביל בקשת רשומת ה-IP בלבד.
- +norecurse עבור שאילתה לא רקורסיבית (מסמנים זאת מפני ש-dig מבצע ברב השאילתות בקשה רקורסיבית כברירת מחדל).

הפלט:

- ANSWER=0 - שרת ה-DNS ענה שהוא אינו מחזיק ברשומה שביקשנו.
- AUTHORITY=6 - כמספר השרתים "המורשים" אשר יכולים "לעזור".

שימו לב שאם היינו שולחים בקשה רקורסיבית אז היינו גורמים לרשומה להיכנס לתוך ה-Cache ואם היינו מבצעים שאילתה נוספת היינו מקבלים פלט שהרשומה נמצאת (כביכול מישהו ביקש אותה קודם לכן) אבל בפועל אנו גרמנו. **באמצעות שאילתה איטרטיבית אנחנו מונעים מצב זה** ויכולים להיות בטוחים שהרשומה אינה מאוחסנת באמצעות מספר חוזר של שאילתות בכדי לוודא את אמינות המידע.

וכעת, נניח כי מחשב פנים ארגוני אכן ניגש לאתר www.linkedin.com בכדי לבצע את עיסוקו. התוקף שוב חוזר על אותה שאילתה, הפעם ניתן לראות את השוני בפלט שהתקבל.

```
C:\dig>dig @[redacted] www.linkedin.com A +norecurse

; <<>> DiG 9.3.2 <<>> @[redacted] www.linkedin.com A +norecurse
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1638
;; flags: qr ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.linkedin.com.              IN      A

;; ANSWER SECTION:
www.linkedin.com.       125     IN      CNAME   la.linkedin.com.
la.linkedin.com.        18      IN      A       216.52.242.80

;; Query time: 3 msec
;; SERVER: [redacted]#53([redacted])
;; WHEN: Tue Sep 27 02:45:51 2011
;; MSG SIZE  rcvd: 79
```

ניתן לראות כי שרת ה-DNS, ענה שהוא אכן מכיר את הרשומה ואינו הפנה אותנו אל שרתים המורשים לעזור לנו. **ניתן לראות זאת ע"י: ANSWER=X > 0 ובנוסף: AUTHORITY=0.**

ניתן גם לראות את שדה ה-TTL, של שני הרשומות שהתקבלו - 125 ו-18.

ניתן להבין ש-Linkedin. מעניקים לתשובה מספר יחסית קטן אשר מקשה על ביצוע סטיסטיקה לגבי זמני הביקור של אותו משתמש/ים.

ניתן לבצע שאילתה לפני כן ולראות מהו מספר ה-TTL הראשוני שהאתר מעניק, ולהשוות אותו עם ה-TTL שהתקבל (חשוב לציין כי המספר מציין "שניות").

קודם לכן, הזכרנו כי יש שימושים לטכניקה הזאת על שרתי DNS ציבוריים. דוגמא קצרה: במהלך הבלאגן שהלך ברשת עם סוני (כאשר דיסק שלהם התקין ללקוחות rootkit ושימש בעצם הכנה לדברים טובים נוספים אשר הגיעו...) החוקר Dan Kaminsky השתמש בטכניקה הזאת בכדי להעריך כמה שרתי DNS "יצרו קשר" עם השרתים אשר היו מעורבים בשליחה וקבלת נתונים עקב התקנתו של ה-rootkit. ההערכה הייתה אז כ-568,200 רשתות שונות.

סיכום

ניתן למנוע שימוש בטכניקה זו באמצעות הגדרות נכונות של שרת ה-DNS, לדוגמא: להגביל תשובות רק למקור אמין, למשל כתובת הרשת הפנימית. טכניקה נוספת היא לחסום מענה לשאילתות איטרטיביות ואז גורם חיצוני לא יכול להיות בטוח האם הוא זה שגרם לשאילתה להכנס ל-Cache או כי הדבר נגרם על-ידי גורם פנימי בארגון.

לשיטה השניה יש חסרון, קיימת דרך לגשש למרות חסימה כזו, שאומרת דבר כזה:

- תחילה, תתשאל לגבי אתר - בין אם הוא מאוחסן ב-Cache או בין אם תגרום לו להיות מאוחסן.
- תתשאל את אותו אתר שוב, ותבדוק כמה זמן לקח לתשובה להגיע. (לצורך הדוגמא, נני כי לקח לו 2 שניות).

- לאחר מכן, תתשאל שוב אתר אחר, ותבדוק כמה זמן לקח לתשובה להגיע. האם הזמן גדול משתי שניות? אם כן סביר להניח כי האתר לא היה קיים ב-Cache, מכיוון ששרת ה-DNS היה צריך לתשאל שרת אחר ובגלל זה לקח לו יותר זמן לענות!

טכניקה זאת מאופשרת באמצעות מנגון הסקריפטינג של Nmap, ובנוסף לאותו סקריפט יש אפשרות גם לבצע שאילתות איטרטיביות למספר רב של אתרים בו זמנית ולבנות מאגר גדול במהירות שמציין מה נמצא ב-Cache כרגע, ומה לא. לא נדגים זאת כאן, אבל ניתן בקלות לקרוא את ה-manual ולהבין כיצד הדבר עובד.

לסיום, טכניקה זאת יכולה לעזור המון באיסוף מידע על היעד ונראה שתתרום רבות בבניית מתקפת Social-Engineering. כי כאשר אנחנו יודעים על התחביבים / עניין / שגרת העובד - נוכל לבנות מתקפה טובה יותר ובעלת סיכויי הצלחה רבים יותר.

מקורות ומידע נוסף

- Dig tool
- Prevent with "simpledns"
- NSE - dns-cache-snoop
- Luis Grangeia - DNS Cache Snooping

Google's GeoLocation API - איפה נמצאות כל המכונות המקוונות בעולם?

מאת: דור זוסמן

הקדמה

פייסבוק שואל "על מה אתה חושב עכשיו?", טוויטר מתעניין "מה קורה עכשיו?", אך אלו כבר חדשות ישנות. היום כולם רוצים לדעת "איפה אתה עכשיו?" פייסבוק הרימו את Places, טוויטר חברו אל Where.com, ורשתות כמו Foursquare, Gowalla , Brightkite ואחרות ממשיכות להופיע...

"איפה אתה?", זאת שאלה שקשה למחשב לענות עליה - אי אפשר לסמוך על המשתמש שיענה עליה, ולא לכל מחשב יש את הציוד המתאים בשביל לענות עליה. למי יש את הציוד המתאים? לסמארטפונים, כל אחד ואחד מהם, מכיל בתוכו רכיב קטן שמסוגל להגיד בדיוק של כמה עשרות מטרים איפה אתה בעולם בכל רגע נתון.

זו הדלת לתוך העולם הזה, אבל כמובן שלא נשאיר את המחשבים מאחור, ולכן קם אדם בשם Stan Wiechers והחליט למצוא לבעיה זו פתרון. בשיתוף עם גוגל, (שבמקרה החזיקה בבעלותה מאגר די נרחב של כתובות MAC של ראוטרים ונ.צ. שלהם...), החליט לכתוב API ב-Javascript, שיותר מאוחר יתחבר ל-HTML5 ויאפשר לכל מתכנת, למצוא את הנ.צ. של גולש, ב-3 שורות קוד פשוטות, נשמע מפחיד? גם לנו.

איך זה עובד ומה נעשה עם המידע?

<u>במחשב האישי:</u>

על מנת למצוא את הנ.צ. של הלקוח, GeoLocation מבקש מהדפדפן כמה דברים:

- כתובת ה-mac, עוצמת הקליטה וה-SSID של ה-AP .
- כתובת ה-mac, של המתאם האלחוטי (אם יש).
- הרשתות האלחוטיות שנמצאות בטווח קליטה, הכתובות הפיזיות שלהם, SSID ועוצמת הקליטה של כל אחת מהם.

כל המידע נאסף ע"י הדפדפן ונשלח לכתובת www.google.com/loc/json בצורה הבאה:

```
{
   "version":"2.0.0",
   "request_address":true,
   "access_token":"2:TOKEN",
   "wifi_towers":
   [
      {"mac_address":"r4-72-20-11-gf-
34","ssid":"MyOwnWifi","signal_strength":-20},
      {"mac_address":"00-33-64-d1-34-
0d","ssid":"OtherWiFi","signal_strength":-64},
      {"mac_address":"00-13-62-e1-36-
cd","ssid":"OtherWiFi2","signal_strength":-84}
   ]
}
```

המידע שחוזר מכיל את הנ.צ. המשוער של הלקוח, חשוב להגדיש שהשרתים של גוגל משתמשים גם בכתובת ה-IP של הלקוח ובכך משפרים את דיוק התוצאה, וכן המסד מבצע התאמות וכיול מחדש של המידע שכבר מאוכסן, ביחס למידע החדש ובכך משפר את דיוקו מפעם לפעם.

בסמארטפון:

אותם הפרטים יאספו כמו במחשב במידה ואין חומרת GPS במכשיר, במידה וקיימת חומרה כזו, מכשיר ה-GPS יאסוף את הנ.צ. מהלווייניים.

נבצר ממני להראות לכם פאקט של סמארטפון ואיני יודע אם המידע מועבר ישירות לאתר או עובר דרך גוגל. אני מניח שהמידע נשמר ע"י גוגל (במיוחד אחרי שקראתי את המחקר של סאמי בנושא סמארטפונים וGPS), ואני בטוח שהמידע הזה גם משמש לייעול ושיפור המאגר.

הסכנות הפוטנציאליות שדבר

כל חוקר אבטחה שקרא עד פה מדמיין כבר את ניצולים לרעה אפשריים. כמה שאני חושב עליהם:

1. הונאות פישינג העושות שימוש במידע זה על מנת לקנות את אמונו של הקורבן.
2. Malware המנצל לרעה מנגנון זה ומוסיף אתר זדוני ל-White-List של הדפדפן, או שולח בעצמו את חבילת המידע ומשתמש במידע לצורך פרסום או חשיפת זהותו האמתית של הקורבן.
3. אתר המשלב מסד של מרשם אוכלוסין (אגרון לדוגמא...), מסד של רחובות והנ.צ. שלהם ומסוגל להעריך את זהותו של הגולש בעזרת נתוני המיקום.
4. Adware מותאם לקורבן מבחינת מיקום ושפה.
5. וכמו כמעט בכל נושא- הדמיון הוא הגבול. כל מה שצריך זה להיות יצירתיים.

תומך בכל פלטפורמה, וזמין לכל מתכנת - איך GeoLocation עובד בפועל?

כמו שניתן לראות בתמונה למטה, טכנולוגית ה-GeoLocation נתמכת בכל גירסה חדשנית של דפדפן גדול ובכל גירסה יחסית חדשה של מערכת הפעלה לסמארטפונים, וברוב אילו שאינו תומך, ניתן להתקין פאטצ'ים פשוטים.

GEOLOCATION API SUPPORT

IE	FIREFOX	SAFARI	CHROME	OPERA	IPHONE	ANDROID
9.0+	3.5+	5.0+	5.0+	10.6+	3.0+	2.0+

הפשטות שבה אפשר לעשות בטכנולוגיה הזאת שימוש מפחידה. בעזרת 3 שורות קוד בלבד, ניתן לזמן בקשה זו ולעשות כאוות נפשנו במידע:

```
function get_location() {
  navigator.geolocation.getCurrentPosition(show map);
}
```

כעת יש לנו ביד שני קבועים (const):

- position.coords.latitude - קו הרוחב המשוער.
- position.coords.longitude- ו- קו האורך המשוער.

כאשר מניחים את שניהם על מפת עולם בגודל סטנדרטי, הנקודה בה הם חוצים זה את זה היא הנקודה שבה הלקוח ממוקם פיזית כביכול.

Google maps יודע לעשות זאת לבד, רק נכניס את הקורדינטות לחיפוש עם פסיק המפריד ביניהם ונראה שתופיע לנו נקודה על המסך, שם הלקוח שלנו נמצא.

ישם עוד מספר משתנים שנוכל לקבל בתוך position שיכולים לעזור לנו, כגון coords.accuracy, שמסמל את רדיוס מרווח הטעות שלנו במטרים, timestamp - שמסמל מתי נדגם מיקומינו וכו'.

כל זה נמצא בתוך מחלקה בשם gears_init.js. מחלקה זו היא המחלקה שמזמנת את Google Gears במידה והוא מותקן (מובנה בתוך כרום) ואם לא, מזמנת את המחלקה geo.js שהיא מחלקת ה-GeoLocation העומדת בפני עצמה.

אם נעיין בקוד נוכל למצוא דברים מעניינים, כמו הקריאה למחלקה בתוך Google Gears במידה והוא מותקן:

```
provider=google.gears.factory.create('beta.geolocation');
```

או הקריאה לרכיב ה GPS במכשירי Palm:

```
r=new Mojo.Service.Request('palm://com.palm.location', {
      method:"getCurrentPosition",
      parameters:parameters,
      onSuccess: function(p){success({timestamp:p.timestamp, coords:
{latitude:p.latitude,
```

ועוד מספר רב של דברים מעניינים, אני ממליץ לעבוד ברפרוף על geo.js על מנת להבין את התפקוד של GeoLocation בצורה טובה יותר.

אחרי שהצגנו את הטכנולוגיה, נעבור לחלק המעניין של מאמר זה ☺

חבילת המידע המכילה את כתובות ה-MAC, SSID של הרשתות וכתובת המתאם שלנו נשלח על ידי הדפדפן, בלי מעורבות של האתר שביקש את המידע. **הפרדת הגורמים הזו חשובה בכדי למנוע מגורמים לא רצויים לגשת למידע רגיש זה**. הנקודה המעניינת היא שמכיוון שהדפדפן בשליטתנו, אנו נוכל לשלוט במה שישלח מהדפדפן וכמובן- במה שיתקבל.

נוכל לעמוד בין הדפדפן לגוגל, להחליף את המידע על הרשתות הסובבות אותנו במידע שאנו נבחר, ובכך נוכל לאתר כתובת MAC של **מכשיר שלא בבעלותינו!**

נערוך את הבקשה שנשלחת לגוגל, בצורה הבאה:

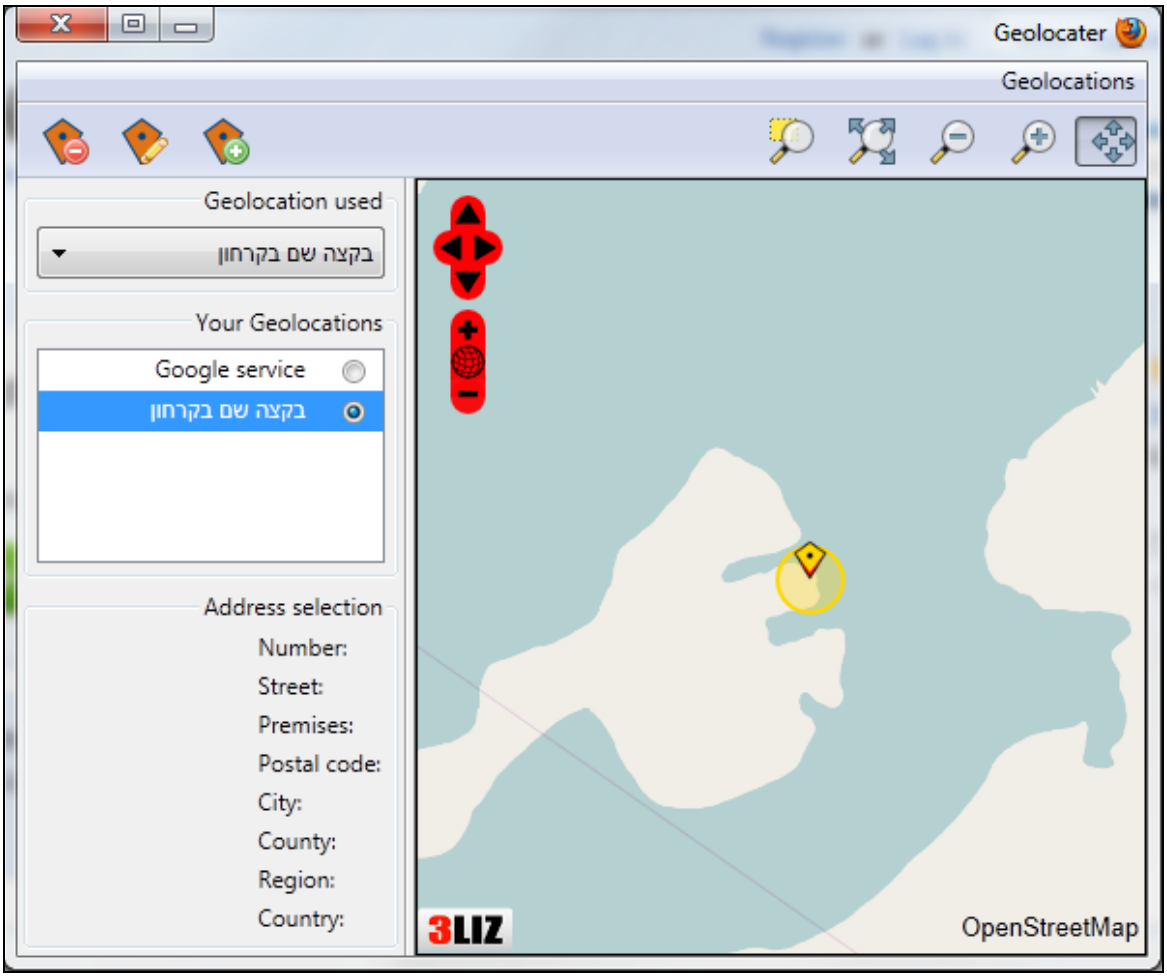
```
{
   "version":"2.0.0",
   "request_address":true,
   "access_token":"2:TOKEN",
   "wifi_towers":
   [
      {"mac_address":"(MAC address of device)"}
  ]
}
```

וזהו זה...

חוקר האבטחה Sami Kamkar, החליט לעשות את זה עוד יותר פשוט: הוא בנה אתר נוח עם ממשק Ajax, השולח שולח את הבקשה בשמנו, ומדפיס את התוצאה בתוך מפה של Google maps. כל מה שנותר הוא להדביק את כתובת ה-MAC וללחוץ על הכפתור...

זה רק כיוון אחד של ניצול, מה עם הכיוון השני? הרי אם אנחנו שולטים במה ישלח לגוגל, אנו יכולים גם לשלוט במה גוגל יחזיר לנו לא? כמובן שאפשר לעשות זאת עם כל sniffer בעזרת עריכת חבילת המידע הספציפית ואין טעם להסביר איך עושים זאת, אבל נסביר כיצד ניתן לעשות זאת עם שימוש ב-Addon לפיירפוקס בשם GeoLocater ו"לעבוד" על פייסבוק Places.

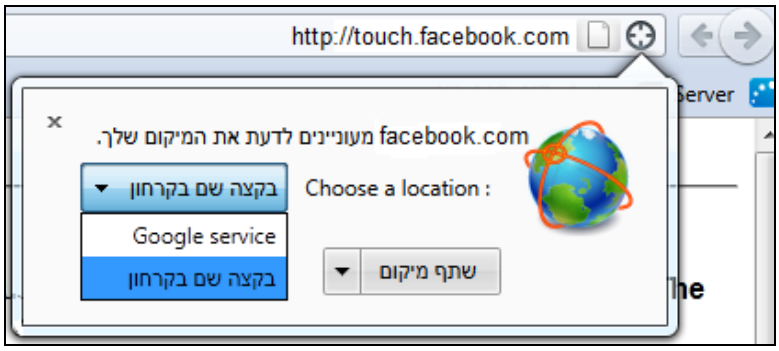
את GeoLocater ניתן להוריד באתר התוספות הרשמי של מוזילה, והקונפיגירציה שלו די ברורה. נוסיף נקודת ציון מזוייפת וניתן לה שם:



כעת, נשתמש בתוספת נוספת בשם User Agent Switcher שיעזור לנו להכנס לגירסה לניידים של פייסבוק (זאת מכיוון שאין גישה ל-Places דרך הממשק הרגיל של הדפדפן).

נפעיל מחדש את הדפדפן ונשתמש ב-UA של Chrome. נגלוש ל-facebook.com ונראה שהועברנו ל-m.facebook.com, אבל זה כמובן לא מספיק לנו, אנו צריכים גירסה יותר חדישה של הממשק, נגלוש ל-touch.facebook.com. נבחר ב-More ושם ננווט ל-Places כאשר נתבקש לבחור בספק שירות שיתן לנו את הקורדינטות, נבחר בשם נקודת הציון שהוספנו ונלחץ "שתף מיקום".

והתוצאה לפניכם:



אם נמשיך כרגיל, נראה שבסוף התהליך נוצר לנו מקום חדש, שטוען שאנחנו באנטרקטיקה, מה שנשאר לעשות זה רק להעלות תמונות שלנו עם פינגווינים ודובי קוטב, ולפרסם בפייסבוק ☺

זוהי רק דוגמא בסיסית לשימוש בטכנולוגיה הזו לרעה. ובמקרה הנ"ל מדובר במקרה לא מזיק. אך ניתן לחשוב על סיטואציות שבהם התבססות על טכנולוגיה זאת לאיכון המשתמש יכולים להיות פחות תמימים, כגון מנגנוני הגנה המאפשרים גישה רק ממקומות מסוימים בעולם.

סיכום

מי יודע באילו רוגלות ורשעות יש ניצול של מאגר זה. לטעמי אסור לאף חברה להחזיק כזה מאגר, ובטח שלא בצורה כל כך לא מאובטחת, הרי גוגל כבר הוכיחו שדאגתם האחרונה היא מי ניגש אל המאגר...

אני מניח שהשאלה שלי היא בעצם - מי הסמיך את גוגל להיות אוצרת המיקומים של כל רכיב אלקטרוני הניגש לאינטרנט בעולם, ולמה לאף אחד לא אכפת מזה?

על המחבר

אני דור, חוקר אבטחה די צעיר מהמרכז, עוד לא בגיל צבא אפילו, לאחרונה החלטתי לעשות משהו מועיל ולהציע עזרה לDigitalWishper, מתוך רצון לשפר את הקהילה הישראלית וכמובן לתמוך במגזין הנהדר הזה, אחרי כמה מיילים עם cp77fk4r מצאנו נושא והתחלתי את המחקר, זה המאמר הראשון שאני כותב בצורה מקצועית אי פעם ומפרסם אותו, ולכן אשמח למשוב מכל סוג שהוא, אם למישהו יש הערות / הארות ניתן תמיד לפנות אלי הדוא"ל: risomrisom@gmail.com ואחזור אליכם בהקדם האפשרי.

דברי סיום

בזאת אנחנו סוגרים את הגליון ה-25 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 (או בעצם - כל יצור חי עם טמפרטורת גוף בסביבת ה-37 שיש לו קצת זמן פנוי [אנו מוכנים להתפשר גם על חום גוף 36.5]) ואנשים המעוניינים לעזור ולתרום 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

"Talkin' bout a revolution sounds like a whisper"

הגליון הבא ייצא ביום האחרון של חודש אוקטובר 2011.

אפיק קסטיאל,

ניר אדר,

1.10.2011