Technologic Papers

רישום CVE-ים - מדריך לחוקר המתחיל

מאת אייל איטקין

הקדמה

אז התחלת מחקר, מצאת חולשה מגניבה במוצר מסחרי / ספריית קוד פתוח ואפילו הצלחת להרים הדגמה מגניבה, סחתיין ☺ אבל זה לא הסוף, עכשיו צריך להתחיל בתהליך הבירוקרטי של "responsible disclosure" והוצאת CVE רשמי לחולשה שמצאת.

מדריך זה נכתב על מנת שתהליך רישום ה-CVE ילך לכם בצורה החלקה ביותר, מקווה שתמצאו אותו מועיל.

הערת שוליים:

- מדריך זה (כמעט) ולא יעסוק בתהליך ה-"responsible disclosure". עם זאת, המדריך יכלול המלצות בנוגע לסנכרון הזמנים בין שני התהליכים הנ"ל.
- מדריך זה נכתב מנסיוני האישי כחוקר עצמאי וכחוקר כיום בחברת Check Point. עם זאת, המדריך נכתב מנקודת מבטי האישית כחוקר, וביוזמתי האישית, ללא קשר למעסיק שלי.

למרות שמדי פעם ישנם שינויים בבירוקרטיית רישום ה-CVE-ים, המדריך נכון למועד כתיבתו וכולי תקווה שיהיה מועיל ככל הניתן גם בעתיד.

מבוא

קצת מונחים

לפני שנתחיל, בואו נסביר את המונחים השונים:

- CVE - **C**ommon **V**ulnerability and **E**xposure: מזהה ייחודי (חד-חד-ערכי) לחולשה
- CNA - **C**VE **N**umbering **A**uthority: גוף או חברה אשר מורשה להנפיק מזהי חולשות
- CWE - **C**ommon **W**eakness **E**numeration: מזהה חד-חד-ערכי לסוג טכני של חולשה
- MITRE (לינק): הגוף הראשי האחראי על רישום והפצת מזהי החולשות
- NVD - **N**ational **V**ulnerability **D**atabase (לינק): מאגר פומבי נוח של מזהי החולשות שהתפרסמו לאורך ההיסטוריה. המאגר מנוהל ומתוחזק על ידי ממשלת ארה"ב.

עד לשנים האחרונות, מזהה החולשה היה מהצורה הבאה: CVE-2016-1234. כלומר, בדוגמה זו:

- החולשה **נרשמה** בשנת 2016 - במידה והתיקון מתעכב, החולשה לפעמים תתפרסם בשנה העוקבת (2017 למשל). על כן החלק הראשון במזהה מציין מתי החולשה התחילה את תהליך הרישום.
- 1234 - מזהה חד-חד-ערכי (חח"ע) של החולשה בשנה בה היא נרשמה.

למה קיימים CVE-ים?

בעוד שחוקרים רבים משתמשים ב-CVE-ים כדי לנופף בכמה חולשות הם מצאו, לא זו המטרה המקורית לשמה הוחלט לתת מזהים לחולשות. מזהה חולשה תוכנן כך שיהיה חח"ע, בכדי לעמוד במספר יעדים:

1. תיאום וסינכרון בין צוות המחקר ובין הגוף או הגופים הפגיעים לאותה החולשה, ונמצאים בתהליך התיקון שלה:
   a. דוגמא טובה לכך היא CVE-2014-0160 הידוע גם בכינוי (heartbleed)
   b. במקרה הזה מספר רב של גופים נדרשו לתקן את אותה החולשה שנבעה מספריית קוד פתוח מאוד פופולארית, ששולבה במספר רב של מוצרים ושרתים
   c. השם עצמו ניתן לחולשה רק לאחר הפרסום שלה כאמצעי מיתוג שנוי במחלוקת (פרקטיקה שנהוגה גם כיום, אם כי בעצימות נמוכה יותר)
2. עדכון הציבור בנוגע לתיקון חולשה:
   a. (בעולם תקין) מזהה החולשה יופיע ב-Git Commit שמתקן את החולשה (קוד פתוח)
   b. (בעולם תקין) מזהה החולשה יופיע ב-Release Notes של הגרסה המתוקנת של המוצר
3. במקרה הפחות טוב, עדכון פומבי של הציבור בנוגע לחולשה שטרם תוקנה:
   a. עבור חולשות שנמצאו כשהן מנוצלות בפומבי ("in the wild") ללא דיווח של חוקר ליצרן
   b. במקרים בהם המוצר לא תוקן בחלון הזמנים שהחוקר הגדיר, או שהיצרן בחר שלא לתקן

או בקיצור, CVE הוא השם המדעי של החולשה, והוא תוכנן כך שיהיה ייחודי וברור.

מאחורי הקלעים

בשנים האחרונות דווחה כמות גבוהה יחסית של חולשות, ובכל שנה מחדש נגמר מאגר המזהים בני 4 הספרות של אותה השנה... כדי להתמודד עם הבעיה הוחלט להאריך את מזהי החולשות, ולכן כיום חלק מהמזהים כוללים 5 ספרות ויותר. לדוגמא: המזהה CVE-2018-20174 מתאר חולשה שדווחה בסוף שנת 2018 ולכן קיבלה מזהה של 5 ספרות במקום 4.

מכיוון והמזהים מחולקים לשנים, מאחורי הקלעים מתבצע תהליך הרישום הבא:

1. בתחילת השנה (לרוב) ה-CNA-ים השונים יבקשו מ-MITRE בלוקים של מזהי חולשות, כדי שיוכלו להשתמש בהם במהלך השנה
2. במידה ובלוק המזהים נגמר, CNA יכול לבקש מ-MITRE בלוק נוסף של מזהים לאותה השנה
3. MITRE גם היא מתפקדת בתור CNA, ולכן היא שומרת לעצמה בלוק מכובד של מזהי חולשות
4. במידת הצורך, עם כל הבלוקים "נגמרו", MITRE פשוט תקצה בלוקים חדשים עם מזהים גדולים יותר (ולכן אפשר להגיע גם למזהים של יותר מ-4 ספרות בשנה מסויימת)

לדוגמא, נגיד שבתחילת שנת 2019 חברת Google ביקשה מ-MITRE בלוק של 100 מזהים. MITRE תקצה לה בלוק של מזהים והיא תקבל לאחריותה את כל המזהים בתחום. למשל: CVE-2019-2200 ועד CVE-2019-2300. מרגע שההקצאה התבצעה, האתר של MITRE יראה כי כל אחד מהמזהים בתחום הוא "שמור" (reserved). לעומת זאת, באתר של NVD לא נמצא אזכור לאף אחד מהמזהים הנ"ל, בגלל שהם עדיין לא הוקצו רשמית לחולשות ספציפיות.

שימו לב: קיומו של מזהה בעל 5 ספרות לא יעיד על כך שהיו 10,000 חולשות שדווחו באותה השנה. בסיכוי סביר להרבה מהחברות יש בלוקים של מזהים לא מנוצלים, משום שהקצאה בבלוקים היא מטבעה לא יעילה.

אז מי בכלל CNA ולמה זה חשוב?

כשבאים לדווח לחברה / פרויקט קוד פתוח על חולשה, נרצה לפנות ל-CNA המתאים ביותר, זה שבאחריותו לרשום את החולשה שמצאנו ולהקצות עבורנו את ה-CVE שניתן לה. לשם כך חשוב להבין מי הם ה-CNA השונים:

1. חברות מסחריות (כדוגמאת Adobe) ישמשו בד"כ כ-CNA ויקבלו בלוק של CVE-ים שישמש אותן כדי לרשום את החולשות שנמצאו במוצרים של החברה
2. חברות פיתוח ומחקר ישמשו בד"כ כ-CNA משולב (כדוגמאת Google)
   a. רישום של חולשות במוצרי החברה
   b. רישום של חולשות שנמצאו על ידי חוקרי החברה, במקרים בהם לא נמצא גוף מתאים יותר בכדי לרשום את החולשה בעצמו
3. פרויקטי גוף פתוח גדולים מספיק (קהילת מפתחי הקרנל של Linux למשל)
4. MITRE (ג'וקר)
   a. במידת הצורך אפשר לפנות ל-MITRE ישירות להנפקת CVE (יפורט בהרחבה בהמשך)
5. כל השאר - Distributed Weakness Filing Project (DWF)
   a. לכאן ידווחו כל החולשות שאין למי לדווח, לרוב חולשות בפרויקטי קוד פתוח קטנים, ושנמצאו על ידי חוקרים עצמאיים / חוקרים מחברות שאינן CNA
   b. הוספת ה-CNA הזה היא שינוי של השנים האחרונות, שנועד להפחית עומס מ-MITRE

את הרשימה המלאה של כל גופי ה-CNA אפשר למצוא בלינק הבא לאתר של MITRE.

פנייה ישירה ל-MITRE

באתר של MITRE יש טופס נוח למילוי כל הפרטים הנדרשים לרישום CVE - לינק (בתפריט יש לבחור באופציה "Request a CVE ID"). ישנם מספר מקרי קצה בהם נרשה לעצמנו לפנות ישירות אליהם, אבל העיקרי שבהם הוא שאתם מוגבלים בזמן והתהליך דרך DWF איטי מדי.

בכל פנייה ישירה ל-MITRE אני נוהג לכתוב בהערות למה פניתי ישירות אליהם, ועד עכשיו התהליך היה חלק.

הנפקת CVE

אני צריך לעשות משהו בעצמי?

פעמים רבות, החולשה מדווחת לחברה / פרויקט קוד פתוח שמנהל בעצמו את הקצאת ה-CVE-ים, לרוב מכיוון שהגוף האחראי על המוצר הינו CNA. דוגמאות:

1. באגים בקוד של PHP שמתוייגים כאבטחתיים, יקבלו CVE שיוקצה ישירות על ידי המפתחים של PHP (זמן טוב בד"כ גם להגיש את מה שמצאתם ל-bug bounty שלהם ב-hackerOne).
2. באגים בקוד של מוצרי Google, Microsoft או רוב החברות הגדולות יטופלו על ידי החברה עצמה, והיא תעדכן אתכם מה ה-CVE שהוקצה לה

בעוד שבמקרים אלו נחסך מכם הצורך לרשום בעצמכם את ה-CVE, יש לכך גם מספר חסרונות:

1. תיאור החולשה יכתב על ידי ה-CNA ועלול להכיל טעויות או שבקושי יכלול פרטים טכניים שיועילו לחוקרים אחרים ("נמצאה חולשה במודול X שעלולה לגרום Y" וזהו)
2. חומרת החולשה תדורג על ידי ה-CNA, ובמקרה שהוא היצרן הוא מטבעו ינסה להפחית מחשיבות הנזק שנמצא במוצר שהוא פיתח

או בקיצור, מי שכותב את הפרטים שולט בהיסטוריה.

טיפ: במהלך תהליך דיווח החולשה למפתחים ואחרי שאישרו את נכונות החולשה (ולא ישר במייל הראשון שנשלח), רצוי לשאול אותם אם אפשר "לסייע להם" בתהליך הקצאת ה-CVE. זאת כדי להבין האם אתם צריכים לעשות זאת, או שהם יקחו את האחריות לכך.

מלכוד: לא מעט פרויקטי קוד פתוח יגידו לכם כחלק מתהליך התיקון שאם לא הקצתם עדיין CVE לחולשה, אז הם יעדיפו שכבר לא יוקצה לה מזהה כלל. במקרה כזה תרצו להחליט האם להנפיק בעצמכם מזהה בכל זאת או לוותר על זה. את ההחלטה רצוי לעשות כתלות בחומרת החולשה שמצאתם.

באחריותי להנפיק CVE וגם מצאתי CNA מתאים, מה עכשיו?

חלק זה יתפצל בין שני תרחישים:

1. חוקר עצמאי שאינו חלק מ-CNA - כנראה שזה התרחיש הנפוץ
2. חוקר שעובד בחברה שהיא CNA

אני חוקר עצמאי

יש 3 דרכים נפוצות לפנות אל CNA:

1. טופס אינטרנט של MITRE (הוסבר קודם) - מענה ראשוני יתקבל תוך 24-48 שעות (די מהיר)

2. Google Form של DWF (ראו לינק קודם) - תהליך יחסית איטי אך פשוט
3. שליחת מייל ישירות ל-CNA

שליחת מייל ישירות ל-CNA תתבצע במספר מועט של מקרים, משום שלרוב יהיה מדובר על אותה החברה שדיווחתם לה על החולשה, ואז היא כבר תנפיק את ה-CVE בעצמה. עם זאת, במידה ונאלצתם לשלוח מייל, ההמלצה שלי היא לפתוח במקביל את הטופס האינטרנטי של MITRE, ולרשום במייל את כל הפרטים הטכניים, סעיף-אחר-סעיף כמו בטופס. בצורה זו תהיו בטוחים שלא פספסתם שום פרט.

אני חלק מקבוצת מחקר, והמעסיק הוא CNA

באופן מפתיע, התהליך במקרה הזה יהיה **מורכב** יותר, Go Figure.

היות והחברה שאתה עובד בה היא כבר CNA, זה אומר ש(בתקווה) יש לה בלוק מוכן של CVE-ים. גש אל האחראי על ניהול הבלוק בחברה שלכם, ותבקש ממנו שיקצה לך מזהה. באחריותו לדאוג לרישום כך שלא יהיו כפילויות פנימיות אצלכם בחברה.

במידה ולחברה שלכם עוד אין בלוק מוכן, או שהוא התמלא, גשו אל הטופס האינטרנט של MITRE, ובחרו הפעם באופציה: "Request a block of IDs (For CNAs Only)".

אז מתי MITRE ידעו מה פרטי החולשה?

בטופס של MITRE ישנה אופציה לדווח להם על חולשה שהונפקה על ידי ה-CNA ("Request an update to an existing CVE Entry"). אופציה זו מתפקדת בשתי צורות:

1. חולשה שהונפקה על ידי MITRE או שכבר התפרסמה - פרטי החולשה יעודכנו
2. חולשה שהוקצתה לבלוק של CNA וטרם פורסמה - החולשה תרשם על ידי MITRE והם לא ידעו מה לעשות איתה...

כן, קראתם נכון. מסיבה שעוד לא הבנתי, במקרה הזה, למרות שמילאתם את כל הפרטים בטופס האינטרנטי, אתם תקבלו מ-MITRE מייל שמבקש מכם להגיש שוב את הפרטים המלאים של החולשה באחת מהצורות הנתמכות (ראו רשימה כאן). אני לרוב מגיש את החולשות כ-"flat file" ושולח אותן בגוף המייל.

מלכוד #1: אם בפרטים שתשלחו ל-MITRE לא יהיו הפניות חיצוניות (בלוג שמסביר על החולשה, לינק להודעה של המפתח / יצרן בנוגע לחולשה או לתיקון שלה), הם **יסרבו** לרשום את החולשה, כי אין לה ראיות חיצוניות. וזה מתקשר למלכוד הבא.

מלכוד #2: במידה ו-MITRE קיבלו את הפרטים ולא היו בעיות, הם ככל הנראה **יפרסמו** את הפרטים תוך שבוע-שבועיים ממועד קבלתם. כן, גם אם כתבתם במפורש שלא יפרסמו כי אתם מסנכרנים הרבה חולשות בין מספר גופי פיתוח, לצורך פרסום מחקר גדול.

מסקנה: בשורה התחתונה, MITRE מתייחסים לדיווח שלכם כאל אישור לפרסום, ולכן רצוי לדווח להם על ההקצאה שביצעתם מהבלוק שלכם רק לקראת הפרסום המתוכנן.

אז מה הפרטים שמופיעים ב-CVE שנמלא?

לצורך ההסבר אני אעבור על הטופס הפומבי של MITRE (שלדעתי הוא הנוח ביותר), ואנסה להסביר כל סעיף שרלבנטי לחולשה עצמה (ולא שם היצרן, גרסא או האם היצרן אישר):

1. **סוג חולשה (Vulnerability Type)**
   a. זהו הסיווג הטכני של ה**חולשה** (ולא של המשמעות שלה כלפי המוצר הפגיע)
   b. MITRE מציעים ממש מעט אופציות, וזה די עצוב. אני לרוב בוחר ב-"Other or unknown" ורושם את הסוג המתאים בשדה החדש שמתווסף לטופס.
   c. בפורמט "flat file" השדה יקרא "Problem Type" ואז לרוב אחפש CWE מתאים שיתאר את החולשה (ראו רשימה מלאה כאן)
2. **משמעות החולשה (Impact)**
   a. מה תוקף יוכל להשיג בהינתן החולשה?
3. **רכיבים מושפעים / פגיעים (Affected Components)**
   a. מה הרכיב הפגיע במוצר?
   b. איזה מנגנון במוצר הוא המנגנון שאתם תוקפים?
4. **וקטור תקיפה (Attack vectors)**
   a. דרך איזה ממשק אתם תוקפים את המוצר? (רשת, קריאה מקובץ, וכו')
5. **המלצה לתיאור טכני (Suggested description)**
   a. במידה והתיאור שלכם מנוסח נכון, הוא ישאר כמעט כמו שהוא
   b. לפעמים MITRE יעשו איזה שינוי קוסמטי (למשל: הם שינו לי long packet field ל-long string, אפילו שהשדה המדובר בהודעה לא היה מחרוזת...)
   c. תנסו להסתכל על מזהים מפורטים של חולשות ולנסח כמו שהם ניסחו. לדוגמא:

> "FreeRDP prior to version 2.0.0-rc4 contains an Integer Truncation that leads to a Heap-Based Buffer Overflow in function update_read_bitmap_update() and results in a memory corruption and probably even a remote code execution".

   d. אחרי ניסוח אחד מוצלח, פשוט תחזרו אליו בכל פעם ותנסחו שוב כמוהו
   e. **חשוב:** אל תתהדרו במשמעות שלא הצלחתם להדגים בפועל. רצוי לכתוב "הרצת קוד" ללא מילות סייג, רק כאשר הצלחתם להדגים אחת. אחרת, תחליטו מה לדעתכם הסיכוי להשמשה מוצלחת, ותסייגו בהתאם:
      i. "probably" - אפשר ולא הדגמתי
      ii. "possibly" - יהיה קשה להדגים

6. **קרדיט (Discoverers)**
   a. זה השדה היחידי ש**לא** יופיע בפרסום הסופי - באסה, נכון?
7. **הפניות חיצוניות (references)**
   a. לינקים לכל דבר שימושי (אפשר להשאיר ריק בנתיים):
      i. Release notes של היצרן שמודיע על התיקון
      ii. Git commit שמתאר את התיקון
      iii. Blog post שלכם שמתאר את המחקר שעשיתם

בסיכוי סביר, בזמן מילוי הטופס לא יהיו לכם הפניות חיצוניות. הפניות אלה יתווספו כאשר תרצו לפרסם את החולשה ולהודיע עליה לעולם. כרגע, החולשה תופיע כ"שמורה" (reserved) ולא תהיה פומבית לציבור.

פרסום החולשה - Public Disclosure

אז תהליך התיקונים הסתיים סוף סוף, והגיע הזמן לפרסם את המחקר שלכם. לשם כך יש 3 אופציות:

1. החברה רשמה את ה-CVE וכנראה תפרסם אותו בעצמה או בתיאום איתכם
2. אתם עובדים ב-CNA - תשלחו ל-MITRE את הפרטים כשבוע - שבועיים לפני הפרסום הרצוי
3. אחרת, נמלא את הטופס האחרון של MITRE מיד אחרי הפרסום

הטופס האחרון של MITRE הוא כאשר בוחרים באופציה ("Notify CVE about a publication") ובו נוכל לציין את רשימת כל ההפניות שנרצה להוסיף על אלה שרשמנו קודם.

חשוב: אין אפשרות לפרסם ללא הפניה למקור הפרסום: blog post שלכם או הודעה של היצרן.

טיפ: מכיוון ואין שדה "קרדיט" לאחר שה-CVE מתפרסם, הקפידו להוסיף הפנייה חיצונית לבלוג שלכם בו תכתבו על החולשה שמצאתם. בצורה זו החולשה תשוייך אלייכם, ותופתעו לגלות שלא מעט אנשים יכנסו לקרוא את הבלוג שכתבתם בזכות ההפניות הללו.

עדכון פרטים / סגירת קצוות

במידת הצורך, ניתן לחזור אל MITRE ולמלא "עדכון פרטים" ל-CVE שפורסם. בשביל העדכון כנראה שתצטרכו להוכיח זיקה לפרסום המקורי של ה-CVE, או למחקר שמצא את החולשה שדווחה. עדכון לרוב יתבצע במקרים הבאים:

1. הוספת הפניות חיצוניות - כקבוצת מחקר, את הפרטים שלחתם ל-MITRE לפרסום לפני שיצא Blog post, זה הזמן להוסיף הפנייה אליו
2. תיקון שגיאות - קורה מדי פעם, יחסית נדיר

התוצר הסופי

סיימתם את כל התהליך, ועכשיו יש לכם CVE משלכם, הוא מפנה למחקר פומבי שלכם (אחרת לא תקבלו קרדיט לחולשה) והוא פומבי כך שכל העולם יכול לראות אותו, מזל טוב ☺

אז איך יראה הפרסום, ואיזה פרטים הוא כולל? הפרסום הרחב ביותר יהיה ב-NVD, והוא יכלול את הפרטים הבאים:

1. תיאור החולשה - שדה הטקסט שכתבתם
2. הפניות חיצוניות - רשימת הלינקים שלכם ועוד מיליון לינקים לא מועילים שמתווספים עם הזמן ומתאמים בין יותר מדי אתרים שכל אחד מפרסם העתק משלו של מאגר החולשות הפומבי
3. מפרט טכני של החולשה, לרוב במבנה CVSS 3.0

CVSS מה?

CVSS 3.0 הוא התקן העדכני (בטח ישתנה שוב בזמן הקרוב) לפירוט הפרטים הטכניים של החולשה. מדובר על אוסף שאלות שמגדירות יחד את הסיכון שגלום בחולשה, ובהתאם ינתן לה ניקוד (רציונאלי, אך לאו דווקא שלם) בין 0 ל-10. התקשורת לפעמים מתלהבת כשמדווחות חולשות עם ניקוד מושלם של 10, למרות שהנסיון שלי עד כה מראה שהסיווג שמתבצע לחולשות הוא ברמה די ירודה, ולא תואם בפועל למורכבות הניצול / רמת הסיכון שהחולשה משקפת למערכת בסיטואציה אמיתית.

מחשבון מועיל לחישוב הניקוד של חולשה ניתן למצוא כאן, המחשבון גם מוסיף הסבר בנוגע לכל אחד מהקריטריונים, כולל דוגמאות די טובות לכל סעיף.

אז מה הקריטריונים וממה הם מורכבים?

1. **וקטור תקיפה:** רשתי (אינטרנט), קרבה (רשת מקומית), מקומי (לרוב קריאה מקובץ), גישה פיסית.
2. **מורכבות התקיפה:** פשוטה / מורכבת. תקיפה תהיה מורכבת במידה והיא הסתברותית ו/או תלויה במרכיבים שלא בשליטתו של התוקף ושעלולים להקשות על הניצול. תקיפה יכולה להחשב "פשוטה" גם אם היא דורשת מחוקר להיות מנוסה מאוד בעולם השמשת החולשות.
3. **הרשאות נדרשות:** כלום, מעט (הרשאות חלשות ובסיסיות), חזקות (הרשאות admin למשל)
4. **אינטראקצית משתמש:** נדרשת / לא נדרשת. מדובר בביט בודד של החלטה, אין כאן סקאלה.
5. **הקשר:** השתנה / לא השתנה. האם תקיפה מוצלחת משפיעה גם על רכיבים מלבד זה שנתקף?
6. **פגיעה בסודיות:** אין, מועטה, גבוהה.
7. **פגיעה בשלמות:** אין, מועטה, גבוהה.
8. **פגיעה בזמינות:** אין, מועטה, גבוהה.

קריטריונים אלה לרוב מחושבים על ידי MITRE בעצמה, אך לעיתים נקבעים על ידי ה-CNA שמנפיק את החולשה (דבר שמאפשר לו להפחית מעט בנזק כדי להמנע מרעש תקשורתי פוטנציאלי).

סיכום

תהליך הנפקת מזהה חולשה (CVE-ID) הוא טכני בעיקרו, ומשתנה בהתאם לגוף / מוצר בו נמצאה החולשה. רוב שרשרת הטיפול ברישום מורכבת מפקידים, ולכן אין סיבה לפחד מהתהליך. אני מקווה שהמדריך העביר בצורה טובה את הפרטים ואני מקווה שתמצאו אותו מועיל.

הערת המחבר

בשנים האחרונות מונפקים המון CVE-ים, ורובם המוחלט **לא** מועיל כאשר באים לתקוף מערכת (לצרכי מחקר כמובן). אנא הקפידו כי החולשות שאתם מדווחים עליהן יהיו חולשות אמיתיות ומועילות, והמנעו מהגזמה בחשיבות / תיאור החולשה. כלל האצבע שלי הוא שאשתדל ש**לא לדווח על חולשה שלא אוכל להשתמש בה בפועל**.

על המחבר

אייל איטקין: חוקר אבטחת מידע בקבוצת המחקר של חברת Check Point. עוסק בעיקר בתחומי ה-Embedded והתקשורת, בד"כ שבירת פרוטוקולי רשת ששמם מורכב רק מ-3 אותיות (FAX, I2P, PTP, RDP וכו'). מפעם לפעם עוסק בציד חולשות בפרויקטי Bug bounty, ברשימה הכוללת את: Microsoft, PHP, Perl (C)Python, (C)Ruby, MRuby ועוד.

- **בלוג אבטחה:** https://eyalitkin.wordpress.com
- **אימייל:** eyal.itkin@hotmail.com
- **Twitter:** @EyalItkin