שימוש בטוח בקוד-פתוח

גיל קיני

gil@trinity.co.il

09-7677880

מי אני? ולמה אני פה?



- **גיל קיני**
  - מעל 20 שנות ניסיון באבטחה, תוכנה וניהול
  - זמן-אמת, מערכות משובצות, תקשורת ו- VoIP
  - תהליכי פיתוח מכווני אבטחה כולל הנדסת-מערכת, תיכון וארכיטקטורה מאובטחת
  - פיתוח אלגוריתמים
  - שיפורי ביצועים למערכות קיימות
  - מחקרי אבטחה וכשלים
  - אבטחה ברמת מערכת

gil@trinity.co.il

מטרות ההרצאה

אספקט האבטחה בשימוש בחבילות קוד-פתוח



- להעלות מודעות
- להוציא משאננות
- להסביר ולכוון

gil@trinity.co.il

Agenda

- שימוש בטוח ואחראי
- אבטחה מתחילה בתובנה
- ניהול פגיעויות מוכרות
- ניהול פגיעויות לא-מוכרות
- בחירת חבילת קוד-פתוח המתאימה לכם

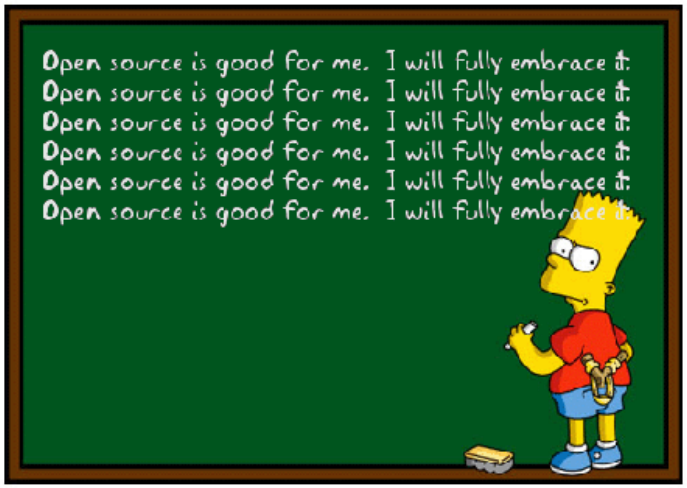
gil@trinity.co.il

שימוש בטוח ואחראי

קוד פתוח מועיל לחברות ...

... אם משתמשים באופן אחראי ובטוח

- אחראי – לפי הכללים והרישיונות
  - פנימיים וחיצוניים
- בטוח
  - יותר או פחות מקוד סגור?
  - כללי שימוש בטוח



gil@trinity.co.il

מהי פגיעות אבטחה?

- Exploit
  - Code that is designed to take advantage of a vulnerability
- Security Vulnerability

According to NIST:

> *"Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source."*

Source: http://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf

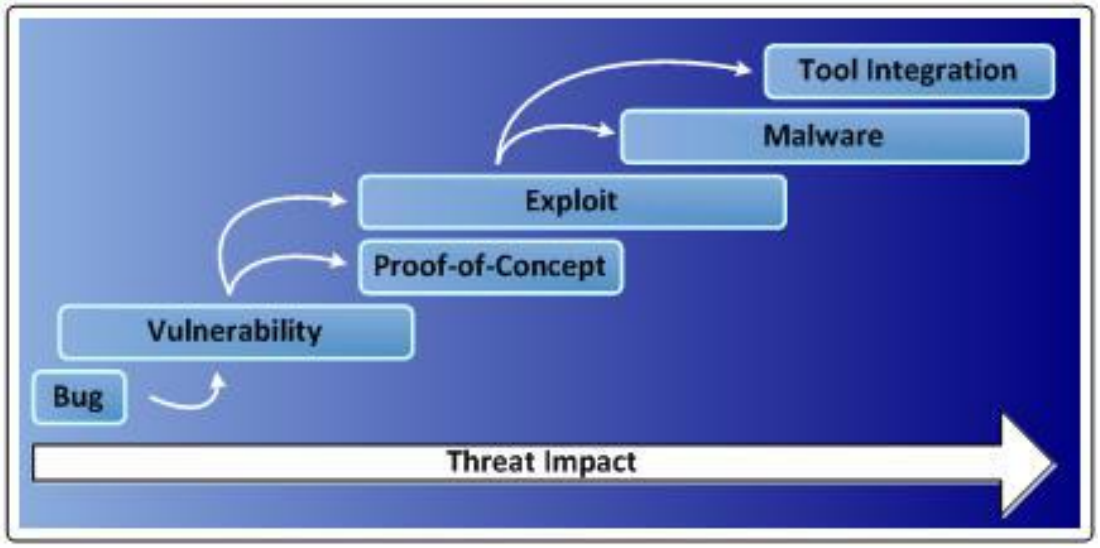
According to Microsoft:

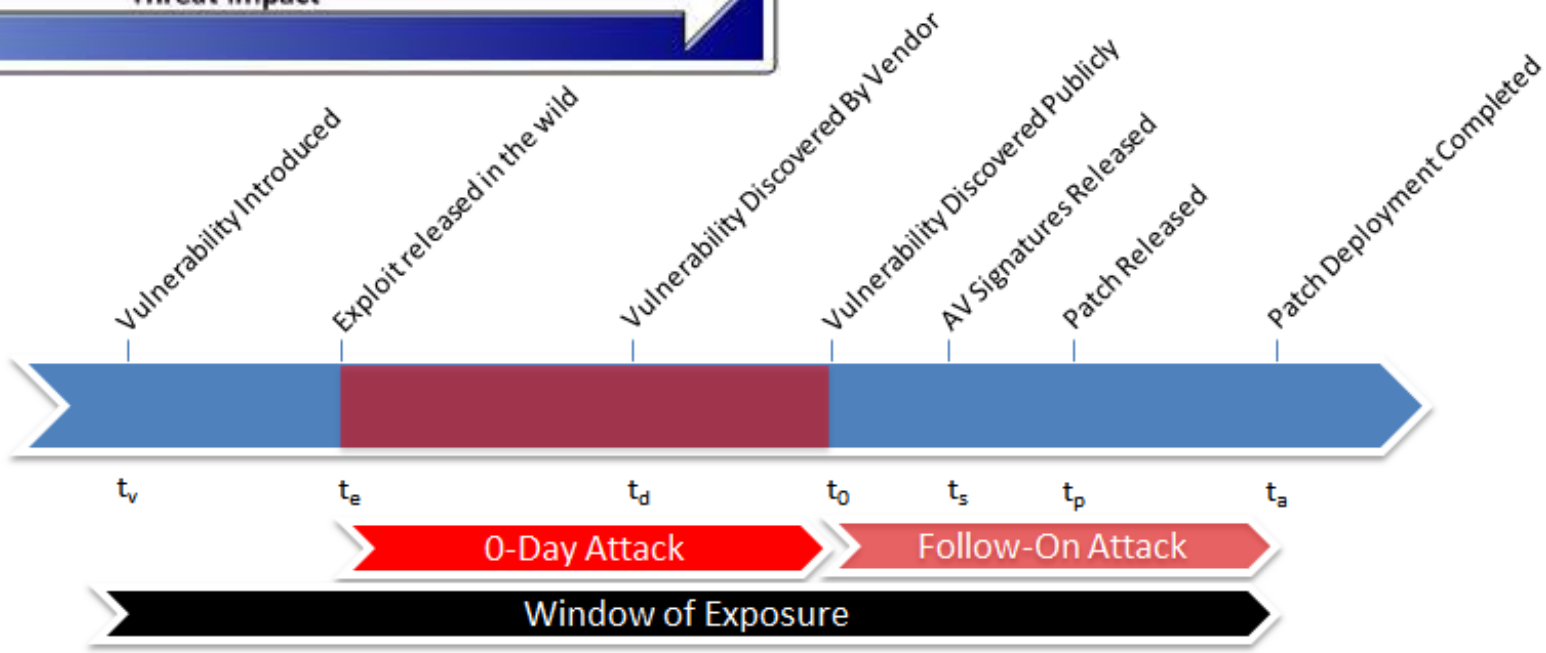
> *"A security vulnerability is a weakness in a product that could allow an attacker to compromise the integrity, availability, or confidentiality of that product."*

Source: http://technet.microsoft.com/en-us/library/cc751383.aspx

Trinity
Software & Beyond

From Vulnerability to Exploit

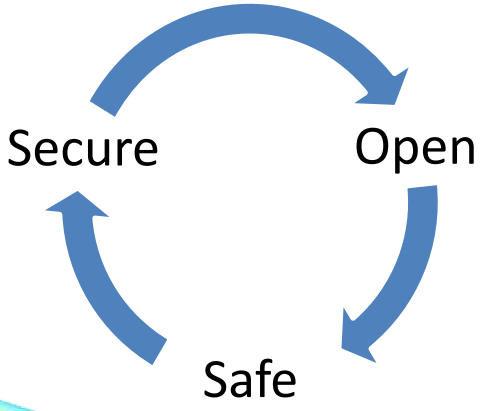




gil@trinity.co.il

יותר בטוח או יותר פתוח?

- בעולם אבטחת המידע והתוכנה ניטש ויכוח סוער על יתרונות וחסרונות האבטחה של קוד-פתוח
- על אף הגילויים האחרונים של פגיעויות אבטחה בקוד-פתוח, מסתמנת מגמה כי הקוד-הפתוח הינו הפתרון ולא הבעיה



gil@trinity.co.il

יותר בטוח או יותר פתוח 2

- הטיעונים לכל צד הינם סובייקטיביים
  - לא נצפה כל קשר בין מספר הפגיעויות באפליקציה והסטטוס קוד פתוח/סגור שלה
- גורמים משמעותיים הם לרוב
  - גיל/ותק האפליקציה
  - רמת ההשכלה של המפתחים
  - מספר המפתחים והבודקים
  - מטרות הפרויקט
  - מורכבות הקוד



gil@trinity.co.il

יתרונות האבטחה של קוד-פתוח

- יותר אנשים חשופים לקוד
  - סקר, בקרה ותיקון
  - יכול להוביל לגילוי מהיר ומניעת הכנסת בעיות חדשות
- תוכנה סגורה מכריחה את המשתמש
  - לקבל את רמת האבטחה שהיצרן מוכן לספק
  - לקבל את קצב התיקונים והעדכונים

gil@trinity.co.il

יתרונות 2

- למשתמש הסופי של קוד-פתוח יש יכולת לשנות את הקוד
  - לשיפור ותוספת של יכולות אבטחה
  - בקרה ותיקון בעיות קיימות



gil@trinity.co.il

יתרונות 3



- גישה לקוד הקומפיילר
  - דלת-אחורית בקומפיילר (Thompson hack)
  - קצת פרנואידי?
  - או לא!?



- עקרון קרקהופס (Kerckhoffs)
  - מערכת הצפנה צריכה להיות בטוחה גם אם כל החלקים במערכת ידועים, פרט למפתח הסודי
  - פתוח זה בטוח

gil@trinity.co.il

Trinity
Software & Beyond

תיקונים מהירים בקוד-פתוח

Faster Fixes for Security Flaws

Vendor	Total Days of Hacker Recess	Total Advisories	Days of Recess Per Advisory
Leading Open Source Vendor	348	31	11.2
Leading Closed Source Vendor A	982	61	16.1
Leading Closed Source Vendor B	716	8	89.5

חסרונות האבטחה של קוד-פתוח

- יותר אנשים חשופים לקוד
  - כולל תוקפים פוטנציאליים
  - יכול להוביל לגילוי מהיר, לא בהכרח ע"י הנכונים
  - כל אחד יכול לתרום קוד – הוספת פגיעויות במזיד
- פתיחות הקוד אינה מבטיחה בקרה
- ריבוי מבקרי-קוד עשוי ליצור תחושת בטחון שווא
  - זו לא הכמות אלא האיכות

gil@trinity.co.il

אבטחה מתחילה בתובנה

מה מכיל הקוד שלכם?



- מיפוי – השלב ראשון בטיפול
  - מל"מ
- מקורות הקוד
  - פנימי / חיצוני
  - עובדים / קבלנים / ספקים
- פרויקטים וחבילות מוכרות
- הפתעות המתווספות בזמן הבילד

gil@trinity.co.il

אחריות משפטית

- פורסם מידע על פגיעות / תיקון
  - מתי אתם מתעדכנים?
  - איך משפיע עליכם / לקוחותיכם?
  - מה אתם עושים?



gil@trinity.co.il

ניהול ידני או אוטומטי?

- של מקורות הקוד
- של החבילות המצויות בקוד
- של ממצאי האבטחה
- של חדשות וחידושי האבטחה



gil@trinity.co.il

מעגלי ושכבות אבטחה



- נקודות בקרה בתהליכי הפיתוח
  - תוכנית 8 השלבים
- בקרה וביקורת
- בדיקות ממוקדות



gil@trinity.co.il

הטמעה בטוחה

- נקודות בקרה בתהליכי הפיתוח



gil@trinity.co.il

ניהול פגיעויות מוכרות

מתודולוגיות סריקה וגילוי

- סריקה חיצונית
  - Rapid7 Nexpose
- סריקת קוד – אוטומטי או ידני

↓ גילוי / דיווח / רישום של קוד-פתוח

↓ SW Bill Of Materials (BOM)

↓ חיפוש והצלבה מול מסדי פגיעויות

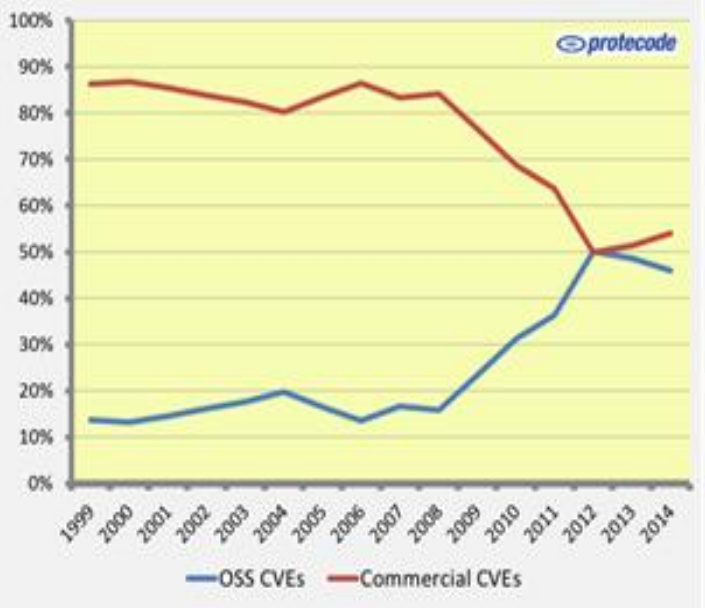
↓ ניתוח רלוונטיות ומשמעות



gil@trinity.co.il

National Vulnerability Database

- US government repository since 2005
- Common Vulnerabilities and Exposures (CVEs)
  - International in scope & free for public use
  - A dictionary of publicly known information security vulnerabilities and exposures
- Currently more than 60K CVEs
  - Both open source and commercial software



gil@trinity.co.il

NVD Nomenclature

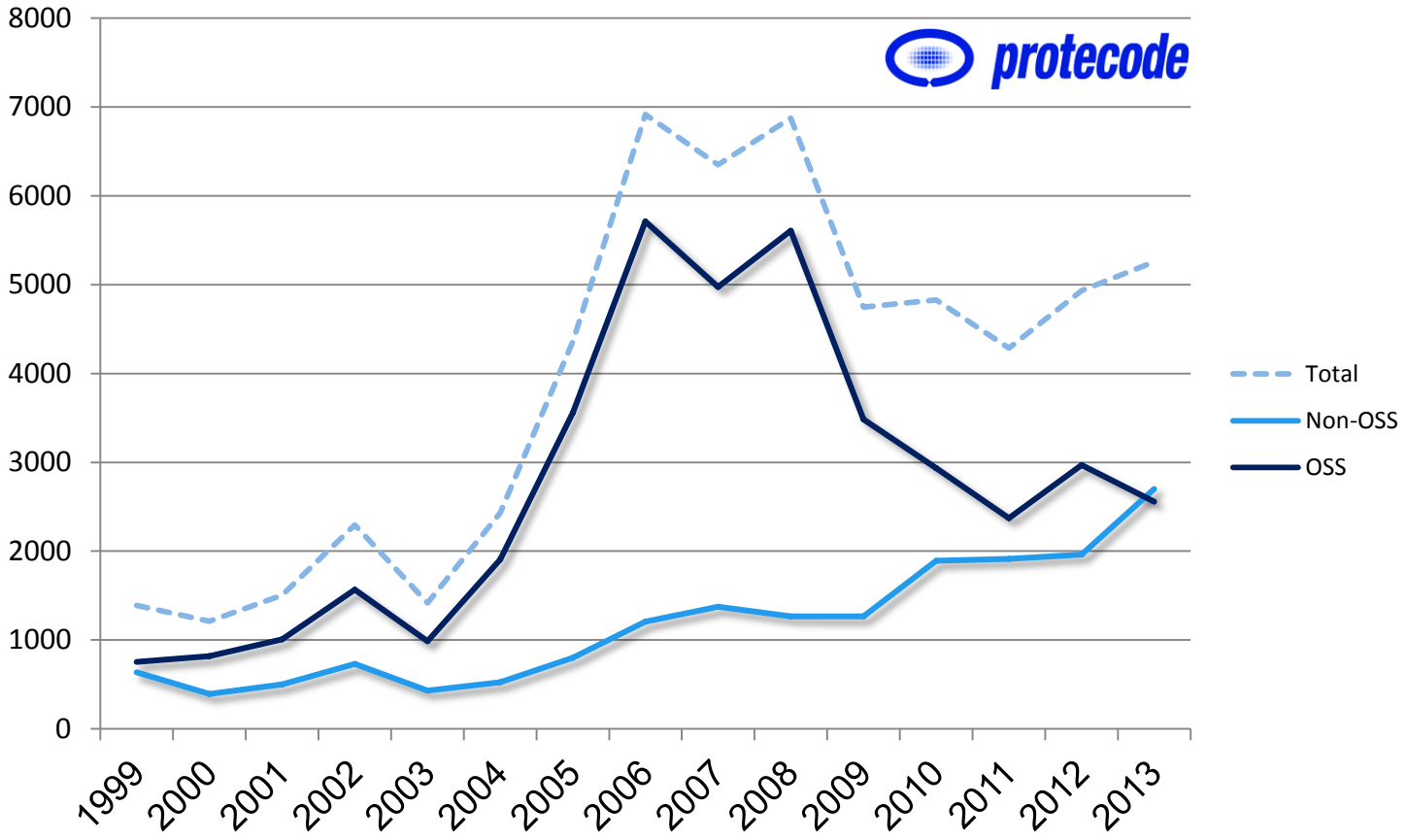
- CVE: Common Vulnerabilities and Exposures
  - Known Vulnerability
- CCSS: Common Configuration Scoring System
  - A severity
- CPE: Common Platform Enumeration
  - An owner, product and version
- CCE: Common Configuration Enumeration
  - A system configuration
- CWE: Common Weakness Enumeration
  - A code, design or architecture weakness



gil@trinity.co.il

Trinity
Software & Beyond

Security Vulnerabilities (CVEs)



gil@trinity.co.il

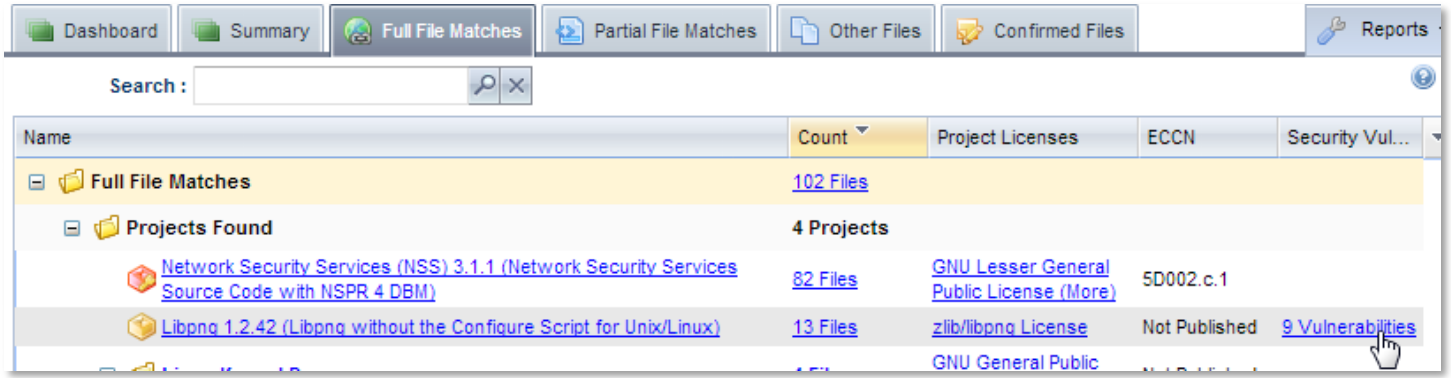
מיפוי פגיעויות

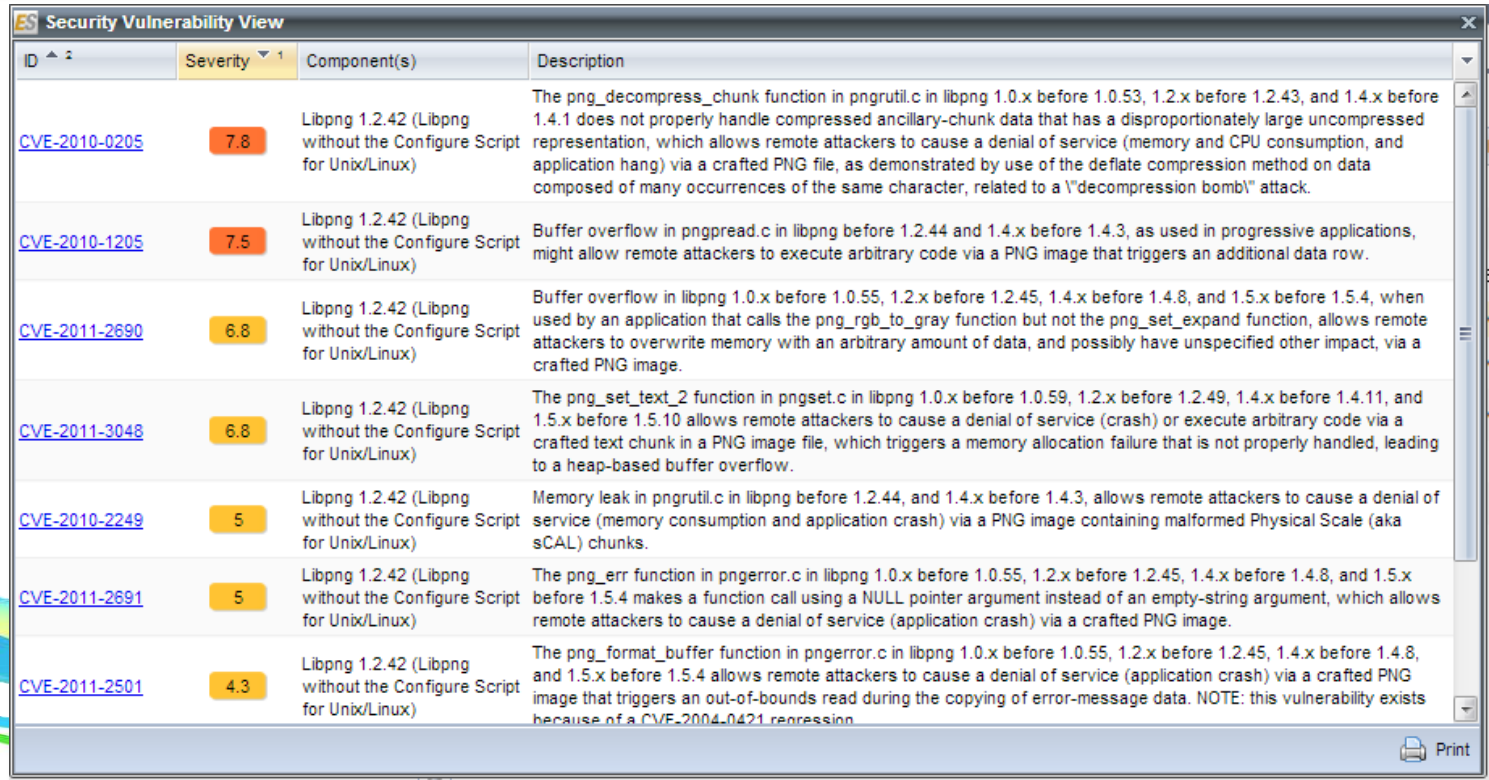
- ממוקד
  - נדיר
- לפי פרויקט קוד-פתוח
  - דורש מיפוי פגיעות לתת-חבילה ו/או רמת הקובץ

gil@trinity.co.il

Trinity
Software & Beyond

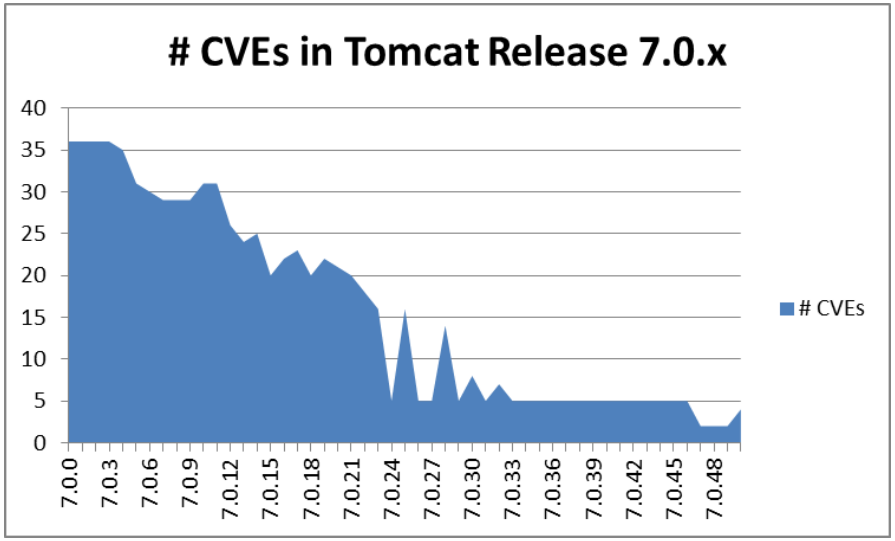
Sample Security Vulnerabilities Report





gil@trinity.co.il

מה עושים עם המידע?



- האם רלוונטי?
- שדרוג
- תיקון
- מציאת אלטרנטיבה
- יהיה בסדר!



gil@trinity.co.il

פיקוח מתמשך

- חזרה תקופתית על הסריקות
  - עם נתונים מעודכנים
- המשך מעקב אחר דיווחים על החבילות בשימוש
- ניתוח מחודש על משמעות הפגיעויות



gil@trinity.co.il

ניהול פגיעויות לא-מוכרות

Zero-Day

יגעת ומצאת – תאמין

- ניתוח סטטי
  - לקוד
  - למבנה (ארכיטקטורה)
- פאזינג
  - לממשק משתמש
  - לתקשורות
    - שליחת הודעות שגויות, באופן מבוקר
    - במטרה להכשיל את המערכת תחת בדיקה
    - לשם מציאת נקודות כשל

gil@trinity.co.il

תיקון ודיווח

- תלוי פרויקט
  - איך ומי מתקן ומדווח
  - העברת התיקון למובילי הפרויקט
- מודלים / גישות / פילוסופיות
  - גילוי מלא
  - ללא גילוי
  - גילוי אחראי = Responsible Disclosure

gil@trinity.co.il

תקנים בדיווח

- ISO/IEC 30111:2013
  - Guidelines for how to process and resolve potential vulnerability information in a product or online service
  - Applicable to vendors involved in handling vulnerabilities
- ISO/IEC 29147:2014
  - Guidelines for the disclosure of potential vulnerabilities in products and online services
  - Details the methods a vendor should use to address issues related to vulnerability disclosure
  - Applicable to vendors who respond to external reports of vulnerabilities in their products or online services

gil@trinity.co.il

בחירה מוכוונת אבטחה

אבטחה – גורמים משמעותיים

- גיל/ותק האפליקציה
- רמת ההשכלה של המפתחים
- מספר המפתחים והבודקים
- מטרות הפרויקט
- מורכבות הקוד



gil@trinity.co.il

רישוי

- רישיונות
  - רישיון כפול או תמיכה בתשלום
- פטנטים
- איך משפיע על גילוי ותיקון בעיות אבטחה
  - דיווח מאחרים
  - מכם
- ישנם פרויקטים עם מדיניות ייחודית
  - Rails, PostgreSQL, Oracle, etc.

gil@trinity.co.il

בשלות ובסיס-משתמשים



- בדיקה ברשימות
  - "generally recognized as mature" (GRAM)
  - "Enterprise Open Source Directory"
- בידקו מה כלול בחבילות לינוקס
  - דביאן במיוחד מכיל מגוון רחב של פרויקטי קוד-פתוח
- נפוצות
- פעילות ותחזוקה

gil@trinity.co.il

אבטחה

- בדיקה ברשימות
  - "generally recognized as safe" (GRAS) OSS/FS
- יוזמת ה- DHS לשיפור אבטחת קוד-פתוח
- בדיקת וספירת פגיעויות
  - למשל ב-MITRE's CVE
- חפשו "פגיעויות בלתי-נסלחות"
  - הן כמו קנריות במכרה פחם
- לאום המפתחים



gil@trinity.co.il

שאלות? טענות? מענות?

- איפה אפשר ללמוד עוד על הנושא?
- בואו נדבר ...

תודה !

גיל קיני

Gil @ Trinity.co.il **09-7677880**