Fallstricke in der FLOSS-Nutzung- & Entwicklung

Lehren aus fremden Fehlern

Till Zimmermann, Mat.Nr. 956242

Seminar ”Open-Source Software” an der Universität Osnabrück, SoSe2018

Linux is a cancer that attaches itself in an intellectual property sense to everything it touches [...] — *Steve Ballmer*

Einführung

Einführung - Überblick

Einführung - Ziele

- Sinnvoll zu erreichen:
  - Schwierigkeiten bedenken
  - Fehler verstehen
  - Verständnis entwickeln
- Nicht sinnvoll:
  - Bösen Willen unterstellen
  - Unfähigkeit unterstellen
  - ”Company-Shaming”

Lizenz lesen & verstehen

Lizenz lesen & verstehen - AVMs FritzBox-GPL-Verletzung

- netfilter/iptables: GPLv2
- Fritzbox - Router: Abgewandelte Kopie
- ”Surf-Sitter DSL” - Kinderschutz-Filter
- AVM: Urheberrecht verletzt

> *Für Sammelwerke bestimmt §2 GPL, dass Werke, die Open Source Software enthalten, als Ganzes den Bedingung der GPL unterliegen [...] Danach stehen der Klägerin an der Firmware als Ganzes - und so sind ihre Anträge zu verstehen - keine urheberrechtlichen Unterlassungsansprüche zu.* — *Landgericht Berlin, Urteilsbegründung*

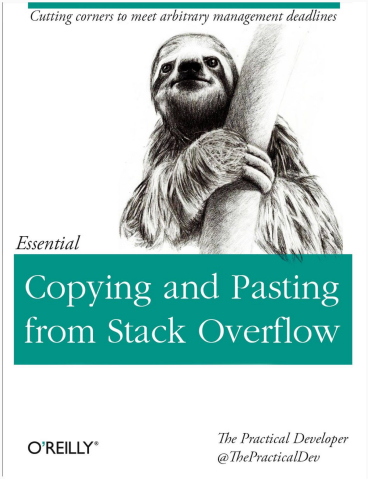
- Wir lernen: Implikationen der FOSS-Nutzung bedenken

Lizenz lesen & verstehen - GPL-Missbrauch in Linux

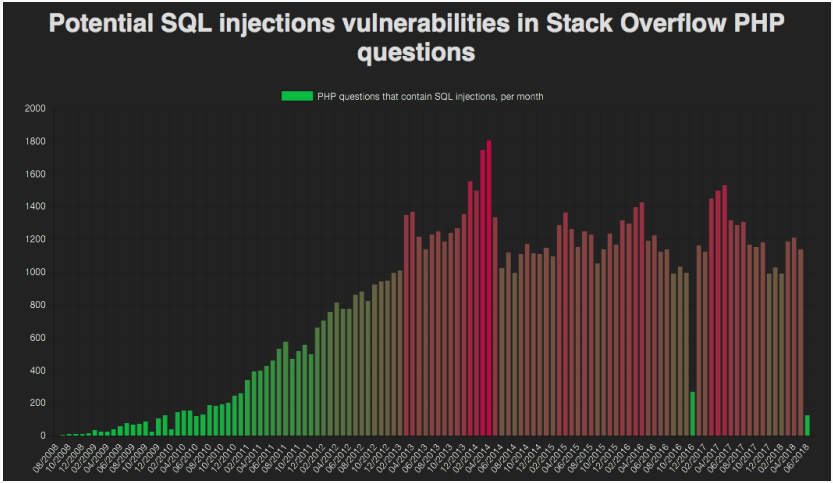
- 2001: Patrick McHardy entwickelt netfilter (Linux-Kernel)
- 2016: geschätzt* 50 Abmahnungen
- Forderung: bis zu 250.000€ Ordnungsgeld
- Probleme
  - Software profitiert nicht
  - Unterschiedliche Ziele d. Entwickler
  - Rechtliche Unsicherheit verhindert Linux-Nutzung

Sicherheitsimplikationen in FOSS

Sicherheitsimplikationen in FOSS - Stackoverflow - C&P



Sicherheitsimplikationen in FOSS - Stackoverflow - C&P



Sicherheitsimplikationen in FOSS - Stackoverflow - C&P

- Offener Code kann kopiert werden
- häufig Stackoverflow (beliebtes ”Forum”)

Sicherheitsimplikationen in FOSS - Stackoverflow - C&P

- Untersuchung 2017:
  - ˜4000 Sicherheitsrelevante Code-Snippets
  - ˜1100 davon unsicher
  - 1.3 Millionen Android-Apps
- 15,4% enthalten **sicherheitsrelevanten**, kopierten Code
- 97,9% davon mit Sicherheitslücken
- Häufig keine Hostname-Verifizierung bei TLS
- Veraltete/Unsichere Hash-Mechanismen

Sicherheitsimplikationen in FOSS - Stackoverflow - C&P

- Offener Code kann kopiert werden
- häufig Stackoverflow (beliebtes ”Forum”)
- Probleme:
  - keine Qualitätskontrolle
  - kein Verständnis für eigenen Code

Sicherheitsimplikationen in FOSS - Heartbleed - Herkunft

- Feb. 2012 – RFC 6520: Vorschlag für Heart**beat**
- Referenzimplementierung in OpenSSL

Sicherheitsimplikationen in FOSS - Heartbleed - Herkunft

- Feb. 2012 – RFC 6520: Vorschlag für Heart**beat**
- Referenzimplementierung in OpenSSL

```
+        /* Read type and payload length first */
+        hbtype = *p++;
+        n2s(p, payload);
+        pl = p;
```

Sicherheitsimplikationen in FOSS - Heartbleed - Herkunft

- Feb. 2012 – RFC 6520: Vorschlag für Heart**beat**
- Referenzimplementierung in OpenSSL

```
+        /* Read type and payload length first */
+        hbtype = *p++;
+        n2s(p, payload);
+        pl = p;
```

- 07.04.2014: Bekanntwerden/Fix
- Probleme:
  - Unnötiger Code wird hinzugefügt, da 'kein Schaden'
  - u.U. keine gemeinschaftliche Kontrolle*

Organistorische Probleme

Organistorische Probleme - NPM

- NodeJS Packet Manager - Datenbank für freien Code
- Entwickler von ”left-pad” nimmt Paket offline
- Zigtausend Projekte laufen nicht mehr
- Probleme:
  - Abhängig von Öffentlichkeit
  - Verlass auf einen dritte Partei

Organistorische Probleme - OracleJava

- Java: Dual-Lizensiert, GPL/kostenpflichtig
- Googles Wunsch: Android Java-**Kompatibel**, ohne GPL-Verpflichtungen
- August 2010: Oracle reicht Klage ein wg. Copyright-Verletzung
- Gerichtsurteile 2012, 2014, 2015, 2016 & 2017
- Letzter Stand: Google hat Copyright verletzt
- Probleme
  - Technisch Schwierig: Nachbau nicht erlaubt
  - Unsicherheit bei Entwicklern

Zusammenfassung

Zusammenfassung - Lehren

- Bei der Nutzung
  - Folgen der Lizenzwahl bedenken
  - Juristische Absicherung
  - Abhängigkeit von Verfügbarkeit/Weiterentwicklung
- Bei der Entwicklung
  - Aktoren mit anderen Hintergründen bedenken
  - Eigener Verantwortung bewusst sein
  - Sicherheitsprüfung etablieren/nicht blind vertrauen

Referenzen

- https://laurent22.github.io/so-injections/
- http://law.siu.edu/_common/documents/law-journal/articles-2016/8%20-%20Unni%20Article%20Proof%205%20FINAL%20-%20sm.pdf
- https://bib.irb.hr/datoteka/771573.Technological_risks_of_Open_Source_Software_Adoption_in_the_organizational_context.pdf
- https://www.heise.de/newsticker/meldung/FSFE-wirft-AVM-GPL-Verletzung-vor-1263357.html
- https://fsfe.org/activities/ftf/kg-avm-vs-cybits.pdf
- https://fsfe.org/activities/ftf/avm-gpl-violation.de.html
- https://fsfe.org/activities/ftf/lg-urteil-20111118.pdf
- https://www.heise.de/newsticker/meldung/Linux-Klaeger-McHardy-zieht-Antrag-gegen-Elektronik-Hersteller-zurueck-3988556.html
- https://opensource.com/article/17/8/patrick-mchardy-and-copyright-profiteering
- http://research.njms.rutgers.edu/m/it/Publications/docs/Heartbleed_OpenSSL_Vulnerability_a_Forensic_Case_Study_at_Medical_School.pdf
- https://saschafahl.de/papers/stackoverflow2017.pdf