Beschreibung des Einsatzes von Werkzeugen für Sicherheitstests in Open Source

Referent: Michael Osipov, ossipov@inf.fu-berlin.de
Betreuer: Prof. Dr. Lutz Prechelt, Martin Gruhn
Seminar: Beiträge zum Software Engineering
FU Berlin
17. Juli 2008

Motivation

Freie Universität Berlin

- Fast täglich erscheinen Meldungen über sicherheitskritische Vorfälle auf heise Security, Secunia, SecurityFocus, CVE oder US-CERT
- Anzahl der Meldungen hat sich in den letzten 5 Jahren fast verdoppelt (Secunia)
- Open Source-Software ist Teil dieser Meldungen
- OSS wird aktiv auch von Unternehmen eingesetzt und weiterentwickelt
- Auch Unternehmen haben also Interesse an einem sicheren Produkt
- Was tut dann die OSS-Gemeinde um diese Meldungen zu vermeiden bzw. zu reduzieren?

Fallstudie – Forschungsfragen

Freie Universität Berlin

- Zweck der Studie ist zwei Fragen zu klären
- Welche Arten von Werkzeugen werden zur Qualitätssicherung bezüglich Sicherheit eingesetzt und welche wurden verworfen?
  - Über Beschreibung einer Zuordnung zu einer Werkzeugart
- Wie werden sie eingesetzt und welchen Mehrwert ergibt sich für die Sicherheit?
  - Deskriptive Darstellung, wie Werkzeuge im Prozess eingesetzt werden, was sie konkret leisten und was sie verbessern können
- Also: Für wiederkehrende Probleme Werkzeuge nutzen (Automatisierung)

Fallstudie – Vorgehen

Freie Universität Berlin

- Untersucht wurden 15 Web-Informationssysteme (Server und Webapplikationen)
- Auswahlkriterien waren u.a. Popularität, Projektgröße, Zugängigkeit der Ressourcen
- Von den 15 haben sieben verwendbare Informationen geliefert, sprich der Kontakt ermöglichte die Daten schneller zu verstehen und zu analysieren
- Informationen wurden bezogen aus Homepage/Wiki, Repo/Skripte, Mailinglisten/Foren, Kontakt mit Projekt
- Im weiteren Verlauf wurden untersucht: Joomla!, JAMWiki, Bugzilla, MediaWiki, Apache Tomcat, Gallery, phpBB

Fallstudie – FF1

Freie Universität Berlin

- Welche Werkzeuge werden benutzt?

Kategorie	***Projekt***	***Werkzeug***
Musterabgleich, Datenflussanalyse (selbst entwickelt)	Bugzilla	runtests.pl
Modultests	JAMWiki	JUnit
Modultests	Gallery	PHPUnit
Modultests, Fuzzer, Leistungsanalyse	Gallery	Burp Suite
Fuzzer (selbst entwickelt)	MediaWiki	fuzz-tester.php
Musterabgleich (selbst entwickelt)	phpBB	grep-Erweiterung

- Mehrere Projekte nutzen selbst entwickelte Werkzeuge
  - Sie sind speziell für den Bedarf des Projektes angepasst bzw. geschrieben worden

Fallstudie – FF1

Freie Universität Berlin

- „Fertige“ Werkzeuge wurden aber auch verworfen, weil
  - sie eine hohe Rate an *false positives* liefern (Gallery, Tomcat, Joomla!, phpBB)
  - sie bereits bekannte Fehler in alten Codebasen nicht auffinden können (Tomcat, phpBB)
  - sie neue/noch unbekannte Fehler auch nicht auffinden (Tomcat, phpBB)
  - Auch kommerzielle Werkzeuge versagen (phpBB/Acunetix, Tomcat/Fortify SCA)
  - Sie durch ihre verallgemeinerte Funktionalität an den falschen Stellen testen bzw. gar nicht an die zu testenden Stellen vordringen (Joomla!, phpBB)

Fallstudie – FF2

Freie Universität Berlin

- Wie werden diese selbst entwickelten und fertigen Werkzeuge eingesetzt und welchen Mehrwert ergibt sich für die Sicherheit?
- Modultests (Gallery, JAMWiki):
  - Sind zwar keine Sicherheitstestmaßnahmen, eher Frameworks
  - Aber: Sinnvoll für Regressionstests, gefundene Schwachstellen sollen nicht noch einmal auftauchen
  - Oder: Um vermeintliche programmatische-präventive Maßnahmen zu testen
  - MT müssen aber sinnvoll gestaltet werden, damit sie einen Effekt haben

Fallstudie – FF2

Freie Universität Berlin

- Statische Analyse (Bugzilla, phpBB):
  - Bugzilla (Perl-basierendes Werkzeug): Suche nach gefährlichen Mustern und Zugriffen; jeder Commit muss Testlauf durchführen und bestehen
  - phpBB (grep-basierendes Werkzeug): Suche nach gefährlichen Mustern und Zugriffen
- Fuzzer (MediaWiki, Gallery):
  - MediaWiki (PHP-basierendes Werkzeug): funktionsreiches Werkzeug, das den WikiSyntax-Parser und DB mit „kaputten“ Eingaben bombardiert; bereits mehrere Schwachstellen entdeckt und behoben
  - Gallery (Burp Suite): HTTP-seitige Tests einzelner PHP-Skripte mit Modul- und Fuzztests

Weitere Beobachtungen

Freie Universität Berlin

- Es gibt noch andere interessante Ergebnisse und Gegensätze bei der Fallstudie aufgrund der negativen Erfahrungen mit Werkzeugen
- Mehrere Projekte erachten Code Reviews als sehr effizientes bzw. effizienteres Mittel im Vergleich zu Werkzeugen
- Apache Tomcat:
  - Führt ein 3x Review von jedem Commit durch
  - Extern durchgeführte Commits können eine sehr hohe Sicherheitsabdeckung erreichen
- phpBB/Gallery:
  - Führen selbst regelmäßig interne Code Reviews durch
  - Verlassen sich auch auf bezahlte Reviews (SektorEins, Gotham Digital Science)

Herausforderungen

Freie Universität Berlin

- Analyse vieler Daten langwierig ohne externe Hilfe
- Vollständige Analyse zeitintensiver als gedacht, da
  - Daten individuell für jedes Projekt als Kontakt aufbereitet werden müssen
  - Emailwechsel sich generell hinziehen kann, bis man überhaupt eine Antwort bekommt

Kritik

- Einseitige Sicht, da nur Webinformationssysteme
- Andere Projekte wie Betriebssysteme (OpenBSD) oder Verschlüsselungssysteme (GPG, Open[SSH|SSL]) wären prädestinierte Forschungsobjekte
- Eine Nichtuntersuchung einer Projektes bedeutet **nicht**,
  - dass dieses Projekt nichts unternimmt
  - sondern dass die Informationen nicht hinreichend genug analysiert werden konnten

Zusammenfassung

Freie Universität Berlin

- Selbst entwickelte Werkzeuge, Modultests und zusätzlich Code Reviews sind hier die Favoriten
- Für gute Sicherheitstests ist eine Kombination auf verschiedenen Werkzeugen/Techniken notwendig (Gallery, phpBB)
- **Aber**: Einige Projekte sind der Meinung, dass Sicherheit auch eine Sache der Erfahrung ist (phpBB, Tomcat, Gallery)

Ausblick

- Situation weiter beobachten, ob Trends sich abzeichnen oder Weiterentwicklungen stattfinden
- Auf Basis dieser Studie eine tiefergehende Studie durchführen, die
  - fähige Programmierer in Projekte wie phpBB, MediaWiki injiziert
  - den Testprozess in dem Projekt beschreibt
  - die Werkzeuge verbessert und für andere Projekte verallgemeinert ohne auf Funktionalität zu verzichten
- Werden auch kostenpflichtige Werkzeuge besser bzw. wird es neue, bessere Werkzeuge geben?
- Es bleibt weiter spannend...

Vielen Dank!