Beschreibung des Einsatzes von Werkzeugen für Sicherheitstests in Open Source

Referent: Michael Osipov, ossipov@inf.fu-berlin.de
Betreuer: Martin Gruhn, mgruhn@inf.fu-berlin.de
Seminar: Beiträge zum Software Engineering
FU Berlin
24. April 2008

Übersicht

Freie Universität Berlin

1) Einleitung
    1) Was bedeutet Sicherheit?
    2) Warum interessiert uns Sicherheit?
    3) Was wollen wir herausfinden?
2) Überblick
    1) Werkzeugkategorien
    2) Testdurchführung
3) Fallstudien
    1) Was wollen wir herausfinden?
    2) Wie wird untersucht?
    3) Wer wird untersucht?
4) Selbstversuche
    1) Anwendung von Werkzeugen auf x Projekte
5) Diskussion

Einleitung

- Was bedeutet Sicherheit?
  - Im Deutschen gibt es nur Sicherheit für Safety *und* Security
- Begriffserklärung:
  - Safety: Gewährleistet angemessene Reaktion bei einer unbeabsichtigten Aktion (z.B. falsche Parametereingabe)
  - Security wird in der Literatur als Verbindung aus Vertraulichkeit, Integrität und Authentizität bezeichnet (vgl. Eckert, 2004)
  - Security bedeutet Schutz gegen böswillige Dritte

Einleitung (2)

- Im Rahmen von verteilen bzw. Web-Projekten genügt:
  - Informationsschutz: Lesen und modifizieren durch Dritte soll unterbunden werden (z.B. erh. Zugriff durch SQL-Injection)
  - Funktionsschutz: Dritte soll die Möglichkeit genommen werden Funktionen zum Nachteil des Systems und deren Benutzern zu ändern (z.B. Parameter Tampering)
  - Systemschutz: Der Zugriff zum System soll so gesichert sein, dass Dritte keine höheren Rechte erhalten bzw. das System abschalten können (DoS: Sun Bug ID 6339385: Shutdown der JVM durch falsches JDWP-Handshake)

Schlussfolgerung: Security ist kein Addon zum Nachpatchen, es muss bereits in die Entwicklung und Planung einfließen!

Einleitung (3)

Freie Universität Berlin

- Warum interessiert mich Sicherheit (in Open Source)?

- Zum einen tägliche Meldungen wie:
  - Millionenfache Ausbreitung von virtuellen Würmen mit immensem Schaden
  - Datendiebstahl (Kreditkarten, Sozialdaten, etc) bei Finanzunternehmen, Militär, Regierungen
  - Verdopplung der Meldungen und der höchkritischen Lücken in den letzten 5 Jahren (vgl. Secunia)

Einleitung (4)

- Zum anderen Umgang mit Sicherheit in Unternehmen:
  - IT-Sicherheit wird als Erfolgskomponente betrachtet
  - Security Awareness ist je nach Größe und Sektor minimal bis stark ausgeprägt
  - Es wird an der falschen Stelle gespart (Wissens, Geld, Zeit, andere Prioritäten)
  - Mittlerweile gibt es ein Bundesamt für Sicherheit in der Informationstechnik (BSI), das sich mit der Thematik beschäftigt

Einleitung (5)

Freie Universität Berlin

- Was hat das alles mit Open Source zu tun?
  - Open Source wird vielfach von Unternehmen, Regierungen und Privatpersonen eingesetzt
  - Open Source versucht sich entgegen Security by Obscurity zu setzen (SbO Gegensatz zu Kerckhoff, 1883)
  - Best Practices aus der Open Source-Welt kann auch für Closed Source-Projekte von Vorteil sein

Schlussfolgerung: Jeder kann direkt oder auch indirekt von Open Source profitieren!

Einleitung (6)

- Was will ich herausfinden?
  - Wie kann man Werkzeuge sinnvoll einteilen?
    - Ziel: Kategorien geben Aussagen zur Einsatzfähigkeit ab
  - Werkzeuge und Testverfahren in OS-Projekten:
    - Werden Werkzeuge eingesetzt?
    - Welche (frei verfügbare, selbst geschriebene) und wie?
    - Wie ist der Einsatzaufwand?
    - Sinnvolle Ergebnisse nach Testeinsatz?
    - Wenn kein Werkzeugeinsatz erfolgt => Warum?
  - Selbstversuche: Einsatz von Werkzeugen auf Projekte
    - Ziel: Selbst herausfinden, ob die obigen Ergebnisse nachvollziehbar sind?!
    - Grenzen derzeitiger Werkzeuge
    - Empfehlung an andere

Überblick

- Versuch der Erstellung von Kategorisierungs- und Beurteilungsoptionen für Werkzeuge
- Im Klartext: Darlegung der Einsatzmöglichkeiten und Grenzen einer Kategorie
- Achtung: Werkzeuge werden einer Kategorie zugeordnet, d.h. Aussagen werden **nicht** über Werkzeuge sondern Kategorien getroffen

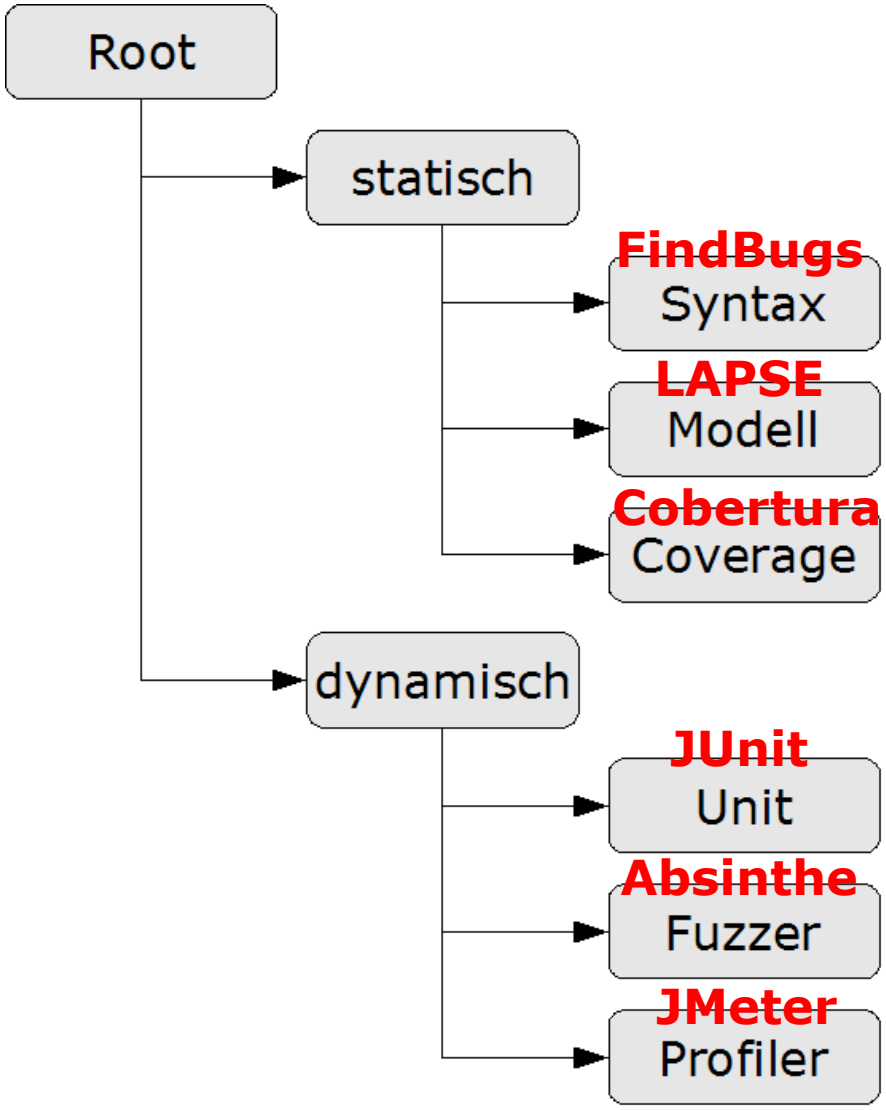
Überblick (2)

- Kategorisierung:
  - Beschreibt Möglichkeiten und Grenzen
  - Beurteilung nach:
    - Leistungsfähigkeit in Bezug auf Sicherheit
    - Notwendiges Verständnis für Sicherheit
    - Einsatzaufwand
    - Informationsgehalt des Ergebnisses in Bezug auf Sicherheit

- Durchführung:
  - Ausführungempfehlungen nach:
    - Manuell
    - Integrierbar (Ant, Maven, Shellskripte, etc.)
    - Eigenes System

Überblick (3)

- Kategorisierung:
  - Zwei große Kategorien
    - Statisch:
      - Keine Ausführung notwendig
      - Kein Vorbereitungszeit notwendig
      - Analysiert gesamten Programmcode
    - Dynamisch:
      - Testen ein Codefragment durch Ausführung mit vorgegebenen Werten
      - Testfälle müssen erstellt werden
      - Codeabdeckung bedeutet für alle Teile Testfälle erstelle



Überblick – Fazit

Freie Universität Berlin

- Keine Kategorie kann als „silver bullet“ bezeichnet werden
- Aufwand steht gegen Nutzen => Abschätzung notwendig
  - Problemstellung
  - Erstellungs-/Durchführungsaufwand
  - Erwartetem Ergebnis
  - Bsp:
    - Statische Analyse hat minimalen Aufwand und sinnvolles Ergebnis, **aber**
    - Nur dynamisches Testen kann Funktionsfähigkeit *wirklich* testen
- Durchführung:
  - Zeitpunkt ist stark abhängig vom Tool: Statisch vs. Dynamisch
  - Häufige Ausführung nur bei automatisierbaren Tools sinnvoll, aber Ergebnisse müssen auch analysiert werden

Fallstudien

Freie Universität Berlin

- Was will ich herausfinden? (Wiederholung)
  - Werkzeuge und Testverfahren in OS-Projekten:
    - Werden Werkzeuge eingesetzt?
    - Welche (frei verfügbare, selbst geschriebene) und wie?
    - Wie ist der Einsatzaufwand?
    - Sinnvolle Ergebnisse nach Testeinsatz?
    - Wenn kein Werkzeugeinsatz erfolgt => Warum?

- Wie wird untersucht?
  - Top-down
  - Bottom-up

Fallstudien – Top-down

Freie Universität Berlin

- Top-down: Suche nach Projektenreferenzen bei Werkzeugen
  - In ca. 50 Werkzeugen:
    - Verschiedene Werkzeugarten (vgl. Kategorien)
    - Allgemeine und auch spezielle für Sicherheit(sschutz)
    - Expliziter Einsatz für Sicherheit gesucht
    - Versuch Projektnamen zu finden

- Suchort und -methodik
  - Durchsucht wurden Homepages, Wikis, Foren, Mailinglisten
  - Gesucht wurde zu aller erst nach Schlüsselwörtern wie: security, testing, fuzzing, injection, exploit, automated, etc.
  - Bei kleinen Datenmengen wurde auch manuell gesucht

Fallstudien – Top-down (2)

Freie Universität Berlin

- Ergebnisse:
  - Es lies sich kein einziger Projektname ermitteln
  - Schlagwort *Security* kam zwar oft vor, aber in einem komplett anderen Zusammenhang
  - Im eigentlichen Sinn wurde nur eine Anfrage zu Security Testing gefunden bei „Watir Recorder" vom 5. März 2007

Kavitha1980: „Hi All; I needed information about security testing. How are test cases written to implement security testing. If any one can share information on this topic pls mail me at [...] Thanks and Regards; Kavitha"

Antwort:

rutgetsmit: „What do you mean by 'security testing'. Can you give us more details about your approach?"

Fallstudien – Bottom-up

Freie Universität Berlin

- Bottom-up: Suche nach Werkzeugen in Projekten
  - 2 Ansätze: passiv und aktiv
  - In 15 Projekten:
    - Verteilt
    - Hohe Verbreitung
    - Zum größten Teil Web-Projekte
    - Gewisse Größe
    - Große Angriffsfläche
  - Projekte: OpenCms, JAMWiki, Bugzilla, Apache Tomcat, Apache HTTPd, MediaWiki, Joomla!, phpMyAdmin, Gallery, Openbravo ERP, XOOPS, OFBiz, Alfresco, phpBB, WebCalendar

Wiederholung: Ich suche nach dem Einsatz von Sichertestwerkzeugen!

Fallstudien – Bottom-up (2)

Freie Universität Berlin

- Passive Analyse: Analyse folgender Ressourcen
  - Homepage
  - Buildskripte
  - Repository/Code
  - Bugtracker
  - Mailinglisten/Foren
  - Gemeldete Sicherheitslücken

- Aktive Analyse: Kontakt mit/per
  - Developer Mailinglist
  - Maintainer
  - Web-Umfrage
  - Persönliches Interview

Fallstudien – Bottom-up (3)

Freie Universität Berlin

- Homepage
  - Kein einziger Verweis auf entsprechende Werkzeuge
  - Vielfach Security Advisory-Seiten mit eMail-Adresse vorhanden (Tomcat, HTTPd, Joomla!, phpMyAdmin)
  - Wenige haben einen Testplan oder QA überhaupt (HTTPd, Openbravo, XOOPS)
  - Einige weisen auf interne und externe Security Code Audits hin (HTTPd, Tomcat, phpBB)
  - Manche haben Ratgeber zu Lücken und sogar programmatischer Vermeidung derer (Bugzilla, Joomla!, phpMyAdmin)

Fallstudien – Bottom-up (4)

Freie Universität Berlin

- Buildskripte/Repository
  - Vorhandene Tests beschränken sich bestenfalls auf {J,PHP,Perl}Unit-Testfälle, manche sehr sehr mager
- Bugtracker
  - Suche nach sicherheitskritischen Issues
  - Bandbreite der Meldungen erstreckt sich von keine bis viele
- SecurityFocus.com/CVE
  - Von den gelisteten Projekten hat fast jedes Projekt viele (10+) Meldungen
  - Die wenigsten dieser Meldungen tauchen in den Trackern oder Mailinglisten auf

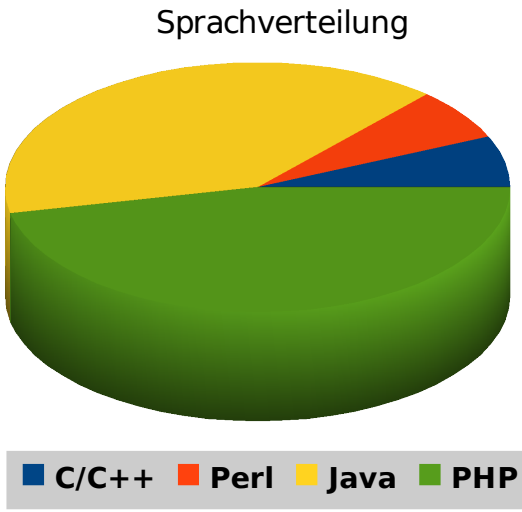
Fallstudien – Bottom-up (5)

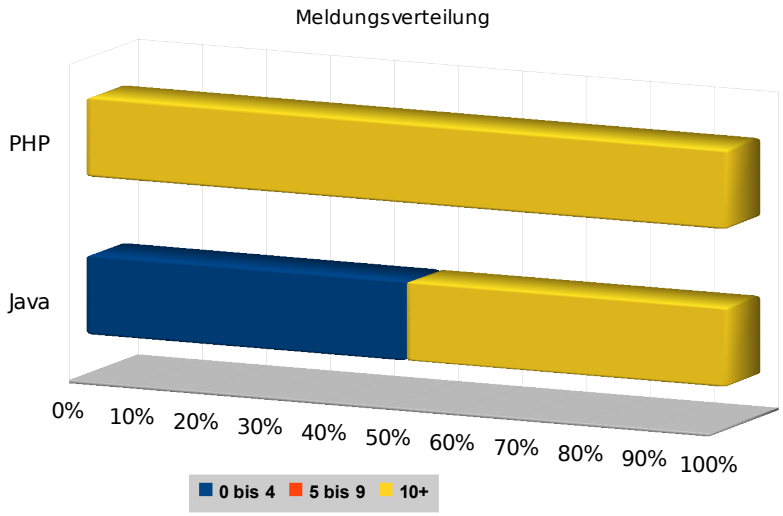
Freie Universität Berlin

- Mailinglisten/Foren
  - Gesucht wurde nach Schlüsselwörtern wie: security, testing, fuzzing, injection, exploit, automated, etc.
  - Hoffung: Finden von Werkzeugnamen, Einsatz, Testplänen
  - Antworten auf Gewährleistung der Sicherheit:
    - Manche Fragen versickern
    - Mache verweisen auf die Sicherheit durchs Framework
    - Manche geben zu, dass sie schlicht und ergreifend nichts in der Hinsicht Sicherheitstests machen
    - Manche planen langfristig ein Sicherheitstestframework zu entwickeln (phpBB)
    - Manche geben an, dass sie mit „security in mind" programmieren (Bugzilla, Joomla, Gallery, phpBB)
    - Manche Testen (allgemein) unorganisiert lokal (phpBB)
  - Nur in einem Projekt erwähnte ein Entwickler (!), dass BeEF interessant wäre (Browser Exploitation Framework)

Fallstudien – Sprachverteilung

- Ausgewogene Projektverteilung zwischen zwei Sprachen
- *Unausgewogene Meldungsverteilung*
  - Von den 15, die gelistet sind bei SecurityFocus.com/CVE
  - Mehr PHP- als Java-Projekte gelistet
  - PHP-Projekte haben bedeutend mehr Meldungen
  - ***Ist das Zufall?***





Fallstudien – Fazit

Freie Universität Berlin

- Zusammenfassung
  - Aktive Tätigkeiten im Sicherheitsbereich sind nur minimal vorhanden
  - Eindruck: Beschränkte, unorganisierte, eher gar keine Durchführung
  - PHP-Projekte haben sehr viele Security-Meldungen
  - Viele Releases sind Security-Fixes
- Einfluss auf weiteren Verlauf
  - Erweiterung auf andere Softwaretypen wie *BSD, Open*
  - Web-Umfrage und direkte Kontakte werden wenig ergiebig sein
  - Konzentration auf wenige, vielversprechende Projekte und persönliche Interviews auf dem LinuxTag

Selbstversuche

Freie Universität Berlin

- Eigentlich sollte aus dem Fallstudien ein Testplan bzw. Empfehlungen für andere Entwickler ausgearbeitet werden
- Idee erscheint nicht sinnvoll, da passive Analyse recht erfolglos war
- Neue Fragestellung durch Selbstversuche:
  - Ist es schwierig die Problematik zu verstehen?
  - Kann man gute Testwerkzeuge finden?
  - Lassen sie sich sinnvoll einsetzen?
  - Wie sehr muss man sich mit Projekt und Testfällen beschäftigen?
  - Findet ein Laie (ich) eine Sicherheitslücke?
  - Rückfluss dieser Erkenntnisse bei den Interviews am LinuxTag

Diskussion

Freie Universität Berlin

Habt ihr eine Meinung oder Kritiken?
Habt ihr noch Fragen?

Vielen Dank!