Beschreibung sicherheitsrelevanten Verhaltens in der Open Source Softwareentwicklung

Zwischenpräsentation Masterarbeit

Tobias Opel
Institut für Informatik
FU Berlin
08.05.2008

Agenda

Freie Universität Berlin

- Ziel der Arbeit
- Aufgaben
- Forschungsmethode
- Datenerhebung
- Unterstützende Praktiken
- Untersuchte Projekte
- Zwischenergebnisse
- weitere Aufgaben auf meinem Weg
- Diskussion

Ziel der Arbeit

- Entwicklung einer Beschreibung sicherheitsrelevanten Verhaltens in der Open Source Softwareentwicklung
  - Identifikation von Verhaltensweisen und Rollen, welche bei der Behebung von sicherheitsrelevanten Schwachstellen involviert sind
  - durch qualitative Datenanalyse diese Verhaltensweisen, Rollen und deren Beziehungen untereinander beschreiben
  - betrachtet werden dabei Open Source Webanwendungen und Bibliotheken

Aufgaben

Freie Universität Berlin

1. Datenbestände zur Behebung sicherheitsrelevanter Schwachstellen finden und auswählen
2. Verhaltenskategorien, Rollen und ihre Beziehungen untereinander aus den Daten ableiten
3. Vergleich von Literatur zur Qualitätssicherung mit den eigenen Erkenntnissen
4. Entwicklung von Erfolgsmaßen und Bewertung unterschiedlicher Formen sicherheitsrelevanten Verhaltens
5. Beschreibung von Verhaltensweisen und Mechanismen zur Vermeidung sicherheitsrelevanter Schwachstellen

Forschungsmethode

Freie Universität Berlin

- Grounded Theory (gegenstandsverankernde Theoriebildung nach Strauss & Corbin):
  - Qualitativer Forschungsansatz
  - Datenerhebung basierend auf dem zu untersuchenden Phänomen
  - Konzeptualisieren der Daten (Kodieren) und Gruppieren der Konzepte zu Kategorien
  - Beziehungen zwischen den Kategorien finden
  - Kernkategorie auswählen und diese in Beziehung zu den anderen Kategorien setzen zur Bildung der Theorie
  - Datensammlung, Analyse und Theorie stehen in wechselseitiger Beziehung zueinander

Datenerhebung

Wahl der Projekte: Erste Annäherung

- erste Projekte gewählt aufgrund persönlichen Bezugs
- Drupal, Horde
- Schwachstellen vorhanden
- aufgetretene Probleme
  - Drupal
    - geschlossene Security Mailingliste
    - auch nach Nachfrage keine Zugriffsmöglichkeit
  - Horde
    - wenig Schwachstellen
- Fazit
  - zu wenig Daten vorhanden für die Analyse

Datenerhebung

Freie Universität Berlin

Wahl der Projekte

- Anfang März
- Ausgangspunkt National Vulnerability Database (http://nvd.nist.gov/)
- vorhandene Schwachstellenmeldungen in absteigend chronologischer Reihenfolge angeschaut
- Filterung relevanter Projekte
  - Open Source Software
  - Webanwendungen oder Bibliotheken
  - Zeitraum der Schwachstellenmeldungen 01.01.2007 bis Anfang 2008
  - Behebung der Schwachstelle sollte vorliegen (Gesamtprozessbetrachtung)

Datenerhebung

Freie Universität Berlin

Wahl geeigneter Projekte

- neben Vulnerability Datenbanken auch andere Quelle durch Erweiterung des Beobachtungsraumes
  - Meldungen auf Security Seiten (z.B. heise security)
  - Blogs zum Thema Sicherheit
- über die beiden Vorgehenswege Projekte gefunden für die Datenanalyse
  - Initialprojekt Wordpress
    - wird später vorgestellt

Datenerhebung

Freie Universität Berlin

Daten in den Projekten

- zu den Projekten gehörende Schwachstellen aus dem Jahr 2007 herausgesucht
- Datenquellen zu den Schwachstellen identifiziert
  - Mailinglisteneinträge, Forenbeiträge, Chatlogs
  - Schwachstellenmeldungen aus Vulnerability Datenbanken
  - Blogeinträge, Newsposts, Patchankündigungen
  - Bugtrackereinträge
  - Wikieinträge

Datenerhebung

Freie Universität Berlin

- Episode
  - Gesamtheit aller zu einer Schwachstelle gehörenden Datenquellen
  - Abbildung des Behebungsprozesses einer Schwachstelle

- Eigenschaften der Datenbasis
  - Daten liegen in Dokumenten in Textform vor
  - eine Person ist an dem vorliegenden Text beteiligt
  - oder mehrere Personen sind innerhalb eines Dokumentes in einer Diskussion involviert
  - über Hyperlinks können verschiedene Dokumente in Beziehung stehen

Unterstützende Praktiken

Freie Universität Berlin

- Ausgangspunkt: große Datenbasis, kein vorgegebenes Kodierschema
- [SalPloPre07] schlagen Praktiken vor, welche die Entwicklung eines Kodierschemas erleichtern sollen

Praktik 1 - Perspektive festlegen

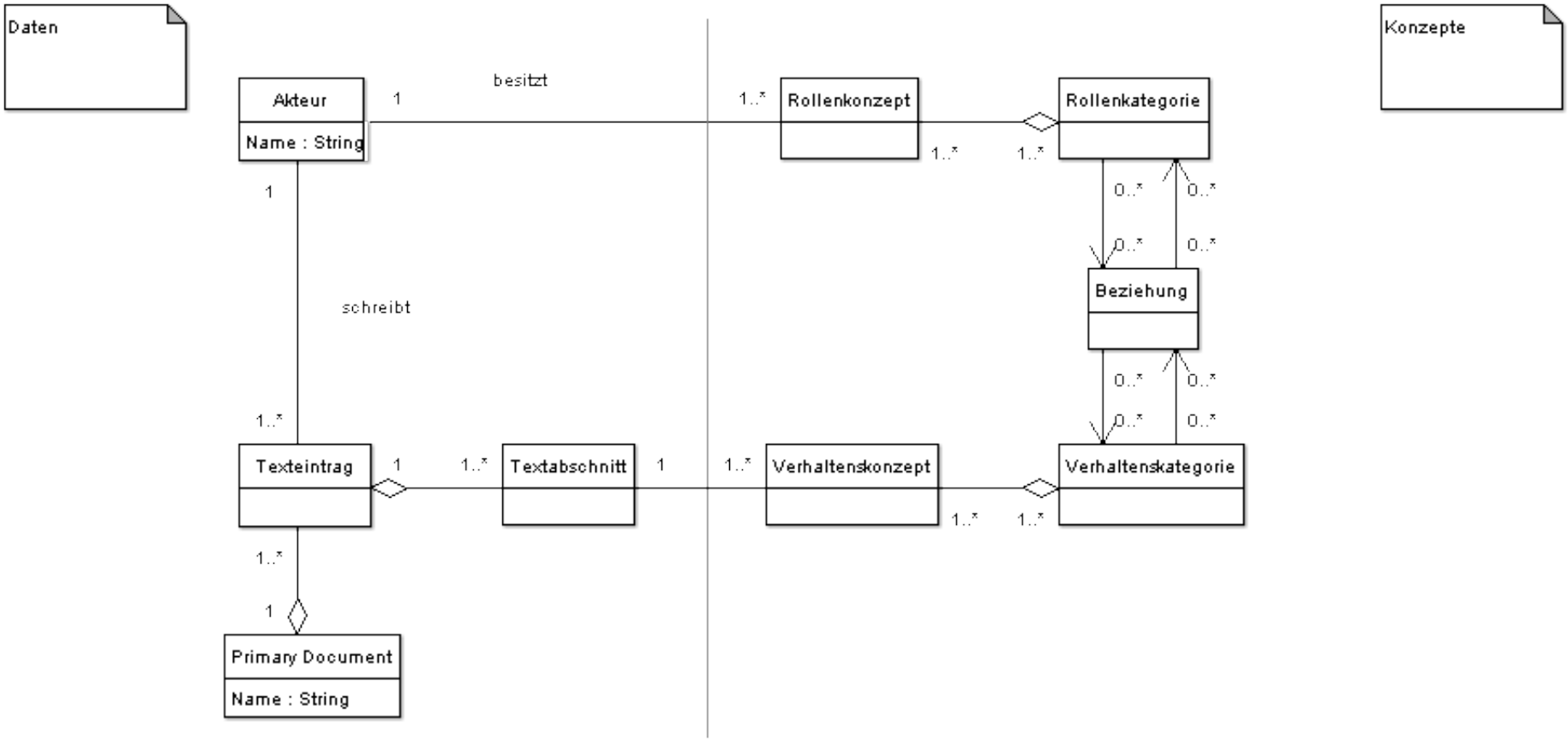
- mit Hilfe durch Beantwortung der Fragen:
  1. Unter welchen Gesichtspunkten erwarte ich in den Daten Erkenntnisse zu finden?
  2. Welche Arten von Phänomenen möchte ich bei der Betrachtung der Daten zulassen?
  3. Zu welchen Ergebnissen will ich gelangen?

Unterstützende Praktiken

1. Unter welchen Gesichtspunkten erwarte ich in den Daten Erkenntnisse zu finden?
   - Die Daten helfen mir zu Verstehen, welche Verhaltensweisen und Rollen bei Behebung von sicherheitsrelevanten Schwachstellen in OSS relevant sind.
2. Welche Arten von Phänomenen möchte ich bei der Betrachtung der Daten zulassen?
   - Phänomene die direkt aus den Daten erhoben werden können.
3. Zu welcher Art von Ergebnissen will ich gelangen?
   - ein System aufstellen von Verhaltens- und Rollenkategorien, welche innerhalb des Behebungsprozesses relevant sind
   - Beziehungen zwischen diesen Kategorien finden

Freie Universität Berlin

Einschub: Analyse Modell



Warum der Akteur?

- Akteur als Hilfsmittel bei der Kodierung
  - Texteinträge in den Dokumenten lassen sich genau einem Akteur zuordnen
  - innerhalb dieser Texteinträge werden Verhaltenskonzepte bestimmt
  - dem Akteur selber werden Rollenkonzepte zugeordnet
  - Vergleiche zur Bildung von Rollenkategorien finden auf Ebene des Akteurs statt
  - verschiedene Rollenzuordnungen innerhalb des Behebungsprozesses bei den Akteuren sichtbar

Praktik 2 – Konzeptnamensschema

- nach [SalPloPre07] folgendes Schema:

```
code = <actor>.<description>
actor = P1 | P2 | P
```

- der Akteur als eindeutiger Bezeichner kennzeichnet bei mir den kompletten Texteintrag
- Kodierung innerhalb des Texteintrags findet nur mit <description> statt (nächste Folie)
  - Verhaltenskonzepte werden innerhalb des Eintrags identifiziert
  - kodierte Textabschnitte sind automatisch dem zum Texteintrag gehörenden Akteur zugeordnet

Unterstützende Praktiken

Freie Universität Berlin

`description = <verb>_<object>[_<criterion>]`

- <verb> steht in dritter Person Singular
  - wird erweitert um Präfix und ggf. einer Präposition
- <object> ist ein Substantiv, welches den Gegenstand der Aktivität darstellt
- der Optionale Teil <criterion> dient Spezialisierungen und Ausprägungsmerkmalen/Eigenschaften
  - darüber findet auch die Negation des Konzeptes statt

- Bsp:
  - bietet_Details/Informationen
  - spezifiziert_Version
  - beschreibt_Programmverhalten_ungewöhnliches

Unterstützende Praktiken

- Rollenkodierung

```
Rolle: <Rollenkonzept>
```

- wird dem Akteur zugeordnet
- Rollen können Beziehungen untereinander besitzen
  - dient der Darstellung der Entwicklung eines Akteurs bzw. der Rollenkategorie

Untersuchte Projekte

- Wordpress
  - Open Source Webanwendung
  - Weblog-Publishing-System für Weblogs
  - Technologien: PhP und Mysql
  - unter GNU General Public License

- Datenerhebung bei Wordpress
  - Schwachstellenidentifikation für einzelne Episoden über den offenen Bugtracker von Wordpress
  - alle behobenen Tickets der Komponente "Security" betrachtet
    - viele Verweise auf weitere Daten (Foreneinträge, Mailinglisten..)
    - Behebungsprozess ersichtlich aus den Daten im Ticket

Untersuchte Projekte

Freie Universität Berlin

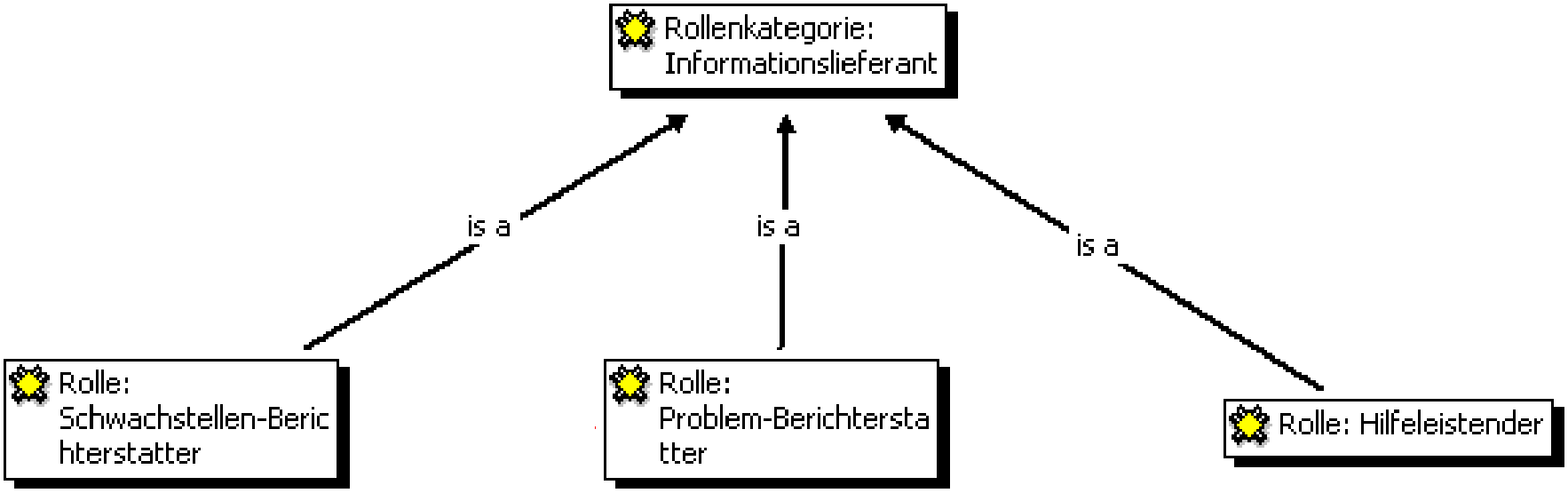
- Wordpress als Einstiegsprojekt mit vielversprechenden Episoden
- Wahl nächster Projekte
  - funktional vergleichbar mit Wordpress (Webanwendung, Content Management ähnliches System)
  - genügend Daten für meine Untersuchung (offener Zugang zu mit Sicherheitsbehebungsprozessen)

- Ausblick: Joomla
  - Open Source Content Manegement System
  - Technologien: PhP und Mysql
  - unter GNU General Public License
  - bietet Tracker, Securityforen, Security FAQ und eigene Schwachstellenübersicht zu den Modulen

Rollen

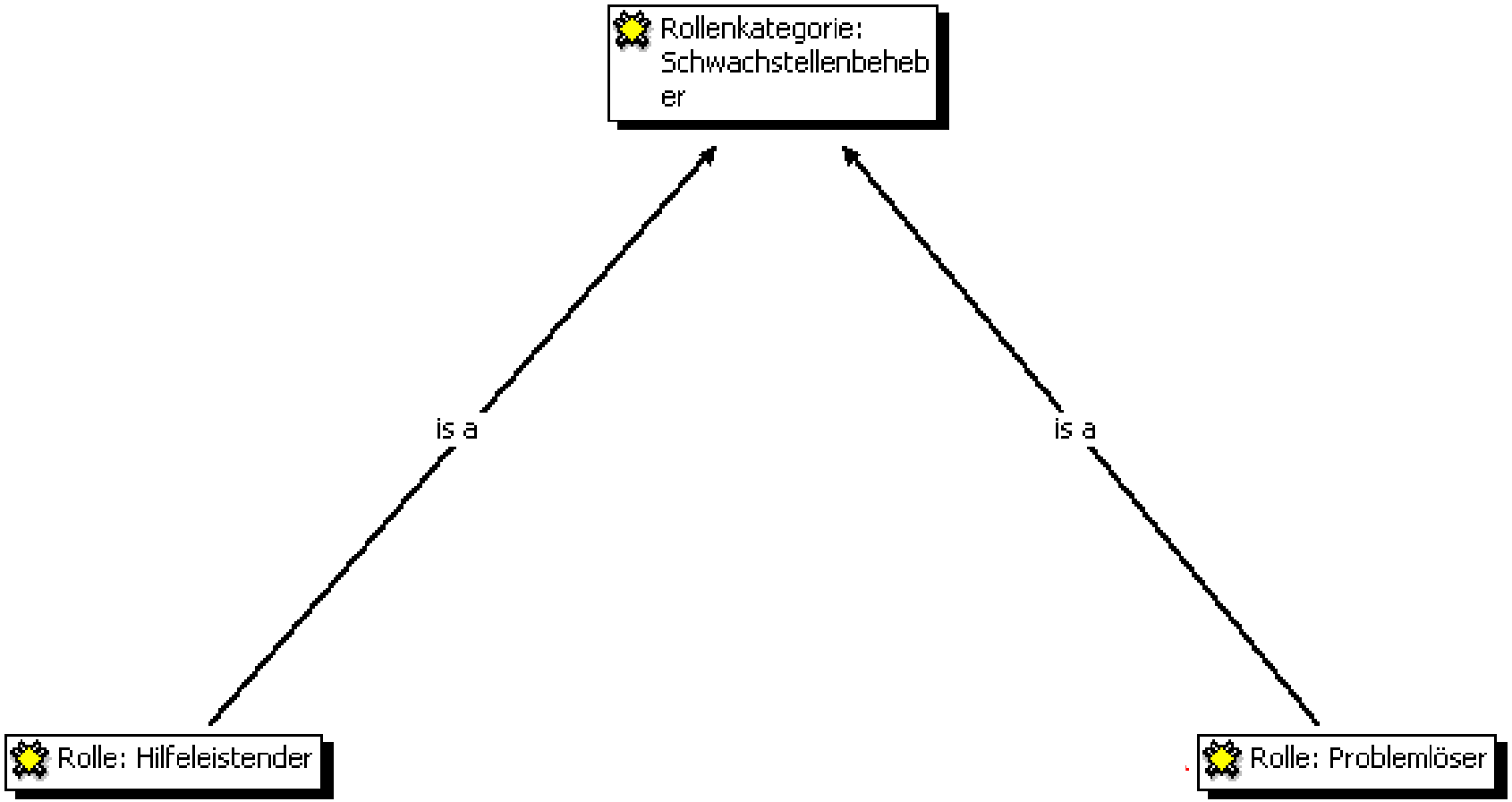
- den Akteuren zugewiesene Rollenkonzepte
  - Beitragender/Vertrauter, Dokumentatoren, Ehemaliger, Externer, Forumsmoderator, Hilfeleistender/Problembeheber, Interner, Kernteam, Patchankündiger, Problem-Berichterstatter, Schwachstellen-Berichterstatter, Site Betreiber/OSS Benutzer, Supporter
- vorläufige Rollenkategorien
  - Informationslieferant
  - Schwachstellenbeheber

- Rollenkategorie: Informationslieferant



Zwischenergebnisse

- Rollenkategorie: Schwachstellenbeheber



Zwischenergebnisse

- Verschiedene Rollen während des Behebungsprozesses

Verhalten

- folgende vorläufige Kategorien bisher gefunden:
  - Informationsbereitstellung
    - beschreibt_Schwachstelle, bietet_Details/Informationen, grenztEin_Schwachstellenvoraussetzung, spezifiziert_Version etc.
  - wechselseitiger Dialog
    - zitiert_Diskussionspartner, stellt_Verständnisfrage, bewertet_Lösung, bedanktSichBei_Berichterstatter etc.
  - Unterstützung geben
    - gibtAn_Workaround, stelltBereit_Patch, leitetAn_Symptombekämpfung, verlinkt_Ticket etc.

Zwischenergebnisse

Freie Universität Berlin

Weitere Erkenntnisse

- Veröffentlichungsproblematik
  - Darf die Schwachstelle zur Behebung öffentlich bekannt gegeben werden? Welche Folgen hat das? Gibt es Einschränkungen/Zwischenlösungen dafür?
- Behebungsprozess/Phasen
  - (Sicherheits-)Problemmeldung – Informationsbeschaffung – Ursachenfund/-meldung – Behebung(Fix/Review/Commit) – Release
  - weitere Varianten:
    - Bugbehebung und Veröffentlichung – Schwachstelle – Release
    - Schwachstellenveröffentlichung – Behebung – Release
    - geheime Schwachstellenmeldung – interne Behebung - Release - Schwachstellenveröffentlichung

Weitere Aufgaben

- Rollen- und Verhaltenskategorien weiter entwickeln
- Beziehungen zwischen Kategorien aufstellen
- weitere Projekte finden
  - Webanwendungen ähnlicher Funktion (z.B. Joomla)
  - Webanwendungen mit gänzlich anderer Funktion
  - Bibliotheken
- Daten aus weiteren Projekten für den Vergleich zur Erweiterung, Bestätigung oder Änderung meiner Kategorien
- Behebungsprozess weiter untersuchen

Vielen Dank!