Windows-Legacy-Applikationen unter Linux

- Vortrag von David Gümbel

Windows-Legacy-Applikationen unter Linux

- Problemstellung
  - Fallbeispiel Stadt Böblingen
  - allgemein
- Lösung
  - PE Header Analysis
  - Codeflow Analysis
  - Graphen-Algorithmen

Fallbeispiel Stadt Böblingen

- 400 Arbeitsplätze
- Windows NT 4 + Office 97
- 60-80 Applikationen (!)
- Probleme
  - NT nicht mehr supported
  - Wartbarkeit
  - Sicherheit
- Linux eine Alternative?

WINE

- WINE Is Not An Emulator
- Loader und Windows-API für Linux/IA32
  - aber: keine vollständige Implementierung
- Open Source unter der LGPL
- lauffähig u.A.:
  - MS Office
  - Internet Explorer
  - diverse Spiele

Fallbeispiel Stadt Böblingen

- Exemplarische Untersuchung
  - Zeiterfassung
  - Baugenehmigung
  - Sozialwesen
  - Dokumentenverwaltung
- Ergebnisse
  - WINE ist gut
  - ..aber: alle APIs ist besser
  - ...und welche werden gebraucht (!?)

Highway to Heaven

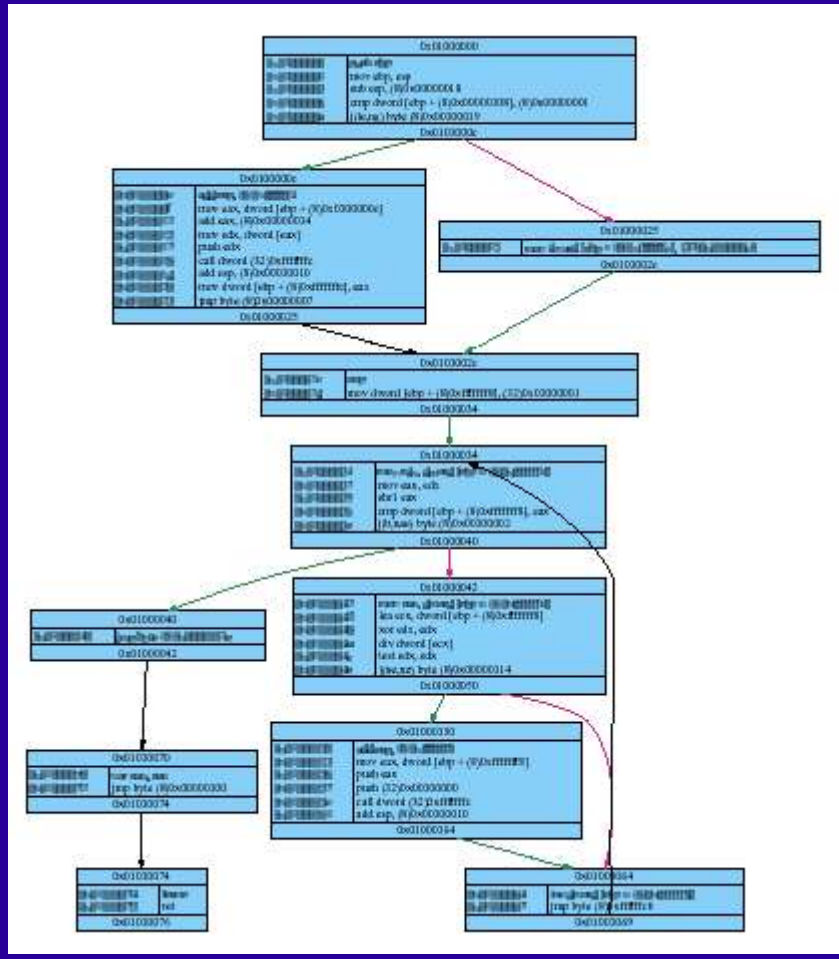
- APIs erfassen
  - importierte
  - exportierte
  - Abhängigkeits-Graph
- Abgleich mit WINE-Implementierungsstand
- APIs gewichten: welche APIs sind
  - häufig benutzt
  - selten benutzt
  - wichtig
  - entbehrlich

Schritt 1 – PE Header Analysis

- Portable Executables
  - DLL, OCX, EXE
  - Header
    - Import Table
    - Export Table
- Automatische Auswertung
  - Abhängigkeiten (Graph)
  - Abgleich mit WINE-Implementierungsstand
- Probleme
  - Unterscheidung wichtig/unwichtig
  - wo im Code erfolgt Zugriff

Schritt 2 – Codeflow Analysis

- Kontrollfluß betrachten
- Unterteilung in „basic blocks“
- Disassembly
- Graph
- Abgleich mit Headern



Fazit

- Ziele waren:
  - APIs erfassen (importierte, exportierte, Abhängigkeiten)
  - Unterscheidung nach Wichtigkeit, Zahl der Aufrufe etc.
  - seriöse Problemabschätzung, automatisch und formal
- Ergebnis:
  - Ziele erreicht
  - Ansatz ausbaubar

Ausblick

- Graphenalgorithmen
  - kürzeste Pfade
  - Zyklen
  - Gewichtung (Knotengrad o.Ä.)
- Sicherheit
- Qualität (z.B. von Kompilaten)
- Performance
- Hardware