Projekte Praktikum KI

(6. Semester)

Sommersemester 2005

1 Journalling Filesysteme

- XFS – `http://oss.sgi.com/projects/xfs/`
- JFS – `http://jfs.sourceforge.net/`
- EXT3 – `http://batleth.sapienti-sat.org/projects/FAQs/ext3-faq.html`

Aufgaben:

- Beschreibungen der Filesysteme lesen
- Kurze Zusammenfassung der wichtigsten Informationen
  - Features
  - Mountoptionen
  - Interaktion mit NFS/Samba
- Vergleichende Tabelle der
  - Vor–/Nachteile,
  - maximalen Werte,
  - Performance
- C–Programme/Shell–Skripte schreiben
  - Testen der Features
  - Performance–Test

2 Krypto Filesysteme

Zusammenfassung

`http://www.finux.org/Reprints/Reprint-Halcrow-OLS2004.pdf`

lesen, hierbei Schwerpunkt auf Abschnitt 5 legen.

Aufgaben:

- Informationen zu Filesystemen in Abschnitt 5
- Kurze Zusammenfassung der wichtigsten Informationen
  - Features
  - Mountoptionen
- Vergleichende Tabelle der
  - Vor–/Nachteile,
  - maximalen Werte,
  - Performance
  - verwendeten Verschlüsselungsalgorithmen
- C–Programme/Shell–Skripte schreiben
  - Testen der Features
  - Performance–Test (evtl. unterschiedlicher Verschlüsselungsalgorithmen)

3 Code Audit Webserver `pserv`

Der PICO Server ist eine schlanke Webserver–Implementierung.

`http://pserv.sourceforge.net/`

Aufgaben:

- schematische High–Level Beschreibung der Funktionsweise von `pserv` mit Verweis auf verwendete Funktionsnamen
- besonderes Augenmerk auf die Behandlung von Strings dynamischer Länge, die von einem (evtl. böswilligen) Browser beeinflußt werden können
- alle gefährlichen C–Library–Aufrufe auf Manipulierbarkeit untersuchen (siehe auch `http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/dangers-c.html`)
- mit einem TCP–Client–Testprogramm versuchen, den `pserv` zum Absturz zu bringen
- CGI–Schnittstelle untersuchen, testen

4 Vergleich von Content–Management–Systemen `OpenCMS` und `Mambo`

Projekt–Webseiten:

- OpenCMS: `http://www.opencms.org/opencms/en/`
- Mambo: `http://www.mamboserver.com/`

Aufgaben:

- Eigenschaften erarbeiten
  - Abhängigkeiten von anderer Software (PHP, Apache,...?)
  - Bedienbarkeit
  - Konfigurierbarkeit
  - Insert/Update/Delete, Versionskontrolle von HTML–Seiten
  - Entdeckung veralteter oder nicht (mehr) verlinkter HTML–Seiten
  - Entdeckung veralteter Links
  - Einbindung verschiedenster Fileformate
  - Aussagekraft der Fehlermeldungen
  - gegenseitige Abschottung der Webautoren
- Vergleichstabelle anlegen
  - Vorteile/Nachteile
  - Features

5 Gültigkeit von Links / 2 Tools erstellen

C– oder C++–Programm, das für eine angegebene Mozilla–Bookmark–Datei

- Mozilla–Bookmarks liest,
- die Links auf Gültigkeit überprüft
- Mozilla–Bookmarks in eine andere Datei schreibt, hierbei die ungültigen wahlweise
  - wegläßt oder
  - markiert

C– oder C++–Programm, das für eine angegebene URL

- alle von dort aus erreichbaren Links findet, welche
  - entweder lokal sind
  - oder der erste nicht–lokale Link sind
- die Links auf Gültigkeit überprüft
- einen Report in eine Ausgabedatei schreibt
- Parallelisierung des Testens der Links mit Hilfe von `fork()`
- Limitierung der Anzahl der durch `fork()` erzeugten Prozesse

Da die Tätigkeiten der beiden Programme verwandt sind, sollen gemeinsame Funktionalitäten in einer Library zusammengeführt werden.

6 Hashverfahren SHA–256/–384/–512/–224

Eine sichere Hashfunktion h berechnet aus einem String x einen Hashwert $h(x)$ fixer Länge in der Weise, daß in ,,vernünftiger" Zeit kein String x' gefunden werden kann mit $h(x) = h(x')$. Sichere Hashfunktionen werden zur Erzeugung sicherer elektronischer Unterschriften eingesetzt. Die seit 2004 gefundenen Schwächen, der als sicher eingeschätzten Hashfunktion SHA–1 haben die Kryptoforscher dazu gezwungen nach Alternativen zu SHA–1 Ausschau zu halten. Als Nachfolger–Kandidaten werden u.a. die Funktionen SHA–256/–384/–512/–224 gehandelt.

Aufgaben:

- Implementierung der vier (verwandten) Hashfunktionen
- Validierung mittels Testvektoren
- Performance–Tests

Grundlage ist FIPS 180-2, Secure Hash Standard

`http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf`

7 Hashverfahren Whirlpool

Eine sichere Hashfunktion h berechnet aus einem String x einen Hashwert $h(x)$ fixer Länge in der Weise, daß in „vernünftiger" Zeit kein String x' gefunden werden kann mit $h(x) = h(x')$. Sichere Hashfunktionen werden zur Erzeugung sicherer elektronischer Unterschriften eingesetzt. Die seit 2004 gefundenen Schwächen, der als sicher eingeschätzten Hashfunktion SHA–1 haben die Kryptoforscher dazu gezwungen nach Alternativen zu SHA–1 Ausschau zu halten. Als Nachfolger–Kandidat wird u.a. der Whirlpool–Algorithmus gehandelt.

Aufgaben:

- Implementierung des Algorithmus
- Validierung mittels Testvektoren
- Performance–Tests

`http://planeta.terra.com.br/informatica/paulobarreto/WhirlpoolPage.html`