

# 2nd Kassel Student Workshop on Security in Distributed Systems (KaSWoSDS'09) – CfP

Thomas Weise, Philipp Andreas Baer  
University of Kassel  
Wilhelmshöher Allee 73  
34121 Kassel, Germany  
`weise@vs.uni-kassel.de`, `baer@vs.uni-kassel.de`

8. Oktober 2008

## Zusammenfassung

Der 2. Kassel Student Workshop on Security in Distributed Systems, kurz KaSWoSDS'09, findet im Rahmen des Seminars Sicherheit in Verteilten Systemen im Wintersemester 2008 an der Professur Verteilte Systeme statt. In diesem Artikel möchten wir einige einleitende Hinweise zum Workshop geben. Gleichzeitig stellt er ein Beispieldokument dar, wie die Ausarbeitung anzufertigen ist.

## 1 Einleitung

Im Wintersemester 2008 veranstaltet die Professur Verteilte Systeme (siehe Grafik 1) der Universität Kassel (siehe Grafik 2) das Seminar *Sicherheit in Verteilten Systemen*. An diesem Seminar können sowohl Studenten im Hauptstudium (Master) als auch im Grundstudium (Bachelor) teilnehmen. Nachdem ausgesprochen positiven Feedback vom letzten Jahr [1] wird das Seminar zum zweiten Mal als Workshop (*2nd Kassel Student Workshop on Security in Distributed Systems, KaSWoSDS'09*) organisiert und veranstaltet.

- Die Vorträge des Seminars werden im Rahmen einer eintägigen Blockveranstaltung (dem eigentlichen Workshop), gegliedert in mehrere Sessions, gehalten.



Abbildung 1: Das Logo der Professur Verteilte Systeme

# U N I K A S S E L V E R S I T Ä T

Abbildung 2: Das Logo der Universität Kassel

- Nach der Themenvergabe erstellt jeder Seminarteilnehmer eine Ausarbeitung zu seinem Thema. Diese wird dann über das Konferenzsystem hochgeladen.
- Wie bei einem Workshop werden die Seminarbeiträge von mehreren Reviewern begutachtet, die zum Beispiel den Inhalt, die Verständlichkeit, sowie Rechtschreibung und Grammatik bewerten.
- Diese Bewertungen werden von anderen Seminarteilnehmern durchgeführt und erfolgen vor dem Workshop und vor der eigentlichen Benotung. Sie sollen den einzelnen Teilnehmern Gelegenheit geben, ihre Ausarbeitungen zu verbessern.
- Nach Erhalt der Bewertungen fertigt jeder Teilnehmer eine Finalversion seiner Ausarbeitung an. Diese wird wieder über das Konferenzsystem hochgeladen.
- Das Seminar findet als Blockveranstaltung statt, in der alle Teilnehmer ihre Vorträge halten.
- Die Benotung des Seminars setzt sich zu  $2/3$  aus der Ausarbeitung und zu  $1/3$  aus der Bewertung des Vortrags zusammen. Dabei spielen die Reviewergebnisse keine Rolle. Es wird jedoch berücksichtigt, ob *gerechtfertigte* Verbesserungshinweise der Reviewer in der Finalversion umgesetzt wurden. Wenn nicht, hat das natürlich negativen Einfluss auf die Bewertung.
- Alle Ausarbeitungen, die eine Bewertung von mindestens 2.0 erhalten haben, werden in die Workshop-Proceedings aufgenommen. Wenn die Proceedings mindestens fünf Ausarbeitungen umfassen, so werden sie in den Kasseler Informatikschriften als Technischer Bericht veröffentlicht. Dies wurde bereits im vergangenen Jahr so gemacht [1]. Dadurch haben dann auch alle Teilnehmer ein schönes, vorzeigbares und vor allem dauerhaftes Endprodukt ihrer Arbeit.

Die Eckdaten des Workshops sind in Tabelle 1 gelistet. Zu beachten ist, dass es sich bei den Daten um *harte Deadlines* handelt – wir verwenden ein automatisches Konferenzsystem, dass z.B. eine Einreichung nach dem 19.12. dann nicht mehr zulässt.

## 2 Themen

Die Themen des Seminars (ausgenommen des Übersichtsvortrags) setzen sich jeweils aus einem praktischen und einem theoretischen Teil zusammen. Der

Tabelle 1: Eckdaten des KaSWoSDS'09

Was	Wert
Titel:	2nd Kassel Student Workshop on Security in Distributed Systems
Abkürzung:	KaSWoSDS'09
Lehrveranstaltung:	Sicherheit in Verteilten Systemen
Website:	<a href="http://www.vs.uni-kassel.de/teaching/ws-20082009/seminar/">http://www.vs.uni-kassel.de/teaching/ws-20082009/seminar/</a>
Veranstalter:	Prof. Dr. Kurt Geihs
Leiter:	Thomas Weise und Philipp Andreas Baer
Vorbesprechung:	21.10.2008, 16:00-17:00, Raum 1405
Einreichung:	19.12.2008
Review-Deadline:	09.01.2009
Final-Version:	16.01.2009
Workshop:	27.01.2009, 09:00-18:00, Raum 1405
Voriger Workshop:	KaSWoSDS'08 [1]

praktische Teil erfordert das Erstellen eines vorführbaren Beispiels für die jeweilige Sicherheitslücke. Beim Thema *Denial of Service* wird z.B. ein Client-/Serversystem programmiert, das während der Präsentation dann durch eine Attacke kompromittiert wird. Dieses Beispiel zählt auch als Teil der Ausarbeitung, ebenso wie der theoretische Teil, der jeweils die grundlegende Natur der Sicherheitslücke sowie etwaige Maßnahmen zu deren Schließung behandelt.

In der KaSWoSDS'08 [1] wurden bereits folgende Themen behandelt: Buffer Overflows, Cross-Site Scripting (XSS), SQL-Injection, Spoofing, Viren und Klassische Verschlüsselungs- und Angriffsverfahren. Diese stehen daher nicht mehr zur Verfügung, können jedoch als Referenz her genommen werden, wie eine Ausarbeitung oder eine Demo aussehen sollte.

## 2.1 Übersicht

Die grundlegenden Angriffspunkte in einem Rechnernetz dienen als Einführung. Dazu zählen Schwachstellen in der Software und den Datenkanälen, aber auch Sicherheitsgefährdungen die aus unsachgemäßem Umgang resultieren.

## 2.2 Unsicherheit in Softwaresystemen

Es wird jeweils eine allgemeine Sicherheitslücke, die durch unvorsichtige Programmierung entsteht, allgemein diskutiert und dann anhand eines konkreten Beispiels demonstriert.

### 2.2.1 Denial-of-Service

*Denial-of-Service*-Attacken<sup>1</sup> (DoS) haben das Ziel, einen oder mehrere Dienste auf dem angegriffenen Host arbeitsunfähig zu machen. Dies wird meistens erreicht, indem das entsprechende System mit Paketen überlastet wird. DoS Attacken erfolgen meistens verteilt von mehreren Quellen gleichzeitig (DDoS), oftmals werden auch Varianten eingesetzt, die IP-Spoofing nutzen (siehe IP-Spoofing in [1], [2]). [3, 4, 5, 6]

Die Attacke wird allgemein analysiert, ein praktisches Beispiel wird erarbeitet und es wird diskutiert, welche Gegenmaßnahmen getroffen werden können.

### 2.2.2 Sniffing

*Sniffer*<sup>2</sup> sind Programme, die den Datenverkehr auf einem LAN mithören. Sie können zur LAN-Analyse verwendet werden, aber auch um beispielsweise das Surfverhalten von Internetnutzern aufzuzeichnen oder um Informationen abzuhören. [7, 8, 9]

Die Funktionsweise von Sniffern ist zu erklären, und ein eigener kleiner Sniffer wird programmiert und vorgeführt.

### 2.2.3 Trojaner

*Trojaner*<sup>3</sup> sind Programme, die einer nützlichen Applikation gleichen aber in Wirklichkeit Schadcode beinhalten. Sie werden z.B. erzeugt, indem man ein vorhandenes, nützliches Programm durch Umlinken mit einem anderen Programm verbindet, das den Schadcode enthält. Beim Starten wird dann immer dieser Code ausgeführt. [10, 11, 12, 13].

Ziel ist es hier, einen solchen Trojaner zu bauen, seine Funktionsweise zu zeigen und allgemeine Schutzmaßnahmen zu diskutieren.

### 2.2.4 Würmer

Ein *Wurm*<sup>4</sup> ist ein Programm, das sich über ein Netzwerk von Computern verbreitet. Es nutzt dazu höherwertige Ressourcen, Protokolle, Netzwerkdienste oder Benutzerinteraktionen. Würmer können auch dazu benutzt werden, Trojaner auf dem Zielsystem abzusetzen. [14, 15]

Genau wie bei Thema Viren in [1] soll ein harmloser Beispielwurm erstellt werden, der sich auf eine vorher definierte Menge von Testsystemen verbreiten soll. Es wird analysiert, wie der Wurm funktioniert (und wie Würmer allgemein vorgehen) und wie man allgemein die Verbreitung von Würmern verhindern kann.

---

<sup>1</sup>[http://de.wikipedia.org/wiki/Denial\\_of\\_Service](http://de.wikipedia.org/wiki/Denial_of_Service)

<sup>2</sup><http://de.wikipedia.org/wiki/Sniffer>

<sup>3</sup>[http://de.wikipedia.org/wiki/Trojanisches\\_Pferd\\_%28Computerprogramm%29](http://de.wikipedia.org/wiki/Trojanisches_Pferd_%28Computerprogramm%29)

<sup>4</sup><http://de.wikipedia.org/wiki/Computerwurm>

### 2.2.5 Phishing

Beim *Phishing*<sup>5</sup> wird versucht, über gefälschte Internetadressen oder gefälschte Emailabsender an die Daten des Benutzers zu kommen. Dabei wird der originale Webauftritt bzw. eine originale Email z.B. einer Bank täuschend echt nachempfunden bzw. eine fiktive Nachricht verbreitet, die die Benutzer dazu verleiten soll, ihre Informationen preiszugeben. Hier sind z.B. auch die einige Punkte aus der Spoofing-Sektion von [1] interessant. [16, 17, 18]

Ziel dieses Seminars ist, die Website des Fachgebiets "Verteilte Systeme" nachzuempfinden und mit einer gefälschten, aber ähnlichen URL bereitzustellen. Zusätzlich sollen bekannte Phishing-Attacken analysiert und diskutiert werden und ebenso wie man sich dagegen schützen kann (auch auf Provider/Email-Server-Seite).

### 2.2.6 Cross Site Request Forgery (XSRF, CSRF) und Session Riding

*Cross Site Request Forgery*<sup>6</sup> ist ein Angriff bei dem der Browser (oder das Client-Programm) eines Opfers ohne dessen Wissen und Einverständnis kompromittierende Anfragen/Kommandos an eine Website schickt. Oftmals sind Aktivitäten, die ein Benutzer auf einer Website ausführen kann, an bestimmte URLs geknüpft. Gelingt es einem Angreifer, einen eingeloggten User dazu zu bringen, die von ihm gewünschten Links zu besuchen, so wird dieser die vom Angreifer geplanten Aktivitäten durchführen. [19, 20, 21]

Es soll eine Website erstellt werden, die für CSRF Prinzipiell anfällig ist. Das Verhalten eines normalen Benutzers wird nachgestellt und dann anschließend durch eine der einschlägigen Methoden eine CSRF-Attacke durchgeführt. Danach werden Gegenmaßnahmen vorgestellt, die der Betreiber der Website und der Benutzer ergreifen können, um sich gegen CSRF zu wehren. Es wird demonstriert, wie wirkungsvolle oder unnütz diese Maßnahmen gegen die Attacke sind.

## 2.3 Unsicherheit in Kryptographiesystemen

Es werden jeweils Chiffren diskutiert, die heute als nicht mehr sicher gelten. Der Grund dafür wird ausgeführt und anhand eines konkreten Beispiels vorgeführt.

### 2.3.1 Security by Obfuscation

*Obfuscation*<sup>7</sup>, zu deutsch verdunkeln oder verwirren, wird im Kontext von Sicherheit oft als Ansatz verwendet, um Programmcode unkenntlich zu machen, der dann aber trotzdem noch ausführbar bleibt. Die Programmiersprache Perl ist in diesem Zusammenhang besonders hervorzuheben, da sie diese Technik implizit unterstützt.

---

<sup>5</sup><http://de.wikipedia.org/wiki/Phishing>

<sup>6</sup>[http://de.wikipedia.org/wiki/Cross-Site\\_Request\\_Forgery](http://de.wikipedia.org/wiki/Cross-Site_Request_Forgery)

<sup>7</sup><http://de.wikipedia.org/wiki/Obfuscator>

Codeobfuscation ist besonders bei Script-Sprachen verbreitet, wo es keine binäre Repräsentation bin, für die eine Rückübersetzung schwierig wäre. [22, 23]

Es sollen bekannte Verfahren zur Code Obfuscation vorgestellt, analysiert und diskutiert werden. Des Weiteren soll der Bereich Sicherheit, im Speziellen die Sicherheit vor Diebstahl betrachtet werden. Ist es sinnvoll und sicher, Programmcode auf diese Weise zu sichern?

### 2.3.2 Security by Obscurity

*Security by Obscurity*<sup>8</sup> bedeutet im Prinzip, Sicherheit darauf beruhen zu lassen, dass ein Angreifer nicht weiß, mit was er es zu tun hat. In der Kryptographie wird also zum Beispiel das eigentliche Verfahren geheim gehalten.

Diese Prinzip ist sehr umstritten und garantiert nicht für Sicherheit, da, sobald das Prinzip des zugrundeliegenden Algorithmus' bekannt geworden ist, das Verfahren nicht mehr eingesetzt werden kann. Bei modernen kryptographischen Algorithmen beruht die Sicherheit auf einen bekannten Algorithmus, der durch einen Schlüssel parametrisiert wird. [24, 25]

Es sollen die Vor- und Nachteile von Security by Obscurity dargelegt und diskutiert werden. Besonders wichtig sind hier bekanntgewordene Fälle aus der näheren Vergangenheit, bei denen die Geheimhaltung des Verfahrens die Sicherheit nicht garantieren konnte.

### 2.3.3 Sicherheit im Mobilfunk

Sicherheit im *Mobilfunk*<sup>9</sup> ist ein recht aktuelles Thema, denn eigentlich jede oder jeder besitzt (zumindest) ein Mobiltelefon. Da die Datenübertragung zwischen Mobiltelefon (Mobile Station, MS) und dem Mobilfunksendesysteme (Base Station Subsystem, BSS) kabellos vonstatten geht, ist hier ein möglicher Ansatzpunkt für Angriffe wie zum Beispiel das Abhören von Gesprächen. [26, 27, 28, 29]

Es sollen die Sicherheitsfunktionen und -schwachstellen in modernen Mobilfunk-Infrastrukturen diskutiert werden. Dies beinhaltet unter Anderem die Identifikation und Verschlüsselung im GSM und UMTS Netz. Desweiteren soll ein Überblick über aktuelle Entwicklungen gegeben werden.

## 2.4 Spionage und Forensik

Nach dem erfolgreichen Eindringen in ein Rechensystem beginnt die eigentliche Aufgabe des Angreifers: das (unauffällige) Sammeln und Auswerten der gewünschten Informationen.

---

<sup>8</sup>[http://de.wikipedia.org/wiki/Security\\_through\\_obscurity](http://de.wikipedia.org/wiki/Security_through_obscurity)

<sup>9</sup>[http://de.wikipedia.org/wiki/Global\\_System\\_for\\_Mobile\\_Communications](http://de.wikipedia.org/wiki/Global_System_for_Mobile_Communications), <http://de.wikipedia.org/wiki/UMTS>, und <http://de.wikipedia.org/wiki/IMSI-Catcher>

#### 2.4.1 Techniken zum Ausspähen von Daten (Spyware)

*Spyware*<sup>10</sup> bezeichnet eine Kategorie von Computerprogrammen, die Daten in einem Computer ausspäht und zum Beispiel an den Urheber der Software (oder an andere Personen/Programme) weiterleitet, ohne dazu berechtigt zu sein. Durch Spyware kann nicht nur Daten, sondern auch das Verhalten von Benutzern überwacht werden. Beispiele hierfür sind Programme, die das Surf-Verhalten von Computerbenutzern überwachen und an die, für das Spyware-Programm verantwortliche Firma weiterleiten. Auch sogenannte Keylogger, also Programme, die Tastatureingaben mitschneiden, fallen in die Kategorie von Spyware-Programmen.

Spyware richtet im Allgemeinen keine Schäden an, es wird jedoch die Privatsphäre verletzt, indem Daten ungewollt an Dritte ausgeliefert werden.

Phishing kann als eine andere, eher passive Art des Ausspionierens von Daten gesehen werden. Dabei wird zum Beispiel versucht, Daten mittels vermeintlich authentischer Emails oder Webseiten von Benutzern zu erfragen. [30, 31, 32]

Es sollen Informationen zu Spyware-Programmen, deren Programmierung und Funktionsweise beschaffen, analysiert und aufbereitet werden. Die Funktionsweise kann durch eine Beispielimplementierung veranschaulicht werden.

#### 2.4.2 Auswertung der Ausgespähten Daten (Forensik, siehe u.a. auch Bundestrojaner)

Das Thema *Datenforensik*<sup>11</sup> beschreibt die Aufgabe, komplexe Zusammenhänge aus vormals unzusammenhängenden Datenmengen abzuleiten. Dies ist vergleichbar mit den forensischen Aufgaben von Ermittlern bei Sicherheitsbehörden.

Datenforensik hat durch die aktuellen Entwicklungen nach dem 11. September 2001 und die damit einhergehende, zunehmende Überwachung besonders an Interessenspotential zugelegt.

Datenforensik wird aber auch im Zusammenhang mit der Rettung von Daten (Plattencrash) genutzt, um Zusammenhänge rekonstruieren zu können. [33, 34, 35]

Es sollen hier Informationen zur Datenforensik im Allgemeinen, aber auch zu speziellen Teilgebieten wie zum Beispiel dem sogenannten "Bundestrojaner" (auch Remote Forensic Software) gesammelt, aufbereitet und diskutiert werden. Hier ist auch besonders der rechtliche Rahmen zu betrachten.

<sup>10</sup><http://de.wikipedia.org/wiki/Spyware>

<sup>11</sup><http://de.wikipedia.org/wiki/Online-Durchsuchung> und <http://de.wikipedia.org/wiki/Forensik>

### 2.4.3 Datensicherheit in Social Networks

In den letzten Jahren haben Social Networks wie Facebook<sup>12</sup>, StudiVZ<sup>13</sup>, MySpace<sup>14</sup>, Xing<sup>15</sup>, LinkedIn<sup>16</sup> oder SchülerVZ<sup>17</sup> extrem an Bedeutung gewonnen. Diese ermöglichen den Benutzern, ein soziales Kontaktnetz aufzubauen und darin ihre Hobbies, Vorlieben, Gefühle und Aktivitäten darzustellen. Diese Daten können oftmals per Passwort geschützt werden. [36, 37, 38]

Dieses Thema kann in zwei Seminarvorträge aufgeteilt werden: Zum einen sind konkrete Statistiken über die bekanntesten Social Networks (mindestens fünf) interessant, die über die Entwicklung der Benutzerzahlen, deren Verteilung auf Geographie, Alter, Soziale Stufe etc. hinausgehen und auch analysieren, wie viele und welche Informationen über die Mitglieder speichern können, wie diese geschützt werden und wie (oder ob) die Mitglieder die Schutzmöglichkeiten nutzen.

Der zweite Vortrag würde sich auf konkrete Maßnahmen konzentrieren, wie diese Daten von dritten ausgespäht und ausgewertet werden können. Welche Informationen über eine Person kann man mit Google & Co. erhalten, wie viel findet man über dessen soziales Netzwerk heraus? Welche Mittel hat man dazu zur Verfügung und wie können diese eingesetzt werden? Wie lassen sich Informationen aus den sozialen Netzwerken mit Informationen von anderen Quellen (private Websites, Newsgroups, Blogs, Vereinswebsites, etc.) verbinden? Gibt es Möglichkeiten, sich effektiv in diesen Netzwerken zu schützen und trotzdem alle sozialen Aspekte mit seinen Freunden auszunutzen?

## 3 Ausarbeitung

Jeder teilnehmende Student reicht bis zum vorgegebenen Datum seine Ausarbeitung über das Konferenzsystem ab. Es erfolgt ein blinder Reviewprozess, bei dem die Ausarbeitung von drei anderen Studenten (ebenfalls Workshop-teilnehmer) kontrolliert wird. Die studentischen Reviewer geben Bewertungen und Verbesserungshinweise ab. Diese sind dann entsprechend zu beachten und umzusetzen, bevor die Final-Version eingereicht wird.

Die Finalversion der Ausarbeitung muss zwischen 15 und 20 Seiten bei Einzelabgaben, bzw. 30-35 Seiten bei Abgaben von einer Zweiergruppe, umfassen. Sie werden in dem Corporate-Design des KaSWoSDS geschrieben, dass über die Konferenzwebseite zugänglich ist. Dabei handelt es sich um einen L<sup>A</sup>T<sub>E</sub>X-Style, der exakt wie in diesem Dokument hier zu verwenden ist. Das vorliegende Dokument, welches mit demselben Style erstellt wurde, liegt ebenfalls als .tex-Datei auf der Konferenzseite vor.

---

<sup>12</sup><http://www.facebook.com/>

<sup>13</sup><http://www.studivz.net/>

<sup>14</sup><http://www.myspace.com/>

<sup>15</sup><http://www.xing.com/>

<sup>16</sup><http://www.linkedin.com/>

<sup>17</sup><http://www.schuelerVZ.de/>

```

1 latex meinPaper.tex
2 bibtex meinPaper
3 latex meinPaper.tex
4
5 dvips -Ppdf -o meinPaper.ps meinPaper.dvi

```

Listing 1: Übersetzung von meinPaper.tex nach .pdf

## 3.1 $\LaTeX$

Die Ausarbeitungen sind in  $\LaTeX$  zu erstellen. Da nun mal noch nicht jeder  $\LaTeX$  hat bzw. weiß, was das ist, wo er das herkriegern soll, und wie man damit umgeht, hier einige kurze Hinweise.

### 3.1.1 Wo man das herkriegt

$\LaTeX$  ist eine kostenlose Software, die sowohl für Windows als auch für Linux zur Verfügung steht. Die Windows-Variante heißt MikTeX und kann von <http://www.miktex.org/> heruntergeladen und anschließend installiert werden. Bei Linux hängt das von der Distribution ab, bei Ubuntu oder Debian kann man es z.B. installieren, indem man `apt-get install texlive` in die Shell eingibt und mit Enter bestätigt. Für andere Distribution weiß ich es nicht, aber vielleicht findet man mehr Infos unter <http://www.tug.org/texlive/>.

### 3.1.2 Wie man damit umgeht

$\LaTeX$  [39] ist kein *What You See Is What You Get*-Texteditor wie z.B. Word. Es handelt sich viel mehr um eine Art Mischung aus 80% einer Programmiersprache und einer 20% Markupsprache (wie z.B. HTML) die auf Knuth's  $\TeX$  [40] aufsetzt.  $\LaTeX$ -Dokumente sind also in erster Linie Textdateien, die mit Hilfe eines Compilers ein visuelles Format wie .dvi, .ps, und .pdf übersetzt werden müssen.

Nehmen wir an, man hat das Dokument `meinPaper.tex` erstellt und es befindet sich zusammen mit der Bibliographie-Datei `meinPaper.bib` (enthält die Literaturreferenzen) und unseren KaSWoSDS-Stilvorgaben in einem Verzeichnis. Zur Übersetzung muss man nun mehrere Kommandozeilenprogramme hintereinander aufrufen, wie in 1 dargestellt.

Mit dem Kommandozeilenbefehl `latex` erfolgt eine Übersetzung von `.tex` in das Zwischenformat `.dvi`. Dabei werden aber die Literaturreferenzen noch nicht eingefügt, das macht `bibtex`. Da sich durch das Einfügen der Referenzen einiges verschieben kann, muss man `latex` noch mal ausführen. Danach kann man dann die `.dvi`-Datei in ein Postscript-Dokument übersetzen, indem man `dvips` wie angegeben aufruft. Eine Umwandlung von Postscript nach `.pdf` schließlich lässt sich mit vielen Freeware-Tools bewerkstelligen – mein persönlicher Favorit ist Ghostscript, was man unter <http://pages.cs.wisc.edu/~ghost/> erhalten kann. Der Adobe Acrobat kann das aber auch.

## 3.2 Wie man das verwendet

Die Verwendung von L<sup>A</sup>T<sub>E</sub>X erfordert das Lernen einer Art Programmiersprache, was uns Informatikern eigentlich leicht fallen sollte. Da der Einstieg nicht immer leicht ist, hier ein paar Tipps für Tutorials. Die Referenzen können jeweils angeklickt werden und führen dann zu einem Link, wo man die downloaden kann.

- L<sup>A</sup>T<sub>E</sub>X eine Einführung und ein bisschen mehr... von Jürgens [41],
- L<sup>A</sup>T<sub>E</sub>X – Fortgeschrittene Anwendungen von Jürgens [42],
- The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X $\epsilon$  von Oetiker et al. [43],
- The L<sup>A</sup>T<sub>E</sub>X Companion von Mittelbach et al. [44],
- L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System von L<sup>A</sup>mpport [39].

## 3.3 Referenzen

Referenzen gehen besonders gut in L<sup>A</sup>T<sub>E</sub>X und sind sehr wichtig bei jeder Ausarbeitung und bei jedem wissenschaftlichen Beitrag. Auch in diesem Call-for-Papers werden ja häufig Referenzen, welche all in der Datei `bibliography.bib` abgelegt sind. Ein Eintrag in diese Datei sieht in etwa so aus wie in 2 dargestellt. Im Beispiel handelt es sich um einen technischen Bericht (`@techreport`), es gibt aber auch Einträge für z.B. Bücher (`@book`), Konferenzbeiträge (`@inproceedings`), oder beliebige Dokumente wie beispielsweise Internetseiten (`@misc`, `@manual`). Mit einem Komma vom Eintragstyp getrennt folgt dann der Identifikator, der dann (in unserem Beispiel) mit `\citep{RFC2006IDOSC}` oder `\citet{RFC2006IDOSC}` referenziert werden kann. Das führt dann zu [3] bzw. Handley and Rescorla [3] (die letzte Variante setzt auch die Autorennamen). Im Bibliographieeintrag kommen dann die Informationsfelder, für die unser `bibliography.bib` eine Menge Beispiele bietet. Beachte, dass wir die Kommandos `\OnlineAvailable` und `OnlineAvailableT` für das Einfügen von URLs bereitstellen. Dieser erfordern das Angeben eines Datums, was unbedingt immer gemacht werden muss.

## 4 Zusammenfassung

Der KaSWoSDS ist die Gelegenheit für Studenten der Informatik, sich mit dem Thema *Sicherheit in Verteilten Systemen* auseinanderzusetzen. Er bietet eine wertvolle praktische Erfahrung, der in fast allen Themenstellungen auch das aktive Erstellen von Beispielprogrammen gefordert wird. Er ist organisiert wie eine wissenschaftliche Konferenz und ermöglicht es somit zusätzlich, den Ablauf solcher Veranstaltung kennenzulernen. Wie im letzten Jahr [1] werden die Proceedings (bei ausreichend vielen guten Teilnehmern) als technischer Bericht veröffentlicht, wodurch die Teilnehmer eine erste wissenschaftliche Arbeit vorweisen können.

```

1 @techreport{RFC2006IDOSC,
2   title           = {Internet Denial-of-Service Considerations},
3   number          = {4732},
4   type            = {Request for Comments (RFC)},
5   institution     = {Network Working Group},
6   author          = {M. Handley and E. Rescorla},
7   month          = nov,
8   year            = {2006},
9   note           = {
10  \OnlineAvailable{http://tools.ietf.org/html/rfc4732}{2008-10-08}},
11 }

```

Listing 2: Ein Beispieleintrag aus unserer `bibliography.bib`.

## Literatur

- [1] Stephan Opfer, Stefan Triller, Stephan Scheuermann, Till Amma, Michael Blumenstein, and Ilhan Glogic. 1st kassel student workshop on security in distributed systems (kaswods'08), proceedings. Kasseler Informatikschriften (KIS) 2008, 1, February 13, 2008. urn:nbn:de:hebis:34-2008041421155. Published on April 14, 2008. Online available at <http://kobra.bibliothek.uni-kassel.de/handle/urn:nbn:de:hebis:34-2008041421155> (version 2008-04-15) and <http://www.it-weise.de/documents/index.html#PROC2008KASWOSDS> (version 2008-04-15).
- [2] Steve Gibson. *Distributed Reflection Denial of Service – Description and analysis of a potent, increasingly prevalent, and worrisome Internet attack*. Gibson Research Corporation, February 22 2002. Online verfügbar unter <http://www.grc.com/dos/drDOS.htm> [gelesen: 2008-10-08].
- [3] M. Handley and E. Rescorla. Internet denial-of-service considerations. Request for Comments (RFC) 4732, Network Working Group, November 2006. Online verfügbar unter <http://tools.ietf.org/html/rfc4732> [gelesen: 2008-10-08].
- [4] *Distributed Denial of Service (DDoS) Analyse der Angriffs-Tools – Erkenntnisse, Tendenzen, Auswertung*. Bundesamt für Sicherheit in der Informationstechnik, September 8 2000. Online verfügbar unter <http://www.bsi.de/fachthem/sinet/gefahr/toolsana.htm> [gelesen: 2008-10-08].
- [5] *Denial of Service (Dokumente): Destruktive Angriffe*. Computec.ch, 2005. Online verfügbar unter <http://www.computec.ch/download.php?list.7> [gelesen: 2008-10-08].
- [6] *Denial of Service (DoS) Attack Resources*. DenialInfo.Com, March 1 2006. Online verfügbar unter <http://www.denialinfo.com/> [gelesen: 2008-10-08].
- [7] Vivek Ramachandran. *Packet Sniffing using Raw Sockets*. Security-Freak.Net, September 16 2007. Online verfügbar unter <http://>

- [security-freak.net/raw-sockets/raw-sockets.html](http://security-freak.net/raw-sockets/raw-sockets.html) [gelesen: 2008-10-08].
- [8] *Sniffer: Kleine nützliche Vieche*. easy network, July 27 2007. Online verfügbar unter <http://www.easy-network.de/sniffer.html> [gelesen: 2008-10-08].
- [9] Robert Graham. *Sniffing (network wiretap, sniffer) FAQ*, September 14 2000. Version 0.3.3. Online verfügbar unter <http://web.archive.org/web/20050221103207/http://www.robertgraham.com/pubs/sniffing-faq.html> [gelesen: 2008-10-08].
- [10] Mircea Ciubotariu. *What next? Trojan.Linkoptimizer*. Symantec Security Response, Ireland, December 2006. Originally published by Virus Bulletin, Ltd. Online verfügbar unter <http://www.symantec.com/avcenter/reference/what.next.trojan.linkoptimizer.pdf> [gelesen: 2008-10-08].
- [11] *Die DEUTSCHEN Trojaner-Seiten*. Online verfügbar unter <http://www.trojaner-info.de/> [gelesen: 2008-10-08].
- [12] Alan Solomon and Gadi Evron. The world of botnets. *Virus Bulletin*, August 31 2006. Online verfügbar unter <http://www.beyondsecurity.com/whitepapers/SolomonEvronSept06.pdf> [gelesen: 2008-10-08].
- [13] *Computerviren (Dokumente): Computerviren, Würmer und Trojanische Pferde*. Computec.ch, 2007. Online verfügbar unter <http://www.computec.ch/download.php?list.14> [gelesen: 2008-10-08].
- [14] Konstantin Rozinov. *Reverse Code Engineering: An In-Depth Analysis of the Bagle Virus*. Bell Labs, Government Communication Laboratory, Internet Research, August 12 2004. Online verfügbar unter [http://rozinov.sfs.poly.edu/papers/bagle\\_analysis\\_v.1.0.pdf](http://rozinov.sfs.poly.edu/papers/bagle_analysis_v.1.0.pdf) [gelesen: 2008-10-08].
- [15] Mohammad Ahmadi Bidakhwidi. *Tutorial - Make Your Own mIRC Worm*. [www.ethicalhacker.net](http://www.ethicalhacker.net), 2005. Online verfügbar unter <http://www.ethicalhacker.net/content/view/26/2/> [gelesen: 2008-10-08].
- [16] Christopher Abad. The economy of phishing: A survey of the operations of the phishing market. *Firstmonday*, 10, July 21 2005. Online verfügbar unter [http://www.firstmonday.org/issues/issue10\\_9/abad/](http://www.firstmonday.org/issues/issue10_9/abad/) [gelesen: 2008-10-08].
- [17] David Watson, Thorsten Holz, and Sven Mueller. *Know your Enemy: Phishing - Behind the Scenes of Phishing Attacks*. The HoneyNet Project & Research Alliance, May 16 2005. Online verfügbar unter <http://www.honeynet.org/papers/phishing/> [gelesen: 2008-10-08].
- [18] Gunter Ollmann. *The Phishing Guide (Part 1) Understanding and Preventing Phishing Attacks*. [www.technicalinfo.net](http://www.technicalinfo.net), 2007. Online verfügbar unter <http://www.technicalinfo.net/papers/Phishing.html> [gelesen: 2008-10-08].

- [19] Norm Hardy. The confused deputy: (or why capabilities might have been invented). *SIGOPS Oper. Syst. Rev.*, 22(4):36–38, 1988. ISSN 0163-5980. Online verfügbar unter <http://portal.acm.org/citation.cfm?id=871709> [gelesen: 2008-10-08].
- [20] Thomas Schreiber. *Session Riding - A Widespread Vulnerability in Today's Web Applications*. SecureNet GmbH, December 2006. Online verfügbar unter [http://www.securenet.de/papers/Session\\_Riding.pdf](http://www.securenet.de/papers/Session_Riding.pdf) [gelesen: 2008-10-08].
- [21] Robert Auger. *The Cross-Site Request Forgery (CSRF/XSRF) FAQ*, August 17, 2008. Online verfügbar unter <http://www.cgisecurity.com/articles/csrf-faq.shtml> [gelesen: 2008-10-08].
- [22] Boaz Barak. *Can We Obfuscate Programs?*, 2003. Online verfügbar unter [http://www.cs.princeton.edu/~boaz/Papers/obf\\_informal.html](http://www.cs.princeton.edu/~boaz/Papers/obf_informal.html) [gelesen: 2008-10-08].
- [23] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs, 2001. Online verfügbar unter <http://citeseer.ist.psu.edu/article/barak01impossibility.html> und <http://citeseer.nj.nec.com/barak01impossibility.html> [gelesen: 2008-10-08].
- [24] *Skype bleibt aus Sicherheitsgründen proprietär*. heise online, iX, July 26 2007. Online verfügbar unter <http://www.heise.de/ix/news/meldung/93346> [gelesen: 2008-10-08].
- [25] Bruce Schneier. Secrecy, security, and obscurity. Technical report, Counterpane Internet Security, Inc., May 15 2002. Online verfügbar unter <http://www.schneier.com/crypto-gram-0205.html> [gelesen: 2008-10-08].
- [26] *CCC klont D2 Kundenkarte*. Chaos Computer Club, November 26 2001. Online verfügbar unter <http://www.ccc.de/gsm/> [gelesen: 2008-10-08].
- [27] Jeremy Quirke. *GSM Security Papers*, May 1 2004. Online verfügbar unter <http://www.gsm-security.net/gsm-security-papers.shtml> [gelesen: 2008-10-08].
- [28] Constantinos F. Grecas, Sotirios I. Maniatis, and Iakovos S. Venieris. Introduction of the asymmetric cryptography in gsm, gprs, umts, and its public key infrastructure integration. *Mobile Networks and Applications*, 8(2):145–150, 2003. ISSN 1383-469X. doi: <http://dx.doi.org/10.1023/A:1022285130956>. Online verfügbar unter <http://portal.acm.org/citation.cfm?id=772052.772057> [gelesen: 2008-10-08].
- [29] Erik Zenner, Rüdiger Weis, and Stefan Lucks. Sicherheit des gsm- verschlüsselungsstandards a5. *Datenschutz und Datensicherheit*, 24(7), 2000. Online verfügbar unter <http://www.cryptolabs.org/gsm/ZennerWeisLucksA5.pdf> [gelesen: 2008-10-08].

- [30] Thomas Weise. Spyware toolkit, April 22 2004. Online verfügbar unter <http://www.it-weise.de/documents/index.html#W2004SWTK> [gelesen: 2008-10-08].
- [31] *BSI für Bürger: Spyware*. Bundesamt für Sicherheit in der Informationstechnik, Godesberger Allee 185-189, 53175 Bonn, June 2007. Online verfügbar unter [http://www.bsi-fuer-buerger.de/druck/kap\\_abzocker.pdf](http://www.bsi-fuer-buerger.de/druck/kap_abzocker.pdf) [gelesen: 2008-10-08].
- [32] Nathaniel Good, Rachna Dhamija, Jens Grossklags, David Thaw, Steven Aronowitz, Deirdre Mulligan, and Joseph Konstan. Stopping spyware at the gate: A user study of privacy, notice and spyware. In Lorrie Faith Cranor, editor, *Symposium On Usable Privacy and Security (SOUPS) 2005*. Symposium On Usable Privacy and Security (SOUPS), July 2005. Online verfügbar unter <http://www.truststc.org/pubs/63.html> [gelesen: 2008-10-08].
- [33] *“Bundestrojaner” heißt jetzt angeblich “Remote Forensic Software”*. heise online, August 2007. Online verfügbar unter <http://www.heise.de/newsticker/meldung/93807> [gelesen: 2008-10-08].
- [34] *Bundestrojaner in ELSTER-Software entdeckt*. Chaos Computer Club, April 1 2007. Online verfügbar unter <http://www.ccc.de/updates/2007/bundestrojaner-elster> [gelesen: 2008-10-08].
- [35] *Wie funktioniert der Bundestrojaner?* tagesschau.de, August 30 2007. Online verfügbar unter <http://www.tagesschau.de/inland/meldung490134.html> [gelesen: 2008-10-08].
- [36] *Privatsphärenschutz in Soziale-Netzwerke-Plattformen*. FraunhoferInstitut für Sichere Informationstechnologie SIT, September 23, 2008. Online verfügbar unter [http://www.sit.fraunhofer.de/fhg/Images/SocNetStudie\\_Deu\\_Final\\_tcm105-132111.pdf](http://www.sit.fraunhofer.de/fhg/Images/SocNetStudie_Deu_Final_tcm105-132111.pdf) [gelesen: 2008-10-08].
- [37] Manuel Zamora-Morschhäuser. *Datenschutz und Sicherheit in Sozialen Netzwerken*, December 18, 2006. Online verfügbar unter [http://zamora.de/wordpress/wp-content/2006/12/soziale\\_netzwerke-vortrag.pdf](http://zamora.de/wordpress/wp-content/2006/12/soziale_netzwerke-vortrag.pdf) [gelesen: 2008-10-08].
- [38] Sicherheit in community und social network. Online verfügbar unter <http://www.computerbetrug.de/sicherheit-im-internet/communities-und-soziale-netzwerke-sicher-nutzen/> [gelesen: 2008-10-08].
- [39] Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994. ISBN 0-201-52983-1.
- [40] Donald E. Knuth. *The T<sub>E</sub>Xbook, Volume A of Computers and Typesetting*. Addison-Wesley, Reading, Massachusetts, second edition, 1984. ISBN 0-201-13448-9.

- [41] Manuela Jürgens.  $\LaTeX$  eine einföhrung und ein bisschen mehr... Technical Report A/026/0003, FernUniversität Gesamthochschule in Hagen, Universitätsrechenzentrum, Abt. Wissenschaftliche Anwendungen, Hagen, Deutschland, March 3 2000. Online verfügbar unter <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0260003.pdf> und <http://www.strz.uni-giessen.de/~holste/latex1.pdf> [gelesen: 2008-10-08].
- [42] Manuela Jürgens.  $\LaTeX$  – fortgeschrittene anwendungen – oder: neues von den hobbits... Technical Report A/027/9510, FernUniversität Gesamthochschule in Hagen, Universitätsrechenzentrum, Abt. Wissenschaftliche Anwendungen, Hagen, Deutschland, October 1 1995. Online verfügbar unter <http://www.fernuni-hagen.de/zmi/katalog/A027.shtml> [gelesen: 2008-10-08].
- [43] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. *The Not So Short Introduction to  $\LaTeX 2\epsilon$  – Or  $\LaTeX 2\epsilon$  in 138 minutes*, 4.22 edition, June 30 2007. Online verfügbar unter <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf> und <http://www.cs.usask.ca/grads/wew036/latex/lshort.pdf> [gelesen: 2008-10-08].
- [44] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The  $\LaTeX$  Companion*. Addison-Wesley, Reading, Massachusetts, second edition, 2004. ISBN 0-201-36299-6.