

# 1st Kassel Student Workshop on Security in Distributed Systems (KaSWoSDS'07) – CfP

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

21. Oktober 2007

## Zusammenfassung

Der 1. Kassel Student Workshop on Security in Distributed Systems, kurz KaSWoSDS'07, findet im Rahmen des Seminars Sicherheit in Verteilten Systemen im Wintersemester 2007 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 2007 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 als auch im Grundstudium teilnehmen. Erstmals wird das Seminar jedoch als Workshop (*1st Kassel Student Workshop on Security in Distributed Systems, KaSWoSDS'07*) organisiert und veranstaltet.

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



**VERTEILTE SYSTEME**

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.
- 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.

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 18.12. 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 praktische Teil erfordert das Erstellen eines vorführbaren Beispiels für die jeweilige Sicherheitslücke. Beim Thema *Buffer Overflow* wird z.B. ein Client-/Serversystem

Tabelle 1: Eckdaten des KaSWoSDS'07

Was	Wert
Titel:	1st Kassel Student Workshop on Security in Distributed Systems
Abkürzung:	KaSWoSDS'07
Lehrveranstaltung:	Sicherheit in Verteilten Systemen
Website:	<a href="http://www.vs.uni-kassel.de/teaching/ws2007/seminar/">http://www.vs.uni-kassel.de/teaching/ws2007/seminar/</a>
Veranstalter:	Prof. Dr. Kurt Geihs
Leiter:	Thomas Weise und Philipp Andreas Baer
Vorbesprechung:	23.10.2007, 16:00-17:00, Raum 1405
Einreichung:	18.12.2007
Review-Deadline:	08.01.2007
Final-Version:	15.01.2008
Workshop:	22.01.2008, 09:00-18:00, Raum 1405

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.

## 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 Buffer Overflows

*Pufferüberläufe*<sup>1</sup> sind eine der häufigsten Attacken auf Rechnernetze. Sie nutzen allgemeine Programmierfehler aus, die besonders in Sprachen wie C sehr schnell auftreten können: Eine interne Arrayvariable (meist eine lokale Variable auf dem Stack) wird vom Programmierer fest dimensioniert und als Speicher für eingehende Daten genutzt. Wenn der Programmierer nicht daran gedacht hat, zu überprüfen wie groß die Daten sind, die in diese Variable geschrieben werden, so

<sup>1</sup>siehe <http://de.wikipedia.org/wiki/Puffer%C3%BCberlauf>

kann man sie zum “überlaufen” bringen, indem man einfach sehr viele eingehende Daten hinschickt. Diese überschreiben dann z.B. Rücksprungadressen auf dem Stack oder direkt irgendwelchen Programmcode, wodurch man den Rechner dann übernehmen kann [Gre05, Don02].

Analysieren Sie dieses Szenario und stellen Sie eine solche Attacke nach. Erklären Sie, wie sich solche Attacken verhindern lassen. Tipp: In der Microsoft Windows API werden schon seit geraumer Zeit verbesserte Funktionen für den Umgang mit Daten angeboten [How04].

### 2.2.2 Cross-Site Scripting (XSS)

Beim *Cross-Site Scripting*<sup>2</sup> (auch XSS) geht es hauptsächlich darum, dem Besucher einer Internetseite Schadcode unterzuschieben. Dabei wird häufig eine andere, vertrauenswürdige Seite (die allerdings auch mangelnde Sicherheitsvorkehrungen hat) als Deckmantel benutzt. Ein Angreifer kann unter Umständen z.B. auf seiner Website einen Link zu einer vermeintlich sicheren Seite stellen. Dieser Link beinhaltet jedoch Schadcode, z.B. als Parameter für ein CGI-Skript der anderen Seite getarnt. Klickt der Nutzer auf diesen Link so baut der Server auf der vermeintlich sicheren Seite den Schadcode in seine Ausgabe, den HTML Code den der Internetnutzer angezeigt bekommt, ein. Dieser wird dann beim Klienten ausgeführt [www03, XSS, Djo07]

Analysieren Sie dieses Szenario und stellen Sie eine solche Attacke nach. Erklären Sie, wie sich solche Attacken (sowohl seitens der Server als auch der Surfer) verhindern lassen.

### 2.2.3 SQL-Injection

*SQL-Injection*<sup>3</sup> baut auf Schwachstellen in der Auswertung von eingehenden Daten auf der Serverseite auf. Der Gedanke ist z.B. folgender: Ein Login-Form erhält Benutzername und Passwort als Eingabe. Der Server schlägt diese dann in einer Datenbanktabelle mit Hilfe einer SQL-Abfrage nach, um festzustellen, ob sie korrekt sind. Wenn man anstelle des Benutzernamens nun einen SQL-Befehl eingibt, der z.B. einen neuen Benutzer einfügt oder auch ganze Tabellen löscht, so kann man immensen Schaden anrichten [Spe03, Anl02].

Analysieren Sie dieses Szenario und stellen Sie eine solche Attacke nach. Erklären Sie, wie sich solche Attacken verhindern lassen.

### 2.2.4 Spoofing

*Spoofing*<sup>4</sup> umfasst alle Methoden, die genutzt werden, um sich als jemand anders (Person, Prozess, System, Host, ...) auszugeben. Es gibt viele Spielarten:

---

<sup>2</sup>siehe [http://de.wikipedia.org/wiki/Cross-Site\\_Scripting](http://de.wikipedia.org/wiki/Cross-Site_Scripting)

<sup>3</sup>siehe <http://de.wikipedia.org/wiki/SQL-Injektion>

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

1. *ARP-Spoofing*<sup>5</sup> Das ARP-Protokoll wird genutzt, um IP-Adressen zu Netzwerkadressen zuzuordnen. Mit ARP-Spoofing können so genannte Man-in-the-Middle Attacks ausgeführt werden. Dabei benutzt ein Angreifer gefälschte ARP-Pakete, um jeweils zwei miteinander kommunizierenden Hosts vorzugaukeln, er wäre der jeweils andere Kommunikationspartner. Der Angreifer agiert dann als Relais zwischen den beiden, die ihm nun jeweils ihre Datenpakete zuschicken. Dadurch kann er allen Datenverkehr abhören. [Lei00, Gib05, Mon06]
2. *DNS-Spoofing*<sup>6</sup> DNS-Spoofing basiert auf dem selben Prinzip, setzt jedoch eine Stufe höher an. Anstelle der Zuordnung IP-Adresse↔Netzwerkadresse wird die Zuordnung von URLs zu IP-Adressen gefälscht.
3. *DHCP-Spoofing*<sup>7</sup> DHCP-Server teilen den Rechnern nicht nur IP-Adressen zu, sondern sie geben ihnen auch Informationen darüber, welche Gateways zu benutzen. Gelingt es einem Angreifer, sich als DHCP-Server zu tarnen, so kann er den Internetverkehr über einen anderen Rechner umleiten und somit gegebenenfalls belauschen.
4. *IP-Spoofing*<sup>8</sup> Hier werden IP-Pakete mit gefälschter Quelladresse verschickt, um den Empfänger über den Absender zu täuschen. Befinden sich Angreifer, Ziel und der fälschlicherweise als Absender angenommene Rechner im selben LAN, so kann man unter Umständen sogar TCP-Verbindungen vortäuschen. Es wird jedoch auch gerne für eine Art Denial-of-Service Attacke ausgenutzt. [Gib02, Tan03]
5. *Mac-Spoofing*<sup>9</sup> Hier nimmt der Angreifer die MAC-Adresse eines anderen Benutzers an, um Sicherheitsmechanismen zu täuschen.
6. *Mail-Spoofing*<sup>10</sup> Emails mit gefälschtem Absender werden verschickt. So kann Phishing betrieben werden.
7. *URL-Spoofing*<sup>11</sup> Der Angreifer nutzt eine URL, die einer vertrauenswürdigen URL sehr ähnlich sieht, um Internetnutzer zu täuschen. Wird die hinter der URL stehende Website dem Original entsprechend ähnlich gestaltet, so kann man Phishing betreiben. [Hei03b, Hei03a, Mic05]

Je nach Anzahl der Teilnehmer wird eines oder mehrere Themen dieser Gruppe (ausgenommen der letzten beiden Punkte) bearbeitet. Wieder wird die Attacke allgemein analysiert, ein praktisches Beispiel erarbeitet und es wird diskutiert, welche Gegenmaßnahmen getroffen werden können.

---

<sup>5</sup>siehe <http://de.wikipedia.org/wiki/ARP-Spoofing>

<sup>6</sup>siehe <http://de.wikipedia.org/wiki/DNS-Spoofing>

<sup>7</sup>siehe <http://de.wikipedia.org/wiki/DHCP#Sicherheit>

<sup>8</sup>siehe <http://de.wikipedia.org/wiki/IP-Spoofing>

<sup>9</sup>siehe <http://de.wikipedia.org/wiki/Mac-Spoofing>

<sup>10</sup>siehe <http://de.wikipedia.org/wiki/Mail-Spoofing>

<sup>11</sup>siehe <http://de.wikipedia.org/wiki/URL-Spoofing>

### 2.2.5 Denial-of-Service

*Denial-of-Service*-Attacken<sup>12</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 bei Thema 2.2.4, [Gib02]). [HR06, Bun00, Com05, Den06]

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

### 2.2.6 Sniffing

*Sniffer*<sup>13</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 abzuhehren. [Ram07, eas07, Gra00]

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

### 2.2.7 Trojaner

*Trojaner*<sup>14</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. [Ciu06, DDT, SE06, Com07].

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

### 2.2.8 Viren

Ein *Virus*<sup>15</sup> ist ein Programm, welches seinen Code in (mehrere) andere Programme einfügt und sich somit verbreitet. [VX 07, Bun97, Com07, Roz04]

Ziel dieses Vortrags ist es, ein harmloses, sich in einem vorgegebenen Rahmen selbst replizierendes Programm (also einen Virus) zu erstellen. Die zugrundeliegenden Mechanismen werden erläutert und es wird diskutiert, wie man gegen Viren vorgehen kann und welche allgemeinen Verteidigungsmaßnahmen es gibt.

### 2.2.9 Würmer

Ein *Wurm*<sup>16</sup> ist ein Programm, das sich über ein Netzwerk von Computern verbreitet. Es nutzt dazu höherwertige Ressourcen, Protokolle, Netzwerkdienste

---

<sup>12</sup>siehe [http://de.wikipedia.org/wiki/Denial\\_of\\_Service](http://de.wikipedia.org/wiki/Denial_of_Service)

<sup>13</sup>siehe <http://de.wikipedia.org/wiki/Sniffer>

<sup>14</sup>siehe [http://de.wikipedia.org/wiki/Trojanisches\\_Pferd\\_%28Computerprogramm%29](http://de.wikipedia.org/wiki/Trojanisches_Pferd_%28Computerprogramm%29)

<sup>15</sup>siehe <http://de.wikipedia.org/wiki/Computervirus>

<sup>16</sup>siehe <http://de.wikipedia.org/wiki/Computerwurm>

oder Benutzerinteraktionen. Würmer können auch dazu benutzt werden, Trojaner auf dem Zielsystem abzusetzen. [Roz04, Bid05]

Genau wie bei Thema 2.2.8 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.

### 2.2.10 Phishing

Beim *Phishing*<sup>17</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 letzten beiden Punkte von Sektion 2.2.4 interessant. [Aba05, WHM05, Oll07]

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).

## 2.3 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 Klassische Verschlüsselungs- und Angriffsverfahren

Die klassischen Verfahren in der *Kryptographie*<sup>18</sup> sind wegen der vielen verschiedenen Ansätze, aber vor allem der sich dahinter verbergenden Geschichte interessant. Bereits die alten Ägypter waren in der Lage, Texte oder Informationen zu verschlüsseln.

Ein sehr interessantes, aus heutiger Sicht aber informationstheoretisch unsicheres Verfahren, ist zum Beispiel das Scytale. Es handelt sich hierbei um einen Papierstreifen durch der, schneckenförmig um einen Stock gewickelt, eine Transpositionschiffre repräsentiert.

Weitreichenden Einfluss auf den Ausgang von Kriegen hatte die Kryptographie spätestens seit dem ersten Weltkrieg, als die Deutschen die Chiffren ADFGX und deren Nachfolger ADFGVX im Einsatz hatten. Im zweiten Weltkrieg wurde schließlich die ENIGMA, die wohl bekannteste Verschlüsselungsmaschine eingesetzt. Das Brechen der ENIGMA-Verschlüsselung durch zuerst polnische und später dann britische Mathematiker (Bletchley Park; Bombe), hatte dann eine entscheidende Auswirkung auf den Kriegsausgang. [Ess, Zim]

<sup>17</sup>siehe <http://de.wikipedia.org/wiki/Phishing>

<sup>18</sup>siehe <http://de.wikipedia.org/wiki/Kryptographie>

Ziel dieses Themas ist es, klassische Verfahren der Kryptographie sowie der Kryptoanalyse zu studieren und zu diskutieren. Da es sich hier um einen sehr komplexen und weitreichenden Themenkomplex handelt, ist dieses Thema vorzugsweise von zwei Studenten zu bearbeiten.

### 2.3.2 Security by Obfuscation

*Obfuscation*<sup>19</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.

Code-Obfuscation ist besonders bei Script-Sprachen verbreitet, wo es keine binäre Repräsentation bin, für die eine Rückübersetzung schwierig wäre. [Bar03, BGI<sup>+</sup>01]

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?

### 2.3.3 Security by Obscurity

*Security by Obscurity*<sup>20</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. [hei07b, Sch02]

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.

### 2.3.4 Sicherheit im Mobilfunk

Sicherheit im *Mobilfunk*<sup>21</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. [Cha01, Qui04, GMV03, ZWL00]

<sup>19</sup>siehe <http://de.wikipedia.org/wiki/Obfuscator>

<sup>20</sup>siehe [http://de.wikipedia.org/wiki/Security\\_through\\_obscurity](http://de.wikipedia.org/wiki/Security_through_obscurity)

<sup>21</sup>siehe [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>

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.

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

*Spyware*<sup>22</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 authentischen Emails oder Webseiten von Benutzern zu erfragen. [Wei04, Bun07, GDG<sup>+</sup>05]

Es sollen Informationen zu Spyware-Programmen, deren Programmierung und Funktionsweise beschaffte, 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>23</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.

---

<sup>22</sup> siehe <http://de.wikipedia.org/wiki/Spyware>

<sup>23</sup> siehe <http://de.wikipedia.org/wiki/Online-Durchsuchung> und <http://de.wikipedia.org/wiki/Forensik>

Datenforensik wird aber auch im Zusammenhang mit der Rettung von Daten (Plattencrash) genutzt, um Zusammenhänge rekonstruieren zu können. [hei07a, Cha07, tag07]

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.

## 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 Workshopteilnehmer) kontrolliert wird. Die studentischen Reviwer 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  $\LaTeX$ -Style, der exakt wie in diesem Dokument hier zu verwenden ist. Das vorliegende Dokument, welches mit dem selben Style erstellt wurde, liegt ebenfalls als `.tex`-Datei auf der Konferenzseite vor.

### 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

$\LaTeX$  [Lam94] ist kein *What You See Is What You Get*-Texteditor wie z.B. Word. Es handelt sich viel mehr um eine Mischung aus Programmier- und Markupsprache (wie z.B. HTML) die auf Knuths  $\TeX$  [Knu84] aufsetzt.  $\LaTeX$ -

```

1 latex meinPaper.tex
2 bibtex meinPaper
3 latex meinPaper.tex
4
5 dvips meinPaper.dvi
6 dvi2pdf meinPaper.dvi

```

Listing 1: Übersetzung von meinPaper.tex über .dvi nach .pdf

```

1 pdflatex meinPaper.tex
2 bibtex meinPaper
3 pdflatex meinPaper.tex

```

Listing 2: Übersetzung von meinPaper.tex direkt nach .pdf

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 Listing 1 oder Listing 2 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. `dvipdf` wandelt das `.dvi` direkt in ein `.pdf` um.

`pdflatex` erspart einem den Weg über `.dvi`. Es wird direkt ein `.pdf` Dokument erstellt, ohne dass noch ein Transformationsschritt nötig ist. Hier muss beachtet werden, dass keine `.eps`-Grafiken eingebunden werden können. Stattdessen werden `.pdf`, `.jpg` und `.png` direkt unterstützt. Eventuell vorhandene `.eps` Grafiken können zum Beispiel mit `epstopdf` verlustfrei konvertiert werden. `epstopdf` ist Teil der T<sub>E</sub>XLive-Distribution.

### 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.

- $\LaTeX$  eine Einführung und ein bisschen mehr... [Jür00],
- $\LaTeX$  – Fortgeschrittene Anwendungen [Jür95],
- The Not So Short Introduction to  $\LaTeX 2\epsilon$  [OPHS07],
- The  $\LaTeX$  Companion [MGB<sup>+</sup>04],
- $\LaTeX$ : A Document Preparation System [Lam94].

## 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. Weiterhin werden die Proceedings (bei ausreichend vielen guten Teilnehmern) als technischer Bericht veröffentlicht, wodurch die Teilnehmer eine erste wissenschaftliche Arbeit vorweisen können.

## Literatur

- [Aba05] ABAD, Christopher: The Economy of Phishing: A Survey of the Operations of the Phishing Market. In: *Firstmonday* 10 (2005), Juli 21. – Online verfügbar unter [http://www.firstmonday.org/issues/issue10\\_9/abad/](http://www.firstmonday.org/issues/issue10_9/abad/)
- [Anl02] ANLEY, Chris ; NEXT GENERATION SECURITY SOFTWARE LTD, NGSSSOFTWARE INSIGHT SECURITY RESEARCH (NISR) (Hrsg.): *Advanced SQL Injection In SQL Server Applications*. Next Generation Security Software Ltd, NGSSoftware Insight Security Research (NISR), Januar 31 2002. – Online verfügbar unter [http://www.nextgenss.com/papers/advanced\\_sql\\_injection.pdf](http://www.nextgenss.com/papers/advanced_sql_injection.pdf)
- [Bar03] BARAK, Boaz: *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)
- [BGI<sup>+</sup>01] BARAK, Boaz ; GOLDREICH, Oded ; IMPAGLIAZZO, Russell ; RUDICH, Steven ; SAHAI, Amit ; VADHAN, Salil ; YANG, Ke: *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>

- [Bid05] BIDAHWIDI, Mohammad A. ; WWW.ETHICALHACKER.NET (Hrsg.): *Tutorial - Make Your Own mIRC Worm*. www.ethicalhacker.net, 2005. – Online verfügbar unter <http://www.ethicalhacker.net/content/view/26/2/>
- [Bun97] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (Hrsg.): *BSI-Publikation: "Informationen zu Computer-Viren" (Schriftenreihe zur IT-Sicherheit)*. Bundesamt für Sicherheit in der Informationstechnik, April 1997. – Band 2 - 2. erweiterte Auflage. Online verfügbar unter <http://www.bsi.de/av/virbro/index.htm>
- [Bun00] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (Hrsg.): *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>
- [Bun07] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (Hrsg.): *BSI für Bürger: Spyware*. Godesberger Allee 185-189, 53175 Bonn: Bundesamt für Sicherheit in der Informationstechnik, Juni 2007. – Online verfügbar unter [http://www.bsi-fuer-buerger.de/druck/kap\\_abzocker.pdf](http://www.bsi-fuer-buerger.de/druck/kap_abzocker.pdf)
- [Cha01] CHAOS COMPUTER CLUB (Hrsg.): *CCC klont D2 Kundenkarte*. Chaos Computer Club, November 26 2001. – Online verfügbar unter <http://www.ccc.de/gsm/>
- [Cha07] CHAOS COMPUTER CLUB (Hrsg.): *Bundestrojaner in ELSTER-Software entdeckt*. Chaos Computer Club, April 1 2007. – Online verfügbar unter <http://www.ccc.de/updates/2007/bundestrojaner-elster>
- [Ciu06] CIUBOTARIU, Mircea ; SYMANTEC SECURITY RESPONSE, IRELAND (Hrsg.): *What next? Trojan.Linkoptimizer*. Symantec Security Response, Ireland, Dezember 2006. – Originally published by Virus Bulletin, Ltd. Online verfügbar unter <http://www.symantec.com/avcenter/reference/what.next.trojan.linkoptimizer.pdf>
- [Com05] COMPUTEC.CH (Hrsg.): *Denial of Service (Dokumente): Destruktive Angriffe*. Computec.ch, 2005. – Online verfügbar unter <http://www.computec.ch/download.php?list.7>
- [Com07] COMPUTEC.CH (Hrsg.): *Computerviren (Dokumente): Computerviren, Würmer und Trojanische Pferde*. Computec.ch, 2007. – Online verfügbar unter <http://www.computec.ch/download.php?list.14>

- [DDT] *Die DEUTSCHEN Trojaner-Seiten. : Die DEUTSCHEN Trojaner-Seiten.* – Online verfügbar unter <http://www.trojaner-info.de/>
- [Den06] DENIALINFO.COM (Hrsg.): *Denial of Service (DoS) Attack Resources.* DenialInfo.Com, März 1 2006. – Online verfügbar unter <http://www.denialinfo.com/>
- [Djo07] DJOEDJOE ; ELITEHACKERS.INFO (Hrsg.): *Basic XSS Tutorial.* EliteHackers.info, Juli 2007. – Online verfügbar unter <http://www.elitehackers.info/forums/showthread.php?p=52491>
- [Don02] DONALDSON, Mark E. ; SANS INSTITUTE (Hrsg.): *Inside the Buffer Overflow Attak: Mechanism, Method, & Prevention.* Sans Institute, April 3 2002. – GSEC Version 1.3. Online verfügbar unter [https://www2.sans.org/reading\\_room/whitepapers/securecode/386.php?id=386&cat=securecode](https://www2.sans.org/reading_room/whitepapers/securecode/386.php?id=386&cat=securecode)
- [eas07] EASY NETWORK (Hrsg.): *Sniffer: Kleine nützliche Vieche.* easy network, Juli 27 2007. – Online verfügbar unter <http://www.easy-network.de/sniffer.html>
- [Ess] ESSLINGER, Bernhard: *Zeittafel/Zeitreise durch Kryptographie und Kryptoanalyse.* – Online verfügbar unter [http://www.cryptool.de/menu\\_zeittafel.de.html](http://www.cryptool.de/menu_zeittafel.de.html)
- [GDG<sup>+</sup>05] GOOD, Nathaniel ; DHAMIJA, Rachna ; GROSSKLAGS, Jens ; THAW, David ; ARONOWITZ, Steven ; MULLIGAN, Deirdre ; KONSTAN, Joseph: *Stopping Spyware at the Gate: A User Study of Privacy, Notice and Spyware.* In: CRANOR, Lorrie F. (Hrsg.) ; Symposium On Usable Privacy and Security (SOUPS) (Veranst.): *Symposium On Usable Privacy and Security (SOUPS) 2005* Symposium On Usable Privacy and Security (SOUPS), 2005. – Online verfügbar unter <http://www.truststc.org/pubs/63.html>
- [Gib02] GIBSON, Steve ; GIBSON RESEARCH CORPORATION (Hrsg.): *Distributed Reflection Denial of Service – Description and analysis of a potent, increasingly prevalent, and worrisome Internet attack.* Gibson Research Corporation, Februar 22 2002. – Online verfügbar unter <http://www.grc.com/dos/drDOS.htm>
- [Gib05] GIBSON RESEARCH CORPORATION (Hrsg.): *ARP Cache Poisoning – How one bad machine on your Ethernet Local Area Network (LAN) can ruin your whole day.* Gibson Research Corporation, Dezember 11 2005. – Online verfügbar unter <http://www.grc.com/nat/arp.htm>
- [GMV03] GRECAS, Constantinos F. ; MANIATIS, Sotirios I. ; VENIERIS, Iakovos S.: *Introduction of the asymmetric cryptography in GSM,*

- GPRS, UMTS, and its public key infrastructure integration. In: *Mobile Networks and Applications* 8 (2003), Nr. 2, S. 145–150. <http://dx.doi.org/http://dx.doi.org/10.1023/A:1022285130956>. – DOI <http://dx.doi.org/10.1023/A:1022285130956>. – ISSN 1383–469X. – Online verfügbar unter <http://portal.acm.org/citation.cfm?id=772052.772057>
- [Gra00] GRAHAM, Robert: *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>
- [Gre05] GREG, Isaac: An Overview and Example of the Buffer-Overflow Exploit. In: *IAnewsletter* 7 (2005), Spring, Nr. 4, S. 16–21. – Online verfügbar unter [http://iac.dtic.mil/iatac/download/Vol17\\_No4.pdf](http://iac.dtic.mil/iatac/download/Vol17_No4.pdf)
- [Hei03a] HEISE SECURITY (Hrsg.): *Falsche URLs auch unter Mozilla*. Heise Security, Dezember 15 2003. – Online verfügbar unter <http://www.heise.de/newsticker/meldung/42942>
- [Hei03b] HEISE SECURITY (Hrsg.): *Gefälschte URLs im Internet Explorer [Update]*. Heise Security, Dezember 9 2003. – Online verfügbar unter <http://www.heise.de/security/news/meldung/42768>
- [hei07a] HEISE ONLINE (Hrsg.): *“Bundestrojaner” heißt jetzt angeblich “Remote Forensic Software”*. heise online, August 2007. – Online verfügbar unter <http://www.heise.de/newsticker/meldung/93807>
- [hei07b] HEISE ONLINE, iX (Hrsg.): *Skype bleibt aus Sicherheitsgründen proprietär*. heise online, iX, Juli 26 2007. – Online verfügbar unter <http://www.heise.de/ix/news/meldung/93346>
- [How04] HOWARD, Michael ; MICROSOFT, MSDN DEUTSCHLAND (Hrsg.): *C-Laufzeitbibliotheken sichern*. Microsoft, MSDN Deutschland, November 14 2004. – Online verfügbar unter <http://www.microsoft.com/germany/msdn/library/net/visualstudio/CLaufzeitbibliothekenSichern.msp>
- [HR06] HANDLEY, M. ; RESCORLA, E.: Internet Denial-of-Service Considerations / Network Working Group. 2006 (4732). – Request for Comments (RFC). – Online verfügbar unter <http://tools.ietf.org/html/rfc4732>
- [Jür95] JÜRGENS, Manuela: *L<sup>A</sup>T<sub>E</sub>X – Fortgeschrittene Anwendungen – oder: neues von den Hobbits...* / FernUniversität Gesamthochschule in Hagen, Universitätsrechenzentrum, Abt. Wissenschaftliche Anwendungen. Hagen, Deutschland, Oktober 1 1995 (A/027/9510). – Forschungsbericht. – Online verfügbar unter <http://www.fernuni-hagen.de/zmi/katalog/A027.shtml>

- [Jür00] JÜRGENS, Manuela:  $\LaTeX$  eine Einführung und ein bisschen mehr... / FernUniversität Gesamthochschule in Hagen, Universitätsrechenzentrum, Abt. Wissenschaftliche Anwendungen. Hagen, Deutschland, März 3 2000 (A/026/0003). – Forschungsbericht. – 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>
- [Knu84] KNUTH, Donald E.: *The  $\TeX$ book, Volume A of Computers and Typesetting*. second. Reading, Massachusetts : Addison-Wesley, 1984. – ISBN 0–201–13448–9
- [Lam94] LAMPORT, Leslie:  *$\LaTeX$ : A Document Preparation System*. second. Reading, Massachusetts : Addison-Wesley, 1994. – ISBN 0–201–52983–1
- [Lei00] LEITNER, Felix von ; CODE BLAU SECURITY CONCEPTS (Hrsg.): *arprelay*. Code Blau Security Concepts, Dezember 17 2000. – Online verfügbar unter <http://www.fefe.de/arprelay/>
- [MGB<sup>+</sup>04] MITTELBACH, Frank ; GOOSSENS, Michel ; BRAAMS, Johannes ; CARLISLE, David ; ROWLEY, Chris: *The  $\LaTeX$  Companion*. second. Reading, Massachusetts : Addison-Wesley, 2004. – ISBN 0–201–36299–6
- [Mic05] MICROSOFT, MICROSOFT KNOWLEDGE BASE (Hrsg.): *Schritte, die helfen können, gefälschte (SSpoof) Websites und böswillige Hyperlinks zu erkennen und sich vor ihnen zu schützen*. Microsoft, Microsoft Knowledge Base, September 12 2005. – Artikel-ID: 833786. Version: 11.0. Online verfügbar unter <http://support.microsoft.com/?id=833786>
- [Mon06] MONTORO, Massimiliano ; OXID.IT (Hrsg.): *Cain & Abel – User Manual*. oXid.it, 2001–2006. – Online verfügbar unter [http://www.oxid.it/ca\\_um/](http://www.oxid.it/ca_um/)
- [Oll07] OLLMANN, Gunter ; WWW.TECHNICALINFO.NET (Hrsg.): *The Phishing Guide (Part 1) Understanding and Preventing Phishing Attacks*. www.technicalinfo.net, 2007. – Online verfügbar unter <http://www.technicalinfo.net/papers/Phishing.html>
- [OPHS07] OETIKER, Tobias ; PARTL, Hubert ; HYNA, Irene ; SCHLEGL, Elisabeth: *The Not So Short Introduction to  $\LaTeX$ 2 $\epsilon$  – Or  $\LaTeX$ 2 $\epsilon$  in 138 minutes*. 4.22, Juni 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>. Eine deutsche Version ist unter <ftp://ftp.dante.de/tex-archive/info/lshort/german/l2kurz.pdf> verfügbar.

- [Qui04] QUIRKE, Jeremy: *GSM Security Papers*, Mai 1 2004. – Online verfügbar unter <http://www.gsm-security.net/gsm-security-papers.shtml>
- [Ram07] RAMACHANDRAN, Vivek ; SECURITY-FREAK.NET (Hrsg.): *Packet Sniffing using Raw Sockets*. Security-Freak.Net, September 16 2007. – Online verfügbar unter <http://security-freak.net/raw-sockets/raw-sockets.html>
- [Roz04] ROZINOV, Konstantin ; BELL LABS, GOVERNMENT COMMUNICATION LABORATORY, INTERNET RESEARCH (Hrsg.): *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)
- [Sch02] SCHNEIER, Bruce: *Secrecy, Security, and Obscurity / Counterpane* Internet Security, Inc. 2002. – Forschungsbericht. – Online verfügbar unter <http://www.schneier.com/crypto-gram-0205.html>
- [SE06] SOLOMON, Alan ; EVRON, Gadi: *The World of Botnets*. In: *Virus Bulletin* (2006), August 31. – Online verfügbar unter <http://www.beyondsecurity.com/whitepapers/SolomonEvronSept06.pdf>
- [Spe03] SPENT, Kevin ; SPI LABS (Hrsg.): *SQL Injection – Are your web applications vulnerable?* SPI Labs, September 10 2003. – Online verfügbar unter <http://www.spidynamics.com/support/whitepapers/WhitepapersSQLInjection.pdf>
- [tag07] TAGESSCHAU.DE (Hrsg.): *Wie funktioniert der Bundestrojaner?* tagesschau.de, August 30 2007. – Online verfügbar unter <http://www.tagesschau.de/inland/meldung490134.html>
- [Tan03] TANASE, Matthew ; SECURITYFOCUS (Hrsg.): *IP Spoofing: An Introduction*. SecurityFocus, März 11 2003. – Online verfügbar unter <http://www.securityfocus.com/infocus/1674>
- [VX 07] VX HEAVENS (Hrsg.): *Tutorials*. VX Heavens, Mai 28 2007. – Viele Links zu Tutorials über das Erstellen von Viren. Online verfügbar unter <http://vx.netlux.org/lib/static/vdat/tutorial.htm>
- [Wei04] WEISE, Thomas: *Spyware Toolkit*. April 22 2004 Online verfügbar unter <http://www.it-weise.de/documents/index.html#W2004SWTK>
- [WHM05] WATSON, David ; HOLZ, Thorsten ; MUELLER, Sven ; THE HONEYNET PROJECT & RESEARCH ALLIANCE (Hrsg.): *Know your Enemy: Phishing – Behind the Scenes of Phishing Attacks*. The HoneyNet Project & Research Alliance, Mai 16 2005. – Online verfügbar unter <http://www.honeynet.org/papers/phishing/>

- [www03] WWW.CGISEcurity.COM (Hrsg.): *The Cross Site Scripting (XSS) FAQ*. www.cgisecurity.com, August 2003. – Online verfügbar unter <http://www.cgisecurity.com/articles/xss-faq.shtml>
- [XSS] *XSS Introduction by Steve.* : *XSS Introduction by Steve.* – Online verfügbar unter <http://www.steve.org.uk/Hacks/XSS/index.html>. Sie auch <http://www.steve.org.uk/Hacks/>
- [Zim] ZIMBELMANN, Johanna: *Geschichte der Kryptographie.* – Online verfügbar unter <http://krypto.informatik.fh-augsburg.de/geschichte.htm>
- [ZWL00] ZENNER, Erik ; WEIS, Rüdiger ; LUCKS, Stefan: Sicherheit des GSM- Verschlüsselungsstandards A5. In: *Datenschutz und Datensicherheit* 24 (2000), Nr. 7. – Online verfügbar unter <http://www.cryptolabs.org/gsm/ZennerWeisLucksa5.pdf>