



FernUniversität  
in Hagen

Fakultät für Mathematik und Informatik

## Hausarbeit

im Seminar 21817 „IT-Sicherheit“

<b>Thema:</b>	Aspekte der Sicherheit in der Programmierung
<b>Autor:</b>	Florian Mahlecke <florian.mahlecke@cirosec.de> MatNr. 9.....
<b>Autor:</b>	Kirsten Katharina Roschanski <studium@kirsten-roschanski.de> MatNr. 9053522
<b>Version vom:</b>	20. Mai 2013
<b>Betreuer:</b>	Dipl. Inf. Daniel Berg

## Eidesstattliche Erklärung

### Eidesstattliche Erklärung zur Hausarbeit

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

*Unterschrift :*

*Ort, Datum :*

## Eidesstattliche Erklärung

### Eidesstattliche Erklärung zur Hausarbeit

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

*Unterschrift :*

*Ort, Datum :*

# Inhaltsverzeichnis

<b>Eidesstattliche Erklärung</b>	<b>2</b>
<b>Abkürzungsverzeichnis</b>	<b>5</b>
<b>1 Einleitung</b>	<b>5</b>
<b>2 Programmierfehler</b>	<b>5</b>
<b>3 Angriffsszenarien für bekannte Sicherheitslücken</b>	<b>6</b>
<b>4 Sicherer Code und Robuste Programmierung</b>	<b>6</b>
<b>5 Fazit</b>	<b>6</b>
<b>Literaturverzeichnis</b>	<b>7</b>
<b>Anhang</b>	<b>8</b>
<b>Listingverzeichnis</b>	<b>9</b>
<b>Abbildungsverzeichnis</b>	<b>10</b>
<b>Tabellenverzeichnis</b>	<b>11</b>

## 1 Einleitung

In kommerziellen Softwareprojekten arbeitet ein geschlossener Kreis an Personen an dem Code. Dieser wird dann in der Regel vor Veröffentlichung durch entsprechende Experten geprüft und freigegeben.

Bei OpenSource Projekten, wo der der Quellcode frei zugänglich ist und eine Mitarbeit auch gewünscht ist, kann hingegen jeder der programmieren kann oder auch denkt den Code erweitern und eigene Module entwickeln.

Bei vielen Modulen wird dabei oft die API ignoriert oder es schleichen sich schnell kritische Sicherheitlücken ein.

In der nachfolgenden Hausarbeit geht es um die Sicherheit in der Programmierung. Dabei soll auf typische Fehler eingegangen werden und anhand von unterschiedlichen Angriffsszenarien, Techniken und Technologien gezeigt werden wo die häufigsten Schwachstellen liegen und wie man diese ganz einfach beseitigen kann. Ausgangspunkt ist die Modulare Objekt orientierter Programmierung, die heute in fast allen Softwareprojekten zur Anwendung kommt.

## 2 Programmierfehler

### Überblick

In diesem Kapietel soll es um oft gemacht und gern immer wieder auftretenden Programmierfehler gehen. Dabei soll der Schwerpunkt auf cross site scripting (XSS) gelegt werden und die Verwendung von Access modifiers in der Objektorientierten Programmierung.

Public - If you can see the class, then you can see the method

Private - If you are part of the class, then you can see the method, otherwise not.

Protected - Same as Private, plus all descendants can also see the method.

Static (class) - Remember the distinction between "Class" and "Object" ? Forget all that. They are the same with "static"... the class is the one-and-only instance of itself.

Static (method) - Whenever you use this method, it will have a frame of reference independent of the actual instance of the class it is part of.

### **3 Angriffsszenarien für bekannte Sicherheitslücken**

#### **Überblick**

In diesem Kapitel soll es um bekannte Sicherheitslücken gehen, die häufig von Hackern als erstes Angriffsziel dienen.

### **4 Sicherer Code und Robuste Programmierung**

#### **Überblick**

Hier soll der Frage nachgegangen werden, welche Bedrohungen von schadhaften Code ausgehen.

### **5 Fazit**

## **Zusammenfassung**

### **Abstract**

## **Anhang**



## **Listingverzeichnis**

## **Abbildungsverzeichnis**

## **Tabellenverzeichnis**