

Angewandte Informatik Hardwarenahe Programmierung

SPECIAL

Prof. Dr. rer. nat. Peter Gerwinski

2. Januar 2017

Angewandte Informatik

Hardwarenahe Programmierung

SPECIAL

U Software und Urheberrecht

U.1 Überblick

U.2 Lizenzmodelle

U.3 Beispiel-Lizenzen

U.4 Fazit

X Exploits

X.1 Einfache Angriffe

X.2 Puffer-Überläufe

X.3 Return-Oriented Programming

X.4 Fazit

U Software und Urheberrecht

U.0 Vorab: Dies ist keine Rechtsdienstleistung!

§ 2 RDG: Begriff der Rechtsdienstleistung

(1) Rechtsdienstleistung ist jede Tätigkeit in **konkreten** fremden Angelegenheiten, sobald sie eine rechtliche Prüfung des **Einzelfalls** erfordert.

Quelle: <http://www.gesetze-im-internet.de/rdg/>

Konkreter Fall:

- Konkretes Programm unter Lizenz A
- Konkrete Bibliothek unter Lizenz B
- Beratung: Paßt das zusammen?

→ Rechtsdienstleistung

U Software und Urheberrecht

U.1 Überblick

- Grundlage: Urheberrecht (\approx Copyright)
„Wer Software schreibt, entscheidet, was damit geschehen darf.“
- anderen etwas erlauben: Lizenz
- „gar keine Lizenz“ = alles verboten



U Software und Urheberrecht

U.1 Überblick

- Grundlage: Urheberrecht (\approx Copyright)
„Wer Software schreibt, entscheidet, was damit geschehen darf.“
- anderen etwas erlauben: Lizenz
- „gar keine Lizenz“ = alles verboten



Lizenzmodelle

kommerziell

nichtkommerziell

U Software und Urheberrecht

U.1 Überblick

- Grundlage: Urheberrecht (\approx Copyright)
„Wer Software schreibt, entscheidet, was damit geschehen darf.“
- anderen etwas erlauben: Lizenz
- „gar keine Lizenz“ = alles verboten



Lizenzmodelle

	proprietär	frei
kommerziell		
nichtkommerziell		

U Software und Urheberrecht

U.1 Überblick

- Grundlage: Urheberrecht (\approx Copyright)
„Wer Software schreibt, entscheidet, was damit geschehen darf.“
- anderen etwas erlauben: Lizenz
- „gar keine Lizenz“ = alles verboten



Lizenzmodelle	proprietär	frei	
		freizügig	Copyleft
kommerziell			
nichtkommerziell			

U Software und Urheberrecht

U.1 Überblick

- Grundlage: Urheberrecht (\approx Copyright)
„Wer Software schreibt, entscheidet, was damit geschehen darf.“
- anderen etwas erlauben: Lizenz
- „gar keine Lizenz“ = alles verboten



Lizenzmodelle	proprietär	frei	
		freizügig	Copyleft
kommerziell			
nichtkommerziell			

U Software und Urheberrecht

U.2 Lizenzmodelle

- Freie Software darf man
 0. benutzen,
 1. studieren und anpassen,
 2. weitergeben,
 3. weiterentwickeln und veröffentlichen.

4 Grundfreiheiten – <http://www.gnu.org/philosophy/free-sw>



Quelltext erforderlich!



U Software und Urheberrecht

U.2 Lizenzmodelle

- Freie Software darf man
 0. benutzen,
 1. studieren und anpassen,
 2. weitergeben,
 3. weiterentwickeln und veröffentlichen.

4 Grundfreiheiten – <http://www.gnu.org/philosophy/free-sw>

- Open Source: i. w. dasselbe in 10 Kriterien
Motivation: technisch statt philosophisch

Definition: <http://www.opensource.org/docs/osd>



← Quelltext erforderlich!



U Software und Urheberrecht

U.2 Lizenzmodelle

- Freie Software
 - Copyleft:
Weitergabe nur unter **gleichen Bedingungen** erlaubt
—→ Umwandlung in proprietäre Software nicht erlaubt
 - freizügig:
Weitergabe auch unter anderen Bedingungen erlaubt
—→ **Umwandlung** in Copyleft- oder proprietäre Software erlaubt
 - Teil-Copyleft:
Linken mit proprietärer Software erlaubt

U Software und Urheberrecht

U.2 Lizenzmodelle

- Proprietäre Software / Closed Source
Gegenteil von freier Software / Open Source
 - Benutzen, Weitergeben und/oder Veröffentlichen erfordert individuelle Erlaubnis des Rechteinhabers
 - Studieren, Anpassen und/oder Weiterentwickeln nur dem Rechteinhaber erlaubt **und/oder möglich (Quelltext erforderlich!)**

U Software und Urheberrecht

U.3 Beispiel-Lizenzen

Generell: Gewährleistungsausschluß

Freie Software / Open Source

- strenges Copyleft: GNU GPL, GNU FDL, CC BY-SA
- Teil-Copyleft: GNU LGPL, Mozilla-Lizenz, Microsoft Public License
- freizügig: Modifizierte BSD-Lizenz, Apache-Lizenz, CC BY, CC0, Public Domain

<http://www.gnu.org/licenses/license-list>

U Software und Urheberrecht

U.3 Beispiel-Lizenzen

Generell: Gewährleistungsausschluß

Proprietäre Software

- Lizenz i. d. R. für jedes Programm anders

U Software und Urheberrecht

U.3 Beispiel-Lizenzen

Generell: Gewährleistungsausschluß

Proprietäre Software

- Lizenz i. d. R. für jedes Programm anders
- „Normales“ Beispiel (Dezember 2016):
Adobe Personal Computer Software License Agreement
<http://get.adobe.com/reader/otherversions>

U Software und Urheberrecht

U.3 Beispiel-Lizenzen

Generell: Gewährleistungsausschluß

Proprietäre Software

- Lizenz i. d. R. für jedes Programm anders
- „Normales“ Beispiel (Dezember 2016):
Adobe Personal Computer Software License Agreement
<http://get.adobe.com/reader/otherversions>

Verwenden nur auf PCs erlaubt

Weitergeben nicht erlaubt

Screenshots nicht erlaubt

U Software und Urheberrecht

U.3 Beispiel-Lizenzen

Generell: Gewährleistungsausschluß

Proprietäre Software

- Lizenz i. d. R. für jedes Programm anders
- „Normales“ Beispiel (Dezember 2016):
Adobe Personal Computer Software License Agreement
<http://get.adobe.com/reader/otherversions>

Verwenden nur auf PCs erlaubt

Weitergeben nicht erlaubt

Screenshots nicht erlaubt

Die Software darf jederzeit mit Adobe und Dritten kommunizieren,
zusätzliche Software installieren und Befehle von außen ausführen.

U Software und Urheberrecht

U.4 Fazit

- Generell: Vor Benutzung Lizenz lesen, durch Anwalt prüfen lassen
- „gar keine Lizenz“ = alles verboten

Bearbeitung, Weitergabe und Mitverwendung

- nicht erlaubt oder nicht möglich → proprietäre Software / Closed Source
- erlaubt → freie Software / Open Source
 - bei Mitverwendung Lizenz übernehmen → Copyleft
 - Lizenz umwandelbar → freizügig
- davon unabhängig: kommerziell / nichtkommerziell

Lizenz für Material zu dieser Lehrveranstaltung

- Vortragsfolien, Skript usw.: Copyleft
- Beispiel-Programme: freizügig

Angewandte Informatik

Hardwarenahe Programmierung

SPECIAL

U Software und Urheberrecht

U.1 Überblick

U.2 Lizenzmodelle

U.3 Beispiel-Lizenzen

U.4 Fazit

X Exploits

X.1 Einfache Angriffe

X.2 Puffer-Überläufe

X.3 Return-Oriented Programming

X.4 Fazit

X Exploits

X.0 Vorab: Dies ist keine Einladung, anderer Leute Systeme anzugreifen!

Ziel dieser Lehrveranstaltung ist es, die Methoden der Angreifer zu **verstehen**, um sich und andere dagegen **schützen** zu können, nicht hingegen, dieselben Methoden gegen Dritte einzusetzen.

https://de.wikipedia.org/wiki/Informationssicherheit#Strafrechtliche_Aspekte:

Jegliches rechtswidrige Verändern, Löschen, Unterdrücken oder Unbrauchbar-Machen fremder Daten erfüllt den Tatbestand nach § 303a StGB (Datenveränderung). In besonders schweren Fällen ist dies auch nach § 303b I Nr. 1 StGB („Computersabotage“) strafbar und wird mit Haftstrafe von bis zu fünf Jahren oder Geldstrafe bestraft. [...]

Das Ausspähen von Daten (§ 202a StGB), also die Erlangung des Zugangs zu fremden Daten, die hiergegen besonders geschützt sind, wird mit Haftstrafe bis zu drei Jahren oder mit Geldstrafe bestraft. [...]

X Exploits

X.1 Einfache Angriffe

- falsche Benutzung von `printf()`:
`printf (buffer);` statt `printf ("%s", buffer);`
→ *Formatstring-Angriff*
→ Auslesen des CPU-Stacks

X Exploits

X.1 Einfache Angriffe

- falsche Benutzung von `printf()`:

```
printf ("Your_name,please:");  
gets (name_buffer);  
printf ("Hello,");  
printf (name_buffer);  
printf ("!\n");
```

```
$ ./server-0
```

```
Your name, please: %016llx %016llx %016llx %016llx  
                  %016llx %016llx %016llx %016llx
```

```
Hello, 00000000004007c7 00007fdb2ced2df0 00000000004007c7  
       00007fdb2d0f3007 20786c6c36313025 6373316870216948  
       00007fdb2cbd0068 0000007265746570!
```

```
Your password, please:
```

geheime Daten



X Exploits

X.1 Einfache Angriffe

- falsche Benutzung von `printf()`:
`printf (buffer);` statt `printf ("%s", buffer);`
→ *Formatstring-Angriff*
→ Auslesen des CPU-Stacks
- fehlende Prüfung auf Sonderzeichen beim Verarbeiten von Daten
(z. B. *SQL Injection* – siehe <http://xkcd.com/327/>)
→ Einschleusen von Befehlen an Programme
- *Puffer-Überlauf (Buffer Overflow)* beim Einlesen von Daten
(z. B. mit `gets()`)
→ Überschreiben der Rücksprungadresse
→ Einschleusen von eigenem Code

X Exploits

X.2 Puffer-Überläufe

Schadcode in Eingabe schreiben

Rücksprungadresse überschreiben, zum Schadcode springen

```
#include <stdio.h>
```

```
int main (void)
```

```
{  
    char buffer[20];  
    printf ("Your_name,please:_");  
    gets (buffer);  
    printf ("Hello,_%s!\n", buffer);  
    return 0;  
}
```

CPU-Stack

vor Aufruf von `gets()`

...
Rücksprung zum Betriebssystem
buffer[]

X Exploits

X.2 Puffer-Überläufe

Schadcode in Eingabe schreiben

Rücksprungadresse überschreiben, zum Schadcode springen

```
#include <stdio.h>
```

```
int main (void)
```

```
{  
    char buffer[20];  
    printf ("Your_name, please: ");  
    gets (buffer);  
    printf ("Hello, %s!\n", buffer);  
    return 0;  
}
```

CPU-Stack

nach Aufruf von `gets()`

...
Schadcode
Rücksprung zum Schadcode
buffer[]

X Exploits

X.3 Return-Oriented Programming

Schadcode aus Fragmenten des Programms (*Gadgets*) zusammensetzen
Rücksprungadresse überschreiben, zu den Gadgets springen

```
#include <stdio.h>
```

```
int main (void)
{
    char buffer[20];
    printf ("Your_name, please: ");
    gets (buffer);
    printf ("Hello, %s!\n", buffer);
    return 0;
}
```

CPU-Stack
nach Aufruf von `gets()`

...
Rücksprung zu Gadget 4
Rücksprung zu Gadget 3
Rücksprung zu Gadget 2
Rücksprung zu Gadget 1
buffer[]

X Exploits

X.4 Fazit

Sauber programmieren!

- **Niemals Eingabedaten ungeprüft verwenden!**
(z. B. als ersten Parameter von `printf()`
oder als Bestandteil von SQL-Befehlen)
- **Niemals Pufferüberläufe ermöglichen!**
(z. B. durch Verwendung von `gets()`)

X Exploits

X.4 Fazit

Sauber programmieren!

- **Niemals Eingabedaten ungeprüft verwenden!**
(z. B. als ersten Parameter von `printf()`
oder als Bestandteil von SQL-Befehlen)
- **Niemals Pufferüberläufe ermöglichen!**
(z. B. durch Verwendung von `gets()`)
- **Niemals die Systeme anderer Menschen angreifen!**
Computer-Angriffe sind Straftaten.

Angewandte Informatik

Hardwarenahe Programmierung

SPECIAL

U Software und Urheberrecht

U.1 Überblick

U.2 Lizenzmodelle

U.3 Beispiel-Lizenzen

U.4 Fazit

X Exploits

X.1 Einfache Angriffe

X.2 Puffer-Überläufe

X.3 Return-Oriented Programming

X.4 Fazit

Angewandte Informatik

Hardwarenahe Programmierung

SPECIAL

U Software und Urheberrecht

U.1 Überblick

U.2 Lizenzmodelle

U.3 Beispiel-Lizenzen

U.4 Fazit

X Exploits

X.1 Einfache Angriffe

X.2 Puffer-Überläufe

X.3 Return-Oriented Programming

X.4 Fazit