Softwaresicherheit

Was Sicherheit mit Schlamperei zu tun hat

Alexander Noack
jali@orca-central.de

Chaos Computer Club Bremen e.V.

5. September 2011

Übersicht

Kurze Vorstellung

- Alexander Noack
- Mitglied des Chaos Computer Club
- Seit 15 Jahren Entwickler für Datenbanksysteme

Worum es heute Abend geht

Worum es heute Abend geht

- Kommunikationsinfrastrukturen werden immer wichtiger

Worum es heute Abend geht

- Kommunikationsinfrastrukturen werden immer wichtiger
- Immer mehr Bereiche des öffentlichen Lebens hängen von diesen Systemen ab

Worum es heute Abend geht

- Kommunikationsinfrastrukturen werden immer wichtiger
- Immer mehr Bereiche des öffentlichen Lebens hängen von diesen Systemen ab
- In fast jedem Gerät, vom Kühlschrank bis zum Auto, steckt Software

Worum es heute Abend geht

- Kommunikationsinfrastrukturen werden immer wichtiger
- Immer mehr Bereiche des öffentlichen Lebens hängen von diesen Systemen ab
- In fast jedem Gerät, vom Kühlschrank bis zum Auto, steckt Software
- Wie gut die Funktioniert, darum soll es gehen

Zwei Beispielsysteme, die im Alltag benutzt werden

- Handynetz (GSM)
- World Wide Web

Warum sollte mich das Interessieren?

- Fast jeder hat ein Handy

Warum sollte mich das Interessieren?

- Fast jeder hat ein Handy
- Die meisten nutzen Internet (in Bremen haben 80% der Haushalte Breitbandinternetanschluss)

Warum sollte mich das Interessieren?

- Fast jeder hat ein Handy
- Die meisten nutzen Internet (in Bremen haben 80% der Haushalte Breitbandinternetanschluss)
- Von Dienstleistungen im Netz profitiert jeder

Software in Kritischen Infrastrukturen

- GSM
- World Wide Web

GSM

1. GSM - Was ist das?
2. Sicherheit von GSM
3. Anwendungen von für GSM
4. Implementierung
5. Gegenmaßnahmen

GSM - Was ist das?

- Steht für Global System for Mobile Communications

GSM - Was ist das?

- Steht für Global System for Mobile Communications
- Geplant seit 1983

GSM - Was ist das?

- Steht für Global System for Mobile Communications
- Geplant seit 1983
- Erster Standard 1991

GSM - Was ist das?

- Steht für Global System for Mobile Communications
- Geplant seit 1983
- Erster Standard 1991
- Basis für Handynetz

GSM - Was ist das?

- Steht für Global System for Mobile Communications
- Geplant seit 1983
- Erster Standard 1991
- Basis für Handynetz
- Neuere Dienste: UMTS (3G) LTE (4G)

GSM - Was ist das?

- Steht für Global System for Mobile Communications
- Geplant seit 1983
- Erster Standard 1991
- Basis für Handynetz
- Neuere Dienste: UMTS (3G) LTE (4G)
- Heute allgegenwärtig

GSM - Was ist das?

Aber. . .

GSM wurde ursprünglich nur zum telefonieren konzipiert!

Sicherheit von GSM

Das Alter und die ursprüngliche Ausrichtung von GSM sind heute Ursache für Probleme

Sicherheit von GSM

Welche Sicherheitsprobleme gibt es?

Sicherheit von GSM

Welche Sicherheitsprobleme gibt es?

- Authentifizierung nur einseitig

Sicherheit von GSM

Welche Sicherheitsprobleme gibt es?

- Authentifizierung nur einseitig
- Der Netzprovider kann verifizieren, wer sich in sein Netz einwählt

Sicherheit von GSM

Welche Sicherheitsprobleme gibt es?

- Authentifizierung nur einseitig
- Der Netzprovider kann verifizieren, wer sich in sein Netz einwählt
- Der Telefonkunde kann das nicht

Sicherheit von GSM

Welche Sicherheitsprobleme gibt es?

- Nur die Luftschnittstelle ist verschlüsselt

Sicherheit von GSM

Welche Sicherheitsprobleme gibt es?

- Nur die Luftschnittstelle ist verschlüsselt
- Drei Algorithmen zur Verschlüsselung A5/1..A5/3

Sicherheit von GSM

Welche Sicherheitsprobleme gibt es?

- Nur die Luftschnittstelle ist verschlüsselt
- Drei Algorithmen zur Verschlüsselung A5/1..A5/3
- A5/1 gilt inzwischen praktisch als gebrochen

Sicherheit von GSM

Welche Sicherheitsprobleme gibt es?

- Nur die Luftschnittstelle ist verschlüsselt
- Drei Algorithmen zur Verschlüsselung A5/1..A5/3
- A5/1 gilt inzwischen praktisch als gebrochen
- A5/3 wird in der Praxis nicht genutzt

Sicherheit von GSM

Überraschung!

Sicherheit von GSM

Überraschung!

Es ist gar nicht nötig, die Verschlüsselung zu brechen.

Sicherheit von GSM

Überraschung!

Es ist gar nicht nötig, die Verschlüsselung zu brechen. Kann eine Basisstation nicht verschlüsseln, bleibt die Verschlüsselung aus.

Sicherheit von GSM

Überraschung!

Es ist gar nicht nötig, die Verschlüsselung zu brechen. Kann eine Basisstation nicht verschlüsseln, bleibt die Verschlüsselung aus. Die wenigsten Handys informieren den Benutzer über fehlende Verschlüsselung!

Sicherheit von GSM

Überraschung!

Es ist gar nicht nötig, die Verschlüsselung zu brechen. Kann eine Basisstation nicht verschlüsseln, bleibt die Verschlüsselung aus. Die wenigsten Handys informieren den Benutzer über fehlende Verschlüsselung!
Basisstationen kann man bei eBay für ein paar 100€ bekommen.

Anwendungen für GSM

Wofür wird GSM eingesetzt?

Anwendungen für GSM

Wofür wird GSM eingesetzt?

- Telefonie

Anwendungen für GSM

Wofür wird GSM eingesetzt?

- Telefonie
- Internetzugang

Anwendungen für GSM

Wofür wird GSM eingesetzt?

- Telefonie
- Internetzugang
- Mautsystem (Toll-Collect)

Anwendungen für GSM

Wofür wird GSM eingesetzt?

- Telefonie
- Internetzugang
- Mautsystem (Toll-Collect)
- Steuerung von Windkraftanlagen

Anwendungen für GSM

Wofür wird GSM eingesetzt?

- Telefonie
- Internetzugang
- Mautsystem (Toll-Collect)
- Steuerung von Windkraftanlagen
- *Smart-Grids* Intelligente Strom-/Gaszähler

Anwendungen für GSM

Wofür wird GSM eingesetzt?

- Telefonie
- Internetzugang
- Mautsystem (Toll-Collect)
- Steuerung von Windkraftanlagen
- *Smart-Grids* Intelligente Strom-/Gaszähler
- Online-Banking (mTAN-Verfahren)!

Implementierung von GSM

- Basisstationen und Geräte laufen inzwischen recht stabil

Implementierung von GSM

- Basisstationen und Geräte laufen inzwischen recht stabil
- Telefonhardware ist im wesentlichen genormt

Implementierung von GSM

- Basisstationen und Geräte laufen inzwischen recht stabil
- Telefonhardware ist im wesentlichen genormt
- Wenige Hersteller

Implementierung von GSM

- Basisstationen und Geräte laufen inzwischen recht stabil
- Telefonhardware ist im wesentlichen genormt
- Wenige Hersteller
- Implementierung nur innerhalb des Standards sicher

Implementierung von GSM

- Basisstationen und Geräte laufen inzwischen recht stabil
- Telefonhardware ist im wesentlichen genormt
- Wenige Hersteller
- Implementierung nur innerhalb des Standards sicher
- Diverse Basisstationen stürzen ab, wenn sie falsche Daten erhalten

Gegenmaßnahmen

Was kann man gegen Angriffe tun?

- UMTS benutzen, wann immer möglich

Gegenmaßnahmen

Was kann man gegen Angriffe tun?

- UMTS benutzen, wann immer möglich
- Kein Online Banking über mTAN machen

Gegenmaßnahmen

Was kann man gegen Angriffe tun?

- UMTS benutzen, wann immer möglich
- Kein Online Banking über mTAN machen
- Druck auf den Provider ausüben, dass mehr Sicherheit gewünscht wird.

World Wide Web

1. Das World Wide Web als Dienst
2. SSL - Das Secure Socket Layer

Das World Wide Web als Dienst

- Das WWW ist der am meisten genutzte Dienst im Internet

Das World Wide Web als Dienst

- Das WWW ist der am meisten genutzte Dienst im Internet
- Web 2.0 ermöglicht Nutzern viel Teilhabe

Das World Wide Web als Dienst

- Das WWW ist der am meisten genutzte Dienst im Internet
- Web 2.0 ermöglicht Nutzern viel Teilhabe
- Online-Shops sind bequem und machen Spaß.

SSL - Secure Socket Layer

Was ist SSL?

SSL - Secure Socket Layer

Was ist SSL?

- Erlaubt Ende-zu-Ende-Verschlüsselung und Authentifizierung

SSL - Secure Socket Layer

Was ist SSL?

- Erlaubt Ende-zu-Ende-Verschlüsselung und Authentifizierung
- Benutzen, wo immer es geht (HTTPS - Everywhere-Plugin)

SSL - Secure Socket Layer

Was ist SSL?

- Erlaubt Ende-zu-Ende-Verschlüsselung und Authentifizierung
- Benutzen, wo immer es geht (HTTPS - Everywhere-Plugin)
- Aber:

SSL - Secure Socket Layer

Was ist SSL?

- Erlaubt Ende-zu-Ende-Verschlüsselung und Authentifizierung
- Benutzen, wo immer es geht (HTTPS - Everywhere-Plugin)
- Aber: SSL ist kein Allheilmittel!

SSL - Secure Socket Layer

Wenn die Partner sich nicht kennen, braucht man einen vertrauenswürdigen Dritten, quasi einen *Notar*, im Netz machen das die Certificate Authorities (CA)

SSL - Secure Socket Layer

Wenn die Partner sich nicht kennen, braucht man einen vertrauenswürdigen Dritten, quasi einen *Notar*, im Netz machen das die Certificate Authorities (CA)

Vertrauen?

Was, wenn ich der CA nicht trauen kann?

SSL - Secure Socket Layer

Wenn die Partner sich nicht kennen, braucht man einen vertrauenswürdigen Dritten, quasi einen *Notar*, im Netz machen das die Certificate Authorities (CA)

Vertrauen?

Was, wenn ich der CA nicht trauen kann?

Missbrauch

Im August 2011 wurde die niederländische CA *DigiNotar* gehackt. Der iranische Geheimdienst nutze dies um den Mailverkehr politischer Dissidenten mitlesen.

Fehlerhafte Programmierung

1. Fehler?
2. Kleine Ursachen, Große Wirkung
3. Buffer Overflow
4. SQL-Injection

Fehlerhafte Programmierung

Frage

Wie kann es sein, das solche Fehler in wichtigen Programmen enthalten sind?

Fehlerhafte Programmierung

Frage

Wie kann es sein, das solche Fehler in wichtigen Programmen enthalten sind?

Antwort

- fehlerhafte Programmierung

Fehlerhafte Programmierung

Frage

Wie kann es sein, das solche Fehler in wichtigen Programmen enthalten sind?

Antwort

- fehlerhafte Programmierung
- Fehler im Design

Fehlerhafte Programmierung

Frage

Wie kann es sein, das solche Fehler in wichtigen Programmen enthalten sind?

Antwort

- fehlerhafte Programmierung
- Fehler im Design
- Erweiterung um neue Anwendungsfälle

Kleine Ursachen, große Wirkung

Es gibt verschiedene Fehlerquellen, die Programme angreifbar machen.

- Falsch verwendete Funktionen

Kleine Ursachen, große Wirkung

Es gibt verschiedene Fehlerquellen, die Programme angreifbar machen.

- Falsch verwendete Funktionen
- Buffer-Overflow Fehler

Kleine Ursachen, große Wirkung

Es gibt verschiedene Fehlerquellen, die Programme angreifbar machen.

- Falsch verwendete Funktionen
- Buffer-Overflow Fehler
- Bestimmte Eingaben nicht bedacht (Code Injection)

Kleine Ursachen, große Wirkung

Es gibt verschiedene Fehlerquellen, die Programme angreifbar machen.

- Falsch verwendete Funktionen
- Buffer-Overflow Fehler
- Bestimmte Eingaben nicht bedacht (Code Injection)
- ...

Buffer Overflow

Einer der häufigsten Fehler ist der *Buffer Overflow* (dt.: *Pufferüberlauf*).

Buffer Overflow

Einer der häufigsten Fehler ist der *Buffer Overflow* (dt.: *Pufferüberlauf*).

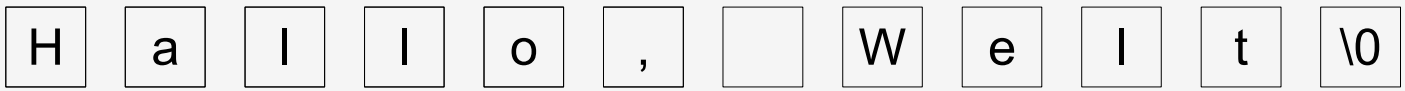
Definition

Ein *Pufferüberlauf* tritt auf, wenn ein Programm versucht mehr Daten in einen zuvor reservierten Speicher zu schreiben, als hineinpassen.

Buffer Overflow

Nehmen wir einen einfachen Text:

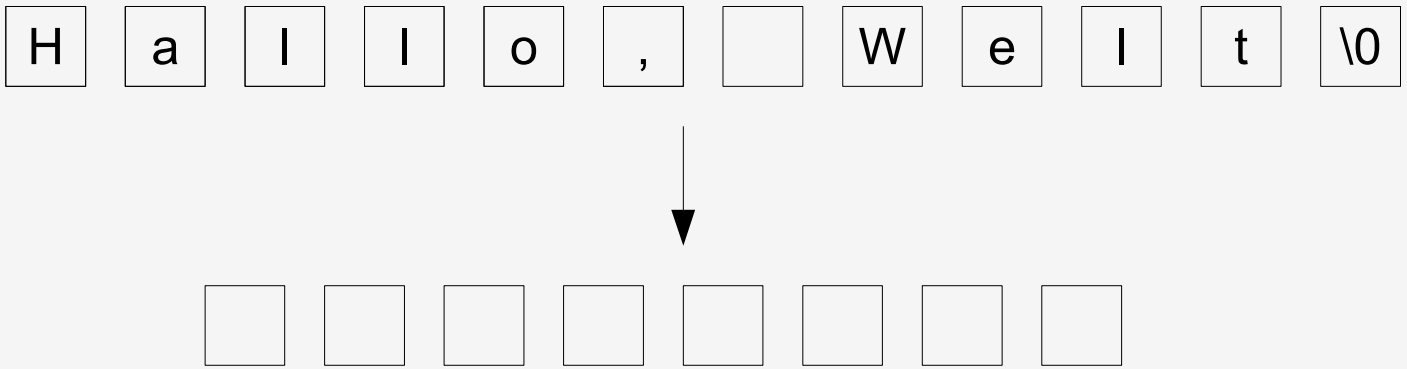
Beispiel



Buffer Overflow

Nun soll der Text in einen anderen Speicher kopiert werden:

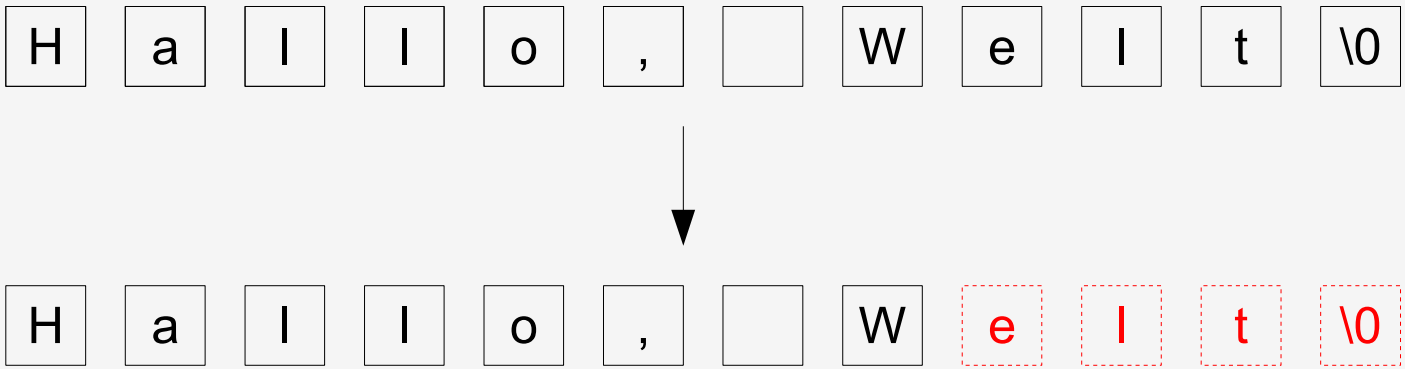
Beispiel



Buffer Overflow

Der Text passt nicht in den Speicher:

Beispiel



Buffer Overflow

Der Computer quittiert das meist mit einer Meldung wie dieser

Absturz des Programs

Segmentation fault

Buffer Overflow

Missbrauch

Wenn man sich geschickt anstellt, kann man statt der Fehlermeldung, den Computer dazu bringen Code auszuführen, den man in das Programm einschleust!

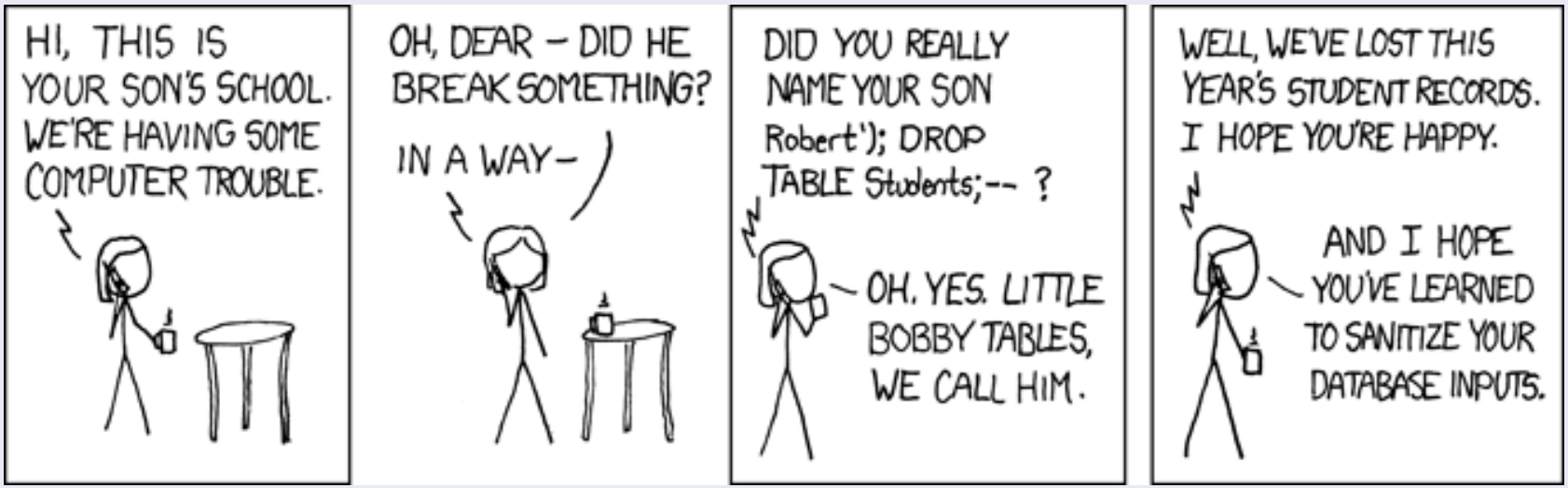
SQL-Injection

Definition

Das Einschleusen von Anweisungen in ein Kommando der Datenbanksprache SQL nennt man *SQL-Injection*

SQL-Injection

Exploits of a Mom



http://xkcd.com/327/

SQL-Injection

Was ist SQL?

Definition

SQL (*Structured Query Language*) ist eine Abfragesprache zum Auslesen und Ändern von Daten in eine Datenbank.

SQL-Injection

Beispiel für eine SQL-Abfrage

```
SELECT * FROM 'users' WHERE 'name' = 'MaxMuster' AND 'password' = 'geheim';
```

SQL-Injection

Nun wird eine Abfrage erstellt, die die Eingabe des Benutzers entgegen nimmt.
Es soll geprüft werden, ob ein Benutzer mit einem bestimmten Namen existiert.

Angreifbare SQL-Abfrage

```
SELECT * FROM 'users' WHERE 'name' = '$USERNAME';
```

$USERNAME wird durch den vom Benutzer eingegebenen Benutzernamen ersetzt.

SQL-Injection

Der böse User *Mallory* gibt jetzt statt seines Benutzernamens folgendes ein:

Angriff mit SQL-Injection

```
' OR '1' = '1
```

SQL-Injection

Der SQL-Parser bekommt nun diesen Code:

Tatsächlich ausgeführter Code

```
SELECT * FROM 'users' WHERE 'name' = '' OR '1' = '1';
```

SQL-Injection

Der SQL-Parser bekommt nun diesen Code:

Tatsächlich ausgeführter Code

```
SELECT * FROM 'users' WHERE 'name' = '' OR '1' = '1';
```

Fehler!

Das ist gültiger SQL Code, der ausgeführt wird!

Entwicklung von Software heute

1. Ursachen von Fehlern
2. Situation in Unternehmen
3. Situation der Open-Source-Community

Ursachen von Fehlern

Wie kommt es zu Programmfehlern?

Ursachen von Fehlern

Wie kommt es zu Programmfehlern?

- Entwickler sind auch nur Menschen

Ursachen von Fehlern

Wie kommt es zu Programmfehlern?

- Entwickler sind auch nur Menschen
- Schon Kleinigkeiten führen zu Fehlern und Sicherheitslücken

Ursachen von Fehlern

Wie kommt es zu Programmfehlern?

- Entwickler sind auch nur Menschen
- Schon Kleinigkeiten führen zu Fehlern und Sicherheitslücken
- Viele Programme werden nachträglich um Funktionen erweitert, für die sie nie konzipiert waren.

Ursachen von Fehlern

Wie kommt es zu Programmfehlern?

- Entwickler sind auch nur Menschen
- Schon Kleinigkeiten führen zu Fehlern und Sicherheitslücken
- Viele Programme werden nachträglich um Funktionen erweitert, für die sie nie konzipiert waren.
- Softwareentwicklung ist schwer

Situation in Unternehmen

Welche Faktoren beeinflussen Unternehmen?

Situation in Unternehmen

Welche Faktoren beeinflussen Unternehmen?

- Hoher Kostendruck

Situation in Unternehmen

Welche Faktoren beeinflussen Unternehmen?

- Hoher Kostendruck
- Wenige Entwickler

Situation in Unternehmen

Welche Faktoren beeinflussen Unternehmen?

- Hoher Kostendruck
- Wenige Entwickler
- „Sieht ja keiner“, da Closed-Source

Situation in Unternehmen

Welche Faktoren beeinflussen Unternehmen?

- Hoher Kostendruck
- Wenige Entwickler
- „Sieht ja keiner“, da Closed-Source
- Viele Unternehmen verzichten aus Kostengründen auf Qualitätssicherung (*Bananenprodukt*)

Situation in Unternehmen

Welche Faktoren beeinflussen Unternehmen?

- Hoher Kostendruck
- Wenige Entwickler
- „Sieht ja keiner“, da Closed-Source
- Viele Unternehmen verzichten aus Kostengründen auf Qualitätssicherung (*Bananenprodukt*)
- Betandskunden vs. Neukunden: Es ist oft lukrativer neue Funktionen zu implementieren, als alte Fehler zu beheben.

Situation in Unternehmen

Frage

Wann soll das Programm an den Kunden gehen?

Situation in Unternehmen

Frage

Wann soll das Programm an den Kunden gehen?

Antwort

Gestern!

Situation in der Open-Source-Community

Welche Faktoren beeinflussen die Open-Source-Community?

Situation in der Open-Source-Community

Welche Faktoren beeinflussen die Open-Source-Community?

- Weil jeder mitmachen kann, sind auch weniger qualifizierte darunter

Situation in der Open-Source-Community

Welche Faktoren beeinflussen die Open-Source-Community?

- Weil jeder mitmachen kann, sind auch weniger qualifizierte darunter
- Interessante Projekte erfahren mehr Aufmerksamkeit als uninteressante

Situation in der Open-Source-Community

Welche Faktoren beeinflussen die Open-Source-Community?

- Weil jeder mitmachen kann, sind auch weniger qualifizierte darunter
- Interessante Projekte erfahren mehr Aufmerksamkeit als uninteressante
- „Wird sich schon jemand kümmern“

Situation in der Open-Source-Community

Welche Faktoren beeinflussen die Open-Source-Community?

- Weil jeder mitmachen kann, sind auch weniger qualifizierte darunter
- Interessante Projekte erfahren mehr Aufmerksamkeit als uninteressante
- „Wird sich schon jemand kümmern“
- Code-Review schwierig, weil wenig dokumentiert wird.

Situation in der Open-Source-Community

Welche Faktoren beeinflussen die Open-Source-Community?

- Weil jeder mitmachen kann, sind auch weniger qualifizierte darunter
- Interessante Projekte erfahren mehr Aufmerksamkeit als uninteressante
- „Wird sich schon jemand kümmern“
- Code-Review schwierig, weil wenig dokumentiert wird.
- Basisdemokratische Prozesse in der Community führen oft dazu, dass nicht die beste, sondern die bekannteste Lösung umgesetzt wird.

Strategien zur Verbesserung von Software

1. Allgemeine Strategien
2. Keep it Simple
3. Test Driven Development
4. Beweisbarer Code
5. Open Source Strategie

Allgemeine Strategien

- Mehr Qualitätskontrolle in Unternehmen.

Allgemeine Strategien

- Mehr Qualitätskontrolle in Unternehmen. Macht Software teurer!

Allgemeine Strategien

- Mehr Qualitätskontrolle in Unternehmen. Macht Software teurer!
- Moderne Programmiersprachen nutzen
- Sprachen wie Ruby, D oder Python ermöglichen das Erkennen von typischen Fehlern schon beim Entwickeln.

Allgemeine Strategien

- Mehr Qualitätskontrolle in Unternehmen. Macht Software teurer!
- Moderne Programmiersprachen nutzen
- Sprachen wie Ruby, D oder Python ermöglichen das Erkennen von typischen Fehlern schon beim Entwickeln.
- Features moderner Betriebsysteme nutzen (Nutzung des NX-Bit der CPU)

Allgemeine Strategien

- Mehr Qualitätskontrolle in Unternehmen. Macht Software teurer!
- Moderne Programmiersprachen nutzen
- Sprachen wie Ruby, D oder Python ermöglichen das Erkennen von typischen Fehlern schon beim Entwickeln.
- Features moderner Betriebsysteme nutzen (Nutzung des NX-Bit der CPU)
- Dokumentation, Dokumentation, Dokumentation

Keep it Simple

- Einzelne Funktionen des Programms in genau definierte Unterprogramme auslagern

Keep it Simple

- Einzelne Funktionen des Programms in genau definierte Unterprogramme auslagern
- Ein Unterprogramm sollte auf eine Bildschirmseite passen

Keep it Simple

- Einzelne Funktionen des Programms in genau definierte Unterprogramme auslagern
- Ein Unterprogramm sollte auf eine Bildschirmseite passen
- Unterprogramme tun, was ihr Name sagt, sonst nichts

Keep it Simple

- Einzelne Funktionen des Programms in genau definierte Unterprogramme auslagern
- Ein Unterprogramm sollte auf eine Bildschirmseite passen
- Unterprogramme tun, was ihr Name sagt, sonst nichts
- Einfache Schnittstellen zwischen Programmteilen definieren

Test Driven Development

Definition

Test Driven Development nennt man eine Entwicklungsstrategie, bei der zunächst für alle denkbaren Einsatzszenarien einer Funktion Tests entworfen werden, und die Funktion dann anhand dieser Tests entwickelt wird.

Test Driven Development

Wie wird *Test Driven Development* gemacht?

- Festlegen, was die Funktion können soll

Test Driven Development

Wie wird *Test Driven Development* gemacht?

- Festlegen, was die Funktion können soll
- Tests entwickeln, mit denen man prüfen kann, ob die Funktion die geforderten Eigenschaften hat.

Test Driven Development

Wie wird *Test Driven Development* gemacht?

- Festlegen, was die Funktion können soll
- Tests entwickeln, mit denen man prüfen kann, ob die Funktion die geforderten Eigenschaften hat.
- Funktion wird Schritt für Schritt implementiert, und immer wieder gegen die Tests geprüft

Test Driven Development

Wie wird *Test Driven Development* gemacht?

- Festlegen, was die Funktion können soll
- Tests entwickeln, mit denen man prüfen kann, ob die Funktion die geforderten Eigenschaften hat.
- Funktion wird Schritt für Schritt implementiert, und immer wieder gegen die Tests geprüft
- Wenn die Funktion alle Tests besteht, kann sie in das Programm integriert werden.

Test Driven Development

Wie wird *Test Driven Development* gemacht?

- Festlegen, was die Funktion können soll
- Tests entwickeln, mit denen man prüfen kann, ob die Funktion die geforderten Eigenschaften hat.
- Funktion wird Schritt für Schritt implementiert, und immer wieder gegen die Tests geprüft
- Wenn die Funktion alle Tests besteht, kann sie in das Programm integriert werden.
- Die Tests umfassen auch Szenarien, in denen eine Funktion unsinnige Daten bekommt, um zu testen, wie sie auf einen Fehler reagiert.

Beweisbarer Code

- Bei Softwareprojekten, die hohe Sicherheit erfordern, kann man die Richtigkeit von Code mathematisch beweisen.

Beweisbarer Code

- Bei Softwareprojekten, die hohe Sicherheit erfordern, kann man die Richtigkeit von Code mathematisch beweisen.
- Wird z.B. in der Luft- und Raumfahrt gemacht

Beweisbarer Code

- Bei Softwareprojekten, die hohe Sicherheit erfordern, kann man die Richtigkeit von Code mathematisch beweisen.
- Wird z.B. in der Luft- und Raumfahrt gemacht
- Einige Sprachen (z.B. *Ada*), erleichtern die Beweisbarkeit ihrer Ausdrücke, und werden im Sicherheitskritischen Umfeld gerne eingesetzt.

Beweisbarer Code

- Bei Softwareprojekten, die hohe Sicherheit erfordern, kann man die Richtigkeit von Code mathematisch beweisen.
- Wird z.B. in der Luft- und Raumfahrt gemacht
- Einige Sprachen (z.B. *Ada*), erleichtern die Beweisbarkeit ihrer Ausdrücke, und werden im Sicherheitskritischen Umfeld gerne eingesetzt.
- *Coq* ist eine funktionale Programmiersprache und ein Tool zum Beweisen von Ausdrücken. Code kann hier nur ausgeführt werden, wenn er zunächst bewiesen wurde.

Beweisbarer Code

Problem

Das Beweisen von Code ist sehr aufwändig und teuer, lohnt daher in seltenen Fällen.

Beweisbarer Code

Problem

Das Beweisen von Code ist sehr aufwändig und teuer, lohnt daher in seltenen Fällen.
Man kann nicht jeden Code beweisen.

Open-Source Strategien

Wann und warum sollte man Open-Source einsetzen?

Open-Source Strategien

Wann und warum sollte man Open-Source einsetzen?

- Quelloffene Software hat viele Vorteile, die besonders in Sicherheitskritiischen Bereichen zum Tragen kommen.

Open-Source Strategien

Wann und warum sollte man Open-Source einsetzen?

- Quelloffene Software hat viele Vorteile, die besonders in Sicherheitskritiischen Bereichen zum Tragen kommen.
- *Security by Obscurity* funktioniert nicht. Es lässt sich *immer* herausfinden, wie ein Sicherheitalgorithmus funktioniert.

Open-Source Strategien

Wann und warum sollte man Open-Source einsetzen?

- Quelloffene Software hat viele Vorteile, die besonders in Sicherheitskritiischen Bereichen zum Tragen kommen.
- *Security by Obscurity* funktioniert nicht. Es lässt sich *immer* herausfinden, wie ein Sicherheitalgorithmus funktioniert.
- Quelloffene Sicherheitslösungen werden von vielen Entwicklern redigiert, Fehler und Hintertüren fallen schnell auf.

Open-Source Strategien

Wann und warum sollte man Open-Source einsetzen?

- Quelloffene Software hat viele Vorteile, die besonders in Sicherheitskritiischen Bereichen zum Tragen kommen.
- *Security by Obscurity* funktioniert nicht. Es lässt sich *immer* herausfinden, wie ein Sicherheitalgorithmus funktioniert.
- Quelloffene Sicherheitslösungen werden von vielen Entwicklern redigiert, Fehler und Hintertüren fallen schnell auf.
- Die Kontrollierbarkeit erhöht das Vertrauen der Anwender.

Open-Source Strategien

Sicherheit durch Transparenz

Kryptographische Software sollte daher immer Quelloffen sein.

Fragen?

Fragen?

Vielen Dank

Vielen Dank für Ihre Aufmerksamkeit!