Softwaresicherheit

Eine Präsentation von

Benedikt Streitwieser und Max Göttl

Einführung Kryptographie und IT-Sicherheit

Gliederung

- Einleitung: Was ist Softwaresicherheit
- Populäre Beispiele
- Anforderungen der Softwaresicherheit
- Gefahren
- Konkrete Beispiele:
  - SQL Injection
  - Buffer Overflow
  - Cross Site Scripting

Einleitung: Was ist Softwaresicherheit?

- Beschäftigt sich mit der Sicherheit von Programmen
- Schutz vor Gefahren bzw. Bedrohungen
- Vermeidung von wirtschaftlichen Schäden und Minimierung von Risiken
- Schwachstellen in jedem noch so gut geplanten und umgesetzten System
- Bedrohung der Grundprinzipien der Informationssicherheit
- Angriffe auf die Schutzziele bedeuten für Unternehmen Angriffe auf reale Unternehmenswerte, im Regelfall das Abgreifen oder Verändern von unternehmensinternen Informationen

Populäre Beispiele

- Stagefrigth
- Shellshock
- Heartbleed
- The Self-Retweeting Tweet
- Explosion of Ariane 5
- Pentium Processor division Error
- Patriot-Missile Error

Anforderungen der Softwaresicherheit

- **Verfügbarkeit** (Availability): Das Programm darf nicht abstürzen und muss immer wieder in einen Zustand zurückkehren in dem neue Eingaben verarbeitet werden können.
- **Vertraulichkeit** (Confidentiality): Geheime Informationen, wie z.B. Passwörter dürfen nicht in öffentlich lesbaren Ausgaben/Speicherbereichen auftauchen
- **Integrität** (Integrity): Öffentliche Eingaben dürfen den Inhalt bestimmter Speicherbereiche sowie das Verhalten des Programms nicht beeinflussen.

Gefahren

- SQL Injection
- Buffer Overflow
- Cross Site Scripting
- DOS Exploits
- Improper Error Handling
- Integer Over- and Underflow
- …

SQL Injection

- Ausnutzen einer Sicherheitslücke in Zusammenhang mit SQL-Datenbanken
- Entsteht durch mangelnde Maskierung oder Überprüfung von Metazeichen in Benutzereingaben
- Angreifer versucht dabei, über die Anwendung, die den Zugriff auf die Datenbank bereitstellt, eigene Datenbankbefehle einzuschleusen.
- Sein Ziel ist es, Daten auszuspähen, in seinem Sinne zu verändern, die Kontrolle über den Server zu erhalten oder einfach größtmöglichen Schaden anzurichten.

SQL Injection

- Veränderung von Daten
- Datenbank-Server verändern
- Ausspähen von Daten
- Einschleusen von beliebigem Code
- Zeitbasiere Angriffe
- Erlangen von Administrationsrechten
- Verwundbarkeit innerhalb des Datenbankservers
- Blind SQL-Injection

SQL Injection

Beispiel:

- Datenbank enthält Informationen zu Studenten
- Abfrage mittels Matrikelnummer: 

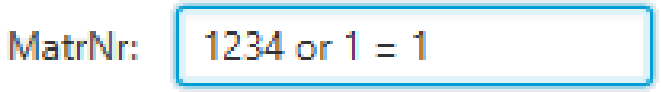
- SQL Statement: SELECT * FROM student WHERE matrNr = 1234;
- Resultat: Details des Studenten mit Matrikelnummer 1234

SQL Injection

Abwehrmaßnahmen:

- Input Validation, z.B. mit regular expression
- Whitelisting: Input, der bestimmte Kriterien erfüllt, wird akzeptiert
- Blacklisting: Input mit bestimmten Inhalten wird blockiert
- Prepared Statements
- DB Zugriffsberechtigungen:
  - Read only
  - Zugriff nur auf benötigte Datenbanken und Tabellen

SQL Injection

- Möglicher Angriff:
- Bösartiger Code wird eingegeben 

- Folgen:
  - Mit Abwehrmaßnahme (Prepared Statement):
    - Details des Studenten mit Matrikelnummer 1234 werden angezeigt
    - Bösartiger Code hat keine Auswirkungen
  - Ohne Abwehrmaßnahmen:
    - Schädliche Eingabe wird ausgeführt
    - Informationen aller Studenten werden angezeigt

Buffer Overflow

- Gehört zu den häufigsten Sicherheitslücken in aktueller Software
- Fehler im Programm
- Zu große Datenmengen in zu kleinem reservierten Speicherbereich
- Folge: Überschreiben von Speicherstellen nach dem Ziel-Speicherbereich
- Schwachstellen, die je nach konkreter Verwundbarkeit als Heap-Overflow, Stack-Overflow, Integer-Overflow oder String-Overflow bezeichnet werden
- Angezeigte Fehler sind beispielsweise „Programm.exe funktioniert nicht mehr“ (Windows) oder „Segmentation fault“ (Linux)

Buffer Overflow

- Absturz des betroffenen Programms
- Verfälschung von Daten
- Injizieren von Code
- Schädigung von Datenstrukturen der Laufzeitumgebung des Programms kann zu ungewollten Zugriffsrechten führen
- Angreifer erlangt Zugang zu System
- Werden in verbreiteter Server- und Clientsoftware wie auch von Internetwürmern ausgenutzt

Buffer Overflow

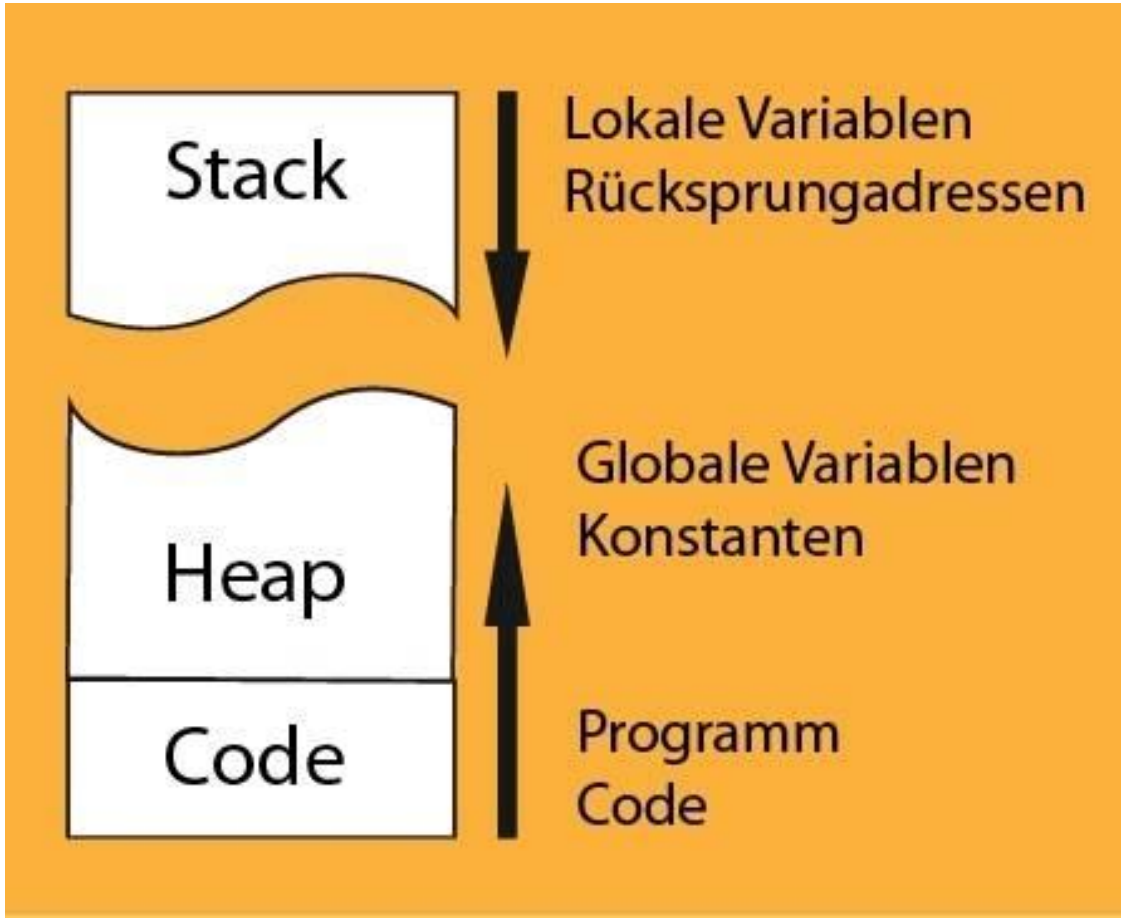
Beim Start eines Programms, weißt ihm das Betriebssystem Speicherbereich zu

- In einem Teil liegt der geschützte und nicht veränderbare Programmcode
- Darüber liegt der **Heap** , in dem das System globale Variablen und Konstanten ablegt
- Dann folgt der **Stack** , der lokale Variablen und den Inhalt von Prozessorregistern aufnehmen kann

Buffer Overflow

- Im Stack liegen auch die Rücksprung-Adressen von Unterprogrammen
- Beim Buffer-Overflow wird eine lokale Variable mit mehr Inhalt gefüllt, als für Sie reserviert ist
- Der Trick der Hacker besteht jetzt darin, die Rücksprungadresse auf Programmsegmente zu lenken, die den eigentlichen Schadcode enthalten.

Buffer Overflow



- Der Stack lässt sich durch lokale Variablen überschreiben
- Dabei kann auch die Rücksprungadresse geändert werden.

Buffer Overflow

Abwehrmaßnahmen:

- Zahl der zu schreibenden Zeichen begrenzen
- Beispielsweise Java statt C oder C++ verwenden
- StackShield: sichert die Return-Adresse und korrigiert sie bei Bedarf
- StackGuard: versucht die Rücksprungadressen zu schützen
- Nicht-ausführbarer Stack
- Prüfen der Abbruchbedingung in Schleifen
- Kontrollierte Rechtevergabe
- moderne Compiler nutzen
- Dem Programmhersteller vertrauen
- Den besten Schutz für den PC gewährleisten auf jeden Fall regelmäßige Updates
- Alternative Software

Cross Site Scripting

- Ausnutzen einer Sicherheitslücke in Webanwendungen
- Informationen werden in einen Kontext eingefügt, in dem sie als vertrauenswürdig eingestuft werden
- Aus diesem vertrauenswürdigen Kontext kann dann ein Angriff gestartet werden
- Ziel ist es meist, an sensible Daten des Benutzers zu gelangen, um beispielsweise seine Benutzerkonten zu übernehmen (Identitätsdiebstahl).

Cross Site Scripting

- HTML Injection
- Übergabe von Parametern an ein serverseitiges Skript, das eine dynamische Webseite erzeugt
- Benutzer-Sessions, Website-Defacements, das Einstellen negativer Inhalte, Phishing-Angriffe und die Übernahme der Kontrolle des Benutzerbrowsers
- Angriffsarten:
  - Persistent
  - Non-persistent

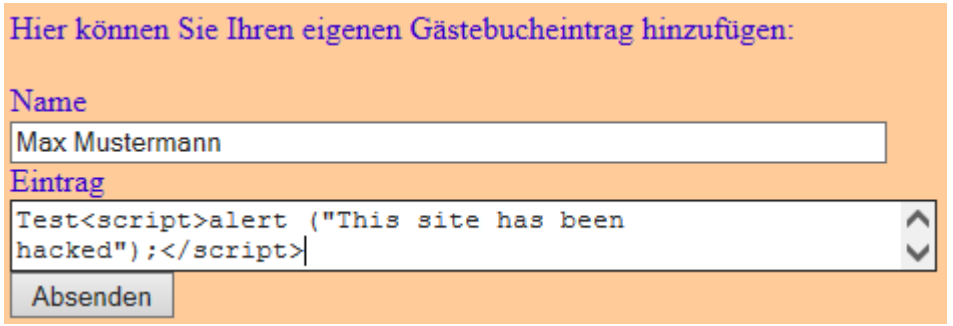
Cross Site Scripting

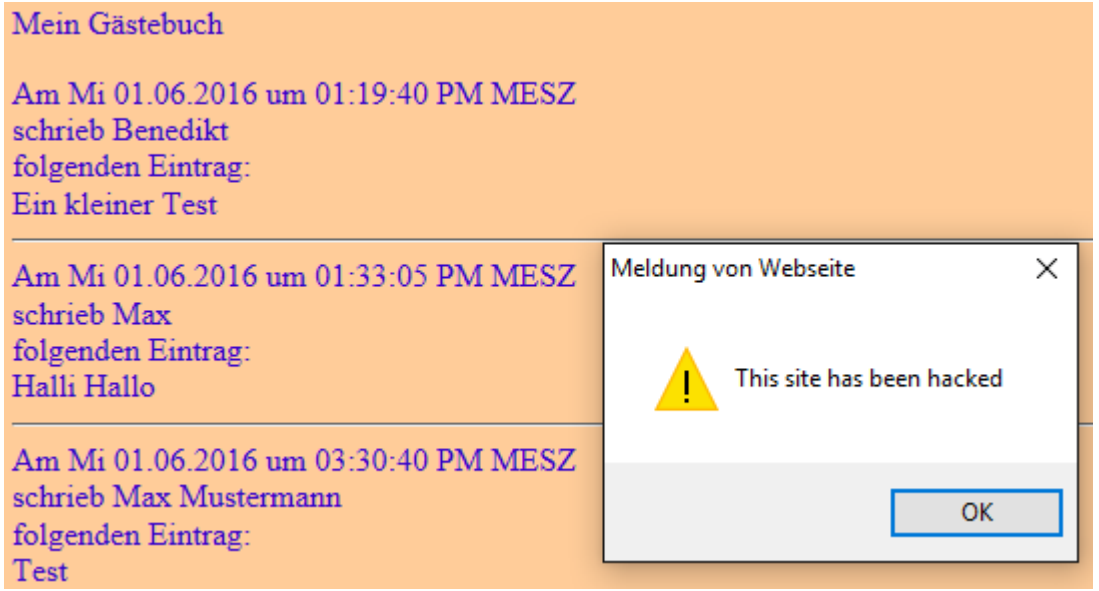
Beispiel:

- In ein Gästebuch wird schadhafter Code eingefügt:

  <script>alert (“This site has been hacked“) ;</script>

- Der Eintrag wird für alle Besucher angezeigt
- Beim Öffnen des Gästebuches erscheint im Browser ein Fenster mit dem Hinweis „This site has been hacked“





Cross Site Scripting

Abwehrmaßnahmen:

- Input Validation, z.B. mit regular expression
- Ersetzung von Metazeichen
- PHP: Funktionen wie strip_tags(), htmlspecialchars(), htmlentities() verwenden
- Content Security Policy: Webadministrator legt vertrauliche Domains von JavaScript fest
- Benutzer:
  - Unterstützung für JavaScript im Browser deaktivieren
  - Browser aktuell halten