Seminar
Sichere Software Systeme

Einführungsveranstaltung

Michael Rodler

Prof. Dr.-Ing. Lucas Davi
Juniorprofessor für Informatik
Universität Duisburg-Essen

UNIVERSITÄT DUISBURG ESSEN
Open-Minded

Sichere Softwaresysteme an der UDE

- Prof. Dr.-Ing. Lucas Davi
  - Juniorprofessur für Informatik

- Wissenschaftliche Mitarbeiter

Dipl.-Ing. Michael Rodler

MSc. Sebastian Surminski

Research Overview

Software Security

- Zero-day exploits
- Control-flow integrity
- Code and data randomization

Stitching Gadgets [**USENIX SEC 2014**]

Trusted Computing

- Remote attestation
- Trusted Platform Module (TPM)

C-FLAT [**ACM CCS 2016**]

Hardware-assisted Security

- Intel Software Guard Extensions (SGX)
- ARM TrustZone
- Rowhammer

HAFIX [**DAC 2015**]

Why Secure Softwaresystems?

Problem of Software Vulnerabilities

- Software increasingly sophisticated and complex
- Developers are not security experts
- Native code

Software bugs and attacks are highly probable

Cybersecurity in the News

Germany admits hackers infiltrated federal ministries, Russian group suspected

German security services have admitted they uncovered a cyberattack on the government in December. Sources say the malware had been planted up to a year earlier and could be the work of a notorious Russian hacking group.

Comments on Cybersecurity

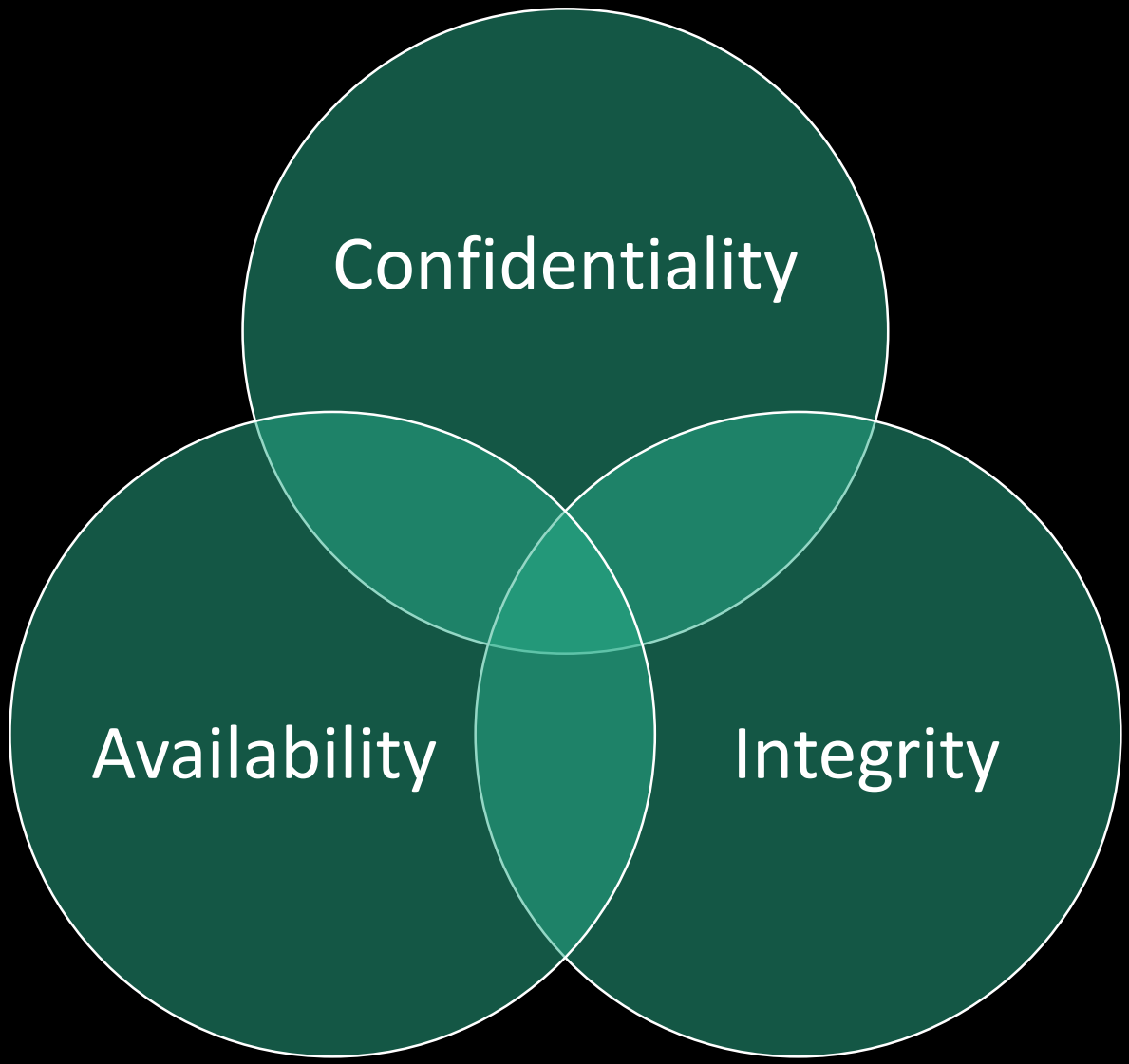
„Solche Cyberangriffe [der Telekom Hack] gehören heute zum Alltag. Wir müssen lernen, damit umzugehen.“

“So we have to get very, very tough on cyber and cyber warfare. It is — it is a huge problem.”

Cyber-Abwehrzentrum

(Quelle: dpa/Berg/Oliver Berg)

C-I-A Principle in Security



C-I-A Principle

- Confidentiality
  - prevention of the unauthorized disclosure of information
  - Covers *data confidentiality* and *privacy*
- Integrity
  - prevention of the unauthorized modification of information
  - covers *data integrity* and *system integrity*
- Availability
  - prevention of the unauthorized withholding of information or resources

Source:
Information Technology Security Evaluation Criteria (ITSEC):
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/ITSicherheitskriterien/itsec-en_pdf.pdf?__blob=publicationFile
and Stallings & Brown. Computer Security, p. 33

Additional Concepts and Security Goals

- Authenticity
  - The property of being genuine and being able to be verified and trusted
- Accountability
  - The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity

Source:
and Stallings & Brown. Computer Security, p. 34

Attacks

- Abuse one or more vulnerabilities
  - Programming Mistakes (Bugs)
  - Design Flaws
- Violates one of the CIA goals
  - A bypass of the applications security policy
- The worst: arbitrary code execution
  - Usually violates all of CIA by giving attacker complete access

Goal of the Seminar

- Introduction into the world of **Software Security**
- Introduction to **one class of vulnerabilities/attacks**
  - By examing 1-2 concrete vulnerabilities/attacks
- Put the attack into scientific context
  - Advanced attacks
  - Countermeasures
    - Currently deployed/available
    - In Research/Academia

1. Understand a Vulnerability/Attack

- Research and select 1-2 vulnerabilities
  - Reproduce an Attack
  - Use whatever resource available
    - Any tool, write-up, exploit PoC is fair game
    - Only requirement you need to **understand what is happening**
- Present the attack(s)
  - Background
  - How does the attack work?
  - What did it take to reproduce the exploit?
  - What resources did you use?
  - (Live) Demo

2. Embed Attack in Scientific Context

- What is the general theme behind the attack?
  - “Vulnerability class”
- How to avoid/fix the vulnerabilities systematically?
- Countermeasures to prevent exploitation?

Where to find Vulnerabilities

- Real-world vulnerabilities
  - CVE database, bugtrackers, security news, bugbounty platforms etc.
- Non-real-world vulnerabilities
  - Capture the Flag, Wargames etc.
  - Damn Vulnerable…
    - Linux
    - Web App
    - Nodejs App
    - …
- You do not have to find new vulnerabilities
  - Search for existing ones and reproduce an attack
  - We'll provide suggestions

Organizational Matters

Ablauf

	Deadline*
Themenauswahl	15. April
Recherchieren und Auswahl von Schwachstelle(n)	24. April
Reproduzieren des Angriffs • Installation der verwundbaren Software • Herstellen der Testumgebung • Angriff durchführen	29. Mai
Zwischenpräsentation • Erklären der Schwachstelle • Erklären des Angriffs	~29. Mai (doodle folgt)
Abgabe eines ersten Entwurfs der Seminararbeit	12. Juni
Abschlusspräsentation • Gegemaßnahmen • Offene Forschungsfragen	~10. Juli (doodle folgt)
Finale Abgabe	24. Juli 23:59

*vorläufig

Themenvergabe

- First come, first served
  - Start: Mi, 11. April 2018 08:00 CEST
  - Ende: So, 15. April 2018 23:59 CEST
- Ablauf
  - E-Mail an michael.rodler@uni-due
  - Betreff: „Themenauswahl Seminar Sommer 2018“
  - **3** Wunschthemen in der E-Mail nennen
  - Ausgewählte Themen nach Präferenz auflisten
- Zuteilung
  - E-Mails werden nach Eingangszeit abgearbeitet und entsprechend der angegebenen Themenpräferenz werden die Themen zugeteilt

Eigene Themen

- Gleicher Ablauf wie bei vorgegebenen Themen
- Zusätzlich in der E-Mail
  - Kurzer „Abstract“ der die Art Schwachstelle beschreibt
  - Motivation
    - Warum ist das Thema relevant für Softwaresicherheit?
    - Warum Sie sich damit beschäftigen wollen?
  - Links zu relevanten Publikationen, „Related Work“
    - Wissenschaftlich und aus der Praxis
    - „Write-Ups“ von echten Schwachstellen
- Backup: Priorisierung von 2 vorgegebenen Themen

Benotung

- Schriftliche Ausarbeitung: 60%
- Mündliche Präsentationen: 40%
  - Zwischenpräsentation 10%
  - Endpräsentation 20%
  - Inhalt und Form der Präsentationsgestaltung: 10%

Erwartungen

- Zuverlässigkeit und selbstmotiviertes Arbeiten
- Professionelles wissenschaftliches Arbeiten
- Thematische Abstimmung mit dem Betreuer

- Offene Problemstellungen im gewählten Thema erkennen und beschreiben
  - Generalisieren von einer konkreten Schwachstelle auf eine Klasse von Schwachstellen und Gegenmaßnahmen

Sprache

- Seminararbeit und Präsentation
  - **Englisch**
  - (auch Deutsch möglich)
- Nutzen Sie die Gelegenheit die Seminararbeit in Englisch zu verfassen
  - Vorbereitung für Bachelor und Masterarbeit
  - Der Großteil der Publikationen/Artikel in der Informatik sind in Englisch
    - Im Besonderen in der Sicherheitsforschung
  - Keine komischen Übersetzungen

Formate

- Seminararbeit in LaTeX
  - Template steht auf der Kursseite zur Verfügung
  - Seitenlimit für die Seminararbeit ist 15 Seiten inkl. Literaturverzeichnis, Appendix etc.
- Freie Wahl des Präsentationsformat
  - z.B. Powerpoint, Flash, OpenOffice, LaTeX Beamer, Keynote, …
- Finale Abgabe in **PDF**
  - Folien der Präsentation am Tag vor der Präsentation um 23:59
  - Seminararbeit am Tag der Deadline 23:59

Topics

Deserialization Vulnerabilities

- (Web) Application Security
- Insecure Formats for Deserialization
  - Often lead to (arbitrary) code execution
  - Unknown “Features” of the format
  - Inherit design flaws
- Examples
  - Java Serialization
  - PHP serialize/unserialize
  - Python pickle
  - Javascript eval of JSON data
  - YAML unsafe load with object construction

Blind SQL Injection

- (Web) Application Security
- SQL injection
  - Modify executed SQL by providing malicious input
  - Extract information from database
- Blind SQL injection
  - Same, but no output given
  - Time-based
  - Boolean Content-based

NoSQL Injection

- Web Application Security
- Similar to SQL injection but for NoSQL databases
  - Different query language, same problem

DOMXSS

- Web Security
- DOM-Based Cross Site Scripting (XSS)
- XSS
  - Manipulate/Inject JavaScript in the victim browser
  - Typically unsanitized input in server response
- DOMXSS
  - Unsanitized input processed by client-side JavaScript code

Hash Length Extension Attacks

- Cryptography
- Application uses flawed custom HMAC
- Abuses a property of Merkle–Damgård hash construction
  - Applicable to MD5, SHA1, SHA2 family
  - Not applicable to SHA3 (Sponge-based desgin)
- Allow attacker to forge „authenticated“ data

Padding Oracle Attacks

- Cryptography
- Attack against encryption in CBC mode with padding
- Allows the attacker to
  - Decrypt and encrypt aribtrary ciphertexts
- Prominent due to attacks against SSL/TLS
  - BEAST (CVE-2011-3389), Lucky Thirteen, POODLE (CVE-2014-3566)

Compression Oracle Attacks

- Cryptography
- Attack against compressed ciphertext data
- Expoits ciphertext length as side-channel
  - Detect content in ciphertext

- Prominent due to attacks against SSL/TLS
  - CRIME, BREACH

Packed Executables

- Binary Reverse Engineering
- A common binary obfuscation technique
  - Make code unreadable for static inspection
  - Hide code as encrypted/compressed data
- Actually executed code is „unpacked“ during runtime
  - Example: UPX packer
  - Many more
- Usual Approach: Dynamic Analysis

Return-Oriented Programming

- Binary Exploitation
- Return-Oriented Programming (ROP)
  - Code-reuse attack technique
  - Current state-of-the-art attack technique
  - Arbitrary computation
- Ususally launched via stack-based buffer overflow
  - Attacker prepares program stack

Heap Feng Shui

- Binary Exploitation
- “Heap Feng Shui” is used for moving the program heap into a certain state beneficial for exploitation
- Exploit technique against modern C/C++ applications
  - Very common in attacks against Web Browser
- Combined with Heap-based vulnerabilities
  - Buffer Overflow
  - Use-After-Free

Format String Attacks

- Binary Exploitation
- Manipulate format string for printf function family
- Allows to
  - Read arbitrary memory contents (information leak)
  - Possibly write to (arbitrary) memory

SigReturn-Oriented Programming

- Binary exploitation
- An extension to return oriented programming
  - More portable exploits

FLUSH+RELOAD Cache Attack

- CPU micro-architecture exploitation
- FLUSH+RELOAD is a Side Channel that abuses caching in modern CPU micro-architectures
- Allows spying on other programs on the same CPU
  - Exfiltrate cryptographic keys
  - Exfiltrate keypresses
- Attack other users

Questions?

Links

- Kurshomepage
  - https://www.syssec.wiwi.uni-due.de/studium-lehre/sommersemester-18/9952/
- Sammlung von Artikeln und Hinweisen zum Verfassen einer Seminararbeit (vom Lehrstuhl Systemsicherheit an der RUB, Prof. Thorsten Holz)
  - https://www.syssec.rub.de/teaching/How-To/
- LaTeX Template
  - https://github.com/uni-due-syssec/thesis-template

Links for Finding Vulnerabilities

- Real Vulnerabilities
  - https://github.com/qazbnm456/awesome-cve-poc
  - https://cve.mitre.org/
- CTF Wargames
  - https://github.com/zardus/wargame-nexus
  - https://liveoverflow.com/capture_the_flag/index.html
  - https://github.com/carpedm20/awesome-hacking
  - https://www.root-me.org/
  - http://www.wechall.net/

Finding Scientific Literature

- Check Program of Top Security Conferences
  - ACM CCS
  - IEEE Security & Privacy
  - USENIX Security
  - NDSS
- https://scholar.google.de/
- http://dblp.uni-trier.de/