Systems Hardening
Premeeting - WS 2021/22 - Season III

Marius Momeu\textsuperscript{1}  Sergej Proskurin\textsuperscript{1,2}

\textsuperscript{1}Chair of IT Security, Department of Informatics, Technical University of Munich (TUM)

\textsuperscript{2}BedRock Systems

July 13, 2021
Intro

Your tutors

- Marius Momeu¹ (momeu@sec.in.tum.de)

¹I'm posting theses / guided research topics at: https://www.sec.in.tum.de/i20/people/momeu-marius
Intro

Your tutors

- Marius Momeu¹ (momeu@sec.in.tum.de)
- Sergej Proskurin (proskurin@sec.in.tum.de)

¹I'm posting theses / guided research topics at: https://www.sec.in.tum.de/i20/people/momeu-marius
Objectives

This seminar is structured to train you for publishing research at scientific conferences or journals.

Consequently, you will exercise and expand a broad spectrum of research skills, such as formulating a clear and novel hypothesis, validating it, and, most importantly, presenting and writing about your findings.

For that, your tutors will pick state-of-the-art mechanisms or infamous issues in the area of systems hardening, and define problem statements to address their limitations. These will then get assigned to you, and you will have to provide a design, prototype, and evaluation for your assigned topic.\(^2\).

Finally, you will write a paper based on the results that you obtain, and present your findings at the end of the semester.

\(^2\) you may also propose your own topic
Content

We are generally interested in mechanisms that improve the security of (low-level) software running in applications, systems, or infrastructures.

As such, the following pool captures some high-level systems hardening areas we will pick topics from:

- CPU extensions (*Intel VT-x/MPK/CET/HLAT, ARM PAC/MTE*) for hardening OS applications, kernels, unikernels, µkernels
  - via code/data isolation, control-flow integrity, data integrity
- Fuzzing low-level software (e.g., OS kernels, device drivers, and hypervisors)
- Static program analysis (especially focusing on large stateful software s.a. OS kernels, device drivers, and hypervisors)
- Microarchitectural flaws and side-channels for leaking secrets, revealing stealthy monitors, etc.
- Security analysis and exploitation of hardware extensions (e.g., *AMD-SEV-*, *Intel CET/MKTME*)
- Heap hardening against use-after-free vulnerabilities
- Confidential/Trusted computing via Trusted Execution Environments: *Intel SGX/MKTME/TXT, ARM TrustZone, AMD-SEV-*
- Live patching
Hands-On

Throughout this seminar, you should expect to touch on the following, including but not limited to:

- Operating machines remotely via the command-line terminal (bash on Unix systems)
- Coding in C/C++, Assembly (x86, ARM), (maybe) *Rust*, various scripting languages
- OS concepts, such as memory management (via paging or nested-paging\(^3\)), interrupts, (bare-metal and emulated) device drivers, syscalls/hypercalls
- LLVM's static analysis framework
- Examining various hardware extensions in architecture manuals (*Intel VT-x/MPK/CET/HLAT*, *ARM PAC/MTE, AMD-SEV-*\(^4\))
- Computer architecture concepts (e.g., speculative execution, return stack buffers, caches, TLBs)
- Dissecting the layout and risks of heap allocators
- Exploitation know-how: code-reuse attacks, data-oriented attacks, secret leaking via covert side-channels
- Compiling/building, dynamic or static linking, binary formats (mostly *ELF*)
- System administration (e.g., spawning VMs, managing partitions)

\(^3\)via *PTs* and *EPTs* on Intel's architecture
Tentative Timeline | Deliverables

- **Paper proposal**
  - motivation
  - research goals

- **First paper draft**
  - introduction
  - technical background
  - threat model
  - related work

- **Final paper draft**
  - design
  - implementation
  - evaluation
  - limitations
  - ...

- **Camera-ready version**

- **Draft slides for the final presentation**

- **Peer review**

- **Task for Students**
  - Students send shortlist of preferred topics
  - Tutors publish catalogue of topics
  - Tutors assign topics to students

- **Task for Tutors**
  - Tutors assign topics to students

- **Deliverables**
  - 5.10
  - 12.10
  - 26.10
  - 16.11
  - 23.11
  - 04.01
  - 11.01
  - 25.01
  - 01.02
  - 04.02
Tentative Timeline | Sessions

- **Deliverables**
  - 5.10
  - 12.10
  - 19.10
  - 26.10
  - 16.11
  - 23.11
  - 21.12
  - 04.01
  - 11.01
  - 25.01
  - 01.02
  - 04.02
  - 12.02

- **Sessions**
  - Hints on Scientific Writing
    - 23.11
  - Proposal presentations
    - motivation
    - research goals
    - 23.11
  - System design presentations
    - 21.12
  - First paper draft
    - introduction
    - technical background
    - threat model
    - related work
    - 21.12
  - Final paper draft
    - design
    - implementation
    - evaluation
    - limitations
    - ... 
    - 04.01
  - Peer review
    - Camera-ready version
    - Draft slides for the final presentation
    - 01.02
  - Final Talks

- **Hints on**
  - Students send shortlist of preferred topics
    - 5.10
  - Tutors publish catalogue of topics
    - 12.10
  - Tutors assign topics to students
    - 19.10
  - Paper proposal
    - motivation
    - research goals
    - 26.10
  - System implementation & evaluation presentations
    - 16.11
  - Draft slides for the final presentation
    - 20.10
  - Peer review
    - Camera-ready version
    - 23.11
  - Final Talks
    - 01.02
  - Final Talks
    - 12.02

- **Tasks**
  - Task for Students
  - Task for Tutors
Tentative Timeline

- **Paper proposal**
  - motivation
  - research goals

- **First paper draft**
  - introduction
  - technical background
  - threat model
  - related work

- **Final paper draft**
  - design
  - implementation
  - evaluation
  - limitations
  - ...

- **Camera-ready version**

**Hints on Scientific Writing**

**System design presentations**

**Proposal presentations**
- motivation
- research goals

**Optimization and evaluation**

**Final Talks**

- **Deliverables**
  - 5.10
  - 12.10
  - 19.10
  - 26.10
  - 16.11
  - 23.11
  - 21.12
  - 04.01
  - 11.01
  - 25.01
  - 01.02
  - 04.02
  - 12.02

- **Sessions**

- **Draft slides for the final presentation**

- **Task for Tutors**
  - Tutors publish catalogue of topics
  - Tutors assign topics to students

- **Task for Students**
  - Read related literature & setup prototyping environment
  - Sketch design
  - Prototype implementation and evaluation
  - Peer review
  - Hints on Paper Reviewing & Hints on Public Speaking
  - Final Talks
Grading

**Graded** deliverables:
- Camera-ready presentation
- Final presentation
- Prototype / design / experiments

**Mandatory ungraded** deliverables:
- Paper proposal
- Paper drafts
- Intermediate presentations
- Peer review

**Optional** deliverables:
- Draft for the final presentation

\[
\begin{array}{lcl}
50 \% & \text{Final Paper (Content, Style, Language, Scope, \ldots)} \\
40 \% & \text{Final Talk (Presentation and Q&A)} \\
10 \% & \text{Design / Prototype / Experiments} \\
\hline
\Sigma & 100 \% & \text{Final Grade}
\end{array}
\]
Deliverables Format

**Paper proposal:**
- 2-3 pages
- one-column
- **note:** the focus should be on motivation of the topic assigned to you and your research goals in this seminar

**Paper:**
- IEEE conference proceedings template\(^4\)
- maximum 10 pages, excluding References and Appendix
- two-column

**Presentation:**
- TUM presentation template\(^5\)
- custom templates can be used as well
- 16:9 aspect ratio if held online, 4:3 if held with beamer

**Peer review:**
- format similar to peer reviews in scientific conferences
- one page with summary, strengths, and weaknesses of reviewed paper

Generally, we encourage you to use \text{LaTeX} \ for writing.

\(^4\)https://www.ieee.org/conferences/publishing/templates.html
\(^5\)https://latex.tum.de/templates/608c2650db4bc7007f58c931
Orga

**When?** irregularly, on Tuesdays, at 10:00 h (subject to change)

**Where?** Onsite (depending on the regulations), online (via BBB), or hybrid

**Capacity:** 10 students

**Language:** English

**Course of study:** both Master’s and Bachelor’s students

**Registration:** via the matching system
Seminar Resources

We will setup a Moodle⁶ page for announcements, for submitting deliverables, and for uploading lecture slides.

We will create Gitlab⁷ repositories on LRZ’s git server for versioning the paper’s and prototype’s source code.

Depending on the topic, we can configure accounts for you in our chair’s test network and let you access our hardware for prototyping.

Matrix⁸ for instantaneous communication.

⁶https://www.moodle.tum.de/
⁷https://gitlab.lrz.de/
⁸https://matrix.tum.de/
Task for Matching Prioritization

Please send us a letter of motivation of maximum 2 pages stating up to 3 mechanisms / hardware technologies / software components from slide 3 that you would like to work on during the seminar. In your letter, describe why do you want to work with these and why do you find them important for systems security?

Send it to: momeu@sec.in.tum.de and proskurin@sec.in.tum.de
In your email, use the subject: Matching - Systems Hardening - WS 2021

Deadline: Sunday, 25th of July

Also, please mention in your report if you have attended any of the following courses:

- Rootkit Praktikum, Binary Exploitation
- Software Security Analysis, Trusted Execution Environment, Reverse Engineering
- IT Security, Secure Mobile Systems
- Computer Architecture, Operating Systems
- Any other course where you have tackled the topics / technologies we have mentioned above
Questions?

Marius Momeu
momeu@sec.in.tum.de
@MariusMomeu

Sergej Proskurin
proskurin@sec.in.tum.de