Chair of IT Security
Department of Informatics
Technical University of Munich

Systems Hardening

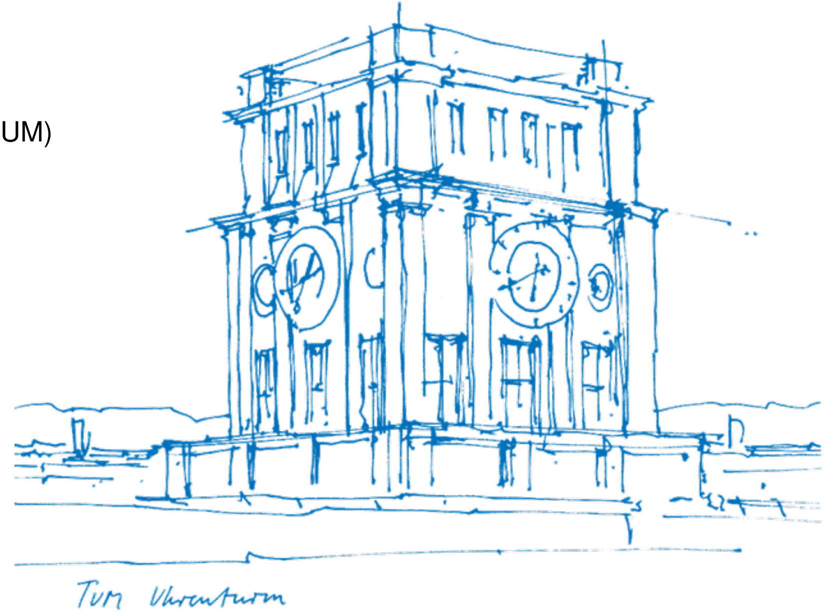
Premeeting - WS 2021/22 - Season III

Marius Momeu[1] Sergej Proskurin[1,2]

[1] Chair of IT Security, Department of Informatics, Technical University of Munich (TUM)

[2] BedRock Systems

September 20, 2021



Intro

Your tutors

- Marius Momeu[1] (momeu@sec.in.tum.de)

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

Intro

Your tutors

- Marius Momeu[1] (momeu@sec.in.tum.de)
- Sergej Proskurin (proskurin@sec.in.tum.de)

[1] I'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

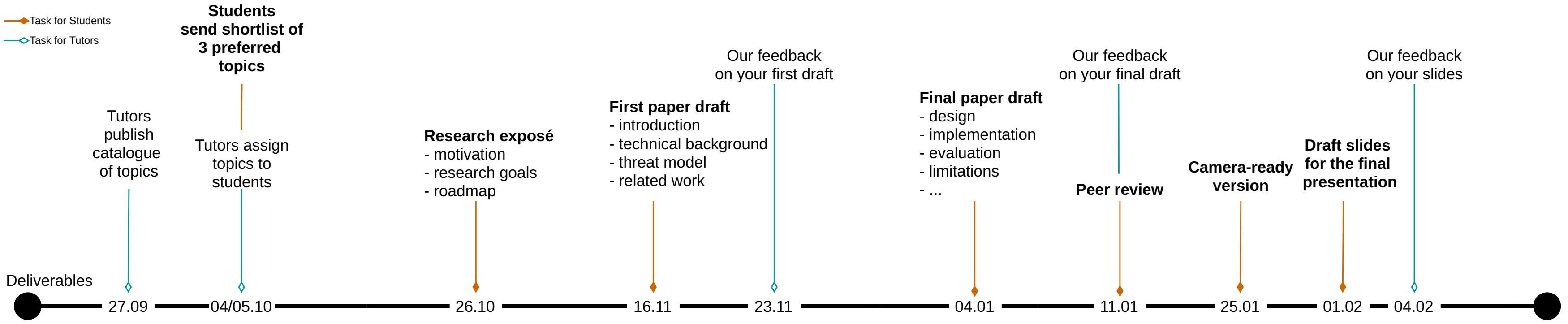
Hands-On

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-**)
- 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



◆ Task for Students

◇ Task for Tutors

Deliverables

27.09 — Tutors publish catalogue of topics

04/05.10 — Tutors assign topics to students

Students send shortlist of 3 preferred topics

26.10 — **Research exposé**
- motivation
- research goals
- roadmap

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

23.11 — Our feedback on your first draft

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

11.01 — **Peer review**

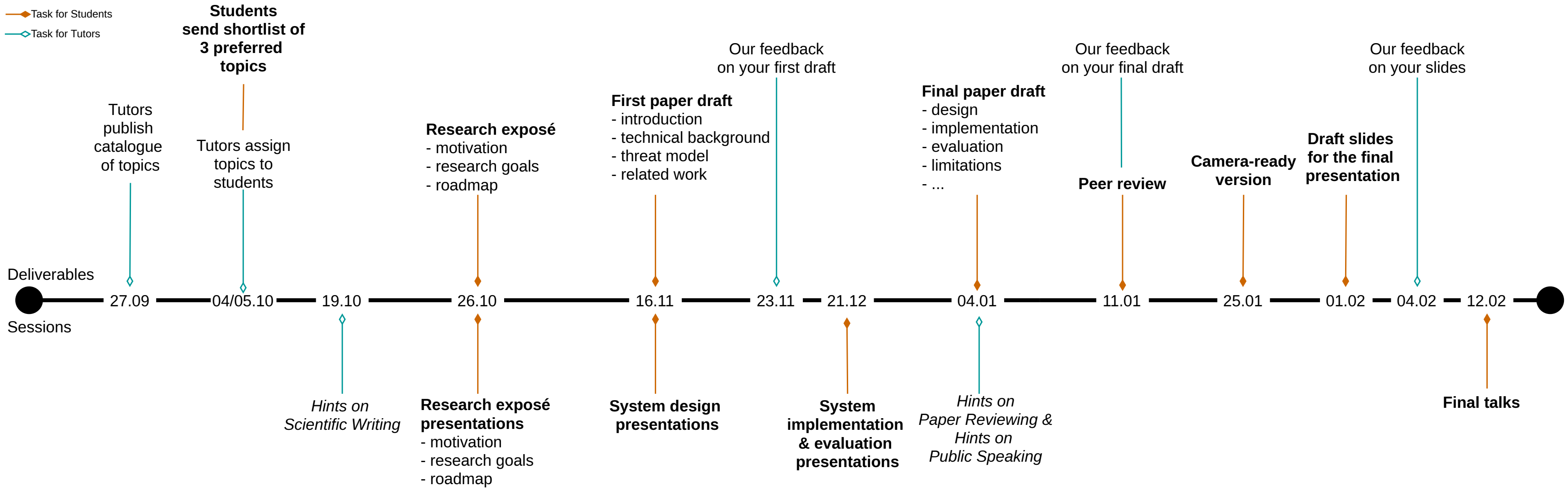
Our feedback on your final draft

25.01 — **Camera-ready version**

01.02 — **Draft slides for the final presentation**

04.02 — Our feedback on your slides

Tentative Timeline | Sessions



◆ Task for Students

◇ Task for Tutors

Deliverables

Sessions

27.09 — Tutors publish catalogue of topics

04/05.10 — Tutors assign topics to students; **Students send shortlist of 3 preferred topics**

19.10 — *Hints on Scientific Writing*

26.10 — **Research exposé**
- motivation
- research goals
- roadmap

26.10 — **Research exposé presentations**
- motivation
- research goals
- roadmap

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

16.11 — **System design presentations**

23.11 — Our feedback on your first draft

21.12 — **System implementation & evaluation presentations**

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

04.01 — *Hints on Paper Reviewing & Hints on Public Speaking*

11.01 — **Peer review**; Our feedback on your final draft

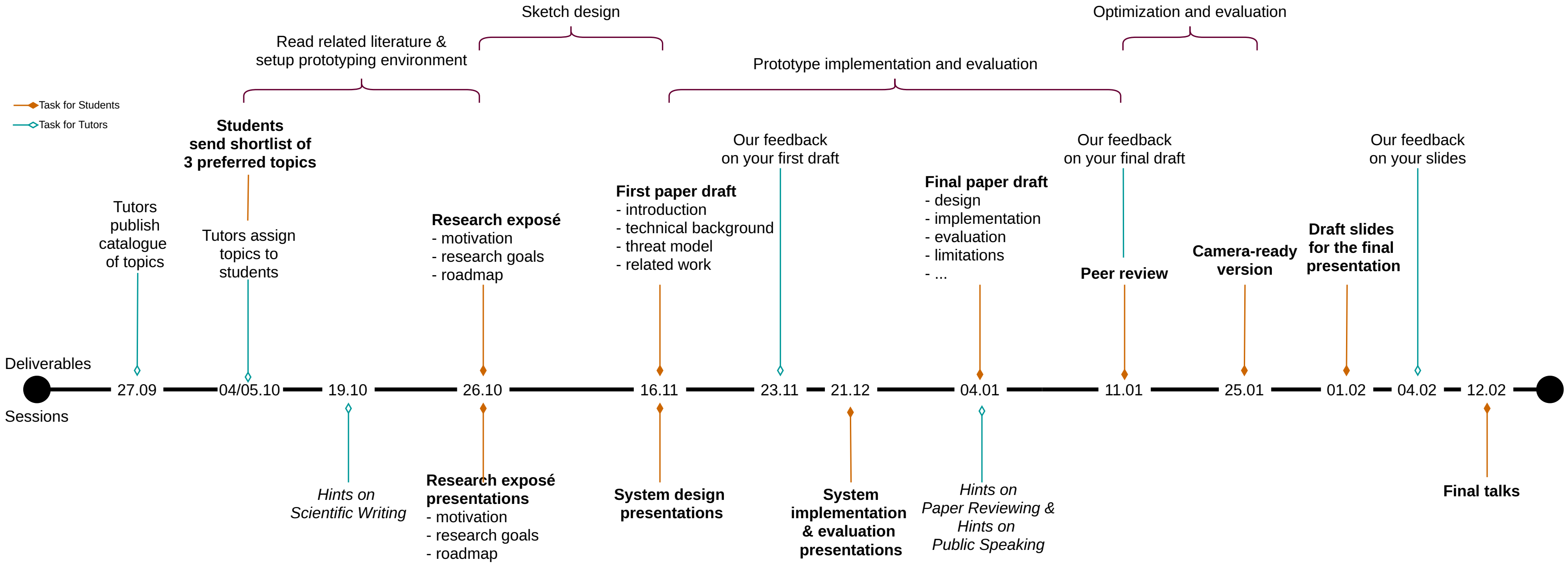
25.01 — **Camera-ready version**

01.02 — **Draft slides for the final presentation**

04.02 — Our feedback on your slides

12.02 — **Final talks**

Tentative Timeline



Sketch design

Optimization and evaluation

Read related literature &
setup prototyping environment

Prototype implementation and evaluation

◆ Task for Students

◇ Task for Tutors

Deliverables

- 27.09 – Tutors publish catalogue of topics
- 04/05.10 – **Students send shortlist of 3 preferred topics**; Tutors assign topics to students
- 26.10 – **Research exposé**
  - motivation
  - research goals
  - roadmap
- 16.11 – **First paper draft**
  - introduction
  - technical background
  - threat model
  - related work
- 23.11 – Our feedback on your first draft
- 04.01 – **Final paper draft**
  - design
  - implementation
  - evaluation
  - limitations
  - ...
- 11.01 – Our feedback on your final draft; **Peer review**
- 25.01 – **Camera-ready version**
- 01.02 – **Draft slides for the final presentation**
- 04.02 – Our feedback on your slides

Sessions

- 19.10 – *Hints on Scientific Writing*
- 26.10 – **Research exposé presentations**
  - motivation
  - research goals
  - roadmap
- 16.11 – **System design presentations**
- 21.12 – **System implementation & evaluation presentations**
- 04.01 – *Hints on Paper Reviewing & Hints on Public Speaking*
- 12.02 – **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

	50 %	Final Paper (Content, Style, Language, Scope, . . .)
	40 %	Final Talk (Presentation and Q&A)
	10 %	Design / Prototype / Experiments
Σ	100 %	Final Grade

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

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

Paper:

- IEEE conference proceedings template[4]
- maximum 10 pages, excluding References and Appendix
- two-column

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 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 (**possible with 3G!**), 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 **Moodle**[6] page for announcements, for submitting deliverables, and for uploading lecture slides.

We will create **Gitlab**[7] **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**.

Matrix[8] for instantaneous communication.

[6] https://www.moodle.tum.de/
[7] https://gitlab.lrz.de/
[8] https://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