

Ausgewählte Kapitel der Systemsoftware

27. Oktober 2022

Rüdiger Kapitza



Lehrstuhl für Verteilte Systeme
und Betriebssysteme



Friedrich-Alexander-Universität
Technische Fakultät

Organisatorisches

Rüdiger Kapitza



(Raum 0.048)

ruediger.kapitza@cs.fau.de

Arne Vogel



(Raum 0.041)

vogel@cs.fau.de

- Termin: TBD
- Webseite: <https://sys.cs.fau.de/lehre/ws22/akss>
- Mailingliste an alle Teilnehmer & Betreuer
akss@lists.cs.fau.de
 - Anmeldung: lists3.cs.fau.de
- Mailingliste an Betreuer
akss-betreuer@lists.cs.fau.de

- Papier: Flicker¹
- Rezension
 - Kurze Zusammenfassung (max. 5 Sätze)
 - Antworten auf die folgenden Fragen, nicht länger als 10 Sätze
 - Erscheint die Struktur des Artikels sinnvoll?
 - Ist der Beitrag verständlich? Bzw. welche Dinge sind schwer zu erfassen oder unzureichend erklärt?
 - Erscheinen die Ergebnisse plausibel?
 - Gibt es ersichtliche offene Punkte bzw. Schwachstellen?
 - Als PDF bis einschließlich 04.11.2022 an vogel@cs.fau.de
 - Dabei enthält die PDF euren Namen!
- Besprechung in Präsenz in der Woche des 07.11.2022

¹Jonathan M. McCune et al. „Flicker: an execution infrastructure for tcb minimization“. In: Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems 2008. Eurosys '08. <https://doi.org/10.1145/1352592.1352625>. New York, NY, USA: Association for Computing Machinery, Apr. 2008, S. 315–328.

- Vortrag²
- Anschließende Diskussion:
 - Inhalt des Vortrags
 - Aufbau der Präsentation
 - Präsentationsstil
- In Präsenz in der Woche des 14.11.2022

²Ines Messadi et al. „SplitBFT: Improving Byzantine Fault Tolerance Safety Using Trusted Compartments“. In: Proceedings of the 23rd ACM/IFIP International Middleware Conference. Middleware '22. <https://doi.org/10.1145/3528535.3531516>. New York, NY, USA: Association for Computing Machinery, Nov. 2022, S. 56–68.

- Eigenständiges Bearbeiten eines Themas
- Literaturrecherche: Vorgegebene Papiere als Ausgangsbasis
- Abgabe eines Exposés: roter Faden
 - Max. eine Seite!
 - Stichpunktartig: Inhalt der Kapitel der Ausarbeitung
 - Liste der Quellen die zusätzlich aus der Literaturrecherche gekommen sind
 - Abgabe bis (spätestens) zum 21.11.2022

- Erstellen einer Ausarbeitung (6 Seiten)
- ACM Standard Proceedings Template (ACM SIG style)
- Sprache: Deutsch oder Englisch



- Keine reine Nacherzählung/Übersetzung
- Direkte Übernahme von Abbildungen vermeiden
- Aufgreifen und Vertiefen einzelner Aspekte
- Herausarbeiten eigener Fragestellung
- Eigene Literaturrecherche

- Foliensatz zur Ausarbeitung
- i4-Beamertemplate (i4neo)
- Sprache: Deutsch oder Englisch
- Vortrag im Rahmen des Seminars
 - min. 20-minütiger (besser 25-minütiger) Vortrag
 - 10-minütige Diskussion zum Vortrag



Semesterplan

- 27.10. Organisation, Einführung und Themenvergabe
- 07.11. Praktische Übung: Lesen & Diskutieren von Papieren
- 14.11. Praktische Übung: Vortragsdiskussion
- 21.11. Abgabe Exposé (kein Präsenztermin)

- *Aus* = Abgabe vollständige Version der Ausarbeitung
- *Fol* = Abgabe vollständige Version der Folien
- *Dry* = Abgabe finale Version der Ausarbeitung + Folien und Probevortrag
- *Prä* = Präsentation + (anonyme) Rezension

12.12	19.12	09.01	16.01	23.01	30.01	06.02
<i>Aus</i> _{1,2}	<i>Fol</i> _{1,2} <i>Aus</i> _{3,4}	<i>Dry</i> _{1,2} <i>Fol</i> _{3,4} <i>Aus</i> _{5,6}	<i>Prä</i> _{1,2} <i>Dry</i> _{3,4} <i>Fol</i> _{5,6} <i>Aus</i> ₇	<i>Prä</i> _{3,4} <i>Dry</i> _{5,6} <i>Fol</i> ₇	<i>Prä</i> _{5,6} <i>Dry</i> ₇	<i>Prä</i> ₇

Arbeitsmittel

- Verwendung von Git empfohlen
→ <https://gitlab.cs.fau.de/>
- Abgabe der Ausarbeitung/Folien per Git (oder E-Mail)

Arbeitsmittel

- Verwendung von Git empfohlen
→ <https://gitlab.cs.fau.de/>
- Abgabe der Ausarbeitung/Folien per Git (oder E-Mail)

Organisation

- Beim Seminar gilt Anwesenheitspflicht:
Bei Abwesenheit bitte (per E-Mail) Bescheid geben
- Technikcheck rechtzeitig vor der Präsentation
- Veröffentlichung der (finalen) Folien und Ausarbeitung auf der Seminarwebseite
(Falls nicht gewünscht, bitte Bescheid geben)

Themen

Intel Software Guard Extensions³: feingranulare Form der vertrauenswürdigen Ausführung

- Welcher Schutzmechanismus wird angeboten?
- Wie kann ich von außen den Zustand der Anwendung überprüfen?
- Wie sieht der Wechsel in und aus der sicheren Umgebung aus?
- (Wie) Können Geheimnisse dauerhaft gespeichert werden?
- Welche zusätzliche Dienste gibt es?
- Wie entwickelt man eine Anwendung die Intel SGX verwendet?

³Intel® Software Guard Extensions. en. <https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html>.

AMD SEV SNP⁴ & Intel TDX⁵: vertrauenswürdigen Ausführung auf der Ebene von virtuellen Maschinen

- Welcher Schutzmechanismus wird angeboten?
- Wie kann ich von außen den Zustand der Anwendung überprüfen?
- Wie sieht der Wechsel in und aus der sicheren Umgebung aus?
- (Wie) Können Geheimnisse dauerhaft Gespeichert werden?
- Welche zusätzliche Dienste gibt es?
- Wie entwickelt man eine Anwendung die AMD SEV SNP oder Intel TDX verwendet?

⁴David Kaplan. „AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More“. en. In: (), S. 20.

⁵Intel® Trust Domain Extensions. en.

<https://www.intel.com/content/www/us/en/developer/articles/technical/intel-trust-domain-extensions.html>.

Keystone⁶: vertrauenswürdige Ausführungsumgebungen auf Basis von RISC-V

- Welcher Schutzmechanismus wird angeboten?
- Wie kann ich von außen den Zustand der Anwendung überprüfen?
- Wie sieht der Wechsel in und aus der sicheren Umgebung aus?
- (Wie) Können Geheimnisse dauerhaft Gespeichert werden?
- Welche zusätzliche Dienste gibt es?
- Wie entwickelt man eine Anwendung die Keystone verwendet?

⁶Dayeol Lee et al. „Keystone: an open framework for architecting trusted execution environments“. In: Proceedings of the Fifteenth European Conference on Computer Systems. EuroSys '20. <https://doi.org/10.1145/3342195.3387532>. New York, NY, USA: Association for Computing Machinery, Apr. 2020, S. 1–16.

ARM CCA⁷: Ein neues Modell des Vertrauens auf der ARM Architektur

- Welcher Schutzmechanismus wird angeboten?
- Wie kann ich von außen den Zustand der Anwendung überprüfen?
- Wie sieht der Wechsel in und aus der sicheren Umgebung aus?
- (Wie) Können Geheimnisse dauerhaft Gespeichert werden?
- Welche zusätzliche Dienste gibt es?
- Wie entwickelt man eine Anwendung die ARM CCA verwendet?

⁷Arm Confidential Compute Architecture. en.
<https://www.arm.com/architecture/security-features/arm-confidential-compute-architecture>.

IBM Protected Execution Facility (PEF)⁸: Ergänzung zur vertrauenswürdigen Ausführung für die POWER9 Architektur

- Welcher Schutzmechanismus wird angeboten?
- Wie kann ich von außen den Zustand der Anwendung überprüfen?
- Wie sieht der Wechsel in und aus der sicheren Umgebung aus?
- (Wie) Können Geheimnisse dauerhaft Gespeichert werden?
- Welche zusätzliche Dienste gibt es?
- Wie entwickelt man eine Anwendung die IBM PEF verwendet?

⁸Nicolae Paladi. Confidential Computing on IBM Protected Execution Facility. en-GB. <https://www.canarybit.eu/2021/paper-review-confidential-computing-for-openpower/>. Mai 2021.

Scone⁹ & Gramine¹⁰: Systeme die basierend auf Intel SGX ganze Anwendungen absichern

- Auf welcher Grundlage wird die Anwendung abgesichert?
- Ob und wie wird die zugrunde liegenden Hardware Erweiterung der Anwendung bereitgestellt?
- Welche Limitierungen haben die Ansätze?

⁹Sergei Arnautov et al. „{SCONE}: Secure Linux Containers with Intel {SGX}“. en. In: <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/arnautov>. 2016, S. 689–703.

¹⁰Chia-Che Tsai, Donald E. Porter und Mona Vij. „{Graphene-SGX}: A Practical Library {OS} for Unmodified Applications on {SGX}“. en. In: <https://www.usenix.org/conference/atc17/technical-sessions/presentation/tsai>. 2017, S. 645–658.

Ryoan¹¹ & AccTEE¹²

- Was ist die Motivation für doppelseitige Sandboxes?
- Was ist das Angreifermodell?
- Welche Technologien werden für die Sandbox der Anwendung verwendet?

¹¹Tyler Hunt et al. „Ryoan: A Distributed Sandbox for Untrusted Computation on Secret Data“. en. In: <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/hunt>. 2016, S. 533–549.

¹²David Goltzsche et al. „AccTEE: A WebAssembly-based Two-way Sandbox for Trusted Resource Accounting“. In: Proceedings of the 20th International Middleware Conference. Middleware '19. <https://doi.org/10.1145/3361525.3361541>. New York, NY, USA: Association for Computing Machinery, Dez. 2019, S. 123–135.

Glamdring¹³ & Panoply¹⁴: Schwierigkeiten bei der teilautomatisierter Partitionierung

- Was ist die Motivation der teilautomatisierten Partitionierung
- Was ist die TCB?
- Welche Probleme werden von den Papieren identifiziert?

¹³Joshua Lind et al. „Glamdring: Automatic Application Partitioning for Intel {SGX}“. en. In: <https://www.usenix.org/conference/atc17/technical-sessions/presentation/lind>. 2017, S. 285–298.

¹⁴Shweta Shinde et al. „Panoply: Low-TCB Linux Applications with SGX Enclaves“. en. In: Proceedings 2017 Network and Distributed System Security Symposium. <https://www.ndss-symposium.org/ndss2017/ndss-2017-programme/panoply-low-tcb-linux-applications-sgx-enclaves/>. San Diego, CA: Internet Society, 2017.

EActor¹⁵ & SplitBFT¹⁶

- Welche Vorteile bieten mehrere vertrauenswürdigen Ausführungskontexte
- Welche Probleme müssen für die Verwendung von mehreren vertrauenswürdigen Ausführungskontexte gelöst werden

¹⁵Vasily A. Sartakov et al. „EActors: Fast and flexible trusted computing using SGX“. In: Proceedings of the 19th International Middleware Conference. Middleware '18. <https://doi.org/10.1145/3274808.3274823>. New York, NY, USA: Association for Computing Machinery, Nov. 2018, S. 187–200.

¹⁶Messadi et al., „SplitBFT“.

ROTE¹⁷ & LCM¹⁸

- Beschreibung des Angriffs
- Welche Gegenmaßnahmen gibt es für den Angriff
 - Wie verändern die Gegenmaßnahmen die Entwicklungsprozesse für Anwendungen?

¹⁷Sinisa Matetic et al. „{ROTE}: Rollback Protection for Trusted Execution“. en. In: <https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/matetic>. 2017, S. 1289.

¹⁸Marcus Brandenburger et al. „Rollback and Forking Detection for Trusted Execution Environments Using Lightweight Collective Memory“. en. In: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). <http://ieeexplore.ieee.org/document/8023119/>. Denver, CO, USA: IEEE, Juni 2017, S. 157–168.

Foreshadow¹⁹ & SgxPectre²⁰

- Beschreibung des Angriffs
- Welche Gegenmaßnahmen gibt es für den Angriff
 - Wie verändern die Gegenmaßnahmen die Entwicklungsprozesse für Anwendungen?
- Wie haben Hardware Erweiterungsanbieter auf Angriffe reagiert?

¹⁹Jo Van Bulck et al. „FORESHADOW: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution“. en. In: (), S. 19.

²⁰Guoxing Chen et al. „SgxPectre: Stealing Intel Secrets from SGX Enclaves Via Speculative Execution“. In: 2019 IEEE European Symposium on Security and Privacy (EuroS&P). Juni 2019, S. 142–157.

Malware in the SGX Supply Chain²¹

- Offenes Thema!
- Wie kann der Entwicklungsprozess gesichert werden?
- Wie kann ich sicher sein, dass ich die richtige Anwendung signiere?

²¹Vlad Craciun et al. „Malware in the SGX Supply Chain: Be Careful When Signing Enclaves!“ In: IEEE Trans. Dependable Secur. Comput. 19.2 (2022).
<https://doi.org/10.1109/TDSC.2020.3024562>, S. 924–935.

- Themen werden nach Windhundverfahren vergeben
 - Intel SGX
 - AMD SEV SNP & Intel TDX
 - Keystone
 - ARM CCA
 - IBM PEF
 - Absicherung ganzer Anwendungen
 - Doppelseitige Sandboxen
 - Teilautomatisierte Partitionierung
 - Nutzung mehrerer vertrauenswürdigen Ausführungskontexte
 - Fork- und Rollback Angriffe
 - Seitenkanalangriffe
 - Abgesicherte Entwicklungsprozesse