Lightening the Shadows

Metadata-Light Exploit Mitigation Based on Novel Cryptography and X86

Michael LeMay, Intel Labs Security and Privacy Research

Copilot & LeMay, M. (2024, October 14). A server room with viruses hidden in the shadows behind the server racks. A glowing canine finding the viruses and illuminating them. [AI-generated image]. Microsoft.

Notices & Disclaimers

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

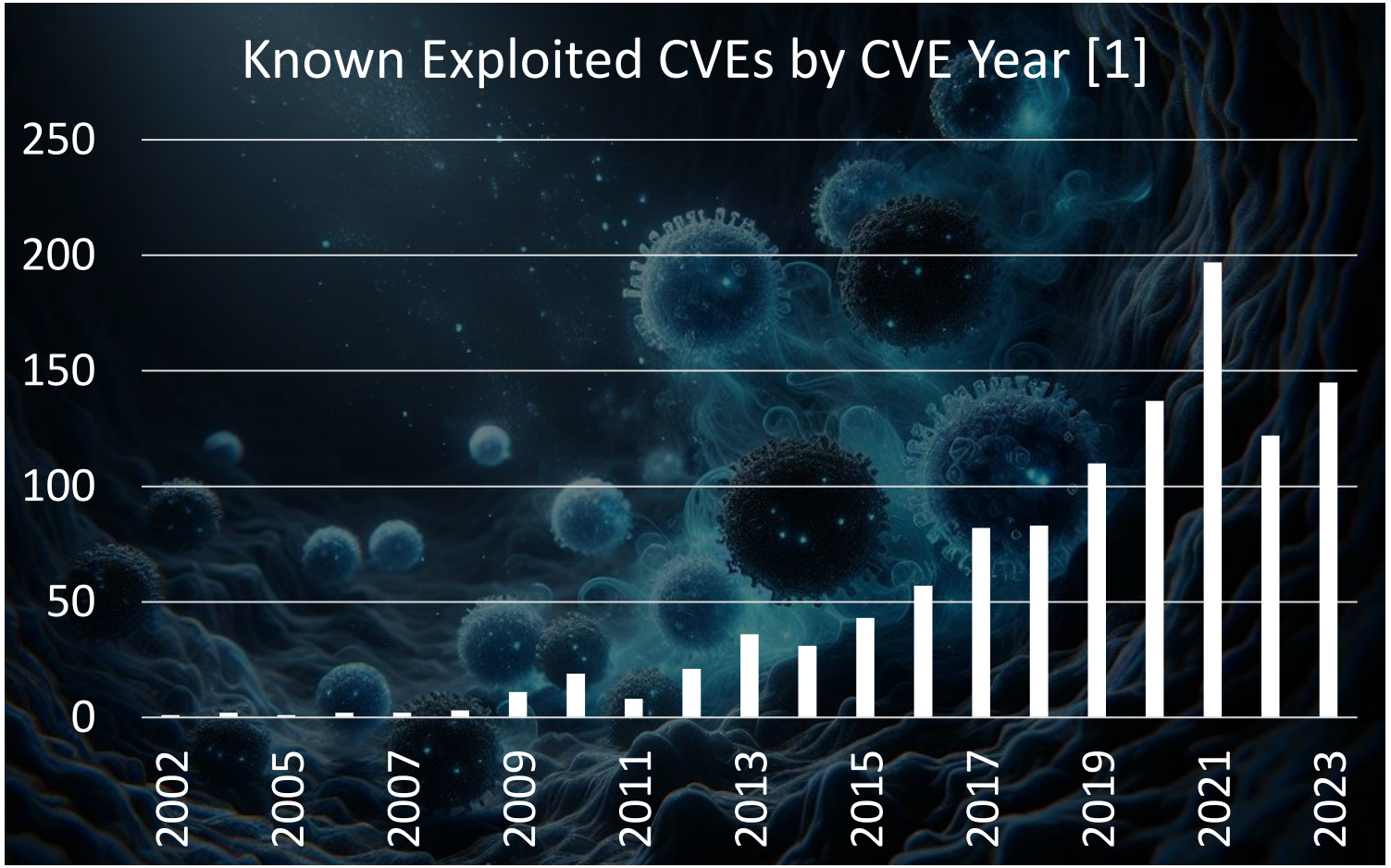
No product or component can be absolutely secure. Your costs and results may vary. Results have been estimated or simulated.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

These materials are provided “as is.” Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Shadows are Deepening as Exploits Proliferate…

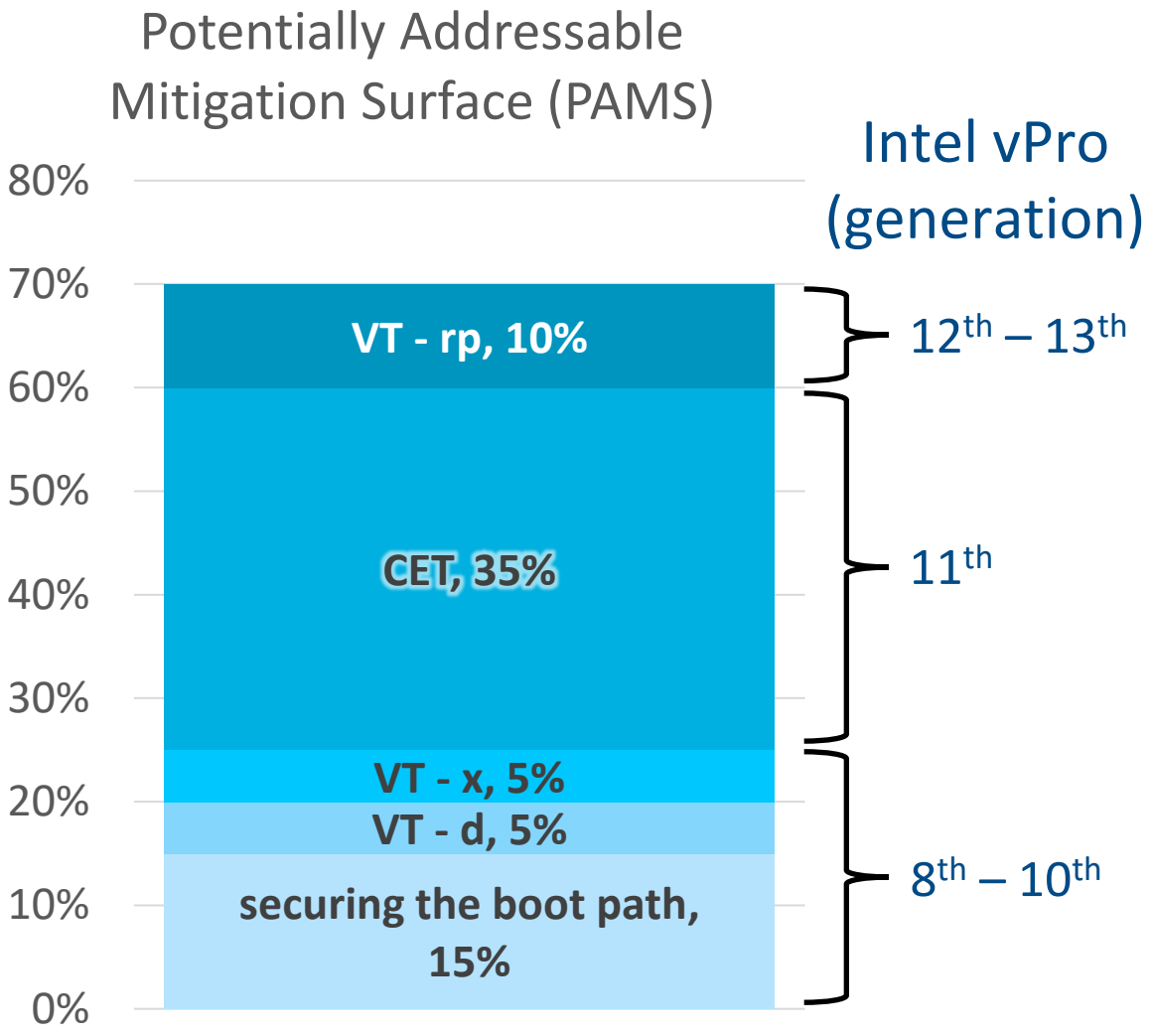


"To reduce the attack surface, we must eliminate vulnerabilities at scale by securing the building blocks of cyberspace."
--- The White House [2]

"Our internal analysis estimates that 75% of CVEs used in zero-day exploits are memory safety vulnerabilities."
--- Google [3]

[1] "Known Exploited Vulnerabilities Catalog", CISA, https://www.cisa.gov/known-exploited-vulnerabilities-catalog (retrieved 10/16/2024)
[2] https://www.whitehouse.gov/oncd/briefing-room/2024/02/26/memory-safety-fact-sheet/
[3] "Safer with Google: Advancing Memory Safety", https://security.googleblog.com/2024/10/safer-with-google-advancing-memory.html
Graph background image: Copilot & LeMay, M. (2024, October 16). A dark abyss with glowing, blue viruses pouring out of it. [AI-generated image]. Microsoft.

…but Intel Technologies Lighten the Shadows

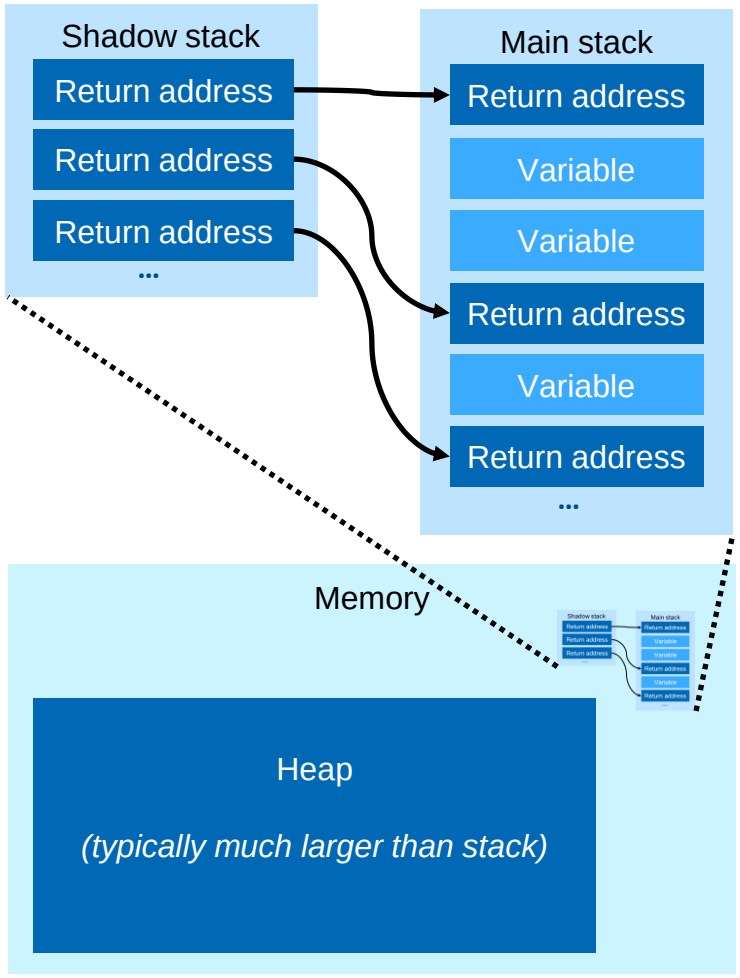


"CET protected software from ROP exploits (which was the most significant technique for real-world exploits on the OS kernel…) and after the deployment of CET attackers were looking at data-only attacks or to modify page tables… Intel® VT-rp and HLAT protection will close that vulnerability gap…"

--- Andrea Allievi, Senior Windows Core OS Developer at Microsoft

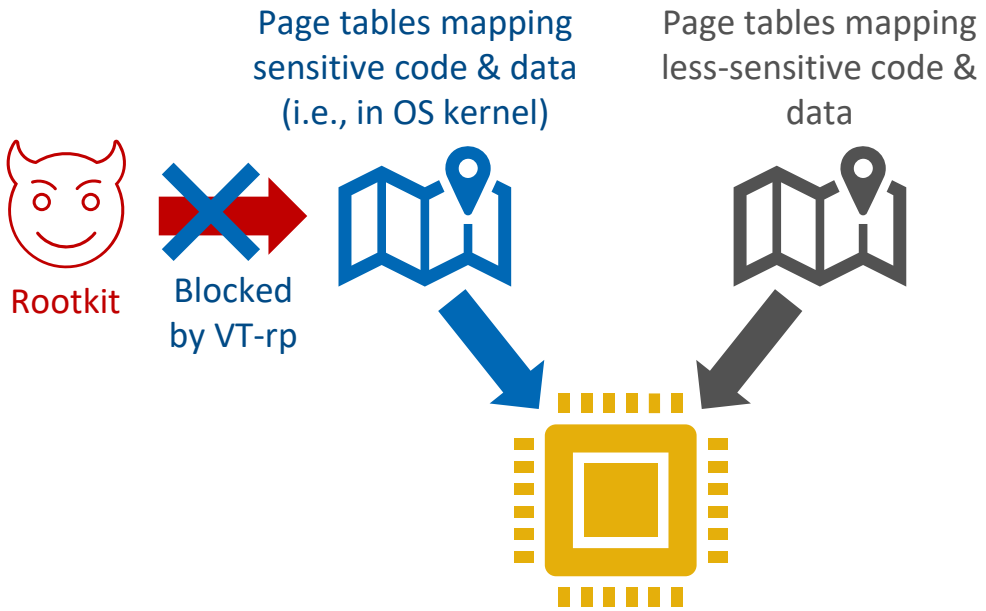
Chart data & quote source: https://www.intel.co.uk/content/dam/www/central-libraries/us/en/documents/2023-03/ioactive-intel-13th-generation-attack-surface-study-summary-report.pdf

Intel® Control-flow Enforcement Technology (CET)



- Shadow stack: Deterministically protects copies of return addresses
- Shadow stack metadata properties:
  - Lighter than stack (only return addresses)
  - High cache locality
  - Predictable access pattern

Intel® VT-Redirect Protections (VT-rp)



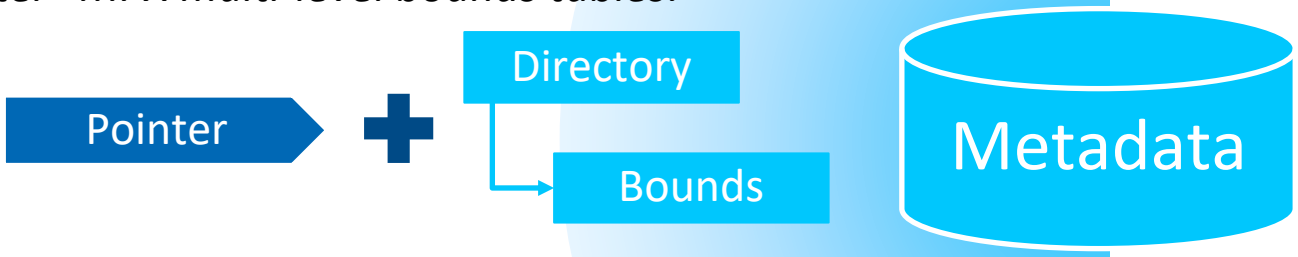
- Deterministically protects privileged memory mappings against rootkits
- VT-rp metadata properties:
  - Lightens overheads by minimizing duplication
  - Selects from distinct page tables

https://www.intel.com/content/dam/www/central-libraries/us/en/documents/intel-virtualization-technologies-white-paper.pdf

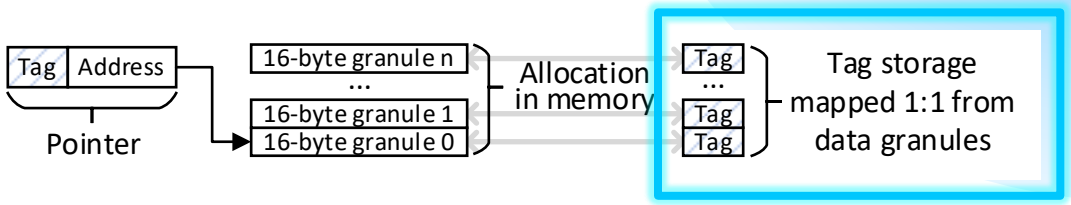
Expanding to Protect All Memory Imposed Heavy Shadows in Prior Approaches

CHERI 128-bit fat pointers:

Intel® MPX multi-level bounds tables:



ARM® Memory Tagging Extension (MTE):



Metadata element per pointer/granule for massive data heaps

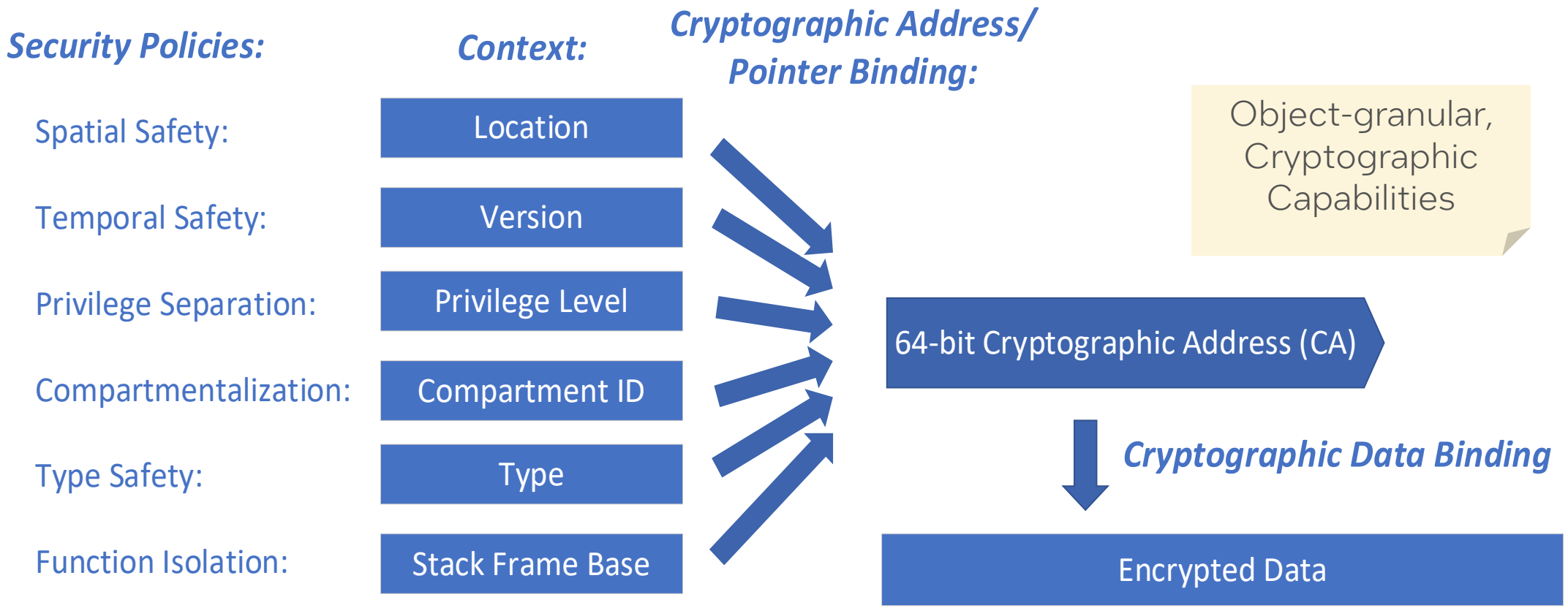
Drawbacks of metadata:

- Excessive memory usage
- Performance overhead from loading and checking metadata for each memory access
- Extensive microarchitectural and software touchpoints

Image source: Michael LeMay et al. Cryptographic Capability Computing. In *MICRO '21*. https://doi.org/10.1145/3466752.3480076

This material is based upon work supported by the Naval Information Warfare Center Pacific and the Defense Advanced Research Project Agency under Prototype Other Transaction Agreement No. N66001-23-9-4004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Information Warfare Center Pacific or the Defense Advanced Research Project Agency.

Stateless Cryptographic Addressing Removes Shadows



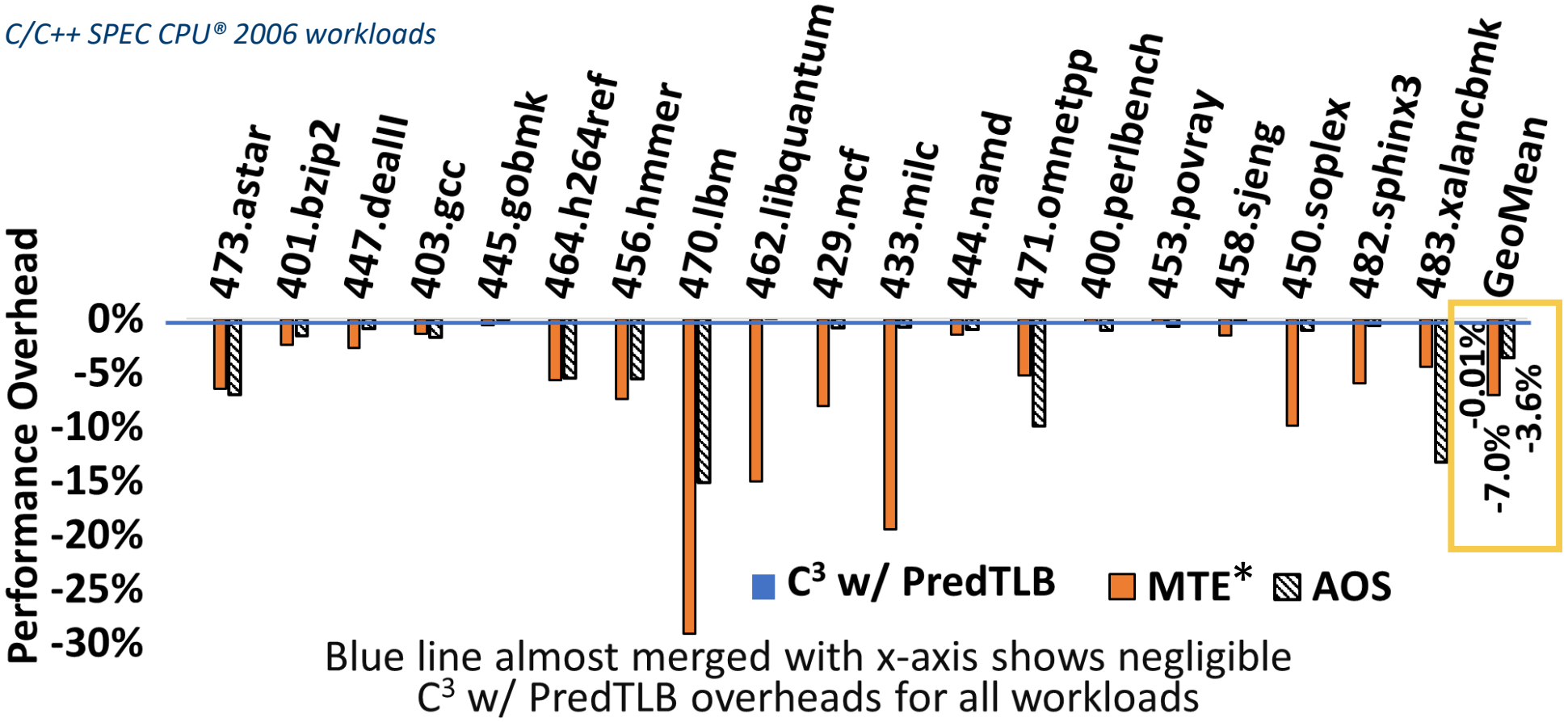
Security context is cryptographically bound to address without needing to be stored as separate metadata

MICRO 2021 paper on Cryptographic Capability Computing (C3) for spatial and temporal safety:

This material is based upon work supported by the Naval Information Warfare Center Pacific and the Defense Advanced Research Project Agency under Prototype Other Transaction Agreement No. N66001-23-9-4004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Information Warfare Center Pacific or the Defense Advanced Research Project Agency.

Negligible Simulated Performance Overhead for Cryptographic Capability Computing (C3)

Simulated C/C++ SPEC CPU® 2006 workloads



No metadata caching for MTE and AOS. Optimistic evaluation of AOS, with geomean overheads lower than the 8.4% reported in original paper.

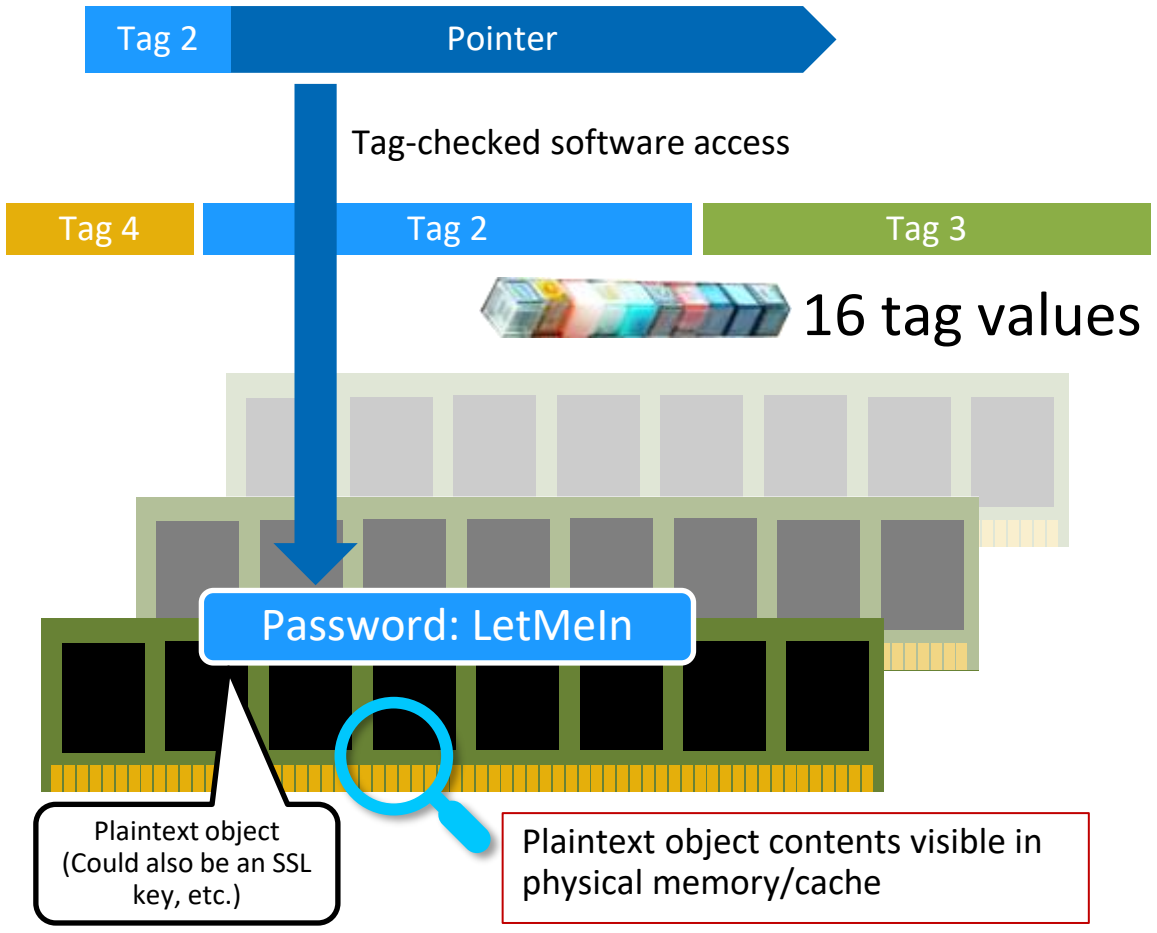
* MTE had not yet been released in any processor implementations at the time of this simulation, so we made assumptions about how metadata is stored, accessed, and cached in our model that may not correspond to current or future MTE releases.

Image source: LeMay et al. Cryptographic Capability Computing. In *MICRO '21*. https://doi.org/10.1145/3466752.3480076 Also see paper for MTE and AOS references.

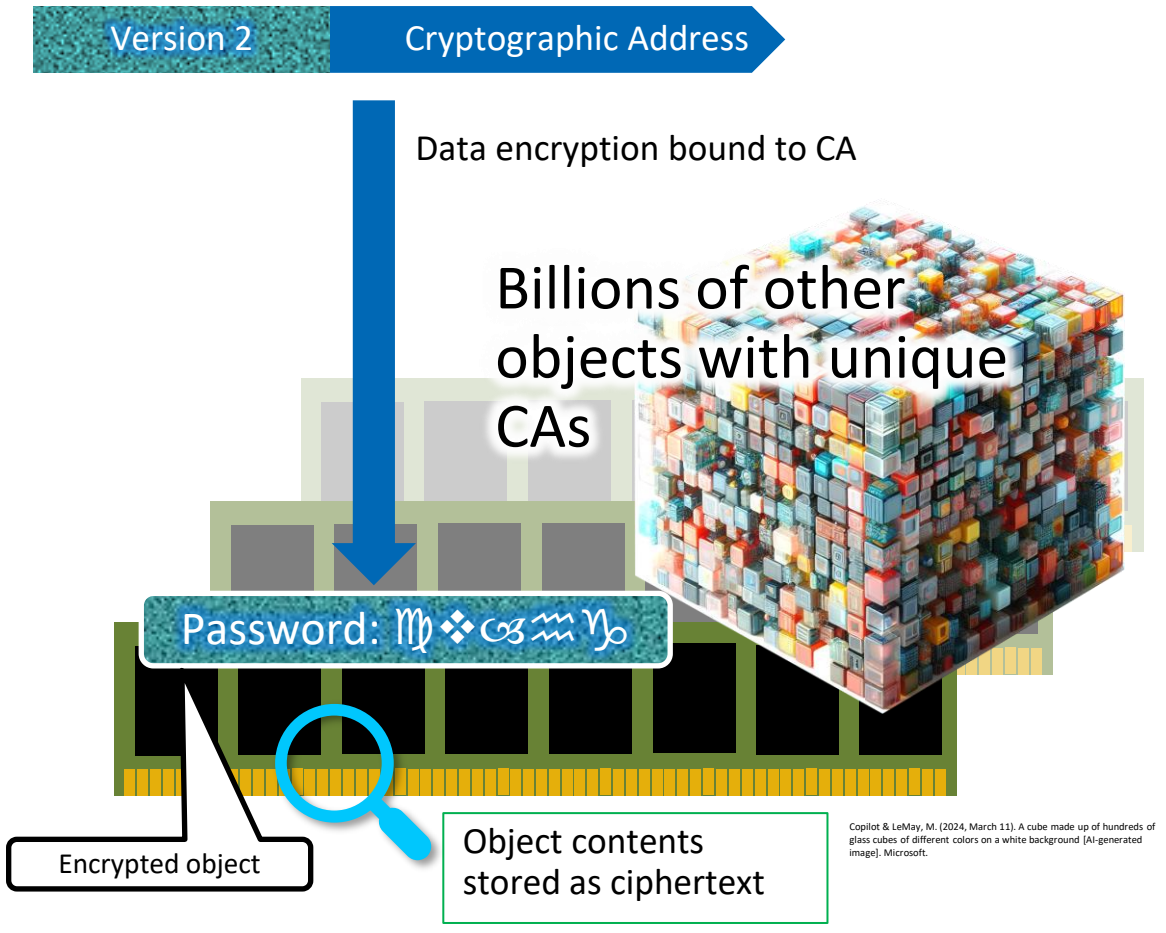
This material is based upon work supported by the Naval Information Warfare Center Pacific and the Defense Advanced Research Project Agency under Prototype Other Transaction Agreement No. N66001-23-9-4004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Information Warfare Center Pacific or the Defense Advanced Research Project Agency.

Object-Granular Crypto: Deeper than “Veneer” Solutions

Architectural security “veneer” (e.g., MTE)



Cryptographic Capability Computing (C3)

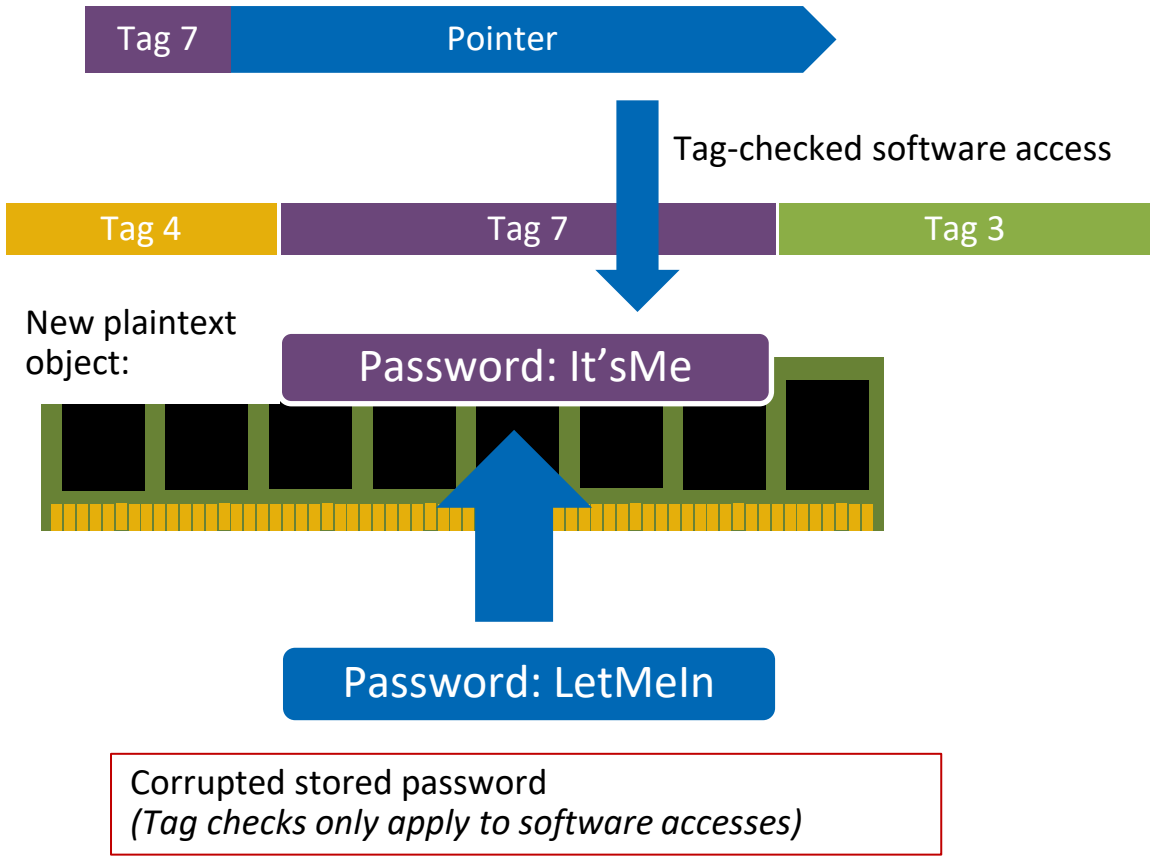


Copilot & LeMay, M. (2024, March 11). A cube made up of hundreds of glass cubes of different colors on a white background [AI-generated image]. Microsoft.

This material is based upon work supported by the Naval Information Warfare Center Pacific and the Defense Advanced Research Project Agency under Prototype Other Transaction Agreement No. N66001-23-9-4004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Information Warfare Center Pacific or the Defense Advanced Research Project Agency.

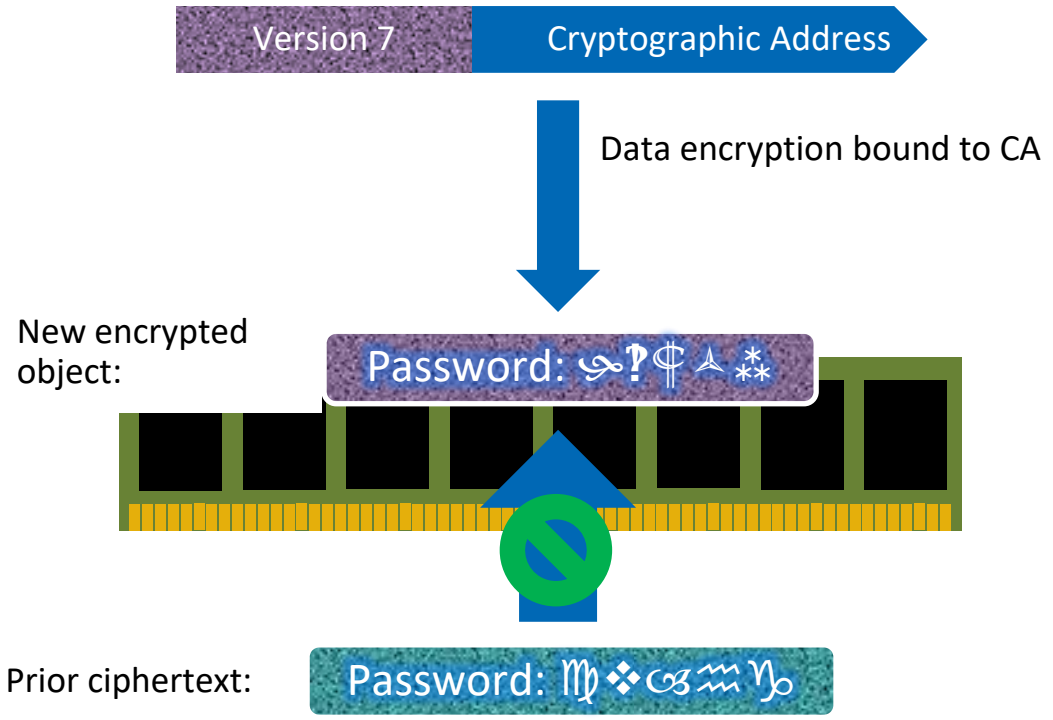
Object-Granular Crypto: Deeper than “Veneer” Solutions (cont.)

Architectural security “veneer” (e.g., MTE)



Corrupted stored password
(Tag checks only apply to software accesses)

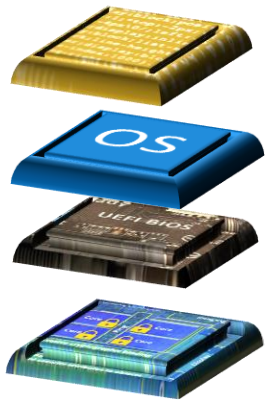
Cryptographic Capability Computing (C3)



Replay fails; CA mismatch
(Note: Ordinary memory encryption lacks CAs)

This material is based upon work supported by the Naval Information Warfare Center Pacific and the Defense Advanced Research Project Agency under Prototype Other Transaction Agreement No. N66001-23-9-4004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Information Warfare Center Pacific or the Defense Advanced Research Project Agency.

Mitigating Vulnerabilities in All Levels and Memory Regions



	Heap	Stack	Globals
Applications	✅ 📊	✅ 📊	✅
Kernel	✅ 📊	✅	✅ 📊
Firmware	✅ 📊	✅	✅ 📊
Hardware ✅ 📊	Side channels, memory errors, HW trojans, memory snooping (e.g., interposers), etc.		

Vision: Day zero automatic coverage of emergent vulnerabilities

✅ = C3 is applicable
📊 = Validated in HARDEN

Open-sourced Simics-based functional simulator, gem5-based performance simulator, and C3-enhanced toolchains.

C3 Mitigated All Relevant Vulnerability Tests in DARPA HARDEN Phase 1

Extensive binary legacy compatibility

Prior approaches rely on new instructions, pointers (e.g., Pointer Authentication, CHERI), …

Unified Cryptographic Address format

Image credits:
- Circuit board: Photo by David Durham.
- Processor with locks: Underlying processor image from an Intel public presentation.

This material is based upon work supported by the Naval Information Warfare Center Pacific and the Defense Advanced Research Project Agency under Prototype Other Transaction Agreement No. N66001-23-9-4004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Information Warfare Center Pacific or the Defense Advanced Research Project Agency.

Surpassing Fragmented Mitigations

Objective	C3	CHERI [1]	MPX [2]	MTE [3]	PAC [4]
Spatial (e.g., OOB)	●	●	●	●	
Temporal (e.g., UAF)	●	~		●	
Pointer integrity	●				●
Uninitialized use	●				
Privilege separation	●	●	●	●	
Compartmentalization	●	●			
Types	●	●			
Function Isolation	●	●			
Physical attacks	●				
Physical errors, RowHammer	●				
Encrypted transient execution	●				
Encrypted by default against HW trojans, SoC errors	●				

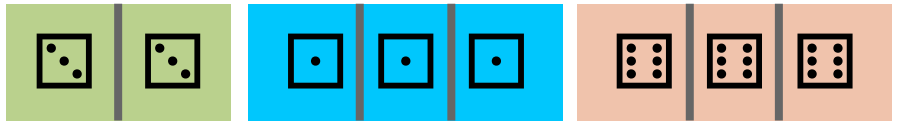
[1] Brooks Davis et al. 2019. CheriABI: Enforcing Valid Pointer Provenance and Minimizing Pointer Privilege in the POSIX C Run-time Environment. In Proceedings of ASPLOS '19, https://doi.org/10.1145/3297858.3304042
[2] Oleksii Oleksenko et al. 2017. Intel MPX Explained: An Empirical Study of Intel MPX and Software-based Bounds Checking Approaches, http://arxiv.org/abs/1702.00719
[3] Kostya Serebryany. 2019. ARM Memory Tagging Extension and How It Improves C/C++ Memory Safety. https://www.usenix.org/publications/login/summer2019/serebryany
[4] Hans Liljestrand et al. 2018. PAC it up: Towards Pointer Integrity using ARM Pointer Authentication. http://arxiv.org/abs/1811.09189

This material is based upon work supported by the Naval Information Warfare Center Pacific and the Defense Advanced Research Project Agency under Prototype Other Transaction Agreement No. N66001-23-9-4004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Information Warfare Center Pacific or the Defense Advanced Research Project Agency.

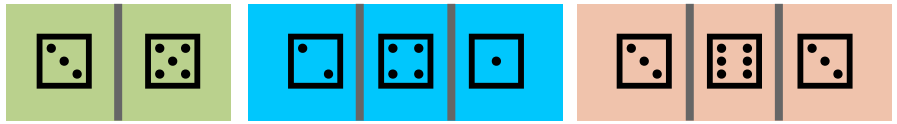
Towards Stateless Integrity for Detection

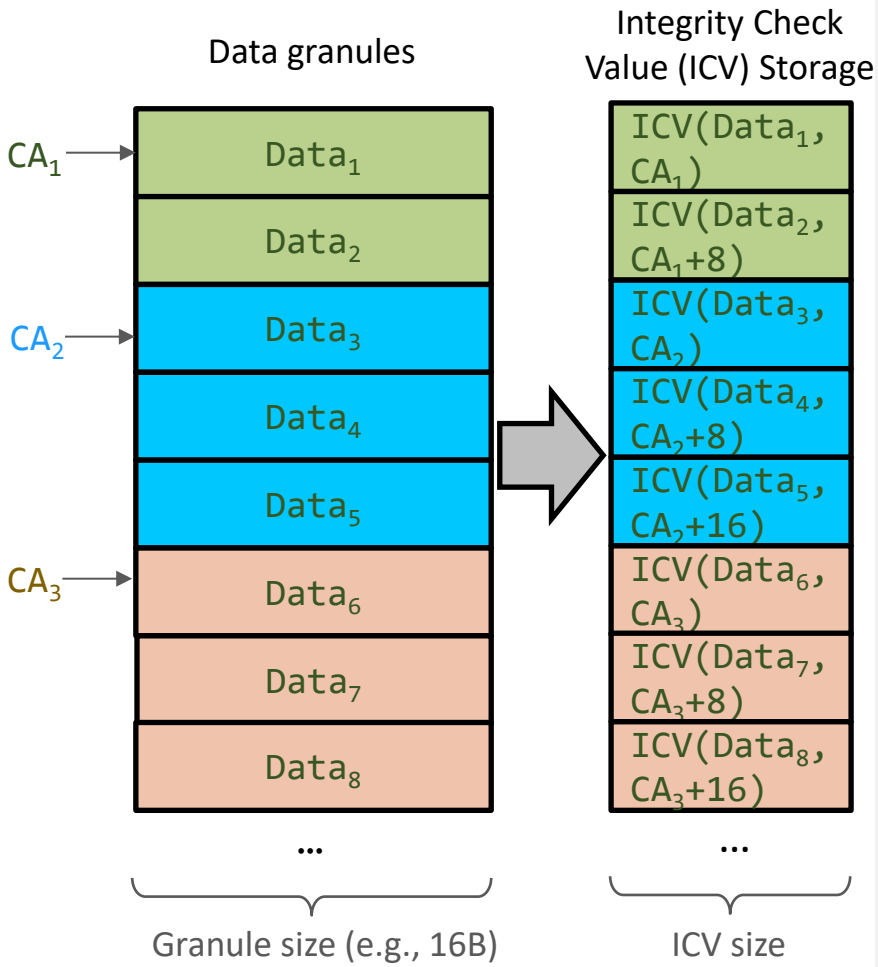
- Integrity checking enhances detection.
- See our HASP paper for how to lighten the metadata [1].
- Unlike memory tagging, invalid CA that passes for one granule is unlikely to pass for others:

Memory tagging *(all granules in allocation have same tag)*:



C3 with integrity checking:



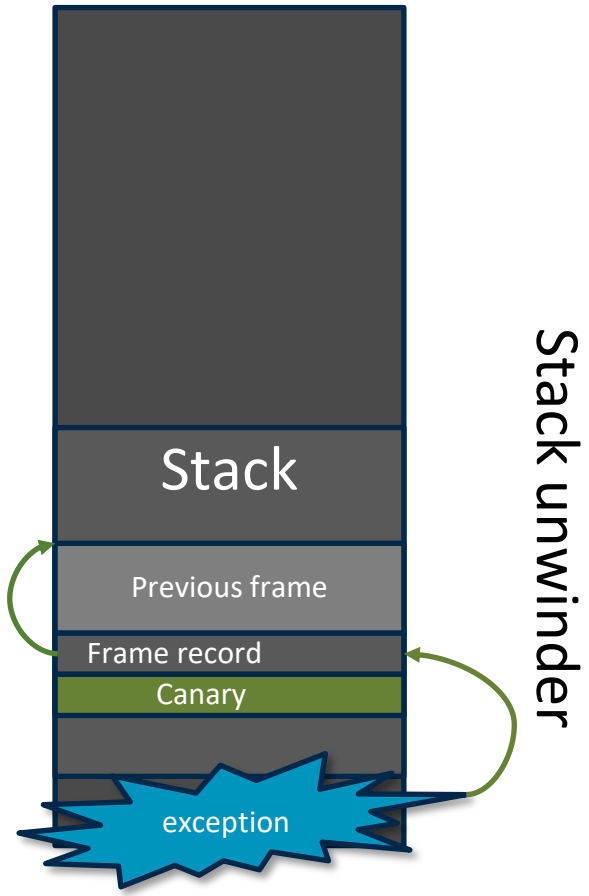


[1] Bharath Namboothiry, David Durham, Christoph Dobraunig, Michael LeMay, Cryptographic Memory Tagging: Towards Stateless Integrity, HASP 2024

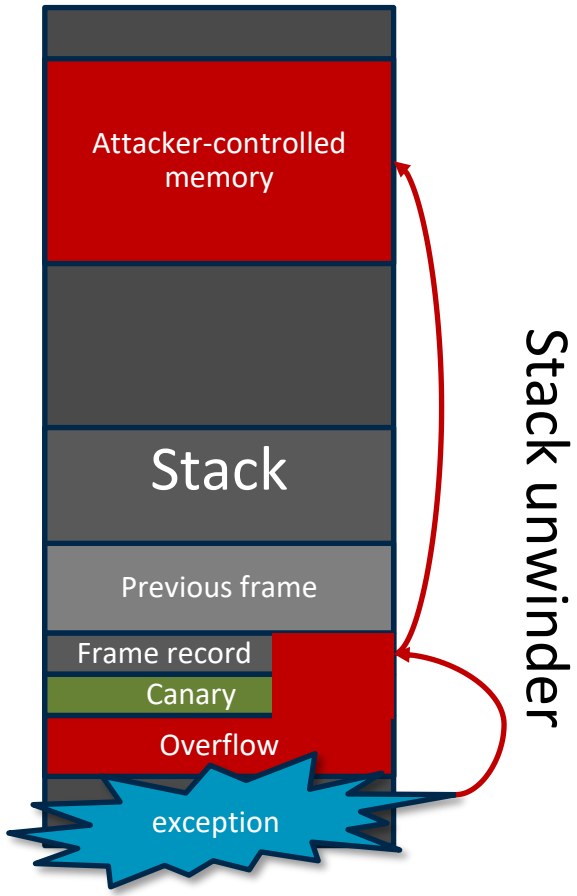
This material is based upon work supported by the Naval Information Warfare Center Pacific and the Defense Advanced Research Project Agency under Prototype Other Transaction Agreement No. N66001-23-9-4004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Information Warfare Center Pacific or the Defense Advanced Research Project Agency.

Mitigating a Recent Weird Machine: Catch Handler Oriented Programming (CHOP)*

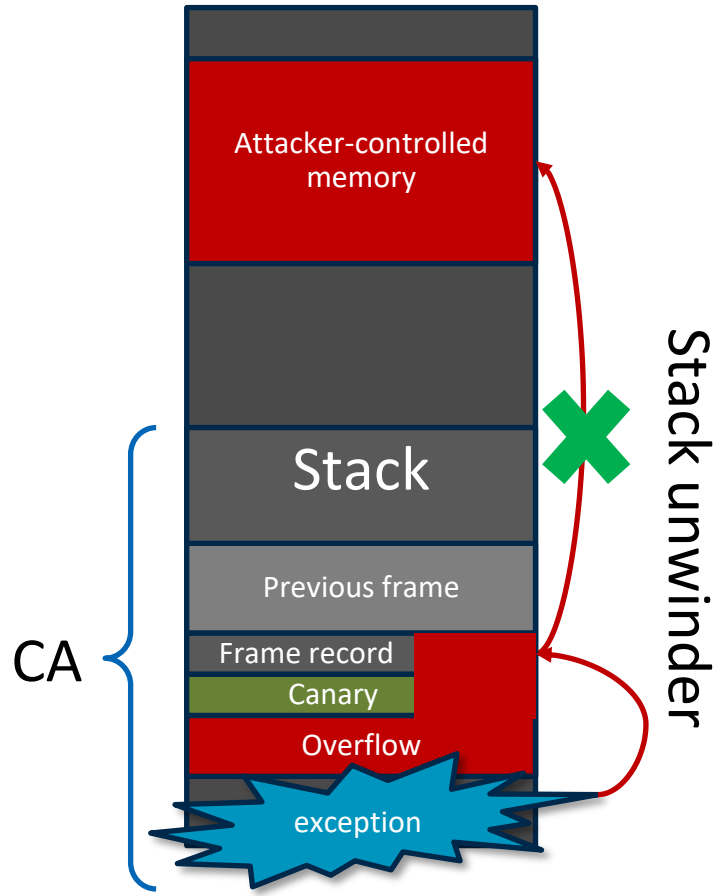
Valid execution:



CHOP co-opts exception handling to bypass canary:



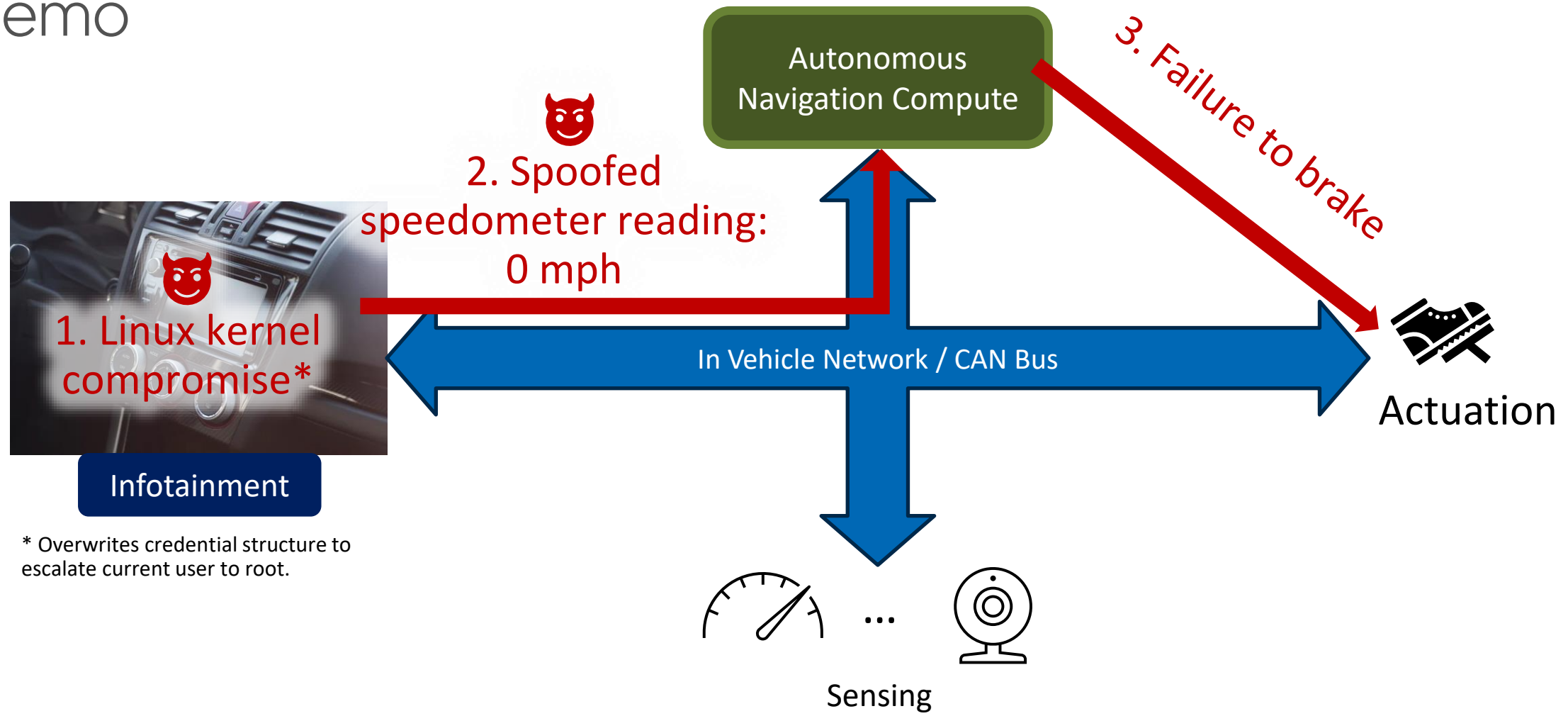
C3-enhanced unwinder enforces CA use and blocks stack pivots:



* Duta, Victor, et al. "Let Me Unwind That For You: Exceptions to Backward-Edge Protection." *NDSS*. 2023.

This material is based upon work supported by the Naval Information Warfare Center Pacific and the Defense Advanced Research Project Agency under Prototype Other Transaction Agreement No. N66001-23-9-4004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Information Warfare Center Pacific or the Defense Advanced Research Project Agency.

Demo



* Overwrites credential structure to escalate current user to root.

C3 mitigates example Linux kernel compromise, hence preserving correct vehicle operation

Image credit:
- Dashboard image: https://www.rawpixel.com/image/5906059/car-interior-photo-free-public-domain-cc0-image

This material is based upon work supported by the Naval Information Warfare Center Pacific and the Defense Advanced Research Project Agency under Prototype Other Transaction Agreement No. N66001-23-9-4004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Information Warfare Center Pacific or the Defense Advanced Research Project Agency.

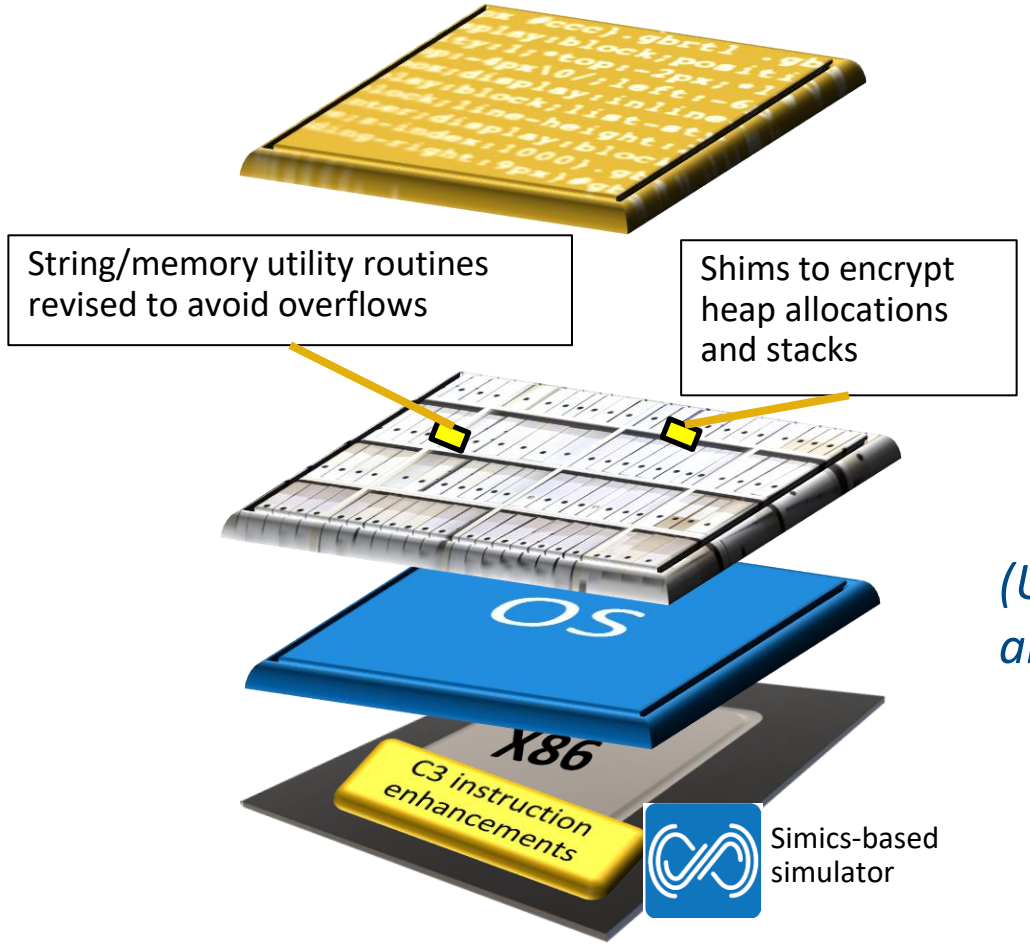
Open Source Simulators for Legacy-Compatible X86 Hardening

Unmodified Linux applications

C3-shimmed C library

Linux kernel

Simulated CPU



(UEFI firmware hardening is also available)

This material is based upon work supported by the Naval Information Warfare Center Pacific and the Defense Advanced Research Project Agency under Prototype Other Transaction Agreement No. N66001-23-9-4004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Information Warfare Center Pacific or the Defense Advanced Research Project Agency.

The opportunity: Continue shrinking the attack surface with stateless hardening

- Simics-based functional simulator: https://github.com/IntelLabs/c3-simulator
  - EDK2 UEFI firmware with C3 hardening: https://github.com/IntelLabs/c3-edk2 https://github.com/IntelLabs/c3-edk2-platforms
  - glibc C library with allocator enhancements and mem/str routine adaptations: https://github.com/IntelLabs/c3-glibc
  - Linux kernel with C3 enlightenment and hardening: https://github.com/IntelLabs/c3-linux
  - LLVM with C3 extensions (for intra-object overflow and uninitialized use hardening): https://github.com/IntelLabs/c3-llvm
- Gem5-based performance simulator: https://github.com/IntelLabs/c3-perf-simulator

Acknowledgments: *Intel Labs:* David M. Durham (C3 project Tech Lead), Chace Clark, Sergej Deutsch, Christoph Dobraunig, Ken Grewal, Christopher Gutierrez, Luis Kida, Yonghae Kim, Hans Goran Liljestrand, Salmin Sultana.
Intel Product Assurance and Security: Gabriel Gomes, Sebastian Österlund
Intel Federal: Cindy Polsky, Jose Salame, Gloria Velazquez
Intel Labs Interns: Yonatan Achamyeleh, Floris Gorter, Bharath Namboothiry, Joey Rudek
University of California San Diego: Prof. Dean Tullsen (C3 project PI), Yash Jain, Nagendra Jamadagni, Joey Rudek, Hosein Yavarzadeh


intel