Linke Song

unik-lif.github.io | Unik-lif

+86-18511165662 | songlinke@iie.ac.cn | Beijing, 100190, China

Research Interests

Broadly interested in computer systems, with an emphasis on OS-level isolation, virtualization, sandboxing, and trusted execution environments (TEEs). Currently focused on secure infrastructure for LLM agents — including container-based sandboxing for untrusted code execution, KV cache security in model serving systems, and scalable confidential computing. Seeking a research internship where I can apply my systems background to building secure and scalable agent infrastructure.

Education

- **University of Chinese Academy of Sciences** Beijing, China
  - **PhD in Computer Systems Organization** *Sept. 2024 – present*
    Institute of Information Engineering | Advisor: Prof. Wei Song, Prof. Wenhao Wang
  - **M.S. in Cyberspace Security** *Sept. 2022 – Jul. 2024*
    Institute of Information Engineering | Advisor: Prof. Wenhao Wang | GPA: 3.83/4.00
  - **B.S. in Cyberspace Security** *Sept. 2018 – Jul. 2022*
    Advisor: Prof. Wenhao Wang, Prof. Dongdai Lin | GPA: 3.70 (6/20)

Research Projects

- **LLM Side-Channel Attack on KV Cache**
  Tools: Python, SGLang, GPT-Cache, LLaMAFactory code paper demo SGLang #1504
  - **Problem:** KV cache reuse in LLM serving systems creates a timing side channel: during decode, a regular user issuing crafted requests can observe TTFT (Time-To-First-Token) reductions whenever their input overlaps with a victim's cached prompt in GPU memory, leaking what the victim has sent. Found exploitable in both SGLang (prefix sharing) and GPT-Cache (semantic similarity matching).
  - **Approach:** In SGLang, sharing a single additional token with Llama 3.1 8B Instruct / 70B GPTQ INT4 yields a measurable latency drop. Designed a token-by-token prompt recovery method that iteratively guesses the next token and verifies via the timing signal. Single-token timing differences are extremely small and easily buried by GPU voltage/frequency noise — developed a countermeasure that stabilizes measurements, raising TPR to **99%**.
  - **GPT-Cache Attack:** Discovered a distinct attack vector in GPT-Cache, where semantically similar queries containing identical sensitive information trigger TTFT speedups due to similarity-based cache matching, enabling privacy inference over a finite set of candidate prompts.
  - **Defense:** Proposed a coarse-grained token-sharing defense that expands the attacker's guessing space, significantly reducing extraction success rate.
  - **Impact:** One of the earliest two independent teams to report KV-cache privacy risks to SGLang (the other from ByteDance Security Research, same week). Shared findings with SGLang core developers at their biweekly meeting (Oct. 19, 2024), receiving strong interest and acknowledgment.
  - **Publication:** Submitted to USENIX Security'25, ACM CCS'25; accepted at **TIFS'25** (CCF-A journal).
- **NaCRE: Native Confidential Containers on RISC-V**
  Tools: C, OpenSBI, Linux Kernel, RunC, Qemu code | Working prototype; preparing for arXiv submission
  - **Problem:** Existing confidential containers repurpose hardware mechanisms (e.g., TEEs) not designed for the container paradigm, sacrificing native-ness and lightweight properties. Alternatives like Arm CCA treat containers as TEE workloads, relying on large contiguous memory regions that cannot be shared across containers — no hardware primitives or abstractions exist that are purpose-built for native confidential containers.
  - **Approach:** Reused RISC-V's native PMP (Physical Memory Protection) mechanism and extended it with a bitmap-based scheme and targeted MMU modifications. This protects user page-table pages, user data pages, and user-space page-table entries *without* fragmenting Linux's native memory allocation — containers remain ordinary processes rather than mini-VMs.
  - **vs. Existing solutions:** Unlike Kata Containers and gVisor — which depend on virtualization or confidential-computing hardware, prohibitively heavy for RISC-V and low-load scenarios — NaCRE introduces a new hardware abstraction layer designed specifically for containers. Native Linux memory allocation is preserved with targeted hardening; no invasive kernel modifications are required.
  - **Security:** Container lifecycle protection is integrated into the full address-allocation pipeline, achieving strong isolation without sacrificing container native-ness or performance.

- **Performance:** Near-zero overhead vs. native Docker on compute-intensive workloads; under 2× slowdown on memory-intensive workloads — significantly outperforming microVM-based approaches.
- **My role:** Sole developer of the full-stack design — hardware ISA primitives, Linux kernel modifications (identified and patched critical paths while preserving native RSS counter semantics and metadata correctness), OpenSBI integration, eCall interface and protocol design. Preparing for arXiv submission.

- **NestedSGX: Nested Enclaves in Confidential VMs**
  Tools: Rust, C, Python, Linux Kernel Module, Qemu, AMD SEV-SNP code paper
  - **Problem:** Confidential VMs face a large TCB in the guest OS, creating an exploitable attack surface. Existing approaches lack a mechanism to establish trusted enclaves *inside* a CVM while keeping the guest OS out of the TCB.
  - **Approach:** Leveraged AMD SEV-SNP's VMPL (Virtual Machine Privilege Level) mechanism to introduce a lightweight hypervisor *within* the confidential VM. This de-privileges the guest OS, so that even a fully compromised guest kernel cannot access enclave memory — drastically shrinking the TCB.
  - **Compatibility:** Built a trusted enclave runtime atop Occlum and Intel SGX SDK, maintaining compatibility with the existing Intel SGX ecosystem. This allows unmodified SGX applications to run inside the nested enclave with minimal porting effort.
  - **Engineering:** Modified low-level Linux kernel drivers; wrote the in-VM hypervisor in Rust, handling page faults, system error paths, and cross-privilege transitions via custom trampoline mechanisms.
  - **Recognition:** Received **2 Artifact Evaluation badges** for reproducibility. Invited by the Asterinas community to present this work at their online seminar. **Submitted to ASPLOS'24, ACM CCS'24; accepted at NDSS'25** (CCF-A conference).

Patents and Publications

C=Conference, J=Journal, P=Patent, S=In Submission, T=Thesis

[J.1] The Early Bird Catches the Leak: Unveiling Timing Side Channels in LLM Serving Systems.
[**Linke Song**, Zixuan Pang], Wenhao Wang*, Zihao Wang, XiaoFeng Wang, Hongbo Chen, Wei Song, Yier Jin, Dan Meng, Rui Hou
TIFS' 25

[C.1] The Road to Trust: Building Enclaves within Confidential VMs.
Wenhao Wang, **Linke Song** (first student author), Benshan Mei, Shuang Liu, Shijun Zhao, Shoumeng Yan*, XiaoFeng Wang, Dan Meng, Rui Hou
NDSS'25

Skills

- **Programming Languages:** C, Rust, Python
- **DevOps & Version Control:** Git, Docker
- **Specialized Area:** Trusted Execution Environment, Operating System, Virtualization, Container
- **Familiar Architecture:** x86, RISC-V