

# Chanyoung Park

Computer Systems Security Lab, Dept. of Computer Science and Engineering,  
Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea

chanyoung.think@gmail.com · sites.google.com/view/chanyoungpark · github.com/blazer502

## EDUCATION

**Ulsan National Institute of Science and Technology (UNIST)** Ulsan, South Korea  
*Combined M.S./Ph.D. in Computer Science and Engineering* Mar. 2022 – Aug. 2026 (expected)

- Advisor: Prof. Hyungon Moon (Computer Systems Security Lab).
- Dissertation: *An Efficient and Secure Memory Allocator for Use-after-Free Prevention.*

**Ulsan National Institute of Science and Technology (UNIST)** Ulsan, South Korea  
*B.S. in Computer Science and Engineering* Mar. 2015 – Feb. 2022

## RESEARCH INTERESTS

My research lies in **systems and software security**, with a focus on memory safety, isolation, program analysis, operating systems, and secure cloud infrastructure. I build mechanisms that define and enforce trust boundaries across compilers, runtimes, operating systems, and hardware.

## PUBLICATIONS

*Bold marks my name; underline marks primary authorship. Venue tags: KIISE grade, BK21 IF, CSRankings; SCI = SCI-indexed journal.*

### International Publications

- [1] **FaaSlim: Partial Caching of Snapshot-based VMs for Serverless Computing.**  
Sanghyeon Eom, **Chanyoung Park**, Gihong Lee, Hyungon Moon, and Young-ri Choi.  
*ACM International Conference on Supercomputing (ICS), Belfast, Northern Ireland, United Kingdom, 2026.* (KIISE A, BK21 IF: 2, CSRankings)
- [2] **Efficient Use-After-Free Prevention with Pooling, OS-assisted, and Opportunistic Page-Level Sweeping.**  
**Chanyoung Park**, Yeongjun Kwak, and Hyungon Moon.  
*IEEE Transactions on Dependable and Secure Computing (TDSC), 2025.* (SCI)
- [3] **Defeating Use-After-Free Bugs Using Memory Sweeper Without Stop-the-World.**  
Junho Ahn, KangHyuk Lee, **Chanyoung Park**, Hyungon Moon, and Youngjin Kwon.  
*IEEE Symposium on Security and Privacy (S&P), San Diego, CA, USA, May 2025.* (KIISE S, BK21 IF: 4, CSRankings) [GitHub]
- [4] **Selective On-Device Execution of Data-Dependent Read I/Os.**  
**Chanyoung Park**, Minu Chung, and Hyungon Moon.  
*USENIX Conference on File and Storage Technologies (FAST), Santa Clara, CA, USA, Feb. 2025.* (KIISE S, BK21 IF: 3, CSRankings) (Artifacts Available & Functional) [GitHub]
- [5] **Efficient Use-After-Free Prevention with Opportunistic Page-Level Sweeping.**  
**Chanyoung Park** and Hyungon Moon.  
*Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, Feb. 2024.* (KIISE S, BK21 IF: 2, CSRankings) [GitHub]

### Domestic Publications

- [6] **Analysis of Memory Allocator Call Sites Used Only Within the Stack Using SVF.**  
Seon Ha, **Chanyoung Park**, Yeongjun Kwak, and Hyungon Moon.  
*Annual Symposium of KIPS (ASK), vol. 31, no. 1, 2024.*

[7] **An Operating System Support-based Prevention Mechanism for Use-After-Free Attacks on the glibc Memory Allocator.**

**Chanyoung Park**, Jaehyu Lee, Daeyeon Kim, and Hyungon Moon.

*Journal of KIISE (JOK)*, 2023.

## PATENTS

---

*Granted patents: 1 U.S. and 1 Korean patent. Pending applications: 1 U.S., 4 Korean, and 2 PCT applications.*

[P1] **Use-After-Free Protection via Pooling, OS-Assisted, and Opportunistic Page-Level Sweeping.** Hyungon Moon, **Chanyoung Park**, and Yeongjun Kwak. *Filed: KR 10-2025-0091241 (2025.07.07); KR 10-2025-0027787 (2025.03.04); PCT/KR2026/003422 (2026.03.04).*

[P2] **Selective On-Device Execution of Data-Dependent Read I/Os.** Hyungon Moon, **Chanyoung Park**, and Minu Chung. *Filed: US 19/439,086 (2026.01.02); KR 10-2025-0008325 (2025.01.20).*

[P3] **Efficient Use-After-Free Prevention with Opportunistic Page-Level Sweeping.** Hyungon Moon and **Chanyoung Park**. *Filed: PCT/KR2025/000618 (2025.01.10); KR 10-2024-0022931 (2024.02.16).*

[P4] **Apparatus and Method for Deallocating Dynamically Allocated Memory.** Hyungon Moon, **Chanyoung Park**, Jaehyu Lee, and Daeyeon Kim. *Granted: US 12,353,322 (2025.07.08); KR 10-2787851 (2025.03.24). Filed: KR 10-2022-0170530 (2022.12.08).*

## RESEARCH PROJECTS

---

*Selected projects as a graduate researcher.*

### **Integrated Platform for Expanding and Safely Applying Memory-Safe Languages**

IITP · 2024.06 – 2026.08

*Role: Intel MPK compartmentalization for unsafe Rust and FFI (Foreign Function Interface), guided by program analysis; manuscript under submission.*

### **Binary Micro-Security Patching under Software Supply-Chain Environments**

IITP · 2024.04 – 2026.08

*Role: Fast, safe security patching of deployed binaries to avoid downtime for long-running services (under submission).*

### **Samsung Electronics Industry-Academic Cooperation Projects**

Samsung Electronics · 2022.05 – 2024.04; 2024.04 – 2026.05

*Role: developed SODE (on-device execution of data-dependent read I/Os) and contributed to FaaSlim (partial caching of snapshot-based VMs for serverless cold starts).*

### **RISC-V-Based Secure CPU for Embedded Malware Detection and Recovery**

IITP · 2022.03 – 2024.12

*Role: HushVac / HushVac+ opportunistic page-level sweeping allocator for use-after-free prevention; contributed to SwiftSweeper.*

## HONORS & AWARDS

---

### **USENIX FAST Student Grant**

*USENIX Association*

Feb. 2025

*Santa Clara, CA, USA*

## RESEARCH EXPERIENCE

---

### **Research Assistant, Computer Systems Security Lab, UNIST**

Mar. 2022 – Aug. 2026

### **Efficient and Secure Memory Management**

Mar. 2022 – Aug. 2026

- Designed and implemented **HushVac / HushVac+**, an opportunistic page-level sweeping allocator that mitigates use-after-free vulnerabilities at low runtime and memory overhead

(NDSS 2024, IEEE TDSC 2025).

- Contributed to **SwiftSweeper**, a concurrent memory-sweeping architecture that removes stop-the-world pauses while reclaiming memory (*IEEE S&P 2025*).
- Designed and implemented **MarKern**, an operating-system-supported use-after-free prevention mechanism for the glibc memory allocator (*JOK 2023*).

**Compartmentalization for Unsafe Rust and FFI** Jan. 2025 – Aug. 2026

- Designing an Intel MPK-based isolation mechanism for unsafe Rust and FFI, using static program analysis to identify memory that crosses the safe/unsafe boundary.

**Serverless Computing & Resource Management** Dec. 2025 – Mar. 2026

- Contributed to **FaaSlim**, a snapshot-based serverless system that partially caches VM pages to reduce cold-start overhead and idle memory use (*ACM ICS 2026*).

**Near-Data / In-Storage Computing** Jan. 2024 – Feb. 2025

- Designed and implemented **SODE**, which offloads data-dependent read I/O operations to the storage device, eliminating host-to-device round trips for data-intensive workloads (*USENIX FAST 2025*).

**Program Analysis for Systems Optimization** Nov. 2023 – Feb. 2024

- Contributed to a static-analysis pass, built on LLVM and Static Value-Flow Analysis (SVF), that identifies allocator call sites confined to the stack to enable targeted allocation-overhead optimizations (*ASK 2024*).

**OPEN-SOURCE SOFTWARE & ARTIFACTS**

**SwiftSweeper** (IEEE S&P 2025; co-author) [github.com/casys-kaist/SwiftSweeper](https://github.com/casys-kaist/SwiftSweeper)

**SODE** (USENIX FAST 2025; *Artifacts Available & Functional*) [github.com/cssl-unist/sode](https://github.com/cssl-unist/sode)

**HushVac** (NDSS 2024) [github.com/cssl-unist/hushvac](https://github.com/cssl-unist/hushvac)

**INVITED TALKS**

**Efficient Use-After-Free Prevention with Opportunistic Page-Level Sweeping** Sep. 2024  
*Samsung Security Tech Forum (SSTF) 2024* Seoul, South Korea

**TEACHING EXPERIENCE**

**Teaching Assistant, UNIST** Mar. 2022 – Jul. 2024  
*Operating Systems; Software Hacking and Defense; Building Customized Computers; Advanced Programming; Principles of Programming Languages* Ulsan, South Korea

- For *Operating Systems*, independently managed the PintOS programming assignments, evaluated student submissions, and provided feedback.
- For *Building Customized Computers*, prepared LLVM-based compiler software stacks that students study and extend.

**PROFESSIONAL SERVICE**

**Artifact Evaluation Committee** Jan. – Feb. 2025  
*ACM European Conference on Computer Systems (EuroSys 2025)* Rotterdam, Netherlands