

# Alexander J. Gaidis

Doctoral Researcher, Brown University CS

+1 (845) 554-8993  
agaidis@cs.brown.edu  
Providence, Rhode Island  
United States

cs.brown.edu/~agaidis/  
alexgaidis  
ajgaidis  
Scholar

*Systems security: passionate about OS kernel protection and software hardening.*

## Education

### Ph.D. in Computer Science

GPA: 4.0/4.0 | Advisor: Vasileios P. Kemerlis

Brown University  
Sept. 2021 – Present

### M.S. in Computer Science

GPA: 4.0/4.0

Brown University  
Jan. 2019 - Dec. 2019

### B.A. in Computer Science

GPA: 3.5/4.0

Brown University  
Sept. 2013 - Dec. 2018

### Full-Time Visiting Student

GPA: 3.9/4.0

MIT  
Sept. 2017 - May 2018

## Work Experience

See “*Research*” below for more thorough explanations.

### Crash Override

Graduate Intern

Remote  
June 2023 – Aug. 2023

- Advisor: Theofilos Petsios
- Designed and implemented test case minimization system with Intel Processor Trace (PT)

### Intel (IPAS STORM)

Offensive Security Research Intern

Remote  
May 2022 – Dec. 2022

- Advisor: Joao Moreira
- Researched applications for Intel Memory Protection Keys (MPK)

### Intel (IPAS STORM)

Offensive Security Research Intern

Remote  
June 2021 – Oct. 2021

- Advisor: Joao Moreira
- Designed and implemented runtime library to improve hardware-assisted CFI schemes

### Secure Systems Lab, Brown University

Research Assistant

Providence, Rhode Island  
Apr. 2020 – May 2021

- Advisor: Vasileios P. Kemerlis
- Researched and improved `sysfilter`'s ability to handle different runtimes
- Developed a kernel exploit to break `kR^X`'s execute-only memory and code diversification protections

### Raytheon

Systems Engineering & Security Graduate Intern

Portsmouth, Rhode Island  
May 2019 – Aug. 2019

- Advisor: Glenn Wojcik
- Engineered a minimal RHEL image for a sea-mine neutralization system

**ABA English***Software Developer**Barcelona, Spain**Jan. 2018 – Feb. 2018*

- Advisors: Brian Subirana & Maria Perillo
- Developed an English proficiency test for Amazon's Echo

**Auto-ID Laboratory, MIT***Research Assistant**Cambridge, Massachusetts**Aug. 2017 – Dec. 2018*

- Advisor: Brian Subirana
- Researched the democratization of IoT voice technologies (e.g., Amazon Echo)

**Viral Communications Laboratory, MIT (Media Lab)***Research Assistant**Cambridge, Massachusetts**Nov. 2017 – May 2018*

- Advisors: David Anderton & Andrew Lippman
- Researched bias in television news through sentiment analysis
- Developed an Ionic mobile phone application to help raise awareness of natural disasters

**Setpoint Medical***Research Intern**Santa Clarita, California**May 2014 – Aug. 2014*

- Advisor: Michael Faltys
- Designed experiments to evaluate RF heating by neuro-modulation devices

## Teaching

*(All comments below are in addition to holding 1:1 office hours with students.)***CSCI1380: Distributed Systems***Graduate Teaching Assistant**Brown University**Spring 2023*

Advised on course development and high-level direction

**CSCI1650: Software Security & Exploitation***Graduate Teaching Assistant**Brown University**Fall 2022*

Helped teach 150 CS students software security &amp; exploitation

**CSCI1650: Software Security & Exploitation***Head Teaching Assistant**Brown University**Fall 2019*

Led 5 TAs to help teach 100 CS students software security &amp; exploitation

**Artificial Intelligence***Teaching Assistant**Northeastern University Software College (China)**Fall 2018*

Developed AI projects and homework on game theory, search algorithms, and more

**CSCI E-11: The Frontiers of Computer Science***Teaching Assistant**Harvard University Extension School**Spring 2018, Fall 2018*

Implemented novel IoT conversational learning tool for 200 students

**Girls Who Code***Instructor**Norton LifeLock**Summer 2017*

Used project-based approach to teach CS from Scratch to Java to high-schoolers

**CSCI0040: Intro to Scientific Computing and Problem Solving***Teaching Assistant**Brown University**Spring 2017*

Developed CS exams, problem sets, &amp; projects for 65 engineering students

(★: Tier-1 venue)

## Conference Papers

1. ★ Marius Momeu, **Alexander J. Gaidis**, Jasper v.d. Heide, and Vasileios P. Kemerlis. IUBIK: Isolating User Bytes in Commodity Operating System Kernels via Memory Tagging Extensions. In *IEEE Symposium on Security and Privacy (S&P)*, 2025
2. ★ Neophytos Christou, **Alexander J. Gaidis**, Vaggelis Atlidakis, and Vasileios P. Kemerlis. Eclipse: Preventing Speculative Memory-error Abuse with Artificial Data Dependencies. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024
3. ★ Di Jin, **Alexander J. Gaidis**, and Vasileios P. Kemerlis. BeeBox: Hardening BPF against Transient Execution Attacks. In *33rd USENIX Security Symposium (USENIX Security 24)*, 2024
4. ★ **Alexander J. Gaidis**, Vaggelis Atlidakis, and Vasileios P. Kemerlis. SysXCHG: Refining Privilege with Adaptive System Call Filters. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2023
5. **Alexander J. Gaidis**, Joao Moreira, Ke Sun, Alyssa Milburn, Vaggelis Atlidakis, and Vasileios P. Kemerlis. FineIBT: Fine-grain Control-flow Enforcement with Indirect Branch Tracking. In *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)*, 2023

## Patents

1. Michael C. Gaidis and **Alexander J. Gaidis**. Magnetic domain wall shift register memory device readout, February 10 2015. US Patent 8,951,811
2. Michael C. Gaidis and **Alexander J. Gaidis**. Magnetic domain wall shift register memory device readout, July 8 2014. US Patent 8,772,889

## Workshops

1. **Alexander J. Gaidis**, Jamie Gabbay, Joao Moreira, and Vasileios P. Kemerlis. NOPoutNG: Improving the Effectiveness of Hardware-assisted Control-flow Integrity via Dynamic Landing Pad Elision. In *Cyber Security Experimentation and Test Workshop (CSET)*, 2025
2. **Alexander J. Gaidis**, Vaggelis Atlidakis, and Vasileios P. Kemerlis. SysXCHG: Refining Privilege with Adaptive System Call Filters. In *New England Security Day (NESD)*, 2024
3. Di Jin, **Alexander J. Gaidis**, and Vasileios P. Kemerlis. BPF-Box: Hardening BPF against Transient Execution Attacks. In *New England Security Day (NESD)*, 2024

## Posters

1. Jamie Gabbay, **Alexander J. Gaidis**, Joao Moreira, Ke Sun, Alyssa Milburn, Vaggelis Atlidakis, and Vasileios P. Kemerlis. NOPout: Dynamic Pruning of Indirect Branch Targets for FineIBT, March 2024
2. Brian Subirana, **Alexander J. Gaidis**, Sanjay Sarma, Richard Cantwell, Jon Stine, Peter Oliveira-Soens, Esteva Tarragò, Ferran Hueto Puig, Prithvi Rajasekaran, and Alex Armengol Urpi. A Secure Voice Name System to Wake Smart Devices, February 2019

## SysXCHG

Secure Systems Lab, Brown University

Software Hardening, Kernel Protection

- **Problem:** Userland processes in Linux are given unfettered access to the system call API, widening the attack surface of the kernel and providing attackers with a rich post-exploitation arsenal.
- **Contribution:** My colleagues and I developed a system call filtering enforcement mechanism called **SysXCHG** that enables programs to run in accordance with the principle of least privilege. At a high-level, **SysXCHG** allows system call filters to be dynamically changed at runtime on `execve`, adapting protection to the currently executing program. To ensure attackers cannot tamper with filters, potentially widening their privileges, **SysXCHG** embeds filters in their corresponding binaries and computes a cryptographic signature over them. **SysXCHG** also includes an optimized installation and filtering mechanism called `xfilter` that improves number-based system call filtering over the current state-of-the-art filtering scheme, `seccomp-BPF`. `xfilter` provides processes with different “views” of the system call table where filtering can occur.
- **Publication:** Appeared in CCS 2023.
- **Paper (link) / Software (link)**

## IUBIK

Secure Systems Lab, Brown University

Data-only Attacks, Kernel Protection

Technical University of Munich

- **Problem:** The effectiveness of many modern mitigations against data-only attacks in OS kernels hinges on their ability to identify data that is security-sensitive and should thus be protected. Unfortunately, researchers continue to uncover new objects in monolithic OS kernels that attackers can target to circumvent defenses. With no concrete definition of what data constitutes as security-sensitive, existing mitigations participate in a reactionary arms race, responding to threats after they have been identified in practice.
- **Contribution:** My colleagues and I introduced IUBIK, moving one step ahead of the attacker in this cat-and-mouse game by redefining sensitive data as any data in the kernel that is not copied from userland—in contrast to the piece-meal approaches that identify sensitive data after their use in an attack. IUBIK leverages this definition to shift the focus of mitigation mechanisms from protecting individual, sensitive objects to neutralizing the primitives abused to corrupt data. Our insight is drawn from our systematic analysis of kernel exploits that revealed most data-only attacks require overlapping a kernel object with a malicious user-controlled object, copied into the kernel via standard interface functions (e.g., `copy_from_user`). Identifying and isolating these “usercopy” primitives forms the basis of IUBIK. To isolate usercopy primitives, IUBIK uses hardware features present in commercially available ARM CPUs—namely, ARM MTE and PAC—to isolate user-controlled objects in “shadow memory,” preventing them from ever accessing (i.e., over-writing) any kernel-controlled, security-critical data.
- **Publication:** Appeared in IEEE S&P 2025.
- **Paper (link)**

## BeeBox

Secure Systems Lab, Brown University

Kernel Protection

- **Problem:** BPF increases the attack surface of the (Linux) kernel and has been used extensively as the underlying mechanism for mounting transient execution attacks against the kernel from userland.
- **Contribution:** My colleagues and I developed a novel security architecture for BPF, called BeeBox, that protects against transient execution attacks by sandboxing BPF’s runtime data in a manner similar to software fault isolation. Additionally, BeeBox uses static analysis and various domain-specific properties to selectively remove enforcement checks to improve performance without sacrificing security. The design is thorough, considering the security of eBPF, cBPF, and BPF helper functions.
- **Publication:** Appeared in USENIX Security 2024.
- **Paper (link) / Software (link)**

## Eclipse

Secure Systems Lab, Brown University

Software Hardening, Kernel Protection

- **Problem:** Memory errors and speculative execution attacks have historically been thought of as two disjoint domains. However, recent attacks, such as speculative probing, have demonstrated that memory corruption and speculative execution vulnerabilities can be combined to bypass memory-safety-based mitigations (e.g., (K)ASLR).
- **Contribution:** My colleagues and I designed and implemented a compiler-assisted solution for blocking such attacks. At a high level, it works by first identifying potentially vulnerable indirect branches before subsequently hardening them with instrumentation that propagates artificial data dependencies to prevent them from dereferencing attacker-controlled pointers during speculative execution. This solution advances the state-of-the-art as it can be implemented solely in software; it does not require complete elimination of side channels or memory corruption vulnerabilities; and it does not completely stop speculation (as with serializing instructions like `lfence`), thus offering good performance.
- **Publication:** Appeared in CCS 2024.
- **Paper (link) / Software (link)**

## FineIBT

Secure Systems Lab, Brown University

Software Hardening

IPAS STORM, Intel

- **Problem:** Control-flow integrity (CFI) solutions built with Intel Indirect Branch Tracking (IBT) technology are too permissive, allowing hijacked control flow to target the entry of most functions.
- **Contribution:** With the help of Intel, I have been designing, building, and testing a fine-grained CFI approach using Intel IBT. The solution adds function prototype hash checks to the Intel hardware feature, ensuring that indirect branches can only target a subset of the total set of functions. Prototype hashes are loaded into a register at a given call site and checked after an “anchoring” `endbr` instruction in the prologue of the callee. Our minimal design is performant and works seamlessly across shared object boundaries.
- **Publication:** Appeared in RAID 2023.
- **Paper (link) / Software (link)**

## NOPoutNG

Secure Systems Lab, Brown University

Software Hardening

IPAS STORM, Intel

- **Problem:** Fine-grained control-flow integrity (CFI) schemes built with Intel Indirect Branch Tracking (IBT) technology are too permissive when applied across shared-object boundaries. They often fail to sufficiently harden a system against control-flow hijacking attacks due to all exported library functions being valid indirect call targets at compile-time.
- **Contribution:** To increase the granularity of IBT-based CFI solutions across shared-objects, I designed and built a run-time library that live-patches code to make unused functions unreachable by eliding `endbr` instructions. The main functionality of the library runs as a constructor, but objects loaded dynamically (e.g., using `dlopen/dlsym`) required me to design a tamper-proof scheme to safely edit non-writable data and code pages.
- **Publication:** Will appear in CSET 2025.

## Awards

---

<b>CSAW Applied Research Paper Award Finalist (top 10) for IUBIK</b>	2025
<b>USENIX Security Distinguished Artifact Reviewer</b>	2024
<b>NDSS Student Support Grant Recipient</b>	2024

## Service

---

### Paper Reviewing

<b>TIFS</b>	IEEE Transactions on Information Forensics and Security	2023
-------------	---	------

### Artifact Evaluation Committee

<b>SEC</b>	USENIX Security Symposium	2025
<b>SEC</b>	USENIX Security Symposium ( <b>Distinguished Artifact Reviewer</b> )	2024
<b>NDSS</b>	Network and Distributed System Security Symposium	2024

### University Service

<b>Ph.D. Admissions Committee</b>	Dept. of Computer Science, Brown University	2025
<b>Ph.D. Admissions Committee</b>	Dept. of Computer Science, Brown University	2023
<b>Ph.D. Mentorship Program (Mentor)</b>	Dept. of Computer Science, Brown University	2022 – Pres.

### Community Service

<b>Wildlife Rehabilitation</b>	Lisinia Doğa, Burdur, Turkey	Spring 2013
--------------------------------	------------------------------	-------------

## Mentorship

---

### Doctoral Students

<b>Simran Kadadi</b>	Dept. of Computer Science, Purdue University	Summer 2017 – Present
<b>Benjamin Spiegel</b>	Dept. of Computer Science, Brown University	Fall 2022 – Spring 2023

### Master's Students

<b>Shukai Ni</b>	Dept. of Computer Science, Brown University	Spring 2023 – Spring 2024
------------------	---	---------------------------

### Undergraduate Students

<b>Isabelle Shapiro</b>	Dept. of Computer Science, Brown University	Fall 2025 – Present
<b>Rebecca Chou</b>	Dept. of Computer Science, Brown University	Spring 2025 – Present
<b>Yi Hao Tan</b>	Dept. of Computer Science, Brown University	Spring 2024 – Spring 2025
<b>Jamie Gabbay</b>	Dept. of Computer Science, Brown University	Spring 2023 – Spring 2025
<b>Sierra Rowley</b>	Dept. of Computer Science, Brown University	Fall 2021 – Spring 2022
<b>Ethan Greenberg</b>	Dept. of Computer Science, Brown University	Summer 2020

## Other Interests

---

<b>Cycling</b>	Raced professionally with wins domestically and a top 20 at the international Tour of Tobago
<b>Visual Art</b>	Charcoal works and experimental darkroom photography displayed on Brown/MIT campus
<b>Music</b>	Played guitar and saxophone in paid gigs and built guitar effects pedals