

# Vaggelis Atlidakis

*Crash Override*

Ch. Trikoupi 19, 5th Floor  
Athens 10679, Greece

+30 698 992 9982

<https://vatlidak.github.io/>

[v.atlidakis@gmail.com](mailto:v.atlidakis@gmail.com)

## Research Interests

I am interested in all aspect of software security and reliability, with a focus on operating systems security, program analysis for large-scale software testing, and robustness of machine learning systems.

## Education

- Aug 2020 **Ph.D. in Computer Science**, *Columbia University*, Department of Computer Science, Graduate School of Arts and Sciences, New York, NY, USA.  
Thesis: *Structure and Feedback in Cloud Service API Fuzzing*  
Advisor: Prof. Roxana Geambasu
- February 2018 **M.Phil. in Computer Science**, *Columbia University*, Department of Computer Science, Graduate School of Arts and Sciences, New York, NY, USA.  
Candidacy: *Automated Software Testing: Approaches and Tradeoffs*
- May 2016 **M.S. in Computer Science**, *Columbia University*, Department of Computer Science, Fu Foundation School of Engineering and Applied Science, New York, NY, USA.  
**GPA:** 3.85/4
- June 2006 **B.S. in Informatics & Telecommunications**, *University of Athens*, Department of Informatics, Athens, Greece.  
**GPA:** 9/10

## Professional Appointments

- 2024–present **Software Engineer**, *Crash Override*, New York, USA.
- 2023–2024 **Software Engineer**, *Lacework*, London, UK.
- 2021–2023 **Postdoctoral Research Associate**, *Secure Systems Lab, Brown University* (mentor: Prof. Vasileios Kemerlis).
- Fall 2018 **Research Intern**, *Microsoft Security Risk Detection*, Microsoft Research (mentors: Dr. Patrice Godefroid, Dr. Marina Polishchuk).
- Summer 2018 **Research Intern**, *Microsoft Security Risk Detection*, Microsoft Research (mentors: Dr. Patrice Godefroid, Dr. Marina Polishchuk).
- Summer 2017 **Research Intern**, *Project Springfield*, Microsoft Research (mentors: Dr. Patrice Godefroid, Dr. Marina Polishchuk).
- 2012–2013 **Research Associate**, *European Organization for Nuclear Research (CERN)* (supervisor: Ignacio Reguero).

---

## Honors and Awards

- Jan 2021 **Computing Innovation Fellowship**, National Science Foundation (NSF/CRA).
- Jan 2015 **Scholarship** (for Ph.D. Studies), Gerondelis Foundation.
- Jan 2013 **Distinguished Award**, Chair's Distinguished Award, Columbia University.
- June 2005 Graduated **summa cum laude**, National and Kapodestrian University of Athens, Department of Informatics and Telecommunications.

---

## Research Activities

2021–2023 **Secure Systems Lab**, Brown University.

► **Software Hardening** [C.1]. 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). My colleagues and I designed and implemented **eclipse** [CCS '24]: a compiler-assisted solution for blocking such attacks. At a high level, eclipse 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.

► **Kernel Protection** [C.2, C.5]. User-space 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. My colleagues and I developed a system call filtering enforcement mechanism called **SysXCHG** [CCS '23]: a compiler-assisted solution for blocking such attacks. At a high level, SysXCHG enables programs to run in accordance with the principle of least privilege and 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 the 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`.

► **Hardware Security** [C.3]. 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. To increase the granularity of IBT-based CFI solutions across shared objects, along with collaborators from Brown university, we designed and built **FineIBT** [RAID '23]: a CFI enforcement mechanism that improves the precision of hardware-assisted CFI solutions—like Intel IBT—by instrumenting program code to reduce the valid/allowed targets of indirect forward-edge transfers. The instrumentation of FineIBT is compact, incurring low runtime and memory overheads, and generic, so as to support different CFI policies. FineIBT has been adopted by Intel and (a variant of it) was more recently upstreamed to the Linux kernel.

► **Automated Vulnerability Detection in Deep Learning Frameworks [C.4].** Deep Learning (DL) frameworks involve code that spans different low- and high-level languages. In this context, it is no surprise that, due to missing sanity checks and mismatched security assumptions, untrusted inputs may transfer through the stack of APIs and reach memory-unsafe code. Given that, how can we automatically detect such code defects and report them to framework developers in an actionable manner? To address this question, my collaborators and I introduced a two-fold bottom-up approach, implemented in our **IvySyn** framework [Usenix Sec '23]. IvySyn leverages the statically-typed nature of native APIs in order to automatically perform type-aware mutation-based fuzzing on low-level kernels. Next, given a set of offending inputs that trigger memory safety (and fatal runtime) errors in low-level, native DL (C/C++) code, it automatically synthesizes code snippets in managed languages (e.g., Python), which propagate offending input through high(er)-level APIs. Such code snippets essentially act as “Proof of Vulnerability” (PoV), as they demonstrate the existence of bugs in native, C/C++ code that attackers can target (and potentially abuse) via high-level APIs. IvySyn has already helped TensorFlow and PyTorch DL framework developers identify and fix numerous previously-unknown security vulnerabilities, implicitly and explicitly affecting millions, if not billions, of users worldwide.

2014–2020 **Software Systems Lab, Columbia University.**

► **Security of Machine Learning Systems [C.9].** Deep Neural Networks (DNNs) perform exceptionally well on many machine learning tasks, including safety- and security-sensitive applications, such as self-driving cars, malware classification, face recognition, and critical infrastructure. Robustness against malicious behavior is important in many of these applications. Yet, in recent years it has become clear that DNNs are vulnerable to a broad range of attacks, including adversarial examples, where the adversary finds small perturbations to correctly classified inputs that cause a DNN to produce an erroneous prediction. Adversarial examples pose a serious threat to security-critical applications. Based on a novel connection between robustness to adversarial examples and differential privacy, my collaborators and I proposed **PixelDP** [S&P '19]: the first certifiably robust defense against adversarial examples that scales to large, real-world DNNs and datasets (e.g., Google’s Inception network for ImageNet) and applies broadly to arbitrary model types. PixelDP also enables a firewall-like security architecture, where a small model is prepended to an existing, already trained one to make it more robust. Such an architecture is common in traditional software systems but unique in ML workloads.

► **Testing Tools for Data-drive Applications [C.10].** A key aspect characterizing modern applications is their increased reliance on data and data-driven decision-making. While often beneficial, this practice can have subtle detrimental consequences, such as discriminatory or racially offensive effects. My collaborators and I argued that such effects are bugs that should be tested for and debugged in a manner similar to functionality, reliability, and performance bugs. To this end, we developed **FairTest** [EuroS&P '17]: a testing toolkit for data-driven applications that identifies unwarranted association between application outputs and user subpopulations, including sensitive groups (e.g., minorities defined by race or gender).

► **POSIX Abstractions in Modern OSes** [C.11, M.1]. In order to shed light into a number of questions regarding the use of POSIX abstractions by modern application, my collaborators and I conducted a broad **measurement study** [EuroSys '16] on Android, OSX, and Ubuntu. We found out that modern applications rely on abstractions not supported by the POSIX API and therefore custom, user-space libraries—providing new, necessary abstractions—are being implemented on top of POSIX. This layering causes mismatches, inefficiencies, and even security risks. For instance, we observed that new abstractions heavily use POSIX extension APIs (such as `ioctl`) to implement their desired functionality, suggesting that POSIX lacks appropriate abstractions for modern workloads. Extension APIs are problematic because their invocations cannot be properly mediated by the OS, putting pressure on user-space libraries and kernel device drivers to implement correct and coherent protections of these invocations. Our findings have broad implications related to the future of POSIX-compliant OS portability, which the systems research community and the relevant standard bodies will likely need to address in the near future.

2017–2018 **Microsoft Security Risk Detection**, Microsoft Research.

► **Testing of Cloud Services** [C.7, C.8, PP.1]. Most cloud services—like those running on Amazon Web Services (AWS) and Microsoft Azure—are programmatically accessed through REST APIs. Yet, tools for automatically testing cloud services through their REST APIs and checking whether those services are reliable and secure are still in their infancy. Along with my collaborators from Microsoft Research, we did work in the area of cloud service API testing. We introduced the idea of stateful REST API fuzzing and built **RESTler** [ICSE '19]: the first Stateful REST API Fuzzer. RESTler is used internally by multiple Microsoft teams for security, reliability, functional correctness, and regression testing, and has found hundreds of bugs in multiple production O365, Azure, and Bing cloud services. In 2020, RESTler became open-source, and its usage has been steadily growing since, both inside and outside Microsoft. Furthermore, we showed how RESTler can be extended with **active checkers** [ICST '20] that automatically test and detect violations of security rules that capture desirable properties of REST APIs in a modular and efficient way. Using active checkers, RESTler found several security-relevant bugs in several deployed, production Azure and Office365 cloud services. Finally, in my most recent work in cloud service testing, we built **Pythia** a new fuzzer that augments stateful REST API fuzzing with coverage-guided feedback and learning-based mutations. Our experimental evaluation showed that Pythia can report previously-unknown errors in production-scale cloud services that were beyond the reach of baseline stateful REST API fuzzing.


---

## Impact and Technology Transfer

- FineIBT** [C.3] Adopted by Intel and (a variant of it) upstreamed to the Linux kernel (April 2022 and onward). <https://tinyurl.com/ha22jf54>
- IvySyn** [C.4] Uncovered 61 previously-unknown security vulnerabilities in TensorFlow and PyTorch (*i.e.*, the two most popular DL frameworks), and was awarded with 39 unique CVEs. <https://bit.ly/3Rq1DZO>
- Cynthia** [C.6] Uncovered 28 previously-unknown bugs in popular ORMs, including 3 release blockers. <https://tinyurl.com/8r7tmdtc>
- RESTler** [C.8] Used internally by multiple Microsoft teams since 2017, and has found hundreds of bugs in multiple production O365, Azure, and Bing cloud services; open-sourced in 2020. <https://tinyurl.com/2jwht4yh>

---

## Software Artifacts

- Eclipse** [C.1]  <https://gitlab.com/brown-ssl/eclipse>
- SysXCHG** [C.2]  <https://gitlab.com/brown-ssl/sysxchg>
- FineIBT** [C.3]  <https://gitlab.com/brown-ssl/fineibt>
- IvySyn** [C.4]  <https://gitlab.com/brown-ssl/ivysyn>
- EPF** [C.5]  <https://gitlab.com/brown-ssl/epf>
- Cynthia** [C.6]  <https://github.com/theosotr/cynthia>
- RESTler** [C.8]  <https://github.com/microsoft/restler-fuzzer>
- FairTest** [C.10]  <https://github.com/columbia/fairtest>
- Libtrack** [C.11]  <https://github.com/columbia/libtrack>

---

## Talks & Presentations

### Invited Talks

- o **Automated Software Testing With Blackbox Fuzzing** [C.4, C.7, C.8]  
November 2023 Department of Computer Science, University of Southern California,  
*Host:* Prof. William G.J. Halfond
- o **Responding to the Trillion-Dollar Threat of Cybercrime** [C.2—C.10]  
October 2023 Department of Informatics and Telecommunications, University of Athens,  
*Host:* Prof. Alexandros Ntoulas

### Conference Presentations

- October 2020 **Checking Security Properties of Cloud Services REST APIs.**  
International Conference on Software Testing, Verification, and Validation (ICST),  
Porto, Portugal. Video: <https://www.youtube.com/watch?v=t7oYbu90ChY>
- August 2019 **RESTler: Stateful REST API Fuzzing.**  
International Conference on Software Engineering (ICSE), Montreal, QC, Canada.  
Video: <https://www.youtube.com/watch?v=M2hmDSkUYsg>
- April 2016 **POSIX Abstractions in Modern Operating Systems: The old, the New, and the Missing.**  
European Conference on Computer Systems (EuroSys), London, UK, 2016.  
Video: <https://www.youtube.com/watch?v=3ZyUN1pCpCA>
- August 2012 **Changing the unchoking policy for an enhanced BitTorrent.**  
European Conference on Parallel Processing (EuroPar), Rhodes Island, Greece.

---

## Funding

- [F.1] Fuzzing of RESTful Cloud Services. **Co-PI**  
Computing Innovation Fellows 2020 program,  
National Science Foundation (NSF/CRA), CIF2020-BU-04,  
**\$318,288**, 01/01/2021 – 05/31/2023.

---

## Patents

- [PT.1] P. Godefroid, L. Zhang, **V. Atlidakis**, C. R. Rioux, C. Y. Kim, and Y. Chen. Vulnerability Remediation Recommendation System. (US patent pending.)
- [PT.2] P. Godefroid, M. Polishchuk, and **V. Atlidakis**. Automatic, Intelligent Cloud Service Testing. U.S. Patent 10,977,161. Issued: December 2019.

---

## Publications

Google Scholar [<https://tinyurl.com/t5f964ut>]

DBLP [<https://tinyurl.com/22s7drb8>]

★: Tier-1 venue

### Conference Proceedings

- [C.1] ★ N. Christou, A. J. Gaidis, **V. Atlidakis**, and V. P. Kemerlis. Eclipse: Preventing Speculative Memory-error Abuse with Artificial Data Dependencies. In *Proceedings of the 31st ACM Conference on Computer and Communications Security (CCS)*, Salt Lake City, UT, USA, October 2024.





- [C.2] ★ A. J. Gaidis, **V. Atlidakis**, and V. P. Kemerlis. SysXCHG: Refining Privilege with Adaptive System Call Filters. In *Proceedings of the 30th ACM Conference on Computer and Communications Security (CCS)*, Copenhagen, Denmark, November 2023.



- [C.3] A. J. Gaidis, J. Moreira, K. Sun, A. Milburn, **V. Atlidakis**, and V. P. Kemerlis. FinelBT: Fine-grain Control-flow Enforcement with Indirect Branch Tracking. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, Hong Kong, HK, October 2023.

- [C.4] ★ N. Christou, Di Jin, **V. Atlidakis**, B. Ray, and V. P. Kemerlis. IvySyn: Automated Vulnerability Discovery for Deep Learning Frameworks. In *Proceedings of the 32nd USENIX Security Symposium (SEC)*, Anaheim, CA, USA, August 2023.



- [C.5] ✪ D. Jin, **V. Atlidakis**, and V. P. Kemerlis. EPF: Evil Packet Filter. In *Proceedings of the 29th USENIX Annual Technical Conference (USENIX ATC)*, Boston, MA, USA, July 2023.
- [C.6] ✪ T. Sotiropoulos, S. Chaliasos, **V. Atlidakis**, D. Mitropoulos, and D. Spinellis. Data-oriented differential testing of object-relational mapping systems. In *Proceedings of 43rd IEEE/ACM International Conference on Software Engineering (ICSE)*, Madrid, Spain, May 2021.
-  
- [C.7] **V. Atlidakis**, P. Godefroid, and M. Polishchuk. Checking Security Properties of Cloud Services REST APIs. In *Proceedings of the 13th IEEE International Conference on Software Testing, Verification, and Validation (ICST)*, Porto, Portugal, August 2020.
- [C.8] ✪ **V. Atlidakis**, P. Godefroid, and M. Polishchuk. RESTler: Stateful REST API Fuzzing. In *Proceedings of 42nd IEEE/ACM International Conference on Software Engineering (ICSE)*, Montreal, QC, Canada, May 2019.
- [C.9] ✪ M. Lecuyer, **V. Atlidakis**, R. Geambasu, D Hsu, and S. Jana. Certified Robustness to Adversarial Examples with Differential Privacy. In *Proceedings of the 40th IEEE Symposium on Security and Privacy*, San Francisco, CA, May 2019.
- [C.10] F. Tramer, **V. Atlidakis**, R. Geambasu, D. Hsu, J. P. Hubaux, M. Humbert, A. Juels, H. Lin. Fairtest: Discovering unwarranted associations in data-driven applications. In *Proceedings of the 2nd IEEE European Symposium on Security and Privacy (EuroS&P)*, Paris, France, April 2019.
- [C.11] ✪ **V. Atlidakis**, J. Andrus, R. Geambasu, D. Mitropoulos, and J. Nieh. POSIX abstractions in modern operating systems: The old, the new, and the missing. In *Proceedings of the 11th European Conference on Computer Systems (EuroSys)*, London, UK, April 2016.
- [C.12] **V. Atlidakis**, M. Roussopoulos, and A. Delis. Changing the unchoking policy for an enhanced BitTorrent. In *Proceedings of the 18th European Conference on Parallel Processing (EuroPar)*, Rhodes Island, Greece, August 2012.

### Journal Articles

- [J.1] **V. Atlidakis**, M. Roussopoulos, and A. Delis. EnhancedBit: Unleashing the potential of the unchoking policy in the BitTorrent protocol. *Journal of Parallel and Distributed Computing (JPDC)*, 74(1), December 2014.

### Preprints

- [PP.1] **V. Atlidakis**, R. Geambasu, P. Godefroid, M. Polishchuk, and B. Ray. Pythia: Grammar-Based Fuzzing of REST APIs with Coverage-guided Feedback and Learning-based Mutations. ArXiv preprint:2005.11498, May 2020.

### Magazine Articles

- [M.1] **V. Atlidakis**, J. Andrus, R. Geambasu, D. Mitropoulos, and J Nieh. POSIX Has Become Outdated. *USENIX ;login: Magazine*, 41(3), Fall 2016.