Kristopher Micinski

4-185 CST Building, Syracuse U, Syracuse, NY, 13201

📱 (940) 399-8924 • ✉ kkmicins@syr.edu • 🌐 kmicinski.com

Personal Information

Education

Michigan State University — **East Lansing, MI**
BS, Electrical and Computer Engineering, Honors: Summa Cum Laude — *2008–2011*

University of Maryland, College Park — **College Park, MD**
PhD, Computer Science — *2011–2017*
Dissertation: Interaction-Based Security Policies for Mobile Apps

Employment

Syracuse University
Tenure-Track Assistant Professor and Director: Cybersecurity MS — *Fall 2019–Present*

Haverford College
Visiting Assistant Professor — *2017–2019*

Online Profiles

WWW: https://kmicinski.com

Twitter: https://twitter.com/krismicinski

Scholar: https://scholar.google.com/citations?user=HpJLJWUAAAAJ

Federal Research Grants ($2.1M total)

National Science Foundation—PPoSS Large: Principles and Practices of Scalable Systems. Award #CCF-2316159. 7/18/2023–7/31/2028. Amount to Syracuse: $1M (of $5M total).

National Science Foundation—PPoSS Planning: Principles and Practices of Scalable Systems. Award #CCF-2217037. 10/1/2022–9/30/2024. Amount to Syracuse: $83.7k (of $250k total).

National Science Foundation—Student Travel Grant: Travel: Student Travel for the Programming Languages Mentoring Workshop (PLMW) at the International Conference on Functional Programming (ICFP). Award #CCF-2328059. 04/18/2023–04/18/2025. Amount to Syracuse: $45k.

US Defense and Research Projects Agency (DARPA, subcontract of Galois Inc.): Verified Security and Performance Enhancement of Large Legacy Software (V-SPELLS). Fall 2021–Spring 2024. Amount to Syracuse: $400k.

US Labratory for Telecommunication Sciences: Scaling Malware Analysis Pipelines. Fall 2021–Present. Amount to Syracuse: $600k (ongoing contract with LPS, $600k has been negotiated so far).

PhD Students

Naveen Ashok. Fall 2024–present. Working on Binary Analysis via AI

Neda Abdolrahimi. Spring 2024–present.

Chang Liu. Fall 2023–present. Planned graduation: Spring 2027.

Yihao Sun. Fall 2020–present. Planned graduation: Spring 2026. Proposing his dissertation in December, 2024.

Arash Sahebolamri (**graduated Spring 2023**). Thesis title: "Improving Logic Programming for Program Analysis."

Conference, Journal, and Workshop Papers

Significant Conference Papers (Refereed)

○ Yihao Sun, Sidharth Kumar, Thomas Gilray, and Kristopher Micinski. "Column-Oriented Datalog on the GPU". in: *Proceedings of the AAAI Conference on Artificial Intelligence* (2025). Acceptance rate: **23.4%**. https://arxiv.org/abs/2501.13051

○ Thomas Gilray, Arash Sahebolamri, Yihao Sun, Sowmith Kunapaneni, Sidharth Kumar, and Kristopher Micinski. "Datalog with First-Class Facts". In: *Proc. VLDB Endow.* 18.3 (Apr. 2025). Acceptance rate **not announced yet**, pp. 651–665. ISSN: 2150-8097. DOI: 10.14778/3712221.3712232. URL: https://doi.org/10.14778/3712221.3712232

○ Yihao Sun, Ahmedur Rahman Shovon, Thomas Gilray, Sidharth Kumar, and Kristopher Micinski. "Optimizing Datalog on the GPU". in: *ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (*ASPLOS '25*) (Mar. 2025). Acceptance rate: **20.5%**, (105 out of 510)

○ Rebecca Saul, Chang Liu, Noah Fleischmann, Richard Zak, Kristopher Micinski, Edward Raff, and James Holt. "Is Function Similarity Over-Engineered? Building a Benchmark". In: *Thirty-Eighth Annual Conference on Neural Information Processing Systems* (*NeurIPS '24 datasets and benchmarks track*) (2024)

○ Chang Liu, Rebecca Saul, Yihao Sun, Edward Raff, Maya Fuchs, Townsend Southard Pantano, James Holt, and Kristopher Micinski. "ASSEMBLAGE: Automatic Binary Dataset Construction for Machine Learning". In: *Thirty-Eighth Annual Conference on Neural Information Processing Systems* (*NeurIPS '24 datasets and benchmarks track*) (2024)

○ Arash Sahebolamri, Langston Barrett, Scott Moore, and Kristopher Micinski. "Bring Your Own Data Structures to Datalog". In: *The ACM's Conference on Object-Oriented Programming, Systems, Languages and Applications 2023* (*OOPSLA '23*) 7 (Oct. 2023). (Acceptance rate: **36%**, won **distinguished paper award** at OOPSLA '23). DOI: 10.1145/3622840. URL: https://doi.org/10.1145/3622840

○ Yihao Sun, Thomas Gilray, Sidharth Kumar, and Kristopher Micinski. "Communication-Avoiding Recursive Aggregation". In: *Proceedings of the IEEE Conference on Cluster Computing* (*IEEE Cluster '23*) Cluster (Oct. 2023). (Acceptance rate: **24%**)

○ Ahmedur Rahman Shovon, Thomas Gilray, Kristopher Micinski, and Sidharth Kumar. "Towards Iterative Relational Algebra on the GPU". in: *2023 USENIX Annual Technical Conference* (*USENIX ATC '23*). (Acceptance rate: **18%**). Boston, MA: USENIX Association, July 2023, pp. 1009–1016. ISBN: 978-1-939133-35-9. URL: https://www.usenix.org/conference/atc23/

`presentation/shovon`

o Ke Fan, Thomas Gilray, Valerio Pascucci, Xuan Huang, Kristopher Micinski, and Sidharth Kumar. "Optimizing the Bruck Algorithm for Non-Uniform All-to-All Communication". In: *Proceedings of the 31st International Symposium on High-Performance Parallel and Distributed Computing* (**HPDC '22**). HPDC '22. (Acceptance rate: 19%). Minneapolis, MN, USA: Association for Computing Machinery, 2022, pp. 172–184. ISBN: 9781450391993. DOI: 10.1145/3502181.3531468. URL: https://doi.org/10.1145/3502181.3531468

o Arash Sahebolamri, Thomas Gilray, and Kristopher Micinski. "Seamless Deductive Inference via Macros". In: *Proceedings of the 31st ACM SIGPLAN International Conference on Compiler Construction* (**CC '22**). (Acceptance rate: 32%). Seoul, South Korea: Association for Computing Machinery, 2022, pp. 77–88. ISBN: 9781450391832. DOI: 10.1145/3497776.3517779. URL: https://doi.org/10.1145/3497776.3517779

o Thomas Gilray, Sidharth Kumar, and Kristopher Micinski. "Compiling Data-Parallel Datalog". In: *Proceedings of the 30th ACM SIGPLAN International Conference on Compiler Construction* (**CC '21**). (Acceptance rate: 37%). Virtual, Republic of Korea: Association for Computing Machinery, 2021, pp. 23–35. ISBN: 9781450383257. DOI: 10.1145/3446804.3446855. URL: https://doi.org/10.1145/3446804.3446855

o Kristopher Micinski, David Darais, and Thomas Gilray. "Abstracting Faceted Execution". In: *2020 IEEE 33rd Computer Security Foundations Symposium* (**CSF '20**). (Acceptance rate: 25%). 2020, pp. 184–198. DOI: 10.1109/CSF49147.2020.00021

o Daniel Votipka, Seth M. Rabin, Kristopher Micinski, Jeffrey S. Foster, and Michelle M. Mazurek. "An observational investigation of reverse engineers' processes". English (US). in: *Proceedings of the 29th USENIX Security Symposium* (**USENIX '20**). Proceedings of the 29th USENIX Security Symposium. (Acceptance rate: 16.1%). USENIX Association, 2020, pp. 1875–1892

o Daniel Votipka, Seth M. Rabin, Kristopher Micinski, Thomas Gilray, Michelle L. Mazurek, and Jeffrey S. Foster. "User Comfort with Android Background Resource Accesses in Different Contexts". In: *Fourteenth Symposium on Usable Privacy and Security* (**SOUPS '18**). Baltimore, MD: USENIX Association, Aug. 2018, pp. 235–250. ISBN: 978-1-939133-10-6. URL: https://www.usenix.org/conference/soups2018/presentation/votipka

o Kristopher Micinski, Daniel Votipka, Rock Stevens, Nikolaos Kofinas, Jeffrey S. Foster, and Michelle L. Mazurek. "User Interactions and Permission Use on Android". In: *Conference on Human Factors in Computing Systems* (**CHI '17**). 2017. URL: http://cs.umd.edu/~micinski/chi-2017.pdf

o Kristopher Micinski, Jonathan Fetter-Degges, Jinseong Jeon, Jeffrey S. Foster, and Michael R. Clarkson. "Checking Interaction-Based Declassification Policies for Android Using Symbolic Execution". In: *European Symposium on Research in Computer Security* (**ESORICS '15**). Vol. 9327. Lecture Notes in Computer Science. Vienna, Austria, Sept. 2015, pp. 520–538. URL: http://www.cs.cornell.edu/~clarkson/papers/clarkson_clickrelease.pdf

o Michael R. Clarkson, Bernd Finkbeiner, Masoud Koleini, Kristopher Micinski, Markus N. Rabe, and César Sánchez. "Temporal Logics for Hyperproperties". English. In: *Principles of Security and Trust* (**POST '14**). Ed. by Martín Abadi and Steve Kremer. Vol. 8414. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014, pp. 265–284. URL: http://cs.umd.edu/~micinski/post14.pdf

Workshop Papers (Refereed)

○ Bruno Rucy Carneiro Alves de Lima, Merlin Kramer, Kalmer Apinis, and Kristopher Micinski. "Incremental Evaluation of Dynamic Datalog Programs as a Higher-order DBSP Program". In: *the 5th International Workshop on the Resurgence of Datalog in Academia and Industry* (**Datalog 2.0 '24**) (2024)

○ Ke Fan, Kristopher Micinski, Thomas Gilray, and Sidharth Kumar. "Exploring MPI Collective I/O and File-per-process I/O for Checkpointing a Logical Inference Task". In: *2021 IEEE International Parallel and Distributed Processing Symposium Workshops* (*IPDPSW '21*). 2021, pp. 965–972. DOI: 10.1109/IPDPSW52791.2021.00153

○ Davis Ross Silverman, Yihao Sun, Kristopher K. Micinski, and Thomas Gilray. "So You Want to Analyze Scheme Programs With Datalog?" In: *CoRR* abs/2107.12909 (2021). (Acceptance rate: 80%). arXiv: 2107.12909. URL: https://arxiv.org/abs/2107.12909

○ Yihao Sun, Jeffrey Ching, and Kristopher Micinski. "Declarative Demand-Driven Reverse Engineering". In: *Workshop on Binary Analysis Research* (*textbfBAR '21*) *at NDSS 2021.* (Acceptance rate: 60%). Feb. 2021. URL: https://bar2021.moyix.net/bar2021-preprint16.pdf

○ Kristopher Micinski, Thomas Gilray, Daniel Votipka, Jeffrey S. Foster, and Michelle L. Mazurek. *Symbolic Path Tracing to Find Android Permission-Use Triggers*. Feb. 2019. URL: https://www.ndss-symposium.org/wp-content/uploads/bar2019_83_Micinski_paper.pdf

○ Kristopher Micinski, Philip Phelps, and Jeffrey S. Foster. "An Empirical Study of Location Truncation on Android". In: *Mobile Security Technologies* (**MoST '13**). San Francisco, CA, May 2013. URL: http://www.cs.umd.edu/~jfoster/papers/most13.pdf

○ Jinseong Jeon, Kristopher Micinski, Jeffrey A. Vaughan, Ari Fogel, Nikhilesh Reddy, Jeffrey S. Foster, and Todd Millstein. "Dr. Android and Mr. Hide: Fine-grained Permissions in Android Applications". In: *ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices* (**SPSM '12**). Raleigh, NC, USA, Oct. 2012, pp. 3–14. URL: http://cs.umd.edu/~jfoster/papers/spsm12.pdf

Journal Articles (Refereed)

○ M. Taghizadeh, K. Micinski, S. Biswas, C. Ofria, and E. Torng. "Distributed Cooperative Caching in Social Wireless Networks". In: *IEEE Transactions on Mobile Computing* 12.6 (June 2013), pp. 1037–1053. ISSN: 1536-1233. DOI: 10.1109/TMC.2012.66

Tech Reports

○ Jinseong Jeon, Kristopher K. Micinski, Jeffrey A. Vaughan, Nikhilesh Reddy, Yixin Zhu, Jeffrey S. Foster, and Todd Millstein. *Dr. Android and Mr. Hide: Fine-grained security policies on unmodified Android*. Tech. rep. CS-TR-5006. Department of Computer Science, University of Maryland, College Park, Dec. 2011

○ Jinseong Jeon, Kristopher Micinski, and Jeffrey S. Foster. *SymDroid: Symbolic Execution for Dalvik Bytecode*. Tech. rep. CS-TR-5022. Department of Computer Science, University of Maryland, College Park, July 2012. URL: http://www.cs.umd.edu/~jfoster/papers/cs-tr-5022.pdf

○ Masoud Koleini, Michael R. Clarkson, and Kristopher K. Micinski. "A Temporal Logic of Security". In: *CoRR* abs/1306.5678 (2013). URL: http://arxiv.org/abs/1306.5678

○ Kristopher K. Micinski, Jonathan Fetter-Degges, Jinseong Jeon, Jeffrey S. Foster, and Michael R. Clarkson. "Checking Interaction-Based Declassification Policies for Android Using Sym-

bolic Execution". In: *CoRR* abs/1504.03711 (2015). URL: http://arxiv.org/abs/1504.03711

Book Chapters

Contributed Android-related material (approximately twenty pages) to the book "Operating Systems, Internals and Design Principles (8th Edition)." I coordinated with the main author, William Stallings, to integrate my material into his book.

Courses Taught

At Syracuse

CIS 352: Spring 2025, 100 students. (**Undergraduate**) **Principles of Programming Languages.** https://kmicinski.com/cis352-s25/. This is a second/third-year programming languages course for undergraduates. Topics include functional programming patterns, λ-calculus, structural type systems, and proof by natural deduction. The course is project-focused, with an emphasis on teaching students to reason about and debug their code.

CIS 700: Fall 2024, 8 students. (**Graduate**) **Modern Symbolic AI and Automated Reasoning.** https://kmicinski.com/cis700-f24/ Covers modern topics in symbolic AI, especially theorem-proving, SAT/SMT solving, first-order reasoning, logic programming, ontological reasoning, and with an application focus on reasoning about system security.

CIS 352: Fall 2024, 39 students. (**Undergraduate**) **Principles of Programming Languages.** https://kmicinski.com/cis352-f24/

CIS 352: Spring 2024, 90 students. (**Undergraduate**) **Principles of Programming Languages.** https://kmicinski.com/cis352-s24/

CIS 700: Fall 2023, 8 students. (**Graduate**) **Modern Symbolic AI and Automated Reasoning.** https://kmicinski.com/cis700-f23/

CIS 352: Fall 2023, 38 students. (**Undergraduate**) **Principles of Programming Languages.** https://kmicinski.com/cis352-f23/

CIS 352: Spring 2023, 94 students. (**Undergraduate**) **Principles of Programming Languages.** https://kmicinski.com/cis352-s23/

CIS 700: Fall 2022, 8 students. (**Graduate**) **Language-Based Security.** This course covered topics such as security type systems, model checking for security properties, information flow, and dynamic monitors.

CIS 352: Fall 2022, 36 students. (**Undergraduate**) **Principles of Programming Languages.** https://kmicinski.com/cis352-f22/ Lecture videos linked on website.

CIS 352: Spring 2022, 94 students. (**Undergraduate**) **Principles of Programming Languages.** https://kmicinski.com/cis352-s22/

CIS 400: Fall 2021, 30 students. (**Undergraduate**) **Compiler Design.** https://kmicinski.com/cis400-f21/ In this undergraduate compilers course, we place an emphasis on the compilation of functional languages via closures and continuation-passing style.

CIS 700: Fall 2021, 8 students. **Principles of Malware Analysis.** Graduate level seminar. This seminar covers modern research that targets using formal methods, symbolic AI, and machine-learning-based mechanisms for analyzing malicious code.

CIS 352: Spring 2021, 98 students. (**Undergraduate**) **Principles of Programming Languages.** https://kmicinski.com/cis352-s21/

CIS 700: Fall 2020, 3 students. (**Graduate**) **Formal Methods in Computer Security.** Graduate-level seminar. This course covers a broad array of formal security properties, such as relational properties (noninterference, information flow) and program analyses (taint analysis, abstract interpretation for reasoning about information flow).

CIS 352: Spring 2020, 95 students. (**Undergraduate**) **Principles of Programming Languages.** `https://kmicinski.com/cis400-f22/`

CIS 700: Fall 2019, 3 students. (**Graduate**) **Program Analysis.** Graduate-level seminar in abstract interpretation, lattices, and related topics.

At Haverford

CMSC 245: Spring 2019, 30 students. Course: Principles of Programming Languages.

CMSC 395: Fall 2018, 18 students. Mobile Apps for Social Change. A project-based course that teaches Android development by pairing students with nonprofits.

CMSC 107: Fall 2018, 30 students. (Honors) Introductory programming in Python. `http://kmicinski.com/cs107`

CMSC 311: Spring 2018, 25-35 students. Computer Security: Attacks and Defenses. `https://kmicinski.com/cybersecurity-course/`

CMSC 245: Fall 2017, 25 students. Course: Principles of Programming Languages.

At Maryland

CMSC 330: Summer 2015, 35 students. First course as instructor of record. Course: Organization of Programming Languages.

PC Membership, Conference Chairing

2025: Program Committee member, OOPSLA '26.

2025: Program Committee member, ICSE '26.

2025: Program Committee member, SIG-KDD '25 (Datasets Track).

2024: General Chair for the Scheme Workshop at ICFP '24. `https://icfp24.sigplan.org/home/scheme-2024`

2024: Program Committee member, AAAI '25. `https://aaai.org/conference/aaai/aaai-25/`

2024: Program Committee member, NeurIPS (datasets and benchmarks track).

2024: Program Committee member, Conference on Generative Programming: Concepts and Experiences (GPCE '24). `https://2024.splashcon.org/home/gpce-2024`

2023: Program Committee member, AAAI '24. `https://aaai.org/conference/aaai/aaai-24/`

2022–Present: Member: SIGPLAN-M mentor. SIGPLAN-M is a professional mentoring organization run by the ACM's special interest group on programming languages (SIGPLAN). I mentor roughly four students per year through SIGPLAN-M, primarily PhD students and post-docs.

2024: Program Committee for NDSS Workshop on Binary Analysis (BAR '24).

2024: Co-Chair (with Garrett Katz) of the AAAI "bridge" workshop at AAAI '24: "AP2S: Automated Program and Proof Synthesis Bridge Program." `https://garrettkatz.github.io/ap2s-bridge/`

2023: Senior chair, Programming Languages Mentoring Workshop (PLMW) at ICFP '23

`https://icfp23.sigplan.org/track/plmw-icfp-2023`. I also administer the three-year ($45k) NSF grant at Syracuse which supports PLMW at ICFP.

2023: Publicity chair for the ACM's 50th Conference on the Principles of Programming Languages (POPL '23)

2023: Co-chair (with Garrett Katz) of the AAAI "bridge" workshop "AP2S: Automated Program and Proof Synthesis Bridge Program." `https://garrettkatz.github.io/ap2s-bridge/`

2022: Program Committee for PLAS (The 17th Workshop on Programming Languages and Analysis for Security) `https://plas2022.github.io/`

2022: Program Committee for Scheme Workshop at ICFP 2022 `https://plas2022.github.io/`

2022: Junior Chair, Programming Languages Mentoring Workshop (PLMW) at ICFP `https://icfp22.sigplan.org/home/PLMW-ICFP-2022`. I ran the workshop in person and applied for the associated NSF grant.

2022: Program Committee Co-chair, NDSS workshop on Binary Analysis Research `https://www.ndss-symposium.org/ndss2022/cfp-bar-workshop/`.

2021: Program Committee, SADFE 2021: Systematic Approaches to Digital Forensic Engineering (SADFE) at 2021 IEEE S&P

2020: External Review Committee, OOPSLA 2021: Served as an external reviewer for OOPSLA '21

2019: General Chair, Scheme Workshop 2019 `https://gilray.org/scheme-2019/`

2017: Web chair for EUSEC '18: the 3rd European Conference on Usable Security.

2017: Member for poster review jury for SOUPS '18: the Symposium on Usable Security and Privacy.

Non-PhD Student Mentoring

Masters Students

Jeffrey Ching (**graduated summer 2021**). "Enhancing Usability of Malware Analysis Pipelines with Reverse Engineering"

Davis Silverman. Fall 2020–Fall 2023.

Chang Liu (Spring 2020–2022). Now PhD student.

Usha Nalabolu. (Fall 2022–2022). Collaborating on Slog.

Satyajeet Jha. (Fall 2022–2022). Collaborating on Slog.

Daniel Lugo (Spring 2021–Summer 2021). Collaborated to build Assemblage

Undergraduate Students

Jay Morrisson. Summer 2021. Working on scaling malware analysis pipeline generation

Chandler Todd. 2017. Working on improvements to the Soot binary rewriting framework to enable dynamic analysis of Android apps.

Skyler Ellenburg. 2017. Working on automatically scraping, archiving, and warehousing apps automatically for automatic analysis.

Linyi Chen. 2017. Working on string analysis of Android apps to detect social-media API usage.

Tosin Alliyu. 2017. Working on a user study of perceptions towards social-media API permissions and their integration with Android apps.

Austin Wan. 2017. Studying how the Abstracting Abstracting Machines approach can be scaled up to real-world analysis frameworks.

Becky Lytle. 2017. Applying abstract interpretation to check algorithmic fairness of machine learning applications.

Daniel Chen. Fall 2016. Applied dynamic analysis to Android apps to decide what kind of data they backed up to cloud storage.

Philip Phelps. Summer 2012 and 2013. Collaborated on work to implement and test location fuzzing for Android apps.

Rebecca Norton. Summer 2012. Collaborated on extending Redexer and doing work to make it publicly released.

Service

To the ECS department at Syracuse University

2024–Present: Director of the Cybersecurity MS program at Syracuse University. In this role, I oversee matters such as accreditation and long-term program direction.

2024–Present: Member of the Assessment and Accreditation Committee.

2024: Member of Syracuse University ECS assessment and accreditation committee.

2023: Member of Syracuse University ECS curriculum committee.

2021–Present: Writer/Grader: PhD qualifier exam questions on Programming Languages.

Service Prior to Syracuse

President of Graduate Student Executive Council
UMD Computer Science   *2012–2017*
As the president of the executive council, I organized social events and served as a representative to the department on behalf of graduate students. During my time I organized numerous events of varying scale each year. These included departmental picnics, happy hours, game nights, and others.

Webmaster, PLUM Lab website
Programming Languages (PLUM) Lab, UMD   *2013–Present*
I redid the PLUM group's website so that it was easier to modify as members and projects were reorganized. I currently maintain the site for the group.

Spring 2013: Organized Mid-Atlantic Programming Langauges Seminar (MAPLS).

Fall 2014: Student volunteer at the International Conference on Functional Programming (ICFP).

Fall 2011–2017: Subreviews for Jeffrey Foster, Michael Hicks, Michelle Mazurek, and Michael Clarkson. Reviewed papers from a variety of conferences including Oakland, PLDI, POPL, CSF, SOUPS, and more.

Fall 2017–Fall 2022: Active volunteer for CodedByKids Philadelphia: an organization that helps underrepresented students learn to program after school.

Selected Public Talks

September 2024: "High-Performance Logic Programming on Servers, Clusters, and GPUs."

Keynote talk at ICFP '24's MiniKanren workshop. Attendance of roughly 30 people.

January 2022: "Massively-Parallel Declarative Analytics." To Galois Inc. Attendance of roughly 35 people.

Summer 2020: "A vision for the future of analysis-assisted reverse engineering." To "Empire Hacking" group (based in NYC). Attendance of roughly 50 people.

Spring 2016: "Interaction-Based Security Policies for Mobile Apps." PhD Proposal Defense.

Fall 2016: "User Interactions and Permission Use on Android." Presented at the New Jersey Programming Languages Seminar.

Fall 2015: "Checking Interaction-Based Declassification Policies for Android Using Symbolic Execution." Presented at ESORICS 2015.

Spring 2013: "An Empirical Study of Location Truncation on Android." Presented at Mobile Security Technologies (MoST).

Fall 2012: "Dr. Android and Mr. Hide: Fine-grained Permissions in Android Applications." Talk given at George Washinton University, Fall 2012.

Software

Assemblage: Generates large-scale datasets for malware analysis pipelines. Currently builds approximately 30-50k Windows binaries per day running at Syracuse University Research Computing Cluster. Funded by Labratory for Telecommunication Sciences. `https://assemblage-dataset.net/`

Ascent: A macro-embedded declarative language in Rust. Outperforms the Soufflé Datalog engine on a single thread, and exhibits good parallel performance at higher thread counts. `https://github.com/s-arash/ascent`

GDLog: A state-of-the-art GPU-based Datalog engine which outperforms the state of the art by $50\times$. `https://github.com/harp-lab/gdlog`

VFLog: A columnar GPU-based Datalog engine which outperforms GDLog by $2.5\times$. `https://github.com/harp-lab/vflog`

Slog: Massively-Parallel Declarative Logic Programming language. Developed in collaboration with the High-Performance Automated reasoning and Programming (HARP) lab, funded by DARPA V-SPELLS and NSF PPoSS planning. `https://github.com/harp-lab/slog-lang1`

Redexer: Joint with Jinseong Jeon. A binary rewriter for Android. Redexer is publicly available at `https://github.com/plum-umd/redexer` and used by a variety of research groups to manipulate Android apps.

SymDroid: Joint with Jinseong Jeon, A symbolic executor for Android bytecode.