1 Contributions to Research and Development

1.1 Articles Published or Accepted in Peer-Reviewed Journals

Findlay, W., Somayaji, A. B., and Barrera, D. (2020) bpfbox: Simple Precise Process Confinement with eBPF. Proceedings of the 2020 ACM Cloud Computing Security Workshop (CCSW'2020). (Master's work)

1.4 Non-Peer-Reviewed Contributions

Findlay, W. (2021) A Practical, Flexible, and Lightweight Confinement Framework in eBPF. MCS thesis. (Master's work)
Findlay, W., Barrera, D., and Somayaji, A. B. (2021) BPFContain: Fixing the Soft Underbelly of Container Security. Archival pre-print, pending submission to Usenix Security 2022. (Master's work)
Findlay, W. (2020) Host-Based Anomaly Detection with Extended BPF. Honours thesis. Carleton University. (Undergraduate work)

2 Most Significant Contributions to Research and Development

BPFBox: Simple Precise Process Confinement with eBPF. This peer-reviewed research paper presents BPFBox[1], the first process confinement mechanism written in Extended Berkeley Packet Filter (eBPF). Process confinement is critical for restricting access to security-sensitive resources on our computers and reducing the attack surface for potential security exploits. BPFBox's advantages over existing process confinement solutions include a simple yet expressive policy language and the ability to express and enforce policy across userspace and kernelspace boundaries, something which no existing process confinement mechanism can do. Furthermore, experimental data presented in the paper shows that BPFBox's performance is competitive with (and in some cases better than) the most popular process confinement mechanisms in Linux.

In this work, I designed and implemented the BPFBox research prototype, including the policy language and enforcement engine. As the first author of the research paper, I conducted and presented the results of all the benchmarks and experiments presented in the paper, described all of the project's technical details, and wrote significant portions of the background material. My co-authors, Dr. Anil Somayaji and Dr. David Barrera, helped with the positioning of the work, writing up portions of the background material, and selecting the appropriate venue for publication.

Our publication venue, the 2020 ACM Cloud Computing Security Workshop (CCSW)—a part of ACM SIGSAC—is a top security workshop in cloud computing and presents an ideal target audience for our experimental work in novel process confinement mechanisms. Submissions to CCSW are competitive, with only a 30% acceptance rate (12 papers out of 40 submissions). This paper was presented in November 2020 at the workshop. I was also invited to give a follow-up presentation at IBM research in December 2020.

BPFContain: Fixing the Soft Underbelly of Container Security. This paper presents the design, implementation, and evaluation BPFContain[2], an extension on top of the original BPFBox design with container security in mind. In particular, we develop a method for incorporating container semantics into kernelspace policy enforcement, using eBPF programs to trace the lifecycle of Linux containers. This approach yields a surprisingly effective policy enforcement mechanism that simultaneously simplifies container confinement policy and improves overall security.

Unlike individual processes, Linux containers group a set of processes and related resources together, defining a semantic relationship between them and establishing a clear boundary between resources within

[1] BPFBox is free and open-source software, available under the GPLv2 license at https://github.com/willfindlay/bpfbox
[2] BPFContain is free and open-source software, available under the GPLv2 license at https://github.com/willfindlay/bpfbox-rs.

and without the container. By tracing the container lifecycle and incorporating these relationships into the policy engine, BPFContain cam enforce container-level policy at a significantly finer granularity than would be possible with existing techniques. Moreover, the implicit security boundary around the container obviates the need to specify access to resources which only exist within the context of the container. This property radically simplifies BPFContain policies compared to traditional approaches.

In this work, I designed and implemented the BPFContain research prototype as an extension on top of my original BPFBox design. As the first author, I conducted all of the experiments, wrote up all the technical details of the implementation. My co-author, Dr. Anil Somayaji and Dr. David Barrera helped with positioning, provided feedback on early drafts, and contributed editorial changes in various sections. This work has been made available as a pre-print on the arXiv paper database, and we plan to submit it for publication in Usenix Security 2022.

3 Applicant's Statement

Research Experience. My research in operating system security has afforded me a strong technical knowledge of the underlying abstractions, security mechanisms, data structures, and algorithms that power our computer systems. This technical understanding has led me to question whether the security mechanisms that are currently in place in most commodity operating systems are sufficient to protect our devices against more sophisticated attacks. My experiences with using operating system observability technologies to build both anomaly detection systems and process confinement mechanisms has motivated me to consider whether it may be possible to bridge the gap between adaptive security approaches and traditionally static approaches like process confinement, a notion that has fundamentally informed my future research directions.

Relevant Activities. I have been a teaching assistant for the COMP3000 (Operating Systems) and COMP4000 (Distributed Operating Systems) course at Carleton University for three years. During this time, I have provided guidance for upper year computer science students and graduate computer science students and taken an active role in the development of course material and other administrative tasks. In particular, I was involved in the design of tutorials and assignments for both COMP3000 and COMP4000. These focused on key aspects of operating system design, implementation, and security. The strong interplay between my research and teaching activities has been quintessential in fostering the growth of my academic career. As a direct result of my passion for operating system security, I have been nominated for three consecutive Outstanding Teaching Assistant awards in the 2018–2019, 2019–2020, and 2020–2021 academic years.

In the final year of my undergraduate studies, I was selected to participate in the Carleton School of Computer Science's Accelerated Master's Program. This option is only offered to top undergraduate students in Carleton's Computer Science program. It allows the student to take two graduate-level courses in the final year of their degree. This opportunity has provided me with a strong background in research early on in my graduate school career and enabled me to complete my Master of Computer Science degree in one year instead of two. After successfully defending my Master's thesis, I was nominated by my committee for a senate medal for academic achievement.

As a member of the Carleton Internet Security Lab (CISL) at Carleton University and a close collaborator with its sister lab, the Carleton Computer Security Lab (CCSL), I have access to the resources, knowledge, and guidance of one of the largest computer security research groups in Canada. This has proved to be a significant advantage in the development and dissemination of my research thus far and will continue to be an invaluable resource in my future research endeavours.