Introduction Canadians today are more reliant on web and cloud-based technologies than ever before. But can we really trust the underlying systems that host our web services and store our data? Misconfiguration of software, zero-day vulnerabilities, unpatched systems, and social engineering are just a few examples of serious factors that impact the security, stability, and reliability of computer systems and services that power our society. Securing these systems is a constant back-and-forth battle between attackers and defenders—a high-stakes game of cat-and-mouse that often has devastating societal and economic consequences, as evidenced by recent attacks on Canadians' security and privacy [3, 4, 11]. These attacks often disproportionately affect vulnerable Canadian minority populations by targeting our social security net, particularly in light of the unprecedented circumstances brought about by the COVID-19 pandemic.

I hypothesize that making measurable and long-lasting security improvements requires constructing adaptive defences that are robust in the presence of attacker innovation. Adaptive defences are those that can respond to novel attacks in real time, require limited or no user intervention to function, and/or enhance the transparency and usability of policy definition and enforcement mechanisms. Existing mechanisms for defining and enforcing security policy do not conform to these specifications; attackers are frequently able to break or circumvent them entirely. Thus, the typical pattern of attackers finding and exploiting large-scale vulnerabilities, while defenders work to patch vulnerabilities and mitigate damage, is perpetuated. As these defences are patched, they grow increasingly complex and convoluted, and so become less accessible to the vulnerable, non-expert populations they are designed to protect.

Objectives In my research, I will use operating system observability and extensibility mechanisms to construct adaptive defences that can protect our computers from a variety of threats and continue to function in the presence of attacker innovation. Computer operating systems provide the necessary hardware abstractions required to support the development of user-facing applications and mediate access to security-sensitive system resources by exposing an *application binary interface* with underlying protection mechanisms. My work will improve operating system security by enhancing the existing access control, isolation, and policy enforcement mechanisms that are critical to the security and stability of our applications, particularly those running in the cloud. Introducing adaptive functionality to these underlying mechanisms will provide an effective and robust basis for securing both the operating system itself and the user-facing applications that run on top of it.

Operating system observability tools are a natural choice for implementing adaptive security mechanisms, since they permit fine-grained system introspection capabilities across multiple layers of abstraction. This provides the necessary framework for developing security solutions that can incorporate multiple aspects of system state and behaviour into their design. *Extended Berkeley Packet Filter* (eBPF) [8], an emergent observability technology in the Linux kernel, is an increasingly prominent example of such a framework. Since its 2014 introduction into the kernel, eBPF has seen rapid development and a steady-growing feature set, allowing creation of complex system services and observation of nearly any aspect of system behaviour, across user and kernelspace boundaries.

Despite its widespread adoption in performance monitoring [10, 19] and networking [1, 5, 13, 20], the security community has only recently started to leverage eBPF for the development of security mechanisms, and much of my work thus far focuses on filling this gap in the security literature. My most recent peer-reviewed publication (co-authored with Somayaji and Barrera) presents the design and implementation of bpfbox[1] [7], the first eBPF-based process confinement mechanism. By leveraging eBPF's safety and flexibility, we were able to rapidly prototype a process confinement mechanism that can enforce policy across userspace and kernelspace boundaries. I plan to continue exploring the development of bpfbox as part of my Master's thesis, to be completed in early 2021. Prior to bpfbox, I designed and implemented ebpH[2], a host-based anomaly detection system written in eBPF, for my honours thesis [6]. A successor

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

[2] ebpH is free and open source software, available under the GPLv2 license at https://github.com/willfindlay/ebph

to my advisor's earlier work, pH [9, 17, 18], ebpH uses eBPF programs to monitor system call sequences in running programs and detect anomalies. We intend to submit a paper that evaluates ebpH's ability to respond to modern attacks to a top security venue (likely USENIX Security) in the near future.

Proposed Research Project Going forward, I propose to design novel operating system security solutions in eBPF. Expanding upon my initial work in eBPF-based process confinement mechanisms [7] and system call sequence analysis [6], I plan to design and implement a new research prototype capable of automatically generating policy language that can be expressed both semantically and at the sequence level. Sequences of observed system events could be combined into semantic policy rules. By providing an interface to assign semantically understandable names to common sequences, we can obtain an auditable, dynamic policy language that adjusts itself to meet the needs of the target environment. Although the generated policy language would be readable and writable by human users, it would be equally reasonable for the policy files themselves to be automatically generated per-application. Following this same pattern, it may be possible to use eBPF to incorporate other aspects of system behaviour into the generated policy, such as memory allocations, network traffic, and per-process stack traces. This work would improve on existing process confinement mechanisms in several ways:

1. Existing process confinement mechanisms lack transparency, which fundamentally undermines usability [14]. While techniques do exist for the automatic generation of policy files [2, 12, 15, 16], they lack integration with the underlying policy enforcement engine, requiring technical knowledge and effort on the part of the user. The proposed research prototype would completely hide the details of automatic policy generation from the user through strong integration with both the policy language itself and the enforcement engine.
2. Traditional policy languages for process confinement are tightly coupled with the underlying operating system interfaces which they are designed to mediate. These interfaces are often low level and difficult to understand without sufficient technical expertise [14]. The key insight behind this research is that sequences of operations on these interfaces can often be semantically understood without necessarily knowing the underlying details. This is analogous to understanding what a computer program *does* without knowing the details of how it is *coded*.
3. Existing process confinement mechanisms require constant patching as new vulnerabilities are discovered. This patching introduces further complexity into their policy languages and often requires existing policy files to be updated. An adaptive process confinement solution will be able to respond to attacker innovation without requiring as much manual intervention or added complexity.

Early design and implementation work will be an iterative process. Striking the right balance between expressiveness, terseness, and auditability of generated policy languages is a non-trivial task and will require careful tuning of design parameters, particularly considering the introduction of other behavioural data beyond system call sequences. Evaluation of such prototypes should consider both the usability and security aspects of the policy language, necessitating both rigorous security analysis and usability studies. To reflect the impact of security concerns on non-expert users and vulnerable populations, diversity considerations will be taken into account when selecting populations to participate in user studies. Security of generated policy languages will be tested by mounting a variety of known attacks against both manually configured and automatically generated policy files. These metrics should establish a strong baseline for comparison against existing process confinement mechanisms.

Conclusion Coming out of this research, we will have an adaptive process confinement solution that will provide increased usability and transparency over existing mechanisms while improving security by shaping the policy language according to the behaviour of the underlying system. This should have a profound impact on the security of the web and cloud-based services which provide the technical infrastructure for many aspects of Canadian society.

References

[1] Z. Ahmed, M. H. Alizai, and A. A. Syed, "InKeV: In-Kernel Distributed Network Virtualization for DCN," in *ACM SIGCOMM Computer Communication Review*, vol. 46, 2018, pp. 1–6. DOI: 10.1145/3243157.3243161.

[2] AppArmor authors, *aa-easyprof*, Linux user's manual. [Online]. Available: https://manpages.ubuntu.com/manpages/precise/man8/aa-easyprof.8.html.

[3] "Canadian accountants' association suffers cyberattack; data of nearly 330K affected," *The Canadian Press via Global News*, Jun. 2020, Anonymous author. [Online]. Available: https://globalnews.ca/news/7025862/cpa-canada-accountants-cyberattack (visited on 09/26/2020).

[4] T. Daigle, "Hackers were paid ransom after attack on Canadian insurance firm, court documents reveal," *CBC News*, Jan. 2020. [Online]. Available: https://www.cbc.ca/news/technology/unnamed-insurance-company-cyberattack-1.5445326 (visited on 09/26/2020).

[5] N. S. Dasineni, N. Shirokov, and R. Dasineni, "Open-sourcing Katran, a scalable network load balancer," *Facebook Engineering*, Nov. 2018. [Online]. Available: https://engineering.fb.com/open-source/open-sourcing-katran-a-scalable-network-load-balancer/.

[6] W. Findlay, "Host-Based Anomaly Detection with Extended BPF," Honours Thesis, Carleton University, Apr. 2020. [Online]. Available: https://williamfindlay.com/written/thesis.pdf.

[7] W. Findlay, A. B. Somayaji, and D. Barrera, "bpfbox: Simple Precise Process Confinement with eBPF," in *Proceedings of the 2020 ACM Cloud Computing Security Workshop (CCSW'2020)*, Nov. 2020. DOI: 10.1145/3411495.3421358.

[8] M. Fleming, "A thorough introduction to eBPF," *LWN.net*, Dec. 2017. [Online]. Available: https://lwn.net/Articles/740157 (visited on 09/26/2020).

[9] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A Sense of Self for Unix Processes," in *Proceedings 1996 IEEE Symposium on Security and Privacy*, May 1996, pp. 120–128. DOI: 10.1109/SECPRI.1996.502675.

[10] B. Gregg, *BPF Performance Tools*. Addison-Wesley Professional, 2019, ISBN: 0-13-655482-2.

[11] R. P. Jones, "Cyberattacks targeting CRA, Canadians' COVID-19 benefits have been brought under control: officials," *CBC News*, Aug. 2020. [Online]. Available: https://www.cbc.ca/news/politics/cra-gckey-cyberattack-1.5689106 (visited on 09/26/2020).

[12] K. MacMillan, "Madison: A new approach to policy generation," in *SELinux Symposium*, vol. 7, 2007. [Online]. Available: http://selinuxsymposium.org/2007/papers/08-polgen.pdf.

[13] S. Miano *et al.*, "Creating Complex Network Services with eBPF: Experience and Lessons Learned," in *2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR)*, 2018. DOI: 10.1109/hpsr.2018.8850758.

[14] Z. C. Schreuders, T. J. McGill, and C. Payne, "Towards Usable Application-Oriented Access Controls," in *International Journal of Information Security and Privacy*, vol. 6, 2012, pp. 57–76. DOI: 10.4018/jisp.2012010104.

[15] J. R. Smith, Y. Nakamura, and D. Walsh, *audit2allow*, Linux user's manual. [Online]. Available: http://linux.die.net/man/1/audit2allow.

[16] B. T. Sniffen, D. R. Harris, and J. D. Ramsdell, "Guided policy generation for application authors," in *SELinux Symposium*, 2006. [Online]. Available: http://gelit.ch/td/SELinux/Publications/Mitre_Tools.pdf.

[17] A. B. Somayaji, "Operating System Stability and Security through Process Homeostasis," Ph.D. dissertation, University of New Mexico, 2002. [Online]. Available: https://people.scs.carleton.ca/~soma/pubs/soma-diss.pdf.

[18] A. B. Somayaji and H. Inoue, "Lookahead Pairs and Full Sequences: A Tale of Two Anomaly Detection Methods," in *Proceedings of the 2nd Annual Symposium on Information Assurance Academic track of the 10th Annual 2007 NYS Cyber Security Conference*, NYS Cyber Security Conference, 2007, pp. 9–19. [Online]. Available: http://people.scs.carleton.ca/~soma/pubs/inoue-albany2007.pdf.

[19] A. Starovoitov, *Bpf at facebook*, Technical talk, Sep. 2019. [Online]. Available: https://kernel-recipes.org/en/2019/talks/bpf-at-facebook/.

[20] K. Suo, Y. Zhao, W. Chen, and J. Rao, "Efficient and flexible packet tracing for virtualized networks using eBPF," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018. DOI: 10.1109/infcomw.2018.8406849.

I. Contributions to Research and Development

a. 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). To appear. 12 page submission accepted August 19^{th}, 2020. (Master's work)

d. Non-Peer-Reviewed Contributions

Findlay, W. (2020) bpfbox: Simple Precise Process Confinement with eBPF. Conference presentation, to be presented at the 2020 ACM Cloud Computing Security Workshop (CCSW'2020). (Master's work)

Findlay, W. (2020) Host-Based Anomaly Detection with Extended BPF. Honours thesis. Carleton University. (Undergraduate work)

Findlay, W. (2020) Host-Based Anomaly Detection with Extended BPF. Technical talk presented in seminar series at the Carleton Computer and Internet Security Lab, Carleton University. (Undergraduate work)

II. Most Significant Contributions to Research and Development

bpfbox: Simple Precise Process Confinement with eBPF This peer-reviewed research paper presents bpfbox[3], 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. 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 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 technical details of the project, and wrote significant portions of the background material. My co-authors, Dr. Anil Somayaji and Dr. David Barrera, helped with 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 excellent target audience for our experimental work in novel process confinement mechanisms. Submissions to CCSW are quite competitive, with only a 30% acceptance rate (12 papers out of 40 submissions). I will present the paper in November 2020 at the workshop.

Host-Based Anomaly Detection with Extended BPF In my undergraduate honours thesis, I presented the design and implementation of ebpH[4], a host-based anomaly detection system written in eBPF. As a successor to the original pH anomaly detection system written by Dr. Anil Somayaji in 2002, ebpH is significant to the fields of operating system security and intrusion detection as the first modern implementation of system call sequence analysis for anomaly detection.

[3] bpfbox is free and open source software, available under the GPLv2 license at https://github.com/willfindlay/bpfbox

[4] ebpH is free and open source software, available under the GPLv2 license at https://github.com/willfindlay/ebph

As the sole author of this work, I designed and implemented the research prototype, conducted experiments to test ebpH's performance and functionality, and wrote up the technical details and challenges associated with ebpH's implementation. My undergraduate thesis advisor, Dr. Anil Somayaji, provided guidance where necessary and his original work on the pH anomaly detection system served as the basis for parts of my design.

While undergraduate theses are not published, an abstract of my work was published in the Carleton Honours Project and Thesis repository. We intend to submit a paper covering a later version of ebpH to a top security conference later this year. I presented ebpH at a technical talk given to members of the CISL and CCSL research groups at Carleton University in April 2020.

III. 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 course at Carleton University for two and a half years. During this time, I have provided guidance for third year computer science students, run weekly lab sessions to introduce concepts relating to operating systems, proctored examinations, and graded assignments and examinations. More recently, I have taken an active role in the development of course material and other administrative tasks; in particular, I designed all of the COMP3000 tutorial labs for the second half of the course in the Winter 2020 semester. Many of these labs will see continued use in the Fall 2020 semester. I was also nominated for two consecutive Outstanding Teaching Assistant awards in the 2018-2019 and 2019-2020 academic years. There is strong interplay between my teaching activities in COMP3000 and the directions I have taken in my area of research, operating system security; my knowledge of operating systems has been a driving factor in my research and I have attempted to bring aspects of my security background into my teaching.

In my 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 and allows the student to take two graduate-level courses in the final year of their degree. This has provided me with a strong background in research early on in my graduate school career.

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.