Proposal for MSE Capstone Project

Project Title: A More Flexible Security Policy for SimpleFlow

Student Name: Daniel Weninger

Project Sponsor: Dr. W. Michael Petullo

Faculty Advisor: Dr. W. Michael Petullo

Date of Submission: April 15, 2022

A More Flexible Security Policy for SimpleFlow

Objective

The objective of this capstone is to enhance the pre-existing project SimpleFlow, an information-flow-based access control system. Enhancements will include porting to a recent Linux kernel that allows for LSM stacking, and developing a new policy compiler for SimpleFlow, which will allow for a more parameterized and arbitrary security policy that is similar to SELinux.

Background

SimpleFlow was created to go beyond most access control systems, which immediately deny all illicit actions on a system. While effective, traditional access controls are not able to monitor the activity of the infiltrator after the initial breech in the file. SimpleFlow takes a different approach so that it is able to closely track and gather information about attacks, acting as an information-flow-based access control system that works on top of the LSM interface. Rather than disallowing read or write access to certain files, as many access control systems do, SimpleFlow allows this access, and it notes if any confidential data was involved. If so, the context of such actions passes through the system as information flows, and ultimately, SimpleFlow can prevent exfiltration attempts based on the information. Similarly, any process that is not whitelisted and is performed by a user, further taints and follows the user's subsequent actions. Therefore, an attacker's intent is captured in the system, whereas without SimpleFlow, an attacker can simply make excuses for such illicit activity. The research prototype SimpleFlow is limited in the way that it is not able to further specify the security policy of a file beyond being confidential and non-confidential, and it also does not currently support LSM stacking which is the ability to chain multiple Linux security modules.

When put to practical use during the 2016 Cyber-Defense Exercise, the SimpleFlow program proved to perform well with a small overhead. While the practical exercise did not make use of all of SimpleFlow's functionality, it was able to successfully use SimpleFlow's network filter which made it easy to observe any attempt of exfiltrating data. Within this exercise and other testing, SimpleFlow confounded experienced attackers who were surprised to find their exfiltration attempts fail.

Current Project

SimpleFlow is a Linux kernel modification that is able to track the flow of information through a system, raising the possibility of preventing malicious activity while allowing a richer set of benign activity. It achieves this by marking specific, potentially sensitive files as confidential. SimpleFlow then allows and mediates what would be otherwise malicious activities up to the point of extracting the sensitive data from the network. Before transmitting this information, any process that a user performs to try to extract the sensitively marked data becomes tainted, and any packets sent by the user are labeled as such by introducing an

RFC3514 evil bit. This bit is easily visible within packets that are being sent and therefore, are easy to track and block.

Regardless of the attack that is being performed, SimpleFlow aims to ultimately protect confidential files from being exfiltrated and study how the attacker attempts to do so. The system administrator is able to set up a whitelist of programs that are trusted by SimpleFlow as well as the confidentiality of a file. This means that if there is a process used that is not whitelisted, the action gets marked as tainted and any subsequent actions do as well.

The following are enhancements that are to be added to the SimpleFlow project as a part of this capstone:

- The student will develop and write a policy compiler for the successor to SimpleFlow, which will allow for arbitrary security policies. This will resemble the tool that compiles SELinux policies that are then loaded into the kernel. The student will design both the policy language and the binary format.
- The student will work with their faculty advisor to port SimpleFlow to a recent Linux kernel to allow for LSM stacking.
- The student will work with their faculty advisor to revise SimpleFlow to support policies beyond confidential and non-confidential.

As an example of a richer security policy, SimpleFlow could potentially support US security clearance levels. Within these policies are three primary levels of classification: confidential, secret, and top secret. Having access to a file within the top-secret level does not give one access to the rest of the files at this level, nor any of the files below this level. Instead, each of these categories requires further qualifications, and in some cases, proof that the individual needs to obtain information from the file. There are additional categories as well that require further clearance and can be stacked onto the three already defined categories. For example, a file may be marked as Secret (Code Word) at which point only a person who has clearance to both Secret files and (Code Word) files may be allowed to view it. Adding a policy such as this to SimpleFlow will allow for more malleable security, allow better control over which processes can interact with which files, and gives SimpleFlow a more expansive and updated security approach from the one-bit policy that it currently has.

Challenges

The following are some of the challenges for this project:

- Designing a new policy for SimpleFlow will be a challenge as the student will need to design both the policy language and the binary format. This may require a refactoring of parts of the SimpleFlow system.
- Porting SimpleFlow to a more recent Linux kernel will pose a challenge because kernel work in general is challenging.

Project Schedule

Phase	From	To	Credits
Develop requirements document and problem analysis	Sept 5, 2022	Oct 15, 2022	3
Port SimpleFlow to recent Linux kernel	Oct 16, 2022	Dec 21, 2022	3
Develop policy compiler for SimpleFlow	Jan 23, 2023	Feb 28, 2023	3
Refine and test	Mar 1, 2023	Mar 31, 2023	2
Deomonstration and project report	Apr 1, 2023	Apr 30, 2023	1

Resources

The student will use his personal computer to complete the project. The project sponsor will provide the SimpleFlow source program to be updated.