A Framework For Integrating Security Services into Software-Defined Networks

Seungwon Shin[1], Phillip Porras[2], Vinod Yegneswaran[2], Guofei Gu[1]

(1) Dept. of Electrical and Computer Engineering
(1) Texas A&M University
College Station, TX 77840
seungwon.shin@neo.tamu.edu
guofei@cse.tamu.edu

(2) Computer Science Laboratory
(2) SRI International
Menlo Park, CA 94025
{porras,vinod}@csl.sri.com

Openflow may in time prove to be one of the more impactful technologies to drive a variety of innovations in network security. It could offer a dramatic simplification to the way we design and integrate complex network security applications into large networks. In particular, OpenFlow offers researchers with an unprecedented singular point of control over network flow routing decisions across the data planes of all OF-enabled network components. Using OpenFlow, security services can implement far more complex logic than simply halting or forwarding a flow. Such applications can incorporate stateful flow rule production logic to implement complex quarantine procedures, or dynamic connection migration functions that can redirect malicious network flows in ways not easily perceived by the attacker. Flow-based security detection algorithms can also be redesigned as OF *security apps*, but implemented more concisely and deployed more efficiently. However, to date there remains a stark paucity of compelling OpenFlow security applications.

Our research team is actively engaged in several projects to help accelerate new research in OpenFlow-enabled network defense. Our latest research result [10] introduces *FRESCO*, an OpenFlow security application development framework that facilitates the rapid design and modular composition of OF-enabled detection and mitigation modules. Inspired by the Click router architecture [6] and Click's modular scripting interface, FRESCO abstracts key data access and security directive controls, fostering a more rapid and collaborative environment for security-focused developers. FRESCO's scripting language enables the linking of modules through data sharing and event triggering. Further, FRESCO provides an API that can facilitate responsive flow rule production decisions using information produced from legacy DPI-based security applications (such as IDS or anti-malware applications).

FRESCO exports a scripting API that enables security practitioners to code security monitoring and threat detection logic as modular libraries. These modular libraries represent the elementary processing units in FRESCO, and may be shared and linked together to emulate complex network defense applications. FRESCO currently includes a library of 16 commonly reusable modules, which we intend to expand over time. Ideally, more sophisticated security modules can be built by connecting basic FRESCO modules. Each FRESCO module includes five interfaces: (i) input, (ii) output, (iii) event,(iv) parameter, and (v) action. By simply assigning values to each interface and connecting necessary modules, a FRESCO developer can replicate a range of essential security functions, such as firewalls, scan detectors, attack deflectors, or IDS detection logic.

To date, we have used FRESCO to implement a rich collection of OpenFlow security mitigation directives (*FRESCO* modules), as well as complex threat detection applications. We have demonstrated OpenFlow-enabled applications such as malicious network scan detectors [5, 9, 4], P2P Malware Locator [12], and BotMiner [2]. In addition, FRESCO exposes an API that enables it to easily support threat responses to alerts produced from legacy DPI-based security applications, such as Snort [11] or BotHunter [3]. The security community has developed a rich set of network-based threat monitoring services. Using FRESCO, we made available several video demonstrations [1] of how these security services can be utilized to drive complex SDN-enabled threat mitigation logic. Further, our *FRESCO* implementations demonstrate over 90% reduction in lines of code when compared to standard implementations and recently published OpenFlow implementations [7].

However, no threat mitigation service can be relied upon when the flow policies decision that it produces cannot be

reliably enforced by the OpenFlow stack. The possibility of multiple (custom and third-party) OpenFlow applications running above an OpenFlow controller introduces a unique policy enforcement challenge: since different applications may insert different control policies *dynamically*, how does the OF controller guarantee that they are not in conflict with each other? Thus, a second critical challenge that our group is investigating is how to provide continued enforcement of potentially conflicting flow constraints imposed by dynamic OF applications.

Our research team has designed and prototyped a security enforcement kernel (SEK), which we have integrated directly into an OF-Controller [8] on which FRESCO is hosted. The SEK offers several important features upon which all OF-enabled security applications may fundamentally rely upon to ensure that their flow rules are prioritized and enforced over competing flow rules produce by non-security critical applications. The SEK introduces a trust model that allows FRESCO applications to digitally sign each candidate flow rule, thereby providing a basis for authenticated prioritization of these rules. The SEK incorporates an inline rule conflict analysis algorithm, called *alias set rule reduction*, which detects flow rule conflicts, including those that arise through set actions that may implement virtual tunnels. Finally, the SEK applies a hierarchical authority model that enables candidate rules to override existing flow rule policies from subordinate applications in the role hierarchy.

The SEK enables *FRESCO* to detect and resolve flow rule contradictions in real time, and is robust even in cases where an adversarial OF application attempts to strategically insert flow rules that would otherwise circumvent flow constraints imposed by *FRESCO* applications. The SEK mediates all flow constraints produced by *FRESCO* scripts, storing them into its internal security constraints table, and forwarding these constraints as flow rules to switches for immediate intervention. The SEK also enforces its resident security constraints against all new flow rules produced by NOX's registered OpenFlow applications. These flow rules are distributed on to switches when no contradiction is detected. Otherwise, the OpenFlow application is notified that the rule is rejected.

We have built an initial reference implementation of an embedded security mediation service in the NOX OF-controller, and with this prototype we continue to explore conflict resolution challenges and demonstrate several advanced security mitigation strategies [1]. We are now focused on the development of a Security-Enhanced Floodlight (SE-Floodlight), which will provide the first public release of a security enforcement kernel for an OpenFlow controller. Combined, we believe that SE-Floodlight and FRESCO offer a powerful new reference framework for enabling the INFOSEC research community to rapidly prototype and field innovative security applications into the rapidly evolving world of software-defined networks.

References

[1] OpenFlowSec. `http://www.openflowsec.org`.

[2] G. Gu, R. Perdisci, J. Zhang, and W. Lee. BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection. In *Proceedings of USENIX Security Symposium (Security'08)*, 2008.

[3] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee. BotHunter: Detecting malware infection through ids-driven dialog correlation. In *Proceedings of the 16th USENIX Security Symposium (Security'07)*, August 2007.

[4] J. Jung, R. Milito, and V. Paxson. On the Adaptive Real-time Detection of Fast Propagating Network Worms. In *Proceedings of Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA)*, 2007.

[5] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan. Fast Portscan Detection Using Sequential Hypothesis Testing. In *Proceedings of IEEE Symposium on Security and Privacy*, 2004.

[6] E. Kohler, R. Morris, B. Chen, J. Jannotti, and F. Kaashoek. The Click Modular Router. *ACM Transactions on Computer Systems*, August 2000.

[7] S. A. Mehdi, J. Khalid, and S. A. Khayam. Revisiting Traffic Anomaly Detection Using Software Defined Networking. In *Proceedings of Recent Advances in Intrusion Detection*, 2011.

[8] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu. A Security Enforcement Kernel for OpenFlow Networks. In *Proceedings ACM SIGCOMM Workshops on Hot Topics in Software Defined Networking (HotSDN)*, August 2012.

[9] V. Sekar, Y. Xie, M. Reiter, and H. Zhang. A Multi-Resolution Approach for Worm Detection and Containment. In *Proceedings of International Conference on Dependable Systems and Networks (DSN)*, June 2006.

[10] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyson. FRESCO: Modular Composable Security Services for Software-Defined Networks. In *Proceedings of Network and Distributed Security Symposium*, 2013.

[11] Snort. `http://snort.org`.

[12] T.-F. Yen and M. K. Reiter. Are Your Hosts Trading or Plotting? Telling P2P File-sharing and Bots Apart. In *Proceedings of IEEE ICDCS*, 2010.