IntelFlow: Towards adding Cyber Threat Intelligence to Software Defined Networks

Javier Richard Quinto Ancieta[1], **Christian Esteve Rothenberg**[1]

[1]School of Electrical and Computer Engineering (FEEC)
University of Campinas (Unicamp)

{richardq,chesteve}@dca.fee.unicamp.br

Abstract. *Security is a major concern in computer networking, which faces increasing threats as the commercial Internet and related economies continue to grow. Our work aims to explore advances in Cyber Threat Intelligence (CTI) in the context of Software Defined Networking (SDN). More specifically, we propose IntelFlow, an intelligence detection system for Software Defined Networking (SDN) that follows a proactive approach using OpenFlow to deploy countermeasures to the threats learned through a distributed intelligence plane. We show through a proof of concept implementation that the proposed system is capable of delivering a number of benefits in terms of effectiveness, altogether contributing to the security of modern computer network designs.*

1. Introduction

Intruder Prevention and Detection Systems (IDPS) are security-oriented networking devices that monitor, analyze, and respond against different threats by using a myriad of methods, including signatures and anomaly detection algorithms. Continuous efforts on the security front on IDPS and networking technologies are devoted to catch up with the innovative attack vectors from the threatening parties, which count (and even sell as a Service) bots in the size of 10s of thousands of infected hosts capable of initiating DDoS attacks in the order of 100s of Gbps.[1]

Over the years, information related to (ongoing) computer security threats are being published with more frequency, though, most of the the information is not correctly evaluated when delivered or is often irrelevant for an organization. Recent advances in open source IDPSs, more specifically BroIDS, includes an `intelligence` interface to read data provided by external sources. Such developments go hand in hand with the recent trend on Cyber Threat Intelligence (CTI) [Johnson et al. 2014], where trusted organizations join their forces and share their `intelligence` about detected threat information [iSIGHT 2014, p. 3].

In our work, we are argue that the only path forward to advance the Internet security and be able to protect against DDoS attacks is having organizations act collaboratively in a distributed, joint approach to fight back security threats. Towards this end, we propose IntelFlow, an intelligent system of intruder detection and prevention that leverages CTI frameworks in the context of Software Defined Networks [Kreutz et al. 2015] to proactively create security rules that allow blocking malicious traffic by programming OpenFlow-enabled switches.

[1]https://blogs.akamai.com/2015/08/q2-2015-state-of-the-internet-security-report-released.html

2. Related Work

SnortFlow [Xing et al. 2013] builds a flexible IPS system in cloud environments, base on the performance evaluation of the virtual machines, reconfiguring the network in case of any abnormal activity. However, SnortFlow only focuses on an intra-domain environment, with a snort agent acting on the domain of the XEN virtualization platform.

BroFlow [Lopez et al. 2014] proposes a system capable of reacting against Denial of Service (DoS) attacks in real time, combining Bro IDS and the OpenFlow application programming interface. However, the authors only use reactive applications to counter those threats, without generating policies based on threats learned.

IPSFlow [Nagahama et al. 2012] is a solution of IPS based on SDN/OpenFlow with automatic blocking of malicious traffic. IPSFlow uses an application that allows the communication with the SDN controller. However, the time taken to detect threats is long, since each IDS waits for the confirmation of the controller to continue sending packets, as it does not mirror interfaces.

3. IntelFlow Architecture

The IntelFlow architecture (Fig. 1) is composed by the following modules: (1) Process Connectors: Receive notifications from Bro about a possible threat, (2) Decider: Make a decision based on the type of analyzed information, (3) False Positive: Evaluate whether the unknown source is a false positive or not, (4) Intel framework: Handle the intelligence received from reliable sources, formatting them to Bro IDS fields with intelligence indicators, (5) Knowledge Plane(KP): Store the information of known or unknown threats previously processed by the Notice and Intel frameworks, (6) Flow Mapping: Perform the flow mapping of the store data, and (7) IntelFlow application: Execute reconfiguration's actions on switches affected. The remaining components are part of the original Bro IDS and the SDN controller architectures. IntelFlow reads intelligence feeds by using of the Collective Intelligence Framework (CIF)[2]and Bro's input framework.

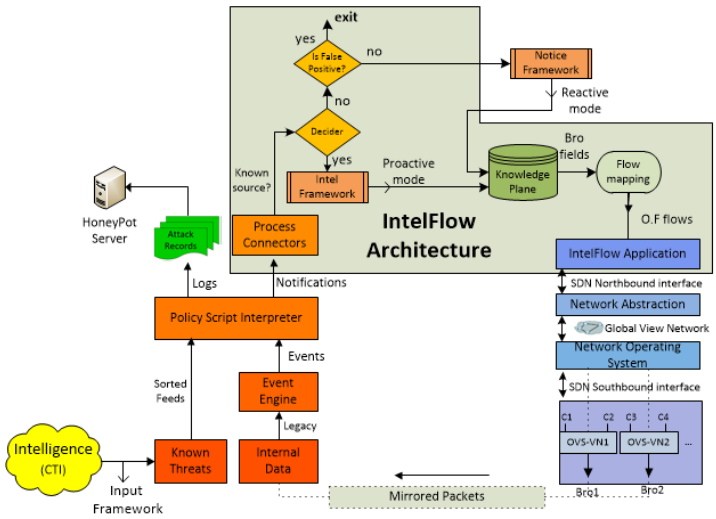

Figure 1. IntelFlow Architecture

[2]http://csirtgadgets.org/collective-intelligence-framework/

SDN architectures can be divided into three types of control applications: reactive, proactive, and hybrid. In this paper, we focus on RESTful APIs for proactive applications where the detection intelligence is based on CIF imported into an IDS.

To take advantage of the intelligence sources, these need to be reliable, and, in turn, relevant, actionable, and valuable for the organization [iSIGHT 2014]. Based on this `intelligence`, organizations can deploy countermeasures based on the experience acquired by their trusted organizations. IntelFlow countermeasures are based on OpenFlow rules set by the SDN controller of choice (OpenDaylight) using the `HTTP PUT` methods. OpenFlow flow entries with drop action discard all packets from the sources responsible for the attack – optionally forwarding the flow packets to a HoneyPot server for further analysis. The essential part of our work consists in defining new flows according to the event triggered, e.g., a flow created for DoS is different that another flow for malicious domains. To avoid duplicate flows, we add a different priority value for each one. Finally, countermeasure are installed in the OpenFlow-enabled data plane devices.

4. Prototype and Experimental Evaluation

The prototype implementation uses Open vSwitch as OpenFlow device, KVM virtual machines to host servers such as Bro, CIF, SDN controller and containers instead of hypervisors to emulate the behavior of malicious hosts with minimum impact over performance.

We evaluate two types of attacks: DoS and malicious websites. When using a proactive approach, new security filters, learned periodically from CIF, are added to the OpenFlow switch flow table before new threats hit the network. In the reactive intelligence approach, our system reacts against different malicious events, that have not been registered yet in the flow table, by matching the threats detected and the intelligence located in the KP. Once an indicator is found, IntelFlow immediately sends countermeasures dropping the origin of the malicious packets, and the same time sends a copy of the attack source to a HoneyPot Server for deep analysis.

The communication between Bro IDS, controller, and CIF is based on a dedicated channel using SSL protocol, ensuring an secure and isolated communication between them. The countermeasure messages are sent in JSON format, and these contain flow data such as “destination and source IPs”, “destination port”, “ethertype”, “priority”, and the “protocol field” that indicate actions to match against a specific threat. All our experiments were executed ten rounds, varying the number of containers and the rate of packets per second involved in each attack. Figure 2(a) illustrates DoS attack experiments and how the attack traffic is mirrored to Bro and in turn dropped by SDN controller. Figure 2(b) compares the response time (t_r) for different packet sending speeds from 200 to 10^4 packets per second (r_{pps}). We consider 10^4 the capacity threshold of each container.

Our experiments evaluate the behavior of both proactive and reactive methodologies. We note better t_r for all values of r_{pps} when using the intelligence. Also, from $4*10^3$ pps to 10^4 pps, t_r grows faster in the conventional methodology because at higher rates the intelligence shows to be more efficient. The experiments also demonstrated that the memory and CPU usage raised quicker to 100% in the conventional methodology (r_{pps} = $6 * 10^3$) when compared to our proposed approach (r_{pps} = 10^4).

In a second experiment, we use the more reliable malicious domain blacklists (MDBLs), such as `malware domain`, to detect and block threats at the time that an

user tries to access one of these malicious website. We demonstrate that the time to disconnect the user from website is nearly small (70 msec). Thus, IntelFlow is capable of intelligently blocking the access attempts from users to websites before or at the beginning of the attack, e.g., preventing the victim getting infected with malware software.

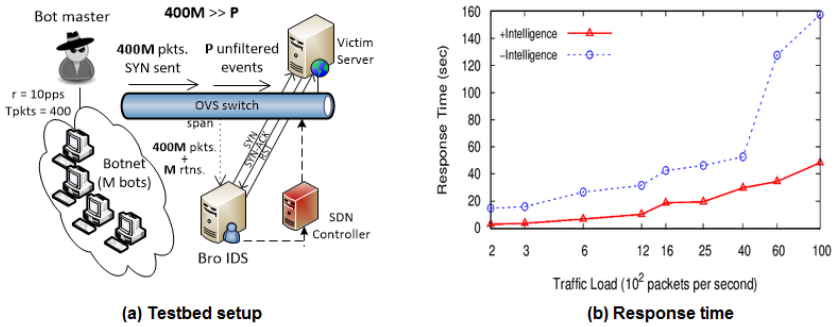

Figure 2. Experimental evaluation of DoS attacks

5. Conclusions

In order to understand how the adversaries work, organizations resort to new security approaches based on trusted, shared information, delivered in a timely manner, and actionable for organizations. This trend is referred to as CTI allows changing the security model from reactive to proactive, enabling to develop new techniques to combat threats based on experience acquired by other organizations. In this work, we present IntelFlow, an architecture that leverages mechanisms of CTI and Bro to proactively drop different types of threats. Our experiments show better response times when used the intelligence located in the KP. As future work, we intend to explore with more detail the process of correlation between the information obtained from reliable sources and from IDS sensors strategically located in different public networks. By using machine learning, we could design new attack models that allow collaborative efforts towards a more secure Internet.

References

iSIGHT (2014). What is Cyber Threat Intelligence and why do I need it? Technical report.

Johnson, C., Badger, L., and Waltermire, D. (2014). Guide to cyber threat information sharing. Technical report, U.S Departament of Commerce.

Kreutz, D., Ramos, F., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proc. of IEEE*, 103.

Lopez, M. A., Figueiredo, U., Lobato, A. P., and DUARTE, O. C. M. B. (2014). Broflow: Um sistema eficiente de detecção e prevenção de intrusão em redes definidas por software. In *CSBC*, Centro de Convenções Brasil 21. CSBC2014.

Nagahama, F. Y., Farias, F., Aguiar, E., Luciano, G., Granville, L., Cerqueira, E., and Antônio, A. (2012). Ipsflow: uma proposta de sistema de prevençao de intrusao baseado no framework openflow. In *III WPEIF-SBRC*, volume 12, pages 42–47.

Xing, T., Huang, D., Xu, L., Chung, C.-J., and Khatkar, P. (2013). Snortflow: A openflow-based intrusion prevention system in cloud environment. In *Proc. of GREE '13*, pages 89–92, Washington, DC, USA. IEEE Computer Society.

