Science DMZ: SDN based Secured Cloud Testbed

Ankur Chowdhary, Vaibhav Hemant Dixit, Naveen Tiwari, Sukhwa Kyung, Dr. Dijiang Huang and Dr. Gail-Joon Ahn
Arizona State University
{achaud16, vdixit2, nktiwar1, skyung1, dijiang, gahn1}@asu.edu

***Abstract*—Software Defined Networking (SDN) presents a unique opportunity to manage and orchestrate cloud networks. The educational institutions, like many other industries face a lot of security threats. We have established an SDN enabled Demilitarized Zone (DMZ) - *Science DMZ* to serve as testbed for securing ASU Internet2 environment. *Science DMZ* allows researchers to conduct in-depth analysis of security attacks and take necessary countermeasures using SDN based command and control (C&C) center. Demo URL:** $https : //www.youtube.com/watch?v = 8yo2lTNV3r4$

I. Introduction

There has been a surge in security breaches in recent years. IRS hack in year 2016 compromised details of 100,000 FAFSA student applicants. Government agencies, corporate, educational institutions and other sectors are facing an increased number of security threats.

SDN allows centralized management of the underlying network as opposed to traditional networks. Recent advances in deployment of experimental testbeds like GENI[1] allow campus networks to be connected to each other.

GENI[1] has also introduced the possibility of designing new network protocols, content management and network service deployment. There is, however limited work on SDN based security testbeds for campus networks.

We demonstrate a secured testbed *Science DMZ* for management of network traffic, identification of suspicious attack vectors and taking necessary countermeasures to prevent security breaches. *Science DMZ* allows us to leverage SDN for security incident, events, network traffic policy management and attack pattern recognition. The Science DMZ framework automatically detects and resolves any SDN flow rule violations.

The key contributions of this research work are the following

- *Science DMZ* security analyzer and *Science DMZ* GUI.
- SDN based smart firewall, honeypot and load balancer.
- SDN Flow rule conflict checker and visualizer.

II. System Architecture

The system architecture is based on Software Defined Networking based command and control (C&C). The gateway physical server at the edge of the network consists of Intrusion Detection System (IDS) and honey proxy. The IDS checks attack signatures for normal and malicious traffic. The normal traffic is allowed to interact with the underlying cloud infrastructure based on Openstack cloud.

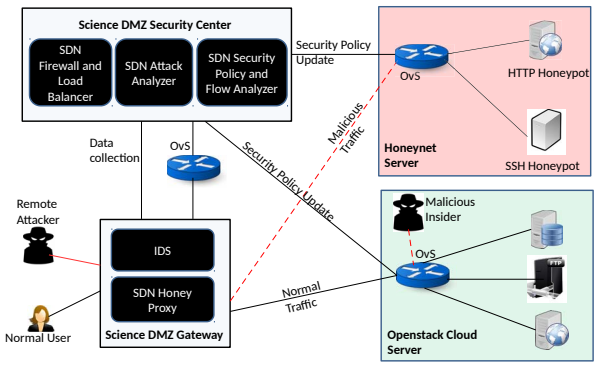

Fig. 1. System Architecture

The malicious traffic is sent to Honeypot server where we log all attack activities to perform security analysis. Various components of the system have been described below:

1) **Network Intrusion Detection System (NIDS)** The IDS performs signature based attack pattern detection on the network gateway. This information is passed to the Opendaylight SDN controller using northbound APIs.
2) **SDN Firewall and Load Balancer** SDN Controller on *Science DMZ Security Center* utilizes streaming traffic information logged by IDS to provide load balancing functionality. The SDN firewall redirects normal traffic to Openstack cloud and the abnormal traffic to the Honeynet server via SDN Honey Proxy.
3) **SDN Honey Proxy** This component is used by the SDN controller to detect fingerprint based attacks[2]. Honey proxy is located at the Science DMZ gateway and informs controller about such attacks. The SDN controller utilizes this information to redirect traffic to low-interaction or high-interaction honeypot based on the type of attack traffic.
4) **SDN Attack Analyzer** The attack analyzer 1 located on *Science DMZ Security Center* uses the system vulnerability information for various VMs running on the system, honeynet server logs, Openstack cloud server logs, firewall policy information to generate scalable attack graphs and perform security analysis based on various attack paths identified from attack graph[3].
5) **SDN Security Policy and Flow Analyzer** automatically detect and resolves any flow policy violations using an adaptive security feature for SDN controller. To find any

978-1-5386-3285-7/17/$31.00 ©2017 IEEE

inconsistencies, the flow rules present in the data plane are matched against the firewall denied authorization space. At the same time, this incident is also reported to the Log Analysis module for attack classification and generating new security constraints.

A. Science DMZ Testbed Flow Chart

The *Science DMZ* security center flow chart above describes how connection requests directed towards services hosted on *Science DMZ* cloud and honeypot servers are handled.

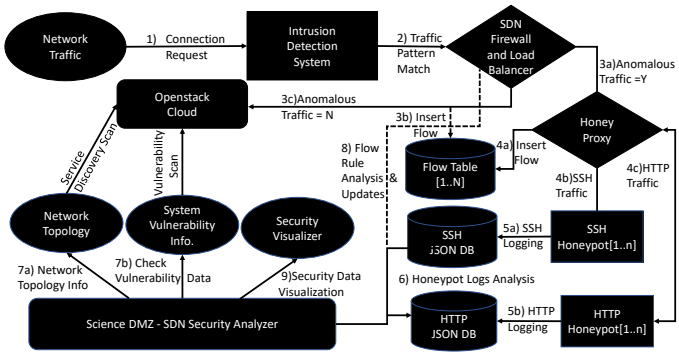

Fig. 2. Science DMZ Information Flow Chart

- **Step 1 and 2:** The traffic is inspected for malicious attack pattern and redirected to respective honeypot.
- **Step 3:** If there is a signature match for anomalous traffic we forward malicious traffic to HoneyProxy. When traffic volume is high SDN controller performs load balancing. Normal traffic is forwarded to Openstack cloud.
- **Step 4:** The anomalous traffic is directed by HoneyProxy to appropriate honeypot and traffic rules are inserted in corresponding flow table - 4a), 4b), 4c).
- **Step 5:** The logs for SSH and HTTP honeypots are stored in Mongo DB collections in JSON format.
- **Step 6:** The SDN security analyzer checks, SSH and HTTP honeypot logs for user behavior recognition.
- **Step 7:** We inspect network topology and vulnerability information to create a scalable attack graph for system and check potential attack impact from a provided source.
- **Step 8:** The security analyzer uses a listener on SDN Opendaylight controller's operational data store for security policy conflicts and flow rule updates. The information collected from steps 6 and 7 is used to update Firewall policies in SDN controller.
- **Step 9:** Information from Steps 7 and 8 is utilized for visualization, attack graphs regeneration and SDN flow rule conflict update.

III. Implementation Details

Science DMZ testbed consists of four Dell R620 servers and two Dell R710 servers all hosted in the ASU datacenter. Each Dell server has about 128 GB of RAM and 16 core CPU. We use HP Openflow switches (Openflow v1.3) to connect the physical servers together. One physical server acts as gateway and provides IDS and proxy services to important components of *Science DMZ* like Openstack Cloud, Honeynet Server, SDN controller, etc. The SDN controller Opendaylight-Carbon has been used for network management and orchestration. *Science DMZ* GUI is based on a PHP lavarel framework with bootstrap and uses the REST API to manage SDN controller, the Openstack Ocata backend and other network segments.

IV. Related Work

We use SDN based honeypot i.e. HoneyMix[2] to identify attackers activity using a low interaction honeypot. We use the cloud system vulnerability and reachability information to generate attack graphs. The attack graph used helps in security assessment and SDN based countermeasure selection for cloud networks as discussed by Chowdhary *et al*[3]. The SDN firewall based on our previous work Flowguard[4] helps in checking flow rules for security conflicts in *Science DMZ* environment. We have utilized flow conflict resolution algorithm from FlowGuard to resolve security and flow rule conflicts in *Science DMZ* testbed. Pisharody *et al*[5] use SDN controller for security policy conflict detection, resolution and visualization. Our security visualization module is based on our previous works for policy conflict visualization[5] and attack graph visualization[3].

V. Conclusion and Future Work

The demo of *Science DMZ* showcases management and security of a cloud network using SDN. We have currently implemented major components such as SDN enabled Firewall and Flow rule conflict checker, SDN honeypot and load balancer, *Science DMZ* security analyzer. Our GUI allows system users to orchestrate and control, security issues in a cloud environment. We plan to extend log analysis part of the system to identify advanced persistent threats (APT) scenarios and perform attacker's behavior analysis.

Acknowledgment

This research is based upon work supported by the National Science Foundation under Grant 1642031 and NSF Secure and Resilient Networking Project under Grant 1528099. Special thanks to the UTO team at ASU for their continuous support.

References

[1] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "Geni: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, pp. 5–23, 2014.

[2] W. Han, Z. Zhao, A. Doupé, and G.-J. Ahn, "Honeymix: toward sdn-based intelligent honeynet," in *Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*. ACM, 2016, pp. 1–6.

[3] A. Chowdhary, S. Pisharody, and D. Huang, "Sdn based scalable mtd solution in cloud network," in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*. ACM, 2016, pp. 27–36.

[4] H. Hu, W. Han, G. J. Ahn, and Z. Zhao, "Flowguard: Building robust firewalls for software-defined networks," in *3rd ACM SIGCOMM 2014 Workshop on Hot Topics in Software Defined Networking, HotSDN 2014*. Association for Computing Machinery, 2014.

[5] S. Pisharody, A. Chowdhary, and D. Huang, "Security policy checking in distributed sdn based clouds," in *Communications and Network Security (CNS), 2016 IEEE Conference on*. IEEE, 2016, pp. 19–27.