Rotating Adaptive Network Defense (RAND)

Design Document

Team ID: sddec18-07

Client: Argonne National Laboratory, Dr. Benjamin Blakely and Joshua Lyle

Faculty Advisor: Dr. Hongwei Zhang

Team Members and Roles:
Andrew Thai - Project Manager
Connor Ruggles - Quality Assurance
Emily Anderson - Delivery Manager
Ryan Lawrence - Communication Manager

Team email: sddec18-07@iastate.edu
Team Website: https://sddec18-07.sd.ece.iastate.edu/

Table of Contents

List of Figures

List of Definitions and Acronyms

SDN: Software-Defined Network

- Software-defined networking is a concept where the actual routing of data packets is moved to a separate layer and is taken care of programmatically by a network controller, that then sends the packets down to the main network switch to route to the individual servers on the network.

MTD: Moving Target Defense

- This is a concept where you detect if a specific machine is being attacked and you have preset rules to mitigate to rotate that machine out of being public facing, and rotate in a "honeypot", or something that looks like a real machine but it distracts the attacker long enough to block them out.

NIC: Network Interface Card

- This is the physical device that connects all of the machines connected to a switch, to the Internet.

CDC: Cyber Defense Competition

- A type of competition where teams try and defend a set of servers against a team of attackers in a pre-defined scenario.

VM: Virtual Machine

- A software emulation of physical aspects needed to run a full computer operating system.

1 Introduction

1.1 Acknowledgement

Our clients, Dr. Benjamin Blakely and Joshua Lyle from Argonne National Laboratory, have been the biggest contributors to our project so far. They have given us topics to research, tools to use, and overall direction on what they are looking for in this project. Our advisor, Dr. Hongwei Zhang, has also helped us throughout this process on larger deliverables such as our project plan, design doc, etc.

1.2 Problem and Project Statement

As the pace of advancement in information and operational technology systems rapidly increases, cyber-attacks become more sophisticated due to the additional resources available to cyber criminals. They have become harder to detect and more effective at penetrating networks. Cyber criminals might spend weeks and months gathering information on target networks to plan out their attack, making sure that they have the right information so that their attacks will work efficiently and effectively.

Our solution consists of a software defined network (SDN) controller that dynamically adjusts where incoming packets are directed when they are being transmitted to a server. By doing so, we will be able to route traffic on the fly to migrate, take down, or add new servers to the network without any downtime. We will utilize a SDN as a moving target defense (MTD) system. The controller we develop will dynamically configure the network to detect any packets that are malicious or come from an information gathering reconnaissance tool and direct them to dummy servers, also known as honeypots. This will prevent or delay an attacker from obtaining any reliable information about the network. This could result in many wasted attempts to grab information regarding the constantly changing network, thus allowing the network to be more difficult to attack than a static configuration.

1.3 Operational Environment

This design will be used in a location where public-facing servers are located. For example, many institutions use a demilitarized zone (DMZ) network segment for web servers or email services which require incoming requests to be served. Such servers could be located in datacenters or on-premises at a facility belonging to the owner. Any physical hardware, such as switches or a server to host the controller, that would be put into place would be able to withstand standard networking environments such as networking closets or datacenter cabinets.

1.4 Intended Users and Uses

The intended users for the developed product are any company with services that use multiple virtual or physical servers, whether internal or not, such as hosting a website or any other service that uses some sort of a network connection between multiple other servers. This design can also be used for government or military institutions to protect from various information gathering attacks.

This SDN MTD product will provide an extra layer of security by dynamically routing traffic to an array of systems thus allowing for a wide variety of maneuvering to impede network scanning.

1.5 Assumptions and Limitations

Assumptions:

1. Physical or virtual switches used must support the OpenFlow protocol.
2. All switches must have a route to connect to the SDN controller.
3. An implementation of SDN can delay an attacker enough so we can mitigate the attack.
4. There are companies/customers willing to implement SDN on their own network.

Limitations:

2 Not ideal for a home network.

2.1 The resources required and scope of the whole system would be inefficient for the size of a typical home network.

2.2 Expected End Product and Deliverables

The end product will consist of :

- A research paper describing:
  - Background on SDN, the protocols selected for our implementation (e.g., OpenFlow);
  - Gaps in existing implementations similar to what is being developed;
  - The threat model defining the scope of attackers the product is designed to defend against;
  - Details of the implementation of the SDN MTD product (including diagrams, where appropriate);
  - The evaluation methodology used to assess the performance of the product and degree to which it counters the in-scope attacks;
  - Results of the assessment;
  - Recommendations for future work;
- Source code or configurations for a SDN controller with basic routing rules (and documentation for creating more specific rules);
- An executable and/or process to aid the end user in deploying the controller onto a virtual machine network; and
- Any other configuration files needed for the system to work as expected.

Research Paper - This is the primary deliverable and will lay out the procedures for the entire project so that the methodology can be assessed, replicated, and extended.

Installer/Install Directions - These will be either a full-fledged installer that will setup the controller for the user automatically, or directions on how to do so manually. This will consist of either the directions only, or both the installer and the directions.

Configuration Files - Will be provided if config files are needed for controller setup

Other deliverables include usability and effectiveness of the system which show tested results that describe the impact of using this system as well as if it actually makes a significant difference than just using a regular network.

2. Specifications and Analysis

2.1 Proposed Design

Our proposed design consist of adding a hypervisor into the production environment of a company with the following machines: Floodlight Controller, Security Onion, and a honeypot. In addition to the added hypervisor, companies will also need OpenFlow compatible switches that will be able to interface with the Floodlight Controller. This setup will allow for any packets that route through the switches to ask the controller for what the next hop will be based on defined rules that the company has implemented. Along with the controller, there will be implementation with Security Onion to monitor network traffic as an intrusion detection system to allow for alerts of attacks. With the use of Security Onion, companies can create rules based on the results and alerts gathered from Security Onion to mitigate attacks and damages that may affect performance or usability.

Standards include:
IEEE standards - Ethernet packets
OpenFlow standards - Switch protocols

An image of our proposed design can be seen in Section 4.3.

2.2 Design Analysis

With our test setup we are able to create a more realistic network that we could perform functional and non-functional requirements and allow us to directly control all machines within the environment. We have found that this setup best allows us to provide the proper environment to determine what packet control we want given a specific scenario.

The strengths of our proposed solution makes it easier for us to determine all endpoints of where packets are transferred, giving us full accessibility to the machines and the switches.

3 Testing and Implementation

3.1 Interface Specifications

Our current test design to mimic a company’s production environment will consist of creating a virtual hypervisor, Citrix XenServer, because we noticed that the hypervisor supported the use of Open vSwitches. Through this we will create a backbone of machines within XenServer to test our the virtual switch controller configuration and determine the packet’s path with multiple virtual machines running. We will interface the Open vSwitch by using the Floodlight Open SDN Controller to direct traffic to and from our virtual machines.

3.2 Hardware and software

Hardware that we used for this design consisted of a Dell PowerEdge R710 Server. This server is running VMware Hypervisor to allow for creation and configuration of virtual machines. Having our own physical server to work on allows for us to easily create and destroy machines as needed in our design without having to get more hardware.

We will be using Floodlight Open SDN Controller as our controller because it was straightforward to use and it supports the OpenFlow Protocol which will allow us to manage the flow of traffic within our testing network to the machines. We will also be using utilities such as Wireshark to analyze packets as well as create determine the effectiveness of our packet control design.

3.3 Functional Testing

Functional tests will include but are not limited to:

- Accessing a web server that will direct to two or three different servers.
- Nmap scan from a Kali Linux box and seeing that the packets route to the correct server.
- Make sure the controller can be inserted into an existing network (this will most likely be a home network of one of our team members').
- Relieving DDOS pressure by blocking connection from an ip if an overload of packets is sensed

3.4 Non-Functional Testing

- Availability - The company's provided services will not be hindered by the network design. There should be no outages as well as no unexpected downtime.
- Usability - Customers should see no change in the services they access over the web.
- Integrity/Security - Data transmitted throughout will stay encrypted within the software defined network.

3.5 Process

We started out the semester with an abundance of research before we did any testing on an actual network. We researched every aspect of the project we could since it was such a new concept to us and determined the main specifications of what we were aiming to accomplish. We then created a test network and tested different ways to implement our project and ultimately decided to use Floodlight Open SDN Controller and Security Onion along with Open vSwitch and our hypervisor. Floodlight supports the OpenFlow protocol, which is what we used to create rules to control the machines, such as disabling/enabling traffic or rerouting traffic.

3.6 Results

After a great deal of experimentation, we decided to use Floodlight, Security Onion, and XenServer as the main services and have successfully created a network where we have been working on prototypes of flow control and performing tests to determine if the system is working.

4 Closing Material

4.1 Conclusion

With the amount of security risks that static networks can face in today's world, a solution to provide extra layers of security to the network is needed. Our goal of creating a Software Defined Network Moving Target Defense (SDN MTD), will help to alleviate this risk. By creating this we will be able to monitor, control, and analyze packets that go through a network and minimize the risk of information gathering and manipulate the flow of traffic to protect the network as a whole.

4.2 References

[1] Jafarian, J. H., Niakanlahiji, A., Al-Shaer, E., & Duan, Q. (2016). Multi-dimensional Host Identity Anonymization for Defeating Skilled Attackers. Proceedings of the 2016 ACM Workshop on Moving Target Defense - MTD16. doi:10.1145/2995272.2995278

[2] Kampanakis, P., Perros, H., & Beyene, T. (2014). SDN-based solutions for Moving Target Defense network protection. Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014. doi:10.1109/wowmom.2014.6918979

[3] Mahler, D. (2014). Netfool Networking. Retrieved from https://www.youtube.com/user/mahler711

[4] Okhravi, H., Rabe, M. A., Mayberry, T. J., Leonard, W. G., Hobson, T. R., Bigelow, D., & Streilein, W. W. (2013). Survey of Cyber Moving Target Techniques. doi:10.21236/ada591804

[5] Skowyra, R., Bauer, K., Dedhia, V., & Okhravi, H. (2016). Have No PHEAR. Proceedings of the 2016 ACM Workshop on Moving Target Defense - MTD16. doi:10.1145/2995272.2995276

[6] Stakhanova, Natalia; Basu, Samik; and Wong, Johnny S., "A Taxonomy of Intrusion Response Systems" (2006). Computer Science Technical Reports. Paper 210. http://lib.dr.iastate.edu/cs_techreports/210

[7] Zhuang, R., Bardas, A. G., Deloach, S. A., & Ou, X. (2015). A Theory of Cyber Attacks. Proceedings of the Second ACM Workshop on Moving Target Defense - MTD 15. doi:10.1145/2808475.2808478

[8] Zhuang, R., S. A., & Ou, X. (2015). Towards a Theory of Moving Target Defense. Proceedings of the First ACM Workshop on Moving Target Defense - MTD 14. doi:10.1145/2663474.2663479

[9] Citrix Systems, Inc., Documentation. https://xenserver.org/overview-xenserver-open-source-virtualization/documentation.html

4.3 Appendices

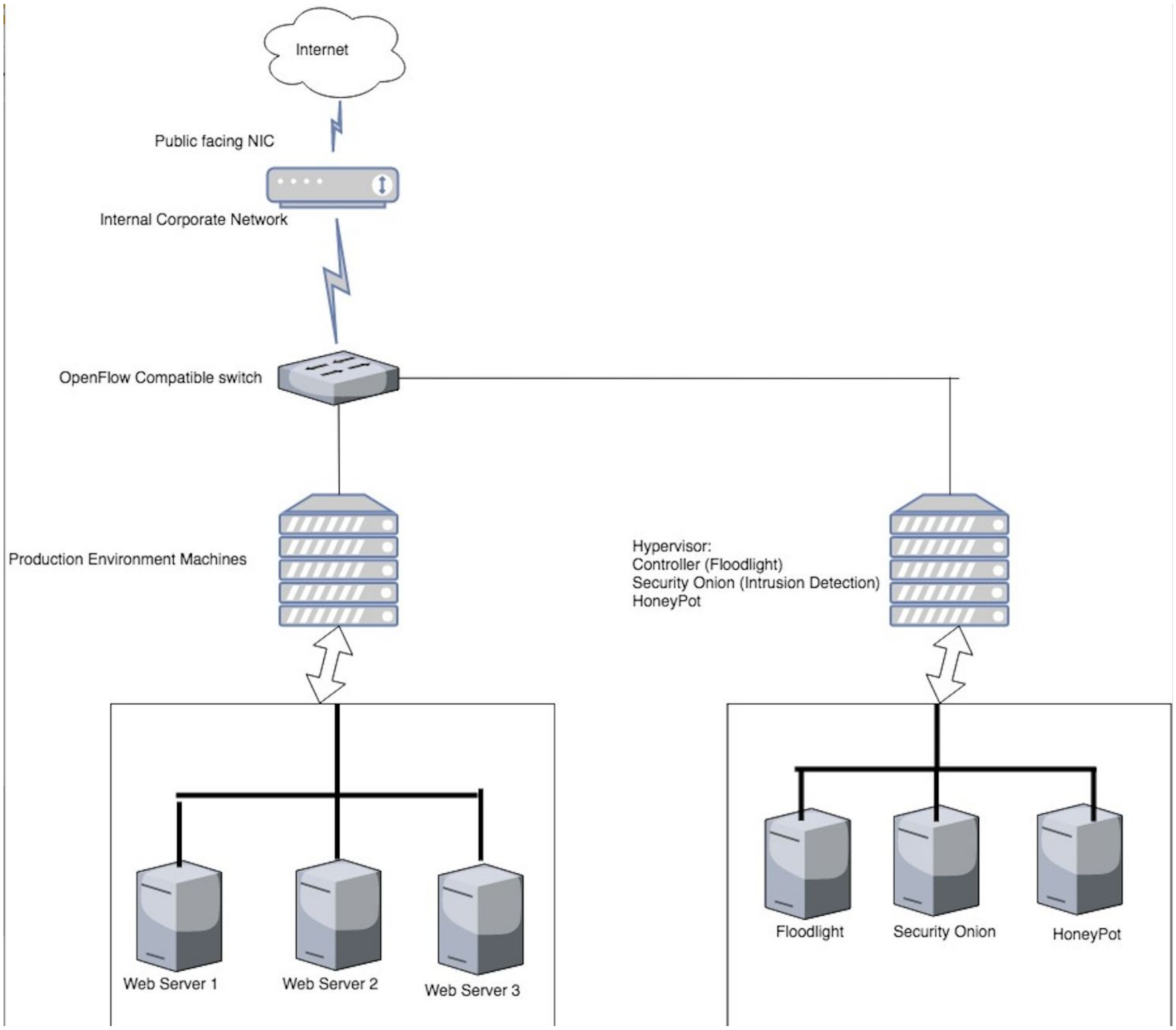

Figure 1: Representation of a software defined network with the deliverable controller installed