Software-Defined Network Moving Target Defense
sddec18-07

THE TEAM

Andrew Thai, Connor Ruggles, Emily Anderson, Ryan Lawrence, Corey Wright
Client: Dr. Benjamin Blakely and Joshua Lyle (Argonne National Laboratories)
Faculty Advisor: Dr. Hongwei Zhang

INTRODUCTION

Problem: Hackers spend weeks or months gathering information on corporate networks to plan out their attack, making sure that they have the right information so that their attacks will work efficiently and effectively.

Solution: Create a software-defined network which consists of dynamically programming where packets are directed to when they are sent to a corporate server. Such a network will be able to quickly route traffic on the fly so that we can migrate, take down, or add servers to the network and minimize, or eliminate, network downtime.

TECHNICAL DETAILS

Function Modules
- Floodlight - Software Defined Network Controller
- Snort - Network-based intrusion detection system that creates alerts based on certain network attacks as well as different types of scanning that we manually create
- XenServer - Used to host our virtual machines that we are load balancing to as well as acting as an open vswitch that can be connected to our Floodlight controller

Software Modules
- Shell Script - Used to create the commands needed to setup load balancing between hosts
- Python - Used to create the interaction between snort and the Floodlight controller such that it creates Floodlight rules on the controller based off the alerts that Snort outputs.
- Angular - Used to create the interface

DESIGN REQUIREMENTS

Functional Requirements
- Snort machine accurately detects nmap scan, nikto scan, ARP spoofing, and DDoS attack traffic and generates alerts
- Floodlight rules are created from the Snort alerts and pushed to the Floodlight controller
- Floodlight controller redirects nmap scans, nikto scans, and ARP spoofing traffic to a honeypot server
- Floodlight controller blocks DDoS traffic
- Load balancing between hosts on the network to decrease packet loss
- Interface to create, edit, view, and delete Floodlight rules

Non-Functional Requirements
- Interface is user friendly
- Floodlight rules last for 3 minutes before disappearing
- Cron job runs every 30 seconds to see if there are new rules to create

Constraints
- OpenFlow 1.1+ compatible
- Switches must communicate with controller
- Load balanced servers in same geographic location
- Snort logging requires large amount of space

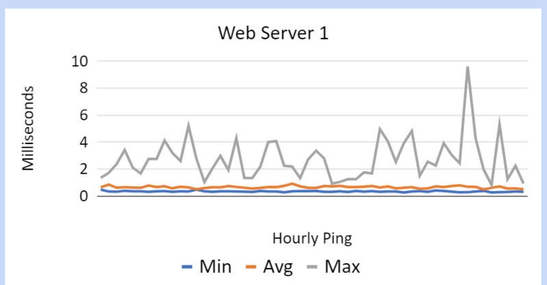
Operating Environment
- Public facing servers including:
  - → Physical servers that use Open vSwitch
  - → Virtual servers that support OpenFlow protocol

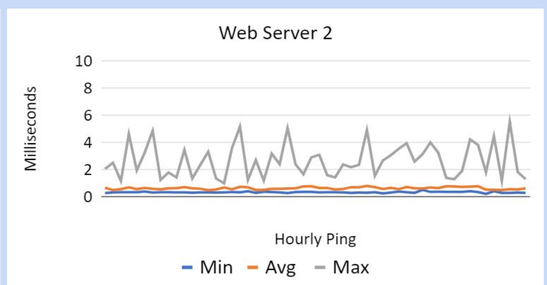
TESTING

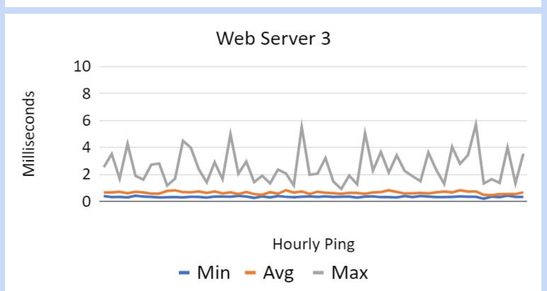
Testing Environment
- Private internal network
- ISU Cyber Defense Competition

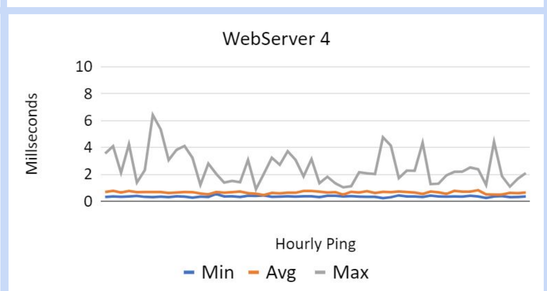
Testing Strategy
- Analyze packet loads under normal conditions vs attack cases
- Min, Max, and Average ping rates
- Packet failures

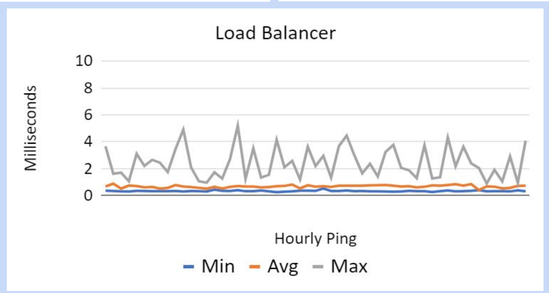
Packet Loads











Standards

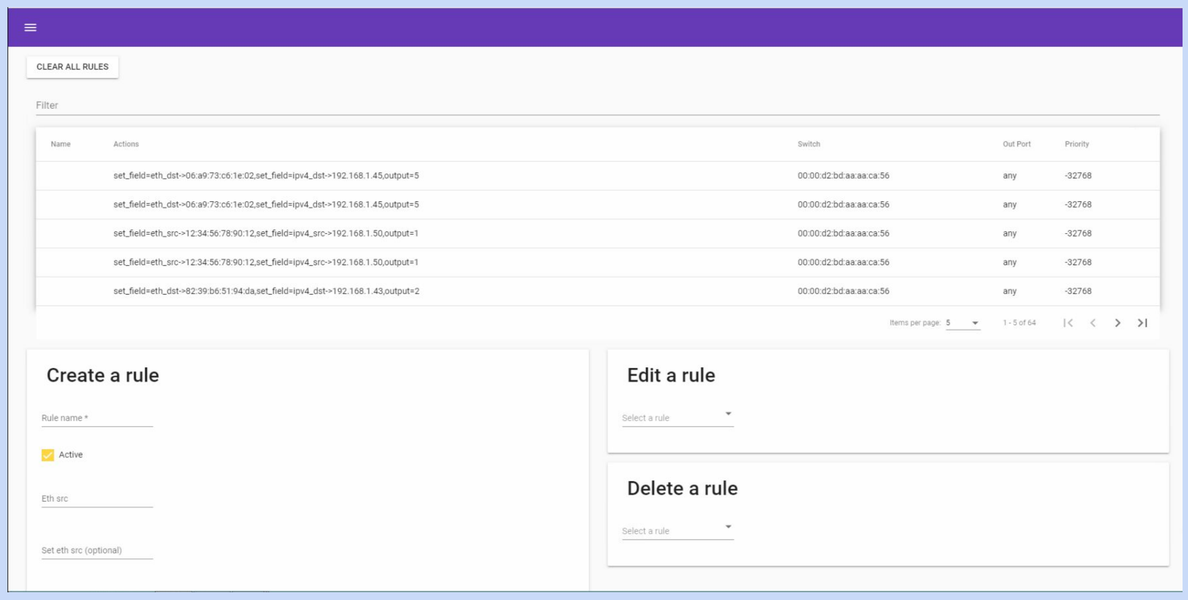
- IEEE 1915.1 Standard for Software Defined Networking and Network Function Virtualization Security
- IEEE 1686-2013 Standard for Intelligent Electronic Devices Cyber Security Capabilities

INTENDED USERS AND USES

The intended user is any company with services that use multiple virtual or physical servers, whether internal or external, such as hosting a website or any other service that uses some sort of a network connection between other servers. This design can also be used for government or military institutions to protect from various information gathering attacks.

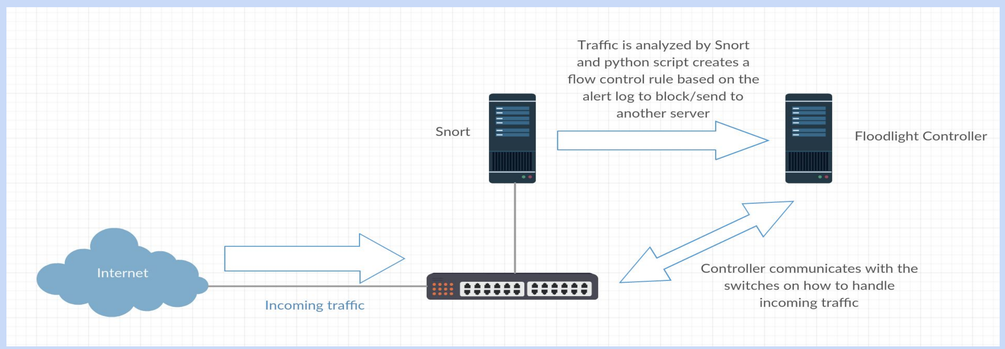
This product will provide an extra layer of security by dynamically routing traffic to an array of systems thus allowing for a wide variety of maneuvering to impede network intrusion.

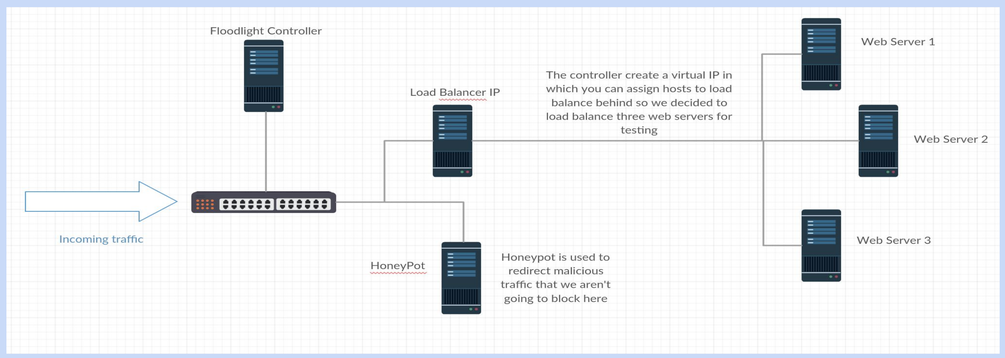
INTERFACE



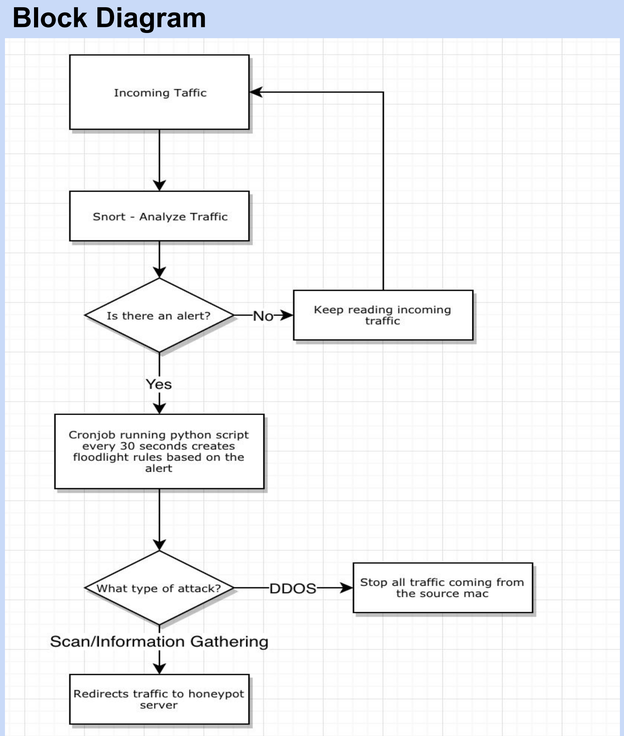
DESIGN APPROACH

Conceptual Sketch





Block Diagram



Main Functional Modules
- Floodlight controller to handle traffic rules
- Snort machine to monitor network traffic
  - → Nikto Scan Detection
  - → Nmap Scan Detection
  - → DDoS Scan Detection
  - → ARP Spoof Detection
- Alert handler to upload rules to Floodlight
- Graphical Interface to upload static rules
- Load balancer to manage traffic across different machines