# A Lightweight rule based Security System for OpenFlow devices

Team Avengers

Prachi Bhole Supriya Singh PK Nidhi Madappa Sunil Venkatraman

Department of Computer Science & Electrical Engineering
University of Southern California
Los Angeles, California

***Abstract -***

Network intrusion detection systems (NIDS) are an important part of any network security architecture. They provide a layer of defense which monitors network traffic for predefined suspicious activity or patterns, and alert system administrators when potential hostile traffic is detected. We suggest a system to be deployed on the OpenFlow protocol to provide a rule based security system for any OpenFlow enabled device. We also suggest some rules that can effectively detect and prevent signature based attacks (like malware signatures), as well as statistical anomaly based attacks. The system will also take custom rule-sets from the users, which can be enabled or disabled at will. We will also present the results we achieved by running the system on the DETER testbed.

***Keywords: DETER, OpenFlow, Network Security, NIDS, Signature based attacks, Statistical anomaly based attack***

## I. INTRODUCTION

The OpenFlow (OF) protocol provides a common interface to control how packets are forwarded by accessing the data plane's internal flow tables, configuration, and statistics. A flow can be defined as a group of packets which belong to the same time interval and share the same specific features.[4]
The data path of an OpenFlow Switch presents a clean flow table abstraction; each flow table entry contains a set of packet fields to match, and an action (such as send-out-port, modify-field, or drop). When an OpenFlow Switch receives a packet it has never seen before, for which it has no matching flow entries, it sends this packet to the controller. The controller then makes a decision on how to handle this packet. It can drop the packet, or it can add a flow entry directing the switch on how to forward similar packets in the future.

Intrusion Detection Systems (IDSs) play an important role detecting various kinds of attacks and defend our computer systems from them. There are basically two main types of detection techniques: signature-based and anomaly based. A signature-based IDS cannot detect unknown attacks because a signature has not been written. To overcome this shortcoming, many researchers have been developing anomaly based IDSs. Although they can detect unknown attacks, there is a problem, they just classify network traffic into normal or abnormal. Therefore, IDS operators have to manually inspect IDS alerts to classify them into known attacks or unknown attacks. Because there are a lot of alerts related to known attacks, it is difficult to extract only unknown attacks from them.
A lightweight intrusion detection system can easily be deployed on most any node of a network, with minimal disruption to operations. Lightweight IDS' should be cross-platform, have a small system footprint, and be easily configured by system administrators who need to implement a specific security solution in a short amount of time. They can be any set of software tools which can be assembled and put into action in response to evolving security situations. Lightweight IDS' are small, powerful, and flexible enough to be used as permanent elements of the network security infrastructure.[3]

### A. *Your Contribution*

As yet, we have not come across a lightweight rule based real time security enforcement system. With this implementation we aim to provide a proof of the concept which is existent in other types of network architectures (eg, SNORT, etc). We realize that the scale of our implementation does not permit much experimentation with different types of rule sets, but the possibilities of such a system are limitless.

## II. RELATED WORKS

In [1] and [2] we see that, the authors have approached the issue of security over an OpenFlow based device in different ways. In [2], the authors concentrate on using NetFPGA devices to generate a pattern matcher which works as a lite NIDS module between the ingress and the egress of the packet. In [1], a more statistical approach is taken to classify and identify a DDoS type attack over OpenFlow systems. This while being a good algorithm to analyze, proves to be only effective on DDoS attacks.

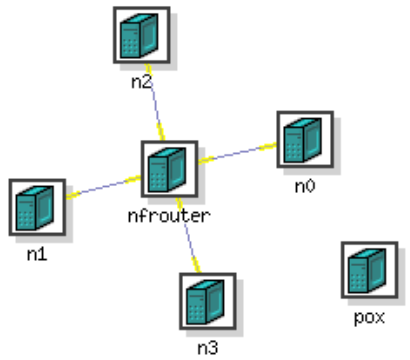
## III. RESEARCH

### A. *Architecture*

The main topology is shown in the Figure. Here we have four nodes (Linux based) and an Open Flow switch. The four nodes are connected to each other through open flow switch. The Open Flow switch is implemented over a NetFPGA board.

Identify applicable sponsor/s here. *(sponsors)*

The Open Flow software is downloaded and installed on the NetFPGA. The topology is designed and controlled using Deter Lab (based on Emulab), which allow the user to access the nodes and NetFPGA board. The nodes will act as the senders and receivers of packets for the experiment simulation. The IDS application will run on the POX controller.




*B. Implementation*

Malicious network traffic has certain patterns to it. We need to automate the task of finding and examining these patterns and execute certain actions. The application will detect and prevent attacks by performing protocol analysis, content searching, and content matching. The incoming network traffic will be analyzed against a pre-defined rule set. For implementing this NIDS, we will design a flexible lightweight rules description language. The rules will have a rule header and rule options. The rule header contains the rule's action, protocol, source and destination IP addresses and netmasks, and the source and destination ports information. The rule option section contains alert messages and information on which parts of the packet should be inspected to determine if the rule action should be taken.

We will have to design the syntax and operators for implementing the rules.

Example sample rules -

1. Log udp traffic coming from any port and destination ports ranging from 1 to 1024 - log udp any any -> 192.168.1.0/24 1:1024
2. Block tcp traffic from any port going to port 6000 - block tcp any any -> 192.168.1.0/24 6000
3. Alert if a virus is detected in the packet content - alert tcp any any -> any any (content:"|trojan|"; msg: "virus detected";)

There will also be a special command line option available to customize the rules for a user.

*C. Execution*

IDS application will sniff packets at the switch and then analyze the packet headers and content to apply the rule sets against it. If a packet matches all the rule options for any rule, it is either logged, blocked or forwarded as per the action specified and an alert is issued.

*D. Experiment Results*

Results of your system experiments

## IV. CONCLUSION

We can publish our results in these conferences :

- IEEEInfocom
- International Conference on Computer Communications and Networks (ICCN)
- ACMSigcom
- ACM Mobicom

## V. REFERENCE

[1] Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow by Rodrigo Braga, Edjard Mota, Alexandre Passito

[2] Pattern Based Packet Filtering using NetFPGA in DETER Infrastructure by Andrew Goodney, Shailesh Narayan, Vivek Bhandwalkar, Young H. Cho

[3] Snort - Lightweight Intrusion Detection fo Networks by Matrin Roesch

[4] http://www.openflow.org/wp/learnmore/

[5] Applying Kernel Methods to Anomaly Based Intrusion Detection Systems by Karim Ali and Raouf Boutaba

[6] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. OpenFlow: enabling innovation in campus networks