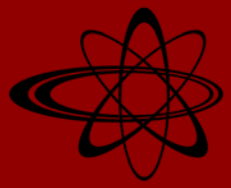


SNRG
Security and Networks Research Group

FIREWALL RULE SET OPTIMIZATION

Author Name: Mungole Mukupa
Supervisor : Mr Barry Irwin
Date : 25th October 2010
Security and Networks Research Group
Department of Computer Science
Rhodes University

RHODES UNIVERSITY
Where leaders learn

Introduction

- Firewalls have been and continue to be used to implement logical security between these networks of different trust levels.
- Network speeds have improved significantly and calls for better packet filtering.
- This research focused on Network layer firewalling.

Problem Statement

- **Sequential Inspection**: Firewalls inspect packets against a set of rules sequentially until a match is found.
- **Matching Patterns**: Not all rules are matched each time a packet is inspected through the rule set.
- **Nature of traffic**: network traffic is dynamic and often flows are not predictable.
- **Network Speed**: Increased network speeds are overwhelming firewalls hence the need for faster filtering decisions by firewalls to avoid waste of device and network resources.

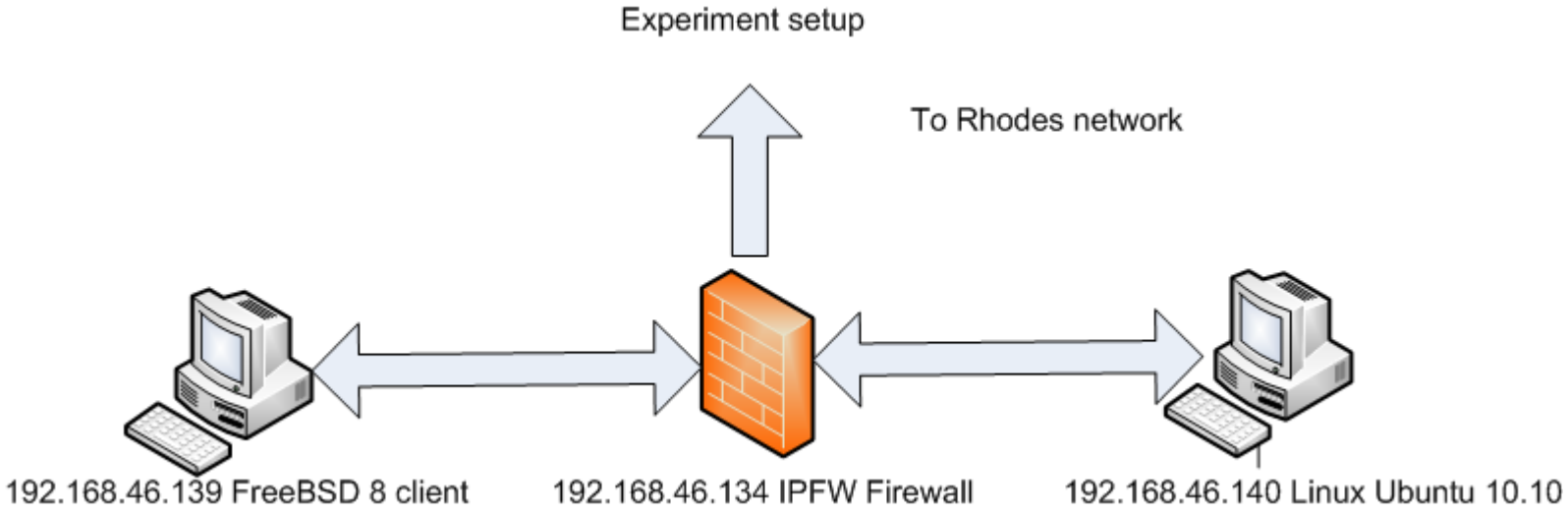
Research Objectives

- To investigate filtering performance improvement gained through optimizing a firewall rule set size.

- Come up with a tool that can aid network administrators in rule set optimization.

Experiment Design

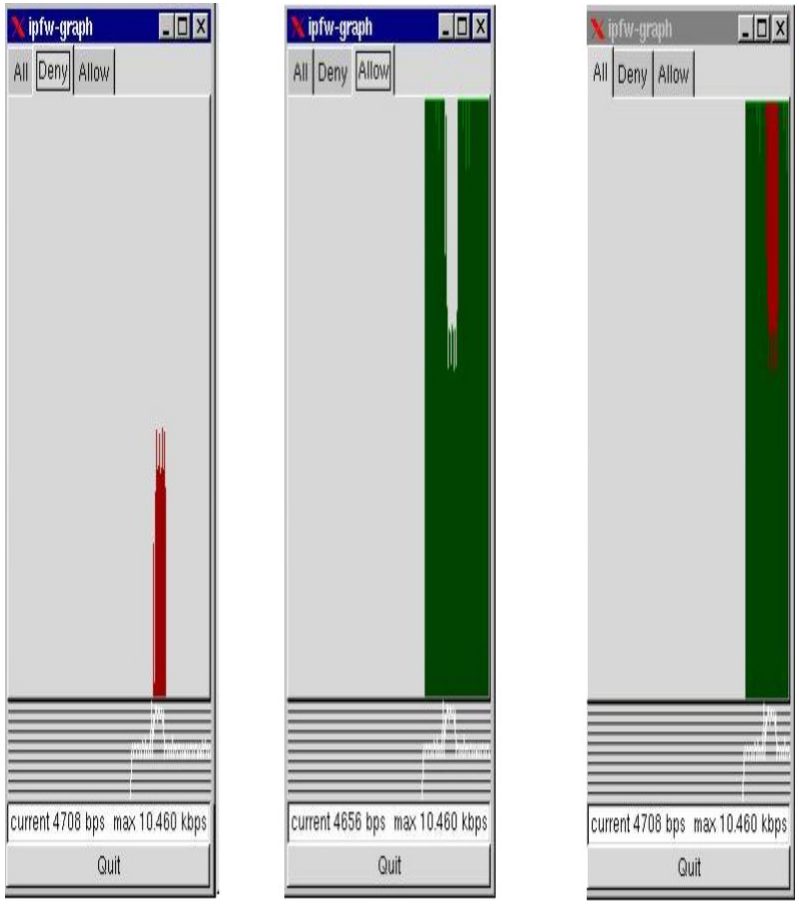
The test was performed in virtual environment using VmWare software and the firewall host is running in **Bridged** mode. IP 192.168.46.134 was configured on the bridge co allow remote access from Linux for graphical use.



Firewall

- FreeBSD firewall was used for tests in this research because of:
  - Flexible rule set logic
  - First match wins
  - Packet accounting
  - Bridge mode
  - Open source
  - IPv6 support
- Ported to other operating systems
  - Mac OS
  - DragonFly
  - Windows

Results Capture



- IPFW-Graph tool was used to capture filtering actions on both rule sets.
  - Gives graphic output of packet matching in real-time.
  - Deny rules shown in red colour.
  - Allow rules shown in green colour.
  - Combines both in the all view.

Traffic Traversal

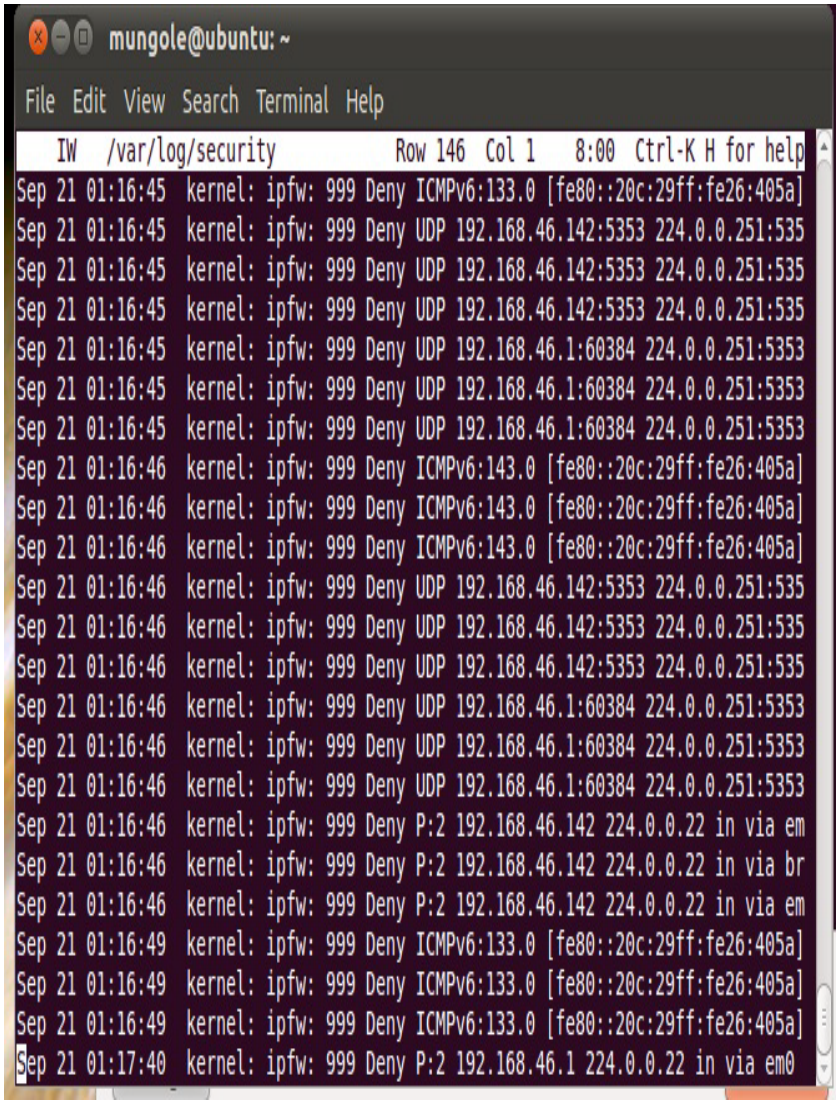
- Multi protocol pcap files were sent through the firewall host running in bridged mode – use two interfaces in different networks (Public and Private to simulate the ideal case).

- Test specific packets were injected into the firewall especially the ones for testing illegal traffic handling after optimizing.
  - # nemesis icmp –S 209.179.21.76 -D 192.168.46.134 -i 0 -c 0
  - The command above tests the firewall's deny action on icmp.

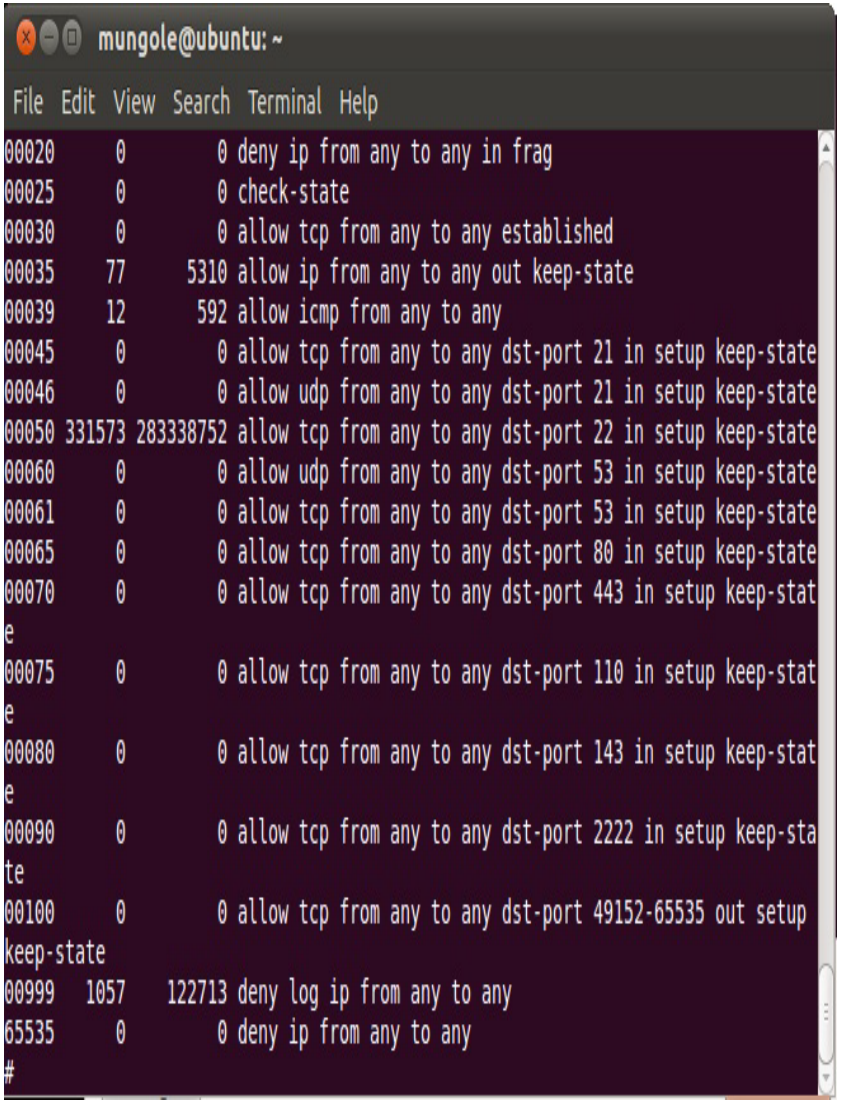
Rule Sets

- Default rule set
  - started with a rule set of 37 rules...
  - Passed traffic from different pcap through the firewall and collected matching statistics.
  - Not all rules matched packets inspected: counter =0.
  - Some rules did not match packets often.

Log file



Counters



Optimization

- Reducing the rule set size was performed based on the collected statistics.
- From the initial 37 rules, an optimized set of 13 rules was created.
- How - continued revision of rule set after several tests using pcap files.
  - Moved frequently matched rules forward
  - Removed non matching rules
  - Introduced ‘skip to’ to skip infrequently matched rules that could match traffic later.
  - Disabled rules for services not available
  - Removing overshadowed rules
  - Avoiding redundancy
  - Reintroducing rules for some denied packets based on their log count and how necessary they are deemed.

Performance Tests

Not standardized – firewall tests depend on what aspect is being measured

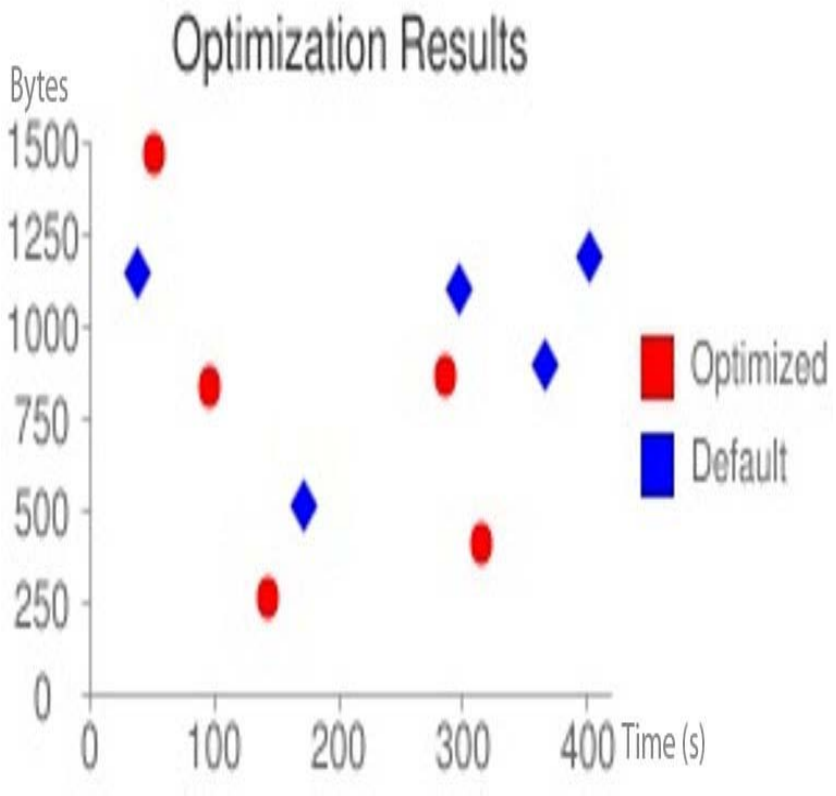
This research concerned itself with throughput based on rule set size and picked the following metrics as discussion points:

- **Connection establishment:** measured how fast the firewall created connections per packets inspected considering the number of rules
- **Forwarding rate** : compared the bits per second transmitted on both rule sets by measuring how long it took for a full pcap file to be inspected.
- **Connection teardown:** paired with connection establishment, the faster the tear down, the less rules a packet is inspected through to find a match.
- **Legal traffic**: checked if optimization denied allowed traffic or stopped certain services allowed from communicating by looking though log file entries.
- **Illegal traffic**: checked if optimization allowed false positives. Tools were used to inject packets denied by a given rule.

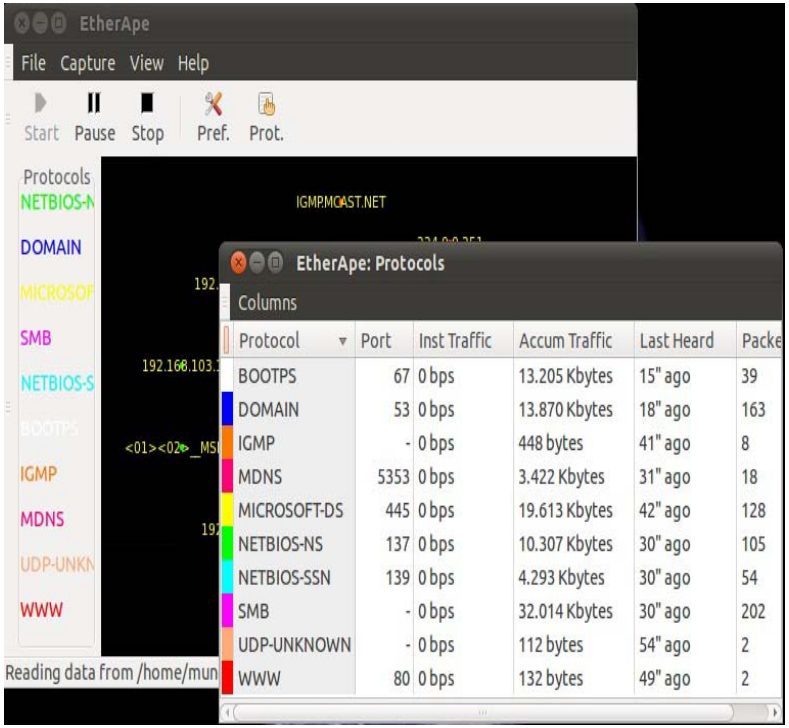
oThese values combined, were used to determine a gain in filtering speed : THROUGHPUT

Results

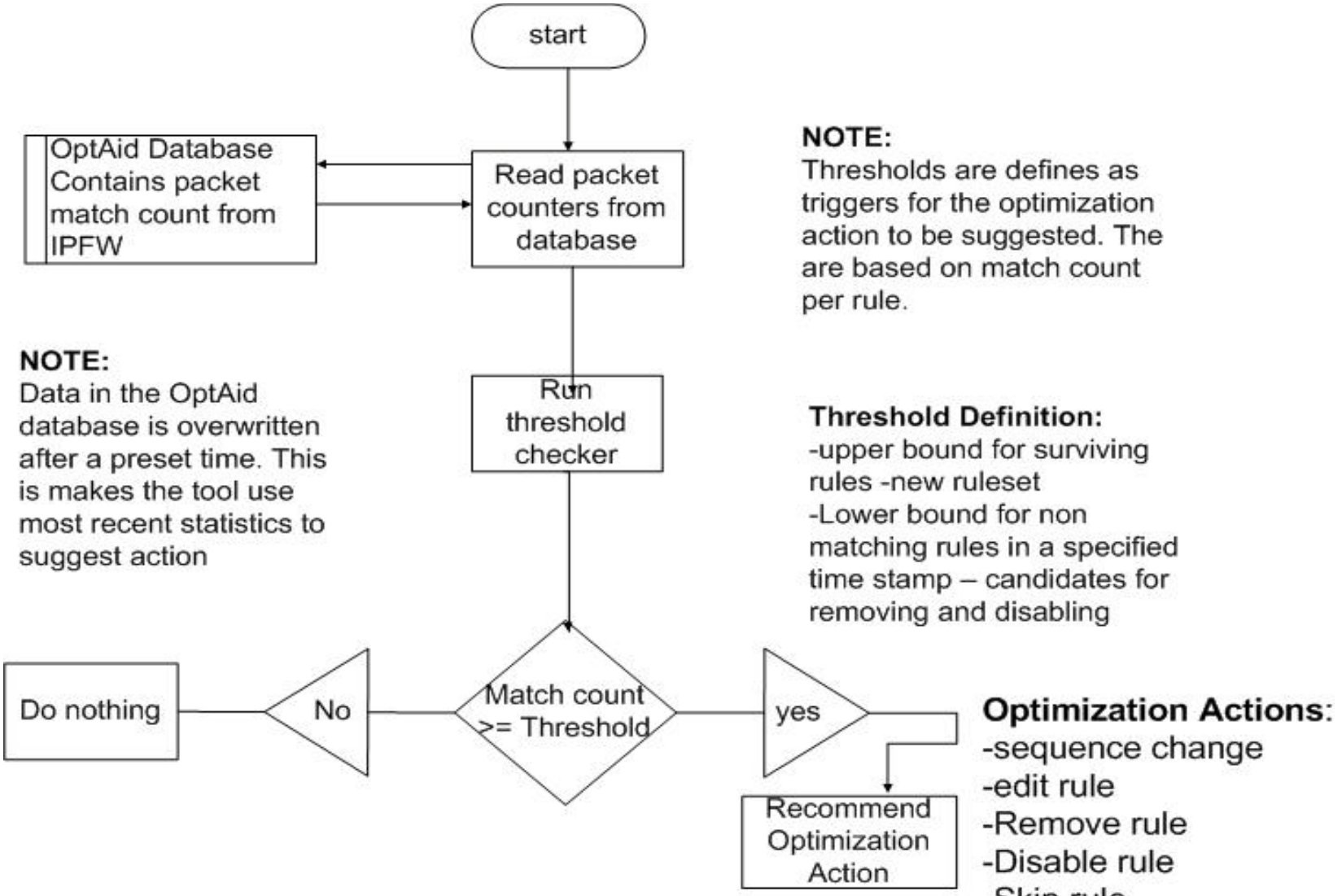
Tests done on 1500 bytes pcap file on both rule sets.



Protocol composition of pcap file used



OptAid: Tool Design



NOTE:
Thresholds are defines as triggers for the optimization action to be suggested. The are based on match count per rule.

NOTE:
Data in the OptAid database is overwritten after a preset time. This is makes the tool use most recent statistics to suggest action

Threshold Definition:
-upper bound for surviving rules -new ruleset
-Lower bound for non matching rules in a specified time stamp – candidates for removing and disabling

Optimization Actions:
-sequence change
-edit rule
-Remove rule
-Disable rule
-Skip rule

OptAid Tool Design

- Considered offline processing and suggesting optimization actions to the administrator.
- Design questions still to be answered before OptAid can be implemented:
  - Statistics aggregation vs. dynamic nature of traffic.
  - Defining triggering counter thresholds for ever changing traffic?
  - How often should statistics be sent to the database?
  - Do we discard rules completely?
  - How much overhead does all this introduce?
- Another consideration was to create a sub-rule-set dynamically. Issues faced and being investigated still are:
  - Processing complications with respect to picking rules for oncoming traffic.
  - How to work out and tell what flows to adapt for.
  - Issues with rule numbering when re-sequencing.
  - How to apply the sub-rule set to traffic.

Conclusion

- Based on the evidence obtained from the tests conducted in this research, it can be concluded that:
  - There is a gain in firewall filtering performance by reducing the rule set. Less rules a checked through sequentially and a match is found faster.
  - The dynamic nature of traffic offers challenges in adapting rule sets to packet flows. This is why it has generally been described as NP-Hard.
  - Care must be taken not to open the network to illegal traffic or denying legitimate traffic by locking the system out when optimizing.
  - Packet filtering is a crucial component of network security and thus needs better methods to avoid wasting of device and network resources.
  - Better filtering methods and approaches are necessary to make firewalls match up to the improved transmission technologies and network speeds.

Future Work

- Investigate more on rule set optimization – technology changes every second.
- Implement OptAid tool having understood and demonstrated matching patterns.
- Investigate further, performance implications with regard to dynamic sub rule set creation from the larger set.
- Add dynamic rule set adaptation to OptAid once performance issues are addressed.

Thank you for your attendance