# Framework for Evaluation of Network-Based Intrusion Detection System

Owen Lo. Beng (Hons) Computer Networks and Distributed Systems 05002961@napier.ac.uk
Supervisor: Prof W. Buchanan.   Second Marker: Dr Jamie Graves.

## 1. Introduction

In testing of IDSs, no accepted standard exists for defining the procedures or metrics of measurement for evaluation.

This project provides a methodology which takes into consideration the types of testing and metrics of evaluation required in order to assess performance of ID systems based on the dynamic nature of network traffic.

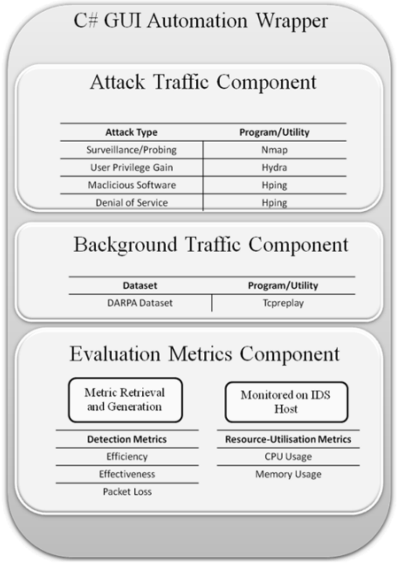
## 2. Literature Review

Research was undertaken which first provided justification for the need of security in computing. A review of the taxonomy and existing methodologies of evaluating ID systems along with the metrics of measurement which are used was then carried out. It was concluded that there are three main requirements in carrying out a effective evaluation:

(1) **Inclusion of Realistic Attack and Benign Network Traffic**
(2) **Ease of Automation**
(3) **Inclusion of Meaningful Metrics for Evaluation**

## 3. Design and Methodology

A prototype framework was developed to meet all three requirements listed in the literature review. The framework and schematic is shown below:

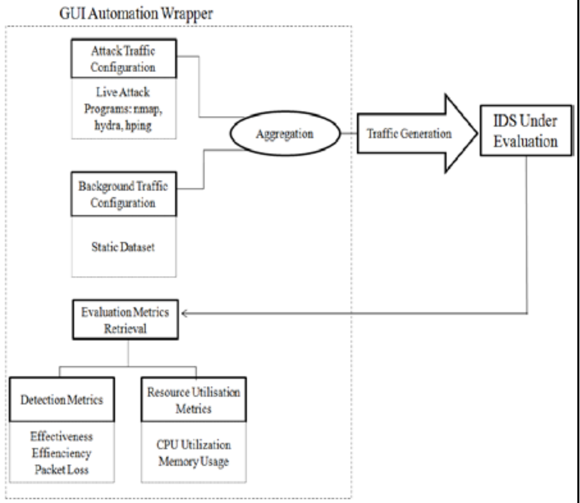

C# GUI Automation Wrapper
Attack Traffic Component
Background Traffic Component
Evaluation Metrics Component
Metric Retrieval and Generation
Monitored on IDS Host

| Attack Type | Program/Utility |
|---|---|
| Surveillance/Probing | Nmap |
| User Privilege Gain | Hydra |
| Maclicious Software | Hping |
| Denial of Service | Hping |

| Dataset | Program/Utility |
|---|---|
| DARPA Dataset | Tcpreplay |

| Detection Metrics |
|---|
| Efficiency |
| Effectiveness |
| Packet Loss |

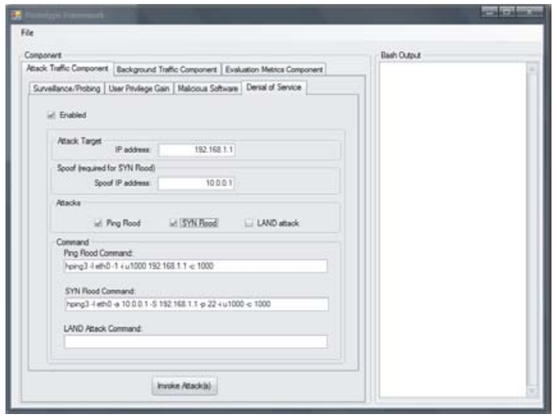
| Resource-Utilisation Metrics |
|---|
| CPU Usage |
| Memory Usage |



A framework was created which consists of three components: **Attack Traffic**, **Background Traffic** and **Evaluation Metrics**. The attack and background traffic component aims to meet the first criterion (realistic traffic) whilst the Evaluation Metrics component meets the third criterion. Finally, a graphical user interface was developed which wraps the three components together in order to achieve both ease of use and, more importantly, automation of testing.

## 4. Implementation

The prototype was developed using Microsoft .NET C# and runs in a Linux environment using Mono (CLR for Linux/Unix/BSD/Mac).



☑ Enabled

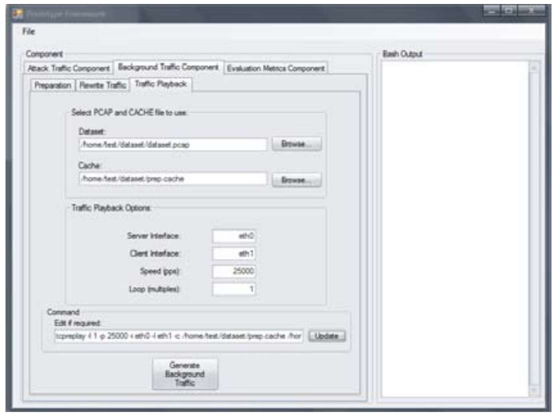
Attack Target
IP address: 192.168.1.1

Spoof (required for SYN Flood)
Spoof IP address: 10.0.0.1

Attacks
☑ Ping Flood ☑ SYN Flood ☐ LAND attack

Command
Ping Flood Command:
```
hping3 -I eth0 -1 -i u1000 192.168.1.1 -c 1000
```
SYN Flood Command:
```
hping3 -I eth0 -a 10.0.0.1 -S 192.168.1.1 -p 22 -i u1000 -c 1000
```
LAND Attack Command:

**Attack Traffic Component**

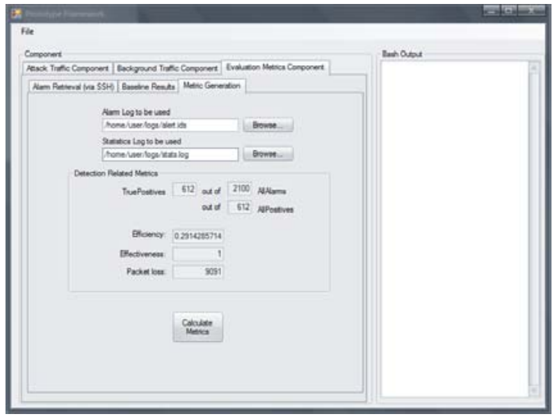


Select PCAP and CACHE file to use:
Dataset: /home/test/dataset/dataset.pcap
Cache: /home/test/dataset/prep.cache

Traffic Playback Options
Server Interface: eth0
Client Interface: eth1
Speed (pps): 25000
Loop (multiples): 1

Command
Edit if required:
```
tcpreplay -l 1 -p 25000 -i eth0 -I eth1 -c /home/test/dataset/prep.cache /hor
```

**Background Traffic Component**



Alarm Log to be used: /home/user/logs/alert.ids
Statistics Log to be used: /home/user/logs/stats.log

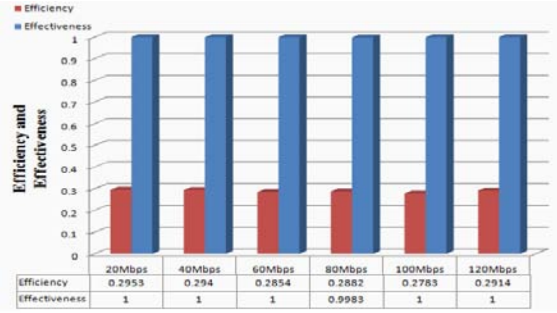
Detection Related Metrics
TruePositives 612 out of 2100 AllAlarms
out of 612 AllPositives
Efficiency: 0.2914285714
Effectiveness: 1
Packet loss: 9091

**Evaluation Metrics Component**

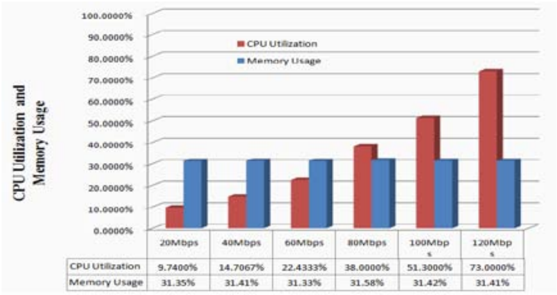
## 5. Results and Conclusion

Using the framework, the evaluation was carried on the NIDS known as Snort. Snort was chosen due to it's popularity and being freely avaliable.

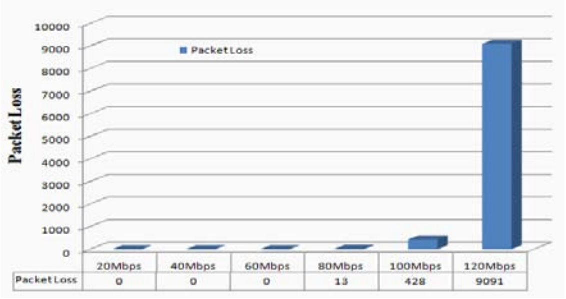
Metrics of evaluation included Efficiency (True-Positive/AllAlarms), Effectiveness (True-Positive/AllPositives). Packet loss, CPU utilisation and Memory usage was also monitored.



| | 20Mbps | 40Mbps | 60Mbps | 80Mbps | 100Mbps | 120Mbps |
|---|---|---|---|---|---|---|
| Efficiency | 0.2953 | 0.294 | 0.2854 | 0.2882 | 0.2783 | 0.2914 |
| Effectiveness | 1 | 1 | 1 | 0.9983 | 1 | 1 |



| | 20Mbps | 40Mbps | 60Mbps | 80Mbps | 100Mbps | 120Mbps |
|---|---|---|---|---|---|---|
| CPU Utilization | 9.7400% | 14.7067% | 22.4333% | 38.0000% | 51.3000% | 73.0000% |
| Memory Usage | 31.35% | 31.41% | 31.33% | 31.58% | 31.42% | 31.41% |



| | 20Mbps | 40Mbps | 60Mbps | 80Mbps | 100Mbps | 120Mbps |
|---|---|---|---|---|---|---|
| Packet Loss | 0 | 0 | 0 | 13 | 428 | 9091 |

The prototype has shown that in the evaluation of Snort, the greater the volume of network traffic the higher the CPU utilisation. This results in packet loss at ≥ 80Mbps playback speeds. Furthermore, the prototype has also shown that Snort is highly effective in logging attacks but, at the same time, does raise a lot of false-positives. It was concluded that the prototype meets the initial aim of this project but further work in this area must still be carried out.