# Virus/Malicious file detection in a shared environment

## Problem Statement

Sharing environment consists an interconnection of two or more computers for communication purpose. Communication can be in many forms such as the distribution of information/ data from a device to another. Organization stores and link data from one location to another with the help of network. This data or information is distributed within networks or from one network environment to another in a different location with the help of the Internet

Malicious activities (malcodes) are self-replicating malware and a major security threat in a network environment. Timely detection and system alert flags are very essential to prevent rapid malcodes spreading in the network. The difficulty in detecting malcodes is that they evolve over time. Despite the fact that signature-based tools, are generally used to secure systems, signature-based malcode detectors neglect to recognize muddled and beforehand concealed malcode executables. Automatic signature generation systems has likewise been use to address the issue of malcodes, yet there are many works required for good detection. Base on the behavior way of malcodes, a behavior approach is required for such detection. Specifically, we require a dynamic investigation and behavior Rule Base system that distinguishes malcodes without erroneously block legitimate traffic or increase false alarms.

Security is an essential part of computer networks, as intrusion attack is a threat to network communication. Network intrusion happens when an unauthorized system/ user gain access into a network system and manipulates data or information. According to Fortinet (2017), There are 184 billion total exploit detections, 1.8 billion average daily attack volume, 6,298 unique exploit detections, Exploit volume per firm averaged 2.5 million, with a median of 456, and 69% of firms saw severe attacks”

## Background

Many research works have been done in the area of malware signature generation, malware clustering and classification or detection of malware and virus attacks using different techniques apart from signature.

Current research focus on improvement of intrusion signature generation for Intrusion Detection, as they ignore the identification of the attack features/Indicators that shows the attack operation or life cycle such as how the attack has operated in the past, similar features of such attack, as the absent of such information will leave a gap in good detection of the attack or similar attack in future. Signature-based Intrusion detection technique faces a challenge in high accuracy detection, due to its approach of detection. This technique uses static approach and intrusion signatures must exist in the Detection System database in other to enable detection of attack.

There are many research and algorithms are proposed to Intrusion and virus Detection

1. Signature Generation using String Algorithm Reshma & Chairag (2011)
2. Intrusion Detection using Machine learning
3. Signature Generation using machine learning Algorithm
4. Clustering and Classification on malware data

## Methodology

This section discusses the proposed solution on traditional/static approach for Malcode detection using intrusion detection system. Malicious code type of attack is unpredictable and dynamic, so a good study and techniques are needed to measure and create an advance way to detect and prevent the network from been compromised. This framework as presented in Figure 1 shows the steps and techniques to detect, collect data, analysis, evaluate and validate the work.

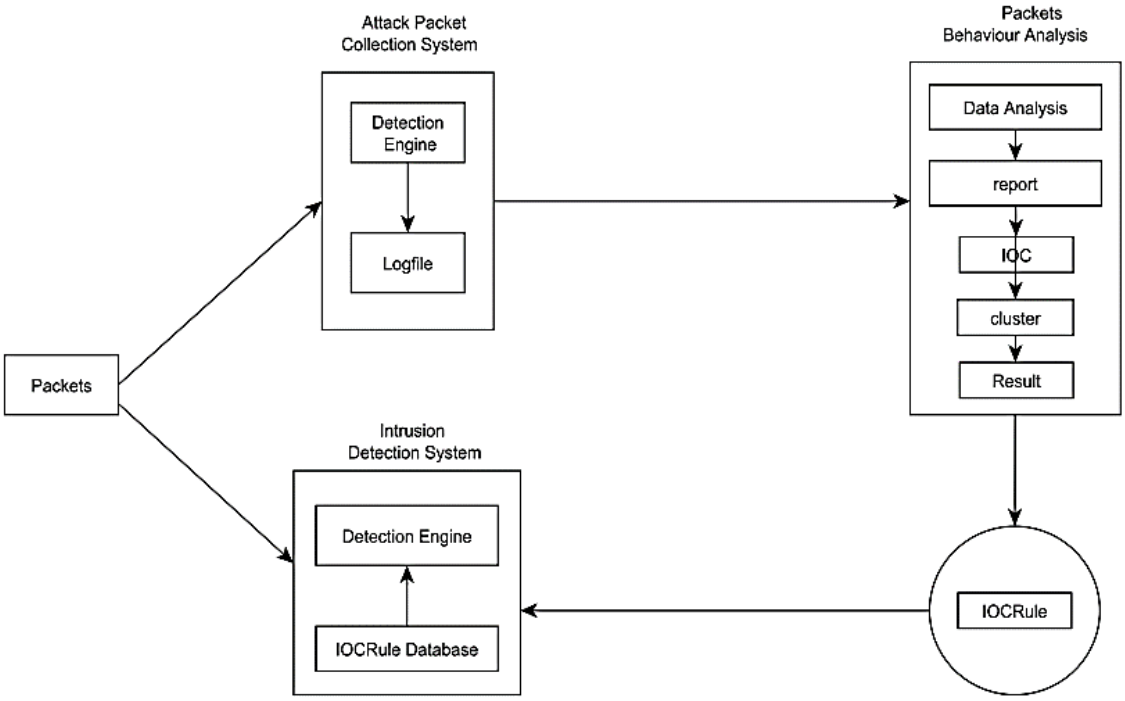


Fig: 1

Network Intrusion Detection System (NIDS): NIDS is a system that monitors and detect network intrusion activities, misuse of systems from internal or external networks. The NIDS monitors networks and servers to detect intrusion activities. has the capacity to analyze real-time traffic and data flow in network. It is a small, lightweight IDS written by Martin Roesch. In project we will be use to evaluate the performance of IOCRule over the traditional detection technique. Basically it consists of the following major components as shown in fig 2.

1) packet decoder: The packet decoder collects packet from different network interfaces which might be Ethernet and SLIP and then send to detection engine.

2) Pre-processors: The Pre-processors work with snort to arrange the packet before detection engine apply some operation if packet is corrupted. Basically it matches pattern of string, by sequence changing, because by adding some extra value and intruder can fool the IDS but the pre-processor rearranges the string and IDS detect the string.

3) Rules: This section is the IOCRule which is the main contribution in this paper. The new IOCrules will add more detection effeicent to Snort IDS and a low false alarm. The IOCRule are incremented into the detection engine which house the Rule database.

4) detection engine: The detection engine finds out intrusion activity in a packet with the help of snort rules and if found appropriate rule action is applying otherwise it drops the packet.

5) Logging and Alerting: Logging and Alerting System work with what the detection engine finds in the packet and it might generate an alert or log activity. All log files are kept under /var/log/snort folder by default. 6) Output modules: Output modules save output generated by the logging and alerting system.

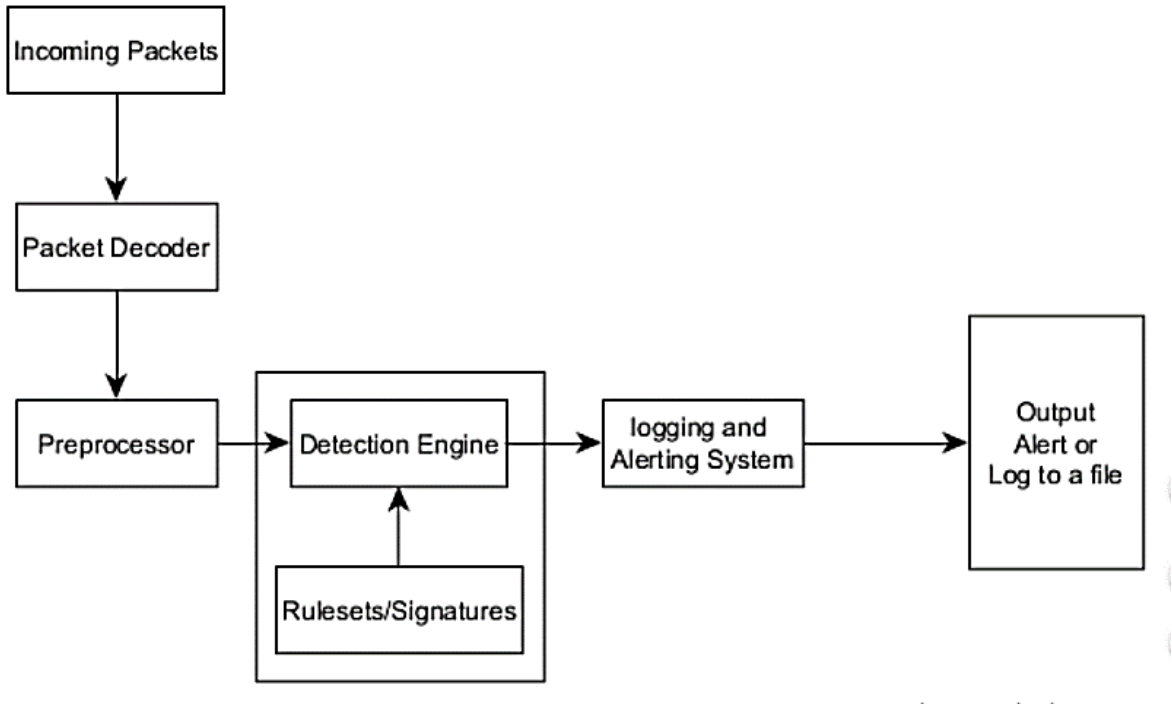


**Fig 2:**

**Experimental Design**

This section explains the evaluation of the baseline study using two different datasets (CTU and Darpa Dataset). CTU dataset is a 2016/2017 network intrusion dataset and Darpa is a 1999 network intrusion dataset. To perform this evaluation study, Snort IDS is configured and setup to analysis incoming and outgoing traffic in the setup network environment for this research. Snort reads every traffic and records numbers of packets that passed through the detection/analysis engine. Each of the datasets will be injected into the snort IDS and run on the eth1 (Ethernet interface). In this thesis, Snort uses the eth1 interface as its packet receiving interface. That's to say, packets go in and out through the interface.

1) Initiate Snort IDS Process and Import Dataset: This stage is the start up of Snort to become active to receive commands and instruction from the terminal window. After the start up Snort, the Datasets are imported into a directory in Snort.

2) Establish Connection and Execute the Dataset: In other for snort to analyze and detect incoming traffic/packets, connection at the Ethernet interface need to be established. After the connection is established, the Dataset is executed in the stored directory. The execution of the

dataset active the dataset to start running as an incoming traffic through the Ethernet interface and snort start analyzing and its detection process.

3) Log Result: The detection result is log into the intrusion alert logfile database.

4) Repeat the Process: Each dataset evaluation was done up to 3 times in other to arrive at a conclusion base on the result achieved.

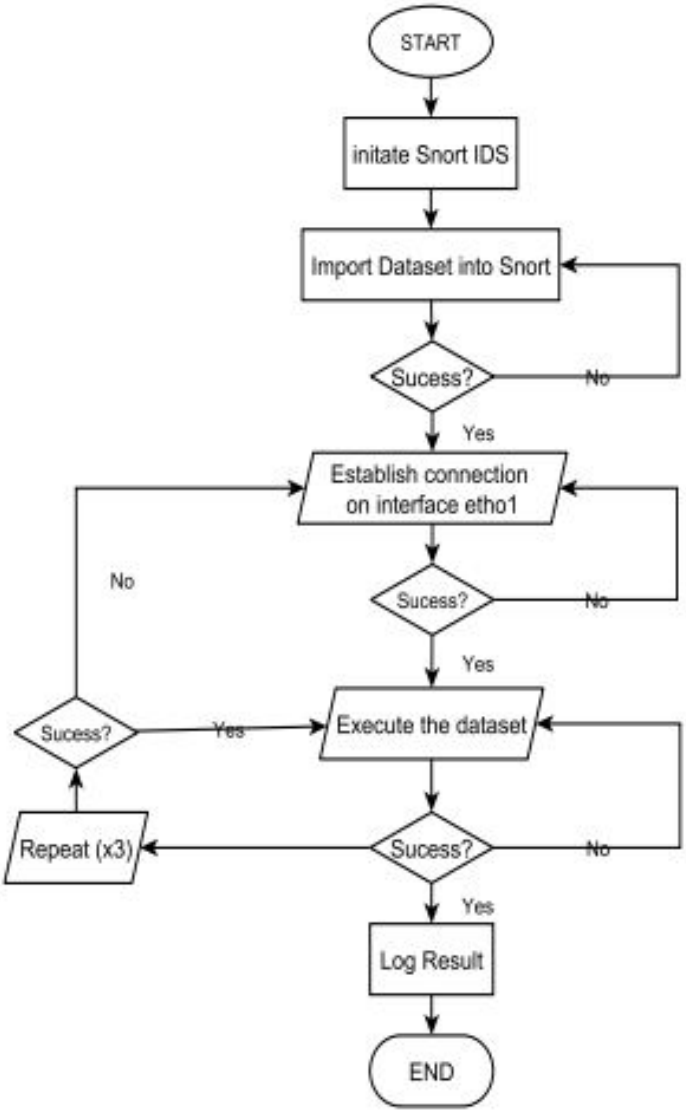


**Fig 3:**