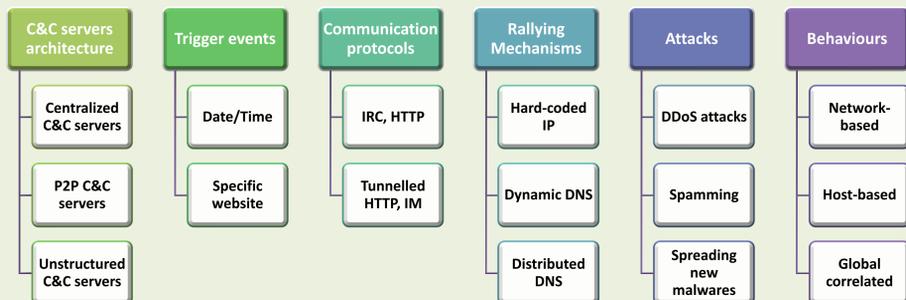


# Experimental Host- and Network-based Analyser and Detector for Botnets

## Background

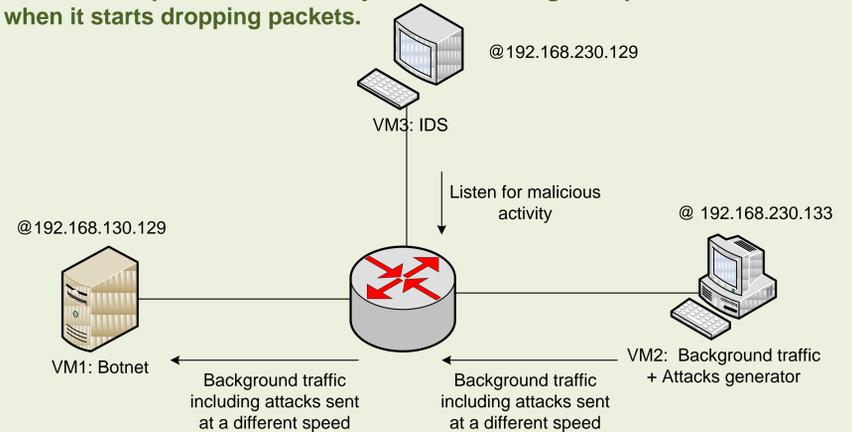
The first Botnet "PrettyPark" was introduced in 1999 on the Internet Relay Chat (IRC), designed for group communication in discussion forums, known as channels. The major function of PrettyPark was to allow an administrator to remotely control a large pool of computers using IRC channels. The idea became popular in the cyber-crime community and over the years Bots have been improved and dedicated to cyber-attacks.

## Literature Review



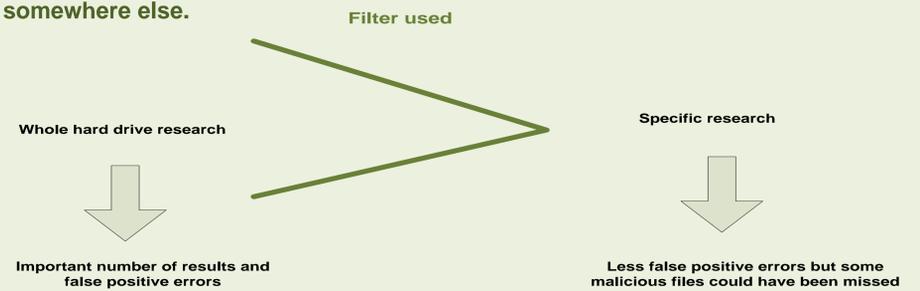
## Test bed implementation

The network traffic will be sent at a different speed using background traffic and attacks. An IDS is listening to the network activity in order to detect these attacks. The aim of the experiment is to analyse at which range of speed the IDS is efficient and when it starts dropping packets.

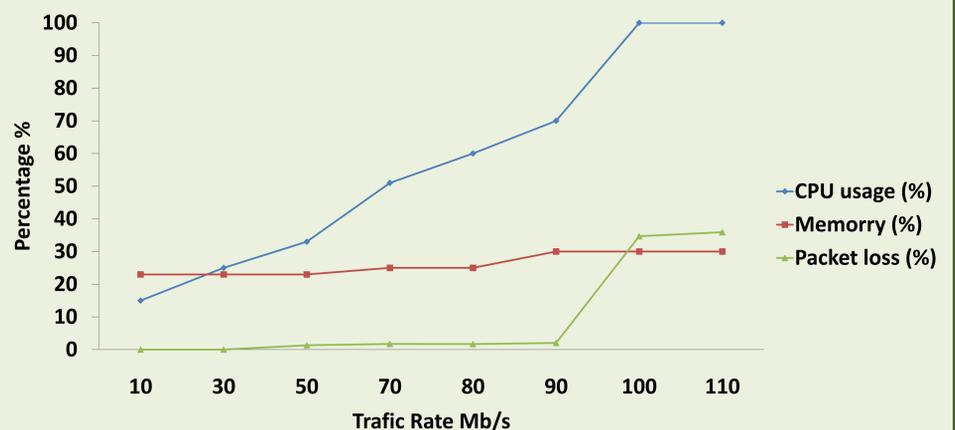


## Results

The host-based experiment has logged an important number of false positive results using Filewatcher. and Processmonitor. Both programmes were able to detect everything going on in the system. But, in respect to the density of files, which were constantly created and modified by the operating system, it was not possible to trigger a file which came from a malicious activity. However, this operation can become more efficient by applying filters inside the Filewatcher, for example, filters triggering modified/created files in a specific folder rather than the whole hard drive. Therefore, the number of false positive errors are reducing because the research of the files activity is more specific but, in another way, the malicious attacks could not have been found at all if the files were saved somewhere else.



On the network side, the IDS showed that under 30Mb/s of all the packets were correctly inspected. However, from 30 to 90Mb/s, there was an average of 2% of the packets that dropped, and after 90Mb/s the number of dropped packets increased to an average of 35%. In conclusion, the IDS was reliable under 30Mb/s, semi-reliable from 30 to 90Mb/s and unreliable after 90Mb/s.



## Conclusion

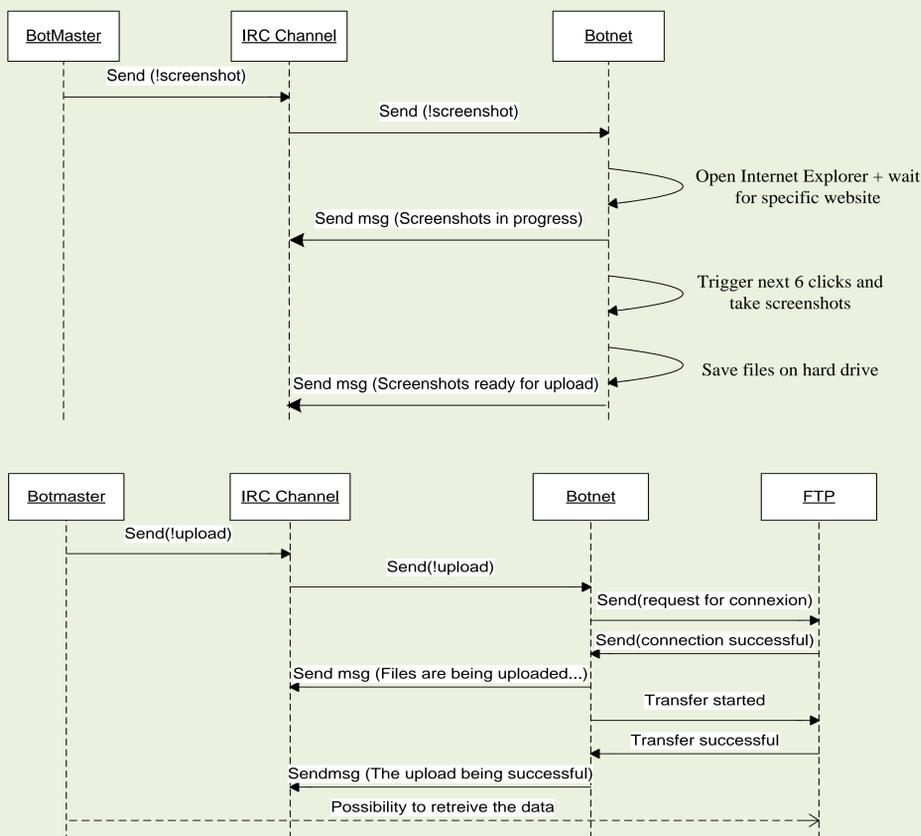
The density of files constantly created on the hard-drive have made the host-based detection fail. The Network-based detection was successful and 0% of the packets were dropped under 30 Mb/s. Therefore, the IDS had analysed correctly all the packets and detected all the attacks. From 30Mb/s to 90Mb/s, less than 5% of the packets were dropped and after 90Mb/s, the amount of dropped packets jump to more than 35%. Any of these dropped packets could have come from an attack which make the IDS un-reliable after 90Mb/s and discussed under 90Mb/s. However, these results can be explained by watching the CPU utilisation during the experiment: when the CPU utilisation raised to 100%, it was normal that the IDS could not analyse all the packets.

## Future Work

The experiment uses Virtual Machines with a CPU shared at 2.0Ghz. Other experiment could be carried on using different type of CPU speeds. This experiment can be useful to companies who do not know how powerful the IDS configuration needs to be if they want to have the maximum efficiency (i.e. no dropped packets) or, if they only need a semi-reliable IDS.

## Design – Botnet Overview

The first aim of this project is to create a script able to mimic a Botnet. The next UML diagrams show the 2 modules included in the Botnet:



## Design – Detection Overview

The second aim of this project is to design the detection/evaluation tools associated to the Botnet. The Botnet will be placed in a Blackbox, which includes multiple security tools. The network activity of the Bot (communication between the Bot and the IRC sever/FTP server) should be detected using Snort and the Host activity (creation of screenshots saved on the hard drive) should be detected using Filewatcher and Processmonitor.

