Entropy – score: A Method to Detect DDoS Attack and Flash Crowd

Design Review

Assigned By:

Eyal Zvi
Sean Etinger

Contents

1. Introduction

1.1 Definitions and abbreviations

Abbreviations:

DDoS – Distributed Denial of Service**.**

FC – Flash Crowd.

Definitions:

FC - An occurrence of many legitimate users accessing a server simultaneously.

Entropy(H) - Measurement of the randomness of the samples.

$p(x_i)$ – Probability of occurrence of the i^{th} sample.

X – Total sample space.

G_i - The i^{th} group

$P(x_i)$ – Current probability of occurrence of the group.

$P(y_i)$ – Stored probability of occurrence of the group.

S_i – The score of the i^{th} group

p_i – The probability of the i^{th} packet.

Th – A predefined threshold value for a normal operation.

T – A calculated threshold using packet scoring method.

Ψ – The probability of current traffic.

Φ – The probability of excitable traffic.

H – The entropy of the system.

g – Number of groups.

n – Number of packets received.

1.2 Overview

DDoS attacks have been affecting the IT industry for many years, attackers are using different attack tools to generate malicious traffic that can affect a victim's system or network by creating heavy congestion and as a result high latency that causes delay in network speed and may even result in network crashes.
There are many defense techniques against DDoS attacks like attack prevention, attack filtering and attack trace back, but the problem with these previously defined methods is that they rely on specific characteristics of incoming packets for DDoS identification.
These basic characteristics are almost identical in DDoS attack and FC, making it very difficult differentiating between them, both in DDoS attacks and in FC, a lot of request packets are generated resulting in slow response times or packet drops at the destination point.
But, besides the fact that differentiating between DDoS and FC is difficult, nowadays attackers can easily imitate these characteristics and bypass the filtering methods, therefore a new solution is needed.
This paper presents an algorithm that is capable of differentiating between DDoS attacks and FC using entropy calculations and packet scoring rather than using packet characteristics. Packet scoring method is used to analyze each packet or group of packets according to their score values and entropy based method is used to categorize an entire group of packets according to their entropy values.

1.3 Goals of the simulation

The goals of the simulation are as follows:

By simulating the use of the algorithm we would like to test and conclude whether or not it categorizes DDoS, FC, and normal hosts correctly.
By examining the simulation results we would like to be able to decide whether the theoretical results we would have expected to get match the simulation's actual results.
After results and data analysis we would like to conclude if this algorithm is more or less efficient than previously mentioned algorithms.

1.4 Assumptions

The assumptions assumed in the simulation are:

-Combining entropy and packet score is enough to differentiate host types.
-All IP addresses are chosen at random, and each host is assigned with an address according to its type (Normal/DDoS/FC).
-Th threshold is predefined to fit the given Example in the paper.
-Each group of hosts has a predefined stored probability of occurrence $P(y_i)$ assigned to it according to its type.

2. Design and Flow

2.1 Overview

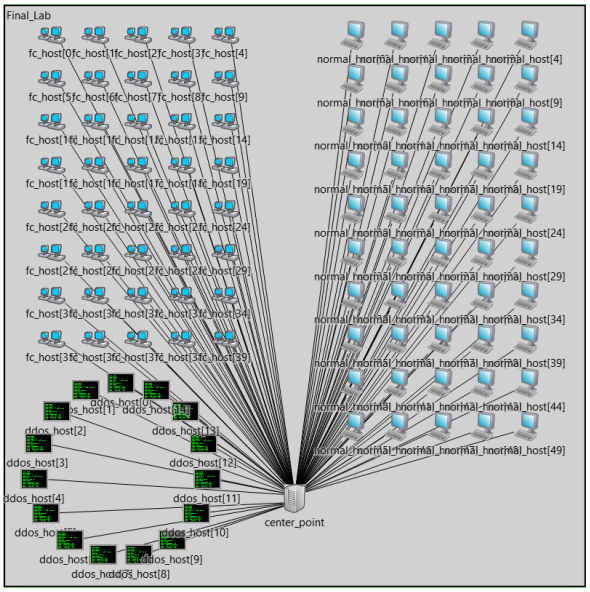
The system is built based on a given topology, there are 50 normal hosts, 15 DDoS hosts, 40 FC hosts and a data collecting server which is named 'Center Point'.
Each of these are made using a simple module, and will be described below:

Normal host – A simple module that represents a host that is considered to be acting normal, meaning that if an entropy H is to be calculated on a group G_i exclusively made of normal hosts, H would not surpass Th threshold and the group will be considered as normal traffic and will not be categorized as DDoS nor FC.

DDoS host – A simple module that represents a harmful attacker that tries to overflow the server with traffic and cause damage to it.
Given that a group G_i contains a DDoS host, if an entropy H is to be calculated on that group, H will surpass Th. Furthermore, if a group score S_i is to be calculated it will also surpass its threshold T, classifying it as a group that has a DDoS host in it, as expected.

Flash crowd host – A simple module that represents one of many legitimate clients trying to connect to a server simultaneously, usually as a cause of some general event.
Given that a group G_i contains a FC host, if an entropy H is to be calculated on that group, H will surpass Th, but if a group score S_i is to be calculated it will not surpass its threshold T, classifying it as a group containing FC.

Center Point – A simple module that represents the server that the hosts are sending messages to, it is connected to each host individually.
In the simulation it is responsible for collecting data, analyzing it, and classifying groups as Normal, DDoS or FC.



Modules overview

This part details the Omnet modules and emphasis the section in which the statistics gathering is taking place. As stated above all the statistics gathering will be implemented in the Center Point module.

Simple Module: Normal_Host
Description: Represents a host that is considered to be acting normal.
Variables: num_of_n_hosts – Number of normal hosts in the topology (50).
Functions: **initialize()** – Initialize normal host parameters and start event.
handleMessage(cMessage* msg) – send normal messages to Center Point.
send_msg() - Generate a normal message and send to Center Point.

Simple Module: DDoS_Host
Description: Represents a host that is considered to be malicious.
Variables: num_of_ddos_hosts – Number of DDoS hosts in the topology (15).
Functions: **initialize()** – Initialize DDoS host parameters and start event.
handleMessage(cMessage* msg) – send DDoS messages to Center Point.
send_msg() - Generate a DDoS message and send to Center Point.

Simple Module: Flash_Crowd_Host
Description: Represents a host that is considered to be a part of a FC group.
Variables: num_of_fc_hosts – Number of FC hosts in the topology (40).
Functions: **initialize()** – Initialize FC host parameters and start event.
handleMessage(cMessage* msg) – send FC messages to Center Point.
send_msg() - Generate an FC message and send to Center Point.

Simple Module: Center_Point
Description: Represents a message receiving server.
Variables: num_of_n_hosts – Number of normal hosts in the topology (50).
num_of_ddos_hosts – Number of DDoS hosts in the topology (15).
num_of_fc_hosts – Number of FC hosts in the topology (40).
Functions: **initialize()** – Create random IPs for all hosts, send the IPs to them.
Generate statistical vector to sample the system's parameters.
handleMessage(cMessage* msg) – Paper's algorithm implementation.
send_ip() – Create new IP_Manage packet with a random IP and send it to the destination host.
system_entropy() – Calculate the system's entropy.
active_groups() – Check how many groups are currently active.
assign_to_group(int src_ip) – Assign a packet to it's relevant group.
update_scores() – Update all packet scores and probabilities.
data_collection(float entropy, int active_g) – Collect statistical data.
detection_summary(struct group, int detection_num) – Inc detection parameters for later evaluation.
finish() – Print a summary of the detection's accuracy.

Compound Module: Final_Lab
Description: Contains all 106 simple modules and defines the network topology.

2.2 Application Interfaces

The parameters defined by omnet.ini for each run:

num_of_n_hosts – Number of normal hosts in the topology. Default value of 50.

num_of_ddos_hosts – Number of DDoS hosts in the topology. Default value of 15.

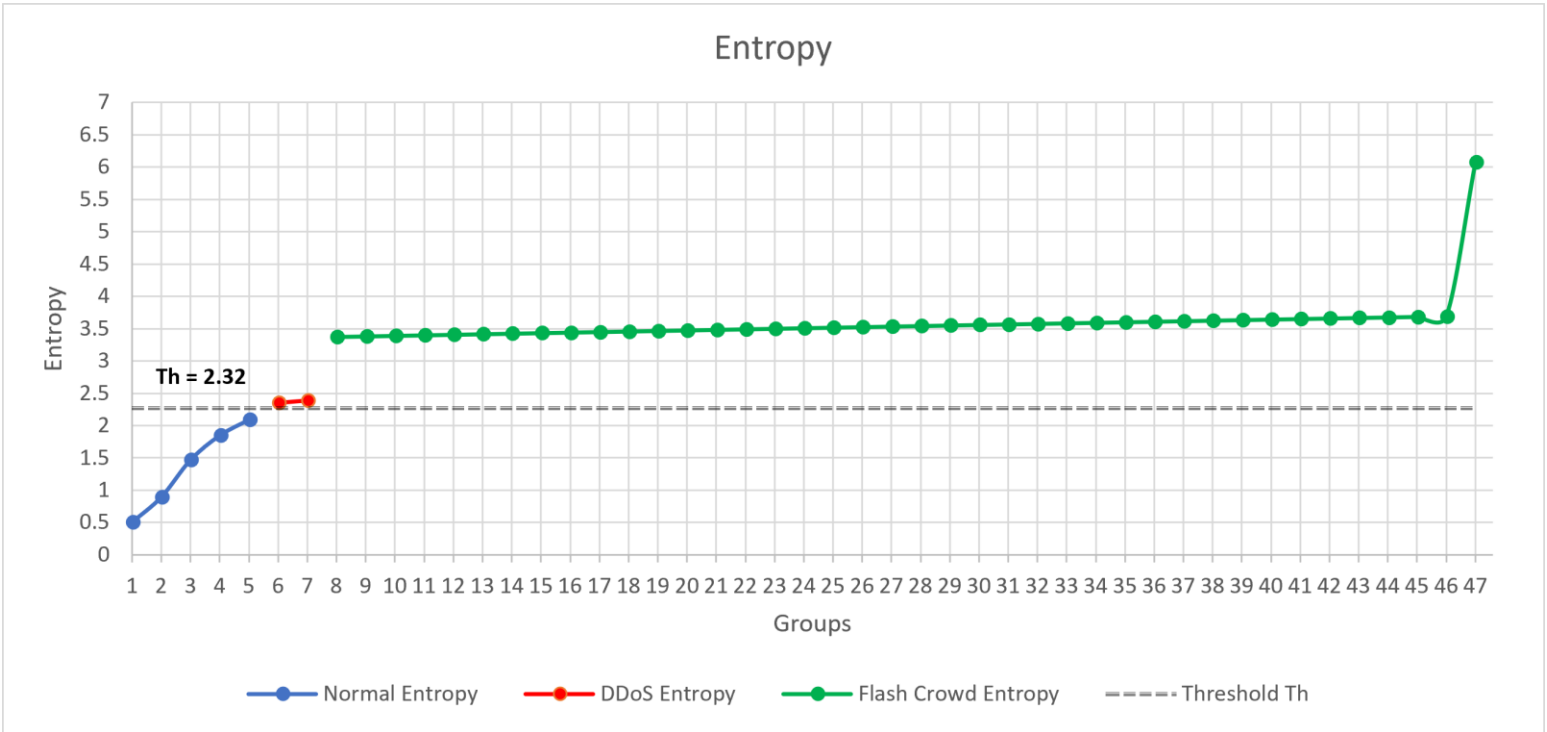
num_of_fc_hosts – Number of FC hosts in the topology. Default value of 40.

There are no parameters that are defined by the user.

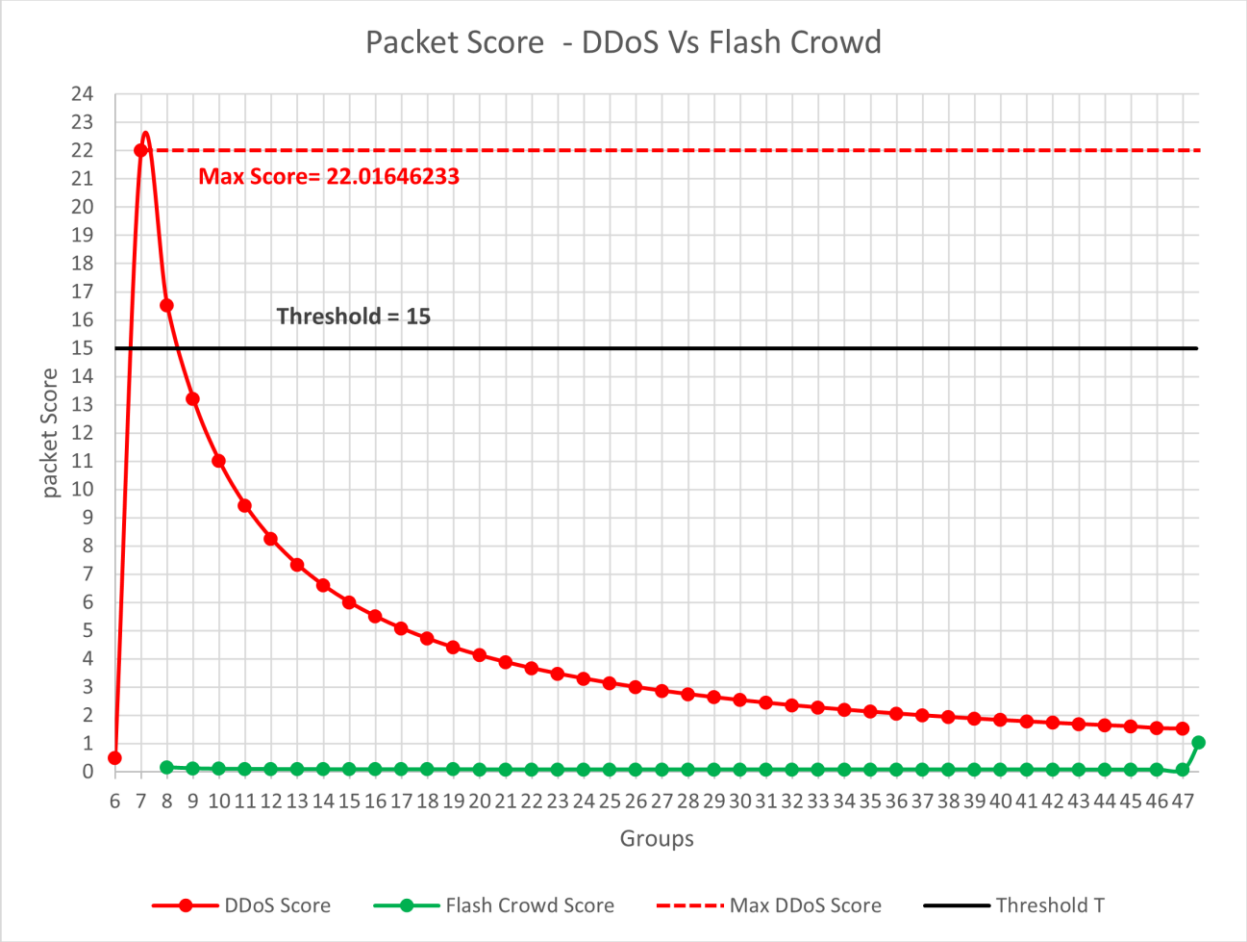
2.3 Results and Test measurements

In our experiment we send messages to a center point from different types of hosts, simulating a scenario where a server needs to be able to differentiate between malicious and legitimate traffic.

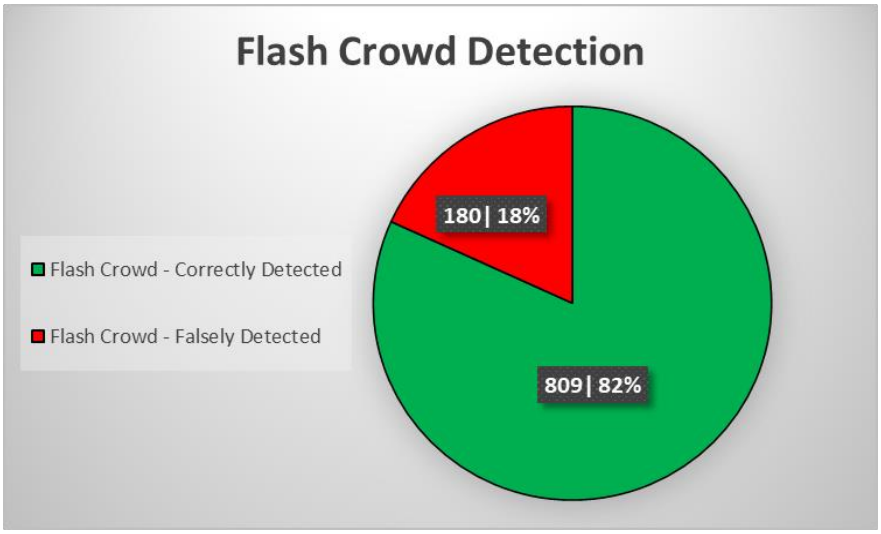
As seen in the Entropy statistics graph below, DDoS attack packets and FC packets clearly overtake the normal traffic entropy-wise, and the predefined threshold Th can efficiently differentiate between normal traffic and DDoS or FC packets.

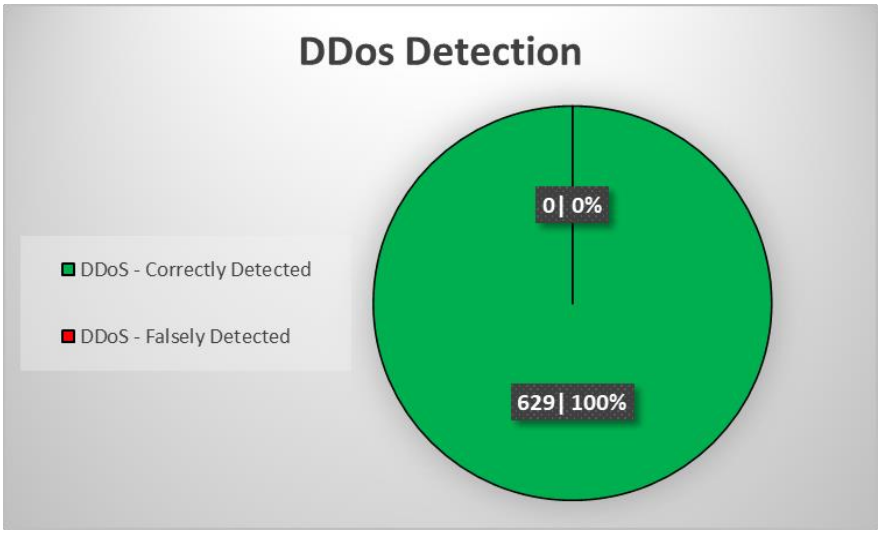


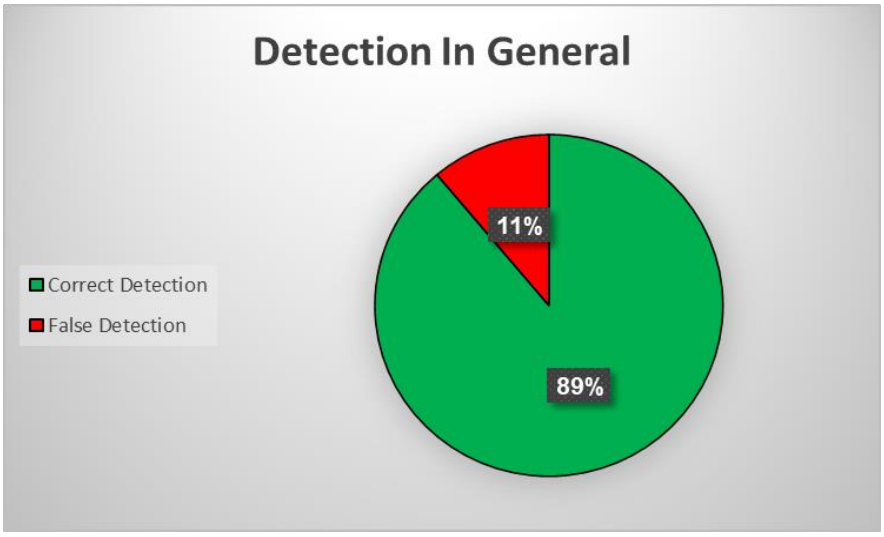
Furthermore, as we see in the Packet Score statistics graph below, for DDoS packets the score peaks at 7 groups and then starts to gradually decrease, although In our algorithm we do not update packet score for the DDoS packet after reaching the Max Score seen below. As for the FC packets, we see a very low valued graph, significantly smaller values than the DDoS one.
By setting the T threshold to be 15 as stated in paper we can most likely say that for most of the time, we can efficiently differentiate between DDoS and FC packet using packet scoring method.



In out simulation we were also able to calculate the detection error as seen in the following graphs:







Summary:

As we can see from our results of 89% decision correctness, we can very efficiently differentiate between DDOS attacks Flash Crowd, also, normal traffic is never mistaken for both.

As seen in our results, the algorithm has an extremely low False – Positive rate, 0% in our simulation.

Also, it is apparent that this algorithm has a very good space complexity compared to other DDoS detection algorithms.