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

By Akshat Gaurav & Awadhesh Kumar Singh

Final report

Tslil Greenberg - 313347841

Hila Gurevich - 204434096

Contents

Preface 1

[Contents 2](#_Toc261787191)

[1. Introduction 1-1](#_Toc261787192)

[1.1 Definitions and abbreviations 1-1](#_Toc261787193)

[1.2 Overview 1-1](#_Toc261787194)

[1.3 Goals of the simulation 1-1](#_Toc261787195)

[1.4 Assumptions 1-1](#_Toc261787196)

[2. Design and Flow 2-2](#_Toc261787197)

[2.1 Overview 2-2](#_Toc261787198)

[2.2 Modules overview 2-2](#_Toc261787199)

[2.3 Application Interfaces 2-2](#_Toc261787200)

[2.4 Results and Test measurements 2-2](#_Toc261787201)

# Introduction

## Definitions

**DDoS** (distributed denial-of-service) **attack** - is a cyber-attack in which the incoming traffic flooding the victim originates from many different sources. This effectively makes it impossible to stop the attack simply by blocking a single source.

Flash Crowd - A flash event is a large increase in traffic to a particular Web site causing a dramatic growth in server load and putting severe and network links under "pressure", leading to the server, which results in considerable increase in packet loss and congestion

Entropy - The degree of disorder or uncertainty in a system.

## Overview

## The article describes attacks that are using different tools to generate malicious traffic which can affect the victim’s system or its network. There are different methods to detect DDoS attacks, but those methods cannot espy the difference between flash crowd and DDoS attacks.

## Goals of the simulation

To detect DDoS and Flash crowd attacks, using entropy calculation and scoring method.

## Assumptions

|  |  |  |  |
| --- | --- | --- | --- |
| **1st** | **2nd** | **3rd** | **4th** |
| **192** | **168** | **072** | **000** |

* **IP addresses** – it’s not specified how IP’s are distributed in the different operations.  
  **In normal** traffic the IP's are not match different from one to another, therefore we randomized only the **4th** octet of the IP**.   
  In DDoS** traffic,we randomized the oneness (1 or 2) of the **3rd** octet and the **4th** octet.  
  **In Flash crowd** attack we randomized all the octets.
* **Center of group** – not specified, if the received IP was not added to any existing group, new group has created with the new IP.
* **Distance** –35, after calculating the mean distance for regular traffic.
* **Threshold for entropy** – was not declared.
* **Average Number of messages** – each host was calculated by   
  (packets generated/number of hosts) that was presented in simulation results.

By this table we assumed the IP's scattering, average number of messages per host, for each traffic type.

|  |  |  |  |
| --- | --- | --- | --- |
| **Different Scenarios** | | | **Simulated Environment** |
| **Packet Generated** | **Number of Groups** | **Number of hosts** |
| **409** | **5** | **50** | **Normal Operation** |
| **809** | **7** | **15** | **DDoS Attack** |
| **809** | **47** | **40** | **Flash Crowd** |

# Design and Flow

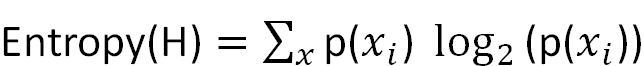
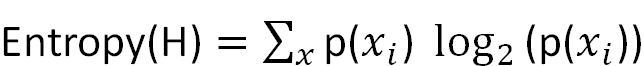
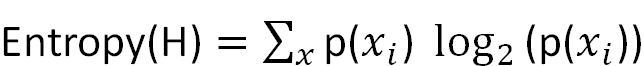
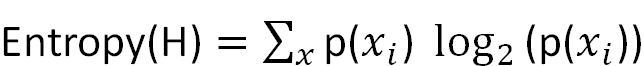
## Overview

The algorithm of the article is:



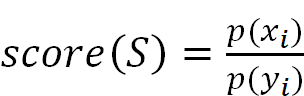
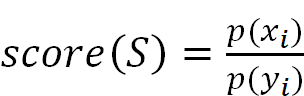
**Stage 1:** Each packet is placed in a specific group if its distance from the centroid of that respective group is less than some distance value.   
If no such group is present then a new group is formed to store that packet.

The group division: we compered the first 3 octets of the incoming message IP to the centroid IP address of each group, if all those octet were equal we calculated whether the absolute of the centroid 4th octet minus the 4th octet of the incoming message, if the result was smaller then the distance value we increased the number of group members' otherwise we kept checking all the other groups, if no match was found we created a new group with the incoming message IP as it's centroid.  
**Stage 2:** **Entropy calculation -**    
were 𝑃(𝑋𝑖 ) = ( 𝑝𝑎𝑐𝑘𝑒𝑡𝑠 𝑖𝑛 𝑔𝑟𝑜𝑢𝑝 𝑖 / 𝑡𝑜𝑡𝑎𝑙 𝑝𝑎𝑐𝑘𝑒𝑡𝑠 𝑟𝑒𝑐𝑒𝑖𝑣𝑒𝑑).



**Stage 3:** **Compare entropy to threshold -** If the entropy is less than the threshold then the traffic is normal, and new score is calculated and stored, as defined in the next slide. The threshold was not predefined, so we assumed it by the results of the entropy we achieved.

**Stage 4:** **Score calculation -** Groups are given scores based on their IP attribute addresses. Packet score assigns score to each attribute based on currently probability of occurrence of the group (*P*(xi*)*) and the stored probability of the occurrence of the group (*P*(yi*)*) .



score calculation indicates witch attack has occurred, by comparison of score value to threshold value. If the score result is smaller then the threshold then we have flash crowd attack, otherwise its DDoS attack.

## Modules overview

**simple Traffic**

parameters:

int nor\_host=default(50) – number of normal hosts in traffic

int dos\_host=default(15) – number of DDoS hosts in traffic

int flash\_host=default(40) – number of Flash Crowd hosts in traffic

gates:

output out – the sending gate to the Target

**simple Target**

parameters:

int distance

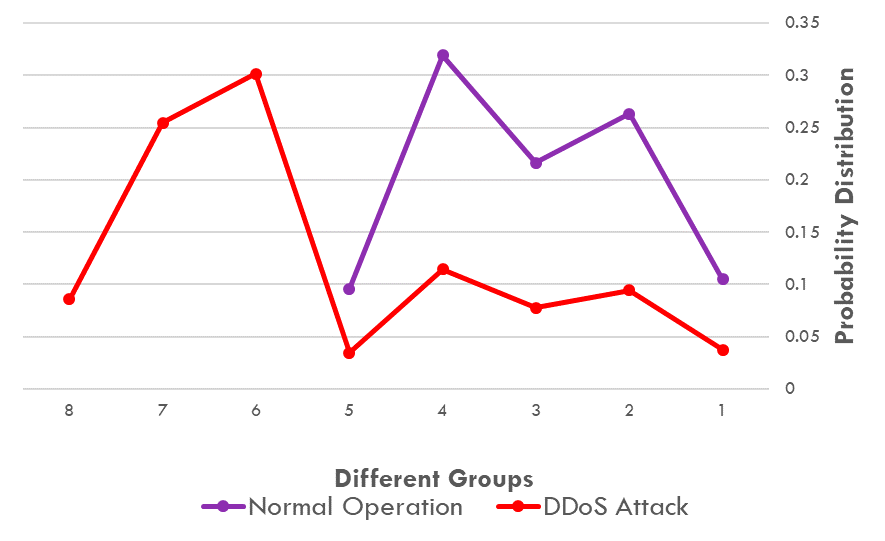
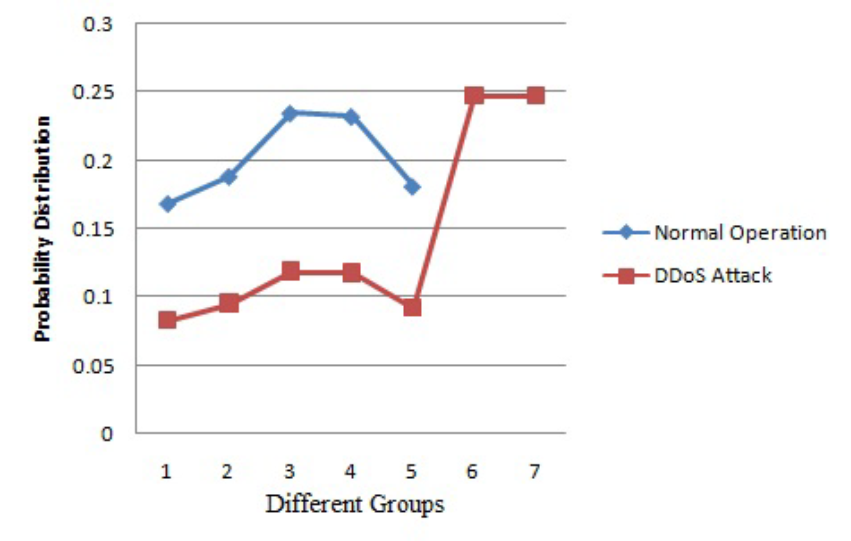
double thresh\_entropy=default(3.5) - threshold for the entropy values

double thresh\_score=default(0.15) - threshold for the score values  
gates:

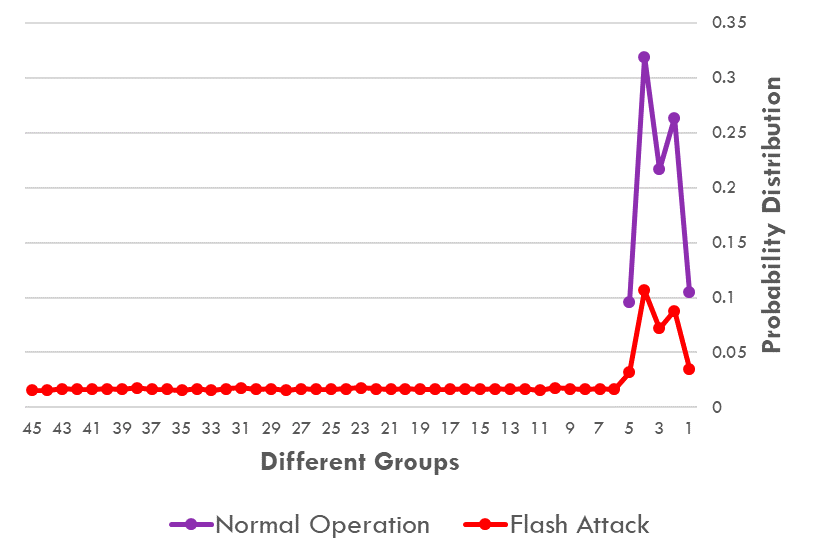
input in – the receiving gate from traffic module

## Results and Test measurements

**2.3.1 DDoS attack – group division**

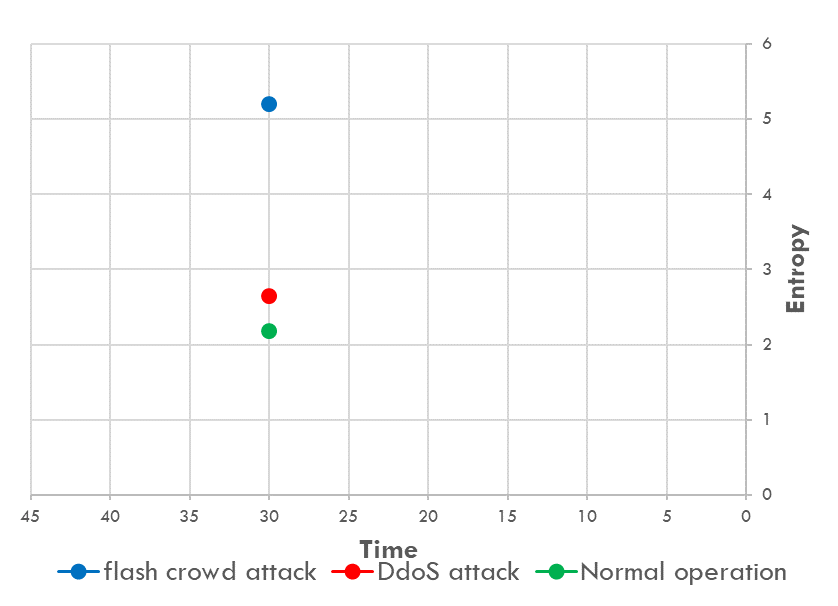
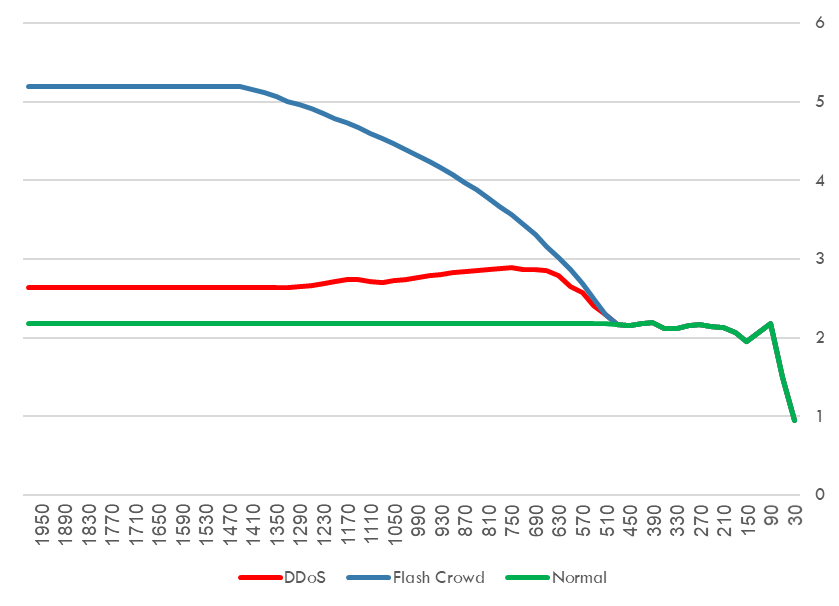


**2.3.2 Flash Crowd attack- group division**



The difference we can see in the graphs is probably because we didn't know how the centroid was chosen(as we can see in group 8 in DDoS attack- we assume that the group was created because of other definition for the centroid), and the way they divided into different group was not defined in the text so the way we applied the division may be different from the article therefore the differences.

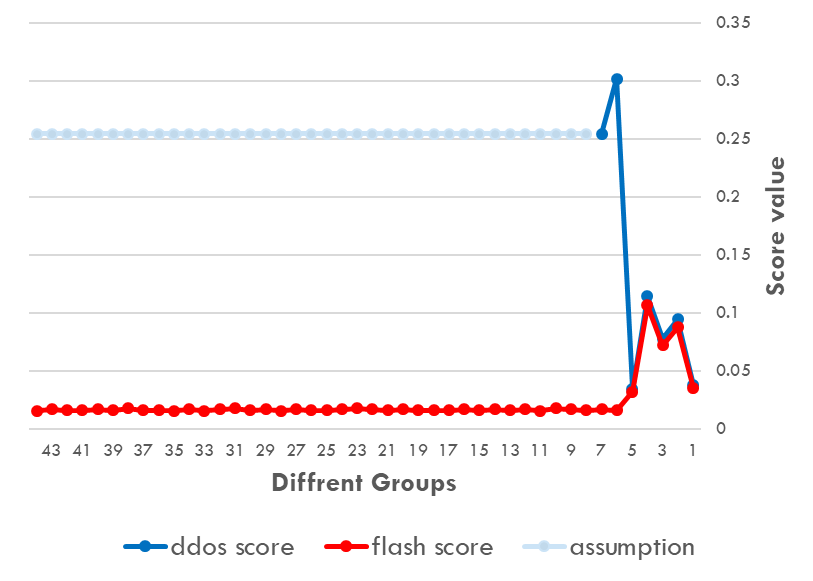
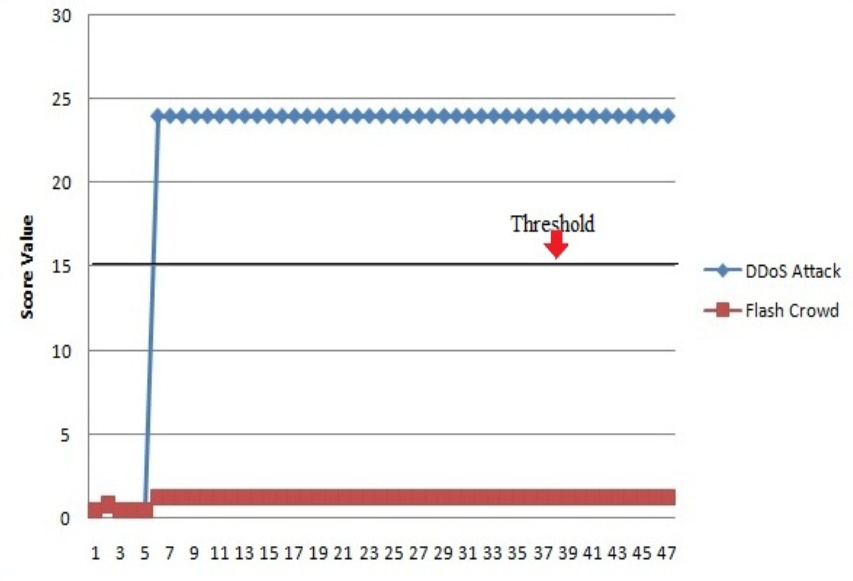
2.3.3 Entropy



The first two top graphs re ours results. The results are quite different, the right graph is created according to 40 seconds simulation time, the entropy is calculated every 30 seconds, so the entropy was actually calculated only once during the simulation. In the left graph we wanted to see the behavior of the entropy after several samples, in order to do so, we increased the time between sending each message, and the simulation time.as we can see from this experiment the entropy stabilized during the running of the simulation, which resembles the article results for the entropy.

In addition, the difference in values, is probably caused by the undefined group division, as we discussed before. The entropy is calculated according to the probability to be in specific group, so it's automatically affected the entropy.

**2.3.4 Score value– different group division**



The right graph is our score results for the different groups. According to the previous described result in DDoS attack there were only 8 groups, therefore we don’t understand the other 40+ groups described for DDoS attack in the article results. In anyway we can see that the score level for DDoS attack is higher than Flash crowd attack as it's supposed to be according to the algorithm, (the light blue in the right graph is an assumption of ours, it's a duplication of the last sampled result ).