**INTRODUCTION**

**1.1 Denial of Service (DoS)**

In computing, a denial of service (DoS) is an attempt to make a machine or network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the internet.

A distributed denial of service (DDoS) is where the attack source is more than one, often thousands of, unique IP addresses. It is analogous to a group of people crowding the entry door or gate to a shop or business, and not letting legitimate parties enter into the shop or business, disrupting normal operations.

Criminal perpetrators of DoS attacks often target sites or services hosted on high- profile web servers such as banks, credit card payment gateways, but motives of revenge, blackmail or activism can behind other attacks. [Ref 1]

**1.1.1 Symptoms**

The United States Computer Emergency Readiness Team (US-CERT) defines symptoms of DoS attacks to include:

* Unusually slow network performance (Opening files or accessing websites etc.).
* Unavailability of a particular website.
* Inability to access any website.
* Dramatic increase in the number of spam emails received. (Email bomb)
* Disconnection of a wireless or wired internet connection.
* Long term denial of access to the web or any internet services

If the attack is conducted on a sufficiently large scale, entire geographical regions of internet connectivity can be compromised without the attacker’s knowledge or intent by incorrectly configured or flimsy network infrastructure equipment. [Ref 1]

**1.2 Distributed Denial of Service (DDoS)**

A DDoS attack occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers. Such an attack is often the result of multiple compromised systems (For example a botnet) flooding the targeted system with traffic. A botnet is a network of zombie computers programmed to receive commands without the owner’s knowledge. When a server is overloaded with connections, new connections can no longer be accepted. The major advantages to an attacker of using a DDoS attack are that multiple machines can generate more attack traffic than one machine, multiple attack machines are harder to tune off than one attack machine, and that the behaviour of each attack machine can be stealthier, making it harder to track and shut down. These attacker advantages cause challenges for defence mechanisms. For example, merely purchasing more incoming bandwidth than the current volume of the attack might not help, because the attacker might be able to simply add more attack machines. This after all will end up completely crashing a website for periods of time.

Malware can carry DDoS attack mechanisms, one of the better known examples of this was My Doom. My doom is a computer worm affecting Microsoft windows. Its DoS mechanism was triggered on a specific date and time. This type of DDoS involved hardcoding the target IP address prior to release of the malware and no further interaction was necessary to launch the attack.

A system may also be compromised with a Trojan, allowing the attacker to download a zombie agent, or the Trojan may contain one. Attackers can also break into systems using automated tools that exploit flaws in programs that listen for connections from remote hosts. This scenario primarily concerns systems acting as servers on the web. Stacheldraht is a classic example of a DDoS tool. It utilizes a layered structure where the attacker uses a client program to connect to handlers, which are compromised systems that issue commands to the zombie agents, handlers by the attacker, using automated routines to exploit vulnerabilities in programs that accept remote connections running on the targeted remote hosts. Each handler can control up to a thousand agents. In some cases a machine may become part of a DDoS attack with the owner’s consent, for example, in operation payback, organized by the group Anonymous. These attacks can use different types of internet packets such as TCP, UDP, and ICMP etc.

These collections of systems compromisers are known as botnets/root servers. DDoS tools like stacheldraht still use classic DoS attack methods centred on IP spoofing and amplification like smurf attacks and fragile attacks (these are also known as bandwidth consumption attacks). SYN floods (also known as resource starvation attacks) may also be used. Newer tools can use DNS servers for DoS purposes. Unlike My Doom’s DDoS mechanism, botnets can be turned against any IP address. Script kiddies use them to deny the availability of well knows websites to legitimate users. More sophisticated attackers use DDoS tools for the purposes of extortion - even against their business rivals.

Simple attacks such as SYN floods may appear with a wide range of source IP addresses, giving the appearance of a well distributed DoS. These flood attacks do not require completion of the TCP three way handshake and attempt to exhaust the destination SYN queue or the server bandwidth. Because the source IP addresses can be trivially spoofed, an attack could come from a limited set of sources, or may even originate from a single host. Stack enhancements such as SYN cookies may be effective mitigation against SYN queue flooding, however complete bandwidth exhaustion may require involvement.

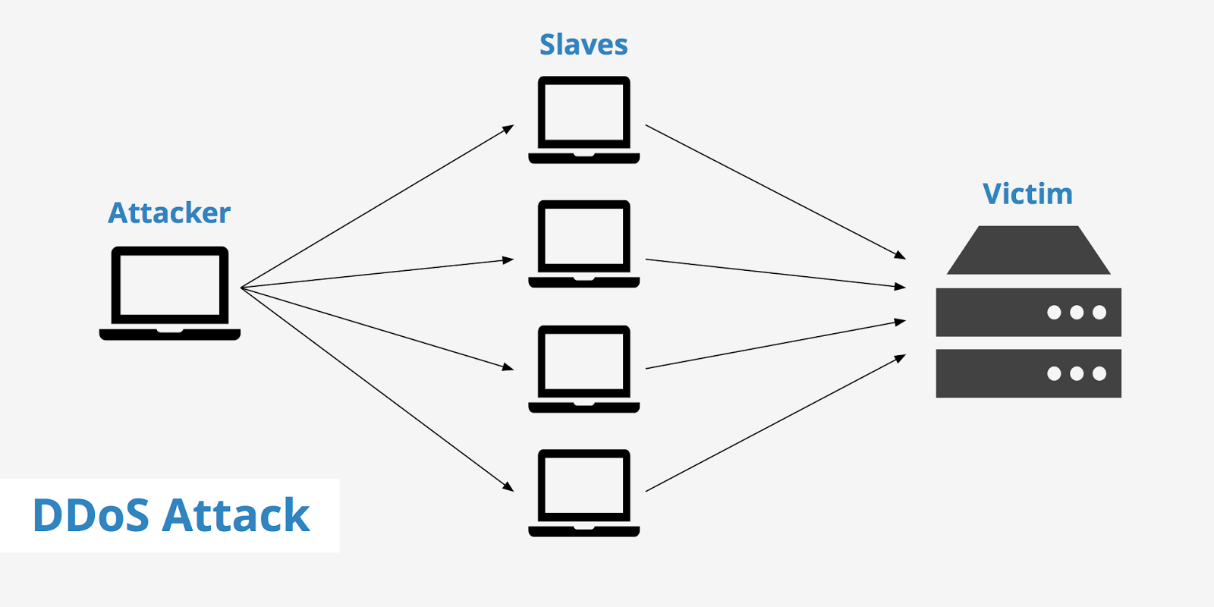
If an attacker mounts an attack from a single host it would be classified as a DoS attack. In fact, any attack against availability would be classed as a DoS attack. On the other hand, if an attacker uses many systems to simultaneously launch attacks against a remote host, this would be classified as a DDoS attack.

It has been reported that there are new attacks from Internet of Things which have been involved in DoS attacks. In one noted attack that was made peaked at around 20,000 requests per second which came from around 900 CCTV cameras.   [Ref 1]

**1.3 How DDoS Attacks Work?**

A website is technically a “service”, a software based system that responds in a particular way to incoming requests from client software - in this case a web browser. But a web browser’s request can be easily faked. A web server can only respond efficiently to a certain number of requests from pages, graphics and other website elements at once. Exceed that number, and it bogs down. Go too far, and the system may become entirely unresponsive. Huge floods of traffic, whether legitimate or not, can thus cripple a server. In recent years beefier hardware and better tools to distribute incoming requests among multiple servers have made things more difficult for attackers. DoS attacks once involved a single computer flooding a webserver. When that become ineffective, DDoS onslaughts conscripted thousands of virus infected computers, knows as zombies, to bombard the target system with bogus requests from many locations at once. This used to be impossible to block without severing the server’s internet link altogether. But now specialised hardware can distinguish between real requests and those intended to harm a site, and block them before they form a tsunami of traffic.

Attackers in turn have also responded with ever more sophisticated software. Past attacks where akin to making a telephone call and never replying to the answer on the other end, thus tying up the line. Mike Rothman, a researcher at Securosis, a security firm, explains in a white paper that hardware designed to repel such attacks can be bypassed using encrypted connections (HTTP sessions), which are typically handled directly by the server. Attackers also tune innocent websites and other internet services elsewhere (such as domain name servers) into unwitting assailants. This involves forging the sending address on queries to these other servers, which obligingly reply to the systems under attack, adding to the load. Attacks can thus be scaled up to well over 100 gigs per second (Gbps). But Mr Rothman notes that some attackers prefer precision attacks that exploit weaknesses in a specific function, rather than the entire server. For instance, sending a huge numbers of legitimate-seeming search requests to a website, each of which uses up substantial computational power, may be more effective and harder to pinpoint than simply flooding it with bogus page requests. [Ref 2]



**(Fig 1.1 Working of DDoS)**

**1.4 The Difference between DoS and DDoS Attacks**

The most significant difference is that in a DoS attack, the attackers use only one computer and one internet connection, while those launching DDoS attacks use a globally distributed network of computers and multiple internet connections. DOS attacks are much simpler and lower in cost.

In addition, it is much more difficult to fight against DDoS attacks as there are hundreds or thousands of sources sending out requests to flood the target, epically when a website or server is under a specifically targeted DDoS attack. Unlike single-source DoS attacks, DDoS assaults tend to target the network infrastructure in an attempt to saturate it with huge volumes of traffic. It is nearly impossible to block out the sources.

But in a DoS attack, if the incoming traffic is identified as being malicious instead of a normal traffic spike, hosts can take actions to absorb and attack and block the source as soon as it is identified. This kind of attacks can be stopped in a short time.

DDoS attacks also differ in the manner of their execution. Broadly speaking, DoS attacks are launched using homebrewed scripts or DoS tools (E.g., Low Orbit Ion Cannon), while DDoS attacks are launched from botnets - large clusters of connected devices (E.g., Mobiles, PCs or routers) infected with malware that allows remote control by an attacker. [Ref 3, 4]

**1.5 Motivation for DDos attack**

It is normally difficult to understand the motivation or goals behind specific DDoS attacks or why they occur. Because the machines or computers performing the attack are being controlled by some hidden external source, it is difficult to pinpoint the origin of the attack. When it is already hard to find out who are conducting the attacks, it is even harder to understand why. Therefore, many explanations of why DDoS occurs are theories based on speculation or small amounts of evidence.

Government websites are a common target for DDoS. Naturally, one can assume that there are some people, organizations or even other governments that do not support this government, and utilize DDoS as a form of cyber warfare to attack this government. Examples include the DDoS attacks against government websites owned by Russia, Georgia, United States, and South Korea. Theories claim that it was one country’s government attacking another country (Russia vs. Georgia or North Korea vs. United States/South Korea), but some evidence shows that the Georgian government’s website was already being attacked before the war began, implying the website could have been attacked by citizens or other people that already did not support the government. For sure, politics is involved in these DDoS attacks and is what motivates them, but it is unclear who is performing the attacks.

With the amount of information and resources available on the Internet, it may even be possible for a user with little technical knowledge to download and run a simple script that performs a DDoS attack. Regular Internet users may attempt to attack a large company’s website simply because they can. Being able to take down a large company or organization’s website can be enticing to the average, insignificant computer user. Personal conflict may even motivate some users to perform DDoS attacks on another user’s home computer, for the sake of revenge. Also, within online communities, including “hacker communities”, users that are able to succeed in taking down a target may receive some form of fame or recognition in their community. However, many of these people are commonly referred to as “script kiddies” in the online community, mainly due to the childishness of their motivation.

Some theories state that company websites under attack are being attacked by competing companies. This can disrupt or damage the target company’s services, which may very well boost the sales or amount of website views for the competing company. Small disruptions or downtime can translate to thousands of dollars lost for companies that conduct most of their business online, so there definitely are huge gains for a competing company performing DDoS attacks (or losses, in the case of the site being attacked). Apparently, DDoS attacks are very common in online gambling websites, where supposedly competing websites constantly DDoS each other.

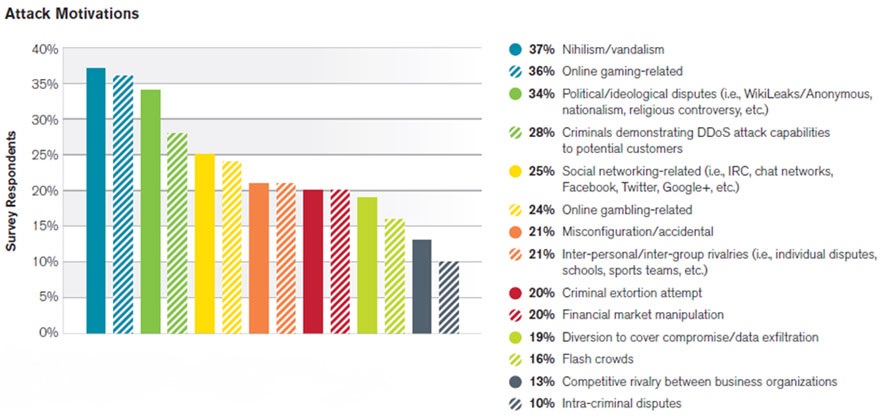
There have also been cases where DDoS has been used against companies for ransom purposes. The attackers may attack first and disrupt the company’s website until it is almost unusable or until it is down and demand ransom money in exchange for stopping the attacks. There have also been cases where attackers do not initially attack the company’s website, but will claim that they will initiate an attack on the company’s website if they do not pay the ransom amount. Some attackers even “rent” out or sell their botnets. All of these cases are motivated by financial reasons, although most reports show they have not been very successful.

Although it is usually very hard or nearly impossible to find out who are conducting DDoS attacks, there are definitely reasons as to how it would benefit them. Conflict between human beings would be more of a philosophical questions, but it is evident that people and companies are using DDoS as an attack medium against their “enemies” or competition now.

While any individual or organization can be the target of a DDoS attack, the attack typically serve to extort money or disrupt the operations of a private or government enterprise.

With that said, an understanding of DDoS motivations is essential for establishing an effective method of mitigating the damage of these attacks.

Participants in the Arbor Networks survey were asked what motivations they believed were behind the DDoS attacks they experienced in 2014. The results are as follows:



**(Fig 1.2 DDoS Attack Motivation)**

[Ref 5 6]

**1.6 Types of DDoS Attacks**

There are many types of DDoS attacks. Common attacks include the following:

**1.6.1 Volumetric attacks**

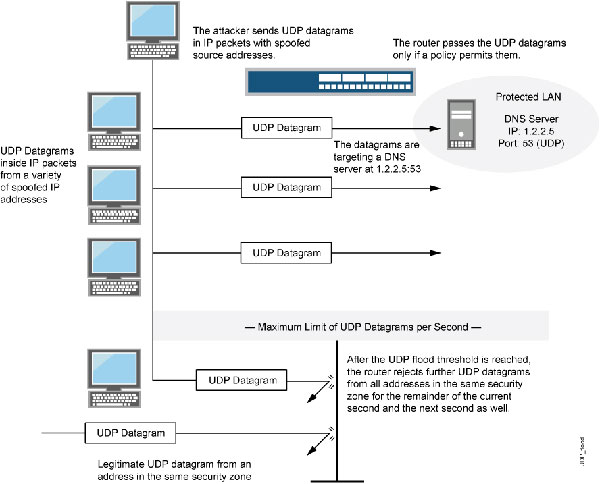
Volumetric are the most common types of DDoS attack, making up for about 65% of the total reported, according to Arbor.

These attacks use multiple infected systems - which are often part of a botnet - to flood the network layers with a substantial amount of seemingly legitimate traffic. This consumes an excessive amount of bandwidth within and/or outside of the network and drives network operations to become painfully sluggish or simply non-functional.

Since volumetric attacks essentially “gang rush” a network, they’re much more difficult to mitigate than attacks from a single source.

Volumetric attacks come in a variety of forms, including:

* **User Datagram Protocol (UDP) floods**: Random ports on a server are flooded with UDP packets, causing the server to repeatedly check for and respond to non-existent applications at the ports. As a result of the UDP Flood, the system is unable to respond to legitimate applications.



**(Fig 1.3 UDP Floods)**

* **ICMP floods**: A server is flooded with ICMP echo requests from multiple spoofed IP addresses. As the targeted server processes and replies to these phony requests, it is eventually overloaded and unable to process valid ICMP echo requests

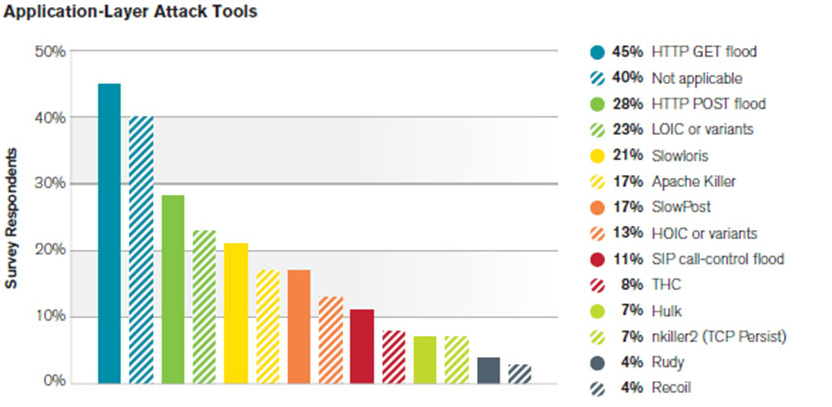
**1.6.2 Application - layer attacks**

Application-layer attacks comprise about 17% of all reported DDoS attacks. They target web application packets in order to disrupt the transmission of data between hosts.

For example, a HTTP Flood uses multiple infected machines to force a target to expend and excessive amount of resources when responding to a HTTP request.

From the attacker’s standpoint, a HTTP Flood is a far more effective threat than other types of attacks since it doesn’t need to consume a great deal of bandwidth to handcuff a server.

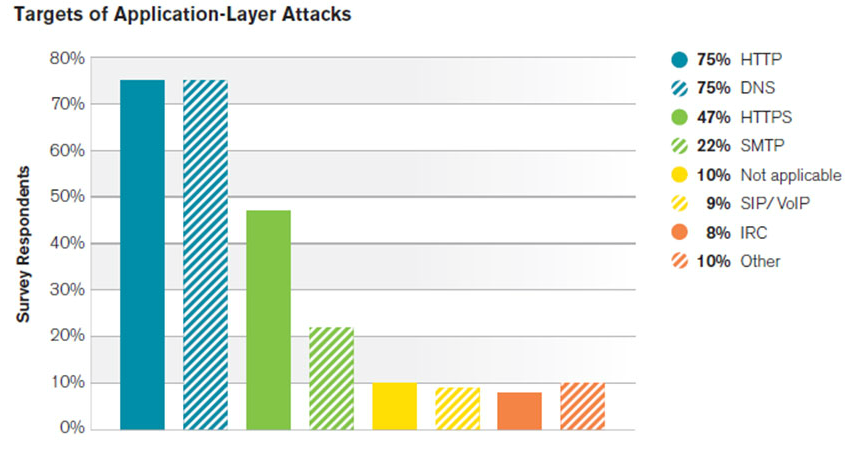
Though a HTTP Flood is typically the most common application layer attack experienced, it’s merely one of many application - layer attack tools available. The table below from Arbor demonstrates how attackers are constantly finding new ways to compromise the application - layer.

**(Fig 1.4 Application Layer Attack Tools Graph)**

Since HTTP floods and other application - layer DDoS attacks mimic human - user behaviour, they’re also much more difficult to detect than other types of attacks. Additionally, application layer attacks can also come from a single machine, which causes less traffic to be generated. In turn, these attacks often go under the radar of detection systems.

While HTTP and DNS services are the primary targets of application - layer attacks, HTTPS and SMTP were also commonly targeted in 2014, although less often, according to the Arbor Network report.

The chart below shows the percentage of respondents who received attacks to the application - layer targets listed.



**(Fig 1.5 Targets of Application-Layer Attack Graph)**

**1.6.3 State - Exhaustion attacks**

Also known as protocol attacks, state - exhaustion attacks target the connection state tables in firewalls, web application servers, and other infrastructure components.

State - exhaustion attacks occur somewhat more frequently than application - layer attacks, accounting for about 20% of reported DDoS attacks in 2014, according to Arbor.

One of the most common state-exhaustion attacks is the notorious **Ping of Death**, in which a 65,536 byte ping packet is defragmented and sent to a target server as fast as possible.

Once the target reassembles the large packet, a buffer overload typically occurs. In the likely scenario that the target attempts to respond to the pings, even more bandwidth is consumed, eventually causing the targeted system to crash.

It’s important to note that these types of DDoS attacks are often used in conjunction with one other to compromise a single target. 42% of respondents in the Arbor Networks report claim to have experienced a multiple-threat attacks in 2014, a 3% increase from 2013.

[Ref 6]

**1.6.4 Some more specific types of DDoS attacks**

* **SYN Flood:** This type of attack is a classic DDoS that sends rapid amounts of packets at a machine in an attempt to keep connection, and eventually that connection times out. If the attack is strong enough it will consume all resources on the server and sends the website offline.
* **Reflected Attack**: Forged packets are sent out to as many computers as possible. When the packets are received the computers reply, but because the packets are received the computers reply, but because the packets are spoofed, instead of responding to the real sender, the machines will all attempt to communicate with the machine at the spoofed address. Eventually, if the attack is strong enough the server will shut down.
* **Slowloris**: Types of DDoS attacks like these are way more complex than some of the other DDoS attacks. Slowloris is a DDoS toolkit that sends out partial requests to a target server in an effort to keep the connections open as long as possible. At the same time it does this, it sends out HTTP headers at certain intervals, which ramps up the requests, but never makes any connections. It doesn’t take long for this type of DDoS attack to take down a website.
* **Peer-to-Peer attacks**: These types of attacks exploit peer-to-peer networks by maliciously redirecting legitimate users to the site of server they want to attack. If the attacker is able to pull it off with enough people, the resulting DDoS shuts down the site.
* **Multi-Vector Attacks**: A Multi-Vector DDoS attack is quite possibly the most complex form of DDoS. This is where attackers not only blend attack strategies, but they often use a variety of tools as well. When you are faced with this type of DDoS attack you will notice the attacker pinpointing applications on your server, while at the same time flooding your site with bad traffic.
* **Zero Day DDoS**: “Zero Day” are simply unknown or new attacks, exploiting vulnerabilities for which no patch has yet been released, the term is well-known amongst the members of the hacker community, where the practice of trading Zero-day vulnerabilities has become a popular activity. Example for a zero day DDoS is Stuxnet.
* **Unintentional DDoS**: This occurs when a system ends up denied, not due to a deliberate attack by a single individual or group of individuals, but simply due to a sudden enormous spike in popularity. This can happen when an extremely popular website posts a prominent link to a second, less well-prepared site, for example, as part of a news story. This result is that a significant proportion of the primary site’s regular users - potentially hundreds of thousands of people - click that link in the space of a few hours, having the same effect on the target websites as a DDoS attack. [Ref 7, 8, 1 ]

**1.7 Why are DDoS attacks so dangerous?**

DDoS represents a significant threat to business continuity. As organizations have grown more dependent on the Internet and web-based applications and services, availability has become as essential as electricity.

DDoS is not only a threat to retailers, financial services and gaming companies with an obvious need for availability. DDoS attacks also target the mission critical business applications that your organization relies on to manage daily operations, such as email, Salesforce automation, CRM and many others. Additionally, other industries, such as manufacturing, pharma and healthcare, have internal web properties that the supply chain and other business partners rely on for daily business operations. All of these are targets for today’s sophisticated attackers. [Ref 9]

**1.8 Preparing for DDoS attacks**

You can’t prevent DDoS assaults. The fast is that cybercriminals are going to attack. Some are going to hit their targets, regardless of the defences in place.

However, there are steps you can take to spot a brewing storm, including:

* Monitoring your traffic to look for abnormalities, including unexpected traffic spikes and visit from suspect IP address and geolocations. All of these could be signs of attackers performing dry runs to test your defences before committing to a full-fledged attack. Recognizing these for what they are can help you prepare for the onslaught to follow.
* Keep an eye on social media (particularly Twitter) and public waste bins (Pastebin.com) for threats, conversations and boasts that may hint on an incoming attack.
* Consider using third-party DDoS testing (i.e., pen testing) to simulate an attack against your IT infrastructure so you can be prepared when the moment of truth arrives. When you undertake this, test against a wide variety of attacks, not just those with which you are familiar.
* Create a response plan and a rapid response team, whose job is to minimize the impact of an assault. When you plan, put in place procedures for your customer support and communication teams, not just for your IT professionals.

**1.9 What are the consequences of a successful DDoS attack?**

When a public facing website or application is unavailable, that can lead to angry customers, lost revenue and brand damage. When business critical applications become unavailable, operations and productivity grind to a halt. Internal websites that partners rely on means supply chain and production disruption.

A successful DDoS attack also means that your organization has invited more attacks. You can expect attacks to continue until more robust defences are deployed. [Ref 9]

**1.10 What are your DDoS protection options?**

Given the high profile nature of DDoS attacks, and their potentially devastating consequences, many security vendors have suddenly started offering DDoS protection solution. With so much riding on your decision, it is critical to understand the strengths, and weaknesses, of your options.

**1.10.1 Application Front End Hardwar**e

Application front end hardware is intelligent hardware placed on the network before traffic reaches the servers. It can be used on networks in conjunction with routers and switches. Application front end hardware analyses data packets as they enter the system, and then identifies them as priority, regular, or dangerous. There are more than 25 bandwidth management vendors.

**1.10.2 Application level Key Completion Indicators**

In order to meet the case of application level DDoS attacks against Cloud based applications, approaches may be based on an application layer analysis, to indicate whether an incoming traffic bulk is legitimate or not and thus enable the triggering of elasticity decisions without the economic implications of a DDoS attack. These approaches mainly rely on an identified path of value inside the application and monitor the macroscopic progress of the requests in this path, towards the final generation of profit, through markers denoted as Key Completion Indicators.

**1.10.3 Black holing and sink holing**

With black holing, all the traffic to the attacked DNS or IP address is sent to a black hole (Null interface or a non-existent server). To be more efficient and avoid affecting network connectivity, it can be managed by the ISP.

Sink holing routes traffic to a valid IP address which analyses traffic and rejects bad packets. Sinking is not efficient for most severe attacks.

**1.10.4 DDS based defence**

More focused on the problem than IPS, a DoS Defence System (DDS) can block connection-based DOS attacks and those with legitimate content but bad intent. A DDS can also address both protocol attacks (Such as Teardrop and Ping of Death) and rate-based attacks (Such as ICMP floods and SYN floods).

**1.10.5 Firewalls**

In the case of a simple attack, a firewall could have a simple rule added to deny all incoming traffic from the attackers, based on protocols, ports or the originating IP addresses.

More complex attacks will however be hard to block with simple rules, for example, if there is an ongoing attack on port 80 (Web Service), it is possible to drop all incoming traffic on this port because doing so will prevent the server from serving legitimate traffic. Additionally, firewalls may be too deep in the network hierarchy, with routers being adversely affected before the traffic gets to the firewall.

**1.10.6 IPS based prevention**

Intrusion-Prevention System (IPS) are effective if the attacks have signatures associated with them. However, the trend among the attacks is to have legitimate content but bad intent. Intrusion-prevention systems which work on content recognition cannot block behaviour-based DoS attacks.

An ASIC based IPS may detect and block DoS attacks because they have the processing power and the granularity to analyse the attacks and act like a circuit breaker in an automated way.

A rate-based IPS (RBIPS) mush analyse traffic granularly and continuously monitor the traffic pattern and determine if there is traffic anomaly. It must let the legitimate traffic flow while blocking the DoS attack traffic.

**1.10.7 Routers**

Similar to switches, routers have some rate-limiting and ACL capability. They too, are manually set. Most routers can be easily overwhelmed under a DoS attack. Cisco IOS have optional features that can reduce the impact of flooding.

**1.10.8 Switches**

Most switches have some rate-limiting and ACL capability. Some switches provide automatic and/or system-wide rate limiting, traffic shaping, delayed binding (TCP splicing), deep packet inspection and Bogon filtering (Bogus IP filtering) to detect and remediate DoS attacks through automatic rate filtering and WAN Link failover and balancing.

These schemes will work as long as the DoS attacks can be prevented by using them. For example, SYN flood can be prevented using delayed binding or TCP splicing. Similarly content based DoS may be prevented using deep packet inspection. Attacks originating from dark address or going to dark addresses can be prevented using bogon filtering. Automatic rate filtering can work as long as set rate-thresholds have been set correctly and granularly. Wan-link failover will work as long as both links have DoS/DDoS prevention mechanism

**1.10.9 Upstream filtering**

All traffic is passed through a “Cleaning centre” or a “Scrubbing centre” via various methods such as proxies, tunnels or even direct circuits, which separates “bad” traffic (DDoS and also other common internet attacks) and only sends good traffic beyond to the server. The provider needs central connectivity to the internet to manage this kind of service unless they happen to be loaded within the same facility as the “Cleaning centre” or a “Scrubbing centre”. [Ref 1]

**1.11 What is Arbor’s approach to DDoS protection?**

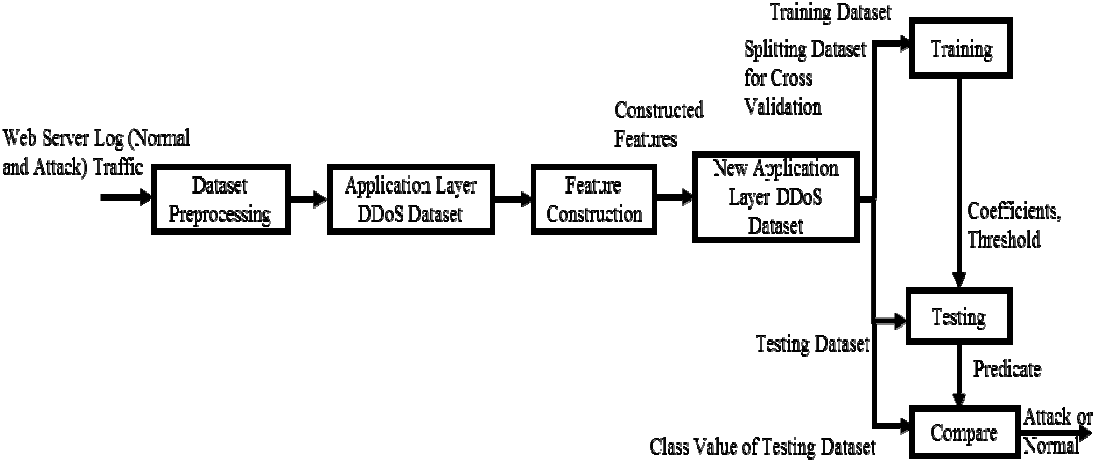
Arbor has been protecting the world’s largest and most demanding networks from DDoS attacks for more than a decade. Arbor strongly believes that the best way to protect your resources from modern DDoS attacks is through a multi-layer deployment of purpose-built DDoS mitigation solutions.

You need protection in the cloud to stop today’s high volume attacks, which are exceeding 300GB/sec. You also need on-premise protection against stealthy application-layer attacks, and attacks against existing stateful infrastructure devices, such as firewall, IPS and ADCs.

Only with a tightly integrated, multi-layer defence can you adequately protect your organization from the full spectrum of DDoS attacks.

* Arbor Networks Cloud (Tightly integrated, multi-layer DDoS protection)
* Arbor Networks APS (On-Premises)
* Arbor Networks SP/TMS (High capacity On-Premise Solution for Large Organizations)[Ref 9]

**1.12 Proposed Method**



**(Fig.1.6 Block schematic diagram of proposed method)**

An anomaly based statistical pattern recognition method is proposed to detect application layer ddos attacks. Fig. 1.12.1, depicts the block schematic diagram of the proposed method. From the web server log (Attack and Normal), after pre-processing, an application layer ddos attack dataset is constructed. From the constructed dataset features are extracted. From the extracted feature set some new characteristics are built.

Then combining extracted and constructed features a new application layer ddos attack dataset is generated. The new application layer ddos attack dataset is split into two sub datasets, viz., training dataset and testing dataset. This has been done for cross validation to avoid over fitting of results.

Here, 10 fold cross validation is used. In training phase, the algorithm is trained to detect application layer ddos attack and in testing phase, the algorithm is tested against upcoming traffic.

**CHAPTER 2 - SOFTWARE REQUIREMENTS SPECIFICATION**

**2.1 Requirements**

**2.1.1 Capability**

The existing solutions cannot detect all types of application layer ddos attacks as they use very less features to model user behaviour. So from this software we require to construct a new feature set from the original feature set, which can increase the accuracy of classification of all types of application layer ddos attack detection, and to use logistic regression. The main goal is to understand the difference between a normal user and an attack user by modelling their behaviour.

**2.1.2 Capacity**

In the application layer, message and package creation happens. In a request, hundreds of attack packets can be sent and requested by millions of bots. The software must be able to detect the bots even if they are larger in number than the normal user.

**2.1.3 Reliability**

In a company, a huge information leak could cause millions of losses therefore it should run at all times of the day without any complications. It should keep modelling every few users.

**CHAPTER 3 - System Design**

**3.1 Introduction**

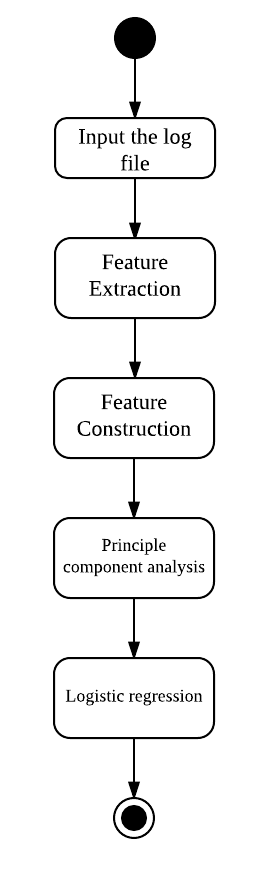
System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. It is a process for problem solving and planning for software solution. It includes low level component and algorithm implementation design.

**3.2 Logical Design**

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of a system. This can be done by modelling or using an over-abstract model of the actual system.

**3.3 High Level Design**

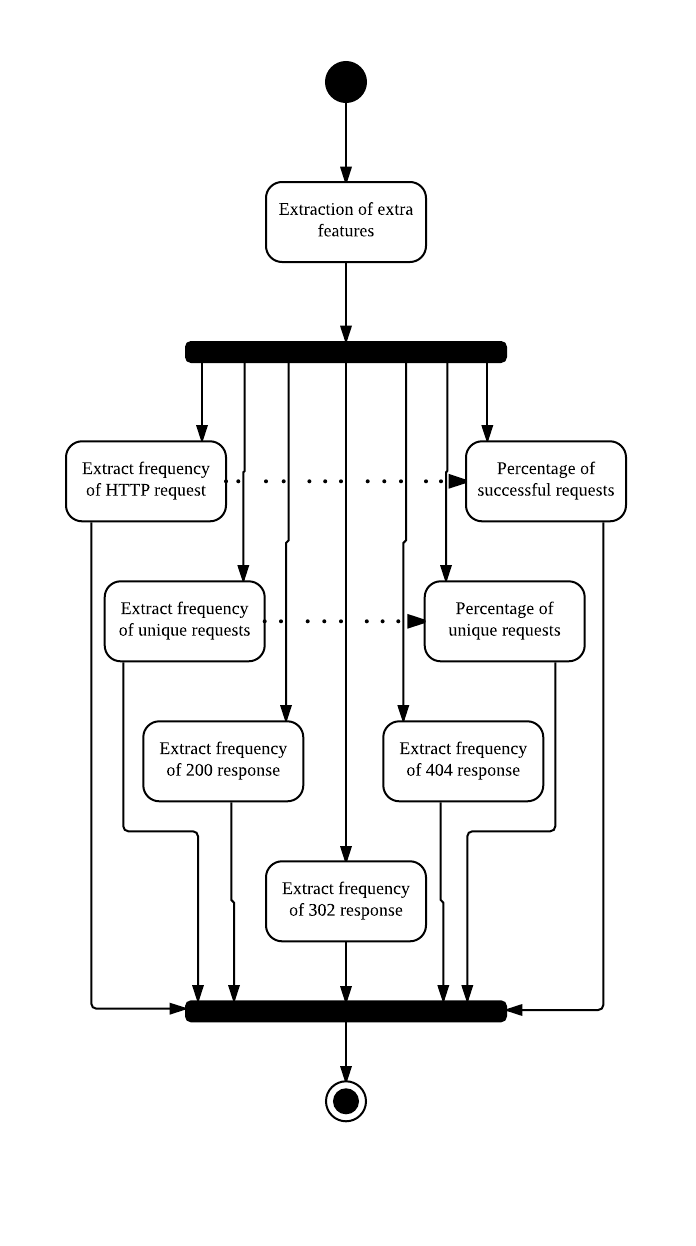
Figure 4.1 depicts a high level design of the project showing the relation between each module, which is shown here in the form of an activity diagram. Activity diagrams are graphical representations of workflow of stepwise activities which support for choice, iteration and concurrency. In the UML, activity diagrams are intended to model both computational and organizational process. Activity diagram show the overall flow of control.



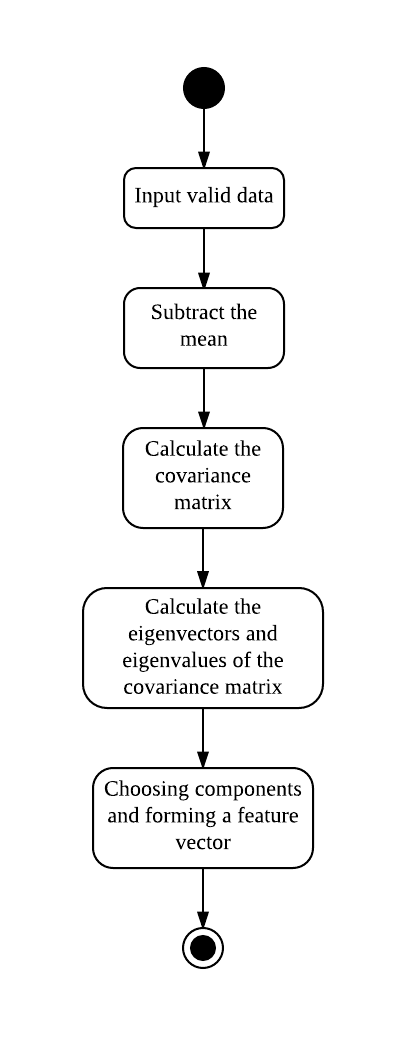
**(Fig 3.1 Activity Diagram for High Level Design)**

**3.4 Low Level Design**

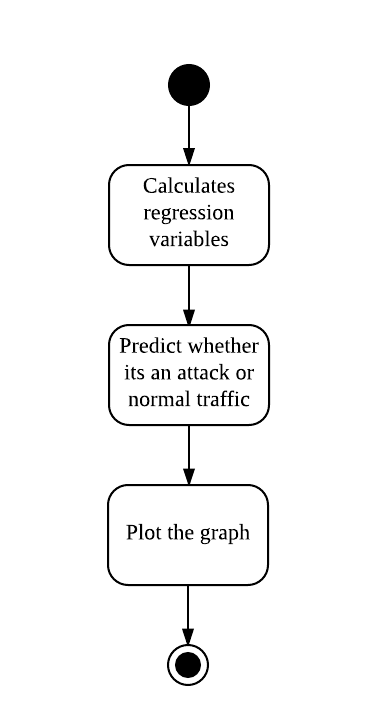
Low level design concentrates on the design of each module to get a clear view in depth of the low level architecture. Low level design is expressed here as an activity diagram.



**(Fig 3.2 Activity Diagram for Activity Extraction)**

****

**(Fig 3.3 Activity Diagram for PCA)**



**(Fig 3.4 Activity Diagram for Logistic Regression)**

**CHAPTER 4 - IMPLEMENTATION**

**4.1 Dataset**

There are normal and attack datasets available online but attack datasets available can only be accessed by prior permission from its owner. But it can be generated.

**4.1.1 Normal Dataset**

In this project, we are using a dataset which is publicly available. [Ref 10]

**4.1.2 Attack Dataset**

Attack dataset has been created by writing a script which uses 10 dummy IPs, some dummy URLs and generated randomly with a running time.

**4.2 Dataset Pre-processing**

The dataset which was prepared is analysed to construct some characteristics of the logs.

**4.2.1 Feature Extraction**

The datasets contain typically the following types of instances:

199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245

1. Host: It (199.72.81.55) is the source IP address of the user.
2. Timestamp: It (01/Jul/1995:00:00:01 -0400) is the date and time at which request was made.
3. Request: It (GET /history/apollo/ HTTP/1.0) represents the page that has been requested by the web user and the method “GET” or “Post”
4. HTTP\_Reply\_Code: It (200) is the response generated by a web server to the request made by the user.
5. Size: It (6245) is the number of data bytes sent from web server to the user for a given request.

All these 5 fields are extracted from both normal and attack dataset. Request extracted is in string format and these features are used to check only their presence or absence as these features could cause a huge impact on detection of application layer DDoS attack.

**4.2.2 Feature Construction**

Features available through a feature extraction process are very less and some of the features are in string format. So, it is difficult to determine all three types of application layer ddos attacks using these features. Hence some more features are also constructed which are as follows:

1. Freq\_of\_HTTP\_Req: It measures the number of HTTP requests made by a user in a particular timeslot. This parameter is used in HTTP request flooding attacks detection.
2. Freq\_of\_diff\_HTTP\_Req: It measures how many unique URL requests have been sent by users.
3. Freq\_of\_200\_Resp: It measures how many requests by the user were successful.
4. Freq\_of\_302\_Resp: It measures how many requests of users have been redirected.
5. Freq\_of\_404\_Resp: It measures how many invalid requests have been made by users.
6. Perc\_diff\_Req: It measures the percentage of different requests made in a particular timeslot, which is calculated as follows:                 Perc\_diff\_Req = (Freq\_of\_diff\_HTTP\_Req / Freq\_of\_HTTP\_Req) \*100
7. Perc\_Succ\_Resp: It measures the percentage of successful responses received by an user, which is calculated as follows:            Perc\_Succ\_Resp = (Freq\_of\_200\_Resp / Freq\_of\_HTTP\_Req)\*100

By combining all these 7 constructed features with the 5 extracted features, the application layer DDoS attack dataset with a total 12 features is created.

**4.3 Methodology**

In this project, Principle Component Analysis has been used to find effective parameters (feature) of these 12 features. The output of the PCA is input to Logistic Regression which is used to compare the upcoming traffic for classifying into normal and attack traffic.

**4.3.1 Principle Component Analysis (PCA)**

PCA is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension.

The Basic steps are:

**Step 1: Get some data**

|  |  |
| --- | --- |
| **X** | **Y** |
| 2.5 | 2.4 |
| 0.5 | 0.7 |
| 2.2 | 2.9 |
| 1.9 | 2.2 |
| 3.1 | 3.0 |
| 2.3 | 2.7 |
| 2.0 | 1.6 |
| 1.0 | 1.1 |
| 1.5 | 1.6 |
| 1.1 | 0.9 |

For example, the data used in this example is:

Table 4.1: Example data set

**Step 2: Subtract the mean**

For PCA to work properly, you have to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension. So, all the value have (the mean of the values of all the data points) subtracted, and all the values have subtracted from them. This produces a data set whose mean is zero. Now the table is:

|  |  |
| --- | --- |
| **X** | **Y** |
| 0.69 | 0.49 |
| -1.31 | -1.21 |
| 0.39 | 0.99 |
| 0.09 | 0.29 |
| 1.29 | 1.09 |
| 0.49 | 0.79 |
| 0.19 | 0.31 |
| -0.81 | -0.81 |
| -0.31 | -0.31 |
| -0.71 | -1.01 |

**Step 3: Calculate the covariance matrix**

The covariance matrix will be:

So, since the non-diagonal elements in this covariance matrix are positive, we should expect that both the X and Y variables increase together.

**Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix**

Since the covariance matrix is square, we can calculate the eigenvectors and eigenvalues for this matrix. These are rather important, as they tell us useful information about our data. I will show you why soon. In the meantime, here are the eigenvectors and eigenvalues:

It is important to notice that these eigenvectors are both unit eigenvectors i.e. their

Lengths are both 1. This is very important for PCA, but luckily, most maths packages, when asked for eigenvectors, will give you unit eigenvectors.

**Step 5: Choosing components and forming a feature vector**

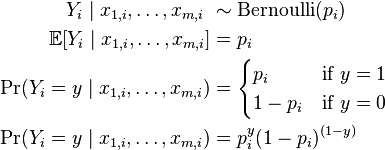
**4.3.2 Logistic Regression**

Logistic regression is one of the most popular classification methods. It has been used for modelling user behaviour. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. Thus, it treats the same set of problems as [profit regression](https://en.wikipedia.org/wiki/Probit_regression) using similar techniques, with the latter using a cumulative normal distribution curve instead. Equivalently, in the latent variable interpretations of these two methods, the first assumes a standard [logistic distribution](https://en.wikipedia.org/wiki/Logistic_distribution) of errors and the second a standard [normal distribution](https://en.wikipedia.org/wiki/Normal_distribution) of errors.

Logistic regression can be seen as a special case of [generalized linear model](https://en.wikipedia.org/wiki/Generalized_linear_model) and thus analogous to [linear regression](https://en.wikipedia.org/wiki/Linear_regression). The model of logistic regression, however, is based on quite different assumptions (about the relationship between dependent and independent variables) from those of linear regression. In particular the key differences of these two models can be seen in the following two features of logistic regression. First, the conditional distribution y \mid xis a distribution rather than a [Gaussian distribution](https://en.wikipedia.org/wiki/Gaussian_distribution), because the dependent variable is binary. Second, the predicted values are probabilities and are therefore restricted to (0, 1) through the [logistic distribution function](https://en.wikipedia.org/wiki/Logistic_function) because logistic regression predicts the probability of particular outcomes.

The application layer DDoS attack can be determined by using the effective features from these 12 features. In this project, logistic regression is used as it is suitable for modelling normal user web browsing behaviour.

To calculate values for logistic regression-



https://lh3.googleusercontent.com/wxbzf4Khm8QbyRFs0wzwdsGiegProq83Puhorw-KFuwTyCRWv_JT0p3aYrIfckxOIDzAJxSFBXYI4GtZbThNbmmMVgpBDe7a_c6qqh73LQrLK5iKCNGDGY_SVxquBpOdZlUJwQnw

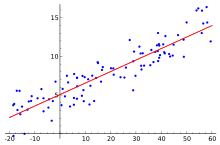
Where,

https://lh5.googleusercontent.com/yCjhCS3GJsnRwmcN3w1qj3uvBx41byh5c3cqXxK89FBmLf5QkwuxcLe_MGPJ21KXVz7HF7eP-8_DSt1OQXh6-jjTYrBDUTNwpsGKr45vl2C23XDzSd4iP3tfK_0P_5qYWq5djvsP=>https://lh6.googleusercontent.com/F8jLEGq9O2srxN8LecWnyGttZDwqzDz295UHeGSO_Y9e57yu1Ivu38g2s89SMO7EVtrZ2LeDItqvVIC2gYddPGUL-IZHIkc_Bba0ZQdp9bIytmBl0V_dXhfUrGAzif5vnbjDzaVq

It is same as,

https://lh5.googleusercontent.com/NkpC_kGYnNg4V3riG969dgXBQQS2Muku0qhQAYnz1q9DJ_iG3_fPE0jty23UTL19WpAKcVy8obErusJbPRvMtdltKnaf_zB8NXtK-HTH1iNXmhKcS1lNOnr8hGBgq0MJVsTTGU6L

 Y_i = \begin{cases} 1 & \text{if }Y_i^\ast > 0 \ \text{ i.e. } - \varepsilon < \boldsymbol\beta \cdot \mathbf{X}_i, \\
0 &\text{otherwise.} \end{cases} 



**Fig 4.1: Example output got by logistic regression**

**4.4 Algorithm**

**4.4.1 Training Dataset Algorithm**

Input: Normal dataset and attack dataset.

Output: Effective feature set, Threshold 𝞃 ,𝛾T

Step 1: Assign class values for normal and attack datasets.

Step 2: Combine normal and attack dataset and generate application layer DDoS attack dataset.

Step 3: Extract features from the dataset.

Step 4: Construct 7 features from the extracted feature set.

Step 5: Select effective parameter using PCA.

Step 6: Compute Threshold 𝞃 ,𝛾T

**4.4.2 Testing Dataset Algorithm**

Input: Effective feature set, Threshold 𝞃 ,𝛾T

Output: Predicted class values.

Step 1. Extract features from the dataset

For each instance

Step 2. Construct effective feature from the extracted

Feature set.

Step 3. Find the probability (p) for instance using 

Where, y = γ 0 +γ 1 x 1 +γ 2 x 2 + ... +γ n x n

Step 4. Compare it with threshold τ

If p > τ

Predicted value = attack

Else

Predicted value = normal

End

**4.5 Languages and Tools**

**4.5.1 Language used: Python**

Python is an application friendly language which is most suitable for working with networks and cyber security because python is dynamic, efficient and easy to program.

**4.5.2 Libraries used:**

1. NumPy - This library function helps calculate various numeric values that we have calculated such as covariance, Eigen values and vectors.
2. MatplotLib- This library function helps in plotting graph.
3. Sklearn- This library helps in calculating values for regressions and classification easily.

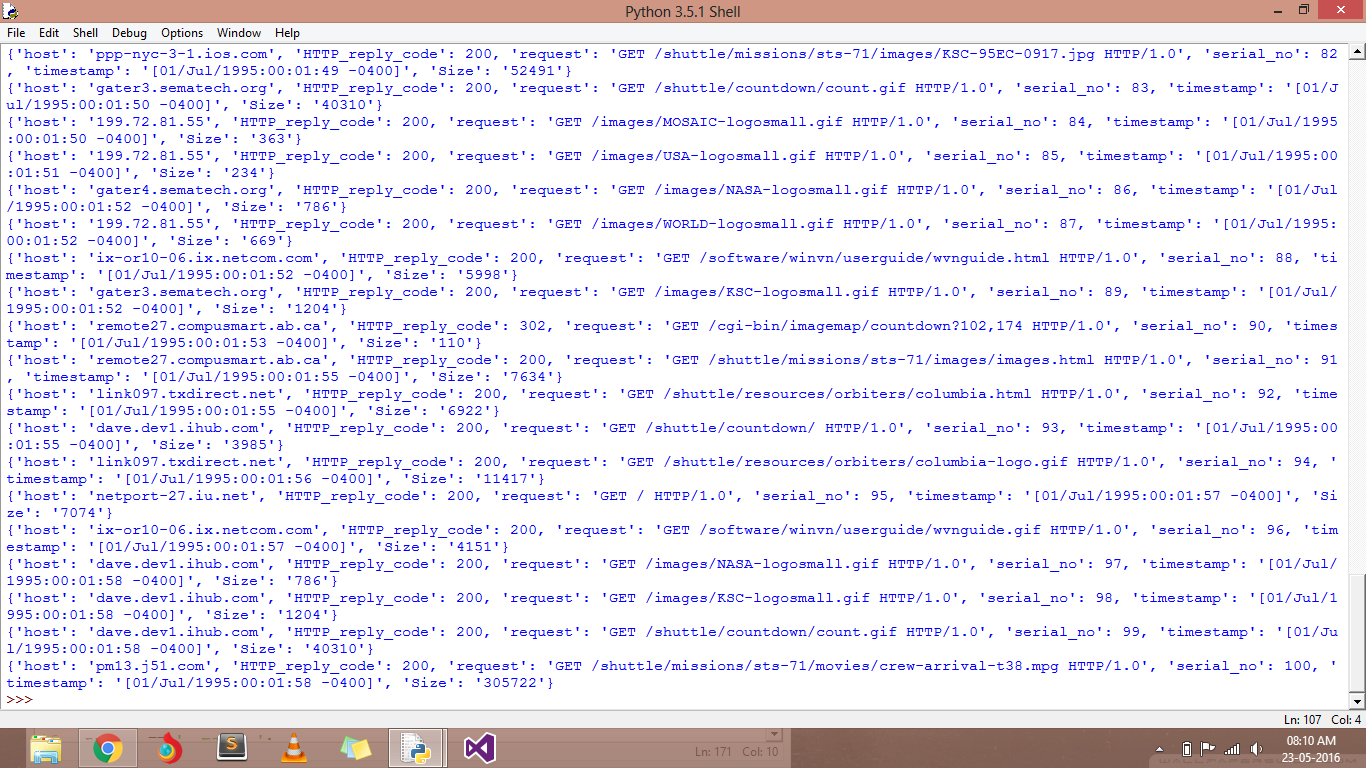
**CHAPTER 5- TESTING**

**5.1 Testing Method**

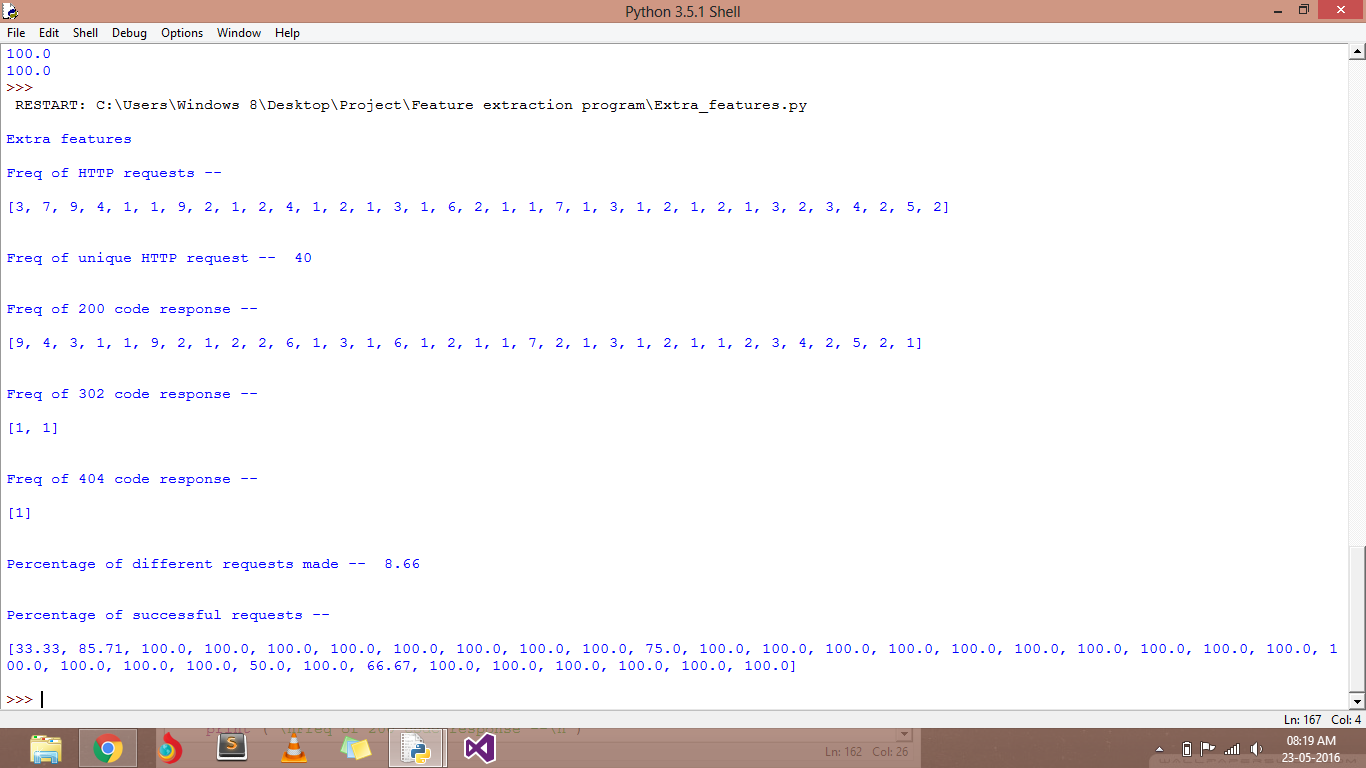
In order to test this programme, we are going to test using the concept of bottom up integration. This way, we test each module and built up to the entire programme. Then test the entire programme as a whole.

**5.1.1 Test Data Pre-processing**

Screenshot of initial datasets--

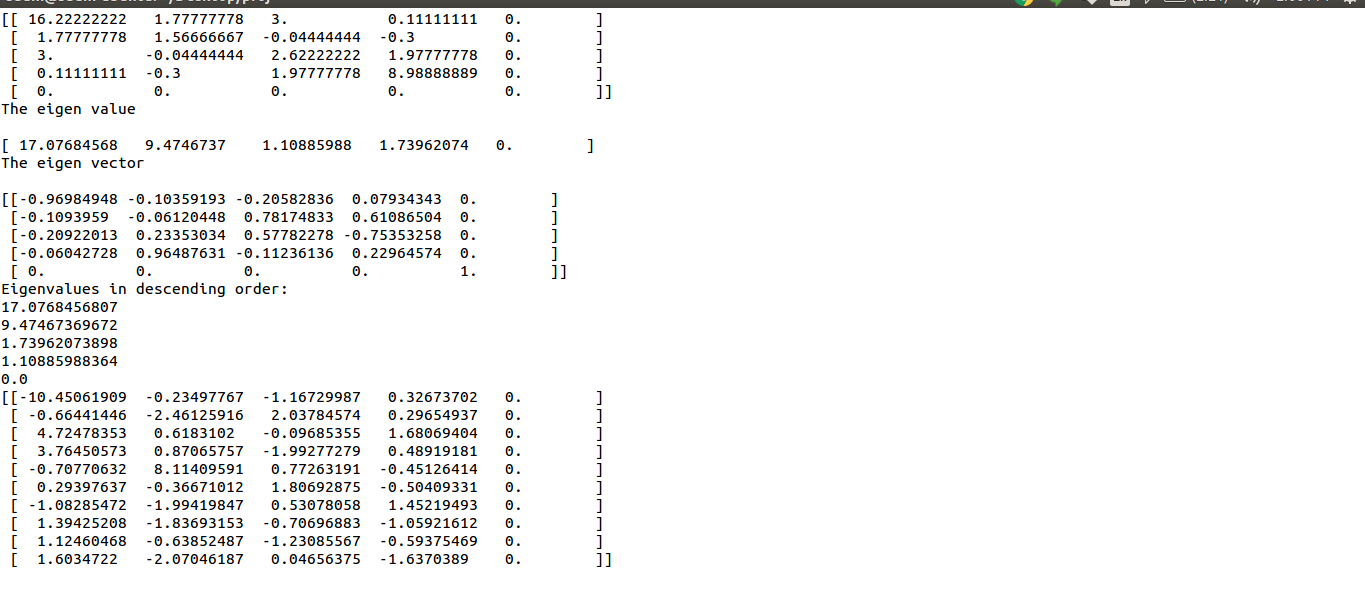


Screenshot of extracted dataset--



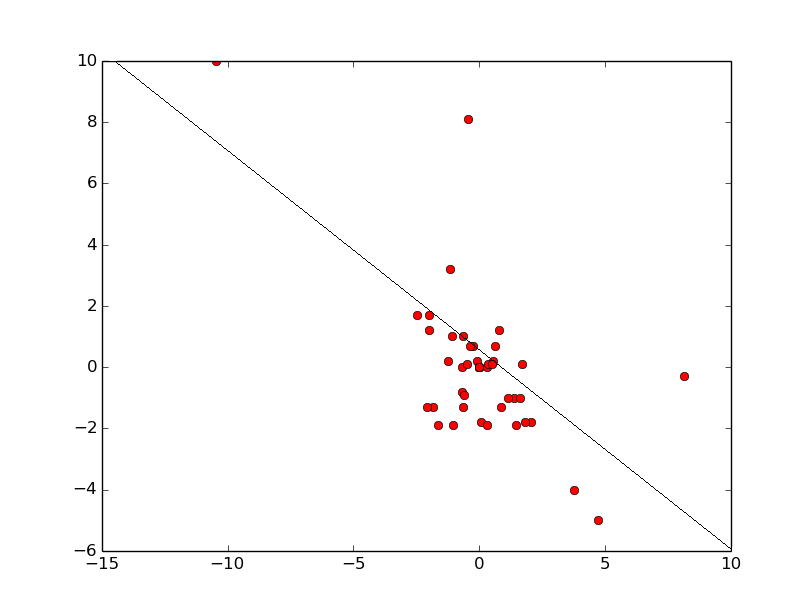
**5.1.2 Test PCA**

Screenshot of result of PCA—



**5.1.3 Test Logistic Regression**

Screenshot of result of log reg--



**CHAPTER 6- CONCLUSION & FUTURE**

A model was proposed using Principle Component Analysis and Logistic Regression for modelling the normal user browsing behaviour for detecting the application layer ddos attack traffic, if any, from the upcoming traffic. To model user behaviour, different features were constructed to differentiate between attacker and normal user. The optimal or near optimal features were selected from newly constructed feature set and used along with existing feature set. The proposed method was tested for different generated datasets. From the experimental results it is evident that the proposed method effectively classifies the attack traffic from the normal traffic.

**CHAPTER 7- FUTURE ENHANCEMENTS**

This project was designed and tested on a dummy data set. It can be further implemented on real-time attack data sets with more number of attacks happening at the same time as the algorithm is working. The future of this research lies in increasing the efficiency of the process and working more on zero day attacks and patching the vulnerabilities.

**CHAPTER -9 REFERENCES**

1. https://en.wikipedia.org/wiki/Denial-of-service\_attack
2. http://www.economist.com/blogs/economist-explains/2015/03/economist-explains-25
3. https://besthostingsearch.net/tutorial/what-is-difference-between-dos-and-ddos-attacks/
4. https://www.incapsula.com/ddos/ddos-attacks/denial-of-service.html
5. https://www.dosarrest.com/ddos-blog/the-motivation-and-goals-behind-ddos/
6. http://www.calyptix.com/top-threats/ddos-attacks-101-types-targets-motivations/
7. http://ddosattackprotection.org/blog/types-of-ddos-attacks/
8. https://www.incapsula.com/ddos/ddos-attacks/
9. https://www.arbornetworks.com/research/ddos-resources
10. http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html

**APPENDIX**

**LIST OF ABBREVIATIONS**

DoS - Denial of Service

DDoS - Distributed Denial of Service

US-CERT - United States Computer Emergency Readiness Team

IP - Internet Protocol

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

ICMP - Internet Control Message Protocol

SYN - SYNchronize

DNS - Domain Name System

HTTP - Hypertext Transfer Protocol

SMTP - Simple Mail Transfer Protocol

CRM - Customer Relationship Management

DDS - DoS Defence System

IPS - Intrusion-Prevention System

ASIC - Application Specific Integrated Circuits

RBIPS - rate-based IPS

WAN - Wide Area Network

ACL - Network Access Control List

PCA - Principle Component Analysis