ISSN:0975-9646

A Novel Methodology for Implementing a DDos Attack and Prevention

Rachna Tewani
ECE Department
Bharati Vidyapeeth's College of Engineering, New Delhi

Saumya Singh
Information Technology
Bharati Vidyapeeth's College of Engineering, New Delhi

Arun Kumar Dubey
IT Department
Bharati Vidyapeeth's College of Engineering, New Delhi

Abstract-Distributed Denial of Service (DDoS) attacks are being used with increasing frequency and force to destabilize internet based resources across the globe. The concept of the attack is to overwhelm the resources of an internet website or cloud resource with spurious requests so that it can no longer cater to the requests of genuine users. Distributed attacks are implements with the help of botnets which are networks of compromised computers throughout the world that work as robots for the master of the botnet or the main attacking agent.

Botnets are a serious threat to the internet and its resources. The botnet is used to perform a broad range of black and grey hat hacking methods from spamming to large scale attacks on networking based defense systems or major financial institutions. Botnets are also extensively used in credit card related crimes and multiple forms of spying. The role of botnets in implementing distributed denial of service attacks is particularly pivotal. The botnet in essence is the platform on which a denial of service attack is launched. The botnet supplies variety and numbers (in terms of computers) to the attacker.

General Terms-Algorithm for Preventing DDoS

Keywords-DDoS, Internet Relay Chat (IRC) , Botnet.

1. Introduction

A 2010 report by Dhamballa, offers alarming statistics on the rate of proliferation of botnets across the globe.[1] The report suggests that during the year 2010, the rate of increase in the number of bots on the internet grew by a steady 8 percent. The same study also suggested that almost half of the bot population was part of the ten largest networks on the web, implying that the power to vandalize internet resources lay in the hands of a small group of hackers or at least a limited set of hacking communities.[2] The most recently recorded major botnet attack was performed but a Dutch web hosting company in March 2013. The attack was so huge in magnitude that data stream surges grew to as much as 300 billion bits per second during its execution. The botnet industry has witnessed a slow and steady grow over the past decade. The first ever botnets were spotted in 1999, during that time botnets were designed and managed by networking aces who could manipulate or modify protocols and IRC environments to eavesdrop on information sources. This later grew into another branch of bot management now popularly known as centralized command and control.[2]

This paper first offers an overview of bots and botnets. It then tries to differentiate between the multiple methods of implementing network control infrastructures. This is followed by a simple implementation of a botnet system and a DDoS attack.[1] The primary emphasis is on inter-bot communication and the implementation of remote bot control. Finally, the paper analyzes various available methods of preventing and mitigating botnet attacks.

Popular definitions identify bots as a piece of software that auto runs computer operations over the web through the use of the host computer's resources. Bots are designed to perform simple and repetitive tasks such as sending spoofed package requests to DNS servers on the web etc.[1] Web spidering is one of the most popular applications of bots. The process involves a computer repeatedly and rapidly accessing multiple web servers and retrieving relevant information. Applications for bots are also found in areas where human actions have to be imitated or when automated tasks have to be performed at a speed higher than that of humans.[2] Bots came into existence during the era of inter relay chats and instant messaging. The bots were primarily needed for communicating with human users and reporting details that could be easily formatted, for eg: report about the day on a particular date of the temperature at a particular location. Software like SmarterChild were developed as application of bots that could function as interactive child games.[1] The widest application of bots however was found in eavesdropping on internet relay chats. Bots could be used to censor out specific words and offer help to new users.

Commercially, bots have been used by third party websites to analyze bargains and other price related information available on bidding and shopping websites like eBay and Betfair to automate the process of locating best offers.[1] This has attracted a lot of negative legislation from target companies. Betfair is one of those websites that ended up launching bot monitoring and managing software at its end to control third party website sending requests or analysing information hosted by them.

The earliest known bot is the SDBot, a C++ based program code that later became known as the SpyBot. The program and its advanced versions were designed to target vulnerabilities in Microsoft remote procedure calls. During this period the proliferation of botnets was fueled by various software vulnerabilities in popular operating systems and other computing platforms.[1] The bots and botnets that ensued had capabilities ranging from performing coordinated spamming to denial of service attacks. Thus, began a rather advanced era in the science or technology of botnets when the success of any botnet became heavily dependent on the use of networking, encryption and internet communication.

The most important development was probably the use of peer to peer systems to control and operate bot networks.[2] Storm Worm somewhere around 2007 became the most powerful architecture of this type. Present day needs for sophistication in bots springs from the wide variety of technological solutions present in the white market today.[1] Broadly speaking the complexity level of botnets has gradually increased over time and begun requiring extensive knowledge of different domains of computer technology. The process of creating a bot has graduated from being a one man show to a sophisticated group initiative with covert goals.

2. Related Work

The literature on botnets and DDoSing is wide ranging and diverse however the aim of this paper is to undertake a study of the present state of botnet implementation. We have subsequently also implement a DDoS attack on our own website and used prevention algorithms to prevent users from making such attacks. In this paper we have undertaken the task of creating a simple DDoS attack from scratch. The idea has been to study the methods used in creation of a command and control botnet and apply one of the simpler prevention algortihms to prevent such attacks.[3] The purpose was to understand how DDoSing changes the traffic on a website and can be detected through information like MAC addresses and IP addresses and has helped us in understanding the basic technology that goes into botnet creation.

Generally, simpler botnet implementations are designed around a command and control type bot-master that implements basic level encryption and manages an entire network of bots.[4] The C&C is capable of tracking the presence and activity of bots as well as receiving updates about their network and other parameters from them. The

C&C is capable of giving sleep, scan and attack instructions to the bots. The bot program is designed to complement the operations of the command and control structure and essentially performs all the tasks related to reporting to the C&C, analyzing the network and making the attack. The bots however are not always designed to remain hidden on the host system. Functions related to hiding have been considered too sophisticated for research in a number of papers and have been dealt with only in the theory portion broadly.

3. Methodology

3.1 Simulation

The mechanism for executing a DDoS attack is to repeatedly ping a website with data or data requests. The more the number of accesses the more the load on the website. It is also generally observed that it is unusual for a user to ask for the same information or web page periodically and in quick succession. Taking these observations under consideration we have designed an algorithm that uses a database to verify the access statistics of each users and bars a user, access to the website after a certain type of behavior is spotted and verified. The primary motivation behind implementing DDoS attacks is to increase the target's down time and capitalize on that result in some manner.

To implement this solution a database is created at the back end of the website. This database essentially records users particulars such as the IP address, time of access and the likes. The idea is to identify a user as single and unique and to record his activity upto a period of 5 minutes after first access. The database is designed such that it flushes entries that have no activity for 5 minutes.

3.2 Unix Command

```
Watch –n 0.2 "GET http://www.bvcoe.info"
```

Command Name : watch

Runs an application repeatedly, also displaying a full screen output. The effect is that the output changes can be monitored continuously and the changes over time can be detected with ease.[5] The program is normally run at a gap of 2 seconds however the interval can be changed.

Command Name : get

The program sends data packet requests to servers. The content of the POST is obtained from stdin and the output is sent to stdout. The most important function of the program is that it returns information about the number of URLs that have returned an error due to any particular reason.

3.3 Iframe Brute Force Attack

```
<meta http-equiv="refresh" content="5">
<iframe src="http://www.bvcoe.info" width="0" height="0">
```

Tag Name: iframe

This command is used to display a web page within a web page. The URL points to the location of the separate page. An inline frame is used to embed another document within the current HTML document. The <iframe> tag is supported in all major browsers. The parameter "height", specifies the height of an <iframe> and the parameter "width" specifies the width of the <iframe>. Both these parameters are set to zero to hide the script and its role. This construct allows for attack multiplication. Each user that accesses the website sends requests to the server of this other website without the slightest knowledge that his/her bandwidth is being used to ping the website with illegal requests.

Tag Name: meta

The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable. Http-equiv, provides an HTTP header for the information/value of the content attribute. The entire construct automatically refreshes the page after a specified duration of time. In this case this duration is 5 seconds but this can be shortened further.

3.4 DOS Command

ping –t –l 40000 198.58.84.147

Command Name : ping

Ping is an abbreviation that stands for Packet InterNet Gopher. The command aids in finding the IP of another host and is also used seamlessly to check internet connections and resolve connectivity issues. The command verifies whether a particular data packet can be delivered to another host on the internet. The present execution using ping is not for determining the IP address but only for sending data packets to the server being attacked. The request passes over the internet and its various nodes and if the route to the host is clear the ping is successful. The flood of ping packets then overwhelms the target's server. The ping message is replied by a pong message once ping is received. The time between the ping and the pong is used to find the average response time. The ping is generally not received when there are errors in the configuration settings of the network of if the sender has been blocked. The command has been used in the following form:

ping [-t] [-a] [-n count] [-l size]

Here the various tags stand for:

-t: is used to ping a particular host until it is paused. A combination of CTRL and Break allows for the generation of statistics and continuation of the ping string. The pinging process can be stopped completely by pressing CTRL and C in combination.

-l: is used to set the length of the echo request packet in terms of bytes. The number can be anything between 32 and 65527. By default the packet size is simply 32 bytes.

4. EVADING DDOS ATTACK

Controlling the Number of Visits and Updating Database

1. The number of allowed visits to the website are limited to two for simplifying execution.
2. We obtain the IP address of the agent trying to access the website and then check if the user is already registered in the database. At this stage particulars such as user's IP address, user's date and time of login are recorded.
3. If IP address is identified as a repeat access-or its time of visit is updated.
4. Else it is understood that the user is either accessing for the first time of accessing after a reasonable gap.
5. At this level we perform a check for whether there is room to accommodate the new user entry
6. If there is space the user is accepted and a new entry in made in the database to register the user otherwise the user's request is rejected and he is not allowed to access the website.
7. Since the user end does not send in data about leaving the website, an elapse of a period of 30 minutes is assumed to be enough to conclude that the user has left the website. On the occurrence of such an event the user's record is deleted from the database.

Identifying DDoS and Displaying Results

1. Through the use of cookies the time interval between two consecutive visits by the user is detected.
2. The minimum number of seconds between visits is set as 3 on the first visit of each user.
3. If the user tries to access the website before the elapse of the set time period, he is spotted and a penalty is levied on him, which may range from anything between 30 seconds and two minutes or more.
4. In this case the user is guided to another page that tells him that he has been blocked temporarily and will be able to use the website's service after the elapse of a certain time.
5. Then the specifics of the user are recorded in the website's database of recent users, as stated in the algorithm above.

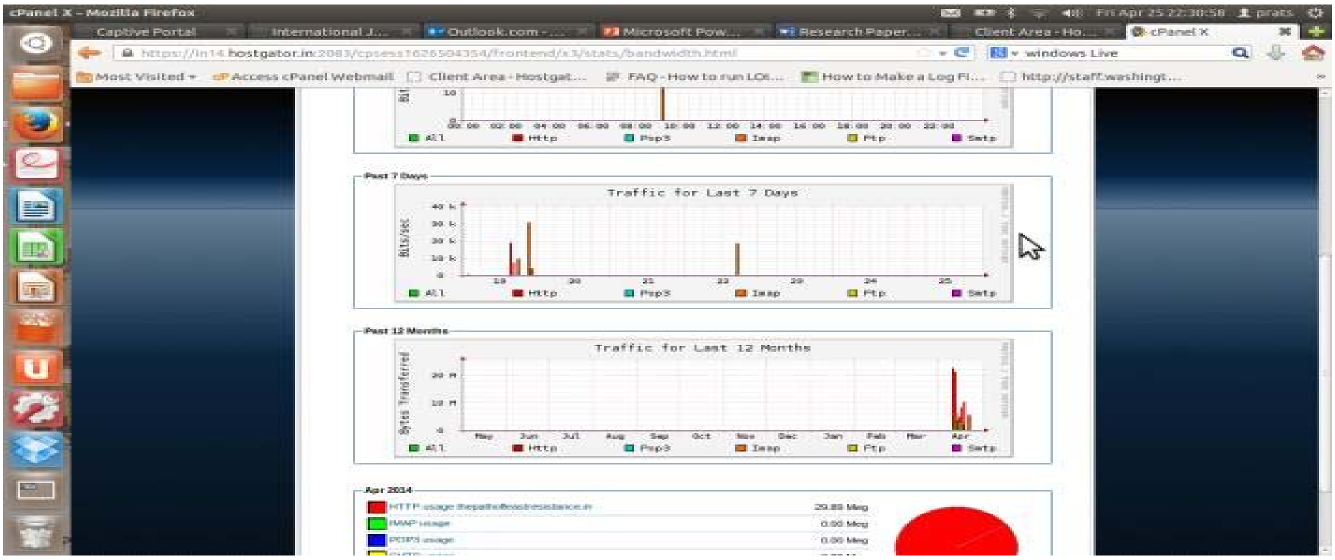
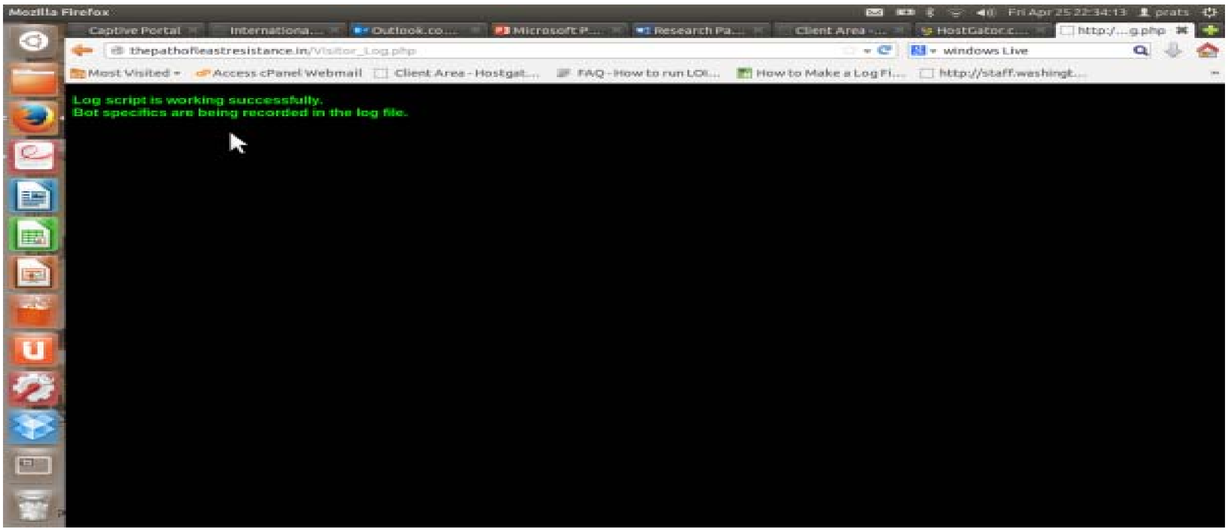


Fig 1: Attack on Specified Website on 19th April 2014



```
Log script is working successfully.
Bot specifics are being recorded in the log file.
```

Fig 2: Attack Detected and Message Displayed

5. Result

The attack on the website "thepathofleastresistance.in" was performed using 3 different methods. The first method used a simple MS-DOS command to repeatedly ping the website server with large bytes of data. The intention behind this type of attack was to overwhelm the resources of the server by sending high amounts of data, quickly and repeatedly. The second method used a set of HTML commands to create a simple script that when attached to another websites code would initiate an attack.[6] The HTML code creates a zero by zero panel in the browser that contains a particular page of the attacked website. This panel is refreshed at short intervals, periodically. Any person trying to access this malicious page on the attacker's website uses his internet bandwidth to not access the information on the web page he has requested but also the information on a web page of attacked web site. The third method is deployed on LINUX systems.[7] The terminal command is used to make access requests to a certain web page of the attacked website on a periodic basis. All the three methods have been designed to work in an infinite loop, automatically. They are generally triggered through the initiation of an application that the user wants to run willfully. This method required the use of a webhosting and domain naming service to obtain results and attack statistics.

6. Conclusion

We have analyzed DDOS attack on the website "thepathofleastresistance.in" using above three method and try to prevent this attack using evading DDOS attack methodology on website which is shown in our result too.

References

1. John Ioannidis, Steven M. Bellovin - "Implementing Pushback: Router-Based Defense Against DDoS Attacks ", AT&T Labs Research, USA.
2. "A Summary of DoS/DDoS Prevention, Monitoring and Mitigation Techniques in a Service Provider Environment", SANS Institute, USA.
3. Alefiya Hussain, John Heidemann, Christos Papadopoulos - "Identification of Repeated Denial of Service Attacks ", Information Sciences Institute, CA, USA.
4. Tao Peng ,Christopher Leckie, Kotagiri Ramamohanarao - "Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring", Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria, Australia .
5. Khaled M. Elleithy , Drazen Blagovic, Wang Cheng, Paul Sideleau - "Denial of Service Attack Techniques: Analysis, Implementation and Comparison", University of Bridgeport, CT, USA.
6. "Online Dispute Becomes Internet Snaring Attack", New York Times Report, URL: http://goo.gl/d675Ae
7. Zheng Bu, Pedro Bueno, Rahul Kashyap, Adam Wosotowsky, "The New Era of Bots", McAfee, URL: http://goo.gl/HkkJm3