Forensics of Random-UDP Flooding Attacks

Anchit Bijalwan
Department of Computer Science and Engineering, Uttarakhand Technical University, Dehradun, India
Emil: anchit.bijalwan@gmail.com

Mohammad Wazid
Center for Security, Theory and Algorithmic Research, Indian Institute of Information Technology, Hyderabad, India
Emil: wazidkec2005@gmail.com

Emmanuel S. Pilli
Department of Computer Science and Engineering, Malaviya National Institute of Technology Jaipur, India
Emil: espilli.cse@mnit.ac.in

R.C. Joshi
Chancellor, Graphic Era University, Dehradun, India
Emil: Chancellor.geu@gmail.com

Abstract—Internet has great impact on various facets of everyone’s life. With the enormous advantage Internet provides to users all around the world, it has some inherent weaknesses because of the protocol stack on which it is built. It can be easily attacked by attackers who exploit the vulnerabilities in the protocols and compromise systems and remotely control them to do further damage. Major attacks are focused on confidentiality, integrity and availability of data or resources.

Flooding attack is one such resource availability attack which is a great cause of concern. Hackers can use the flooding attacks and cause Distributed Denial of Service (DDoS) attack with ease. With the increase and variations in the attack mode makes the investigation of these attacks essential. Random-UDP flooding attack is a different type of attack in which the attacker sends multiple UDP datagrams of different sizes at a time. This causes denial of service to the system and its resources.

In this paper, we have proposed a technique for the forensics of Random-UDP flooding attack. We have tried to get as close as possible to the source of such attacks. The proposed technique is capable to identify the source of Random-UDP flooding bot attack.

Index Terms—Security, DDoS, Flooding, Zero day attack, Bot, Random-UDP flooding

I. INTRODUCTION

Internet is the important part of modern life and is a collection of communicating computers. People connect over the internet with each other and share the information. Internet also facilitates the access to and utilization of the resources around the globe. Though internet has become very useful, it has some inherent weaknesses because of the protocol stack on which it is built. It can be easily attacked by attackers who exploit the vulnerabilities in the protocols and compromise systems. The information exchange over the internetwork is not secure as the protocols used were not designed with security in mind. Sometimes these resources are unavailable to the authorized users, because of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. A compromised system or host (also called a bot) floods the network with large number of packets in a short span of time, which further causes DoS attack.

Various types of flooding attacks using protocols such as HTTP, UDP etc are popular these days.

Intrusion detection/detection and prevention systems (IDS/IPS), firewalls are used to prevent flooding attacks. Some of the deployed IPSs are able to prevent these attacks. But the scenario can be different if attacker varies the flooding characteristics such as variation in packet size etc. They can easily bypass the deployed appliances.

The existing solutions are not sufficient for the investigation of these attacks. The flooding attacks are launched very easily at short intervals of time and Random floods which are generated through zero day attacks have different types of characteristics. Neither the IDS nor the firewall detect the flood, because these appliances work on the basis threshold based flood detection criteria.

In this paper we have proposed a framework for investigation of Random-UDP flooding attack. The framework has three different stages: normal flow of packets, flooding attack and flood source identification. Our contributions in this paper are outlined below.

• We propose a framework for the forensic analysis of random UDP flooding attacks.
• During the forensics analysis process, the technique successfully identified the source of random UDP flood and a corresponding report is generated.
• The proposed technique is evaluated and a performance comparison table is also provided.

The rest of the paper is organized as follows: In section II we review the existing techniques. The proposed
framework is presented in section III. The implementation details and results are explained in section IV. The work is concluded in section V along with scope for future work.

II. RELATED WORK

Hussain et al. [1] showed the effect of UDP flooding on the performance of the number of queuing algorithms like Droptail (DT), Random Early Discard (RED), Deficit Round Robin (DRR), Fair Queue (FQ) and Stochastic Fair Queue (SFQ) is measured. During the experimentation, it has been observed that SFQ performs better for UDP traffic as compared to the other schemes. Yuan Tao et al. [2] proposed DDoS attack detection scheme for local area networks. Flow entropy is employed on the LAN routers to supervise the traffic and to raise the potential flooding alarms. A information distance is used differentiate between false alarms and DDoS attacks. The Mathematical models are implemented for the proposed detection schemes. During the experiments, it has been observed that the proposed schemed is very effective to detect the DDoS attacks. Moustis et al. [3] analyzed DDoS attacks that require only a small number of bots to make a web server unavailable. The bots are simulated by using both Windows and Linux based systems infected with Slowloris (HTTP syn-flooder), targeting to a web server. Several security controls are also applied to test the effectiveness of proposed method against such attacks. In simulations, it has been observed that a combination of carefully selected anti-DDoS controls can reduce the exposure of flooding attack. In Bardas et al. [4], authors present the investigation of proportional-packet rate assumption. The classification of UDP traffic is done, the objective is to detect malicious addresses that cause UDP flooding attack. In the experiments the dataset is created by taking data from ISPs, universities, financial institutions, etc. A prototype classifier is implemented and a method is also discussed, how it can be used to prevent the UDP flooding attacks. Silva et al. [5] reviewed on botnet problem. Author summarized the previous work related to botnet attacks, the problems and some solutions to those problems are also discussed. The open prominent and persistent research problems of botnet are also discussed. The HTTP GET flooding based DDoS attacks increase day by day. So in Kim et al. [6] proposed the web server protection mechanism against HTTP GET flooding attacks. The proposed technique can easily detect HTTP GET flooding attacks. It is implemented in Gigabit Ethernet secure Network Interface Controller (GESNIC) for the high performance DDoS attack prevention. GESNIC protects the Internet servers against various DDoS attacks. GESNIC provides high performance secure logics, a kind of security offload engine to protect against TCP and HTTP related DDoS attacks. Its performance is almost a carrier-class level because its latency time $7 \times 10^{-6}$ seconds. The installation of GESNIC can make the system more secure, highly available and easier to manage. Chuiyi et al. [7] classified the three flooding DoS attacks. A distributed architecture is described and local and global communication methods are proposed, which reduce the overhead produced by fully distributed architecture. In the performance analysis, it has been observed that F-DIDS works effectively. Mohay et al. [8] focused on research on attack detection, with some discussion of mitigation through IP address filtering. It describes the leading edge work on the development of detection techniques capable to identify a high-rate flooding attacks.

There are two types of characteristics of bandwidth attack. The first characteristic is, if during an attack the incoming traffic rate is very high as compare to the outgoing traffic rate. Second is, if the proportion of protocol exploited by the attacker is higher than the exploitation of the other protocols in the traffic. On the basis of these characteristics, a UDP bandwidth attack detection system based on Protocol Share Based Traffic Rate Analysis (PSBTRA) is proposed in Ihsan et al. [9]. In the experiments, it has been observed that the proposed method can effectively detect UDP bandwidth attacks. Vural et al. [10] proposed botnet identity concealment techniques. In order to detects botnet computational intelligence techniques are proposed. A simulate for network anomaly detection is done. In Mansfield et al. [11], a discussion on botnet and whitehats is done. There is a continuous arms race between botnet operators and the whitehats (researchers), anti-malware organization and law enforcement organizations. The most visible action of this conflict is the malware, but there is a less obvious struggle going on to control the infrastructure, supports the unauthorized actions of botnet operators. By the application of malware, the botnet operators can build and manage their infrastructures more effectively, as seen in the past few years. In Rui et al. [12], an artificial immune detection based defense system against UDP flooding attack is proposed. The r-bits matching rule is introduced with eigenvalue matching scheme. The all non self modes are detected by the application of eigenvalue filter windows. In simulation, it has been observed that the proposed defense system detects the fake IP addresses from UDP flooding successfully. In Argyraki et al. [13], proposed an Internet traffic filtering (AITF), a network-layer defense technique against bandwidth consuming flooding attacks. The proposed scheme enables a receiver to contact to the misbehaving source and ask him to stop the flooding traffic. The each flooding source that has been asked to misbehave source and ask him to stop the flooding traffic. The first characteristic is, if during an attack the incoming traffic rate is very high as compare to the outgoing traffic rate.
traffic model can be used to detect large-scale flooding attacks (DDoS). During the experimentations, it has been observed that the proposed Soft Computing based detection technique can successfully detect large scale flooding attacks. Safaa et al.[15] proposed a defense mechanism against SYN flooding is proposed. It makes the use of spoofed IP addresses associated with edge routers to determine whether the incoming SYN-ACK segment is valid or not. A matching table of the outgoing SYN's and incoming SYN-ACKs is maintained. If the incoming SYN-ACK segment is invalid, the edge router resets the connection at the victim host, freeing up an entry in the victim’s backlog queue, and enables it to accept other legitimate incoming connection requests (RQ). The performance evaluation of the proposed technique is also done. Park et al.[16] proposed an SNMP-based lightweight and fast detection technique for traffic flooding attacks. It minimizes the processing and network overhead of the intrusion detection system, the detection time, and provides high detection rate. AITF protects the network against the flooding and also reduces the bandwidth consumption. It is also shown that, two networks deployed with AITF scheme can maintain their connectivity to each other in the presence of flooding. Takemori et al. [17] proposed an IP tracking scheme against bot attacks using the DNS logs.

III. PROPOSED FRAMEWORK

Many authors have worked on flooding attacks and provide the detection techniques for these attacks. UDP flooding is a type of flooding attack in which multiple UDP datagrams are generated by a malicious system (bot). These UDP datagrams flood across the network, and when they reach at a system causes congestion there. This further causes denial of service (DoS) attack. The flooding attack scenario becomes more hazardous, if we change the characteristics of flood. The attacker system (bot) generates multiple UDP datagrams of different sizes. These UDP datagrams flood easily across the network without any restriction because every detection system works on the basis of two criteria: First is signature based detection and second is feature based detection. Though they are UDP datagrams so their signature seems safe, they easily pass the detection system. An intrusion detection system (IDS) also does the task of feature based detection. In feature based detection, it matches the features of packets flow to the feature of a safe packets flow. If they do not match it stops those packets. Here, we have designed a malicious script by using a randomized function, runs on a bot. It sends the multiple UDP datagrams of different sizes to a victim system. These datagrams are captured and analyzed.

The methodology of the work is divided into three parts:

- We conducted experiments involving normal flow of UDP datagrams.
- We launched a DoS attack using Random-UDP flooding.
- We finally conduct a forensic analysis of Random-UDP flooding attack.

A. Normal Flow of UDP Datagrams

In this scenario, we have an internet user, who wants to book a tour package from an online booking website, he searches for that and gets a link (www.fasttoursandtravel.com) showing the cheapest rate, he attracted towards that link and click on that. He successfully books the tour package. Figure 1 shows, the request (HRQP-HTTP Request) and response (HRSP-HTTP Response) coming from the web server.

B. Random-UDP Flooding Attack

If the web server (www.fasttoursandtravel.com) becomes bot, the user who wants to book a tour package can be easily victimized. User sends HTTP request for the web page. At web server (bot) a malicious script is running, extracts the IP of the user from the request and starts sending of UDP datagrams of different sizes. Figure 2 shows, the request (HRQP-HTTP Request) and response (HRSP-HTTP Response) flood coming from the malicious web server (bot).

C. Forensics of Random-UDP Flooding Attack

The methodology of forensics of Random-UDP flooding attack is described here. The Random-UDP flooding attack happens to the user 1’s system. We want to investigate the flooding attack. For this purpose, we have started packet capturing using Wireshark forensics tool. User 1 sends HTTP request for the web page. At web server (bot), malicious script starts sending of UDP datagrams of different sizes. These datagrams are captured and analyzed.

The methodology of the work is divided into three parts:

- We conducted experiments involving normal flow of UDP datagrams.
Figure 3 shows, the request (HRQP-HTTP Request) for the web page. Figure 4, shows the response (HRSP-HTTP Response) flood coming from the web server. User 1 has deployed appliance which starts packet capturing. Figure 5 depicts the steps involved in Random-UDP Flooding Attack. Figure 6 depicts the steps involved in flooding packets capturing. Figure 7 depicts the steps involved in flooding packets analysis.

IV. IMPLEMENTATION AND RESULTS

A. Network Setup and Experimentation

For the experimentation purpose we have created a network by using some systems. The following network components are used:

<table>
<thead>
<tr>
<th>TABLE I. COMPONENTS USED IN EXPERIMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Components</td>
</tr>
<tr>
<td>Number of Users (Under Normal Case)</td>
</tr>
<tr>
<td>Number of Users (Under Attack)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Number of Users (Under forensics process)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Number of deployed bots (Attacker)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II. CONFIGURATION OF BOT (ATTACKER)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
</tr>
<tr>
<td>Operating System</td>
</tr>
<tr>
<td>RAM</td>
</tr>
<tr>
<td>Processor</td>
</tr>
<tr>
<td>Clock Speed</td>
</tr>
<tr>
<td>IP Address</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III. CONFIGURATION OF USER’S SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
</tr>
<tr>
<td>Operating System</td>
</tr>
<tr>
<td>RAM</td>
</tr>
<tr>
<td>Processor</td>
</tr>
<tr>
<td>Clock Speed</td>
</tr>
<tr>
<td>IP Addresses</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

For Random-UDP flooding attack, the following programs are coded, run at the bot (attacker).

Flooding Attack Algorithm

//host IP address/URL of victim system
//ta in mseconds attack duration
//UDP_flood is program to flood the network
1. Start flood_attack
2. Call UDP_flood()
3. Terminate
4. END

UDP Flooding Algorithm

//tpko packet timeout, reestablish a connection if timeout
//sk open socket
// rand randomizer function for selecting different ports for each packet
//RN is used to store the values return by rand
//udp selected flooding packet protocol
//pk=0; packet counter
//exec_time= ta; attack duration
//tin to input the current system time
//tstop=ta + tin ; attack stops time

1. while (1)
   // for infinite loop
   pk=pk+1;
   if (tin greater than tstop)
      break;
2. RN=rand();
   //It returns port number where attack is to be performed
3. sk(udp,host, RN, tpko)
   // open socket by passing parameters
4. Send udp packets and continues

Flooding Source Identification Algorithm
//ip.src source IP address
//IP IP address of the source sending packets
//inf.ip indentified IP of flooding source
//proto protocol field value
//pkt number of packets
//pkt_sz packet size
//rand.pkt_sz random packet size
//det_fl detected flood type
//rand.udp_fl random udp flood
//nc.udp_fl normal udp flood
//pkt_recv received packets
//pkt_rcp_thrsh packet reception threshold

If (ip.src = = IP and proto = = udp
   and pkt_recv > pkt_rcp_thrsh)
then
   for pkt 1 to n
      if pkt_sz == rand.pkt_sz then
         det_fl = rand.udp_fl and
         inf.ip = IP
      else
         det_fl = nc.udp_fl and
         inf.ip = IP

When user 1 sends request for www.toursandtravel.com, its web page opens into the user 1’s system. In this case the total communicated (sending and receiving) packets are 192. The total response packets sent by the web server are 97. Figure 8, depicts the Random-UDP datagrams sent by the bot (web server of www.fasttoursandtravel.com, IP 192.168.1.2). In the other scenario when user 1 sends request for www.toursandtravel.com, its web page opens into the user 1’s system. But his request also activated the malicious script at the bot, which extract his IP address from his HTTP request and starts flooding to his system. In this case the total communicated (sending and receiving) packets are 8895. The response packets (under attack) sent by the bot the sent packets are 8824.

B. Analysis and Results

Deployed tool captures all the packets coming to the system. After the packet capturing, the packet analysis phase is started to detect the source of Random-UDP flooding. The outcomes of packet analysis phase are as follows:

<table>
<thead>
<tr>
<th>TABLE IV. NETWORK STATISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Flow</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>Total Packets</td>
</tr>
<tr>
<td>Request Packets</td>
</tr>
<tr>
<td>Response Packets</td>
</tr>
<tr>
<td>Random-UDP flood datagrams</td>
</tr>
</tbody>
</table>

Figure 7. Flooding Packets Analysis

Figure 8. Flooding UDP datagrams sent by malicious web server (www.fasttoursandtravel.com, IP 192.168.1.2)

Figure 9. Network Statistics under different flows
Figure 8, depicts the Random-UDP datagrams sent by the bot (web server of www.fasttoursandtravel.com, IP 192.168.1.2). The total datagrams sent by the bot are 8804. The datagrams are of different sizes such as 126, 124, and 120 bytes. Figure 9, depicts the network statistics under normal flow and under attack.

In the forensics procedure of Random-UDP flood attack the following report is generated:

**Table V. COMPARISON WITH AVAILABLE TECHNIQUES**

<table>
<thead>
<tr>
<th>Available Techniques/Flooding types</th>
<th>S. M. Hussain et al</th>
<th>D. Moustis et al</th>
<th>Hyoungoo Kim et al</th>
<th>Xu Rui et al</th>
<th>Haider Safa et al</th>
<th>Jun-Sang Park et al</th>
<th>Anchi, Wazid, E. S. Pili</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>TCP</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HTTP</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SYN</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Random-UDP</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**D. Performance Comparison with Existing Techniques**

Table V shows the comparison of proposed technique with available techniques. We compared our technique with Hussain et al. [1], Moustis et al. [3], Kim et al. [4], Rui et al. [12], Safa et al. [15] and Park et al. [16]. Our technique has successfully investigated the source of random UDP flood. By tuning the parameters of flooding source identification algorithm the technique can also be applied for other floods such as HTTP, SNMP etc.

V. CONCLUSION

Internet is one of the essential requirements of society, but it can be easily attacked. Compromised hosts act as bots which further cause the unavailability of resources and services. Flooding attacks can easily cause denial of service attack. Random-UDP flooding attack is a different type of attack in which the attacker sends multiple UDP datagrams of different sizes at a time. Here, we have investigated Random-UDP flooding attack and tried to find out the source of attack and network behavior under the attack. In the presence of this attack, the system is flooded by multiple UDP datagrams, which causes denial of service to the system and its resources. By the help of proposed technique we can easily indentify the sources of such attacks. During experimentation, it has been observed that the system (IP 192.168.1.2) is a bot which floods the users systems with random sized UDP datagrams. This work can be further extended to design a learning based framework for the forensics of other floods generated by zero day attacks.

**REFERENCES**


Anchit Bijalwan is a Research scholar at Department of Computer Science & Engineering, Uttarakhand Technical University, Dehradun, India. His research interests include network security, botnet forensics.

Mohammad Wazid is a Research scholar at International Institute of Information Technology (IIIT), Hyderabad, India. He received the Young Scientist Award by UCOST, Department of Science and Technology, Government of Uttarakhand, India. His research interests include network security.

Emmanuel S. Pilli has completed his Ph. D. from Indian Institute of Technology Roorkee, India. He is an Assistant Professor in the Department of Computer Science and Engineering and is also the Adjunct Faculty, National Center for Disaster Management and Mitigation, in the same institute. He has teaching and research experience of over 15 years.

He actively participates in the professional societies like IEEE, ACM and Cloud Computing Innovation Council of India (CCICI). He is reviewer of reputed Transactions, Journals and Conferences. His areas of interest are Security, Privacy and Forensics; Cloud Computing, Big Data and Internet of Things (IoT).

Prof. R. C. Joshi is a Chancellor at Graphic Era University, Dehradun, India. He is a former Professor in the E & CE Department at IIT Roorkee, India. He received his BE in 1967, ME and Ph.D from IIT, Roorkee, India in 1970 and 1980 respectively.

He has a teaching experience of over 45 years and has guided more than 30 Ph. Ds. He had been a PI for four projects with DST and DIT. His areas of interest include Data mining, Security, Forensics, Reconfigurable Computing, Distributed and Parallel Computing, Wireless Sensor Networks, Cloud Computing, and Big Data.