Protocol Share Based Traffic Rate Analysis (PSBTRA) for UDP Bandwidth Attack

Zohair Ihsan, Mohd. Yazid Idris*, Khalid Hussain, Deris Stiawan, and Khalid Mahmood Awan

Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Skudai, 81310, Johor. Malaysia
{izohair2,hkhalid2,sderis2,makhalid2}@live.utm.my, yazid@utm.my

Abstract. Internet is based on best effort and end to end design principles. Although they are the reasons for the Internet's high efficiency and popularity, they also resulted in many inherent security problems such as the Bandwidth Attacks. There are two main characteristics of bandwidth attack. First, during an attack the incoming traffic rate is much higher than the outgoing traffic rate. Second, the proportion of protocol exploited by the attacker is higher as compare to other protocols in the traffic. Based on these two characteristics, a UDP bandwidth attack detection system based on Protocol Share Based Traffic Rate Analysis (PSBTRA) is proposed. Experiments on real world network shows that this approach can effectively detect UDP bandwidth attacks.

Keywords: Distributed Denial of Service Attack, Bandwidth Attack, UDP Flooding Attack.

1 Introduction

Since the novel ideas of packet switching networks in mid of 1960's and the first packet switching network ARPANET in 1969 [1], computer networks have become highly important components of contemporary societies. At present, computer networks are use everywhere and this trend is not likely to change in the future. The ARPANET was created as a research network sponsored by the Advanced Research Projects Agency (ARPA) of Department of Defense (DoD) in the United States of America. The aim was to develop a communication network for researchers to share their research and that network is now commonly known as the Internet. The Internet's best effort and end to end design principles [2] along with the TCP/IP protocol suite [3] are major factors for the Internet's dominant success, but they also inherent security problems. Although the Internet has been proved extremely robust in cases of random failures, it has also been proven extremely sensitive to specifically targeted attacks [4]. This is due to the fact that Internet was not designed to be use in

* Corresponding author.

A. Abd Manaf et al. (Eds.): ICIEIS 2011, Part I, CCIS 251, pp. 275–289, 2011.

such a way it is being used nowadays, which leads to the poor security design. For instance, already in the late eighties [5], several security problems within TCP/IP protocol suite were pointed out.

One of the Internet's largest security concerns is its intrinsic inability to deal with denial of service attacks. The term denial of service referrers to a situation, where a legitimate requestor of a service cannot receive the requested service for one reason or the other. Denial of service (DoS) attacks are characterized by the attacker's primary intention to cause denial of service to the requestors of the service in question.

DoS attacks can very well be launched both locally and remotely and they range from software vulnerabilities to bandwidth consumption. A majority of DoS attacks can be countered relatively. For instance, attacks that target software can mostly be eliminated by patching the vulnerabilities but unfortunately, the number of vulnerabilities reported each year is increasing according to CERT statistics [6]. Hence, an attacker can control a large number of insecure systems by exploiting their vulnerabilities. Attacks that target network resources are more of a problem such as bandwidth attack. The bandwidth attacks are built within the principles of the Internet and thus it appears that any absolute solution would require a change in the principles themselves.

2 Denial of Service Attacks: The Concept

The Internet is based on best effort and end to end design principles and although they are the reasons for the Internet's high efficiency and popularity, they are also the sources of many inherent security problems as well. As discussed previously, DoS attacks exist due to this fact. The best effort principle accompanied with the end to end principle, means that the Internet's only concerned with the routing packets injected to it as fast as possible to the specified destinations, leaving everything else for the end hosts to handle [2]. This means that at the core level, the Internet is only concerned with what the IP portion of a packet embodies. The IP specifies the network level header according to which the users are ought to construct their packets in order to transfer data through the Internet [7]. In Internet this information is extracted from the packets, specifically from the IP portion and operations are performed on the bases of extracted information. Internet is not concerned with whom created the packets or from where they are coming and where they are heading. This means that everything else is left for the user to construct.

"Denial of service" and *"denial of service attack"* are two completely different concepts where the former refers to an event or a situation and the latter refers to an intention driven illegal act. [8] States, *"The most comprehensive perspective would be that regardless of the cause, if a service is supposed to be available and it is not, then service has been denied."* The definition of denial of service used in this paper was created on these bases. Denial of service is an event or a situation, in which a legitimate client cannot access the requested service to which the client is entitled to and which should be available to it.

According to [9], *"A denial of service attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service."* [8] States the same in a slightly more verbose manner, *"A denial of service attack is considered to take place only when access to a computer or network resource is intentionally blocked or degraded as a result of malicious action taken by another user."* With a slight modification, the definition provided by [9] is the definition for DoS attack used in this paper.

"A denial of service attack is characterized by an exclusive function of the attack and an explicit attempt by one or more attackers to prevent one or more legitimate users of a service from using that service."

With these modifications, the stress is on two important points. First, the number of targets or attackers is irrelevant. Second, the single purpose of the attack must be to cause a denial of service, which means that if the attack has any other functions besides causing denial of service, the attack cannot be categorized as a denial of service attack.

As some other attacks may cause denial of service situations as a side effect. For instance, it is common for viruses and worms to consume much of both host and network resources while propagating and executing their primary functions, this has often leaded to severe denial of service situations. These attacks cannot be characterized as DoS attack, unless their primary objective it to cause denial of service, such a case was witnessed with the Morris Worm in 1989 [10].

Probably the most common definition of a Distributed Denial of Service (DDoS) attack follows the idea of having multiple machines each deploying a DoS attack towards one or more targets [11]. Such a definition is almost correct, however, it fails to include the aspect of coordination between the attackers, which is the most fundamental characteristic of a DDoS attack. For that reason a new definition is formulated as.

"Distributed denial of service attack is a denial of service attack, in which a multitude of attackers performs denial of service attacks in a coordinated manner to one or more targets."

This definition emphasizes three important aspects. First, DDoS attack is essentially a denial of service attack. More accurately, DDoS attacks are a subset of Denial of Service attacks. Second, there must be more than one source attacking. Third, there must be coordination between the attackers. In case either one of these conditions is not met, the attack cannot be characterized as a DDoS attack.

It is important to note that, there exists the concept of denial of service, but there is no such thing as Distributed Denial of Service in the same sense. The service can be denied, but the service cannot be denied distributed unless the service itself is distributed. Only Distributed Denial of Service attack can exist.

The bandwidth attack is any activity that aims to disable the services provided by the target by sending an excessive volume of useless traffic [12]. This is in contrast to the flash crowd which occurs when a large number of legitimate users access a server at the same time. So the bandwidth attack is defined as.

"Bandwidth attack is an attack that consumes a target's resources through a massive volume of useless traffic."

DDoS attacks are always about multiple DoS attacks targeted to one or more specific target. As it was argued in the previously, coordination is a crucial part of DDoS Attack. Coordination of multiple hosts in turn implies the existence of some sort of a network structure, which could be titled as distributed denial of service attack network and define as.

"Distributed denial of service attack network is a network of computers that are being controlled by same entity administrating the distributed denial of service attacks."

3 Distributed Denial of Service Attack

In Distributed Denial of Service attack, the attack traffic is launched from the multiple distributed sources. The attack power of a DDoS attack is based on the massive number of attack sources. A typical DDoS is executed in two stages. First stage is to compromise vulnerable systems over the Internet this is known as turning these computers into zombies. In the second stage, the attacker sends an attack command to these zombies through a secure channel to launch an attack against the victim [13]. Spoofed source IP address are used to hide the identity of the zombies and a possible risk of being trace back to the attacker via zombies.

3.1 Attacks That Target Software

Distributed Denial of Service attacks that target software rely on the attacker's ability to perform a function or an operation against the target software, which either immediately or eventually causes denial of service situation. In other words, the aim of DDoS attack targeting the software is either system or software crash or system resource consumption. The targeted software can be anything ranging from operating systems to lightweight applications.

3.2 Attacks That Target Protocols

Distributed Denial of Service attacks that target protocols rely to the attacker's ability to exploit specifications of the protocols in a way that will result in denial of service. Differentiating protocol attack traffic from normal traffic is more difficult as compared to attacks that target software. The individual packets of the attack traffic stream may not contain any kind of signature diverging from normal packets. The traffic streams, however, may contain distinguishable patterns, such as abnormally high percentage of TCP SYN packets, which could be a sign of an ongoing TCP SYN attack.

3.3 Attacks That Target Bandwidth

Attacks that target bandwidth may appear as the easiest in nature, but in fact they are the most flexible and configurable DDoS attacks. These attacks aim to overwhelm the

target or the links on which the target's network relies, with such an amount of traffic that it causes either partial or complete denial of service. Hence it is not necessary for the attack traffic to reach the target. However, the attack traffic must be able to reach and congest the communication links. For instance, such links could be the routers of the target's Internet Service Provider that routes the target's traffic. Unlike other two classes of DDoS attacks, attacks that target bandwidth always succeeds, given that sufficient amount of attack traffic is able to reach the target.

Bandwidth attack sends an excessive volume of useless traffic. This is in contrast to the flash crowd where a large number of legitimate users access a server at the same time. The comparison between bandwidth attacks and flash crowds is shown in Table 1.

Table 1. Comparison between bandwidth attack and flash crowds [14]

	Bandwidth Attack	Flash Crowd
Network impact	Congested	Congested
Server impact	Overloaded	Overloaded
Traffic	Illegitimate	Genuine
Response to traffic	Unresponsive	Responsive
Traffic	Any	Mostly web
Number of flows	Any	Large number of flows
Predictability	Unpredictable	Mostly predictable

The bandwidth attack consumes the resources of a victim. Since the resources are limited (such as processing of NIC), high volume of traffic will result in dropping of incoming traffic by NIC. This traffic consist of both the legitimate traffic and attack traffic. As a result legitimate client will reduce their sending rate while the attackers will maintain or increase their sending rate. Eventually, the resources on the victim such as CPU and memory will be exhausted and the victim will not be able to provide the service. Bandwidth attack may also dominate the communication links of the network which is more threating then the resource consumption of victim. In this case the legitimate traffic to the server will be blocked, and if these links are the backbone links, any network or subnet which relies on these links will be effected.

4 UDP Flood Attack

The User Datagram Protocol (UDP) is a connectionless protocol that does not have flow control mechanisms, i.e., there is no built in mechanism for the sender and receiver to be synchronized or adapt to changing network conditions. The UDP flood is a type of bandwidth attack that uses UDP packets. Since UDP does not have flow control mechanisms, when traffic congestion happens, both legitimate and attack flows will not reduce their sending rates. Hence, the victim is unable to decide whether a source is an attack source or legitimate source by just checking the source's sending rate. Moreover, unlike TCP, UDP does not have a negotiation mechanism before setting up a connection. Therefore, it is easier to spoof UDP traffic without

being detected by the victim. Fig. 1(a) shows a typical UDP Flooding attack while Fig. 1 (b) gives an illustration of how a single spoofed UDP packet can initiate a never ending attack stream. The attacker sends a UDP packet to victim 1, claiming to be from victim 2, requesting the echo service. Since victim 1 does not know this is a spoofed packet, it echoes a UDP packet to victim 2 at port 7 (echo service). Then victim 2 does exactly the same as victim 1 and the loop of sending echo requests will never end unless it is stopped by an external entity [15].

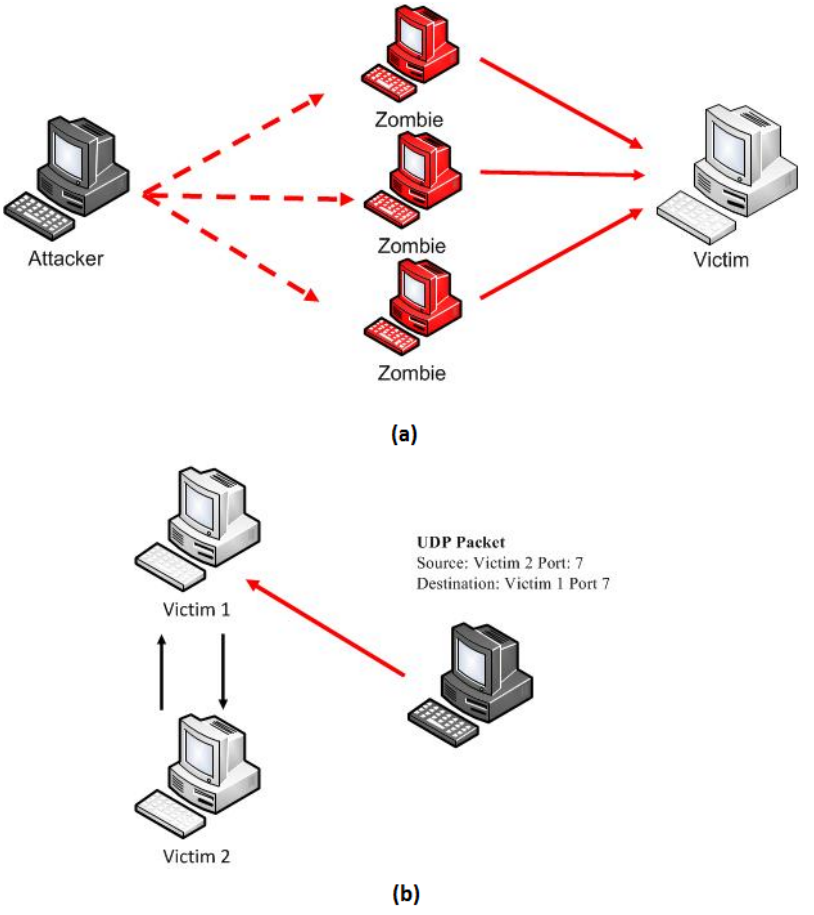

Fig. 1. UDP Attacks

5 Current Countermeasures

The DDoS attacks can be handled at three different levels. The majority of work has been done at the IP Layer. At this level, the defensive mechanism attempts to detect attack by analyzing certain features such as the IP filtering, IP logging, IP trace back, TTL values, Packet marking, Route statistics, IP header information and Flow monitoring

[16,17,18,19,20,21]. The other place to defend from such attacks is the TCP Layer. TCP, ICMP, UDP, SIN, FIN and similar packets are analyze or match against different rules to distinguish the attack traffic from normal traffic [22,23,24,25,26,] The last place to handle such attacks is the application layer. Defense mechanism at this level includes the monitoring of user browsing behavior for anomalies [27,28,], HTTP session analysis and limiting session rates [29], using statistical techniques for constraint random request attacks [30], usage of K-means clusters [31], using probabilistic techniques such as CAPTCHAs, graphic puzzles [32].

DDoS attacks are most commonly about consuming bandwidth and these attacks are the most difficult to defend against. As it was already mentioned, there are no absolute defense solutions to bandwidth consumption attacks, however, certain defense methods might be effective when they are properly implemented. Still, the technical defense methods are only a part of well-constructed risk management. Detection mechanisms refer to the actions performed to identify one or more ongoing attacks. Detection is the process of determining is the target under an attack, an attack must first be detected in order to level an appropriate defensive response.

MULTOPS [18] monitors the packet rate of uplink and downlink of a router. It works on the principle that under normal conditions, there is a proportional traffic transfer rate between two hosts. As a DDoS attack is initiated, a significant disproportional difference occurs in uplink and downlink traffic. Statistical approach for detection of SYN flood attack is proposed by Wang et al. [22]. Ratio of SYN packets to FIN and RST is used to detect such attacks. A similar approach by Blazek et al. [23] detects the DDoS attack using TCP and UDP traffic volume. Both methodologies used the assumption that during a DDoS attack, there will be a statistical change in traffic which can be used for detection of attack. Cheng et al. [21] used spectral analysis of packet arrival in a fixed interval as a sign of DDoS attack. During DDoS attack, large number of similar malicious packets is send from different sources to the victim. However, in case of legitimate traffic there will be many different traffic types. On the bases of this Kulkarni et al. [19] proposed a Kolmogorov detection system which uses the randomness and correlation in traffic flow to detect DDoS attack.

Cabrera et al. [25] used the correlation of traffic behavior at attack source as well as at victim. In the first step key variables from victim are extracted and analyzed in statistical tools to match variable of potential attacks in the second step. In the third step, normal profile is built which is further used with the variable to detect the potential attacks. Statistical approach used by Manikopoulos and Papavassiliou [20] is based on anomaly detection. First a normal profile is build using statistical modeling and neural network. Similarity distance is use for detection of attack. If the distance between monitor traffic and normal profile is greater than the threshold, it is assumed that a DDoS attack is in progress.

6 Proposed Solution (PSBTRA)

We proposed a solution for real time UDP flood attack detection. After evaluating the number of time in connectionless environment the proposed solution seems to be computationally fast and effective. When a server is under bandwidth attack, it cannot

reply to any requested service after maximum waiting time or due to unavailability of bandwidth. So it is assumed that a server under bandwidth attack will have higher incoming traffic and lower outgoing traffic and with a higher variation between them. It is further assumed that even if the server is still able to provide the requested service, the Quality of Service (QoS) will be degraded due to the limited bandwidth available to the legitimate users. Based on these assumptions we defined the traffic ratio *Traffic(T)* as:

$$\text{Traffic(T)} = \text{Traffic(IN)} / \text{Traffic(OUT)} \tag{1}$$

In eq (1), the *Traffic(IN)* is Number of incoming packets per second and *Traffic(OUT)* is Number of outgoing packets per second.

6.1 *Traffic(T)* Ratio

In order to confirm it, a study has been during which network traffic of a proxy server was captured through wireshark [33] and stored in libpcap format for further analysis. The capture traffic contains both the normal traffic and UDP flood attacks.

The analysis shows that during normal time period, the incoming and outgoing traffic is 62% and 38% respectively, while during the attack time period, it changed to 88% and 12% as shown in Fig. 2. The network traffic directed toward victim with a higher load of incoming over outgoing traffic is most likely intrusive and the value of *Traffic(T)* will be higher with a possibility of bandwidth attack. The traffic ratio *Traffic(T)* was also calculated for normal traffic eq (2) and for attack traffic eq (3).

$$Traffic(T_{Normal}) = 178953 / 108121 = 1.655118 \tag{2}$$

$$Traffic(T_{Attack}) = 768502 / 103093 = 7.454454 \tag{3}$$

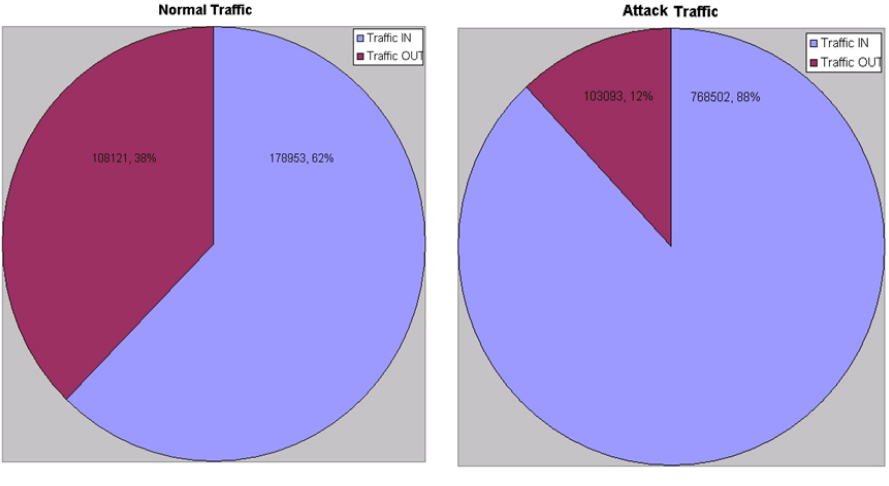

Fig. 2. Incoming and outing traffic under normal condition and during attack

It is clear that during the bandwidth attack the value of *Traffic(T)* is higher as compare to normal traffic. It was also reported by the users that they faced degradation in the Quality of Service (QoS) of Proxy Server during the attack time period, so this confirms the assumptions that during a bandwidth attack, system under bandwidth attack will have higher incoming traffic and low outgoing traffic and with a higher variation between them along with low QoS.Although, the *Traffic(T)* can detect bandwidth attacks, it has a short come. This ratio fails to distinguish between the bandwidth attack and flash crowd. The flash crowd which occurs when a large number of legitimate users access a server at the same time. A better approach is to use protocol composition with traffic ratio.

6.2 Protocol Proportion

When a specific attack is commenced, the proportion of the specific exploited protocol increases abruptly. Since the proportion of each protocol in traffic is related to each other, the increase in the proportion of exploit protocol makes the proportion of other protocols to decease. Therefore a detection technique can be design by monitoring the variation of the incoming and outgoing traffic by each protocol. As stated in Table 1 during flash crowd, the traffic is mostly web traffic. So a more better approach is to use *Traffic(T)* ratio along with the proportion of different protocols in network traffic. It was also observed that during the UDP flood attack, the proportion of UDP was considerably more than the other protocol.

Fig. 3 shows the proportion of incoming and outgoing TCP(96.47%), ICMP(0.71%) and UDP(3.82%) under normal traffic, while fig. 4 shows the proportion of incoming and outgoing TCP(4.43%), ICMP(10.12%) and UDP(85.36%) during attack.

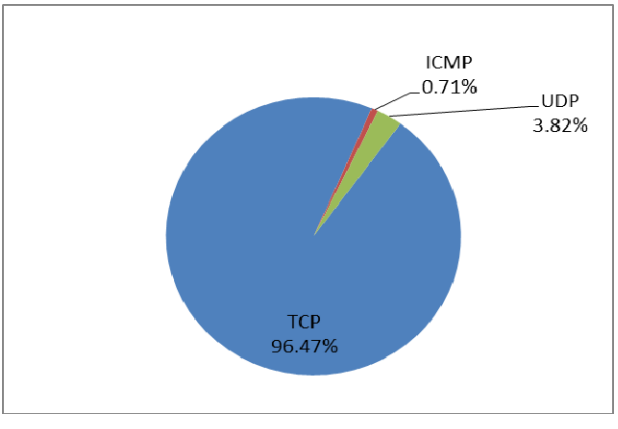

Fig. 3. Proportion of incoming and outgoing protocols in normal traffic

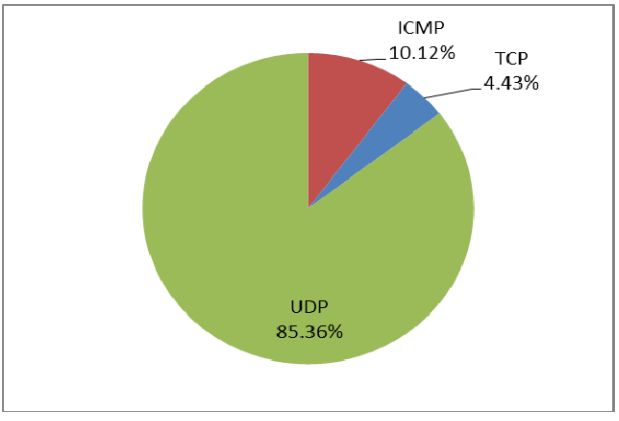

Fig. 4. Proportion of incoming and outgoing protocols during attack

6.3 Detection of UDP Flood Attack

The UDP Flood attack can be detected using the Equation (4).

$$IN_p(UDP)>(IN_p(ICMP)+ IN_p(TCP)) \text{ and}(OUT_p(ICMP)>OUT_p(TCP))\text{and } OUT_P(ICMP_{type})==3 \quad (4)$$

In Equation (4), the

IN_p *(IUDP)*	Proportion of incoming UDP traffic.
IN_p *(ICMP)*	Proportion of incoming ICMP traffic.
IN_p *(TCP)*	Proportion of incoming TCP traffic.
OUT_p *(ICMP)*	Proportion of outgoing ICMP traffic.
OUT_p *(TCP)*	Proportion of outgoing TCP traffic.
$OUT_P(ICMP_{type})$	Outgoing ICMP type 3(ICMP Destination Unreachable)

When an UDP packet is received, the PSBTRA becomes active. PSBTRA first compare the current proportion of incoming UDP traffic with current aggregated proportion of incoming ICMP and TCP traffic. Then it compares the current proportion of outgoing ICMP with the current proportion of outgoing TCP. PSBTRA also checks type of outgoing ICMP traffic. If the incoming proportion of UDP is greater than incoming ICMP and TCP and outgoing proportion of ICMP greater then out proportion of TCP and the ICMP type is equal to 3. The detection system alerts an UDP flooding attack. The reason to compare the current proportion of outing ICMP traffic with the current proportion of outgoing TCP and checking it type equal to 3 is that, when a system is under UDP flood attack, an outgoing ICMP traffic is generated. This outgoing traffic is ICMP Destination Unreachable (type code 3) messages informing the client that the destination cannot be reached.

7 Simulations and Results

In order to validate the PSBTRA, experiments were perform on real network having more than 500 clients. Since our research objective was to develop a real work solution, we didn't use any simulation tool or dataset like KDD[34] or DARPA[35]. During experiments an active proxy server was attack using Tribe Flood Network 2000(TFN2K) [36]. PSBTRA was implemented as Linux based application written in C on the proxy server during experiment to generate the traffic statistics and to detect UDP flood attack. Libpcap APIs [37] were use in the application for live packet capturing. The UDP flood attack using TFN2K was generated from 20 clients having 100 Mbps using packet starting with 20 packets per second and was gradually increased. We assumed these 20 clients as zombies. The source addresses of the UDP packets were spoofed. Fig. 5 shows the network layout of the experiment.

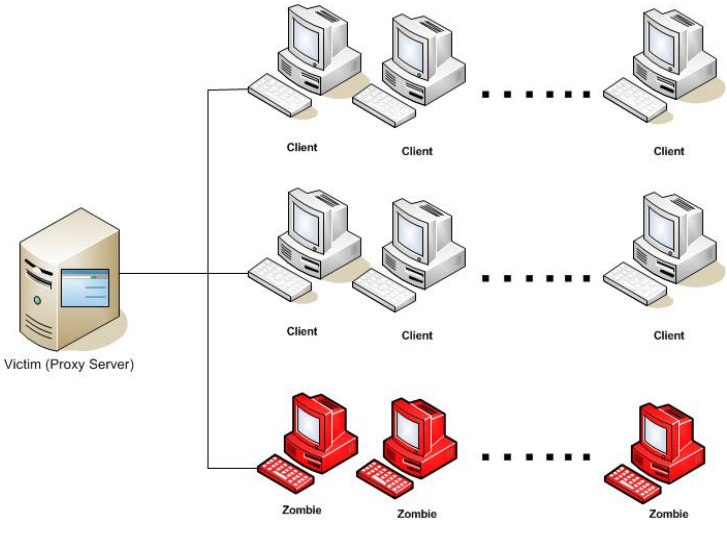

Fig. 5. Network layout of experiment

Fig. 6, 7 and 8 shows the incoming and outgoing TCP, UDP and ICMP traffic. Y axis shows the number of packets per second while X axis shows the time in seconds. A total of 19 minutes traffic was captured and processed. This traffic also includes the UDP flooding attack. As the attack occurs, the incoming UDP traffic rate is increased, at the same time the outgoing ICMP (Destination Unreachable) traffic rate also increases while both the incoming and outgoing TCP rates decreases. Fig. 9 shows the detection of attack by the PSBTRA. As the attack occurs, the PSBTRA immediately detects it and attack alert is generated. Once the attack is finished, the traffic rate of TCP, UDP and ICMP becomes normal.

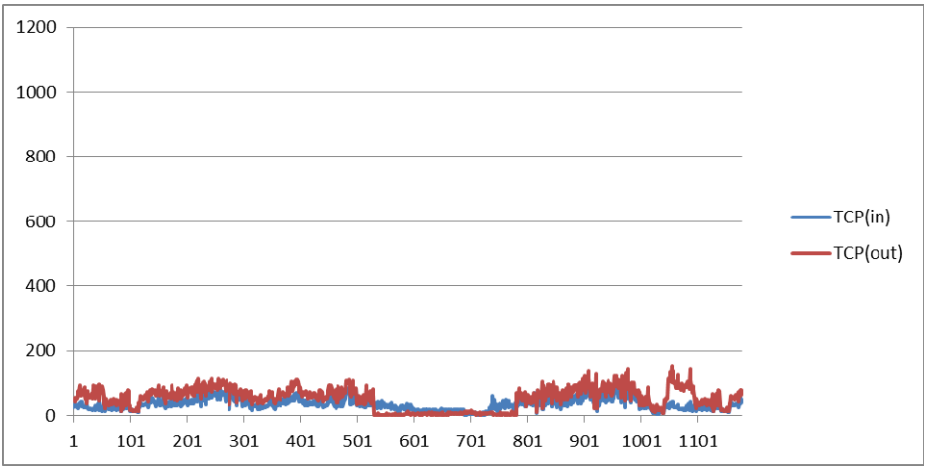

Fig. 6. Number of incoming and outgoing TCP packet. Y axis shows the number of packets per second while X axis shows the time in seconds.

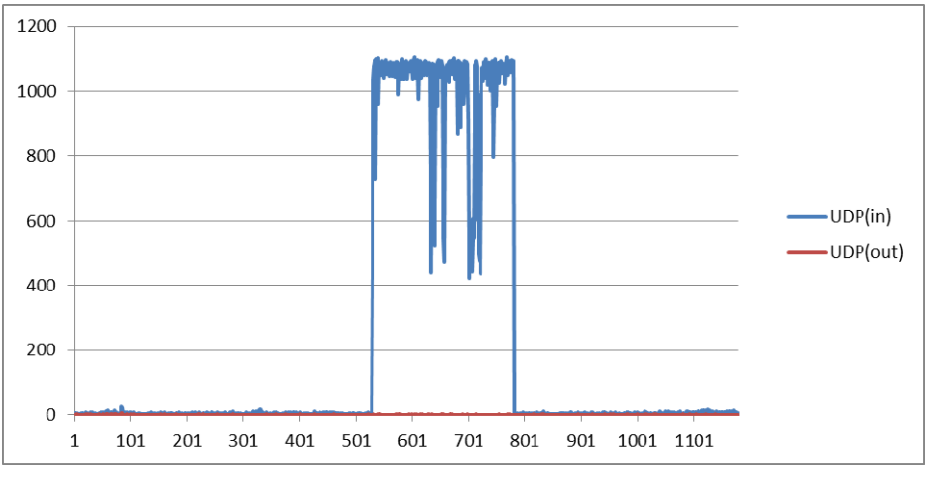

Fig. 7. Number of incoming and outgoing UDP packet. Y axis shows the number of packets per second while X axis shows the time in seconds.

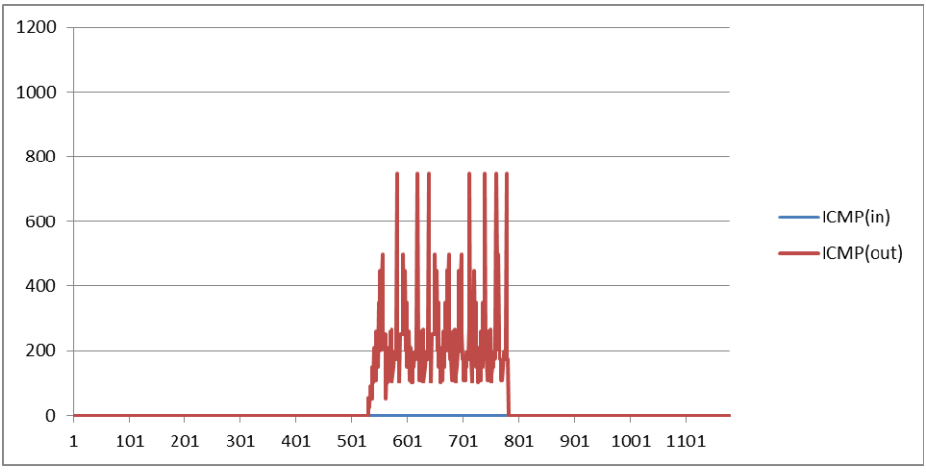

Fig. 8. Number of incoming and outgoing ICMP packet. Y axis shows the number of packets per second while X axis shows the time in seconds.

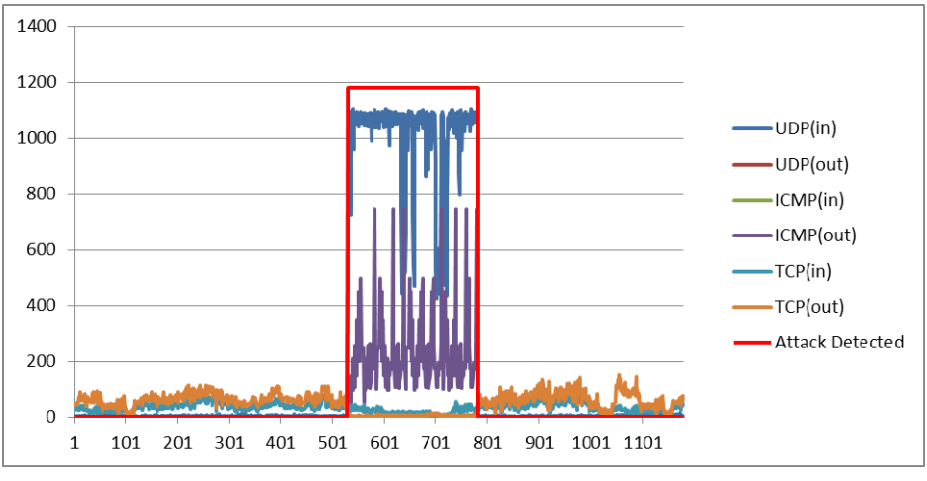

Fig. 9. Number of incoming and outgoing over all traffic. Y axis shows the number of packets per second while X axis shows the time in seconds.

8 Conclusion and Future Work

We have investigated the two main characteristics of bandwidth attack namely the traffic rate and protocol proportion. It is clear from the experiments perform on real network that these two characteristics can detect possible UDP based bandwidth attack. The methodologies presented in this paper has low computational overhead in detection of

UDP bandwidth attacks. The detection scheme based on the monitoring of incoming and outgoing traffic ratio along with the proportion of various protocols for the detection of UDP bandwidth attacks. We are also working to enhance the PSBTRA to detect TCP SYN, ICMP Smurf and other bandwidth attack specially those targeting multimedia traffic/service. The future work can be to add a defense mechanism can also be added to defend against such attacks by filtering the malicious traffic.

Acknowledgment. This research is supported by International Doctrinal Fellowship (IDF) No. UTM.J10.00/13.14/1/128(191) of Universiti Teknologi Malaysia and collaboration with Research Management Center (RMC) Universiti Teknologi Malaysia.

References

1. Lipson, H.F.: CERT CC: Tracking and tracing cyber-attacks: Technical challenges and global policy issues. Special Report CMU/SEI-2002-SR-009 (2002)
2. Blumenthal, M.S., Clark, D.D.: Rethinking the Design of the Internet: The End-to-End Argument vs. the Brave New World. ACM Transactions on Internet Technology 1, 70–109 (2001)
3. RFC 793 Transmission Control Protocol, http://www.faqs.org/rfcs/rfc793.html
4. Albert, R., Jeong, H., Barabási, A.: The Internet's Achilles' Heel: Error and attack tolerance of complex networks. Nature 406, 378–382 (2000)
5. Bellovin, S.M.: Security Problems in the TCP/IP Protocol Suite. ACM Computer Communications Review 19, 32–48 (1989)
6. CERT CC CERT Statistics, http://www.cert.org/stats/
7. RFC 791 Internet protocol, http://www.ietf.org/rfc/rfc0791.txt
8. Howard, J.D.: An Analysis of security incidents on the Internet 1989-1995. In: Ph. D dissertation. Carnegie Mellon University, Carnegie Institute of Technology (1998)
9. CERT CC Denial of Service Attacks, http://www.cert.org/tech_tips/denial_of_service.html
10. Orman, H., Streak, P.: The Morris Worm: A Fifteen-Year Perspective. IEEE Security & Privacy Magazine 1, 35–43 (2003)
11. Jelena Mirkovic, J., Peter Reiher, P.: A Taxonomy of DDoS Attacks and DDoS defense Mechanisms. ACM SIGCOMM Computer Communication Review 34, 39–53 (2004)
12. CERT CC CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks, http://www.cert.org/advisories/CA-1998-01.html
13. Dietrich, S., Long, N., Dittrich, D.: Analyzing distributed denial of service attack tools: The shaft case. In: Proceedings of the 14th USENIX Conference on System Administration, pp. 329–339 (2000)
14. Peng, T., Leckie, C., Ramamohanarao, K.: Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems. ACM Computing Surveys 39, 1–42 (2007)
15. CERT CC CERT Advisory CA-1996-01 UDP Port Denial-of-Service Attack, http://www.cert.org/advisories/CA-1996-01.html
16. El-Atawy, A., Al-Shaer, E., Tran, T., Boutaba, R.: Adaptive Early Packet Filtering for Defending Firewalls Against DoS Attacks. In: IEEE Conference on Computer Communications, pp. 2437–2445 (2009)
17. Wang, X., You-lin Xiao, Y.: IP Traceback Based on Deterministic Packet Marking and Logging. In: International Conference on Embedded Computing, pp. 178 –182 (2009)

18. Gil, T.M., Poletto, M.: MULTOPS, A data-structure for bandwidth attack detection. In: Proceedings of 10th Usenix Security Symposium, pp. 23–38 (2001)
19. Kulkarni, A.B., Bush, S., Evans, S.: Detecting distributed denial-of- service attacks using Kolmogorov complexity metrics. Technical Report 2001CRD176, GE Research & Development Center (2001)
20. Manikopoulos, C., Papavassiliou, S.: Network intrusion and fault detection: A statistical anomlay approach. IEEE Communications Magazine, 76–82 (2002)
21. Cheng, C.M., Kung, H.T., Tan, K.: Use of spectral analysis in defense against DoS attacks. In: IEEE Global Communications Conference, pp. 2143–2148 (2002)
22. Wang, H., Zhang, D., Shin, K.G.: Detecting SYN flooding attacks. In: IEEE Conference on Computer Communications, vol. 3, pp. 1530–1539 (2002)
23. Blazek, R.B., Kim, H., Rozovskii, B., Tartakovsky, A.: A novel approach to detection of denial-of-service attacks via adaptive sequential and batch-sequential change-point detection methods. In: IEEE Systems, Man and Cybernetics Information Assurance Workshop, vol. 54, pp. 3372–3382 (2006)
24. Limwiwatkul, L., Rungsawangr, A.: Distributed denial of service detection using TCP/IP header and traffic measurement analysis. In: International Symposium Communication Information Technology, pp. 605–610 (2004)
25. Cabrera, J.B.D., Lewis, L., Qin, X., Lee, W., Prasanth, R.K., Ravichandran, B., Mehra, R.K.: Proactive detection of distributed denial of service attacks using MIB traffic variables a feasibility study. In: IFIP/IEEE International Symposium on Integrated Network Management, pp. 609–622 (2001)
26. Noh, S., Lee, C., Choi, K., Jung, G.: Detecting Distributed Denial of Service (DDoS) Attacks Through Inductive Learning. In: Liu, J., Cheung, Y.-m., Yin, H. (eds.) IDEAL 2003. LNCS, vol. 2690, pp. 286–295. Springer, Heidelberg (2003)
27. Xie, Y., Yu, S.: A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviors. IEEE/ACM Transactions on Networking 17, 54–65 (2009)
28. Xie, Y., Yu, S.: Monitoring the Application-Layer DDoS Attacks for Popular Websites. IEEE/ACM Transactions on Networking 17, 15–25 (2009)
29. Ranjan, S., Swaminathan, R., Uysal, M., Nucci, A., Knightly, E.: DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks. IEEE/ACM Transactions on Networking 17, 26–39 (2009)
30. Yen, W., Lee, M.: Defending Application DDoS with Constraint Random Request Attacks. In: Asia-Pacific Conference on Communications, pp. 620–624 (2005)
31. Yu, J., Li, Z., Chen, H., Chen, X.: A Detection and Offense Mechanism to Defend Against Application Layer DDoS Attacks. In: Third International Conference on Networking and Services, pp. 54–54 (2007)
32. Ahn, V., Blum, M., Langford, J.: Telling Humans and Computers Apart Automatically. Communications of the ACM 47, 57–60 (2004)
33. Wireshark,· Go deep, http://www.wireshark.org/
34. KDD Cup 1999 Data (1999), http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
35. DARPA Intrusion Detection Evaluation, http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/index.html
36. TFN2K – An Analysis, http://packetstorm.wowhacker.com/distributed/TFN2k_Analysis.htm
37. TCPDUMP/LIBPCAP public repositor, http://www.tcpdump.org