區域網路超量攻擊訊務的監測
Monitoring X-Attack Traffic over Aggregate Network

Su-Chiu Yang
Computer Center of National Central University
center7@cc.ncu.edu.tw

Li-Ming Tseng
Department of Information Engineering of National Central University
tsenglm@cc.ncu.edu.tw

摘要

隨著電腦計算效能的快速提升, 愈來愈多的攻擊病毒利用開放的傳輸協定,發動超量 X -Attack **攻擊: 產出鉅量的** UDP/ICMP Flooding **封包虛耗網路資源, 壅塞** WAN **傳輸.為防止超量攻擊訊務的持續擴散,影響沿徑** routing **網段訊務,本研究擷取區域網路中心** router **的訊務轉送** log, **統計** host-to-host **的非自律性** Packet/ Byte/ Flow **訊務量,實做超量攻擊訊務監測網頁, 與自動化的攻擊訊務阻絕與通告系統,並統計單日的** UDP/ICMP Packet/Byte **標準差分布,提供方便的攻擊訊務監測指標,分析攻擊訊務的阻絕成效。**

Abstract

The notable rise in significant UDP/ICMP flooding events and network worms has increased the need to design effective methods for detecting significant attack traffic and preventing further traffic degradation. This work developed web pages allowing users to monitor abnormal UDP/ICMP flooding attack traffic based on the Netflow transportation traffic logs gathered from the aggregate router. This system has been deployed in one regional network center over a TANet backbone. And the automatic X-Attack traffic detecting and limitation system also was implemented based on known extraordinary attack behaviors.

Keywords：X-Attack traffic measurement, ICMP/UDP Flooding, blocking attack traffic.

1. Introduction

A. Motivation

The convenience of the Internet has driven its acceptance as the main means of data communication, and also has brought various benefits. However, the open transmission protocols also have provided an excellent opportunity for numerous attacking programs and network worms to flourish. The Internet has experienced a rapid increase in the frequency of attack events from network worms or viruses, for example, the CodeRed/ Nimda worm in 2001, the Scalper/Slapper worm found in 2002, and the Slammer and the Blaster worms in 2003 [1]-[4]. The flourishing worms have arisen as a result of the openness of transportation protocols and the shortage of effective and wide-deployed attack traffic measurement tools. The marked rise in attack traffic has increased the need to measure this attack traffic and prevent significant traffic degradation.

Along with the advanced computing and broadband networking resources, attackers can markedly increase the volume of attack traffic using the attacking source codes retrieved from worldwide Internet. For example, attackers can increase traffic volume by maximizing the iteration count and packet size parameters of the attack program to launch UDP/ICMP or SYN flooding. Alternatively, attackers can use a fake source IP address or IP protocol identifier carried in the packet header to launch a Smurf attack without being filtered out by firewalls or network operators. and the extraordinarily huge chunks of useless

packets can severely congest regional networks and jam the network links throughout the transmission path, and strongly affecting inter-networking performance.

Most network worms have carried a payload causing a Denial of Service (DoS) attack on well-known services, and providing the attacker with full remote access to the servers. Consequently, it is extremely plausible that an attacker could infect dozens or even hundreds of servers, and moreover could trigger those compromised hosts to launch a distributed flooding attack and overwhelm the transport routing resources. This kind of attack is called an eXtreme Attack (X-Attack).

The lack of attack traffic measurement and detect tools also helps the rapid spreading of network viruses and the rapid growing of attack events. Internet communication heavily depends on the transmission function on all the transit routing sub-networks throughout the transportation path. The attack could not success if any transit node can detect and block attack traffic promptly. This work thus developed a feasible approach for UDP/ICMP flooding traffic measurement to help detect the compromised machines and block the significant attack traffic automatically. Network users are also encouraged to explore and browse the concrete UDP/ICMP X-Attack traffic via web interfaces, to determine the X-Attack traffic and the compromised hosts, and fix the systems accordingly.

B. The Transportation Traffic Logs

Since all network operators depend on quantifiable traffic log data to evaluate network performance, Traffic measurement has been considered necessary since the early days of networking. WAN routers stand at the entrance of the aggregated networks, and respond to the forwarding IP packets. Consequently, it is feasible to configure a router to cache and total the transit packet headers, including detailed transportation items such as source/destination IP addresses, source/destination transportation application ports, source/destination routing interfaces, protocol identity, number of packets, and number of bytes. Additionally, detailed NetFlow entries are forwarded regularly to a designated collecting and analyzing PC [5]-[7]. And network operators can develop numerous traffic measurements for aggregate networks using the concrete single-direction flow logs.

With some knowledge of the IP stack, network operators can browse and trace the traffic characteristics of different applications or IP hosts using the transportation flow logs collected from aggregate router. For example, operators can measure the top-N input or output traffic by accumulating the byte counts of each flow log with the index of the source or destination IP addresses. Alternatively, operators can monitor the traffic of top-N applications by summing up the byte counts of each flow log with the index of transportation port number. While this work measures the top-N ICMP/UDP communication partners by accumulating the flow count, packet count and byte count with the index of the source and destination IP addresses.

The rest of this paper is organized as follows. Section 2 describes the characteristics of network attack and the measurement of abnormal ICMP/UDP flooding traffic, and also analyzes the X-Attack traffic over the subject network. Section 3 addresses the implementation of the automatic X-attack traffic blocking system, and analyzes the system performance. Finally, Section 4 draws conclusions.

2 Measuring and Monitoring X-Attack Traffic

A. Monitoring ICMP X-Attack Traffic

For avoiding block by firewalls or network operators, the attack programs employ dynamic application ports to spread the massive UDP/ICMP packets to the single victim or a group of destination hosts. It is also highly plausible for the source IP hosts infected by an attacker to generate massive ICMP packets to strike the victim off the Internet and congest inter-networking. Consequently, it is impossible to identify the attack traffic flow logs using fixed transport ports only. As the number of packets generated by the X-Attack machine can be significantly exceed that generated by ordinary Internet applications. This work measures the abnormal UDP/ICMP flooding traffic based on the significantly intensive traffic volume, the character traits of all the extreme attack programs.

The approach for measuring abnormal flooding traffic could be straightforwardly and easily. First, the numbers of packets and bytes that transmitted between each source and destination IP pair were accumulated, as were the numbers of flow connections established between the communication partners. And these data was saved into the corresponding traffic lists, icmp_flow[$pair_i$], icmp_packet[$pair_i$] and icmp_byte[$pair_i$]. After sorting these traffic lists, the abnormal traffic volume could be filtered from the ICMP traffic lists by applying some high thresholds, for example, icmp_flow[$pair_i$] / hour > 5000, and icmp_packet[$pair_i$] / hour > 100,000. When the top-N traffic result was filtered out, a Hypertext Preprocessor (PHP) scripts were written to accept user queries and present the corresponding traffic results in response to these queries [8]-[9].

The obvious huge chunk of X-attack host could be easily detected according to the measured numeric result shown on the web page. Figure 1(a) displays the top-N ICMP traffic associations measured over the subject network on May 5th 2003. Clearly, the number of ICMP packets transmitted from hosts 140.135.135.104, 140.115.220. 87 and 203.68.79.1 significantly exceeded 10^7 packets per hour, and the total traffic volume sent out from attacker could also reach up to several Giga-bytes per hour.

However, the attacker used the forged IP protocol identifier 255 rather than the ICMP protocol identifier of 1, to prevent the traffic being blocked by network routers or firewalls [10]. And the significant huge ICMP flooding traffic generated by the X-Attack machines was all single direction, and all targeting the same destination (Fig 1a). The cause might be that the source IP hosts was compromised by DRDoS worms [11], and be tricked to reflect massive packets to the destination victim. It is also highly plausible that the extreme flooding traffic had been blocked by some transit segment along the transit path. Anyway, the X-Attack traffic was sufficiently large to exhaust the processing resources of several routing sub-networks and congest the regional network severely.

B. Monitoring Blaster ICMP Flooding Traffic

The W32.Welchia, also known as WORM_MSBLAST.D, is a worm that exploits the DCOM RPC vulnerability using TCP port 135. It checks for active machines to infect by sending an ICMP echo request, or PING, which will result in increased ICMP traffic. It selects the victim IP address in two different ways: The worm uses either A.B.0.0 from the infected machine's IP of A.B.C.D, or it will construct a random IP address based on some hard coded addresses. After selecting the start address, the worm counts up through a range of Class B-sized networks; for example, if the worm starts to send an ICMP echo request to A.B.0.0, it will count up to at least A.B.255.255, to check whether the constructed IP address is an active machine on the network [12].

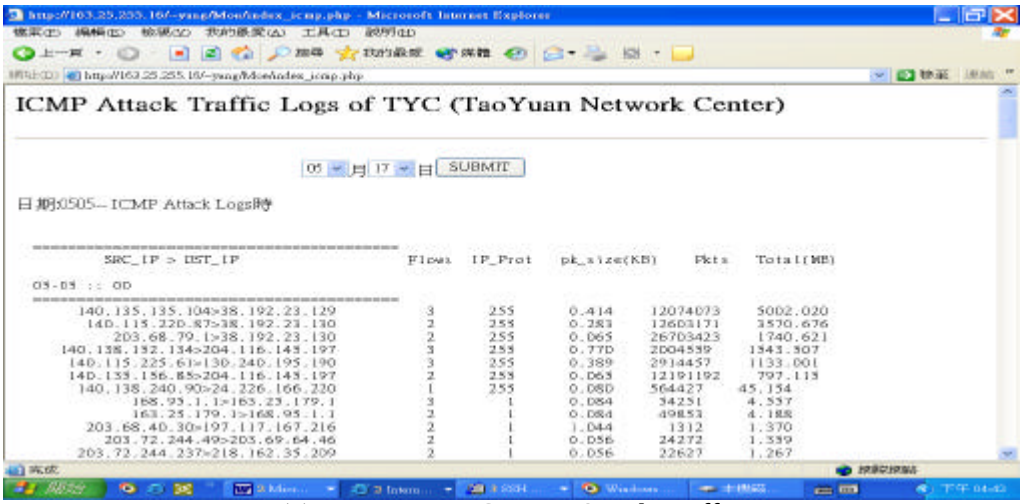
Similarly, the numbers of flows, packets and bytes that transmitted between the modified communication pairs: source IP address and subset of destination IP (S1.S2.S3.S4>D1.D2.#.#) were accumulated, and saved into the corresponding traffic lists, blast_flow[pair$_i$], blast_packet[pair$_i$] and blast_byte[pair$_i$]. After sorting these traffic lists, the abnormal traffic volume could be filtered from the Top-N Blaster communicating lists. Subsequently, the daily top-N Blaster ICMP flooding traffic can be displayed on web page, and the owner of the compromised machine can be noticed accordingly.

Figure 1(b) displays the top-N Blaster flooding traffic send out from the infected machines over the subject networks on Sep 4th 2003. Although the number of ICMP packets transmitted from single compromised hosts was largely lower than X-Attack machine. Network users are encouraged to explore and browse concrete Blaster flooding traffic web page, to determine the IP addresses of compromised victim hosts, and thus fix the infected systems.

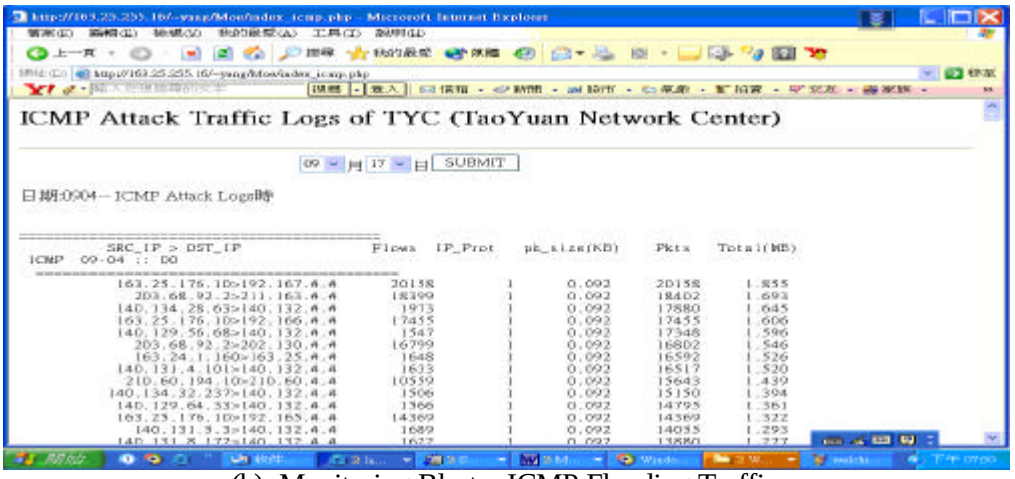
C. Monitoring UDP X-Attack Traffic

Applying the traffic accumulating steps for detecting ICMP X-Attack, the top-N UDP traffic list also could be figured out accordingly. Figure 2 displays the UDP traffic measured over the subject network on Feb 15th 2003. Clearly, the obvious large numbers of UDP packets were sent from IP hosts 140.123.102.184 and 140.136.200.11. The X-Attack packets transmitted from each single attack host can reach up to 10^7 - 10^8 packets per hour, significantly exceeding the number generated by streaming media and game servers, 218.146.254.203 and 163.13.10.141 (Fig. 2). The massive competing traffic can overwhelm the processing resources of routing interfaces and jam all of the routing networks in the transmission path.

Besides the attack traffic, the heavy streaming media and game traffic also can be detected and listed on the web pages. For example, the prevalent streaming media traffic, transmitted between 163.13.10.141 and 61.171.38.242, and had mean packet size of approximately 1500 bytes/packet [13]-[17]. Additionally, the traffic transmitted between the Counter_Strike game servers, 218.146.254.203 and 64.95.80.9, had mean packet size of 70 to 200 bytes/packet; and the TFtp traffic, the mean packet size of 544 bytes/packet, transmitted between 203.242.146.143 and 203.72.179.12 also can be detected and displayed on page.



(a) Monitoring ICMP X-Attack Traffic



(b) Monitoring Blaster ICMP Flooding Traffic

Fig. 1 Monitoring Extraordinary ICMP Attack Traffic

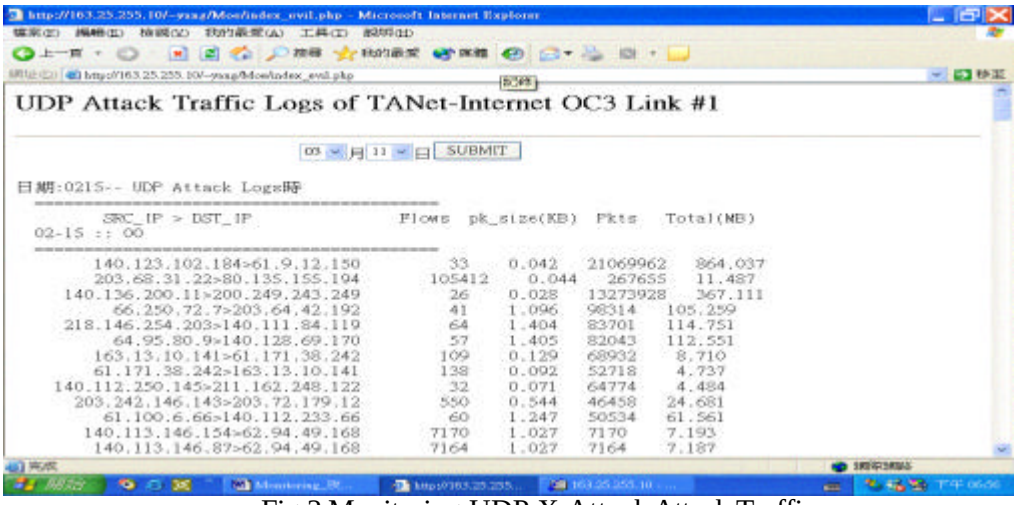


Fig.2 Monitoring UDP X-Attack Attack Traffic

3 Automatic X-Attacking Traffic Blocking System

A. Monitoring Standard Deviation of UDP/ICMP Traffic

The traffic associated with an attack significantly exceeded that generated by normal applications. Moreover, the X-Attack event could clearly be identified based on the standard deviation (std.) of measured traffic. Consequently, the std. of UDP traffic was accumulated following the statistical formulae 1 though 4 to help monitor the evident X-Attack events.

Figure 3(a) illustrates the std. statistics of the UDP traffic measured over the regional network used here (May-29th -2003). Obviously, the UDP X-Attack traffic began from 0:00, and it lasted until the attack tarffic was blocked by system manager at 12:00 by configuring the aggregate router to limit the traffic of the detected attack machines. The standard numerical of UDP packets and bytes evidently reflected the attack traffic (Fig.3a & 3b).

B. Blocking the X-Attack Traffic

Obviously, the massive quantities of packets transmitted between the single X-Attack flow-pair can reach 10^7 - 10^8 packets per hour, and that significantly exceeded those generated from the normal applications. Consequently, the automatic

X-Attack traffic blocking was implemented to maintain high transmission quality.

First, the source IP record of each significant X-Attack flow were filtered out based on the top-N UDP traffic list measured over the present network. Additionally, the aggregate router could be remotely configured to block obvious attack traffic. And the email addresses of the source hosts also can be identified via RWhois query; so the emails can be sent to the owners of attack machines to notify the attack traffic of the compromised hosts.

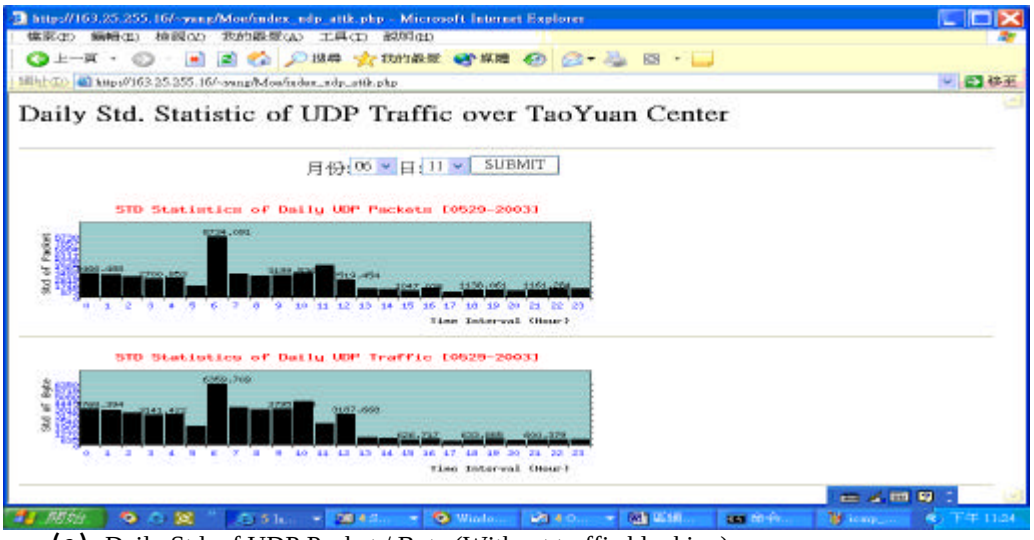
Figure 3(c) displays the std. statistics of the UDP traffic measured over the aggregate network that has been protected against overwhelming by continuous X-Attack traffic (Jun-16th-2003). Obviously, the X-Attack traffic was successfully eliminated within the detecting and blocking hours. The std. UDP packet statistics also clearly reflected the huge packet amounts to the attack traffic (Fig. 3c). Figure 4 displays the automatically blocked X-Attack ICMP/UDP traffic from April 2003 through September 2003. Obviously, the X-Attack traffic had been effectively limited from July 2003.

$$mean\_udp\_pkt = \frac{\sum_{i=1}^{n} \text{udp\_pkt}[pair_i]}{n} \quad , i = 0, 1, 2, \ldots, n \quad \text{---- (1)}$$

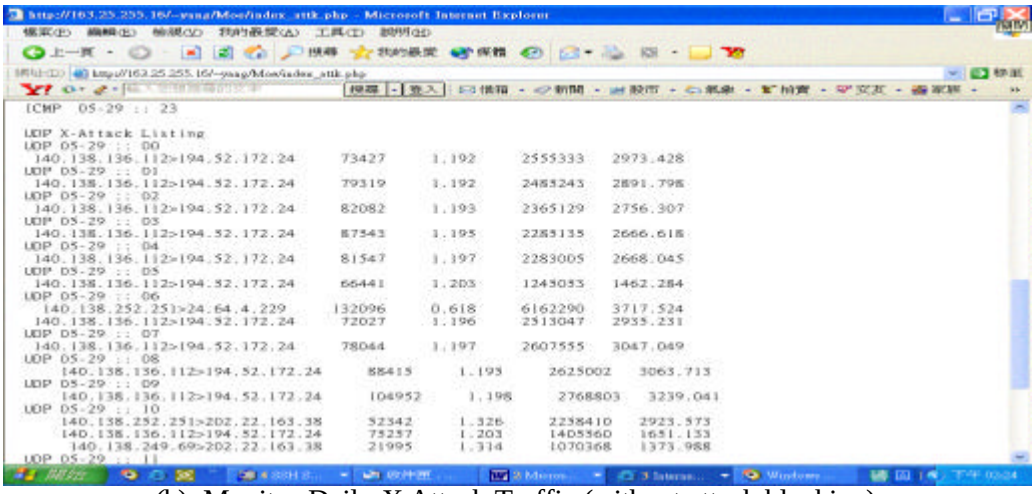
$$(std\_udp\_pkt)_i = \sqrt{\frac{\sum_{i}^{n-1} (\text{udp\_pkt}[pair_i] - mean\_udp\_pkt)^2}{n-1}} \quad \text{---- (2)}$$

$$mean\_udp\_byte = \frac{\sum_{i=1}^{n} \text{udp\_byte}[pair_i]}{n} \quad , i = 0, 1, 2, \ldots, n \quad \text{---- (3)}$$

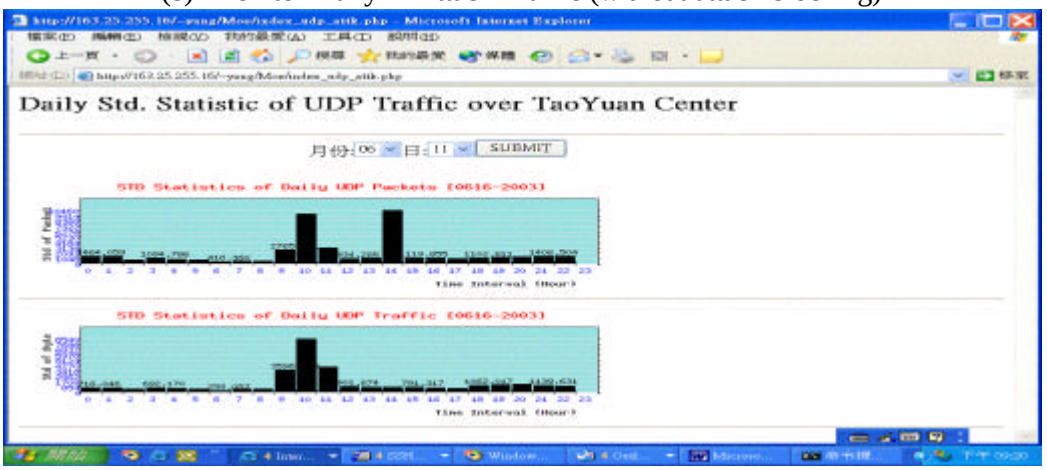
$$(std\_udp\_byte)_i = \sqrt{\frac{\sum_{i}^{n-1} (\text{udp\_byte}[\text{pair}_i]_i - mean\_udp\_byte)^2}{n-1}} \quad \text{---- (4)}$$



(a) Daily Std. of UDP Packet / Byte (Without traffic blocking)



(b) Monitor Daily X-Attack Traffic (without attack blocking)



(c) STD. of UDP Packet/Byte statistics (with traffic blocking)

Fig.3 Standard Deviation of ICMP/UDP Traffic

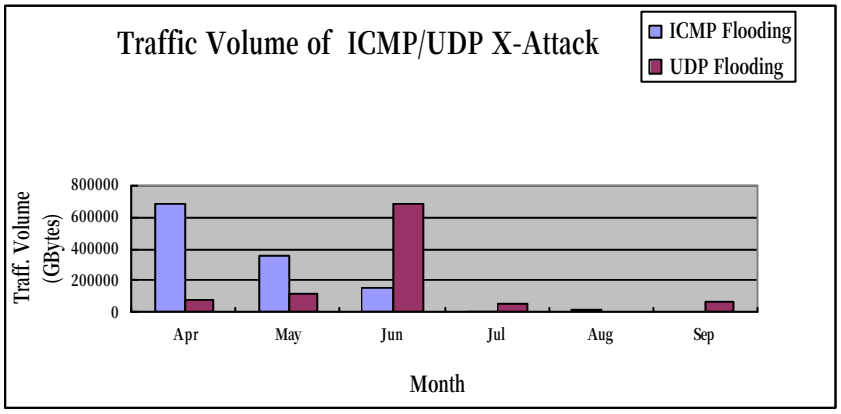


Fig. 4 Traffic Volume of X-Attack ICMP/UDP Flooding

4. Conclusion

This work created web pages to monitor the UDP/ICMP X-Attack traffic and the top-N UDP application traffic using the Netflow transportation traffic logs gathered from the aggregate router. The traffic measuring system was installed on the aggregate network of the Tao-Yuan area center over the TANet backbone. The following observations were made based on the measured results for the last three months.

The massive X-Attack ICMP/UDP packets sent out from a single attack host significantly exceeded those emitted from ordinal media and game servers, and could congest the network links along the transmission path. According feedback from the user of detected machines, most of the attack machines were compromised Windows 2000 systems, and very few attack machines were infected UNIX hosts of Linux and FreeBSD systems.

Since Network worms spread attack traffic locally rather than globally. This study recommends that attack traffic monitoring systems should also be implemented on more campus networks to measure the abnormal attack traffic, and thus help identify and fix the compromised systems. The continuous advance of computing and networking technologies undoubtedly will increase more varieties of network worms and viruses. In the near future, the authors plan to develop the stochastic modeling of some basic TCP service traffic based on detailed knowledge of attack behaviors.

Acknowledgement

The authors would like to thank the Ministry of Education of the Republic of China for financially supporting this research under Contract No. MOE-910-78067.

Reference

(1) Bellovin S.M., Security Problems in TCP/IP Protocol Suit, Computer Communication Review, vol. 19, No. 2,pp. 32-48. April 1989.

(2) Adams E.S., Distributed Reflective Denial of Service Attacks, http://www.unixeunuchs.com/joke.php.

(3) Strategies to protect Against Distributed Denial of Service (DDoS) Attacks, http://www.cisco.com/warp/public/707/newsflash.html

(4) W32.Blaster.Worm,http://securityresponse.symantec.com/avcenter/venc/data/w32.welchia.worm.html

(5) Cho K.; Kaizaki R.; Kato A., An Aggregation Technique for Traffic Monitoring, Proceedings of the 2002 Symposium on Applications and the Internet (SAINT'02w), 2002.

(6) Cisco IOS Netflow Technology, http://www.cisco.com/warp/public/cc/pd/iosw/ prodlit/iosnf_ds.htm.

(7) Flow-tools: Tool set for working with NetFlow data, http://www.splintered.net/sw/flow-tools.

(8) Apache https server project, http://httpd.apache.org/ABOUT_APACHE.html

(9) PHP: Hypertext Preprocessor, http://www.php.net/manual/en/introduction.php

(10) Strategies to protect Against Distributed Denial of Service (DDoS) Attacks, http://www.cisco.com/warp/public/707/newsflash.html.

(11) Adams E.S., Distributed Reflective Denial of Service Attacks, http://www.unixeunuchs.com/joke.php ?request=31.

(12) W32.Welchia.Worm, http://securityresponse.symantec.com/avcenter/venc/data/w32.welchia.worm.html.

(13) Schulzrinne H. et al., RTP: A Transport Protocol for Real-Time Applications, RFC 1889, April 1996.

(14) Schulzrinne H. et al., Real Time Streaming Protocol (RTSP), RFC 2326, April 1998.

(15) Mena A.; Heidemann J., An Empirical Study of Real Audio Traffic", IEEE INFOCOM 2000, Pages: 101-110.

(16) Schulzrinne H. et al., RTP: A Transport Protocol for Real-Time Applications, RFC 1889, April 1996.

(17) Conklin, G.. J.; Greenbaum, G.S.; Lillevold, K.O.; Lippman, A.F.; Reznik, Y.A., Video coding for streaming media delivery on the Internet, IEEE Transactions on Circuits and Systems for Video Technology, Volume: 11 Issue: 3, March 2001, Pages: 269 –281.