Is It Congestion or a DDoS Attack?

Amey Shevtekar and Nirwan Ansari, *Fellow, IEEE*

***Abstract*—We propose a new stealthy DDoS attack model referred to as the "quiet" attack. The attack traffic consists of TCP traffic only. Widely used botnets in today's various attacks and newly introduced network feedback control are integral part of the quiet attack model. We show that short-lived TCP flows can be intentionally misused. The quiet attack is detrimental to the Internet traffic and at the same time difficult to be detected by using current defense systems. We demonstrate the inability of representative defense schemes such as adaptive queue management and aggregate congestion control to detect the quiet attack.**

***Index Terms*—DDoS, router, and TCP.**

I. Introduction

INTERNET has become an integral part of various commercial activities like online banking, online shopping, etc. Denial-of-Service (*DoS*) attacks are becoming a major threat to the Internet infrastructure integrity. These days, attackers are professionals who are involved in such activities because of financial incentives. A low rate DoS attack is such a new threat to the Internet [1]. Instances of this type of attacks on Internet2 experimental network have been reported. Thus, it is becoming more difficult to detect stealthy DoS attacks. We believe that the war against attackers will be won only if we stay ahead of the attackers. In this paper, we expose a similar vulnerability in the current Internet infrastructure. Our model uses the TCP protocol to launch the attack. TCP is considered to be a network adaptive protocol and widely contributes to the current Internet traffic. Our model has the ability to disguise itself as a normal traffic, thereby making detection difficult. It relies on using readily available botnets to send the attack traffic. Section II presents the concept of the quiet attack model. Section III describes how to execute the quiet attack in the Internet. Section IV presents simulation results. Section V compares the quiet attack with other attacks, followed by conclusions and future work in Section VI.

II. Quiet Attack Basics

In this section, we describe the basic attack model, and explain the effects of the attack on a single TCP flow. The attack exploits the effect of short-lived TCP flows on other TCP flows. A TCP flow strictly adheres to the congestion control algorithms like slow start and congestion avoidance. It is a network adaptive protocol. A long-lived TCP flow is typically known to enter and end in the congestion avoidance phase while a short-lived TCP flow to end in the slow start phase. In the quiet attack model, a set of short-lived TCP flows are started so that the aggregate attack traffic rate is approximately proportional to the link capacity under attack, and subsequently after every short time period T, a new set of short-lived TCP flows are injected to the network. A TCP flow rate is congestion dependent unlike a UDP flow, and so we may not impose the exact rate by using TCP as an attack flow. A botmaster will use capacity estimation techniques like capprobe [2] to approximate the link capacity. During this process, the available bandwidth is monitored by using tools like abget [3] or pathchirp [4], and is not allowed to increase. The time period T is kept random between 0s and 1s to remove any deterministic pattern. In fact, the time period is designed to look continuous. The available bandwidth remains negligible that leads to congestion at the router queue, and as the queue fills up, packets at the queue end are dropped by the droptail queue mechanism, i.e., any packet can be dropped as there is no preferential treatment given to any packet. As a legitimate long-lived TCP flow times out and enters the slow start phase, the congestion window is reduced, thus reducing the sending rate and throughput. The persistent congestion in the queue forces random packet drops of the legitimate long-lived TCP flows, thus affecting the throughput each time a packet is dropped. The attack concept is simple: keep the available bandwidth negligible by sending enough short-lived TCP flows.

The quiet attack exploits shortcomings of the end to end window flow control [5], which shows that if the number of connections increases, then delay or congestion will roughly increase proportionally to the number of connections or more precisely to the sum of window sizes of all connections. Basic trade off in end to end window flow control mechanism is selecting the right window size which is a well studied difficult problem. TCP initially does not know the network congestion state, and so it uses slow start in the beginning when the window size is increased exponentially; this, however, gives unfair advantage to short-lived TCP flows in the quiet attack when other legitimate TCP flows are waiting in retransmission timeout. In this attack, short-lived flows are sent persistently that subsequently lead to considerable reduction in throughput.

The major threat of the attack model is to lead us into assuming the attack as a "congestion scenario". Previous studies have considered effects of short-lived flows on other Internet traffic with the assumption that they occupy less volume of the total capacity although they are large in numbers. By using botnets in sending short-lived TCP flows during an attack, such Internet traffic assumptions can no longer be true. This letter is the first to pinpoint such a disguisable attack scenario and to demonstrate the potential threat.

III. Quiet Attack Execution in the Internet

A. *Quiet Attack Reconnaissance Phase*

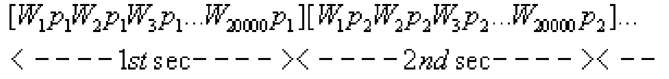
The attack execution of the quiet attack model involves network reconnaissance and finally execution. The network

Manuscript received March 15, 2009. The associate editor coordinating the review of this letter and approving it for publication was X. Cao.

The authors are with the Advanced Networking Laboratory, ECE Dept., NJIT, Newark, NJ 07012, USA (e-mail: ansari@njit.edu).

Digital Object Identifier 10.1109/LCOMM.2009.090628

1089-7798/09$25.00 © 2009 IEEE



$[W_1p_1W_2p_1W_3p_1 \cdots W_{20000}p_1][W_1p_2W_2p_2W_3p_2 \cdots W_{20000}p_2] \cdots$

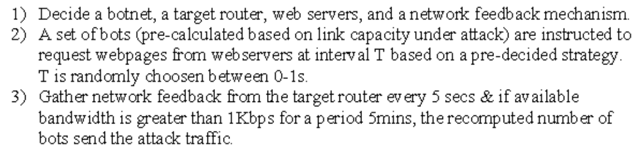
$< - - - - 1st \text{ sec} - - - >< - - - - 2nd \text{ sec} - - - >< - -$

Fig. 1. Web request strategy.

reconnaissance involves deciding a router to be attacked, a botnet to be used, and gathering a list of web servers. We assume an attacker has access to a botnet which is formed by capitalizing on latest exploits. Active bots in a botnet are typically concentrated in the same time zone [6], and such group of bots will participate in the quiet attack. The next step is to determine which router to launch the attack that involves the botmaster issuing a command to the active bots to initiate a simple command like "tracert http://www.bu.edu" which will trace the path from the bot to the bu website. A botmaster selects a website which does not use CDN (content distribution network) for reasons to be explained later. The tracert command even works for users with limited access or non-administrative accounts on windows XP and Vista. All bots then send the output of the tracert to the botmaster. An edge router whose IP address is observed in all the tracert ouputs becomes a target for attack by the botmaster. We assume each time zone will have few big ISPs; for example, there are Verizon, Comcast and AT&T in the eastern standard time zone. The big ISPs will have few transit points from which the traffic enters the Internet backbone. Such transit points of big ISPs will be the targets of the quiet attack. Espionage during a cyber war or stealing confidential information of ISPs can be another way in which an IP address of a router to be attacked can be found. To discover routers to be attacked, an attacker can also use a network mapping technique by using open source tools like cartoreso [7] or traceroute. The last step is collecting web servers that could be attacked. The major component of the attack traffic is short-lived flows, and so an attacker needs web servers from which bots could request files via HTTP. We assume that bots can exploit a CAPTCHA or use websites that do not employ CAPTCHA [8]. Simple scripts [3] can estimate the size of web pages. A botmaster can set a limit of the web page size between 100KB and 1MB for a bot to request. A larger number of web servers will allow a sparser distribution of the attack traffic. This would make attack traffic realistic and evade detection from systems which target sudden increase in the traffic to a destination address. An attacker should avoid websites which use CDNs as it could lead to packets going to proxy web servers at the edges of ISPs. The largest CDN provider akamai claims to carry 20% of web traffic, i.e., 80% of web traffic does not pass through akamai. Most of the government websites, university websites, and small websites cannot afford CDNs.

B. Quiet Attack Execution Phase

Under normal circumstances, most of the DSL download rates are 1-2 Mbps for a typical home customer. Now, consider an OC192 (10Gbps) link under the attack. To fill up 10Gbps, we need 20,000 bots online, assuming each sending at a minimal rate of 500Kbps. In the Internet, there are easily around millions of websites, and so we may consider a botmaster using 20,000 websites in the quiet attack. The webpages (p) of websites (W) used in the attack are accessed by 20,000 bots as shown in Fig. 1.



1) Decide a botnet, a target router, web servers, and a network feedback mechanism.
2) A set of bots (pre-calculated based on link capacity under attack) are instructed to request webpages from webservers at interval T based on a pre-decided strategy. T is randomly choosen between 0-1s.
3) Gather network feedback from the target router every 5 secs & if available bandwidth is greater than 1Kbps for a period 5mins, the recomputed number of bots send the attack traffic.

Fig. 2. Quiet DDoS Attack Algorithm.

Thus, every second each website only experiences one webpage worth of traffic which is clearly non-anomalous for any IDS at the website. For a signature based router IDS, there is nothing anomalous in packet contents or patterns. For an anomaly based router IDS, there is nothing anomalous if such a traffic pattern consisting of so many unique flows is occurring daily. Typically, there are millions of web flows like the attack short-lived TCP flows passing daily via a major router of any ISP. If an attacker has more than 20,000 websites, then the quiet attack can become much more difficult to be tracked and detected at the router. While launching this attack, a botmaster will need network feedback or amount of congestion traffic at the target link. To get network feedback, a botmaster can use a tool like abget. The botmaster will initiate new bot requests at every interval T depending upon the network feedback. The network feedback is useful if ISPs do load sharing so that botmaster can detect less congestion and add more bots to send additional attack traffic. We consider available bandwidth of more than 1000bps for more than 5mins to be a threshold to add more attack short-lived TCP flows. A botmaster will use re-use capacity estimation techniques like capprobe to approximate the size of the bottleneck link to adjust the number of bots. ISPs typically do not use dynamic load sharing techniques as it leads to some negative effects although using BGP they can advertise different CIDR blocks to peering ISPs for load sharing [9]. The attack model can be summarized as an algorithm, as shown in Fig. 2.

IV. Simulation Results

We used ns2 to demonstrate the feasibility of our proposed attack model. We adopted the dumbbell topology which allows us to focus on the bottleneck queue, and provides the ability to modify parameters such as the access link bandwidth, the delay, and the number of clients and servers [10]. The bottleneck link has a capacity of C=10 Mbps, and each of the access links has a capacity of 100 Mbps. The link delay of the client-side access links is uniformly distributed between 50 and 100ms to simulate the varying roundtrip-time for each connection; all other links have a delay of 10ms. The router has a simple droptail queue with size equal to 500 packets, following the specification that the buffer size should be twice that of the delay bandwidth product. The legitimate traffic consists of 5 FTP flows and HTTP traffic generated by PackMime in ns2. The HTTP traffic rate of 40 connections/sec is used to generate an average traffic of approximately 30% of the bottleneck link. Thus, the legitimate traffic contains

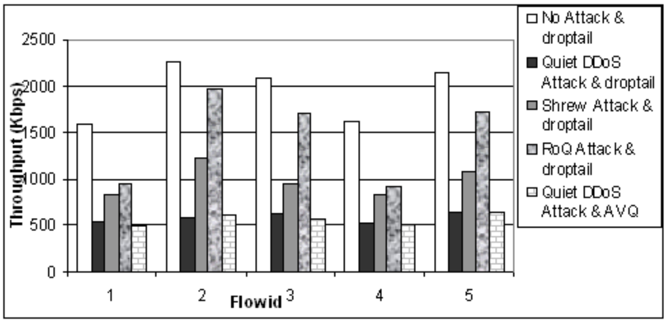


Fig. 3. Effect of Quiet DDoS attack on throughput.

the mix of short-lived and long-lived flows that is found in the Internet where measurements show that short-lived flows account for 30-40% of the traffic volume (in bytes), and the rest is occupied by long-lived flows. In the absence of the attack, the available bandwidth at the bottleneck queue is always negligible. The attack traffic consists of a set of 20 ftp flows, each transferring a small file of size 750KB to simulate bots accessing webpages. These attack flows are introduced at every interval of T from different nodes, thus simulating bots of a botnet. The time period T is kept random between 0s and 1s to remove deterministic pattern. To simulate a real botnet, we have nodes acting as zombies at 1, 2, 3, and 4 hops away from the target router. Each of the nodes has variable access link delays to the target router. We use pathchirp to gather available bandwidth at the bottleneck. The effect of attack traffic on ftp flows is shown in Fig. 3; it is indeed more detrimental than the shrew and RoQ attacks. The throughput of ftp flows is reduced by almost 70% when there is attack. Note that quiet attack does not affect the web traffic significantly due to the lack of burstiness, but we conjecture that increasing more short-lived flows can be damaging. We have left this investigation as part of the future work.

We have seen in our experiments that the attack traffic cannot be mitigated by adaptive queue management system AVQ [11] since the attack traffic adapts to the network congestion which is a property of the TCP protocol. The throughput of long-lived flows under AVQ is shown in Fig. 3. We also tried to use the aggregate congestion control scheme pushback [12] to mitigate the quiet attack. The pushback scheme also fails because the attack traffic uses TCP and network feedback. For the pushback scheme, we had to use a slightly different topology with four routers between sources and sinks. The network traffic was composed of a legitimate TCP flow and the attack traffic. The throughput of TCP flow was 277 Kbps with the attack and pushback scheme, and it was 954Kbps without the attack.

V. Existing Attack Models

The new breed of DoS attacks such as Shrew attacks and RoQ attacks [1],[13] have emerged. Both rely on sending high rate UDP traffic periodically. Their deterministic periodic pattern and high rate make them detectable, and several detection systems have been proposed. TCP vs TCP attack [14] is similar to the shrew attack, but it uses short-lived TCP flows instead of UDP. It suffers from the same issues of sudden high rate and deterministic periodic attack pattern. Thus, in [1], [13], and [14], an attacker sends traffic at rate equal to the target capacity to induce packet drops while our prosposed quiet attack has no such requirement on the aggregate attack rate, thereby making the traffic less bursty. A typical DDoS attack sending high rate UDP, ICMP or TCP SYN packets continuously can affect web traffic, but these attacks can be detected by detection systems like pushback. DDoS attacks such as the SYN attack have distinguishable traffic characteristics like abnormal quantities of SYN packets as compared to normal traffic. ICMP and UDP attack traffic can be easily detected as they normally occupy a small percentage of the regular traffic as compared to the TCP traffic. Our sophisticated attack model based on reconnaissance and execution is unique, and is adapted to be legitimate-like. Since no single server is under constant attack, the web servers are kept in the dark while ISPs who manage routers are made to believe that their routers are under real congestion. Finally, the simulation and comparison results demonstrate that our attack model performs better than those reported in [1], [13], and [14] in terms of evading detection.

VI. Conclusion and Future Work

We have proposed a new DDoS attack model by using botnets that is evadable and can be easily mistaken as real congestion. We believe botnet mitigation, such as better CAPTCHAs, is an important part of the defense strategy against the quiet attack. We will be building a more focused defense against such a grave threat in our future work.

References

[1] A. Kuzmanovic and E. Knightly, "Low-rate TCP-targeted denial of service attacks (The Shrew vs. the Mice and Elephants)," in *Proc. ACM SIGCOMM 2003*, Kalrushe, Germany, Aug. 2003 pp. 75-86.

[2] R. Kapoor, L. Chen, L. Lao, M. Gerla, and M. Sanadidi, "CapProbe: a simple and accurate capacity estimation technique," in *Proc. ACM SIGCOMM 2004*, Portland, USA, pp. 67-78.

[3] D. Antoniades, M. Athanatos, A. Papadogiannakis, E. Markatos, and C. Dovrolis, "Available bandwidth measurement as simple as running wget," in *Proc. PAM 2006*, Mar. 2006.

[4] V. Rebeiro, R. Reidi, R. Baranuik, J. Navratil, and L. Cottrell, "pathChirp: efficient available bandwidth estimation for network paths," in *Proc. PAM 2003*, Apr. 2003.

[5] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice Hall, 1987.

[6] D. Dagon, C. Zou, and W. Lee, "Modeling botnet propagation using time zones," in *Proc. NDSS06*, San Diego, CA, Oct. 2005.

[7] Cartoreso. [Online]. Available: http://cartoreso.campus.ecp.fr/.

[8] "Is captcha moment passing?" *Wired*, Apr. 2008. [Online]. Available: http://blog.wired.com/monkeybites/2008/04/is-captchas-mom.html.

[9] M. Ceasar and J. Rexford, "BGP routing policies in the Internet," *IEEE Network*, vol. 19, no. 6, Nov./Dec. 2005, pp. 1-6.

[10] S. Floyd and E. Kohler, "Internet research needs better models," in *Proc. Hotnets-I*, Princeton, NJ, Oct. 2002.

[11] S. Kunniyur and R. Srikant, "An adaptive virtual queue algorithm for active queue management," *IEEE/ACM Trans. Networking*, vol. 12, no. 2, pp. 286-299, Apr. 2004.

[12] R. Mahajan and S. Floyd, "Controlling high-bandwidth flows at the congested router," in *Proc. IEEE ICNP*, CA, USA, Nov. 2001, pp. 192-201.

[13] M. Guirguis, A. Bestavros, and I. Matta, "Exploiting the transients of adaptation for RoQ attacks on Internet resources," in *Proc. IEEE ICNP 2004*, Berlin, Germany, Oct. 2004, pp. 184-195.

[14] S. Ebrahimi-Taghizadeh, A. Helmy, and S. Gupta, "TCP vs. TCP: a systematic study of adverse impact of short-lived TCP flows on long-lived TCP flows," in *Proc. IEEE INFOCOM 2005*, Miami, USA, Mar. 2005, pp. 926-937.