Proceedings of 10th Global Engineering, Science and Technology Conference
2-3 January, 2015, BIAM Foundation, Dhaka, Bangladesh, ISBN: 978-1-922069-69-6

A Realistic Approach and Mitigation Techniques for Amplifying DDOS Attack on DNS

Muhammad Yeasir Arafat, Muhammad Morshed Alam and Feroz Ahmed

Domain name system (DNS) amplification attacks extremely exploit open recursive DNS servers generally for performing bandwidth consumption amplifying distributed denial of service (DDoS) attacks. The amplification effect lies in the fact that DNS response messages substantially larger than DNS query messages. In this paper, authors present and evaluate a practical approach that is able to distinguish between valid and bogus DNS replies. The propose scheme can effectively protect DNS servers acting both proactively and reactively. In this paper, authors shown DNS DDoS attack and also suggest a mechanism that can protect a DNS server from amplifying DDoS attacks especially the attacks targeting the bandwidth consumption of the victim server. We propose a new defence based on Iptables and routine fail2ban detection. The attack flow detection mechanism detects attach flows based on the indication or stress at the server, since it is getting more difficult to identify bad flows only based on the incoming traffic patterns. Our analysis and the corresponding real-usage experimental results demonstrate that the propose scheme offers a flexible, strong and effective solution for amplifying DDoS attack on DNS.

Keywords: DDoS, DNS Amplification Attack, Recursive, UDP, iptables, fail2ban

1. Introduction

Denial of service (DoS) attack is a malicious attempt to disrupt the service provided by networks or servers. The power of a DoS attack is amplified by incorporating over thousands of zombie machines through botnets and mounting a distributed DDoS attack. Although many defence mechanisms have been proposed to counter DDoS attacks, this remains a difficult issue, especially because the attack traffic tends to mimic normal traffic recently.

Over the past few years, the size and frequency of DDoS attacks have grown dramatically as attackers take advantage of botnets and other high-speed Internet access technologies to overcome their target's network infrastructure. In fact, according to Arbor's sixth annual worldwide Infrastructure security report, the largest-recorded DDoS attack has grown ten times in size from 2005 (10 Gbps) to 2013(300 Gbps). To make matters worse, the report also highlights a growing new trend with DDoS attacks. Not only are DDoS attacks getting larger and more frequent, but they are also becoming more sophisticated as they pinpoint specific applications (e.g., DNS, hyper text transport protocol (HTTP) or voice over internet protocol (VoIP) with smaller, stealthier attacks.

A DNS amplification attack is a popular form of DDoS, in which attackers use publically accessible open DNS servers to flood a target system with DNS response traffic. The primary technique consists of an attacker sending a DNS name lookup request to an open DNS server with the source address spoofed to be the target's address. When the DNS server sends the DNS record response, it is sent instead to the target. Attackers normally submit a request for as much zone information as possible to maximize the amplification effect. In case of most attacks of this type observed by United States computer emergency readiness team (US-CERT), the spoofed queries sent by the attacker are of the type, "ANY" which returns all known information about a DNS zone in a single request. Because the size of the response is significantly larger than the request, the attacker is able to increase the amount of traffic directed at the victim. By

---

Department of Electrical and Electronic Engineering, School of Engineering and Computer Science, Independent University, Bangladesh, Email: yeasir08@yahoo.com, morshed380@yahoo.com & fahmediub@gmail.com

leveraging a botnet to produce a large number of spoofed DNS queries, an attacker can create an immense amount of traffic with little effort. In addition, because the responses are legitimate data coming from valid servers, it is extremely difficult to prevent these types of attacks. While the attacks are difficult to stop, network operators can apply several possible mitigation strategies.

The remainder of this paper is organized as follows. We first discuss related study of amplify DoS attack in DNS in Section 2. Then, described the characteristic of amplifying DDoS attack in Section 3. In Section 4, we showed how to amplifying DDoS attack in DNS. In Section 5, we propose a mitigation technique based on iptables, open recursive solution in bind. In Section 5, we also showed how to save DNS server using linux IPtables rules called fail2ban and showed how to find the BOT from DNS log file. Finally, conclusions are presented in Section 6.

2. Background and Related Study

In the area of DNS traffic analysis, the most related work in this area is rendered by Oberheide et al. who analyse DNS queries that target dark net sensors. The authors characterize these traces and propose a mechanism to implement a secure DNS service on dark net sensors. Moreover, Paxson is among the first to pinpoint the threats of DNS reflectors on making DDoS attacks harder to defend. In another work, Tong, Xiao, WANG analyse corrupted DNS resolution paths and pinpoint an increase in malware that modified these paths and threatened DNS authorities. In comparison to our work, Oberheide et al. have not linked or investigated any DNS DDoS traces through their analysis but solely focused on analysing DNS traffic. On the other hand, Paxson did not investigate dark net data. Therefore, all DNS amplification traces destined to unused IP addresses (dark net) cannot be detected through their analysis. However, dark net and other sources of data could be associated to extract further intelligence on DNS amplification DDoS activities such as the approximate number of infections. Future work could consider the latter task.

DNS queries and responses are mostly user datagram protocol (UDP) based, it is vulnerable to spoofing-based DoS attacks, which are difficult to defeat without incurring significant collateral damage. The key to prevent this type of DoS attacks is spoof detection. There is little research work towards the DNS amplification attacks. Adam, Zare provides a thorough analysis about reflection-based DoS attack. Two attack strategies against DNS are analysed. Unfortunately, these two attacks can be controlled by filtering out replies to spoofed request at the victim site and restricting recursive servers to serve local machines only. The DNS security extension (DNSSEC) is designed to provide data integrity and authentication instead of authenticating the requester. It has no protection against DoS attacks. Xi YE, Yiru YE present a simple and practical method that is able to distinguish between authentic and bogus DNS replies. The proposed scheme, acts proactively by monitoring in real time DNS traffic and alerting network administrators when necessary. Once the attack is confirmed, our approach is then activated to filter out all the illegitimate DNS responses by automatically updating firewall rules to ban bogus packets.

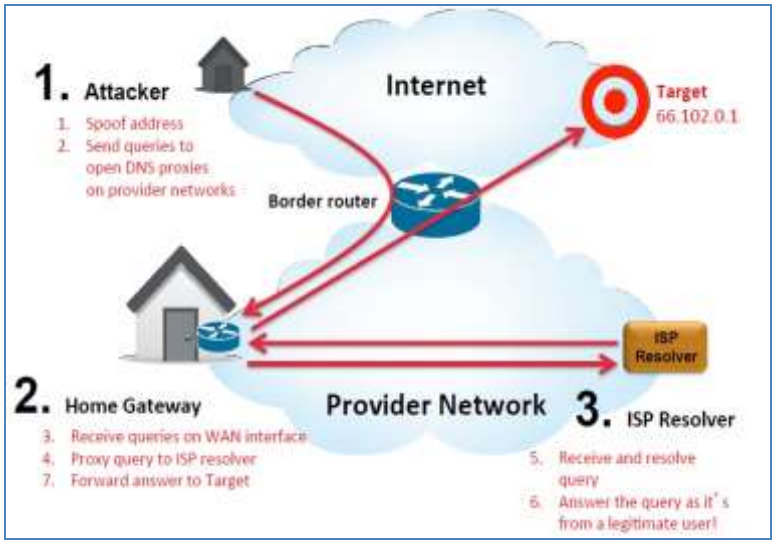
3. Characteristics of the DNS Amplification Attack

In arrange to initiate a DNS amplification refection attack the attacker desires to execute two tasks. First task the attacker spoofs the address of the victim. This is the refection part; it wills origin all the replies from the DNS server to be directed to the victim's server. This can easily be done since in UDP no handshake like in transmission control protocol (TCP) is being done between the client and the server. Secondly the supplicant searches for responses that are several times bigger than the request. The attacker achieves an amplification factor because the response is several times larger than the request. The amplification can even be larger when DNSSEC is used, because of the signatures used the size of the response increases. The

amplification can even be larger when DNSSEC is used, because of the signatures used the size of the response increases. Now the attacker is ready to perform the attack. The attackers launch a stream of small queries originating from a group of infected computers (referred to as a botnet) to one or multiple authoritative DNS servers. The DNS servers will then reply to the resolver. However, because the attacker spoofed the address of the victim, all the traffic is directed to the victim.

The victim gets overloaded with the amount of traffic send to it and possibly cannot make use of the internet connection anymore. Not only bandwidth is exhausted but also the resource on the client's machine becomes flooded. The client's machine can be so busy processing the incoming traffic that is exhaust the resources; this could lead to a halt of the client's machine. So a DNS refection amplification attack could lead to two types of Denial of Service. DNS amplification attack process showed in Figure 1.

Figure 1: DNS amplification DDoS attack.



The relation between a request and the corresponding response is known as the amplification factor and is computed by the following formula:

Amplification Factor = size of (response) / size of (request)

The bigger the amplification factor is, the quicker the bandwidth and resource consumption at the victim is inflicted. From the study of the DNS amplification attack, three major characteristics are identified. The first characteristic is that a DNS amplification attack must use port 53 and UDP protocol. The second characteristic of a DNS amplification attack is a massive volume of UDP packets over a very short time period (over 4000 UDP packets in response per second). The third characteristic of DNS amplification attack is that incoming and outgoing IP addresses of the packets do not match. Because attackers exploit IP spoofing, the incoming and outgoing IP addresses do not match in the victim server. Therefore through comparing the incoming and outgoing IP addresses an intelligent algorithm can detect if a DNS amplification attack has occurred.

4. Amplification DoS Attack on DNS

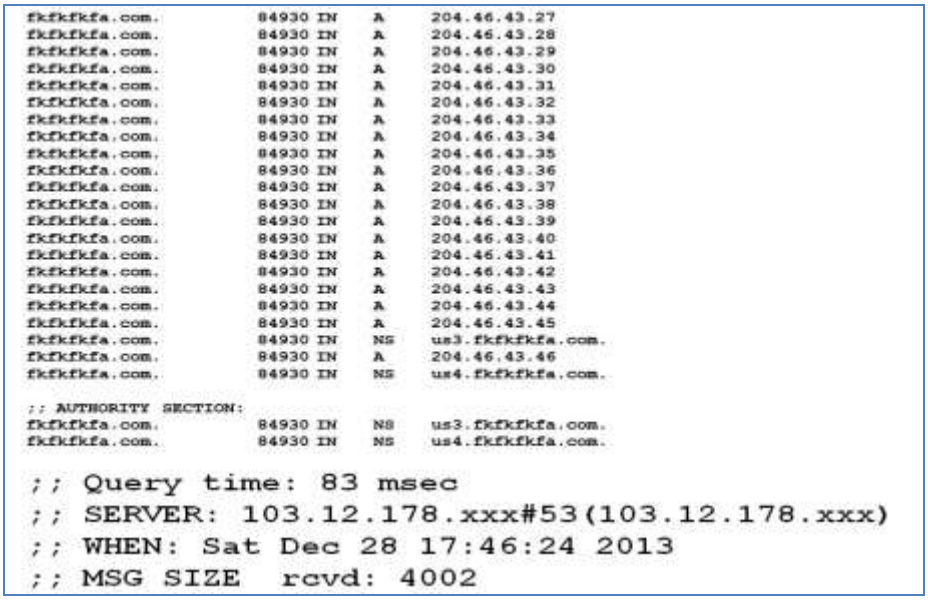
An ANY query returns all the records for a specific domain name despite of the record type. When launch to a recursive server, the server can solely return the records that it has cached. The server can have to be compelled reply, despite of available recursion. This is currently the

most frequent attack because the ANY request usually returns a large collection of resource records, making a high amplification ratio.
Hacker creates their own domain and increases the DNS response size so that they can get higher amplification. In this case they use the domain fkfkfkfa.com which is not a legitimate domain name. Now check the interesting part not the animation. Command of DNS attack is given below

[root@ns3 ~]# dig ANY fkfkfkfa.com @103.12.178.XXX +edns=0 +notcp +bufsize=4096

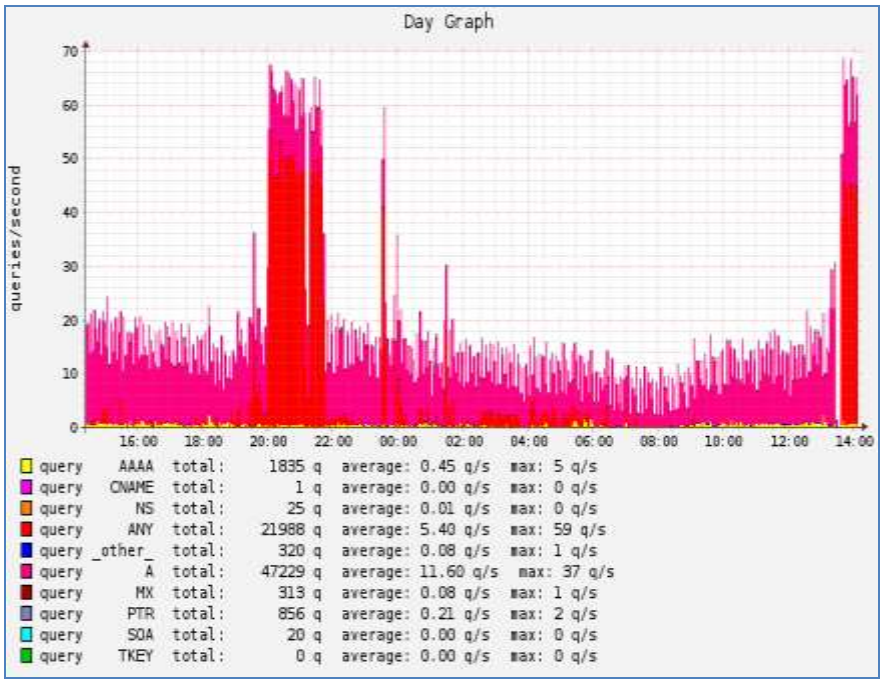
In this command using UDP packet with buffer size 4096. It says that the query takes 83 msec, server who response to this query. This is the part where are interested. It's a 64 byte query and response is 4002 byte. Average DNS query size is 64 bytes but if we look at the response it is 4002 bytes. That means it's amplifying the request by roughly 4002/64 = 62x times amplification. Query output showed in Figure 2.

Figure 2: Response size of query



So 83.69.230.xxx can launch 1Mbps of DNS query, he can amplify it by 64 times and can send 64Mbps of traffic to ietf.org. It's really impressive. That's why it's important to secure your DNS. Usually I have 20/25 queries/second. But there are few spikes where I have 70 queries/second and most of them are ANY query. When I check my DNS query log what I get is really interesting. To track my DNS query I have configured bind graph. Bellow is the output in Figure 3:

Figure 3: DNS DDoS attack showed in Bindgraph



5. Proposed Mitigation Mechanism

In this paper, we approach some mechanism that can protect DNS amplifying attack, especially, the attacks targeting the resources, including consume bandwidth. It is not difficulty to lunch a DNS amplification attack, because 75% name server in world is an open resolver. Therefore, DNS amplification attacks may be stealthier and more dangerous for the DNS servers, owing to its amplification in attack effect and its difficulty to trace the attacker. The configuration information has been limited to Berkeley Internet Name Daemon (BIND9) and Microsoft's DNS Server, which are two widely deployed DNS servers on federal networks.

5.1 Approach 1: Iptables

Conventional host based firewall using IPtables. In practice there may be thousands of nodes. Billions of packets can be directed at the victim, taking up all available bandwidth or causing DoS. The following Perl script has been developed to stop DoS attacks. There is a script for dropping packets from a offending IP/range if it exceeds 30 requests per second let's say for our purposes the range is 202.4.96.0/24

```
#!/bin/bash
/sbin/iptables -I INPUT 1 -p udp --dport 53 -m limit --limit 30/sec -s 202.4.96.0/24 -j ACCEPT
/sbin/iptables -I INPUT 2 -p udp --dport 53 -s 202.4.96.0/24 -j DROP

/sbin/iptables -I OUTPUT 1 -p udp --dport 53 -m limit --limit 30/sec -d 202.4.96.0/24 -j ACCEPT
/sbin/iptables -I OUTPUT 2 -p udp --dport 53 -d 202.4.96.0/24 -j DROP
/sbin/iptables -I FORWARD 1 -p udp --dport 53 -m limit --limit 30/sec -s 202.4.96.0/24 -j ACCEPT
/sbin/iptables -I FORWARD 2 -p udp --dport 53 -s 202.4.96.0/24 -j DROP

/sbin/iptables -I FORWARD 1 -p udp --dport 53 -m limit --limit 30/sec -d 202.4.96.0/24 -j
ACCEPT
/sbin/iptables -I FORWARD 2 -p udp --dport 53 -d 202.4.96.0/24 -j DROP
```

5.2 Approach 2: Disabling Recursion on Authoritative

Many of the DNS servers currently deployed on the Internet are exclusively intended to provide name resolution for a single domain. In these systems, DNS resolution for private client systems may be provided by a separate server and the authoritative server acts only as a DNS source of zone information to external clients. These systems do not need to support recursive resolution of other domains on behalf of a client, and should be configured with recursion disabled. To stop recursion need to add following lines in vi /etc/bind9/ named.conf in the public view to prevent bind from responding with root referrals.

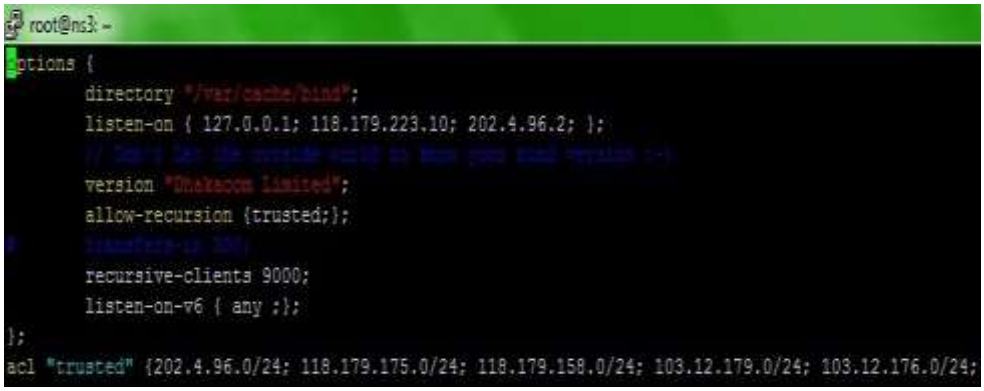
options

```
{
    allow-query-cache { none; };
    recursion no;
};
```

5.2.1 Limiting Recursion to authorized Clients

For DNS servers that are deployed within an organization or Internet Service Provider, the resolver should be configured to perform recursive queries on behalf of authorized clients only. These requests typically should only come from clients within the organization's network address range. We highly recommend that all server administrators restrict recursion to only clients on the organization's network. In this is case we need to add trusted acl list in named.conf. Recursion to authorized network showed in Figure 4.

Figure 4: Adding authorized network in trusted list.



5.3 Approach 3: Response Rate Limiting (RRL)

RRL is a mechanism for limiting the amount of unique responses returned by a DNS server. This can limit the effectiveness of a DNS amplification attack by dropping responses that exceed the configured rate limit. When using RRL the victim might still notice it is under attack, because it receives DNS responses for which no request was sent out for a limited time. An attacker might also be able to circumvent this defence mechanism by distributing its attack over a large number of DNS servers, to stay under the RRL limits of the DNS servers. On BIND9 implementation running the RRL patches, include the following lines to the options block of the authoritative views.

```
rate-limit {
slip 2;  // Every other response truncated
window 15; // Seconds to bucket!
responses-per-second 5; // # of good responses per  prefix-length/sec
referrals-per-second 5;  // referral responses
nodata-per-second 5;  // nodata responses
nxdomains-per-second 5;  // nxdomain responses
errors-per-second 5;  // error responses
all-per-second 20;  // When we drop all
}
```

5.4 Approach 4: Fail2ban

Fail2ban [11] operates by monitoring log files (e.g. /var/log/pwdfail, /var/log/auth.log, etc.) for selected entries and running scripts based on them. Usually this is used to block selected IP addresses that may belong to hosts that are trying to break the system's security. It can ban any host IP that makes too many login attempts or performs any other unwanted action within a time frame defined by the administrator.

5.4.1 Configuring Fail2ban

“enabled” defines whether or not a given section is enabled or nor, it’s possible values are true or false. “filter” this is not used in the default section as it is used to tell fail2ban client what it is looking for in the log file, its values could be among others likes apache-badbots, sshd, https, asterisk etc. Basically it is how the service is identified on the log file being parsed. “action” this option tells fail2ban what action to take once a rule is broken, could be specified a default action in the default section, and overwritten on each jail section may need to change the default value. “ignoreip” this option is used to set one or some IPs that should not be blocked, no matter how many times a users fail in login from those IPs. “maxretry” this option is used to set the limit of retries a user have before he gets blocked. Edit vi /etc/fail2ban/jail.conf file and add this section. Configuration snap shown in Figure 5.

Figure 5: Filter Configuration in Fail2ban

```
[iptables-dns]
enabled = true
ignoreip = 127.0.0.1
filter = iptables-dns
action = iptables-multiport [name=iptables-dns, port="53",
protocol=udp]
logpath = /var/log/iptables/dns_reqs.log
bantime = 86400
findtime = 120
maxretry = 1
[named-refused-udp]
enabled  = true
[named-refused-tcp]
enabled  = true
```

5.4.2 Filter Configuration

Now need to create the filter. Configuration snap shown in Figure 6. The following filter configuration files are stored in /etc/fail2ban/filter.d/:

Figure 6: Jail Configuration in Fail2ban

```
[Definition]
failregex = fw-dns.*SRC=<HOST> DST
failregex = ^.* security: info: client #.*: query \(cache\)
'./(NS|A|AAAA|MX|CNAME)/IN' denied

ignoreregex =
```

Now verify that fail2ban is doing something by checking out the log file located at /var/log/fail2ban.log it should contain something like in Figure 7.

Figure7: Fail2ban banned log from the server

```
[root@ns3 ~]# $ sudo tail -f /var/log/fail2ban.log
2014-05-21 09:44:42,800 fail2ban.actions: WARNING [named-refused-udp] Ban
118.179.4.5
2014-05-21 07:46:12,902 fail2ban.actions: WARNING [named-refused-tcp] Ban
202.4.96.2
```

6 Conclusion

In this paper, we investigated a new two-stage mechanism that can protect DNS servers from amplification DoS attacks. The proposed mechanism is based on three key ideas. The first one is an iptables scheme in the first stage, which protects the servers from a sudden surge of attack flows. In second one we stop unauthorized recursion by trusted network list allow in bind. We also set response rate limit in bind. The amplification of illegitimate responses can be limited by implementing RRL on authoritative name servers. RRL can prevent false positives by setting SLIP. We also investigated the condition to detect the victim servers and freeze the whitelist based on the server response time in detail. The third key idea is to detect attack flows based on the concept of a whitelist-based admission control defined for each pair of client and server IP addresses in the second stage. The experiment results show that whitelist-based admission control policies attack flow detection mechanism distinguishes attack flows from normal flows and effectively filters the IP addresses of the attackers from the band list. Although we focused on protecting DNS servers from amplify DDoS attack in this paper, the proposed approach will be extended to other types of DNS attack in future study.

References:

Adam Ali.Zare Hudaib, Esra'a Ali Zare Hudaib (2014), “DNS Advanced Attacks and Analysis,” *International Journal of Computer Science and Security (IJCSS), Vol. 8, No. 2, Pp.63-74*

Application-layer attacks are on the rise, according to Arbor’s sixth annual Worldwide Infrastructure Security Report, *www.arbornetworks.com, 2014*

J. Oberheide, M. Karir, and Z. M. Mao (2007), "Characterizing dark dns behavior," *In Fourth GI International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA).*

Muhammad Yeasir Arafat, Muhammad Morshed Alam, Feroz Ahmed (2014), "Study on Security issue in open source SIP server," *Journal of Modern Applied Science, vol.8, No. 2, Pp. 114-124.*

Tong Guang NI, Xiao Qing GU, Hong Yuan WANG (2014), "Detecting DDoS Attacks Against DNS Servers Using Time Series Analysis," *Indonesian Journal of Electrical Engineering, Vol.12, No.1, Pp. 753 ~ 761.*

V. Paxson (2011),"An analysis of using reflectors for distributed denial-of-service attacks," *SIGCOMM Comput. Commun.Rev., vol. 31, no. 3, Pp. 38 – 47*.

Vlajic N, Andrade M, Nguyen U (2012), "The Role of DNS TTL Values in Potential DDoS Attacks," *Procedia Computer Science, 52(10), Pp. 466- 473.*

Xi YE, Yiru YE (2013), "A Practical Mechanism to Counteract DNS Amplification DDoS Attacks," *Journal of Computational Information Systems, 9(1), Pp.265–272.*

Xie Y, Tang S, Huang X (2013), "Detecting latent attack behavior from aggregated Web traffic," *Computer Communications, 8(36), Pp. 895-907*.

Yan R, Zheng Q, Li H (2013), "Combining Adaptive Filtering and IF Flows to Detect DDoS Attacks within a Router," *KSII Transactions on Internet and Information Systems, (4), Pp. 428-449.*

Z. Zhu, G. Lu, Y. Chen, Z. J. Fu, P. Roberts, K. Han (2008), "Botnet Research Survey," in *Proc. of IEEE COMPSAC, Pp. 967-972.*