A proposal of a countermeasure method against DNS amplification attacks using distributed filtering by traffic route changing

Yuki Katsurai[*], Yoshitaka Nakamura[**] , and Osamu Takahashi[**]

[*]Graduate School of Systems Information Science, Future University Hakodate, Japan
[**]School of Systems Information Science, Future University Hakodate, Japan
{g2114005, y-nakamr, osamu}@fun.ac.jp

Abstract - In recent years, victims of DDoS attacks have been increasing rapidly all over the world, and it has become a very serious problem for network service providers. In particular, DNS amplification attacks have attracted attention. These attacks utilize DNS servers to cause huge damage to services using network systems. There are some methods that network administrators can introduce as countermeasures to DNS amplification attacks. Examples include a method to change the setting of DNS servers and a method to perform packet filtering on a firewall or routers. However, in these methods, it is not possible to suppress the damage to the network due to the large amount of packets passing through the system. Also, in the method of applying filtering, there is the problem that network congestion occurs on the processing terminals. In this paper we propose a countermeasure method against DNS amplification to reduce damage to the network. Our method is implemented on multiple routers on a network and performs distributed filtering using route-changing to prevent attack packets from reaching the target server. We also evaluate the utility of our method from the viewpoint of reducing the number of processes of each filtering terminal and the load on the network.

Keywords: DNS amplification attacks, DDoS, iptables, UDP, filtering

1 INTRODUCTION

In recent years, services that use the Internet have become familiar to people because of the development of the information society. However, damage from cyber-attacks, which are typified by DoS attacks (Denial of Service attacks) and DDoS attacks (Distributed Denial of Service attacks), has also increased. DoS attacks are a kind of cyber-attack which attacks the devices constituting the network, and thereby inhibit the provision of services. DDoS attacks are DoS attacks carried out using several dispersed sources. Among these DDoS attacks, the kind that has been typically exploited for many years is DNS amp attacks (DNS amplification attacks). A DNS amp attack refers to an amplification attack using a DNS server. The server reflexively responds to inquiries from a source, and acts as both a reflector and an amplifier. DNS amp attacks exploit these characteristics. In RFC 5358 / BCP 140, DNS amp attacks have been defined as Reflector Attacks [1], but in this research we unify such attacks under the name which by which they are commonly referred to. Figure 1 shows an overview of a DNS amp attack.

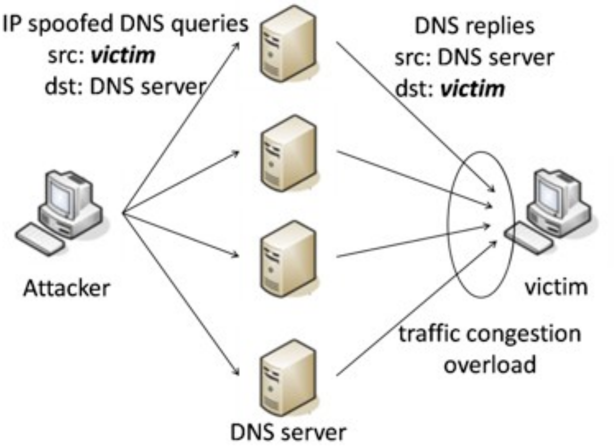


Figure 1: Overview of DNS amp attacks

In DNS amp attacks, name resolution requests of which the source IP address is spoofed are transmitted from an attacker to DNS servers. DNS servers that have received them return a response towards the terminal of which the IP address has been spoofed. Thus, DNS amp attacks lead to network congestion and overload the processing capacity of the victims. The hazards of this attack have been pointed out from 2001 [2].

There are two main types to DNS server: authoritative DNS server and cache DNS server. An authoritative DNS server shares a part of the domain name space it manages with multiple other DNS servers, and manages the distributed data by forming a tree structure. A cache DNS server is also called a resolver. It queries the authoritative DNS server when receiving a name resolution request, and it returns the results to a client. An authoritative DNS server that has a name resolution function enabled which is not inherently necessary, and a cache DNS server that processes a name resolution request sent from outside the network are called open resolvers. Incidentally, a home router can also function as a cache DNS server, so cases in which a home router is used to attack as an open resolver have occurred.

2 RELATED WORK

As a countermeasure to DNS amp attacks, there is a method to prevent DNS servers being used as an amplifier. Also there is a self-defense method that can be used by potential victims. In this chapter, we describe each measure.

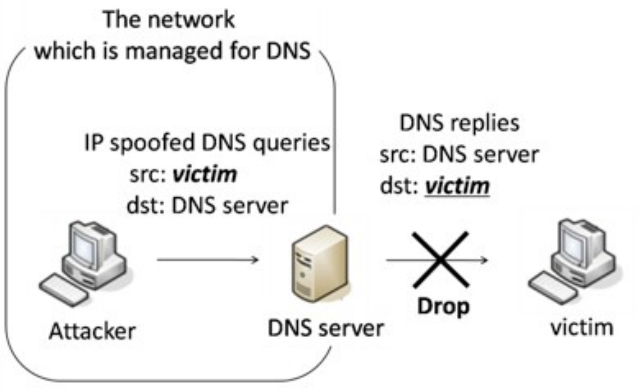


Figure 2: Measure using access control

2.1 Measures to be applied to the DNS servers

In the measure to ensure that DNS servers will not be used to attack others, the server acts as a protection. Regarding this type of countermeasure method, there are separate methods to ensure that cache DNS servers authoritative DNS servers are not used for DNS amp attacks. We describe each method.

2.1.1 Measures for cache DNS servers

There are two main approaches, which are performing access control and performing packet filtering using the router.

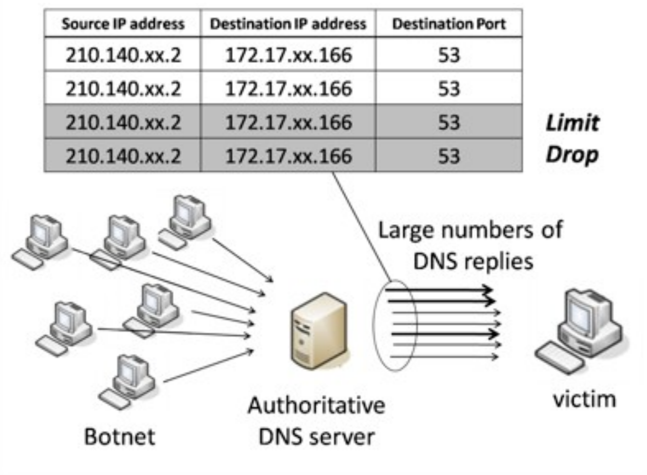
In access control, cache DNS servers only permit access to DNS queries from a client that the cache DNS servers regard as a target user, based on IP address. Figure 2 shows a schematic diagram of access control.

Figure 2 represents a situation in which a victim present on the outside of a network managed by the DNS server is set as a target of DNS amp attacks. In this case, the response a cache DNS server sends to a victim who is outside the scope of services is discarded, as dictated by a setting of the server. Therefore, the risk of the server being used as a stepping- stone in DNS amp attacks on the outside of the network is reduced. The details of this countermeasure are set out in the RFC 5358 / BCP 140 [1].

In packet filtering, a setting that prevents the transmission and reception of packets with spoofed source IP addresses is added to network devices such as routers. Spoofed packets do not reach a victim or a DNS server, thus attacks are eventually prevented. The packet filtering method is described under the name of Source Address Validation in RFC 2827 / BCP 38 [3], and also in RFC 3704 / BCP 84 [4]. In addition, this method has been used as a countermeasure against not only DNS amp attacks but various other kinds of cyber-attack.

2.1.2 Measures for authoritative DNS servers

In an authoritative DNS server, the source of a name resolution request is a cache DNS server. Also, authoritative DNS servers are providing services to the entire Internet, so it would be disadvantageous to use the access control method in the same way as cache DNS servers. If authoritative DNS servers receive inquiries from a wide range, sent through technologies such as botnets, this cannot be addressed with access control. Moreover, in DNS amp attacks using the authoritative DNS servers, there is a tendency that the size of response packets of the DNS servers is larger than that of the packets in DNS amp attacks using the cache DNS servers. Thus, further measures have been required. In response to this fact, Paul Vixie et al proposed DNS RRL (DNS Response Rate Limiting) [5]. This method is a countermeasure that utilizes the fact that authoritative DNS servers return the same response at a high frequency to the same destination in a short time during DNS amp attacks. It monitors the response frequency, and if it exceeds a certain percentage, it limits and discards response packets. Also in this case, it is possible to respond to a variety of attacks by flexibly changing the conditions for determining the same responses. In Fig. 3, we show an example of the measures used DNS RRL.



Source IP address	Destination IP address	Destination Port	
210.140.xx.2	172.17.xx.166	53	
210.140.xx.2	172.17.xx.166	53	
210.140.xx.2	172.17.xx.166	53	Limit
210.140.xx.2	172.17.xx.166	53	Drop

Figure 3: Response limit using DNS RRL

A typical example of the problems of applying the DNS RRL is the occurrence of false detection. Since this determination is made based on statistics, whether it is an attack or not, if there are packets which should not originally have been detected it is determined that these are attack packets. To prevent this false detection, a retransmission request is sent to a cache DNS server using TCP. This action achieves a correct name resolution. In relation to this, Rozekrans et al have shown the results of field trials of DNS RRL [6]. In this reference, a method of giving an evaluation value for each client is applied, and this is called DNS dampening. The author wrote that it is necessary to verify the usefulness of attacks that currently exist, and to perform source verification in order to respond to development attacks in the future.

2.2 Measures that can be applied by victims for self- defense

An ideal countermeasure against DNS amp attacks is the simultaneous application of source verification to all of the

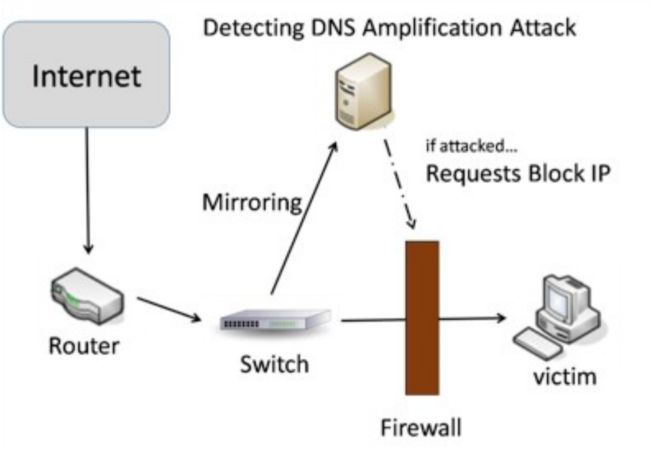


Figure 4: Filtering method using DDAA

network devices around the world, but it is not practical. The next best option is to apply access control and DNS RRL to DNS servers. However, DNS RRL is still under study. Also, these measures are intended to be applied to e each device by administrators of the network and DNS servers. In addition, devices and DNS servers to which measures have not been applied would still be exploited in attacks as stepping-stones. Therefore, a victim requires measures against DNS amp attacks as a means of self-defense. The method used at these times is filtering performed on the victim's side of the network. A method for filtering by attack detection and firewall on the victim's side of the network has been proposed by Ye et al [7]. In this approach, a switch is interposed between the Internet, which is the 'backbone', and the network on the victim's side, and it copies packets. The copied packets are sent to an installed system named DDAA (Detecting DNS Amplification Attacks). This system records the information of the packets, and then blocks packets that are considered attack packets with a firewall. Figure 4 shows this method including DDAA.

This method stores the IP address and destination port of the packets that pass through the switch in the DDAA's internal database. Therefore, the performance of the system is reduced as time elapses. For this reason, if the packet information stored in the database exceeds 10000, it is set to delete all the information which has been in the database more than three seconds. The advantage of this approach is that the information of the filtering can be dynamically updated and saved by the parameter settings of DDAA. In addition, by managing the detection and blocking at the same terminal, using the firewall it can directly drop DNS reply packets that are sent to a victim intermittently. The problem of this approach is that it does not consider the burden on devices and network congestion. Depending on the nature of the firewall, it may not be able to respond to the congestion of the network. Also, saving packets and constantly performing the matching process with the database results in a high load on the firewall and DDAA, which can affect performance. Paola et al proposed a method that maintains low loads on devices [8]. In this approach, packets passing through devices are retrieved efficiently from the database by using a Bloom Filter. Accordingly, the burden on devices is reduced, and it is possible to perform accurate packet filtering. However, regarding this approach, the influence on the network is not taken into consideration, and damage due to congestion in performing filtering is also overlooked.

3 PROPOSED METHOD

3.1 Research task

In this research, we assume a case in which DNS servers, to which countermeasures of the entire network are not applied, to have been used in DNS amp attacks. Our research deals with packet filtering as a means of self-defense means on the victim's side of the network. After attack detection, to ensure that attack packets do not reach the victim's service, we perform filtering along the network path. Further, by performing distributed filtering using multiple routers, the burden of the routers that perform filtering and the networks on either side of them is reduced.

3.2 System configuration

The system of the proposed method consists of the following contents; the Internet, as a backbone, a switch that exists close to the victim in the Internet, multiple routers which perform filtering, and a router which integrates packets that have changed route. In Fig. 5, a schematic view of the system is shown.

We place a switch at the connection point between each network and the Internet. It distributes the packets to an arbitrary number of routers to perform filtering. They operate as filtering routers. After completing the filtering, they send packets to a router that is used for integration of the packets, and it sends packets in a fixed order to the victim server.

3.3 Performance details

In the proposed method, there are four stages. They are: the time until a DNS amp attack is detected, distribution of packets after attack detection, distributed filtering, and

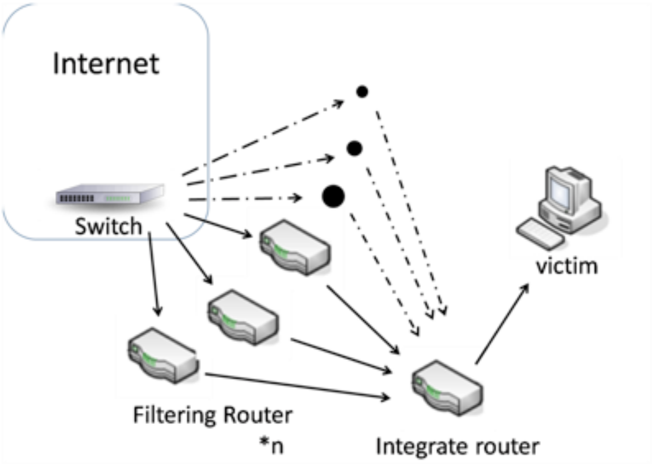


Figure 5: Schematic diagram of the proposed method

integration of packets after filtering. We will describe each of the steps.

3.3.1 Until a DNS amp attack is detected

In the proposed method, all routers and switches perform in the same way as existing devices until a DNS amp attack is detected. In the case of there being several routers, each one functions normally as a router within the network.

3.3.2 Distribution of packets

When a DNS amp attack is detected, a switch sends attack packets which are distributed to filtering routers. Immediately prior to distributing the packets, this switch sends commands ordering the commencement of filtering and stating which router will perform. As an example, router 1, router 2 and router 3 are set as filtering routers. In this case, it is assumed that router 1 is a router which is used as a general path. The switch adds fragment information to the packet head of packets being subjected to encapsulation. This fragment information has a similar meaning to 'flag field' and 'fragment offset field' used in the IP header, so it shows the information of what number this mass of packets is, out of all those that passed through after the detection of a DNS amp attack. Then it changes the route from router 1 to router 2. Similarly to the case of router 1, a switch encapsulates a defined amount of the packets, adds to them the fragment information and then transmits them to router 2. The same is true when the switch sends packets to router 3. Then the object returns to router 1, whereafter the same operation is repeated. Further, if the fragment offset has reached the upper limit, it is set to repeat from 0 again. Figure 6 shows the status of packet distribution.

If the packet distribution is stopped, the switch finally sends the packets to a router that has been used as a path before a DNS amp attack was detected. In this example, after the switch has finally sent the mass of packets to router 1, it also sends a number of masses of packets t to router 1, and then stops the division of the packets to resume communication as usual. This is in order to prevent the communication eventually being affected due to abnormalities in the order of the packets.

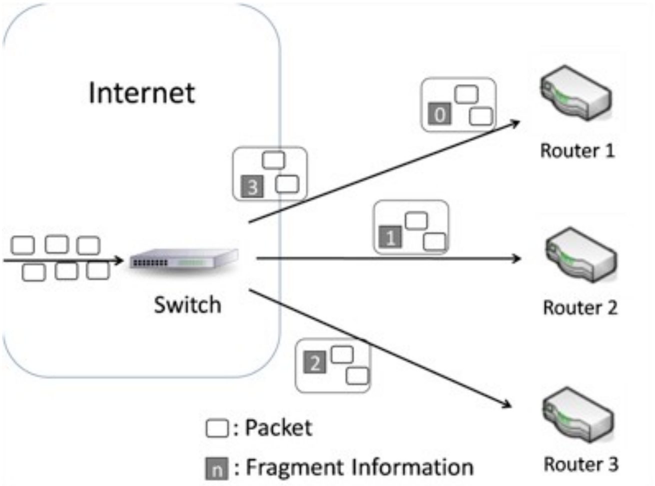
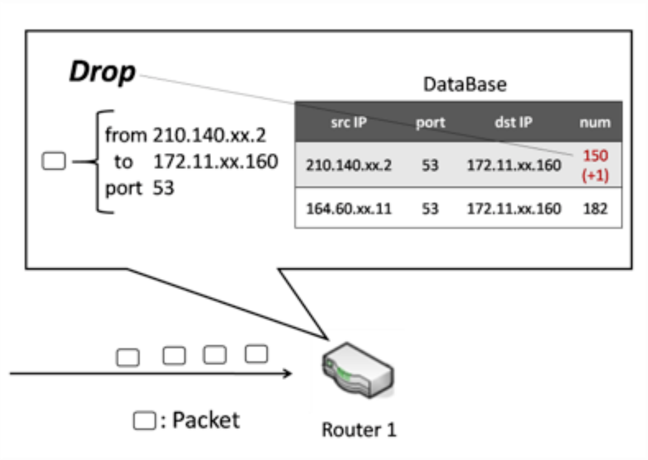


Figure 6: Distribution of packets by the switch



src IP	port	dst IP	num
210.140.xx.2	53	172.11.xx.160	150 (+1)
164.60.xx.11	53	172.11.xx.160	182

Figure 7: Overview of filtering by database collation

3.3.3 Distributed filtering

Each filtering router starts performing filtering after the switch notified it to do so and sends a mass of packets. First, packets are transmitted to the UDP53 port. If a packet filtered is addressed to the UDP53 port, filtering routers register the source IP address of the packet to their own database. Then the filtering routers repeat the same operation. They recognize a DNS server which has transmitted a certain amount or more packets within a specified time as a server being used as a stepping stone in DNS amp attacks. Subsequently, they share the information of the source IP address with the other filtering routers. Following these operations, they discard all DNS reply packets coming from a DNS server that is considered to be an attack source.

Figure 7 shows the state of filtering. In addition, when performing filtering, excepting information that is shared with other filtering routers, they reset the database at regular intervals. This is a measure to maintain a certain level of search efficiency regarding the information of the packets which are sequentially registered. Filtering routers send packets to an integration router after finishing the filtering for each mass of packets.

3.3.4 Integration of packets

The integration router sorts the mass of packets into the correct order by referring to the fragment offset information that was added by the switch. Then it transmits packets to the server of a victim in ascending order of number.

3.3.5 Relation of the processing

Figure 8 shows the relationship to other devices of each device in the filtering process of the proposed method.

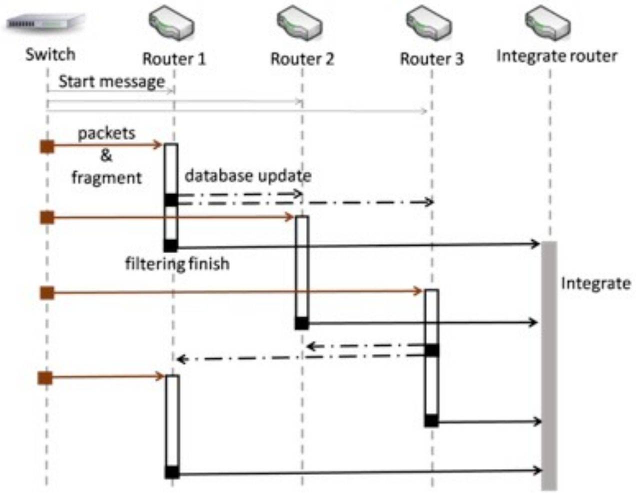


Figure 8: Relationship diagram of the filtering process

4 EXPERIMENTATION

About the proposed method in this research, we reproduced a DNS amp attack in a virtual environment and conducted an evaluative experiment. Hereinafter, we describe the experimental environment, and the details of the experiment contents the evaluation.

4.1 Experimental environment

As an experimental environment, we established virtual machines to act as an attacker and a victim, DNS servers, a switch, filtering routers and an integration router, and we applied the appropriate settings to each one. Thereafter, we created a scenario of a DNS amp attack. Table 1 shows the experimental environment.

4.2 Experiment contents

We gave the role of the devices that are used in the proposed method to virtual machines, and allowed a large number of packets to be sent to a victim using a virtual machine that was configured as a DNS server. After a certain time had elapsed from the start of the DNS amp attack , filtering operations began along the router path. For distributed filtering, filtering routers use the database and iptables to manage packet dropping and communication permission. In this experiment, we performed distributed filtering using two routers.

Table 1: The parameters used in the experiment

Using distribution	Ubuntu 12.04 32bit
Programming language	Python 2.7.3, PHP 5.3.3
Database	My SQL 5.1.73
Number of attack packets / s	1000

4.3 Evaluation contents

We will now evaluate the results of the above experiments.

i. Throughput between the switch and the victim
ii. The number of Processing packets and the percentage of blocking

For these comparisons, we compared single filtering and distributed filtering by calculating each value. The results are shown in the following section.

5 RESULTS AND DISCUSSION

Figure 9 shows throughputs from a switch to a victim. Figure 10 shows the number of packets that filtering routers processed and the percentage of these packets that were blocked.

These results indicate the utility of this research. Comparison of the throughput reveals that the adverse effect on a victim's side of the network is smaller when filtering using the proposed method than when filtering using a single router. It's a measure of the throughput of the normal communication packet with the exception of the attack packets. Furthermore, it is possible that the throughput can be raised by increasing the number of

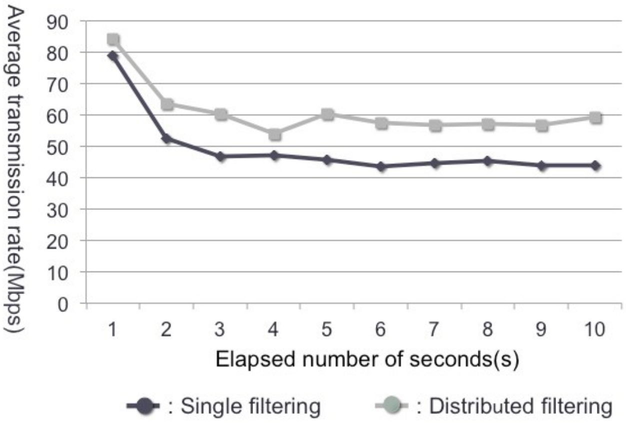


Figure 9: Graph of comparison of throughput from switch to victim

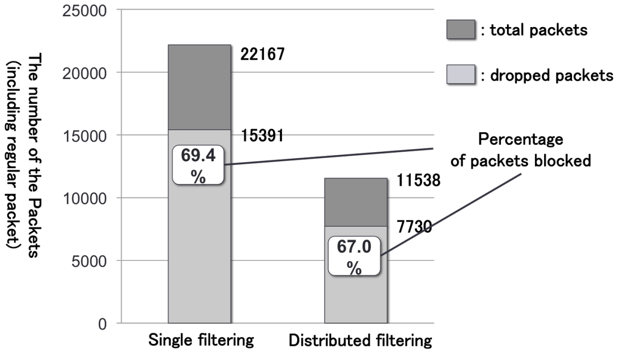


Figure 10: Graph of the number of packets filtering routers processed and the percentage of packets blocked

routers used for distributed filtering. Also, regarding the number of processing packets, the numbers of discarded packets and probability that the attack packet is blocked, the experimental results indicate that using the proposed method decreases the load per single router. By the difference between the time required for the distribution of the packet, the amount of normal communication packets transmitted is increased and block rate has somewhat changed. However, it seems that there is no significant impact on the accuracy of the block. From these results, the usefulness of this research has been proved.

6 CONCLUSION

In this paper, we have described a method of distributed filtering as a counter measure against DNS amp attacks. The purpose of this research is to reduce network congestion and the load on a router. We conducted an experiment to evaluate it, and it indicated the improvement of throughput in the network on the victim's side and decrease in the amount of processing packets per single router. From these results, the purpose of reducing the burden of the network and devices while maintaining the performance can be said to have been achieved. As future challenges, there are a survey of numerical change when the number of routers is increased, and an investigation into the performance of the switch when a large number of attack packets are sent to a victim.

REFERENCES

[1] J. Damas, and F . Neves, Preventing Use of Recursive Nameservers in Reflector Attacks, RFC 5358, BCP 140, https://www.ietf.org/rfc/rfc5358.txt (2008).

[2] V. Paxson, An analysis of using reflectors for distributed denial-of-service attacks, ACM SIGCOMM Computer Communication Review, Vol.31, No.3, pp.38-47 (2001).

[3] P. Ferguson, and D . Senie, Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, RFC 2827, BCP 38, https://www.ietf.org/rfc/rfc2827.txt (2000).

[4] F. Baker, and P. Savola, Ingress Filtering for Multihomed Networks, RFC 3704, BCP84, https://tools.ietf.org/rfc/rfc3704.txt (2004).

[5] P. Vixie, and V. Schryver, DNS Response Rate Limiting (DNS RRL), ISC-TN-2012-1-Draft1 (2012).

[6] T. Rozekrans, and J. Koning, Defending against DNS reflection amplification attacks, University of Amsterdam System & Network Engineering RP1 (2013).

[7] X. Ye, and Y. Ye, A Practical Mechanism to Counteract DNS Amplification DDoS Attacks, Journal of Computational Information Systems, Vol.9, No.1, pp.265-272 (2013).

[8] S. Paola, and D. Lombardo, Protecting against DNS Reflection Attacks with Bloom Filters, Proceedings of the 8th international conference on Detection of intrusions and malware, and vulnerability assessment (DIMVA'11), pp.1-16 (2011).