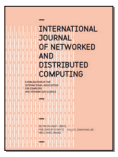
International Journal of Networked and Distributed Computing
Vol. **8**(**1**); December (2019), *pp.* 34–40
DOI: https://doi.org/10.2991/ijndc.k.191121.001; ISSN 2211-7938; eISSN 2211-7946
https://www.atlantis-press.com/journals/ijndc



Research Article

Proposal and Prototype of DNS Server Firewall with Flexible Response Control Mechanism

Hideo Masuda[1,*], Shun Segawa[2], Masayuki Mori[1]

[1]*Center for Information Science, Kyoto Institute of Technology, 1 Matsugasaki-Hashikami-cho, Sakyo-ku, Kyoto 606-8585, Japan*
[2]*Department of Information Science, Kyoto Institute of Technology, 1 Matsugasaki-Hashikami-cho, Sakyo-ku, Kyoto 606-8585, Japan*

ARTICLE INFO

Article History
Received 13 April 2019
Accepted 18 May 2019

Keywords
Internet security
Domain Name System (DNS)
Distributed Denial of Service (DDoS).

ABSTRACT

Domain Name System (DNS) is an important system for the Internet communication. DNS is a system for distributed management and operation of domain names, and it is possible to associate with the resources such as IP address, instruct the destination host of the e-mail, and so on. On the one hand, it is very serious problem that the damage caused by the service of the DNS server being stopped, and stable operation of the DNS server is essential for stable operation of the Internet. DNS servers may be illegally accessed to make it target or springboard server for attacks such as Distributed Denial of Service (DDos) attacks and DNS reflector attacks. In this paper, we show the analysis of the queries received by our university DNS server. In addition, we propose the method to suppress attacks to DNS servers by deploying the system to monitor access from DNS clients and adaptively manipulating responses of queries from attackers in front of the DNS server based on the analysis. Moreover, we developed the prototype system and evaluated performance of it.

1. INTRODUCTION

The use of the Internet is increasing due to the diversification of services by clouding and the spread of mobile devices. Domain Name System (DNS) [1,2] is an important system in Internet communication. Communication between all computers connected to the Internet is performed by unique IP address. For this reason, the mechanism devised to treat IP addresses with human-recognizable names is the Internet domain name. DNS is a system for distributed management and operation of domain names, and it is possible to associate with the IP address, instruct the destination host of the e-mail, and so on. Also, the role of DNS on the Internet is expected to grow more and more in the future by the development of new technologies in DNS such as E.164 NUmber Mapping [3] and Internationalized Domain Name [4].

On the one hand, the damage caused by the service of the DNS server being stopped is significant, and stable operation of the DNS server is essential for stable operation of the Internet. Servers connected to the Internet are always at risk of attack, and DNS servers are no exception. DNS attacks include Denial of Service (DoS) and Distributed DoS (DDoS) attacks [5] that increase the load on the DNS server so that responses cannot be obtained, and DNS amplifier attacks [6] that exploit the characteristic that the DNS server responds to queries in a reflective. In our laboratory, a method for finding IPv6 devices using DNS has been found [7]. These attacks can cause service outages and network paralysis. The DNS server operator needs to prevent such attacks while providing the service requested by the client.

*Corresponding author. Email: h-masuda@kit.ac.jp

Domain name system response rate limiting [8] proposed by Vixie and Schryver is a technology to prevent attacks on DNS servers. In this method, attacks on DNS servers are prevented by monitoring and limiting the response rate. Also, in our laboratory, an access source classification system [9] is proposed, which is a method of guiding illegal communication using DNS response. In this research, by changing the response to the DNS query, it is possible to distinguish between offensive communication and legitimate communication, and propose a method that can guide illegal communication. However, it is difficult to follow version upgrades such as security measures by adding new functions to existing DNS servers for implementation.

Therefore, in this paper, we analyze the queries received by the university DNS server. In addition, we propose the method to suppress attacks to DNS servers by introducing the system to monitor access from DNS clients and adaptively manipulating responses of queries from attackers in front of the DNS server without changing the existing DNS server based on the analysis. Moreover, we developed the prototype system and evaluated performance of it.

2. ATTACK ON DNS

2.1. DoS Attack and DDoS Attack

A DoS attack is an attack method that uses server software vulnerabilities to stop services, increase the load on DNS servers, make responses unobtainable, flood the communication path and prevent responses. In addition, there is also a DDoS attack type in

which a large number of machines simultaneously launch a DoS attack on one server.

2.2. DNS Amplifier

Domain name system amplifier attack is an attack method that uses DNS server as communication amplifier and exploits the characteristic that DNS server responds to queries in a reflective. DNS uses UDP, which is vulnerable to spoofing IP address. If DNS server receives a query spoofing the source IP address, the DNS server sends an attack packet because it returns a response to the spoofed IP address. In addition, DNS is characterized in that it can perform DoS and DDoS attacks more efficiently because the responses are often larger than queries.

2.3. Communication Analysis

In this chapter, we analyze the queries received by DNS server that is actually used on the Internet. In this analysis, we use DNS server with authority of kit.ac.jp zone and kit.jp zone in Kyoto Institute of Technology. The IPv4 address block corresponding to the zone is 133.16.0.0/16. Analysis is performed using the results of monitoring the communication of the DNS server. The analysis term was seven days from January 12, 2018 to January 18, 2018.

2.4. Analysis of Query Content

An overview of the observed DNS queries is shown in Table 1. The DNS server observed 1,593,411 queries. Among them, 1,283,277 queries were normal responses, and 310,134 queries were response errors. The result was that response errors were very high, but about half were queries from proxy servers on campus. It is considered to be caused by sending a query with cis.kit.ac.jp added at the end of the domain name if name resolution fails.

Among the observed queries, the normal response queries are regarded as a regular query, and the response error queries are analyzed in detail.

2.4.1. Queries to out of zone

One of the reasons that DNS server responses a "Response error" code is that DNS query indicates the resource of outside the authoritative zone. An example of these queries is shown in Table 2. These queries were roughly classified into two types of queries according to the search type.

analytics.ff.avast.com, dnsscan.shadowserver.org, cc595656.openresolverproject.org, etc. that are the domain name inquired by the query whose search type is A is the domain name under the control of the research organization that is investigating open resolvers on the Internet. *sttp.1f1f1085.wc.syssec.rub.de* is a domain name managed by Ruhr-University Bochum, which was also used for surveys on the Internet. About 25% of queries whose search type is A were queries for domain names under the control of research institutes. Thus, although the domain name inquired by the query is outside the zone and the query with search type A is a response error of the DNS server's response, it is considered that there are many queries that do not lead to incorrect communication. In this way, when the inquired domain name is outside the zone and the search type is A, the DNS server response resulted in a response error, but it is considered that many queries do not lead to illegal communication.

We used full-service resolver for name resolution for activum.nu and leth.cc as domain names whose search type is ANY. The result is shown in Table 3. The amplification rate is the packet size of the response to the packet size of the query. The amplification rate of general domain names whose search type is ANY are shown in Table 4. General domain names is the top 10 domain names of popular site rankings published in Alexa [10]. Since the average amplification rate of general domain names is about 642%, it can

Table 1 | Observed DNS queries

All queries	1,593,411
Normal responses	1,283,277
Responses error	310,134
Unique source IP address	33,313

Table 2 | Queries to out of zone

Search types	Domain names
A	c.afekv.com analytics.ff.avast.com dnsscan.shadowserver.org cc595656.openresolverproject.org sttp.1f1f1085.wc.syssec.rub.de
ANY	activum.nu leth.cc svist21.cz isc.org (.)

Table 3 | Amplification rate of queries whose search type is ANY

Domain names	Amplification rates (%)
activum.nu	8207
leth.cc	15564
svist21.cz	8922
isc.org	4387
(.)	3348

Table 4 | Amplification rate of general domain names

Domain names	Amplification rates (%)
google.com	812
youtube.com	909
facebook.com	176
baidu.com	400
wikipedia.org	675
reddit.com	104
yahoo.com	400
google.co.in	758
qq.com	266
amazon.com	981
average	642

be seen that the amplification rate of domain names whose search type is ANY is very large. From these facts, it is considered that these domain names are easily used for DNS amplifier attacks, and are queries that lead to illegal communication. About 99% of queries whose search type is ANY were queries for domain names that greatly increase the amplification rate.

In addition, there is a need to analyze further about queries whose search type is A but not a query for a domain name under the control of research institutes, or queries whose search type is ANY but whose amplification rate is not different from general ones.

2.4.2. Queries to zone

One of the reasons that DNS server responses a "Response error" code is that DNS query indicates the non-existence resources. An example of these queries is shown in Table 5. These queries include forward DNS queries for domain names in the zone such as smtp.mail.edu.kit.ac.jp and resmail.cis.kit.ac.jp but not defined, and reverse DNS queries for IP addresses in the zone such as xxx.yyy.16.133.in-addr.arpa but not associated with a domain name.

Forward DNS queries that are for domain names in the zone and are not defined, such as imap.kit.ac.jp and smtp.mail.edu.kit.ac.jp can be queries mail software and others used to find mail server corresponding to kit.ac.jp or kit.jp. Also, such queries may be for domain names that has been used in the past but is not currently used. Therefore, it is considered that these queries do not lead to illegal communication.

Reverse DNS queries that are for IP addresses in the zone such as xxx.yyy.16.133.in-addr.arpa. and are not associated with domain names may have scan preparation for unregistered IP addresses. However, not all IP addresses do not perform domain name registration in the IPv4 address block (133.16.0.0/16) managed by Kyoto Institute of Technology, so it is conceivable that this queries are for access sources survey (connection restriction and log recording) from existing IP addresses. Application methods may be considered in which automatic extraction of unregistered addresses or automatic generated response in cooperation with the IP address usage status survey system on campus.

2.4.3. Domain name "version.bind"

Queries whose response is not error and the end of the domain name is not kit.ac.jp or kit.jp include queries whose domain name is version.bind. The DNS server sends version information of the DNS server when it receives a query whose domain name is version.bind, search type is TXT and network class is CH. Such queries may be used by attackers to find vulnerable DNS servers by verifying DNS server versions.

Table 5 | Queries to zone

Search type	Domain names
Forward DNS lookup	imap.kit.ac.jp smtp.mail.edu.kit.ac.jp resmail.cis.kit.ac.jp ipc.kit.ac.jp www.ad-global.kit.ac.jp
Reverse DNS lookup	xxx.yyy.16.133.in-addr.arpa

2.5. Number of Queries

We checked the queries per second (qps) of the query received by the DNS server. The result of qps about the query received by the DNS server is shown in Table 6. The largest qps was 528 qps on January 13th.

3. CONSIDERATION

This chapter considers methods to prevent attacks on DNS servers and attacks using DNS in order to realize the security of DNS.

3.1. Improve the DNS Protocol

Since the DNS protocol is widely used, it is not realistic to realize the security of DNS by improving the DNS protocol.

3.2. Add Functions on Existing DNS Servers

It is difficult to follow security upgrades by adding new functions that prevents DNS attacks to existing DNS servers such as BIND [11] and NSD [12]. Therefore, it is necessary to avoid modifying the existing DNS server.

3.3. DNS Firewall

Consider creating the firewall for DNS. If an attack can be detected by the firewall in front of the DNS server, the DNS server is protected by adaptive control such as changing the response timing, thinning out, or rewriting a part of the payload and the firewall prevent the DNS server to attack. It is possible to maintain the proper action of DNS by passing the regular communication.

4. PROPOSED SYSTEM

This chapter explains a method to suppress attacks on DNS by placing the system to monitor the communication from the DNS client in front of the DNS server as shown in Figure 1 and controlling the response adaptively for the attacks based on the consideration in Chapter 4.

Table 6 | Queries per second (qps) of the query received by the DNS server

Date	Maximum qps
Jan 12	204
Jan 13	528
Jan 14	82
Jan 15	208
Jan 16	102
Jan 17	356
Jan 18	92

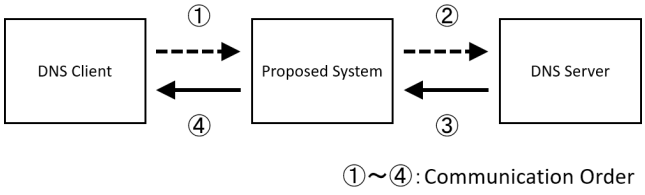


Figure 1 | Figure of proposed system.

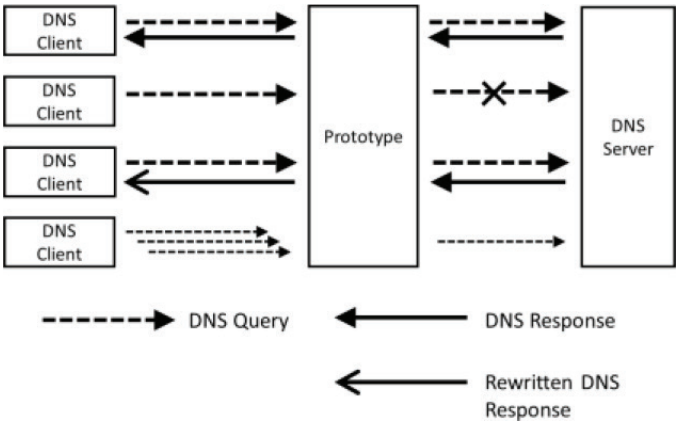


Figure 2 | Figure of prototype.

4.1. DNS Server Attack Countermeasures

The proposed system relays, discards, and rewrites packets without improving the DNS protocol. If queries from DNS clients can be monitored and analyzed in this system, the aggression of queries be detected and responses be adaptively controlled to protect the DNS server based on the results. This leads to the prevention of attacks such as DDoS attacks and DNS amplifier attacks.

4.2. System Independent of DNS Server

Since the system protects the DNS server, the security of DNS can be improved without modifying the existing DNS server.

5. PROTOTYPE

This chapter explains the prototype of the system described in Chapter 5.

5.1. Control Function

We used Python, which can use Scapy [13] as a library to implement the prototype. This enabled us to implement the process of sending and receiving DNS packets with about 400 lines of source code. In this program, DNS query is received with pcap [14] which is an API for packet capture. The outline of the prototype is shown in Figure 2. The prototype controls responses as follows:

1. Relay the communication between DNS client and DNS server without change.
2. Discard the query from DNS client without sending it to DNS server.
3. Rewrite the response from DNS server.
4. Thinning out queries from DNS client to DNS server.

Additionally, the prototype records timestamp, source IP address, destination IP address, ID, QNAME, QTYPE, and QCLASS as access logs.

5.2. Examples of Use

An example of using the control function of the prototype is explained. Relay communication between DNS client and DNS server for regular queries from DNS clients. For queries that lead to illegal communication, it is possible to protect the DNS server by discarding the query from the DNS client without sending it to the DNS server. Also, when a large number of queries are sent from the same IP address, it may be possible to protect the DNS server by limiting the number of queries from that DNS client to the DNS server, delaying the response, etc.

6. PERFORMANCE EVALUATION

Measure qps as processing performance by sending queries with resperf [15] that is a DNS stress tool when the prototype is placed in front of the DNS server and it is not placed. The prototype is evaluated by comparing these results. Furthermore, we compare the latency when the prototype is placed in front of the DNS server and it is not placed. We referenced the value of "Query time" by dig command to investigate latency.

6.1. Without Prototype

We measure the processing performance when the prototype is not placed in front of the DNS server. We prepared virtual machines on the virtualization platform operated in the laboratory as a DNS server. The specification of the DNS server is shown in Figure 3. We used NSD as a DNS server software. The domain in the domain name space is not delegated and the zone data

CPU	Dual-Core AMD Opteron(tm) Processor 8222, 3 GHz x4
Memory	128 GB
Network	1000baseT x4
Host OS	CentOS release 6.8 (Final)
Type of VM	KVM
vCPU on VM	QEMU Virtual CPU (cpu64-rhel6)
OS on VM	Ubuntu 16.04.5 LTS

Virtual machine hosts for experiment	
DNS server	NSD version 4.1.7 8 GB mem
DB server	MariaDB 10.0 1 GB mem
Load balancer	ipvsadm v1.28 1 GB mem

Figure 3 | Experiment servers specification.

file describes information for fictitious zones. The DNS server responses packets that include a response part, an authoritative part, and an additional information part for the queries with domain name of www.example.com.

The resperf measurement result is 44,438 qps when the prototype is not placed in front of the DNS server.

As a result of measuring ten times, the average of latency is 3 ms.

6.2. With Prototype

We measure the processing performance when the prototype is placed in front of the DNS server. We prepared virtual machines on the virtualization platform operated in the laboratory as the prototype. The specifications of the prototype is shown in Table 7. Also, access logs are stored in the database server. The specifications of the database server is shown in Table 3. We used MariaDB as a database software.

6.2.1. Experiment 1

The resperf measurement result is 26 qps when the prototype placed in front of the DNS server. It is clear that the processing performance as a DNS server is significantly degraded by the prototype in comparison with the results of Section 7.1. In addition, it can be seen that the prototype system cannot withstand the actual query on the Internet in comparison with the results of Section 7.1. As a result of measuring ten times, the average of latency is 58.8 ms. This shows that the processing performance as a DNS server is greatly degraded by the prototype. We examined the bottlenecks of the prototype by using cProfile [16], which is a Python profiler. The cause is to handle pcap.

6.2.2. Experiment 2

The processing performance is improved by increasing the number of processes in the prototype system so that it can be received by multiple ports and the load balancer distributing queries to each port. The specifications of the load balancer is shown in Table 3. We used ipvsadm as a load balancer software. The figure of Experiment 2 is shown in Figure 4. The maximum throughput by resperf is 400 qps.

The results of processing performance with increased the number of processes are shown in the Figure 5. Processing performance was improved by increasing the number of processes up to about 24 or 25, but after that it was not improved even if the number of processes is increased. From this result, it was found that the processing performance is improved to some extent by increasing the number of processes, but it did not reached the processing performance required for the DNS server by just increasing the number of processes.

As a result of measuring ten times, the average of latency is 61.8 ms. This shows that the difference between Experiment 1 and 2 in latency is small and there is little delay by the load balancer.

Table 7 | Prototype system specification

CPU	Intel(R) Xeon(R) CPU X5650 @ 2.67GHz x6
Memory	96 GB
Network	1000baseT x2
Host OS	CentOS Linux 7 (Core)
Type of VM	KVM
vCPU on VM	Intel Nehalem Class Core i7 x8
Memory on VM	8 GB
OS on VM	Ubuntu 16.04.5 LTS
Software	Python 2.7.12

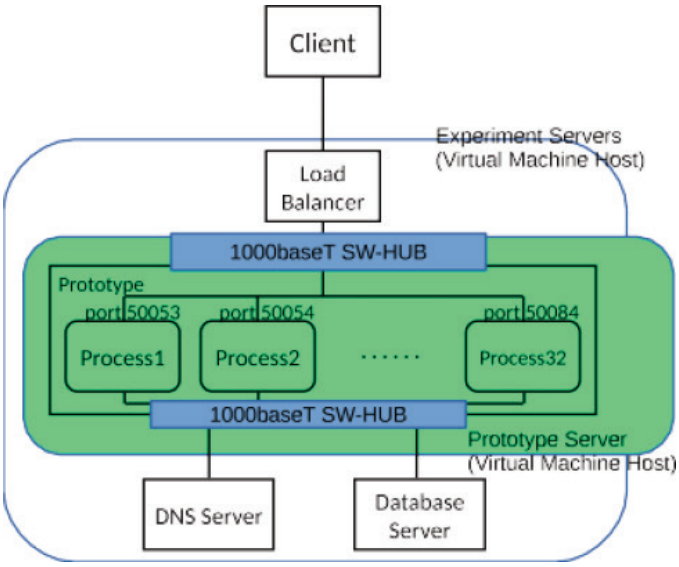


Figure 4 | Figure of Experiment 2.

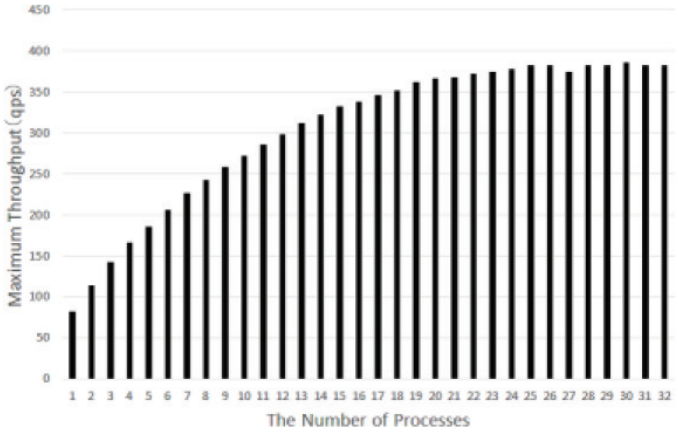


Figure 5 | Result of Experiment 2.

6.2.3. Experiment 3

The processing performance is improved by increasing the number of prototypes. The Figure of Experiment 3 is shown in Figure 6. From the results of Experiment 2, the number of processes on each machine is 24.

The results of processing performance with increased the number of prototypes are shown in the Figure 7. From this result, it was found that when the number of prototypes is 2, the processing performance exceeds the result of the Section 3.2.

As a result of measuring ten times, the average of latency is 85.8 ms.

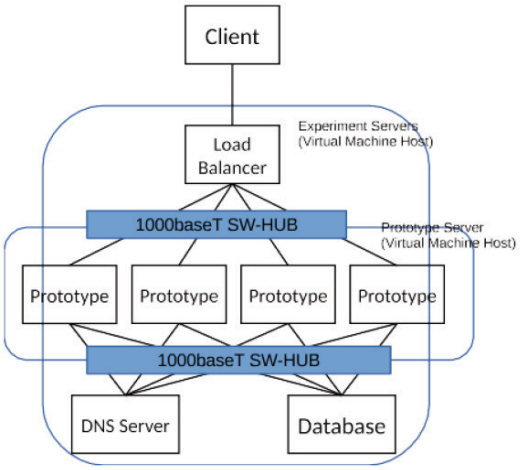


Figure 6 | Figure of Experiment 3.

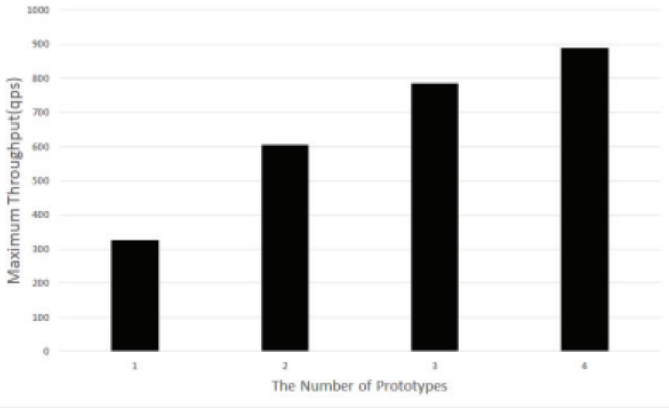


Figure 7 | Result of Experiment 3.

7. DISCUSSION

This chapter examines the results of Chapters 3 and 7.

7.1. How to Use the Proposed System

From the analysis result of Chapter 7, it is confirmed that many domain names inquired by the queries whose search type is ANY and whose response code is error are easily exploited for DNS amplifier attack. From this, it is conceivable as a usage of the proposed system to discard and thin out queries whose increase amplification rate is large such as queries whose search type is ANY.

Queries to confirm the version of DNS server may be used to search for vulnerable DNS servers by attackers. From this, it is conceivable as a usage of the proposed system to block the search for DNS servers by discarding queries to confirm the version of DNS servers.

7.2. About Prototype System

From the results in Sections 7.1 and 7.2.1, when the prototype place in front of the DNS server, the throughput is 1/1700 of throughput without the prototype and the response increased by about 56 ms. The results of Section 3.2 indicate that the processing performance of the prototype cannot withstand the requests observed by our university's DNS server of the kit.ac.jp zone and the kit.jp zone.

It was found that the processing performance is insufficient when the prototype is a single unit. We increase the processing performance by distributing the load to multiple instances in Experiment 2 and 3. These experiments indicate that the processing performance of multiple instances can respond to the requests observed by our university's DNS server. However, to compare with the prototype and without the prototype shows the processing performance has dropped significantly. Furthermore, it is expected that the processing performance is further decreased if the prototype system is equipped with functions to analyze the access log and judge the adaptive control method.

8. CONCLUSION

In this research, we analyzed the queries received by our university DNS server in order to consider how to realize the security of DNS server. As a result, the queries considered to lead to the attack was confirmed.

In addition, we propose the method to suppress attacks to DNS servers by deploying the system to monitor access from DNS clients and adaptively manipulating responses of queries from attackers in front of the DNS server based on the analysis. Moreover, we developed the prototype system and evaluated performance of it. Performance evaluation showed that when the prototype is placing in front of the DNS server, the processing performance decrease. However, it was found that the processing performance can respond to the requests observed by our university's DNS server by distributing the load to multiple instances.

Future topics of study include continuing analysis of queries to DNS servers, examining methods of judging queries that lead to illegal communication and regular queries, and implementing the judgment method in the prototype.

CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest.

REFERENCES

[1] P. Mockapetris, Domain names - concepts and facilitie, (RFC1034, 1987) Available from: https://tools.ietf.org/rfc/rfc1034.txt
[2] P. Mockapetris, Domain names - implementation and specification, (RFC1035, 1987) Available from: https://tools.ietf.org/rfc/rfc1035.txt
[3] P. Faltstrom, E.164 number and DNS, (RFC2916, 2000) Available from: https://tools.ietf.org/rfc/rfc2916.txt
[4] J. Klensin, Internationalized domain names for applications (IDNA): definitions and document framework, (RFC5890, 2010) Available from: https://tools.ietf.org/rfc/rfc5890.txt

[5] M. Handley and E. Rescorla, Internet denial-of-service considerations, (RFC4732, 2006) Available from: https://tools.ietf.org/rfc/rfc4732.txt

[6] J. Damas and F. Neves, Preventing use of recursive nameservers in reflector attacks, (RFC5358, 2008) Available from: https://tools.ietf.org/rfc/rfc5358.txt

[7] K. Soga, H. Masuda, A consideration of finding IPv6 devices by DNS resolver under IPv4/IPv6 dual stack environment and resolving methods, (Information Processing Society of Japan (IPSJ), Internet and Operation Technology (IOT) Symposium 2013 WIP, 2013), pp. 107–110.

[8] P. Vixie, V. Schryver, DNS Response Rate Limiting (DNS RRL), in: ISC-TN-2012-1-Draft1, 2012, pp. 1–5. Available from: http://ss.vix.su/~vixie/isc-tn2012-1.txt (accessed Oct 2017).

[9] T. Ogawa, H. Masuda, Proposal of a source host labeling method using customized DNS response for server protection in IPv6 environment, Master Thesis, Department of Information Science, Kyoto Institute of Technology, Kyoto, Japan, 2016, pp. 1–30.

[10] Alexa Internet, The top 500 sites on the web, Available from: https://www.alexa.com/topsites (accessed Feb 2019).

[11] Internet Systems Consortium, BIND 9 - Versatile, classic, complete name server software, Available from: https://www.isc.org/bind/

[12] NLnet Labs, nsd - Name Server Daemon, Available from: https://www.nlnetlabs.nl/projects/nsd/

[13] P. Biondi, Scapy community, Scapy - Packet crafting for Python2 and Python3, Available from: https://scapy.net/

[14] L. MartinGarcia, tcpdump/libpcap public repository, Available from: https://www.tcpdump.org/

[15] DNS-OARC, dnsperf - gather accurate latency and throughput metrics for DNS, Available from: https://www.dns-oarc.net/tools/dnsperf/

[16] Python Software Foundation, cProfile - the python profilers, Available from: https://do cs.python.org/2/library/profile.html