ISBN: 978-981-18-3959-7
2022 the 12th International Workshop on Computer Science and Engineering (WCSE 2022)
doi: 10.18178/wcse.2022.06.047

Policy-based Detection and Blocking System for Abnormal Direct Outbound DNS Queries using RPZ

Hikaru Ichise [1 +], Yong Jin [1] and Katsuyoshi Iida [2]

[1] Tokyo Institute of Technology, Tokyo, Japan
[2] Hokkaido University, Sapporo, Japan

Abstract. Bot-infected computers sending direct outbound DNS queries without obtaining the information of authoritative DNS servers from the DNS full resolvers set up in the internal network have become a critical security issue nowadays. In DNS protocol, the domain name resolution process obtains the information of necessary authoritative DNS name servers (NS records) at the beginning and then achieves the answers of the original DNS queries which is accomplished via the DNS full-service resolvers. However, some types of bot programs violate the DNS protocol process and send the direct outbound DNS queries to its Command and Control (C&C) servers (malicious DNS servers) for bot communication. We have investigated the detection and blocking the direct outbound DNS queries by using MySQL at an early stage. However, the network latency was arising as a critical issue. In this advanced research, we propose a policy-based detection and blocking system for abnormal direct outbound DNS queries using DNS Response Policy Zones (DNS RPZ) in order to solve the issues. In this paper, we describe the design of the proposed system and introduce an implemented prototype system. In addition, we also describe the preliminary evaluation results per feature of the proposed system conducted on the prototype, and finally, we introduce the tasks planed for future work.

Keywords: Botnet, abnormal DNS traffic, DNS, NS record, RPZ, SDN, and direct outbound DNS query.

1. Introduction

Botnet is one of the critical threats in cyber security nowadays [1]. Once a computer is infected by some types of bot programs, the bot-infected computer searches its Command and Control (C&C) servers first. Then the C&C servers will send instructions to the bot-infected computer for further actions such as cyber-attacks including Advanced Persistent Threat (APT), Distributed Denial of Service (DDoS), ransomware attacks, etc. Consequently, a huge number of bot-infected computers execute cyber-attacks based on the instructions of the C&C servers [2]. Therefore, the detection and blocking botnet communication between bot-infected computers and the C&C servers become very important.

Recent reports have indicated that the Internet communication protocols such as Internet Relay Chat (IRC), HTTP, and Peer-to-Peer (P2P) had been used in botnet communication [3]. Moreover, some researchers have shown that some botnet communication used DNS protocol [4,5]. As being well-known, the DNS protocol is mainly used for domain name resolution which translates the hostnames to IP addresses in the Internet and vice versus. However, with the increase of Internet services, some minor DNS resource records like TXT records [6], [7] also become widely used nowadays. According to [6], [7] on the use of DNS TXT resource records, some types of bot programs have been identified as using DNS TXT resource records for botnet communication. In [8],[9], the authors proposed a machine-learning-based detection method of botnet communications using DNS over HTTPS protocol, which is a privacy enhancement of DNS. As a result, the DNS traffic which so far has been considered secure also has become a target of being monitored communication since the network administrators cannot simply block all DNS traffic. Thus it becomes necessary to detect and block DNS-based botnet communication from bot-infected computers at the early stage.

+ Corresponding author. Tel.: + 81-3-5734-3424; fax: +81-3-5734-2491.
E-mail address: hichise@nap.gsic.titech.ac.jp.

In most organization networks, one or more DNS full resolvers will be set up for providing the domain name resolution service to the internal computers. Firstly, the DNS full resolver obtains information of the authoritative name server (NS record and the IP addresses of the NS record, which is called glue A record) at the beginning of the domain name resolution process. However, in DNS-based botnet communication, a bot-infected computer sends DNS queries to the C&C servers directly without the processes of obtaining the NS records and glue A records. Note that the bot-infected computers are likely to communicate with the C&C servers by using DNS full resolvers in the organization network but network administrators may detect botnet communication by monitoring DNS traffic or DNS full resolvers. Moreover, there is an exception in using the direct outbound DNS queries which can be legitimate such as using the public DNS servers like Public DNS operated by Google [10], or Cloudflare [11]. In the exceptional procedure, it is possible to communicate between internal computers and the Public DNS servers by using the direct outbound DNS queries.

Based on the above observation, we have ever considered and investigated the detection and blocking system for the botnet communication based on monitoring the direct outbound DNS queries by using NS record history database, which store NS records and the corresponding glue A records and verifies every direct outbound DNS query [12],[13]. Particularly, we created a database that stores the NS records and the corresponding glue A records in order to differentiate the normal and abnormal direct outbound DNS queries. Next, the destination IP address of a DNS query in a normal DNS name resolution will be stored in the database and will be passed through, otherwise, the DNS query will be blocked because the destination IP address of the DNS query is not included in the database which indicates that the direct outbound DNS query is abnormal. However, we found an issue in the use of a database in the system which may cause long network latency during the domain name resolution due to the verification for every direct outbound DNS query by accessing the database. Reducing the network latency must be accomplished to satisfy the quality demand of users in the organization. In this paper, we proposed a policy-based detection and blocking system for abnormal direct outbound DNS queries by using DNS Response Policy Zones [14] (DNS RPZ) instead of the database in order to solve the existing problem in the previous system. We implemented a prototype system and confirmed that the proposed features worked as expected in the preliminary evaluations. Then we plan to deploy the proposed system in a real network environment based on the performance evaluation.

2. Proposed System

In this section, we describe the design and the architecture of the proposed system that detects and blocks the DNS traffic from bot-infected computers using direct outbound DNS queries in an organization network.

2.1. Design

The proposed system is based on three observations.

(1) All the destination IP addresses of direct outbound DNS queries should be obtained by legitimate DNS-based domain name resolution including the NS records and the glue A records in an organization network.

(2) A direct outbound DNS query in which the destination IP address is not obtained by legitimate DNS-based domain name resolution should be considered abnormal.

(3) Some exceptions need to be considered such as the use of public DNS servers.

In an organization network, one or more DNS full resolvers will be set up for providing domain name resolution service to the internal computers. The DNS full resolvers provide recursive domain name resolution service to the computers in the organization network by querying the corresponding authoritative DNS servers on the Internet. In the proposed system, all the DNS traffic, sent out to and received from the Internet, will be captured and analyzed. Then all the DNS NS records and glue A records included in the authoritative answers and authoritative sections will be extracted and stored in the DNS RPZ. A DNS RPZ is a special authoritative zone in a DNS server for successive communication to a special application server based on the policy. By practically using DNS RPZ in the proposed system, a direct outbound DNS query can be effectively controlled based on the policies.

Figures 1 illustrate the brief network architecture of an organization with the proposed system. We assume that in the organization network, some internal computers use the DNS full resolver for the domain name resolution and some directly send out DNS queries to the public DNS servers. The DNS traffic analyzer of the proposed system achieves all the DNS traffic from the border router of the organization network and stores the NS records as well as the glue A records to the DNS RPZ. Then all the direct outbound DNS queries will be monitored in the organization network and the destination IP address will be verified on the DNS RPZ. If the destination IP addresses are stored in the DNS RPZ the DNS query will be passed through, otherwise, the DNS query will be blocked.

In the previous system [12],[13], we used MySQL database to store the DNS NS records and the glue A records. Accordingly, the destination IP address of each direct outbound DNS query needs to be verified by accessing the MySQL database using TCP connection which may cause unnecessary overhead. Instead, by using the DNS RPZ in the proposed system, one pair of UDP packets can accomplish the verification for every direct outbound DNS query and it is expectable to reduce the unnecessary overhead.

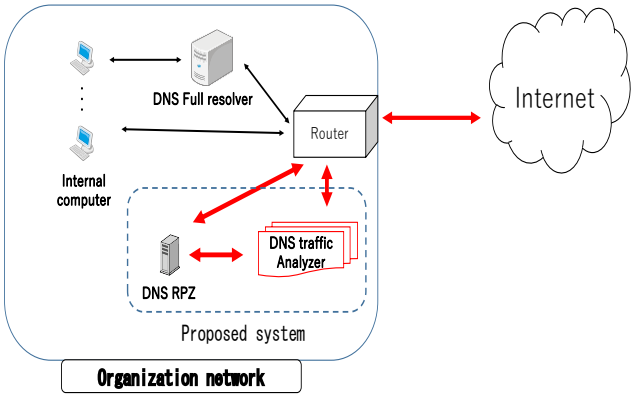

Fig. 1: Network architecture with the proposed system.

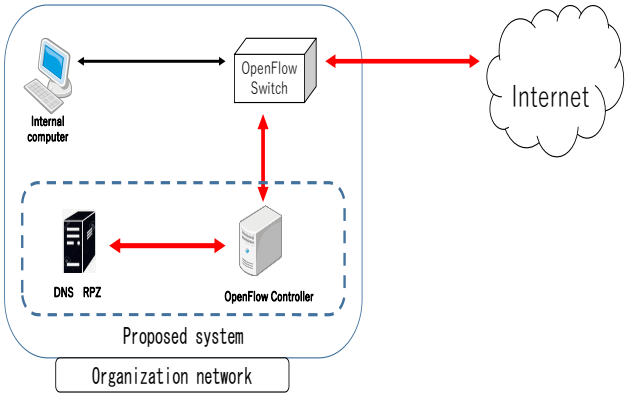

Fig. 2: Network architecture of the prototype system

2.2. System Architecture

In the proposed system, we use the SDN technology [15] to control the network traffic since the separation of the control plane and data plane makes it easier to apply the policies in the proposed system. The following terminologies will be used in the proposed system architecture.

- Query Fully Qualified Domain Name (Query FQDN): The hostname or domain name that the internal computer's query for domain name resolution.
- Query destination IP Address: The destination IP address to which the internal computer sends the DNS queries.
- DNS RPZ: For storing the legitimate DNS NS and glue A records including legitimate public DNS servers.
- OpenFlow Switch: Used in the SDN network for forwarding the packets based on the flow tables constructed under the instructions of the OpenFlow controller.
- OpenFlow Controller: Used in the SDN network to control network traffic by instructing OpenFlow switch construct flow tables.

In an SDN-based network, all packets will be sent to the OpenFlow switch and the OpenFlow controller determines whether pass them through or not. In our proposed method, the destination IP address of a direct outbound DNS query packet will be checked by the OpenFlow controller on the DNS RPZ. Figure 2 shows a simple system architecture of the proposed system within an SDN-based network. We briefly describe the procedure of the proposed system using an example in which an internal computer sends a direct outbound DNS query to the Internet in the following.

1) An internal computer sends a direct outbound DNS query to the Internet and the packet will be transferred to the OpenFlow switch.

2) The OpenFlow switch receives the DNS query and forwards the packet to the OpenFlow controller since the information about the destination IP address of the DNS query packet is not in the flow table.
3) When the OpenFlow controller receives the DNS query packet, it analyzes the packet to obtain the query name and the destination IP address. The OpenFlow controller then checks the destination IP address of the DNS query packet in the DNS RPZ.
4) The DNS RPZ replies to the corresponding entry (can be multiple entries) if the destination IP address is in the DNS RPZ.
5) The OpenFlow controller decides whether or not to pass through the DNS query packet based on the result and sends the decision to the OpenFlow switch. If the destination IP address of the DNS query packet is in the DNS RPZ, then the OpenFlow controller passes it through as a normal DNS query, otherwise, the OpenFlow controller will block the DNS query packet.
6) Based on the instructions of the OpenFlow controller, the OpenFlow switch conducts the corresponding actions, passes through the DNS query packet, or blocks it.

With the above procedures, the direct outbound DNS queries to the IP addresses achieved from the legitimate DNS-based domain name resolution in an organization network will be passed through as a normal DNS query and others will be detected and blocked as abnormal DNS traffic.

3. Implementation and Evaluation

Based on the proposed system, we implemented a prototype including two new features: one is for extracting authoritative NS records and glue A records and storing them in the DNS RPZ, the other is OpenFlow controller feature to control the DNS traffic. Then we conducted preliminary evaluations and checked the proposed features using the prototype system.

3.1. Implementation

We implemented the DNS traffic analyzer feature using the Python programing language with including the DPKT module in order to process captured PCAP network traffic files. Then we used Berkeley Internet Name Domain (BIND) to construct the DNS RPZ since the RPZ is one of the features of the BIND. In addition, we used Ryu and Open vSwitch as the OpenFlow controller and OpenFlow switch to construct the SDN-based network environment.

In order to evaluate the prototype system, we have set up a local experimental network environment consisting of an Open vSwitch, an OpenFlow controller named Ryu, and a client PC. The Open vSwitch was installed on a physical machine and the Ryu and the client were installed as KVMs in the host machine. Moreover, the Open vSwitch and the external switch, which was connected to the physical machine, were set all packet forwarding in different network segments in order to send the DNS query to Public DNS servers. By this setting, the Open vSwitch handled all packets. Particularly, Ryu program created the Python language monitored the destination IP address of all DNS traffics. Table 1 shows the detailed version information for the operating system, SDN solutions, and RPZ.

Table 1: Software versions used in the implementation

Operating System, Software and RPZ	Version
Host OS, guest OS in KVM	7.9
Open vSwitch	2.13
RPZ	4.34
Python	3.6

Table 2: The results of feature evaluation

Command	Behavior of OpenFlow switch
dig @8.8.4.4 www.google.com	Pass
dig @8.8.8.8 www.google.com	Blocked
dig @1.1.1.1 www.google.com	Blocked

3.2. Evaluation

To show the basic functionality of the proposed system, we evaluated the proposed features using the prototype system on a local SDN-based experiment network. We evaluated the proposed features using the prototype system on a local SDN-based experiment network. First, we constructed the DNS RPZ using PCAP files achieved from our campus network, which were captured for about three months from November 01, 2014. A PCAP file containing the DNS traffic with the size of 200MB needed about 40 minutes to be

analyzed to extract the authoritative NS records and glue A records. After constructing the DNS RPZ in the experimental network environment, we evaluated the policy-based detection and blocking features for the direct outbound DNS queries using the DNS RPZ. Specifically, we attempted to send a DNS query from the internal computer and checked the results of the Ryu controller and OpenFlow switch as shown in Table 2. We stored the IP address "8.8.4.4" which is one of the Google Public DNS server IP addresses as legitimate in the DNS RPZ while "8.8.8.8" and "1.1.1.1" which is another IP address of the Google Public DNS server and one of the Public Cloudflare DNS server IP addresses as malicious. When the Open vSwitch receives the DNS query packet for the first time it will be forwarded to the Ryu controller via the "packet in" process since there are no entries for the DNS query packet from the client in the flow table. Then the Ryu program monitors the DNS query packet and checks its destination IP address with the glue A record stored in the RPZ. We confirmed that the Ryu controller correctly made the decisions for the direct outbound DNS queries based on the DNS RPZ. That is if the destination IP address of a direct outbound DNS query has been registered in the DNS RPZ, the Ryu controller added a flow entry into the OpenFlow switch with passing through the packet, otherwise, a flow entry with blocking the packet has been added. We also confirmed that the OpenFlow switch worked correctly based on the flow entries.

Table 3: The results of query time using dig in latency evaluation

Command	DNS resolver	Conventional		Previous system using MySQL		Proposed system using DNS RPZ	
		Average [ms]	Standard Deviation	Average [ms]	S.D	Average [ms]	S.D
dig @172.16.100.13 www.google.com (without cache)	Internet	1,626.9	121.6	3,409.6	381.6	2,406.4	419.2
dig @172.16.100.13 www.google.com (with cache)	Internet	0	0	0	0	0	0
dig @8.8.8.8 www.google.com	8.8.8.8	6.2	0.4	8.5	2.4	8.2	2.6
dig @1.1.1.1 www.google.com	1.1.1.1	6.5	0.5	8.1	2.5	7.3	1.5

Then we conducted the latency evaluation in the same SDN-based network environment by using "dig" command. Before the latency evaluation, we stored the destination IP addresses of the public DNS servers "8.8.8.8" and "1.1.1.1" to the DNS RPZ as legitimate in order to compare the latency of conventional domain name resolution. Furthermore, we also set up the previous system [12], [13] which used MySQL database in order to compare the latency with the proposed system using the DNS RPZ. As shown in Table 3, in the case of using the conventional DNS full resolver (172.16.100.13), the average latency for ten times of domain name resolution was 3,409.6ms in the previous system used MySQL database while that in the proposed system using the DNS RPZ was 2,406.4ms. The results confirmed that the overhead in the DNS RPZ is less than that in the MySQL database. When we used the cache in the DNS full resolver, we can see that all the latencies were "0" since all the DNS responses were sent back from the cache of the DNS full resolver. Moreover, we also measured the latency when using the public DNS servers. As shown in the result, the latency was a little much smaller than that of the DNS full resolver. We consider that the Public DNS servers have cache so that they can provide high performance than the internal DNS full resolver without using cache.

4. Conclusion

In this paper, we proposed a policy-based detection and blocking system for abnormal direct outbound DNS queries by using DNS RPZ. We implemented a prototype for the proposed system and constructed a local SDN-based network environment for the feature evaluations. Based on the preliminary evaluation

results, we confirmed that the proposed system worked correctly as we designed, and it also could obtain higher performance compared to the previous system that used MySQL database.

This paper only covers the preliminary evaluation per function of the proposed system. Therefore, we plan the include the following tasks in future work. First, a complete feature evaluation through the entire proposed system is necessary. Next, we plan to deploy the DNS traffic capture and analysis feature in a real network environment. Finally, we plan to realize the collaboration of the DNS traffic capture analysis feature and the policy-based detection and blocking feature in the real network environment. Furthermore, an approach to protect the user privacy during the DNS traffic analysis will be considered after the above task have been realized.

5. References

[1] Avast Threat Labs, https://decoded.avast.io/martinchlumecky/dirtymoe-1/. Last accessed 12 Nov 2021.

[2] Sood, A.K., Zeadally, S. :A taxonomy of domain-generation algorithms, IEEE Security & Privacy, vol. 14, no. 4, pp. 46–53, July-Aug. 2016. DOI: 10.1109/MSP.2016.76

[3] OpenDNS inc.: The role of DNS in botnet command and control (C&C), online http://info.opendns.com/rs/opendns/images/WB-Security-Talk-Role-Of-DNS-Slides.pdf, 2021.

[4] Feily, M. et al. :A survey of botnet and botnet detection, Proc. IEEE Int'l Conf. Emerging Security Information, Systems and Technologies, Athens, Glyfada, June 2009, pp. 268–273. DOI:10.1109/SECURWARE.2009.48

[5] Hands, N.M., Yang, B., Hansen, R.A. :A study on botnets utilizing DNS. In Proceeding of ACM Conference on Research in Information Technology (RIIT'15), pp. 23–28, Chicago, IL, USA, (2015). DOI:10.1145/2808062.2808070

[6] Ichise, H., Jin, Y., Iida, K. :Analysis of via-resolver DNS TXT queries and detection possibility of botnet communications. In Proceeding of IEEE Pacific Rim Conf. Communications, Computers and Signal Processing (PACRIM). pp. 216–221 (2015) DOI: 10.1109/PACRIM.2015.7334837

[7] Ichise, H., Jin, Y., Iida, K.: Analysis of DNS TXT Record Usage and Consideration of Botnet Communication Detection. IEICE Transactions on Communications E101(1), 70–79 (2018). https://doi.org/10.1587/transcom.2017ITP0009

[8] MontazeriShatoori, M., Davidson, L., Kaur, G., Habibi Lashkari, A.: Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic. In: Proceedings of 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech). pp. 63–70 (2020)

[9] Mitsuhashi, R., Satoh, A., Jin Y., Iida, K., Shinagawa, T., Takai Y.:Identifying Malicious DNS Tunnel Tools from DoH Traffic Using Hierarchical Machine Learning Classification. In Proceeding of International Conference on Information Security (ISC). pp. 238-256 (2021) DOI: 10.1007/978-3-030-91356-4_13

[10] Google, :Introduction to Google public DNS, https://developers.google.com/speed/public-dns/docs/intro. Last accessed 12 Nov 2021

[11] Cloudflare, :Introduction to Cloudflare Public DNS, https://www.cloudflare.com/learning/dns/what-is-1.1.1.1/. Last accessed 12 Nov 2021

[12] Ichise, H., Jin, Y., Iida, K., Takai, Y. :Detection and Blocking of Anomaly DNS Traffic by Analyzing Achieved NS Record history. In Proceeding Asia-Pasific Signal and Information Processing Association, Annual Summit and Conference 2018. pp. 1586–1590. (2018) DOI: 10.23919/APSIPA.2018.8659739

[13] Ichise, H., Jin, Y., Iida, K., Takai, Y.: NS record History Based Abnormal DNS traffic Detection Considering Adaptive Botnet Communication Blocking. IPSJ Journal of Information Processing 28, 112–122 (2020).https://doi.org/10.2197/ipsjjip.28.112

[14] Internet Systems Consortium, https://www.isc.org/rpz/. Last accessed 12 Nov 2021

[15] Open Networking Foundation, SDN Architecture, https://opennetworking.org/wp-content/uploads/2014/11/TR SDN-ARCH-1.0-Overview-12012016.04.pdf. Last accessed 12 Nov 2021.