ISBN: 978-981-18-3959-7
2022 the 12th International Workshop on Computer Science and Engineering (WCSE 2022)
doi: 10.18178/wcse.2022.06.043

Forged Cache Isolation on DNS Full-Service Resolvers and Identification of Infected End Clients

Yong Jin [1 +], Masahiko Tomoishi [2] and Satoshi Matsuura [3]

Tokyo Institute of Technology, Japan

[1] yongj@gsic.titech.ac.jp, [2] tomoishi@noc.titech.ac.jp, [3] matsuura@gsic.titech.ac.jp

Abstract. Domain Name System (DNS) plays an indispensable role in the Internet nowadays. Meanwhile, the cyber-attacks via the DNS based domain name resolution have become a critical issue especially DNS cache poisoning attack. DNS Security Extensions (DNSSEC) is one solution for mitigating the threats but the deployment rate is still very low throughout the whole Internet due to the high overhead on the DNS full-service resolvers and the high operational cost. DNS over TLS (DoT)/DNS over HTTPS (DoH), which are two ongoing standards only cover the communication between the end clients and the DNS full-service resolvers thus they cannot effectively mitigate the cache poisoning attacks. In this research, we propose a mechanism that isolates the forged or poisoned cache on the DNS full-service resolvers and identifies the infected end clients in order to mitigate further infections within an internal network. In this paper, we describe the design of the proposed mechanism and introduce a simple prototype implementation in a local network environment first. Then we show the preliminary evaluation results of basic functions of the proposed mechanism. Finally, we discuss some extra features may require for the further approach against DNS cache poisoning attacks and describe some future work regarding the deployment of the proposed system in a real network environment.

Keywords: DNS, full-service resolver, cache poisoning attack, forged cache isolation, DNSSEC, DoH

1. Introduction

Domain Name System (DNS) [1, 2] based domain name resolution is one of the most fundamental Internet services in the Internet for both of the Internet users and the Internet service providers. In the meanwhile, the cyber threats never stop in the Internet with the development of the Internet technologies and the increase of the Internet users. Many security reports indicated that numerous cyber-attacks had been conducted by practically using the DNS services due to the network administrators cannot simply block all DNS traffic in an organization network. One of the well-known cyber-attacks related to the DNS is cache poisoning attack [3], in which, the attacks inject fake DNS resource records into a DNS full-service resolver in order to make the end clients access the malicious web sites. Many approaches have been proposed in order to mitigate DNS cache poisoning attack and one of them is DNS Security Extensions (DNSSEC) [4]. The DNSSEC covers the communication between the DNS full-service resolver and DNS authoritative servers and provides the integrity of DNS responses. Consequently, the DNS cache poisoning attacks can be effectively mitigated by deploying the DNSSEC on all DNS full-service resolvers and all DNS authoritative servers. However, the enabling DNSSEC functions on a DNS full-service resolver can significantly raise the overhead and reduce the performance of domain name resolution service. Moreover, the administration cost on DNS authoritative servers including the key management and troubleshooting will be extremely increased. Due to these negative reasons, the deployment rate of the DNSSEC over the Internet is very low and DNS cache poisoning attacks targeting on DNS full-service resolvers is still increasing.

Another ongoing approach is DNS over TLS (DoT)/DNS over HTTPS (DoH) [5, 6] which provide secure communication between the end clients and the DNS full-service resolver. The main purpose of DoT/DoH is to protect the privacy of the Internet users and secure communication between the end clients and the DNS full-service resolvers which are not included in the DNSSEC. However, the communication between the DNS full-service and the DNS authoritative servers is as usual and the cache poisoning attacks

+ Corresponding author.
E-mail address: yongj@gsic.titech.ac.jp.

are still dangerous for the DNS full-service resolvers even the DoT/DoH has been deployed in an organization network.

In this research, we propose a novel mechanism to solve the existing issues regarding DNS cache poisoning attacks which includes two main features, one is the forged cache isolation on DNS full-service resolvers and the other is the identification of infected end clients. It should be noted that the phrase "forged caches" [7] used in this research indicates the cache in DNS full-service resolvers including malicious DNS resource records injected by the attackers and also wrong DNS resource records from misconfigured DNS authoritative servers by the network administrators.

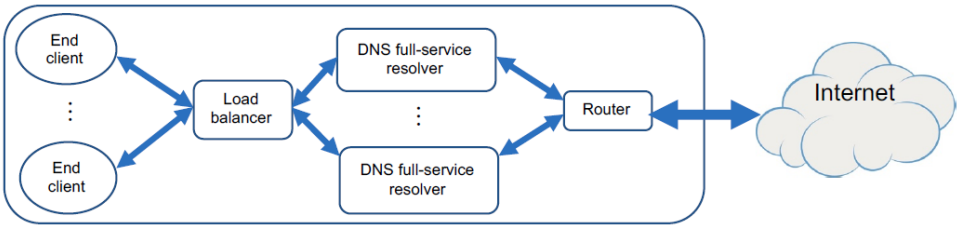

Fig. 1: Domain name resolution in an Internal network

2. Proposed mechanism

2.1. Target Network Architecture

The target network architecture of this research is as shown in Fig. 1. Almost of all organization networks run one or more DNS full-service resolvers in order to provide domain name resolution service for the internal computers. In such a network environment, an internal computer (hereafter end client) sends the domain name resolution requests to one of the DNS full-service resolvers setup in the internal network and the DNS full-service resolver performs the domain name resolution on behalf of the end client and replies the answers back to it. It should be noted that setting up a load balancer in front of the multiple DNS full-service resolvers is a common operational strategy in order to realize the redundancy of the domain name resolution service in an organization network. Consequently, the domain name resolution requests from the end clients can be sent any one of the running DNS full-service resolvers and therefore if one of the DNS full-service resolvers is infected by cache poisoning attack all end clients can be the victims. Accordingly, the approach against cache poisoning attacks should be applied on all the running DNS full-service resolvers in an organization network.

2.2. Design

The key idea of the proposed system is forged cache isolation on DNS full-service resolvers and identification of infected end clients. Since a successful DNS cache poisoning attack starts by sending DNS queries to DNS full-service resolvers and ends with injecting fake DNS resource records into the DNS full-service resolvers. Accordingly, in an organization network, there must exist the infected end client caused the DNS cache poisoning attacks and successively any end clients can be the victim of accessing the malicious servers indicated by the fake DNS resource records cached in the DNS full-service resolver.

Based on the above rationales, in the proposed system, firstly a feature for isolating forged cache (including the poisoned cache) is necessary in order to avoid being used by the end clients. This feature needs to be applied to all DNS full-service resolver running in an organization network simultaneously. Secondly, a feature for identifying the end client caused the cache poisoning attacks is also necessary since the end client may have been infected by some types of malwares. After that the end client needs to be investigated and processed before being reused in the organization network.

2.3. Forged Cache Isolation on DNS Full-Service Resolvers

In the proposed system, we intend to use DNS Response Policy Zones (DNS RPZ) [8] feature of Berkeley Internet Name Domain (BIND) [9] DNS server program in order to realize the forged cache isolation on all DNS full-service resolvers in an organization network. The network administrators are able to register any domain names in a DNS RPZ and reply and DNS resource records based on the policies. More importantly, the DNS RPZ can be configured on DNS full-service resolvers and all of them can be

updated by using zone transfer simultaneously. Consequently, if some DNS resource records are identified as forged or poisoned the network administrator can add them into the DNS RPZ with a secure IP address so that end clients which send the DNS query for the domain name can be avoid accessing the malicious servers.

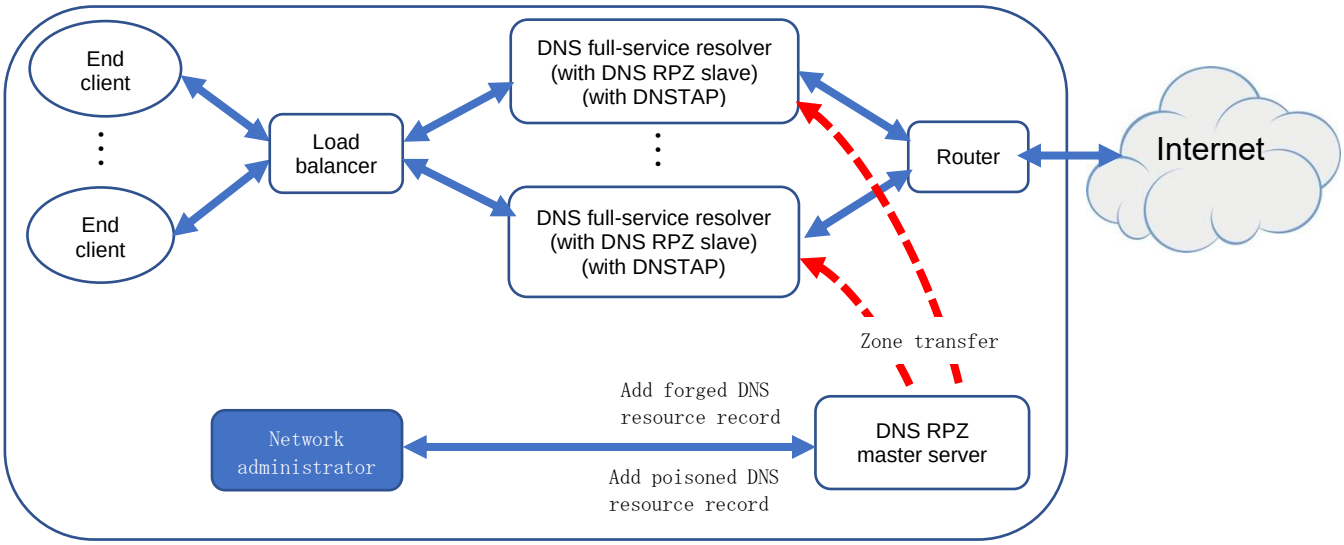

Fig. 2: Forged cache isolation feature in the proposed system.

As shown in Fig. 2, a DNS RPZ slave will be configured in each DNS full-service resolver in an organization network and a DNS RPZ master server will be setup for the administration. When the network administrators achieve forged cache information, they only need to add the corresponding DNS resource records in to the DNS RPZ master server. Then all the forged cache information will be updated to the DNS RPZ slave configured on each DNS full-service resolver by zone transfer feature of DNS protocol. Accordingly, if any end clients send any DNS query about the domain names added in the DNS RPZ, the DNS full-service resolvers will reply a secure (unused in real communication) IP address such as ``127.0.0.1" which will not cause any real Internet communication. Consequently, the forged cache of DNS full-service resolvers will be isolated and the end clients can avoid being affected by the fake DNS resource records. This procedure is similar to ``DNS blacklists and whitelists" [10] for determining whether allow or deny a connection before the establishment. It should be noted that the achievement of forged cache and poisoned cache needs collaboration with security facilities deployed in an organization network such as firewall system, Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) thus we omit the detailed description here.

2.4. Identification of Infected End Clients

As being mentioned in section 2.2, there must exist victim end clients when DNS cache poisoning attacks happen and the victim end client may have been infected by some types of malwares already. Therefore, in case of DNS cache poisoning attacks happen, we need to identify the victim end clients and perform detailed investigation in order to mitigate further spread of the malware. It should be noted that misconfiguration on authoritative DNS server may also cause wrong cache on DNS full-service resolvers. In this case, the network administrators only need to add the correct DNS resource records into the DNS RPZ until the cache TTL (Time To Live) expiration.

One solution for identifying the victim end clients when DNS cache poisoning attacks happen is the deployment of security facilities. However, we believe that the security facilities not only require high financial and administrative cost but also cannot guarantee 100\% detection and blocking. Therefore, we consider that there should be alternative solution for the cases of no high performance security facilities deployed and the cases of false negative.

In the proposed system, as shown in Fig. 3, we intend to enable DNSTAP [11] feature on all DNS full-service resolvers running in an organization network in order to effectively achieve and analyze the domain name resolution logs including end client and domain name information. Accordingly, when the network

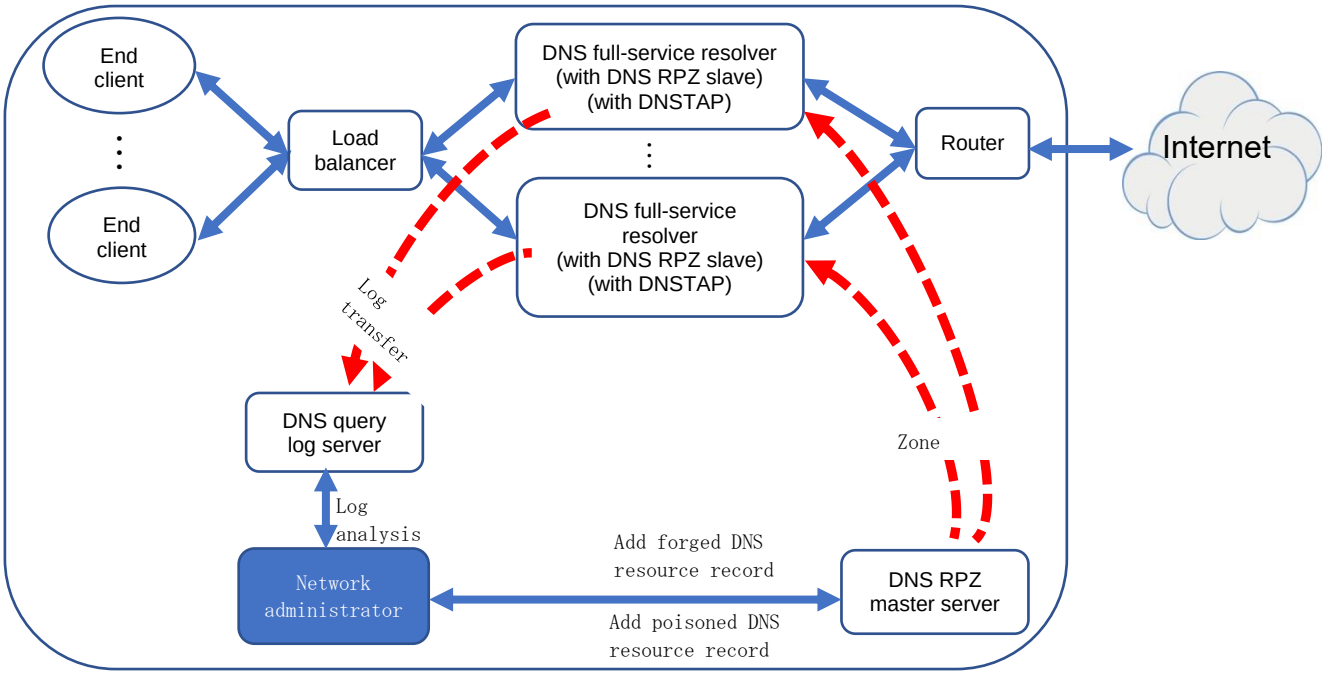

Fig. 3: Victim end client identification feature in the proposed system.

administrators detect or achieve forged cache information; they can immediately look out the involved end clients performed the domain name resolution related to the suspicious domain names. It should be noted that the information of the end clients logged by DNSTAP is the IP address of the DNS query packets. Thus when NAT (Network Address Translation) router is used on the client side, the network administrator cannot precisely identify the real end client under the NAT router. In this case, the network administrators cannot simply block all the DNS queries from the IP addresses since other end clients under the NAT router may not involve with the suspicious domain name resolution. Therefore, all the network administrators can do is the forged cache isolation and further investigation for the victim end clients.

3. Implementation and Evaluation

3.1. Prototype Implementation

Based on the design of the proposed system, we implemented a prototype within a local experimental network environment. All the components of the proposed system were implemented constructed on one physical machine as virtual machines using Linux KVM technology [12] as shown in Fig. 4. We only setup two DNS full-service resolvers for playing the roles of a normal and forged DNS full-service resolver respectively and two end clients for victim and normal respectively. All the virtual machines are connected to the Internet and the proposed system can be evaluated.

For the simplicity, we used CentOS 7 as the operating system on all of the components and selected the most widely used DNS server program BIND as the DNS full-server resolver and the DNS RPZ master server. Since we used KVM technology and constructed all the components on one physical machine, all the virtual machines can communicate with each other and the end clients used the DNS full-service servers as the name server for domain name resolution. Since we did not setup a load balancer for the DNS full-service resolvers, we assume the victim end client uses the forged DNS full-service resolver and the normal end client uses the normal DNS full-service resolver for the domain name resolution. In fact, the normal end client also can send DNS queries to the forged DNS full-service resolver in a real network environment in which a load balancer has been deployed.

In the prototype system, a DNS RPZ named ``example.com'' is configured in each DNS full-service resolver as slave and the DNS RPZ master server as master and once the master DNS RPZ has been updated the other two slave DNS RPZ will be updated too using zone transfer of the DNS protocol.

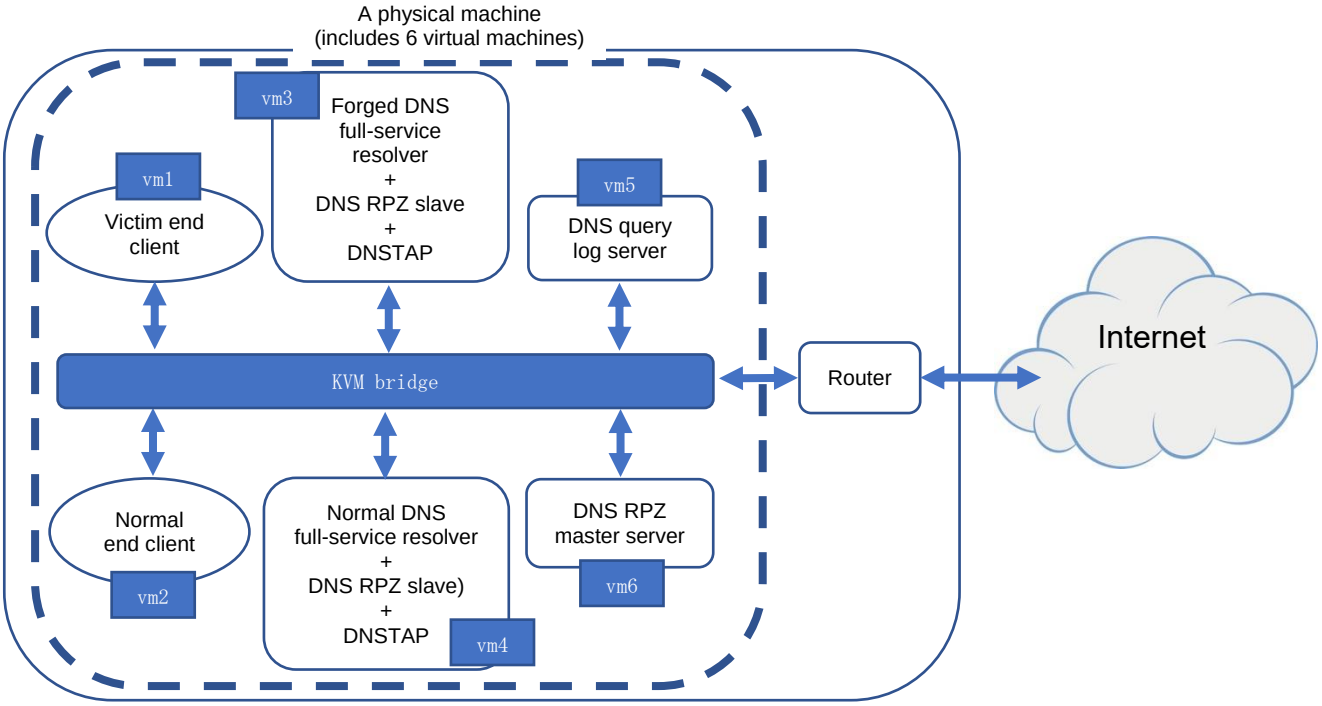

Fig. 4: The architecture of the prototype system

3.2. Preliminary Evaluation

We performed preliminary evaluations for the prototype system and checked the features of the proposed system. First, we accessed several web sites such as ``www.google.com" in the Internet on the end clients (victim and normal) using command ``Wget" [13] with the Fully Qualified Domain Name (FQDN) [14] and confirmed the domain name resolution requests were sent to the DNS full-service resolvers before accessing the web sites. Next, we added the DNS resource record ``www.google.com. IN A 127.0.0.1" in the DNS RPZ master and confirmed that the record was updated in the slave DNS RPZ of the DNS full-service resolvers. Then, we tried again from the end clients to access the web site ``www.gogole.com" and confirmed that the end clients attempted to access the IP address ``127.0.0.1" which ended with failure. Finally, we checked the DNS full-service resolver for the DNS query log and confirmed that the DNSTAP feature worked correctly and the DNS query logs from the end client were achieved as expected and they were correctly transferred to the DNS query log server.

4. Discussion

In the proposed mechanism, we only included the features for forged cache isolation and identification of the infected end clients as a solution for DNS cache poisoning attacks. In reality, the victim end clients may successively conduct Internet communication until the network administrators identify them and block the communication. Accordingly, in order to mitigate the damage of from DNS cache poisoning attacks more effectively and promptly it is necessary to identify and block the victim end clients as fast as possible. Therefore, as an improvement plan, we intend to add a feature to check if the source IP address of a DNS query has been involved with the suspicious domain name related to the detected DNS poisoning attacks in the proposed mechanism. Based on the check results, the DNS query will be determined whether or not to be sent out so that the successive Internet communication can be effectively detected and blocked as fast as possible.

As a specific solution for realizing the feature in the proposed mechanism, we intend to add two new components named ``DNS proxy" and ``Check list database" respectively. The ``DNS proxy" [15] receives all DNS queries from the end clients and checks the source IP address of the packet in the ``Check list database" before forwarding it to the DNS full-service resolvers. If the source IP address is listed in the database the DNS query will be dropped, otherwise will be passed through. For the feature of network control (drop or pass the DNS query packets) feature, we intend to use SDN (Software Defined Network) technology [16].

5. Conclusion

In this paper, we proposed a mechanism for forged cache isolation on DNS full-service resolvers and identification of infected end clients. In the proposed mechanism, the feature of forged cache isolation on DNS full-service resolvers is realized by using DNS RPZ function of BIND (the most widely used DNS server program) and the identification of the infected end clients is realized by using DNSTAP function. By using the proposed mechanism, when some suspicious DNS cache was detected, it is able to be isolated on the DNS full-service resolvers without stop the domain name resolution service. Furthermore, the corresponding victim end client can be identified by logging all the DNS queries using the DNSTAP function. We implemented a prototype system in a local experimental network environment and conducted preliminary evaluations for the proposed features. The evaluation results confirmed that each individual feature worked correctly as designed.

In the future work, we plan to include the implementation of the new features mentioned in section 4 and the complete evaluation through the entire system. Moreover, in addition to the feature evaluations on a local experimental network, we also plan to deploy the prototype system in a real network environment and conduct performance evaluation.

6. Acknowledgement

This work was partially supported by JSPS KAKENHI (Grants-in-Aid for Scientific Research) Grant Number 19K20254, 18K11291.

7. References

[1] P. Mockapetris, ``Domain names - concepts and facilities," IETF RFC1034, November 1987.

[2] P. Mockapetris, ``Domain names - implementation and specification," IETF RFC1035, November 1987.

[3] Kaspersky, ``What is DNS Cache Poisoning and DNS Spoofing?" (online), available from https://usa.kaspersky.com/resource-center/definitions/dns

[4] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, ``DNS Security Introduction and Requirements," IETF RFC4033, March 2005.

[5] S. Dickinson, D. Gillmor, T. Reddy, ``Usage Profiles for DNS over TLS and DNS over DTLS," IETF RFC8310, March 2018.

[6] P. Hoffman, P. McManus, ``DNS Queries over HTTPS (DoH)," IETF RFC8484, October 2018.

[7] A. Hubert and R. van Mook, ``Measures for Making DNS More Resilient against Forged Answers," IETF RFC5452, January 2009.

[8] Internet Systems Consortium, ``Response Policy Zones (RPZ)" (online), available from https://www.isc.org/rpz/

[9] Internet Systems Consortium, ``BIND 9", available from https://www.isc.org/downloads/bind/

[10] J. Levine, ``DNS Blacklists and Whitelists," IETF RFC5782, February 2010.

[11] Robert Edmonds, ``dnstap: high speed DNS server event replication without packet capture" (online), available from https://dnstap.info/slides/dnstap.html\#(1)

[12] Red Hat OpenShift Online, ``Kernel Virtual Machine" (online), available from https://www.linux-kvm.org/page/Main_Page

[13] ``GNU Wget" (online), https://www.gnu.org/software/wget/

[14] M. Stapp, ``Resolution of Fully Qualified Domain Name (FQDN) Conflicts among Dynamic Host Configuration Protocol (DHCP) Clients," IETF RFC4703, October 2006.

[15] ``Net::DNS::Dynamic::Proxyserver - A dynamic DNS proxy-server" (online), available from https://metacpan.org/pod/Net::DNS::Dynamic::Proxyserver

[16] ``Open Networking Foundation: SDN Architecture" (online), https://opennetworking.org/wp-content/uploads/2014/10/TR-521_SDN_Architecture_issue_1.1.pdf