DNSSEC as a service – A prototype implementation

Adnath Hemanthindra
School of Computing
Middlesex University Mauritius
Flic-en-Flac, Mauritius
HA91@live.mdx.ac.uk

Amreesh Phokeer
Research & Innovation Dept.
AFRINIC
Ebene, Mauritius
amreesh@afrinic.net

Visham Ramsurrun
School of Computing
Middlesex University Mauritius
Flic-en-Flac, Mauritius
v.ramsurrun@mdx.ac.uk

Panagiota Katsina
School of Computer Science
Middlesex University London
London, UK
p.katsina@mdx.ac.uk

Sumit Anantwar
School of Computer Science
Middlesex University London
London, UK
s.anantwar@mdx.ac.uk

Amar Kumar Seeam
School of Computing
Middlesex University Mauritius
Flic-en-Flac, Mauritius
a.seeam@mdx.ac.uk

***Abstract*—Domain Name System (DNS) plays a massive role in today's technological era. While initially designed to facilitate communications over the Internet and over networks, the DNS in itself is not secure enough considering the type and criticality of information being shared today. Considering its worldwide acceptance and popularity, securing the DNS without breaking its operation has become vital. DNSSEC is seen as a viable option to protect the integrity of the data and prevent on the fly modifications. However, its adoption rate is not encouraging. Research shows that complexity associated with currently proposed solutions were major turn off for organizations. This paper proposes the creation of a DNSSEC signing service whereby customers register themselves with the service provider and the latter deploys a signing environment for them which includes a DNSSEC signer, a database and web services for access purposes. Customers will only have to use the web services to create and manage their zones and the zone signing can be done automatically or with a simple click of a button. Signed zones are sent back to customer authoritative DNS servers securely using Transaction SIGnature (TSIG) and incoming DNS requests are signed. This solution involves open-source tools and service providers make use of Linux containers for customer environment and space for resource efficiency. All the complexity and additional maintenance involving the system are taken off the customer's shoulders and managed by the provider while also facilitating their tasks through GUI operations.**

Keywords—DNSSEC, zone signing service, security, TSIG, container

I. INTRODUCTION

The Internet has wiped out geographical barriers and the sharing of information has never been easier around the globe. One of the pillars of today's Internet is the DNS, which enables easily remembered text-based domain names to be translated into machine-readable IP addresses for network communication. DNS enables the formulation of URLs through Fully Qualified Domain Names (FQDNs) used by the Internet users to access their favourite sites without the need to remember each web server's IP address. Domain names are also used by various other Internet technologies like SIP protocol for VoIP and the Email system. However, since its inception in 1983, DNS was designed for scalability in mind and less focused on security. In 1990, the first DNS flaws were detected and today the types of DNS attacks are numerous [1].

A. Problem Definition and Motivation

On the 22nd February 2019, Internet Corporation for Assigned Names and Numbers (ICANN) published an announcement issuing a warning of an "ongoing and significant risk" on the DNS infrastructure following months of multifaceted attacks [2]. In an attempt to mitigate attacks, ICANN called for collaboration and encouraged the adoption of DNSSEC across all domains. However, despite risks regarding DNS are well known and acknowledged by the IT society, DNSSEC's adoption has been quite slow. According to data gathered on the Asia Pacific Network Information Centre (APNIC) [3] website, it is seen that as at September 2019, less than 25 % of DNS requests are validated by resolvers. As stated by Adrichem *et al.* [1], the main causes for lack of DNSSEC's adoptions are:

- Management of DNSSEC infrastructure is complex and requires skilled individuals to maintain it. Corporates are reluctant to invest into new hardware and skills which brings no profit.
- It was observed that although DNSSEC adoption was increasing, the number of signed zones were still low mainly due to misconfigurations. For example, errors in key rollovers process could lead to unreachability and lead to users not getting Internet access. This was the case for the *.nl*, Dutch country code Top-Level Domain (ccTLD), which in turn affected nearly 5 million sub-domains under *.nl*. [4]. So these risks make organizations simply abandon their efforts.
- Companies wanting to adopt DNSSEC on their name servers would also need to find a registrar that supports it. As per Jacques Latour [5], in his article, he noticed that not many registrars support DNSSEC. At the time of writing of this paper, out of a sample of more than 50 CA Registrars, only 11 of them could offer DNSSEC capability.

B. Aims & Objectives

The goal of this research work is to devise a way to help service providers like a registry or Network Information Centre (NIC) to provide DNSSEC as a service to DNS Operators. A prototype system was developed which would sign zones on behalf of the DNS operator using suitable tools and push back the signed zones to them. Infrastructure and DNSSEC environment would be managed by the service provider itself in an attempt to facilitate DNSSEC adoption by organizations. The latter would only have to register to the service and provide suitable information so that an isolated DNSSEC environment could be deployed for the user.

The objectives of this project were as follows:

- The design of a signing service which would sign zones on behalf of DNS operators and to transfer them back to their DNS servers.
- The solution should allow isolated environments whereby each customer registering to this service

would not have to share the same system with other customers.

- The front end management should be simple and easy to use by the customer when managing their zones. The solution should provide an easy to use web interface allowing users to add, delete and modify zone information for their DNS.
- The DNSSEC service provider should do the heavy lifting in terms of maintenance of the environment and key management. This includes key roll overs and as well as patching of servers and applications.
- The solution should also be easy for service providers to deploy DNSSEC environments for customers as quickly as possible with minimal redundant configurations and installations to be done. Managing the set-up should not be complicated as this could lead to service unavailability in case of issues.
- The proposed solution should also ensure security and employ HTTPS for web access as well as secure transfer of zones to DNS operator's servers.

II. Related works

This section reviews other related research works involving the development of DNSSEC-based solutions. These works were analyzed in terms of their strengths and weakness.

Yong *et al.* [6] list the additional overload on recursive DNS servers for DNSSEC validation as one of the reasons listed for the low DNSSEC adoption. Consequently, the authors have suggested an enhancement of the system referred to as a client-based DNSSEC Validation Mechanism with Recursive DNS server separation. The prototype consists of a Resource Record Signature (RRSIG) recursive DNS server and makes use of Google Public DNS server as RR recursive server. The evaluation of the new proposed system are acceptable as per results obtained, which attempts to reduce workload on public recursive servers and improve DNSSEC adoption and also validation of records by end clients. The main shortcoming of this work was that this solution was not tested in real network environment.

Migault *et al.* [7] state that complexity is a major concern with DNSSEC, and hence, is a reason for its low adoption. This paper focuses on evaluating the performance of popular DNS software and implementations like Berkeley Internet Name Domain (BIND), Unbound and Name Server Daemon (NSD). Results from tests shows that overall NSD have better performance than BIND with lower network latency and is less impacted by DNSSEC additional workload. BIND in turn shows lower performance than UNBOUND but the latter seems to be more impacted by DNSSSEC workloads.

Ju *et al.* [8] provided in their paper an alternative solution to DNSSEC. As per the article, the main concern with DNSSEC is that it has to be implemented through all the concerned parties along the DNS hierarchy starting from the root. Also, it is suggested that DNSSEC may degrade the overall performance of the DNS. Hence, a real-time detection mechanism for cache poisoning attack and post-checking mechanism in recursive DNS was proposed. The proposed system termed as Cache Poisoning Detection Method for Improving Security of Recursive DNS (CPDS) consisted of a recursive query manager, an iterative query manager, a database manager, a cache validation manager and an alert manager. Simulations and tests showed that the CPDS solution could provide a good alternative to DNSSEC with better performances since it does not employs any cryptographic technique.

Adrichem *et al.* [1] tried to explain how misconfigurations represent another reason for the poor statistics for DNSSEC adoption. Misconfigurations could cause network unreachability and organizations prefer not to risk their operation. This paper analyzes the *.br*, *.co* and *.se* domains for their experimental research. The article classifies DNSSEC misconfigurations into three parts namely: zone-based, delegation-based or anchor-based. It is stated that out of a sample of 1456 signed zones, 194 of them were misconfigured and the impact on reachability as per research were clear and considerable. This paper showed that implementation of DNSSEC was not simple as misconfigurations were on a high as per sampled data.

III. Analysis & Design of Proposed solution

The proposed solution consists of a DNSSEC signer, a database and web services for portal access per customer.

- *DNSSEC Signing service:* This service is responsible for zone management, zone signing, and the configurations of the DNS Master server.
- *Web Services:* The web services will allow customers access web services on the environment for GUI management and monitoring.
- *Database:* The database will hold all the data for the customer DNS as well as the required keys and signatures.

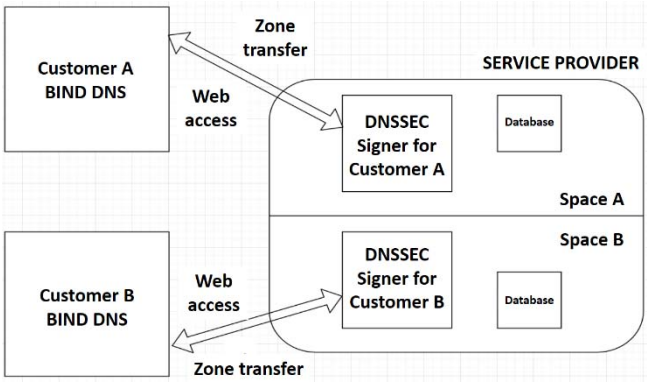

Fig. 1. System Architecture

The DNSSEC signer would provide DNSSEC signing as a service to organizations in order to sign their zones without much hassle. Customers will register to the service with the service provider who is responsible for the signing environment, resource allocation and service availability. The service provider will create a DNSSEC signer for the customer on their infrastructure. The environment will be isolated per customer to ensure privacy and security. Customers will then get access to the DNSSEC signer through a web portal which they can access using set and agreed credentials. The portal allows the customer to create zones, add, modify and delete records as per their requirements. The customer will also be able to activate/deactivate DNSSEC on their zone easily. After saving their zone configurations and validating, these signed-zones will be transferred to the customer DNS servers configured as slaves in a secure manner. This solution does

not intend to alter the customer's DNS setup and will work on most popular DNS software like BIND and PowerDNS.

Fig. 1 illustrates an overview of the planned system architecture showing as example two customers each having different DNS software. Access to service provider are performed through HTTPS protocol and Zone transfer protocols only. The figure also shows separate spaces for each customer for their signing environment.

IV. IMPLEMENTATION

The software versions used for developing this prototype are listed in the table below:

TABLE I. SOFTWARE COMPONENTS

Software	Version
Linux OS	Ubuntu 18.04.3 LTS
LXD	3.0.3
PowerDNS Authoritative Server	4.2.0
MySQL Database	5.7.27
PowerDNS-Admin (web management for PowerDNS)	Required: python3-dev
Webmin(web management for Linux)	1.930

a) Environment Set-up: The first task is the setting up of the environment for the implementation. For security reasons, a Virtual Machine is used for this setting up as VMs can be easily backed-up and provide sandboxes for experimentation.

b) Linux with LXD: In the Virtual machine created on top of VMware Workstation, the sandbox environment for the prototype is created with the installation of Linux Ubuntu Server 18.04 LTS.

c) DNSSEC signer:

For the DNSSEC signer, an instance of Ubuntu 18.04 Bionic is used and spawned using the command *lxc launch images:ubuntu/18.04*. After the process, the commands *lxc list* shows that the container spawned has the name optimal-sawfish with IP address of 192.168.64.134. The container can be accessed directly from the host using the command *lxc exec optimal-sawfish /bin/bash*. Using the *apt-get install* command, the following packages are installed: *mysql-server, pdns-server, and pdns-backend-mysql*. The database for the signer is constructed with the database name set to *pdns*. In order to link the PowerDNS server with the MySQL database, the file *pdns.local.gmysql.conf* is edited to add the connection parameters. The base configuration for the PowerDNS server can be done by editing the file *pdns.conf*. For this prototype, only some basics configuration were done and required. The api was enabled for web application access and the server was defined as master since the PowerDNS Authoritative server will act as a DNS Master. Before starting the PowerDNS service, the default resolver on the Ubuntu system is deactivated since both uses the same port 53. After a system reboot, the PowerDNS server is up and running on the container. The *Powerdns-admin* application can be pulled from github [9]. Pre-requisites and necessary packages for the application involves the installation of *python3-dev, libmysqlclient-dev, libsasl2-dev, libldap2-dev, libssl-dev, libxml2-dev, libxslt1-dev, libxmlsec1-dev, libffi- dev* and *pkg-config*. The *Powerdns-admin* application is then set-up behind an Nginx reverse proxy for web access. The creation of a service file ensures that the service can be monitored and managed using system commands. The *Powerdns-admin* web interface can be accessed by typing the ip address of the server. For the Linux system management and control, *Webmin 1.930* was downloaded from [10] for debian-based linux and it was installed.

d) Test DNS slaves: After the implementation of the DNSSEC signer, test DNS Slave servers need to be implemented so that the functionality could be tested. On the same container environment, two additional containers are deployed whereby one of the BIND DNS server is installed and the other one a PowerDNS DNS server. The details of the containers and service are as follows:

TABLE II. DNS TEST SLAVES

Container Name	IP address	Service Name
Relieved-clam	192.168.64.138	Bind9 DNS server
Novel-muskrat	192.168.64.135	Powerdns authoritative server

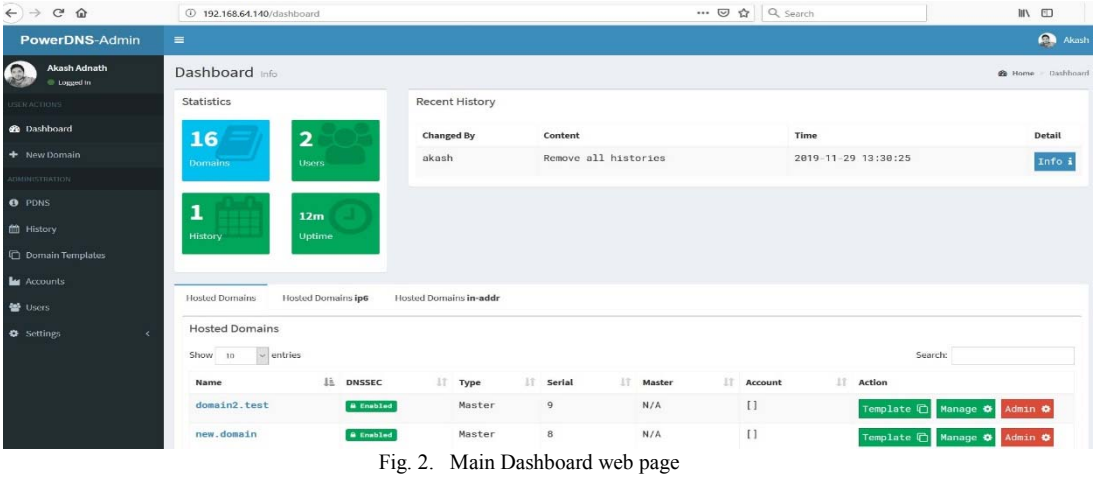

Fig. 2. Main Dashboard web page

e) Web Interfaces: The prototype consisted of two web interfaces, one for management of the DNSSEC signer and one for the monitoring and management of the environment.

- DNSSEC Signer Web application:

Fig. 2 shows the main dashboard after a successful login on the service. The dashboard using the administrator access having all access on the DNSSEC signing service and two test domains are shown for illustration. On the left pane, all the accessible view options are shown for configuration purposes or viewing.

- System Monitoring and Management Dashboard:

The system monitoring and management on the DNSSEC signing environment per customer's role is to facilitate the task of system administrators. This access is different to the DNSSEC service that is mostly used by the customer, and this web service is accessible on port 10000.

V. System Testing & Evaluation

A series of test cases were developed to ascertain the correct functioning of the prototype, namely:

- Setting up a new signer and deployment environment for a customer, and testing service access credentials
- New domain creation with DNSSEC
- DNS Records manipulations (add, delete, modify)
- Testing DNSSEC on Customer DNS (DNS Slave)
- Testing TSIG and configuration of signer
- Real-time monitoring of system and Key Rollovers by Administrators
- Testing with PowerDNS slave DNS server

The major strengths and weaknesses of the proposed system are as follows:

a) Strengths

- DNSSEC operation is no more difficult and does not require expert-level skills to use it.
- Providers can deploy new customers on the fly with little configuration required.
- Zone transfers are secure between Service provider and customer DNS and done within acceptable delays.
- Customers have full visibility of zone change history and can apply restricted access on their environment.
- Monitoring and maintenance of the environment are easier through intuitive web services.
- The solution works with popular DNS software like BIND DNS and PowerDNS and does not limit customers with a particular product.

b) Weaknesses:

- DNSSEC Keys are stored without a Hardware Security Module (HSM) and are therefore less secure.
- Customer zone information are hosted at service providers. A legal agreement is needed to avoid any privacy breach.
- This prototype supports only a few containers. A more robust environment to manage containers is required.

VI. Conclusion & Future Works

This paper has presented a prototype involving DNSSEC and TSIG protocols packaged in single system and offered as a service to organizations rather than the latter making all the configurations and setting up. Service providers can use this DNSSEC signing system to sign customer unsigned zones so that DNSSEC is adopted from the bottom of the DNS hierarchy. The solution provides an easy to use GUI, which facilitates zone signing with proper keys and secure zone transfers performed between service provider and customer. Tests have shown that queries on customer authoritative servers are signed while the system is maintained by the service provider. The system also helps service providers deploy with minimal configurations required new customers on a container based environment for efficient usage of compute resources. System monitoring and maintenance are performed through a web interface and the availability of the signing service does not affect the customer DNS operation allowing system patching and maintenance to be performed on agreed timeframe. The system requirements defined earlier are met by choosing the right protocols for this prototype.

Future works may look at the following:

- Since the service provider will have to deal with DNSSEC keys for customer zones and TSIG key for transfers, an HSM will provide a more robust and secure storage of those keys and increase trust between provider and customer.
- Zone changes notifications by email could be set-up to help administrators or customers to validate those changes and avoid impersonation attack.

Acknowledgment

This project was done as part of AFRINIC Research Collaboration (ARC) 2019 programme.

References

[1] N. L. M. v. Adrichem, A. R. Lua, X. Wang, M. Wasif, F. Fatturrahman and F. A. Kuipers, "DNSSEC Misconfigurations: How Incorrectly Configured Security Leads to Unreachability," 2014 IEEE Joint Intelligence and Security Informatics Conference, The Hague, 2014, pp. 9-16, doi: 10.1109/JISIC.2014.12.

[2] ICANN.ORG, "Icann Calls For Full Dnssec Deployment, Promotes Community Collaboration To Protect The Internet," [Online] Available At: https://www.icann.org/news/announcement-2019-02-22-en [Accessed 21 Aug. 2019].

[3] APNIC, "DNSSEC Validation Rate by country (%)," [Online] Available at: https://stats.labs.apnic.net/DNSSEC [Accessed 19 Sept. 2019].

[4] M. Müller, T. Chung, A. Mislove and R. van Rijswijk-Deij, "Rolling With Confidence: Managing the Complexity of DNSSEC Operations," in IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 1199-1211, Sept. 2019, doi: 10.1109/TNSM.2019.2916176.

[5] J. Latour, "Increasing DNSSEC Adoption - What if We Put DNSSEC Provision in the Hands of Registries?," [Online] Available at: http://www.circleid.com/POSTS/20150715_DNSSEC_ADOPTION_PUTING_DNSSEC_PROVISION_IN_HANDS_OF_REGISTRIES/ [Accessed 30 Oct 2020].

[6] Y. Jin, M. Tomoishi and N. Yamai, "A Client Based DNSSEC Validation Mechanism with Recursive DNS Server Separation," 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2018, pp. 148-153, doi: 10.1109/ICTC.2018.8539727.

[7] D. Migault, C. Girard and M. Laurent, "A performance view on DNSSEC migration," 2010 International Conference on Network and Service Management, Niagara Falls, ON, 2010, pp. 469-474, doi: 10.1109/CNSM.2010.5691275.

[8] Y. W. Ju, K. H. Song, E. J. Lee and Y. T. Shin, "Cache Poisoning Detection Method for Improving Security of Recursive DNS," The 9th International Conference on Advanced Communication Technology, Okamoto, Kobe, 2007, pp. 1961-1965, doi: 10.1109/ICACT.2007.358755.

[9] PowerDNS-Admin, [Online] Available at: https://github.com/ngoduykhanh/PowerDNS-Admin [Accessed 30 Jul 2020].

[10] Webmin 1.930, "Installing on Debian," , [Online] Available at: https://www.webmin.com/deb.html [Accessed 30 Oct 2019]