DNS Security Challenges and Best Practices to Deploy Secure DNS with Digital Signatures

M. H. Jalalzai, W. B. Shahid and M. M. W. Iqbal
National University of Sciences and Technology Islamabad Pakistan
haseeb.msis12@students.mcs.edu.pk

***Abstract*—This paper is meant to discuss the DNS security vulnerabilities and best practices to address DNS security challenges. The Domain Name System (DNS) is the foundation of internet which translates user friendly domains, named based Resource Records (RR) into corresponding IP addresses and vice-versa. Nowadays usage of DNS services are not merely for translating domain names, but it is also used to block spam, email authentication like DKIM and the latest DMARC, the TXT records found in DNS are mainly about improving the security of services. So, virtually almost every internet application is using DNS. If not works properly then whole internet communication will collapse. Therefore security of DNS infrastructures is one of the core requirements for any organization in current cyber security arena. DNS are favorite place for attackers due to huge loss of its outcome. So breach in DNS security will in resultant affects the trust worthiness of whole internet. Therefore security of DNS is paramount, in case DNS infrastructure is vulnerable and compromised, organizations lose their revenue, they face downtime, customer dissatisfaction, privacy loss, confront legal challenges and many more. As we know that DNS is now become the largest distributed database, but initially at the time of DNS design the only goal was to provide scalable and available name resolution service but its security perspectives were not focused and overlooked at that time. So there are number of security flaws exist and there is an urgent requirement to provide some additional mechanism for addressing known vulnerabilities. From these security challenges, most important one is DNS data integrity and availability. For this purpose we introduced cryptographic framework that is configured on open source platform by incorporating DNSSEC with Bind DNS software which addresses integrity and availability issues of DNS by establishing DNS chain of trust using digitally signed DNS data.**

***Index Terms*—DNS Vulnerabilities, DNS Security, PKI, Digital Signatures, DNSSEC, Network and Computer Security**

I. Introduction

The Domain Name System (DNS) is the foundation of internet which translates user friendly named based Resource Records (RR) into corresponding IP addresses and vice-versa. But nowadays DNS is not only addresses translation it is more than this to provide authentication and improve security services of many internet applications. Now DNS becomes most critical part of internet and it should work properly otherwise whole internet communication will collapse. Therefore security of DNS infrastructure is one of the core requirements for any organization. The NIST document [1] elaborated the importance of secure DNS deployment, this documentation supplements the importance of security in DNS and provides concrete examples of openDNS software as well in order to achieve secure DNS infrastructure. DNS are favorite place for attackers due to huge loss of its outcome, DNS failure will collapse all live hosts and internet applications published over an internet. Hence the breach in DNS security will in resultant affects the trust worthiness of internet. Therefore security of DNS is paramount, in case DNS infrastructure is compromised organizations lose their revenue, they degrade their reliability due to downtime, customer dissatisfaction, privacy loss, confront legal challenges and many more named few. DNS (Domain Name System) is hierarchical mapping of dynamic database scattered globally which provides numerous internet related application services. The database comprises of various resource records (RRs) like Address ”A” record, reverse pointer record ”PTR”, mail exchanger ”MX” record, Name Server ”NS” record, TXT record and other authentication services like DKIM, DMARC, DANE [2] and many more. These RRs are assembled into zones and which are retained locally on a name server or hosted DNS and then they are globally accessible from a widely dispersed architecture of DNS. The destination port of DNS is 53, which works on both TCP and UDP. DNS forms a hierarchical tree-like data structure of domain names, in which nodes (domains) have resource records (RRs) which may or may not contain information for a particular domain name space. The top most hierarchy is root zone denoted by dot ”.”, which is typically not mentioned in applications. Under the root, domains are categorized into Top Level Domains (TLDs), Country Code ccTLDs or generic gTLDs. For example Fully Qualified Domain Name (FQDN) of ICANN is www.icann.org. in this the right most dot ”.” denotes root zone. Domains are composed of one or more labels, which are separated by ”.” and the maximum length of each label is 63 characters. The maximum length for FQDN is 255 characters inclusive of dot. Labels start from right to left, where TLD is at far right of label for a particular domain. Following the example www.icann.org. TLD for ICANN domain is identifying as:

org is the TLD for www.icann.org which is furthest to right.

Now to identify a domain name space following the diagram, everything below the ”.org” domain name space belongs to org domain and everything underneath ”.icann.org”

978-1-4799-6369-0/15/$31.00 ©2015 IEEE

domain names space is in the .icann.org domain.

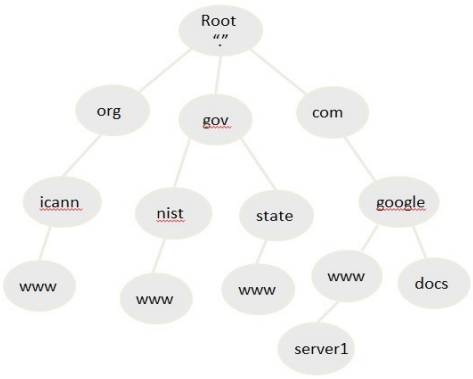

Fig. 1. DNS hierarchical structure

II. DNS Function

Before discussing the security challenges of DNS, we need to understand the DNS functionality. Hosts request for a particular resource by sending a ”recursive” query to its configured DNS Servers. Clients will get the name resolution answer of forwarded query or error message that it could not found anywhere. As we have discussed DNS servers are distributed globally, so before giving an error message they query other name servers until it gets the answer or query failed response. So, basically DNS functions on two queries one is recursive and other is iterative. Queries which are initiated from clients are recursive queries and other search queries are iterative which are initiated from local DNS servers to authoritative name servers. So, when a local DNS server receive query from clients and it doesn't have information in its cache, then they forward ”DNS Referral Message” back to clients for the name server that may have the answer which can be authoritative name server or a lower level DNS in a hierarchical structure as we talked earlier. Figure below illustrates recursive and iterative queries used in DNS.

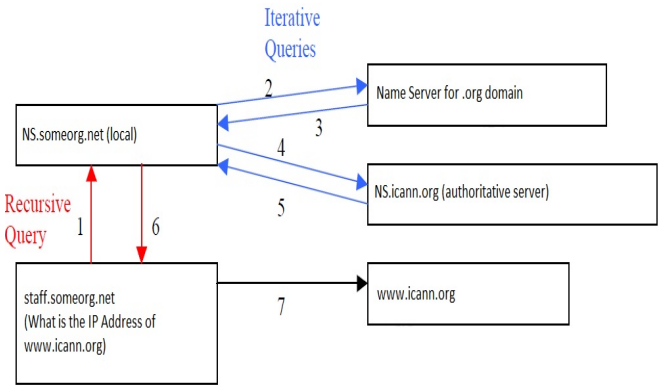

Fig. 2. Recursive versus iterative queries

A figure without cache search of www.icann.org name resolution process is shown below.

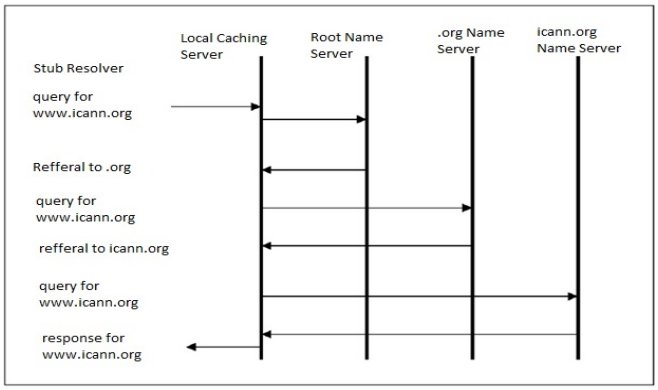

Fig. 3. DNS name resolution without cache search

III. DNS Vulnerabilities and Security Challenges

The DNS flaws exist if DNS server will not configure properly. DNS or Name servers play a most critical role in overall internet communication. This service is being used on multiple platforms, operating systems, countless applications, various operators and technical experts. This must be hardened in order to avoid from its malicious threats and attacks. Most common flaws are mentioned in this document to inform DNS community how vulnerabilities can be exploited and DNS become vulnerable. Along with this best practices and techniques are also discussed in this paper to prevent these types of malicious activities.

IV. DNS Vulnerabilities and Known Attacks

A. DNS Open Resolver

DNS Open Resolver is DNS server which is open to all clients to provide name resolution regardless the requestor is part of its domain or not. So, they are entertaining requests (queries) to every client by responding them back to their queries and they are not configured with any administrative controls. Open resolver is a hot cake for attackers and can be vulnerable to perform following malicious activities and attacks against DNS servers.

- DoS or DDoS attacks
- DNS Cache Poisoning Attacks
- Resource Utilizing Attacks
- DNS ID Hacking Attacks

B. Denial of Service and Distributed Denial of Service (DDoS) attacks

are possible against the name server which is open for every client. Attacker uses open DNS servers to maximize the attack volume and they also spoofed IP addresses in order to hide the attack source, in result DoS or DDoS attack will occur. In this case the resolver sends the query to these open recursive name servers spoofing the IP Address of the target and that is really easy to do because in this case the DNS is UDP based so attacker just tempering IP address in the IP message header. Attacker utilizes these open DNS servers to perform malicious

activities, they forged and spoofed the source IP address of target by sending query to open DNS servers and these servers respond back to target which will eventually malfunctioned. To magnify the attack volume attacker chooses multiple open DNS servers, in this way DoS or DDoS attack will be perform against target.

C. DNS Cache Poisoning attacks

DNS Cache Poisoning attacks [3] are very serious threat against DNS infrastructure. It's very easy idea to understand inducing a name server to cache bogus resource records where these bogus resource records might be machines or hosts run by hacker. For example bogus address for some.bank.com, or any insurance company's website poisoned with the IP address of webserver run by hacker where hackers hosted digitally identical replica of targeted original website and all users will be false victim to cache poisoning attack where end users are victimized for login, password, account info, credit card numbers are captured and so on will be recorded and used later on. Another example of cache poisoning attack is to falsify email communication with rerouting email by using poisoned MX record of hacker and then email may be modified without the sender knowledge.

D. Resource Utilizing Attacks

Resource Utilizing Attacks are used to degrade the performance of open DNS server by utilizing device resources like memory, CPU, socket buffers. These attacks will consume all available resources of open server and directly impact on the operations of open resolver and servers may have to stop or restart DNS services or may be reboot the server in order flush from occupied system resources.

E. DNS ID hacking

DNS ID hacking is another common type of threat in DNS environment in which attacker impersonates by MITM DNS response requesting by a client to misdirect clients without utilizing the cache of impersonate DNS server. In BIND DNS id is generated through Pseudo Random Number Generators (PRNG) and analysis has been done with BIND 9.0 that if 13 to 15 consecutive IDs will be spoofed the whole PRNG function can predict and sequence of ID can be generate. This is the known vulnerability in BIND specific software version.

F. DNS change malware

DNS change malware is known malware which changes clients systems stub resolvers to direct malicious recursive servers, where clients will connect to malicious sites or other non-intended resources. As we see attacks are numerous but the problem was here that being DNS community we trusted [4] on our personal name server and worthy trusting more on caching data that were unrelated to question that who have asked query and the answer we are receiving is the same that is originally with domain authoritative name server. For this organizations have to follow best practices in deploying secure domain name systems. SANS whitepaper [5] also discusses security issues of DNS.

V. BEST PRACTICES FOR SECURING DNS

Now step by step approaches will be discuss to accomplish a secure DNS environment so that discussed attacks and flaws will be addressed. For this purpose open source software Berkeley Internet Name Domain (BIND), DNS software provided by ISC (Internet Systems Consortium) is demonstrated in this paper in order to achieve secure DNS infrastructure.

There are lots of things to do for securing DNS server or group of servers. Some of security measures especially the one that must start from hardening the DNS hosting environment are given below.

Host hardening, DNS server should be harden keeping in view of OS, DNS placement and services. Only DNS program is installed on DNS host and all other ports should be closed. On UNIX machine one can check running services and open ports by using the command "nmap localhost".

Always use Updated software for DNS or update DNS program as often as possible to limit software bugs.

DNS placement, to avoid single point of failure, name servers should not be placed on same subnet, behind same router or even the leased line. It is recommended that organization must have multiple DNS servers and put on multiple locations. Disable recursion on master or primary DNS and limit or restrict recursive queries on slave servers to prevent spoofing. In BIND version 9.5 and prior version recursion is enabled by default but one can disable and restrict by modifying BIND configuration file as mentioned below. Restricting

```
//Disable recursion for Primary DNS
  options {
        recursion no;
  };

//Tursted Network Subnet 172.18.18.0/16
  options {
  allow-recursion {171.18.18.0/16;};
  };
```

Fig. 4. Disabling recursion feature

queries, allow possible queries from allowed possible hosts for minimum query as sample configuration is given below in BIND. In BIND, query version is used to return the software

```
options {
            allow-query {172.18.18.0/16;};
};
```

Fig. 5. Setting allow-query parameter

version. This feature is vulnerable if attacker is looking for a specific version of BIND with discovered flaws. Therefore DNS administrators should configure to hide type of software that is in use and its version. To configure BIND will refuse

this query following configuration is made in configuration file.

```
options {
        version none;
   };
```

Fig. 6. Disabling DNS software (bind) version.

Another most recommended best practice is to restrict the zone transference and updating to only approve slave servers or even if there is not a requirement of frequent dynamic change in zones then allow-transfer feature can disable after successful synchronization of zones and can restricted to allowed slave servers through ACLs as exhibited below. One of

```
acl slaves {
        203.82.49.35;
        96.43.23.35;
        114.43.24.34;
        };
```

Fig. 7. Restricting zone data to only allowed slave servers.

the best practices is to segregate external and internal queries on DNS server. Split-Service will split the DNS server into two compartments, one is used for resource records name resolution for its authoritative domain zones and other part will resolve the queries receive from its internal trusted hosts. In this case, if external part of DNS will hack, the DNS service will not affect internal hosts.

In BIND version 9.10 [6] introduces DNS Response-Rate Limiting (DNS RRL) feature to prevent DoS and DDoS attacks against DNS. In which token bucket is allocated to a name server. In this every response consumed a token, say 5 token to particular response to particular NS, when a token bucket becomes empty for particular response to concern NS is limited, so this limitation is not sufficient to launch a DoS or DDoS attack. The DNS RRL watches that over and over high rate of requests come from same IP address is an attack, so it stopped. This is a new technique adopted in BIND version 9.10 an patch for version 9.9 is also available. Recently ISC has renamed project Bundy [7] after release of version 1.2 of Bind 10 software to dedicate this project for further DNS application framework with integrated DNS and DHCP server [8] which supports IPv6 and refined security features. Configuring DNS security on other network devices is also suggested, sample output for Cisco firewall [9] is mention below and same should be configure for other firewalls.

```
fw# configure terminal
fw(config)# dns-guard
fw(config)# exit
```

VI. DNS WITH SECURITY EXTENIONS (DNSSEC)

By summarizing discussed attacks here with respect to CIA triad, they fall into Integrity and Availability categories. So we need authenticated mechanism where these two goals will be achieved to address discussed DNS vulnerabilities. For this purpose here we introduced public key cryptographic framework using Bind DNS opensource software on Linux platform. In this framework DNS security extensions are used to apply digital certificates to DNS data, which has been successfully implemented and tested in IS department. There are other cryptographic mechanisms available like DNSCurve [10] which used Elliptic Curve Cryptography. OpenDNS software has adopted and launched support for DNSCurve back in 2010 [11] but still most of the DNS servers are static without having authenticated DNS traffic.

Proposed mechanism adds asymmetric cryptographic RSA keys to DNS, it digitally signs DNS data, respective name server (NS) that supports validating that digitally signed data belongs to domain originator and proof the authenticity of that data by protecting the integrity of data, it actually able to sign by who supposed to be responsible for that domain. Before going into the practical implementation of DNSSEC, its complete understanding is essential for successful roll out and follows the best practices required for intended organizations. **DNSSEC** [12] has two tails one is signing and other is validation for making the cryptographic chain of trust. As far as practical deployment is concern, in lab we developed discussed framework by incorporating the BIND DNS software with RSA and SHA-256 algorithm to digitally signed DNS data. Because of hierarchical mapping of DNS architecture, a trust relationship begins from the top of root [13] towards down leveled TLDs, then to second level domain (”icann.org”) and so on to build the chain of trust. A traditional DNS is called static DNS that is robust but we have seen it have many flaws with its robustness. The security extensions to apply digital certificates to DNS guaranteed higher integrity but with decreased robustness. DNS with security extensions is technique that requires expertise, skills and precision. To ensure its successful deployment, following are prerequisites. TLDs must be signed, a large number of top level domains like .com, .org, .gov etc all have signed and numerous country code TLDs as well like .uk, .ca, .cn, .in etc. Complete list of signed TLD can be obtained from ICANN website [14].
Domain Registrar and hosting provider must support DNSSEC, it is essential that they must be in position to accept and sign Delegation Signer (DS) which contains the necessary information about keys used to sign the zone. Registrar then forwards this DS record to parent domain (typically a TLD). The operating model of DNSSEC is quite different from traditional DNS. In DNSSEC signs are valid for certain time period and need to be re-signed before expiration. Some organization’s security policy requirement is to change the keys with the passage of time and if changes were not plan properly, then keys can be compromised or lost. For handling lost or compromised keys, key rollovers activity has to launch in addition to signatures. Before going into DNS security extensions implementation for production environment, let’s

have a looked on operations required for DNSSEC implementation to each zone.

- Generation of Keys
- Signing records using keys
- Publishing signed zone
- Managing the each KSK (Key Signing Key) rollover with the publication of key summary in parent zone.
- Checking and monitoring the assigning of new key before signing it.

Precise and proper planning is essential for successful deployment of DNSSEC and it's fundamental. Prerequisite are mention to monitor and make necessary changes to obtain optimum level of accuracy in DNS configuration.

1) Synchronizing clock is mandatory element in DNSSEC, as signatures are valid for specific time slot having start and end date. So clock synchronizing is best practice, but for DNSSEC if not properly handled consequences will be severe. For centralized control and uniform clock synchronizing it can be manage through NTP along with monitoring application of NTP.
2) Securing storage of DNSSEC key is one of the prime objectives, like other cryptographic systems DNSSEC private keys have to be secure. So, if not properly secured and attacker have approach to access these keys then whole security of DNSSEC will become vulnerable. So recommended approach is to use Hardware Security Module a specialized system to generate and store cryptographic keys with primary and secondary mode. If organization cannot afford HSM then keys must be stored under lock by saving it on USB or other media.
3) To achieve a reliable DNS environment, monitoring is a key for prompt triggering of any falsie change occurred in DNS system like a server may be crash or firewall may be blocking legitimate requests etc. For DNSSEC if server signatures are outdated, then whole zone will be consider invalid. So, it's necessary to check all servers are responding with updated signatures and signatures are not about to be expire.
4) DNSSEC Key Management has two type of keys ZSK (Zone Signing Key) and Key Signing Key (KSK). If DNSSEC is managed manually then acceptable approach is to use single key. Alternatively if automated key management tool like OpenDNSSEC is used, then key pair can be use, an automated tool will take care of everything.
5) As far as practical aspects of cryptography are concern, the choice of using NSEC3 with salt value is preferred which make more difficult zone contents counting attacks and dictionary attacks while zone walking. Finally, choice of cryptographic algorithm, choosing RSA plus SHA256 is recommended.

VII. PROTOTYPE CRYPTOGRAPHIC DNS FRAMEWORK

DNS Security Extensions can be deployed on wide range of operating systems and DNS software. In this paper we have chooses most widely used open source platform, UNIX. We have tested these set of security extensions using BIND version 9.9 software.

In this framework we only illustrate DNS cryptographic keys generation and necessary configuration, as other name server configuration secure settings have been demonstrated in best practices initial section of this paper.

Generating ZSK (Zone Signing Key) and KSK (Key Signing Key) pair using RSA + SHA256 algorithm with 1024 and 2048 entropy bits respectively.

```
[root@ns1 named]# dnssec-keygen -a RSASHA1 -b 1024 -n ZONE mcs.edu.pk
Generating key pair.....................................................
.....................++++++ ..............................++++++
Kmcs.edu.pk.+005+55227
[root@ns1 named]# ls
192.168.228.db  dynamic                     mcs.edu.pk.db  named.localhost
chroot          Kmcs.edu.pk.+005+55227.key  named.ca       named.loopback
data            Kmcs.edu.pk.+005+55227.private  named.empty   slaves
[root@ns1 named]# dnssec-keygen -a RSASHA1 -b 4096 -n ZONE -f KSK mcs.edu.pk
Generating key pair.....................................................
.....................................................................
.....................................................................
.....................................++ ..............................++
Kmcs.edu.pk.+005+05779
[root@ns1 named]# ls
192.168.228.db  Kmcs.edu.pk.+005+05779.key  mcs.edu.pk.db  named.loopback
chroot          Kmcs.edu.pk.+005+05779.private  named.ca      slaves
data            Kmcs.edu.pk.+005+55227.key  named.empty
dynamic         Kmcs.edu.pk.+005+55227.private  named.localhost
```

Fig. 8. Generating ZSK and KSK RSA Asymmetric Key pairs.

Signing the zone file

Here we have used NSEC3 with salt `(head -c 1000 /dev/random | sha1sum | cut -b 1-16)` to make dictionary attacks more difficult and it creates a new file "db.icann.org.signed" which have RRSIG (Resource Record Signature) digital signature for each DNS record. Directing zone to use this signed file in a bind configuration. Initial signing process and configurations completed here, the default

```
Verifying the zone using the following algorithms: RSASHA1.
Zone signing complete:
Algorithm: RSASHA1: KSKs: 1 active, 0 stand-by, 0 revoked
                    ZSKs: 0 active, 1 stand-by, 0 revoked
mcs.edu.pk.db.signed
Signatures generated:                       14
Signatures retained:                         0
Signatures dropped:                          0
Signatures successfully verified:            0
Signatures unsuccessfully verified:          0
Signing time in seconds:                 0.144
Signatures per second:                  96.641
Runtime in seconds:                      0.194
[root@ns1 named]# ls -lh
total 76K
-rwxr-xr-x. 1 root  named  464 Mar 28 22:46 192.168.228.db
drwxr-x---. 6 root  named 4.0K Mar 28 22:20 chroot
drwxrwx---. 2 named named 4.0K Mar 30 03:28 data
-rw-r--r--. 1 root  root   163 Apr  1 10:08 dsset-mcs.edu.pk.
drwxrwx---. 2 named named 4.0K Mar 31 22:29 dynamic
-rw-r--r--. 1 root  root   948 Apr  1 10:03 Kmcs.edu.pk.+005+05779.key
-rw-------. 1 root  root  3.3K Apr  1 10:03 Kmcs.edu.pk.+005+05779.private
-rw-r--r--. 1 root  root   429 Apr  1 10:02 Kmcs.edu.pk.+005+55227.key
-rw-------. 1 root  root  1010 Apr  1 10:02 Kmcs.edu.pk.+005+55227.private
-rwxr-xr-x. 1 root  named 1.9K Apr  1 10:04 mcs.edu.pk.db
-rw-r--r--. 1 root  root   14K Apr  1 10:08 mcs.edu.pk.db.signed
-rw-r-----. 1 root  named 1.9K Feb 18  2008 named.ca
-rw-r-----. 1 root  named  152 Dec 15  2009 named.empty
-rw-r-----. 1 root  named  152 Jun 21  2007 named.localhost
-rw-r-----. 1 root  named  168 Dec 15  2009 named.loopback
drwxrwx---. 2 named named 4.0K Jan 20 09:40 slaves
```

Fig. 9. Signing Zone files.

```
zone "icann.org" IN {
        type master;
        file "db.icann.org.signed";
        allow-transfer { slaves; };
        allow-update { slaves; };
        notify yes;
   };
```

Fig. 10. Pointing domain zone to use digitally signed file.

validity for zone signatures is 30 days after this it expires. Zone file should be re-sign before its expiration, zone resigning process has to plan with respect to TTL value set in DNS server. Re-signing process is automated by creating a monthly cron job and placed in /etc/cron.monthly/, this will re-sign zone file automatically on every month. After completing the signing process, next to provide validation to these signatures from registrar end, for this purpose forward DS (Delegation Signer) records to registrar. The one more file "dsset-db.icann.org" apart from .signed file, which contains DS records. Sample output of DS record file is shown below. The above DS

```
db.icann.org.           IN DS 5779 5 1 896CD9F5DADDDA
95C39FCCEBEC1951F990AFA99D
db.icann.org.           IN DS 5779 5 2 8971AD035AED97
0CCDD6AD6F13ABF7A04B92D159A28ACF5003069556 C73514F7
```

Fig. 11. Content of DS records

records need to be forward for registration at concern registrar. Most registrars provide control panel for registering these records. Once these records have been saved, it takes some time to propagate. To check successful implementation of DS records, query name server of TLD or query any open DNS resolver like google as shown below. Once confirmed that

```
[root@ns2 named]# dig +trace +noadditional DS db.icann.
org. @8.8.8.8 | grep DS
```

Fig. 12. Pointing domain zone to use digitally signed file.

everything is synchronized and working, as mentioned earlier thorough testing and monitoring is essential for DNSSEC and one cannot afford leniency specifically in DNSSEC. Here are some online web based tools for testing the performance of DNSSEC and continuous monitoring of its health. http://dnssec-debugger.verisignlabs.com [15] http://dnsviz.net [16]

VIII. CONCLUSION

This document is meant to understand existing DNS security challenges and demonstrates best practices for secure deployment of DNS using digital signature for DNS data. As discussed DNS with Security Extensions of digital signatures is an ultimate solution till now for securing DNS infrastructure, which is not only beneficial for end users, but it also provide safety to overall internet by protecting DNS data with authentication and encryption end to end in making a hierarchical chain of trust in domains to avoid from forgery. With reference of NIST documentation [1] and discussed DNS vulnerabilities deploying DNSSEC based framework is the essential requirement. Deploying and managing DNSSEC is a technique which requires proper planning, skillful DNS administrators and prcised approach. Therefore to ensure DNSSEC proper deployment and its understanding, an effort is made in this paper to discuss security challenges of DNS and also demonstrated cryptographic framework with industry based best practices to deploy a secure DNS infrastructure with DNSSEC. It is suggested to implement DNSSEC based trusted DNS infrastructure, specifically for those domains whose parent domain have already signed and support DNSSEC. For DNS chain of trust as discussed TLD must be signed and support DNSSEC. Future work is required to enhance DNSSEC support in devices and strategy should adopt for deploying DNSSEC on TLDs domains which are still not signed with DNSSEC to form global DNS chain of trust, so that internet communication could be secure from forgeries at large scale.

REFERENCES

[1] R. Chandramouli, S. Rose Author, *Secure Domain Name System (DNS) Deployment Guide* in 800-81-2 REV Sep, 2013. NIST
[2] IETF Working Group Document on DNS-based Authentication of Named Entities
[3] IEEE Research Paper "Prevent DNS Cache Poisoning Using Security Proxy" Oct, 2011
[4] Master Thesis, Christoph L. Schuba, Purdue University West Lafayette, IN.
[5] SANS Whitepaper: Security Issues with DNS.
[6] Internet Systems Consortium (ISC) Knowledge Base BIND 9.10
[7] ISC Concludes Bind 10 release 1.2 renamed project Bundy 2014.
[8] Integrated authoritative DNS and DHCP server, BUNDY Nov, 2014.
[9] Cisco Whitepaper: DNS Best Practices, Network Protections, and Attack Identification
[10] Usable Security for DNS, DNSCurve
[11] ISC document, OpenDNS adopted DNSCurve to secure DNS
[12] IANA DNSSEC DNS Security Algorithm Numbers RFC, Rev Mar, 2014.
[13] Root-Anchor, ICANN Trust anchor for the root zone of Domain Name System.
[14] ICANN Research. List of Top Level Domains (TLDs) DNSSEC Report. May, 2014.
[15] US VeriSign Labs Tool, DNS security analysis and reporting.
[16] US Sandia National Laboratories, DNSviz, A DNS Visualization Tool