UPGRADING AND SECURING EXTERNAL DOMAIN SPACE IN THE CITY OF NIŠ ADMINISTRATION INFRASTRUCTURE

Đorđe Antić,
Mladen Veinović

Singidunum University,
32 Danijelova Street, Belgrade, Serbia

Abstract:
Being the key Internet infrastructure component, DNS (Domain Name System) is vital for any organization that requires external visibility and availability of its services. Public administration is especially sensitive, due to the nature of services offered to citizens. This paper describes the strategy, implemented to the administration of the city of Nis, for making external domain space more robust and resilient, as well as securing it with DNSSEC (DNS Security Extensions).

Key words:
DNS, security, cryptography, DNSSEC.

1. INTRODUCTION

Domain name system (DNS) is a distributed hierarchical database, operating as a mechanism for mapping hostnames to IP addresses. All Internet services rely on DNS as an infrastructure, making it essential and fundamental. Although robustly designed and improved over years, security was never its strong point. Most notable problems are DNS client flooding (a denial of service attack)[1] and cache poisoning, which makes it possible to insert false information into the cache of a DNS resolver, as was made widely known in 2008[2].

In order to improve the security of the system, the Domain Name System Security Extensions (DNSSEC) were introduced. It is a security protocol based on public-key cryptography, using asymmetric cryptography to generate digital signatures of data in DNS[3]. Through these signatures, resolving clients can be provided with origin authentication, data integrity and authenticated denial of existence. Signatures are following the hierarchical model of DNS architecture, forming a chain of trust from the root zone to all levels of subdomains.

The City of Niš administration relies on DNS for its presence on the Internet through various online services offered to citizens. Although not visible to end users, name resolution plays a critical role. Any disruptions can render the services unavailable or can, through abuse, provide false or misleading information to citizens, which can result in legal issues or material damage. This makes the need for reliable and secure DNS system even more emphasized.

Correspondence:
Đorđe Antić

e-mail:
djordje.antic@gmail.com

DOI: 10.15308/Sinteza-2016-148-153

2. BACKGROUND AND SET GOALS

Background

DNS is a distributed database, deployed on name servers, linking domain names with IP addresses and other data. The data is organized hierarchically, similar to the structure of a tree. Root domain is on top, as shown in Fig. 1. The domain name system tree is divided into zones (such as .com, .net, .org). Zones are the sections of the tree delegated to a single administrative authority. Each zone is required to have multiple authoritative name servers that provide name resolution for all parts of domains contained within.

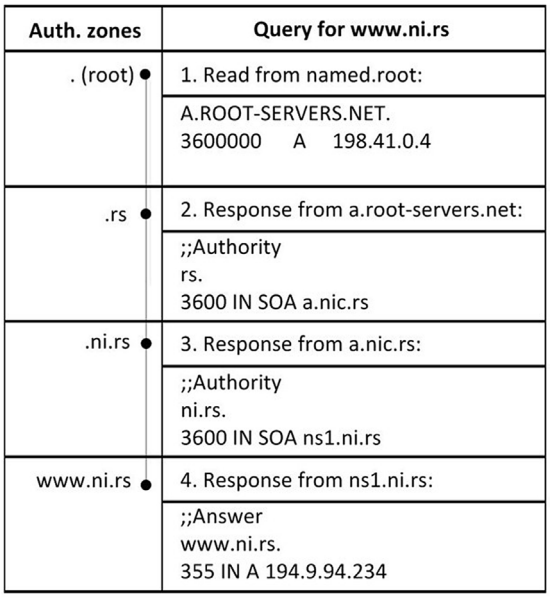

Fig. 1. DNS hierarchy

DNS for ni.rs zone

In 2015, it was determined that the current DNS infrastructure of the city of Nis administration is in need of an audit and upgrade. The task was aimed at identifying the current state, weak points and shortcomings, setting the desired goals and proposing and implementing the appropriate solutions.

At the start of the project, current state of the DNS was reviewed. It was determined that the ni.rs zone (along with a number of others) was served by a pair of Networks Defender ND410 appliances, acquired in 2007, which were also used as antivirus scanning points for HTTP traffic. Internally, DNS service was provided by BIND 8, deprecated as of August 2007. Both of them were 9 years old, with no support or warranty or new antivirus updates. One device was put permanently offline due to hardware failure. The other device experienced occasional software problems that caused it to stop answering queries (rendering the zones it was authoritative for unreachable). This made the ni.rs and other zones hosted unreliable to reach, which was a problem emphasized by the fact that it was also the location where domains of various online services provided by the city administration were hosted. A solution was needed to replace the existing appliances and several goals for the project were defined.

Goals set

1. Any solution must take into consideration limited funds available for the project. Solution should be cost-efficient but with making as few compromises as possible.
2. Solution must follow best current industry practices for operation of authoritative name servers. Proper configuration and maintenance of name servers should have critical part in the project.
3. Special consideration should be given to security issues. DNSSEC extensions should be part of the solution.

3. DESIGN AND IMPLEMENTATION

In order to make the solution cost effective, decision was made to use the existing capacities on two older server machines with hypervisors and create name servers as virtual machines. Servers were both dual Xeon with 24GB RAM, both running VMWare ESXi hypervisors.

The software choice for virtual name servers was based on stability, security, hardware requirements and total cost of ownership. Linux was the obvious choice for operating system, and for diversity reasons, two different distributions were chosen, CentOS (Red Hat based) and Ubuntu (Debian based). Both distributions were installed in their minimal server variants.

The name server software was chosen among currently most widespread implementations. ISC's BIND was the first choice, being the industry standard. Others were NLnet Labs's NSD, CZ.NIC's Knot and EURid's YADIFA, all being high-performance authoritative only name server implementations. The most recent versions of the software were installed, with support for all important DNS

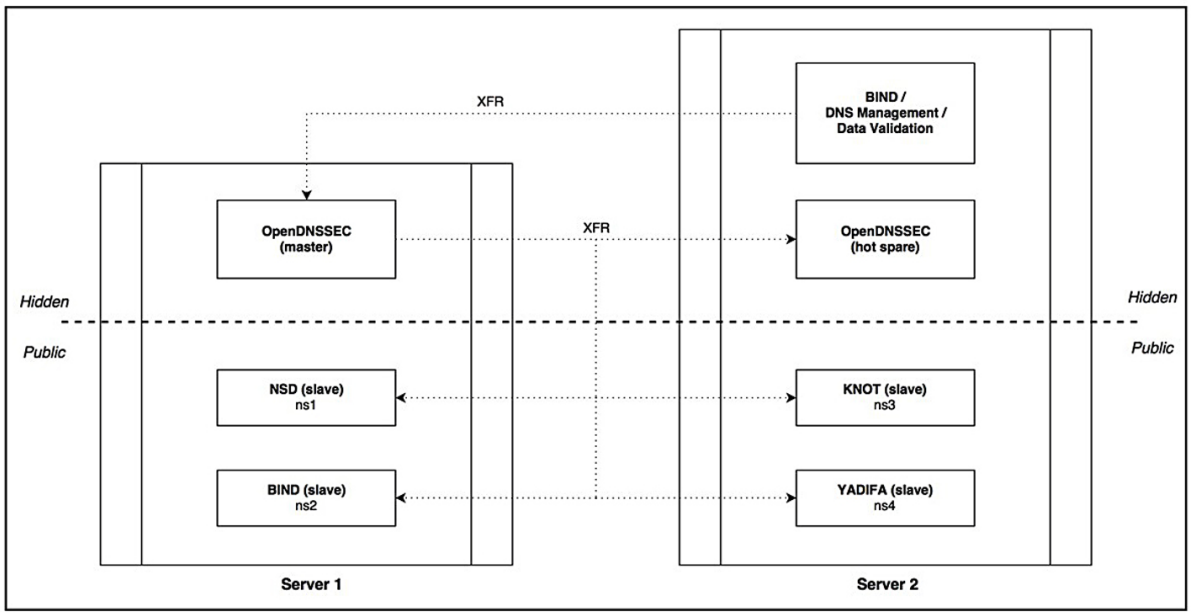

Fig. 2. DNS concept with hidden master and bump-in-the-wire DNSSEC signing

protocols, such as full and incremental zone transfers, dynamic updates, EDNS0[4] and DNSSEC extensions with NSEC3[5], response rate limiting[6] and NSID[7].

The concept, illustrated in Fig. 2, was to have one hidden server where zone data would be prepared, validated and loaded to a master server, running BIND. This server would then notify DNSSEC signing machine (running OpenDNSSEC) about the change in zone data. OpenDNSSEC machine would then initiate a zone transfer from the master server using an encrypted zone transfer (AXFR/IXFR). This machine works as a "bump-in-the-wire" between editing and publishing a zone. Transferred zones would then be signed with appropriate cryptographic keys. Keys are stored in a software implementation of a cryptographic store accessible through a PKCS#11 interface, SoftHSM, which is developed as part of the OpenDNSSEC project. After the zones have been successfuly signed, slave servers are notified about the zone change. The slave servers (public name servers, running NSD, BIND, Knot and YADIFA) would then initiate zone transfers, again using an encrypted zone transfer. All chosen name server implementations on slave servers use the same format of BIND's master zone file. Virtual DNS server machines are split among two physical servers.

Configuration of public servers

General recommendations for operation of public name servers have been implemented:

- Servers are running on virtual machines dedicated to DNS. This minimizes the risk of unauthorized access or negative impact of other applications on DNS. It also enhances the capability to monitor server performance or troubleshoot problems.
- DNS software is running as an unprivileged user
- Access control mechanisms are set to restrict zone transfers capability to master server only. Transfers are secured with HMAC-SHA256 TSIG.
- Recursion queries are not allowed, since servers are authoritative-only. Recursive servers intended for internal clients exist on separate part of infrastructure.
- Time to live (TTL) values of NS records and their associated A and AAAA records are set long enough to help reduce the impact of DDoS attacks, as recommended in [8].
- Response Rate Limiting (RRL) with appropriate values is deployed on servers.

Response Rate Limiting

Response rate limiting (RRL) is an enhancement that helps mitigate DNS amplification attacks. DNS amplification attack is a type of reflection attacks, in which an attacker sends traffic to the victim by reflecting it off a third party, effectively concealing his identity. Amplification is combined into this attack when the amount of traffic the victim receives is considerably larger than the amount of traffic sent by the attacker.

DNS servers are often misused for this type of DDoS attack because of the protocol characteristics. UDP (User Datagram Protocol) protocol is suitable for this purpose because it is relatively easier for an attacker to spoof his IP address over UDP (there is no source validation) than it would be over TCP protocol. As DNS replies can be significantly larger than a DNS query, an attacker can spoof a small query for which he knows will generate a large answer. Sending many queries in this manner to a large number of "open" DNS resolvers can generate enormous traffic directed to the victim. Target is flooded with unrequested DNS query responses, and although they are discarded on arrival, they have already consumed network resources, potentially rendering the target unavailable.

RRL mitigates this type of attack by limiting the rate at which servers respond to large number of malicious queries. RRL can detect patterns in queries that are received and, according to set parameters suggesting abuse, reduce the rate at which the replies are sent. Along with making the attack lose bandwidth, RRL decreases the attractiveness of the DNS system as DoS amplifier.

DNSSEC

DNSSEC introduces four new resource records: RRSIG (Resource Record Signature), DNSKEY (DNS Public Key), DS (Delegation Signer) and NSEC (Next Secure). RRSIG is a digital signature produced by hashing and RRset and encrypting it with a private key for a zone. That key is then published as a DNSKEY RR. DS RR, which resides with the parent zone, represents a hash of the DNSKEY of the child zone. DS RR is a point of delegation between the zones, which can be authenticated, because it works as a form of "certificate", binding the child zone with the parent. These relationships form a chain of trust that a resolver can follow through the DNS tree (as in Fig. 3).

DNSSEC in ni.rs zone has been implemented as a "bump-in-the-wire" between the hidden master and publicly visible servers. This has allowed for a gradual setup on dedicated machines, implementing DNSSEC only after the rest of the DNS system (without DNSSEC) has been put online and operating as expected.

OpenDNSSEC implementation has been chosen for several main reasons:

- Seamless integration into existing non-DNSSEC DNS environment without changes to the current model.
- High level of automation. When it is set up, no manual intervention is needed, but still possible if necessary (for example, in case of emergency key rollover). Also, since DNSSEC requires that certain number of procedures be performed in a strict timeframe, higher automatisation reduces chances of errors.
- Security – support for HSM. Current setup uses software emulation of HSM (SoftHSM) in order to avoid cost, but OpenDNSSEC can also use hardware HSM using industry standard PKCS#11 interface.

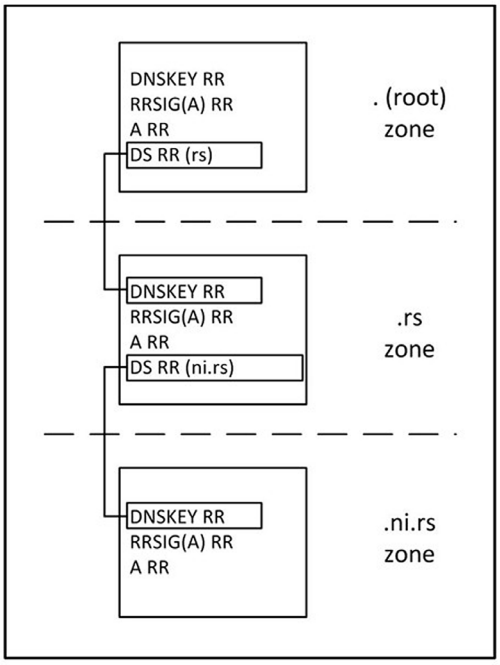

Fig. 3. DNSSEC chain of trust

Configuration of DNSSEC was done according to the practices described in [9]:

- Keys are operationally separated to have a role of Key Signing Keys (KSK) and Zone Signing Keys (ZSK).
- Zone Signing Key size is set to 1024 bits while the Key Signing Key size is set to 2048 bits.
- Algorithm used for KSK and ZSK is RSA/SHA-256, as referenced in [13] (algorithm number 8 per IANA registry[14])
- Maximum validity period of signatures is 14 days (both KSK and ZSK), with inception time of one hour.
- Resign interval (runs of signer engine) is 2 hours, with refresh interval (time after signature is refreshed) of 3 days.

- TTL values of signature resource records match the TTL values of the RRsets they cover, as recommended in [15].

DNSSEC Key rollover can take place in the event of compromise of existing keys or in case of policy demands. Two strategies for key rollover are implemented:

- For ZSK, Pre-Publication method is used, as recommended in [10]. New key is introduced to DNSKEY RRset, which is then resigned. After sufficient time, when all cached RRsets are considered to contain both keys, signatures created with old key are removed. Again, after sufficient time, after signatures created with old keys have expired from caches, old key can be removed from DNSKEY.
- For KSK, Double-Signature method is used. New KSK is generated and DNSKEY record for new key is added to the zone. Key is then sent to parent zone, and parent replaces old DS record with a new one. After sufficient time, when all cached RRsets are considered to contain new DS record, DNSKEY record for old KSK can be removed. Although [10] recommends Double-RRset as the most efficient for KSK rollover due to the ability to have new DS records and DNSKEY RRsets propagate in parallel, this method is not yet supported in OpenDNSSEC.

NSEC3

One of the things that DNSSEC provides is authenticated denial of existence, which is a mechanism that can prove that domain names and resource records do not exist. This is achieved by listing of all domain names and resource records that do exist and securing them with NSEC. However, this introduced the zone enumeration issue, which can allow an attacker to gather all domain names in a zone. To prevent this scenario, NSEC3 is used.

NSEC3 creates hash of each name in a zone and links these hashed names. Any query for these hashed names will give back a response stating that the requested name does not exist. Queries directed to names that do not exist will receive the same answer, as it can be proven that there is no hash record for them.

There are three main configuration parameters for NSEC3:

- Opt-Out mechanism: Since the ni.rs zone is relatively small and contains no insecure delegations, opt-out mechanism is not used.
- Iterations: This parameter is used to counter the brute-force breaking. Number of iterations is set according to recommendations in [5]. The limits are 150 for key size of 1024 bits and 500 for key size of 2048 bits.
- Salt: Used to prevent creation of a rainbow table. Salt size is set according to recommendations in [5], *i.e.* at least 64 bits long. It is worth noting here that according to the study[12], NSEC3 salt is ineffectual and inadequate. Since "the value of the salt is publicly accessible via DNSSEC RR lookup…any attacker may obtain the salt to use as input into its dictionary computation, effectively negating the required increasing in dictionary size."

NSEC3 TTL value is identical to SOA minimum TTL value, as recommended in [5].

4. FUTURE WORK

Anycasting

Anycast is a network methodology in which traffic is routed from a single source to several topologically dispersed targets using the same IP address. Layer 3 routing is used to send packets to the nearest server in the anycast group.

Adding anycast servers is planned as the next future upgrade of the name server infrastructure described here, as the benefits for using anycast for DNS servers are increased reliability, load balancing, improved performance, better protection from DoS and increased availability. The tradeoffs are complexity, cost and increased difficulty in troubleshooting and monitoring. Support for NSID by all implemented server software should help with anycast deployment.

DNSSEC Policy and practice Statement (DPS)

DPS is a document, written according to recommendations in [11], that describes the policies and procedures relevant to DNSSEC that have been implemented. The document should „provide a means for stakeholders to evaluate the strength and security of the DNSSEC chain of trust...comprising statements describing critical security controls and procedures relevant for scrutinizing the trustworthiness of the system"[11].

It is planned to prepare and publish this document as it should help with understanding of everything that has been done to secure our zone. It can be significant for all stakeholders, including regulatory authorities. It will also serve the purpose of helping people learn about

the security implemented in our zone and decide if they can trust it. Other implementators may find it useful for planning all significant aspects of using DNSSEC.

DANE

DNS-based Authentication of Named Entities (DANE) is a method of binding X.509 certificates to DNSSEC secured domain names, with the purpose of using the secure DNS infrastructure to „store and sign keys and certificates that are used by TLS (Transport Layer Security)" as described in [16].

DANE is another feature planned for testing and implementation in our zone, as it provides a potential alternative to trust currently placed in commercial Certificate Authorities and offers a standard for encrypted email, as described in [17].

5. CONCLUSION

DNSSEC is still in the test phase for our zone and our parent .rs zone has not yet been signed at the time of this writing. Thus, being still unable to verify DNSSEC operation in environment with established chain of trust, it is early to say that DNS setup and configuration presented here are final or fully optimized. All efforts were made to follow industry standards as well as recommended best practices. Experience gained during the project will serve to further improve DNS for our zone and other implementators may benefit from data presented herin.

Acknowledgment

The authors would like thank to Mr. Žarko Kecić, CTO of Serbian National Internet Domain Registry for his help and guidance.

REFERENCES

[1] Moti Geva, Amir Herzberg, Yehoshua Gev, "Bandwidth Distributed Denial of Service: Attacks and Defenses", IEEE Security & Privacy, vol.12, no. 1, pp. 54-61, Jan.-Feb. 2014, doi:10.1109/MSP.2013.55

[2] Alexiou, N.; Basagiannis, S.; Katsaros, P.; Dashpande, T.; Smolka, S.A., "Formal Analysis of the Kaminsky DNS Cache-Poisoning Attack Using Probabilistic Model Checking," in High-Assurance Systems Engineering (HASE), 2010 IEEE 12th International Symposium on , vol., no., pp.94-103, 3-4 Nov. 2010

[3] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, "DNS Security Introduction and Requirements", RFC 4033, Internet Engineering Task Force, March 2005.

[4] J. Damas, M. Graff, P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", RFC 6891, Internet Engineering Task Force, April 2013.

[5] B. Laurie, G. Sissons, R. Arends, D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, Internet Engineering Task Force, March 2008.

[6] T. Rozekrans, J. de Koning, "Defending against DNS reflection amplification attacks", University of Amsterdam, February 2013.

[7] R. Austein, "DNS Name Server Identifier (NSID) Option", RFC 5001, Internet Engineering Task Force, August 2007.

[8] V. Pappas, E. Osterweil, "Improving DNS Service Availability by Using Long TTL Values", draft-pappas-dnsop-long-ttl-04, Internet Engineering Task Force, February 2012.

[9] O. Kolkman, W. Mekking, R. Gieben, "DNSSEC Operational Practices, Version 2", RFC 6781, Internet Engineering Task Force, December 2012.

[10] S. Morris, J. Ihren, J. Dickinson, W. Mekking, "DNSSEC Key Rollover Timing Considerations", RFC 7583, Internet Engineering Task Force, October 2015.

[11] F. Ljunggren, AM. Eklund Lowinder, T. Okubo, "A Framework for DNSSEC Policies and DNSSEC Practice Statements", RFC 6841, Internet Engineering Task Force, January 2013.

[12] Bau, J. and Mitchell, J.C., 2010. A Security Evaluation of DNSSEC with NSEC3. IACR Cryptology ePrint Archive, 2010, p.115.

[13] J. Jansen, "Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC", RFC 5702, Internet Engineering Task Force, October 2009.

[14] http://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml

[15] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, Internet Engineering Task Force, March 2005.

[16] P. Hoffman, J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, Internet Engineering Task Force, August 2012.

[17] V. Dukhovni, W. Hardaker, "SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS)", RFC 7672, Internet Engineering Task Force, October 2015.