CIA Lab Assignment:
DNS Security Extensions (DNSSEC)*

A. Safin K.Saltabov R.Hussain

Feedback deadline:
September 26, 2016 00:00 CET

Abstract

In this lab assignment, you will set up a DNSSEC-validating DNS resolver, use DNSSEC to secure your own zone, and learn how to do key maintenance.

1 Introduction

DNS has been designed without security in mind. Since security researcher Dan Kaminsky demonstrated the ease of DNS cache poisoning, DNSSEC, a DNS security extension, has become popular. DNSSEC is an open standard, documented in various RFCs that adds cryptographic protection to DNS making it hard to forge DNS replies.

Before the DNS root servers and the Dutch ccTLD got signed, people had to work with "islands of trust". Nowadays, we can use DNSSEC by knowing just the DNSSEC keys of the root servers. If you are interested in legacy DNSSEC operation, "islands of trust" are nice to experiment with.

2 Setting Up A Validating Resolver

In order to get familiar with DNSSEC, we first set up a DNSSEC-validating resolver. You can use the DNS server installation from the previous assignment.

1. What does a validating resolver do?
2. Add support for DNSSEC to your BIND or Unbound configuration.
   (a) What changes do you have to make to your configuration?
   (b) Verify the root key used against a trusted source.

Since the DNS root servers have been signed, "native" DNSSEC validation is possible for spme domains.

*Version August 25, 2016.
This material based on prev version of document, prev doc prepared by A. Bakker N. Sijm J. van der Ham M. Pouw†

Use *dig/devl* or *drill* to verify the validity of DNS records for isc.org and os3.nl. Show the results.

3. How does `dig/drill` show whether DNSSEC validation was successful or not?

The only way to really test your resolver is by breaking the chain of trust. In more recent versions of BIND, this is not as simple as you would think it is.

4. Where does BIND/Unbound store the DNSSEC root key?
5. How do "managed keys" differ from "trusted keys"? Which RFC describes the mechanisms for managed keys?

Modify the DNSSEC root key of your installation and try to validate the chain of trust again.

6. How did you modify the DNSSEC root key?
7. What problems did your server encounter, and how did it react?

3 Setting Up A Secure Zone

Adding DNSSEC to an authoritative nameserver is a delicate process. First one has to decide upon the chain of trust: does the parent zone offer secure delegation, or do I create an "island of security"? Cryptographic algorithms, key sizes, key lifetimes and key rollover schemes have to be chosen, generated and configured.

9. Look up which cryptographic algorithms are available for use in DNSSEC. Which one do you prefer, and why?

Although the specification does not require it, all major DNSSEC tools use two different kinds of keys: a *key-signing key (KSK)* and a *zone-signing key (ZSK)*. When a ZSK changes, the parent zone is not involved. When the KSK changes, the parent zone needs to be involved in order to maintain the chain of trust.

10. In practice, different algorithms, key sizes and key lifetimes are chosen for KSKs and ZSKs. Discuss what are these differences in:
    (a) algorithms
    (b) key sizes
    (c) key lifetimes

    and give the motivation for the difference.
11. Choose appropriate algorithms, key sizes and key lifetimes for your KSK and ZSK.

There are various tools that can generate keys and sign zones. If you are running BIND, you should use the BIND9 tools. Use `dnssec-keygen` to generate the KSK and ZSK for your zone and `dnssec-signzone` to sign your `<stX>.stY.os3.su` and `stX.os3.su` zones. If you are running NSD+Unbound, use the `ldns` tools from NLnet labs, in particular, `ldns-keygen` and `ldns-signzone`.

12. Show the signed version of your zone file. How does it differ from the unsigned version? Any unexpected differences?

Edit the BIND/NSD configuration to include the signed version of your zone file. Restart the authoritative server and look at the syslog for errors. If the server appears to be up and running, test DNSSEC by querying your server for the DNSKEY of `stX.os3.su`.

Unfortunately, we can't create full chain of trust, because nameserver used for os3.su domain is not support DNSSEC. We will create island of security/trust by adding DS records for `stY.stX.os3.su` domain in `stX.os3.su` zones.

13. Which DS record do you need to send to your neighbors, and why that one?

    Once your partner has implemented your DS record, use a DNSSEC debugger to exam- the chain of trust, how you can be sure that you successfully create signed chain. Is it possible? What are you needed to make it possible.

    For example: you can visit http://dnssec-debugger.verisignlabs.com/ (use 'more detail') or http://dnsviz.net/ and check os3.nl domain.

14. Show the results of the examination of your secured domain.

15. Describe which DS and DNSKEY records are important for your domain. Which keys are used where?

4 Key Rollovers

At some point it may be necessary to change one or more of your keys that you use for DNSSEC. Fortunately, there is a document that describes some possibilities for this process: RFC 6781.

16. Start planning for a Zone Signing Key rollover.
    (a) Describe the options for doing a ZSK rollover, make a motivated choice for one procedure.
    (b) How do you implement this procedure with the tools for signing your zone?
    (c) Which timers are important for this procedure?
    (d) Implement the procedure and use a DNSSEC debugger to verify each step. Don't forget to show the results of each verification.

17. Can you use the same procedure for a KSK rollover? What does this depend on?

5 Extra Assignments (Optional)

5.1 Configure zone transferring for secured zones.

18. What is necessary to set for zone transferred signed zone?
19. Do you need transfer the keys to secondary server?

5.2 NSEC / NSEC3

NSEC is used to prove that a certain resource record does not exist. A nasty side effect of NSEC is that it enables zone traversal. Using NSEC3, a DNS server can prove the non-existence of a resource record without facilitating zone traversal.

20. What are the main differences between NSEC and NSEC3?
21. Does the root use NSEC or NSEC3? What about the `.org` domain?
22. How common is NSEC3? How do tools respond to it?