Project Proposal - RP1 Monitoring DNSSec

Martin Leucht
martin.leucht@os3.nl

Julien Nyczak
julien.nyczak@os3.nl

January 9, 2015

1 Introduction

As statistics demonstrate it, DNSSEC becomes more and more popular. Indeed, 77% of TLDs have trust anchors published as DS records in the root zone [1]. However, to function properly, resource records of a DNSSEC-enabled zone need to be regularly signed since signatures have an expiration date. If signatures have not been refreshed in time, the zone becomes unreachable. This is not acceptable.
Monitoring DNSSEC parameters like signature expiration dates, could avoid those undesired effects. For instance, a monitoring system would send notifications to the zone maintainer when a RRSIG of a resource record is about to expire.
There exists a lot of approaches to monitor DNSSEC. For example, service providers offering DNSSEC services have developed their own utilities, mostly strongly related to their infrastructures and demands. All of these solutions do not use a standard protocol designed for monitoring, such as SNMP. SNMP provides an abstraction model for data which is required to be monitored. That data is represented in a common format, known as an SNMP MIB. Thus, it makes it applicable for a generic use in monitoring systems, rather than for custom made solutions. The absence of a Management Information Base for DNSSEC clearly demands to put effort into developing such a solution.

2 Research questions

- What are vital life signs for monitoring DNSSEC?
- What are vital life signs for monitoring DNS?
- How to construct a MIB for DNSSEC?
- How to conduct monitoring based on such a MIB?
- How do architectures for monitoring DNSSEC compare?

3 Related Work

As pointed out earlier, there are plenty of monitoring system already existing[2], but we are a looking for an integrated SNMP-compliant tool. We would like to write our SNMP sub-agent in Python, and some work has already been done towards that direction. Indeed, *python-netsnmpagent*[3] is a Python module that allows to easily write Net-SNMP subagents in Python. In addition, *snmp-passpersist*[4] facilitates the creation of a MIB subtree. However, MIBs for DNSSEC monitoring do not seem to have ever been created.

4 Scope

The scope of our research project is to develop an SNMP-compliant monitoring prototype solution which spots relevant DNSSEC data of DNSSEC-enabled zones. That implies the construction of a DNSSEC-SNMP MIB module. The research team will define a generic structure for the MIB and the data that should go into it.

Moreover, the MIB module needs to be implemented in an SNMP sub agent. The SNMP variables inside that SNMP MIB module need to get updated frequently in order to provide fresh and valuable data to SNMP-based monitoring probes.

Finally, if time permits, the research team will try to implement the MIB module in such a way that new zone data will be appended automatically to the SNMP MIB module tree structure as soon as a new DNS domain is detected.

5 Approach and Methods

As a first step, the research team needs to evaluate and define which DNSSEC related data will be collected and how the data can be retrieved.

Possible data sources are included in signer instances like OpenDNSSEC [5] (e.g. signing policies) or data from authoritative name servers.

Candidates for this data are for example the expiry time of the signature of the SOA record, the SOA record itself, discrepances in serial numbers in the SOA record or clock skews between the signer process and the authoritative DNS servers.

Based on that collected data, a design and implementation concept for the SNMP MIB has to be developed by the research team. That requires studying the common syntax and format of SNMP MIB modules which are defined in several RFC's [6]. The research team will build the SNMP MIB module by obeying the standard recommendations given in RFC2578-2580 for SMIv2. MIB browsers [8] and tools to verify the syntax [9] of our created SNMP MIB module will be used during the research.

Once the MIB has been constructed it needs to be registered to an SNMP subagent and the values need to get updated frequently to make it accessible for

monitoring purposes. This subagent will be implemented by using an existing Python class, as described in 3 or by writing our own sub agent in Python.

An approach of the research team is to decouple the data collecting mechanism from the mechanism which provides the data to our SNMP MIB module. That means that a separate process needs to be created to collect the relevant data in advance in a machine parsable format. The SNMP variables of our MIB module will be updated frequently by another process subsequently. We have chosen this approach because of scalability and security reasons.

Finally a monitoring probe will be created and implemented in a standard monitoring system like Nagios [7].

6 Requirements

To conduct our research we need two physical servers. On both servers an authoritative DNS server instance will be active, serving at least one DNSSEC enabled zone. Furthermore we will setup an OPENDNSSEC signing instance. We will run our SNMP AgentX subagent on top of the NET-SNMP application [10]. NET-SNMP will run on one server instance. To show our solution and its use in practice, we will implement a simple monitoring check and implement it in the well known monitoring system Nagios [7].

7 Planning

Week 1:

- Getting familiar with the topic
- Writing the project proposal

Week 2:

- Research on MIB modules
- Investigate what data to feed the SNMP MIB module
- Building a MIB module for DNSSEC monitoring

Week 3:

- Research on SNMP agent
- Implementing the MIB module in a Python SNMP subagent
- Implementing the monitoring probe in a standard monitoring system (e.g. Nagios)

Week 4:

- Writing the report
- Preparing the presentation

8 Proof of Concept

The research team will create an SNMP MIB module that will cover DNSSEC critical parameters. Hence, we will collect data from different sources (DNSSEC signer instance, authoritative DNS data). To make the defined SNMP variables and tables accessible to monitoring tools, our MIB module needs to be registered to an SNMP master agent through an AgentX [11] sub agent.

9 Ethical Concerns

Ethical issues are not expected during this project. Indeed, the research team will build a proof of concept to monitor DNSSEC. Thus, data that might be collected will be public information only.

References

[1] ICANN, *TLD statistics regarding DNSSEC*, January 2014, `http://stats.research.icann.org/dns/tld_report/`

[2] Dagmar Hilmarsdottir, *not exhaustive list of DNSSEC monitoring tools*, May 2011, `info.menandmice.com/blog/bid/58099/DNSSEC-monitoring-tools`

[3] Pieter Hollants, *Python-netsnmpagent Module*, 2013, `https://github.com/pief/python-netsnmpagent`

[4] Nicolas Agius, *SNMP passpersist*, October 2013, `https://pypi.python.org/pypi/snmp-passpersist`

[5] OpenDNSSEC, *DNSSEC Signing Instance*, 2015 `https://www.opendnssec.org/`

[6] SNMP RFC, *Overview of SNMP related RFC's*, 2013 `http://www.snmp.com/protocol/snmp_rfcs.shtml`

[7] Nagios, *Nagios Monitoring System*, 2014, `http://www.nagios.org`

[8] iReasoning, *iReasoning MIB browser*, 2014, `http://ireasoning.com/mibbrowser.shtml`

[9] smilint, *Syntax and semantic checks of SMIv1/v2*, 2014, `http://www.ibr.cs.tu-bs.de/projects/libsmi/smilint.html`

[10] NET-SNMP, *SNMP tool suite*, 2013, `http://www.net-snmp.org/`

[11] RFC2741, *Agent Extensibility (AgentX) Protocol*, January 2000, `https://www.ietf.org/rfc/rfc2741.txt`