There MUST be an RRSIG for each RRset using at least one DNSKEY of each algorithm in the zone apex DNSKEY RRset. – WHY,OH,WHY???

The root of all this particular evil

Edward Lewis

DNS Security Workshop at ICANN78
25 October 2023

For Context

- RFC 4035: *Protocol Modifications for the DNS Security Extensions*
- Section 2: *Zone Signing*
- Subsection 2.2: *Including RRSIG RRs in a Zone*

- Note – not validation, not even including RRSIG RRs in a response! This ***was only, simply,*** about preparing signatures

Where I’m Coming From

- I didn’t invent the protocol but I did write the first working DNSSEC signer and DNSSEC validator

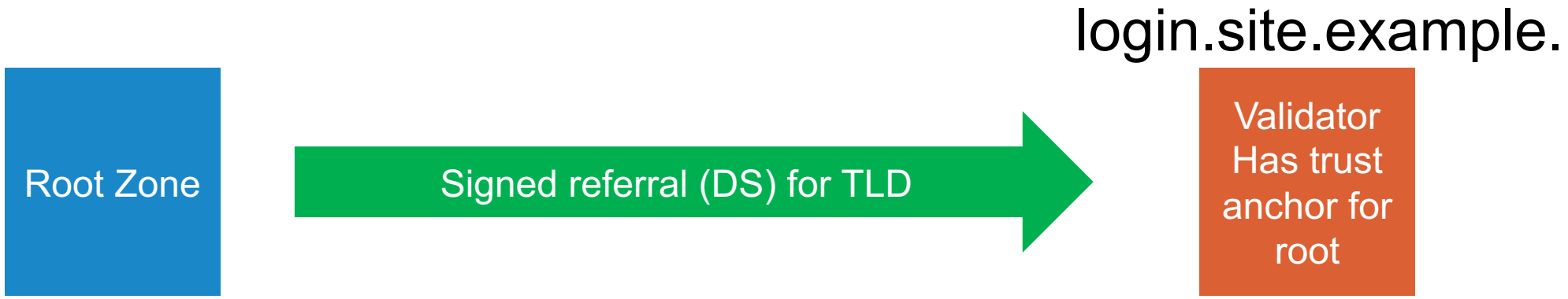
- I pushed hard for this requirement to establish the expectation of a validator – specifically to determine if a signature was removed from a response in transit

Important concept

- A receiver must know what to **expect** in a response
  - For a stable protocol the recipient of a message must know what should be received, how to parse, and how to react
  - Protocols are all about how one reacts to stimuli

- A fundamental element of security is **expectation**
  - When the environment is “as expected” it is safer
  - When exceptional events occur, suspicion is aroused

The Issue, step 1

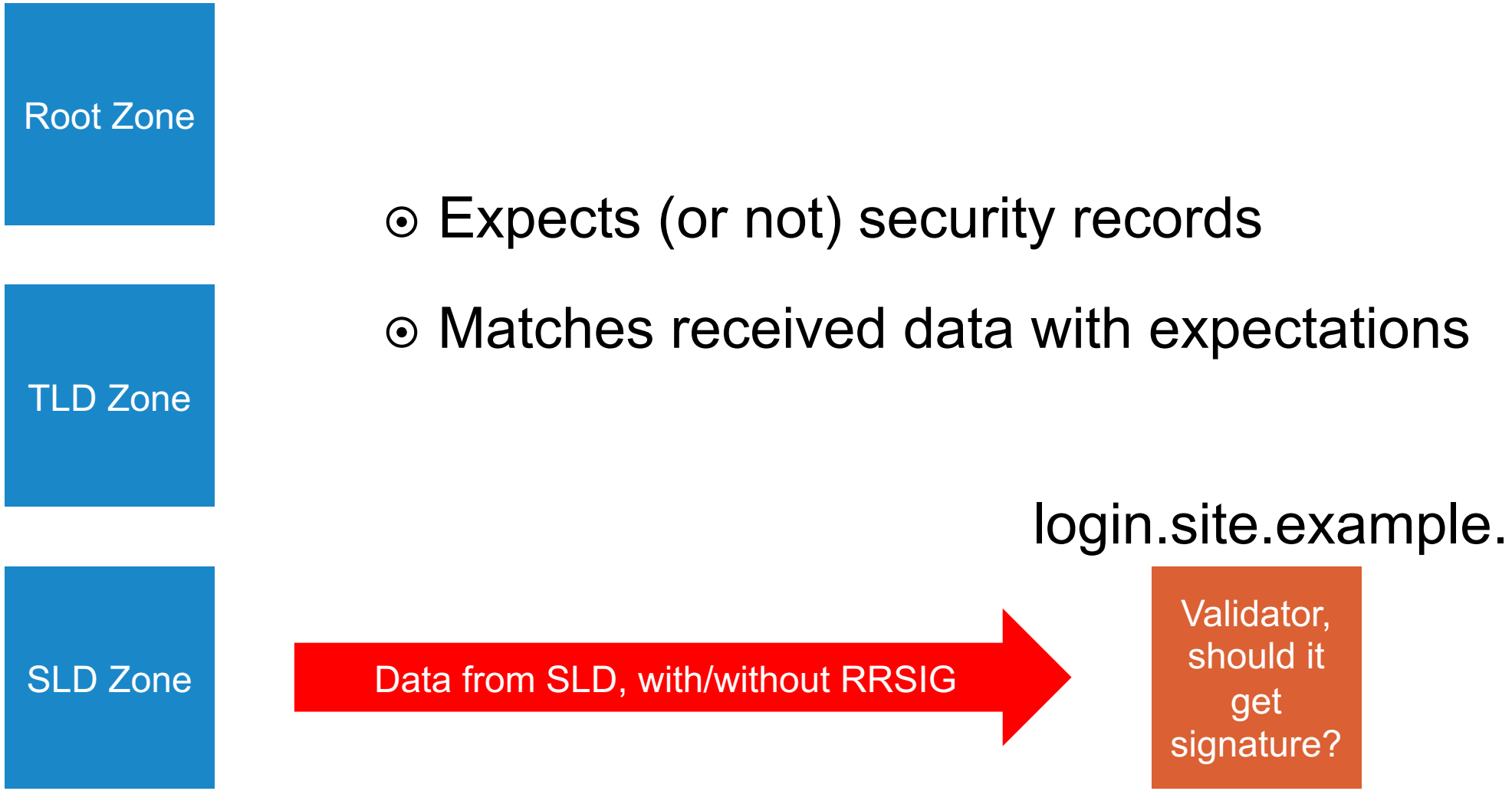


- Expects either a signed DS resource record set or signed NSEC/NSEC3 resource record set(s) in response
- Sets expectations of security within the TLD

The Issue, step 2

- Expects security based on information from root zone
- Sets expectations of security in the SLD

The Issue, step 3



The simple question is…

- Should the validator **expect** a signature?

Signature State	DNSKEY says yes	DNSKEY says no
Signature arrives	Use it to validate	Immaterial
No signature sent	Security failure	Good, but insecure

In the 1990's:

- The world of cryptography was much different than today
- There were no standard cryptographic libraries
  - SSL didn't exist for most of the decade (23 Dec 1998)
  - SSLeay was still state of the art (Eric Young/Tim Hudson)
- Cryptography was patent encumbered (RSA)
- Cryptography was highly regulated (No export rules on s/w)
- Cryptography use was "an art", no "normal set of crypto"

Impact of the 90's on DNSSEC Development

- We assumed that operators would deploy many different DNSSEC Security Algorithms at once
  - DNS Security Algorithm = hash function + cryptographic algorithm
- We assumed nothing about the commonality of cryptographic software libraries
- And we did not consider response size as an issue
  - Probably the most glaring omission in the original design effort

The complex question is…

- Should the validator **expect** a signature **it can use**?
  - Usable = cryptographic algorithm supported in code

Signature State, per algorithm	DNSKEY says yes, per algorithm	DNSKEY says no, per algorithm
Signature arrives	Use it to validate	Immaterial
No signature sent	Security failure	Good, but insecure

- For each algorithm…then combine the results

90’s Result on DNSSEC rules

- Written in a very abstract manner – lack operational reality
- Assume zone administrator is the zone (DNS[SEC]) operator
- Expectations are set by specifying the signer and sender actions
  - Should have been clearer on how recipients ought to react to received messages

Today's Reality

- Openly available cryptographic libraries are available
  - Open source DNS and many available proprietary DNS code bases adhere to a common set
  - Private proprietary DNS is supported by informed staff
- There are popular cryptographic algorithms ("operational norm")
- DNS Zone managers are wary of large responses
  - Traffic shift to TCP is an increased load
  - DDoS, especially flooding attacks powered via amplification

Operations today…signing

- Do use only one DNS Security Algorithm
  - When two co-exist, it is usually an “algorithm roll”
  - Or, it may be a mistake (they happen)
- Does sometimes separate zone admin from zone operator
- Don’t use specialized cryptography nor different keys for different data within a zone
- Don’t dwell much over cryptographic nuance (“go with the herd”)

Operations today…validating

- Do base validation service on a widely available or actively supported code base
  - Many open source implementations are available and used in service or modified and put into service
  - Most of these code bases include widely available cryptographic libraries, may be the same, hopefully with code diversity!
- Do try to keep cryptography, and DNSSEC, as simple as possible for the sake of service delivery

How does this change things?

- The rules need to be re-worded, which of course, means a new understanding
  - Probably an impact on code, necessitating roll-outs

- Perhaps the rule ought to be … no, now is not the time to propose a solution (not yet) …
  - We need to derive rules reflecting network reality, while still maintaining the goals of DNSSEC

Engage with ICANN

Thank You and Questions

Visit us at **icann.org**
Email: edward.lewis@icann.org

@icann

facebook.com/icannorg

youtube.com/icannnews

flickr.com/icann

linkedin/company/icann

soundcloud/icann

instagram.com/icannorg