GEORGE MASON UNIVERSITY

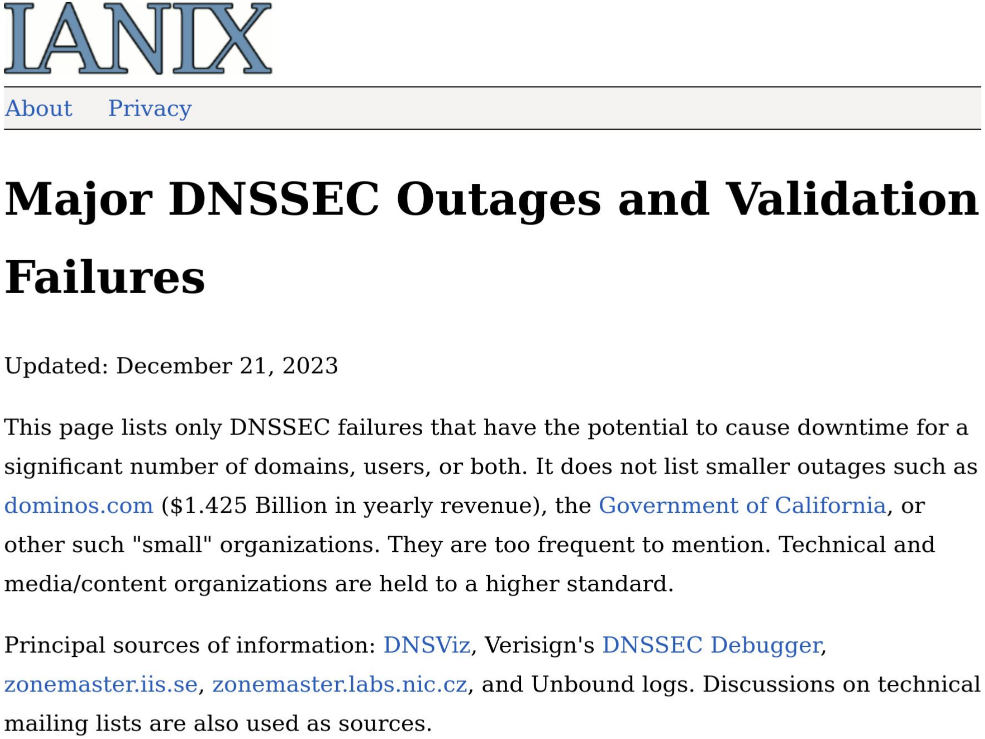
UIC

ICANN

On DNSSEC-related Outages

John Kristoff, Quan Minh Nguyen,
Steve Sheng, Eric Osterweil

Motivation



IANIX

About Privacy

Major DNSSEC Outages and Validation Failures

Updated: December 21, 2023

This page lists only DNSSEC failures that have the potential to cause downtime for a significant number of domains, users, or both. It does not list smaller outages such as dominos.com ($1.425 Billion in yearly revenue), the Government of California, or other such "small" organizations. They are too frequent to mention. Technical and media/content organizations are held to a higher standard.

Principal sources of information: DNSViz, Verisign's DNSSEC Debugger, zonemaster.iis.se, zonemaster.labs.nic.cz, and Unbound logs. Discussions on technical mailing lists are also used as sources.

Research Questions

Classification

Is there more than one type of DNSSEC-related outage?

Methodology

How are DNSSEC-related outages detected?

Results

Can we quantify DNSSEC-related outages and impact?

DNSSEC-related Outage Definition

A DNSSEC-related outage exists when queries or system components **would not have failed albeit for DNSSEC** being enabled on the end-to-end DNS resolution and processing path.

Furthermore, DNSSEC-related outages are not just authentication failures, but can occur whenever any system or software component such as zone loading or offline signing results in resolution discrepancies.

Not all outages are equal

Is a single NS serving stale signatures an “outage”?

- Probably just a partial outage

Is a lame delegation a DNSSEC-related outage?

- Maybe, but might not have anything to do with DNSSEC

“Impact” seems to matter. How do we measure it?

- How many other zones implicitly impacted?
- Don’t parts of name space have varying tolerance?
- Do recovery mechanisms (e.g., retries) limit impact?

Current Scope

Longitudinal study using SecSpider active polling data

Infrastructure records, DNSKEY RRset expirations

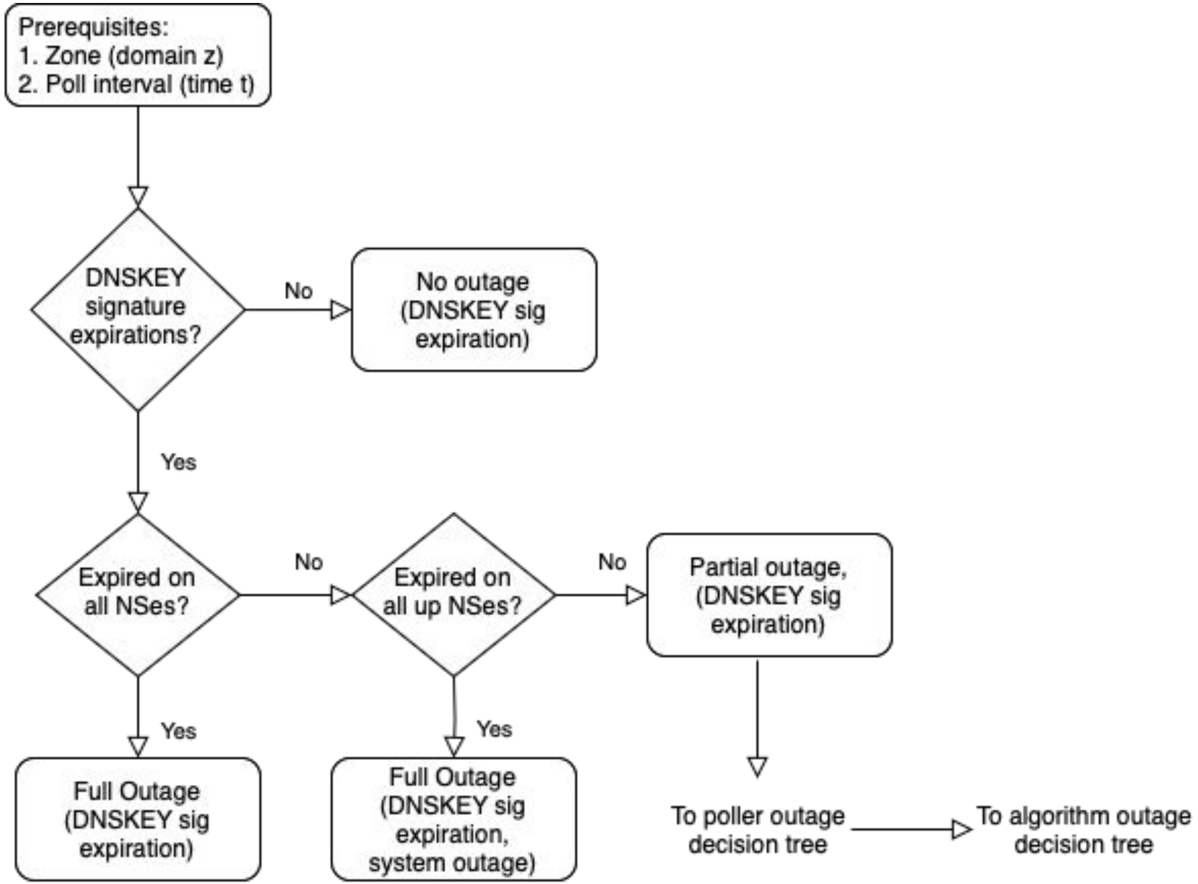
Decision Tree driven analysis

Classifying outages by zone, NS RRs, pollers, and algorithms

Methodology - classifier

DNSSEC Outage Decision Classifier

DNSKEY Expiration



Prerequisites:
1. Zone (domain z)
2. Poll interval (time t)

DNSKEY signature expirations?

No → No outage (DNSKEY sig expiration)

Yes → Expired on all NSes?

Yes → Full Outage (DNSKEY sig expiration)

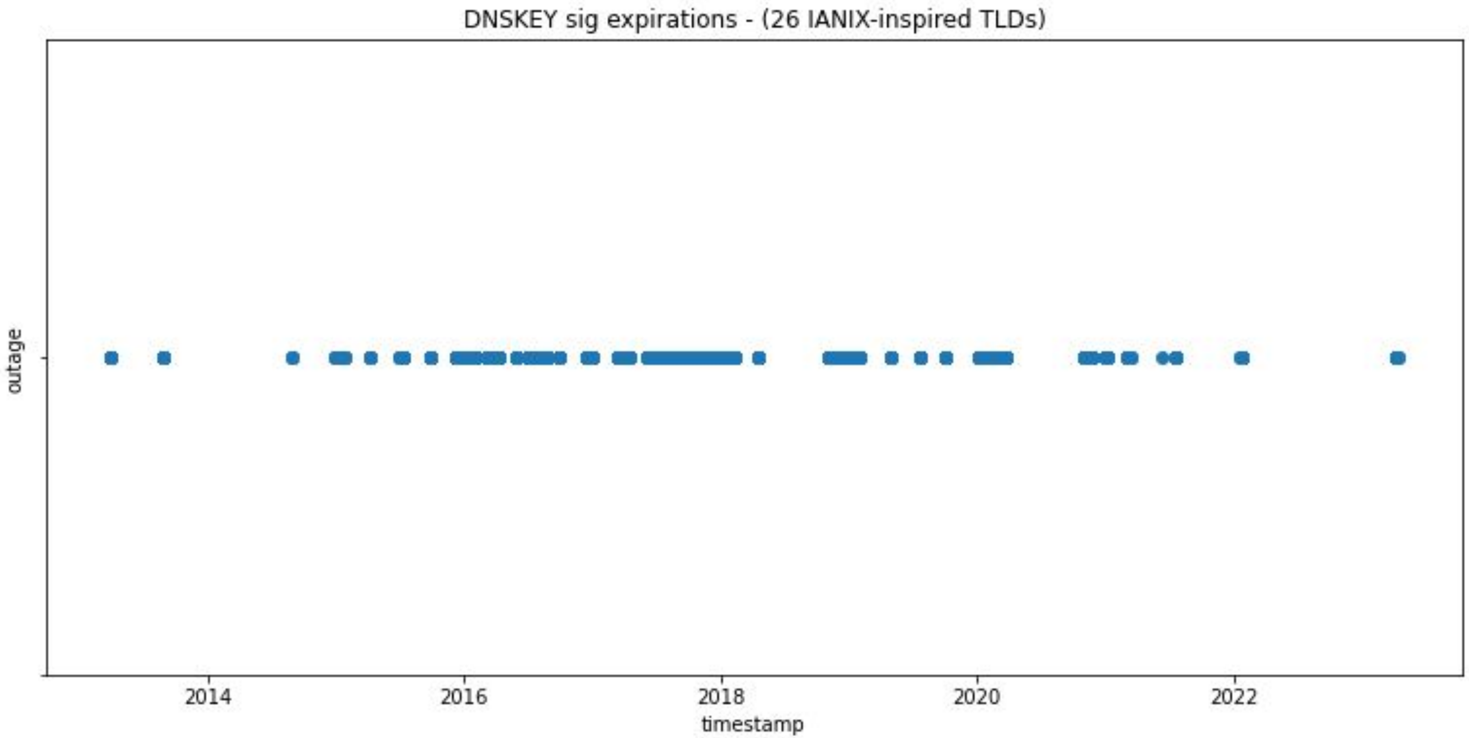
No → Expired on all up NSes?

Yes → Full Outage (DNSKEY sig expiration, system outage)

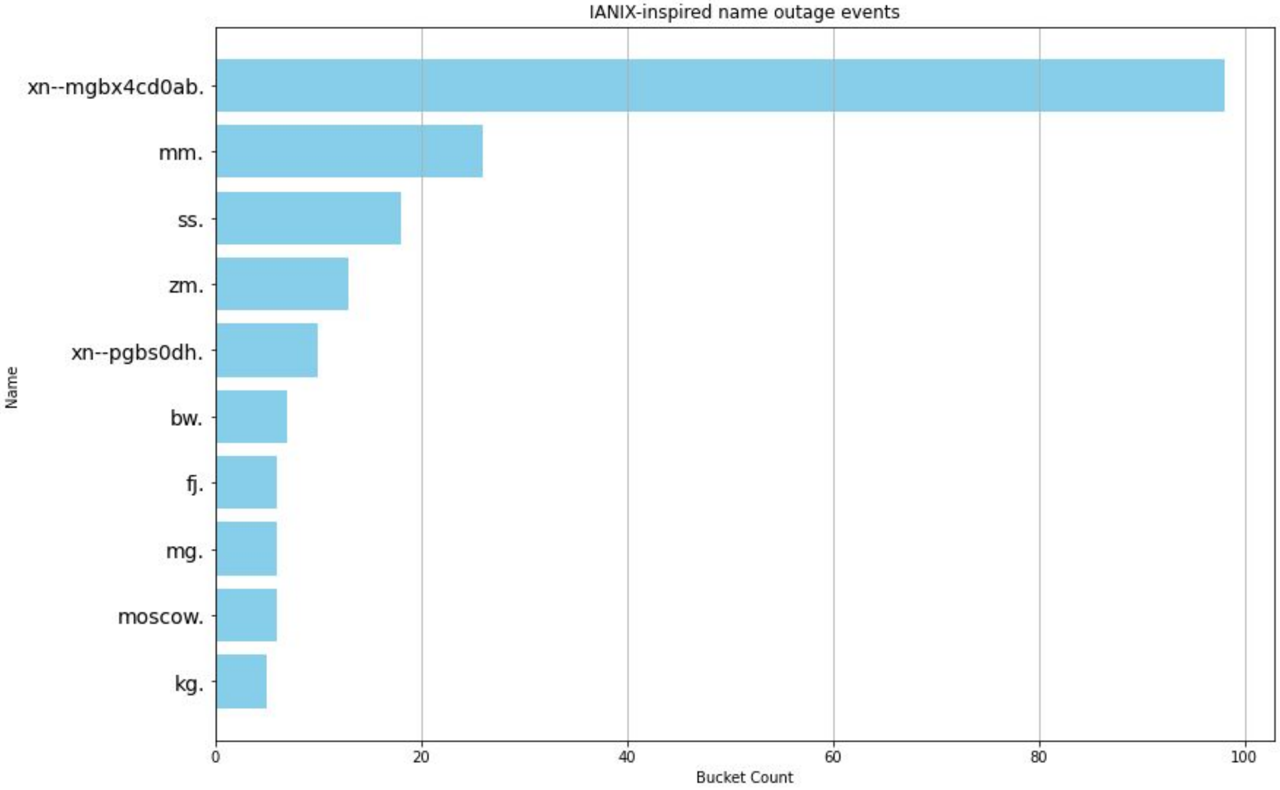
No → Partial outage, (DNSKEY sig expiration)

→ To poller outage decision tree → To algorithm outage decision tree

Results - IANIX-inspired TLDs



Results - IANIX-inspired TLDs - outage event frequency



Results - IANIX-reported full outage correlation (.mm)

IANIX	Date	SecSpider
✅	2013-03-29	✅
✅	2014-07-30	
	2015-09-27	✅
✅	2015-09-29	✅
✅	2015-12-20	✅
	2015-12-24	✅
	2016-01-20 to 2016-02-01	✅
✅	2016-03-02	✅
✅	2018-11-05	✅

Polling Resolution

Zones are polled approximately once per day

Maybe we can infer short outages from RRsig time stamps

Nonetheless, we see a lot of what IANIX sees (and misses)

Notable Impact Analysis Statistic

We looked for top 10K CrUX name/parent outages

In our secspider_2023 data set… we saw zero outages

We looked at the Public Suffix List (except *. and IDNs)

In our secspider_2023 data set… we saw one outage.

Tentative Conclusions

We see many outage-related events

But there are a LOT uneventful names/zones

secpider_2023 monitored approximately 24.5 million names

DNSKEY expiry events for only 6115 (~0.025%)

Impact appears to be limited and/or short-lived

TODO

Additional IANIX outage report corroboration/refutation

Dependency impacts and MTBF/MTTR trends

Zone performance reports (overall availability vs. outage)

Other types of DNSSEC-related outages

Tools, solutions, and operator guidance

Academic publication with full results and measurements

Overflow

Data - measurement record (combined and simplified)

zone

poller

NS address

RR qtype

RRset lastseen timestamp

RRsig inception timestamp

RRsig expiration timestamp

algorithm

Methodology - bucket and sort data hourly

```
# Sort data by lastseen timestamp

bucket_id = 0
bucket_time = event[lastseen].min()

for event in data
    if event[lastseen] >= bucket_time + 1 hour
        bucket_id++
        bucket_time = event[lastseen]
    output(bucket_id, event)
```