Availability Problems in the DNSSEC Deployment

Eric Osterweil
Dan Massey
Lixia Zhang
Motivation: Why Use DNSSEC?

• DNS cache poisoning has been a known attack against DNS since the 1990s [1]
  – Now there is a new variant: the Kaminsky attack

• Patches to existing resolvers and name servers have helped mitigate recent threats

• However, DNSSEC offers a more structured solution to ensure data’s origin authenticity and integrity
  – European operational efforts have (arguably) lead the way on the deployment front
Has DNSSEC Overstressed the DNS?

• DNSSEC added a lot to DNS packets
  • We added crypto keys (DNSKEYs)
    – Anywhere up to 4,096 bits each
    – Zones should have at least 2 (ZSK + KSK) and maybe more
  • We added crypto signatures (RRSIGs)
    – At least one in each RRset and sometimes one for each DNSKEY
    – Varying in size, based on DNSKEY sizes
• Resolvers and name servers need to send and receive these large DNS packets

• In this talk we examine a prominent availability problem in DNSSEC’s deployment
Outline

• DNSSEC background
• The network path and large packets
• How SecSpider measures
• Observations
• What can be done
DNSSEC Background

- DNSSEC provides *origin authenticity, data integrity, and secure denial of existence* by using public-key cryptography
- **Origin authenticity:**
  - Resolvers can verify that data has originated from authoritative sources.
- **Data integrity**
  - Can also verify that responses are not modified in-flight
- **Secure denial of existence**
  - When there is no data for a query, authoritative servers can provide a response that proves no data exists
How DNSSEC Works

• DNSSEC zones create public/private keys
  – Public portion goes in DNSSEC record type: DNSKEY

• Zones sign all RRsets and resolvers use DNSKEYs to verify them
  – Each RRset has a signature attached to it: RRSIG

• So, once a resolver has a zone’s DNSKEY(s) it can verify that RRsets are intact by verifying their RRSIGs
Signing Example

Using a zone’s key on a standard RRset (the NS)

Signature (RRSIG) will only verify with the DNSKEY if no data was modified.
Large Message Support in DNSSEC

• Originally, DNS messages were limited to 512 bytes
  – Resolvers use EDNS0 “negotiation” (RFC 2671) to advertise how much DNS buffer space they have for DNS messages

• Name servers try to fit data into buffers of that size
  – If data won’t fit, servers indicate response is “truncated”
  – Resolvers should explore alternate message size, “...considered preferrable to the outright use of TCP...”

• Without exploration, both sides hope the path between them will tolerate UDP packets of that size
  – This can result in false advertising

• We will show that this has lead to problems
Outline

• DNSSEC background
• The network path and large packets
• How SecSpider measures
• Observations
• What can be done
The Network Path and PMTU

• A network path is a sequence of links
• Each link can only support packets of a certain size (MTU)
• The smallest MTU for a network path is its bottleneck, or its Path Maximum Transmission Unit (PMTU)
Further Complications with DNS’ Large Packets

• DNS messages are further limited by “middle boxes” (firewalls, NAT, etc.)
  – Some firewalls drop “suspicious” DNS traffic
  – A recent study found this was quite common in SOHO routers [2]

• Because of middle boxes, network paths that may support large packets may fail to deliver large DNS messages

• We overload the term PMTU to apply in these cases too
How One Can Identify PMTU Problems

• Suppose a resolver advertizes a buffer size to a name server, but that size exceeds the PMTU
  – Result: message is dropped along the network path

• Distinguishing random drops from PMTU failures
  – Retry queries 3 times

• Distinguishing name server failures from PMTU failures
  – Reissue queries with different EDNS0 buffer sizes
  – Query from different network vantages
  – Verify the problem exists over time
  – Check if TCP works
Outline

• DNSSEC background
• The network path and large packets
• How SecSpider measures
• Observations
• What can be done
SecSpider’s Vantage Points

• We poll all of our DNSSEC zones from 8 vantages in:
  – Europe
  – Asia
  – North America

• We’re always looking for more
  – Please consider hosting a lightweight poller for us
  – Please drop me a note if you might be interested
eoster@cs.ucla.edu
SecSpider’s PMTU Walking

• To trigger a PMTU walk there must be 3 successive DNSKEY query timeouts
• After 3 timeouts, we try TCP
• Then we perform a binary search between 4,096 and 512 to see if any size will work
  – Find out precisely what size works before a failure or truncation
Outline

• DNSSEC background
• The network path and large packets
• How SecSpider measures
• Observations
• What can be done
What We Have Observed

• A recent study [4] showed that roughly 60% of queries seen at one root server advertise buffer sizes of 4,096.
• In this talk we use our distributed pollers to illustrate:
  – How often does the default behavior of using 4,096 byte buffers work for DNSSEC
  – When it fails, is it possible to advertise smaller buffer sizes that will work
  – How often are key sets just too large to fit over paths
• To illustrate, consider how different 2 pollers results’ can be
  – For example, NL NetLabs and a SOHO router (cable modem)
NL NetLabs Poller

PMTU Rates Over Time

4096 Worked
Smaller Works
Packets Don't Fit

Count

Date

Jun/08 Jul/08 Jul/08 Jul/08 Jul/08 Aug/08 Aug/08 Aug/08 Aug/08 Aug/08

100000
10000
1000
100
10
1

18
SOHO Router in Los Angeles

PMTU Rates Over Time

- 4096 Worked
- Smaller Works
- Packets Don't Fit
It Matters Where You Look From

• NL NetLabs only has trouble with roughly 10 zones (for the most part)

• However, at the same time, our SOHO router has PMTU problems with roughly 100 zones
As Seen From All of Our Pollers

- Green bars indicate the number of times a poller needed to do a PMTU walk
- Red bars indicate the percentage of times a PMTU was able to find a buffer size the allowed DNSKEYs to be received
How Many Zones Have Trouble?

- Fraction of queries (x-axis) that cause PMTU exploration (y-axis)
- For Ex: from poller 0: ~70% of the production zones only need PMTU walks ~20% of the time (or less)
- Poller 6: ~60% of the zones need PMTU walks up to 90% of the time
More Succinct

Availability Dispersion of DNSSEC Zones

- We use a metric from [3] to quantify the “availability dispersion” of each zone
  - Captures how different each poller’s view of each zone is
- Using a weighted average over time, we see that most zones have suffered dispersion
Something Interesting...
A Correlated Jump in Walks

• In September of 2008, roughly 100 zones began serving DNSKEYs that didn’t “fit” their PMTUUs

• In November, availability seems restored, but only with PMTUU walks

• Still investigating causes, but zones can check their status at
  – http://secspider.cs.ucla.edu/
Outline

- DNSSEC background
- The network path and large packets
- How SecSpider measures
- Observations
  - What can be done
What Can be Done (Tactically)

• Check your zones’ availability at: http://secspider.cs.ucla.edu/
  – We are more than happy to work with anyone that has questions
What Can be Done (Strategically)

- Try different DNSKEY configurations then monitor and observe availability through SecSpider
- Use results to collaborate on best-practices documents
- Continue to raise awareness of the problem
- Develop availability dispersion and PMTU recommendations
Summary

• We use Availability dispersion to allow us to expresses how different all of the resolvers’ views are

• Distributed monitoring needs to be a service that lets zone operators to assess their zones’ availability dispersion

• SecSpider been helping to reveal problems (such as a spike in PMTU walks) before they become insurmountable challenges to the deployment
References


4 https://www.dns-oarc.net/node/146
Thank You

Questions?