# A Quick 20 Minute Introduction to DNSSEC for the session "DNS: Don't Call It Insecure Any More"

Joe St Sauver, Ph.D.
Internet2 Nationwide Security Program Manager
(joe@internet2.edu or joe@uoregon.edu)

Internet2 Member Meeting
April 19th, 2011, 1:15-2:30PM Salon C

http://pages.uoregon.edu/joe/dnssec-intro/

# DNS: A Key Service

- The domain name system (DNS) is a fundamental service – virtually every application relies on DNS to translate names (such as www.internet2.edu) to IP addresses (such as 207.75.164.151)
- Without DNS, a lot of fairly cool tricks would be impossible. Some simple examples:

  -- we'd all have to remember IP addresses instead of fully qualified domain names (yuck)

  -- you couldn't easily and transparently move a well known server from one address to another

  -- high density virtual web hosting would be difficult

  -- you couldn't do round robin DNS (e.g., you couldn't bind multiple IP addresses to a single fully qualified domain name)

  -- etc., etc., etc.
- Bottom line, we really need DNS, and yet we know that DNS is continually under attack.

# Why Would Someone Attack DNS?
# If I Could Successfully Attack DNS, What Might I Do?

- If cyber attackers can modify the answers that DNS servers return, they can mislead users into going to the wrong places.

- For example, if you were trying to go to your bank, you might be taken to a "knock off" bank site in the Ukraine instead, or you might be involuntarily taken to a malware infected site.

- I could also DDoS key network resources by poisoning your DNS: www.google.com could be set to 127.0.0.1 (e.g., localhost)

- That intentional DNS misdirection is normally accomplished via a technical attack known as "cache poisoning." (This is not just a theoretical attack; cyber criminals are actually using this one)

# What’s DNS Caching?

- DNS resolvers remember, or “cache,” the DNS information they receive. By doing this, they don’t need to continually re-resolve popular DNS names from scratch.  For example, having resolved www.cnn.com once, they can then remember the answer they received “for a while” and then use that info for additional users.
- How long do DNS resolvers remember those answers? Well, the authoritative DNS server can recommend a duration, and normally that’s how long a response will be cached. Typical cache lifetimes might range from 7200 seconds (a couple of hours) to 172800 seconds (e.g., multiple days) or more.
- Caching of DNS data is generally a Really Good Thing: it makes DNS faster, and reduces the load on the DNS infrastructure.
- But, caching also means that if I can convince you to accept and remember an incorrect response, that one unlucky moment can have a protracted impact.

# So How Would I Poison a DNS Cache?

- If an attacker wanted to convince a resolver’s cache to remember bogus information, that is, to “poison” the resolver’s cache, an attacker would generate a stream of spurious DNS responses that would compete with real answers that might be coming in for the same domain name.
- As you might expect, those answers would need to correspond to a pending query – for example, if you wanted to poison yahoo.com, there would need to be a pending query for that domain name.
- But the answer needs to match more than just the domain name: it needs to also match the source port associated with the query, and the transaction ID number. In the ideal world, given a random distribution of values, it would be hard to guess these numbers. Unfortunately, many implementations have source port and transaction IDs that are insufficiently random and too predictable

# Testing Your Resolver's Entropy

-
  ```
  % dig +short porttest.dns-oarc.net TXT

  "128.223.32.36 is GREAT: 26 queries in 0.6
  seconds from 26 ports with std dev 17311”
  ```

-

| Rating | Standard Deviation | Bits of Entropy |
|---|---|---|
| GREAT | 3980 -- 20,000+ | 13.75 -- 16.0 |
| GOOD | 296 -- 3980 | 10.0 --13.75 |
| POOR | 0 -- 296 | 0 -- 10.0 |

- Note: some name servers break this test… For example, if you are told “Only received 3 queries.  Please try again in 60 seconds” then a network middlebox is interfering with your DNS service.

# How Can I Prevent DNS Cache Poisoning?

- Cache poisoning attacks can be prevented IF…

  (a) the authoritative DNS servers (which ultimately provide DNS queries) cryptographically sign their DNS data, and

  (b) recursive resolvers (which generate DNS queries) cryptographically validate the DNS signatures they receive.

  (c) if that signature doesn't validate, the DNS answer that's been received gets ignored as invalid/broken/bogus

- The rest of the DNSSEC story is just a matter of "details."

- Cough! :-;

# An Example of A “Detail:” Keys and Chains of Trust

- All cryptographic protocols need some solution for managing keys and maintaining “chains of trust.”

- DNSSEC uses a so-called “tree model,” beginning with cryptographic keys for the root of the DNS tree (“.”)

- For a long time, the root wasn’t signed, which meant that you needed to manually maintain trust anchors for each DNSSEC-enabled top level domain, or look to a third party to maintain those trust anchors for you

# Key Parts of the DNS Tree Have Been Signed

- The root (".") got signed on July 15th, 2010.
- Dot edu domain got signed on August 2nd, 2010.
- Dot info got signed on September 1st, 2010.
- Dot net got signed on December 10th, 2010.
- Dot com got signed on March 31st, 2011.
- Plus plenty of other top level domains are also DNSSEC signed…
- The good news is that this means that those of us who want to DNSSEC-validate 2nd level domains won't need to manually collect and maintain trust anchors for each such 2nd level domain or top level domain– we can simply have ONE cryptographic key for the root, and then leverage that single key to bootstrap all the TLDs that have been signed by the root key, and all the 2nd level domains that have been signed by those TLDs, etc.
- Are 2nd level domains, including university TLDs, getting signed?

# Are Any Schools Currently <u>Signing</u> Their 2nd Level dot edu domains?

- Yes. http://secspider.cs.ucla.edu/ tracks (most) domains that are DNSSEC signing their zones, including (most) dot edu's.

- Schools that are known to have signed their 2nd level dot edu zones include:
  (1) baker.edu, (2) berkeley.edu, (3) carnegiemellon.edu, (4) cmu.edu,
  (5) desales.edu, (6) eunc.edu, (7) fhtc.edu, (8) gtc.edu, (9) indiana.edu,
  (10) internet2.edu, (11) iu.edu, (12) iub.edu, (13) iup.edu, (14) iupui.edu,
  (15) jhuapl.edu, (16) k-state.edu, (17) ksu.edu, (18) lctcs.edu, (19) lsu.edu,
  (20) ltc.edu, (21) merit.edu, (22) minnesota.edu, (23) monmouth.edu,
  (24) okstate.edu, (25) oxford-university.edu, (26) pacificu.edu, (27) penn.edu,
  (28) psc.edu, (29) southern.edu, (30) suu.edu, (31) ucaid.edu, (32) ucr.edu,
  (33) uiowa.edu, (34) umbc.edu, (35) upenn.edu, (36) upf.edu, (37) valencia.edu,
  (38) washjeff.edu, (39) weber.edu, and (40) wnec.edu

- Note that some schools may have multiple related domains (e.g., carnegiemellon.edu and cmu.edu for example)

# How Many Schools Currently Validate DNSSEC Signatures?

- Unfortunately, we don't have solid data for that question.
- A site can enable DNSSEC validation with no externally discernible sign that they're doing so, and many sites might enable DNSSEC validation even if they don't sign their own zones. For example, while UOregon doesn't sign its own zones yet, it does validate DNSSEC signatures from other domains on its production resolvers (that part's pretty painless).
- Why doesn't EVERYONE enable DNSSEC validation? Multiple potential reasons:
  -- Some recursive resolver software may not support DNSSEC
  -- DNSSEC works silently; there's no discernible indication that DNSSEC is doing anything for you when everything is working the way it should. If I fix a problem and no one knows, should I bother? Some people apparently think "nah, why bother?"
  -- DNSSEC can "break" domains that would otherwise be accessible, if a site accidentally screws up their DNSSEC signatures (e.g., by letting them expire); this may not be viewed as a "feature" by your users
  -- DNSSEC requires support for EDNS0 ("extra long" DNS replies); some sites may have older or misconfigured firewalls that are unable to handle EDNS0 extensions
  -- Chicken and egg issues ("no one's signing, so why bother trying to validate?")

# What If You Want To Try DNSSEC?

- Start by having a conversation with your DNS administrators -- they may already be testing or doing planning with respect to DNSSEC.
- Before embarking on DNSSEC, make sure that your DNS infrastructure is otherwise up to snuff (e.g., if you're running an ancient version of BIND on end-of-life hardware, you need to get the meat-and-potatoes handled before you get dessert!).
- http://dnscheck.iis.se/ can help ID many DNS configuration issues.
- Recognize that you can "ease into" doing DNSSEC. For example:
  -- you can try offering DNSSEC-enabled test resolvers for opt-in use (or you can try the validating resolvers that DNS-OARC is making available; see https://www.dns-oarc.net/oarc/services/odvr )
  -- you can try signing some less-critical ("toy") domains to get some signing experience w/o putting critical institutional assets at risk
  -- you can decide you only want to sign, or only want to validate – you don't need to do both at once

# DNSSEC Resources for Your DNS Admin Team

- Begin with your current DNS vendor – DNSSEC aware vendors (such as ISC BIND) will often have specific DNSSEC documentation that will walk you through what you need to do, and obviously O'Reilly's "DNS and BIND" (now in its 5th edition) is a bible that every DNS admin should own.
- There are also many freely available community documents, see the list at http://www.dnssec.net/practical-documents (I'm particularly fond of ISC's "DNSSEC in 6 Minutes" (79 slides) from that list, see alan.clegg.com/files/DNSSEC_in_6_minutes.pdf )
- Some sites may prefer to buy rather than build. If that's you or a site you know, you should know that there are multiple DNSSEC-enabled appliance vendors you can consider, both for DNSSEC-enabled authoritative servers and for validating resolvers (but at least some of them may not be cheap – federal agencies are prime customers, and pricing sometimes reflects that target audience).

# What Might Internet2 and Our Campuses Do To Help?

- -- Publicly highlight the importance of DNS as a critical (but potentially vulnerable) service
  -- Explicitly endorse DNSSEC as one important way to help improve the trustworthiness of DNS results
  -- Lead by example/commit to “eating their own dog food” by working to deploy DNSSEC on their own campuses
  -- Acknowledge community participants who have made the effort to deploy DNSSEC (“Map of glory” with stars for DNSSEC-enabled Internet2 participants? Plaque or other tangible award for particularly enthusiastic community DNSSEC boosters?)
  -- Explicitly encourage DNSSEC appliance vendors to participate as part of the Internet2 corporate membership
  -- [your ideas here]

# Thanks For The Chance to Talk Today!

- Are there any questions?