Port 53 Wars

Security of the Domain Name System and Thinking About DNSSEC

Internet2/ESNet Joint Techs, Minneapolis, MN 9:10AM February 14th, 2007

Joe St Sauver, Ph.D. (joe@uoregon.edu)
Internet2 and University of Oregon Computing Center
http://www.uoregon.edu/~joe/port53wars

Disclaimer: All opinions expressed are solely those of the author and do not necessarily represent the opinions of any other entity.

My Earlier DNS-Related Recommendations Are Included/Reiterated in This Talk By Reference

- At the Spring 2006 Internet2 Member Meeting in Arlington VA, in my talk, **A Brief Practical Security "Punch List"** (see http://www.uoregon.edu/~joe/punchlist/punchlist.ppt (or .pdf)), I made a number of security recommendations relating to DNS:
  -- insure your network is doing **BCP38** ingress traffic filtering
  -- insure that your domain name servers aren't **open recursive**, thereby reducing the likelihood that your servers will participate in DNS amplification attacks or be subject to cache poisoning
  -- insure that you're running a **current version of BIND** (or whatever DNS software you may be using)
  -- consider monitoring your own DNS traffic with **DNStop**
  -- consider donating DNS log data to the **RUS-CERT Passive DNS Replication Project**
- All those recommendations continue to be very important, but I'm not going to explicitly belabor them again as part of THIS talk.[2]

Basic DNS Sanity Check

- **If you do NOTHING else recommended in this talk, I strongly encourage <u>everyone</u> to at least go to**

  http://dnsreport.com/

  and conduct a basic test of your university's DNS.

That free DNS check will do 56 basic tests, reporting many DNS-related inconsistencies and DNS-related security issues.

- The output is easy to understand, and once you know an issue exists, you can then work on getting it fixed.

Beyond That, What Will I Cover Today?

1. The Importance of DNS

2. A Brief Hand-Waving Overview Of How DNS Works

3. DNS and Malware

4. DNSSEC: What Is It?

5. Why Aren't People Using DNSSEC?

1. The Importance of DNS

You Should Pay Attention to DNS Because:

- **"Everything" relies on DNS** (email, Usenet, IM, the world wide web, P2P, VoIP, you name it), it is ALL is built on top of DNS -- DNS is the foundation technology (or at least DNS is one of just a handful of particularly key foundation technologies – I'll certainly concede that BGP is equally as important as DNS, for example).
- **If I can control your DNS, I control your world.** Going to eBay? Maybe, maybe not, depending on what sort of DNS resolution occurs (and no, SSL certificate issues will not be sufficient to flag DNS misdirection as an issue -- users just don't get the whole certificate thing, and will just blindly accept any SnakeOil, Inc. self-signed certificate they've been handed for a "secure" site).
- **Miscreants can (and have!) attacked the trustworthiness of DNS data** on a variety of levels (cache poisoning and malware that tweaks host file entries and/or DNS registry entries on the PC are just two examples)

You Should Also Pay Attention To DNS Because… (cont. 1)

- **DNS uses UDP**. Because of that, **DNS has issues when it comes to accepting and processing spoofed query sources**. Because DNS accepts a tiny query as input, and potentially generates a huge response as output, **DNS operates as a high-gain online traffic amplifier**. Couple those two phenomena and you can do the online equivalent of vaporizing small cities with a **DNS "death ray."**
- Name servers aren't just a tool for conducting distributed denial of service attacks, **DNS servers are also a target for distributed denial of service attacks** (if I can kill your DNS service, you are off the network even if your transit links aren't flooded with traffic)
- **DNS has traditionally not been a focus of institutional love and investment**; lots of people are running old gear, old code, using part time or student DNS staff, and generally treating DNS very casually despite how operationally critical it has become.

You Should Also Pay Attention To DNS Because… (cont. 2)

- DNS is used for a lot more than just translating FQDNs to dotted quads these days.
- **DNS has effectively become a general-purpose distributed database**. DNS block lists are one example of non-traditional data distributed via DNS, RouteViews IP-to-ASN data is another, and ENUM data (see www.enum.org) is a third.
- A comment from Eric A. Hall, ca. April 16, 2001, noted in passing:

  "The current DNS will only keep working if it is restrained to lookups, the very function that it was designed to serve. It will not keep working if the protocol, service, tables and caches are overloaded with excessive amounts of data which doesn't benefit from the lookup architecture."

  http://www.ops.ietf.org/lists/namedroppers/namedroppers.2001/msg00247.html

You Should Also Pay Attention To DNS Because… (cont. 3)

- **Some people are doing some wild stuff via DNS**. Personal favorites in the "**no,-this-is-not-what-we-intended**" category relate to DNS-based "covert channel" apps such as…
  -- "DnsTorrent" (see http://www.netrogenic.com/dnstorrent/)
  -- "IP over DNS" (see http://thomer.com/howtos/nstx.html or "DNS cat" (see http://tadek.pietraszek.org/projects/DNScat/), or
  -- "Tunneling Arbitrary Content in DNS" (part of Dan Kaminski's "Attacking Distributed Systems: The DNS Case Study," see http://www.doxpara.com/slides/BH_EU_05-Kaminsky.pdf)
  Two other great Kaminski DNS-related talks are "Black Ops 2004@LayerOne," see http://www.doxpara.com/bo2004.ppt , and "Black Ops of TCP/IP 2005," see http://www.doxpara.com/slides/Black%20Ops%20of%20TCP2005_Japan.ppt
- **Note well:** sites may view "atypical" DNS usage as hostile/illegal.

You Should Also Pay Attention To DNS Because… (cont. 4)

- **Your DNS (or, more precisely, your rDNS) may determine how some people treat your email or other network traffic**.
- For example, some ISPs check that rDNS exists for the sending host; others look for **"non-dynamic"-looking rDNS host names** when deciding whether to accept or reject direct-to-MX email. See, http://postmaster.aol.com/guidelines/standards.html or Steve Champeon's very thorough listing at http://enemieslist.com/
- There are efforts underway in the IETF to encourage consistent use of rDNS, and to standardize rDNS naming practices:
  -- http://www.ietf.org/internet-drafts/
  draft-ietf-dnsop-reverse-mapping-considerations-01.txt
  -- http://tools.ietf.org/wg/dnsop/
  draft-msullivan-dnsop-generic-naming-schemes-00.txt
  [What do <u>your</u> campus rDNS naming conventions look like?]

You Should Also Pay Attention To DNS Because… (cont. 5)

- **DNS ties into a ton of other things.**

  -- Where do dynamic hosts get their DNS information? Why, often from **DHCP**, of course (so you really want to pay attention to DHCP-related security issues, too).

  -- DNS can be used for **load balancing**, and DNS can selectively deliver **different answers** based on a query's source.

  -- Planning on doing **IPv6**? How you handle DNS is an integral part of that, whether that's numbering plans, provisioning quad A records, making local DNS servers available via IPv6, etc.

  -- DNS ties into broader Domain Name-related **policy issues** in myriad interesting ways (for example: how do you handle evil DNS glue records? what about IP whois/rwhois privacy? how do you manage the rate of routing table growth while still allowing for provider independent addresses and easy multihoming? etc.)

You Should Also Pay Attention To DNS Because… (cont. 6)

- Some current approaches to dealing with DNS insecurities may negatively impact Internet end-to-end transparency, and ironically, foreclose other approaches to securing DNS (such as DNSSEC). The IAB recently noted in an IETF technical plenary:

"DNSSEC deployment may be hampered by transparency barriers."

[…]

"DNS Namespace Mangling

"– Recursive forwarders modifying responses are incompatible with DNSSEC."

Reflections on Internet Transparency
http://www3.ietf.org/proceedings/06nov/slides/plenaryt-2.pdf

Last of All, You Will Also Want to Pay Attention To DNS Because…

- **DNS is on the Research Radar as a Big Deal**: CoDNS is a perfect example in that space (see http://codeen.cs.princeton.edu/codns/) but there are plenty of others.
- **DNS is on the Federal Radar as a Big Deal**: DNSSEC is receiving significant federal interest; see for example DHS's http://www.dnssec-deployment.org/ and NIST SP 800-81)...
- **DNS is on the Corporate Radar as a Big Deal**: VeriSign Site Finder (see http://en.wikipedia.org/wiki/Site_Finder) is a nice example of some folks who expected to make **big money** via DNS
- **So... bottom line, I think DNS is a very important and timely area that "punches through" a lot of background noise.**
- **What characteristics <u>should</u> DNS have?**

Important DNS Characteristics

- **Be available** (remember, if the domain name system is unavailable, for most users, the "Internet is down")
- **Be trustworthy** (if the domain name system returns untrustworthy values, you may be sent to a site that will steal confidential data, or to a site that could infect your computer with malware)
- **Be fast** (rendering even a single web page may require tens -- or hundreds! -- of domain name system queries; can you imagine waiting even a second for each of those queries to get resolved?)
- **Be scalable** (there are billions of Internet users who rely on DNS, all around the world)
- **Be flexible** (different sites may have different DNS requirements)
- **Be extensible** (there are still many things that DNS will be called upon to do, but we don't know what all those things are yet! We need to have the flexibility to evolve DNS as time goes by)
- **Let's begin by talking a little about how DNS currently works.**

2. A Quick Hand Waving DNS Tutorial

What The Domain Name System Does

- Pretty much everyone at Joint Techs conceptually understands how the Domain Name System (DNS) works, but just for the sake of completeness, or those who may look at this talk after the fact, let me begin with a brief (and very incomplete) functional definition:

  "DNS is the network service that translates a fully qualified domain name, such as *www.uoregon.edu*, to a numeric IP address, such as *128.223.142.89*. DNS can also potentially do the reverse, translating a numeric IP address to a fully qualified domain name."

- Whenever we use the Internet we're using DNS, and **without DNS, using the Internet would become very inconvenient**. Can you imagine having to remember to go to http://66.102.7.147/ instead of http://www.google.com/ for example?

How Does the DNS System *Currently* Work?

- While the fine points can vary, the basic process is:
  1) An application (such as a web browser) requests resolution of a fully qualified domain name, such as www.uoregon.edu
  2) If the desktop operating systems includes a caching DNS client, the DNS client checks to see if that FQDN recently been resolved and cached (stored locally) -- if yes, it will use that cached value.
  3) If not, the desktop DNS client forwards the request for resolution to a recursive DNS server which has been manually pre-configured (or to a recursive DNS server which may have been designated as part of DHCP-based host configuration process)
  4) If the recursive DNS server doesn't have a recently cached value for the FQDN, the recursive DNS server will begin to make queries, if necessary beginning with the DNS root zone, until it has resolved a top level domain (e.g., .edu), primary domain name (uoregon.edu), and finally a FQDN (such as www.uoregon.edu)

We can simulate that process with dig….

The process begins by bootstrapping via pre-specified name servers for the root ("dot"):

% dig +trace www.uoregon.edu

.	**417141**	**IN**	**NS**	**B.ROOT-SERVERS.NET.**
.	417141	IN	NS	C.ROOT-SERVERS.NET.
.	417141	IN	NS	D.ROOT-SERVERS.NET.
.	417141	IN	NS	E.ROOT-SERVERS.NET.
.	417141	IN	NS	F.ROOT-SERVERS.NET.
.	417141	IN	NS	G.ROOT-SERVERS.NET.
.	417141	IN	NS	H.ROOT-SERVERS.NET.
.	417141	IN	NS	I.ROOT-SERVERS.NET.
.	417141	IN	NS	J.ROOT-SERVERS.NET.
.	417141	IN	NS	K.ROOT-SERVERS.NET.
.	417141	IN	NS	L.ROOT-SERVERS.NET.
.	417141	IN	NS	M.ROOT-SERVERS.NET.
.	417141	IN	NS	A.ROOT-SERVERS.NET.

;; Received 436 bytes from 128.223.32.35#53(128.223.32.35) in 0 ms

Next, one of the root servers identifies the NS's for the .edu TLD:

edu.	**172800**	**IN**	**NS**	**L3.NSTLD.COM.**
edu.	172800	IN	NS	M3.NSTLD.COM.
edu.	172800	IN	NS	A3.NSTLD.COM.
edu.	172800	IN	NS	C3.NSTLD.COM.
edu.	172800	IN	NS	D3.NSTLD.COM.
edu.	172800	IN	NS	E3.NSTLD.COM.
edu.	172800	IN	NS	G3.NSTLD.COM.
edu.	172800	IN	NS	H3.NSTLD.COM.

;; Received 306 bytes from 192.228.79.201#53(B.ROOT-SERVERS.NET) in 30 ms

One of those TLD name servers then identifies the NS's for uoregon.edu:

uoregon.edu.	**172800**	**IN**	**NS**	**ARIZONA.edu.**
uoregon.edu.	172800	IN	NS	RUMINANT.uoregon.edu.
uoregon.edu.	172800	IN	NS	PHLOEM.uoregon.edu.

;; Received 147 bytes from 192.41.162.32#53(L3.NSTLD.COM) in 85 ms

And then finally, via one of the name servers for uoregon.edu, we can then actually resolve www.uoregon.edu:

www.uoregon.edu.	**900**	**IN**	**A**	**128.223.142.89**
uoregon.edu.	86400	IN	NS	phloem.uoregon.edu.
uoregon.edu.	86400	IN	NS	arizona.edu.
uoregon.edu.	86400	IN	NS	ruminant.uoregon.edu.
uoregon.edu.	86400	IN	NS	dns.cs.uoregon.edu.

;; Received 228 bytes from 128.196.128.233#53(ARIZONA.edu) in 35 ms

DNS is An Inherently Distributed Service

- What you should glean from that example is that DNS is **inherently distributed** – every sites doesn't need to store a copy of the the complete Internet-wide mapping of FQDN's to IP addrs.
- This differs dramatically from **pre-DNS** days, when mappings of host names to IP addresses happened via **hosts files**, and each server would periodically retrieve updated copies of the hosts file. (Can you imagine trying to maintain and distribute a hosts file with hundreds of millions, or **billions**, of records each day?)
- Fortunately, because DNS is distributed, it scales very well, far better than replicating host files!
- Unfortunately, because DNS is distributed, it is more complex than the conceptually simple (if practically unworkable) hosts file solution, and there can be substantial variation in how, and how well, sites and DNS administrators do DNS-related activities.
- There are a few things we can generally note, however.

DNS Efficiencies

- Most common DNS queries do not require re-resolving the TLD (.edu, .com, .net, .org, .biz, .info, .ca, .de, .uk, etc.) name servers, or even the name servers for 2nd level domains such as google.com or microsoft.com -- those name servers change rarely if ever, and will typically be statically defined via "glue" records, and cached by the local recursive name server. (Glue records assist with the DNS bootstrapping process, providing a static mapping of name server's FQDNs to its associated dotted quad.)
- Cached data which has been seen by a DNS server will be reused until it "cooks down" or expires; cache expiration is controlled by the TTL (time to live) associated with each data element. TTL values are expressed in seconds.
- Negative caching (the server may remember that a FQDN **doesn't** exist) may also help reduce query loads; see "Negative Caching of DNS Queries (DNS NCACHE)," RFC2308.

A Few More DNS Notes

- The DNS entries for domains are contained in **zones**. For example, there would normally be one zone for uoregon.edu and another zone for oregonstate.edu
- The **primary** DNS server for a given domain normally is augmented by a number of **secondary** (or "slave") DNS servers. Secondary servers are deployed to help insure domains remains resolvable even if a primary server becomes unreachable.
- Secondary DNS servers periodically retrieve updated zone data for the zones they secondary from the primary DNS server. Most sites limit who can download a complete copy of their zone file because having a definitive listing of all hosts in a given domain may be useful for cyber reconnaissance and attack purposes.
- It is common for universities to agree to provide secondary DNS service for each other, e.g., Arizona does runs a secondary for UO. But ALSO see the excellent http://www.ripe.net/ripe/meetings/ripe-52/presentations/ripe52-plenary-perils-transitive-trust-dns.pdf

Despite Being Critical to the Functioning of the Internet, DNS Is Seldom Given Much Attention

- Doing DNS for a university or a company is not a particularly glamorous or high prestige job (unlike being a network engineer, few novices aspire to some day become a DNS administrator)
- DNS servers seldom receive the care or lavish attention that mail servers, web servers, firewalls, or switches and routers receive, and enterprise DNS architectures and operational approaches are frequently quite simple
- To the best of my knowledge, there are no routinely scheduled reoccurring conferences devoted exclusively to DNS-related research or operational praxis
- DNS is thus simultaneously operationally critical **and** managerially insignificant to the point of often being obscure/unknown.
- **Are you paying attention to YOUR DNS servers?**

DNS Can Be Misused Very Many Ways

- The bad guys (and gals) "get" the potential of DNS, and are now interested in DNS for use in a variety of capacities, including:

  -- as a **distributed denial of service (DDoS) attack tool**

  -- as a way to affirmatively **misdirect** (increasingly careful) users who are learning to spot and be wary of phishing solicitations; this is often called "pharming," and can involve MITM attacks or cache poisoning

  -- as a way to **limit user access to resources,** such as antivirus updates, needed for the remediation of malware infections

  -- as a key **botnet command and control technology**, and even

  -- use of DNS as a distributed **content delivery system**.

Others Are Becoming Interested in DNS Because of New Potential Roles, Including

- … as a new way of **identifying** infected systems (see, e.g., http://aharp.ittns.northwestern.edu/talks/bots-dns.pdf)
- … as a new way of **mitigating** infected systems
- … as a new way of "**monetizing**" typos and other domain name resolution "misses"
- … as something which will **needs to be fixed** after miscreant name servers get taken off the air.

- **And then there are all the rest of us, who probably "just" want DNS to continue to work!**

- **Let's look at malware and DNS for a minute or two…**

3. Malware and DNS

Spam-Related Malware Relies on DNS

- Much of the most virulent malware out there has been deployed to facilitate spamming, and that spam-related malware is notorious for generating large numbers of DNS queries for MX host information (so the spamware can determine where it should connect to dump its spam).
- Spam related malware may also refer to upstream command and control hosts by their FQDNs, thereby making it possible for the miscreants to repoint their mailware's command and control host from one dotted quad to another, should the system currently "hosting" their C&C get filtered or cleaned up.

- At the same time that malware critically **relies** on DNS, ironically other malware may **also** be actively working to interfere with legitimate DNS uses.

Why Would Malware Interfere With DNS?

- Authors of viruses, trojan horses and other malware may interfere with user DNS for a variety of reasons, including:

  -- attempting to block access to remediation resources (such as system patches, AV updates, malware cleanup tools)

  -- attempting to redirect users from legitimate sensitive sites (such as online banks and brokerages) to rogue web sites run by phishers

  -- attempting to redirect users from legitimate sites to malware-tainted sites where the user can become (further) infected

  -- attempting to redirect users to pay-per-view or pay-per-click web sites in an effort to garner advertising revenues

Examples of Malware Interfering with DNS

- **Trojan.Qhosts** (discovered 10/01/2003)
  http://www.sarc.com/avcenter/venc/data/trojan.qhosts.html
  "Trojan.Qhosts is a Trojan Horse that will modify the TCP/IP settings to point to a different DNS server."
- **MyDoom.B** (published 1/28/2004)
  http://www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=38114
  "The worm modifies the HOSTS files every time it runs to prevent access to the following sites [list of sites deleted]"
- **JS/QHosts21-A** (11/3/2004)
  http://www.sophos.com/virusinfo/analyses/jsqhosts21a.html
  "JS/QHosts21-A comes as a HTML email that will display the Google website. As it is doing so it will add lines to the Windows Hosts file that will cause requests for the following websites to be redirected: www.unibanco.com.br, www.caixa.com.br, www.bradesco.com.br"

Another Example

- **Win32.Netmesser.A** (published 2/1/2005): http://www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=41618

  "[the trojan] then enumerates the following registry entry: *HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\ Parameters\Adapters*

  checking for references to dial up adapters. If found, the adapters' DNS servers are changed by altering the value 'NameServer' in the referenced key."

  […]
  "Computer Associates have seen the following DNS server IPs used by these trojans in the wild: 69.50.166.94, 69.50.188.180, 69.31.80.244, 195.225.176.31"

  [you can do the whois on all the dotted quads :-)]

More Examples of Malware Tweaking DNS

- **Trojan.Flush.A** (discovered 3/4/2005)
  http://www.sarc.com/avcenter/venc/data/trojan.flush.a.html
  'Attempts to add the following value […]:
  "NameServer" = "69.50.176.196,195.225.176.37"'

- **DNSChanger.a** (added 10/20/2005)
  http://vil.mcafeesecurity.com/vil/content/v_136602.htm
  "Symptoms: […] Having DNS entries in any of your network adaptors with the values: 85.255.112.132, 85.255.113.13"

- **DNSChanger.c** (added 11/04/2005)
  http://vil.nai.com/vil/Content/v_136817.htm
  "This program modifies registry entries pertaining to DNS servers to point to the following IP address: 193.227.227.218"

ZLOB Trojan (9/3/2006)

- ZLOB is a piece of "fake video codec" DNS-tinkering malware, see http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=TROJ_ZLOB.ALF&VSect=Sn and http://www.trendmicro.com/vinfo/secadvisories/default6.asp?VNAME=The+ZLOB+Show%3A+Trojan+poses+as+fake+video+codec%2C+loads+more+threats&Page= , which notes:

TROJ_ZLOB.ALF, for instance, modifies an affected system's registry to alter its DNS (Domain Name System) settings, such that it connects to a remote DNS server that is likely controlled by a remote malicious user. Thus, using this setup, the said remote user can decide what IP address the affected system connects to when the affected user tries to access a domain name.

At the time when it was first detected, TROJ_ZLOB.ALF redirects users to adult-themed sites. Of course, by now the DNS server could have been changed already -- perhaps by the highest bidder it was rented to -- so that connections are redirected to other, possibly malicious, sites instead.

Trojan.Flush.K (1/18/2007)

- http://www.symantec.com/enterprise/security_response/writeup.jsp?docid=2007-011811-1222-99&tabid=2 states:

  'The Trojan then creates the following registry entries: […]
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\[RANDOM CLSID]\"DhcpNameServer" = "85.255.115.21,85.255.112.91"
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\[RANDOM CLSID]\"NameServer" = "85.255.115.21,85.255.112.91"'

- And there are MANY, MANY more. The bad guys ARE attempting to accomplish their goals via your users' reliance on DNS.

The Mechanics: 53/UDP and 53/TCP

- Most DNS queries are made over port 53/UDP, but some queries may return more data than would fit in a normal single DNS UDP packet (512 bytes). When that limit is exceeded, DNS will normally truncate, and retry the query via 53/TCP.
- Occasionally you may run into a site where either 53/**UDP** or 53/**TCP** has been blocked outright for all IP addresses (including real name servers!) at a site. That's a really bad idea.
- Blocks on **all** 53/**TCP** traffic sometimes get temporarily imposed because of the misperception that "all" normal DNS (at least all traffic except for zone transfers) happens "only" via UDP; that is an incorrect belief. Real DNS traffic (other than zone transfers) **can, may and will** actually use 53/TCP from time to time.
- Blocks on **all** 53/**UDP** may sometimes get installed because of concerns about spoofed traffic, or worries about the non-rate adaptive nature of UDP traffic in general, or simply by mistake.

(Less?) Crazy Tweaks to User DNS Traffic

- Because of the high cost of handling user support calls, some ISPs may attempt to avoid user support calls (and associated costs) by "managing" user DNS traffic.
- What does "managing" mean?
  -- **blocking/dropping all** port 53 traffic, **except** to/from the DNS server(s) that the ISP provides for their customers (this will often be implemented via router or firewall filters)
  -- **redirecting** all user DNS traffic that isn't destined for the ISP's customer DNS servers (e.g., redirecting DNS is something that's common enough that Cisco even includes redirecting DNS as an example for its Intelligent Services Gateway, see: http://www.cisco.com/en/US/products/ps6566/products_configuration_guide_chapter09186a0080630d65.html#wp1048400)
  -- **selectively redirecting user DNS traffic**, if it appears that the customer is infected (e.g., Simplicita's commercial DNS switch)

We ARE Coming Up To A Crossroads Again

- Do you remember…
  -- **the good old days before everything was behind a firewall** (or NAT box, or other middlebox), and transparent end-to-end connectivity was still possible?
  -- simpler times **when you had the ability to manage your own desktop**, and configuration and management of your desktop wasn't controlled by a desktop domain admin for security's sake?
  -- **when you could store content locally**, taking responsibility for the management of that data, including its backup and its definitive deletion?
  -- **when you could even run your own mail or web server?**
- As a result of the increasing interest in DNS, you may soon be able to add to that list, "***Do you remember when you could directly access domain name servers other than just those provided for your use by your provider?***"

Just "For the Record…"

- I am generally **not** a big fan of **redirecting or rewriting all customer DNS traffic, or limiting users to just their provider's DNS servers** as a "solution." Why?
  -- doing DNS filtering/redirection breaks Internet transparency in a very fundamental and bad way, as I've mentioned
  -- if the provider's designated DNS servers end up having issues, DNS filtering/redirection substantially reduces customer options
  -- port-based filtering/redirection can be surmounted by technically clued people thru use of non-standard ports for DNS
  -- port-based filtering/redirection (or even deep packet inspection approaches) can be overcome by VPN-based approaches
  -- some services (such as commercial DNSBLs) may be limited to just subscribing DNS servers; the DNS server that you redirect me through may not be allowed to access that data.
- **I would encourage you to consider passive DNS monitoring as an alternative way of identifying systems which need attention.**

What About Blocking *JUST* Malicious DNS Servers at the Network Level?

- Assume you succeed in identifying one or more malicious name servers being used by your users. Most security folks would then be inclined to do the "logical" thing and block access to those name servers. Good, right? You're protecting your users by blocking access to just those servers, eh? Well… *yes*, you are, but when you do so, when you block those malicious name servers, ALL name resolution for those infested users (crumby though it may be), will typically suddenly cease. "The Internet is down!"
- **Suggestion: IF you DO decide to block specific malicious DNS servers, and I CAN sympathize with the desire to do that, be SURE to notify your support staff so that they can add DNS checks to their customer troubleshooting processes.**
- **A nice resource for folks who want to do this sort of blocking: http://www.bleedingsnort.com/blackhole-dns/**

Note: You May End Up Blocking Bad DNS Servers W/O Knowing You're Doing That

- For example, assume you're using the Spamhaus DROP (Do Not Route or Peer list, see http://www.spamhaus.org/DROP/), an excellent resource you should all know about and consider using.
- Some of those DROP listings **may** happen to cover bad DNS servers which will no longer be reachable by infected clients once you begin using DROP.
- Thus, even though you may not be focused on blocking bad DNS servers, by filtering some prefixes at the network level, you may inadvertently end up filtering name servers your users may be using.
- Isn't this all just so much "fun?"

Users May Tinker With The Hosts File, Too

- Remember those old host files I mentioned earlier? Well, you can still statically define FQDN to dotted quad relationships using a hosts file, and some folks take advantage of that, particularly in an effort to thwart adware or spyware or online advertising (when that's the objective, unwanted sites are generally mapped to 127.0.0.1, a special address that always maps to the local system). Examples of hosts files that are in circulation for that sort of purpose include:

  http://mvps.org/winhelp2002/hosts.htm

  http://www.hosts-file.net/
- Features in Vista may attempt to deter this, but workarounds exist, (e.g., see http://support.microsoft.com/kb/923947)
- Speaking of Microsoft and hosts files, note that Microsoft sometimes intentionally ignores hosts files (see http://www.securityfocus.com/archive/1/431032/30/0/threaded[41])

Interesting Things Can Happen to DNS on An Application-by-Application Basis, Too…

- http://www.codeproject.com/internet/DnsHijack.asp …

  "Here's what DnsHijack enables you to do:

  -- It allows you to rewrite DNS requests for a single Windows process (in this case, it's hard-coded to firefox.exe, but the technique works equally well for any standard Winsock-using application).

  -- You can rewrite to another DNS name instead of to just an IP address. There's no need to manually perform DNS lookups when creating the configuration file.

  -- It supports Perl-compatible regular expressions (using the PCRE library and some C++ wrapper classes I created for my xp_pcre library). This means you can rewrite multiple DNS names using a single line in the configuration file. [continues]"

MS Windows and DNS Cache Pollution

- While we're talking about DNS and Windows, some early versions of MS Windows, such as Windows NT and pre-SP1 versions of Windows 2000, are vulnerable to what Microsoft refers to as "cache pollution" (for Microsoft's description of this vulnerability, see: http://support.microsoft.com/kb/316786). While Windows NT should not be used at all at this time, and Windows 2000 users should be running with the latest Service Pack installed, if you **do** happen to have someone running an early version of MS Windows, make **sure** they upgrade or see: "How to prevent DNS cache pollution," http://support.microsoft.com/kb/q241352/
- What about Windows 2003? With 2003 you'll be protected by default but make sure that Windows Server 2003 admins **do NOT uncheck** the pre-checked "prevent cache pollution" box!
- For a listings of sites known as attempting to do poisoning see: dns.measurement-factory.com/cgi-bin/poison_browser.pl

4. DNSSEC: What Is It?

DNSSEC "By the [RFC] Numbers"

- DNSSEC is defined by three RFC's:
  -- RFC4033, "DNS Security Introduction and Requirements,"
  -- RFC4034, "Resource Records for the DNS Security Extensions,"
  -- RFC4035, "Protocol Modifications for the DNS Security Extensions"

  If you really want to know about DNSSEC, read those RFCs.
- A couple of other RFC's you may also find useful along the way:
  -- RFC3833, "A Threat Analysis of the Domain Name System"
  -- http://www.ietf.org/internet-drafts/draft-ietf-dnsext-nsec3-09.txt "DNSSEC Hashed Authenticated Denial of Existence" (expires July 9, 2007

- RFCs can make for rather dry reading, however, so let me just dive right in with my personal take on DNSSEC…

DNSSEC in a Nutshell

- DNSSEC uses public key asymmetric cryptography to guarantee that if a DNS resource record (such as an A record, or an MX record, or a PTR record) is received from a DNSSEC-signed zone, and checks out as valid on a local DNSSEC-enabled recursive name server, then we know:

  -- it came from the authoritative source for that data
  -- it has not been altered en route
  -- if the server running the signed zone says that a particular host does not exist, you can believe that assertion

- But what about other things, like insuring that no one's sniffing your DNS traffic, or making sure that DNS service is always available?

DNSSEC Intentionally Focuses on Only One of The Three Traditional Information Security Objectives

- While there are three "C-I-A" information security objectives:

  -- Information **Confidentiality**
  -- Information **Integrity**, and
  -- Information **Availability**

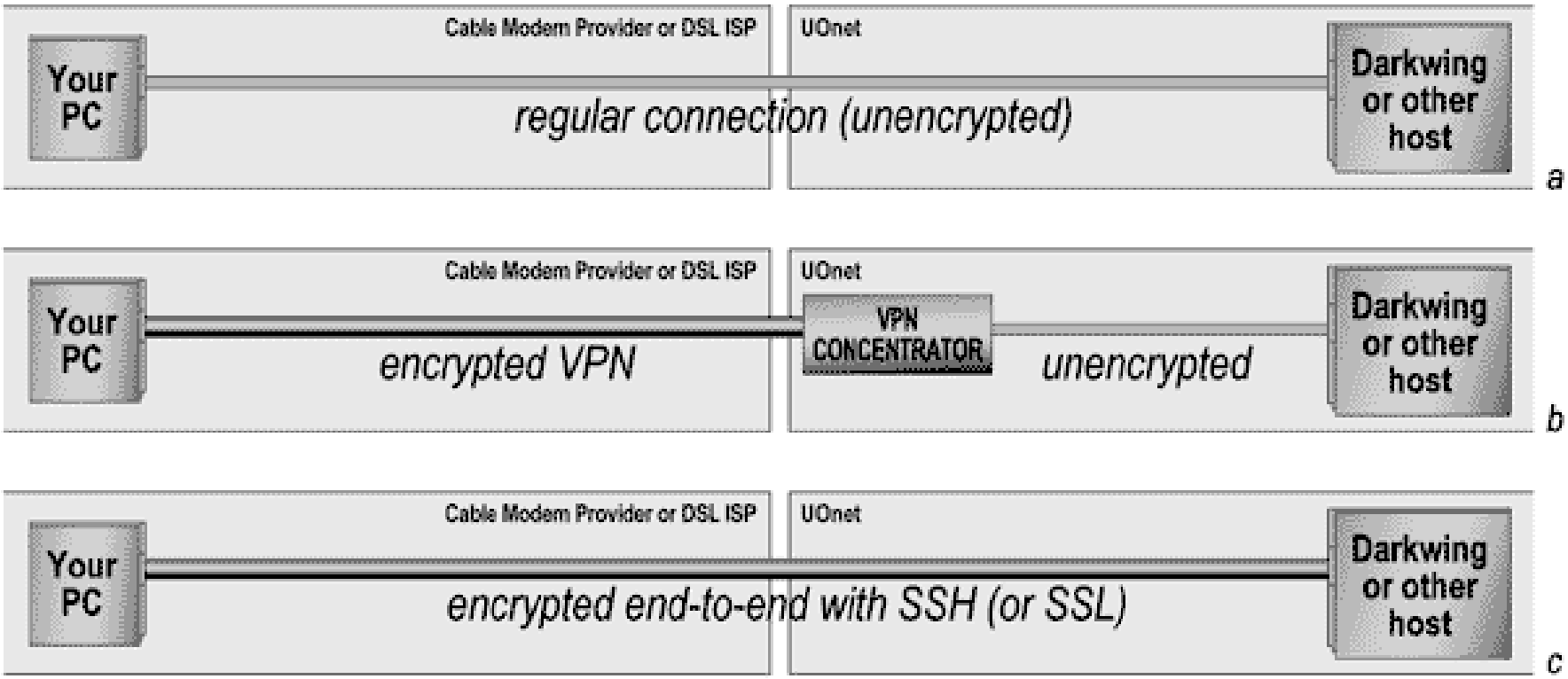
  DNSSEC is intentionally **NOT** designed to keep DNS data confidential, and it is also intentionally **NOT** designed to improve the availability of DNS data -- it's sole focus is on insuring the **integrity** of DNS data.

- And, to the extent that DNSSEC is not an end-to-end protocol, its ability to even insure information integrity is imperfect.

DNSSEC As A Non-"End-to-End" Protocol

- To understand the difference between an end-to-end protocol and one that works only along part of a complete path (e.g., to or from some intermediate point), consider the difference between using SSH and using a typical VPN.
- SSH secures traffic all the way from one system (such as your laptop) to the other system you're connecting to (perhaps a server running Linux) – it is "end-to-end."
- A VPN, however, may terminate on a hardware firewall or VPN concentrator, and from that point to the traffic's ultimate destination, traffic may travel unsecured. This is NON end-to-end.
- DNSSEC is more like the VPN example than the SSH example: **DNSSEC only secures traffic to the local recursive name server**, it typically cannot and will not secure traffic all the way down to the desktop. Thus, a bad guy can still attack DNS traffic that is in flight from the local recursive name server to the endhost.

Non-End-to-End and End-to-End Protocols



What About Using TSIG To Secure The Last Hop for DNSSEC?

- TSIG is defined by RFC2845, and was originally created to improve the security of zone transfers, and to provide a secure way by which trusted clients could dynamically update DNS.
- For the purpose of providing DNSSEC with last hop integrity, TSIG has a number of potential shortcomings, including:
  -- it uses a form of symmetric cryptography, so all clients need to be given a copy of a shared secret key (yuck)
  -- the only hashing mechanism defined for TSIG in the RFC is HMAC-MD5, which is no longer particularly robust
  -- clocks need to be roughly in sync (user laptops or desktops often have system clocks which aren't very well synchronized)
- The DNSSEC data validation check could be moved from the local recursive DNS server all the way down to the laptop or desktop itself, IF the DNS server running on the laptop or desktop knew how to do DNSSEC (but that would probably be painful).

Microsoft DNS Client Support for DNSSEC

- Quoting technet2.microsoft.com/WindowsServer/en/library/264820c4-55c7-42d6-9747-432af9556acc1033.mspx?mfr=true

  "Client support for DNSSEC

  "The DNS client does not read and store a key for the trusted zone and, consequently, it does not perform any cryptography, authentication, or verification. When a resolver initiates a DNS query and the response contains DNSSEC resource records, programs running on the DNS client will return these records and cache them in the same manner as any other resource records. This is the extent to which Windows XP DNS clients support DNSSEC. When the DNS client receives the SIG RR relating to the RRset, it will not perform an additional query to obtain the associated KEY record or any other DNSSEC records."

Speaking of Client Layer Stuff, What Would a User See If a DNS Resource Record Failed DNSSEC Validation?

- **Answer: nothing. Users would see nothing** that would indicate a DNSSEC validation failure had occurred. Such a failure is normally "silent" and indistinguishable (to the user) from many other types of DNS failures. It is probably just me, but I've got mixed feelings about DNSSEC validation failures being opaque to users. Instinctively, we know that DNSSEC validation might fail due to:

  -- operational error: it would be good to make sure that's noticed and corrected, and users could act as "canaries in the coal mine"

  -- an active attack; it would be REALLY good to know that's happening!

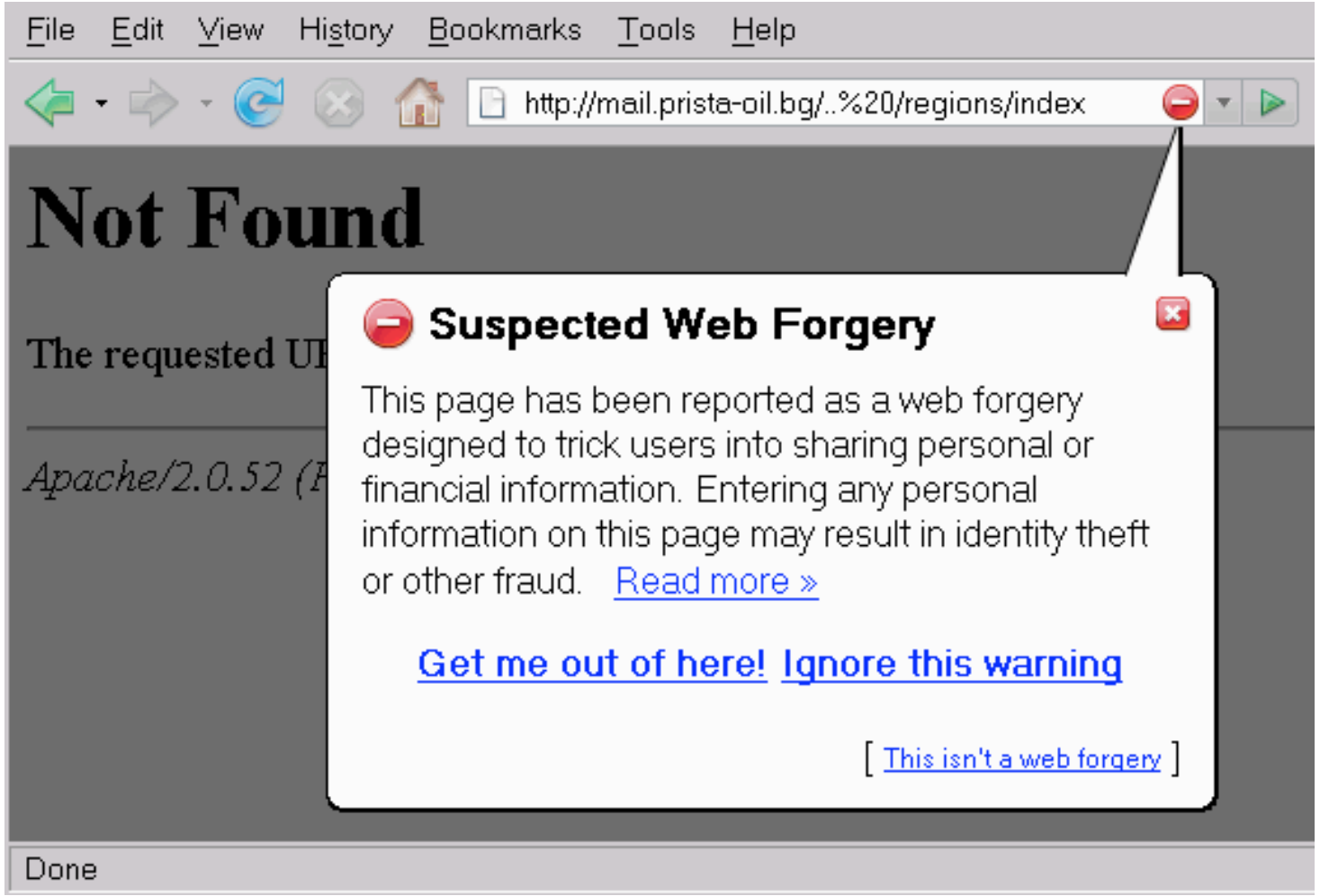
  -- something completely unrelated to DNSSEC might be busted

- Silent failure modes that confound several possible issues just strike me as a bad idea.

DNSSEC and Application Layer Visibility

- DNSSEC **needs** application layer visibility for all the times when it works, kin to the little padlock icon for SSL encrypted secure web sessions (or certificate failure notices for when things are self signed, expired, or otherwise not trustworthy).
- In this, DNSSEC is potentially like Internet2 itself. I'm convinced that one of the biggest (and best!) things about Internet2 AND one of the biggest problems with Internet2 is that it "just works." People use Internet2 all the time with no idea that they're doing so.
- If DNSSEC similarly "just works" (except for when it silently breaks attempts to do bad things), will people even know they're receiving a benefit from it?
- Contrast invisible DNSSEC protection with the anti-phishing protection that Firefox delivers, something that's FAR more "in your face" and visible…

What A Firefox User Sees When Attempting to Visit A Phishing Site



Another Issue: The DNSSEC Trust Model

- Talking about phishing makes me think about trust models.
- Trust models focus on the question of, "Why should I believe you're really you?" "Why should I accept 'your' credentials as being authentic?" This is a pivotal question in cryptography.
- Some crypto protocols, such as GPG/PGP, are decentralized, and employ a "web-of-trust" trust model where I trust your public key because it has been signed by other keys which I recognize/trust.
- Other crypto protocols, such as PKI, are more centralized or "top down." In the PKI model, I trust a particular PKI certificate because it has been signed by a trusted certificate authority ("CA")
- **DNSSEC was originally intended to use a centralized top-down trust model, with a signed root.** The trusted signed root would then sign immediately subordinate TLDs; those TLDs would sign second level domains immediately below them, etc.
- **One slight problem: the root still hasn't been signed.**

Signing The Root (".")

- There are 13 root servers, A through M, representing 155 locations (some of the DNS roots anycast a single root server IP from multiple geographically diverse locations).

- **26th rssac [DNS Root Server System Advisory Committee] meeting - 05nov2006**

  San Diego, prior to IETF67

  http://www.rssac.org/meetings/04-08/rssac26.pdf

  ") **SSAC**
  what is the status of support for a signed root zone?"

  [continues over the next two slides]

- "**A** [Verisign, Dulles VA] yes by eoy [e.g., end of year]

  B [ISI, Marina Del Rey CA] yes by eoy

  C [Cogent, 4 locations] need software upgrade but yes hoping by eoy; asking to be asked

  D [University of Maryland, College Park] not present

  E [NASA Ames, Mountain View CA] not present

  F [ISC, 40 sites] ready needs enabling

  G [US DOD, Columbus OH] ready

  H [US ARL, Aberdeen MD] not present

  I [Autonomica/Nordunet, 29 sites] ready needs enabling

  J [Verisign, 22 sites, going to 70 sites*] yes end of year

  K [RIPE, 17 sites] yes needs enabling

  L [ICANN, Los Angeles CA] not ready. in burn-in by end of year

  M [WIDE] ready"

  * see http://www.nytimes.com/2007/02/08/technology/08net.html

But Someone Needs to Formally Ask…

- "**Root server operators point out that they have not yet been asked to do this**, and that they would need a formal request from the zone administrator with a date on which they will be expected to serve a signed zone. There are concerns regarding discussions of signed .arpa since it is not the root, .arpa discussion should be somewhere else. The zone owner should include the root ops in any discussion of planning, not just dates when they think they might be ready. **Actual target dates would be very helpful**, preferably with at least 30 days notice."
- **Who asks?** From: http://www.icann.org/general/bylaws.htm …

  "ICANN: […] 2. Coordinates the operation and evolution of the DNS root name server system."

  [bracketed additions and bolding by me; root server operator identities and location counts from http://www.root-servers.org/]

What About The TLDs? Are The TLDs At Least Signed and Supporting DNSSEC?

- A very limited number are. For example, .se (Sweden) is signed:

  % dig +dnssec +bufsize=4096 se @catcher-in-the-rye.nic.se

  [snip]

  ;; AUTHORITY SECTION:

se.	7200	IN	SOA	catcher-in-the-rye.nic.se. registry.nic-se.se. 2007021008 1800 […]
se.	172800	IN	TYPE46 \# 150	000605010002A30045D5084B45CDD157E86502736500E […]
se.	7200	IN	TYPE47 \# 17	03302D3002736500000722008000000380
se.	7200	IN	TYPE46 \# 150	002F050100001C2045D3453445CC9BF7E865027365000 […]

- Most other TLDs (including .edu, .com, .net, .gov, .mil, .ca, .cn, .de, .fr, .jp, .uk, etc.) are **NOT** signed nor supporting the use of DNSSEC at this time. This does not prevent domains **under** those TLDs from doing DNSSEC, but when a domain under one of those TLDs does do DNSSEC, they exist as an "island of trust."

Islands Of Trust

- Remember, DNSSEC was designed to work using a **centralized, top-down trust model**. If the root isn't signed, all the stuff under the root must establish **alternative trust anchors**. In some cases (such as .se), the trust anchor may be the TLD, but in other cases, the trust anchor may be 2nd-level domain (such as nanog.org).
- Because there is **no central trust anchor**, unless you can come up with an alternative way of establishing a chain of trust, **you must obtain trustworthy keys for each of those individual islands of trust**. (Key management is the 2nd thing, after trust models, to always scrutinize when considering about a crypto effort!)
- If each site that wants to do DNSSEC has to do a "scavenger hunt" for each island of trust's DNSSEC keys, that's **rather inconvenient** particularly if (1) trust islands periodically **rekey**, (2) there are **thousands** of domains, and (3) given that if a site **fails** to keep each trust island's keys current, any data served by that trust island with their new key will be mistakenly viewed as bogus and get dropped.

DLV

- To avoid these problems, ISC has proposed DLV (Domain Lookaside Validation) as a temporary/transitional model.
- In the DLV model, even if the root or a TLD isn't ready to support DNSSEC and sign its zone, perhaps a trusted third party can collect, authenticate and deliver the required keys. Someone attempting to do DNSSEC then has only to configure the DLV server or servers as an anchor of trust, thereafter automatically trusting domains that are anchored/validated via the DLV.
- DLV is described at http://www.isc.org/pubs/tn/isc-tn-2006-1.html and in http://www.ietf.org/rfc/rfc4431.txt
- DLV is supported in BIND 9.3.3, 9.4.0 and later.
- One sample DLV registry: http://www.isc.org/index.pl?/ops/dlv/ (and there may/will be others). Obviously, assuming you need to trust the data that a DLV registry secures, you will want to be extremely careful when adding trusted DLV registries. (Needless to say, I'm quite comfortable trusting ISC's DLV registry)

What About the In-Addr Zones?

- In addition to the root and the TLDs, the rDNS ("inverse-address") zones would also be a top priority for DNSSEC signing.
- RIPE has signed the in-addrs that it is responsible for (see https://www.ripe.net/projects/disi/keys/), however other registries (such as ARIN, APNIC, LACNIC, etc.) have yet to do the same for the in-addr zones they control.
- It would be great to see progress in that area, along with getting the root and/or major TLDs signed.

The Zone Enumeration Issue And NSEC3

- As originally fielded, DNSSEC made it possible to exhaustively enumerate, or "walk," a zone, discovering all known hosts. An example of such as tool is Zonewalker, http://josefsson.org/walker/
- Zone enumeration give miscreants a real "boost up" when it comes to reconnoitering a domain, and this was a real problem for some TLDs in countries with strong privacy protections.
- NSEC3, currently in draft (see http://www.ietf.org/internet-drafts/draft-ietf-dnsext-nsec3-09.txt), addresses the zone enumeration issue through use of salted hashes, which handles both that concern as well as the problem that "the cost to cryptographically secure delegations to unsigned zones is high for large delegation-centric zones and zones where insecure delegations will be updated rapidly."
- For our purposes, it is sufficient to know that NSEC3 effectively eliminates the zone enumeration problem.

Are Name Servers (the Software Programs) DNSSEC-Ready?

- Another potential stumbling block might be the name server software. If the name server software you use doesn't support DNSSEC, your ability to do DNSSEC will obviously be limited.

- First, what name server products do people run?

BIND Dominates The DNS Server Market

- http://dns.measurement-factory.com/surveys/200608.html …

BIND 9	201,723	60.74%	
BIND 8	45,547	13.71%	
BIND 4	1,387	0.42%	**(74.87% total)**
Embedded Linux	51,720	15.57%	
Microsoft Windows DNS 2000	11,548	3.48%	
Microsoft Windows DNS 2003	3,246	0.98%	
Microsoft Windows DNS NT4	868	0.26%	(4.72% total)
PowerDNS	14,448	4.35%	
Other (including Cisco CNR)	1,623	0.49%	

["122,188 additional nameservers could not be identified"]

Let's Start With The Good News: Current Versions of BIND Support DNSSEC

- The good news for folks interested in deploying DNSSEC is that the current version of BIND supports DNSSEC, and BIND has the lion's share of the current DNS server market, as shown by the table on the proceeding page.

- I must admit that I am a little disconcerted to see ancient versions of BIND still in use – are people REALLY running BIND 4? You really don't want to be running ancient versions of **anything** on systems exposed to the Internet these days! Job one is to get current!

What About Microsoft's DNS Servers?

- Quoting technet2.microsoft.com/WindowsServer/en/library/264820c4-55c7-42d6-9747-432af9556acc1033.mspx?mfr=true (updated January 31st, 2005):

  "Windows Server 2003 DNS provides basic support of the DNS Security Extensions (DNSSEC) protocol as defined in RFC 2535." *[however, note that RFC2535 dated March 1999, was made obsolete by RFC4033, RFC4034, and RFC4035 ca. March 2005]* "**The current feature support allows DNS servers to perform as secondary DNS servers for existing DNSSEC-compliant, secure zones.** DNS supports the storing and loading of the DNSSEC-specific resource records (RRs). **Currently, a DNS server is not capable of signing zones and resource records (creating cryptographic digital signatures) or validating the SIG RRs.** The DNSSEC resource records are KEY, SIG, and NXT." [the March 2005 RFC's deprecated those earlier DNSSEC record types]

The Most Recent News From MS on DNSSEC Support in Windows Server

- See "DNSSEC in Windows Server" from http://public.oarci.net/files/workshop-2006/Microsoft-DNSSEC.pdf

  -- driven by NIST 800-53 and SC-20 and SC-21 requirements

  -- implements RFC4033, RFC4034, RFC4035

  -- "Beta: middle of 2007

  RTM: late 2007 or early 2008

  – General availability by first service pack of Longhorn Server"

How About PowerDNS?

- PowerDNS appears to **lack support** for DNSSEC.
- PowerDNS may provides DNS for 10%-20% of all the world's domains according to Bert Hubert's PowerDNS presentation (http://ds9a.nl/pdns/pdns-presentation-ora.pdf), including doing DNS for Tucows, Schlund, etc. However, that same talk states:

  "Things PowerDNS doesn't do
  DNSSEC
  – Perhaps too complicated in its current form."

- See also http://downloads.powerdns.com/documentation/html/changelog.html at "1.3.8. Version 2.9.19, Released 29th of October 2005," which states "support for DNSSEC records is available in the new infrastructure, although is should be emphasised that there is more to DNSSEC than parsing records. There is no real support for DNSSEC (yet)."

What About The Large Number of "Unidentified" Name Servers?

- In some cases those may be sites running one of the mentioned products, but they may have disabled version strings and/or taken other steps to limit the ability of potential miscreants to successfully "fingerprint" the name server software running on their servers.
- In other cases, however, sites may be running an alternative DNS implementation, such as D. J. Bernstein's DJBDNS (aka TinyDNS), see http://cr.yp.to/djbdns.html or http://tinydns.org/
- If you're considering doing DNSSEC and you're currently using those products, you should note that the author of those products explicitly does NOT support DNSSEC in DJBDNS, and to the best of my knowledge has no plans to change that stance. You can see his discussion and rationale for this at http://cr.yp.to/djbdns/blurb/security.html and at http://cr.yp.to/djbdns/forgery.html

What About The "Embedded Linux" Name Servers Which Were Mentioned in The Survey of DNS Software Usage?

- Embedded Linux is a stripped down version of Linux that's often run on hardware network appliances, including at least some DSL or cable modems, and some "firewall"/"broadband router" devices.
- Based on the survey numbers, I believe at least some those hardware network devices offer DNS service as well as other functions.
- I'm not sure anyone has even begun to think about how DNSSEC might interact with those home hardware firewall class devices.

EDNS0

- While we're on the topic of network hardware devices such as firewalls, you should know that name servers doing DNSSEC requires a feature known as EDNS0, as defined in RFC2671, "Extension Mechanisms for DNS (EDNS0)," August 1999.
- Normally, DNS UDP responses are limited to just 512 bytes, a size that's too small for the much larger DNSSEC records. To better handle delivery of DNSSEC records, EDNS0 allows clients and servers to negotiate the maximum size datagram which can be handled, with the expectation that at least some hosts might negotiate datagram sizes as high as 4KB. Name servers doing DNSSEC must do EDNS0.
- Why's that a problem? Well… some firewalls may block UDP DNS traffic > 512 bytes. If you've got a firewall in front of your DNS server, please see http://dnssec.nic.se/fw/en.html to make sure you won't need to upgrade your firewall to handle EDNS0.

5. Why Aren't People Using DNSSEC?

Deployment of DNSSEC to Date? NIL

- "The first version (RFC 2535, March 1999) defines the KEY, SIG, and NXT record types. The second version (RFC 4035, March 2005) essentially obsoletes the first-generation RR types and adds four new ones: DNSKEY, NSEC, RRSIG, and DS. We queried the set of nameservers for both old and new RR types. Among the **1,756,827** zones with at least one working nameserver, we found **16 (0.001%)** with **first-generation DNSSEC records**. Coincidentally, we also found **16** zones publishing **second-generation** DNSSEC records. There is no overlap between the two first- and second-generation subsets. Needless to say, DNSSEC adoption is still very small. Unfortunately, our use of the COM and NET zones probably under-represents DNSSEC adoption across the whole Internet. Some European CCTLDs have been more proactive in encouraging the use of DNSSEC." [emphasis added]
- http://dns.measurement-factory.com/surveys/200608.html

Another View of DNSSEC Penetration: UCLA's SecSpider

- SecSpider: The DNSSEC Monitoring Project http://secspider.cs.ucla.edu/ reports (as of Saturday, February 11, 2007) that it knows about just 718 DNSSEC-enabled zones (please note that many of those zones are NOT major/well known zones)

- See also http://public.oarci.net/files/workshop-2006/Osterweil-SecSpider.pdf …

"From our web crawl (of 18M zones), we estimate that the deployment status of DNSSEC is roughly 0.0015%"

Why Aren't Folks Currently Using DNSSEC?

- **Do people simply not know DNSSEC exists?** Well at least that's no longer an excuse for the folks at this Joint Techs session. :-)
- **Are people willing to try DNSSEC, but simply don't know the "recipe" to get going?** If so, let me recommend three resources:
  -- Olaf Kolkman/NLNet Lab's "DNSSEC HOWTO, a tutorial in disguise," see http://www.nlnetlabs.nl/dnssec_howto/
  -- Geoff Huston's three part exploration of DNSSEC:
  http://www.potaroo.net/ispcol/2006-08/dnssec.html
  http://www.potaroo.net/ispcol/2006-09/dnssec2.html
  http://www.potaroo.net/ispcol/2006-10/dnssec3.html and
  -- The RIPE NCC's DNSSEC Training Course:
  http://www.ripe.net/training/dnssec/material/dnssec.pdf
- **Are people waiting for the root zone (or major TLDs) to be signed?** Or are people waiting for more of their peers to take the plunge and report back, first? (EDU land is prone to herd behavior!)

Or Are There More Fundamental Problems?

- Are people just really busy, with slow uptake just the normal resistance to yet one more thing – *ANYTHING* MORE! – to handle without substantial additional resources?
- Does DNSSEC solve what's perceived by the community to be a **"non-existent" or "unimportant" problem?**
- Are there **critical administrative tools** missing? (if that's the issue, then see http://www.dnssec-tools.org/ and http://www.ripe.net/disi/dnssec_maint_tool/)
- Does DNSSEC **demand too many system resources (e.g., does it make zone files too large, or is the CPU crypto overhead too great, or would it swamp the network with additional DNS-related network traffic?)** (Nice discussion of some of increased resource issues at http://www.nominet.org.uk/tech/dnssectest/faq)
- Are people waiting to see what the "big guys" do w.r.t. DNSSEC?

The Biggest Guy Out There

- One of the largest and most influential entities out there is the U.S. Federal government. With adoption of "Recommended Security Controls for Federal Information Systems," NIST 800-53 Rev. 1 (see http://csrc.nist.gov/publications/nistpubs/800-53-Rev1/800-53-rev1-final-clean-sz.pdf) in December 2006, agencies now have a year from December 2006 to begin doing DNSSEC. Relevant security controls from 800-53 Rev. 1 include:
  -- SC-8 "TRANSMISSION INTEGRITY
  -- SC-20 "SECURE NAME / ADDRESS RESOLUTION SERVICE (AUTHORITATIVE SOURCE)"
  -- SC-21 "SECURE NAME / ADDRESS RESOLUTION SERVICE (RECURSIVE OR CACHING RESOLVER)"
- See also NIST SP 800-81, "Secure Domain Name System (DNS) Deployment Guide," May 2006.
- Will required Federal adoption be enough to kick start DNSSEC?

Unfortunately…

- Federal agencies face a HUGE number of information security requirements under FISMA, and in many cases while agencies are working hard to try to comply, they simply haven't been able to fully do so yet. The 6th FISMA Report Card, released March 16th, 2006, shows many federal agencies still able to make only a D or F grade overall (http://republicans.oversight.house.gov/FISMA/FY2005FISMAreportcard.pdf).
- Given the many fundamental computer security issues in play, is there reason to believe that the comparatively obscure issue of DNSSEC, out of all the FISMA requirements laid on Federal agencies, will end up becoming a noteworthy and ubiquitous Federal cyber security success story?
- It is probably fundamentally unfair to expect the federal government to do something which even the most security conscious private entities haven't yet done…

Federal Agencies And Commercial Partners

- Many federal agencies work closely with commercial partners (such as commercial DNS providers & content delivery networks):

gov.	172800	IN	NS	g.gov.**zoneedit.com**.
gov.	172800	IN	NS	f.gov.**zoneedit.com.**
gov.	172800	IN	NS	e.gov.**zoneedit.com.**
gov.	172800	IN	NS	d.gov.**zoneedit.com.**
gov.	172800	IN	NS	c.gov.**zoneedit.com.**
gov.	172800	IN	NS	b.gov.**zoneedit.com.**
gov.	172800	IN	NS	a.gov.**zoneedit.com.**

www.irs.gov.	900	IN	CNAME	www.irs.gov.**edgesuite.net.**
www.navy.mil.	86400	IN	CNAME	prpx.service.**mirror-image.net.**

- Because of that, DNSSEC-ifying some "federal" online resources will likely require active involvement of commercial partners.[80]

Something to Note: DNSSEC Adoption Doesn't Need to Be Symmetric

- When deploying DNSSEC (just as when deploying SPF or DK/DKIM for email), adoption doesn't need to be symmetric:

  -- you can sign your own zones with DNSSEC on your authoritative name servers, yet **not** check DNSSEC on your recursive customer-facing name servers, or
  -- you can check DNSSEC on your recursive customer-facing facing name servers, yet **not** publish DNSSEC records for your own domains on your authoritative name servers

- Most sites will eventually want to "take the whole plunge" (or skip the technology entirely), but sometimes different people have decision making authority for different parts of the organization, and you should recognize that asymmetric adoption is a possibility.

Thanks for the Chance to Talk Today!

- Are there any questions?