Securing DNS:
Doing DNS As If DNS Actually Mattered

An Educause Security Professionals Conference
Pre-Conference Seminar
1:00-4:30PM, Monday, April 12th, 2010
International E (6th Floor), Westin Peachtree Plaza, Atlanta

Joe St Sauver, Ph.D. (joe@uoregon.edu)
Internet2 and University of Oregon Computing Center
http://www.uoregon.edu/~joe/secprof10-dns/

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

Welcome to the Security Professionals Conference and to Atlanta, Georgia!

- Let me be among the first to welcome you to this year's Security Professionals Conference, and to welcome you back to Atlanta if you were also here at Security Professionals in 2009.
- Let me also specifically thank you for coming to this preconference seminar on securing the domain name system.
- I'd like to begin by taking a minute to introduce myself, and then having each of you introduce yourself to the group... if you would, please mention:
  -- your name and the school you're with
  -- where you're at when it comes to DNS issues (beginner? highly skilled? somewhere in between?)
  -- and if you want to, please mention one DNS-related issue, concern or question you'd like to see us discuss during the course of this seminar

Format and Mechanics

- We'll go till 2:30 or so, take a break from 2:30 to 3:00 at the International Foyer area on the 6th floor, and then finish up. If we don't get done by 4:30, I'm happy running later, and conversely, if we finish up ahead of time, I'm okay with that too.
- Because this is a seminar, and we only have a comparatively small number of attendees, I'd like you all to feel free to speak up at any time, whether that's to share your expertise or opinion, or to ask a question. I've prepared some material, but I don't mean for the prepared material to be the only thing we cover today.
- Also note that some topics we'll cover in depth, other topics we only allude to, perhaps providing a link for more information.
- Speaking of links, copies of these slides are available online at http://www.uoregon.edu/~joe/secprof10-dns/ in PowerPoint and PDF formats.

1. Why Worry About DNS?

DNS is powerful, ubiquitous and largely ignored.
That's a very dangerous combination.

Virtually All Applications Rely on DNS

- Email
- The world wide web
- Peer to peer applications
- Instant messaging
- Voice over IP, etc., etc., etc.

- Virtually ALL applications are built on top of DNS, and rely on DNS to function. This puts DNS in a radically different role than an application such as FTP – if FTP doesn't work, everything else will continue to function, but that's not true of DNS! If DNS is down, everything else also tends to come to a screeching halt.
- 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 can control your world.**

- Going to eBay? Doing some online banking? Sending important email? Maybe, maybe not, depending on what sort of DNS resolution occurs. If a bad guy controls your DNS, he can send you to a convincing alternative site under his control…

- "But, but… even if the bad guys hijack my DNS, the fake website they might have set up won't have the right SSL certificate!"

  In my experience, SSL certificate issues are not enough to flag DNS misdirection as an issue -- users just don't get the whole certificate thing, and will just blindly accept any self-signed certificate they've been handed for a "secure" site.

Users Really Don't "Get" DNS, Either…

- Just as most non-technical users don't "get" subtle SSL certificate-related issues, most non-technical users also don't "get" DNS.
- Because DNS is, or can be, complex, and because non-technical users generally don't need to understand DNS to use the Internet (at least when everything is working the way it is supposed to), many people never bother to learn anything about DNS -- it just works, and they blindly and trustingly rely on it.
- Unfortunately, because DNS usually "just works," users are not sensitized to the ways that DNS can be perverted or corrupted by a miscreant, and DNS-related areas are not the focus of most consumer-grade system security review tools.
- This increases the need for technically-oriented security professionals -- you folks! -- to pay attention to DNS on behalf of your non-technical users.

The Bad Guys and Gals Are Interested in DNS & Do Understand DNS-Related Vuln's

- **Miscreants can (and have!) attacked the trustworthiness of DNS data** on a variety of levels, including:
  -- doing cache poisoning, where misleading results are seeded into the DNS data that many DNS servers save locally, eventually getting provided to local users even though it's inaccurate
  -- releasing malware that tweaks host file entries and/or DNS registry entries on the PC, so the bad guys send you directly to the wrong web site rather than the web site you'd intended
- Some hacker/crackers also view DNS as a convenient mechanism whereby they can limit user access to key resources, such as antivirus updates needed for the remediation of infections
- The bad guys also recognized DNS is a key enabling technology for botnet command and control survivability

DNS: A City Vaporizing Death Ray?

- Sometimes security guys are accused of sowing fear, uncertainty and doubt (FUD), but truly, DNS is potentially an incredibly potent "death ray." Why do I say that?

  -- There are **millions** of DNS servers deployed on the Internet.

  -- **DNS uses UDP**. Because of that, **DNS has issues when it comes to accepting and responding to 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**.

  There's also the simple reality: we've seen DNS servers used to conduct some of the largest DDoS attacks we've seen to date.

- We'll talk more about this later in this talk.

Speaking of DDOS, DNS Servers Are A Prime Target for DDoS, Too…

- Name servers aren't just a tool for conducting distributed denial of service attacks, customer-facing recursive **DNS servers are also a target for distributed denial of service attacks**: if I can kill the DNS servers your customers are using, you are off the network even if your transit links aren't flooded with traffic.

DNS Services Have Been Broadly Neglected

- **DNS has traditionally not been a focus of institutional love and investment**. When it comes to DNS, lots of people are running:

  -- old code,
  -- on old gear,
  -- with crude operational tools,
  -- a low level of redundancy,
  -- poor service monitoring and
  -- part time or student (rather than fulltime) DNS administrators.

- DNS isn't "cool."

"When I Grow Up, I Want to Be A *DNS Administrator*!"

- Doing DNS for a university is not a particularly glamorous or high prestige job (few novices aspire to some day become a DNS administrator – they all want to work in Marketing, instead. :-))
- To the best of my knowledge, there are no routinely scheduled reoccurring conferences devoted exclusively to DNS-related research or operational praxis, with the exception of ISC's OARC meetings (see https://www.dns-oarc.net/)
- An effort by ICANN staff to create a DNS-CERT has not exactly been enthusiastically embraced (see http://www.icann.org/en/public-comment/#dns-cert)
- 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 Is No Longer Just for Translating Domain Names to IP Addresses

- **DNS has become a general-purpose distributed database**.
- DNS block lists, as used to block spam, are one example of non-traditional data distributed via DNS, and 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, which I'd like to note 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
- That comment notwithstanding, people are now doing wild stuff.

Some Personal Favorites…

- …in the "**no,-this-is-<u>not</u>-what-we-intended DNS to be used for**" 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," www.blackhat.com/presentations/bh-europe-05/BH_EU_05-Kaminsky.pdf)
  Two other great Kaminski DNS-related talks are "Black Ops 2004@LayerOne," see http://www.defcon.org/images/defcon-12/dc-12-presentations/Kaminsky/dc-12-kaminsky.ppt
  and "Black Ops of TCP/IP 2005," see http://www.blackhat.com/presentations/bh-jp-05/bh-jp-05-kaminsky/bh-jp-05-kaminsky.pdf

- **Note well:** sites may view "atypical" DNS usage as hostile/illegal.

Always Keep Your Hair Cut, Your Shoes Shined and Your Tie Carefully Knotted…

- **Your DNS (or, more precisely, your rDNS) may determine how some people decide to treat your email and other network traffic**. For example, some ISPs check that rDNS exists for a host that is attempting to send mail. **No rDNS?** For a growing number of sites that means, "Sorry, we won't be able to accept email from that dotted quad…" For instance, see http://postmaster.aol.com/guidelines/standards.html and http://help.yahoo.com/l/us/yahoo/mail/postmaster/basics/postmaster-15.html

- Other sites may also be on the lookout for dynamic-looking rDNS host names when deciding whether to accept or reject direct-to-MX email. Have rDNS which looks dynamic? Again, for many sites, that means "Sorry, but we won't be accepting email directly from you, send it via your provider's official SMTP servers…"

Examples of "Dynamic Looking" rDNS

- adsl.nuria.telefonica-data.net
  cable.mindspring.com
  dhcp.vt.edu
  dialup.hawaii.edu
  dorm.ncu.edu.tw
  dsl.telesp.net.br
  dyn.columbia.edu
  dynamic.hinet.net
  dynamicip.rima-tde.net
  fios.verizon.net
  resnet.purdue.edu
  student.umd.edu
  user.msu.edu
  wireless.indiana.edu
- See Steve Champeon's rDNS-based list at http://enemieslist.com/

Standardizing rDNS Nomenclature

- There are efforts underway in the IETF to encourage consistent use of rDNS, and to standardize rDNS naming practices. Two drafts you should be aware of:

  -- Considerations for the Use of DNS Reverse Mapping
  http://www.ietf.org/internet-drafts/
  draft-ietf-dnsop-reverse-mapping-considerations-06.txt
  (expired)
  -- Suggested Generic DNS Naming Schemes for Large Networks and Unassigned hosts
  http://tools.ietf.org/id/
  draft-msullivan-dnsop-generic-naming-schemes-00.txt
  (also now expired)

- **What do <u>your</u> campus rDNS naming conventions look like?**

DNS Interacts With Lots of Other Things

- For example, how do hosts learn which DNS servers they should be using? Users of static IP addresses may be given static DNS server configuration information, but most users who are using dynamic addresses will get their DNS server information from **DHCP** at the same time they receive an IP address to use.
- Thus, if you care about the security of DNS, you really want to pay attention to the security of DHCP, too. Why? If you don't pay attention to the security of DHCP, the bad guys and gals can attack the security of your DNS indirectly, by **attacking DHCP**.
- The attack would not have to be hard: for example, imagine a rogue DHCP server sitting on the wire and listening for DHCP requests… first server to respond to a DHCPDISCOVER with a DHCPOFFER typically "wins”
- Sample DHCP malware: isc.sans.org/diary.html?storyid=6025
- Nice tool: http://www.net.princeton.edu/software/dhcp_probe/

DNS Also Interacts With NTP (Time)

- Just as DNS and DHCP are tightly coupled, you should also know that DNS can also rely critically on accurate system clocks (so you're heavily pushing NTP on your campus, right?)
- Two examples:

  -- From the the BIND FAQ
  (http://www.isc.org/software/bind/faq):
  "**Q:** I'm trying to use TSIG to authenticate dynamic updates or zone transfers. I'm sure I have the keys set up correctly, but the server is rejecting the TSIG. Why?
  "**A:** This may be a clock skew problem. Check that the clocks on the client and server are properly synchronised (e.g., using ntp)."

  -- If you're trying to identify who was using a dynamic IP address at a given time, it can be critical to have accurate time stamps (including time zone information!)

DNS May Control Access To Resources

- Consider, for example, a site-local resource, like a USENET News server, or a site-licensed database. Access to those resources may be controlled by password, or by limiting access to a particular network range, **but** many times access is controlled by limiting access to a particular domain, e.g., "If the connection is coming from an IP address which has the rDNS of *.uoregon.edu, allow access to that resource."
- Of course, it is entirely possible that a bad guy or bad gal might create a bogus in-addr for a non-institutional address, thereby pretending to be part of a domain to which they really don't belong; checking to make sure that the forward address and the reverse addresses seen agree helps reduce the magnitude of this issue, but this is still a fundamentally weak approach to the problem of controlling access.
- Relying on rDNS means that location can be a replacement for identity (all I need is an open jack somewhere and I'm "okay").

DNS May Play An <u>Infrastructural</u> Role

- For example, DNS can be used for traffic management and load balancing, perhaps with DNS selectively returning different dotted quads based on a query's geographical or organizational source.
- Yes, for most of us this is inconsistent with the goal of having consistent information returned regardless of query source, but highly tailored non-uniform DNS operation is highly valued by some commercial sites which may want to do things like:
  -- send users to a topologically "close" server farm
  -- serve a internationalized, language appropriate version of their web site, perhaps in German for users coming from IP's known to be located in Germany, French for users coming from IP's known to be in France, etc.
  -- display a specially tailored version of their web site for particularly important customers, or a version that has had unacceptable content removed for particular cultural venues

Round Robin DNS vs. Load Balancers

- Another example of how DNS may be used to manage traffic can be seen in the use of round robin DNS, where multiple IPs are bound to a single fully qualified domain name (FQDN).
- When doing round robin DNS, name servers sequentially return each defined dotted quads in turn, providing a sort of crude (and potentially multi-site) alternative to dedicated load balancers such as Ultramonkey (see http://www.ultramonkey.org/)
- The down side to doing round robin DNS instead of something more sophisticated? Potentially many things, including:
  -- caching can screw things up (delay changes in configurations)
  -- load division is crude at best, and not load aware in any way
  -- if you "lose" a host in an N-host round robin, every 1-in-N times someone tries to access that site, there will be a failure
  -- failed hosts do not get automatically removed from the rotation
  -- debugging round robin DNS issues can be a real pain

DNS Can Affect Network Planning

- How much load will your DNS servers (and network) see? Choice of DNS TTLs (time to live) may directly impact that…
- Speaking of DNS TTLs, if your DNS servers are temporarily down, how long will sites on the network continue to use cached values? (And is this caching good, or does it just help us conceal (rather than fix) substandard DNS infrastructure?)
- Still thinking about DNS TTLs, if you experience a disaster and need to move servers, how long will it take for cached values to "cook down" so that new DNS values can be noticed?
- What about dynamic addresses? How long should dynamic address leases be? How big should DHCP pools be?
- 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 Can Interact With And Impact Policy

- DNS can interact with policy issues in myriad interesting ways.
- For example, what does your campus privacy policy say about DNS server logs? Has your site even thought about why DNS server logs may be sensitive? (Perhaps some member of your community has an embarrassing health condition, and the DNS server logs expose that condition by documenting visits to a site for those suffering from chronic hemorrhoids (or acute leukemia)). Or what if a key employee is suddenly resolving domain names associated with executive recruiters or online job sites?
- A second, completely unrelated DNS policy example: will you allow non-campus domains to be registered and pointed at campus IP addresses? Will you allow campus domains to be hosted on non-campus IP addresses? Why or why not? Does it matter if your campus "official athletics" site has a non-institutional domain name and uses a non-institutional IP address? (think about searching!)

Some DNS Policy Areas

- Who/what organization does DNS for the campus?
- Who can get DNS service from that organization?
- Is there a charge for this service?
- What's an acceptable DNS name?
- What if the FQDN I want is already taken? Can I “bump” them?
- Can I get a subdomain?
- What determines if I get a static or dynamic address?
- Can institutional FQDNs point at non-institutional IPs?
- Can non-institutional FQDNs point at institutional IPs?
- Does it matter if a domain is a .edu instead of a .com or .org or .net or .us or something else?

- And many more areas…

Does Your Campus Have a DNS Policy?

- Quite a few colleges and universities now have DNS policies. Some sample policies (by no means an exhaustive list!) include:

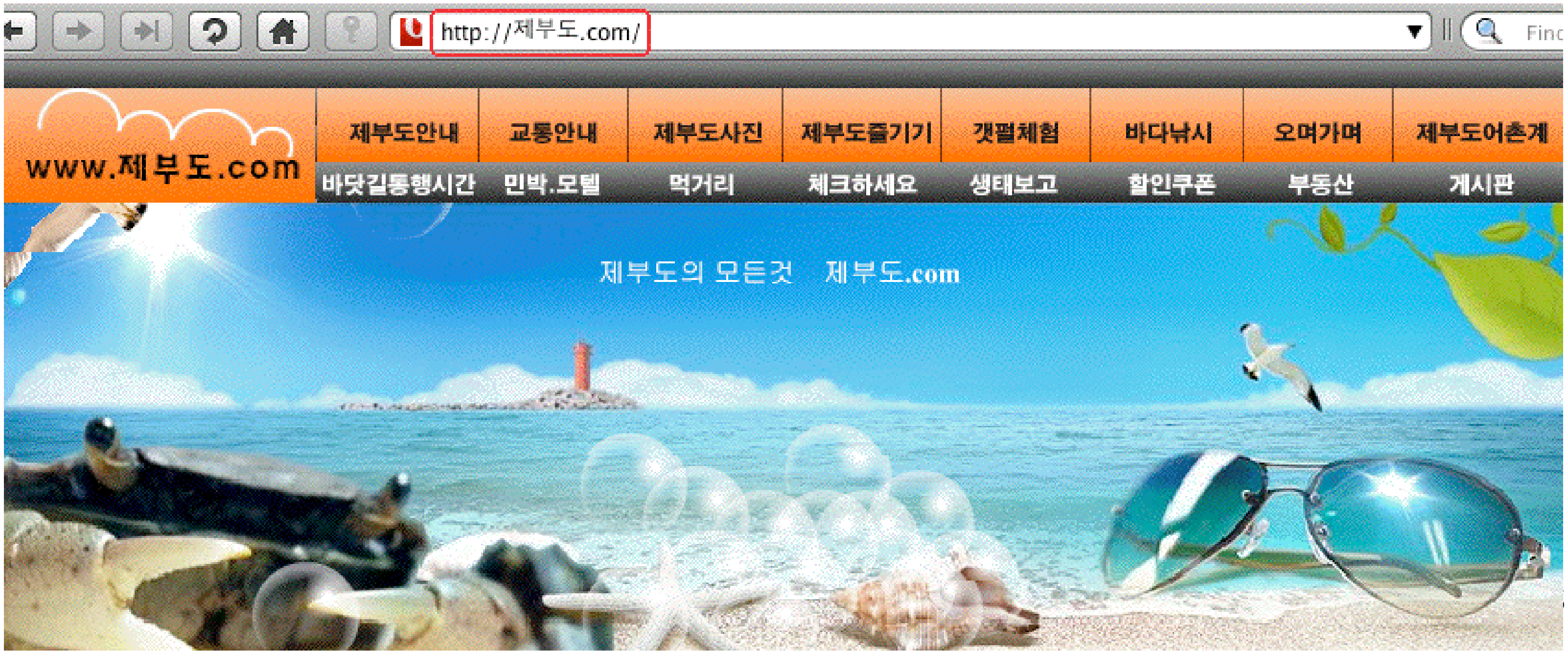
Arkansas:	http://www.uark.edu/~uarkinfo/CAC/CAC4-4-00_DNS.html
Berkeley:	http://net.berkeley.edu/policy_review/DNS.new.shtml
Cincinnati:	http://www.uc.edu/ucomm/web/dns.html
Cornell:	http://www.dfa.cornell.edu/dfa/cms/treasurer/policyoffice/policies/volumes/informationtech/upload/vol5_6.pdf
Florida:	http://www.webadmin.ufl.edu/policies/domain_name/
Indiana:	http://kb.iu.edu/data/aqeo.html
Iowa:	http://cio.uiowa.edu/Policy/domain-name-policy.shtml
Iowa State:	http://policy.iastate.edu/policy/dns
KS State:	http://www.k-state.edu/cns/policy/dns_policy.html
Michigan:	http://spg.umich.edu/pdf/601.15-1.pdf
Nevada Reno:	http://www.it.unr.edu/pages/policy-domain-name.aspx
Oregon State:	http://oregonstate.edu/net/info/policy/domain_policy.php
NYU:	http://www.nyu.edu/its/policies/dnsserv.html
Penn State:	http://tns.its.psu.edu/networking/psuDNS.cfm
Vanderbilt:	http://its.vanderbilt.edu/dns_policy
WU St Louis:	http://www.wustl.edu/policies/domain.html

Another Int'l Policy Example: IDN

- Since we're westerners and use a Roman alphabet, we probably give scant thought to all the folks abroad who may wish they could use accented characters, or Greek letters, or Kanji, or Hangul, or Cyrillic letters as part of domain names…
- Surely accommodating the diverse needs of those with non-Roman character sets can only be good, right? Why would that raise policy issues? There are many reasons, including:
  -- can all name servers technically accommodate non-Roman names?
  -- what representation should be used for foreign character sets? Choices are potentially legion (and sometimes highly political)
  -- what about internationalized names which look *almost* the same as already registered names belonging to banks or other phishing targets? (this is often called a homographic attack; see http://www.shmoo.com/idn/homograph.txt for more info)

Internationalized Domain Names Today

- IDNs have come a long way in the last few years.
- Most web browsers now support for IDNs, and 19 internationalized TLDs representing 11 languages have been requested as of April 2010 (see http://icann.org/en/topics/idn/fast-track/ and http://icann.org/en/topics/idn/fast-track/string-evaluation-completion-en.htm)
- IDNs are currently available for some existing TLDs (e.g., in dot com one can register punycoded domains: http://xn--hq1bp8p1yi.com/)



http://제부도.com/

Some Additional Reasons Why You Will Also Want to Pay Attention To DNS…

- **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 commercial 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

We don't want to turn you into DNS administrators, but we do need to agree on some terminology and provide a little historical background.

What The Domain Name System Does

- Pretty much everyone here 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** or "master" 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

Some <u>Are</u> 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's everyone else, who just wants DNS to keep working…

- Let's talk about one of the biggest threats to DNS, spoofed traffic used as a denial of service attack tool

3. Spoofed (DNS and Other) Traffic and Distributed Denial of Service Attacks

First Important Job:
Please check that your network is configured to prevent spoofed traffic from leaving your network.

Distributed Denial of Service (DDoS) Attacks

- As discussed in my May 3, 2005 Internet2 Member Meeting talk, "Explaining Distributed Denial of Service Attacks to Campus Leaders," http://www.uoregon.edu/~joe/ddos-exec/ddos-exec.pdf), in a distributed denial of service (DDoS) attack network traffic from thousands of hacked computer systems -- often systems located all over the Internet -- gets used in a coordinated way to overwhelm a targeted network or computer, thereby preventing the target from doing its normal work.

- Unlike that earlier general talk, today we **do** need to talk a little about a specific technical vulnerability. We need some quick background, first.

TCP and UDP Traffic

- There are basically two types of network application traffic: TCP and UDP.

- TCP traffic is associated with relatively persistent connections (such as ssh sessions, web traffic, email, etc.), and has a variety of characteristics which are desirable from a network application programmer's point of view, including retransmission of lost packets, congestion control, etc.

- UDP traffic, on the other hand, is designed for "send-it-and-forget-it" applications where you don't want to/can't afford to maintain state or you don't want a lot of connection setup overhead.

- DNS, NFS, and IP video traffic all normally run as UDP.

The Spoofability of UDP Connections

- Unlike a fully established TCP connection (which only gets established after a bidirectional handshake is negotiated and which is therefore robust to spoofing attempts),* UDP traffic can be created with virtually **any** apparent source address -- including IP addresses which have no relationship to the traffic's actual origin.

- Network traffic that's intentionally created with a bogus source address is said to be "spoofed."

- If allowed to reach the global Internet, spoofed traffic is generally indistinguishable from legitimate traffic.

* Yes, of course, naked TCP SYNs are also spoofable.

Why Would Anyone Bother to Spoof Traffic?

- If you don't spend time "thinking like an attacker," you might not immediately "get" why an attacker would be interested in spoofing his attack traffic. The answer is actually quite simple: the attacker wants the systems he's using as part of his attack to stay online and unblocked as long as possible.

- Spoofing the source of the attack traffic…

  -- hinders backtracking/identification/cleanup of the system that's sourcing the traffic; and

  -- makes it harder for the attack victim to filter the attack traffic (the spoofed source addresses may be constantly changed by the attacker, and thus do not provide any sort of stable "filterable characteristic").

"So Why Not Just Block All UDP Traffic?"

- Given that UDP can be easily spoofed by the bad guys/bad gals, sometimes you'll hear folks naively propose simply blocking all inbound or outbound UDP traffic (or at least heavily rate limiting all UDP traffic).

- Unfortunately, because some pretty basic services (like DNS) require support for UDP, blocking (or heavily rate limiting) all inbound or outbound UDP traffic is generally **not** a good idea. :-;

- Warts and all, you have no choice but to learn to to live with UDP traffic. :-;

"Well, Can We Block SOME UDP Traffic?"

- For once, the answer is positive: yes, you can block some UDP traffic.

- For example, if you're the University of Oregon and your school has been assigned the IP address range 128.223.0.0-128.223.255.255 there's no reason for systems on your network to be sourcing packets that pretend to be from some other IP address range. We'd filter that spoofed traffic before it leaves our campus.

- This is a pretty basic sanity check, but you'd be surprised how many sites don't bother with even this trivial sort of filter.

Subnet-Level Filtering

- While it is great to prevent spoofing at the university-wide level, that sort of border router anti-spoofing filter does not prevent a miscreant from forging an IP address taken from one of your subnets for use on another of your subnets.

- *Cue subnet-level anti-spoofing filters….*

  You KNOW that hosts on each subnet should ONLY be originating packets with IP addresses legitimately assigned to that subnet, so at the uplink from each subnet, drop/block outbound packets that appear to be "from" any other IP address – another very basic sanity check.

Filtering at Other Levels of Granularity

- Although we've talked about filtering at your border and at each subnet uplink, you could also filter all the way upstream at the regional optical network (“RON”) level/the gigapop level, or all the way downstream at the host level.

- Obviously, the closer you get to the traffic source, the more effective the anti-spoofing filter will be.

  That said, catching at least some problematic traffic at the RON/gigapop level is better than nothing if you can't get your downstream customers to do the right thing closer to the traffic source (but the larger your gigapop, the harder it will be to keep accurate track of all the prefixes that may be in use).

BCP38/RFC2827

- Let me be clear that ingress filtering of traffic with spoofed IP addresses is not new and is not my idea – it is Best Current Practice (BCP) 38/RFC2827, written by Ferguson and Senie in May 2000.

- Unfortunately, despite being roughly ten years old, **many** sites still do **NOT** do BCP38 filtering -- currently 15-24% Internet wide depending on whether you count netblocks, dotted quads or ASNs (see http://spoofer.csail.mit.edu/summary.php)

- **Does YOUR university do BCP38 filtering?**

"So Why Doesn't Everyone Do BCP38 Filtering?"

- "Too hard given the complexity of my network"

- Asymmetric costs/benefits: filtering my network protects you (which is nice), but filtering that traffic "costs" me w/o any tangible/economic "benefits." So what are these “horrible” "costs?"
  -- engineer time to configure and maintain the filters (one time/negligible for most relatively static .edu networks)
  -- overhead on the routers (but if that overhead is material enough to be a "show stopper," you should be upgrading your hardware anyway)

- "Too busy" (or other (frankly sort of lame) excuses)

"What's It To You Anyhow, Bub? Butt Out…"

- Some may question why others should care what they do with their networks – your network, your rules, right? Well, generally yes.

- However in this case, remember that if you're NOT doing BCP38 filtering, your network may be getting used to generate spoofed attack traffic that's pretending to be "from" someone else's network, and that's the point at which what you do (or don't do) potentially affects a lot of other people including the attack target itself, the entity whose IP addresses are being spoofed, etc.]

"So <u>How</u> Should I Be Doing This Filtering?"

- Only you and your network engineering colleagues can make the final decision about the best approach for your network, but you may want to see BCP84/RFC3704, March 2004.
- I would note, however, that strict mode unicast reverse path forwarding ("strict uRPF") is **not** a good idea for the multihomed environment typical of I2 universities due to route asymmetry issues. I would also urge you to review "Experiences from Using Unicast RPF" (January 23rd, 2008) tools.ietf.org/html/draft-savola-bcp84-urpf-experiences-03
- Quoting RFC3704, "Ingress Access Lists require typically manual maintenance, but are the most bulletproof when done properly…"

4. Open Recursive DNS Servers and DNS Amplification Attacks

Second Important Job:
Please make sure your name servers aren't answering recursive DNS queries for random domains for random users.

A Specific Example of UDP Spoofing…

- Since we just got done covering UDP spoofing, talking a little about open recursive domain name servers and DNS amplification attacks seems like a "nice" segue/practical example of why BCP38 filtering is important, while also pointing out another specific vulnerability you should be addressing.

- Again, let's begin with a little more background, however, first.

Authoritative and Recursive DNS Servers

- There are different types of name servers, with "authoritative" and "recursive" DNS servers being the two most important types:

  -- Authoritative servers are definitive for particular domains, and should provides information about those domains (and ONLY those domains) to anyone.

  -- Recursive servers are customer-facing name servers that should answer DNS queries for customers (and ONLY for customers) concerning any domain.

- DNS servers that aren't appropriately limited can become abused.

For Example…

- Consider a situation where a DNS server is recursive AND is open for use by anyone (a server that's cleverly termed an "open recursive DNS server").

- While it might seem sort of "neighborly" to share your name server with others, in fact it is a really bad idea (the domain name system equivalent of running an open/abusable SMTP relay, in fact).

- The problem? Well, there are actually **multiple** problems, but one of the most important ones is associated with spoofed UDP traffic (see how this all ties together? :-;)

Spoofed DNS Attack Scenario

Dramatis personae:

- Attacker, who's working from non-BCP38 filtered network. Let's call him/her "A"
- Attack target – let's refer to that entity as "T"
- Open recursive domain name server on large, high bandwidth pipe, denoted below as "NS"

Act 1, Scene 1:

- "A" generates spoofed DNS queries with "T"'s address as the "source" address of the queries
- "NS" receives the spoofed queries and dutifully returns the "responses" for those queries to "T"
- "A" repeats as desired, thereby DoS'ing "T" via "NS"

Some Spoofed DNS Attack Scenario Notes

- -- From "T"'s point of view, the attack comes from "NS" not from "A"

  -- DNS queries are small and use UDP, so an attacker can readily generate a "large" query volume

  -- DNS response traffic is also UDP, which means that it is insensitive to net congestion.

  -- DNS responses can be **large** relative to size of DNS queries (output/input ratios can run over 8X on most DNS servers, and on servers supporting RFC2671 EDNS0 extensions, observed amplification can exceed 70X).

  -- "A" can employ **multiple spoofed query sources**, and **use multiple NS's** for more traffic (oh boy!)

This Is A Well Known Vulnerability

- I'm not letting the "cat out of the bag" about a big secret; this is a well known/documented threat:

  -- "The Continuing Denial of Service Threat Posed by DNS Recursion, " see http://www.us-cert.gov/reading_room/DNS-recursion121605.pdf

  -- "DNS Amplification Attacks," see http://www.isotf.org/news/DNS-Amplification-Attacks.pdf

  -- "DNS Distributed Denial of Service (DDoS) Attacks," see http://www.icann.org/committees/security/dns-ddos-advisory-31mar06.pdf

Open Domain Name Servers Worldwide

- Unfortunately, despite this being a well known problem, at one point it was estimated that 75% of all name servers worldwide run as open recursive name servers (see http://dns.measurement-factory.com/surveys/sum1.html)
- Kristoff and Monnier estimated that 45% of .edu name servers were open recursive (see "Explorations in the .edu DNS Namespace," http://www.internet2.edu/presentations/ jt2007feb/20070213-kristoffmonnier.pdf at slide 5)
- And in a spirit of self-criticism, feel free to note that even UO's name servers were open until we secured them in February 2006. See, for example: http://cc.uoregon.edu/cnews/winter2006/recursive.htm
- **If *our* domain name servers were open recursive, *how about yours*? You NEED to get them secured if you haven't already done so!**

Many Other Schools Have Also Fixed Their Open Recursive DNS Servers…

- **Berkeley:** “Access to Caching DNS Servers to Be Restricted - Details,” http://net.berkeley.edu/DNS/recursion-detail.shtml

- **Merit Networks:** "Merit Network DNS Service Change," http://www.merit.edu/news/newsarchive/article.php?article=20060516_recursive

- **Northwestern University:** "NUIT Discontinues Recursive Queries on Central DNS Servers," http://www.it.northwestern.edu/transitions/2006/dns-queries.html

- **Penn State:** “Restrict Recursive Lookups on Central DNS Servers,” http://tns.its.psu.edu/networking/recursivedns.cfm

- **UAlbany:** “Non-Ualbany Recursive Access to Ualbany DNS Servers will End Monday, March 12,” http://www.albany.edu/its/news_DNS_access.htm

How Can I Find Open Recursive DNS Servers At My Campus?

- Team Cymru will happily send you notifications about open recursive DNS resolvers on your campus; to sign up to receive these notifications: http://www.team-cymru.org/Services/Resolvers/

- If you'd rather test things yourself, one tool which you can use to scan your network for open recursive DNS servers is dnsscan, see http://monkey.org/~provos/dnsscan/

- NOTE: Please do **NOT** scan for open recursive DNS servers on any network unless you are explicitly authorized by that network's owner/administrator to do so. Unauthorized scans will likely be considered hostile/illegal and may be treated as a computer intrusion and result in legal action against you.

What About Google's Public DNS Servers?

- Some in the audience may be aware that Google announced that it would be running publicly available recursive DNS servers that anyone could use by changing their name servers to point to 8.8.8.8 and/or 8.8.4.4 (see http://code.google.com/speed/public-dns/)

- Google is explicitly aware of the risks associated with the service that they're offering, and you can read the discussion of how they address/plan to address that issue at http://code.google.com/speed/public-dns/docs/security.html

- I should mention that Google is NOT the only site intentionally making recursive name servers available; other examples include:
  -- http://www.opendns.com/ (free and paid versions are available)
  -- https://www.dns-oarc.net/oarc/services/odvr (intended for those who want to try using a DNSSEC-enabled name server)

Coming Back to The General Problem of Open Recursive DNS Servers, The Problem Isn't "Just" About DDoS, Either

- If you aren't yet sufficiently motivated to "bite the bullet" and fix your DDoS-exploitable domain name servers by the discussion I’ve provided about DNS amplification, let me add a little more thrust to help launch that hog: if you're not controlling access to your domain name servers, you may also be leaving yourself vulnerable to **DNS cache poisoning attacks**, whereby vulnerable caching name servers can be made to return bogus results for a user's name service queries: www.secureworks.com/research/articles/dns-cache-poisoning

What's a Cache Poisoning Attack?

- In a nutshell, in cache poisoning attacks, the attacker "primes" the caching name server to respond to queries with an IP address of his/her choice, rather than the real/normal IP address for that site.

  An innocent victim then asks the caching name server for the IP address of a site of interest, such as the IP address of their bank's website.

  If the domain name of that site happens to be one that the attacker has poisoned, the victim is automatically and transparently misdirected to a website of the attacker's choice, rather than to their bank's real web site, and confidential data can then end up being lost.

Another Cache Poisoning Scenario

- Another cache poisoning scenario uses cache poisoning to redirect queries for popular sites (such as google.com or hotmail.com) to a site that contains a virus or other malware.

  If your caching name server has been poisoned, when you try to visit one of these popular sites, you can unknowingly be redirected to another site that stealthily tries to infect your PC with malware.

  Blocking open access to your recursive name servers won't completely eliminate the possibility of your servers participating in such attacks, but it will reduce the likelihood of that sort of abuse.

Recommendations to Deal With Open Recursive DNS Servers

- Insure that you're running a current version of BIND (or whatever DNS software you use)
- Insure that you've separated your Internet-facing authoritative name server from your customer-facing recursive name server
- Protect your customer-facing recursive name server from access by non-customers
- Consider implementing the additional DNS server hardening measures described in the Team Cymru BIND Template (see http://www.cymru.com/Documents/secure-bind-template.html)

5. Malware and DNS

It's time to start thinking about how malware interacts with DNS, and what will happen when DNS hijacking malware gets disrupted.

Spam-Related Malware Relies on DNS

- Much of the most virulent malware out there has been deployed to facilitate spamming, and 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 deliver its spam).
- Spam related malware may also refer to its upstream command and control hosts using their FQDNs, thereby making it possible for the miscreants to repoint their malware's command and control host from one dotted quad to another should the systems currently "hosting" their C&Cs get filtered or cleaned up.
- At the same time that malware critically **relies** on DNS, ironically other malware may **also** be actively working to block or 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"'

DNSChanger.F (3/27/2007)

- http://vil.mcafeesecurity.com/vil/content/v_141841.htm states that "the main objective of this trojan is to change the default DNS entries to its own [preferred] DNS server."

    #HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
    Services\Tcpip\Parameters\NameServer: "85.255.115.46
85.255.112.154" (This is just an example and IP can vary)
    #HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
    Services\Tcpip\Parameters\DhcpNameServer: "85.255.115.46
    85.255.112.154" (This is just an example and IP can vary)

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

DNS Tinkering Malware Is Driving an Architectural Change Among ISPs

- Confronted with malware that's targeting user DNS settings, providers are forced to think about scalable (“network-centric”) ways to deal with those threats.
- Coming up with a solution requires understanding the mechanics of how DNS is transported across the network.

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 actively "managing" user DNS traffic at the network level.

- What does "managing" mean?

  -- **blocking/dropping** **<u>all</u>** 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** some or all user DNS traffic that isn't destined for the ISP's customer DNS servers at Layer 4 (e.g., see: http://www.cisco.com/en/US/docs/routers/10000/10008/configuration/guides/ancp/isbl4rdt.pdf at PDF pages 12-13)

"Fixing" Some DNS-Related Things May Make Other DNS-Related Things Worse

- Some approaches to dealing with DNS insecurities (such as DNS-rewriting network middleboxes) may negatively impact Internet end-to-end transparency, and ironically, foreclose other approaches to securing DNS (such as DNSSEC). The IAB 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

We ARE Coming 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.**

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/Windows 7 may attempt to deter this, but workarounds exist, (e.g., see 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)

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

- http://www.codeproject.com/KB/IP/DnsHijack.aspx …

  "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 and Windows 2000 users should be used at this time (and even Windows Server 2003 R2 loses mainstream support 7/13/2010), 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 some sites known as attempting to do poisoning see: dns.measurement-factory.com/cgi-bin/poison_browser.pl

6. Hardening DNS

If you're running a DNS server, what steps can you take to help harden or protect it?

A True Factoid About BIND 9

- Appropos of nothing, a true factoid: the "security considerations" section of the BIND 9 manual runs just a tiny bit more than two pages. See: http://www.isc.org/files/Bv9.6ARM.pdf at page 91 As you now know, I'm a bit more verbose. :-)
- In fairness, BIND does offer detailed discussions of all the vulnerabilities they've detected and patched, see http://www.isc.org/advisories/bind , and many security related topics are handled in other sections of the documentation

Basic DNS Sanity Check

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

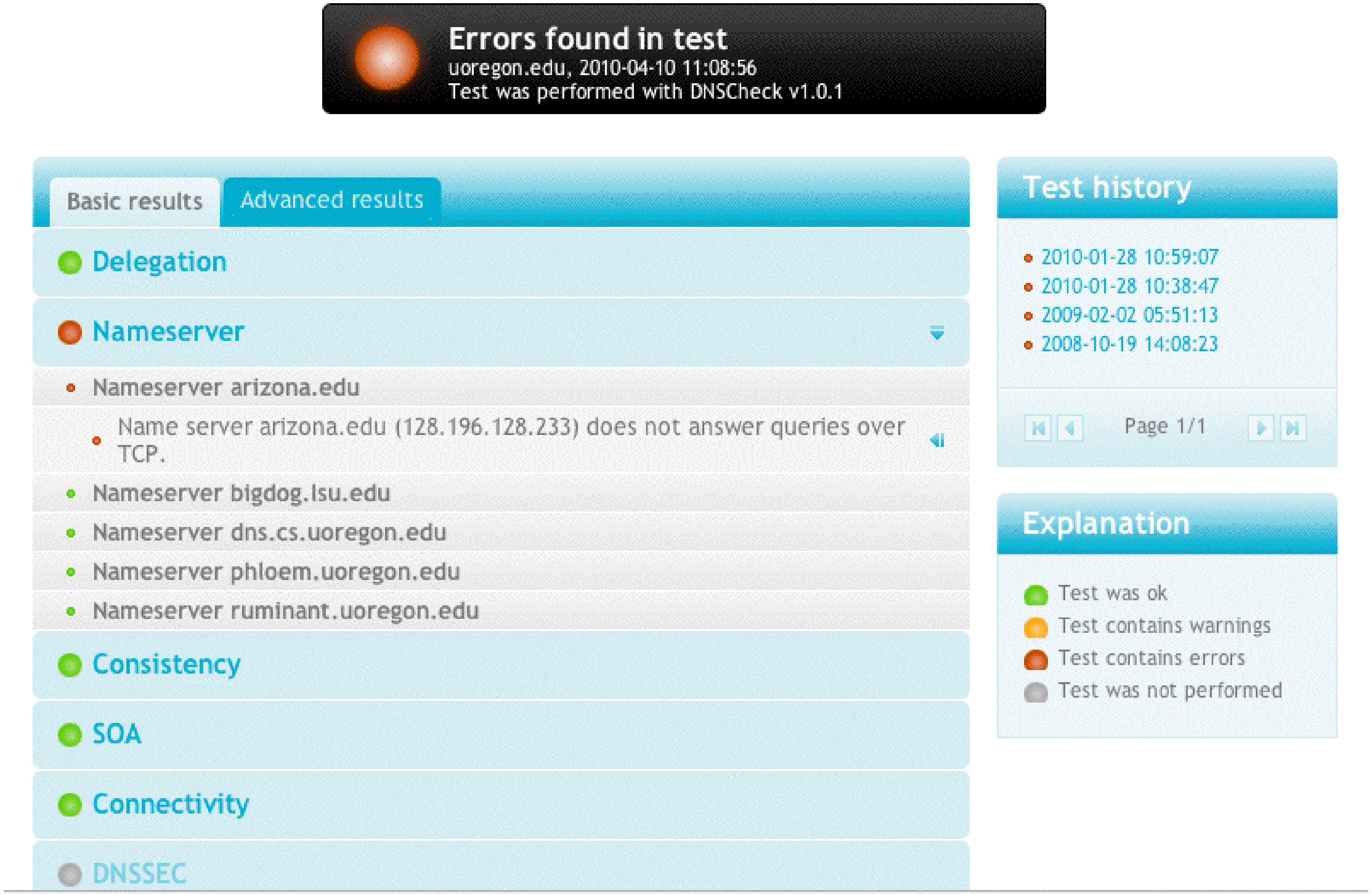
  http://dnscheck.iis.se/

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

  That free DNS check will do many 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.

- There are also other online DNS checking tools you can use -- try several and see which works best for you.

Example Output



Errors found in test
uoregon.edu, 2010-04-10 11:08:56
Test was performed with DNSCheck v1.0.1

🟢 Delegation

🔴 Nameserver

- Nameserver arizona.edu
  - Name server arizona.edu (128.196.128.233) does not answer queries over TCP.
- Nameserver bigdog.lsu.edu
- Nameserver dns.cs.uoregon.edu
- Nameserver phloem.uoregon.edu
- Nameserver ruminant.uoregon.edu

🟢 Consistency

🟢 SOA

🟢 Connectivity

⚪ DNSSEC

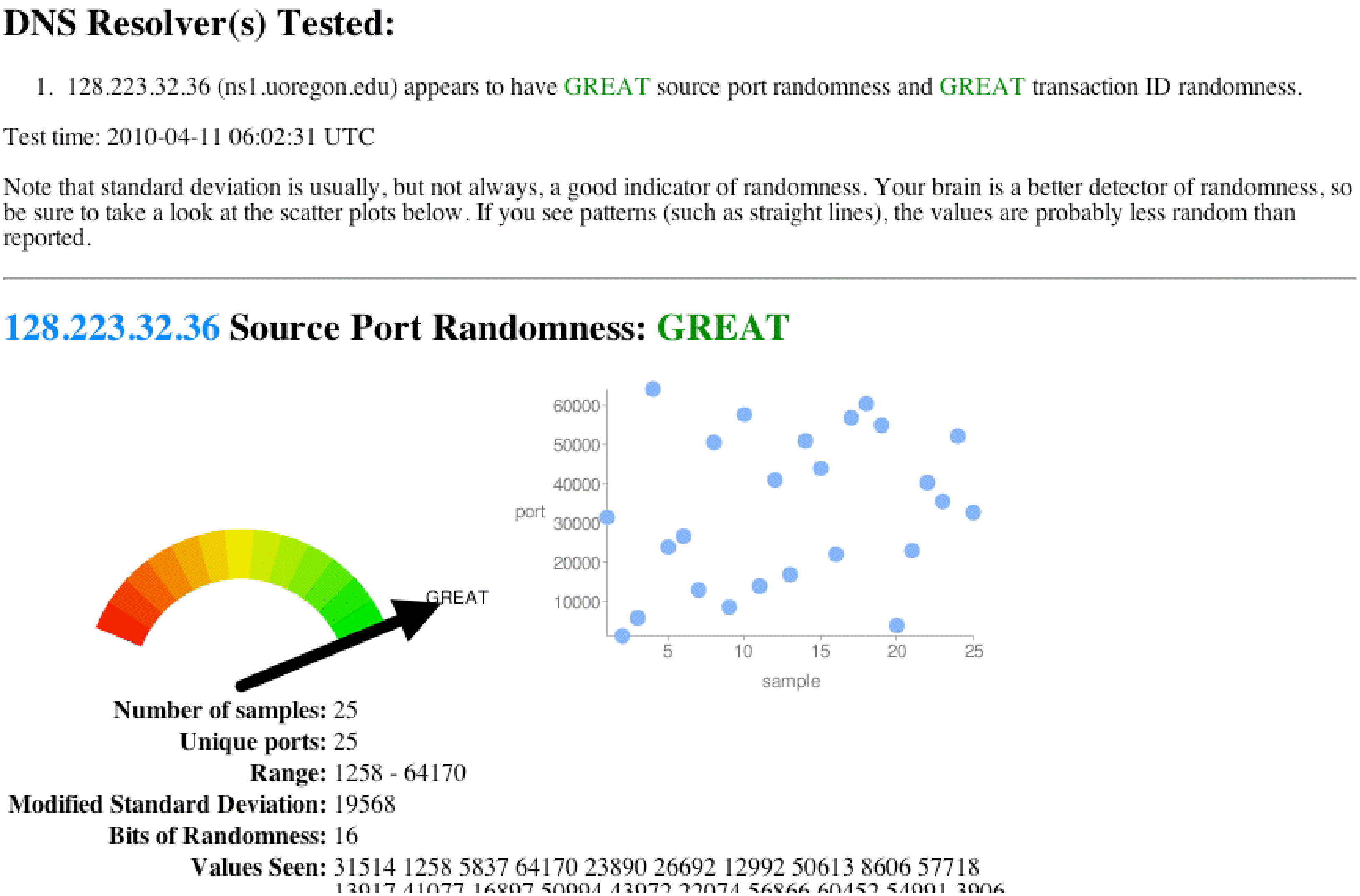
Test history

- 2010-01-28 10:59:07
- 2010-01-28 10:38:47
- 2009-02-02 05:51:13
- 2008-10-19 14:08:23

Explanation

- 🟢 Test was ok
- 🟠 Test contains warnings
- 🔴 Test contains errors
- ⚪ Test was not performed

One Other Test You Should Do Back Home…
https://www.dns-oarc.net/oarc/services/dnsentropy



DNS Resolver(s) Tested:

1. 128.223.32.36 (ns1.uoregon.edu) appears to have GREAT source port randomness and GREAT transaction ID randomness.

Test time: 2010-04-11 06:02:31 UTC

Note that standard deviation is usually, but not always, a good indicator of randomness. Your brain is a better detector of randomness, so be sure to take a look at the scatter plots below. If you see patterns (such as straight lines), the values are probably less random than reported.

128.223.32.36 Source Port Randomness: GREAT

Number of samples: 25
Unique ports: 25
Range: 1258 - 64170
Modified Standard Deviation: 19568
Bits of Randomness: 16
Values Seen: 31514 1258 5837 64170 23890 26692 12992 50613 8606 57718
13917 41077 16897 50994 43972 22074 56866 60452 54991 3906

DNS Server Software Versions

- Unless you have compelling reason to do otherwise, **run the latest version of the DNS server software you're using**.

- For BIND users, as of April 10th, 2010, this means 9.7.0-P1

  -- If you're on an earlier version, it is highly desirable that you upgrade to the current version

  -- Updated versions of BIND can be downloaded from http://www.isc.org/downloads

- **Note:** some vendors may not do a great job of keeping their vendor customized versions up to date. If you are using a vendor-supplied version of Bind, you need to carefully weigh the convenience of running an older vendor supported version of BIND against the strong desirability of running the latest version.

BTW It Isn't Just The Name Server Software

- If/when you upgrade BIND, you may notice that BIND isn't the **only** thing that may needs upgrading – how about the status of OpenSSL, for example? Problems with stale versions of OpenSSL are so common that BIND explicitly checks OpenSSL as part of the build process! Note that OpenSSL-1.0.0 was released March 29th, 2010, for example. Updated versions of OpenSSL are available from http://www.openssl.org/source/

- Are you periodically running a package management tool to check for ALL the software that may need updating?

  yum or apt-get can be your friend…

Determining the Version of BIND in Use

- % dig @ns1.uoregon.edu version.bind chaos txt

```
version.bind.        0      CH      TXT     "9999.9.9"
```

```
options {
    directory "/var/named";
    version "whatever";
};
```

- If you have shell access to the name server, try: % named –v (you may also want to use the unix find command to look for multiple/additional installations of named)
- If you don't have local access, you may also be able to fingerprint a name server using fpdns (see http://code.google.com/p/fpdns/), but it may not always be able to distinguish dot release versions.
- Of course, once they've identified your name server(s), the bad guys can also just try each and every exploit they know, regardless of whether or not they know the version of the code you're running!

OS Hardening

- It does little good to run a secure version of the name server software if the operating system that system is running is insecure. Making sure that you're running current versions of OS software and applications are part (but not all) of that picture.
- OS hardening is generally beyond the scope of this tutorial, however a few good starting points include:
  -- Center for Internet Security “Benchmarks” (checklists), see http://cisecurity.org/en-us/?route=downloads.benchmarks (some sites tailor their own recommendations from that, e.g., see http://security.utexas.edu/admin/redhat-linux.html)
  -- See als the National Security Agency’s Operating System Guides, http://www.nsa.gov/snac/
- In addition to hardening your name server OS, you may also want to consider running a tool (such as tripwire) which checksums critical executables, related libraries, and key configuration files.

The Art of Securely Configuring and Operating BIND

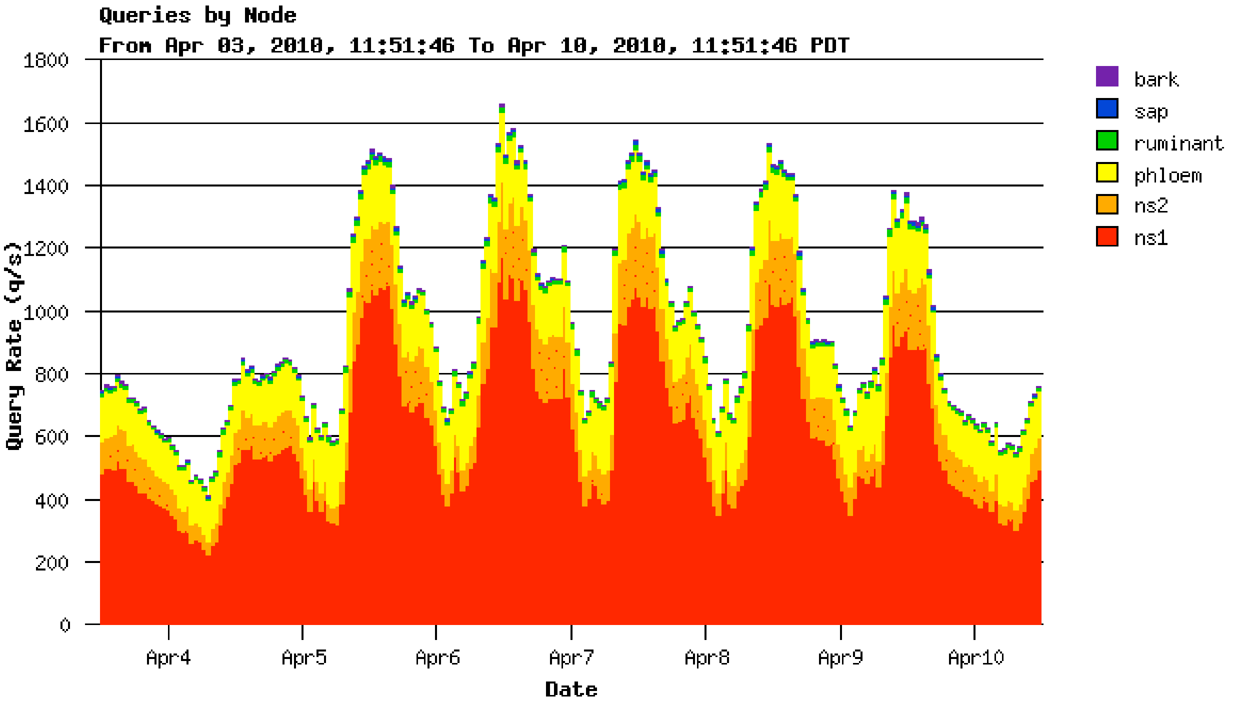
- Even if you're running a current version of BIND, it is still possible to configure it in more (or less secure) ways.
- A nice secure template to use for configuring BIND is the **Team Cymru Secure BIND Template**, available from http://www.cymru.com/Documents/secure-bind-template.html
- That configuration template will improve the security of BIND in a number of ways, including handling the open recursion problem, appropriately limiting zone transfers, and coaching you through running BIND in a chroot jail.
- Caution: do not "configure and forget" if you use the Team Cymru template since it includes some things (like lists of bogon IP space!) which *will* evolve over time.

Digression: Name Servers Other Than BIND

- I would also be remiss if I didn't mention that there are name servers other than BIND, both free/open source and commercial products, some of which I discuss in the DNSSEC part of this talk.
- A great topic for discussion over beers sometime is the question of which name server software is better, faster, more secure, has the best/most appropriate set of features, etc.
- For the most part, however, because of BIND's empirical dominance in the market place, that's what we'll (continue to) focus on.
- Noted for the record: there may be survivability value to running more than one name server software product (arguably, however, you're just complicating your support load and increasing your exposure to bugs in two, three or N products, rather than just picking one product and developing true expertise with it)

DNS Monitoring

- You should graphically monitor DNS query traffic just as you monitor things like transit bandwidth. A nice tool for this is DNS Stats Collector (DSC), see http://dns.measurement-factory.com/tools/dsc/ (sample below) You can see samples of some other possible graphs at http://dns.measurement-factory.com/tools/dsc/sample/index.html



Additional DNS Tools

- Beyond doing graphical DNS monitoring with DSC, there are additional DNS tools that you may find helpful listed at

  -- https://www.dns-oarc.net/oarc/tools
  -- http://dns.measurement-factory.com/tools/
  -- http://www.dns.net/dnsrd/tools.html

A Potential Op Sec Issue: Zone Transfers

- Zone transfers allow an entity to obtain a complete copy of a DNS zone. In some cases this may just be a small vanity domain, but in other cases it may be a complete ccTLD. For example:

```
% dig pk @ns.pknic.net.pk axfr
pk.             38400   IN      SOA     ns.pknic.net.pk. ashar.pknic.net.pk.
1137374758 14400 7200 1664000 21600
pk.             38400   IN      NS      ns.pknic.net.pk.
pk.             38400   IN      NS      m-2.pknic.net.pk.
pk.             38400   IN      NS      AUTH51.NS.UU.NET.
pk.             38400   IN      NS      AUTH101.NS.UU.NET.
ns1.0000.pk.    38400   IN      A       74.117.232.51
ns2.0000.pk.    38400   IN      A       74.117.232.51
01net.pk.       38400   IN      NS      ns1.mailclub.fr.
01net.pk.       38400   IN      NS      ns2.mailclub.fr.
0321.pk.        38400   IN      NS      nm.thebighosting.com.
0321.pk.        38400   IN      NS      cobalt.thebighosting.com.
0322.pk.        38400   IN      NS      nm.thebighosting.com.
0322.pk.        38400   IN      NS      cobalt.thebighosting.com.
[etc]
```

Why Are Zone Transfers An Issue?

- Zone transfers are a security issue because the first step in an attack is often reconnoitering the target, whether we're talking about a physical attack or an online attack.
- Having a copy of a target's zone file allows a miscreant to easily do a thorough and exhaustive review of the target's systems or domains, looking for vulnerabilities or exploitable weaknesses.
- For that reason, zone transfers should be strictly limited to just the sites that need to be able to transfer the zone files for legitimate purposes, such as those who provide secondary service for the zone.
- You may even want to consider blocking **all** conventional zone transfers, doing zone synchronization via rsync over ssh instead (see http://www.seebq.com/dns-replication-using-rsync/). Rsync over ssh has the additional advantage of eliminating the possibility of miscreants attempting zone file denial of service attacks via RFC1996 NOTIFY messasges, too.

Security-As-Availability: Avoid Single Points of Failure

- A key step to hardening your DNS service is to look at your architecture with an eye to any single points of failure:
  -- Do you have multiple physical DNS servers, or just one?
  -- Assuming you have multiple servers, are they on different subnets?
  -- Are at least some of your name servers at a different physical location, preferably in a different part of the country?
  -- If your site uses a border firewall, have you taken steps to make sure all your name servers are not behind a single common firewall?
  -- Are all of your servers running the same operating system and the same name server software?
  -- Don't forget your DNS admin, either – do you have at least two people who can handle DNS responsibilities at your site?

Network and System Capacity

- Because DNS servers may be the target of a denial of service attack, you may want to insure that those systems and the connectivity that services them are overprovisioned. While normal traffic loads may require trivial levels of connectivity, if your name server is the target of an attack, you'll find that fast ethernet is better than regular ethernet, and gigabit ethernet is better still. Similarly, a server class system with redundant power supplies an redundant power sources, running as multicore system with plenty of RAM, is also a good idea.
- Run your name servers on dedicated hardware. No other services should be delivered from the name servers – your name servers should be dedicated to just delivering name service!
- Try to run your customer facing recursive caching name servers and your Internet-facing authoritative servers on separate systems.

A Brief Digression: Name Server Architectures and Anycasting

- If you're like most network folks, you're probably familiar with unicast traffic, broadcast traffic, and maybe even IP multicast traffic, but anycast traffic is sort of an odd bird that may be less familiar. In a nutshell, anycasting involves advertising the *same* network prefix (typically a /24) from multiple locations. When someone attempts to query a name server which resides in an anycast range, they automagically use the closest server.

- A number of the root name servers are currently using Anycast to scale the number of servers available, and to improve performance among other reasons. See: http://www.root-servers.org/ and http://www.icann.org/meetings/vancouver/jlc-anycasting.pdf

Dynamic DNS (Commercial and RFC2135)

- "Dynamic DNS" can refer to two completely different things:

  -- commercial dynamic DNS service provided by a third party, designed to allow a user to map a vanity domain name or other hostname to a dynamic (rather than static) IP address

  -- RFC 2135 "Dynamic Updates in the Domain Name System" either as implemented by BIND or Microsoft

- Commercial dynamic DNS service should generally not be needed at most universities (if someone wants a static IP address, they should generally be able to request and receive one from the school); some universities/some commercial providers actually forbid use of 3rd party commercial dynamic DNS services

RFC2135 Dynamic Updates

- RFC2135 dynamic updates can cause issues with unnecessary traffic under some circumstances, particularly when they occur in conjunction with NAT'd users, see Section 2.8 of "Observed DNS Resolution Misbehavior" (RFC4697, October 2006). CAIDA also has an excellent page on disabling dynamic updates at: http://www.caida.org/research/dns/disable_dns_updates.xml or see http://support.microsoft.com/support/kb/articles/q246/8/04.asp
- While it is quite tempting to simply recommend avoiding dynamic DNS updates for philosophical reasons, dynamic updates can have a role in some special circumstances (IPv6, IP mobility, and Active Directory come to mind). If you decide that you do need dynamic updates (e.g., for ActiveDirectory-related reasons, I'd encourage you to review Yale's excellent web page on this at http://amtweb.its.yale.edu/yalead/ddns.asp
- Note that dynamic updates and DNSSEC are also incompatible.

AS112 Project

- Speaking of dynamic updates, do you all know about the AS112 Project, the "Nameservers at the end of the universe?"
- As noted at public.as112.net:
  "Because most answers generated by the Internet's root name server system are negative, and many of those negative answers are in response to PTR queries for RFC1918, dynamic DNS updates and other ambiguous addresses, as follows:
  -- 10.0.0.0/8
  -- 172.16.0.0/12
  -- 169.254.0.0/16
  -- 192.168.0.0/16

  There are now separate (non-root) servers for these queries…"
- Nice paper, "The Windows of Private DNS Updates," at http://www.caida.org/publications/papers/2006/ private_dns_updates/private_dns_updates.pdf

7. 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 <u>really</u> 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"
  -- RFC5155, "DNSSEC Hashed Authenticated Denial of Existence"

- 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**

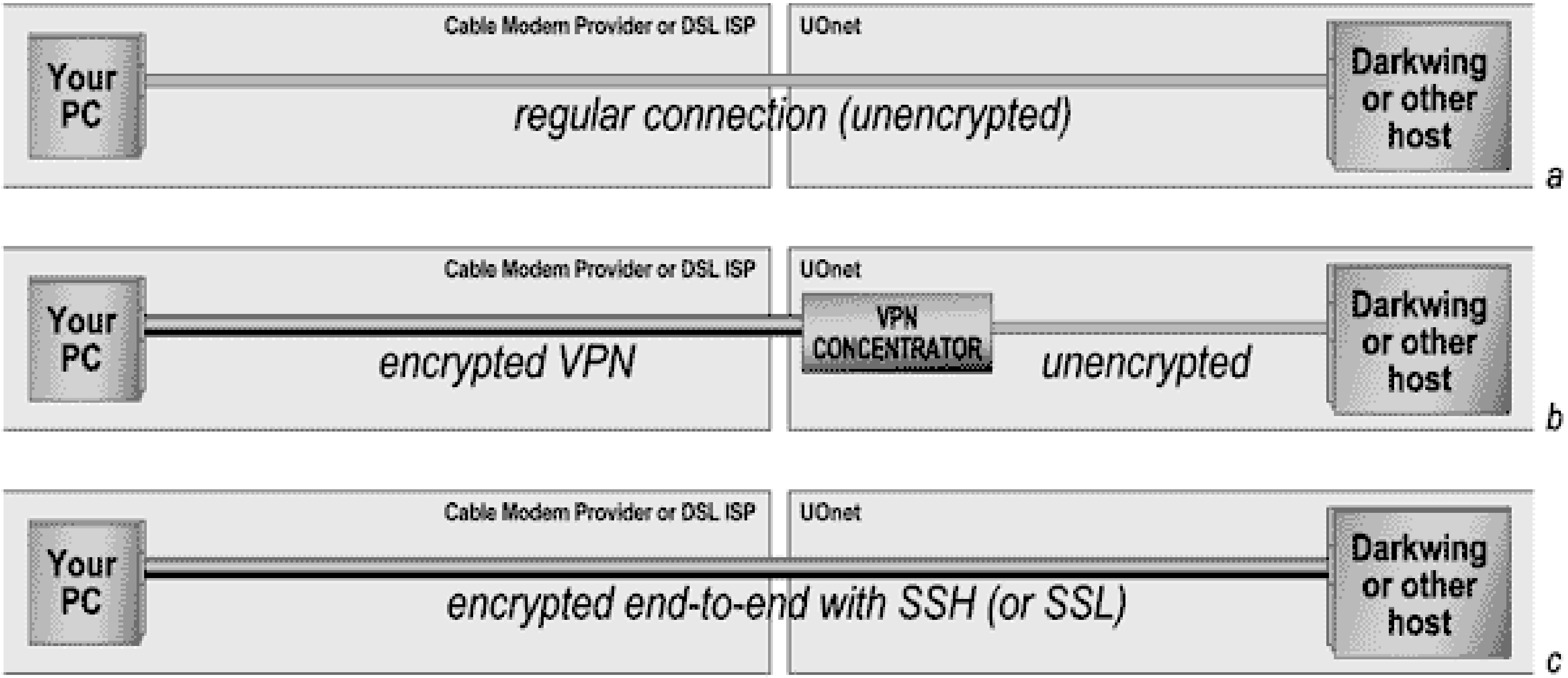
  DNSSEC is intentionally **NOT** designed to keep DNS data confidential, and it is also intentionally **NOT** designed to improve the availability of DNS data -- it's sole focus is on insuring the **integrity** of DNS data.

- 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


Cable Modem Provider or DSL ISP
UOnet
Your PC
Darkwing or other host
regular connection (unencrypted)
a
Cable Modem Provider or DSL ISP
UOnet
Your PC
VPN CONCENTRATOR
Darkwing or other host
encrypted VPN
unencrypted
b
Cable Modem Provider or DSL ISP
UOnet
Your PC
Darkwing or other host
encrypted end-to-end with SSH (or SSL)
c

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).

Windows DNS Client Support for DNSSEC

- Quoting http://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 DNSSEC User See If a DNS Resource Record Failed DNSSEC Validation?

- **Answer: <u>nothing</u>. Users would see <u>nothing</u>** 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:

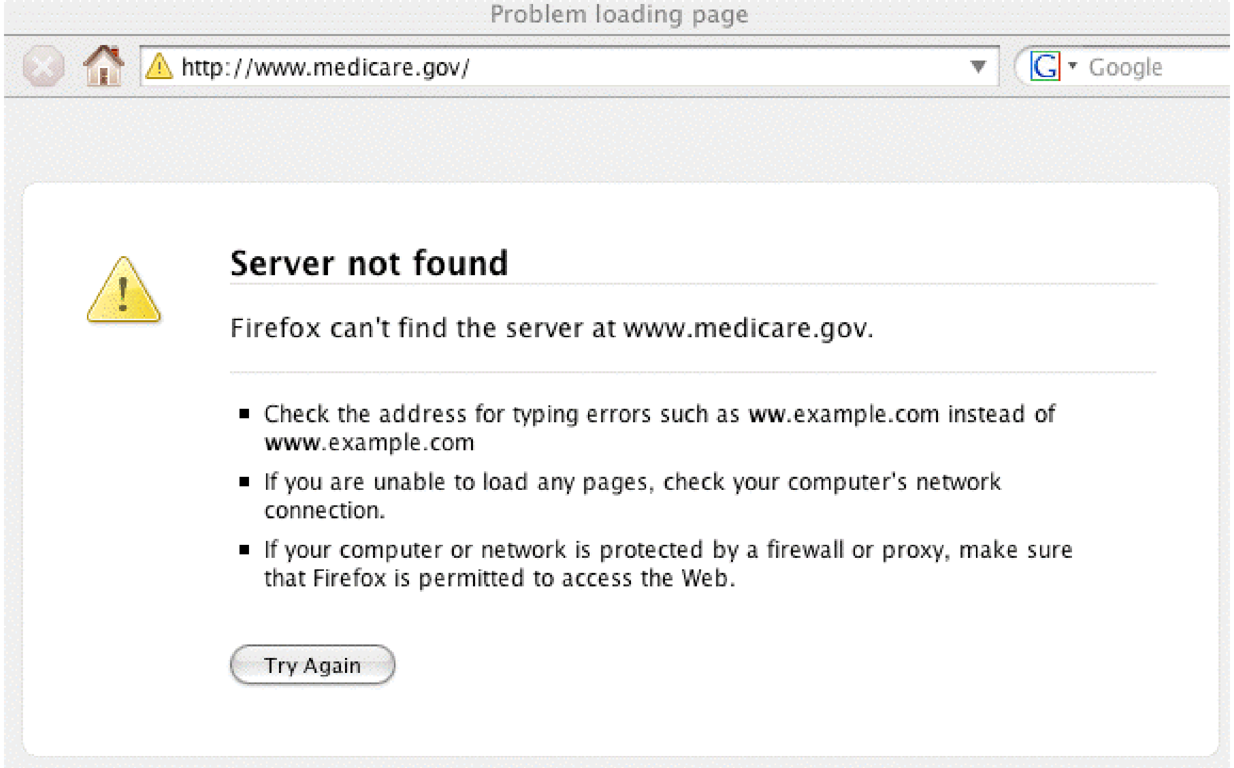
  -- operational error: it would be good to make sure that's noticed and corrected, and users could act as "canaries in the coal mine"

  -- 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.

What Would a DNSSEC User See If a DNS Admin Screws Up Signing DNSSEC Signing Their Zone?

- The zone wouldn’t resolve. Thus web pages under that zone would be inaccessible to user doing DNSSEC, although users who AREN’T doing DNSSEC would still be just fine.
- Example: try to access www.medicare.gov on 4/10/2010 from UO:



Server not found

Firefox can't find the server at www.medicare.gov.

- Check the address for typing errors such as ww.example.com instead of www.example.com
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

www.medicare.gov on 4/10/2010

- **% dig www.medicare.gov @ns1.uoregon.edu [does DNSSEC]**

  ```
  [snip]
  ;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 65323
  ;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0,
  ADDITIONAL: 0
  [snip]
  ```

- **% dig www.medicare.gov @149.20.64.20 [does DNSSEC]**

  [same result as for ns1.uoregon.edu]

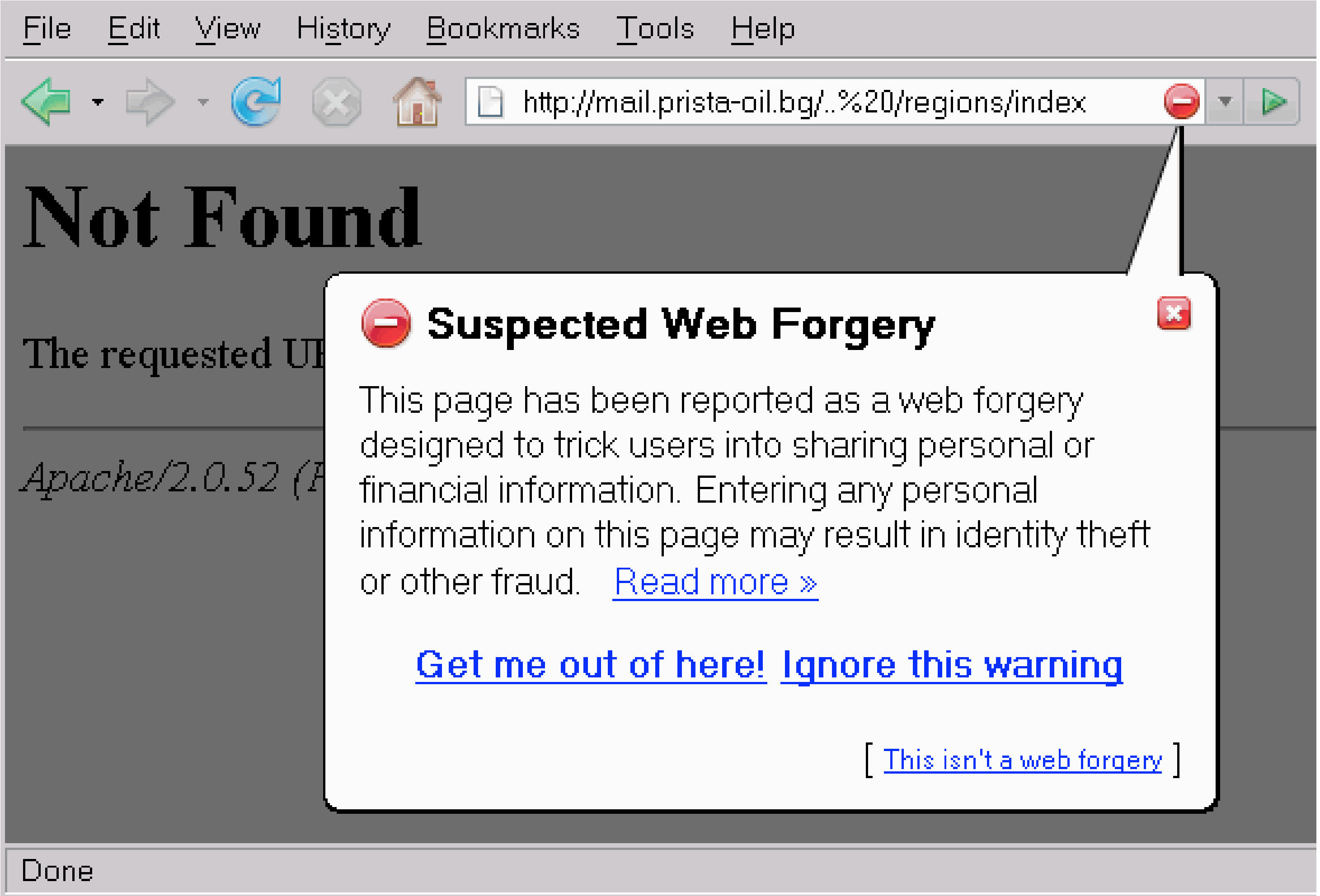
- **% dig www.medicare.gov @8.8.8.8 [does NOT do DNSSEC]**

  ```
  ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48657
  ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
  ADDITIONAL: 0
  [snip]
  www.medicare.gov.     541    IN     A      146.123.140.204
  ```

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, or someone screws up and it breaks attempts to do legitimate things), will people even know they're using DNSSEC?
- Contrast invisible DNSSEC protection with the anti-phishing protection that Firefox delivers, something that's FAR more "in your face" and visible…

What A <u>Firefox</u> User Sees When Attempting to Visit A Phishing Site



Not Found

The requested UI

Apache/2.0.52 (F

Suspected Web Forgery

This page has been reported as a web forgery designed to trick users into sharing personal or financial information. Entering any personal information on this page may result in identity theft or other fraud. Read more »

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 fully signed.**

Signing The Root (".")

- All the DNS root servers are supposed to be signed by July, 2010. As part of a phased deployment, currently seven root servers are… A (Verisign), D (U Maryland), E (NASA Ames), I (Autonomica/ Nordunet), K (RIPE), L (ICANN) and M (WIDE). For example:

```
% dig . +dnssec @k.root-servers.net
[snip]
;; AUTHORITY SECTION:
.        86400    IN         SOA      a.root-servers.net. nstld.verisign-grs.com. 2010041001
1800 900 604800 86400
.        86400    IN         RRSIG    SOA 8 0 86400 20100417000000 20100409230000
55138 . wflJAzJQaVrNedJSJrtE8yYsBUsygr1V8iqsJdhvOkiq99ZswiMED5dN
rVI3N56pnPfwCVbejdK1c3JFfJjHYU9fAGMR0mbvL3fVq/MMoIMIgipB
fm4dsZ48ULCt6Jg5lcWMQSMlwsb8S/PViBaHwXGdyGkbCz1RGh8ZxAGa gvQ=
.        86400   IN     NSEC    ac. NS SOA RRSIG NSEC DNSKEY
.        86400   IN     RRSIG   NSEC 8 0 86400 20100417000000 20100409230000 55138
. E/1kHbUWunXPv5KK/Jb6iIRZfe172m5OsFBtTCHylzTnYFMC5NEigjjA
LBu1NjTrTctu7MfCyh7cPfjJrft+72G3zWPE102ihz9D2PvlN2NUrtMP
Yn0meWMi+FphIFy5rjR1ihS6aNgkieE5Q9RuKoCVRHGURY4cKnDS2Ej4 4bw==
```

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

- A limited number are, see https://itar.iana.org/anchors/anchors.mf (if you download that file, save it as a text file despite the weird file extension)
- Signed TLD domains include .arpa (the in-addrs), .bg (Bulgaria), .br (Brasil), .cz (Czech Republic), .gov, .li (Liechtenstein), .na (Namibia), .nu (Niue), .org, .pr (Puerto Rico), .se (Sweden), .th (Thailand), .tm (Turkmenistan), .uk (the United Kingdom) and .us
- There are also trust anchors for a number of IDN'd TLDs.
- Most other TLDs (including .edu, .com, .net, .info, .mil, .biz, .int, .ca, .cn, .de, .fr, .jp, etc.) are still NOT signed 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, or the TLD above them isn’t signed, all the stuff below that point 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).
- If 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, then that zone will “do a medicare.gov”

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/solutions/dlv
- DLV is supported in current versions of BIND
- DLV is the most popular approach to dealing with the problem of maintaining trust anchors until the root and TLDs are signed.
- If you don’t want to rely on DLV, and you’re willing to use ONLY TLD-level trust anchors, you can also use the IANA interim trust anchors (https://itar.iana.org/anchors/anchors.mf)

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 as defined by RFC5155, addresses the zone enumeration issue through use of salted hashes, which handles both the zone enumeration 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/200910.html

Using dataset II, authoritative 2nd level com/net/org servers:

Recent **BIND 9**	173,590	69.23%	
Other versions of **BIND**	11,583	4.62%	**(73.85% total)**

Using dataset I, nameservers found on random IPv4 addresses:

Recent **BIND 9**	235,358	31.44%	
Other versions of **BIND**	16,828	2.26%	**(33.70% total)**

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? (Yes, unfortunately, people are!)

- 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]

DNSSEC and Windows Server 2008 R2

- The situation is less dire for Windows Server 2008 R2. Windows Server 2008 R2 now provides at least basic DNSSEC support, and an 87 page guide to Windows DNSSEC deployment guide released in October 2009 is now available from Microsoft (see http://tinyurl.com/windows-and-dnssec).

- The current Microsoft DNSSEC implementation has not exactly won universal acclimation, unfortunately. See, for example, "DNSSEC: Will Microsoft Have Enough Time?", Jan 29, 2010, www.circleid.com/posts/dnssec_will_microsoft_have_enough_time/

- See also "NIST SP 800-81r1 Checklist Items and Microsoft Windows Server 2008 R2," http://www.dnsops.gov/vendors/ MS-Win2008R2-SP800-81r1-Checklist.pdf (note, for example, that NSEC3 support is still lacking)

What About DJBDNS aka TinyDNS?

- If you're considering doing DNSSEC and you're currently using DJBDNS or TinyDNS, you should note that the author of those products explicitly does NOT support DNSSEC in DJBDNS/TinyDNS, and to the best of my knowledge has no plans to change that stance. You can see his discussion and rationale at:

  http://cr.yp.to/djbdns/blurb/security.html and at
  http://cr.yp.to/djbdns/forgery.html

EDNS0

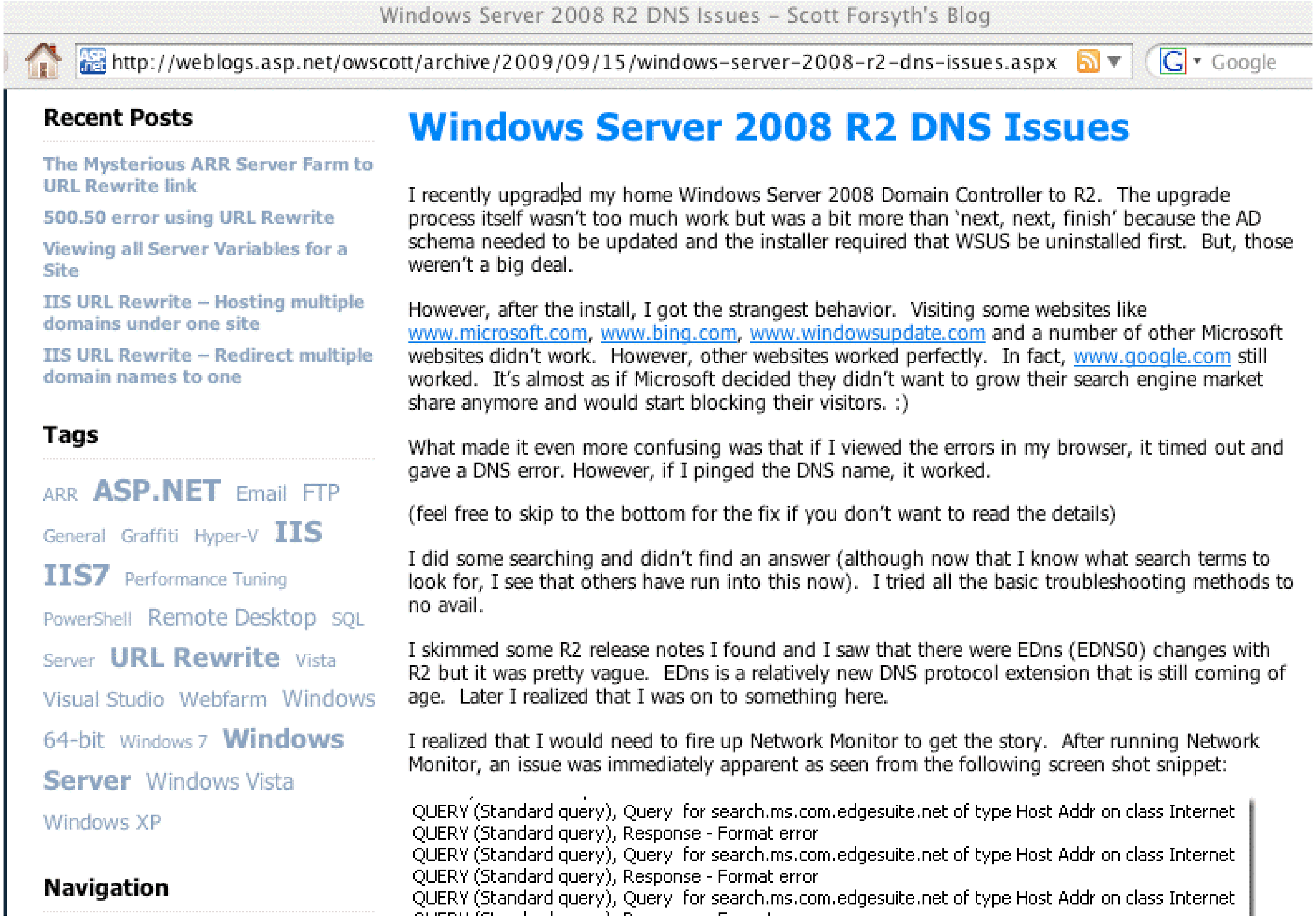
- You should know that name servers doing DNSSEC requires a feature known as EDNS0, as defined in RFC2671, "Extension Mechanisms for DNS (EDNS0)," August 1999.
- Normally, DNS UDP responses are limited to just 512 bytes, a size that's too small for the much larger DNSSEC records. To better handle delivery of DNSSEC records, EDNS0 allows clients and servers to negotiate the maximum size datagram which they can handle, with the expectation that at least some hosts might negotiate datagram sizes as high as 4KB. Name servers doing DNSSEC **must** also do EDNS0.
- Why's that a problem? Well… some firewalls may be configured to block UDP DNS traffic > 512 bytes. If you've got a firewall in front of your DNS server, please test to see if you’re broken: https://www.dns-oarc.net/oarc/services/replysizetest

Sample Results from an EDNS0 Test

-
  ```
  % dig  +short rs.dns-oarc.net  txt  @ns1.uoregon.edu
  rst.x996.rs.dns-oarc.net.
  rst.x1956.x996.rs.dns-oarc.net.
  rst.x2442.x1956.x996.rs.dns-oarc.net.
  "128.223.32.36 sent EDNS buffer size 4096”
  "128.223.32.36 DNS reply size limit is at least 2442”
  "Tested at 2010-04-10 23:21:41 UTC”
  ```

-
  ```
  % dig  +short  rs.dns-oarc.net  txt  @8.8.8.8
  rst.x476.rs.dns-oarc.net.
  rst.x485.x476.rs.dns-oarc.net.
  rst.x490.x485.x476.rs.dns-oarc.net.
  "74.125.154.94 DNS reply size limit is at least 490”
  "74.125.154.94 lacks EDNS, defaults to 512”
  "Tested at 2010-04-10 23:25:21 UTC"
  ```

EDNS0 In Some MS Windows Environments


Windows Server 2008 R2 DNS Issues - Scott Forsyth's Blog
http://weblogs.asp.net/owscott/archive/2009/09/15/windows-server-2008-r2-dns-issues.aspx
Google

Recent Posts

- The Mysterious ARR Server Farm to URL Rewrite link
- 500.50 error using URL Rewrite
- Viewing all Server Variables for a Site
- IIS URL Rewrite – Hosting multiple domains under one site
- IIS URL Rewrite – Redirect multiple domain names to one

Tags

ARR ASP.NET Email FTP General Graffiti Hyper-V IIS IIS7 Performance Tuning PowerShell Remote Desktop SQL Server URL Rewrite Vista Visual Studio Webfarm Windows 64-bit Windows 7 Windows Server Windows Vista Windows XP

Navigation

Windows Server 2008 R2 DNS Issues

I recently upgraded my home Windows Server 2008 Domain Controller to R2. The upgrade process itself wasn't too much work but was a bit more than 'next, next, finish' because the AD schema needed to be updated and the installer required that WSUS be uninstalled first. But, those weren't a big deal.

However, after the install, I got the strangest behavior. Visiting some websites like www.microsoft.com, www.bing.com, www.windowsupdate.com and a number of other Microsoft websites didn't work. However, other websites worked perfectly. In fact, www.google.com still worked. It's almost as if Microsoft decided they didn't want to grow their search engine market share anymore and would start blocking their visitors. :)

What made it even more confusing was that if I viewed the errors in my browser, it timed out and gave a DNS error. However, if I pinged the DNS name, it worked.

(feel free to skip to the bottom for the fix if you don't want to read the details)

I did some searching and didn't find an answer (although now that I know what search terms to look for, I see that others have run into this now). I tried all the basic troubleshooting methods to no avail.

I skimmed some R2 release notes I found and I saw that there were EDns (EDNS0) changes with R2 but it was pretty vague. EDns is a relatively new DNS protocol extension that is still coming of age. Later I realized that I was on to something here.

I realized that I would need to fire up Network Monitor to get the story. After running Network Monitor, an issue was immediately apparent as seen from the following screen shot snippet:

```
QUERY (Standard query), Query  for search.ms.com.edgesuite.net of type Host Addr on class Internet
QUERY (Standard query), Response - Format error
QUERY (Standard query), Query  for search.ms.com.edgesuite.net of type Host Addr on class Internet
QUERY (Standard query), Response - Format error
QUERY (Standard query), Query  for search.ms.com.edgesuite.net of type Host Addr on class Internet
```

One View of DNSSEC Penetration: UCLA's SecSpider Project

- SecSpider: The DNSSEC Monitoring Project http://secspider.cs.ucla.edu/ reports (as of Saturday, April 10th, 2010) that there are now 12,961 production zones doing DNSSEC (although some of those zones may be zones from non-US ccTLDs or zones from obscure signed 2nd level domains)
- Signed 2nd level edu zones include:
  -- berkeley.edu
  -- internet2.edu (and ucaid.edu)
  -- merit.edu
  -- penn.edu (and upenn.edu)
  -- psc.edu
- There are also a fair number of signed 3rd level edu zones such as netsec.colostate.edu, lcs.mit.edu, cs.rpi.edu, engr.wisc.edu, etc.

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 SecProf2010 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/)

- Are people waiting to see what the other folks do w.r.t. DNSSEC?

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 <u>is</u> a possibility.

8. DNS Case Studies

Applying What We've Talked About

- Given that we're a small group, and this is a meeting of practitioners, what do we see if we actually look at DNS and related areas at some sites?
- We could pick arbitrary sites, but since we're a small group, let's look at the sites of the folks who've actually signed up for the seminar. (My apologies to you if you're a last minute walk in participant – we'll try to "do" your site in at the end on the fly if we have the opportunity)
- Please note that if we find issues with your site (and I think I could find issues with any site I'll ever look at if I look hard enough!), please do not take that as a criticism – that's not how it is intended. When we flag things that seem odd, our goal is solely to help you (and others) harden their sites. Sometimes sites only show issues for a brief period, and it is just luck that I happened to check during at just the wrong time… the important thing is that issues get fixed!

7.1 Penn State University

Penn State Routing

- AS3999 (Pennsylvania State University)
- Upstream ASNs:
  AS5050 (Pittsburgh Supercomputing Center) and
  AS174 (Cogent Communications)
- whois.radb.net knows about AS3999
- IPv4 Network blocks advertised via AS3999:
  66.71.0.0/17
  75.102.64.0/18
  128.118.0.0/16
  130.203.0.0/16
  146.186.0.0/16
  150.231.0.0/16
- IPv6 Netblocks advertised via AS3999:
  2610:8::/32

Penn State Name Servers

- Potential name server-related issues:

  -- Name servers are open for psu.edu zone transfers (e.g.,f0fs03.cac.psu.edu, isengard.cse.psu.edu, ns1.ems.psu.edu (IPv4 and IPv6), otc2.psu.edu (IPv4 and IPv6), psu-ns.acns.msu.edu, sodium.tns.its.psu.edu (IPv4 and IPv6)

  -- Some name servers may be running older versions of BIND (9.4.3-P3 and 9.6.1-P3)

  -- While IPv4 psu.edu name servers are advertised by multiple ASNs, IPv6 name servers are only advertised by a single ASN

  -- psu.edu is not DNSSEC signed

Penn State Miscellaneous

- abuse.net knows about: security@psu.edu
- No SPF record defined
- Reasonable looking sending pattern on senderbase.org
- Domain whois updated 09-Jul-2007
  ASN whois all updated 24-Jun-2005
- wpad.psu.edu and wpad.la.psu.edu are NOT defined
  isatap.psu.edu and isatap.la.psu.edu are NOT defined
- Some indication that blog/guestbook/wiki spam is occurring (google for “cheap phentermine” site:psu.edu)

7.2 North Carolina State University

NCSU Routing

- AS11442 (NCSU)
- Upstream: AS81 (NCREN)
- No whois.radb.net entry for 11442, but whois.radb.net has interesting entries for AS81:

```
[whois.radb.net]
aut-num:     AS81          <== this is really MCNC/NCREN.NET per ARIN
as-name:     RoadRunner
descr:       RR-RC-Rockingham County Schools-Greensboro
import:      from AS-ANY   accept ANY
export:      to AS-ANY   announce AS-ROADRUNNER
admin-c:     IPADDREG
tech-c:      IPADDREG
notify:      ipaddreg@rr.com
mnt-by:      MAINT-RR
changed:     ipaddreg@rr.com 20080805
source:      RADB
```

[RADB also has an entry for AS81 ASN-NCREN from SAVVIS ~1995]

NCSU Address Space

- Network block advertised via AS11442:
  152.1.0.0/16
  152.7.0.0/16
  152.14.0.0/16
  204.84.244.0/22 (part of NCREN’s 204.84.0.0/15)

- NCSU.EDU domain whois last updated 07/2007
  AS11442 whois last updated 09/2008
  Some netblock whois entries last update 09/1998

- NCREN-B14 and NCSU3 cover exactly overlapping ranges:
  [whois.arin.net]
  North Carolina Research and Education Network NCREN-B14
  (NET-152-14-0-0-1)
  152.14.0.0 - 152.14.255.255
  North Carolina State University NCSU3 (NET-152-14-0-0-2)
  152.14.0.0 - 152.14.255.255

NCSU Name Servers

- Superfluous name server listed at parent: uni00ns.unity.ncsu.edu and uni10ns.unity.ncsu.edu
- Additional name server listed at child: ns1.ncsu.edu, ns2.ncsu.edu
- No IPv6 name servers found.
- No DNSSEC

NCSU Miscellaneous

- Abuse.net knows about: abuse@ncsu.edu
- No SPF record
- NCSU.EDU has some blocklisted hosts on Senderbase.org
- wpad.ncsu.edu is NOT defined
- Isatap.ncsu.edu is NOT defined
- Domain is showing material signs of guestbook/blog/wiki spam (google for site:ncsu.edu “cheap phentermine” and ask to see all results)

7.3 Indiana University

Indiana University Routing

- AS87
  upstreams:
  -- AS19872 (Indiana Gigapop)
  -- AS11069 (Egix, Inc.)
- Whois.radb.net knows about AS87
- Originates
  129.79.0.0/16
  134.68.0.0/16
  140.182.0.0/16
  149.159.0.0/16
  149.160.0.0/14
  149.165.0.0/17
  149.166.0.0/16
  156.56.0.0/16
  198.49.177.0/24

Indiana University Name Servers

- Actual/potential name server related issues:
  -- open recursive name servers on IPv6
  (2001:18e8:3:220:0:0:0:6, 2001:18e8:2:8:0:0:0:6)
  -- SOA MNAME for indiana.edu (ns.indiana.edu) not listed as NS.
  -- No answer received from 129.79.1.1 when querying for
  indiana.edu/IN/SOA.
  -- IPv6 name servers announced from only one ASN

Indiana University Miscellaneous

- Abuse.net knows about abuse@indiana.edu
- No SPF record defined (weird TXT record for indiana.edu: "ReleaseWLIDNamespace=true”)
- Are all three MX records for indiana.edu on the same subnet?

  ```
  external-relay.indiana.edu. 19075 IN    A       129.79.1.61
  belushi.uits.indiana.edu. 5030  IN      A       129.79.1.188
  hartman.uits.indiana.edu. 4820  IN      A       129.79.1.194
  ```
- wpad.indiana.edu is NOT defined
  isatap.indiana.edu is NOT defined
- Checking for “cheap phentermine” site:indiana.edu doesn’t return much indication of spamming

7.4 City University of New York

City University Routing

- AS31822
- Upstreams: AS3754 (Nysernet), AS209 (Qwest), AS3356 (Level3)
- AS31822 is NOT registered in whois.radb.net

City University Address Blocks

- 128.228.0.0/16
  134.74.0.0/16
  146.95.0.0/16
  146.96.0.0/16
  146.111.0.0/16
  146.245.0.0/16
  148.84.0.0/16
  149.4.0.0/16
  150.210.0.0/16
  163.238.0.0/16
  198.61.16.0/20
  198.83.28.0/22
  198.83.112.0/20
  198.180.141.0/24
  199.219.128.0/18
  199.219.192.0/20
  199.219.208.0/21
  199.219.216.0/24
  207.159.192.0/18
  209.2.54.0/23

City University Whois

- Cuny.edu domain whois last updated Dec 2008
- ASN whois information last updated Jan 2004
- Some IP whois was last updated some time ago (e.g., 146.111.0.0 was last updated September 1998)

City University Name Servers

- No IPv6 name servers
- No DNSSEC
- Other than that, pretty sweet! :-)

City University Miscellaneous

- Abuse.net knows about security@mail.cuny.edu
- Weird cuny.edu txt record (like indiana.edu!), "ReleaseWLIDNamespace=true"
- Only one MX record
- wpad.cuny.edu is NOT defined
  isatap.cuny.edu is NOT defined
- cuny.edu appears to be getting abused by guestbook/blog/wiki spammers (as an example, google for “cheap phentermine” site:cuny.edu although some of those pages may already be down)

7.5 Morehouse School of Medicine

MSM Routing and Netblocks

- AS29972 (Morehouse Medical)
- Upstreams:

  AS10490 (SOX)

  AS14745 (Internap)
  AS3549 (GBLX)

  (Routes heavily prepended in favor of AS10490 and against AS14745 and AS3549; AS3549 only used for 204.246.192.0/21)
- Netblocks:

  70.42.183.0/24
  192.83.232.0/24
  204.246.192.0/21
- Site also uses 174.46.102.16/28 for www.msm.edu, routed by AS4323 (TWTelecom)

MSM Nameservers

- Name server issues/notes:

  -- saturn.msm.edu has old version of BIND (8.4.7-REL-NOESW)
  -- saturn.msm.edu open for zone transfers of msm.edu
  -- name servers appear to be on consecutive Ips (192.83.232.40, 192.83.232.41)
  -- No offsite name server (for survivability)
  -- No IPV6 name servers
  -- No DNSSEC
  -- superfluous name server listed at parent: ns2.twtelecom.net

MSM Miscellaneous

- abuse.net has no entry for msm.edu (except the default of postmaster@msm.edu)
- Msm.edu has an SPF record
- Msm.edu’s MX’s route via Postini, except for InFilter2.msm.edu (test anti-spam product?)
- wpad.msm.edu doesn't exist
  isatap.msn.edu doesn’t exist
- Msm.edu does NOT appear to have any entries touting cheap phentermine web pages, good job keeping the blog/guestbook/wiki spammers at bay!

7.6 Rochester Institute of Technology

RIT Routing and Netblocks

- AS4385 (last updated 1/2002)
- whois.radb.net doesn't know about AS4385
- Upstreams AS3754 (Nysernet), AS3356 (Level3), and AS4323 (Time Warner Telecom)
- 129.21.0.0/16 (IP whois last updated 10/2002)
  192.77.9.0/24 (IP whois last updated 10/2002)

RIT Name Servers

- Actual/potential name server issues/notes:

  -- accuvax.northwestern.edu open for zone transfer of rit.edu
  -- different serial numbers found:
  SOA at address 129.21.3.17 has serial 23648168
  SOA at address 129.21.4.18 has serial 23648168
  SOA at address 129.105.49.100 has serial 23647780
  -- refresh and retry are somewhat short (3600 and 600 respectively vs. 14400 and 3600)
  -- No IPv6
  -- No DNSSEC

RIT Miscellaneous

- Abuse.net knows about
  abuse@rit.edu
- No SPF record for rit.edu
- MX records appear to be on successive dotted quads:

  ```
  mxgate02.rit.edu.    590    IN    A    129.21.3.39
  mxgate03.rit.edu.    593    IN    A    129.21.3.40
  mxgate01.rit.edu.    586    IN    A    129.21.3.38
  ```

- wpad.rit.edu is NOT defined
  isatap.rit.edu is NOT defined
- Some guestbooks/blogs/wikis appear to be getting abused;
  google for “cheap phentermine” site:rit.edu to see examples

7.7 Ithaca College

Ithaca College Routing

- 147.129.0.0/16 is routed by AS4323 (Time Warner Telecom); consider getting own ASN to enable eventual HPC connectivity?
- Domain and IP whois both updated in 2010 (excellent!)

Ithaca College Name Servers

- Actual/potential issues:
  -- Superfluous name server listed at parent: resolver3.ithaca.edu
  -- Additional name server listed at child: siren.ithaca.edu
  -- ns1.ithaca.edu --> 209.51.64.22 --> dynamic.apogeenet.net --> NXDOMAIN
  -- ns2.ithaca.edu --> 66.152.113.102 --> NXDOMAIN
  -- siren.ithaca.edu --> 147.129.56.13 --> NXDOMAIN
  -- siren.ithaca.edu doesn’t answer over TCP or UDP (master name server mistakenly advertised publicly?)
  -- MX servers on successive Ips (147.129.30.79 and 147.129.30.80)
  -- SOA refresh and retry TTLs too low (1200 < 14400, and 2400 < 3600m respectively)
  -- No IPv6 name servers
  -- No DNSSEC

Ithaca College Miscellaneous

- Abuse.net knows about abuse@ithaca.edu
- No SPF record
- Senderbase looks fine for the ithaca.edu domain
- wpad.ithaca.edu is NOT defined
  isatap.ithaca.edu is NOT defined
- No indication that guestbooks/blogs/wikis are currently being spammed

7.8 University of San Francisco (usfca.edu)

USFCA Routing

- AS22700 (University of San Francisco)
- Upstream AS2152 (Calren)
- Netblock: 138.202.0.0/16
- Also uses 208.88.129.81 (part of Websolutions Technology's 208.88.128.0/22) for web hosting

USFCA Name Servers

- Real/potential name server issues/notes:

  -- Uses ns2.cenic.org, which appears to be recursive
  -- hostmaster@lovelace.usfca.edu not possible?
  -- Refresh TTL is 7200 (recommended is 14400)
  -- No IPv6 nameservers
  -- No DNSSEC

USFCA Miscellaneous

- Abuse.net has abuse@usfca.edu
- Does not have an SPF record
- Usfca.edu looks fine at senderbase
- wpad.usfca.edu is NOT defined
  isatap.usfca.edu is NOT defined
- A handful of usfca.edu pages are being hit by blog/guestbook/ wiki spam (google for "cheap phentermine" site:usfca.edu)

7.9 Union College

Union College Routing

- AS19999 (Union College), cool ASN :-)
- Upstreams are AS4323 (Time Warner) and AS11351 (RoadRunner), with prepending used to pref 11351 and deprioritze 4323
- whois.radb.net thinks AS19999 is RoadRunner, even though ARIN knows better :-)
- Union has 192.52.218.0/24 (but note Sprint Rtech handle for that netblock) plus 149.106.0.0/16. The Union /16 is being broken up into a bunch of separately announced individual /19's in addition to being announced as the /16.

Union College Name Servers

- Real/potential name server issues/notes:
  -- dutch.union.edu and eliphalet.union.edu appear to be open recursive
  -- dutch.union.edu and eliphalet.union.edu don't answer queries via TCP
  -- both name servers on same subnet? (149.106.160.{3,14})
  -- root@eliphalet.union.edu not accepting email
  -- TTLs may need tweaking (900<3600, refresh is 10800 but should be at least 14400, minimum is 172800, but should be no more than 86400
  -- no IPv6 name servers
  -- no DNSSEC

Union College Miscellaneous

- Abuse.net has no registered reporting address; using postmaster@union.edu by default
- No SPF record
- Senderbase looks clean
- wpad.union.edu is NOT defined
  isatap.union.edu is NOT defined
- 4 union.edu pages being hit by blog/guestbook/wiki spam (google for “cheap phentermine” site:union.edu)

7.10 McGill

McGill Routing and Netblocks

- AS15318 (McGill University)
  upstreams AS376 (RISQ) and AS17356 (Vermont Telephone)
- whois.radb.net has point of contact info for AS15318 (from Bell), but no routing policy
- 132.206.0.0/16
  132.216.0.0/16
  142.157.0.0/16
  142.157.0.0/17
  142.157.128.0/18
  142.157.192.0/18
  192.197.121.0/24
  198.168.128.0/18
  199.202.80.0/22
  199.202.84.0/23
  199.202.98.0/23

McGill Name Servers

- Actual/potential name server issues/notes:
  -- mcgill.edu: total parent/child glue mismatch -- mismatch between name servers declared for domain (ns1.mcgill.edu and ns2.mcgill.edu) and name servers reported by ns1.mcgill.edu (moka.cc.mcgill.ca, ns1.cim.mcgill.ca, ns1.mcgill.ca, kona.cc.mcgill.ca, oolong.cc.mcgill.ca)
  -- moka.cc.mcgill.ca, ns1.mcgill.ca, kona.cc.mcgill.ca, oolong.cc.mcgill.ca are open recursive
  -- ns1.cim.mcgill.ca allows zone transfer of mcgill.edu
  -- TTLs may need adjustment, 600<3600, refresh (3600<14400)
  -- No IPv6 name servers
  -- No DNSSEC

McGill Miscellaneous

- Abuse.net knows about postmaster@mcrcim.mcgill.edu for mcgill.edu, but postmaster@mcgill.ca and contact.ncs@mcgill.ca for mcgill.ca
- No SPF record for mcgill.edu or mcgill.ca
- no mx for mcgill.edu, one mx for mcgill.ca
- wpad.mcgill.edu, wpad.mcgill.ca are NOT defined
  isatap.mcgill.edu, isatap.mcgill.ca are NOT defined
- Mcgill.ca has pages that are being hit by blog/guestbook/wiki spam (google for “cheap phentermine” site:mcgill.ca)

7.11 Missouri State

Missouri State Routing

- Missouri State has 146.7.0.0/16
- That netblock is routed via AS2572 (more.net)
- whois.radb.net has point of contact info for AS2572, but no routing policy

Missouri State Name Servers

- Real/potential name server issues/notes:
  -- argus.more.net doesn’t answer queries over UDP or TCP
  -- Additional name server listed at child:
  canopus.missouristate.edu
  -- canopus.missouristate.edu --> 146.7.4.137 --> NXDOMAIN
  -- sirius.missouristate.edu --> 146.7.4.136 --> NXDOMAIN
  -- canopus.missouristate.edu doesn’t answer over TCP
  -- sirius.missouristate.edu doesn’t answer over TCP
  -- canopus and sirius use non-date-format serial numbers
  -- No IPv6 nameservers
  -- No DNSSEC

Missouri State Miscellaneous

- Abuse.net has netabuse@missouristate.edu and postmaster@missouristate.edu
- Missouristate.edu has an SPF record
- Senderbase is mostly good (one poor host, 146.7.213.11, listed on NJABL; additional hosts sending mail but w/o in-addr's defined)
- wpad.missourstate.edu is NOT defined
  isatap.missouristate.edu is NOT defined
- Bunch of missouristate.edu pages being hit by blog/guestbook/wiki spam (google for "cheap phentermine" site:missouristate.edu)

7.12 Lancaster University (www.lancs.ac.uk)

Lancaster University Routing

- Netblock 148.88.0.0/16
- Routed by AS786 (JANET, UK)

Lancaster University Name Servers

- Real/potential name server issues/notes:
  -- Additional name servers at child: dns.lancs.ac.uk and dns2.lancs.ac.uk
  -- SOA refresh TTL is 10800, smaller than recommended 14400
  -- No offsite name servers?
  -- No IPv6 name servers
  -- No DNSSEC

Lancaster University Miscellaneous

- Abuse.net has postmaster@lancs.ac.uk, abuse@ja.net, abuse@lancs.ac.uk, and irt@csirt.ja.net
- No SPF record
- Senderbase looks okay
- wpad.lancs.ac.uk IS defined (Gold star! I was hoping at least one example domain would be taking care of this!)
  isatap.lancs.ac.uk is NOT defined
- Lancs.ac.uk does NOT appear to be getting hit with blog/guestbook/wiki spam (e.g., googling for “cheap phentermine” site:lancs.ac.uk finds nothing)