Securing DNS

An Educause Security Professionals Conference
Pre-Conference Seminar

1:00-4:30PM, April 10th, Nat Hill Room
Denver, Colorado

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

Welcome to the Security Professionals Conference and to Denver, Colorado!

- Let me be among the first to welcome you to this year's Security Professional Conference, and to the mile high city!
- 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 Denver Ballroom prefunction area on lower level 2 near the registration desk, 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/dns-tutorial/

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://oarc.isc.org/)

- 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-not-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," see http://www.doxpara.com/slides/BH_EU_05-Kaminsky.pdf) Two other great Kaminski DNS-related talks are "Black Ops 2004@LayerOne," see http://www.doxpara.com/bo2004.ppt , and "Black Ops of TCP/IP 2005," see http://www.doxpara.com/slides/Black%20Ops%20of%20TCP2005_Japan.ppt

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

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 help.yahoo.com/l/us/yahoo/mail/original/abuse/abuse-58.html
- Other sites may 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-02.txt
  (expires August 18, 2007)
  -- Suggested Generic DNS Naming Schemes for Large Networks and Unassigned hosts
  http://tools.ietf.org/wg/dnsop/
  draft-msullivan-dnsop-generic-naming-schemes-00.txt
  (expired October 2006)

- **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" and a DHCPREQUEST and a DHCPACK later its all over…
- 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 campus, right?)
- Two examples:
  -- From the the BIND FAQ (http://www.isc.org/index.pl?/sw/bind/FAQ.php):
  "**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 Infrastructural 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
  -- 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/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 web 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?

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 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 .com instead of a .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:

Berkeley:	http://net.berkeley.edu/policy_review/DNS.new.shtml
Cornell:	http://www.policy.cornell.edu/vol5_6.cfm
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
KS State:	http://www.k-state.edu/cns/policy/dns.html
Michigan:	http://spg.umich.edu/pdf/601.15-1.pdf
NYU:	http://www.nyu.edu/its/policies/dnsserv.html
Penn State:	http://tns.its.psu.edu/policies/dns.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)

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

- … as a new way of **identifying** infected systems (see, e.g., http://aharp.ittns.northwestern.edu/talks/bots-dns.pdf)
- … as a new way of **mitigating** infected systems
- … as a new way of "**monetizing**" typos and other domain name resolution "misses"
- … as something which will **needs to be fixed** after miscreant name servers get taken off the air.
- And then there'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 retrans-mission 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 doesn't provide a 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) requires 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 gigapop level, or all the way downstream at the host level.

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

  That said, catching at least some problematic traffic at the 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 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 six years old, **many** sites still do **NOT** do BCP38 filtering -- perhaps as many as 20-25% Internet wide. (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 .edu networks)
  -- overhead on the routers (but if that overhead is material enough to be a "show stopper," you should be upgrading anyway)

- "Too busy" (or other 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 How 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.
- I would also urge you to review (April 19, 2006) draft-savola-bcp84-urpf-experiences-00.txt
- 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 background, first.

Thinking A Little About DNS

- Most users never really think about how DNS works* -- they just take it for granted that entering http://www.uoregon.edu/ in their web browser will take them to the University of Oregon home page. In order for that to happen, however, the web browser needs to be able to find out that www.uoregon.edu resolves to the IP address (or "dotted quad") 128.223.142.13

- The web browser, and ultimately the user, relies on the domain name system to do that name-to-dotted quad translation.

- DNS is thus a critical network service.

* Geeks please see RFC1035

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 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 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, it is 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 estimate that 45% of .edu name servers are 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 UO's name servers were open until we secured them this past February 1st, 2006. See, for example: http://cc.uoregon.edu/cnews/winter2006/recursive.htm
- **If *our* domain name servers were open recursive until Feb 2006, *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…

- **Carnegie Mellon:** "Recursive DNS Server Operation Guideline," http://cmu.edu/computing/documentation/policies_dnsservers/dnsservers.html
- **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
- **University of Chicago:** "Curtailing Chicago Recursive Domain Name Service Access" http://support.uchicago.edu/announcements/secure/dns/index.html

The Problem Isn't "Just" About DDoS, Either

- By the way, if you aren't yet sufficiently motivated to "bite the bullet" and fix your DDoS-exploitable domain name servers, 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. (see, for example: http://www.lurhq.com/dnscache.pdf)

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.

  The innocent victim 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 that spam-related malware is notorious for generating large numbers of DNS queries for MX host information (so the spamware can determine where it should connect to dump its spam).
- Spam related malware may also refer to upstream command and control hosts by their FQDNs, thereby making it possible for the miscreants to repoint their malware's command and control host from one dotted quad to another, should the system currently "hosting" their C&C get filtered or cleaned up.
- At the same time that malware critically **relies** on DNS, ironically other malware may **also** be actively working to interfere with legitimate DNS uses.

Why Would Malware Interfere With DNS?

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

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

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

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

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

Examples of Malware Interfering with DNS

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

Another Example

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

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

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

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

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

More Examples of Malware Tweaking DNS

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

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

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

ZLOB Trojan (9/3/2006)

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

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

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

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

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

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

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** all user DNS traffic that isn't destined for the ISP's customer DNS servers (e.g., redirecting DNS is something that's common enough that Cisco even includes redirecting DNS as an example for its Intelligent Services Gateway, see: http://www.cisco.com/en/US/products/ps6566/products_configuration_guide_chapter09186a0080630d65.html#wp1048400)
  -- **<u>selectively</u> redirecting user DNS traffic**, if it appears that the customer is infected (e.g., Simplicita's commercial DNS switch)

"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 recently noted in an IETF technical plenary:

  "DNSSEC deployment may be hampered by transparency barriers."

  […]

  "DNS Namespace Mangling

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

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

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.**
- **A nice resource for folks who want to do this sort of blocking: http://doc.bleedingthreats.net/bin/view/Main/BlackHoleDNS**

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

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

Users May Tinker With The Hosts File, Too

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

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

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

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

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

  "Here's what DnsHijack enables you to do:

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

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

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

MS Windows and DNS Cache Pollution

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

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 two pages.
  See: http://www.isc.org/index.pl?/sw/bind/arm93/

- As you now know, I'm a bit more verbose. :-)

Basic DNS Sanity Check

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

  http://dnsreport.com/

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

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

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

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, at this time, this means 9.4.0
  -- If you're still on the 9.3 branch, make sure you get to 9.3.4
  -- If you're on 9.2, 9.2.8 will go EOL in August 2007; upgrade
  -- If you're on 8.x, upgrade
  -- If you're on 4.x, upgrade
- Updated versions of BIND can be downloaded from http://www.isc.org/index.pl?/sw/bind/
- **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 against the 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!

- Updated versions of OpenSSL are available from http://www.openssl.org/source/

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

  yum or apt-get can be your friend…

Determining the Version of BIND in Use

- % dig @phloem.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://www.rfc.se/fpdns/), but note that this may not always be able to distinguish dot release versions.
- 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:

  -- Bastille Linux, http://www.bastille-linux.org/

  -- National Security Agency Operating System Guides, http://www.nsa.gov/snac/

  -- Team Cymru IP stack tuning http://www.cymru.com/Documents/ip-stack-tuning.html
- 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

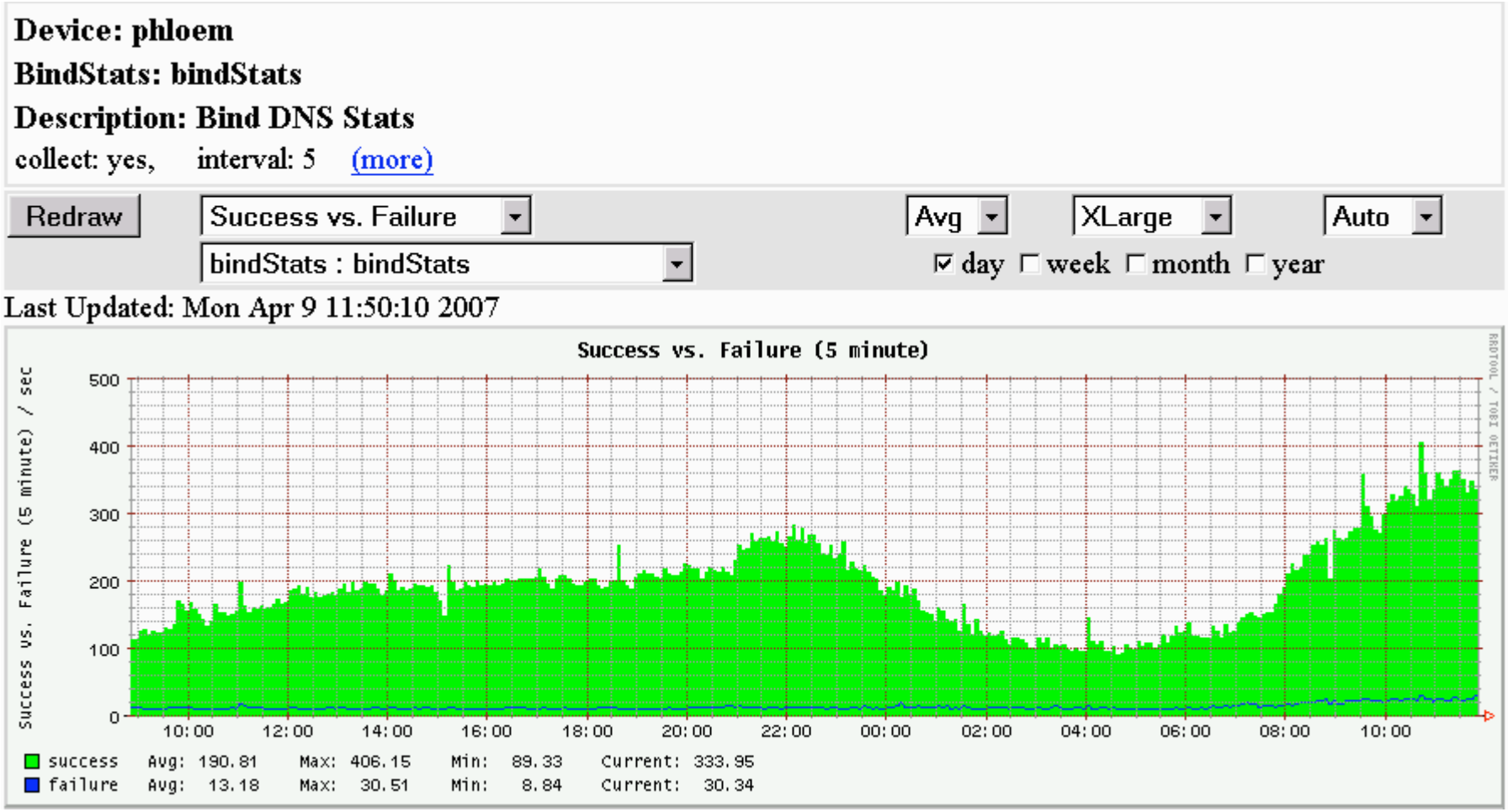
- Even if you're running a current version of BIND, it is still possible to configure it in more (or less secure) ways.
- A nice secure template to use for configuring BIND is the **Team Cymru Secure BIND Template**, available from http://www.cymru.com/Documents/secure-bind-template.html
- That configuration 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: MRTG/RRDtool

- You can (and should) graphically monitor DNS query traffic just as you monitor other network elements (such as transit bandwidth)…



DNS Monitoring: DNSTOP

- Beyond doing coarse DNS monitoring for things like query volume, you may also want to consider running DNSTOP (see http://dns.measurement-factory.com/tools/dnstop/).

  DNSTOP watches DNS query traffic via libpcap/tcpdump, and can report on things such as src/dst addresses, query types, TLDs, second level domains, third level domains, etc.

An Operational 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 country code. For example:

```
% dig pk @ns.pknic.net.pk axfr
pk.                    38400   IN     SOA     ns.pknic.net.pk.
ashar.pknic.net .pk. 1137367538 14400 7200 864000 21600
pk.                    38400   IN     NS      ns.pknic.net.pk.
pk.                    38400   IN     NS      AUTH51.NS.UU.NET.
pk.                    38400   IN     NS      AUTH101.NS.UU.NET.
0800.pk.               38400   IN     NS      dns3.websitehostings.us.
0800.pk.               38400   IN     NS      dns4.websitehostings.us.
0800shifa.pk.          38400   IN     NS      ns1.veriqual.NET.
0800shifa.pk.          38400   IN     NS      ns2.veriqual.NET.
1-world.pk.            38400   IN     NS      ns4.hsphere.net.pk.
[etc]
```

- See also: http://atrey.karlin.mff.cuni.cz/~mj/sleuth/

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 <u>at least</u> 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, running as multiprocessor/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/somecommercial providersactually 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, I'd encourage you to review Yale's excellent web page on this at http://babs.its.yale.edu/yalead/ddns.asp
- Nah, on second thought, just avoid dynamic updates. :-)

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

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

DNSSEC in a Nutshell

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

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

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

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

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

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

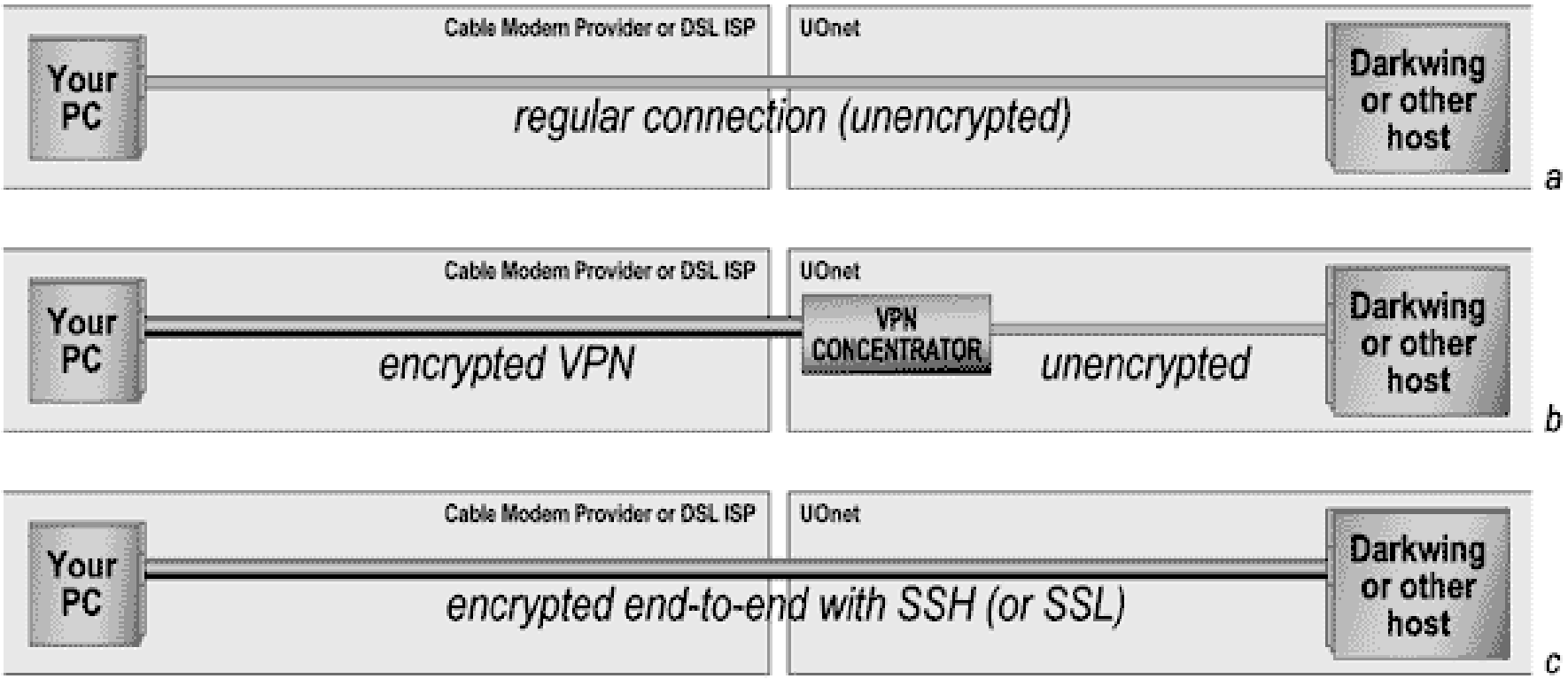
  DNSSEC is intentionally **NOT** designed to keep DNS data confidential, and it is also intentionally **NOT** designed to improve the availability of DNS data -- it's sole focus is on insuring the **integrity** of DNS data.

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

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

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

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



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

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

Microsoft DNS Client Support for DNSSEC

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

  "Client support for DNSSEC

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

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

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

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

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

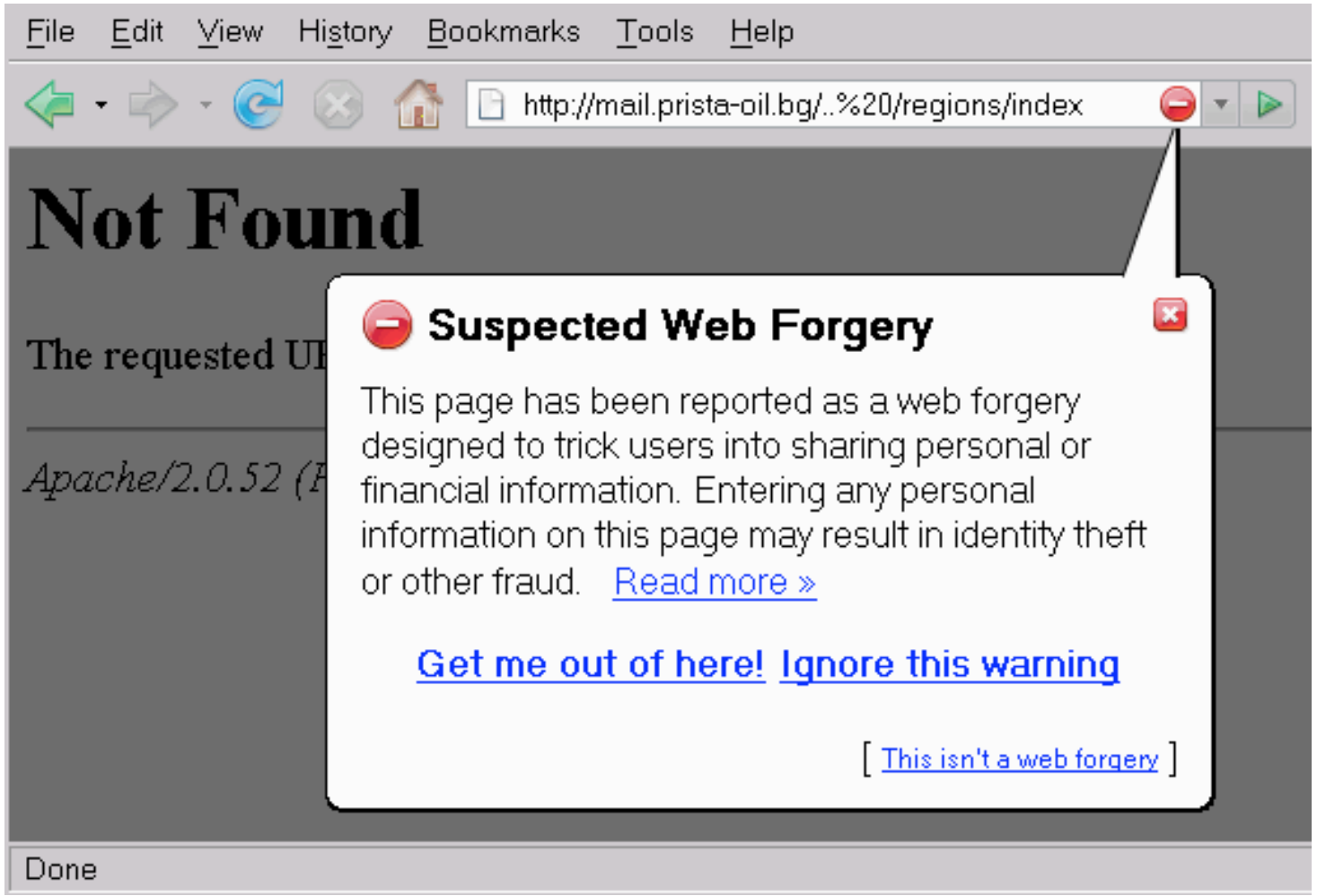
  -- something completely unrelated to DNSSEC might be busted

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

DNSSEC and Application Layer Visibility

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

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



Another Issue: The DNSSEC Trust Model

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

Signing The Root (".")

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

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

  San Diego, prior to IETF67

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

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

  [continues over the next two slides]

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

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

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

  D [University of Maryland, College Park] not present

  E [NASA Ames, Mountain View CA] not present

  F [ISC, 40 sites] ready needs enabling

  G [US DOD, Columbus OH] ready

  H [US ARL, Aberdeen MD] not present

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

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

  K [RIPE, 17 sites] yes needs enabling

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

  M [WIDE] ready"

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

But Someone Needs to Formally Ask…

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

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

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

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

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

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

  [snip]

  ;; AUTHORITY SECTION:

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

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

Islands Of Trust

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

DLV

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

What About the In-Addr Zones?

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

The Zone Enumeration Issue And NSEC3

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

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

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

- First, what name server products do people run?

BIND Dominates The DNS Server Market

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

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

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

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

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

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

What About Microsoft's DNS Servers?

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

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

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

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

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

  -- implements RFC4033, RFC4034, RFC4035

  -- "Beta: middle of 2007

  RTM: late 2007 or early 2008

  – General availability by first service pack of Longhorn Server"

How About PowerDNS?

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

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

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

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

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

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

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

EDNS0

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

Deployment of DNSSEC to Date? NIL

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

Another View of DNSSEC Penetration: UCLA's SecSpider

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

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

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

Why Aren't Folks Currently Using DNSSEC?

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

Or Are There More Fundamental Problems?

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

The Biggest Guy Out There

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

Unfortunately…

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

Federal Agencies And Commercial Partners

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

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

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

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

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

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

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

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

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 Washington University in St Louis

WUSTL Routing

- AS2552
  upstream AS7911 (L3); Abilene via AS19782 (Indiana Gigapop)
  downstream AS25887 (St. Louis Internet Access Consortium)
- **whois.radb.net doesn't know about AS2552**
- Network blocks advertised via AS2552:
  65.254.96.0/19 (SLIAC)
  128.252.0.0/16
  73 more specific announcements:
  128.252.0.0/23
  128.252.4.0/22
  128.252.8.0/21
  128.252.16.0/22
  128.252.20.0/23
  [etc]

WUSTL Name Servers

- 6 name servers:

NS1.wustl.edu	**<== open recursive**
WUGATE.wustl.edu	**<== open recursive**
WUMSDNS1.wustl.edu	**<== open recursive**
alpha.louisiana.edu	**<== open recursive**
SEASDNS.wustl.edu	**<== stealth**
WUMSDNS2.wustl.edu	**<== stealth**

- Actual/potential name server-related issues:

  -- Open recursive servers and stealth servers
  -- Inconsistent SOA serial numbers, w/o correct RFC1912 2.2 date
  -- Bind versions reported; **8.2.4 and 9.3.2** are in use

WUSTL rDNS Format

- rDNS format fails to distinguish between static and dynamic hosts for the purpose of direct-to-MX mail servers; these and many others all emit email…

128.252.17.170	ip-17-170.wustl.edu
128.252.124.110	spgwa.wustl.edu
128.252.192.192	pathmasq.wustl.edu
128.252.17.171	ip-17-171.wustl.edu
128.252.17.206	mail2-wusm-pcf.wustl.edu
128.252.223.130	expurgate2.wustl.edu
128.252.132.36	gwb-po.gwb.wustl.edu
128.252.117.18	biosmail2.biostat.wustl.edu
128.252.124.88	beckermail.wustl.edu
128.252.17.156	imspammer.im.wustl.edu
128.252.17.199	imspam2.im.wustl.edu
128.252.117.3	mailgate.biostat.wustl.edu

WUSTL and dnswalk

% dnswalk wustl.edu.
Checking wustl.edu.
[…]
Getting zone transfer of wustl.edu. from wumsdns2.wustl.edu...done.
SOA=WUGATE.wustl.edu contact=noc.WUGATE.wustl.edu
WARN: W.C.-Generator.wustl.edu A 128.252.97.230: no PTR record
WARN: 000C414ED65F.wustl.edu A 65.254.112.102: no PTR record
WARN: 0016B6E09375.wustl.edu A 65.254.112.66: no PTR record
WARN: 207_50.wustl.edu: invalid character(s) in name
WARN: 207_51.wustl.edu: invalid character(s) in name
WARN: 207_53.wustl.edu: invalid character(s) in name
[etc]

WUSTL Miscellaneous

- No SPF record defined
- Domain, netblock and ASN whois all updated 08/2006
- abuse.net knows about:

  postmaster@wustl.edu

  abuse@wustl.edu
- Both MX are in the same subnet?
  zippy.wustl.edu ==> 128.252.29.129
  mcfeely.wustl.edu ==> 128.252.29.1
- wpad.wustl.edu is defined
- Some indication that blog/guestbook/wiki spam is occurring (google for phentermine site:wustl.edu)

7.2 Ohio Northern University

ONU Routing

- Not using its own ASN; advertised as part of AS3112 (OARNet); Abilene connectivity also via AS3112
- **whois.radb.net doesn't know about as3112**
- Network block advertised via AS3112: 140.228.0.0/16 (**netblock whois last updated 11/1993**)
- Domain whois last updated 02/2006

ONU Name Servers

- 5 name servers

  ns1.onu.edu <== 140.228.10.14 (**same subnet** as ns2?)
  ns2.onu.edu <== 140.228.10.22 (**same subnet** as ns1?)
  ns1.oar.net <== **open recursive**
  ns2.oar.net <== **open recursive**
  ncnoc.ncren.net <== resolves to multiple IPs
  (128.109.193.1 and 192.101.21.1)

- Actual/potential name server related issues:
  -- both local name servers on the same subnet?
  -- open recursive name servers
  -- **SOA expire value high** at 3,600,000 seconds (41.67 days)
  -- ONU BIND version not displayed (good!); fpdns suggests BIND 9.2.3rc1 -- 9.4.0a0 (verify not a vulnerable version); OAR NS's using 9.2.6; NCREN NS's using 9.3.4

ONU and dnswalk

dnswalk onu.edu.
Checking onu.edu.
[…]
Getting zone transfer of onu.edu. from ns1.oar.net...done.
SOA=ns1.onu.edu contact=dns.onu.edu
WARN: acid.onu.edu CNAME factoids.onu.edu: unknown host
WARN: base.onu.edu CNAME factoids.onu.edu: unknown host
WARN: base-new.onu.edu CNAME factoids.onu.edu: unknown host
WARN: dgw-new.onu.edu CNAME austin-new.onu.edu: unknown host
WARN: fw.onu.edu CNAME fw-inside-fe-0-1.onu.edu: unknown host
WARN: law.onu.edu CNAME onulaw2.onu.edu: unknown host
[…]

ONU Miscellaneous

- Abuse.net knows about:
  abuse@onu.edu
  postmaster@onu.edu
- Has SPF record published, good!
- Very reasonable looking sending profile on Senderbase.org
- Both MX's on the same subnet?
  mx3.onu.edu ==> 140.228.10.73
  mx4.onu.edu ==> 140.228.10.74
- wpad.onu.edu is NOT defined
- Domain is not showing material signs of guestbook/blog/wiki spam

7.3 University of Kentucky

UKY Routing

- AS10437
  upstreams: AS7029 (Windstream), AS10490 (Southern Crossroads)
  Internet2 connectivity via Southern Crossroads
- **whois.radb.net doesn't know about AS10437**
- Originates
  128.163.0.0/16
  199.76.144.0/20
  199.76.160.0/19
  199.76.192.0/24
  204.198.72.0/22
  204.198.76.0/23
  206.240.24.0/22
- AS10437 provides transit for additional prefixes including 147.133.0.0 (Morehead State), 161.6.0.0 (Western Kentucky), 170.180.0.0/14 and 170.185.0.0 (KY Dept of Ed), and others

UKY Name Servers

- 3 name servers

  ncc.uky.edu **<== open recursive**
  nic.net.uky.edu **<== open recursive**
  nit.net.uky.edu **<== stealth**

- Actual/potential name server related issues:
  -- open recursive name servers
  -- stealth name server
  -- non-standard serial number for the SOA (1641)
  -- **short expiration for the SOA** (43200)
  -- BIND version **9.3.4**

UKY Miscellaneous

- Abuse.net knows about
  postmaster@uky.edu
- No SPF record defined
- All three MX host IPs are on the same subnet:
  mg1.uky.edu <== 128.163.184.178
  mg2.uky.edu <== 128.163.184.179
  mg3.uky.edu <== 128.163.184.180
- **wpad.uky.edu is NOT defined**
- Uky.edu appears to be getting *heavily* abused by **blog/guestbook/wiki spammers** (as an example, google for phentermine site:uky.edu)

7.4 Oakland University

Oakland University Routing

- Doesn't have its own ASN, advertised as part of Merit's AS237; Abilene connectivity also via Merit
- AS237 is registered in whois.radb.net
- Has 141.210.0.0/16, netblock whois last updated 06/2006
- Domain whois last updated 05/2006

Oakland University Name Servers

- 5 name servers

  ns1.oakland.edu <== **141.210.2.2**
  ns2.oakland.edu <== **141.210.2.3**
  dns1.merit.net
  dns2.merit.net
  dns3.merit.net

- Actual/potential name server issues/notes:

  -- all servers are closed to recursion! Yeah! :-)
  -- are both Oakland name servers on the same subnet?
  -- Oakland name servers suppress version info, so confirm that a non-vulnerable name server version is in use; Merit name servers are using 9.3.1
  -- non-standard SOA serial number format

Oakland University Miscellaneous

- Abuse.net knows about
  abuse@oakland.edu
- SPF record defined (good!)
- Only one MX record
- **wpad.oakland.edu is NOT defined**
- Oakland.edu appears to be getting abused by **guestbook/blog/wiki spammers** (as an example, google for phentermine site:oakland.edu)

7.5 University of Auckland

Auckland Routing

- AS9431
- AS9431 is registered in whois.radb.net
- Upstream AS4768 (TelstraClear), heavily prepended, and AS38022 (REANNZ National Research Network)
- 130.216.0.0/16 Auckland University
  202.36.244.0 (Auckland College of Education)
  202.36.245.0 (Auckland College of Education)
  202.37.88.0 (**APNIC notes that the whois contacts for this range are historical/non-contactable**)

Auckland Nameservers

- 3 name servers

  dns1.auckland.ac.nz <== **130.216.1.2**
  dns2.auckland.ac.nz <== **130.216.1.1**
  pubsec.domainz.net.nz <== 202.46.160.4

- Actual/potential name server issues/notes:

  -- both Auckland name servers appear to be on the same subnet
  -- all name servers closed to recursion
  -- Auckland name servers are running **BIND 9.3.2**; domainz.net.nz is not advertising its version information, but fingerprints as BIND 9.2.3rc1 -- 9.4.0a0; confirm version manually
  -- Auckland name servers are returning NS records with comparatively **short TTLs (1800 seconds)**

Auckland Miscellaneous

- abuse.net knows about
  abuse@auckland.ac.nz
  postmaster@auckland.ac.nz
- No SPF record
- All MX IP's appear to be in the same subnet:
  chico.itss.auckland.ac.nz <== 130.216.190.12
  harpo.itss.auckland.ac.nz <== 130.216.190.13
  zeppo.itss.auckland.ac.nz <== 130.216.190.14
  groucho.itss.auckland.ac.nz <== 130.216.190.11
- **wpad.auckland.ac.nz doesn't exist**
- Auckland appears to be getting *heavily* abused by some **blog/guestbook/wiki spammers** (as an example, google for phentermine site:auckland.ac.nz)

7.6 Medical University of South Carolina

MUSC Routing

- AS13429 (last updated 06/1999)
- **whois.radb.net doesn't know about AS13429**
- Upstream AS209 (Qwest), AS10490 (Southern Crossroads); Abilene connectivity via Southern Crossroads
- 128.23.0.0/16, netblock whois last updated 01/2005
- Domain whois last updated 01/2004

MUSC Name Servers

- 2 name servers

  chimera.musc.edu <== **128.23.34.1**
  tangent.musc.edu <== **128.23.34.2**

- Actual/potential name server issues/notes:

  -- both name servers are closed to recursion, good!
  -- are both name servers on the same subnet?
  -- neither name server provides their versions (unless those are biblical name servers :-)); both fingerprint as BIND 9.2.3rc1 -- 9.4.0a0 and should be manually checked to confirm they're not vulnerable
  -- **TTLs are both sort of on the short side at 3600**

MUSC Miscellaneous

- Abuse.net knows about
  abuse@musc.edu
- SPF record is defined, good! (however what do you want to have happen for mail from undefined sources? Softfail or ?)
- There appears to only be a single MX server defined
- Senderbase.org looks nice and clean for this domain
- **wpad.musc.edu is NOT defined**
- Some guestbooks/blogs/wikis appear to be getting abused; google for phentermine site:musc.edu to see examples (caution: some pages returned by that query may very well be legitimate, as you might expect for a medical university)

7.7 George Mason University

GMU Routing

- AS11279 (last updated 06/1998)
- AS11279 is in whois.radb.net, however there's the remarks:

  remarks: George Mason University's Multi-homed AS
  Test with two Class C's, move main network later
  mnt-by: MAINT-AS11279
  changed: [deleted] **19981024**
  source: RADB
- Upstream AS6461 (Metromedia Fiber)
- Abilene connectivity via AS40220 (Mid-Atlantic Terascale Partnership, Virginia Tech) on to AS10886 (MAX Gigapop)
- 129.174.0.0/16 (netblock whois last updated 08/2004)
  199.26.254.0/25 (part of 199.26.254.0/24, **last updated 04/1995**)
- Domain whois last updated 3/2002

GMU Name Servers

Hmm….

```
gmu.edu.            172800  IN      NS      UVAARPA.VIRGINIA.edu.
gmu.edu.            172800  IN      NS      THALASSA.gmu.edu.
gmu.edu.            172800  IN      NS      PORTAL-0-8.gmu.edu.
;; Received 152 bytes from 192.5.6.32#53(A3.NSTLD.COM) in 81 ms

gmu.edu.            86400   IN      A       129.174.1.52
gmu.edu.            86400   IN      NS      mulhall.gmu.edu.
gmu.edu.            86400   IN      NS      uvaarpa.virginia.edu.
gmu.edu.            86400   IN      NS      thalassa.gmu.edu.
gmu.edu.            86400   IN      NS      portalknot.gmu.edu.
gmu.edu.            86400   IN      NS      ruth.gmu.edu.
gmu.edu.            86400   IN      NS      magda.gmu.edu.
gmu.edu.            86400   IN      NS      archon.gmu.edu.
gmu.edu.            86400   IN      NS      henoch.gmu.edu.
gmu.edu.            86400   IN      NS      sargon.gmu.edu.
;; Received 388 bytes from 128.143.2.7#53(UVAARPA.VIRGINIA.edu) in 81 ms
```

GMU Nameservers (continued)

- portalknot.gmu.edu <== **open recursive; stealth**;
  SOA, but not listed at the parent servers as an NS
  (replace PORTAL-0-8.gmu.edu)
- PORTAL-0-8.gmu.edu <== aka portalknot
- thalassa.gmu.edu <== **open recursive**
- uvaarpa.virginia.edu <== **open recursive**
- magda.gmu.edu <== **stealth**
- ruth.gmu.edu <== **stealth**
- sargon.gmu.edu <== **stealth**
- archon.gmu.edu <== **stealth**
- henoch.gmu.edu <== **stealth**
- mulhall.gmu.edu <== **stealth**

GMU Nameservers (continued 2)

- Actual or potential name server issues/notes:

  -- PORTAL-0-8.gmu.edu vs. portalknot.gmu.edu issue
  -- open recursive name servers
  -- stealth name servers
  -- BIND version **9.3.2 in use on at least some servers**; uvaarpa.virginia.edu conceals the version information, but it fingerprints as BIND 9.2.3rc1 -- 9.4.0a0 (manually confirm that it isn't running a vulnerable version)

GMU and dnswalk

% dnswalk gmu.edu.
Checking gmu.edu.
[…]
Getting zone transfer of gmu.edu. from thalassa.gmu.edu...done.
SOA=portalknot.gmu.edu contact=postmaster.portalknot.gmu.edu
WARN: www.1stplace.gmu.edu A 129.174.25.174: no PTR record
WARN: www.4jrs.gmu.edu A 129.174.125.49: no PTR record
WARN: abyss.gmu.edu A 129.174.15.16: no PTR record
BAD: ad.gmu.edu NS addc1.ad.gmu.edu: unknown host
BAD: ad.gmu.edu NS addc2.ad.gmu.edu: unknown host
WARN: adobo.gmu.edu A 129.174.15.36: no PTR record
WARN: www.akpsi.gmu.edu CNAME www.som2.gmu.edu: CNAME (to SOMWEB27.gmu.edu)
[etc]

GMU Miscellaneous

- Abuse.net knows about
  postmaster@gmu.edu
  abuse@gmu.edu
- No SPF record
- Only one MX host (mx-h.gmu.edu); that house banners as talos2.gmu.edu ESMTP SonicWALL (5.0.2.8439) but
  mx-h.gmu.edu ==> 129.174.0.99
  talos2.gmu.edu ==> 129.174.0.107

  (and just in passing, I don't know if I'd be as forthcoming as you are at https://antispam.gmu.edu/about.html ; GMU appears to be one of six sites mentioning that particular release of SonicWall)
- Senderbase looks fine for the gmu.edu domain
- **wpad.gmu.edu is NOT defined**
- Some indication that guestbooks/blogs/wikis are being spammed; check google for phentermine site:gmu.edu

7.8 Fort Lewis College

fortlewis.edu Routing

- Using IPs such as 129.19.131.74 (www.fortlewis.edu) and 129.19.131.99 (mail.fortlewis.edu), however **I'm not seeing a block specifically SWIP'd or rwhois'd to fortlewis.edu**; those IP's are announced as part of the 129.19.0.0/16 aggregate
- That block includes IPs associated with
  AS12145 (Colorado State University),
  AS16519 (CU Denver), and
  AS31991 (Platte River Power Authority)
  and those ASNs are aggregated by AS14041 (UCAR)
- **whois.radb.net doesn't know about AS12145 or AS14041**
- Abilene connectivity is via Front Range Gigapop
- Whois data for 129.19.0.0/16 was last updated 01/1991
- Note: Fortlewis.edu may have some potential for confusion with Ft Lewis, Washington. Fort Lewis, Washington is the home of numerous military elements, including special operations units.

fortlewis.edu Name Servers

- 4 name servers
  edns.fortlewis.edu **<== 10.38.1.51, stealth, RFC1918!**
  ns2.fortlewis.edu **<== open recursive**
  yuma.acns.colostate.edu **<== open recursive**
  ns1.westnet.net
- Real/potential name server issues/notes:
  -- open recursive name servers
  -- leaking stealth name server with RFC1918 address
  -- **SOA serials do not agree**
  -- **long SOA expire (2592000 seconds, e.g., 30 days)**
  -- ns2.fortlewis.edu fingerprints as Windows 2003; carefully review the security of that host (at a minimum, run MSBSA 2.1)
  -- yuma.acns.colostate.edu reports that it is running "**8.2.2-P5+Fix_for_CERT_till_10_15_03**"… hmm…
  -- ns1.westnet.net reports that it is running "unixops standard isc" and fingerprints as BIND 9.2.3rc1 -- 9.4.0a0; confirm version

Fort Lewis and dnswalk

% dnswalk fortlewis.edu.
Checking fortlewis.edu.
[…]
Getting zone transfer of fortlewis.edu. From yuma.acns.colostate.edu...done.
SOA=ns2.fortlewis.edu contact=postmaster.fortlewis.edu
WARN: edns.fortlewis.edu A 10.38.1.51: no PTR record
WARN: myflc.fortlewis.edu CNAME ghs.google.com: CNAME (to ghs.l.google.com)
WARN: choose.fortlewis.edu A 216.83.6.65: no PTR record
WARN: edtoc.fortlewis.edu CNAME k12.fortlewis.edu: unknown host
WARN: news.fortlewis.edu CNAME news-2.sni.net: unknown host
2 failures, 5 warnings, 0 errors.

Fort Lewis Miscellaneous

- Abuse.net is using postmaster@fortlewis.edu by default (may want to register a preferred abuse reporting address)
- Has an SPF record, good!
- Only one MX host, banners as antispam.fortlewis.edu ESMTP SonicWALL (5.0.2.8415) (14 other sites have that same version of SonicWall according to Google; this is an older version than the version seen from GMU…)
- Senderbase isn't reporting any traffic for this domain or for 129.19.131.0/24
- **wpad.fortlewis.edu is NOT defined**
- No indication that fortlewis.edu is being hit by blog/guestbook/ wiki spam

7.9 Franklin and Marshall College

Franklin and Marshall Routing

- AS31746 (last updated 12/2003)
  upstream AS3593 (EPIX)
  no Internet2 connectivity
- **whois.radb.net doesn't know about AS31746**
- EPIX connectivity may be DSL-based, e.g., last hop is:
  plns-208-111-192-22-pppoe.dsl.plns.epix.net (208.111.192.22)
- 155.68.0.0/16 (netblock whois last updated 10/2002)
- domain whois last updated 10/2006
- Also seeing (as of 2006-10-11):
  D&E Communications DANDE (NET-66-109-224-0-1)
  66.109.224.0 - 66.109.255.255
  Franklin & Marshall College DECM-1080 (NET-66-109-240-48-1)
  66.109.240.48 - 66.109.240.55
  routed by AS20124 (D&E Communications, Ephrata PA)

Franklin and Marshall Name Servers

- 5 name servers:
  DNSONE1.fandm.edu <== 155.68.1.122, **open recursive**
  DNSONE2.fandm.edu <== 155.68.1.123, **open recursive**
  DNS4.fandm.edu <== 66.109.240.50, **open recursive**
  DNSONESERVER.fandm.edu <== 155.68.1.100, missing at the domain, **open recursive**
  dnsone3.fandm.edu <== 155.68.1.105, **stealth** name server
- Real/potential name server issues/notes:
  -- open recursive name servers
  -- stealth name server; name server mentioned at the parent server is missing at the domain
  -- wrong SOA serial number format
  -- most name servers do not return version, but fingerprint as BIND 9.2.3rc1 -- 9.4.0a0, confirm version; DNS4.fandm.edu reports version **9.2.2** (if accurate, should be upgraded)

Franklin and Marshall Miscellaneous

- SOA: fandm.edu. 900 IN SOA dnsone1.fandm.edu. **please_set_email.absolutely.nowhere.** [etc]
- Abuse.net reports postmaster@fandm.edu used by default; might want to register preferred abuse reporting address
- No SPF record
- Two MX hosts:
  spammy1.fandm.edu <== 155.68.1.14
  spammy2.fandm.edu <== 155.68.1.16
  same subnet for both mail hosts?
- Senderbase is showing one interesting host:
  pcp006733pcs.fandm.edu (155.68.47.195)
- **wpad.fandm.edu is NOT defined**
- Seeing some indication that fandm.edu is being hit by blog/guestbook/wiki spam (google for phentermine site:fandm.edu)

7.10 University at Buffalo

Buffalo Routing

- AS3685
  upstream AS6395 (Broadwing)
- **whois.radb.net doesn't know about AS3685**
- Abilene connectivity via Nysernet
- Six prefixes:
  67.99.160.0/21 (Nysernet)
  128.205.0.0/16
  199.33.167.0/24 (Western New York Health Science Consortium)
  204.68.186.0/23 (Sisters of Charity Hospital)
  204.124.132.0/23 (Independent Health Association)
  204.124.134.0/24 (Independent Health Association)
  205.232.18.0/23

Buffalo Name Servers

- 4 name servers

  ns.buffalo.edu
  sybil.cs.buffalo.edu
  butler.acsu.buffalo.edu
  accuvax.northwestern.edu **<== open recursive**

- Actual/potential name server issues/notes:
  -- open recursive server
  -- unable to confirm version of name server software, but they fingerprint as 9.2.3rc1 -- 9.4.0a0; confirm version manually
  -- non-standard SOA serial number format

Buffalo rDNS Format

- rDNS format fails to distinguish between static and dynamic hosts for the purpose of direct-to-MX mail servers; these and many others all emit email…

128.205.119.15	mail.ap.buffalo.edu
128.205.7.58	defer.acsu.buffalo.edu
128.205.7.57	deliverance.acsu.buffalo.edu
128.205.134.23	urh-exch01.urh.buffalo.edu
128.205.6.88	warmfront.acsu.buffalo.edu
128.205.25.5	fate.eng.buffalo.edu
128.205.6.89	coldfront.acsu.buffalo.edu
128.205.4.140	upfront.acsu.buffalo.edu
128.205.25.103	thebrain.nsm.buffalo.edu
128.205.2.93	itsa-vpsamail2k.vpsa.buffalo.edu
128.205.2.9	prv-mail1.pn.buffalo.edu

[etc]

Buffalo Miscellaneous

- Abuse.net knows about
  postmaster@buffalo.edu
  abuse@buffalo.edu
  abuse@suny.edu
  abuse@sysadm.suny.edu
  abuse@broadwing.net
- No SPF record
- Domain MX hosts don't announce FQDN when bannering
- **"Special" high numbered MX (mx.buffalo.edu)**
- **wpad.buffalo.edu is NOT defined**
- Seeing some indications that buffalo.edu is being hit by blog/guestbook/wiki spam
  (google for phentermine site:buffalo.edu)

9. Some Miscellaneous DNS Topics

RUS-CERT Passive DNS Replication

- The RUS-CERT Passive DNS Replication server, see http://cert.uni-stuttgart.de/stats/dns-replication.php , allows you to do synthetic DNS queries in a very powerful way. (Consider contributing log data from your site!)
- For example, assume you wanted to know what FQDNs were associated with a given IP address. You could check for a PTR record, but many times a PTR will not be defined – that's when RUS-CERT Passive DNS can come to the rescue! It will return DNS records for a given IP based on A records that it has seen. It can also return NS records, and the domains which use those NS records, etc.
- For example, assume you happened across the domain porav.hk and you wanted to know if there were any additional related domains associated with it…

Query string: porav.hk

Query

The server returned the following data:

porav.hk	NS	ns1.badreze.net
porav.hk	NS	ns2.badreze.net
porav.hk	NS	ns1.porav.hk
porav.hk	NS	ns2.porav.hk

Query string: ns1.badreze.net

Query

The server returned the following data:

qveda.hk	NS	ns1.badreze.net
terfa.hk	NS	ns1.badreze.net
peola.hk	NS	ns1.badreze.net
posta.hk	NS	ns1.badreze.net
kysta.hk	NS	ns1.badreze.net
okiva.hk	NS	ns1.badreze.net
petva.hk	NS	ns1.badreze.net
kisub.hk	NS	ns1.badreze.net
purac.hk	NS	ns1.badreze.net
masoc.hk	NS	ns1.badreze.net
fihad.hk	NS	ns1.badreze.net
misad.hk	NS	ns1.badreze.net
pilod.hk	NS	ns1.badreze.net
sinod.hk	NS	ns1.badreze.net
klaud.hk	NS	ns1.badreze.net
zvabe.hk	NS	ns1.badreze.net
jyrce.hk	NS	ns1.badreze.net
ontverde.hk	NS	ns1.badreze.net
dinistine.hk	NS	ns1.badreze.net

Query string: 210.22.15.22

Query

The server returned the following data:

22.15.22.210.blacklist.spambag.org	CNAME	cncgroup.black
22.15.22.210.zz.countries.nerd.dk	A	127.0.0.156
ns1.qveda.hk	A	210.22.15.22
ns2.qveda.hk	A	210.22.15.22
www.qveda.hk	A	210.22.15.22
ns1.terfa.hk	A	210.22.15.22
ns2.terfa.hk	A	210.22.15.22
www.terfa.hk	A	210.22.15.22
ns1.peola.hk	A	210.22.15.22
ns2.peola.hk	A	210.22.15.22
www.peola.hk	A	210.22.15.22
ns1.posta.hk	A	210.22.15.22

Fast Flux Domains

- From time to time you may run into web pages which are "fast flux" domains, hosted on consumer broadband IP addresses, and changing over a large number of IP's. For example one pr0n name was seen on over 300 different IPs, including:

4.228.159.20	dialup-4.228.159.20.Dial1.Denver1.Level3.net
62.108.8.253	k8253.upc-k.chello.nl
68.44.215.111	c-68-44-215-111.hsd1.nj.comcast.net.
71.159.138.190	adsl-71-159-138-190.dsl.rcsntx.sbcglobal.net
75.31.214.13	adsl-75-31-214-13.dsl.irvnca.sbcglobal.net
80.243.24.100	243-24-100.elekta.lt
85.193.2.99	user2_99.ktkadan.cz
89.178.58.103	89-178-58-103.broadband.corbina.ru
91.122.57.0	ppp91-122-57-0.pppoe.avangard-dsl.ru
122.133.174.206	FL1-122-133-174-206.kng.mesh.ad.jp
212.220.101.118	NXDOMAIN

Other Topics of Interest