

Notes:

- DNS is one of the first (if not THE first) large scale, hierarchical, distributed, fault tolerant databases.
- Remarkable accomplishment that has stood up extremely well, especially considering it was designed in 1983
- Like most Internet protocols, it was designed to be reliable, not secure.
- Showing it's age.

Cloud background photo by turtlemom4bacon

http://www.flickr.com/photos/turtlemom_nancy/2046347762/ Creative Commons Attribution Generic 2.0 license

Stone angel by jaz1111 <http://www.sxc.hu/profile/jaz1111> Royalty Free SXC license

Presentation overview

- Why DNS security matters.
- DNS vulnerabilities – a quick recap.
- Real world exploits and consequences.
- Seven Deadliest Sins.
- How to protect your critical DNS infrastructure.

Notes:

Photo by ralphbijker <http://www.flickr.com/photos/17258892@N05/2588347668/>
Creative Commons Attribution 2.0 Generic License

DNS: Who cares?



Almost every OS and network application depend on DNS to function

OSs and applications trust DNS data **completely**.

No verification or sanity checks on data.

Notes:

Who cares?

Consider that nearly every operating system and networked application depend on DNS to function:

- Connecting to hosts
- Part of authentication

Data received from DNS are trusted completely by almost every application and OS: even security applications rarely perform any sanity checks on the data.

Photo source: Hemera. Used with permission. All rights reserved



DNS Compromise

- Defeat SSL protections (web and VPN).
- Redirect clients and staff to malicious sites.
- Intercept and modify email.
- Install malicious OS and application software updates.
- Hard-to-detect phishing attacks.
- Defeat of spam controls.
- Easy denial of service.

Notes:

Impact of compromised DNS or spoofed records:

Man-in-the-middle attacks against SSL, including SSL connections to web sites and SSL “dissolvable client” (Java and ActiveX-based) VPNs

Redirect your customers to malicious recreations of your web sites and web services.

Redirect your internal staff to malicious sites.

Fool servers and desktop automatic update mechanisms to download malware

Defeat your spam and phishing controls

Really easy denial of service attacks – DNS zone transfers can be large. DNS queries can be amplified.
http://www.us-cert.gov/reading_room/DNS-recursion033006.pdf

Photo by hotblack <http://www.morguefile.com/archive/display/189364> Morguefile free photo license

DNS: a security blind spot

- Few recognize how critical DNS is.
- Few recognize how vulnerable it is.
- Do **YOU** address DNS security in your:
 - Network architecture?
 - Threat risk assessments?
 - Vulnerability assessments?
 - Network monitoring?

Notes:

For the risk assessment folks in the audience:

DNS is often ignored. Rarely seen as any part of a TRA, Vulnerability assessment or pen test.

(Though this has started to change since the excitement in 2008 with Dan Kaminsky's exploit)

The DNS Protocol

- DNS invented in 1983.
 - One of the first globally distributed databases.
 - Global, scalable, redundant, adaptable.
- An outstanding achievement!
- Like all original Internet protocols, security was not a design goal.

Photo by ralphbijker <http://www.flickr.com/photos/17258892@N05/2588347668/>
Creative Commons Attribution 2.0 Generic License

DNS vulnerabilities

Protocol	Vulnerabilities in the DNS protocol itself
Architecture	Network architecture mistakes
Software	Server & client software vulnerabilities
Operations	Configuration and administration
Endpoints	Vulnerable client configurations

Notes:

DNS vulnerabilities exist at several levels

The DNS protocol itself is broken and cannot be fixed. That is part of what caused last summer's excitement with the Dan Kaminsky vulnerability. See doxpara.com

Due to the unfixable issues in the protocol, compensating controls must be used. BUT...

Few networks are designed with DNS properly deployed

Software (server and client) are often buggy as hell

Operational issues abound, as we'll see

Endpoints: If an attacker can change DNS settings on workstations and servers, they can do tremendous damage.

Protocol Vulnerabilities

- UDP based:
 - No “three way handshake.”
 - Easy to spoof origin IP of queries and replies to appear to come from authorized sources.
- Weak message authentication:
 - Data not cryptographically signed.
 - Transaction IDs are the only assurance of message authenticity (match DNS queries with replies).
 - Only 65525 possible transaction IDs.

Protocol Vulnerabilities

- July 2008: Researcher Dan Kaminsky shows DNS cache poisoning is easier than previously thought.
- Patches were issued to add random ports to DNS queries in addition to random transaction IDs.
- But the problem is a protocol design flaw. It cannot be fixed.
- The patch only makes exploits take longer (10 hours vs. 10 minutes)



Notes:

Photo by bb_matt http://www.flickr.com/photos/bb_matt/207102084/
Creative Commons Attribution Generic 2.0 license

Cache poisoning

- Inserting false records in your DNS cache.
- Every DNS query is a race condition:

The first reply with the correct transaction ID wins*

- Attackers can spoof IP addresses, but also replace the authoritative server IPs for all future queries for that domain.

*(Some conditions apply)



Notes:

Cache poisoning is the most popular objective with DNS attacks

DNS queries are cached locally to reduce bandwidth and improve performance

Poisoned records live in the cache until they expire (could be weeks)

Cached in multiple places:

Endpoints (desktop and servers)

Firewalls and proxy servers

Your organization's DNS forwarder

Your upstream's DNS

Insert fake data into your DNS cache

- Not just IP addresses. Attacker can also replace the NS records for a domain, making all subsequent queries go to the attacker's DNS

Photo source: Hemera. Used with permission. All rights reserved

Evilgrade

- Emulates the update servers of

Java runtime	OpenOffice
WinZip	Winamp
MacOS	iTunes
+ pluggable framework to add more	

- Tells client software an update is available
- Packages any code you like into in the right format for the target



Notes:

Cache poisoning attack example: Evilgrade

- Not malware. A proof of concept by a researcher
- Attacker first poisons your DNS cache to point (for example) update.java.com to their own IP address
- Evilgrade emulates update servers for multiple applications
- Packages arbitrary code in the update package format expected by the endpoint software (headers, version numbers, etc)
- Can integrate into Metasploit
- Many endpoint updaters run with admin privileges: attacker can affect more than just the app being updated: install rootkits, malicious drivers, etc.
- Windows update uses digital signatures so more difficult to attack this way

<http://www.infobyte.com.ar/down/isr-evilgrade-Readme.txt>

Photo source: Hemera. Used with permission. All rights reserved



Notes:

http://www.theregister.co.uk/2009/04/22/msn_hijacking/

<http://blogs.zdnet.com/security/?p=1356>

But are DNS attacks actually happening? You betcha! Let's look at some recent ones.

- Compromised domain registrar web interface via an SQL injection.
- Able to change the WHOIS settings to point domains to other DNS servers.

Same group apparently hijacked ICANN and IANA DNS servers in 2008.

Photo by chrisdlugosz <http://www.flickr.com/photos/chrisdlugosz/2805048271/>

Creative Commons Attribution Generic 2.0 license

Famous DNS Hijacks (2)

- April 2008: Puerto Rico registrar Gauss Research DNS host for local versions of:

Google Microsoft Yahoo
Paypal Nike Dell Nokia

- DNS host compromised via SQL injection.
- Attackers redirected lookups to blank sites and "you are hacked" pages.

Notes:

http://www.theregister.co.uk/2009/04/22/msn_hijacking/

<http://blogs.zdnet.com/security/?p=1356>

Photo by chrisdlugosz <http://www.flickr.com/photos/chrisdlugosz/2805048271/>

Creative Commons Attribution Generic 2.0 license

Famous DNS Hijacks (3)

- April 2008: DNS cache poisoning attack against major Brazilian bank “Bandesco.”
- Clients redirected to fake websites that attempted to steal passwords and install malware.

Notes:

<http://www.spamfighter.com/News-12243-Leading-Brazilian-Bank-in-Grip-of-DNS-Cache-Poisoning-Attack.htm>

http://www.theregister.co.uk/2009/04/22/bandesco_cache_poisoning_attack/

Photo by chrisdlugosz <http://www.flickr.com/photos/chrisdlugosz/2805048271/>

Creative Commons Attribution Generic 2.0 license

Famous DNS Hijacks (4)

- June 2008: DNS of **ICANN** and **IANA** hijacked by an activist group!
- Attackers managed to change the WHOIS records of the domains to point to their own DNS servers.
- Domains were redirected to a site with a message from the group.

Notes:

<http://blogs.zdnet.com/security/?p=1356>

- But the most embarrassing DNS hijack took place the previous year
- June 2008, the DNS servers of ICANN, the Internet Corporation for Assigned Names and Numbers, and IANA, the Internet Assigned Numbers Authority were hijacked by altering the WHOIS records
- Redirected to a page with the group's message on it.

Photo by chrisdlugosz <http://www.flickr.com/photos/chrisdlugosz/2805048271/>

Creative Commons Attribution Generic 2.0 license



Notes:

And here are the seven deadliest sins of DNS security.

These are the ones that in my experience present the greatest risk and are most commonly seen in the real world.

Cloud background photo by turtlreemom4bacon

http://www.flickr.com/photos/turtlemom_nancy/2046347762/

Creative Commons Attribution Generic 2.0 license

Stone angel by jaz1111 <http://www.sxc.hu/profile/jaz1111>

Royalty Free SXC license

One server for everything

Authoritative server,
DNS resolver,
Query cache,
floor wax,
dessert topping...



Notes:

How many have just one DNS server doing everything:

- Authoritative for your public records?
- Authoritative for your internal records? (“split DNS”)
- Internal hosts use the same box as their forwarder?
- Also caches responses?

There are multiple DNS functions:

- Authoritative records (external)
- Authoritative records (internal)
- resolving
- caching & forwarding

Authoritative and caching functions should NEVER be combined.

Public and private records should NEVER be combined. Using ACLs such as in BIND to implement “split DNS” is not sufficient. Use separate instances on same server or (better) separate servers / VMs

Photo by tnz http://www.flickr.com/photos/jesse_sneed/2383953694/

Creative Commons Attribution Generic 2.0 license



Notes:

Cache poisoning is much easier if the attacker can do arbitrary queries on your DNS servers.

Recursive queries: queries for domains the server is NOT authoritative for.

Only allow recursive queries for your own users.

Zone transfers reveal all your DNS records. You should never put non-public data into DNS, but many do. A zone transfer reveals all that.

Zone transfers should be restricted to your secondaries. Use IP address ACLs or even better, use TSIG as well.

Dynamic updates: BIND and Active Directory allow clients to update their records. If an attacker can send dynamic updates, they can rewrite your DNS records.

Photo by Gaetan Lee <http://www.flickr.com/photos/gaetanlee/2757513472/>

Creative Commons Attribution Generic 2.0 license



Notes:

Example from a recent VA:

Large, very popular local ISP:

3 public DNS servers. All allow recursive queries to the entire world.

Same servers also host their client's authoritative records. (oops)

Software was an older version of BIND

Engineers say they cannot change this for legacy reasons (too many clients and ex-clients using the servers as forwarders)

Using your ISP's DNS



Use your own cache & resolver

Point directly to the root name servers

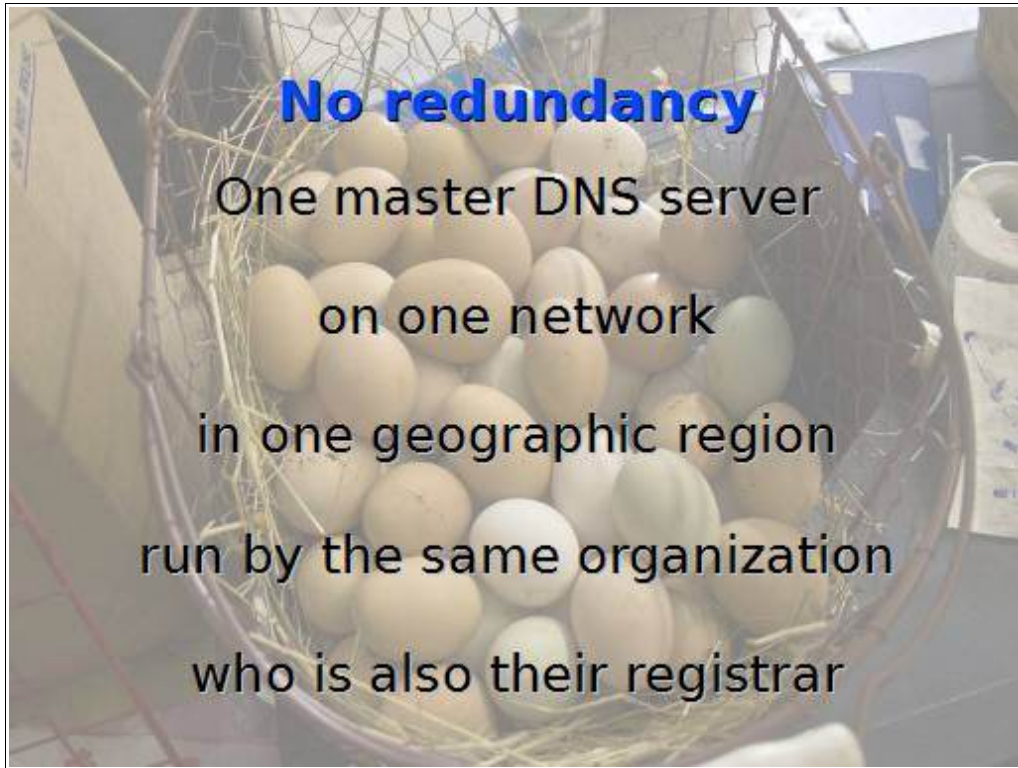
Notes:

Unless you are certain otherwise, it is best to NOT use your provider's DNS as a forwarder.

Install your own caching resolver and send queries direct to the root name servers and authoritative servers for each domain.

“Unbound”, mentioned later, is excellent for this.

Photo by by Mykl Roventine <http://www.flickr.com/photos/myklroventine/1430139705/>
Creative Commons Attribution Generic 2.0 license



No redundancy

One master DNS server
on one network
in one geographic region
run by the same organization
who is also their registrar

Notes:

Don't put all your eggs in one basket!

A good design will have:

- A master DNS and at least two secondaries
- Each secondary on separate networks run by different organizations, physically hosted in different parts of the world.
- Your DNS registrar, DNS host provider, and your network/hosting provider should all be different organizations.
- Not just to survive network outages: Combining them gives too much power to the provider.

Photo by woodleywonderworks <http://www.flickr.com/photos/wwworks/2623295415/>

Creative Commons Attribution Generic 2.0 license



Notes:

It's a big world. Spread your DNS servers across geographic regions and organizations.

- If you have DNS servers in Ottawa, maybe also put some in
- West coast
- Europe
- Asia
- Australia

There are many free and pay secondary DNS providers (e.g. Everydns, twisted4life etc.)

NEVER allow your provider to also be your DNS host and domain registrar!

Map photo by Norman B. Leventhal Map Center at the BPL

<http://www.flickr.com/photos/normanbleventhalmapcenter/2710792122/>

Creative Commons Attribution Generic 2.0 license

Pushpin photo by Kalan <http://commons.wikimedia.org/wiki/File:Thumbtack.png>

GNU Free Documentation License, Version 1.2



Notes:

How many people use their DNS servers as the only location of their DNS data?

Do you back up your DNS files, or just rely on secondaries to provide your “backup”?

How about change control? Keep a record of who changed what when and why?

How often do you review your DNS? Without separate notes, how can you determine the validity of each DNS record?

Photo by hotblack <http://www.morguefile.com/archive/display/189364>

Morguefile free photo license



Notes:

Use a caching resolver, Unix-based secondary or DNS proxy (many firewalls have DNS proxies) to protect Active Directory and other directory services that are not designed for Internet use.

Amazing how many are still running BIND 4 on the Internet (“the one true BIND”)

Amazing how many never update their DNS server software after deployment.

Misconfigured software: open queries, no access control, but also revealing version numbers

Not configuring / verifying the server is emitting random transaction Ids and random source ports To check: visit

<https://www.dns-oarc.net/oarc/services/dnsentropy>

Photo by perenijsje <http://www.flickr.com/photos/26365331@N03/2486768279/>

Creative Commons Attribution Generic 2.0 license



Notes:

- Many examples of domain registrars or DNS providers yanking domains without due process due to Unsubstantiated claim from a copyright holder or law enforcement e.g. <http://nodaddy.com/>
- Most DNS providers have a web interface for client updates. Oh no... web application security!
- But even with good software, users often pick weak easy to guess passwords that allow attackers to brute force.
- Historically, many DNS providers are only too happy to transfer domain ownership or transfer to another registrar. E.g. requiring only a faxed request on company letterhead as authentication.
- Example: 2005 domain of Panix.com a New York ISP hijacked <http://www.internetnews.com/xSP/article.php/3460871>
- Example: sex.com ownership hijacked in 1995 using a forged fax document <http://en.wikipedia.org/wiki/Sex.com>
- Most top level domains and domain registrars support domain locking which prevents transfer to another registrar or domain deletion <http://en.wikipedia.org/wiki/Registrar-Lock>.

Photo by Deborah Fitchett <http://www.flickr.com/photos/deborahfitchett/2970373235/>
Creative Commons Attribution Generic 2.0 license

Forgetting to renew

- Jan 2010: South African Airlines forgets to renew flysaa.com. Millions in sales lost.
- 2008: Turner forgets to renew cnnpolitics.com (in an election year).
- 2007: Google forgets to renew google.de:
 - German Google properties unreachable
- 2003: Microsoft forgets to renew hotmail.co.uk
- 1999: Microsoft forgets to renew passport.com:
 - Hotmail and MSN unreachable
 - Domain renewed by a *Slashdot* reader



Notes:

It sounds obvious, but serious issues have resulted from organizations simply forgetting to renew their domains.

Once lapsed, a squatter can buy it. Unless name is a trademark, you have few rights to get it back.

Passport.com is (was?) microsoft's attempt at central authentication for all their services: hotmail, MSN etc.

Domain fee of \$35 graciously renewed by Slashdot reader Michael Chaney who donated the domain ownership back to Microsoft <http://www.doublewide.net/>

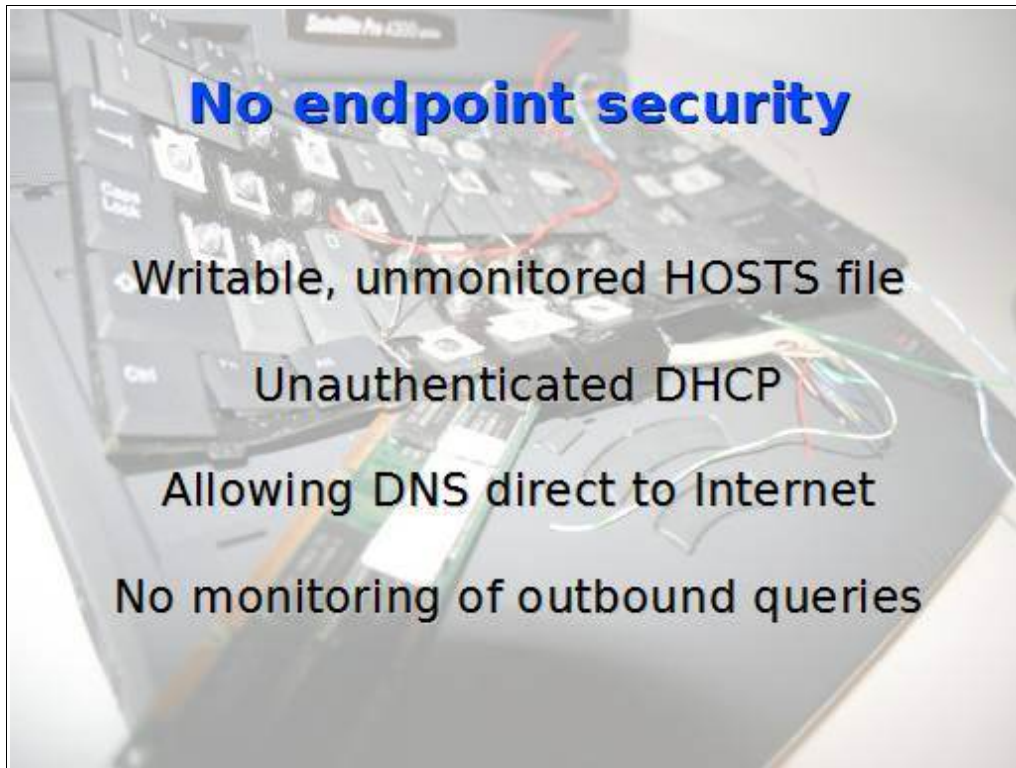
It happened to them, it can happen to you.

http://www.theregister.co.uk/2009/04/22/msn_hijacking/

<http://blogs.zdnet.com/security/?p=1356>

Photo by Fabio Bruna http://www.flickr.com/photos/_fabio/326363546/

Creative Commons Attribution Generic 2.0 license



Notes:

Changing DNS settings on endpoints is a favorite trick of malware:

Trick one:

- Hosts file: common to Windows, Unix/Linux
- Records are looked up there before DNS consulted
- Inserting false records in the HOSTS file bypasses DNS

Trick two:

- Changing the DNS server settings on the endpoint to point to the attackers own DNS server (either changing registry, resolv.conf or via DHCP)
- Easy safeguard: do not allow outbound DNS from internal network except to your own DNS forwarders (block port 53 TCP and UDP)

Unmonitored queries:

- Workstations compromised in the “Aurora” attack in January 2010 against Google and others made large number of queries to dynamic DNS names. Many bots do the same.
- Also monitor rates of failed queries. This is one good indicator of presence of malware.

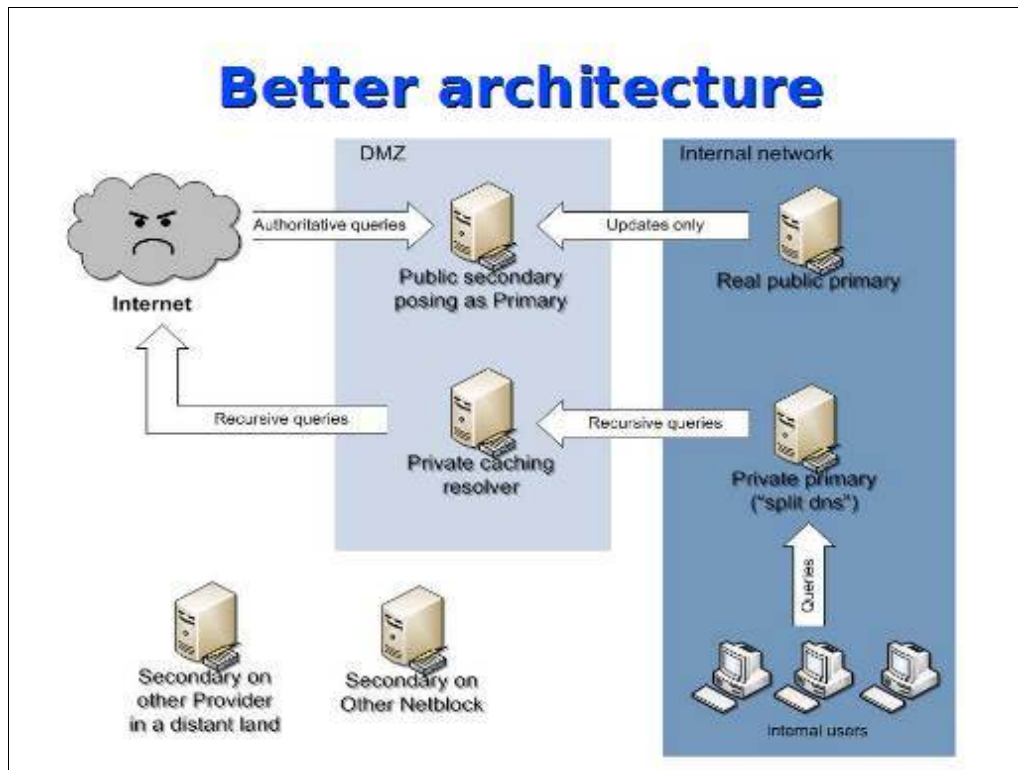
Photo by S Baker <http://www.flickr.com/photos/sarahbaker/280931618/>

Creative Commons Attribution Generic 2.0 license

Summary

- Don't mix DNS functions on one server.
- Implement access controls & monitoring.
- Avoid your provider's DNS
- Have Redundancy.
- Have Redundancy.
- Configure and update your DNS software
- Remember your renewal date.
- Harden your endpoints.

Notes:



Notes:

In case anyone is wondering what a sound DNS architecture looks like, here is one example.

- Top right: Real external primary DNS holding records for external authoritative domains. Inside firewall to protect from attack.
- Top middle: A public secondary published in whois records as the primary nameserver for the domains. If compromised, can be easily rebuild and reloaded from real master. Non-recursive. Only answers queries for domains it is authoritative for.
- Bottom left & middle: secondaries on other subnets and other geographic locations. Should also be located on networks run by different organizations, differing upstreams etc.
- Bottom right: Internal hosts query internal DNS domains and records from dedicated internal primary DNS. Queries for other domains go to a private caching resolver in a DMZ. Internal primary can use fake master as on Internet if you're concerned about internal attack.
- Firewall blocks outbound DNS queries from internal hosts.
- How much of this you actually implement depends on your risk tolerance.

Unbound

- Validating, caching recursive resolver
- Developed by Verisign, NLnet Labs, Nominet, Kirei, and EP.net
- Lightweight, easy to configure
- Many sanity checks on replies
- Supports DNSSEC and “views”
- Use it at the perimeter or on individual servers
- www.unbound.net



Notes:

Sanity checks:

For example: set a minimum time to live on DNS records to avoid fast flux DNS records favored by phishing web sites.

Views:

Override replies for specific queries (eg. To redirect internal users to your internal web site rather than public one when they query www.yourcompany.com).

What about DNSSec?

- DNS SEC adds cryptographic signatures to DNS replies to eliminate spoofing.
- Integrity and authentication only: does *not* encrypt data.
- Developed over 10 years ago by "a committee of experts."
- Crypto ≠ Security.
- DNSSec adds millions of additional lines of complex code to DNS server implementations.
- BIND uses the buggy OpenSSL crypto library.
- Will prevent spoofing, but expect many years of serious DNS software vulnerabilities and configuration errors.

Notes:

DNSSEC is the official solution to cache poisoning and spoofing.

It adds on-the-fly generated crypto signatures to DNS replies

Does not encrypt data: only provides integrity and authentication

Too many people to this day still equate encryption with security

Crypto code is complex. As we know from history, the more complex a thing is, the more problems it has, in security, functionality, managability, and so on

DNSSec is implemented in BIND (the most widely used DNS server software) using OpenSSL which has a poor security history.

The DNS root will have a DNSSEC signature added soon (this year?), The US Gov't ordered all public US gov't DNS servers to implement DNSSEC by this year (now 2011), and the .org and other TLDs are now signed.

DNSSEC eventually will make cache poisoning nearly impossible, but with the complexity of the code and configuration, we can probably expect many years of serious exploits before overall DNS security improves.



Notes:

Photo by -bast- <http://www.flickr.com/photos/-bast-/349497988/>
Creative Commons Attribution Generic 2.0 license

Resources

US CERT DNS cache poisoning alert
<http://www.us-cert.gov/cas/techalerts/TA08-190B.html>

"It's the end of the cache as we know it" (Dan Kaminsky)
http://www.doxpara.com/DMK_BO2K8.ppt

NIST SP-800 81 "DNS Deployment Guide"
<http://csrc.nist.gov/publications/PubsSPs.html>

DNS Server Randomness Test
<https://www.dns-oarc.net/oarc/services/dnsentropy>

RFC 3833 Threat Analysis of the Domain Name System
<http://www.rfc-archive.org/getrfc.php?rfc=3833>

Evilgrade (software update emulator)
<http://www.infobyte.com.ar/down/isr-evilgrade-Readme.txt>

Unbound, a validating, recursive and caching DNS resolver
<http://www.unbound.net/>

List of DNS testing and developer tools
<http://www.bind9.net/dns-tools>

Team Cymru bogon list
<http://www.cymru.com/Documents/bogon-list.html>

US CERT DNS cache poisoning alert

<http://www.us-cert.gov/cas/techalerts/TA08-190B.html>

"It's the end of the cache as we know it" (Dan Kaminsky)

http://www.doxpara.com/DMK_BO2K8.ppt

NIST SP-800 81 "DNS Deployment Guide"

<http://csrc.nist.gov/publications/PubsSPs.html>

DNS Server Randomness Test

<https://www.dns-oarc.net/oarc/services/dnsentropy>

RFC 3833 Threat Analysis of the Domain Name System

<http://www.rfc-archive.org/getrfc.php?rfc=3833>

Evilgrade (software update emulator)

<http://www.infobyte.com.ar/down/isr-evilgrade-Readme.txt>

Unbound: a validating, recursive and caching DNS resolver

<http://www.unbound.net/>

Good list of DNS testing and developer tools

<http://www.bind9.net/dns-tools>

Team Cymru bogon list

<http://www.cymru.com/Documents/bogon-list.html>