Introduction to DNS and its vulnarabilities

Olaf M. Kolkman
olaf@nlnetlabs.nl

DNS has a distributed nature

- Authoritative servers all provide part of the name space
- User devices query a local server that maintains a cache
    - For better performance
    - For scalability of the system as a whole

Terminology

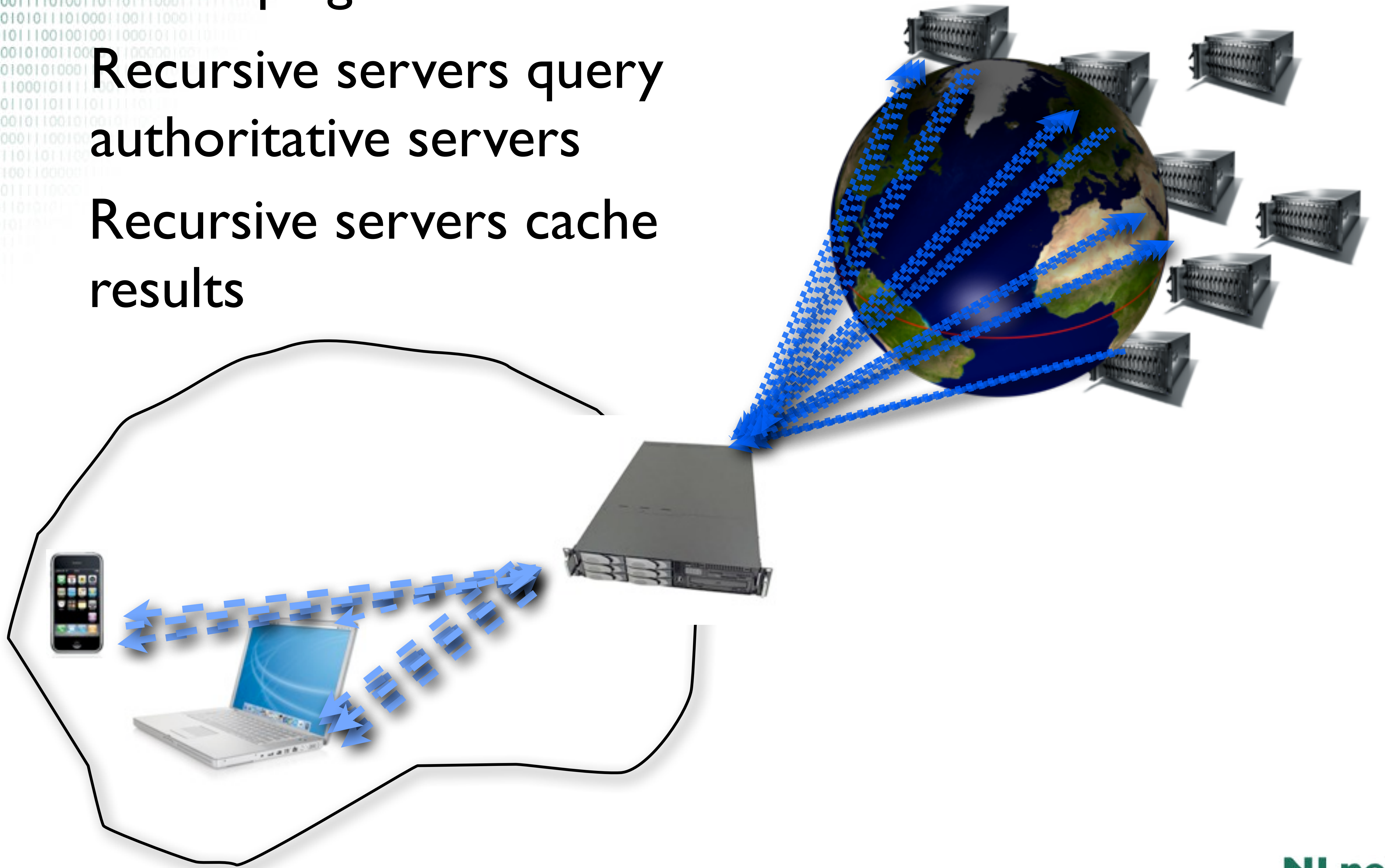
- Authoritative Nameserver: Maintains an authoritative copy of the data.
- Recursive Nameserver: Contacts Authoritative servers to compose an answer for stub resolvers. Also called Caching Nameserver or Cache
- Stub Resolver: fires off queries to pre-configured addresses and expects an answer. Usually implemented in OS library
  - gethostbyname()

Animation

Look up against recursive servers

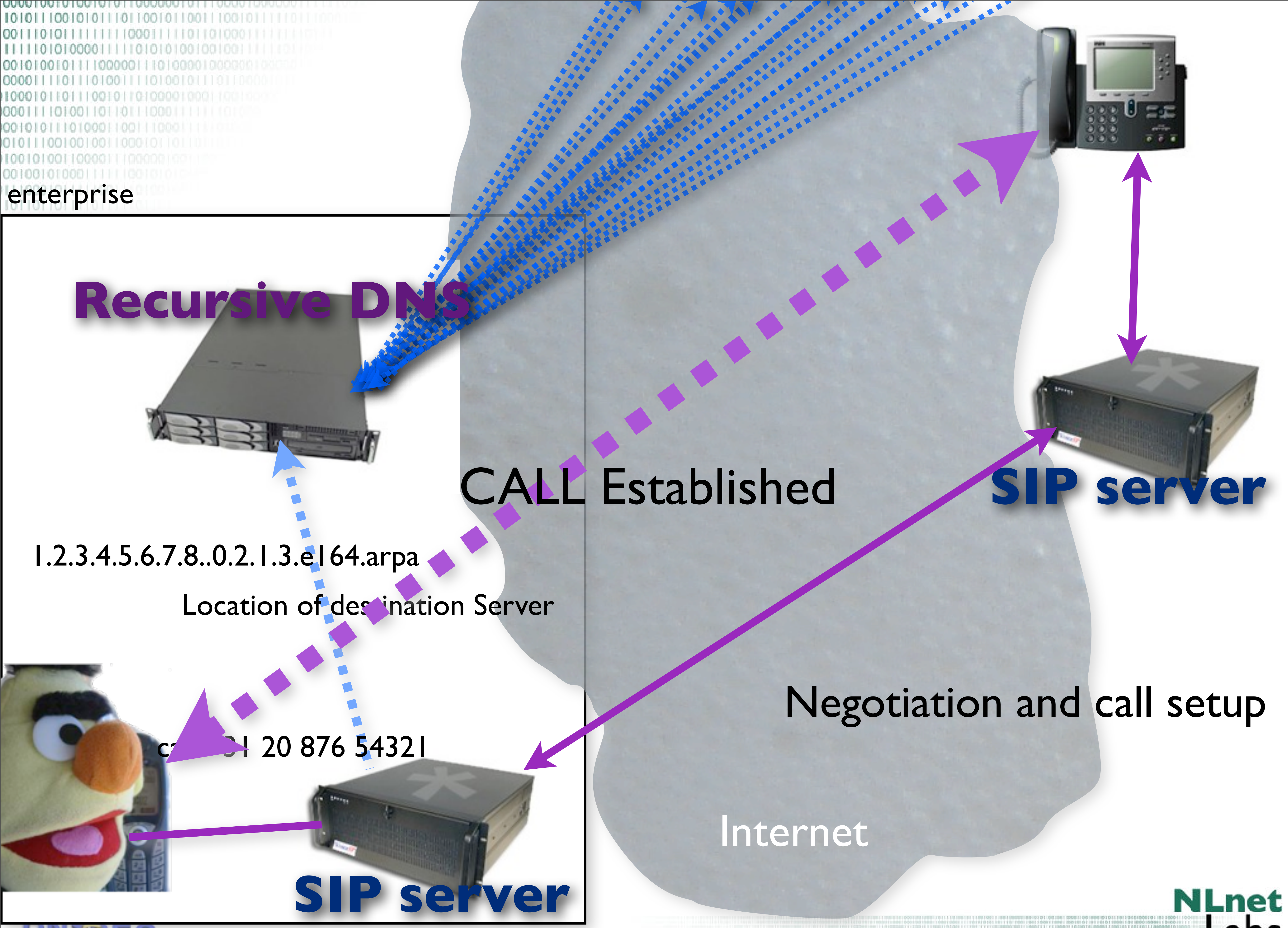
Recursive servers query authoritative servers

Recursive servers cache results



When do you use the DNS

- Anytime that you need to know where the other guy is
- DNS is the phone book of the Internet
- So it is used when people make a voice over IP call



enterprise

Recursive DNS

CALL Established

SIP server

1.2.3.4.5.6.7.8..0.2.1.3.e164.arpa

Location of destination Server

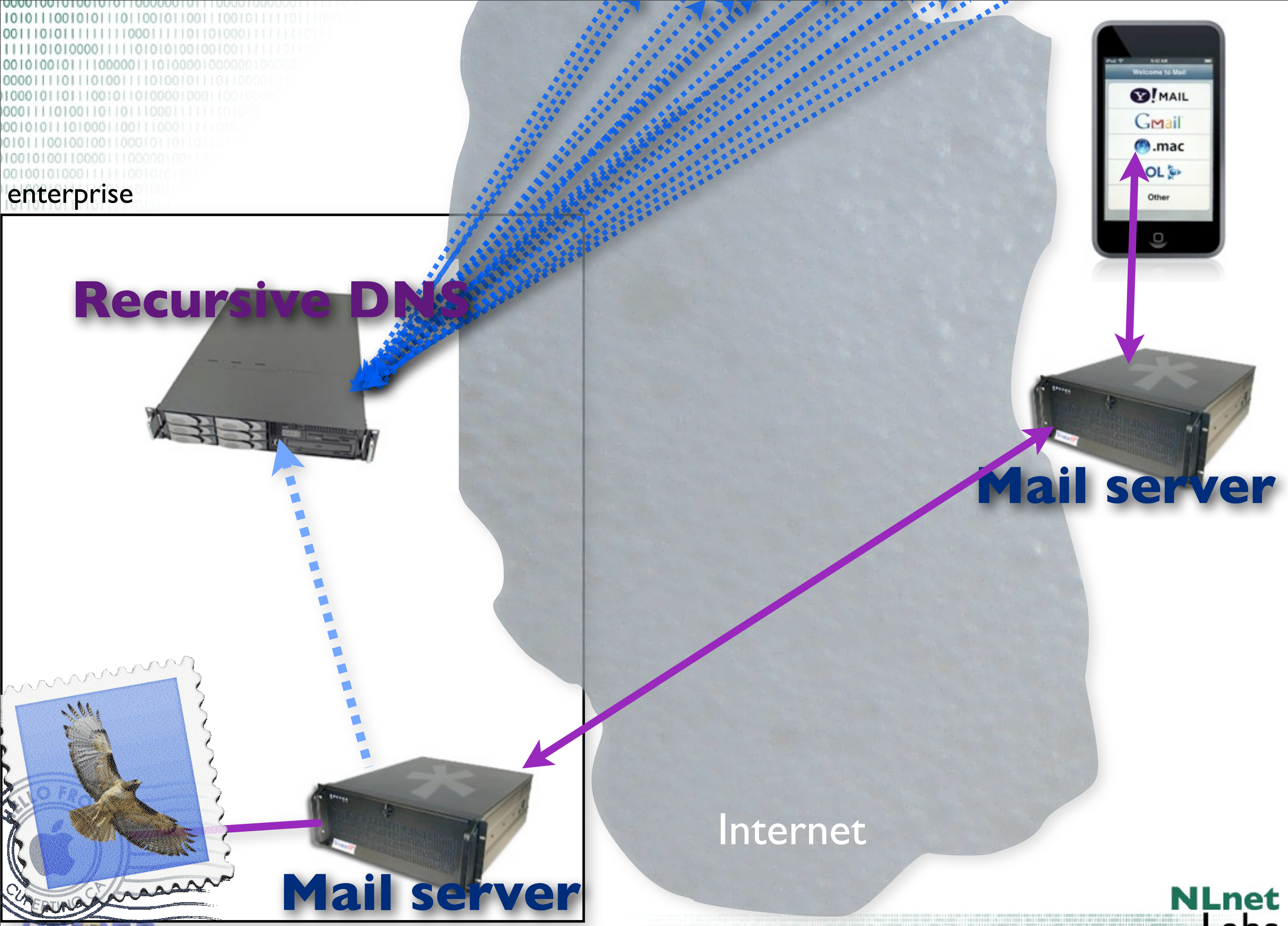
Negotiation and call setup

ca 31 20 876 54321

Internet

SIP server

Or they use the DNS when sending MAIL



enterprise

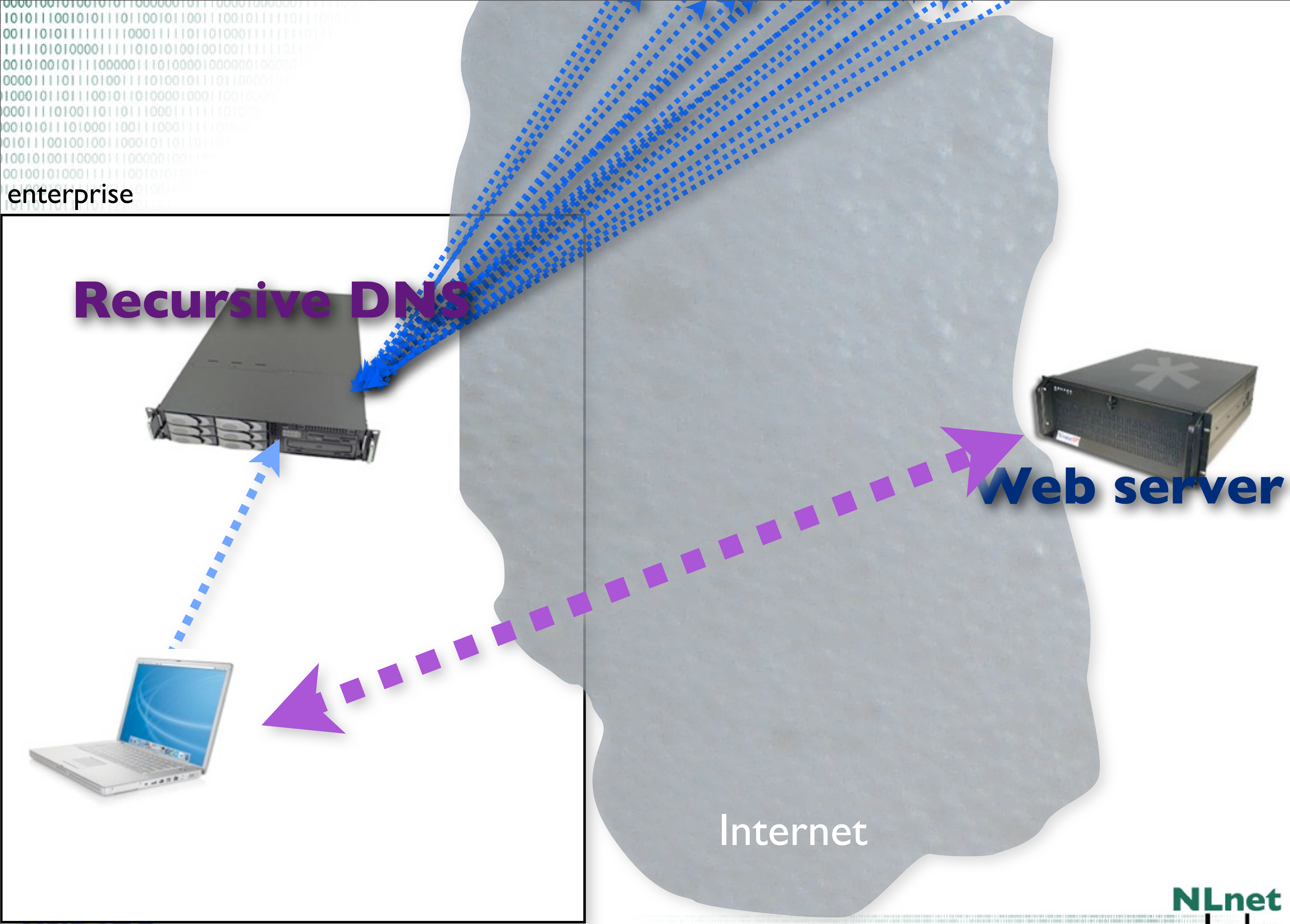
Recursive DNS

Mail server

Mail server

Internet

Or they use the DNS when browsing the Web


enterprise
Recursive DNS
Web server
Internet

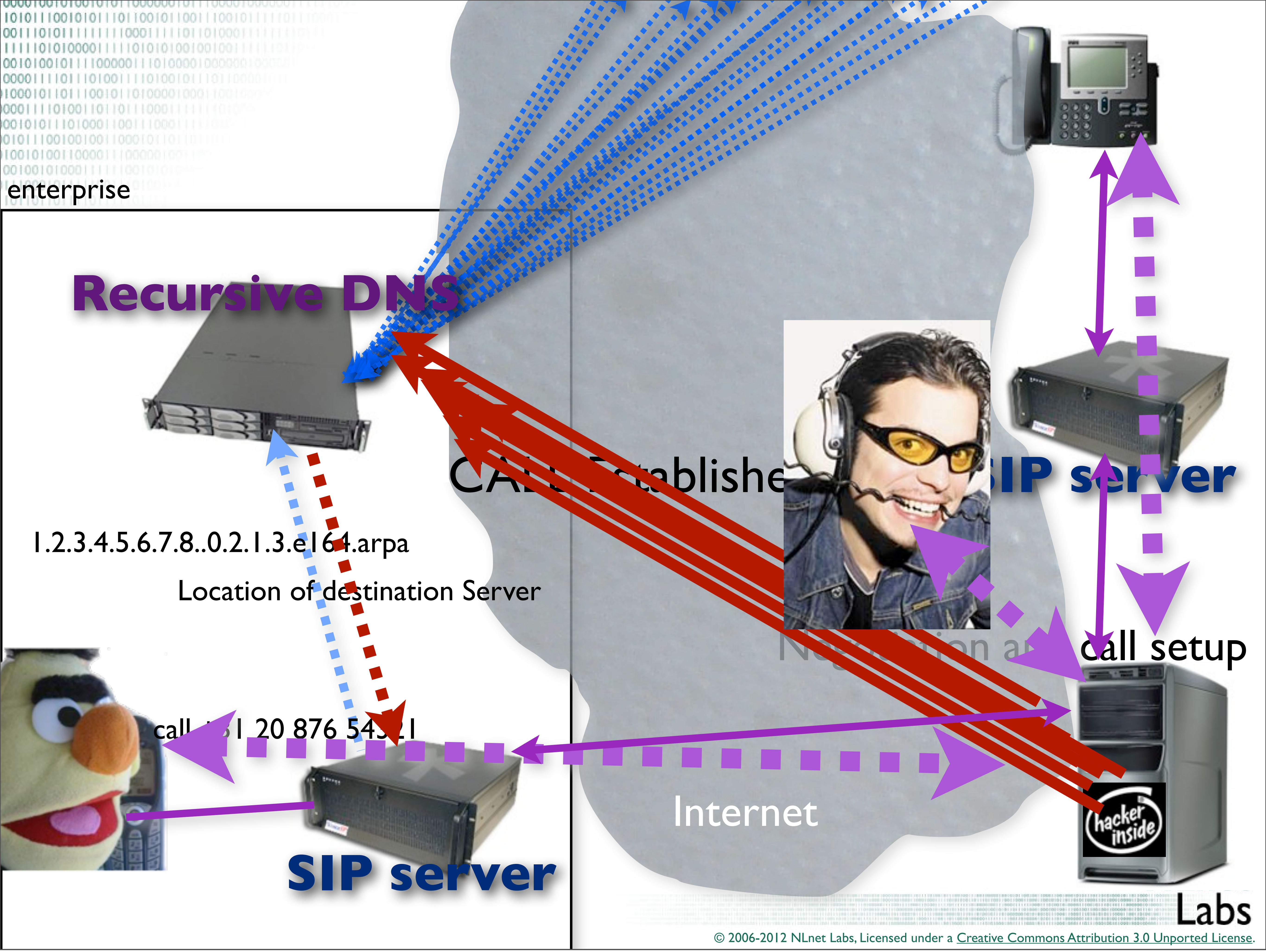
Or they use the DNS

- When downloading Software upgrades
- Sharing their agenda
- Uploading tax forms
- Instant messaging with friends
- Connect to their security camera
- Figure out the latest news about that merger

So DNS is IMPORTANT

- How would an attacker use the DNS for attacks?
- By fooling the receiver that a service lies elsewhere

Back to our VOIP example



enterprise

Recursive DNS

CALL Establishe

SIP server

1.2.3.4.5.6.7.8..0.2.1.3.e164.arpa

Location of destination Server

call +31 20 876 54321

Negotiation a

call setup

Internet

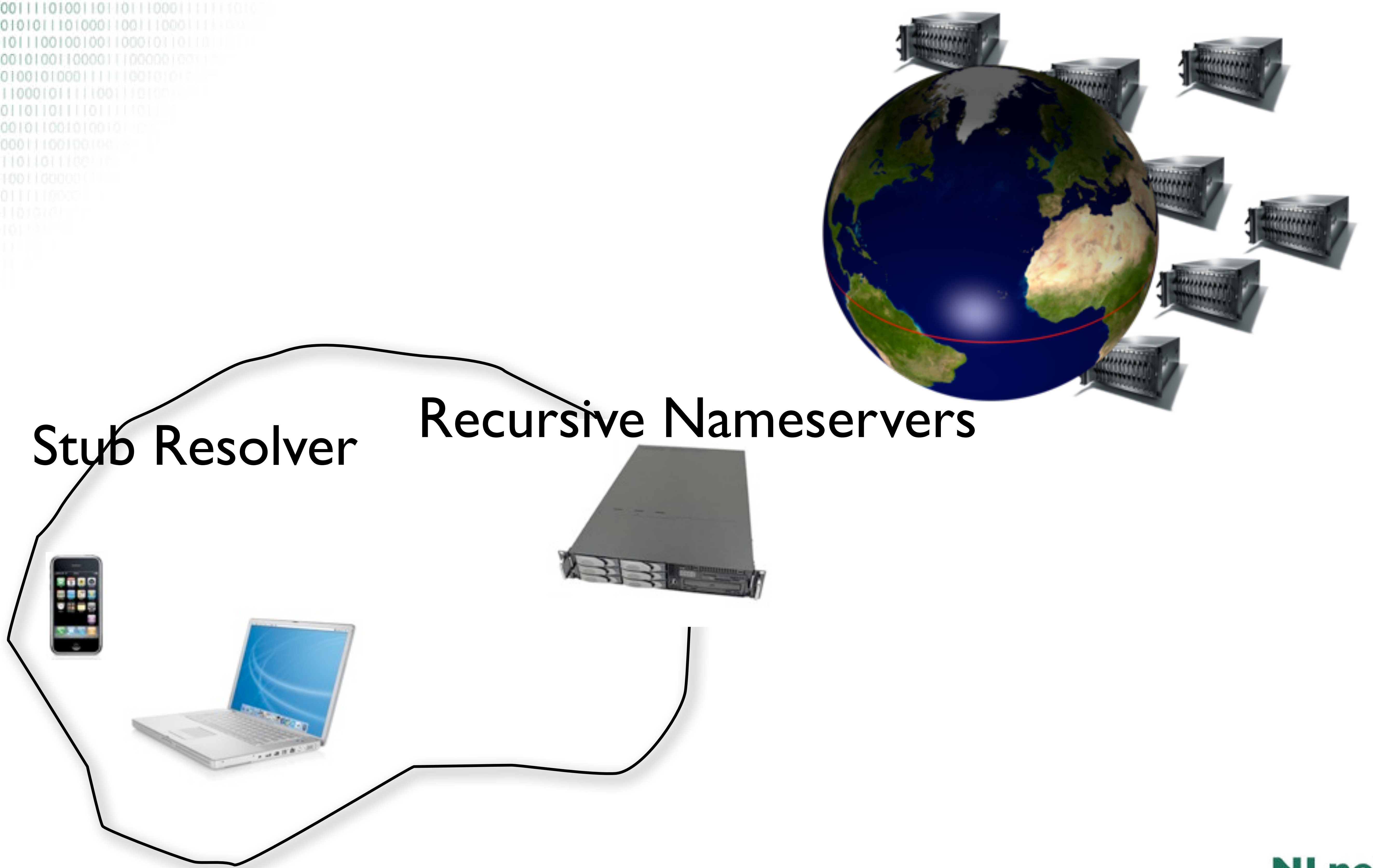
SIP server

Labs

Cache Poisoning

- The attack you just saw is called cache poisoning
- Inserting false data into the cache of recursive name servers
- This form of attack has been known for years
- One of the reasons to work on DNSSEC

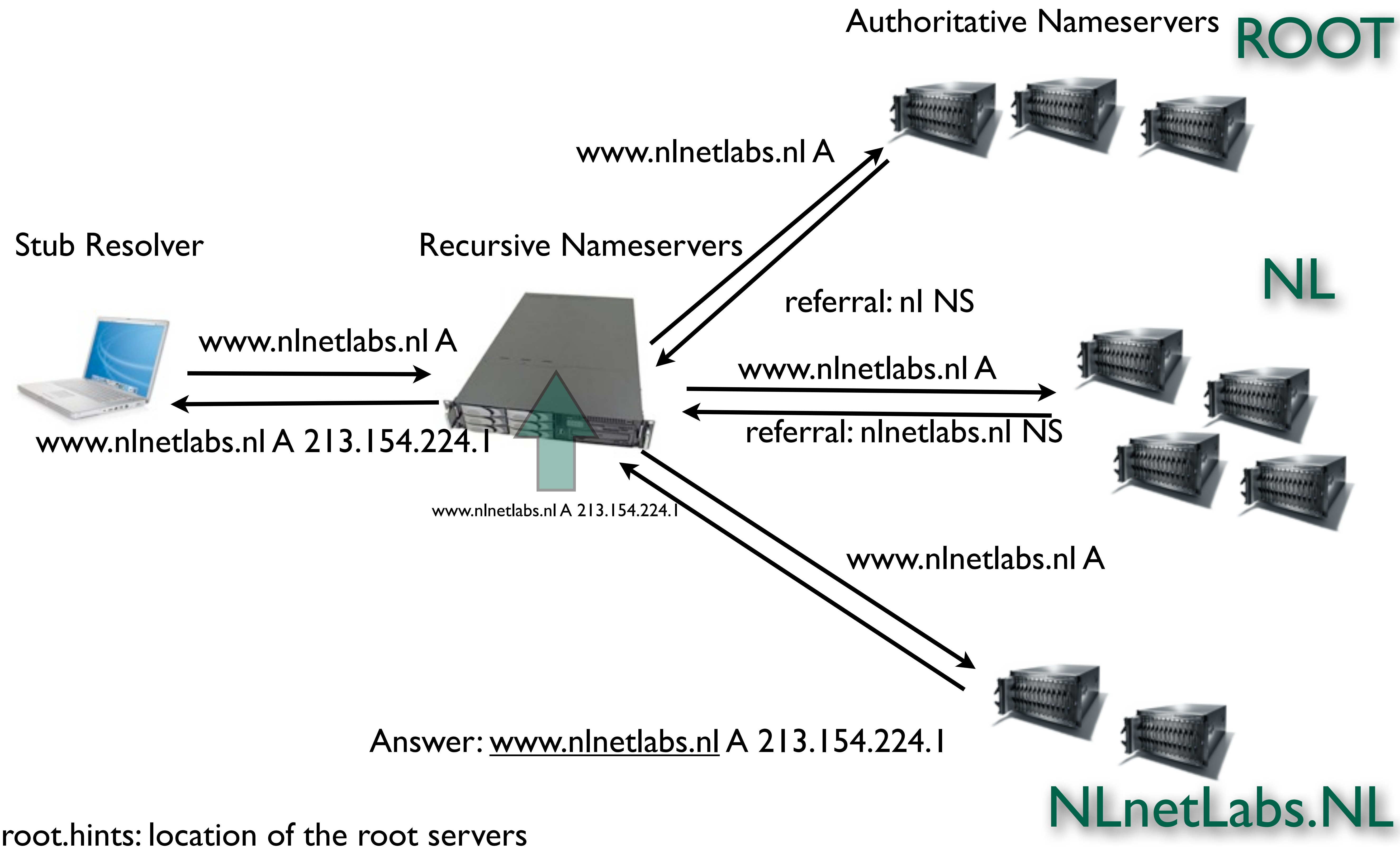
DNS Architecture and Protocol



Authoritative Nameservers

Recursive Nameservers

Stub Resolver



Authoritative Nameservers

ROOT

Stub Resolver

Recursive Nameservers

www.nlnetlabs.nl A

www.nlnetlabs.nl A 213.154.224.1

www.nlnetlabs.nl A

referral: nl NS

NL

www.nlnetlabs.nl A

referral: nlnetlabs.nl NS

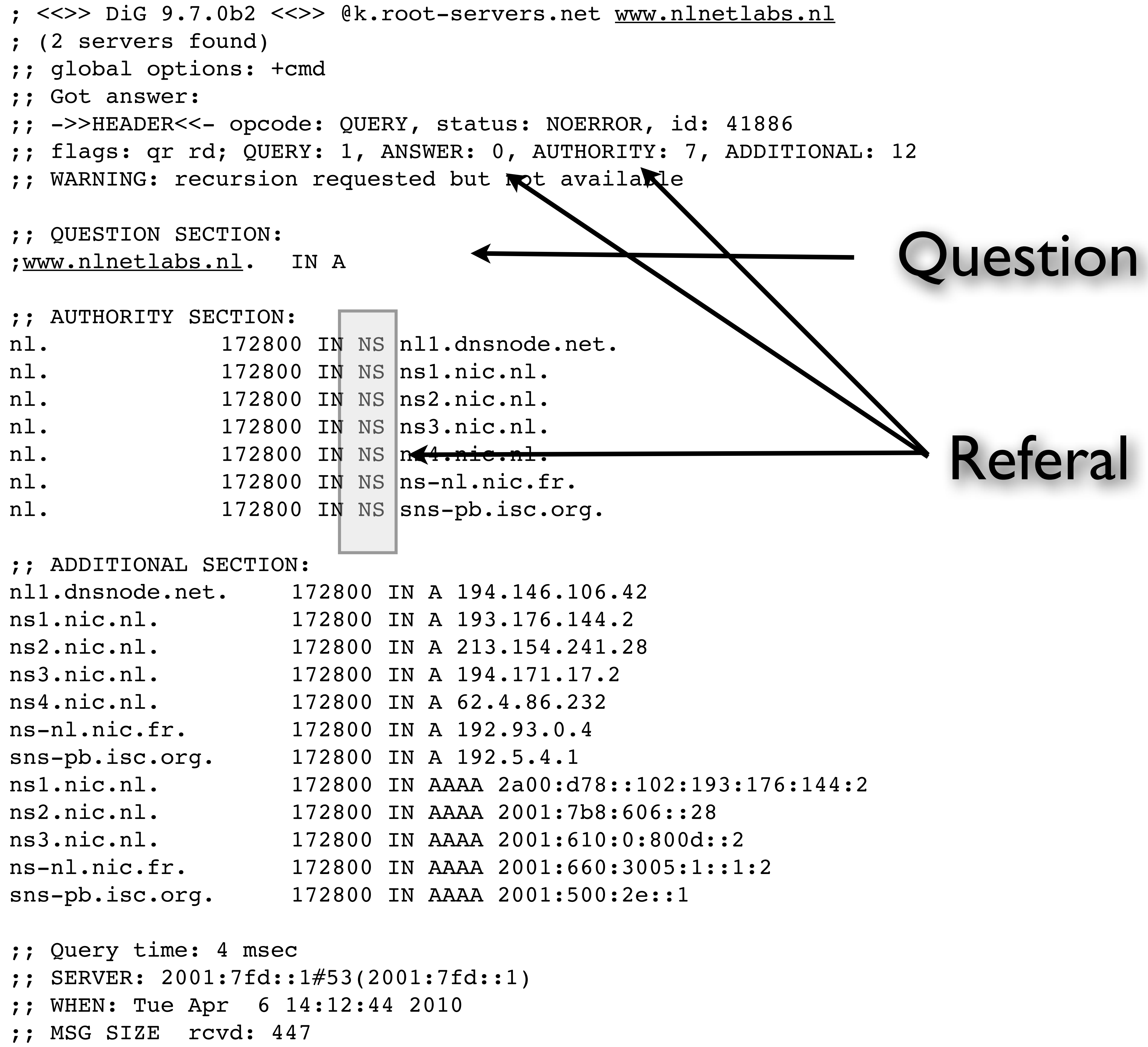
www.nlnetlabs.nl A 213.154.224.1

www.nlnetlabs.nl A

Answer: www.nlnetlabs.nl A 213.154.224.1

NLnetLabs.NL

root.hints: location of the root servers



```
; <<>> DiG 9.7.0b2 <<>> @k.root-servers.net www.nlnetlabs.nl
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41886
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 7, ADDITIONAL: 12
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.nlnetlabs.nl.   IN A

;; AUTHORITY SECTION:
nl.             172800 IN NS nl1.dnsnode.net.
nl.             172800 IN NS ns1.nic.nl.
nl.             172800 IN NS ns2.nic.nl.
nl.             172800 IN NS ns3.nic.nl.
nl.             172800 IN NS ns4.nic.nl.
nl.             172800 IN NS ns-nl.nic.fr.
nl.             172800 IN NS sns-pb.isc.org.

;; ADDITIONAL SECTION:
nl1.dnsnode.net.     172800 IN A 194.146.106.42
ns1.nic.nl.          172800 IN A 193.176.144.2
ns2.nic.nl.          172800 IN A 213.154.241.28
ns3.nic.nl.          172800 IN A 194.171.17.2
ns4.nic.nl.          172800 IN A 62.4.86.232
ns-nl.nic.fr.        172800 IN A 192.93.0.4
sns-pb.isc.org.      172800 IN A 192.5.4.1
ns1.nic.nl.          172800 IN AAAA 2a00:d78::102:193:176:144:2
ns2.nic.nl.          172800 IN AAAA 2001:7b8:606::28
ns3.nic.nl.          172800 IN AAAA 2001:610:0:800d::2
ns-nl.nic.fr.        172800 IN AAAA 2001:660:3005:1::1:2
sns-pb.isc.org.      172800 IN AAAA 2001:500:2e::1

;; Query time: 4 msec
;; SERVER: 2001:7fd::1#53(2001:7fd::1)
;; WHEN: Tue Apr  6 14:12:44 2010
;; MSG SIZE  rcvd: 447
```

Question

Referal

Cache and TTL

```
;; ANSWER SECTION:
www.nlnetlabs.nl.   10200 IN A 213.154.224.1
```

- TTL is a parameter that indicates how long data is to remain in a cache
- TTL value is set by the zone owner
- TTL decreases while in the cache

Back to Cache Poisoning

Cache Poison

- Attack is based on 'predicting' properties
  - e.g. when asking a question to a female you expect a female voice to answer
- If you ask a question with a specific QID you expect that QID in the answer
  - Cache poisoner will take a wild guess

```
; <<>> DiG 9.7.0b2 <<>> @k.root-servers.net www.nlnetlabs.nl
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41886
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 7, ADDITIONAL: 12
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.nlnetlabs.nl.   IN A

;; AUTHORITY SECTION:
nl.             172800 IN NS nl1.dnsnode.net.
nl.             172800 IN NS ns1.nic.nl.
nl.             172800 IN NS ns2.nic.nl.
nl.             172800 IN NS ns3.nic.nl.
nl.             172800 IN NS ns4.nic.nl.
nl.             172800 IN NS ns-nl.nic.fr.
nl.             172800 IN NS sns-pb.isc.org.

;; ADDITIONAL SECTION:
nl1.dnsnode.net.    172800 IN A 194.146.106.42
ns1.nic.nl.         172800 IN A 193.176.144.2
ns2.nic.nl.         172800 IN A 213.154.241.28
ns3.nic.nl.         172800 IN A 194.171.17.2
ns4.nic.nl.         172800 IN A 62.4.86.232
ns-nl.nic.fr.       172800 IN A 192.93.0.4
sns-pb.isc.org.     172800 IN A 192.5.4.1
ns1.nic.nl.         172800 IN AAAA 2a00:d78::102:193:176:144:2
ns2.nic.nl.         172800 IN AAAA 2001:7b8:606::28
ns3.nic.nl.         172800 IN AAAA 2001:610:0:800d::2
ns-nl.nic.fr.       172800 IN AAAA 2001:660:3005:1::1:2
sns-pb.isc.org.     172800 IN AAAA 2001:500:2e::1

;; Query time: 4 msec
;; SERVER: 2001:7fd::1#53(2001:7fd::1)
;; WHEN: Tue Apr  6 14:12:44 2010
;; MSG SIZE  rcvd: 447
```

Varying properties in a packet

- The sender can vary the following properties for the attacker to match
- DNS:
  - Query ID (16 bits)
- Transport:
  - Fire the question from a random source port (16 bits)

Isn’t Query ID only sufficient?

Chance that *n* people have different birthdays

$$\bar{p}(n) = 1\times\left(1-\frac{1}{365}\right)\times\left(1-\frac{2}{365}\right)\cdots\left(1-\frac{n-1}{365}\right) = \frac{365\times 364\cdots(365-n+1)}{365^n} = \frac{365!}{365^n(365-n)!}$$

Chance that *n* people have the same birthday

$$p(n) = 1 - \bar{p}(n).$$

n	P(n)
10	11.17%
20	41.1%
23	50.7%
30	70.6%
50	97%
57	99%
100	99.99997%

from: http://en.wikipedia.org/wiki/Birthday_problem

Bits	50%	5%	Aka
16	10 s	1 s	Unpatched server, random ID
26	2.8 h	17 m	Patched, using only 1024 ports
34	28 days	2.8 days	unbound with defaults
44	28444 days	2844.4 days	unbound with 0x20 and source addresses configured

50%-5%-0.5%-0.05%

Besides: randomness is non-trivial

- For example: BIND9.4.1 and earlier used a pseudo random number generator that provided predictable sequences
  - Current ID even: next ID one out of 10 possible numbers
  - Only order 15 queries needed to predict rest of the stream
- Discovered by Amit Klein of trusteer

Using all ports, not easy

- Some architectures did not use a sufficiently large range of ports
- The patches issued as response to the so called Kaminsky attack, early 2008, all had to do with increasing the randomness in port use

Still until 2007 folk seemed happy

- Attacker only got one try:
  - Query for www. onlinebank.example
  - Bombard with answers hoping for the the mala-fide answer to get in first
  - Wait for timeout of the TTL
  - Then try again

Kaminsky’s variant

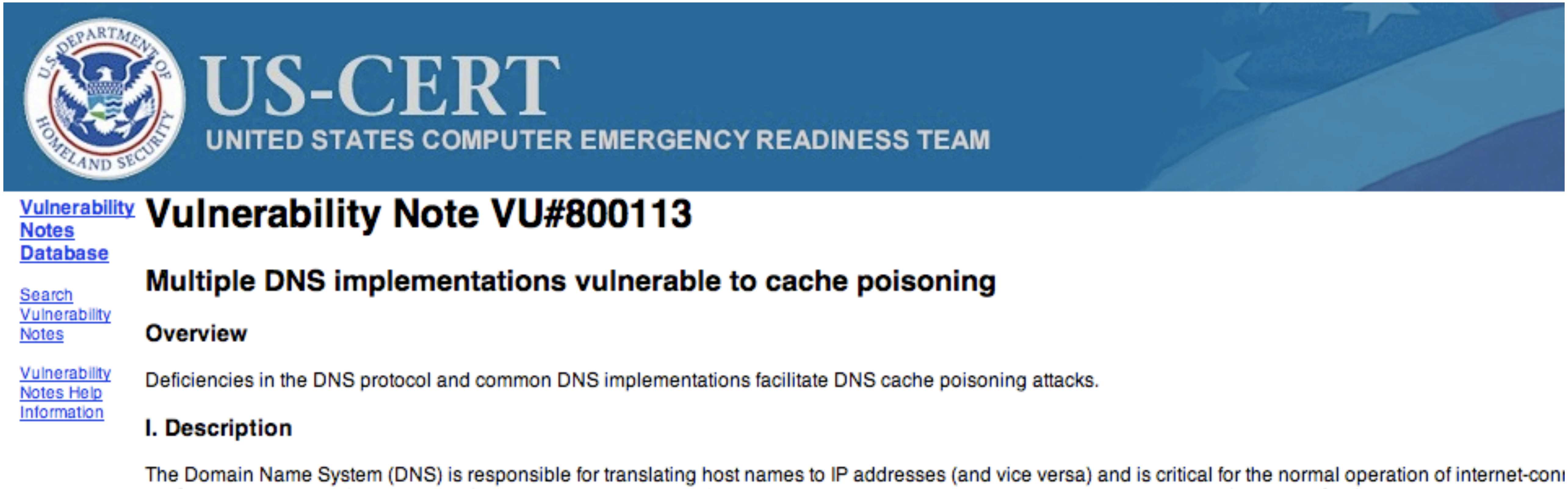
- Classic cache poisoning gave you ‘a few tries’ to get in between the outgoing question and incoming answer
- Kaminsky came with a scheme where the culprit can keep trying
  - Surprisingly simple, a wonder nobody thought of the variety before

And how does it work

- Attacker queries: <randomcruft>.www.importantbank.example
- respond with fake delegation to: www.importantbank.example with glue
- There are other varieties too, but this is the one that has no real workaround

problem?

There is Recognition



US-CERT

UNITED STATES COMPUTER EMERGENCY READINESS TEAM

U.S. DEPARTMENT OF HOMELAND SECURITY
Vulnerability Notes Database
Search Vulnerability Notes
Vulnerability Notes Help Information

Vulnerability Note VU#800113

Multiple DNS implementations vulnerable to cache poisoning

Overview

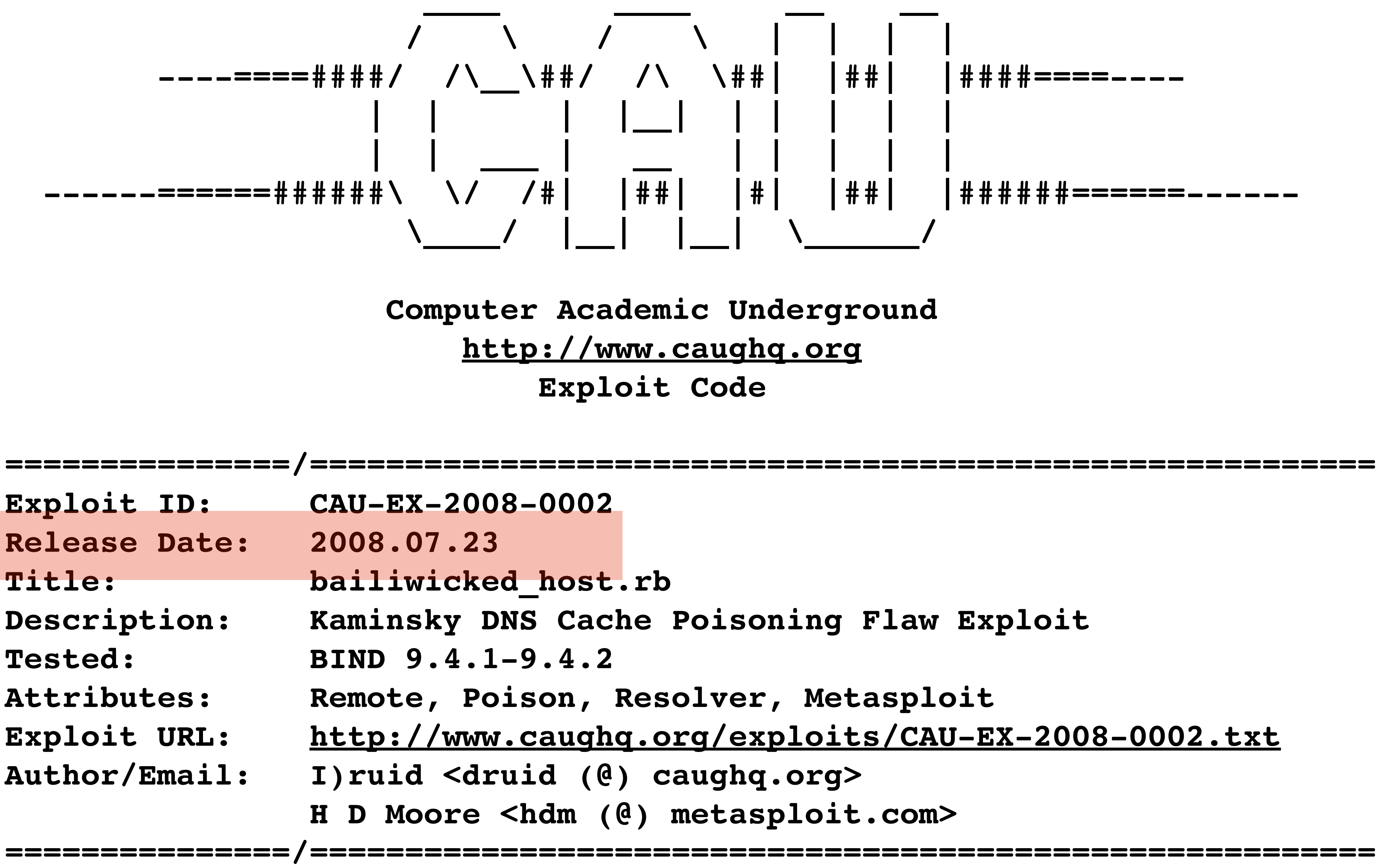
Deficiencies in the DNS protocol and common DNS implementations facilitate DNS cache poisoning attacks.

I. Description

The Domain Name System (DNS) is responsible for translating host names to IP addresses (and vice versa) and is critical for the normal operation of internet-con

http://www.kb.cert.org/vuls/id/800113

There is Exploit Code



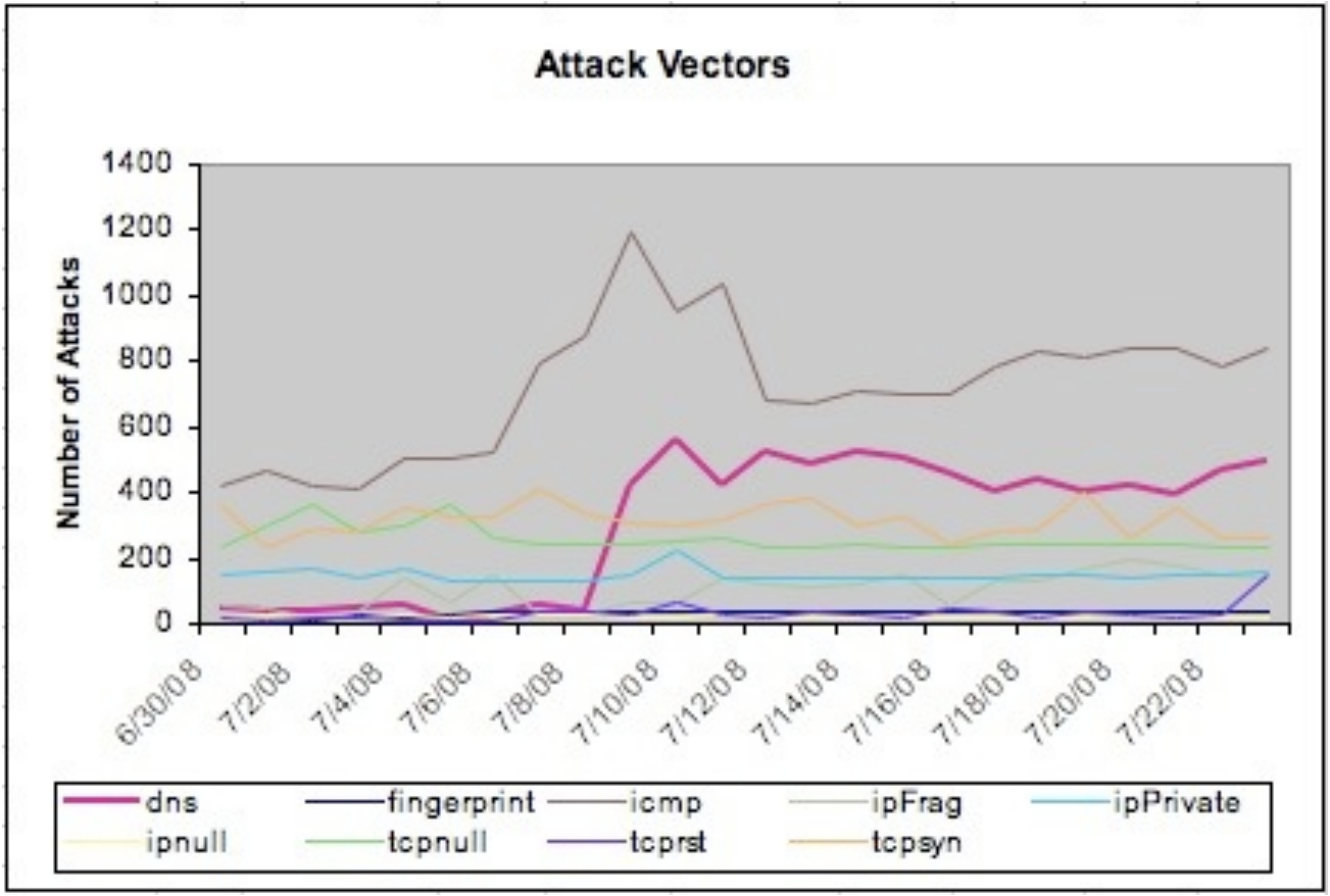
```
                    ____      ____     __    __
                   /    \    /    \   |  |  |  |
      ----====####/  /\__\##/  /\  \##|  |##|  |####====----
                 |  |      |  |__|  | |  |  |  |
                 |  |  ___ |   __   | |  |  |  |
------======######\  \/  /#|  |##|  |#|  |##|  |######=====------
                   \____/  |__|  |__|  \______/

                   Computer Academic Underground
                      http://www.caughq.org
                           Exploit Code

===============/========================================================
Exploit ID:     CAU-EX-2008-0002
Release Date:   2008.07.23
Title:          bailiwicked_host.rb
Description:    Kaminsky DNS Cache Poisoning Flaw Exploit
Tested:         BIND 9.4.1-9.4.2
Attributes:     Remote, Poison, Resolver, Metasploit
Exploit URL:    http://www.caughq.org/exploits/CAU-EX-2008-0002.txt
Author/Email:   I)ruid <druid (@) caughq.org>
                H D Moore <hdm (@) metasploit.com>
===============/========================================================
```

And more exploit code

```
/*
 * 2008+ Copyright (c) Evgeniy Polyakov <johnpol@2ka.mipt.ru>
 * All rights reserved.
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 */
```

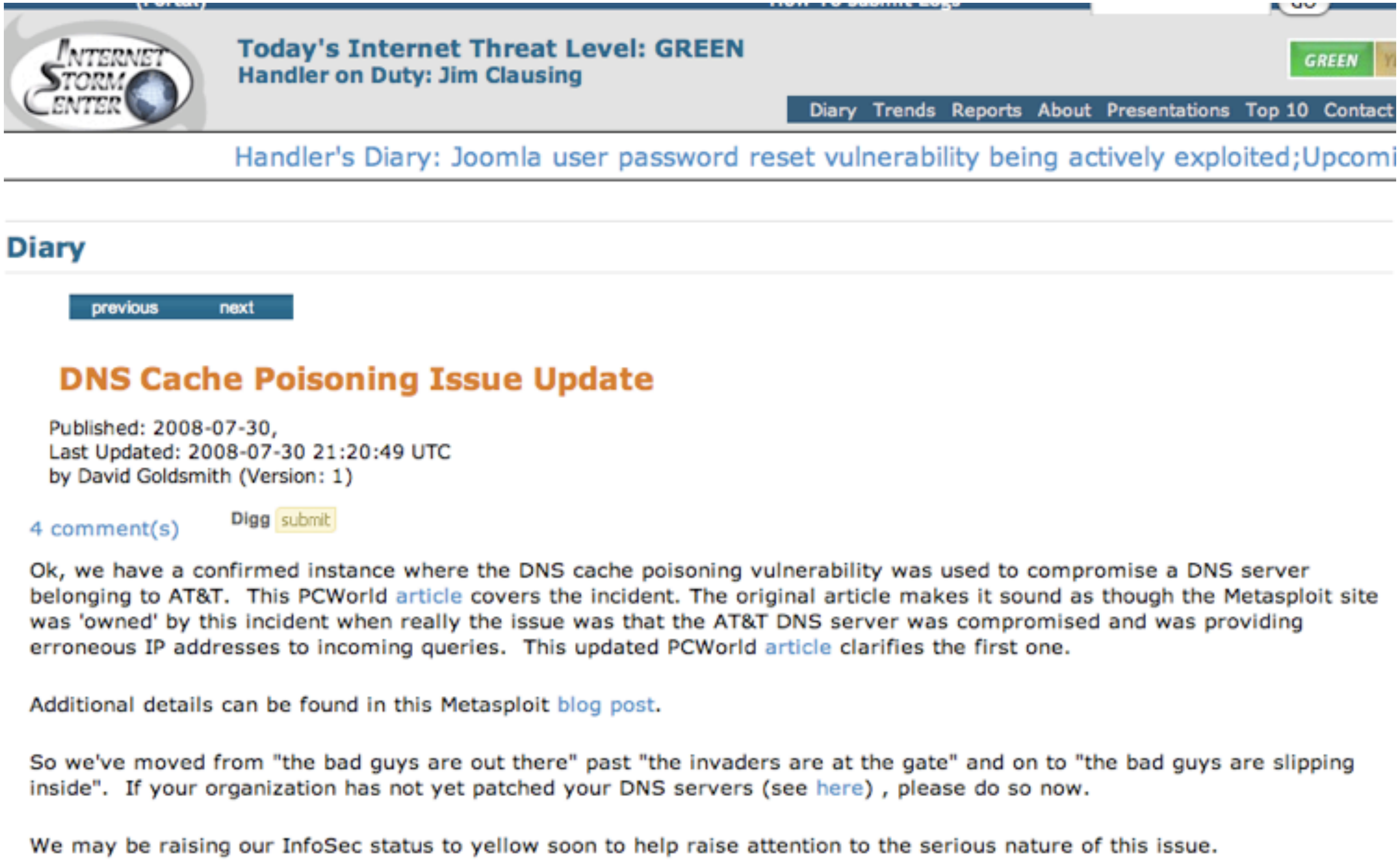
http://tservice.net.ru/~s0mbre/archive/dns/

The networks are scanned



http://asert.arbornetworks.com/2008/07/30-day-of-dns-attack-activity/

There have been succesful attacks



Today's Internet Threat Level: GREEN
Handler on Duty: Jim Clausing

Handler's Diary: Joomla user password reset vulnerability being actively exploited;Upcomi

Diary

DNS Cache Poisoning Issue Update

Published: 2008-07-30,
Last Updated: 2008-07-30 21:20:49 UTC
by David Goldsmith (Version: 1)

4 comment(s)

Ok, we have a confirmed instance where the DNS cache poisoning vulnerability was used to compromise a DNS server belonging to AT&T. This PCWorld article covers the incident. The original article makes it sound as though the Metasploit site was 'owned' by this incident when really the issue was that the AT&T DNS server was compromised and was providing erroneous IP addresses to incoming queries. This updated PCWorld article clarifies the first one.

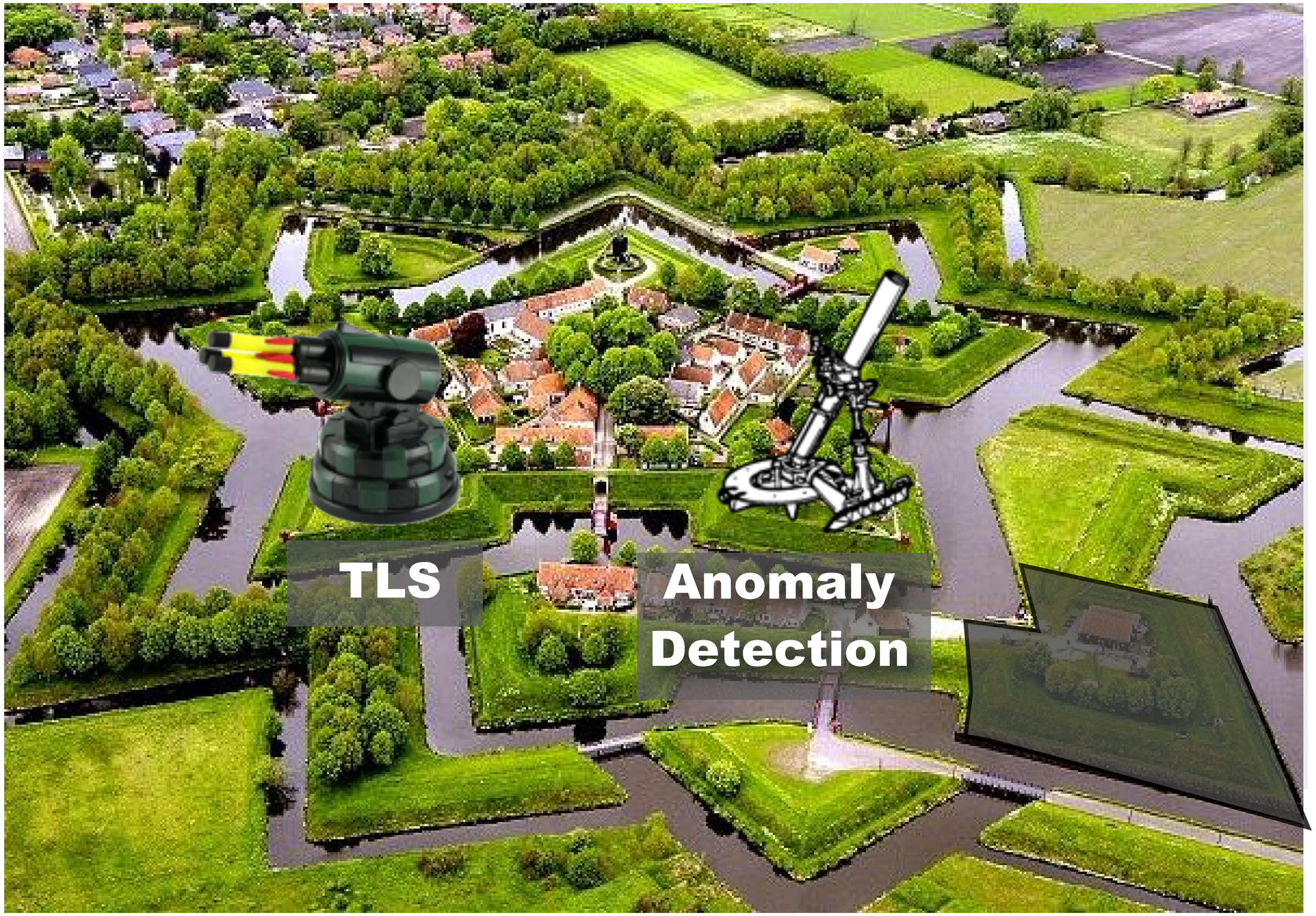
Additional details can be found in this Metasploit blog post.

So we've moved from "the bad guys are out there" past "the invaders are at the gate" and on to "the bad guys are slipping inside". If your organization has not yet patched your DNS servers (see here) , please do so now.

We may be raising our InfoSec status to yellow soon to help raise attention to the serious nature of this issue.

http://isc.sans.org/diary.html?storyid=4801

Yes, Problem

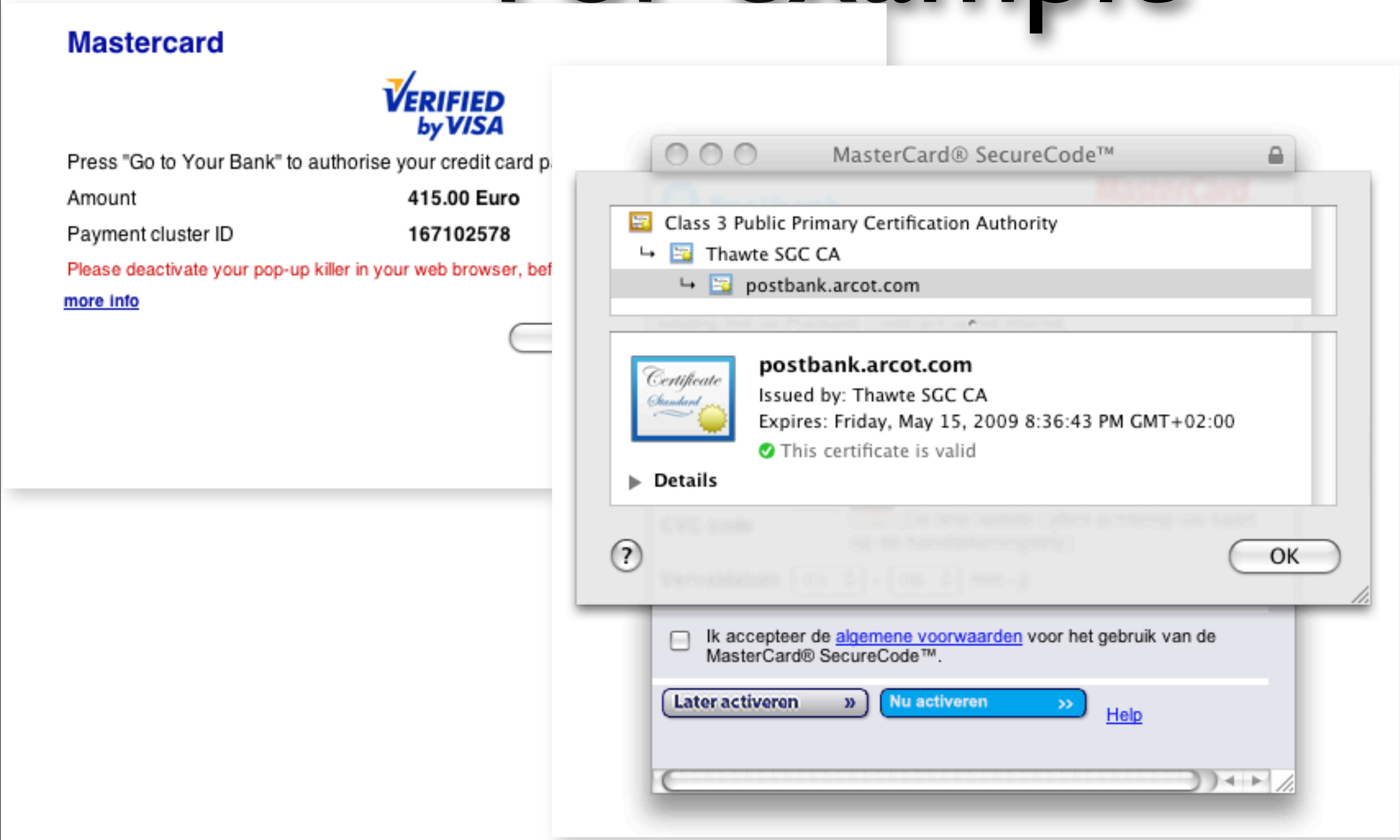


We lost DNS...
How about the other defenses ?

SSL?

- Current practices are sloppy
- Users connect to their banks
- Get redirected to unrelated domains
- User interfaces only show padlocks

For example



Mastercard

Press "Go to Your Bank" to authorise your credit card p

Amount	**415.00 Euro**
Payment cluster ID	**167102578**

Please deactivate your pop-up killer in your web browser, bef

more info

Class 3 Public Primary Certification Authority
↳ Thawte SGC CA
↳ postbank.arcot.com

postbank.arcot.com
Issued by: Thawte SGC CA
Expires: Friday, May 15, 2009 8:36:43 PM GMT+02:00
✓ This certificate is valid

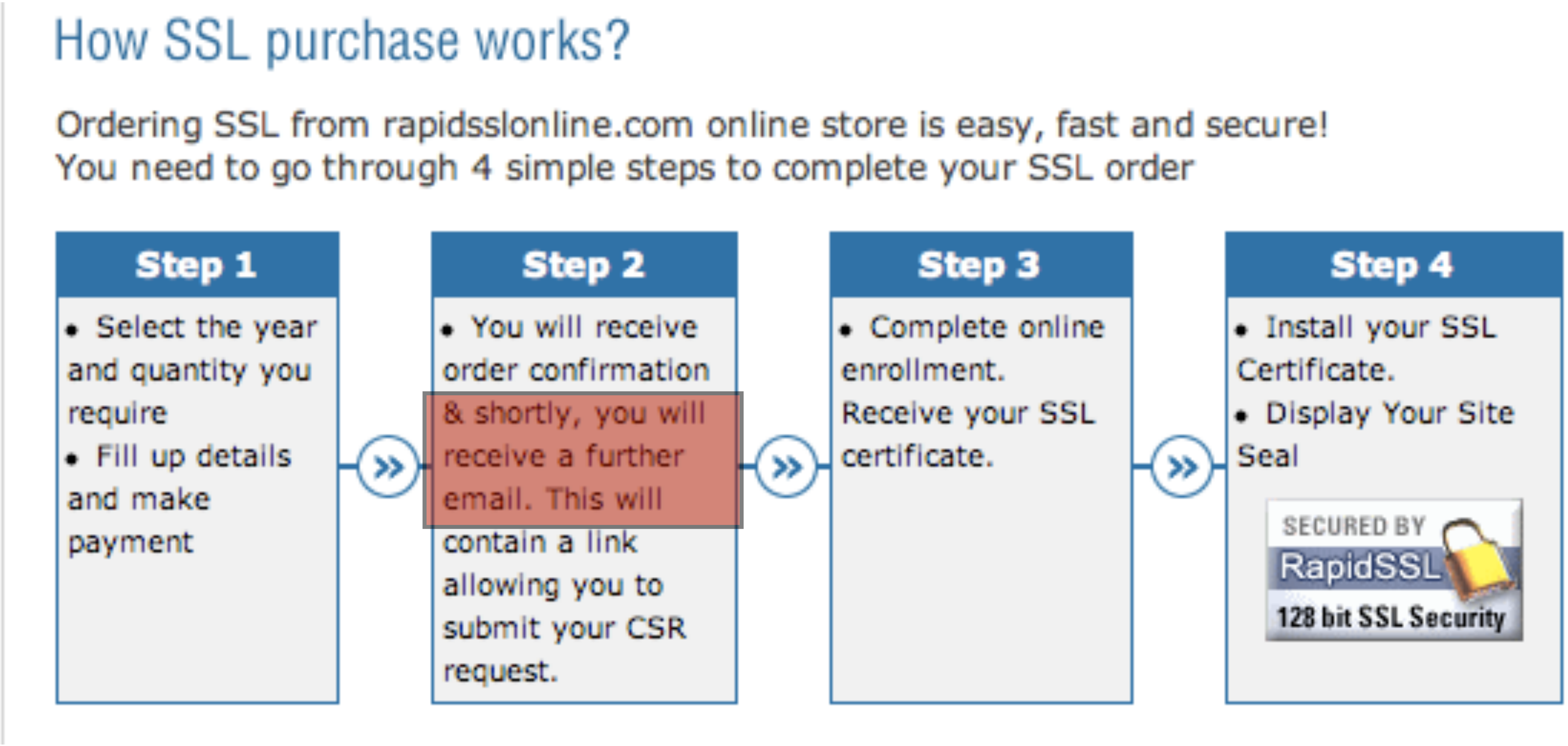
□ Ik accepteer de algemene voorwaarden voor het gebruik van de MasterCard® SecureCode™.

Exploit

- Attacker poisons DNS for www.postbank.nl
- Fake www.postbank.nl redirects to postbank.webbanksecurity.com
    - Obtaining the domain name and certificate is trivial for organized criminals
- Users are used to these sort of redirections and the domainname looks trustworthy

Things get worse

- Fake www.postbank.nl redirects to fake https://www.postbank.nl
- SSL protects agains that?
- Not if the attacker has a signed certificate
    - How would an attacker do that?



How SSL purchase works?

Ordering SSL from rapidsslonline.com online store is easy, fast and secure!
You need to go through 4 simple steps to complete your SSL order

Step 1

- Select the year and quantity you require
- Fill up details and make payment

Step 2

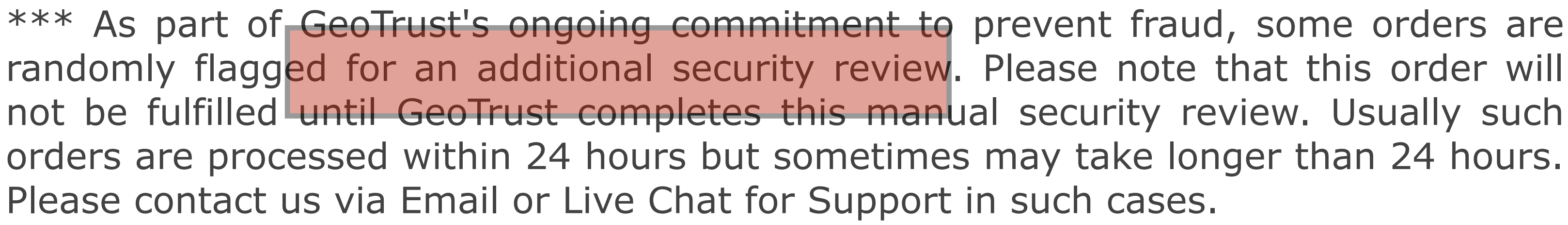
- You will receive order confirmation & shortly, you will receive a further email. This will contain a link allowing you to submit your CSR request.

Step 3

- Complete online enrollment. Receive your SSL certificate.

Step 4

- Install your SSL Certificate.
- Display Your Site Seal



*** As part of GeoTrust's ongoing commitment to prevent fraud, some orders are randomly flagged for an additional security review. Please note that this order will not be fulfilled until GeoTrust completes this manual security review. Usually such orders are processed within 24 hours but sometimes may take longer than 24 hours. Please contact us via Email or Live Chat for Support in such cases.

http://www.rapidsslonline.com/index.php

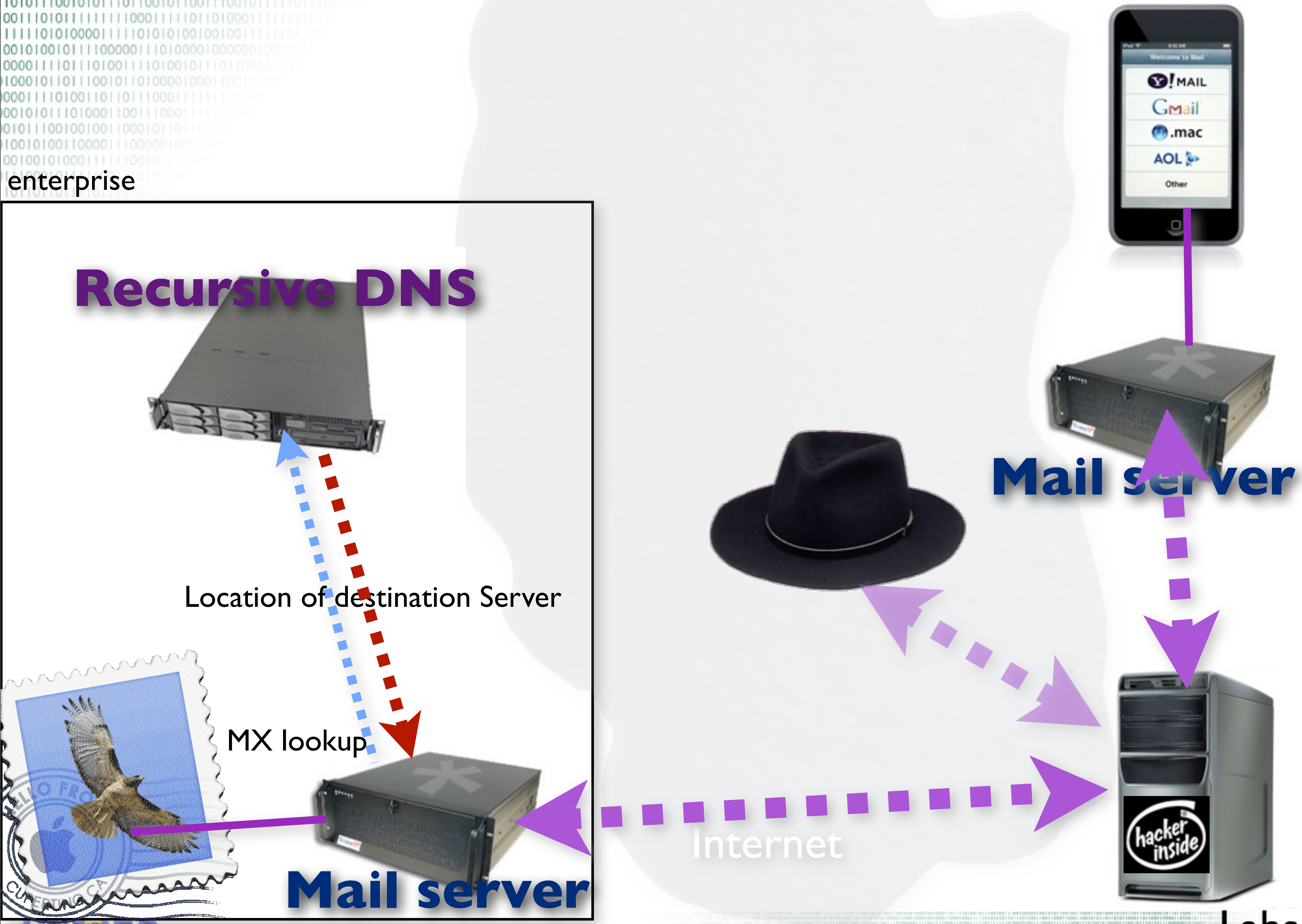
Disclaimer: GeoTrust policy is just used as an example. This is not a claim that GeoTrust does not perform due-diligence. For instance GeoTrust may actually do the necessary out of band checks, flush their DNS cache, or take other measures to prevent this sort of fraud.

Don't rely on DNS for the Security review

- Don't get the contact details out of the WHOIS, getting to WHOIS is DNS based
- Don't send confirmation e-mails to typical addresses in the domain
  - Mail uses the DNS
- Don't try to see if domain already has a SSL certificate installed. That uses the DNS

Lower hanging fruit: email

- Just attack e-mail
- Eavesdropping on e-mail
- Modifying text
- Inserting malicious content



enterprise

Recursive DNS

Location of destination Server

MX lookup

Mail server

Internet

Mail server

Technique to notice these attacks

- SPF protocol for spam recognition
  - Based on... DNS
- TSL based connections and certification
  - In practice only used for encryption of the channel
  - Often misconfigured, or with fallback in place
  - And remember the problems wrt TLS

EV vs DV

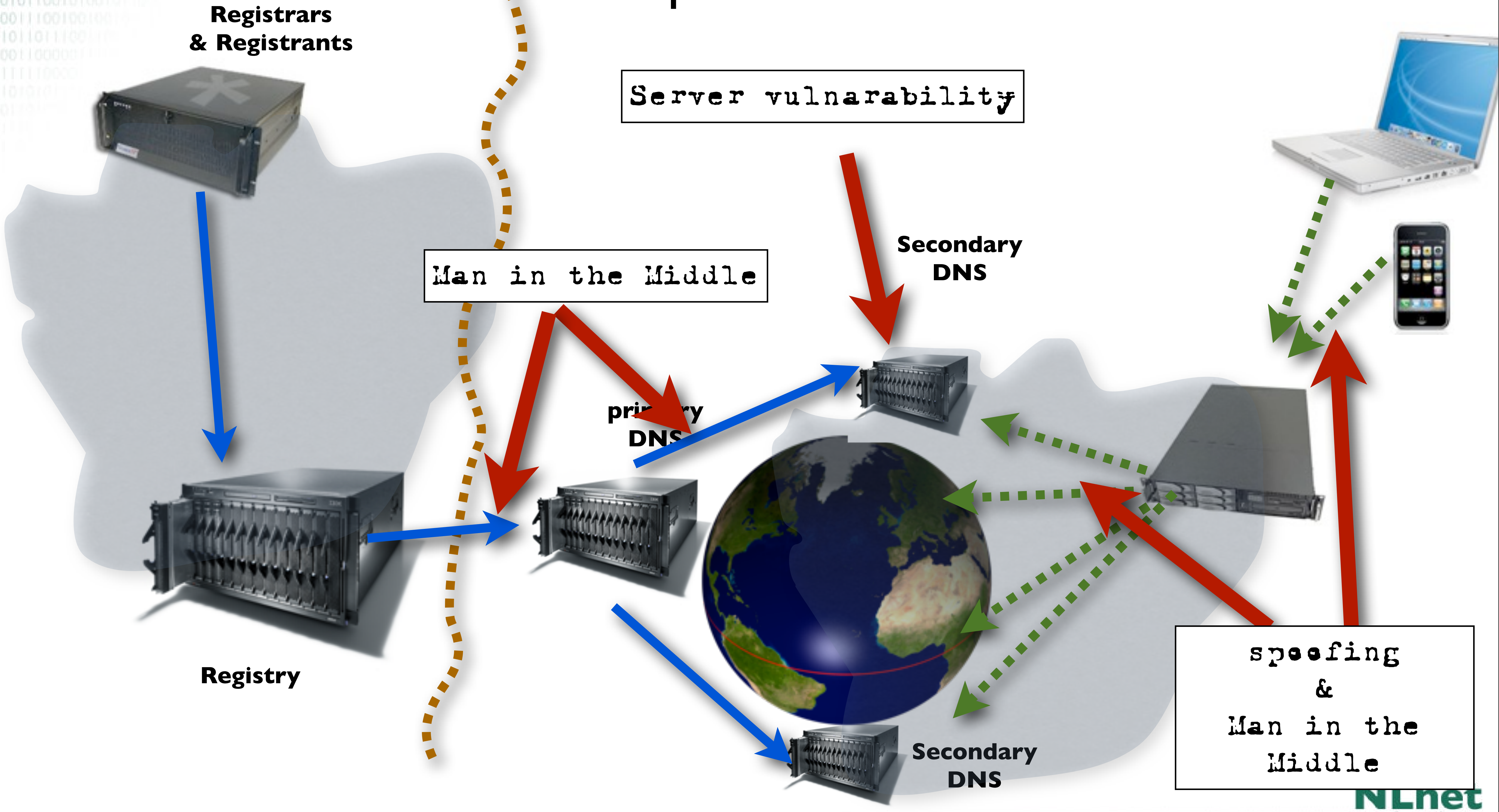
- Certificates come in two types: Domain Validation (DV) and Extended Validation (EV)
- Cert. Authorities hand out DVs purely on DNS based knowledge
- Difference: Green Glow in the browser (assuming UI is available)

Back to the DNS

Is cache poisoning the only vulnerability?

Data flow through the DNS

Where are the vulnerable points?



Registrars & Registrants

Registry

Server vulnarability

Man in the Middle

primary DNS

Secondary DNS

Secondary DNS

spoofing & Man in the Middle

Protecting (Authoritative) Servers: Host Security

- Harden your OS
  - No unnecessary services/software
  - SSH with public keys only
  - Audit
- Harden your DNS secondary service provider
  - SLAs

Protecting (Authoritative) Servers: Host Security II

- Run up-to-date software
  - OS stack
  - Nameserver software
- Software protection
  - chroot/jail environment
  - drop elevated permissions

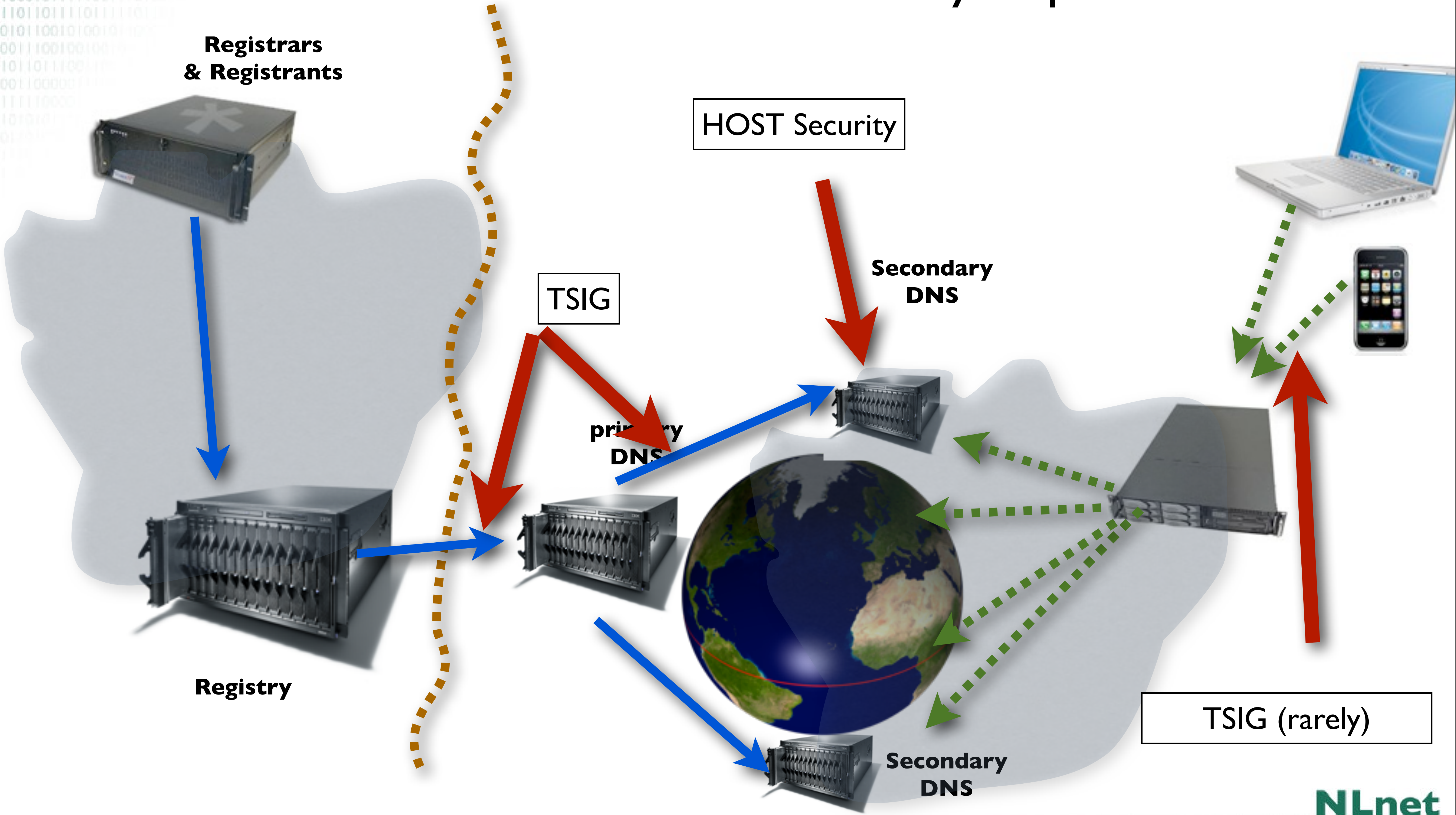
Protecting (Recursive) Servers

- Who do you accept questions from?
- Risks:
  - DOS by others (others my use your resources)
  - DOS to others (amplification attacks)
  - Have you implemented BCP 38?

Securing Host-Host Communication

Data flow through the DNS

What should you protect...



Registrars & Registrants

Registry

TSIG

primary DNS

HOST Security

Secondary DNS

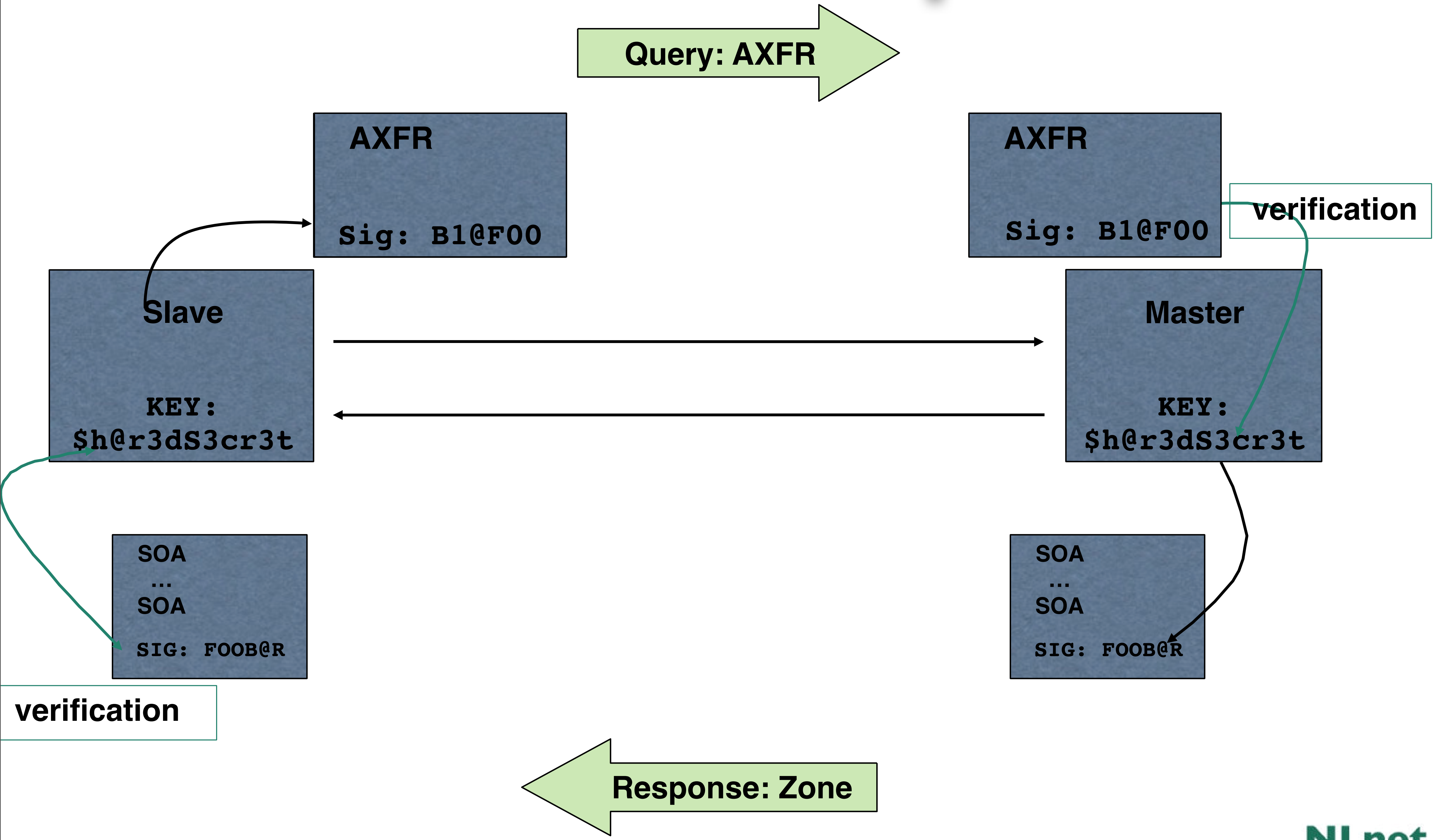
Secondary DNS

TSIG (rarely)

Transaction Signature: TSIG

- TSIG (RFC 2845)
  - –Authorising dynamic updates and zone transfers
  - –Authentication of caching forwarders
  - –Independent from other features of DNSSEC
- One-way hash function
  - –DNS question or answer and timestamp
- Traffic signed with “shared secret” key
- Used in configuration, **NOT** in zone file

TSIG Example



TSIG for Zone Transfers

1. Generate secret
2. Communicate secret
3. Configure servers
4. Test

Importance of the Time Stamp

- TSIG/SIG(0) signs a complete DNS request / response with time stamp
  - To prevent replay attacks
  - Currently hardcoded at five minutes

- Operational problems when comparing times
  - Make sure your local time zone is properly defined
  - **`date -u`** will give UTC time, easy to compare between the two systems
  - Use NTP synchronisation!

Authenticating Servers Using SIG(0)

- Alternatively, it is possible to use SIG(0)
  - –Not yet widely used
  - –Works well in dynamic update environment
- Public key algorithm
  - –Authentication against a public key published in the DNS
- SIG(0) specified in RFC 2931

Cool Application

- Use TSIG-ed dynamic updates to configure configure your laptops name
- My laptop is know by the name of aagje.secret-wg.org
  - http://ops.ietf.org/dns/dynupd/secure-ddns-howto.html
  - Mac OS users: there is a bonjour based tool.
    - www.dns-sd.org

How about Unbound?

Security Choices in Unbound

- In general, a modern paranoid resolver
- DNSSEC support.
- RFC 2181 support completely
  - Fine grained. Keeps track of where RRSets came from and won't upgrade them into answers.
  - Does not allow RRSets to be overridden by lower level rrsets

Filtering

- Scrubber:
- Only in-bailiwick data is accepted in the answer
  - The answer section must contain only answer
  - CNAME, DNAME checked that chain is correct
    - CNAME cut off and only the first CNAME kept
      - Lookup rest yourself do not trust other server
    - DNAME synthesize CNAME by unbound do not trust other server. Also cut off like above.
    - DNAME from cache only used if DNSSEC-secure.

Filtering II

- No address records in authority, additional section unless relevant – i.e. mentioned in a NS record in the authority section.
- Irrelevant data is removed
  - When the message only had preliminary parsing and has not yet been copied to the working region of memory

Entropy

- Randomness protects against spoof
  - Arc4random() (OpenBSD): crypto strong. May not be perfectly random, but predicting it is a cryptographical breakin.
  - Real entropy from OS as seed
- Query id – all 16 bits used.
- Port randomisation – uses all 16bits there, goes out of its way to make sure every query gets a fresh port number

Entropy II

- Destination address, and ipv4/ipv6. RTT band of 400msec (=everything).
    - Its not the timewindow but the randomness
- Query aggregation – same queries are not sent out – unless by different threads
- Qname strict match checked in reply
- 0x20 option
- Harden-referral-path (my draft) option
- Can use multiple source interfaces!
    - 4 outgoing IP address add +2 bits

Other measures

- Not for the wire itself
  - Heap function pointer protection (whitelisted)
  - Chroot() by default
  - User privileges are dropped (lots of code!)
  - ACL for recursion
  - No detection of attacks – assume always under attack
  - version.bind hostname.bind can be blocked or configured what to return (version hiding)
  - Disprefer recursion lame servers – they have a cache that can be poisoned