


# Paraguayan ccTLD Hacked, Google.com.py Redirected, Internal Database Leaked

Unknown Lamer posted yesterday | from MrJones 

MrJones writes "Last February 20th, hackers supposedly from Iran [accessed and modified \(English\)](#) the [www.NIC.py](#) database, redirecting [www.google.com.py](#) to another site. The hackers [posted the whole NIC.py database](#) containing full names, national ID numbers, street addresses, phone numbers, and more of registrants. This is [not the first time \(English\)](#) that NIC.py, managed by the 2 most respectful Computer Science Universities of Paraguay, was hacked. Since the entire database was released, local white hat hackers were able to calculate [how much money NIC.py was making annually \(English\)](#) by charging \$44 US per .py domain. The local CS community are urging the NIC.py administrators to do all whats possible to protect the .py domain names since the hack was done by exploiting a [simple remote code execution](#) vulnerability. If they can modify [google.com.py](#), just imagine what they can do to banks and financial institutions. Maybe Google can helps us."

# DNS and BGP



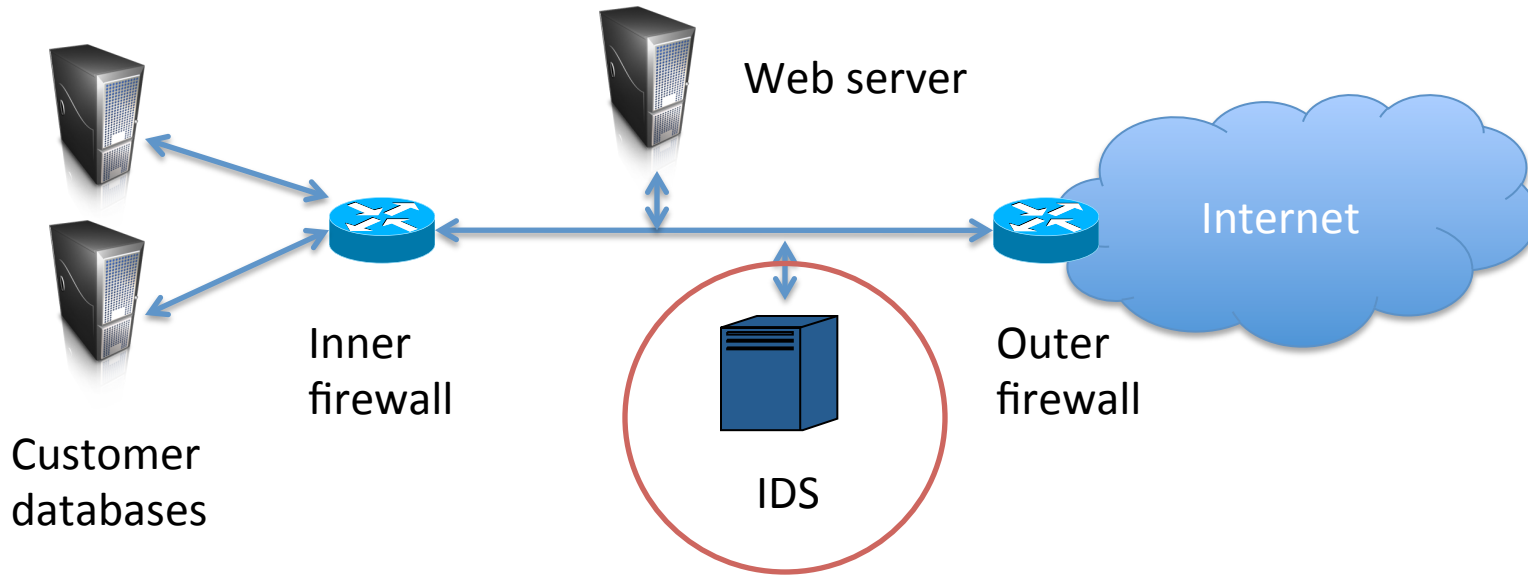
## CS642: Computer Security

Professor Ristenpart

<http://www.cs.wisc.edu/~rist/>

rist at cs dot wisc dot edu

# Network DMZ

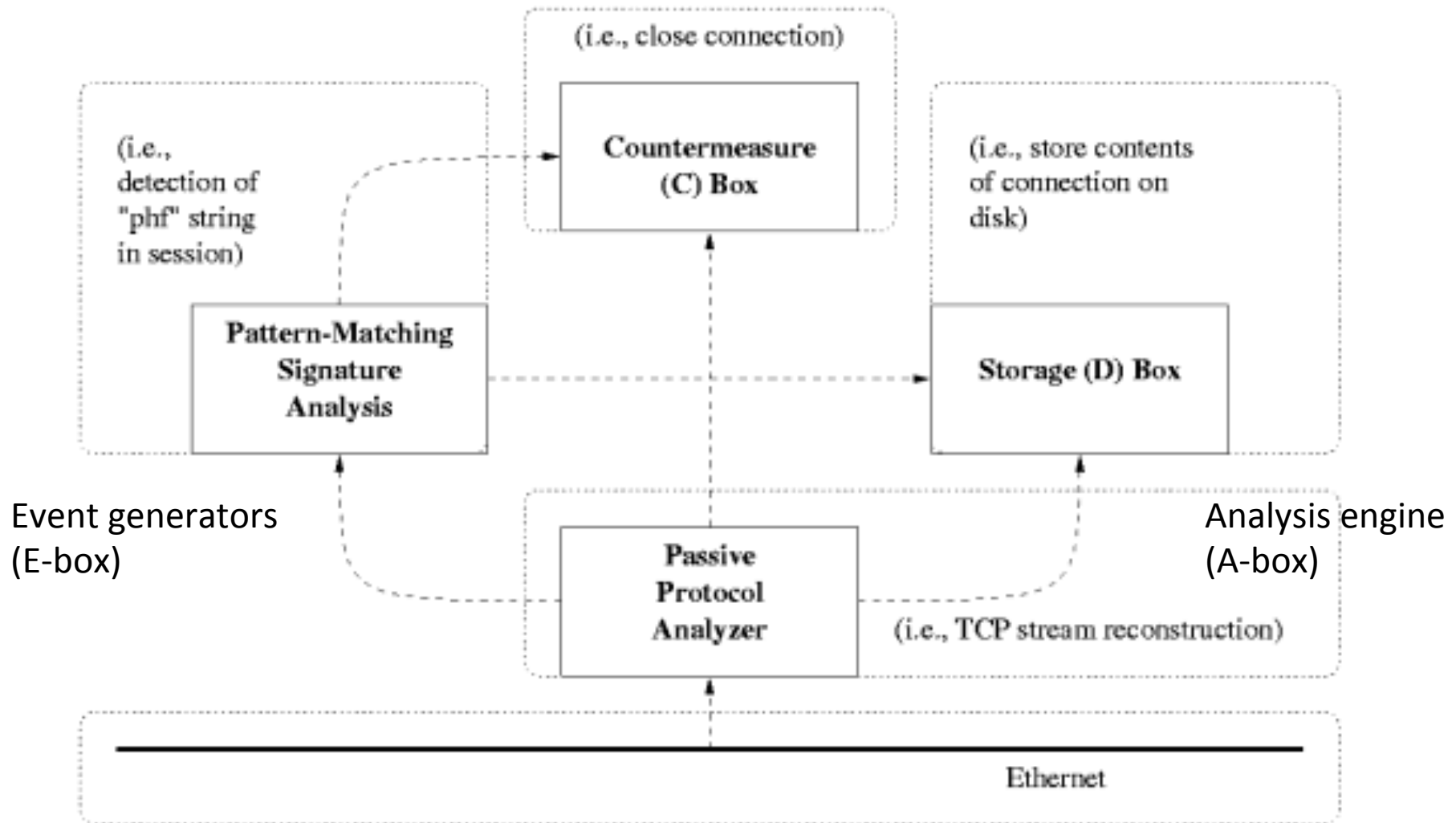


DMZ (demilitarized zone) helps isolate public network components from private network components

Firewall rules to disallow traffic from Internet to internal services

# CIDF

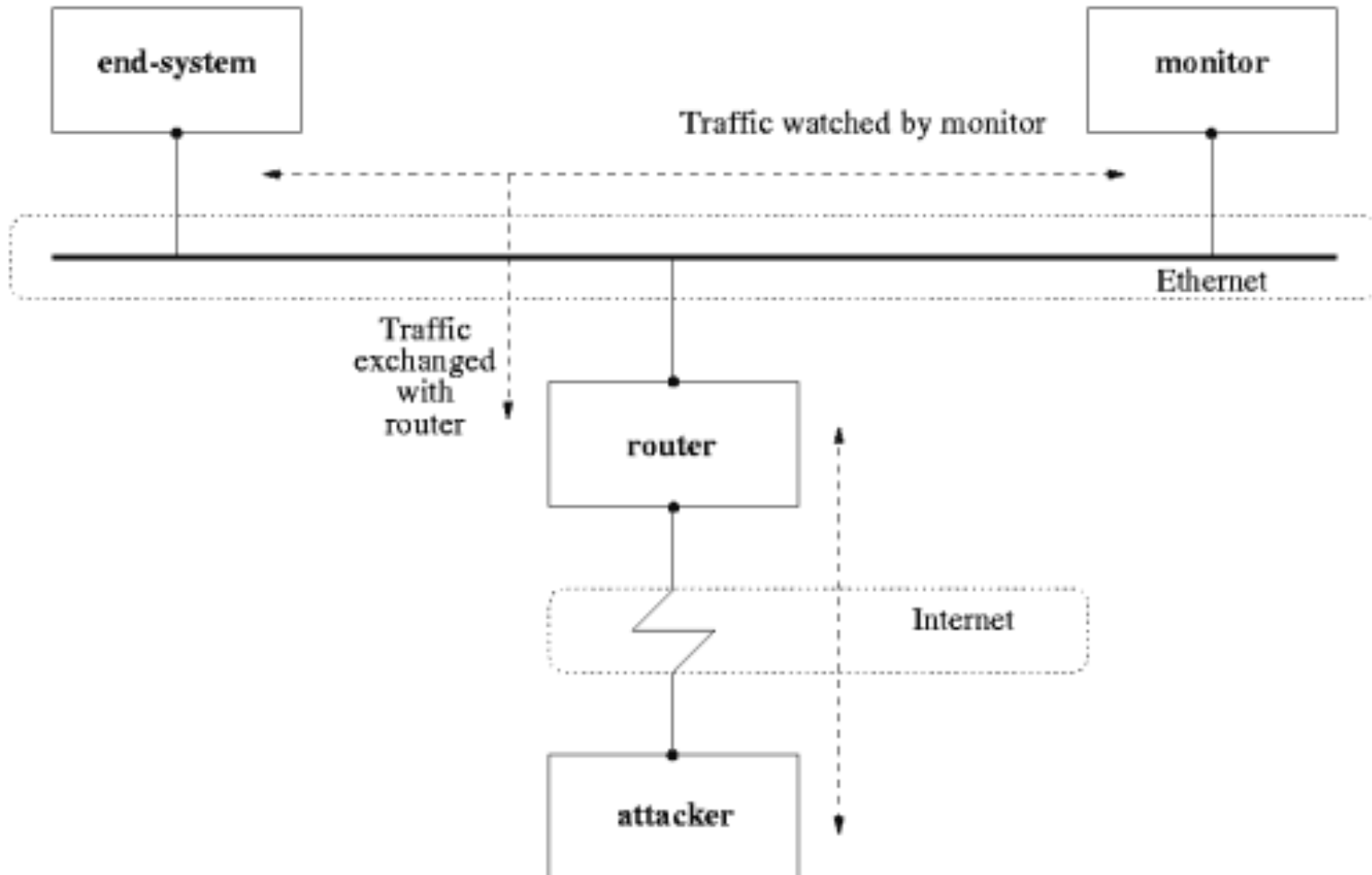
(Common intrusion detection framework)



# Two broad classes

- Anomaly detection
  - What does “normal” traffic look like?
  - Flag abnormal traffic
- Signature based
  - Define some explicit traffic patterns as bad
  - Flag them
  - E.g., regular expressions

# Basic NIDS setup



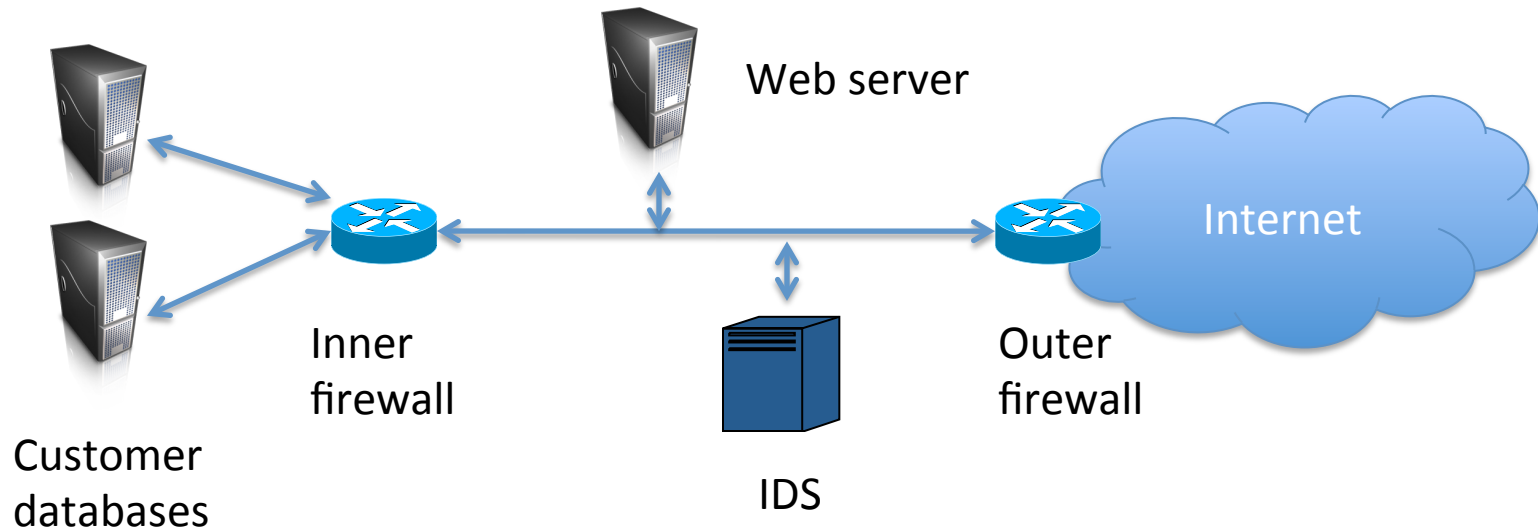
From [http://insecure.org/stf/secnet\\_ids/secnet\\_ids.html](http://insecure.org/stf/secnet_ids/secnet_ids.html)

# Some examples

- Snort (Martin Roesch)
- Bro (Vern Paxson)
  - 1999: 27,000 lines of C++ code

# Attacking or bypassing NIDS

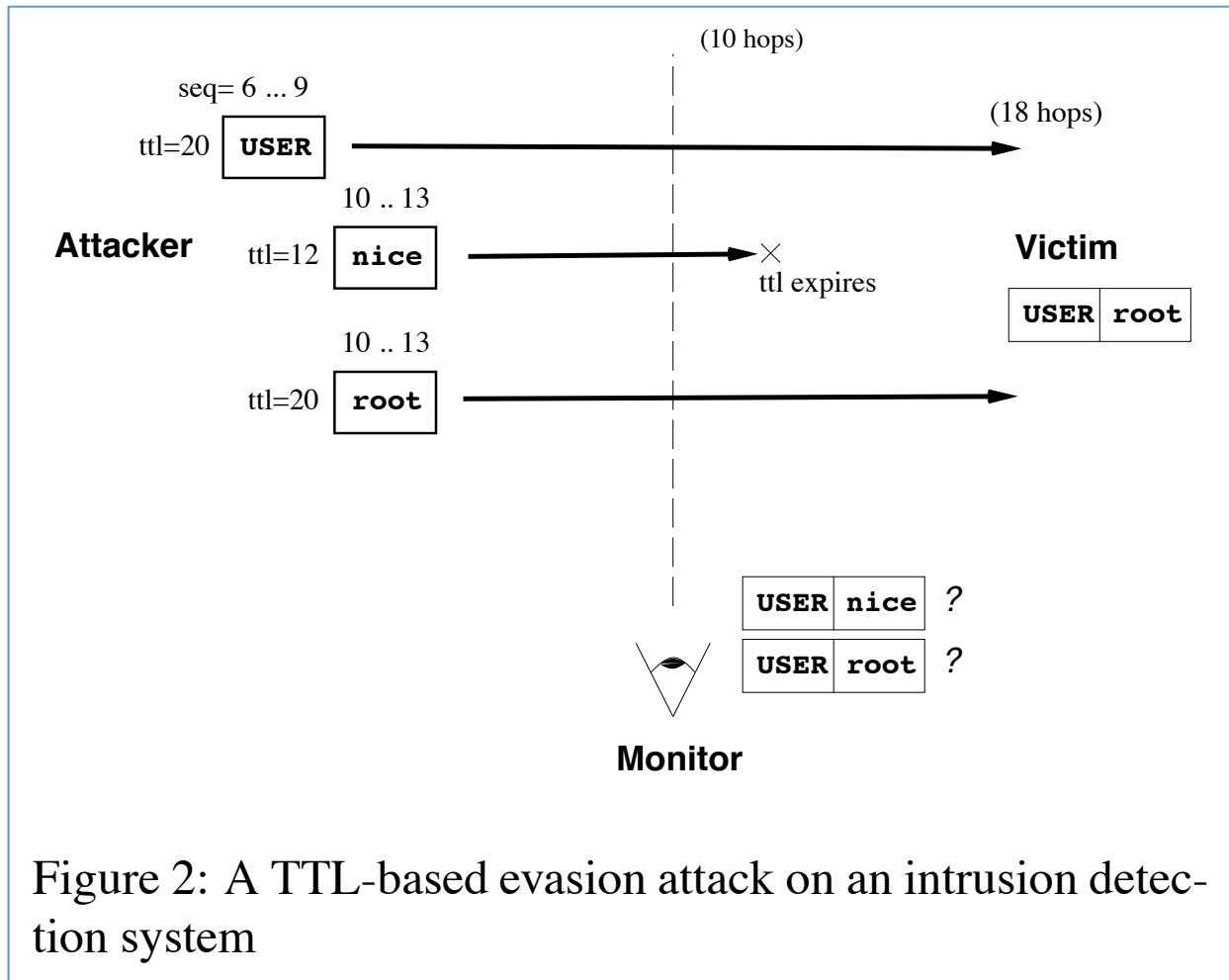
- How can one circumvent a NIDS?



Overload attacks, crash attacks, subterfuge attacks



# Subterfuge attack example



# Anomalous, non-attack traffic

- “Storms” of 10,000s of FIN or RST packets due to protocol implementation error
- “Storms” due to foggy days
  - Fog in SF bay area killed a connection, causing routing flaps and in turn routing loops
- SYN packet with URG flag set
  - Flags == SYN fails

# Honeypots

- Systems that should have no legitimate traffic
  - Isolated and monitored
  - Any traffic routed to it is spurious
- High interaction (e.g., a full system)
- Low interaction (e.g., Honeyd)
- Honeynets, honeyfarms
  - lots of honeypots
- Honeytoken
  - email address
  - credit card number

# DNS and BGP

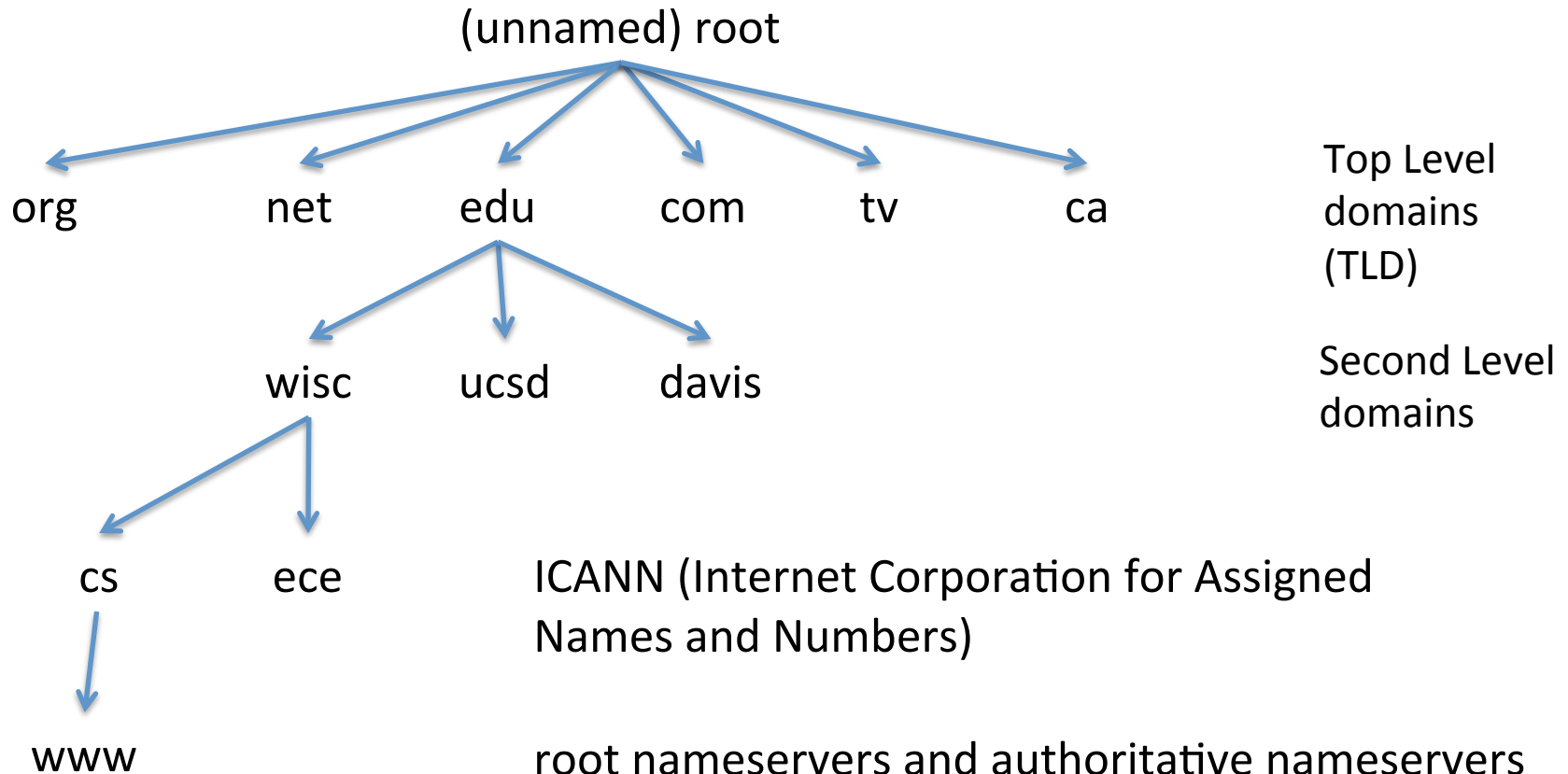
# 128.105.5.31

We don't want to have to remember IP addresses

```
[rist@seclab1] (17)$ head hosts
#
#      Wisconsin CS Local Host Table
#
127.0.0.1      localhost
128.105.6.39   smtp.cs.wisc.edu smtp
128.105.6.40   spam.cs.wisc.edu spam spam-test
128.105.6.42   spam.cs.wisc.edu spam spam-test
128.105.6.38   spam.cs.wisc.edu spam spam-test
128.105.1.1    ge-5-1.cisco-border1.cs.wisc.edu ge-5-1.cisco-border1
128.105.1.2    ge-1-2.cisco1.cs.wisc.edu ge-1-2.cisco1
[rist@seclab1] (18)$ █
```

Early days of ARPANET: manually managed hosts.txt served from single computer at SRI

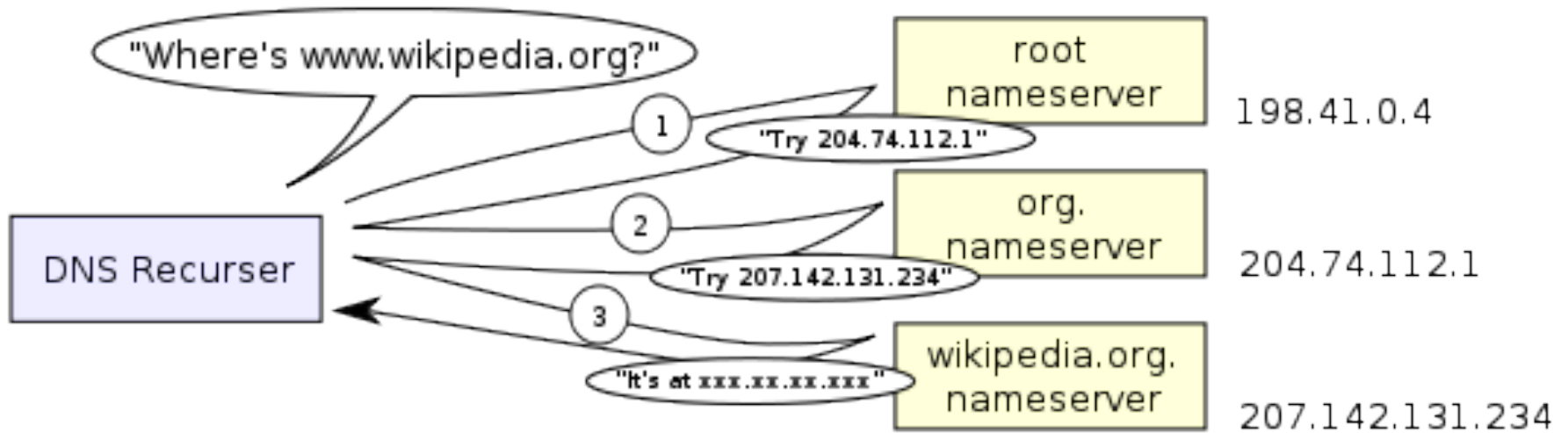
# Heirarchical domain name space



max 63  
characters

Zone: subtree

# Resolving names



From

[http://en.wikipedia.org/wiki/File:An\\_example\\_of\\_theoretical\\_DNS\\_recursion.svg](http://en.wikipedia.org/wiki/File:An_example_of_theoretical_DNS_recursion.svg)

# Example DNS query types

A	address (get me an IPv4 address)
AAAA	IPv6 address
NS	name server
TXT	human readable text, has been used for some encryption mechanisms
MX	mail exchange



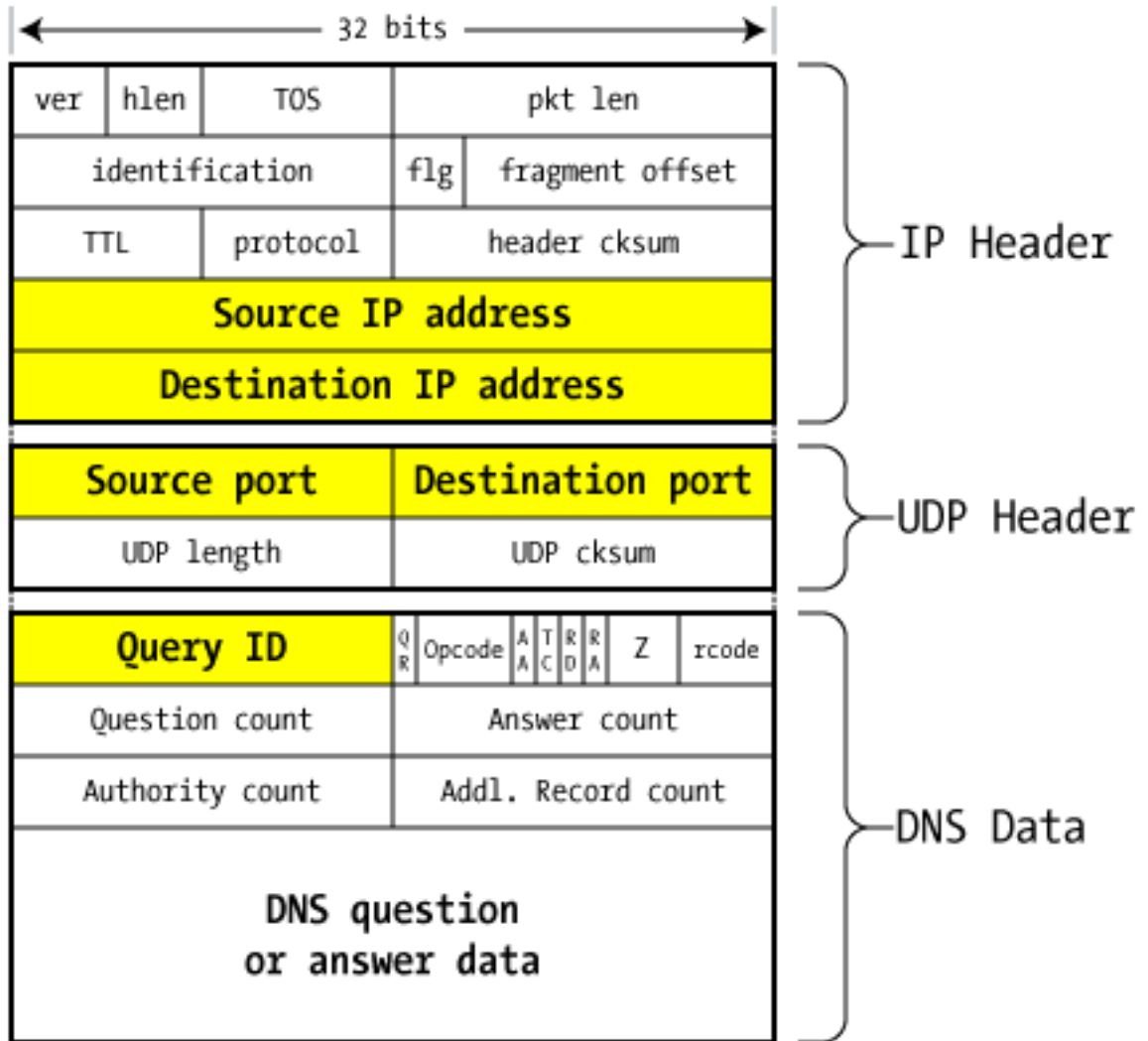
# Caching

- DNS servers will cache responses
  - Both negative and positive responses
  - Speeds up queries
  - periodically times out. TTL set by data owner

# DNS packet on wire

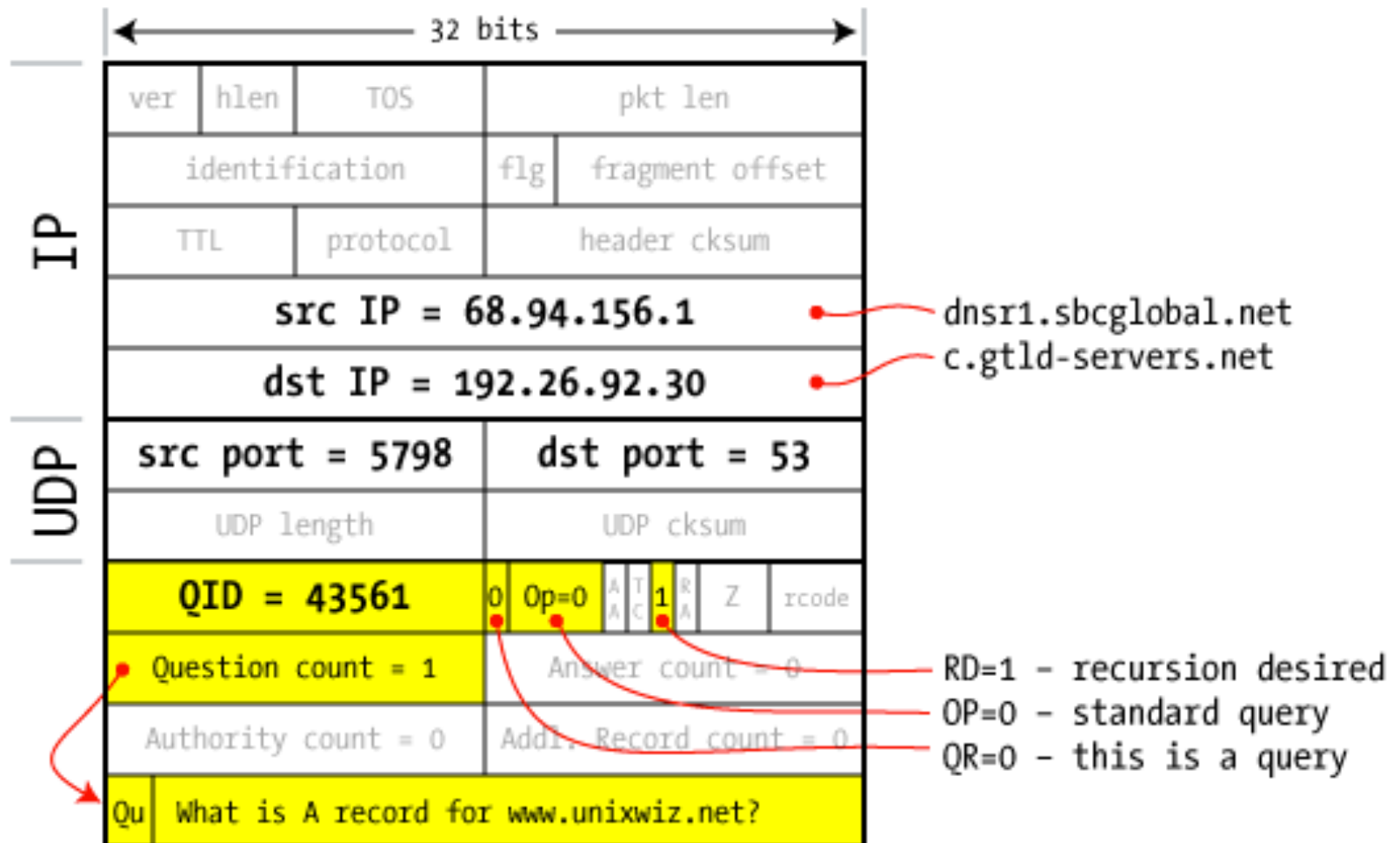
Query ID is 16-bit random value

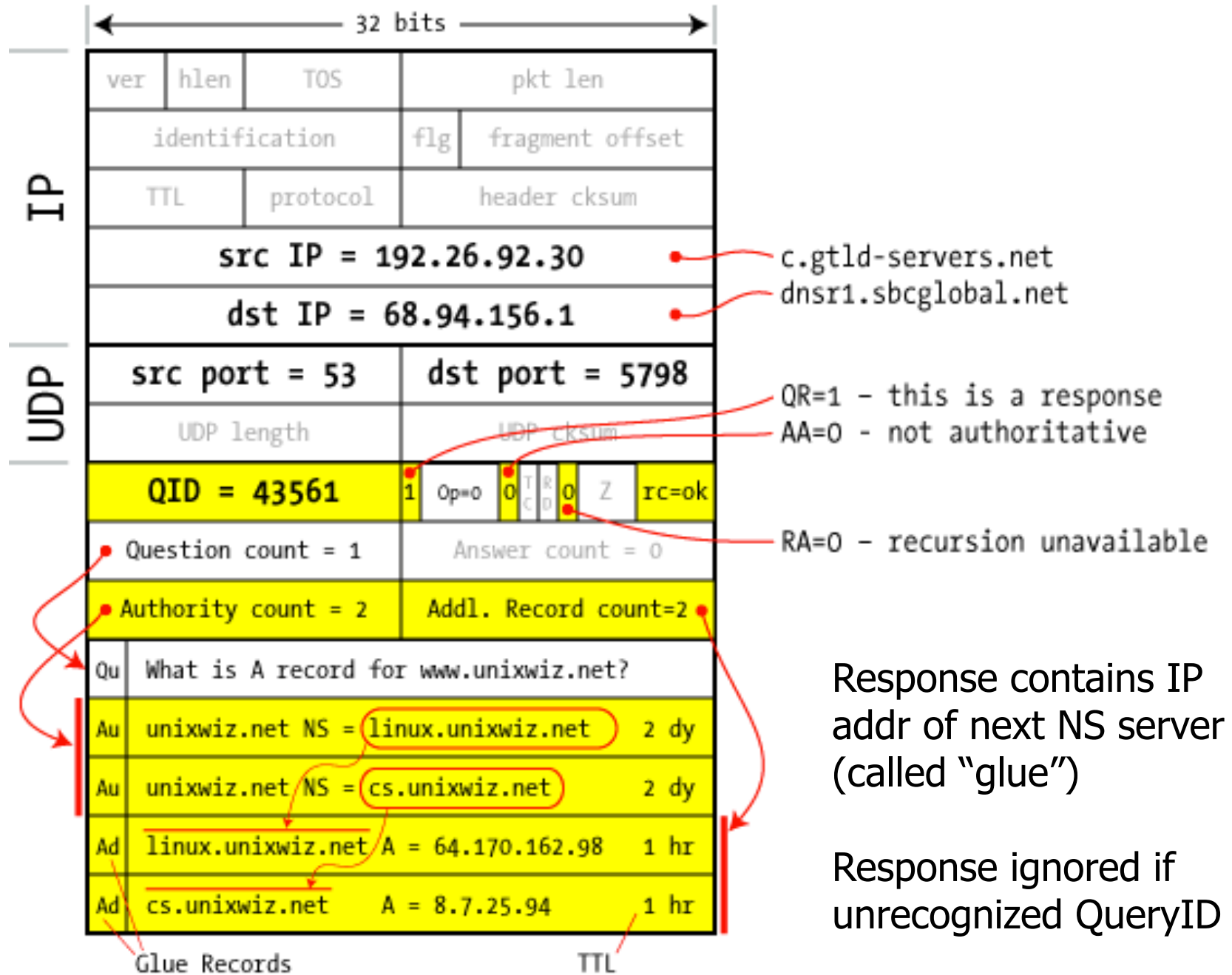
We'll walk through the example from Friedl's document



From Friedl explanation of DNS cache poisoning, as are following diagrams

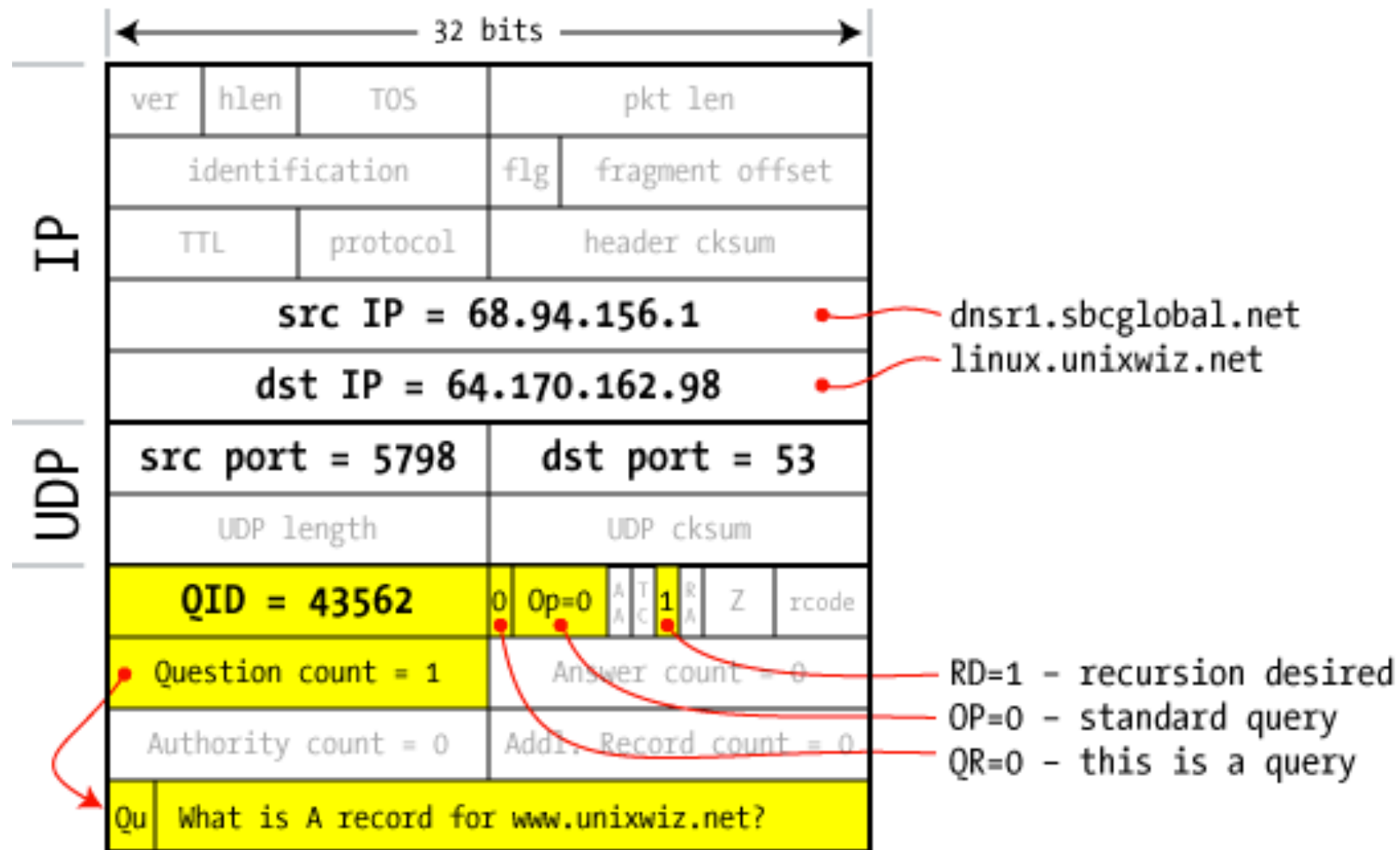
# Query from resolver to NS

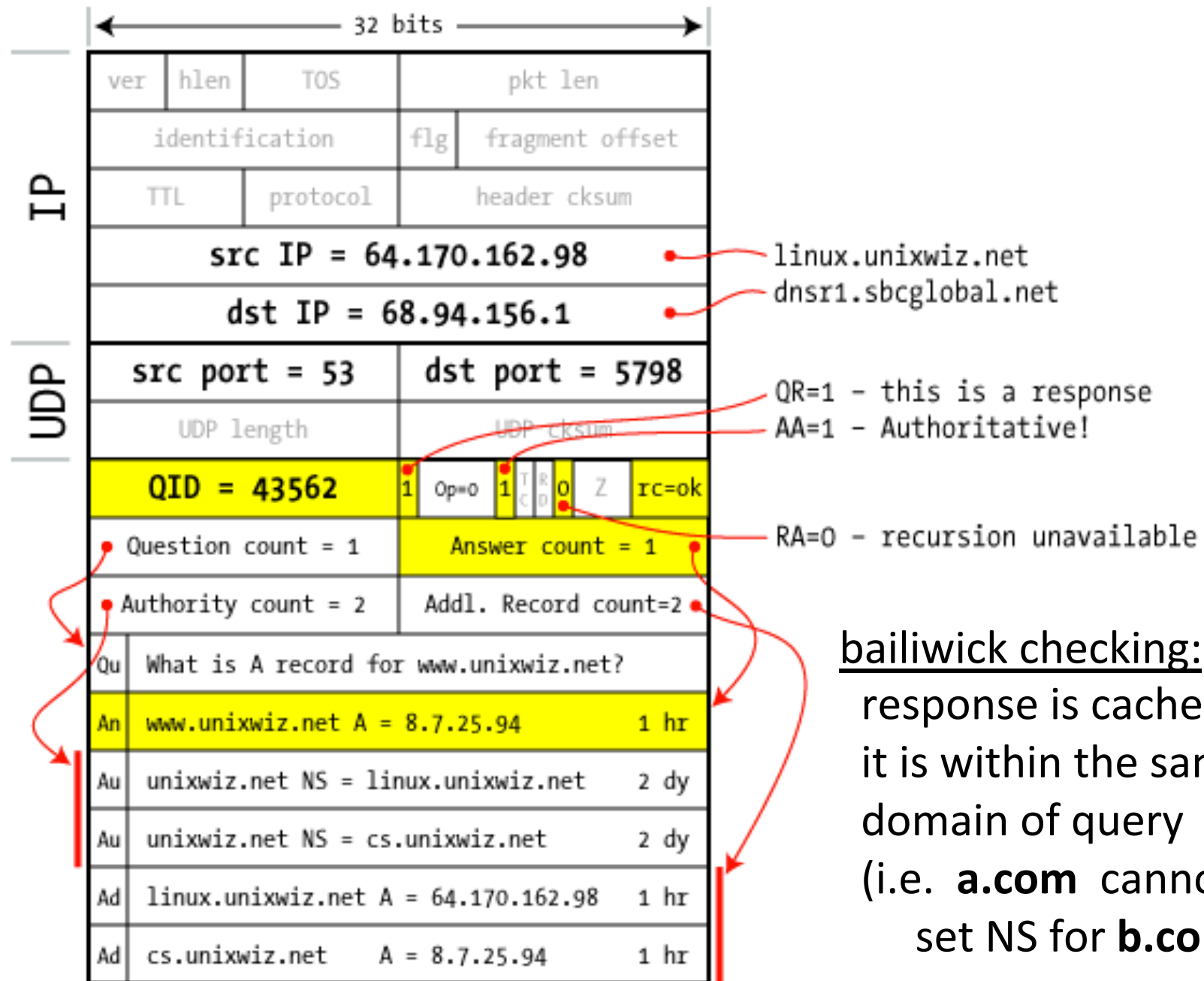




Response contains IP  
addr of next NS server  
(called "glue")

Response ignored if  
unrecognized QueryID





# Here we go again...

- What security checks are in place?
  - Random query ID's to link responses to queries
  - Bailiwick checking (sanity check on response)
- No authentication
  - DNSsec is supposed to fix this but no one uses it yet
- Many things trust hostname to IP mapping
  - Browser same-origin policy
  - URL address bar

# What are clear problems?

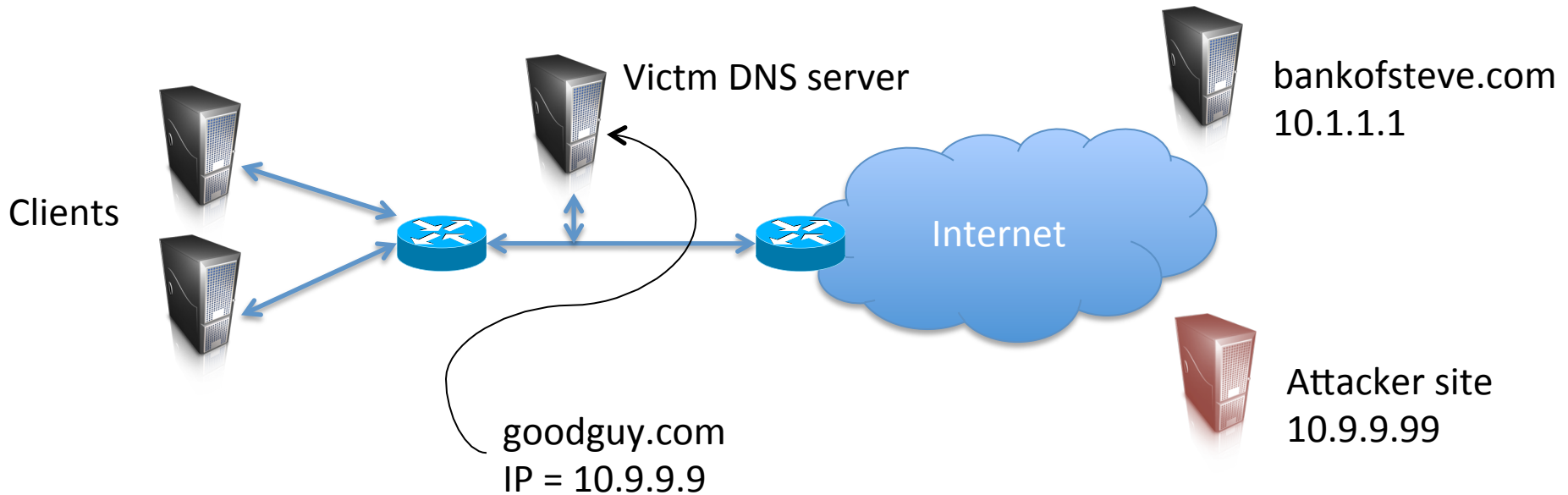
- Corrupted nameservers
- Intercept & manipulate requests
- Other obvious issues?



# DDoS against DNS

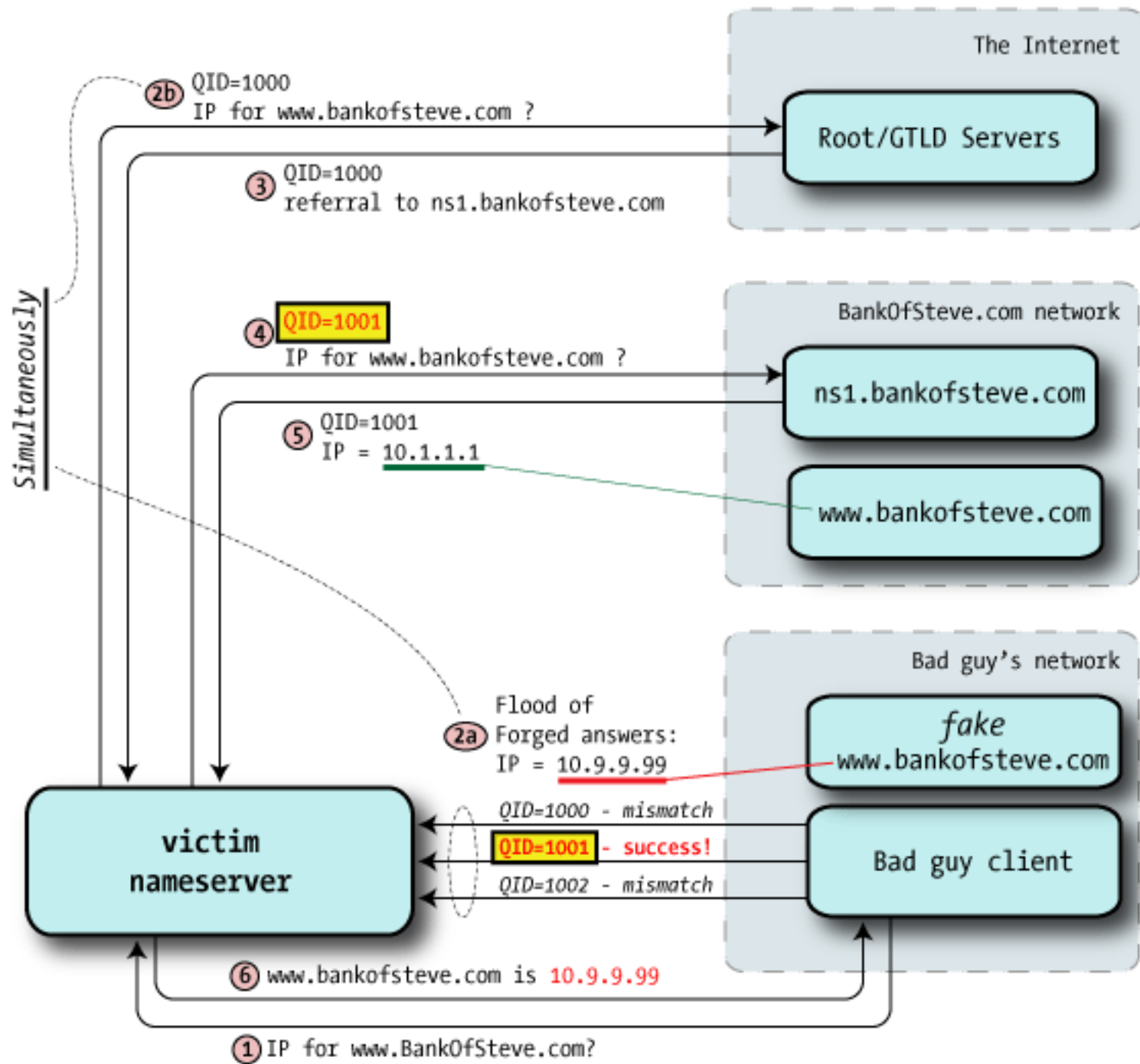
- Denial of Service
  - take down DNS server, clients can't use Internet
  - Feb 6, 2007 attack against 6 of 13 root servers:
    - 2 suffered very badly
    - Others experienced heavy traffic
- DoD purportedly has interesting response:
  - “In the event of a massive cyberattack against the country that was perceived as originating from a foreign source, the United States would consider launching a counterattack or bombing the source of the cyberattack, Hall said. But he noted the preferred route would be warning the source to shut down the attack before a military response.”
  - [http://www.computerworld.com/s/article/9010921/RSA\\_U.S.\\_cyber\\_counterattack\\_Bomb\\_one\\_way\\_or\\_the\\_other](http://www.computerworld.com/s/article/9010921/RSA_U.S._cyber_counterattack_Bomb_one_way_or_the_other)

# DNS cache poisoning



How might an attacker do this?

Assume DNS server uses predictable UDP port

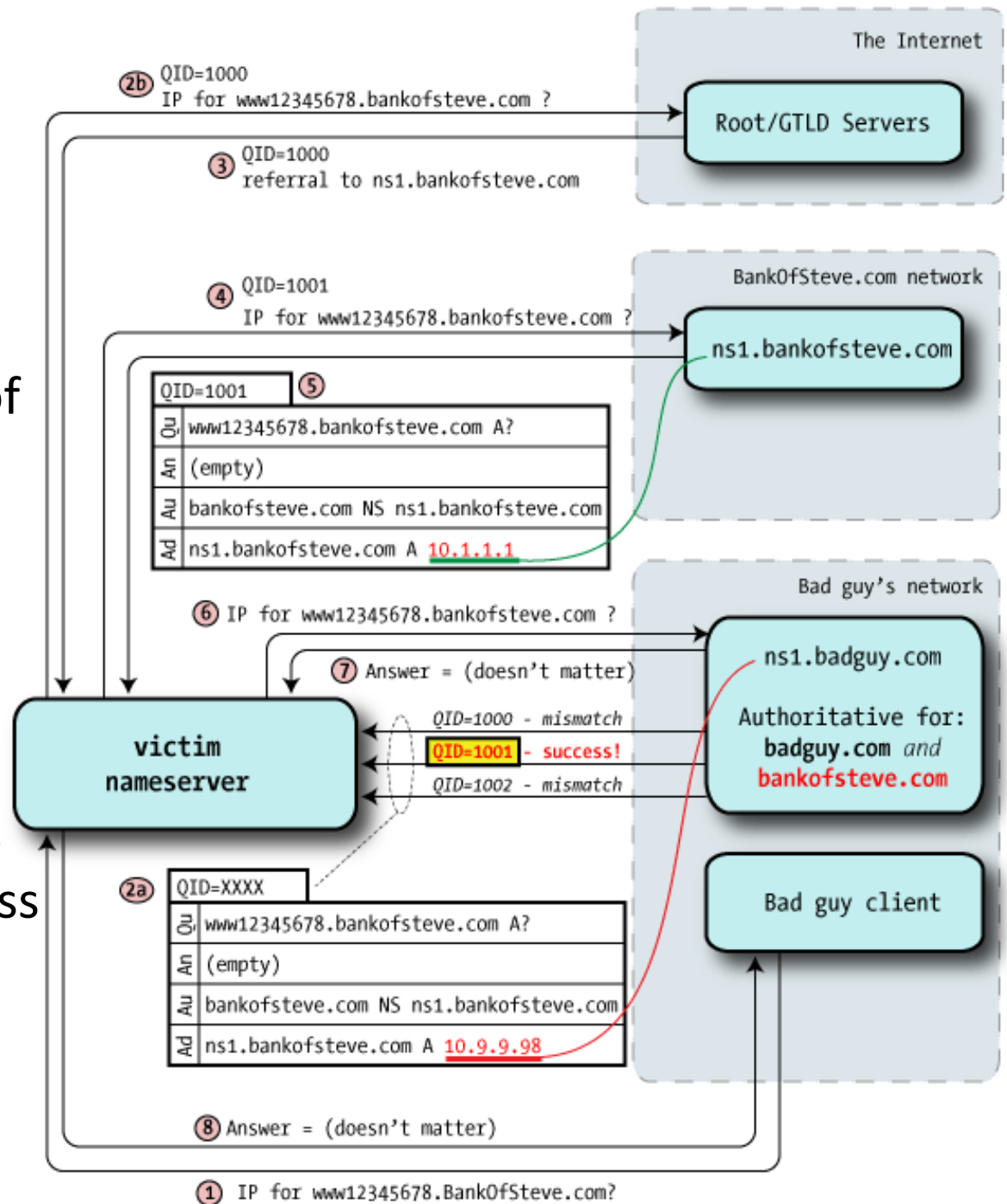


Another idea:

- Poison cache for NS record instead
- Now can take over all of second level domain

How many tries does this require?

- Try 256 different QIDs
- Good chance of success



# Does happen in the wild

## HD Moore pwned with his own DNS exploit, vulnerable AT&T DNS servers to blame

By Dancho Danchev | July 30, 2008, 8:08am PDT

**Summary:** *A week after |)ruid and HD Moore release part 2 of DNS exploit, HD Moore's company BreakingPoint has suffered a traffic redirection to a rogue Google site, thanks to the already poisoned cache at AT&T servers to which his company was forwarding DNS traffic : "It happened on Tuesday morning, when Moore's company, BreakingPoint had some [...]*

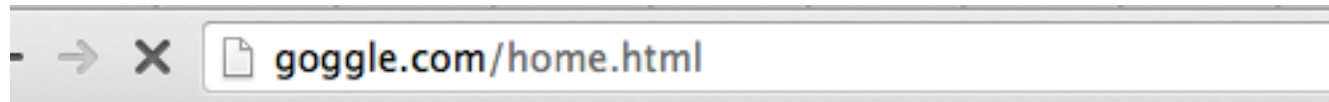
<http://www.zdnet.com/blog/security/hd-moore-pwned-with-his-own-dns-exploit-vulnerable-at-t-dns-servers-to-blame/1608?tag=content;siu-container>

# Defenses

- Query ID size is fixed at 16 bits
- Repeat each query with fresh Query ID
  - Doubles the space
- Randomize UDP ports
  - Dan Bernstein's DJBDNS did this already
  - Now other implementations do, too
- DNSsec
  - Cryptographically sign DNS responses, verify via chain of trust from roots on down

# Phishing is common problem

- Typo squatting:
  - [www.ca.wisc.edu](http://www.ca.wisc.edu)
  - [www.goggle.com](http://www.goggle.com)
- Other shenanigans:
  - [www.badguy.com/\(256 characters of filler\)/www.google.com](http://www.badguy.com/(256 characters of filler)/www.google.com)
- Phishing attacks
  - These just trick users into thinking a malicious domain name is the real one



**The page at goggle.com says:**

\*\*\*\*\*

Congratulations!

You are Todays Lucky Visitor.

Click OK to continue

\*\*\*\*\*

OK



## SEO Via DNS "Piggybacking"

Posted by samzenpus on Wednesday October 12, @04:03PM  
from the slip-in-there dept.



An anonymous reader writes

"There is an interesting story over at the SANS Internet Storm Center that shows details on about 50 organizations that have had new machine names added to their DNS zone information. These were then pointed to sites used to [boost the search engine cred](#) of pharma, personals, and porn sites. If you outsource your DNS, how would you ever catch something like this?"

Read the **67** comments



fraud seo it

# DNS piggybacking

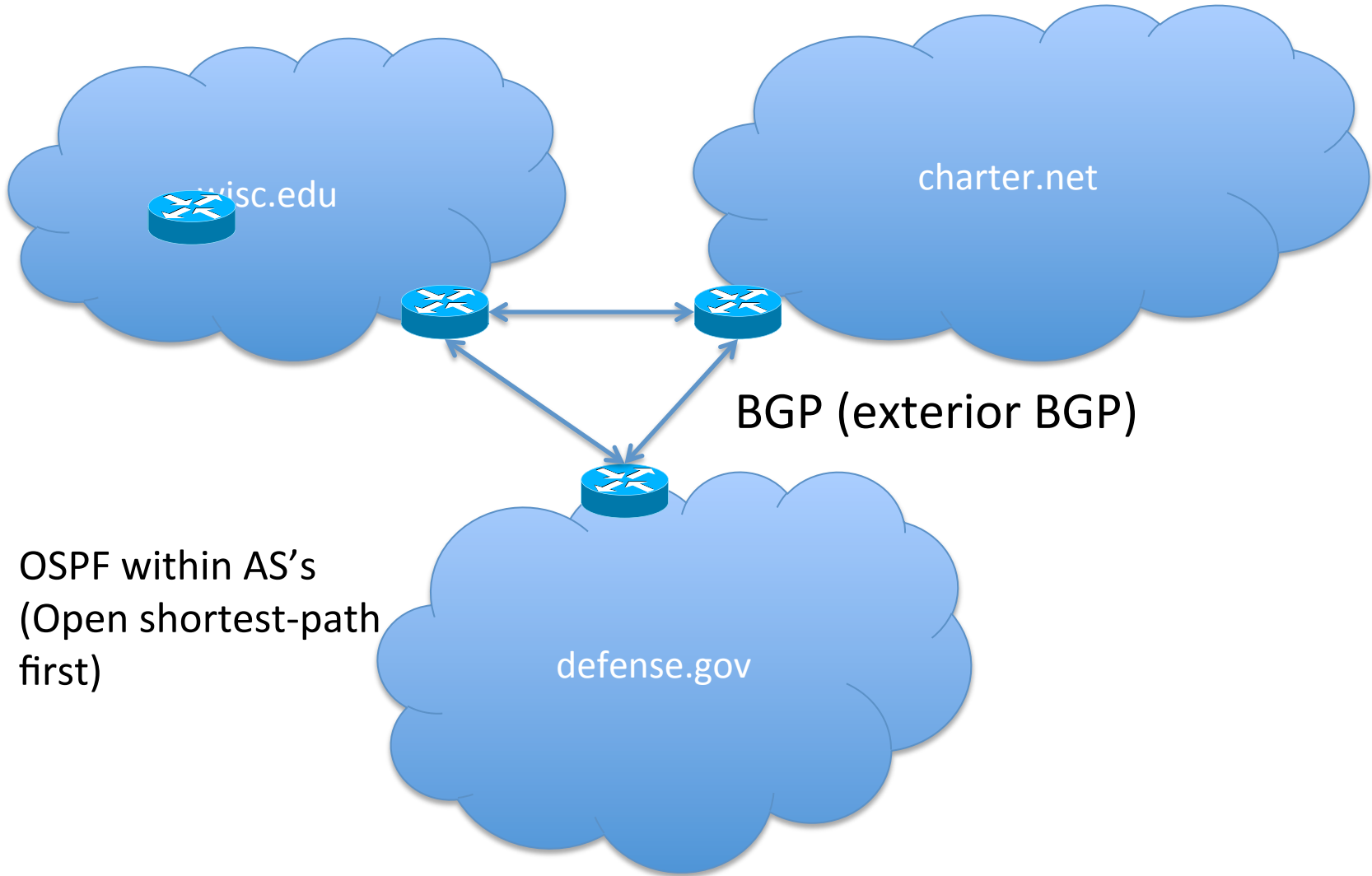
buy-cialis.sacmetrofire.ca.gov			
buy-viagra.sacmetrofire.ca.gov			
drugs.sacmetrofire.ca.gov	74.220.215.210	www.sacmetrofire.ca.gov	66.147.240.176
mgdrugs.sacmetrofire.ca.gov			
rxdrugs.sacmetrofire.ca.gov			

From <https://isc.sans.edu/diary/What+s+In+A+Name+/11770>

Attackers maliciously added extra lower level domain names to valid domain name

This is helpful for search engine optimization

# BGP and routing

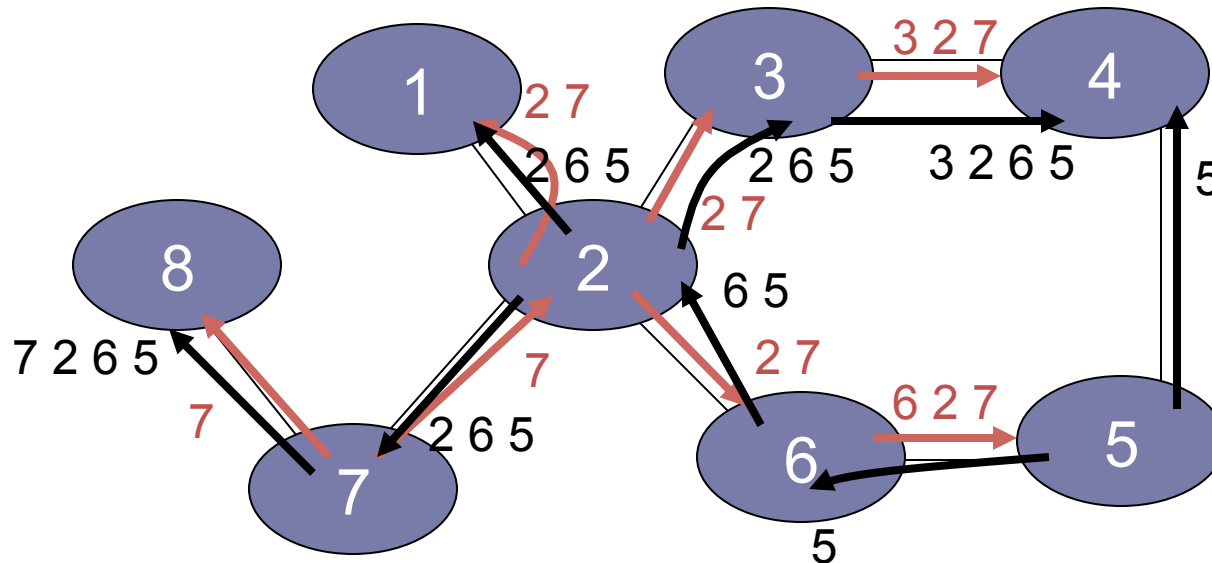


# BGP

- Policy-based routing
  - AS can set policy about how to route
    - economic, security, political considerations
- BGP routers use TCP connections to transmit routing information
- Iterative announcement of routes

# BGP example

[D. Wetherall]



- 2, 7, 3, 6 are Transit AS
- 8, 1 are Stub AS
- 4,5 multihomed AS
- Algorithm seems to work OK in practice
  - BGP does not respond well to frequent node outages

# IP hijacking

- BGP unauthenticated
  - Anyone can advertise any routes
  - False routes will be propagated
- This allows IP hijacking
  - AS announces it originates a prefix it shouldn't
  - AS announces it has shorter path to a prefix
  - AS announces more specific prefix

# Malicious or misconfigurations?

- AS 7007 incident in 1997
  - “Okay, so panic ensued, and we unplugged \*everything\* at 12:15PM almost to the second.” [sic]
  - <http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html>
- China Telecom hijacks large chunks of Internet in 2010
  - <http://bgpmon.net/blog/?p=282>

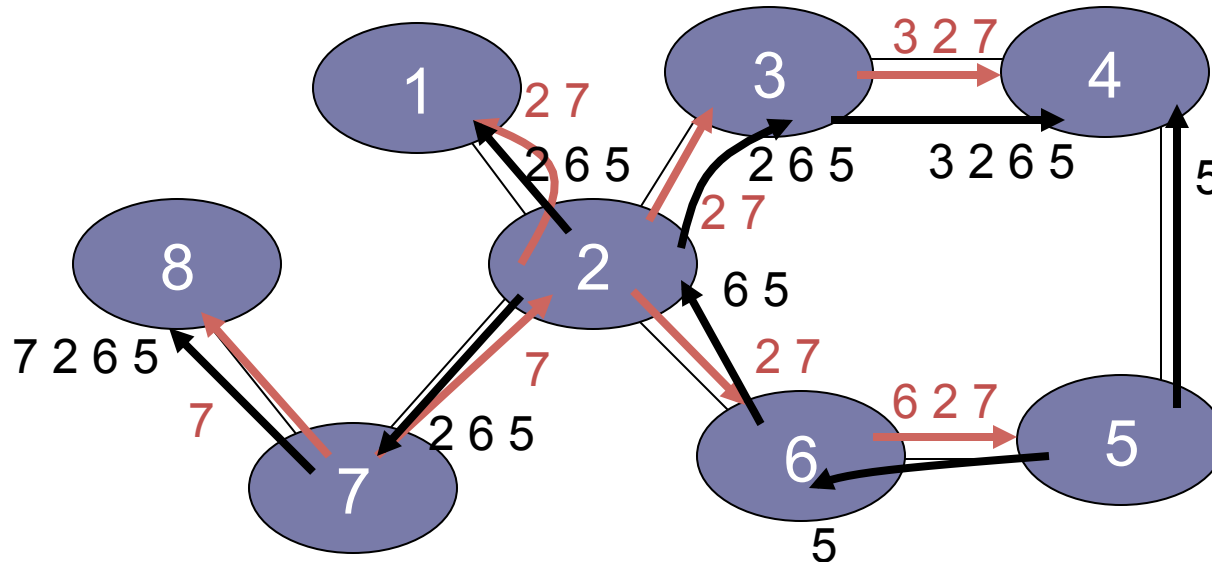
# Youtube incident

- Pakistan attempts to block Youtube
  - youtube is 208.65.152.0/22
  - youtube.com = 208.65.153.238
- Pakistan ISP advertises 208.65.153.0/24
  - more specific, prefix hijacking
- Internet thinks youtube.com is in Pakistan
- Outage resolved in 2 hours...



# BGPsec

[D. Wetherall]



- Route announcements must be cryptographically signed
  - AS can only advertise as itself
  - AS cannot advertise for IP prefixes it does not own
- Requires a public-key infrastructure (PKI)
- Still in development:
  - <http://tools.ietf.org/html/draft-lepinski-bgpsec-protocol-00#ref-7>

# Internet Security

- Recurring themes:
  - Built without any authenticity mechanisms in mind
  - Functionality mechanisms (sequence #'s) become implicit security mechanisms
  - New attempts at backwards-compatible security mechanisms
    - IP -> IPsec
    - DNS -> DNSsec
    - BGP -> BGPsec

