



# Tracing a DNS Reflection Attack via Anycast Routing Changes

Duane Wessels  
October, 2011

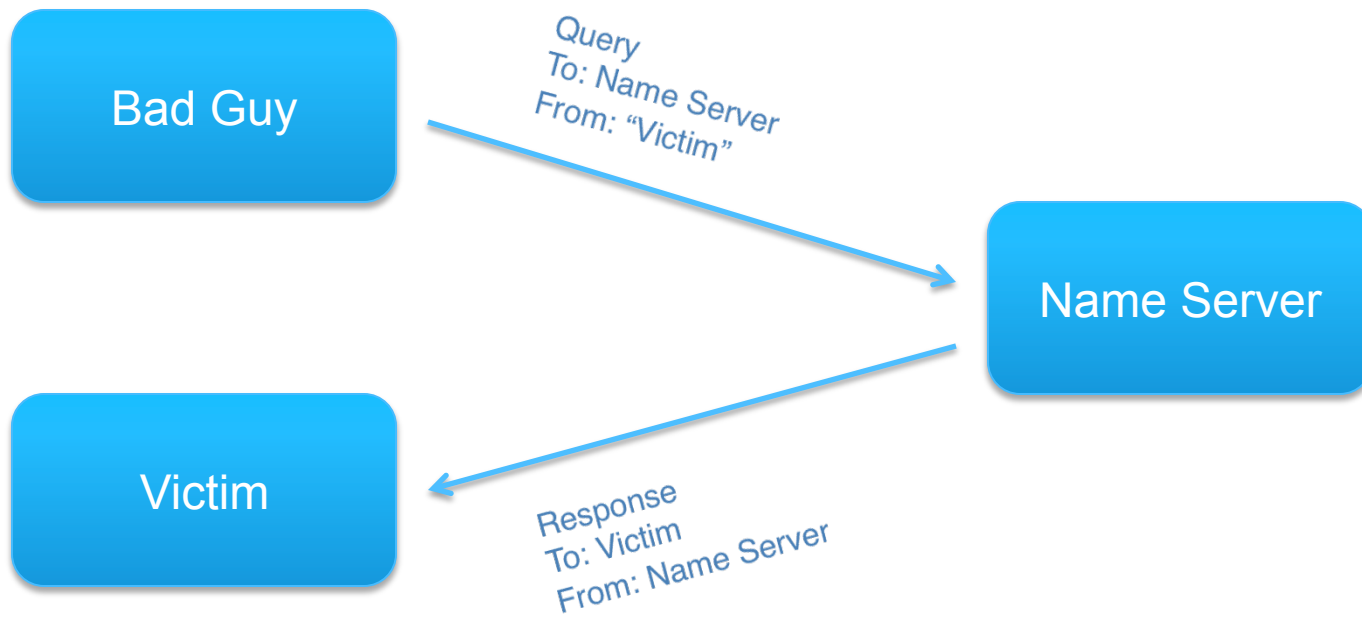


# A Brief History



- Big Bang
- The Earth cooled
- Internet (TCP/IP) invented
- Domain Name System invented
- People realize that DNS/UDP makes a great DDoS attack vector.

# A DNS Reflector Attack



## Subject of this Talk



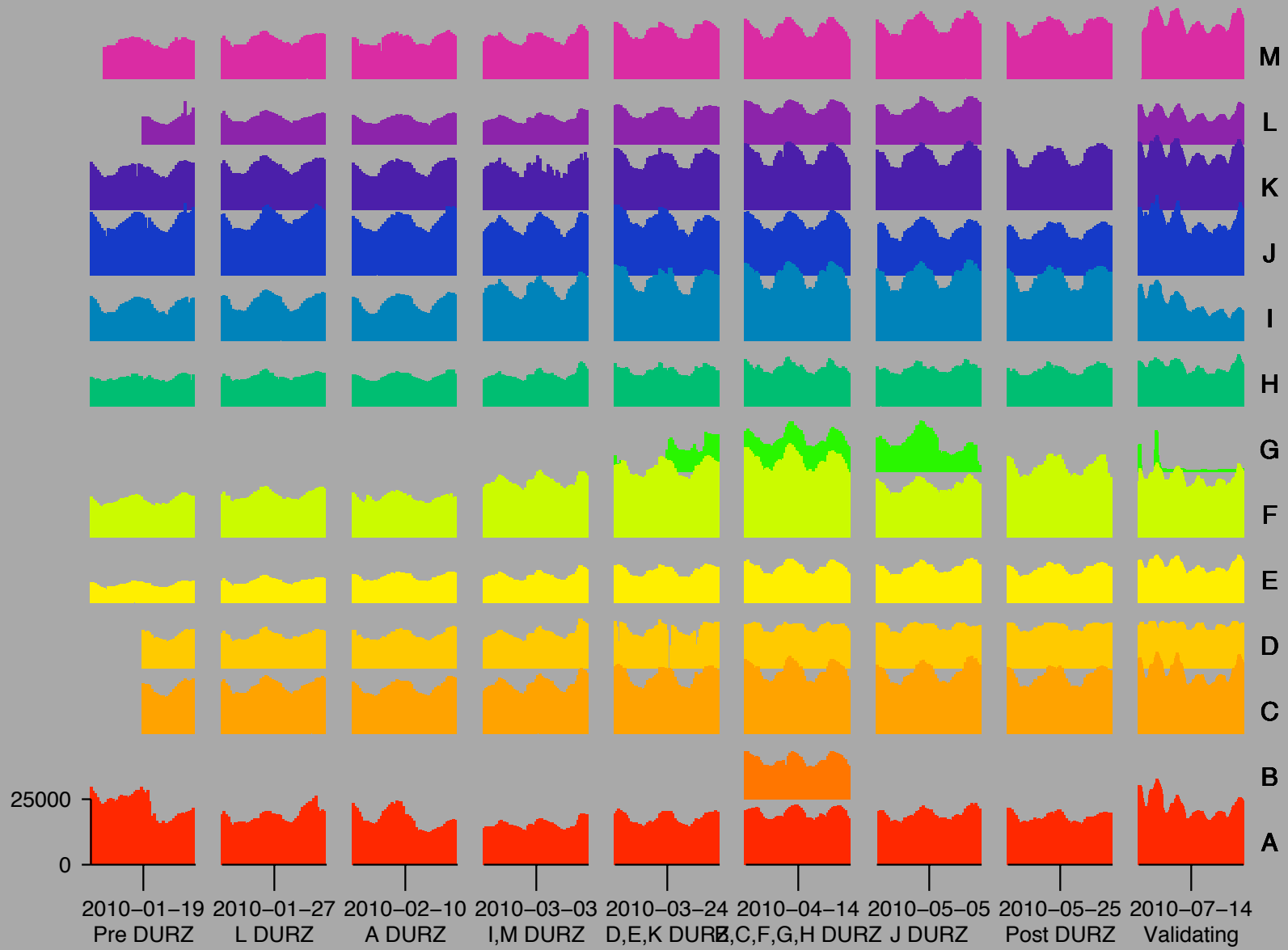
- Occasional attacks reflecting off the Root Name Servers
- First noticed in early 2010
- Attack load:
  - 6 kqps (January)
  - 15 kqps (December)
- Each “letter” of the Root Server system typically sees between 10 and 30 kqps
- Typical duration: 1--2 days
- Consistent query names

## Getting Data



- Normally difficult to get actual data from Root Operators
- We were adding DNSSEC to the Root in 2010.
  - Anyone remember the DURZ?
- Many Root Operators contributed data (full packet capture) to DNS-OARC during this time.
- Data is available to OARC members for analysis on OARC systems.

# Data Collections



# Looking for Attacks

- Extracted \$NAME1 and \$NAME2 queries from pcap files at OARC
- Wrote new pcap files, using dnscap -x
- Took a long time
  - It's a good thing OARC servers have many months uptime
- Resulted in about 275 GB

```
04:15:00.196679 IP 69.49.96.8.9544 > 198.41.0.4.53: 11645+ A? $NAME1. (23)
04:15:00.196715 IP 69.49.96.8.47402 > 198.41.0.4.53: 39784+ A? $NAME1. (23)
04:15:00.196792 IP 69.49.96.8.28022 > 198.41.0.4.53: 8708+ A? $NAME1. (23)
04:15:00.196951 IP 69.49.96.8.19023 > 198.41.0.4.53: 46782+ A? $NAME1. (23)
04:15:00.197082 IP 69.49.96.8.64610 > 198.41.0.4.53: 52331+ A? $NAME1. (23)
04:15:00.197140 IP 69.49.96.8.31253 > 198.41.0.4.53: 39260+ A? $NAME1. (23)
04:15:00.197189 IP 69.49.96.8.62176 > 198.41.0.4.53: 57977+ A? $NAME1. (23)
04:15:00.197295 IP 69.49.96.8.59463 > 198.41.0.4.53: 44059+ A? $NAME1. (23)
04:15:00.197395 IP 69.49.96.8.13442 > 198.41.0.4.53: 61958+ A? $NAME1. (23)
04:15:00.197489 IP 69.49.96.8.58885 > 198.41.0.4.53: 7953+ A? $NAME1. (23)
```

## Populated an SQL database

Column	Type	Modifiers
fid	integer	not null
unixtime	integer	not null
qname	text	not null
qtype	integer	not null
src	inet	not null
count	bigint	
eid	integer	

- Count the number of queries by name, type, and source at one-second intervals and store in SQL.



## Defining an Attack Event

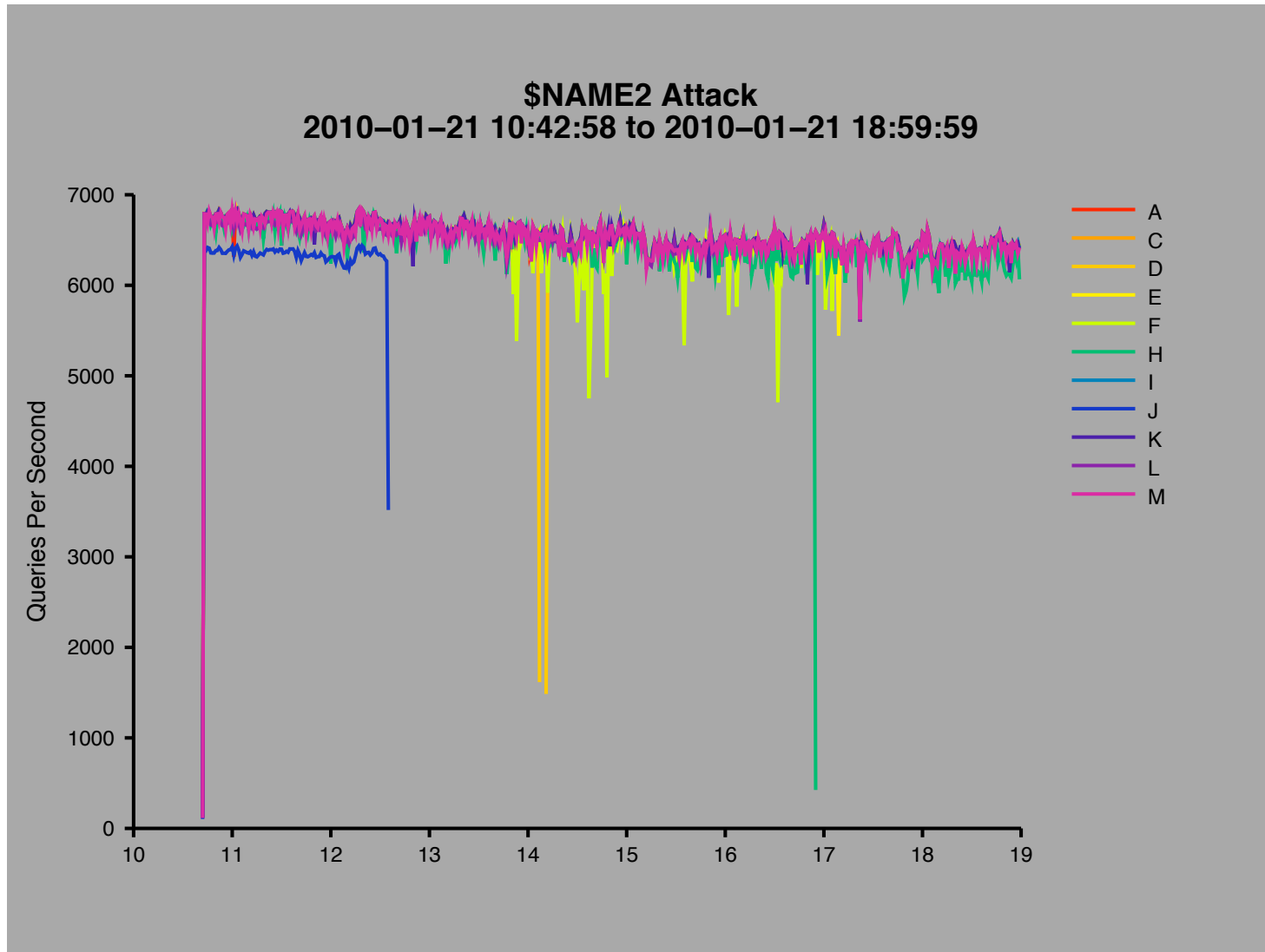


- Attack Event begins when a server sees at least 50 qps of queries for \$NAME1 or \$NAME2
- Attack Event ends when no server sees more than 50 qps
- A gap of 5 minutes or more marks the start of a new Attack Event

# Number of Attacks Captured

Collection Date	# attack events
2010-01-19	1
2010-01-26	1
2010-02-09	3
2010-03-02	1
2010-03-23	4
2010-04-13	0
2010-05-04	0
2010-05-25	0
2010-07-14	4

# Characterizing an Attack Event



# Characterizing Cont'd

Date	2010-01-21
Start	10:42:58
Stop	18:59:59
Duration	497 min
Max	7466 qps
Mean	5995 qps

Qname	Rate
\$NAME2	5995

Qtype	Rate
1	5995

Source	Rate
174.123.170.3	2393
174.123.170.5	5
174.123.170.6	2269
174.123.170.254	953
174.123.170.255	4



# Characterizing Cont'd



Server Node	Rate
a-root anr2-lax2-a	6570
c-root jfk1a.c.root-servers.org	4072
c-root jfk1b.c.root-servers.org	2498
d-root d-mon	6503
e-root crystal	5
e-root falcken	1262
e-root palace	5300
f-root f-lga1a	2745
f-root f-lga1b	3760
h-root H	6388
i-root was	6571
j-root evrsn2-bom1-j	1438
k-root sniffer.nap	6565
l-root dsc1.mia	6736
m-root MROOT-CDG	6529

Site locations are usually encoded in node names, often using airport codes (LAX, JFK, LGA).

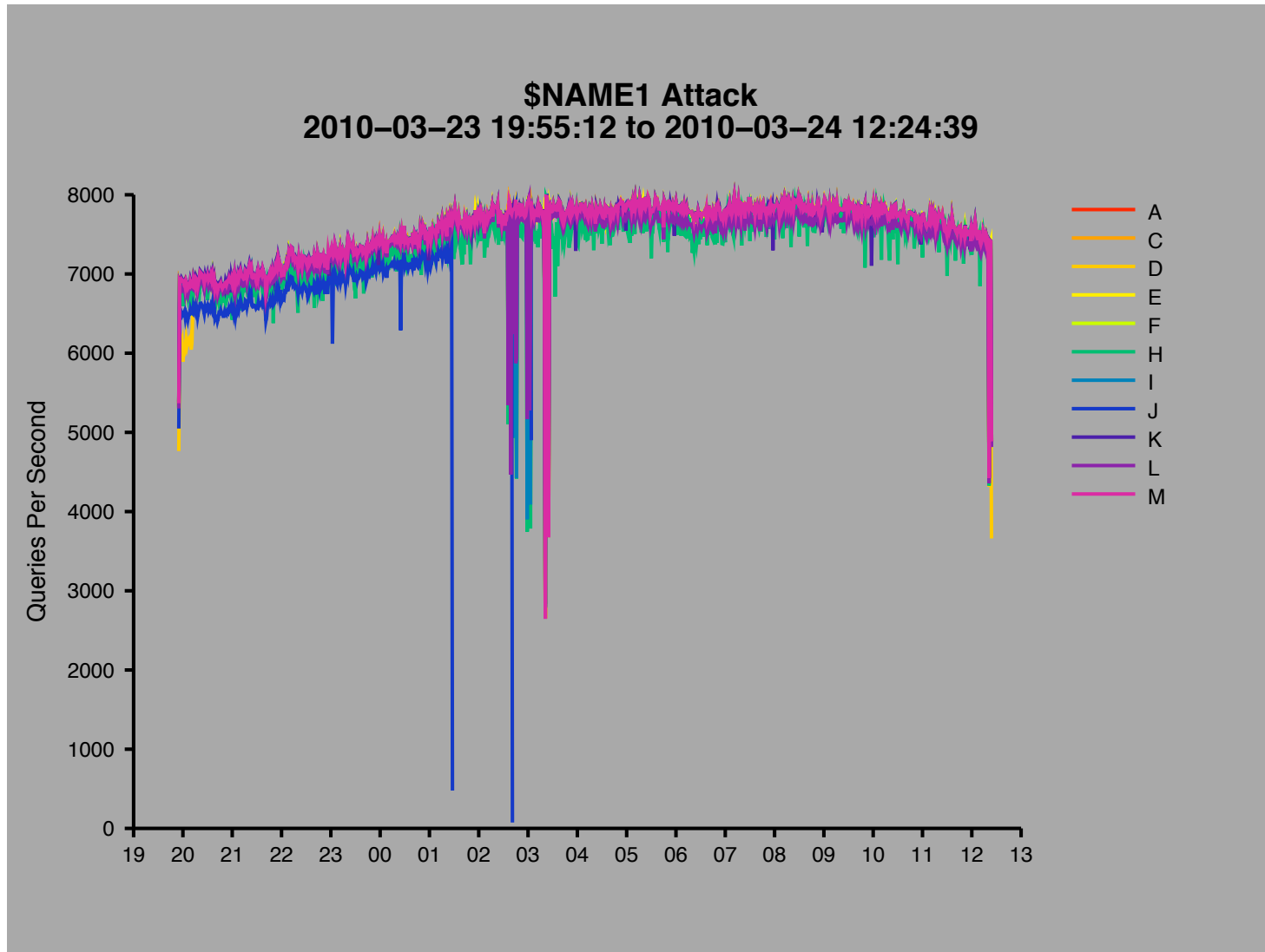
Each data provider chooses their own naming scheme.

# Observations







- Most of the other attacks look very similar
- Always \$NAME1 or \$NAME2 for the query name
- Always a small number of sources
  - Assume they are spoofed
  - But sources change for each Attack Event
- Always query type 1 (A)
- A lot of consistency in servers that see the attack traffic
  - US East Coast
  - Europe
- Attack affects some operators ability to collect data
  - J-root (Mumbai)

# This one was different though

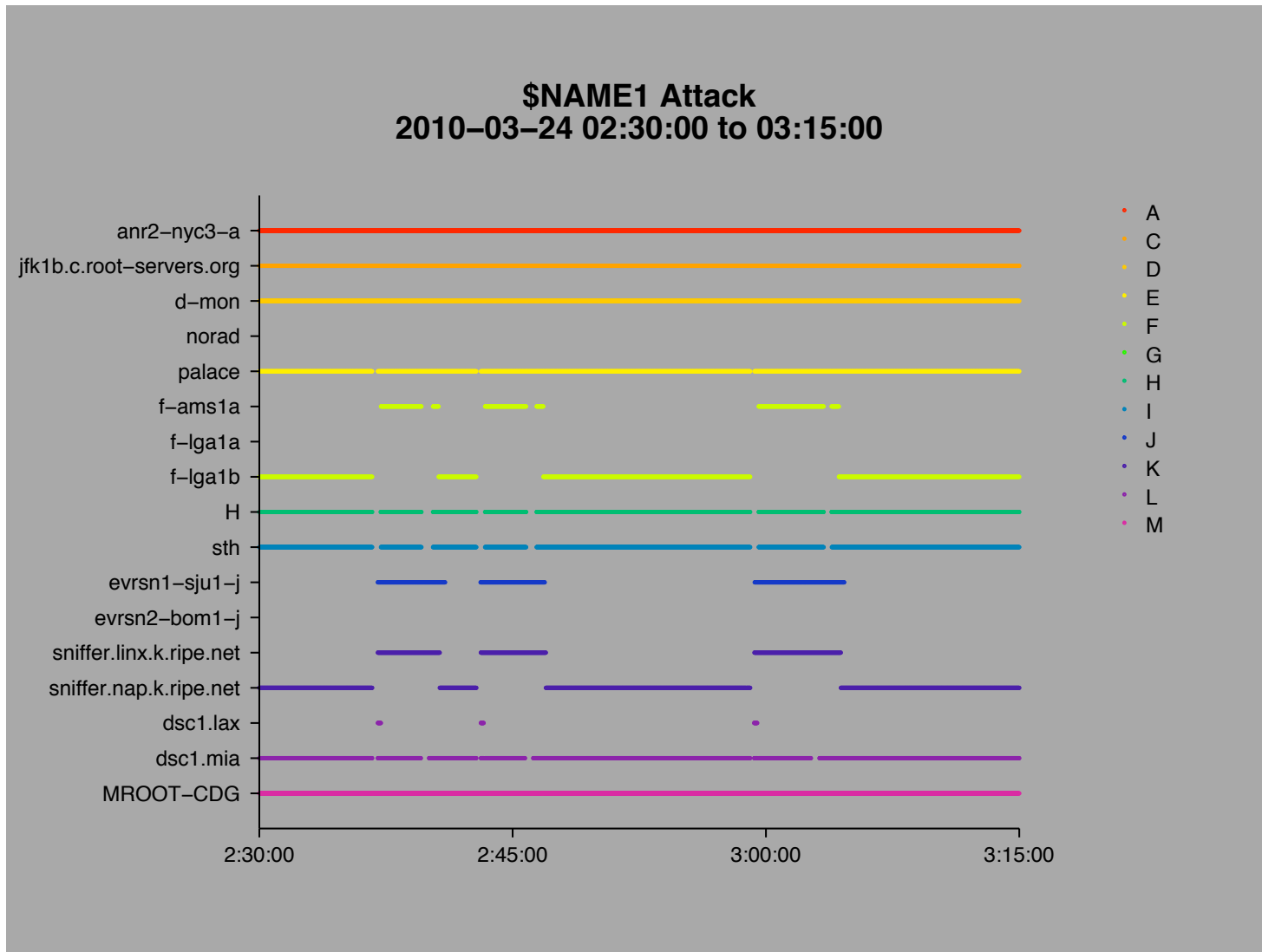


# 2010-03-23 Reflectors

Server Node	Rate
a-root anr2-nyc3-a	7558
c-root jfk1b.c.root-servers.org	7555
d-root d-mon	7537
e-root norad	21
e-root palace	7530
 f-root f-ams1a	78
f-root f-lga1a	21
f-root f-lga1b	7433
h-root H	7368
i-root sth	7532
 j-root evrsn1-jsu1-j	103
i-root evrsn2-bom1-j	2302
 k-root sniffer.linx	100
k-root sniffer.nap	7455
 l-root dsc.lax	2
l-root dsc.mia	7442
m-root MROOT-CDG	7566



# A Closer Look at Nodes Receiving Queries

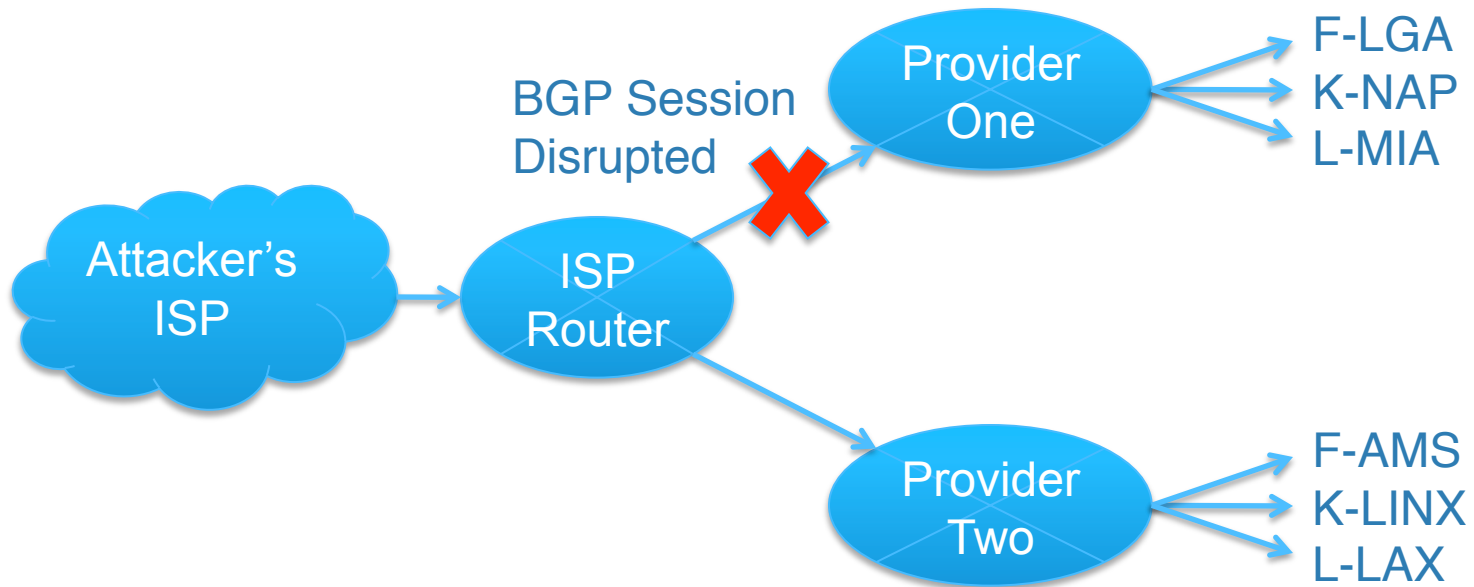


## What Have We Here?



- We found a “glitch” in which attack traffic to three anycast-enabled servers shifted at exactly the same time.
- We assume the shift is due to a change in routing topology. (what else could it be?)
- We assume this glitch happened close to the true source of the attack because of the simultaneity.
  
- Wouldn't normal traffic going through the glitch point be affected in exactly the same way??

# The Model



# A Fingerprint?

- Let's use the shift in anycast traffic to look for other sources which shifted in exactly the same way, at exactly the same time.
- For example here are times when F-root nodes received queries during the glitch:

Node	From	To
lga1	02:30:00	02:36:41
ams1	02:37:12	02:40:37
lga1	02:40:37	02:42:51
ams1	02:43:22	02:46:49
lga1	02:46:49	02:59:04
ams1	02:59:34	03:04:19
lga1	03:04:19	03:14:59

# Fingerprints Found



F-root		K-root		L-root	
Source	Count	Source	Count	Source	Count
69.49.96.8	19515354	69.49.96.8	20646817	69.49.96.8	19760819
96.9.156.4	1479	66.96.224.89	4747	96.9.156.4	1311
96.9.156.5	1457	64.120.132.244	1463	96.9.156.5	1283
96.9.142.101	535	64.120.132.245	1463	96.9.142.101	91
66.197.185.149	138	96.9.156.4	1057	173.212.215.218	47
96.9.131.165	89	96.9.156.5	995	64.191.50.173	38
64.120.163.117	67	96.9.142.101	136	64.191.6.148	23
173.212.242.178	54	96.9.139.229	73	66.197.250.165	22
66.197.187.117	54	64.120.174.37	71	96.9.176.5	14
64.191.46.21	52	173.212.242.178	64	204.124.182.119	7
66.197.212.165	50	66.197.160.85	61	144.126.1.10	4
64.120.165.149	48	64.120.167.98	54		
64.191.50.93	36	64.120.165.149	46		
173.212.215.218	35	173.212.215.218	46		
96.9.185.165	34	64.191.78.21	42		
66.197.209.21	34	204.124.183.221	39		
64.191.37.149	34	66.197.247.101	33		
66.96.216.149	34	64.120.227.74	33		
64.191.76.53	32	64.191.50.173	32		
66.197.204.70	32	64.191.37.149	31		
...	...	...	...		


# Netblocks

- Nearly all of the addresses found by the fingerprint search are within these 6 netblocks:

IP	BGP Prefix	AS	AS Name
66.96.224.89	66.96.192.0/18	21788	NOC – Network Operations Center Inc.
66.197.212.165	66.197.128.0/17	21788	NOC – Network Operations Center Inc.
64.120.165.149	64.120.128.0/18	21788	NOC – Network Operations Center Inc.
64.191.50.173	64.191.0.0/17	21788	NOC – Network Operations Center Inc.
96.9.142.101	96.9.128.0/18	21788	NOC – Network Operations Center Inc.
173.212.242.178	173.212.192.0/18	21788	NOC – Network Operations Center Inc.

# Whois

```
OrgName:      Network Operations Center Inc.
OrgId:        NOC
Address:      PO Box 591
City:         Scranton
StateProv:    PA
PostalCode:   18501-0591
Country:      US
RegDate:      2001-04-04
Updated:      2010-03-30
Comment:      Abuse Dept: abuse@hostnoc.net
Ref:          http://whois.arin.net/rest/org/NOC
```



Also known as BurstNET, which advertises managed servers, co-location, and VPS hosting.

# Confirmation



- Do AS21788's sources always hit the same servers as attack traffic?
- For each anycast root server, and for each attack event, let's calculate the fraction of AS21788's legitimate traffic going to the same anycast sites as the attack traffic.
- Its nearly a 100% match...



# Confirmation



		20100323 #1 Attack							
Net		20100714 #1 Attack							
	Net	a-root	c-root	f-root	j-root	k-root	l-root	root	root
64.12	64.120.128.0/17	1.00	1.00	1.00	0.97	1.00	1.00	1.00	1.00
66.96	64.191.0.0/17	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
66.19	66.96.192.0/18	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96.173	66.197.128.0/17	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
173.184	96.9.128.0/18	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
184.184	173.212.192.0/18	1.00	1.00	1.00	-	1.00	1.00	1.00	1.00
184.96	184.22.0.0/16	-	-	-	-	-	-	-	-
96.173	184.82.0.0/16	1.00	1.00	1.00	1.00	1.00	1.00	-	-
173.184	66.197.128.0/17	1.00	1.00	1.00	-	1.00	1.00	1.00	1.00
184.184	96.9.128.0/18	1.00	1.00	1.00	-	1.00	1.00	-	1.00
184.96	173.212.192.0/18	1.00	1.00	1.00	-	1.00	1.00	-	1.00
184.96	184.22.0.0/16	-	-	-	-	-	-	-	-
184.96	184.82.0.0/16	-	-	-	-	-	-	-	-

## Why This Technique Worked



- Diverse set of servers (the root server letters)
- Many of them highly anycast
- Apparent routing glitch close to the source
  - Self inflicted?
- High quality data collection for post-mortem analysis
  - Could it be done in real-time?



# Questions?