Conficker
Conflicker
Confnicker

Emile Aben, CAIDA

early 21st century

Outline

- UCSD network telescope
- Data sharing
- Conficker

- DoS Backscatter data (<- original subject)

UCSD Network Telescope

- /8 almost darknet that we capture pcap on
  - high noise (good) to signal (legit traffic) ratio
  - sees Internet background radiation:
    - scanning (worms and other),
    - DoS backscatter (spoofed),
    - misconfigurations

UCSD Network Telescope

- /8 almost darknet that we capture pcap on
  - high noise (good) to signal (legit traffic) ratio
  - sees Internet background radiation:
    - scanning (worms and other),
    - DoS backscatter (spoofed),
    - misconfigurations
- no responders/honey-whatevers, but
  - every disadvantage has its advantage
    - *http://en.wikiquote.org/wiki/Johan_Cruijff (Dutch philosopher)*
    - *eg. observation of tcp-retransmits*

UCSD Network Telescope

- /8 almost darknet that we capture pcap on
  - high noise (good) to signal (legit traffic) ratio
  - sees Internet background radiation:
    - scanning (worms and other),
    - DoS backscatter (spoofed),
    - misconfigurations
- no responders/honey-whatevers, but
  - every disadvantage has its advantage
    - *http://en.wikiquote.org/wiki/Johan_Cruijff (Dutch philosopher)*
    - *eg. observation of tcp-retransmits*
- we continue to work making this a community infrastructure
  - happy to hear your ideas (protected data sharing)

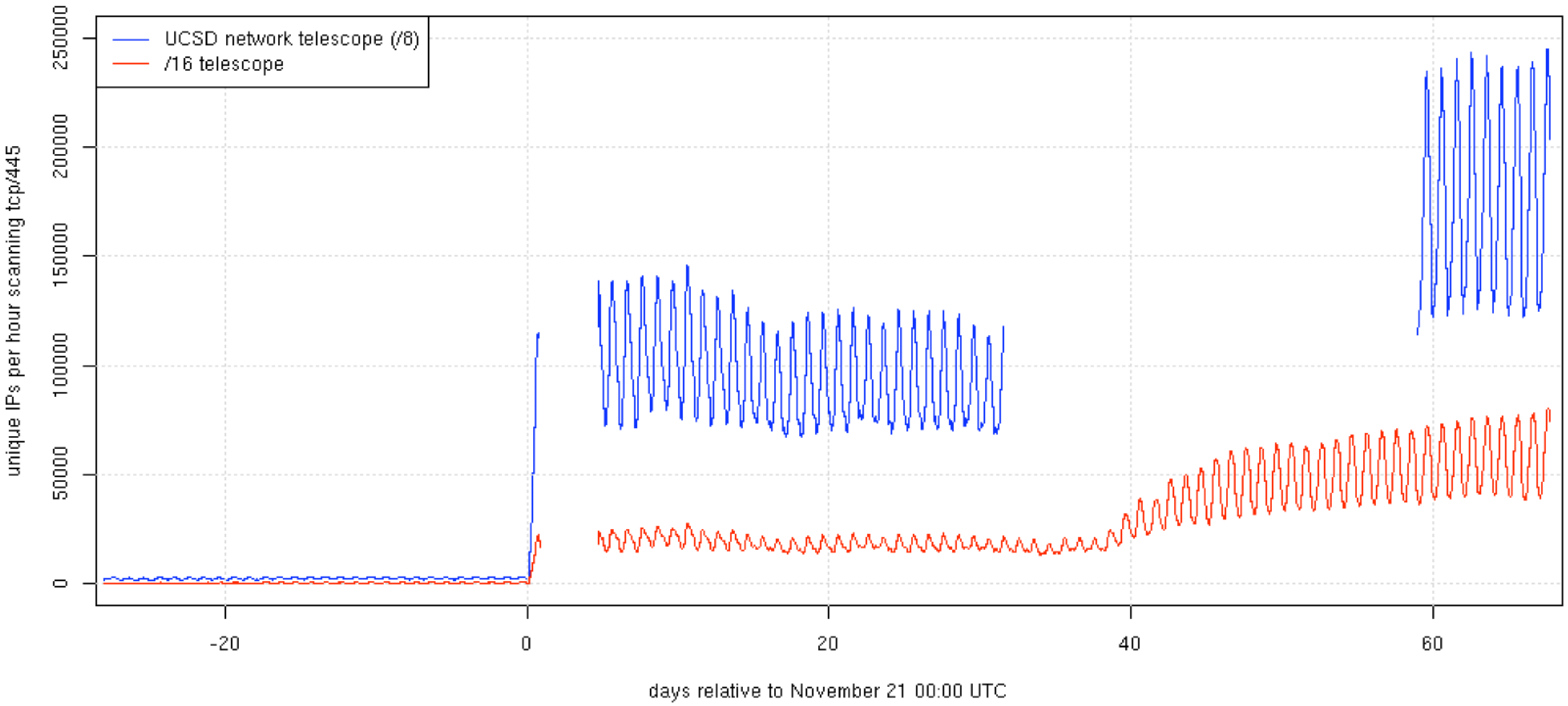
Data sharing efforts

- PREDICT project (DHS)
  - http://www.predict.org/
  - data sharing effort created to help to protect and defend cyber infrastructure
  - CAIDA provides data from:
    - Internet backbone links (“tier1”)
    - active measurements (traceroute-like, topology)
    - UCSD network telescope
      - DoS Backscatter
      - Worm data (Witty, Code-Red, ..)

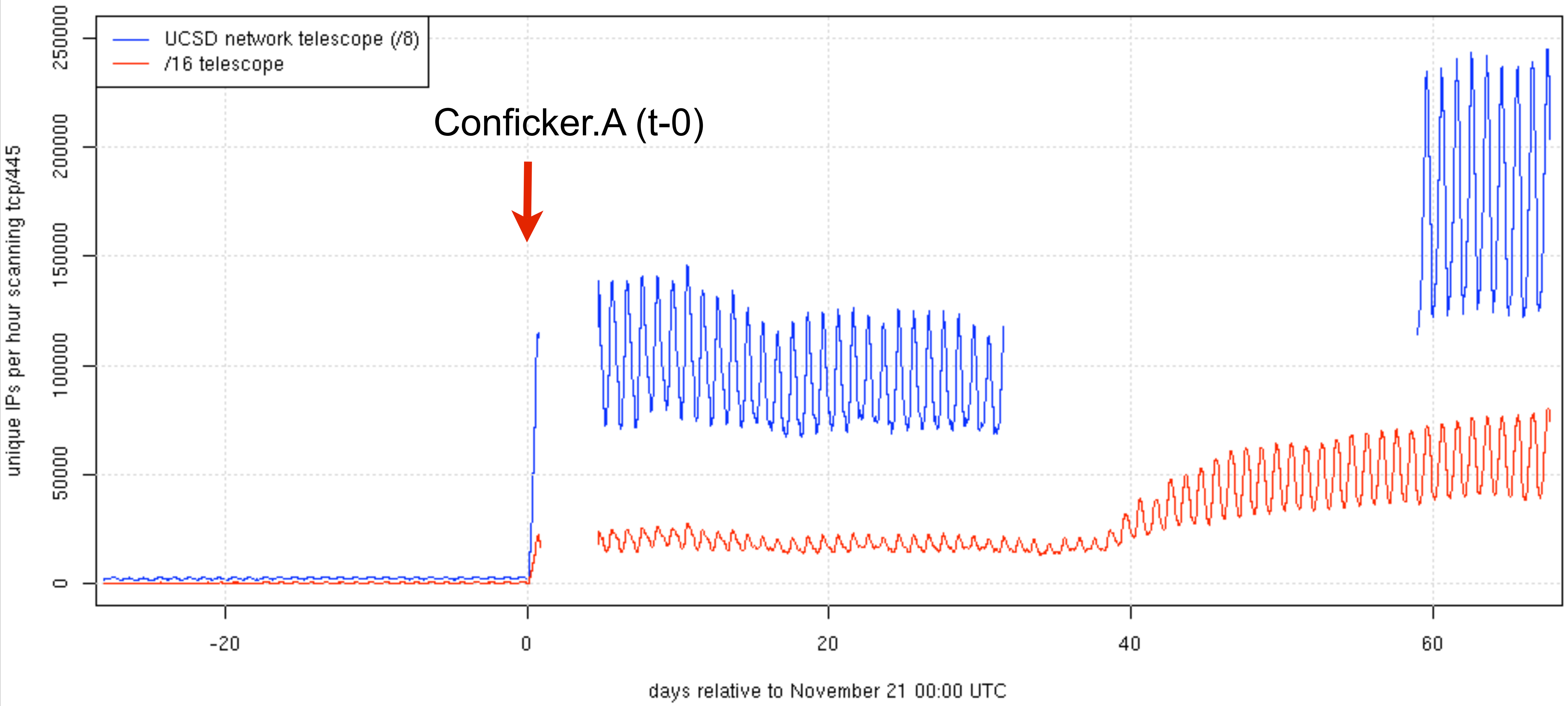
MS08-067 / Conficker

- some observable behavior:
    - 3 types of TCP/445 scanning (local network, “random”, around infected hosts)
    - HTTP “phone-home” to 250 different domains per day
- “random” TCP/445 scanning expected to show up on darknets
- noisy signal: background of other MS-RPC exploits, but significant differences between pre- and post-conflicker

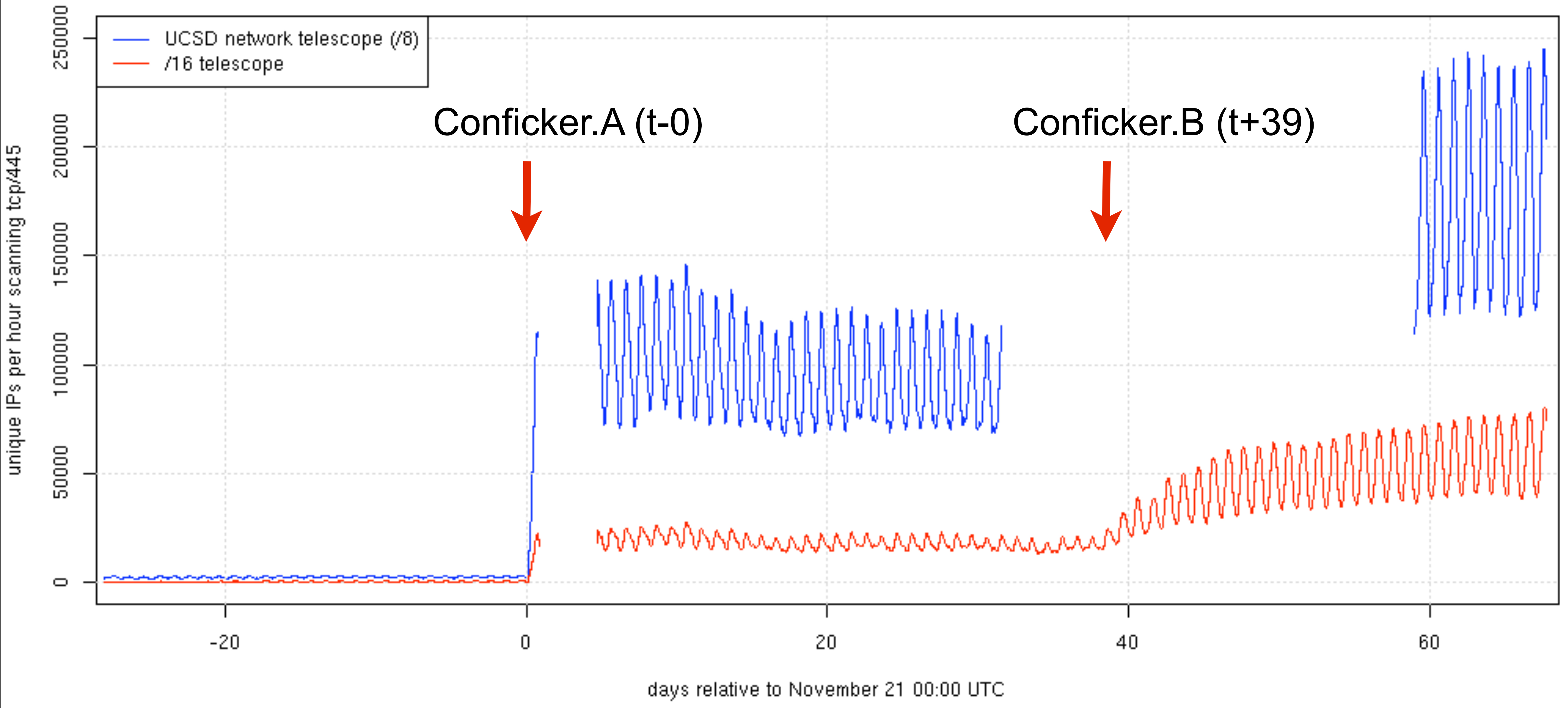
srcIPs/hour scanning TCP/445



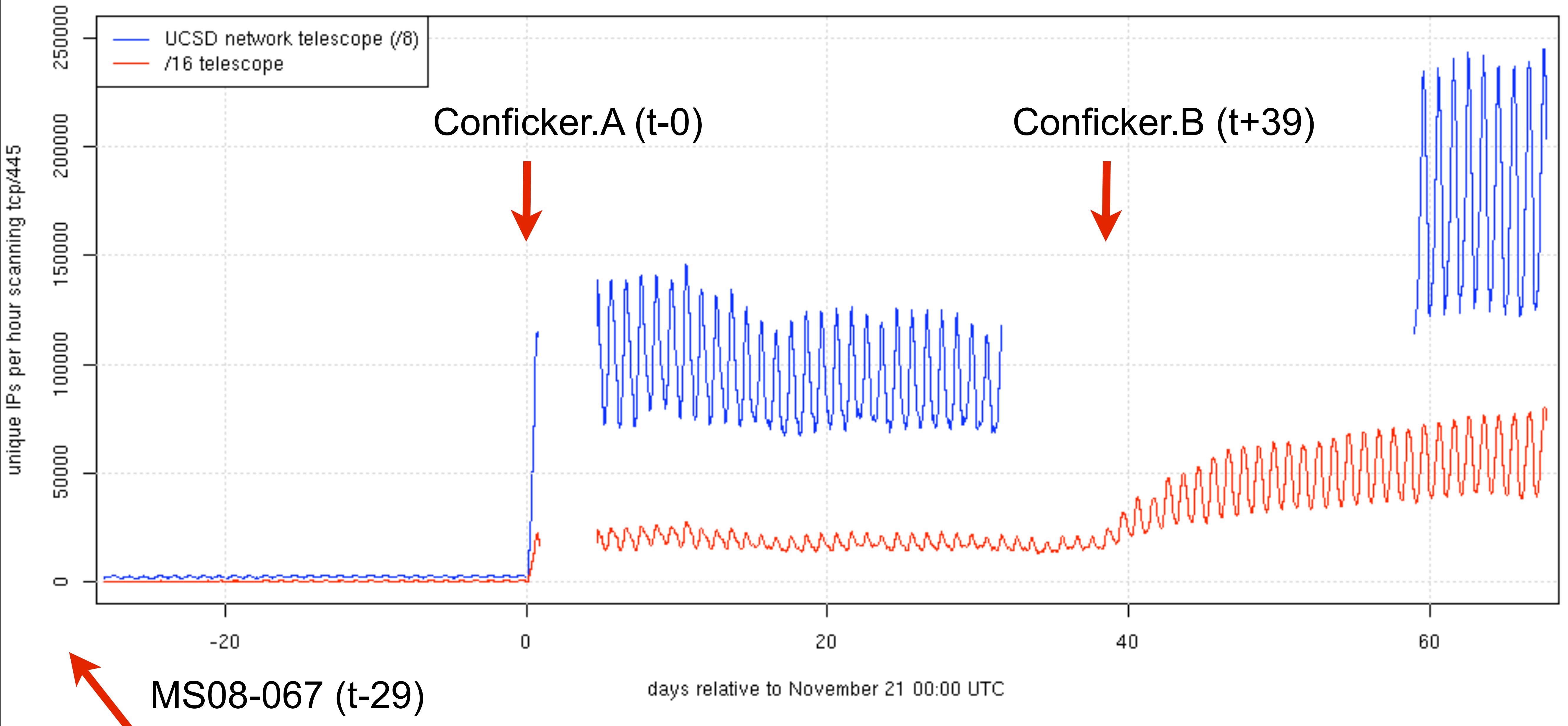
srcIPs/hour scanning TCP/445



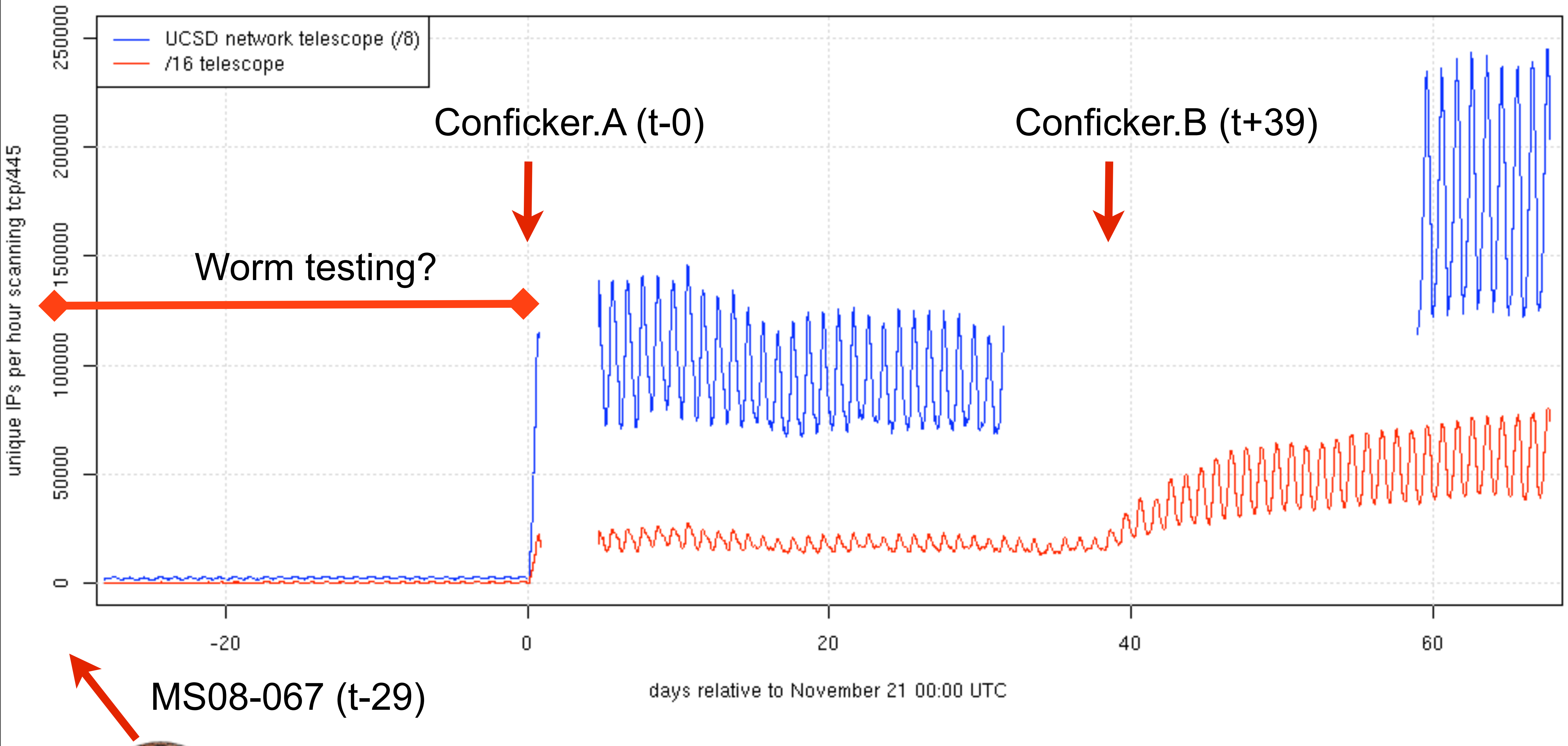
srcIPs/hour scanning TCP/445



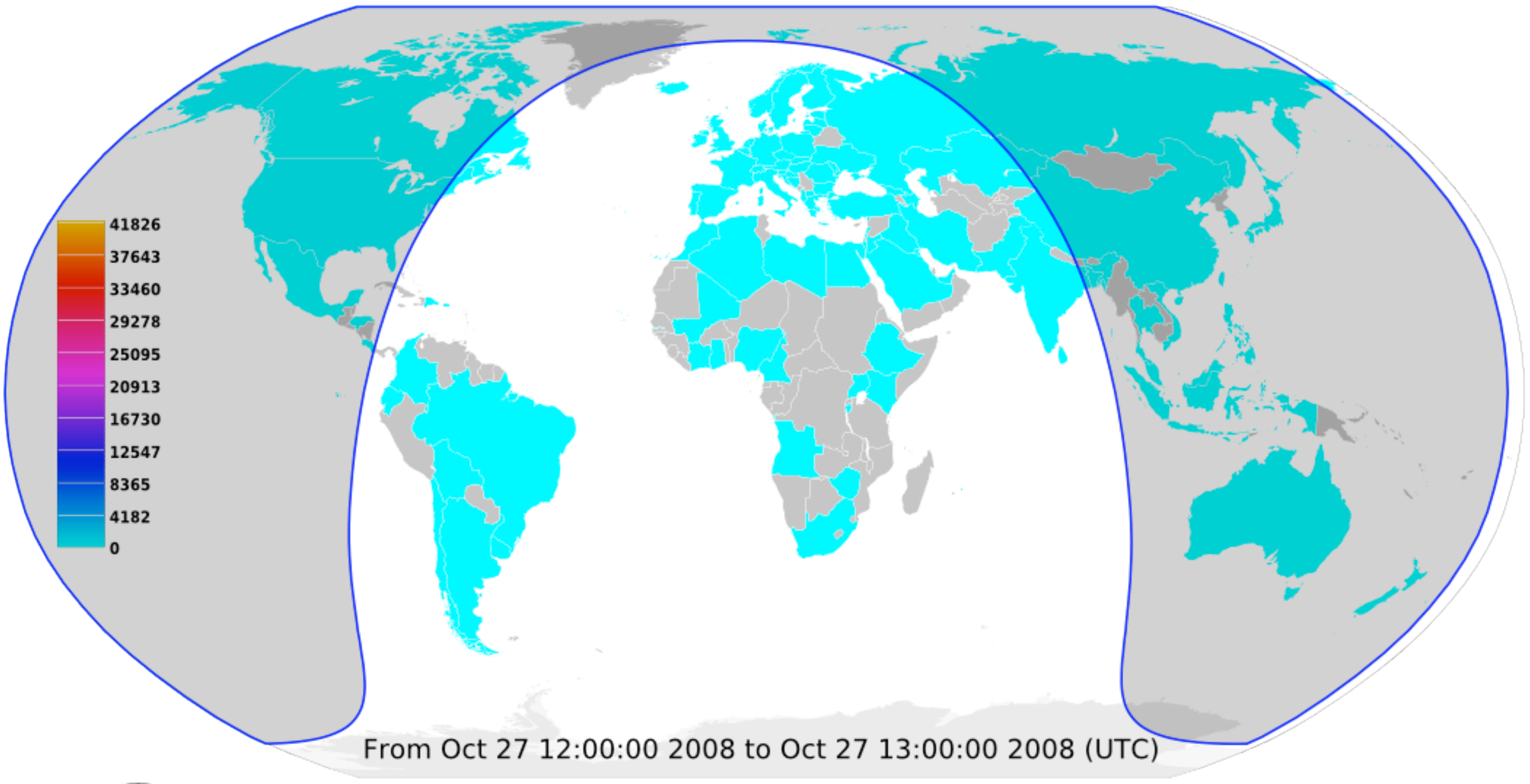
srcIPs/hour scanning TCP/445



srcIPs/hour scanning TCP/445

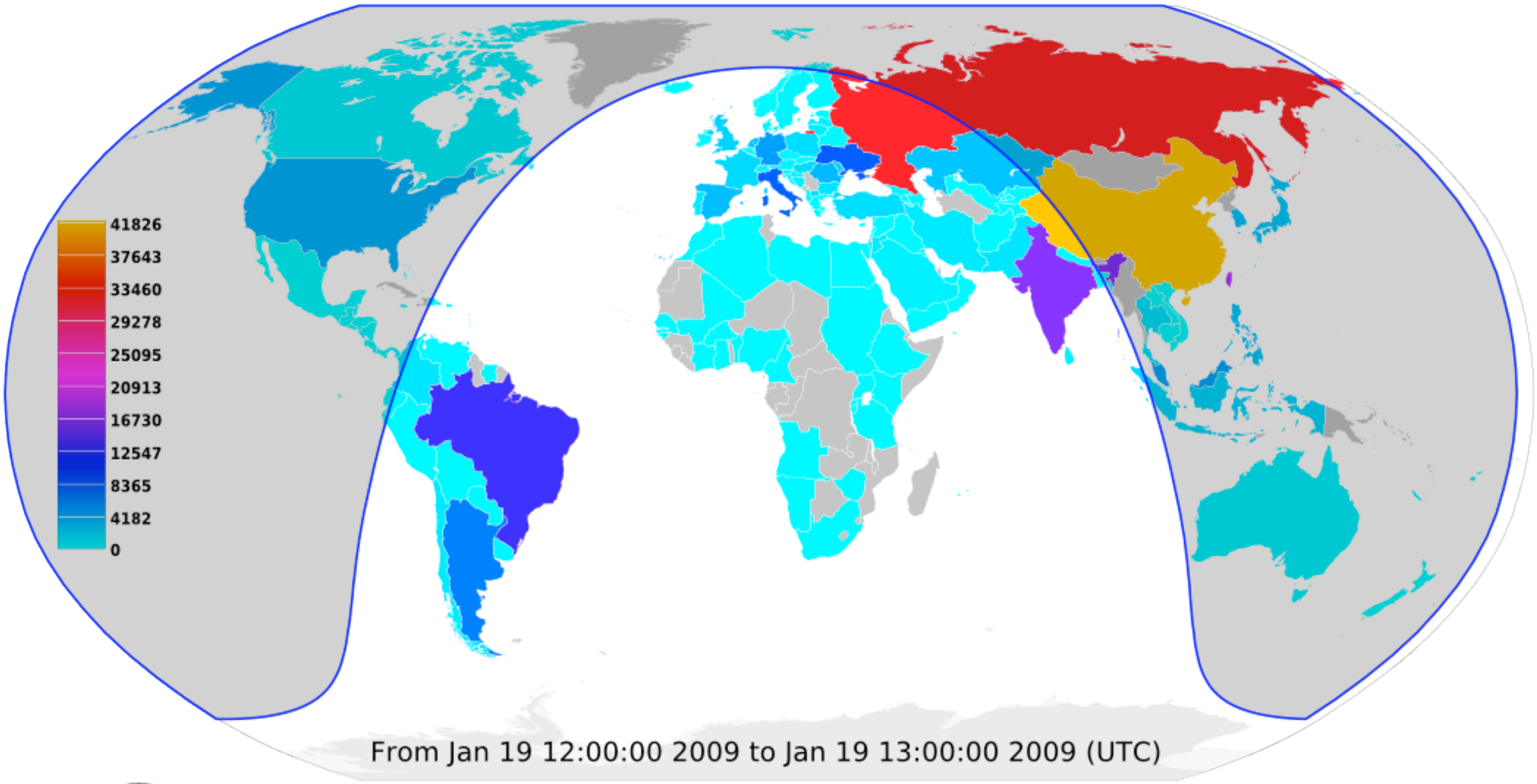


Unique IPs scanning on TCP/445 2008-10-27 (pre conficker)



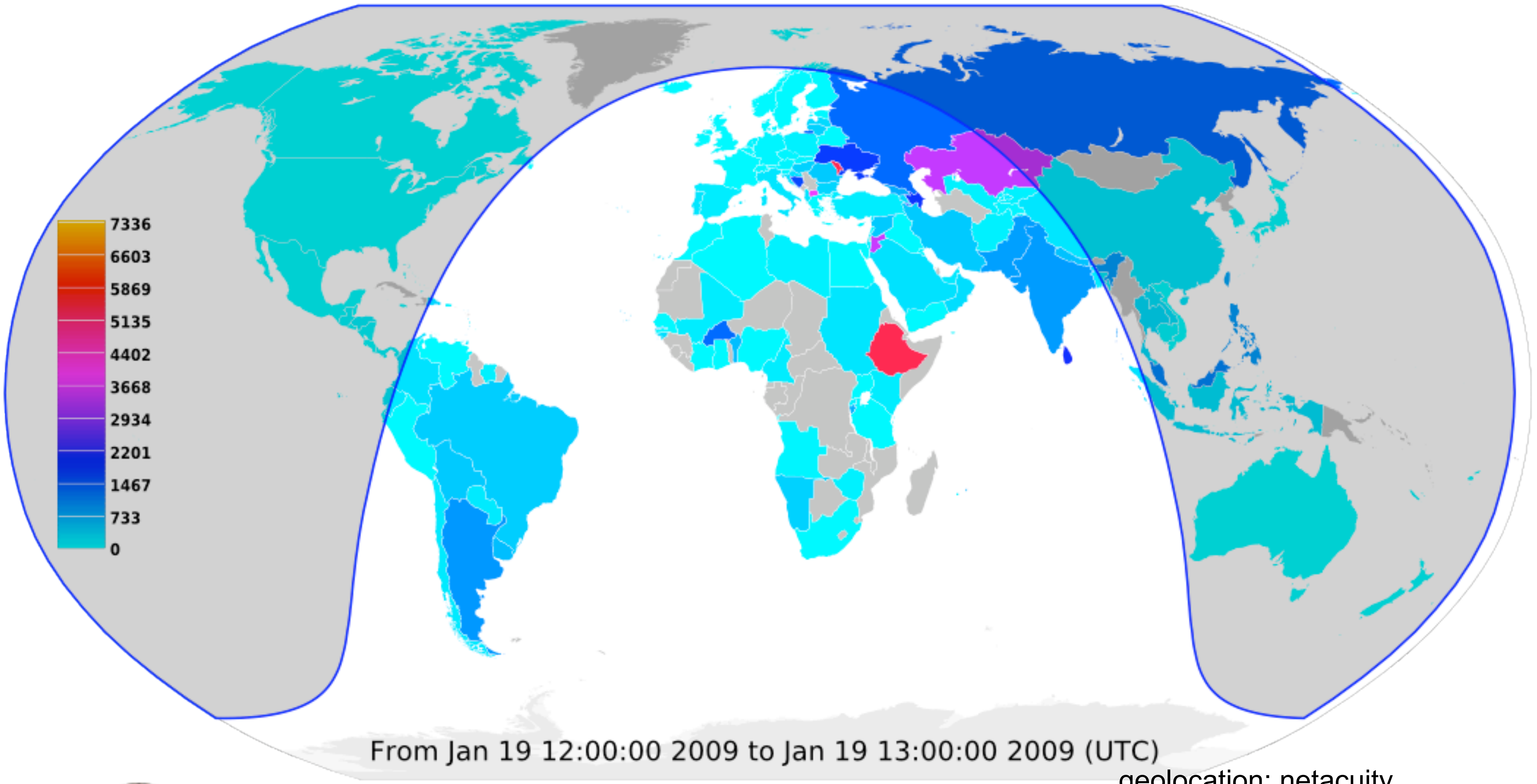
geolocation: netacuity
visualization: Sebastian Castro

Unique IPs scanning on TCP/445 2009-01-19 (post conficker)



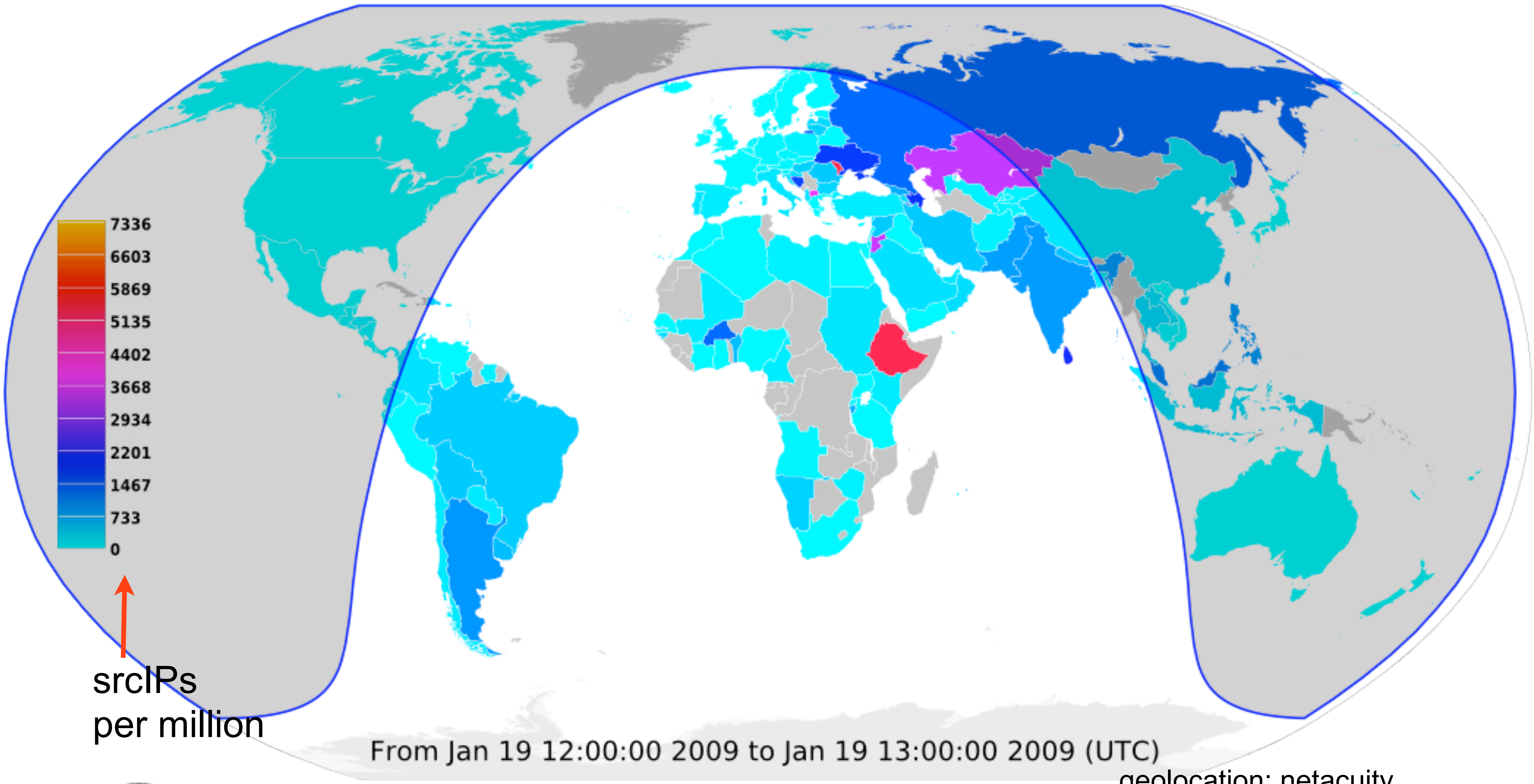
geolocation: netacuity
visualization: Sebastian Castro

Normalized by country IP pool scanning on TCP/445 2009-01-19



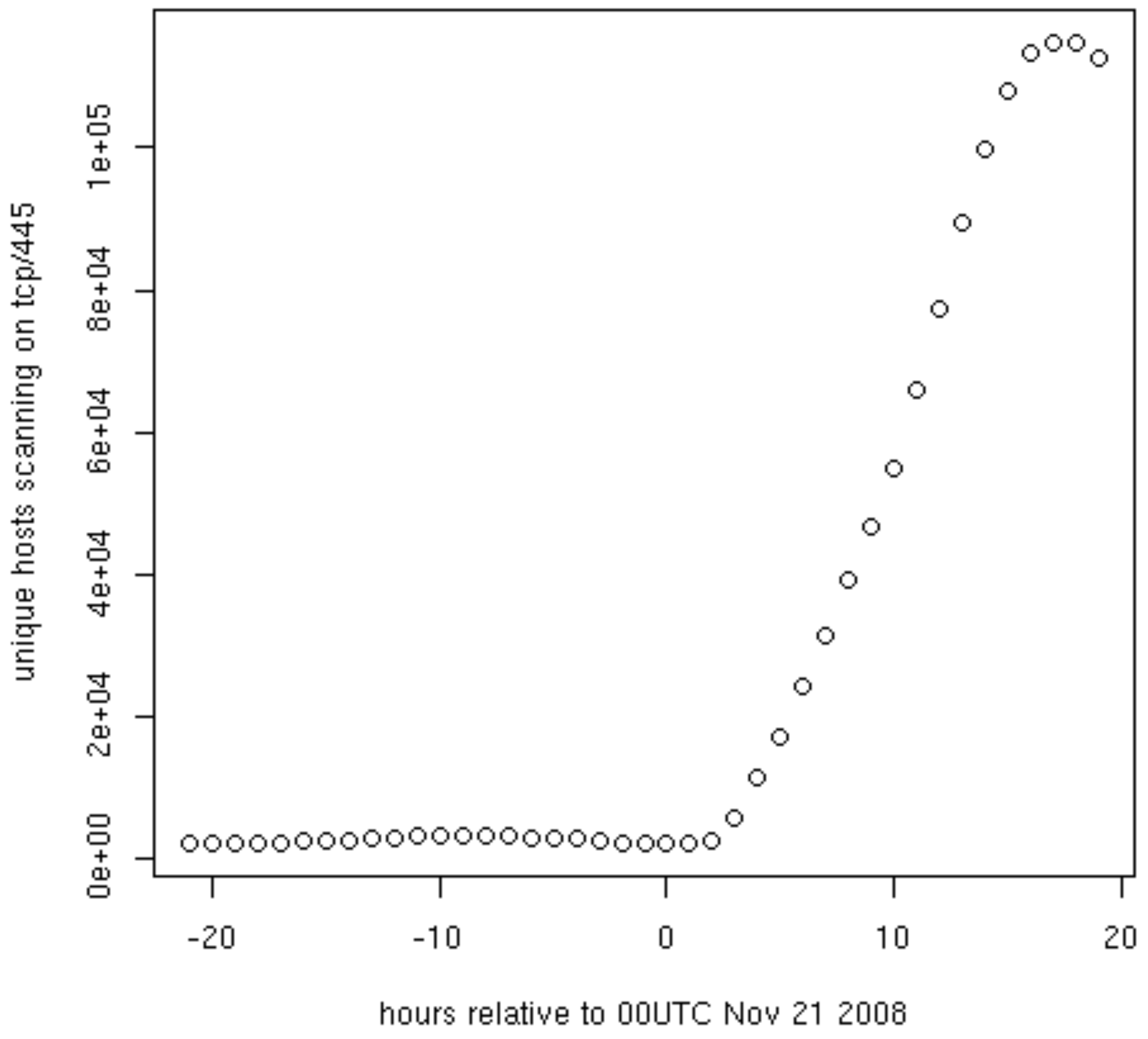
geolocation: netacuity
IPs/country: maxmind
visualization: Sebastian Castro

Normalized by country IP pool scanning on TCP/445 2009-01-19

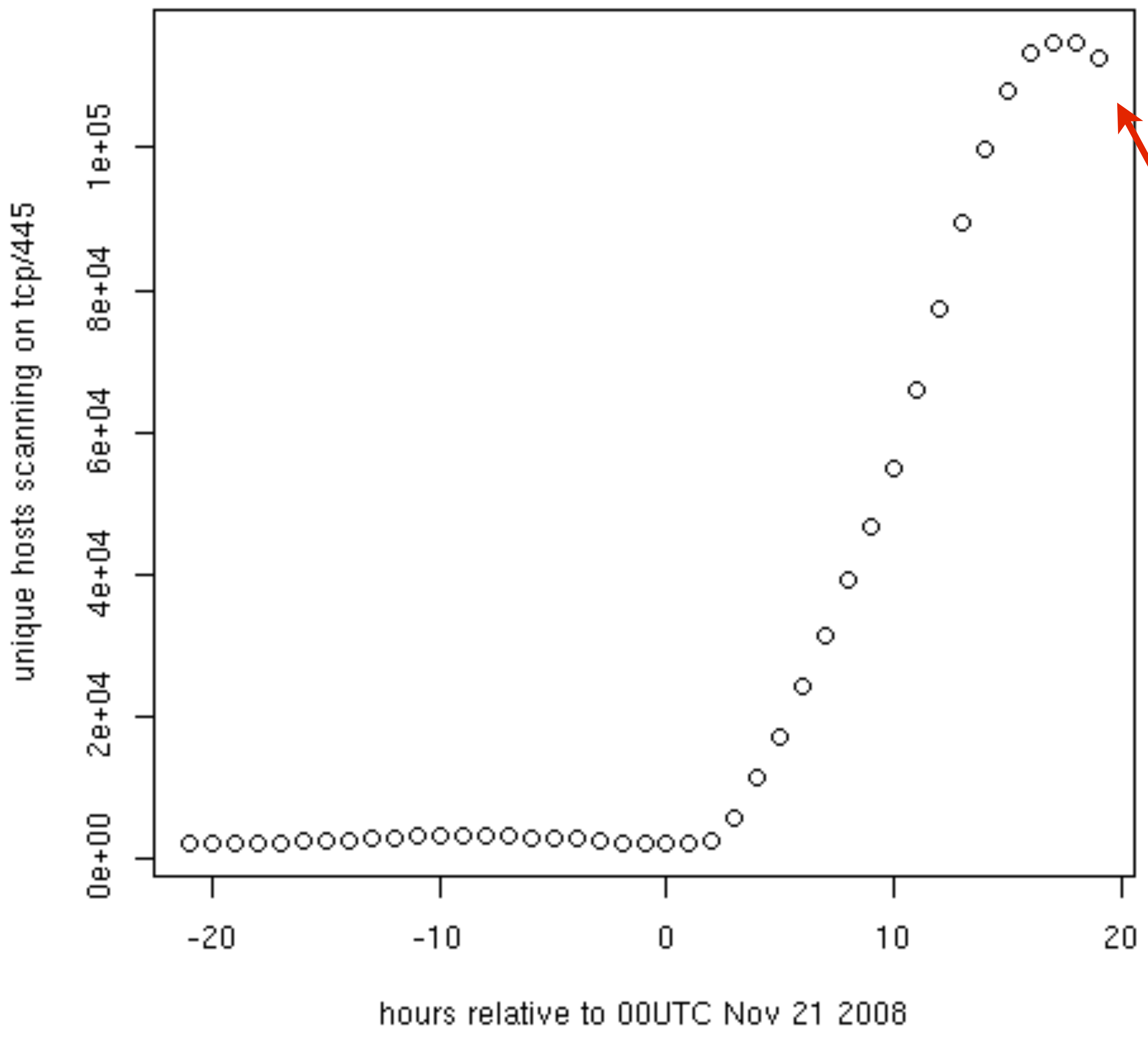


geolocation: netacuity
IPs/country: maxmind
visualization: Sebastian Castro

Conficker A lift-off

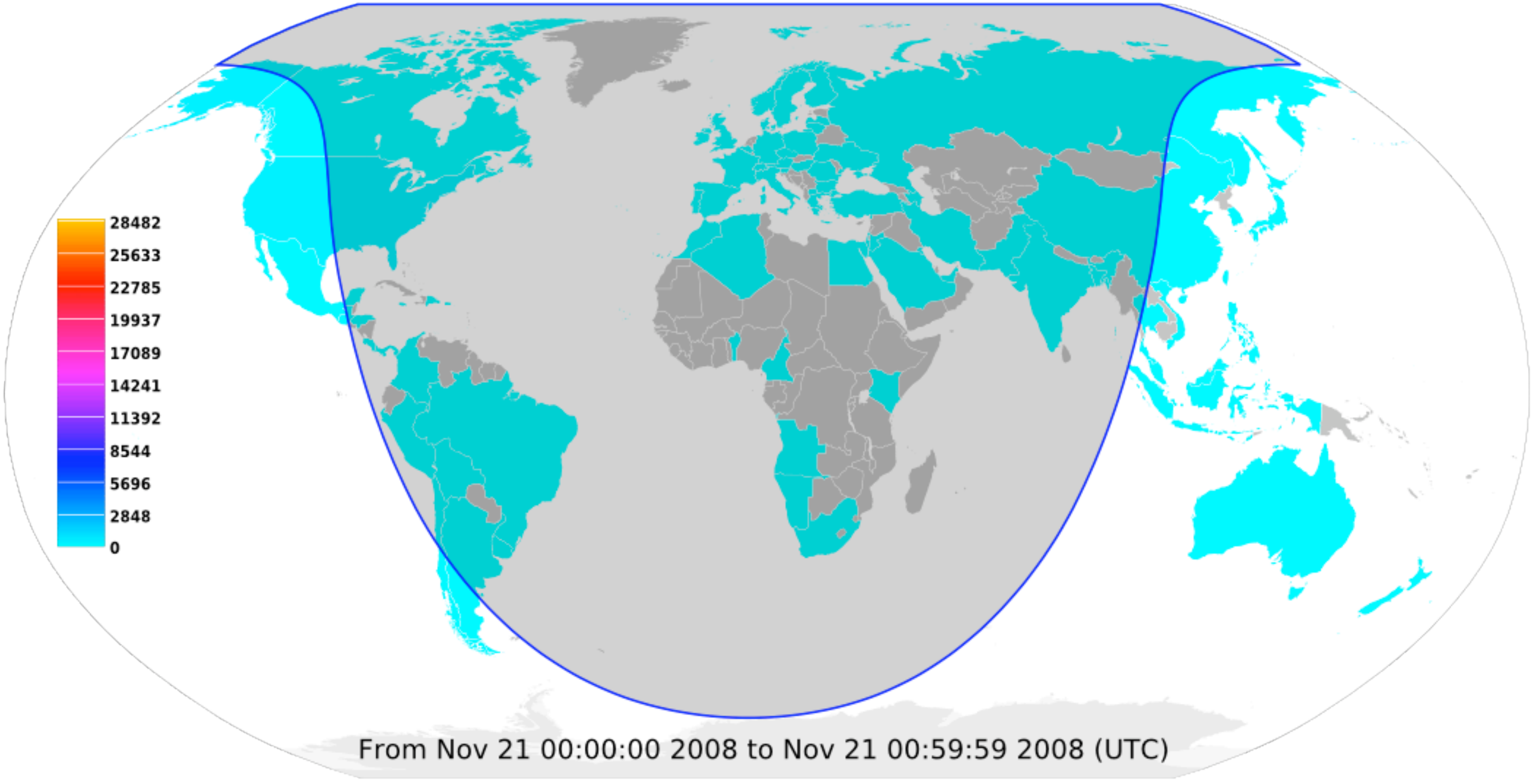


Conficker A lift-off



our data collection stops here :(

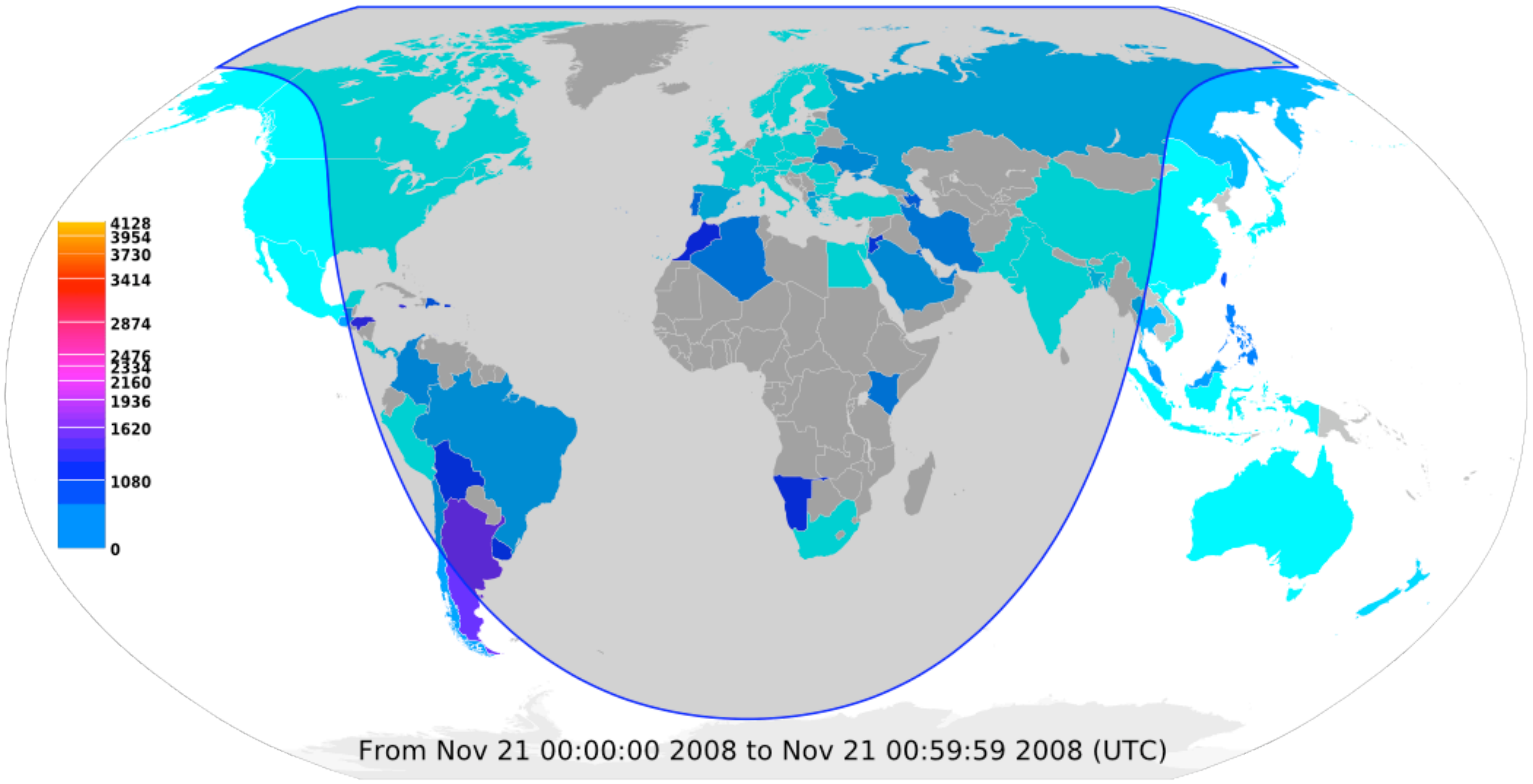
Unique IPs scanning on TCP/445 2008-11-21 (t-0)



animation at:
http://www.caida.org/~emile/conficker-telescope/
telescope.tcp445.nov21.linear.animated.gif

geolocation: netacuity
IPs/country: maxmind
visualization: Sebastian Castro

same normalized (only countries with 100k+ IPs)



animation at:
http://www.caida.org/~emile/conficker-telescope/
telescope.tcp445.nov21.norm.log.animated.gif

geolocation: netacuity
IPs/country: maxmind
visualization: Sebastian Castro

Typical scan

- tcp-retransmit (only if you don't respond)
- incrementing IPID (makes pps estimate for host possible)
- incrementing source port (makes connections/per second estimate for host possible)

```
#if          time      src addr          dst addr          len  pro  ts  ip.id  ttl  sport  dport  tcp.seq   tcp.ack   flags
 0   1232324970.000000 X.X.X.X           X.102.106.33       48    6  00   1541  100   1209    445  b8b021a8  -         -S----
 0   1232324973.000000 X.X.X.X           X.102.106.33       48    6  00   1643  100   1209    445  b8b021a8  -         -S----
 0   1232324978.000000 X.X.X.X           X.67.245.42        48    6  00   1856  109   1303    445  3cace550  -         -S----
 0   1232325014.000000 X.X.X.X           X.105.36.120       48    6  00   3053  109   1643    445  2ac65d2e  -         -S----
 0   1232325017.000000 X.X.X.X           X.105.36.120       48    6  00   3109  109   1643    445  2ac65d2e  -         -S----
 0   1232325017.000000 X.X.X.X           X.76.162.11        48    6  00   3113  109   1647    445  49dfb8bc  -         -S----
 0   1232325017.000000 X.X.X.X           X.84.188.57        48    6  00   3116  109   1650    445  30e40077  -         -S----
 0   1232325020.000000 X.X.X.X           X.121.233.40       48    6  00   3179  101   1693    445  538dcd2c  -         -S----
 0   1232325023.000000 X.X.X.X           X.121.233.40       48    6  00   3235  101   1693    445  538dcd2c  -         -S----
 0   1232325036.000000 X.X.X.X           X.5.253.93         48    6  00   3543  100   1855    445  5a488859  -         -S----
 0   1232325053.000000 X.X.X.X           X.97.151.88        48    6  00   3887  109   2006    445  8e2650e0  -         -S----
<continues>
```

28 pps avg   9 cps avg

Quick and dirty looking for non-randomness in scanning

- For each scanning IP, look at probability each bit the destination IP address is set, example:

```
 octet2 . octet3 . octet4

01010011.00000000.00011001
01101001.10101010.00001100
00011101.00000000.01110101
00010010.01010101.01110001
01000010.11010101.00010010
01110001.10001100.00010010
00011100.10001010.01001001
00110001.01100111.01001001
01000110.01110001.01001001
00000000.10000000.01000000

05363345.54333434.06255226
```

Patterns pre and post

- patterns in dst IPs (per src IP scanning tcp/445 and sending > 100 pkts)

```
      2008-10-27 00UTC (pre)
count octet2   .octet3   .octet4
  900
  516 --------.--------.--------
   27 0-------.--------.0-------
    5 ---1-1-1.--------.--------
    4 10011011.01110101.11101010
      <continues>
```

```
0 = 0 (in >99% of pkts)
1 = 1 (in >99% of pkts)
- = “random”
```

Patterns pre and post

- patterns in dst IPs (per src IP scanning tcp/445 and sending > 100 pkts)

```
      2008-10-27 00UTC (pre)
count octet2   .octet3  .octet4
  900
  516 --------.--------.--------
   27 0-------.--------.0-------
    5 ---1-1-1.--------.--------
    4 10011011.01110101.11101010
      <continues>
```

```
          2009-01-19 00UTC  (post)
count octet2   .octet3  .octet4
30805
30108 0-------.--------.0-------
  318 --------.--------.--------
   18 0-------.--------.--------
    9 0-------.--------.0------1
    8 10011011.01110101.11101010
      <continues>
```

```
0 = 0 (in >99% of pkts)
1 = 1 (in >99% of pkts)
- = "random"
```

Patterns pre and post

- patterns in dst IPs (per src IP scanning tcp/445 and sending > 100 pkts)

```
      2008-10-27 00UTC (pre)
count octet2   .octet3   .octet4
  900
  516 --------.--------.--------
   27 0-------.--------.0-------
    5 ---1-1-1.--------.--------
    4 10011011.01110101.11101010
      <continues>
```

oxo

```
      2009-01-19 00UTC  (post)
count octet2   .octet3   .octet4
30805
30108 0-------.--------.0-------
  318 --------.--------.--------
   18 0-------.--------.--------
    9 0-------.--------.0------1
    8 10011011.01110101.11101010
      <continues>
```

```
0 = 0 (in >99% of pkts)
1 = 1 (in >99% of pkts)
- = "random"
```

Patterns confirmed

- reversing conficker code (Brandon Enright, Michael Hale Ligh @ iDefense):
  - Conficker random IP generation routine can only reach `xxxxxxxx.0xxxxxxx.xxxxxxxx.0xxxxxxx`
- reversed routine showed bias for certain dests (1:2:3 ratio), these ratios also exist in scanning destinations, suggests:
  - reversing was done right
  - most observed tcp/445 scanning is conficker

Overlap with other data?

	src IPs TCP/445 UCSD telescope	src IPs HTTP logs	overlap
2008-01-25 01:00-02:00 UTC	124,106 (1,167 scanning non-oxo)	117,707 (Ricky’s HTTP logs)	26,261 (22% of HTTP logs)
2008-01-25 03:00-04:00 UTC	127,693 (1,082 scanning non-oxo)	63 (single domain HTTP log we got forwarded)	15 (23% of HTTP logs)
2008-01-20 24 hrs	1,314,526 (13,190 scanning non-oxo)	just over 1M (f-secury blog)	?

- differences: NAT, proxy, firewalling TCP/445, private nets scanning

Next steps

- more analysis
  - forensics on pre Nov 21 data
  - correlate with HTTP logs (sinkhole data)
  - 0x0 pattern vs. other (pre and on Nov 21)
- distill useful data to share, possibly ongoing
- “vigilante” : tarpitting?
  - can we slow this thing down with large pool of responders that make tcp connections “get stuck”?
  - success makes us likely appear on blacklists next time around and/or malware adapting
    - so success = failure
    - and failure = failure (but measurable)

Conclusion

- Telescopes continue to provide a valuable macroscopic view of the Internet
  - Conficker / Backscatter are examples of what telescopes can do
- Issues to be solved regarding data sharing
  - real IPs / victim IPs
  - legit traffic interspersed in raw data
  - dataset size (2TB and counting compressed pcap)
    - What would you like:
      - open up the floodgate (2TB of compressed pcap and counting, but we can't give out as-is)
      - sample in time: day per week, hour per day, Nov 21, pre Nov 21
      - level of detail: pcap, flow, more reduction (list of IPs scanning TCP/445?)

DoS Backscatter

- TCP SYN/ACK, RST, ICMP replies from attack victims
  - distillation technique described in Moore et. al (Inferring Internet Denial-of-Service Activity)

DoS Backscatter

- TCP SYN/ACK, RST, ICMP replies from attack victims
  – distillation technique described in Moore et. al (Inferring Internet Denial-of-Service Activity)
- <blink>NOT ATTACK TRAFFIC</blink>

DoS Backscatter

- TCP SYN/ACK, RST, ICMP replies from attack victims
  - distillation technique described in Moore et. al (Inferring Internet Denial-of-Service Activity)
- <blink>NOT ATTACK TRAFFIC</blink>
- 2001 - 2008 data, 1 week per quarter available for download (after user vetting)

DoS Backscatter

- TCP SYN/ACK, RST, ICMP replies from attack victims
  - distillation technique described in Moore et. al (Inferring Internet Denial-of-Service Activity)
- <blink>NOT ATTACK TRAFFIC</blink>
- 2001 - 2008 data, 1 week per quarter available for download (after user vetting)
- hot in 2001, obsoleted by botnets now?
  - researchers ask for (D)DoS attack traffic regularly (we can anonymize and serve if you have any to share)
  - we'd hoped more research published from this backscatter data:
    - PR problem or uninteresting data?

DoS Backscatter

- TCP SYN/ACK, RST, ICMP replies from attack victims
  - distillation technique described in Moore et. al (Inferring Internet Denial-of-Service Activity)
- <blink>NOT ATTACK TRAFFIC</blink>
- 2001 - 2008 data, 1 week per quarter available for download (after user vetting)
- hot in 2001, obsoleted by botnets now?
  - researchers ask for (D)DoS attack traffic regularly (we can anonymize and serve if you have any to share)
  - we'd hoped more research published from this backscatter data:
    - PR problem or uninteresting data?
- Latest dataset available: Nov 12-19 2008 (BS2008Q4)

BS2001Q1 vs. BS2008Q4

	2001-02 (8d)	2008-11 (8d)
# IPs attacked at > 1k observed pps	110	661
# IPs attacked at > 10k observed pps	0	32
max observed pps	2.70k pkts (SYN flood)	36.2k pkts (SYN flood)
max extrapolated pps (if randomly spoofed)	692k pkts	9.26M pkts

- 40 B pkts => 3 Gbit
- extrapolate the extrapolation
  - if it were1.5 kB pkts => 108 Gbit (Arbor 2008: 40 Gbit max)

BS2008Q4 geolocation (max pps 1k+)

Country	Attack victims (IPs)
RFC1918-land	1
Brazil	1
Germany	1
Hong Kong	1
Netherlands	1
Philippines	1
Australia	2
South Korea	11
USA	23
China	619 (!)

- Why China?

Random spoofing?

- Some examples of backscatter in 2008Q4
  - (reply pkts so: dst IP is spoofed src IP)

```
random dst:
#if          time      src addr         dst addr           len pro ts ip.id ttl  sport dport tcp.seq  tcp.ack  flags
 0  1226448412.093350 X.X.X.X          T.113.118.71        48   6 00     0  42     80  3072 28db3086 3ef24e7a -S----
 0  1226448412.128537 X.X.X.X          T.35.235.80         48   6 00     0  42     80  1024 280a94da 50a00705 -S----
 0  1226448412.181751 X.X.X.X          T.142.36.117        44   6 00     0  42     80  1024 76035c3b 39f15e76 -S----
 0  1226448412.204474 X.X.X.X          T.29.53.87          44   6 00     0  42     80  3072 c0e8e536 74428d52 -S----
 0  1226448412.208140 X.X.X.X          T.58.220.93         44   6 00     0  42     80  1024 4e47b5dc 3f2ae717 -S----
 0  1226448412.238628 X.X.X.X          T.163.58.99         44   6 00     0  42     80  1024 d2403700 cd1c1551 -S----
 0  1226448412.241120 X.X.X.X          T.150.205.63        44   6 00     0  42     80  3072 6ffdb5a0 f79b6d60 -S----

slightly less random dst:
#if          time      src addr         dst addr           len pro ts ip.id ttl  sport dport tcp.seq  tcp.ack  flags
 0  1226451258.357809 Y.Y.Y.Y          T.204.1.41          40   6 00   256 119     80 12816 00000000 0000710b -S----
 0  1226451258.360012 Y.Y.Y.Y          T.204.1.122         40   6 00   256 119     80 12897 00000000 0000715c -S----
 0  1226451258.373950 Y.Y.Y.Y          T.204.1.120         40   6 08   256 118     80 12895 00000000 0000715a -S----
 0  1226451258.392873 Y.Y.Y.Y          T.204.1.185         40   6 08   256 119     80 12960 00000000 0000719b -S----
 0  1226451258.398600 Y.Y.Y.Y          T.204.1.126         40   6 08   256 119     80 12901 00000000 00007160 -S----
 0  1226451258.404475 Y.Y.Y.Y          T.204.1.99          40   6 08   256 118     80 12874 00000000 00007145 -S----
 0  1226451258.408447 Y.Y.Y.Y          T.204.1.105         40   6 08   256 118     80 12880 00000000 0000714b -S----

even less random dst:
#if          time      src addr         dst addr           len pro ts ip.id ttl  sport dport tcp.seq  tcp.ack  flags
 0  1226448084.075671 Z.Z.Z.Z          T.207.241.73        44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.075957 Z.Z.Z.Z          T.207.241.73        44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.075964 Z.Z.Z.Z          T.207.241.73        44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.076978 Z.Z.Z.Z          T.207.241.73        44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.077123 Z.Z.Z.Z          T.207.241.73        44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.077709 Z.Z.Z.Z          T.207.241.73        44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.078443 Z.Z.Z.Z          T.207.241.73        44   6 00     0  42     80  4218 b7278c80 01000001 -S----
```

Random spoofing?

- Some examples of backscatter in 2008Q4
  - (reply pkts so: dst IP is spoofed src IP)

```
random dst:
#if          time      src addr         dst addr          len pro ts ip.id ttl sport dport tcp.seq  tcp.ack  flags
 0  1226448412.093350 X.X.X.X          T.113.118.71       48   6 00     0  42    80  3072 28db3086 3ef24e7a -S----
 0  1226448412.128537 X.X.X.X          T.35.235.80        48   6 00     0  42    80  1024 280a94da 50a00705 -S----
 0  1226448412.181751 X.X.X.X          T.142.36.117       44   6 00     0  42    80  1024 76035c3b 39f15e76 -S----
 0  1226448412.204474 X.X.X.X          T.29.53.87         44   6 00     0  42    80  3072 c0e8e536 74428d52 -S----
 0  1226448412.208140 X.X.X.X          T.58.220.93        44   6 00     0  42    80  1024 4e47b5dc 3f2ae717 -S----
 0  1226448412.238628 X.X.X.X          T.163.58.99        44   6 00     0  42    80  1024 d2403700 cd1c1551 -S----
 0  1226448412.241120 X.X.X.X          T.150.205.63       44   6 00     0  42    80  3072 6ffdb5a0 f79b6d60 -S----

slightly less random dst:
#if          time      src addr         dst addr          len pro ts ip.id ttl sport dport tcp.seq  tcp.ack  flags
 0  1226451258.357809 Y.Y.Y.Y          T.204.1.41         40   6 00   256 119    80 12816 00000000 0000710b -S----
 0  1226451258.360012 Y.Y.Y.Y          T.204.1.122        40   6 00   256 119    80 12897 00000000 0000715c -S----
 0  1226451258.373950 Y.Y.Y.Y          T.204.1.120        40   6 08   256 118    80 12895 00000000 0000715a -S----
 0  1226451258.392873 Y.Y.Y.Y          T.204.1.185        40   6 08   256 119    80 12960 00000000 0000719b -S----
 0  1226451258.398600 Y.Y.Y.Y          T.204.1.126        40   6 08   256 119    80 12901 00000000 00007160 -S----
 0  1226451258.404475 Y.Y.Y.Y          T.204.1.99         40   6 08   256 118    80 12874 00000000 00007145 -S----
 0  1226451258.408447 Y.Y.Y.Y          T.204.1.105        40   6 08   256 118    80 12880 00000000 0000714b -S----

even less random dst:
#if          time      src addr         dst addr          len pro ts ip.id ttl sport dport tcp.seq  tcp.ack  flags
 0  1226448084.075671 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.075957 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.075964 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.076978 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.077123 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.077709 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.078443 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
```

Random spoofing?

- Some examples of backscatter in 2008Q4
  - – (reply pkts so: dst IP is spoofed src IP)

```
random dst:
#if          time      src addr         dst addr          len pro ts ip.id ttl  sport dport tcp.seq  tcp.ack  flags
 0  1226448412.093350 X.X.X.X          T.113.118.71       48   6 00     0  42     80  3072 28db3086 3ef24e7a -S----
 0  1226448412.128537 X.X.X.X          T.35.235.80        48   6 00     0  42     80  1024 280a94da 50a00705 -S----
 0  1226448412.181751 X.X.X.X          T.142.36.117       44   6 00     0  42     80  1024 76035c3b 39f15e76 -S----
 0  1226448412.204474 X.X.X.X          T.29.53.87         44   6 00     0  42     80  3072 c0e8e536 74428d52 -S----
 0  1226448412.208140 X.X.X.X          T.58.220.93        44   6 00     0  42     80  1024 4e47b5dc 3f2ae717 -S----
 0  1226448412.238628 X.X.X.X          T.163.58.99        44   6 00     0  42     80  1024 d2403700 cd1c1551 -S----
 0  1226448412.241120 X.X.X.X          T.150.205.63       44   6 00     0  42     80  3072 6ffdb5a0 f79b6d60 -S----

slightly less random dst:
#if          time      src addr         dst addr          len pro ts ip.id ttl  sport dport tcp.seq  tcp.ack  flags
 0  1226451258.357809 Y.Y.Y.Y          T.204.1.41         40   6 00   256 119     80 12816 00000000 0000710b -S----
 0  1226451258.360012 Y.Y.Y.Y          T.204.1.122        40   6 00   256 119     80 12897 00000000 0000715c -S----
 0  1226451258.373950 Y.Y.Y.Y          T.204.1.120        40   6 08   256 118     80 12895 00000000 0000715a -S----
 0  1226451258.392873 Y.Y.Y.Y          T.204.1.185        40   6 08   256 119     80 12960 00000000 0000719b -S----
 0  1226451258.398600 Y.Y.Y.Y          T.204.1.126        40   6 08   256 119     80 12901 00000000 00007160 -S----
 0  1226451258.404475 Y.Y.Y.Y          T.204.1.99         40   6 08   256 118     80 12874 00000000 00007145 -S----
 0  1226451258.408447 Y.Y.Y.Y          T.204.1.105        40   6 08   256 118     80 12880 00000000 0000714b -S----

even less random dst:
#if          time      src addr         dst addr          len pro ts ip.id ttl  sport dport tcp.seq  tcp.ack  flags
 0  1226448084.075671 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.075957 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.075964 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.076978 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.077123 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.077709 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42     80  4218 b7278c80 01000001 -S----
 0  1226448084.078443 Z.Z.Z.Z          T.207.241.73       44   6 00     0  42     80  4218 b7278c80 01000001 -S----
```

Random spoofing?

- Some examples of backscatter in 2008Q4
  - (reply pkts so: dst IP is spoofed src IP)

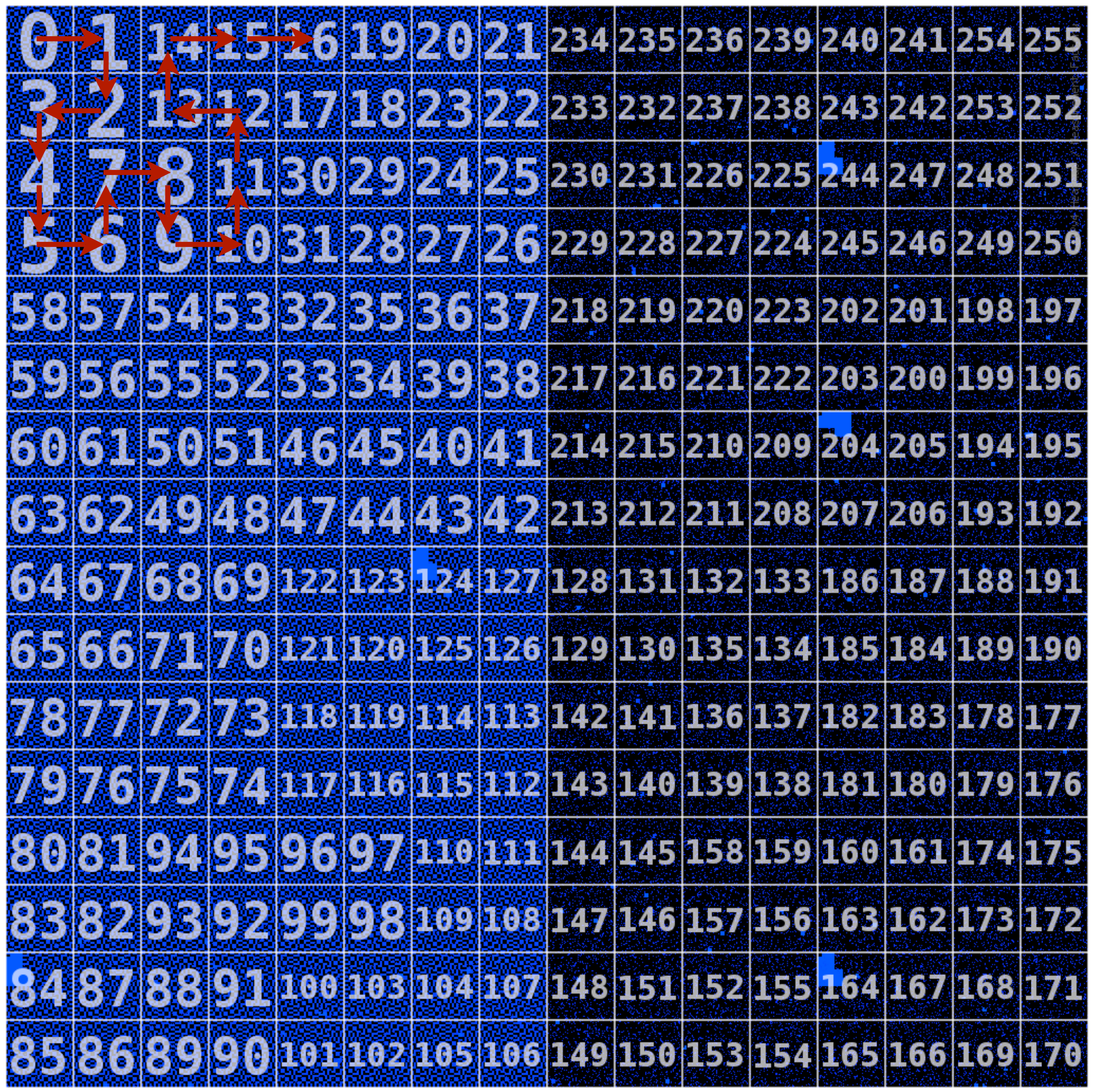
```
random dst:
#if           time       src addr          dst addr          len pro ts ip.id ttl sport dport tcp.seq  tcp.ack  flags
 0  1226448412.093350 X.X.X.X           T.113.118.71       48   6 00     0  42    80  3072 28db3086 3ef24e7a -S----
 0  1226448412.128537 X.X.X.X           T.35.235.80        48   6 00     0  42    80  1024 280a94da 50a00705 -S----
 0  1226448412.181751 X.X.X.X           T.142.36.117       44   6 00     0  42    80  1024 76035c3b 39f15e76 -S----
 0  1226448412.204474 X.X.X.X           T.29.53.87         44   6 00     0  42    80  3072 c0e8e536 74428d52 -S----
 0  1226448412.208140 X.X.X.X           T.58.220.93        44   6 00     0  42    80  1024 4e47b5dc 3f2ae717 -S----
 0  1226448412.238628 X.X.X.X           T.163.58.99        44   6 00     0  42    80  1024 d2403700 cd1c1551 -S----
 0  1226448412.241120 X.X.X.X           T.150.205.63       44   6 00     0  42    80  3072 6ffdb5a0 f79b6d60 -S----

slightly less random dst:
#if           time       src addr          dst addr          len pro ts ip.id ttl sport dport tcp.seq  tcp.ack  flags
 0  1226451258.357809 Y.Y.Y.Y           T.204.1.41         40   6 00   256 119    80 12816 00000000 0000710b -S----
 0  1226451258.360012 Y.Y.Y.Y           T.204.1.122        40   6 00   256 119    80 12897 00000000 0000715c -S----
 0  1226451258.373950 Y.Y.Y.Y           T.204.1.120        40   6 08   256 118    80 12895 00000000 0000715a -S----
 0  1226451258.392873 Y.Y.Y.Y           T.204.1.185        40   6 08   256 119    80 12960 00000000 0000719b -S----
 0  1226451258.398600 Y.Y.Y.Y           T.204.1.126        40   6 08   256 119    80 12901 00000000 00007160 -S----
 0  1226451258.404475 Y.Y.Y.Y           T.204.1.99         40   6 08   256 118    80 12874 00000000 00007145 -S----
 0  1226451258.408447 Y.Y.Y.Y           T.204.1.105        40   6 08   256 118    80 12880 00000000 0000714b -S----

even less random dst:
#if           time       src addr          dst addr          len pro ts ip.id ttl sport dport tcp.seq  tcp.ack  flags
 0  1226448084.075671 Z.Z.Z.Z           T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.075957 Z.Z.Z.Z           T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.075964 Z.Z.Z.Z           T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.076978 Z.Z.Z.Z           T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.077123 Z.Z.Z.Z           T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.077709 Z.Z.Z.Z           T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
 0  1226448084.078443 Z.Z.Z.Z           T.207.241.73       44   6 00     0  42    80  4218 b7278c80 01000001 -S----
```

Hilbert to the rescue

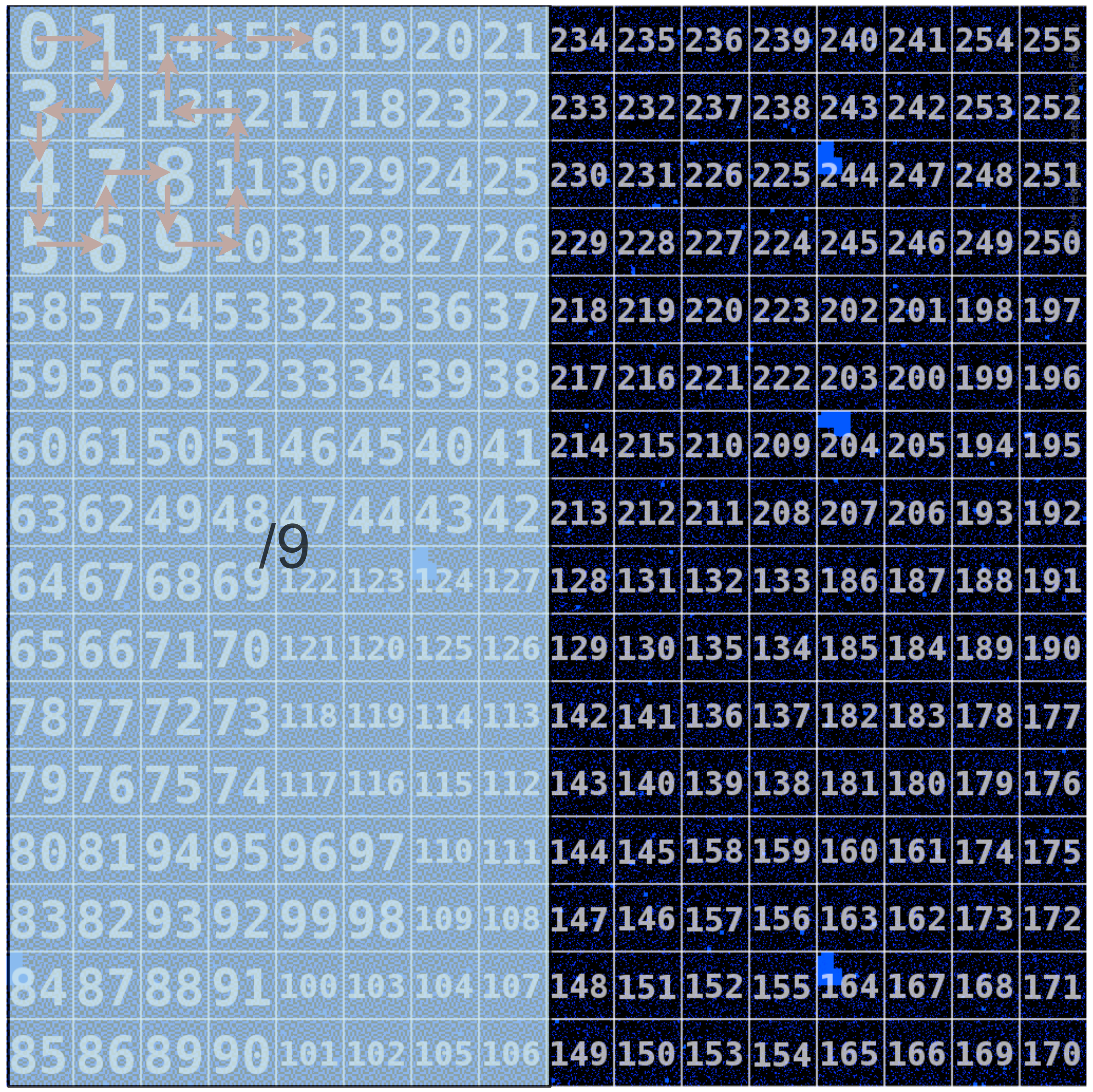
- 1-dimensional telescope IPv4 address space is mapped into a 2-dimensional image using a Hilbert curve, as inspired by xkcd. This means that CIDR netblocks always appear as squares or rectangles in the image.



software: http://maps.measurement-factory.com/software/index.html

Hilbert to the rescue

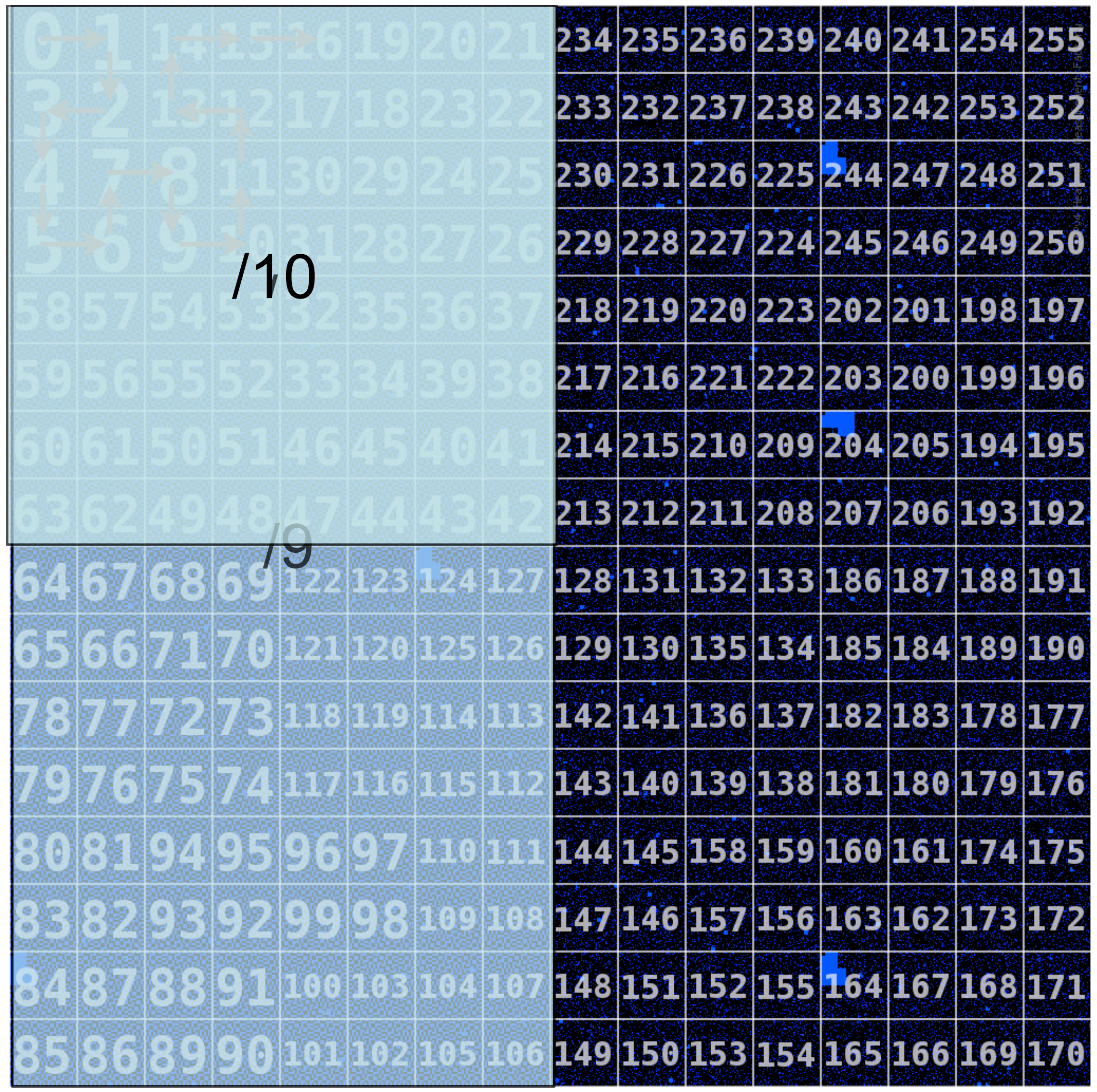
- 1-dimensional telescope IPv4 address space is mapped into a 2-dimensional image using a Hilbert curve, as inspired by xkcd. This means that CIDR netblocks always appear as squares or rectangles in the image.



software: http://maps.measurement-factory.com/software/index.html

Hilbert to the rescue

- 1-dimensional telescope IPv4 address space is mapped into a 2-dimensional image using a Hilbert curve, as inspired by xkcd. This means that CIDR netblocks always appear as squares or rectangles in the image.



software: http://maps.measurement-factory.com/software/index.html

Animation

- intensity: no. pkts to this destination IP address
- each frame: 1hr of backscatter (ie. responses from spoofed attack victim)
- what causes non-random patterns?
- http://www.caida.org/~emile/bs-2008-q4/backscatter.dst-pkts.animated.map.gif

Patterns found

- look at patterns for 661 attack victims (the population with a max pps > 1k), counting everything between 0.01 and 0.99 as random (quite a forgiving definition).
- By this definition: only 16% of spoofed IP attacks somewhat resembling 'random'

```
octet2   .octet3   .octet4    IPs  percentage

-----100.00------.--------- 270  40.85     0 = 0 (>99% pkts)
--------.--------.--------- 109  16.49     1 = 1 (>99% pkts)
-1---100.00------.--------- 28    4.24     - = "random"
11110100.00------.--------- 27    4.08
01010100.00------.--------- 23    3.48
11001100.00------.--------- 21    3.18
10100100.00------.--------- 21    3.18
01111100.00------.--------- 20    3.03
--------.--------.0-------- 17    2.57
     <list continues>
```

Possible use for patterns

- mitigation
  - filter
    - ip&0x00800080=0
    - `pcap: ip[17]&128=0 and ip[19]&128=0`
    - any products doing nonstandard bitmasks?
- forensics:
  - correlate attacks
  - correlate DoS attack-tools (?)
  - correlate (pseudo)random-number generators in active use