Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority

David Dagon

www.gtisc.gatech.edu

ISC/OARC Workshop 2007

Outline

based on joint work with:

- *Google:* Niels Provos
- *GaTech ECE:* Chris Lee
- *GaTech CS:* Wenke Lee

Summary: Resolution Path Corruption

Context: Localized Poisoning

- We measure a “new” DNS “poisoning”: resolution path corruption
- Previous: DNS Poisoning against servers
- Current: stub attacks
  - Of course, stub attacks are hardly new
  - We summarize recent trends surrounding open resolvers

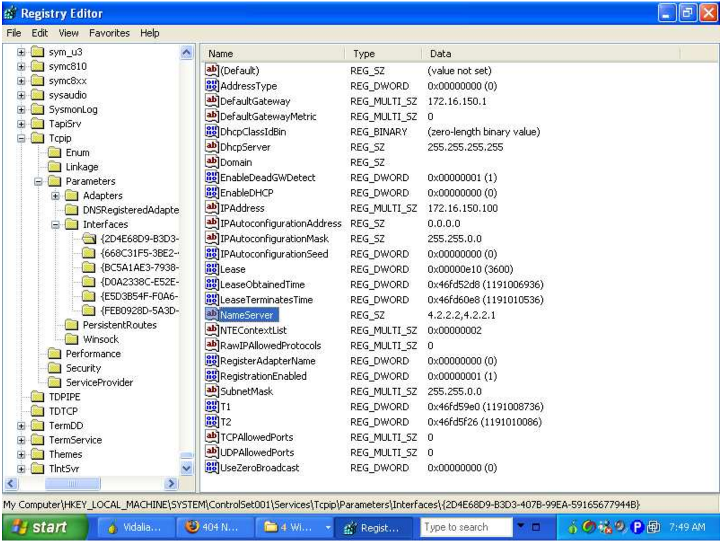
Context: Other (better) Parallel work

- You are presumed to have attended the better talk by John Kristoff @Chicago OARC workshop
- Our work touches on this area

Background

- We have noted a rise in malware that changes default DNS settings
- Many binaries (PE32) point users to malicious DNS servers
- Alarmingly, numerous web pages performed drive-by registry changes
- We decided to investigate

“DNS Changer” Malware: Normal Setup



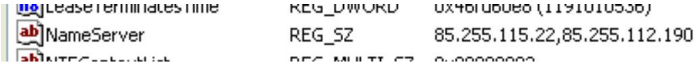
“DNS Changer” Malware: Normal Setup

Windows stub resolver users many registry keys, notably `\\HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\`*`(UID)`*`\NameServer`

“DNS Changer” Malware

- Malware is introducted through the usual vectors (e.g., e-mail spam, web link spam, social engineering)
- Anecdote: Site distributing DNS-changing `zcodec` trojan was top 15,000 page on Internet (3 Yr. Alexa Ave.)

"DNS Changer" Malware: Result

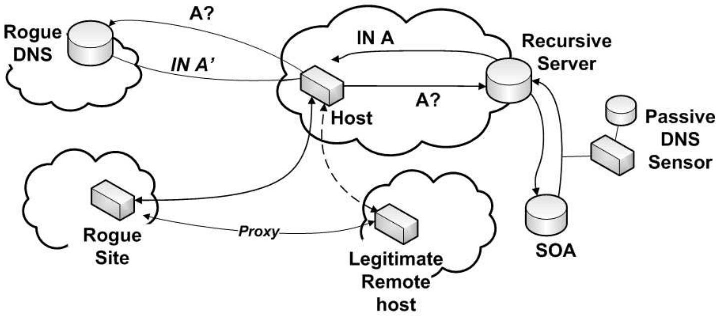


- Sometimes, additional malware dropped (banner/adware)
- Beyond that, the only evidence is the DNS change.
- Consider the challenge this presents to anti-virus detection
  - How does an AV know a DNS server is malicious?
  - Nascent DNS reputation feeds need to materialize
  - Perhaps shoe-horn with NS reputation used in spam detection

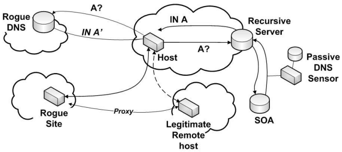
"DNS Changer" Malware: Autopsy

- Malfease execution trace
- `[PID: 844, TID: 468]`
  `[CALL:ADVAPI32.dll:RegCreateKeyExW:1:77DB93AD]`
  `[3:HKEY:LPCWSTR:PHKEY][80000002,`
  `53006F006600740077006100720065005C004D006900630`
  `72006F0073006F00660074005C00570069006E0064006F00`
  `...`
- Essentially the malware changes the default DNS server.
- Get Vetted and download at: `https://malfease.oarci.net`
  - See previous OARC talk on the malware repo
  - (Some DNS-related malware RSS notices may be offered)

"DNS Changer" Malware: The Big Picture



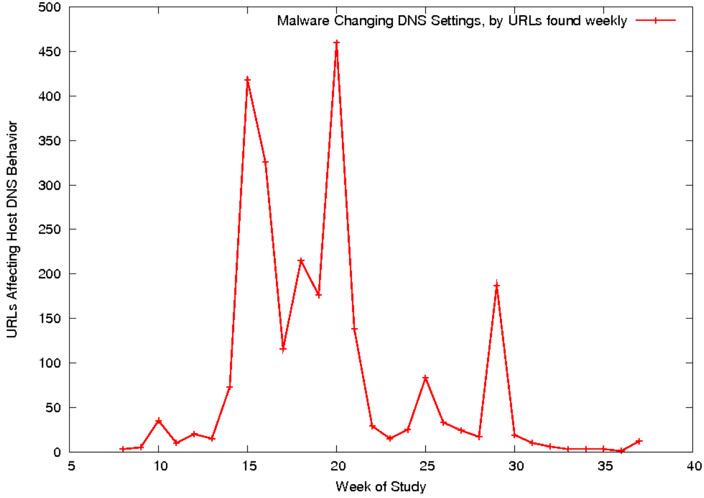
"DNS Changer" Malware: The Big Picture



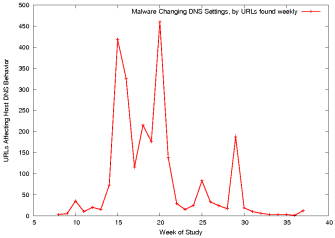
- Malware trivially changes resolution settings
- Rogue DNS server selectively provides malicious answers
- Web servers proxy connections/logins (even without complete MIM)
- Farms of "rogue" DNS servers spotted. (See also Trend Micro's blog[1] entries).

[1] http://blog.trendmicro.com/rogue-domain-name-system-servers-5breposted5d/

"DNS Changer" Drive-By Web Attacks



“DNS Changer” Drive-By Web Attacks

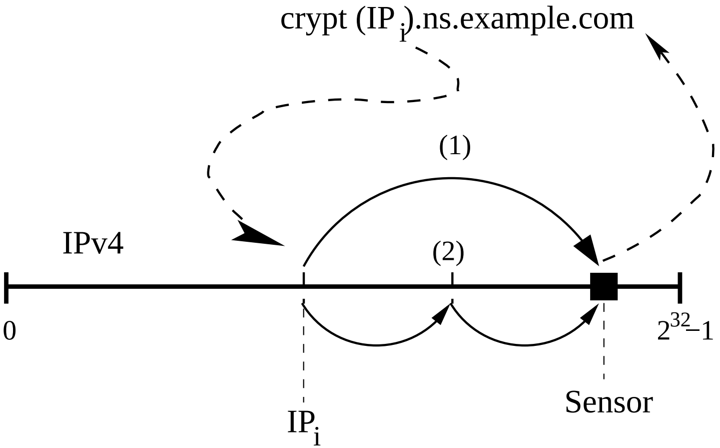


- Google checked the previous months of crawls
- Hundreds of web pages per week were discovered that change DNS settings
- No insight as to age of page; given the source, one suspects the pages were discovered early.
- Note Google offers a related domain reputation API.

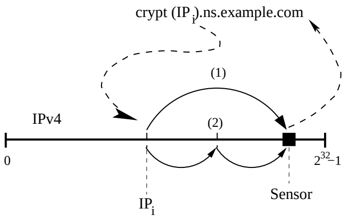
Sourcing Resolution Path Corruption

- We verified this attack using passive DNS (and full captures) at campus border
- Who is behind this?
- Note: registry key changes are *trivial*
  - One merely has to run a rogue DNS server
  - ... or become an affiliate of such a rogue server
- Beyond these anecdotal IPs, we know:
  - These attackers use IPv4;
  - These run open resolvers (by necessity, absent complicated victim ACLs)
- We decided to round up the usual suspects and question them in the lab.
  - We first needed to locate open resolvers.

Study Methodology



Study Methodology



- Unique label queried to all IPv4
- SOA wildcard for parent zone
- Script used to return srcIP of requestor
- Logging at NS yields open recursive and recursive forwarding hosts
- See Kristoff for operational experiences

Design Goals for Survey

- Policy, policy, policy
  - My apologies to any bothered
  - The PTR gave clues (“`dnsstudy1`”)
  - Web page provided means of self-exclusion
- Save state (stop, restart)
- Avoid caching (unique labels)
- Trivially reversible (avoid `SELECT`)
  - Embed srcIP in RR
  - Lamport hash of IPs (cf. SSH Scan tools)

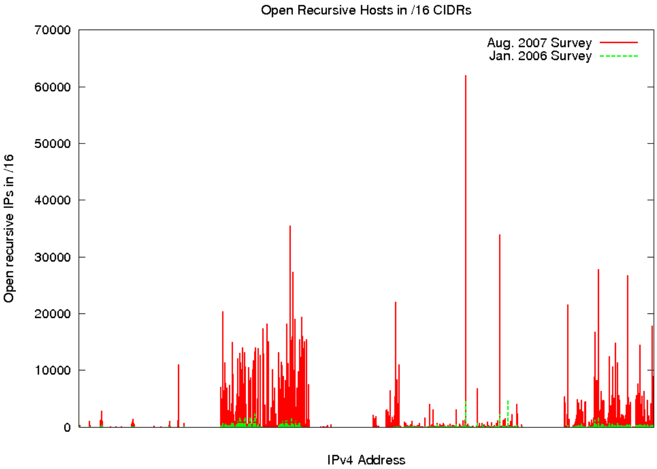
Probe Strategies: Policy

- Avoid bogons, and gov/mil
  bogons = ('0.0.0.0/7', '2.0.0.0/8', '5.0.0.0/8', '7.0.0.0/8', '10.0.0.0/8', '23.0.0.0/8', '27.0.0.0/8', '31.0.0.0/8', '36.0.0.0/7', '39.0.0.0/8', '42.0.0.0/8', '49.0.0.0/8', '50.0.0.0/8', '94.0.0.0/7', '100.0.0.0/6', '104.0.0.0/5', '112.0.0.0/6', '127.0.0.0/8', '169.254.0.0/16', '172.16.0.0/12', '173.0.0.0/8', '174.0.0.0/7', '176.0.0.0/5', '184.0.0.0/6', '192.0.2.0/24', '192.168.0.0/16', '197.0.0.0/8', '198.18.0.0/15', '223.0.0.0/8', '224.0.0.0/3')
  nosolicit = ('3.0.0.0/8', '6.0.0.0/8', '7.0.0.0/8', '11.0.0.0/8', '21.0.0.0/8', '22.0.0.0/8', '26.0.0.0/8', '28.0.0.0/8', '29.0.0.0/8', '30.0.0.0/8', '33.0.0.0/8', '34.0.0.0/8')
- (Note: need to add AS13506's prefixes)
- Listen patiently to those who complain
- Provide documentation and path for self-exclusion

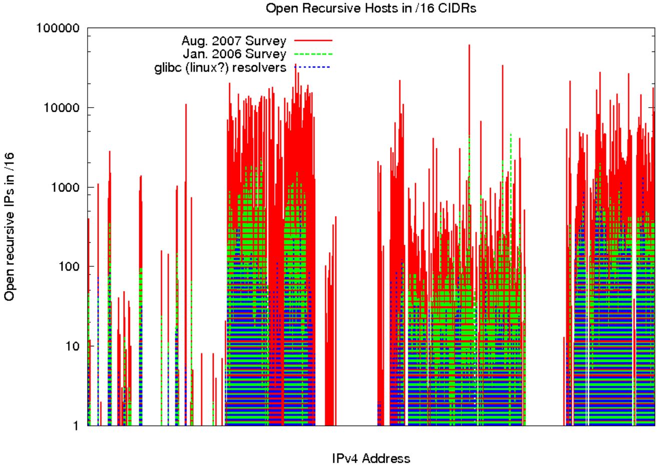
Methodology (cont'd)

- Phase1
  - If response given...
  - Exclude authority open resolvers
  - `fpdns` taken of answering host
  - Perform http request of host
- Phase2
  - Pick 600K open resolvers
  - Ask them repeatedly to resolve phishable domains
  - Note which ones gave incorrect answers
  - If "incorrect", http request to the answered IP

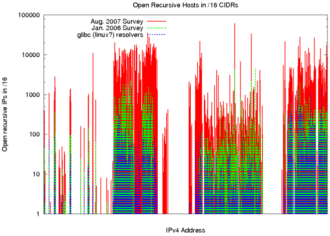
Open Recursion: Comparison of OpenRec in /16s, in IPv4



Open Recursion: Putative GNU libc /16s

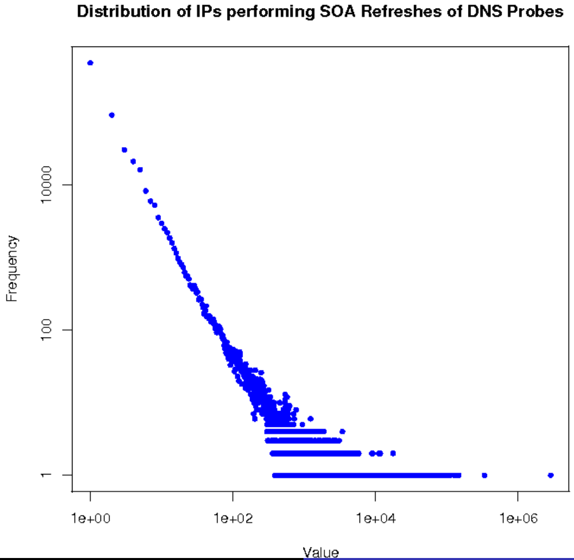


Open Recursion: Putative GNU libc /16s

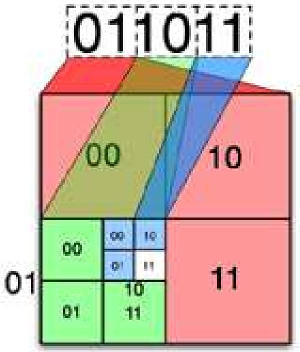


- gnu libc logic of AAAA? $\longrightarrow$ A? queries.
- Other heuristics: Windows DNS servers answered authoritatively for queries for `1.in-addr.arpa`,
- TODO item: update fpdns

Open Recursion: Histogram of Queries to NS



Mapping Mass IPv4 Infections

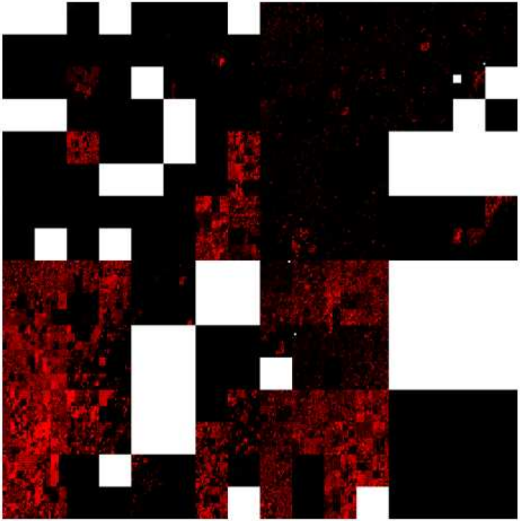


IPMap Visualization of Mass Infections

How to visualize mass infections?

- Complex visualization problem; /16s are too course grained for linear plots
- Solution: IPMap representation
  - Evan Cooke, `http://monkey.org/~phy/ipmaps`
  - Superior to (largely irrelevant) geoip plots
- An alarming note: `ipmap` is usually used for visualising BGP information (i.e., scale is large, prefixes usually $\geq$ /24). But botnets/mass infections are so large, they require the visual metaphors use for BGP visualization. (This alone is a disturbing note.)

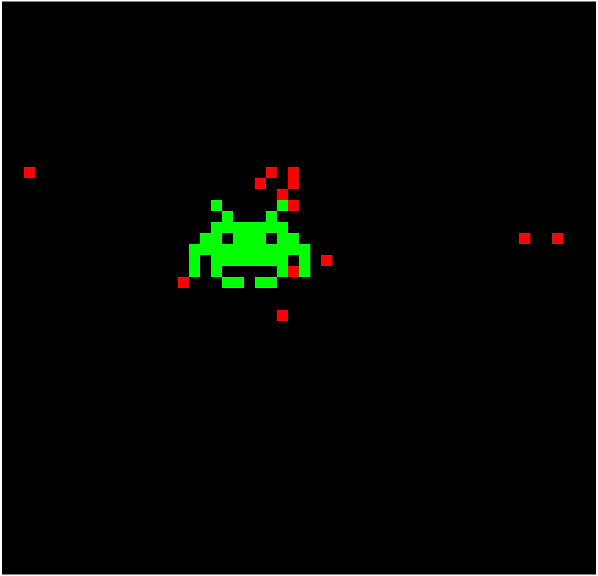
Open Recursive IP Map Visualization; August 2007



A Fun Tangent: Open Recursion in Georgia Tech's Network

- Adding some firewall rules to Georgia's Tech research cluster allows us to selectively highlight CIDRs plotted on ipmap representations:
- These CIDRs (mapped to `RFC 1918`) performed a recursive forward: `10.0.0.29/32, 10.0.0.30/31, 10.0.0.37/32, 10.0.0.49/32, 10.0.0.50/29 10.0.0.55/32, 10.0.0.57/32, 10.0.0.59/32, 10.0.0.60/29, 10.0.0.72/32, 10.0.0.73/32, 10.0.0.96/32, 10.0.0.97/31,10.0.0.102/32, 10.0.0.104/32, 10.0.0.106/32, 10.0.0.108/32, 10.0.0.141/32, 10.0.0.145/32, 10.0.0.146/29, 10.0.0.151/32, 10.0.0.153/32, 10.0.0.156/32, 10.0.0.157/31,10.0.0.159/32, 10.0.0.181/32, 10.0.0.192/32, 10.0.0.194/32, 10.0.0.198/32, 10.0.0.200/32, 10.0.0.201/32, 10.0.0.224/32, 10.0.0.225/32`
- When someone scans us, and plots the result, they find our secret base... (enjoy the next image!)

A Fun Tangent: Mapping Georgia Tech's Secret Base



Analysis: Open Resolvers

- Two sweeps of IPv4:
- Aug 2007, 10,427,000 open recursive
- Sep 2007, 10,573,000 open recursives
- Union: 17,365,000 open recursives over 2 weeks
- Intersection: 3,634,000 in common
  - Some packet loss perhaps
  - However, *union count* points to mass migration of 7M hosts

Analysis: HTTP Server Version

- Appendix A, table 7 of paper
- In general, three classes
  - All open recursive resolvers
  - Intersection of open recursives and visitors to Google's authority server
  - Intersection of open recrusives and Storm victims
- Found numerous embedded devices: RomPager, Agranat-EmWeb
  - Vendor outreach via OARC?

Analysis: "DNS Liars"

- Phase 2: We explore DNS liars. Paper; table 1 (p. 10)
- In general, three classes
  - selected 200K random open recs, 200K open recs contacting Google authority servers, 200K overlap storm
  - Repeatedly queried for "phishable"; 15 min window; 220M probes total over 4 days
  - Diurnal pattern noted (see paper)
  - Approx. 310K-330K resolvers answer; 460K out of 600K total answered
    - Recall migration among 10M open resolvers, noted above
- Creating database of "proxied" webpages
  - Porn, advertising, and proxied pages(!)

New Probe Strategies: Stealth

- Stealth: dictionary words (Markov transition for “likely” labels at SLD/3LD; (Seed via harvest of TLD zones, etc.)
- Passive DNS: validation
- Passive-Aggressive DNS: poison detection
  - Interesting problem: passive DNS data may contain failed poisoning attempts
  - This is not a flaw in passive DNS; we merely desire a convenient means of identification.

Probe Strategies: Ongoing Mapping

- About every 2-3 months, rescan IPv4
- About 2x/month, rescan “hot CIDRs”
- Poll to known “old” DNS servers for early poison detection
- Diversity of srcIPs and SOAs.

Thanks

- Nicholas Bourbaki
- Paul Vixie
- Dave Ulevitch
- The entire Georgia Tech, OIT, abuse staff
- OARC membership, and ICANN