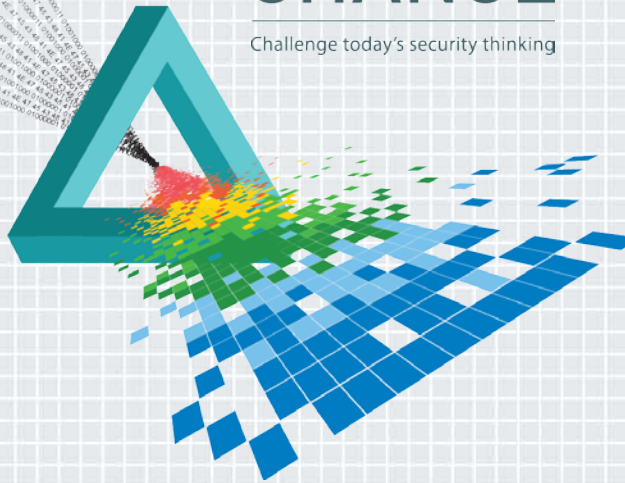
RSAConference2015
San Francisco | April 20-24 | Moscone Center

CHANGE
Challenge today's security thinking



SESSION ID: HTA-R02

Domain Name Abuse: How Cheap New Domain Names Fuel The eCrime Economy

Dr. Paul Vixie

CEO
Farsight Security, Inc.
@paulvixie

Today's Agenda

- DNS History At-A-Glance
- New Domain Name Churn
- Reducing New Domain Name Risk
- The Value of Passive DNS
- Conclusion

RSAConference2015

San Francisco | April 20-24 | Moscone Center

DNS History At-A-Glance

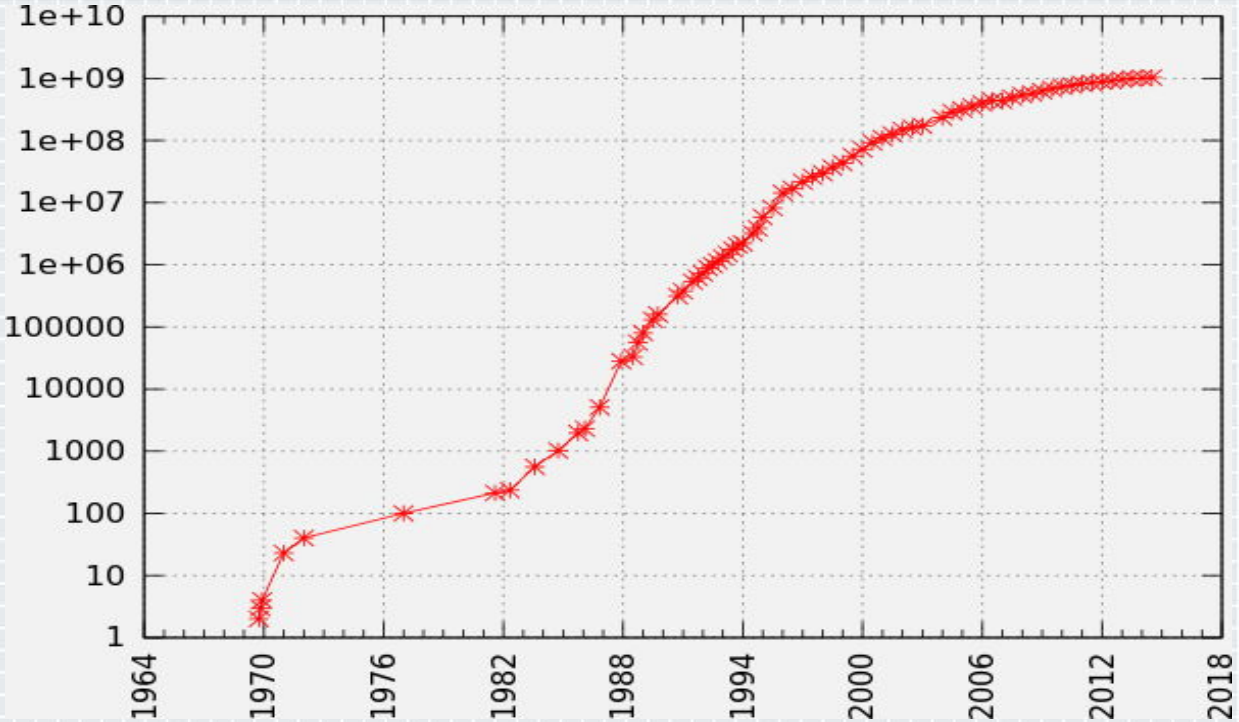
A Brief Recap of (Some) DNS History

- In the beginning, each host had simple (flat) alphanumeric names. Names were manually registered by emailing HOSTSMASTER@SRI-NIC.ARPA
- The Network Information Center (NIC) at Stanford Research Institute maintained a flat text file (HOSTS.TXT) that contained the complete list of such hosts. Sites periodically grabbed copies.
- Nodes translated names to numeric address by doing a search of their local copy of that flat file.
- Clearly this was not a scalable solution (imagine a billion line HOSTS.TXT file, copied to a billion nodes every day!)

The New Era

- A replacement hierarchical and distributed domain name system was specified in 1983 and 1984:
  - "The Domain Names Plan and Schedule," RFC881, Postel, Nov. 1983
  - "Domain Names – Concepts & Facilities," RFC882, Mockapetris, Nov. 1983 "Domain Names – Implementation and Specifications," RFC883, Mockapetris, Nov. 1983
  - "Domain Requirements," RFC980, Postel & Reynolds, Oct. 1984, etc.
- A hierarchical and distributed domain name system was critical to enable growth of the Internet (ironically, today it also threatens it)

It's no coincidence that material growth in the # of connected hosts happened post-DNS



"Internet Hosts Count log" by Kopiersperre (talk) - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons – http://commons.wikimedia.org/wiki/File:Internet_Hosts_Count_log.svg#mediaviewer/File:Internet_Hosts_Count_log.svg

The Evolution

- Symbolics.com, the first dot com, registered March 15th, 1985. Domain names were free for the next ten years.
- 1995-99: Network Solutions era. Price goes to $100 for two years.
- 1999-date: ICANN and the shared registration system.
  New cost for a .com? $7.85 to the registry ($0.25 goes to ICANN) + whatever the registrar adds on (typically just a few bucks)
- Domains are often bundled at nominal cost in packages with web hosting, web design, name service, privacy protection, etc.
- Some domains available at zero cost to drive market share, etc.

Example of One Domain Policy Gone Awry

- Domain Tasting: "In February 2007, 55.1 million domain names were registered. Of those, 51.5 million were canceled and refunded just before the 5 day grace period expired and only 3.6 million domain names were actually kept." [Source: Godaddy]
- Driver? Empirical evaluation of pay-per-impression advertising revenues
- Eliminated in 2008/2009 by ICANN reforms correcting exploitable cost structure
- ***Note:*** **anything free (or cheap) will be prone to exploitation.**

RSAConference2015

San Francisco | April 20-24 | Moscone Center

New Domain Name Churn

Internet as Substrate; Domains as Identities

- IP packets, IP addresses and BGP routes, *underlay* everything
- We *overlay* that substrate with many applications, such as *the web*
- The most important overlay layer is, in many ways, DNS
- For most sites, DNS is totally good -- and operationally critical
- Can you imagine Amazon, Apple, Cisco, eBay, Microsoft, PayPal, without DNS? No. It's unimaginable. Domains are literally priceless to the online operations of these and many other companies.
- Their domains ARE these companies' identities.

Domain Names Are Also Important to Criminals (Just Not The Same Way As For Corporations)

- Cyber criminals aren't interested in long-lived domain names.
- For *criminals*, domains are free (or cheap) & short-lived assets
- "Honest" bad guys? ~$10/name is just a "cost of doing business," too inconsequential to mention, even if using 100's of them per day
- Other bad guys? Fraudulently use stolen cards to get domains. Use those names until the card is reported; lather/rinse/repeat.
- And then there's all the intentionally free domain/free subdomain/free domain name redirection services out there...

Free... And Liable to Being Abused As A Result

- **Domains:** .cf, .ga, .gq, .ml, .tk
- **Subdomains:** .eu.nu, .web.gg, us.nf, int.nf, tv.gg, co.gp, online.gp, asia.gp, biz.uz, pro.vg, name.vu, info.nu, edu.ms, mobi.ps, .co.nr, or tens of thousands of other domain names offering subdomains to those interested (see http://freedns.afraid.org/domain/registry/)
- **URL Redirector Services:** One list of hundreds of URL shorteners and redirectors http://longurl.org/services
- These free domains/services aren't meant to be abused and their operators try to police them, but criminals are relentless.

Why Criminals Need New Domain Names

- These days, if you use a domain (for good/ill) the world will notice
- Domain intelligence services are very efficient, listing misused or abused domains very quickly (often within just minutes).
- Domains – once listed – are worthless (or even become liabilities):
  - Any content that includes the listed domain is "dead on arrival" due to domain-based block lists (SURBL, Spamhaus DBL, etc.)
  - Domain names may even act as a connection back to the cyber criminal (WHOIS POC info, credit card info, etc.)
- Blocklists make life very unpleasant for spammers/cyber criminals.

A Historical Aside About Blocklists

- Wikipedia says, "The first DNSBL was the Real-time Blackhole List (RBL), created in 1997, at first as a BGP feed by Paul Vixie, and then as a DNSBL by Eric Ziegast as part of Vixie's Mail Abuse Prevention System (MAPS) [...] The inventor of the technique later commonly called a DNSBL was Eric Ziegast while employed at Vixie Enterprises."
- I'm proud to say that Eric is still a valued part of the Farsight Security family today. We all owe Eric a debt of thanks.
- So how do the bad guys counter blocklists? Many approaches, most notably, they begin to continually use new domains

Constantly Using New Domain Names Makes A Lot of Sense For The Bad Guys...

- Besides complicating use of blocklists...
- *Continual new domains complicate prioritization of investigations:*
  - "Who's the worst/hottest bad guy, our top priority for attention?" In order to tell, investigators need to aggregate all the relevant domains – but which ones belong to each particular bad guy? [And can we *prove* that attribution?]
- *Continual new domains exacerbate evidence management issues:*
  - Imagine thousands of domains, spread across multiple registrars, each using privacy/proxy services to hide contact information, and each of which may need court paperwork to "pierce the veil."

Fast Flux Hosting

- Just as bad guys churn through domains, at one point they also churned through IPs, leveraging bots for "bulletproof hosting"
- Lots o' bots were (and are) available. Bad guys could use those to host content as well as send spam, conduct DDoS attacks, etc.
- They'd use short TTLs and constantly rotate through new botted hosts, continually updating DNS to point to 6-to-12 botted hosts, each acting as proxy to a hidden backend real server.
- This basically worked pretty well, at least a few years ago, and some fast flux hosts continue to be seen today...

A Domain Tagged as Fast Flux by Zeus Tracker

- deolegistronf[dot]com. 150 IN A 31.202.17.249 [AS34700]
  deolegistronf[dot]com. 150 IN A 178.158.131.20 [AS50780]
  deolegistronf[dot]com. 150 IN A 77.122.150.5 [AS25229]
  deolegistronf[dot]com. 150 IN A 136.169.129.8 [AS24955]
  deolegistronf[dot]com. 150 IN A 92.113.61.139 [AS6849]
  deolegistronf[dot]com. 150 IN A 46.36.143.223 [AS39824]
  deolegistronf[dot]com. 150 IN A 81.4.149.82 [AS6866]
  deolegistronf[dot]com. 150 IN A 176.195.204.168 [AS12714]
  deolegistronf[dot]com. 150 IN A 212.76.8.221 [AS13082]
  deolegistronf[dot]com. 150 IN A 123.194.248.221 [AS9924]
  deolegistronf[dot]com. 150 IN A 188.214.33.160 [AS50886]

Why Doesn't Everyone Use Fast Flux Hosting?

- Fast flux hosting generally isn't necessary if you're constantly churning through new domains, instead.
- New domains can just be assigned to IPs from a regular hosting company (by the time the complaints come pouring in, the bad guy will have moved on), or you can always use bots
- So how, then, to cope with these hit-and-run domain name strategies?

Insight: No One Needs to Immediately Use a New Domain (Except Cyber Criminals)

- Cyber criminals get new domains, abuse and then abandon them – *within minutes*
- While the good guys are still figuring what they're seeing, the bad guys are making a "lightning strike:" in, out, gone.
- The trick is to "help" these cyber criminals slow down a little. What's the rush? No honest person, no legitimate domain, is in *that* big of a hurry...

RSAConference2015

San Francisco | April 20-24 | Moscone Center

Reducing New Domain Name Risk

Simple Strategy: Temporarily Defer The Resolution of ALL Newly Observed Domains

- Temporarily deferring the resolution of ALL new observed domains is a simple strategy, but one that's surprisingly effective....
- By ignoring new domains for a specific period of time, you'll frustrate cyber criminals' "no-huddle offense."
- Following this approach, domain reputation companies have more time to review new domains and block those found to be bad.

How Long Is Enough? How Long Is Too Long?

- We won't pretend to dictate a single "right" answer. Users normally can find an "ignore" duration that works for them from:

  5 minutes
  10 minutes
  30 minutes
  60 minutes
  3 hours
  12 hours
  24 hours

What Counts As A "Newly Observed Domain?"

- Domains are "new" if they haven't been seen in use on network -- - it isn't a function of when a domain was just registered.
- Newly detected domain information is exceedingly time sensitive: need to publish in real time (or near real time) to block resolution
- This implies a need for a low latency real-time (stream) computing approach rather than asynchronous (batch) computing paradigm.
- **This has been operationally proven in production.**

Operationalizing Use of Newly Observed Domains

- A couple of examples of how one could practically employ a feed of newly observed domains:
  - Download an rbldnsd-formatted file via rsync; use that data as an input to SpamAssassin or another spam scoring/filtering systems, or
  - Download a Response Policy Zone-formatted file via IXFR, blocking the new domains for ALL applications by using BIND with RPZ (thereby creating a "DNS-firewall")

"DNS Firewalls" with RPZ (Response Policy Zones)

- Uses DNS zones to implement DNS Firewall policy
  - If it doesn't resolve in DNS, it's blocked (to a first approximation)
- Pub-sub is handled by NOTIFY/TSIG/IXFR
  - Many publishers, many subscribers, one format
- Pay other publishers, or create your own
  - Or do both, plus a private exception list
- Simple failure or walled garden, as you choose
  - We call this “taking back the streets” (“the DNS”)

RPZ Capabilities

- Triggers (RR owners):
  - If the query name is $X
  - If the response contains an address in CIDR $X
  - If any NS name is $X
  - If any NS address is in CIDR $X
  - If the query source address is in CIDR $X
- Actions (RR data):
  - Synthesize NXDOMAIN
  - Synthesize CNAME
  - Synthesize NODATA
  - Synthesize an answer
  - Answer with the truth
  - But remember, it's not a sin to lie to criminals

Why Use RPZ?

- Easy stuff:
  - Block access to DGA C&C's
  - Block access to known phish/driveby downloaders
  - Block e-mail if envelope/header is spammy
- More interesting stuff:
  - Block DNS A/AAAA records in bad address space
    - E.g., import Team Cymru Bogons or Spamhaus DROP list
  - Block DNS records in your own address space
    - After allowing your own domains to do so, of course

RPZ Status

- Implications:
  - Controlled Balkanization (your network, your rules)
  - Open market for producers and consumers
  - Differentiated service at a global scale
  - Instantaneous effective takedown
- Deployment:
  - The RPZ standard is open and unencumbered
  - So far implemented only in BIND
  - Performance is pretty reasonable
  - New features will be backward compatible
  - This is not an IETF standard

RSAConference2015

San Francisco | April 20-24 | Moscone Center

The Value of Passive DNS

Exterminators seldom find just *one* termite

- ... and cyber investigators seldom find bad guys with just *one* evil domain. Bad guys almost *always* have multiple domains for the reasons we've previously discussed.
- But *how* to find them? This is where we can leverage the inherent relationships that almost always exist among domain names:
  - Given the IP of *one* bad domain (or bad name server), often there will be additional bad domains (or bad name servers) using that same IP
  - Bad guys will often share a single set of name servers for multiple related domains
  - Over time, bad domain names will often move from one bad IP to another, which can lead to still more IP that merit investigation

Passive DNS

- Passive DNS makes it possible to synthetically derive implicit DNS relationships based on empirically observed query/response data.
- Sensors collect DNS data from recursive resolvers across Internet (we collect data above recursive resolvers to help protect end-user privacy).
- This collected DNS data gets stored in a database, and indexed. Hunt teams can query the database using one indicator of badness to find others.
- Farsight's passive DNS data is called DNSDB™, but there are others, too
- Let's use DNSDB to explore a few **NON-MALICIOUS** examples.

Given an IP (or CIDR netblock) of interest, what domains have used that address?

$ dnsdb_query.py -i 63.241.205.21
oregon.gov. IN A 63.241.205.21
gis.oregon.gov. IN A 63.241.205.21
egov.oregon.gov. IN A 63.241.205.21
courts.oregon.gov. IN A 63.241.205.21
education.oregon.gov. IN A 63.241.205.21
insurance.oregon.gov. IN A 63.241.205.21
healthoregon.org. IN A 63.241.205.21
healthykidsoregon.org. IN A 63.241.205.21
[etc]

Given a base domain, what FQDNs are known to be associated with that base domain?

$ dnsdb_query -r *.rsaconference.com/a | grep "nce.com. "

rsaconference.com. A 128.221.203.14
rsaconference.com. A 168.159.218.92
rsaconference.com. A 204.13.110.98
e.rsaconference.com. A 204.13.110.98
ae.rsaconference.com. A 68.142.139.80
ae.rsaconference.com. A 136.179.0.37
cm.rsaconference.com. A 68.142.139.116
ec.rsaconference.com. A 68.142.139.117
[etc]

Given a particular name server, what domains have we seen using that name server?

$ dnsdb_query.py -n ns1.ieee.org/ns
ieee.com. IN NS ns1.ieee.org.
myieee.com. IN NS ns1.ieee.org.
ieeeexplore.com. IN NS ns1.ieee.org.
trynanotechnology.com. IN NS ns1.ieee.org.
ieeeconfpublishing.com. IN NS ns1.ieee.org.
photonicssociety.net. IN NS ns1.ieee.org.
ieee.org. IN NS ns1.ieee.org.
computer.org. IN NS ns1.ieee.org.
[etc]

Given a domain, what IP or IPs has that domain used over time?

$ dnsdb_query.py -s time_last -r www.farsightsecurity.com/a
;; bailiwick: farsightsecurity.com.
;; count: 164
;; first seen: 2013-07-01 17:37:26 -0000
;; last seen: 2013-09-24 17:14:08 -0000
www.farsightsecurity.com. IN A **149.20.4.207**

;; bailiwick: farsightsecurity.com.
;; count: 4,289
;; first seen: 2013-09-25 20:02:10 -0000
;; last seen: 2015-01-23 02:20:22 -0000
www.farsightsecurity.com. IN A **66.160.140.81**

Given a domain, what name servers has that domain used over time?

$ dnsdb_query.py -s time_last -r fsi.io/ns/fsi.io
[...]
;; first seen: 2013-06-30 17:28:00 -0000
;; last seen: 2013-07-15 16:51:10 -0000
fsi.io. IN NS **ns.lah1.vix.com.**
fsi.io. IN NS **ns1.isc-sns.net.**
[...]
;; first seen: 2013-07-15 17:26:55 -0000
;; last seen: 2015-01-23 15:33:31 -0000
fsi.io. IN NS **ns5.dnsmadeeasy.com.**
fsi.io. IN NS **ns6.dnsmadeeasy.com.**
[...]

Prefer JSON to plain text output? Just add a -j

$ dnsdb_query.py -r f.root-servers.net/a/root-servers.net -j

{"count": 2676912802, "time_first": 1277349038, "rrtype": "A", "rrname": "f.root-servers.net.", "bailiwick": "root-servers.net.", "rdata": ["192.5.5.241"], "time_last": 1424978882}

Json format output is perfect for those frustrated with plain text, including those who like to use json slicing/dicing/formatting tools such as ./jq

Why Passive DNS Methods Matter

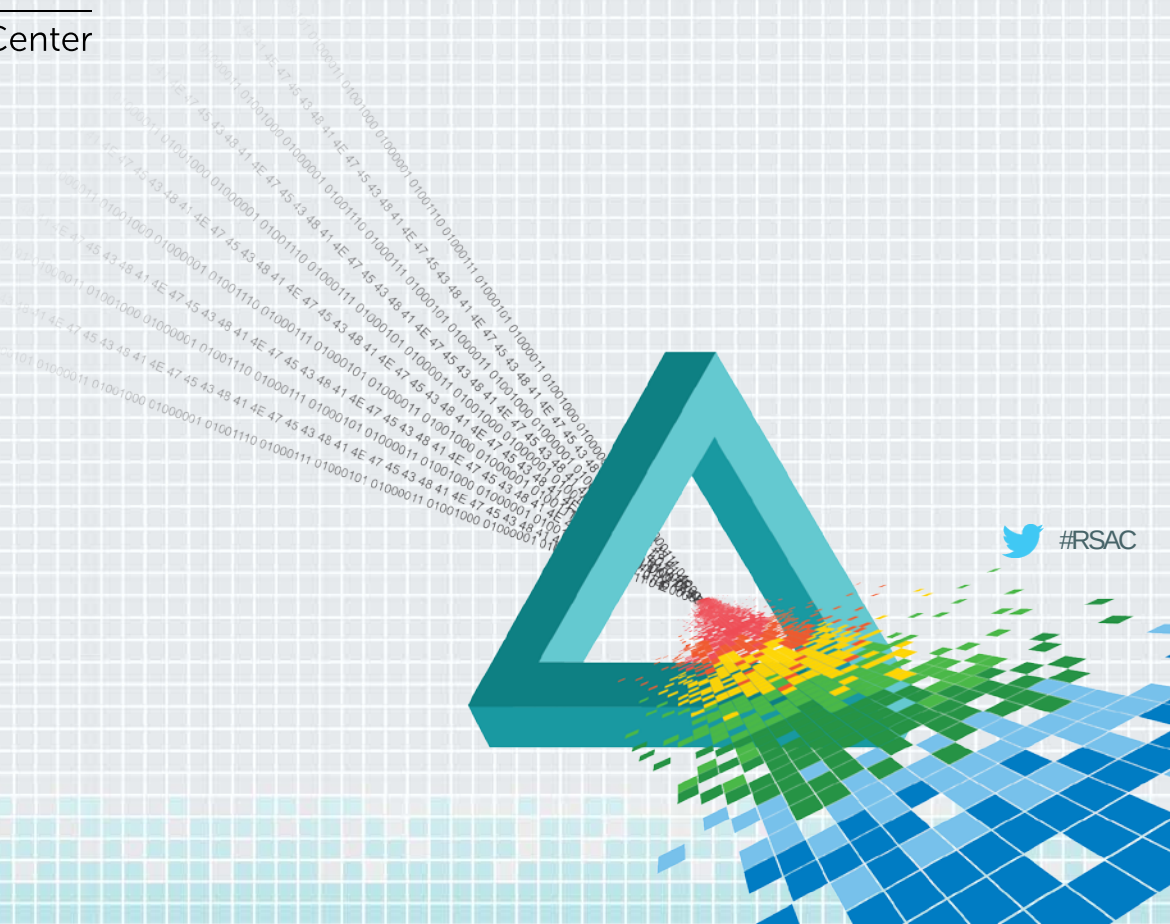
- Investigators can use sometimes-scarce "clues" (such as even a single malicious domain) to find MANY other related domain names they might otherwise have missed, thereby avoiding the frustration of "incomplete takedowns"...

  "He hit 5 of my domains but missed 8,750 other ones!"

- Agencies or enterprises planning takedowns or local blocks can avoid embarrassment by checking for potential 'collateral damage:'

  "Um, there are 14,000 apparently innocent domains on that IP, as well as the three bad ones we noticed. Maybe we should hold off blocking that IP for now..."

Conclusion



Applying What You've Learned Today

- There are now massive volumes of untraceable junk domains
  - Use Passive DNS methods to make forensics possible
  - Use DNS RPZ to block unwanted domains locally/collaboratively
- There are also massive volumes of forged DNS queries
  - Deploy Source Address Validation (aka BCP38/BCP84) to limit emission of spoofed DNS queries
  - Use DNS Response Rate Limiting to protect your authority servers
  - Use IP ACLs to limit unauthorized access to your recursive resolvers
- Pay attention to DNS and treat it as if it matters (because it does!)

Q&A

- Thank you
- Farsight Security Whitepaper: *Passive DNS for Threat Intelligence*
- Contact information: info@farsightsecurity.com

Limited Bibliography

https://www.farsightsecurity.com/
http://www.redbarn.org/dns/ratelimits
http://www.redbarn.org/internet/save
http://dnsrpz.info/