

# Forming General Filter Rules Using BGP Prefix Data and Passive DNS

Joe St Sauver, Ph.D. (stsauver@fsi.io)

PGP: 54A7 02D4 E156 1037 4ADF 2290 9D54 F6B4 36AD 91D7

Scientist, Farsight Security, Inc.

November 10th, 2016 16:30 – 17:30

FIRST TC

Palau Robert, Barcelona, Catalonia

<https://www.stsauver.com/joe/first-tc-barcelona/>

TLP White: Disclosure is unlimited.

# **I. Introduction**

# Thanks For The Chance To Talk Today!

- I'd like to thank FIRST TC and **Xavier Panadero** of Centre de Seguretat de la Informació de Catalunya (CESICAT) for inviting me to present. I'm honored!
- Thanks to to all of **you** for attending – I know we're close to the end of this week's meetings, and many of you will be catching trains or flights home soon.
- Thanks, too, to my boss at Farsight Security, Inc., for giving me the time to attend today's meeting.
- For those who may not know me, I live in Oregon -- that's the red state on the map of the United States. Some other background...



## Background

- My Ph.D. is in Production and Operations Management from the University of Oregon.
- I currently work as a Scientist for Farsight Security, Inc. Prior to working for Farsight, I was with the University of Oregon Computing Center/Information Services for about 28 years.
- During part of that time, I worked under contract as Internet2's Nationwide Security Programs Manager, including running the very popular InCommon SSL/TLS Certificate Program and the popular InCommon Multifactor Authentication programs (at that time it included Duo Security and Safenet PKI hard tokens).
- I'm currently one of 6 Sr Technical Advisors for M3AAWG, the Messaging, Malware and Mobile Anti-Abuse Working Group, and I'm involved with a variety of other cyber security efforts, such as the REN-ISAC TAG, the Global Cyber Alliance, and Cybergreen.



## Content and Format

- Despite all those affiliations and more, the content of today's slides represent solely my own opinions.
- Speaking of my slides: as you've seen by now, I produce **very detailed slides**. Some people don't get why I create slides like this, so I now routinely explain my thinking on this..
- I've tried the more-typical 3-4 bullets/slide (with maybe 15 slides for an hour long talk), but I found myself getting **sidetracked, rambling/running over**, or I end up **missing/skipping stuff**.
- My slide style prevents a lot of those problems, and means that **you don't need to try to take notes**.
- That said, I'm **not going to read my slides word-for-word for you**. You don't need to try to do so, either, although they are a sort of "closed captioning" if you are deaf or hard-of-hearing.
- I also write detailed slides for those who think I talk too fast or find my American drawl hard to understand. :-)

## What We're Going To Cover Today

- I'm always torn when it comes to what to cover during a given talk. I usually end up talking about **what's on my mind**.
- Most recently, **Ponemon** released a study that concerns me greatly: **reportedly many sites do not know how to use network intelligence to effectively block basic incoming attacks** such as brute force password cracking attacks against ssh, or email spam.
- This is such a mundane (but highly critical) skill!
- **I know that you all already get it, and I know that it is dangerous to even bring this topic up at a meeting about incident response analytics, but I really want to enlist your help to ensure that every one can effectively use threat intelligence data!**
- I also specifically wanted to talk about how **BGP routing data** and **passive DNS** can be used to help you intelligently block threat sources while not causing unacceptable collateral damage.

## Is This Really A Problem? Yes. Ponemon Reports...

- "Seventy percent of security pros said that their companies have problems taking actions based on threat intelligence because there is too much of it, or it is too complex, according to a report by Ponemon Research [....]"
- "In particular, 69 percent said that their companies lacked staff expertise. As a result, only 46 percent said that incident responders used threat data when deciding how to respond to threats, and **only 27 percent said that they were effective in using the data.**"



-----  
"Flood of threat intelligence overwhelming for many firms: The amount of threat information coming in from security systems is overwhelming for many companies,"  
Nov 3, 2016 [emphasis added] <http://www.csoonline.com/article/3138003/security/flood-of-threat-intelligence-overwhelming-for-many-firms.html>

## And This is True Even Though Many Of The Attacks Are Basic "unsophisticated, brute-force attacks."

- Continuing with the Ponemon study...

"According to a report based on two years of sensor data, **57 percent of attacks that get through firewalls and antivirus systems are unsophisticated, brute-force attacks.** This is due to ongoing, automated activity by attackers running scans looking for unpatched software, default passwords, and misconfigured systems."

- "These probes are constantly looking for ways that attackers can grab a foothold in a system, and **there isn't much that companies can do to stop it without also locking out customers, partners, employees, and other legitimate services.**"
- **I DISAGREE.**

## This Is Just Crazy

- If you can't effectively use security data to protect yourself, you're "going to war unarmed."
- And as a person who works for a security data provider, hearing that people can't effectively use security data is also an obvious concern, kind to a car maker hearing that people find it hard to drive cars
- **The Ponemon report demands corrective training for the community.**
- **Attacks should be -- and are -- block-able, and usually without unacceptable levels of collateral damage.**
- Let's make this concrete.

# A Hypothetical System's Attack Surface

- Let's think about a typical system.
- Let's assume that it is a simple but well-run one, with just two Internet-accessible services:
  - sshd (e.g., perhaps OpenSSH)
  - smtpd (e.g., perhaps Postfix)
- All other services are off/handled elsewhere (authoritative DNS, http and https web service, etc.)
- We assume that the host is directly exposed to the Internet, and unmanaged, so it needs to do its own monitoring and defensive filtering.

## Defensive Options (In Skeleton Form)

- **Option 1: Reactive (but better than nothing)**
  - Attack observed
  - Block <something>
  - Attack observed
  - Block <something>
  - ....
- **Option 2: Proactive (generally better than just being reactive)**
  - Leverage one or more threat feeds to block threats already detected elsewhere
  - Continue to take action to block any unique local threats, too.
  - Share data to help others?

## Out of Scope Today

- I'm **NOT** going to address:
  - **Doing housekeeping of local block lists** (they can be like "Hotel California" -- "you can check in, but you can never leave")
  - **Particularly sophisticated attacks** (let's just focus the 57% of attacks that are basic/brute force attacks as first step)
  - **Malware issues on MS Windows or Android platforms**
  - **DDoS attacks**
  - **IoT** (looked like a problem with well-known passwords to me)
  - **SCADA/control systems**
  - **Social media**
  - **The crypto wars**
  - **STIX/TAXII wars**
  - **Insider threat, etc., etc., etc.**



# We're Just Going to Cope With Two Simple Attacks

- **Persistent attempts to login via ssh, you know:**

"Hmm. Someone just tried to login as root from Africa... Do we have a sysadmin vacationing there? No? I didn't think so..."

- **Unwanted email, aka "spam."**

"Ahh. The toenail fungus email spammer. That guy just never takes no for an answer, does he?"

## You Could Just Ignore Those Attacks, Or, You Could Try Blocking Them.

- But what size filters should be applied?
- **Individual IP addresses**, as actually seen delivering unwanted traffic?
- **Constant-size network blocks?** (maybe the size of a subnet, such as an IPv4 /24, or an IPv6 /64? (trying to block individual IPv6 IPs is just nuts))
- What if there are "rogue" net blocks sourcing unwanted traffic that are relatively large (say an IPv4 /15 or /16), while others are known to be relatively small (/27 or /28)? Should we use common sense and block those **varying-size network ranges, small or large though they may be** instead of blocking IP-by-IP or /24-by-/24?
- What's current best-of-breed practice?

# The Spamhaus SBL: It Often Lists Individual /32's

https://www.spamhaus.org/sbl/latest/

▼

↺

🔍 Search

IN

SBL319880

04-Nov 13:30 GMT

186.215.124.186/32

Canadian Pharmacy : spamtrap hit (hijacked client)

gvt.net.br

IN

SBL319879

04-Nov 13:29 GMT

182.75.249.94/32

Canadian Pharmacy : spamtrap hit (hijacked client)

airtel.in

IN

SBL319877

04-Nov 13:25 GMT

162.253.153.189/32

spam support (domains)

reprisehosting.com

IN

SBL319876

04-Nov 13:05 GMT

67.227.85.171/32

pennypickstocks.com

colocationamerica.com

**Note:** While this example shows only /32's being listed, Spamhaus DOES also list larger net blocks on the SBL when it is appropriate to do so.

## A Feed That Formerly Listed Just Fixed-Size CIDRs

# Entries consist of fields with identifying characteristics of a  
# a source IP address that has been seen sending HTTP requests  
# to Dragon Research Pods. This report lists hosts that are highly  
# suspicious and are likely conducting malicious HTTP attacks.  
# Each entry is sorted according to a route origination ASN.  
[...]

# **netblock** The source IPv4 or IPv6 network that is being reported.  
# For IPv4 this will be the /24 the actual host IP is in.  
# For IPv6 this will be the /64 the actual host IP is in.  
[...]

<https://www.dragonresearchgroup.org/insight/http-report.txt>  
(sadly this feed is now apparently discontinued)

## Another Spamhaus Feed That Routinely Uses Varying Size Filter Rules...

115.47.0.0/**18** ; SBL299435

119.227.224.0/**19** ; SBL237235

120.46.0.0/**15** ; SBL262362

[...]

<https://www.spamhaus.org/drop/edrop.txt>

2803:5380:ffff::**48** ; SBL262056

2404:d880::**32** ; SBL297339

2401:c580::**32** ; SBL246818

2a05:4f80::**29** ; SBL257361

[...]

<https://www.spamhaus.org/drop/dropv6.txt>

## Bottom Line: Block What "Makes Sense" To Block

- If dealing with **static point sources** of unwanted traffic, adjacent IPs may be totally innocent, blocking IPv4 /32's make a lot of sense. Feel free to do that. It's the conservative choice, and easy enough if automated.
- However, if we know (for example, from Whois or from a trusted source) that an attacker has gained exclusive control over a specific network range, or if the attack source is from a dynamic address range where a single abused host may be constantly renumbering, **it may make more sense (or at least be more convenient) to just block a specific network range, instead.** This is where judgment (or third party data) may be most helpful.
- If you have no external data that will help you to more precisely localize, but you're seeing a concentrated range of IPs emitting unwanted traffic, blocking "typical" subnet-size chunks can be a pragmatic compromise.
- Blocking specific IPs is usually best, IF you can get away with it...

# How Many IPv4 /32's Might We Have to Block?

## Consider the Spamhaus CBL (Composite Block List)

Country	Listings	%total	% Total Listings	%cumulative Total Listings	Rank	Traffic	%TrafficS
Total	13516052	100				212040051	100
IN	2081481	15.40	15.40	15.40	1	2539920	1.20
CN	1450094	10.73	10.73	26.13	2	441896	0.21
BR	1101730	8.15	8.15	34.28	3	10916232	5.15
VN	1017885	7.53	7.53	41.81	4	1543952	0.73
US	882750	6.53	6.53	48.34	5	183846560	86.70
IR	603908	4.47	4.47	52.81	6	389152	0.18
RU	548861	4.06	4.06	56.87	7	658648	0.31
MX	390603	2.89	2.89	59.76	8	352560	0.17
PK	388031	2.87	2.87	62.63	9	304432	0.14
ID	343129	2.54	2.54	65.17	10	227776	0.11

- **13.5+ million listed v4 addresses** as of Nov 5 17:00:10 2016 UTC
- Note: just 10 countries account for nearly 2/3rds of all listed hosts
- The US alone accounts for nearly 87% of all spam traffic as sent!
- **Note:** you will *not* normally be loading the whole CBL as a filter!:-;

## Just How Patient Are You?

- Beyond the issue of system capacity, **how willing are you to play "whac-a-mole"\* all day?** If there is evidence that hundreds of hosts in a given subnet are sourcing problematic traffic, are you really going to handle that **IP-by-IP**? Or will you recognize that a pattern exists and deal with that **entire netblock** as an entity? Do you even HAVE any other realistic option if you're dealing with IPv6 addresses?
- Or are you willing to go still further and block **all the prefixes associated with an entire ASN or organization?**
- The key consideration is probably one of avoiding harm to innocent third parties, or "collateral damage." This implies keeping block sizes appropriate and understanding exactly what you're blocking. An example may help explain what I mean by this.

\* <https://en.wikipedia.org/wiki/Whac-A-Mole>



## **II. A Worked Example: ssh probes as seen in syslog data**

## syslog: A Terrific Resource That Too Few Look At

- syslog is a terrific basic security diagnostic tool, but (apparently) all too few bother to look at what gets logged.
- **PLEASE ENCOURAGE PEOPLE TO LOOK AT THEIR LOGS!**
- It will improve your situational awareness/motivation: people really are trying to gain access to your systems! (no, you're not paranoid)
- syslog also provides **hard data** to inform your blocking decisions. Making decisions based on data epitomizes the **scientific process**. Making decisions based on anything else is, well, an "artistic process" that I frankly don't get.

## Centralize Your syslogging?

- Use of a centralized syslogging facility (e.g. <https://syslog-ng.org/> ) can potentially take many individually-unscrutinized distributed logs and convert them into a single unified resource. It's almost like having your own network of security data collection nodes!
- Using centralized syslogging will:
  - Increase the chance that syslog data will get looked at (**who is responsible for looking at syslog data at YOUR site?**)
  - Eliminate the scenario where local/on-machine syslog data can end up being **wiped** by the attacker if a local node is successfully breached
  - Allow an integrated all-systems/"360 degree view" of attackers that may be doing an **interleaved "low-and-slow" attack** against multiple machines on your network all at once.

# The Basics of Logging Authentication Failures

- Checking **/etc/syslog.conf** (or just rummaging around `/var/log` a little), you should be able to figure out where authentication failures (failed logins) are getting logged on your local Unix systems. For example, one system's `syslog.conf` says:

`auth.info`

`/var/log/authlog`

- If you then look at `/var/log/authlog`, you should see a list of failed login attempts, such as:

`Nov 3 20:38:04 hostname sshd[6330]: Failed password for root from 116.31.116.23 port 51925 ssh2`

`Nov 3 20:41:40 hostname sshd[5282]: Failed password for root from 116.31.116.23 port 13412 ssh2`

`[etc]`

## root logins and sshd

- Just for the record, if you're seeing lots of sshd root login attempts, folks should know about a simple defense against that particular attack:
  - After verifying that you can su or sudo from some other account
  - In `/etc/ssh/sshd_config` set  
  
    PermitRootLogin no
  - Restart sshd  
  
    # `/etc/init.d/sshd restart`

## Attributing IPs to Responsible Parties: Whois

- **\$ whois 116.31.116.23**  
[...]  
inetnum:           **116.16.0.0 - 116.31.255.255**  
netname:           CHINANET-GD  
descr:             CHINANET Guangdong province network  
descr:             China Telecom  
descr:             No.31,jingrong street  
descr:             Beijing 100032  
country:           CN
- That's a big netblock, 116.16.0.0/12 (=16 x /16) or **1,048,576 IPv4 addresses** – sure would be nice if there was some sort of indication of how those addresses were re-assigned in whois, eh?

## Is There At Least a Crumby PTR Record for 116.31.116.23?

- A PTR record, or inverse-address record, maps an IP to one or more fully qualified domain names (potentially in an untrustworthy sort of way, but nonetheless).
- Unfortunately, many IP addresses (particularly in the APNIC region for some reason) will be found to NOT have PTR records. This is true for 116.31.116.23:

```
$ dig -x 116.31.116.23
```

```
[...]
```

```
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 5943
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1,  
ADDITIONAL: 0
```

## What Do We Find If We Check Passive DNS?

- There's not anything there for this specific IP in passive DNS:

```
$ dnsdb_query.py -i 116.31.116.23  
HTTP Error 404: Not Found
```

- *"What's that dnsdb\_query.py command?"*

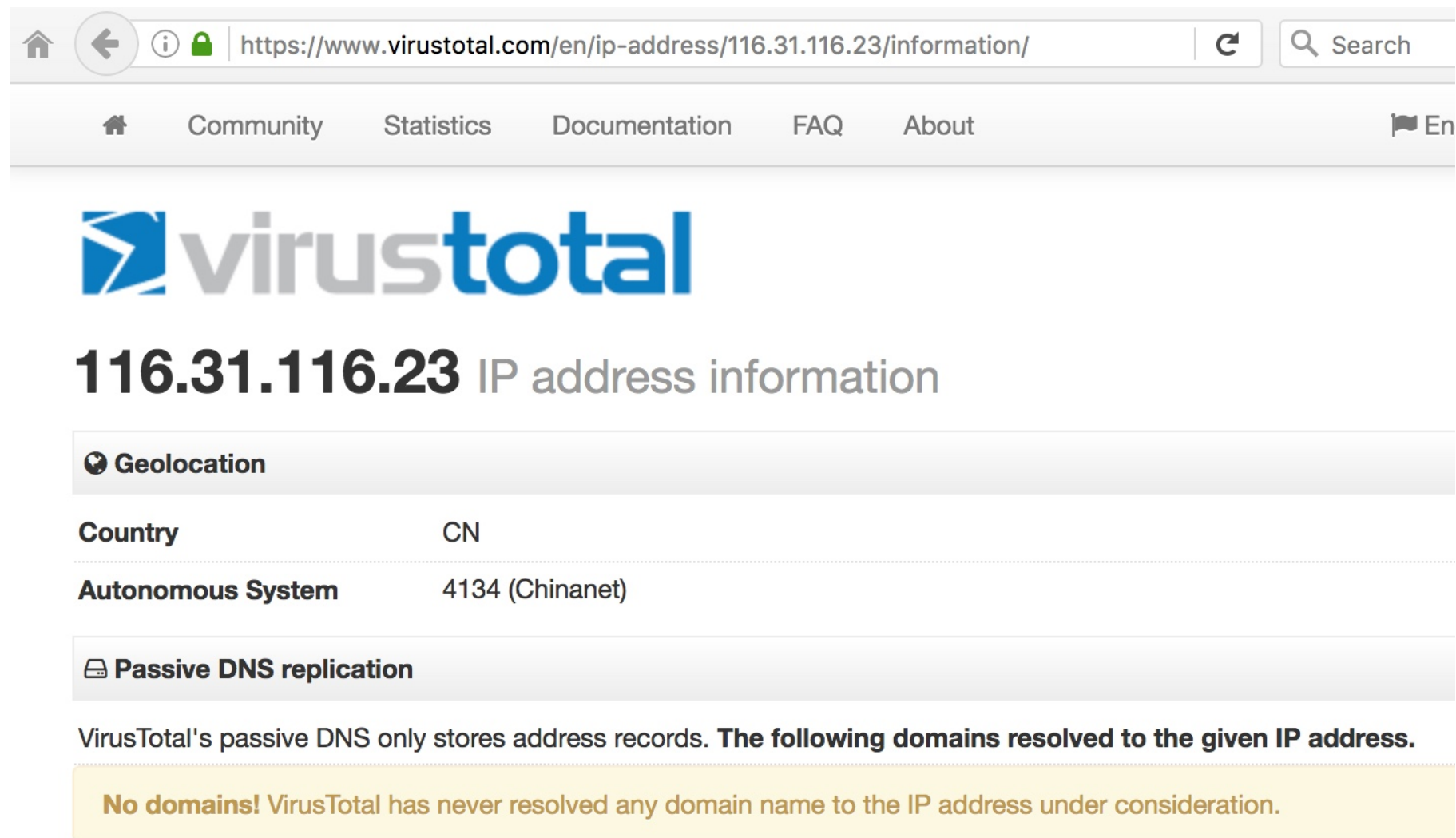
That's one (of multiple) ways you can access Farsight's DNSDB passive DNS API. See:

```
-- https://api.dnsdb.info/  
-- https://github.com/dnsdb/dnsdb-query
```

Other passive DNS implementations will have analagous query capabilities...



# Example of Checking Another Passive DNS Provider



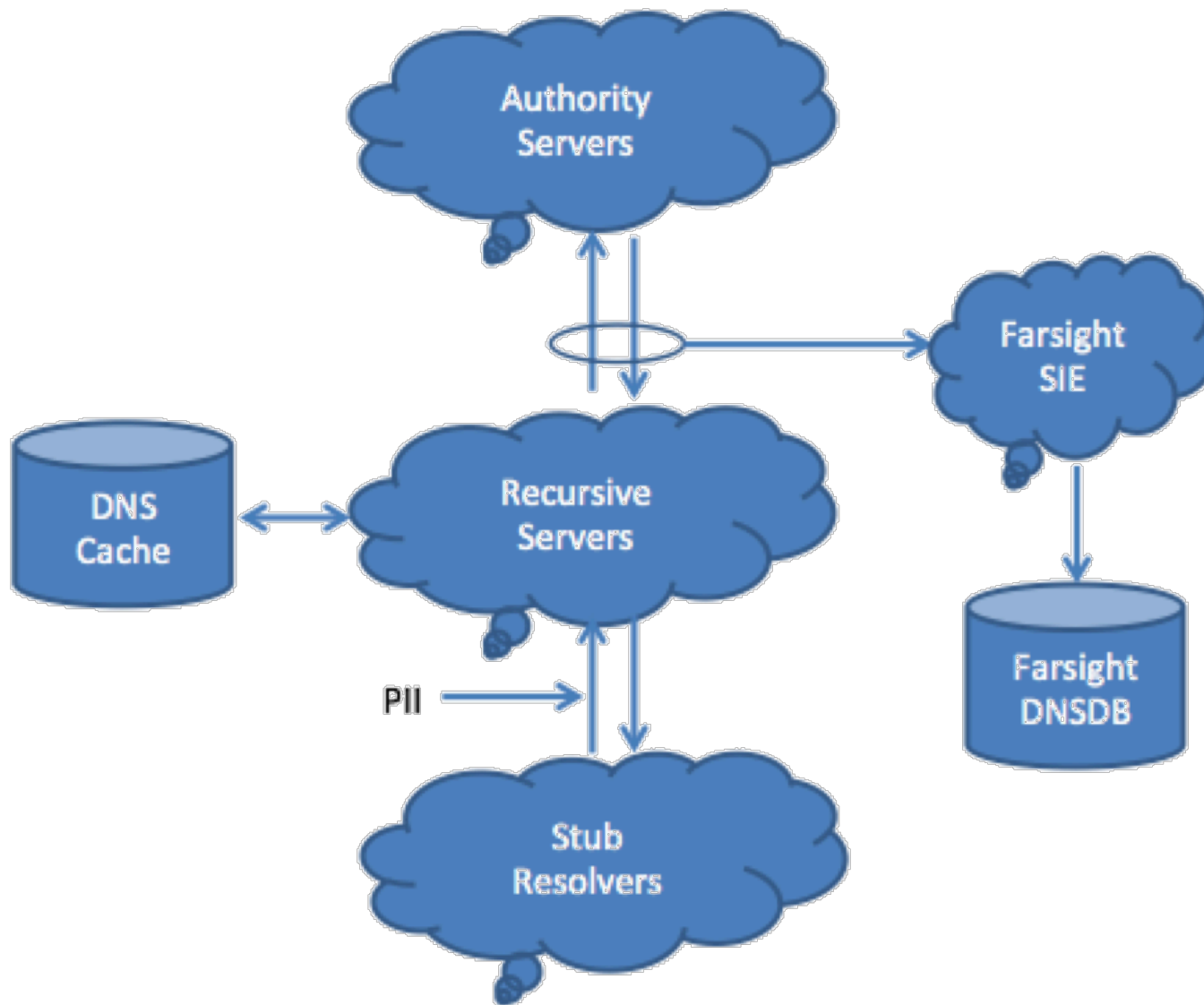
The screenshot shows the VirusTotal website interface. The browser's address bar displays the URL <https://www.virustotal.com/en/ip-address/116.31.116.23/information/>. The navigation bar includes links for Home, Community, Statistics, Documentation, FAQ, and About, along with a language selector set to English. The main header features the VirusTotal logo. Below the logo, the page title is "116.31.116.23 IP address information". A section titled "Geolocation" contains a table with the following data:

Country	CN
Autonomous System	4134 (Chinanet)

Below this, a section titled "Passive DNS replication" contains the text: "VirusTotal's passive DNS only stores address records. The following domains resolved to the given IP address." A yellow warning box at the bottom states: "No domains! VirusTotal has never resolved any domain name to the IP address under consideration."

## [Pink == Backfill]... What's "Passive DNS?"

- Passive DNS is a NoSQL database created by watching DNS traffic, typically cache miss traffic (queries and responses) captured above large recursive resolvers. By collecting passive DNS data above large recursive resolvers, queries seen appear to come from the resolver, not a specific user (so privacy gets preserved).
- Passive DNS allows a security analyst to get answers to queries that regular DNS can't address, including:
  - "What FQDNs have been seen on IP foo or in CIDR block bar?"
  - "What FQDNs use name server ns1.example.com?"
  - "Given FQDN alpha.bravo.charlie, where (what IP addresses) has that FQDN been seen using?"
  - "Given base domain bravo.charlie, what FQDNs are known to live under that base domain?"
- Bottom line -- passive DNS is useful for "pivoting" or taking an initial clue and finding related items ("guilt by association").



## Some Entities Offering Passive DNS Services

- Farsight Security, Inc. offers DNSDB (see <https://dnsdb.info/> )
  - DNSDB is a commercial (licensed) product, but **individual** law enforcement officers (LEOs), vetted academic researchers, and vetted-but-unfunded "Internet superheroes" can request free (grant) access to DNSDB.
- Some other passive DNS implementations include:
  - Florian Weimer's BFK, [http://www.bfk.de/bfk\\_dnslogger.html](http://www.bfk.de/bfk_dnslogger.html)
  - CERT.at/Aconet Passive DNS (inquire: [kaplan@cert.at](mailto:kaplan@cert.at) or [lendl@cert.at](mailto:lendl@cert.at))
  - CIRCL Passive DNS, <http://www.circl.lu/services/passive-dns/>
  - <http://passivedns.mnemonic.no/search/>
  - <https://www.opendns.com/enterprise-security/resources/data-sheets/investigate/>
  - <https://www.cs.auckland.ac.nz/research/groups/sde/dhdb-index.php>
  - VirusTotal, <https://www.virustotal.com/#search> (shown a few slides ago)
  - 360.cn Passive DNS, <https://www.passivedns.cn/help/>
- If I missed any other passive DNS sites, please drop me a note...

## Let's Also Check **Oregon Routeviews**

- **\$ telnet route-views.oregon-ix.net**  
login: **rviews**  
route-views>**show ip bgp 116.31.116.23**  
BGP routing table entry for **116.31.96.0/19**, version 93702655  
[...]  
3549 3356 4134 **134764**  
[...]  
**q**  
route-views>**exit**
- What does this tell us? **Well, we've now localized that abuse source from an IPv4 /12 (1,048,576 addresses) all the way down to an IPv4 /19, just 8,192 IPv4 addresses.**
- We also now have a responsible party, the ASN that's routing 116.31.96.0/19, AS134764.

## Don't Like telnet? There's Also Two DNS Zone

```
$ dig -t txt 23.116.31.116.aspath.routeviews.org +short  
"47872 3356 4134 134764" "116.31.96.0" "19"
```

You can easily write a little shell script or perl script to form asn or aspath zone queries for you. E.g., if you just want the origin asn:

```
$ ip2asn 116.31.116.23  
134764 116.31.116.23
```

See the next page.

Want the raw zone to run from a local server for lower latency, higher throughput, or total control? Mirror that zone locally. See <http://ftp.routeviews.org/dnszones/>

# The Little ip2asn shell Script

```
#!/bin/sh
```

```
origip=`echo $1`
```

```
revip=`echo $1 | sed 's/\([0-9]*\)\. \([0-9]*\)\. \([0-9]*\) \. \([0-9]*\) / \4.\3.\2.\1/'`
```

```
listing=`host -w -t txt ${revip}.asn.routeviews.org  
2>/dev/null | tail -1`
```

```
listing2=`echo ${listing} | awk '{print $4}' | sed  
's/"//g'`
```

```
echo "${listing2} ${origip}"
```

**Notes:** Required "host" command arguments may vary from system to system. This simple script shows only a single Route Origin ASN, and does not attempt to deal with Multiple ROAS. Beware tick vs **backtick** in the above, and **multi-line** commands. Inputs are not sanitized—not for use as a public Internet gateway

## Sources of BGP Routing Data

- In addition to Oregon Routeviews which we've already mentioned, see also:
- **RIPE's RIS Raw Data Service**  
<https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data>  
(including data from rrc18.ripe.net right here in Barcelona!)
- **Team Cymru's terrific IP to ASN mapping service**, see <http://www.team-cymru.org/IP-ASN-mapping.html>  
(please be sure to note and honor their terms of service for bulk queries!)



## But What Is "Routeviews"?

- Numerous ISPs contribute BGP routing table views to the Oregon Routeviews project, see <http://www.routeviews.org/>
- Routeviews was "originally conceived as a tool for Internet operators to obtain real-time information about the global routing system from the perspectives of several different backbones and locations around the Internet."
- That said, it (and similar BGP resources, such as RIPE's BGP data) are **also** terrific tools to support **data driven security research**.
- BGP data excels for four security-research-related purposes:
  - Identifying the **origin/length of the most compact enclosing CIDR prefix** that's routing an IP of interest
  - Finding **additional related network blocks** that are linked to an initial network address of interest
  - Tying **address blocks to ASNs for reputational assessment purposes**
  - Helping to identify appropriate **reporting targets** for problematic activity.

## Closest Encompassing CIDR Netblock

- In an ideal world, each provider would have **one contiguous netblock**, and that netblock would be advertised in aggregate via BGP with a consistent worldwide routing policy, etc.
- In the real world, providers often have **multiple discontinuous blocks**, or they **deaggregate large netblocks into smaller blocks** with each of those blocks taking up a slot in global routing tables.
- **In thinking about what to block, we'd normally want the most closely encompassing CIDR netblock around a source of badness.**
- We may find that by looking at IP Whois, inspecting PTR records, or in the current case, by checking BGP data.
- Yes, a bad guy certainly could have IP assets that span multiple netblocks as defined in IP Whois, or multiple BGP routing table entries, but these are the best options we have for hints about scoping initial filter blocks that I'm aware of.

## Backup, Please -- What's An "ASN?"

- Routing tables show who's announcing each CIDR prefix, and what's the best path to each of those prefixes. That path is expressed in terms of "autonomous system numbers," or "ASNs."
- Each ASN represents a network. Large organizations running their own networks (like a university, ISP, or large company), will usually have their own ASN.
- A convenient list of ASNs showing owners is available online at <http://bgp.potaroo.net/cidr/autnums.html>
- You can also lookup ASNs in Whois, just like any other Internet number resource.

## Whois AS134764? whois.apnic.net says...

- **\$ whois -h whois.apnic.net as134764** [select records only]  
aut-num: AS134764  
as-name: CT-FoShan-IDC  
descr: CHINANET Guangdong province network  
descr: Data Communication Division  
descr: 510000  
country: CN

"Foshan, formerly romanized as Fatshan, is a prefecture-level city in central Guangdong Province in southeastern China. [...] It forms part of the western side of the Pearl River Delta Economic Zone, which includes Guangzhou to the north and Shenzhen to the east. [...] Foshan is regarded as the home of the Cantonese forms of Chinese opera, kung fu, lion dancing, and dragon boat racing."

<https://en.wikipedia.org/wiki/Foshan>

# Have Others Seen Probes From That Same IP? Yep...

# Entries below consist of fields with identifying characteristics of a  
# a source IP address that has been seen attempting to remotely login to  
# a host using SSH password authentication. This report lists hosts  
# that are highly suspicious and are likely conducting malicious SSH  
# password authentication attacks. Each entry is sorted according to a  
# route origination ASN.

[...]

134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.4</b>	2016-11-06 00:59:44	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.5</b>	2016-11-06 00:59:50	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.6</b>	2016-11-06 01:00:00	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.7</b>	2016-11-01 14:47:57	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.8</b>	2016-11-06 00:20:31	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.9</b>	2016-11-06 00:59:56	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.10</b>	2016-11-06 00:59:29	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.11</b>	2016-11-06 00:59:36	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.14</b>	2016-11-06 00:59:31	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.17</b>	2016-11-06 00:59:48	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.18</b>	2016-11-06 00:59:41	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.20</b>	2016-11-06 01:59:52	sshpwauth
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.21</b>	2016-11-03 02:35:36	sshpwauth
<b>134764</b>	<b>  CT-FOSHAN-IDC CHINANET Guangdo</b>	<b>  116.31.116.23</b>	<b>  2016-11-06 00:59:59</b>	<b>  sshpwauth</b>
134764	CT-FOSHAN-IDC CHINANET Guangdo	<b>116.31.116.24</b>	2016-11-05 01:28:19	sshpwauth

[etc]

Source: <https://dataplane.org/sshpwauth.txt> (selected records only, sorted, emphasis added)

## Maybe Just Block Using 3rd Party Feeds?

- If <https://dataplane.org/sshpwauth.txt> site (or some other publicly available ssh threat feed) is reflecting the same attack sources that we're seeing, maybe it would make sense to pre-emptively block those traffic sources when they first appear in that threat feed? Doing so might let you automatically block most of the specific /32's that might eventually be hitting you.
- However, two points: **you must trust the operator of any third party operator you rely on**, and **the badness that anyone else sees might or might not correspond to the badness that you see now or will see in the future** (although I will say that the dataplane.org attackers seem to line up well with what I personally see, and I completely trust the operator of that resource).

## Default Deny, Whitelisting Exceptions?

- For some small sites with well defined populations, another option is to go to a default deny approach, only permitting connections from a closely defined list of allowed addresses. That is, for a first-match filter:

Allow as an exception: IP\_address\_1

Allow as an exception: IP\_address\_2

Otherwise (default option): deny

[etc]

- You can certainly try that approach if your needs allow, but we'll assume that this isn't a practical option for most larger sites, particularly if you have a user base that connects via dynamic IPs, etc.
- More common alternatives that can work well for larger sites or sites with authorized users connecting via dynamic IPs? See the next slide...

## Block Heuristically? Rate Limit? Preshared Keys Only?

- You could also just block ssh brute force attack traffic **automatically** with **fail2ban** ( <http://www.fail2ban.org/> ), etc.
- Some people don't like that approach, worrying that it may impact legitimate users or cause problems in heavily NAT'd environments, preferring a simpler approach that **rate limits** login attempts per IP per unit time, see <http://serverfault.com/questions/216995/is-it-worth-the-effort-to-block-failed-login-attempts> , suggesting, e.g.:

```
iptables -I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent --set
iptables -I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent --update
--seconds 60 --hitcount 4 -j DROP
```

- Use **ssh preshared keys** (only) (potentially leveraging smartcards such as [https://developers.yubico.com/PGP/SSH\\_authentication/](https://developers.yubico.com/PGP/SSH_authentication/) ) or require **multifactor** (e.g., see <https://duo.com/docs/duounix> )



## OR You Could Block Locally (And AT SCALE)

- Yet another option: maybe you and your authorized users don't ssh in from Asia and you're fed up with being probed, and you're willing to be more aggressive, perhaps blocking the encompassing /19 we saw in the Routeviews data (or even the entire original /12 we saw from the original IP Whois)?
- Before doing that, wouldn't it be nice to get an idea of what you'd be blocking, so you can avoid any unexpected false positives?
- And maybe, if you do check, you might find an interesting potential connection to the unwanted traffic you may be seeing, either a resource purchased/controlled by the attacker, or a third party whose site has been successfully attacked and compromised by an attacker? Let's check passive DNS, either via the web interface at <https://dnsdb.info> (max 10,000 results) or via the API (next slide).

## What Has DNSDB Seen For 116.31.96.0/19 During Just The Last Month?

```
$ dnsdb_query.py -l 1000000 --after=30d -i 116.31.96.0/19  
--sort=rdata > 116.31.96.0.txt
```

```
$ wc -l 116.31.96.0.txt
```

```
8879 116.31.96.0.txt
```

```
$ less 116.31.96.0.txt
```

```
[...]
```

```
ly.ff768.com. IN A 116.31.115.92
```

```
pay.qiqiuc.net. IN A 116.31.116.130
```

```
server1.qiqiuc.net. IN A 116.31.116.130
```

```
lixingnz.com. IN A 116.31.116.185
```

```
frenchgo.servegame.com. IN A 116.31.116.40
```

```
digitalpartnerconnect.com. IN A 116.31.116.7
```

```
yfwin.com. IN A 116.31.118.10
```

```
[...]
```

← note the IP

## What Does Domain Whois Say About **digitalpartnerconnect.com?** (selected records)

\$ **whois digitalpartnerconnect.com**

Domain Name: digitalpartnerconnect.com

Creation Date: 2016-04-11T05:12:55Z

Registrant Name: **DigitalPartner Solutions**

Registrant Organization: Digita Partner Solutions

Registrant Street: 0

Registrant Street: Ortigas Extebsion

Registrant City: Pasig

Registrant State/Province: ph

Registrant Postal Code: 1600

Registrant Country: PH

Registrant Phone: +63.09236800000

Registrant Email: info@**digitalpartnersolutions.com** ← note domain

## Chaining to **digitalpartnersolutions.com** (selected records)

\$ **whois digitalpartnersolutions.com**

Domain Name: DIGITALPARTNERSOLUTIONS.COM

Creation Date: 2015-03-09T14:47:24Z

Registrant Name: **R G**

Registrant Organization: Digita Partner Solutions

Registrant Street: One Oasis Ortigas Extension

Registrant City: Pasig

Registrant State/Province: NCR

Registrant Postal Code: 1600

Registrant Country: PH


Registrant Phone: **+63.0000000000**

← Cool phone number

Registrant Email: info@digitalpartnersolutions.com

[etc]

# Connecting A Person To That Company...



**R. G. F.**  
Lead Generation/ Telemarketing /Call Center Executive  
NCR - National Capital Region, Philippines | Marketing and Advertising


Current Digital Partner Solutions  
Education AMA University

Send R. G. InMail ▼

---

[!\[\]\(7e0d6a31a51eb3952a6a6daebf7e401c\_img.jpg\) https://ph.linkedin.com/in/](https://ph.linkedin.com/in/)

## Background

 Experience

**CEO**  
Digital Partner Solutions

October 2014 – Present (2 years 2 months) | Pasig, Philippines

Operate and manage call center.

Digital Partner Solutions is a marketing and sales firm. Our services focus on activities that promotes growth and revenue. We specialized on lead generation, customer acquisition, prospecting, sales conversion, direct marketing, telemarketing and online marketing.

## Netblock Reputation: What Else Has Been Seen In 116.31.96.0/19 During The Last 30 Days? ["Snow"]

**\$ less 116.31.96.0.txt**

[...]

yiye1.top. IN A 116.31.100.161  
yiye2.top. IN A 116.31.100.161  
yiye3.top. IN A 116.31.100.161  
yiye4.top. IN A 116.31.100.161  
yiye5.top. IN A 116.31.100.161  
yiye6.top. IN A 116.31.100.161  
yiye7.top. IN A 116.31.100.161  
dongpo1.top. IN A 116.31.100.161  
dongpo2.top. IN A 116.31.100.161  
dongpo3.top. IN A 116.31.100.161  
dongpo4.top. IN A 116.31.100.161  
dongpo5.top. IN A 116.31.100.161  
dongpo6.top. IN A 116.31.100.161  
juntao1.top. IN A 116.31.100.161

juntao2.top. IN A 116.31.100.161  
juntao3.top. IN A 116.31.100.161  
juntao4.top. IN A 116.31.100.161  
juntao5.top. IN A 116.31.100.161  
juntao6.top. IN A 116.31.100.161  
juntao7.top. IN A 116.31.100.161  
juntao8.top. IN A 116.31.100.161  
juntao9.top. IN A 116.31.100.161  
jzbao81.top. IN A 116.31.100.161  
jzbao82.top. IN A 116.31.100.161  
jzbao83.top. IN A 116.31.100.161  
jzbao84.top. IN A 116.31.100.161  
jzbao85.top. IN A 116.31.100.161  
jzbao86.top. IN A 116.31.100.161  
jzbao87.top. IN A 116.31.100.161  
[etc]

## Also From 116.31.96.0/19 During The Last 30 Days? [almost looks algorithmic/random, eh?]

**\$ less 116.31.96.0.txt**

[...]

aghwg.cn. IN A 116.31.123.191  
akavx.cn. IN A 116.31.123.191  
aogrz.cn. IN A 116.31.123.191  
apzls.cn. IN A 116.31.123.191  
bftzi.cn. IN A 116.31.123.191  
ccouw.cn. IN A 116.31.123.191  
dkcvh.cn. IN A 116.31.123.191  
drfle.cn. IN A 116.31.123.191  
dvrhp.cn. IN A 116.31.123.191  
dwsau.cn. IN A 116.31.123.191  
ekntf.cn. IN A 116.31.123.191  
elzjo.cn. IN A 116.31.123.191  
fabqy.cn. IN A 116.31.123.191  
fvmdt.cn. IN A 116.31.123.191

idvif.cn. IN A 116.31.123.191  
ifuvq.cn. IN A 116.31.123.191  
jdrfe.cn. IN A 116.31.123.191  
jkzbr.cn. IN A 116.31.123.191  
jpkur.cn. IN A 116.31.123.191  
ofeuj.cn. IN A 116.31.123.191  
oiasi.cn. IN A 116.31.123.191  
okivw.cn. IN A 116.31.123.191  
onfys.cn. IN A 116.31.123.191  
rjuyn.cn. IN A 116.31.123.191  
rkdhb.cn. IN A 116.31.123.191  
robkd.cn. IN A 116.31.123.191  
sijre.cn. IN A 116.31.123.191  
syubf.cn. IN A 116.31.123.191  
tvkkf.cn. IN A 116.31.123.191  
[etc]

## The Blocking Of That Which You Don't Want

- A bake-off of blocking options is really beyond the scope of today's talk
- We'll just mention one example by way of illustration: if your system uses **pf** (\*BSD, Mac, etc) and you like **pf**, try adding to `/etc/pf.conf`

```
table <unwanted> persist file "/etc/unwanted.txt"  
block in on interfacename from <unwanted> to any
```

where `/etc/unwanted.txt` contains a list of prefixes you want to filter.

- Remember to sanity check and then reload after any changes with:

```
pfctl -v -n -f /etc/pf.conf  
pfctl -f /etc/pf.conf
```

- **The Book of PF** by Hansteen is a nice potential addition to your library, or see the "PF - User's Guide" at <https://www.openbsd.org/faq/pf/>



## What If I Just Want to Block Prefixes in Postfix?

- Sometimes you may not want to block at the pf level. For example, maybe you just want to block access for a specific application, such as smtpd. We assume that you're running Postfix.
- If so, create /etc/postfix/client.cidr, with one CIDR prefix to block per line
- Reload after any changes with:

**# postmap client.cidr**

- Some Postfix documentation worth consideration:
  - **The Book of Postfix** by Hildebrandt and Koetter
  - **Postfix: The Definitive Guide** by Kyle Dent
  - The online documentation: <http://www.postfix.org/documentation.html>

## Having Blocked 116.31.96.0/19, 117.21.224.127 Soon Made Its Appearance As A Replacement...

- Nov 6 19:16:20 *hostname* sshd[30266]: Failed password for root from **117.21.224.127** port 3780 ssh2  
Nov 6 19:28:27 *hostname* sshd[12513]: Failed password for root from **117.21.224.127** port 3907 ssh2
- **\$ telnet route-views.oregon-ix.net**  
Username: rviews  
route-views>show ip bgp 117.21.224.127  
BGP routing table entry for **117.21.0.0/16**, version 92690576  
[etc]  
(that's "CHINANET Jiangxi province network" per whois)
- Lather, rinse, repeat...

## And Then There's...

- Nov 7 13:32:29 *hostname* sshd[4592]: Failed password for invalid user admin from **91.200.12.61** port 1604 ssh2

\$ **ip2asn 91.200.12.61**

35804 91.200.12.61

AS35804 = PP SKS-LUGAN (Ukraine), just 20,992 total IPv4 IPs

- Nov 7 18:25:36 *hostname* sshd[15332]: Failed password for root from **58.115.32.106** port 52768 ssh2

\$ **ip2asn 58.115.32.106**

9416 58.115.32.106

A9416 = Hoshin Multimedia Center, Inc (TW), **1,157,632 IPs**  
**(large provider)**

## And Then There's... (2)

- Nov 7 17:48:59 *hostname* sshd[1838]: Failed password for invalid user ubnt from **123.31.35.50** port 60391 ssh2

\$ **ip2asn 123.31.35.50**

45899 123.31.35.50

AS45899 = VNPT Corp (VN), **5,146,112 IPs (large provider)**

- Nov 7 19:38:38 *hostname* sshd[17544]: Failed password for root from **37.247.101.241** port 38376 ssh2

\$ **ip2asn 37.247.101.241**

199366 37.247.101.241

AS199366 = Yesilbir Bilisim Teknolojileri Bilgisayar Yayıncılık Sanayi ve Ticaret Ltd. Sti. (Turkey), **just 4,352 IPs**

## And Then There's... (3)

- Nov 7 20:16:27 *hostname* sshd[25639]: Failed password for root from **198.154.63.92** port 41213 ssh2

\$ **ip2asn 198.154.63.92**

26272 198.154.63.92

AS26272 = Fortaccloud (US), **just 10,240 IPs**

- Nov 7 23:12:50 *hostname* shd[8272]: Failed password for root from **155.133.82.159** port 53134 ssh2

\$ **ip2asn 155.133.82.159**

197226 155.133.82.159

AS197226 = "SPRINT" S.A. (PL), **just 15,360 IPs**

[etc., etc., etc.]

## And Then There's... (4)

- Nov 9 18:12:03 *hostname* sshd[3657]: Failed password for invalid user admin from **195.154.57.248** port 8746 ssh2

\$ **ip2asn 195.154.57.248**

12876 195.154.57.248

Online S.A.S, **311,296 IPv4 addresses (large provider)**

- Nov 9 18:23:33 *hostname* sshd[18770]: Failed password for root from **81.90.13.26** port 48392 ssh2

\$ **ip2asn 81.90.13.26**

12739 81.90.13.26

CJSC Netline, **just 10,240 IPv4 addresses**

[etc., etc., etc.] – do you have a new (boring) hobby?

## *Really* "Going For It:" Blocking Entire ASNs

- You may encounter some ASNs where legitimate uses of the ASN appears to be hard to find
- In that case, at least on more specialized systems, you may decide to block entire ASNs, particularly if the ASN is obscure and only routing a comparative handful of prefixes.
- **To be clear: this is a big step, and not one to be undertaken casually. I officially recommend that you only take this approach VERY RARELY.**
- If you decide you ARE going to go down this road, you'll need a list of **prefixes** to block since most end-user systems can't block an ASN (as such) directly. Remember to think about IPv4 **and** IPv6. Routeviews can give you a list of prefixes associated with a given ASN. Once you have those, you can filter them as you otherwise would. (Note that any new additional prefixes will not somehow automatically get added over time)

## Example: Finding All Prefixes Belonging to UO Itself

- `$ telnet route-views.oregon-ix.net`

login: **rviews**

route-views>**show ip bgp regex \_3582\$**

Network	Next Hop	Metric	LocPrf	Weight	Path
* <b>128.223.0.0</b>	208.51.134.254	2515		0	3549 3356 3701 35
[blah blah blah]					
* <b>163.41.128.0/17</b>	208.51.134.254	2515		0	3549 3356 3701 35
[blah blah blah]					
* <b>184.171.0.0/17</b>	208.51.134.254	2515		0	3549 3356 3701 3582 i
[blah blah blah]					
* <b>207.98.72.0/21</b>	208.51.134.254	2515		0	3549 3356 3701 3582 i
[blah blah blah]					

route-views>**quit**


- **Notes:**

- **This is just an example. I have no reason to believe that you should block all traffic from UO!**
- When searching for prefixes originated by an ASN, be careful not to overmatch (the `_` before UO's ASN mean "must be a space before the ASN" and the `$` after the ASN says that the "ASN must be at the end of the ASPath" (e.g., AS3582 is the **originating** ASN, not a **transit** ASN)



# An Easier Alternative Approach To Finding Prefixes

bgp.he.net/AS3582#\_prefixes





 HURRICANE ELECTRIC  
INTERNET SERVICES

AS3582 University of Oregon

**Quick Links**

- [BGP Toolkit Home](#)
- [BGP Prefix Report](#)
- [BGP Peer Report](#)
- [Exchange Report](#)
- [Bogon Routes](#)
- [World Report](#)
- [Multi Origin Routes](#)
- [DNS Report](#)
- [Top Host Report](#)
- [Internet Statistics](#)
- [Looking Glass](#)
- [Network Tools App](#)
- [Free IPv6 Tunnel](#)
- [IPv6 Certification](#)
- [IPv6 Progress](#)
- [Going Native](#)
- [Contact Us](#)

AS Info Graph v4 Graph v6 Prefixes v4 Prefixes v6 Peers v4 Peers v6 Whois IRR


Prefix	Description
<a href="#">128.223.0.0/16</a>	University of Oregon 
<a href="#">163.41.128.0/17</a>	University of Oregon 
<a href="#">184.171.0.0/17</a>	University of Oregon 
<a href="#">207.98.72.0/21</a>	Oregon Exchange 

Updated 08 Nov 2016 12:30 PST © 2016 Hurricane Electric

[An amusing aside: I once had someone ask, "Is this sort of information supposed to be available?" :-) ]

# And Don't Forget About IPv6, Too...

Home Back Info bgp.he.net/AS3582#\_prefixes6 Search



 **HURRICANE ELECTRIC**  
INTERNET SERVICES  Search

**AS3582 University of Oregon**

**Quick Links**

- [BGP Toolkit Home](#)
- [BGP Prefix Report](#)
- [BGP Peer Report](#)
- [Exchange Report](#)
- [Bogon Routes](#)
- [World Report](#)
- [Multi Origin Routes](#)
- [DNS Report](#)

AS Info Graph v4 Graph v6 Prefixes v4 **Prefixes v6** Peers v4 Peers v6 Whois IRR

Prefix	Description
<a href="#">2001:468:d01::/48</a>	Oregon GigaPOP 
<a href="#">2607:8400::/32</a>	University of Oregon 

Updated 09 Nov 2016 12:24 PST © 2016 Hurricane Electric

## If You Are Trying To Block An Entire Entity...

- While most entities use only a single ASN, some entities may use **more than one ASN**.

If you're attempting to block an entire specific entity, be sure to identify ALL the ASNs they may be using so that you can expand ALL those ASNs into the corresponding prefixes to block. Sometimes you can readily find these on a list of ASNs, otherwise you may need to scrutinize the Hurricane Electric site's BGP graphs for hints.

- In other cases, **some entities may NOT have an ASN of their own**. They may simply be announced as part of some larger organization's network. **Blocking large shared ASNs is virtually ALWAYS a BAD idea.**
- **If you see lots of IPs or lots of prefixes, I'd ALSO strongly encourage you to think twice about blocking by ASN.** The bigger the ASN, the greater the likelihood you'll cause unacceptable collateral damage.

# Factors Impacting Collateral Damage Levels

- You're at more risk of interfering with legitimate traffic if the candidate ASN is from:
  - Your own home country
  - Uses the same language(s) that you and your users use
- Other attributes that correlate with a higher probability of problems:
  - A **large** user base (the bigger the user base, the greater the statistical probability that **someone** will want to interact with any given prefix)
  - **A diverse international user base** (universities tend to have students from "everywhere;" likewise international businesses may transact business with overseas customers from all over the place, for example)
  - **High value interactions** (if the loss of even a single transaction might have an impact involving millions of dollars, broad filtering can become more potentially expensive)
  - **High user expectations for ease of use and universal availability** (if you say "we'll bring you the whole world," don't block big chunks of it!)
  - **No way for users to opt out of the filtering.**

## "Courtesy Notices" To The Party Getting Blocked?

- I should also touch on the idea of courtesy notices – some polite folks believe that it is good form to drop a responsible party at a network that's about to be blocked a note **before** you block them. I get that notion in theory, however in practice...
- **Abuse reporting contact addresses at troubled networks will often bounce** (the address may now be non-existent, or the mailbox may be full), or the mailbox may go unread even if it's able to be delivered. (You may not be the **only one** sending courtesy notices/complaints)
- **You're likely wasting your time** (a well run network will be monitoring its traffic with Netflow, and shouldn't need reports from 3<sup>rd</sup> parties to alert it to problematic behaviors (like outbound ssh to thousands of hosts))
- This may be a bit like an **ant notifying an elephant** that he will no longer be saying "hello" in the morning (**the third party provider may not care**)
- **You may actually be helping to educate the bad guys** (your notice may be passed along verbatim, directly to them, or the bad guys may actually be running some networks)

## Sharing Data With **Trusted Third Parties**

- On the other hand, if you do have the time and inclination, you may want to **share information about badness you've noticed with trusted third parties**, including national or regional CERTs, ISACs, etc.
- By sharing data, you have the ability to positively impact not just the security of your own site, but also the security of other sites.
- Sharing data will often result in others reciprocally sharing data with you.
- Data sharing can be as simple as manual email to a trusted mailing list, or you may want to investigate a scalable data sharing format such as the Collective Intelligence Framework (see <http://csirtgadgets.org/> )
- Be sure to pay attention to any applicable laws or regulations:  
I am not an attorney and cannot provide legal advice, but you **should talk with an attorney** before embarking on any new data sharing project.
- **If your network is potentially interested in sharing passive DNS data with Farsight, we'd be glad to hear from you (the better/more inclusive the data in passive DNS, the better the decisions that people can make from it... including decisions NOT to block due to collateral damage)**

### **III. Worked Example Two: Unwanted Mail**

# A Piece of Blocked Email That Was Meant For Me

- Connecting host: **yeua7oi.mastann.top** [69.162.69.121]
- EHLO **gaui8as.ffcheek.top**
- MAIL FROM:<**Toenail-Fungus-Awareness@gaui8as.ffcheek.top**> BODY=7BIT  
RET=HDRS
- Let's begin with **three points** relating to the connecting host's domain name...
  - 1) What do we know about the **"top"** TLD?
  - 2) What does the **Valli Multi RBL checker** think about the base domain (mastann.top) – is it listed on any block lists?
  - 3) Once we've checked that, then let's check the **domain Whois info**



# Spamhaus' Assessment of Most Abused TLDs

← ⓘ 🔒 <https://www.spamhaus.org/statistics/tlds/> ↻ 🔍 Search

## The World's Most Abused TLDs

TLD Check

TLD Result

Top Level Domain (TLD) registries which allow registrars to sell high volumes of domains to professional spammers and malware operators in essence aid and abet the plague of abuse on the Internet. Some registrars and resellers knowingly sell high volumes of domains to these actors for profit, and many registries do not do enough to stop or limit this endless supply of domains.

A TLD may be "bad" in two ways. On one side, the ratio of bad to good domains may be higher than average, indicating that the registry could do a better job of enforcing policies and shunning abusers. However, some TLDs with a high fraction of bad domains may be quite small, and their total number of bad domains could be relatively limited with respect to other, bigger TLDs. Their total "badness" to the Internet is limited by their small total size.

## The 10 Most Abused Top Level Domains

As of 06 November 2016 the TLDs with the worst reputations for spam operations are:

1	<b>.science</b>	<b>Badness Index: 9.93</b>	Domains seen: 44,121 Bad domains: 41,227 (93.4%)
2	<b>.top</b>	<b>Badness Index: 7.22</b>	Domains seen: 647,451 Bad domains: 364,859 (56.4%)
3	<b>.stream</b>	<b>Badness Index: 6.28</b>	Domains seen: 11,472 Bad domains: 8,015 (69.9%)
4	<b>.download</b>	<b>Badness Index: 5.84</b>	Domains seen: 15,670 Bad domains: 9,941 (63.4%)
			Domains seen: 21,744

**For those in the back, that's 364,859 bad domains, 56.4% of all .top domains are bad... wow.**

# Partial Report From The Valli Multi-RBL Checker for mastann.top: It **\*IS\* Block Listed**

<div> <input type="text" value="multirbl.valli.org/lookup/mastann.top.html"/> <input type="text" value="Search"/> </div>				
533	mastann.top	Spam Eating Monkey SEM-FRESH10	fresh10.spameatingmonkey.net	Listed
	Query:	mastann.top.fresh10.spameatingmonkey.net		
	A Record:	127.0.0.2		
	TTL:	300		
	TXT:	<ul style="list-style-type: none"> <li>Domain first seen 2016-10-27, see <a href="http://spameatingmonkey.com/lookup/mastann.top">http://spameatingmonkey.com/lookup/mastann.top</a></li> </ul>		
534	mastann.top	Spam Eating Monkey SEM-FRESH15	fresh15.spameatingmonkey.net	Listed
	Query:	mastann.top.fresh15.spameatingmonkey.net		
	A Record:	127.0.0.2		
	TTL:	300		
	TXT:	<ul style="list-style-type: none"> <li>Domain first seen 2016-10-27, see <a href="http://spameatingmonkey.com/lookup/mastann.top">http://spameatingmonkey.com/lookup/mastann.top</a></li> </ul>		
530	mastann.top	Spam Eating Monkey SEM-URI	uribl.spameatingmonkey.net	Not listed
531	mastann.top	Spam Eating Monkey SEM-URIRED	urired.spameatingmonkey.net	Not listed
640	mastann.top	Spamhaus DBL Domain Block List	dbl.spamhaus.org	Listed
	Query:	mastann.top.dbl.spamhaus.org		
	A Record:	127.0.1.2		
	TTL:	60		
	TXT:	<ul style="list-style-type: none"> <li><a href="https://www.spamhaus.org/query/domain/mastann.top">https://www.spamhaus.org/query/domain/mastann.top</a></li> </ul>		
813	mastann.top	Suomispam Domain Blacklist	dbl.suomispam.net	Not listed
457	mastann.top	SURBL multi (Combined SURBL list)	multi.surbl.org	Listed
	Query:	mastann.top.multi.surbl.org		
	A Record:	127.0.0.64		
	TTL:	180		
	DB_rc:	<ul style="list-style-type: none"> <li>jp data source (jwSpamSpy + Prolocation sites)</li> </ul>		
	TXT:	<ul style="list-style-type: none"> <li>Blocked, mastann.top on lists [abuse], See: <a href="http://www.surbl.org/lists.html">http://www.surbl.org/lists.html</a></li> </ul>		
40	mastann.top	Swiss URIBL	uribl.swiss.ch	Not listed

## Domain Whois For Connecting Host (Select Records)

\$ **whois mastann.top**

Domain Name: mastann.top

Updated Date: **2016-10-26T06:54:29Z**

← Recently registered  
and updated

Creation Date: 2016-10-26T06:34:39Z

Sponsoring Registrar: **Alpnames Limited**

← See next slide

Domain Status: clientTransferProhibited

<https://www.icann.org/epp#clientTransferProhibited>

Registrant Name: Cecille Wynn

Registrant Street: **Roissy Charles de Gaulle**

← The French airport

Registrant City: Roissy-en-France

[and M3AAWG was meeting  
in Paris at the time...]

Registrant State/Province: Paris

Registrant Postal Code: 95700

Registrant Country: FR

Registrant Phone: +33.174258418

Registrant Email: **alw6nrdlpdysambm3aum3krrwhva32j6pjeyqqln@yahoo.com**

← That's sure a memorable email address!

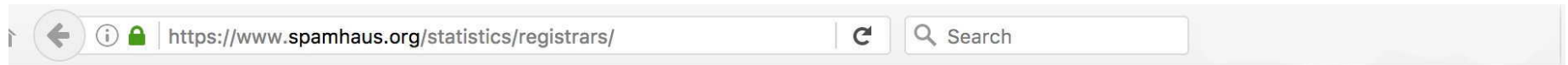
[etc]

Name Server: **mimi.ns.cloudflare.com**

← Look forward 2 slides

Name Server: **brett.ns.cloudflare.com**

# The Relevance of Alpnames?



## The World's Most Abused Domain Registrars

Among the reasons spam, malware and other threats continue to plague the internet is that abusers find it easy to obtain an endless supply of domain names. Some [gTLD](#) and [ccTLD](#) resellers (called registrars) sell large volumes of domains to professional spammers and other miscreants for profit. Some registrars have been directly owned and operated by abusers, while others simply do not do enough to stop or limit bad guys' access to an unlimited supply of domains. Abusers destroy the reputation of those domains (and along with them, possibly the reputation of registrars and registries) and just move on to new ones in a vicious cycle.

A registrar may be "bad" in two ways. On one side, the ratio of bad to good domains may be higher than average, indicating that the registrar could do a better job of enforcing policies and shunning abusers. However, some registrars with a high fraction of bad domains may be quite small businesses, and their total number of bad domains could be relatively limited with respect to other registrars. Their total "badness" to the Internet is limited by their small total size.

## The 10 Most Abused Domain Registrars

As of 05 November 2016 the registrars with the worst reputations for spam domains are:

1	<b>Domainers Choice (Nanjing Imperiosus Technology)</b>	Badness Index: <b>8.71</b> Domains seen: 10,260 Bad domains: 9,735 ( <b>94.9%</b> )
2	<b>Alpnames</b>	Badness Index: <b>8.64</b> Domains seen: 163,730 Bad domains: 120,877 ( <b>73.8%</b> )
3	<b>Todaynic</b>	Badness Index: <b>6.00</b> Domains seen: 12,112 Bad domains: 8,076 ( <b>66.7%</b> )
4	<b>GMO</b>	Badness Index: <b>5.44</b> Domains seen: 182,480 Bad domains: 87,259 ( <b>47.8%</b> )
		Badness Index: <b>3.50</b>

That's 73.8% bad domains according to Spamhaus

## And Cloudflare?

- Cloudflare acts as a **reverse proxy**, interposing itself between a web site and the Internet. This is very useful service if your site has become the target of a DDoS attack.
- HOWEVER, from the point when you put your "real" web site behind Cloudflare, traffic for your web site then goes to one of Cloudflare's IPs; **only you and Cloudflare know your "real IP."**
  - This means that third parties can't lookup the real IP address in the hope that you'll find clues to the identity of the web site's owner
  - Nor can you potentially determine the span of IPs the site owner controls
- Cloudflare protected domains also typically use **Cloudflare name servers; those servers are shared by many unrelated sites.**
  - This means that you also can't use passive DNS to find just a set of closely related domains that all use a unique name server
  - And you can't do name server → IP, and look for other name servers sharing the same IP using passive DNS.
- This arrangement frustrates both domain and IP reputation work



## An Example Of A Cloudflare-Protected Domain

Domain Name: **int-security-paypal[dot]com**

Registrant Name: fuzay nsidik

Registrant Street: ABC

Registrant City: ABC ← ABC is lovely in the springtime :-)

Registrant State/Province: Arizona

Registrant Postal Code: 87102 ← That's an Albuquerque, NM zip)

Registrant Country: US

Registrant Phone: +1.3331143019 ← Guadalajara, Mexico area code

Registrant Email: sidikccuk@gmail.com

[...]

Name Server: **MERLIN.NS.CLOUDFLARE.COM**

Name Server: **TANI.NS.CLOUDFLARE.COM**

# And Another Example...

The screenshot shows the homepage of **Microsoft Online Key**. The browser's address bar displays **www.microsoftonlinekey.com**. The website's navigation bar includes links for **Windows 10**, **Windows 8**, **Windows 7**, **Windows Server**, and **Office**, along with a **Cart: (0)** and a **Search** button. The main heading, **Windows 7/8/10 Product Key Sale & Office 2016/2013 ISO Download**, is highlighted with a red box. Below this, a welcome message states: "Welcome to MicrosoftOnlineKey.com, Windows 10,8.1,7 Product Key & Office 2016/2013 ISO Download online store. Our quality license Key helps us succeed in the severe competition. It's genuine, affordable and backed by professional services and dedicated support from our computer experts, 24x7. Our whole-life guarantee releases all your worries and concerns. Besides, our dedicated customer service, Free download and lowest prices are another kind of guarantee." The **Hot Products** section features four items:

Product	Price	Description
<b>Windows 10 Pro</b>	\$47.99	100% Genuine Windows 10 Pro Key & ISO Download
<b>Office Professional Plus 2016</b>	\$49.99	100% Genuine Office Professional Plus 2016 Key & ISO Download
<b>Windows 8.1 Professional</b>	\$39.99	100% Genuine Windows 8.1 Professional Key & ISO Download
<b>Windows 7 Ultimate SP1</b>	\$32.99	100% Genuine Windows 7 Ultimate SP1 Key & ISO Download

Domain Name: MICROSOFTONLINEKEY[dot]COM  
[...]

Name Server: **ASHLEY.NS.CLOUDFLARE.COM**

Name Server: **DILBERT.NS.CLOUDFLARE.COM**

## Our Original Domain...

- Coming back to yeua7oi.mastann.top, it's clearly suspicious
- In this case, however, the FQDN is **NOT** using a CloudFlare IP.

The connecting mail server is actually using the IP address  
**69.162.69.121**

Is that IP address suspicious?

- Let's start by checking IP Whois for it...



## IP Whois (Selected Records)

\$ whois 69.162.69.121

**Private Customer** LSN-DLLSTX-1 (NET-69-162-69-112-1) 69.162.69.112 - 69.162.69.127

Limestone Networks, Inc. LSN-DLLSTX-2 (**NET-69-162-64-0-1**) 69.162.64.0 - 69.162.127.255

\$ whois -h whois.arin.net NET-69-162-69-112-1

CIDR: **69.162.69.112/28**

NetName: LSN-DLLSTX-1

Customer: **Private Customer** (C06228828) ← <https://whois.arin.net/rest/customer/C06228828.html> says that's "this customer's" only netblock

RegDate: 2016-10-04

Updated: 2016-10-04

CustName: **Private Customer**

Address: **Private Residence**

City: Pathankot

StateProv: AG

PostalCode: 145101

Country: **IN**

RegDate: **2016-10-04**

Updated: 2016-10-04

# What Does Passive DNS Know About That /28?

```
$ dnsdb_query.py -i 69.162.69.112/28 --after=90d | awk '{print $1}' | 2nd-level-dom |  
sort -u | reverse-domain-names | sort > temp.txt
```

com.e8m	top.bloodsugaropened	top.manylottowinner
com.fmcgdirectory	top.chatasianwomenactive	top.mastann
com.jodhpurhomestay	top.colourp	top.oancomp
com.kickerrorstoday	top.continuerewards	top.onlylosshearingback
com.kristalklaws	top.eranamt	top.ourbrainhealthboss
com.nepalipoem	top.fallenp	top.pastrrx
com.nepalityper	<b>top.ffcheek</b> ← EHLO	top.pencilp
com.roomstoletin	top.ffmarch	top.presentsnoringcure
com.timeandupdate	top.ffraven	top.reallyloanpersonal
in.zebralive	top.fftread	top.seflare
np.com.suryodayalaghubitta	top.figurelosshearing	top.seprobe
np.org.upakarsaccos	top.humogra	top.somerrx
org.aicfellowship	top.inbyeneuropathypain	top.stakedx
org.bethelagindia	top.jeffort	<b>top.stseize</b> ← See next slide
org.christianrevelation	top.kusmoon	top.tiedrrx
org.hamrosambandha	top.lczflex	top.wyseare
top.amaterialswoodworking	top.lczitem	
top.backearlossalthough	top.lczknow	

## stseize.top: This email pattern look familiar?

\$ **whois stseize.top**

Domain Name: stseize.top

Updated Date: 2016-10-21T08:21:55Z

Creation Date: 2016-10-21T06:56:30Z

Registry Expiry Date: 2017-10-21T06:56:30Z

Sponsoring Registrar: **Alpnames Limited**

Domain Status: clientTransferProhibited <https://www.icann.org/epp#clientTransferProhibited>

Registrant Name: Mathilde Craft

Registrant Street: Antistaseos 36 ← <https://www.flickr.com/photos/mckroes/5502159284>

Registrant City: Chalandri says that's a McDonald's...

Registrant State/Province: Attiki

Registrant Postal Code: 15232

Registrant Country: GR

Registrant Phone: +30.2106800852

Registrant Email: **b45gc5ictwlu7z7tpgyg2xwb7cwiguqexgdp5qe2@yahoo.com**

[...]

Name Server: **sue.ns.cloudflare.com**

Name Server: **jeff.ns.cloudflare.com**

## "2nd-level-dom? reverse-domain-names?"

- 2nd-level-dom is a small Perl script which, given a fully qualified domain name, returns the effective 2nd-level domain name.

```
$ echo "www.google.com" | 2nd-level-dom  
google.com
```

- reverse-domain-names is another Perl script; this one reverses the labels of a domain for ease of sorting and display, and for reduced risk of accidental blockage if "interesting" domains are mentioned in a document that's subsequently emailed, etc.

```
$ echo "www.springfield.k12.or.us" | reverse-domain-names  
us.or.k12.springfield.www
```

- See the next slides for copies of these small scripts...

## 2nd-level-dom

```
#!/usr/bin/perl
use strict;
use warnings;
use IO::Socket::SSL::PublicSuffix;
my $pslfile = 'your_path_here/effective_tld_names.dat';
my $ps = IO::Socket::SSL::PublicSuffix->from_file($pslfile);
my $line;
foreach $line (<>) {
    chomp($line);
    my $root_domain = $ps->public_suffix($line,1);
    printf( "%s.\n", $root_domain );
}
```

The required data file? See <https://publicsuffix.org/list/>

## reverse-domain-names

```
#!/usr/bin/perl

my @lines = <>;
chomp @lines;

@lines =
    map { join ".", reverse split /\./ }
    sort
    @lines;

print "$_\n" for @lines;
```

## It Isn't *Just* One Netblock...

- Our first netblock was **69.162.69.112/28**. Now look at this observation, part of **69.162.93.96/28**
- Connecting host: p0kilee.yardalb.top[**69.162.93.110**]  
EHLO p0kilee.yardalb.top  
MAIL FROM:<**Grocery\_Coupons**@p0kilee.yardalb.top> BODY=7BIT RET=HDRS

Private Customer LSN-DLLSTX-1 (NET-69-162-93-96-1) 69.162.93.96 - 69.162.93.111  
Limestone Networks, Inc. LSN-DLLSTX-2 (NET-69-162-64-0-1) 69.162.64.0 -  
69.162.127.255

NetRange: 69.162.93.96 - 69.162.93.111  
CIDR: 69.162.93.96/28  
Customer: Private Customer (C06233555)  
RegDate: 2016-10-11  
CustName: Private Customer  
Address: Private Residence  
City: Silay City  
StateProv: AG  
PostalCode: 6116  
Country: PH  
RegDate: 2016-10-11

## Or How About *This* One...

- Not **69.162.69.112/28** or **69.162.93.96/28**, now **69.162.124.128/28**
- Connecting host: unknown[69.162.124.140]  
EHLO a3houre.racketn.top  
MAIL FROM:<**Medicare\_Supplemental**@a3houre.racketn.top> BODY=7BIT RET=HDRS

Limestone Networks, Inc. LSN-DLLSTX-2 (NET-69-162-64-0-1) 69.162.64.0 - 69.162.127.255

Private Customer LSN-DLLSTX-1 (NET-69-162-124-128-1) 69.162.124.128 - 69.162.124.143

**\$ whois -h whois.arin.net NET-69-162-124-128-1**

NetRange: 69.162.124.128 - 69.162.124.143  
CIDR: 69.162.124.128/28  
Customer: Private Customer (C06242492)  
RegDate: 2016-10-22  
CustName: Private Customer  
Address: Private Residence  
City: Pathankot  
StateProv: AG  
PostalCode: 145101  
Country: IN



## Or *This* One...

- Not **69.162.69.112/28**, **69.162.93.96/28**, **69.162.124.128/28**, but **69.162.64.48/28**
- Connecting host: azik2lep.takefreeenergydevice.top[69.162.64.53]  
EHLO azik2lep.takefreeenergydevice.top  
MAIL FROM:<**Tesla-Free-Energy**@azik2lep.takefreeenergydevice.top> BODY=7BIT  
RET=HDRS
- Limestone Networks, Inc. LSN-DLLSTX-2 (NET-69-162-64-0-1) 69.162.64.0 - 69.162.127.255  
Private Customer LSN-DLLSTX-1 (NET-69-162-64-48-1) 69.162.64.48 - 69.162.64.63
- **\$ whois -h whois.arin.net NET-69-162-64-48-1**  
NetRange: 69.162.64.48 - 69.162.64.63  
CIDR: 69.162.64.48/28  
Customer: Private Customer (C06243694)  
RegDate: 2016-10-24  
Address: Private Residence  
City: Pathankot  
StateProv: AG  
PostalCode: 145101  
Country: IN  
RegDate: 2016-10-24

## Rather Than Blocking /28 After /28, Could We Get Away With Just Blocking 69.162.64.0/18 ?

- After all, that's "just" 16,384 IPv4 IPs, right? How "hot" is that entire block?
- ```
$ dnsdb_query.py -l 1000000 --after=90d -i 69.162.64.0/18 > temp.txt
```

```
$ wc -l temp.txt
```

```
791007 temp.txt
```
- **That's a pretty busy netblock...** collateral damage would likely be significant.
- *And would blocking that range catch everything relevant, anyway?*

## Other "Private Customer" "LSN-DLLSTX-1"?

If we were to block the one large block, would that catch everything relevant, such as all LSN-DLLSTX-1 customers with StateProv: AG?

Maybe not. Sorted by date, a partial list:

**\$ whois -h whois.arin.net NET-69-162-80-48-1**

|                   |                              |
|-------------------|------------------------------|
| NetRange:         | 69.162.80.48 - 69.162.80.63  |
| CIDR:             | 69.162.80.48/28              |
| NetName:          | LSN-DLLSTX-1                 |
| Customer:         | Private Customer (C05599535) |
| RegDate:          | <b>2015-02-16</b>            |
| Address:          | Private Residence            |
| City:             | Wanchai                      |
| <b>StateProv:</b> | <b>AG</b>                    |
| PostalCode:       | <b>72022</b>                 |
| Country:          | <b>HK</b>                    |

## Related To?

**\$ whois -h whois.arin.net NET-216-245-210-80-1**

NetRange: 216.245.210.80 - 216.245.210.95

CIDR: 216.245.210.80/28

NetName: LSN-DLLSTX-1

Customer: Private Customer (C05690479)

RegDate: **2015-04-21**

Address: Private Residence

City: Xiamen

StateProv: **AG**

PostalCode: **72022**

Country: **CN**

## Plus Of Course...

**\$ whois -h whois.arin.net NET-216-245-206-88-1**

NetRange: 216.245.206.88 - 216.245.206.95

CIDR: 216.245.206.88/29

NetName: LSN-DLLSTX-1

Customer: Private Customer (C05696618)

RegDate: **2015-04-24**

Address: Private Residence

City: xiamen

**StateProv: AG**

**PostalCode: 72022**

**Country: CN**

## And Its Friend...

**\$ whois -h whois.arin.net NET-74-63-243-176-1**

NetRange: 74.63.243.176 - 74.63.243.191

CIDR: 74.63.243.176/28

NetName: LSN-DLLSTX-1

Customer: Private Customer (C05706595)

RegDate: **2015-04-30**

Address: Private Residence

City: Jaipur

StateProv: **AG**

PostalCode: **72022**

Country: **IN**

## And...

**\$ whois -h whois.arin.net NET-216-144-241-172-1**

NetRange: 216.245.223.160 - 216.245.223.175

CIDR: 216.245.223.160/28

NetName: LSN-DLLSTX-1

Customer: Private Customer (C05911796)

RegDate: **2015-09-24**

CustName: Private Customer

Address: Private Residence

City: Canterbury

**StateProv: AG**

PostalCode: **CT2 8DD** ← well, that's a change!

Country: **GB**

## And...

**\$ whois -h whois.arin.net NET-74-63-197-168-1**

NetRange: 74.63.197.168 - 74.63.197.171  
CIDR: 74.63.197.168/30  
NetName: LSN-DLLSTX-1  
Customer: Private Customer (C05976890)  
RegDate: **2015-12-01**  
Address: Private Residence  
City: Olivos  
StateProv: **AG**  
PostalCode: **1636** ← Ditto  
Country: **AR**



## Plus...

**\$ whois -h whois.arin.net NET-216-144-241-172-1**

NetRange: 216.144.241.172 - 216.144.241.175

CIDR: 216.144.241.172/30

NetName: LSN-DLLSTX-1

Customer: Private Customer (C06043883)

RegDate: **2016-02-16**

Address: Private Residence

City: Belo Horizonte

**StateProv: AG**

PostalCode: **30580270** ← Ditto

Country: **BR**

## And Also...

**\$ whois -h whois.arin.net NET-216-245-219-104-1**

NetRange: 216.245.219.104 - 216.245.219.111

CIDR: 216.245.219.104/29

NetName: LSN-DLLSTX-1

Customer: Private Customer (C06086184)

RegDate: **2016-04-06**

Address: Private Residence

City: Curitiba

**StateProv: AG**

PostalCode: **82130390** ← Ditto

Country: **BR**

## And Then There Is...

**\$ whois -h whois.arin.net NET-74-63-253-184-1**

NetRange: 74.63.253.184 - 74.63.253.187

CIDR: 74.63.253.184/30

NetName: LSN-DLLSTX-1

Customer: Private Customer (C06129634)

RegDate: **2016-05-18**

Address: Private Residence

City: Medellin

StateProv: **AG**

PostalCode: **05001000** ← Ditto

Country: **CO**

## Don't Forget...

**\$ whois -h whois.arin.net NET-192-169-81-136-1**

NetRange: 192.169.81.136 - 192.169.81.139

CIDR: 192.169.81.136/30

NetName: LSN-DLLSTX-1

Customer: Private Customer (C06179624)

RegDate: **2016-07-27**

Address: Private Residence

City: joinville

**StateProv: AG**

PostalCode: **89203100** ← Ditto

Country: **BR**

**etc., etc., etc.**

**This very quickly becomes boring/tedious.**

## Alternative Strategy

- Get a list of all the prefixes for **AS46475** from bgp.he.net
- Use your favorite editor to compose a command file that looks like:

```
dnsdb_query.py -l 1000000 --after=90d -i 103.3.166.0/24 > temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 135.84.217.0/24 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 162.253.40.0/22 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 192.169.80.0/20 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 192.169.80.0/22 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 192.169.92.0/22 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 208.115.192.0/18 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 216.144.240.0/20 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 216.245.192.0/19 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 2400:ed00::/48 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 2400:ed00::/64 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 2607:ff68::/32 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 63.143.32.0/19 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 64.31.0.0/18 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 69.162.64.0/18 >> temp3.txt
dnsdb_query.py -l 1000000 --after=90d -i 74.63.192.0/18 >> temp3.txt
```

## Alternative Strategy (continued)

- What do we end up with after we run those commands in bash?  
\$ **wc -l temp3.txt**  
1,449,381 temp3.txt ← lot of hits! (commas added for readability)
- Decide if you want to look at a particular TLD, such as .top...  
\$ **grep "top\. " temp3.txt > temp4.txt**  
\$ **wc -l temp4.txt**  
9156 temp4.txt ← 0.6% the size of the original file
- Now extract the associated IPs:  
\$ **awk '{print \$4}' < temp4.txt | sort -u > temp5.txt**  
\$ **wc -l temp5.txt**  
826 new-temp5.txt ← 0.06% the size of the original file
- Condense the discovered IPs into CIDRs:  
\$ **aggregate-cidr-addresses.pl < temp5.txt > temp5-cidr.txt**  
\$ **wc -l temp5-cidr.txt**  
469 temp5-cidr.txt ← 0.03% of the size of the original file
- For one cidr-aggregation tool, see:  
<https://gist.github.com/denji/17e30bddb9ce9e50294a>

## **IV. Conclusion**

## Take Aways

- Blocking unwanted traffic based on logged events is a fundamental skill, albeit one that many don't manage well.
- We've now talked about a variety of approaches you can try when you need to manage unwanted traffic, from the simple to the complex.
- You've seen how you can leverage passive DNS and BGP routing data to enhance what you can do with conventional tools such as Whois.
- This material will only make sense and become useful if you try using these techniques on your own networks and with your own data. Once you do try it, I think you'll be pleased with the power of these approaches.
- Finally, we're always interested in talking with about potentially running a passive DNS sensor. The data you contribute may help document relevant potential collateral damage, and help keep your site and its users safe from becoming collateral damage from overly broad blocking.
- **Are there any questions?**