

# **Modern Internet Scale Reconnaissance**

---

HD MOORE

BSIDES LAS VEGAS 2017

# Howdy!

Work as a penetration tester / vulnerability researcher / hacker at large

Lots of time writing public exploits, blogs, whitepapers, etc

A few years of internet scanning projects

# Introduction

A practical guide to building your own reconnaissance platform

- Gather raw data for the internet as a whole
- Query it locally, fast, and cheap
- Make security work better!

# Shiny New Recon Tools!

New tools released since the BSidesLV submission

- XRay - <https://github.com/evilsocket/xray>
- Aquatone - <https://github.com/michenriksen/aquatone>
- Web Sight - <https://github.com/lavalamp-/ws-docker-community>

# Problem Space

Most companies don't actually know their external footprint

Penetration testing scope is rarely accurate or fully known

This complicates M&A, IT management, security testing, etc

Existing solutions

- DNSDB (Robtex)
- PassiveTotal
- Farsight Passive DNS
- OpenDNS Umbrella
- Open source OSINT tools
- Manual OSINT lookups

# Current Challenges

Discovery is becoming dependent on third-party APIs and services

Coverage drastically changes by source and technique

Difficult to identify frequently changing infrastructure

Cloud discovery is difficult without credentials

DevOps deployment tools complicate things

# Moving to Local Data

Pull data from as many sources as possible and cross-reference

Use local datasets instead of querying third parties

Avoid leaking target information to third parties

Complement existing active discovery efforts

Dig wider and deeper as needed

Find weird & interesting stuff!

# Build a Platform

We want domains, whois information, DNS data, TLS cert data, anything!

Collect data on regular intervals, search data for stuff we care about

We don't want to spend a lot of money getting it

Storage is relatively cheap, computers are fast

Lets go data shopping!



<a href="#">Sonar</a>	FDNS, RDNS, UDP, TCP, TLS, HTTP, HTTPS scan data	<b>FREE</b>
<a href="#">Censys.io</a>	TCP, TLS, HTTP, HTTPS scan data	<b>FREE</b>
<a href="#">CT</a>	TLS Certificates	<b>FREE</b>
<a href="#">CZDS</a>	DNS zone files for new global TLDs	<b>FREE</b>
<a href="#">ARIN</a>	American IP registry information	<b>FREE</b>
<a href="#">CAIDA PFX2AS IPv4</a>	Daily snapshots of ASN to IPv4 mappings	<b>FREE</b>
<a href="#">CAIDA PFX2AS IPv6</a>	Daily snapshots of ASN to IPv6 mappings	<b>FREE</b>
<a href="#">US Gov</a>	US government domain names	<b>FREE</b>
<a href="#">UK Gov</a>	UK government domain names	<b>FREE</b>
<a href="#">RIR Delegations</a>	Regional IP allocations	<b>FREE</b>
<a href="#">PremiumDrops</a>	DNS zone files for com/net/info/org/biz/xxx/sk/us	<b>\$24.95/mo</b>
<a href="#">WWWS.io</a>	Domains across many TLDs (~198m)	<b>\$9.00/mo</b>
<a href="#">WhoisXMLAPI.com</a>	New domain whois data	<b>\$109.00/mo</b>

**<https://github.com/hdm/inetdata>**

# Downloading Data

Grab the *inetdata* repository:

- <https://github.com/hdm/inetdata>
- git clone, cp config/inetdata.json.sample config/inetdata.json
- Sign up for APIs, enter credentials and keys as needed

Setup *inetdata-parsers*

- Full steps at <https://github.com/hdm/inetdata-parsers>
- Parallel processing friendly (written in Golang)

Run *inetdata* downloader and normalizer

- *inetdata/bin/download.sh* && *inetdata/bin/normalize.sh*
- Wrapped up in the *inetdata/bin/daily.sh* script
- More RAM & more cores helps!

# Crunching Data

Raw data is nice, but cooked data is much more useful

Structure the data to match the query use cases

Make lookups fast

- By IP CIDR
- By domain prefix

Useful cooked outputs

- Sonar DNS (FDNS, RDNS)
- CZDS (ICANN gTLDs)
- PremiumDrops (Legacy TLDs)
- Certificate Transparency logs
- Censys.IO IPv4

# Server Specifications

As much RAM and as many cores as possible (4c/16Gb+)

- Google Cloud: **n1-highmem-4** (4 vCPUs, 26 GB memory) [\$122/mo]

Lots of storage space (1Tb+ HDD) for long-term archives

Fast working directory (SSD/NVMe/etc) for scratch space

Ubuntu Linux 16.04 LTS is the easy-mode option for tools

Two weeks to bootstrap \*everything\*

A few hours the first day otherwise

# CPU > IOPS == pigz

CPU cores are substantially cheaper than higher IOPS

Reduce required IOPS by compressing data inline

Across every pipe, temp directory, artifact file

Use parallel versions of compression tools

- pigz (gzip)
- pbzip2
- lz4

Stick with gzip format for compatibility

- Support for Hadoop processing
- Support within Java parsers

# MTBL Databases

Sorted String Table (key-value) database by Farsight Security

- <https://github.com/farsightsec/mtbl>
- Build key names for each use case
- Built-in compression!

*inetdata-parsers* includes the *mq* mtbl query utility

- -domain something.com
- -cidr 8.8.8.0/24
- -j for output and pipes
- -v / -k for just keys/values
- Swiss army knife for searching cooked inetdata output

CPU intensive to build (*inetdata-parsers*), but insanely fast to query

Search 1Tb of MTBLs with ~8Gb of memory instantly\*

# Convert Censys.io to MTBL

Sign up to obtain credentials, add them to `./config/inetdata.json`

Clear about 3Tb of space for raw + processed data

Install *liblz4-tool* for *inetdata* to unpack raw files

Download the latest IPv4 dataset with

- `$ inetdata/bin/download.sh -s censys_ipv4`

Convert to MTBL with

- `$ inetdata/bin/normalize.sh -s censys_ipv4`

Query with

- `$ mq -v -n -cidr 8.8.8.0/24 censys_ipv4/normalized/ipv4-[date].mtbl`

```
{"ip":"8.8.8.8","ipint":134744072,"p53":{"dns":{"lookup":{"additional": [], "answers": [{"name": "c.afekv.com", "response": "192.150.186.1", "type": "A"}, {"name": "c.afekv.com", "response": "173.194.103.8", "type": "A"}], "authorities": [], "errors": false, "metadata": {}, "open_resolver": true, "questions": [{"name": "c.afekv.com", "type": "A"}], "resolves_correctly": true, "support": true, "timestamp": "2016-11-22 00:13:21"}}, "location": {"city": "Mountain View", "continent": "North America", "country": "United States", "country_code": "US", "latitude": 37.38600000000003, "longitude": -122.0838, "postal_code": "94035", "province": "California", "registered_country": "United States", "registered_country_code": "US", "timezone": "America/Los_Angeles"}, "autonomous_system": {"asn": 15169, "country_code": "", "description": "GOOGLE - Google Inc., US", "name": "GOOGLE", "organization": "Google Inc., US", "path": [15169], "routed_prefix": "8.8.8.0/24"}}
```

# JSON Line Format

JSON is a bulky format, but still better than XML

Line-delimited JSON records make life easy

- jq
- jsawk
- dap

ARIN to JSONL conversion makes easy greps

- `$ egrep -i "'email':".*@microsoft\.com"' pocs.json | jq .city | head`
  - "New York"
  - "Redmond"
  - "Dallas"
  - "BOULDER"
  - "ASHBURN"
  - "Redmond"



# Text Files Forever

Everything not in MTBL or JSON is CSV or plain text files

Make it easy to pipe data through other tools

Unix model for data management

# Storage Usage by Source

ARIN (XML + JSONL): **8Gb/day**

Sonar FDNS/RDNS (Raw + CSV + MTBL): **200Gb/week**

ICANN CZDS (Raw + MTBL): **1.5Gb/day**

PremiumDrops (Raw + MTBL): **4.3Gb/day**

WWWS.IO (Raw + MTBL): **6.5Gb/day**

Censys IPv4 (Raw + MTBL): **3Tb/snapshot (huge!)**

Pick and choose data sources with *inetdata/bin/download.sh -s <src>*

Two years of selective daily datasets is approximately 30Tb

# Platform Capabilities

Regular drops of new data via *inetdata + inetdata-parsers*

Fast lookup by domain name or IP range

Common use cases with existing dataset

- Find all hostnames for a given domain name (subdomains)
- Find all IP ranges for a given domain name
- Find all SSL/TLS sites for a given domain name
- Find all domains for a given nameserver
- Find all usable domain fronting hostnames
- Find typo and keyword matching domains
- Find all domains with the same registrant
- Historical ownership of domains & IPs

# Next Steps

After bootstrapping, add **inetdata/bin/daily.sh** to cron

Add custom scripts to monitor, match, and notify

Will dive into specific scenarios during demos

Query the datasets to win at security!

# Certificate Transparency

A quick diversion into Certificate Transparency

- CT is a Google-run project to track TLS certificates globally
- CT logs are append-only historical logs of x509 certificates
- CT logs are append-only and publicly readable
- CT submissions are mandatory for Chrome support of a CA

Home: <https://www.certificate-transparency.org/>

Search: <https://crt.sh/>

# Certificate Transparency Logs

Anyone can operate a log, public logs are documented online

- <https://www.certificate-transparency.org/known-logs>

## Example logs

- pilot: <https://ct.googleapis.com/pilot>
- aviator: <https://ct.googleapis.com/aviator>
- rocketeer: <https://ct.googleapis.com/rocketeer>
- submariner: <https://ct.googleapis.com/submariner>

## Log servers expose API endpoints (json)

- `/ct/v1/get-sth` (return the head of the log)
- `/ct/v1/get-sth-consistency` (return sth consistency)
- `/ct/v1/get-entries` (return encoded CT records)
- `/ct/v1/add-pre-chain` (submit cert pre chain)
- `/ct/v1/add-chain` (submit cert chain)
- `/ct/v1/add-json` (submit cert chain)

# Extended Validation in Chrome

EV certs must be logged to Certificate Transparency for Chrome support

Identify new EV-certificate sites as they are being deployed

- Staging sites, pre-production, development environments
- Certs with CNs for internal resources

## Extended Validation in Chrome

In order to improve the security of Extended Validation (EV) certificates, Google Chrome requires Certificate Transparency (CT) compliance for all EV certificates issued after 1 Jan 2015.

A plan to bring CT to all certificates in Chrome has been [published on the ct-policy Chromium group](#).

Full details, as well as up-to-date CT compliance specification, are available on the Chromium Certificate Transparency page:

<https://www.chromium.org/Home/chromium-security/certificate-transparency>

# Lets Encrypt + CT

LetsEncrypt sends all new certificates to the Pilot CT server

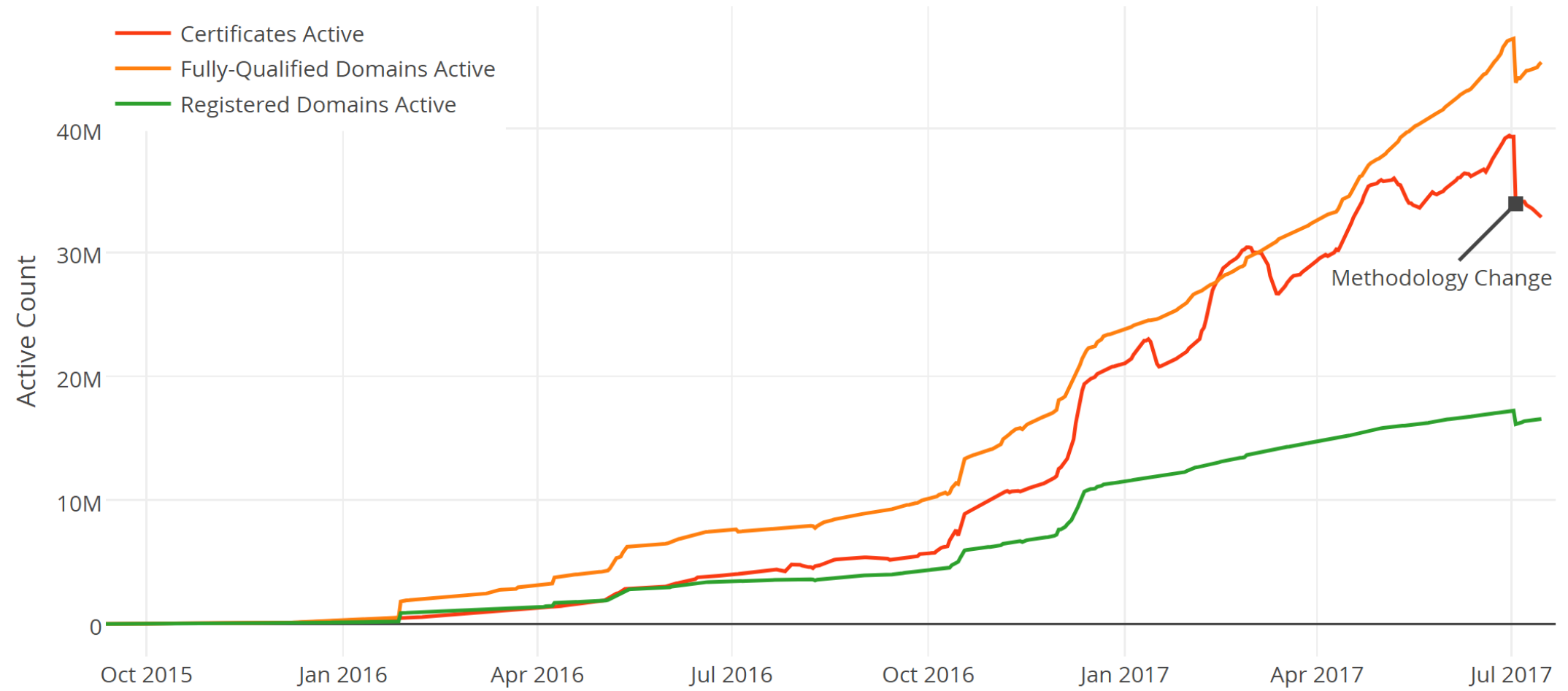
- LetsEncrypt market share continues to increase
- LetsEncrypt integrations are everywhere
- This happens almost in real-time

Services that use LetsEncrypt are being advertised in CT

- Dynamic infrastructure becomes discoverable
- New assets become visible immediately

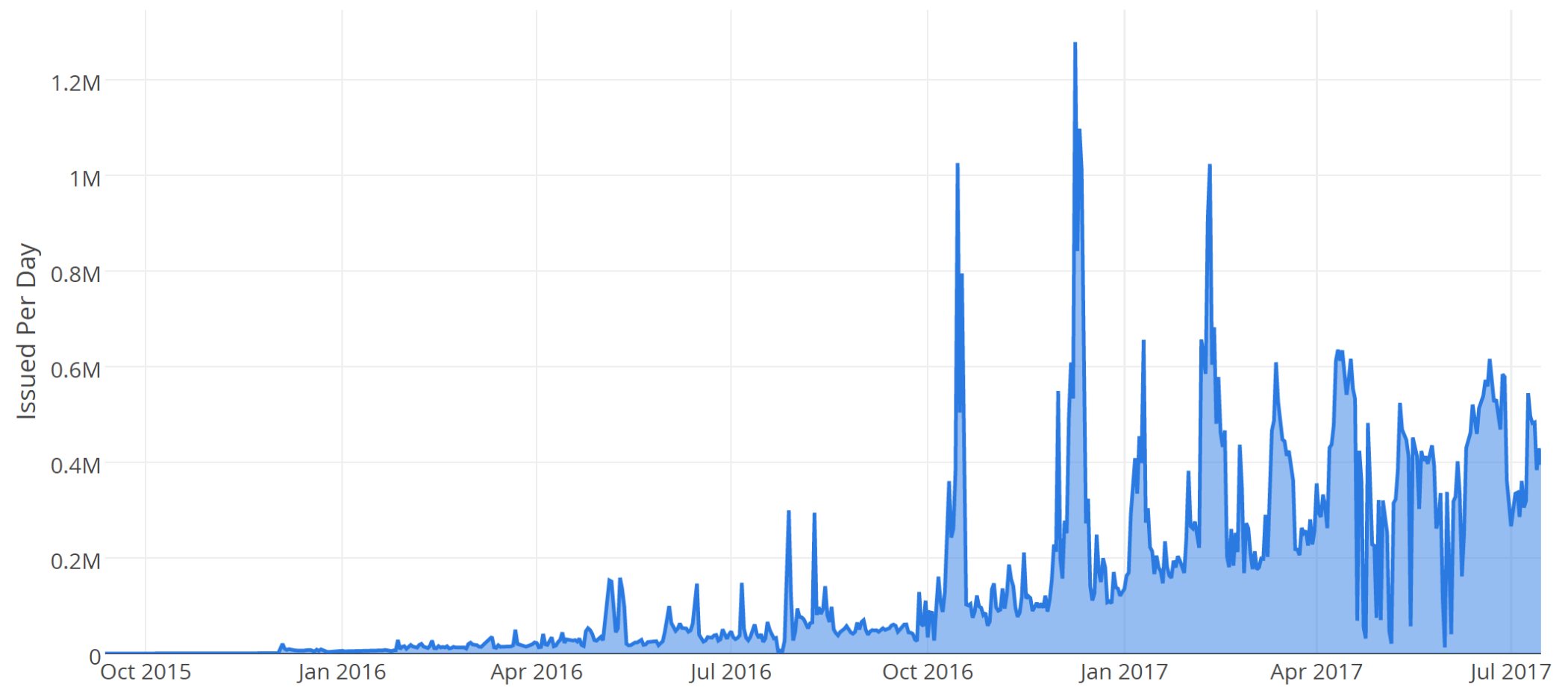


# Let's Encrypt Growth



Source: [Firefox Telemetry](#)

## Let's Encrypt Certificates Issued Per Day



Source: [Firefox Telemetry](#)

# Real-time CT Monitoring

*inetdata-ct-tail* provides a firehose of TLS certificate names

- Install golang 1.8+ from <https://golang.org/dl/>
- `$ sudo apt-get install libmtbl-dev`
- `$ go get github.com/hdm/inetdata-parsers/cmd/inetdata-ct-tail`
- `$ inetdata-ct-tail -f | grep vpn`

Add a bloom filter to the pipeline to deduplicate\*

Feed the output into automated scanning tools

Identify dynamic assets as they are provisioned

This has a fun security implication...

# Racing to First Setup

Many apps provide admin access to the first person to visit the site

We can beat the legitimate user by tailing CT into nmap

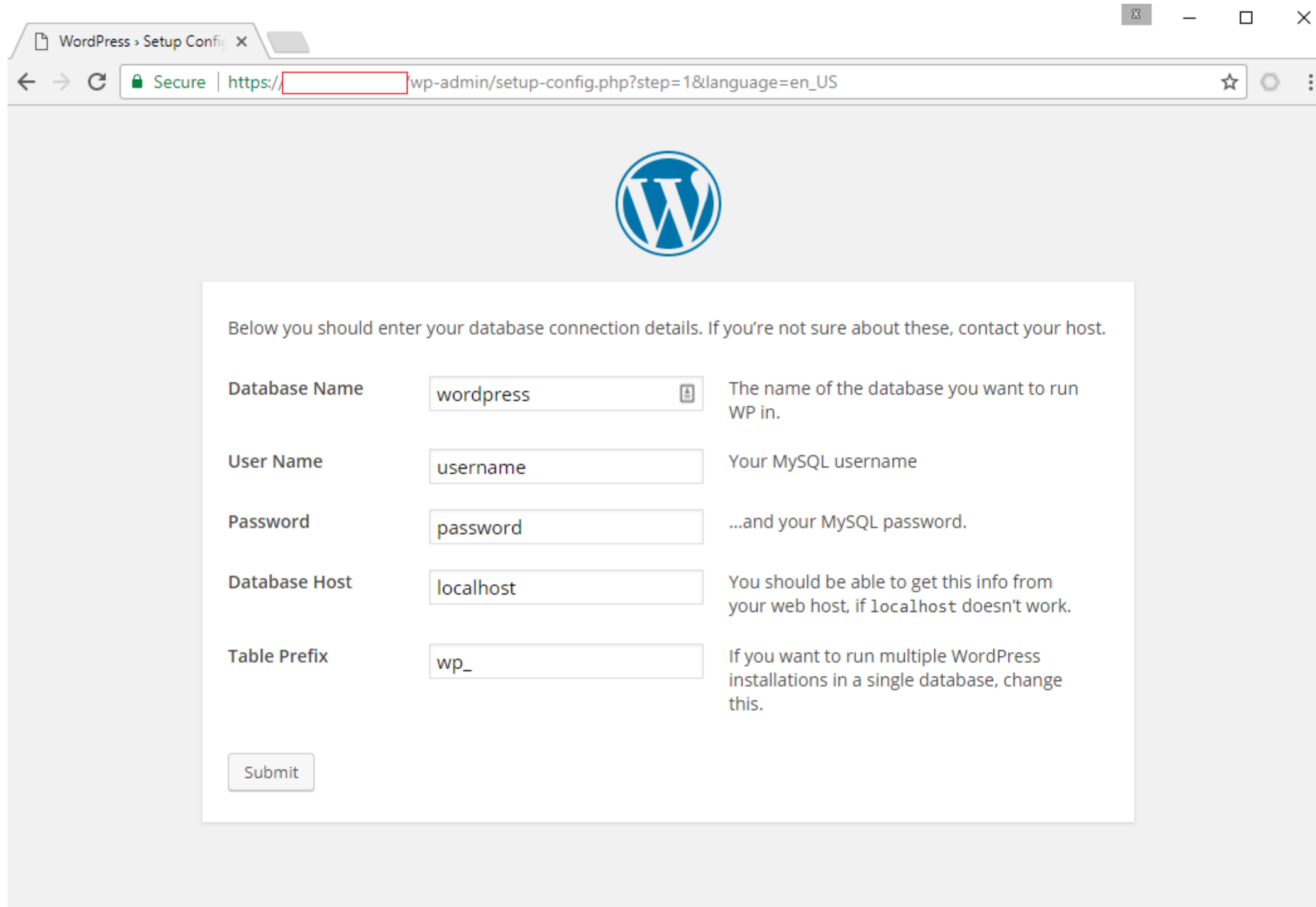
...then backdoor the server and reset the setup =)

```
$ inetdata-ct-tail -f 2>/dev/null | perl -pe 's/,dns,/\n/g' | cut -f 1 -d , |  
bloom | grep -v ^\*. | nmap -iL - --min-rate=1000 -PS443 -p 443 --max-  
retries=1 --script=http-title --min-parallelism=64 -oA ct-tail
```

...

```
|_http-title: Did not follow redirect to https://[nooooo]/wp-  
admin/setup-config.php
```

# Winning a WordPress



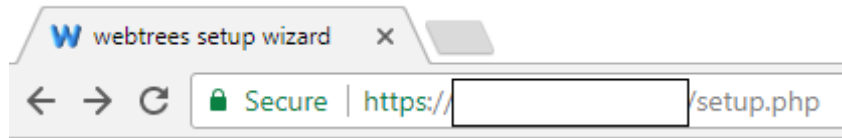
The image shows a browser window with the title "WordPress > Setup Config". The address bar shows a secure connection to a URL ending in "wp-admin/setup-config.php?step=1&language=en\_US". The page features the WordPress logo at the top center. Below the logo, a white box contains the following text and form fields:

Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="wordpress"/>	The name of the database you want to run WP in.
User Name	<input type="text" value="username"/>	Your MySQL username
Password	<input type="text" value="password"/>	...and your MySQL password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost doesn't work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

Submit

# Lots More!



## Setup wizard for webtrees

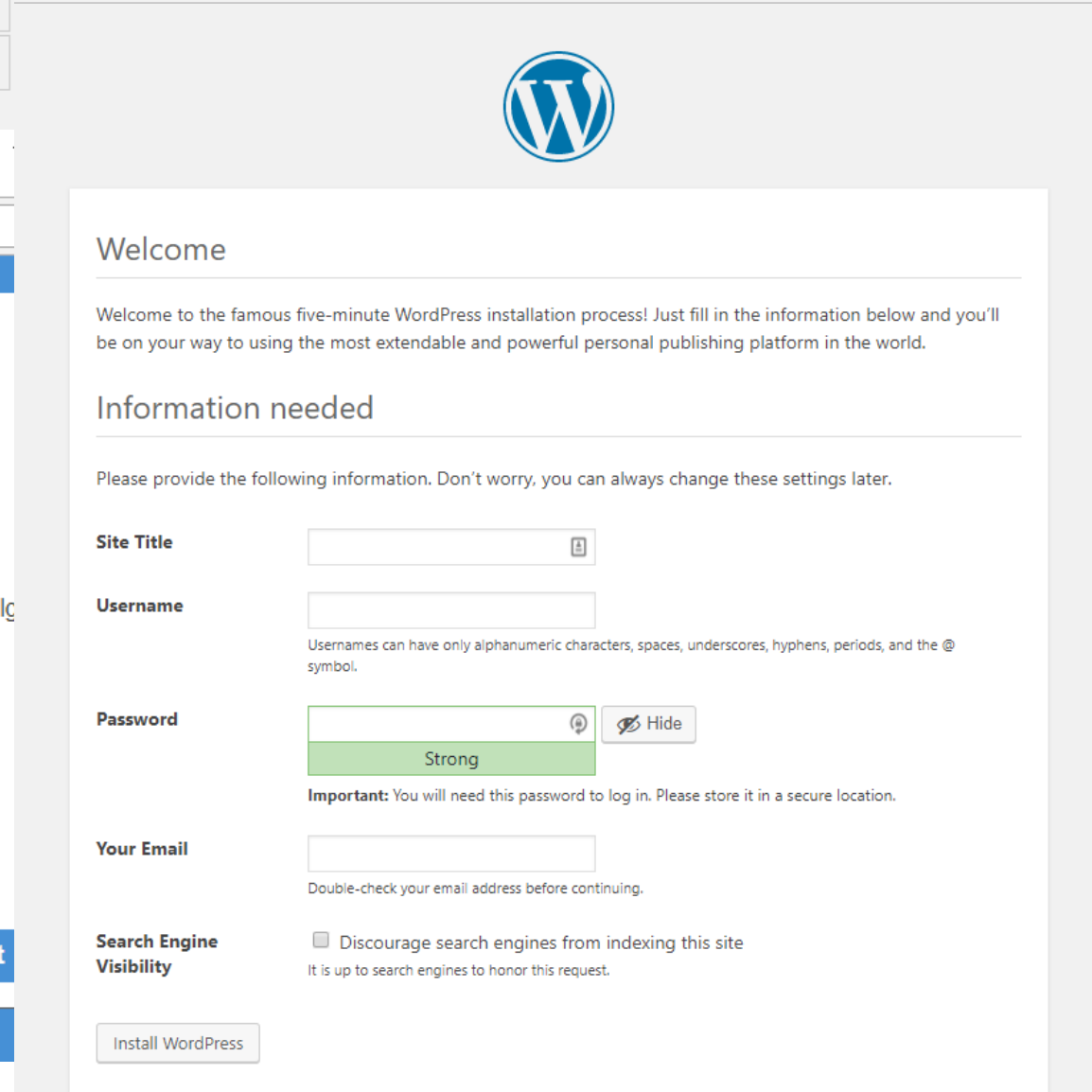
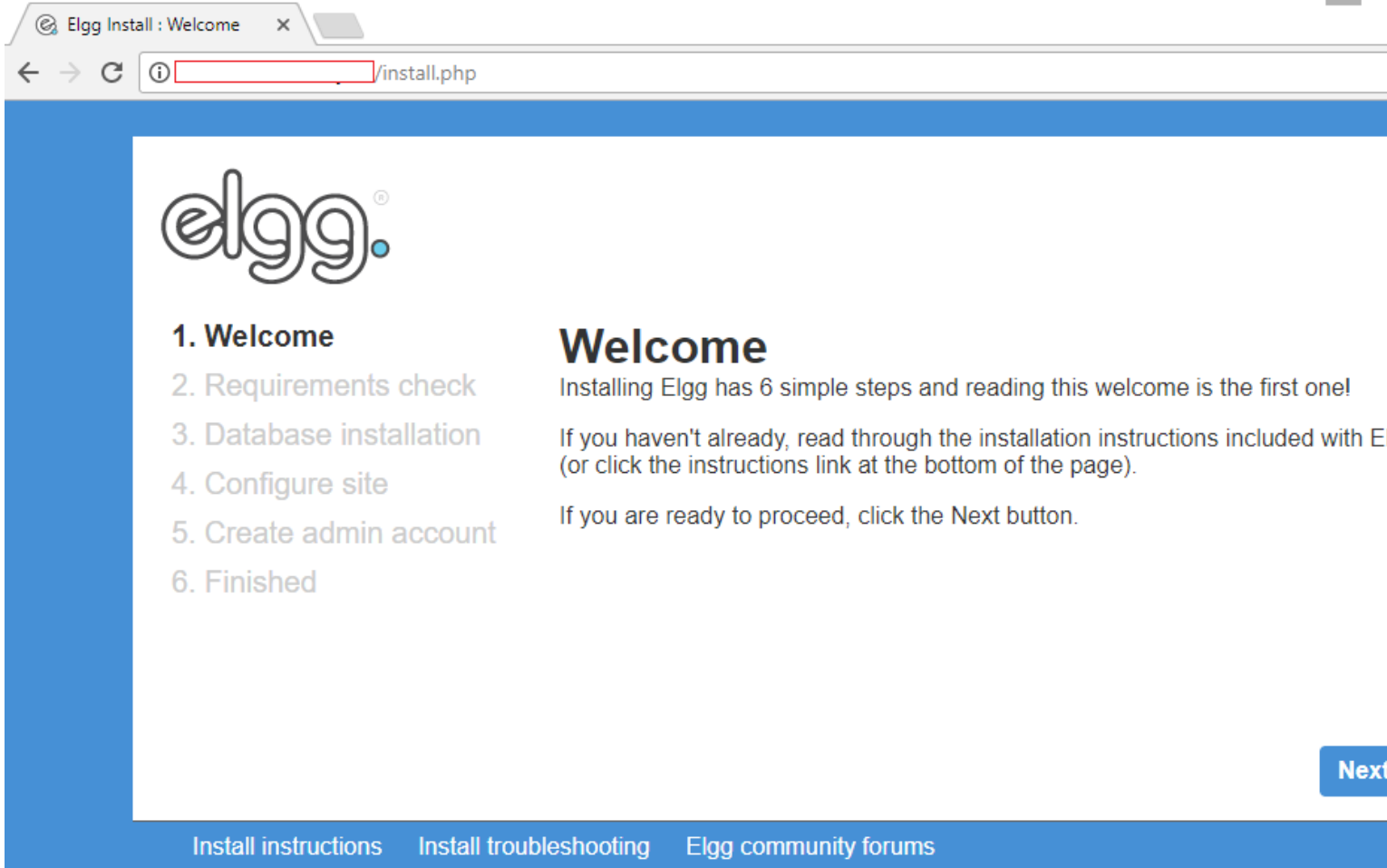
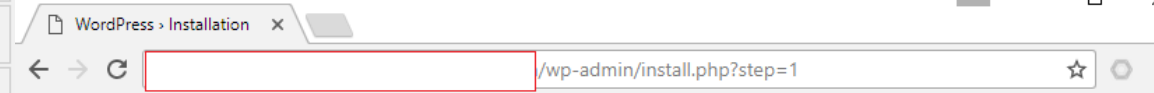
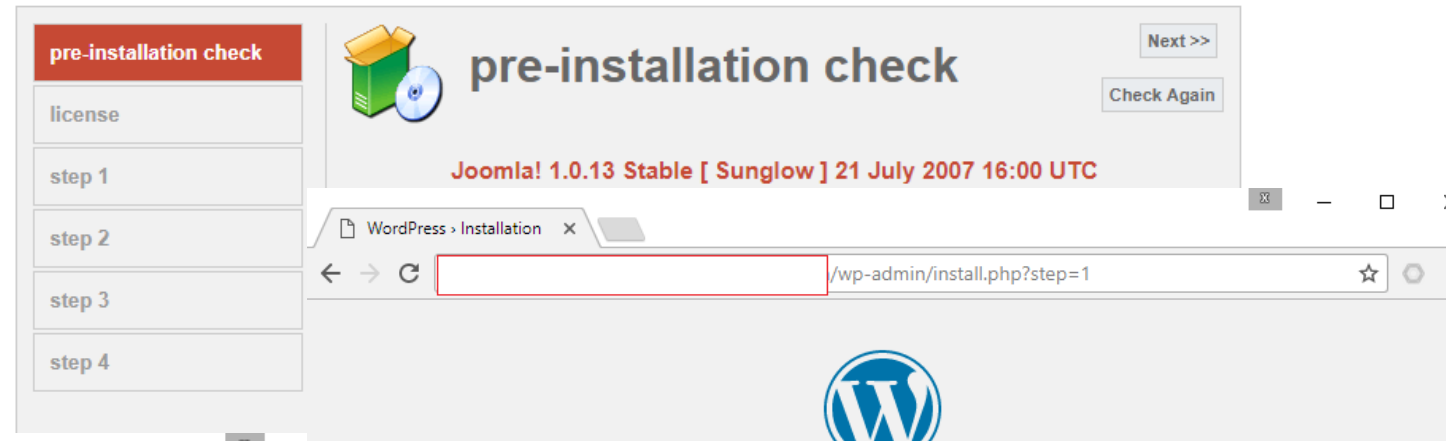
### Connection to database server

webtrees needs a MySQL database, version 5.0.13 or later.

Your server's administrator will provide you with the connection c

#### Database connection

Server name  Most sites a your web se



# Cloud Discovery: Azure DNS

Identify CNAMEs that point into cloudapp.azure.net

- `$ mq -n -domain cloudapp.azure.com sonar/201707*.mtbl | wc -l`  
> 8529
- `$ mq -n -domain cloudapp.azure.com ct/*.mtbl | wc -l`  
> 2865
- Misconfigured DNS can leak the \*.internal.cloudapp.net hostnames
- Easy attribution from Azure assets back to a known organization

# Domain Fronting

Leverage millions of cloud-hosted domains for your C2

Discover frontable domains by querying Sonar MTBLs

- Sonar FDNS generates forward/inverse lookups
- Leverage inverse lookups for reverse CNAME
- Dump all hostnames within a domain



# Domain Fronting: Azure

```
$ mq -k -n -domain azureedge.net 201707*.mtbl | wc -l
```

```
$ mq -v -n -domain azureedge.net sonar/normalized/201707*.mtbl | jq .  
-r |grep -A 1 r-cname | grep \" | grep -v r-cname | shuf | head
```

- software-download.microsoft.com
- static.cdn.salewa.com
- www.shama.com
- amici.iccf.com
- www2.pepsico.com
- www.mosaic-collection.com
- www.duddingston-golf-club.com
- www.eyerecommend.ca
- vidzapper.vidzapper.com

# Domain Fronting: Cloudfront

```
$ mq -k -n -domain cloudfront.net 201707*.mtbl | wc -l
```

```
$ mq -v -n -domain cloudfront.net sonar/normalized/201707*.mtbl | jq .  
-r |grep -A 1 r-cname | grep \" | grep -v r-cname | shuf | head
```

- static.demobi.us
- wac-cdn.atlassian.com
- cdn.stage2.consumerreports.org
- www.awtaxi.com.au
- dev.makeitsocial.com
- static.101hacks.com
- www.yourfoodjob.com
- eu1static.oktacdn.com

# Domain Fronting: Fastly

```
$ mq -k -n -domain fastly.net 201707*.mtbl | wc -l
```

```
$ mq -v -n -domain fastly.net sonar/normalized/201707*.mtbl | jq . -r |  
grep -A 1 r-cname | grep \" | grep -v r-cname | shuf | head
```

- shop.tinypencil.com
- mjdele.github.io
- sol-roar-cdn.rebelmouse.com
- revan.yelp.com
- b2g.bigcartel.com
- helmutzechmann.com
- eightmedia.github.com
- cdn3.skybride.com

# Discovery: M&A

**Blackstone acquires Clarion Events for £600m, lets find their hosts:**

Sonar FDNS: **17 hostnames**

- `$ mq -k -domain clarionevents.com ./data/sonar/*fdns*.mtbl | sort -u`

Sonar RDNS: **4 hostnames**

- `$ mq -k -domain clarionevents.com ./data/sonar/*rdns*.mtbl | sort -u`

Certificate Transparency: **10 hostnames**

- `$ mq -k -domain clarionevents.com ./data/ct/*.mtbl | sort -u`

Combined: **20 hostnames**

# Discovery: Full Asset List

**All hostnames / IP addresses for a company (McAfee)**

Sonar DNS: **2216 hostnames**

```
$ mq -k -domain mcafee.com sonar-dns/201705*.mtbl | sort -u
```

Certificate Transparency: **537 hostnames**

```
$ mq -k -domain mcafee.com ct/*.mtbl | sort -u
```

Combined: **2546 hostnames**

# Summary

Local internet datasets can improve your security game

Discovery, monitoring, exploitation, exfiltration

Costs are relatively low compared to value

Keep your client/company identify safe

# Roadmap

More data sources, more normalizers, more configuration options

Support for real-time streaming sources (pdns, inetdata-ct-tail)

Performance improvements for low-end servers

Example analysis scripts for common tasks

Split early CT dataset into smaller blocks

Automatic data expiration & deletion

Build out post-normalize hooks

MTBL API daemon + clients

# Contribute!

Fork, fix, expand! Add new datasources! Add new utilities!

- <https://github.com/hdm/inetdata>
- <https://github.com/hdm/inetdata-parsers>

Build your own API or service, internal or external

Monitor your company's footprint for changes

Dig up fun research for your next talk!



Demo Time!

# Q & A

Contact: [underflow@hdm.io](mailto:underflow@hdm.io)