EFFICIENT BUG BOUNTY AUTOMATION TECHNIQUES

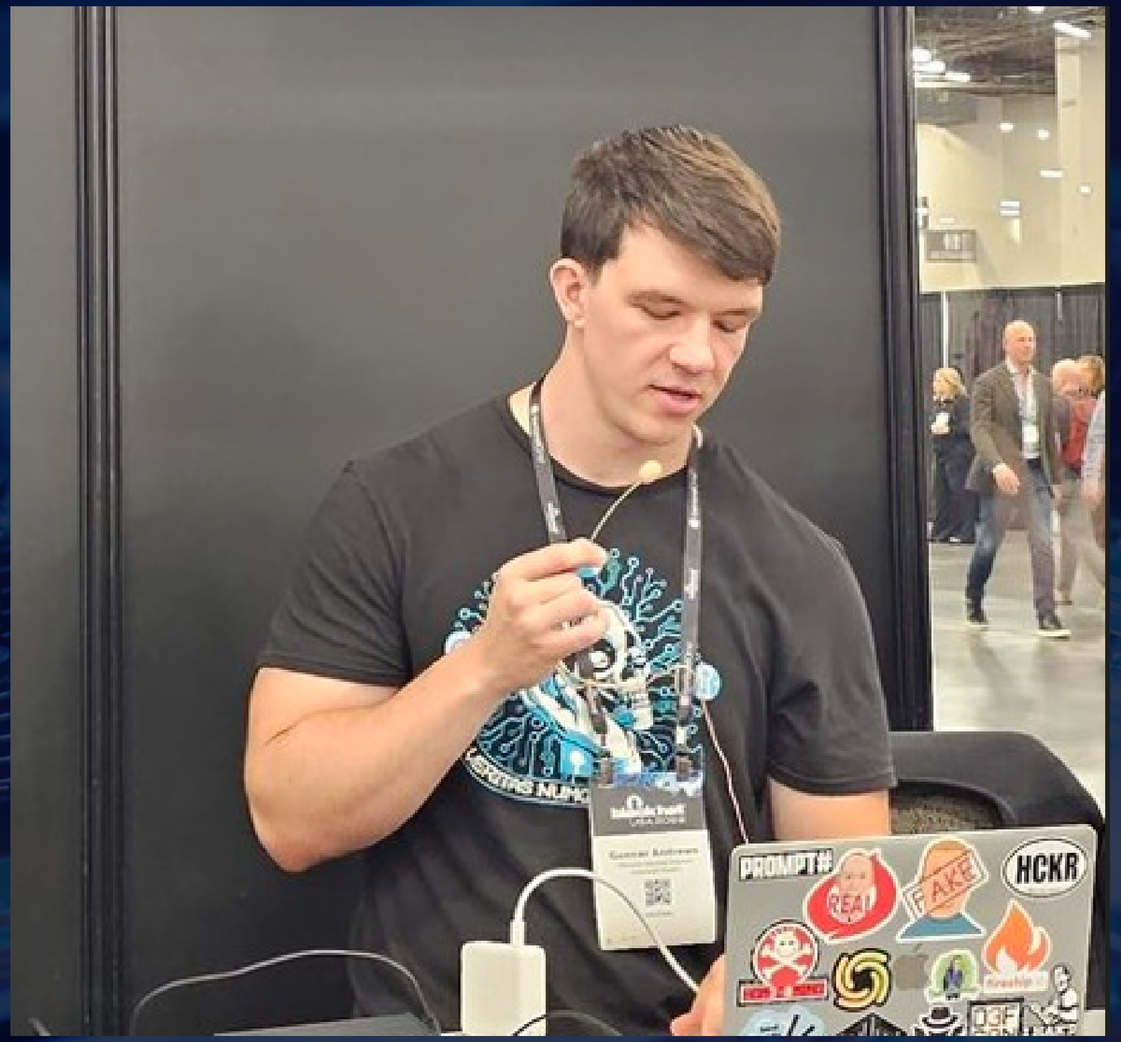
Gunnar Andrews

Welcome!

App Sec Engineer

Bug Hunter

Content Creator

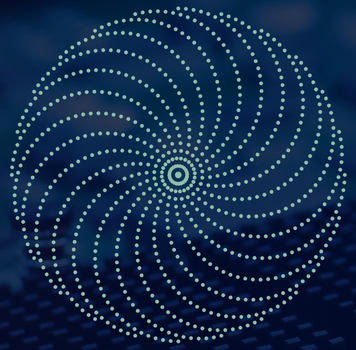


Goals



Resources vs. Findings

Getting from 0 => 100 with the least cost possible.



Automating Collection

What I collect and how.



Data Aggregation

Data hoarding or bug bounties?...



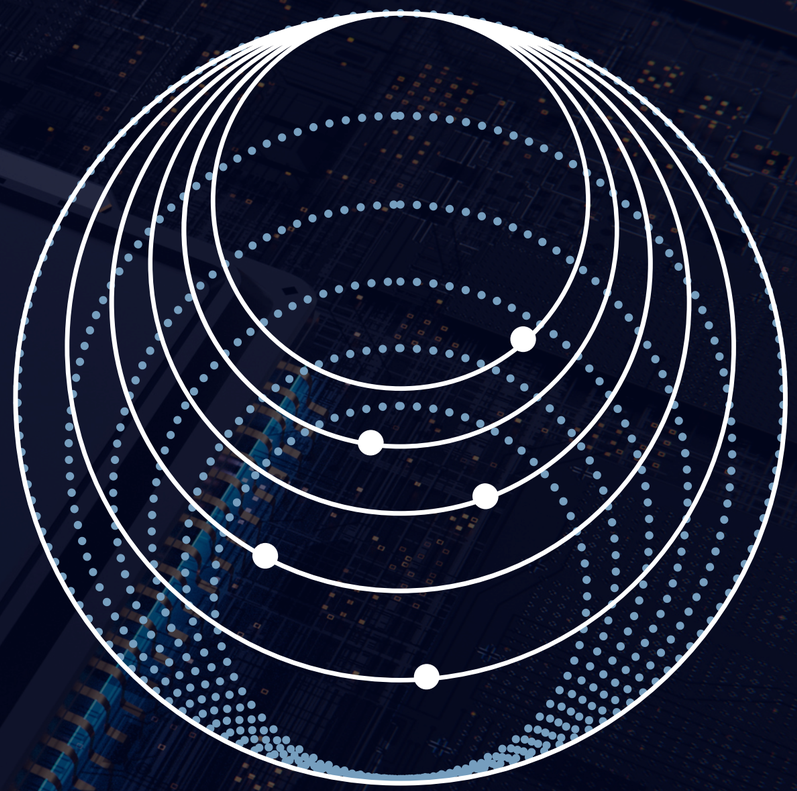
I don't brute force until I have to

Brute forcing is guessing.

We will cover what to do before you start guessing.

Apex Domains

Some programs have “Any asset owned by <company>” in scope!



The journey begins

- ASN sources
  - bgp.he.net
  - dnschecker.org
- Not a complete picture if they are cloud friendly...

- Reverse WHOIS via whoxy.com
  - Org name
  - Registrant name
- Paid API

Not all of this can/should be automated

Scan the planet!

- Scan IPv4 for certificates
- There are a few tools for this, but I built my own
  - Caduceus
- Ports to start with
  - 443, HTTPS
    - Other common HTTPS ports
  - 990, FTPS
  - 563, NNTPS
  - 636, LDAPS
  - 993, IMAPS
  - 995, POP3S

- Reverse DNS Lookup IPv4 addresses
- Lots of tool options:
  - DNSx
  - Hakrevdns
  - ZDNS
  - etc.

- These scans give new apex domains!
  - grep or AI for parsing
- **TIP**: Scan the cloud from the cloud!

At this point it’s basically like Pokémon
for bug bounty hunters...

Subdomains



Passive DNS

- Passive DNS sources have become incredibly good
- Some are pay to use, in my opinion it is worth it
- Always use the APIs in the way they are intended
  - READ THE DOCS
  - Rate limiting
  - data pagination
  - credit usage
  - etc.

- There are tools for this
  - Subfinder
  - Assetfinder
  - Amass
  - BBOT
  - Finddomain
  - and more
- I wrote my own logic to query the specific APIs I want to use
  - why?

Audit your tools!

- Read the source code, and make sure the tool is doing what you expect
- I noticed a lot of tools were not interacting with the APIs the way that I think they should/could
- So I write my own so I know EXACTLY how my data is being collected
- Start with existing tools, and build your own to fill the gaps or create something more suited to you

Audit your tools!

```
[INF] Found 1267 subdomains for tesla.com in 43 seconds 525 milliseconds
[INF] Printing source statistics for tesla.com

 Source                  Duration      Results     Errors
-------------------------------------------------------
 alienvault              2.969s             13          0
 anubis                  213ms             320          0
 builtwith               0s                  0          0
 certspotter             3.994s              0          0
 chaos                   1.44s              46          0
 commoncrawl             43.505s         -70782          0
 crtsh                   8.54s               0          0
 digitorus               450ms               1          0
 dnsdb                   0s                  0          0
 dnsdumpster             1.108s              0          0
 hackertarget            327ms               0          0
 leakix                  675ms               0          0
 netlas                  707ms               0          1
 rapiddns                1.883s              7          0
 securitytrails          10.175s           104          1
 shodan                  195ms             776          0
 sitedossier             231ms              -2          0
 virustotal              0s                  0          0
 waybackarchive          30.023s             0          0


 The following sources were included but skipped...
```

Certificates... Again!

- IPv4 certificate scanning from Caduceus works for subdomains too
- Certificate transparency logs
- Crt.sh works usually but not always
- Takes to a day or two for some data to make its way into crt.sh
- Made my own tool that constantly reads the top of the CT log trees
    - Gungnir
- Certificates have multiple domains (virtual hosts, internal domains, etc.)
- We will talk more about virtual hosts later...
- These go into my automation to be monitored over time

Other Good Stuff

- Finding subdomains on GitHub
  - github-subdomains (by gwen001)
- IPv4 RDNS data
- Light spidering on known assets for new domains
- Virtual host scanning with Harpe

The simplest, yet MOST ANNOYING, process ever.

DNS Resolution



Issues Resolving

- A lot of current tooling will silently fail, return servfails, etc.
- If you only attempt to resolve once, you ARE missing assets.
- Trickest has a list of trusted resolvers. It is what I started with. But I moved on later
- Public resolvers have rate limits and timeouts. This is normally handled by tooling by just moving on.

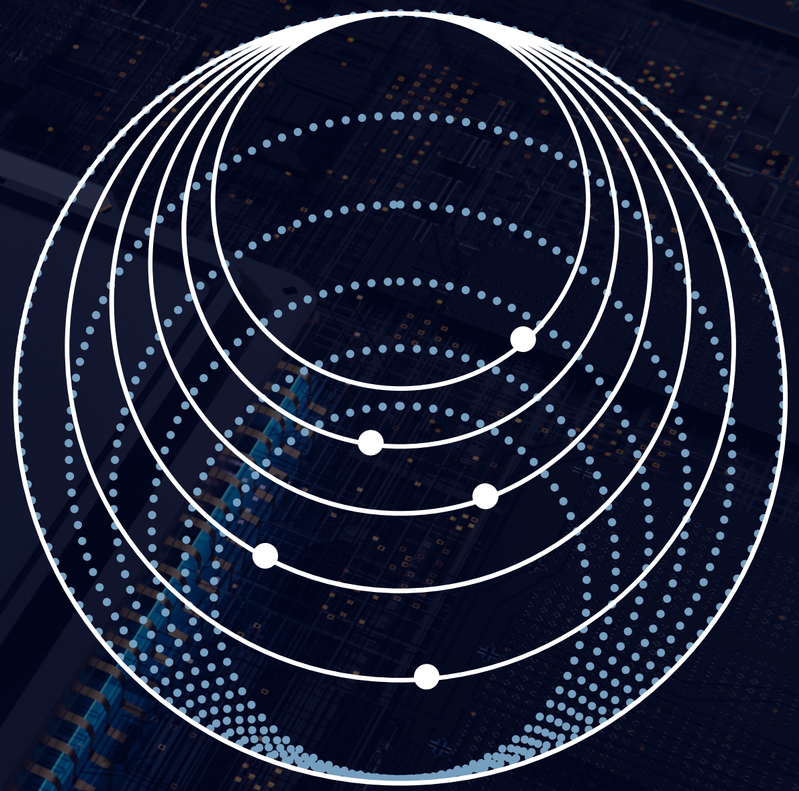
My DNS Experience

- I host my own DNS resolution stack
  - VIP using keepalived
  - PowerDNS dnsdist
  - PowerDNS recursor
- I attempt to resolve a domain multiple times
  - One sorted set of currently resolving domains with timestamps
  - One sorted set of ALL domains with a count of attempts

- Records I currently keep:
  - A
  - AAAA
  - CNAME
  - MX
  - NS
  - TXT
- The records you pull/store is up to you. More later

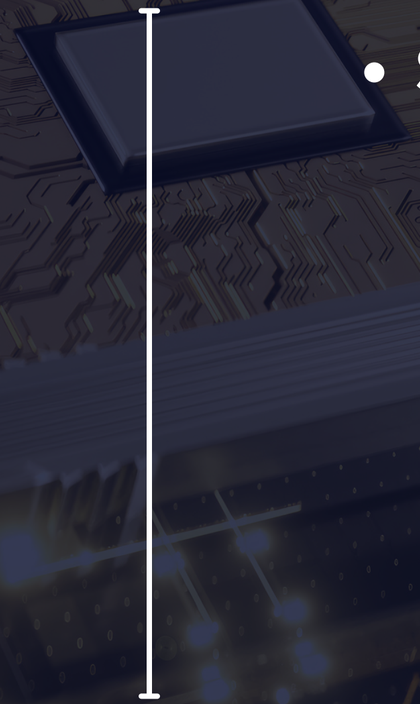
Port Scanning





Active vs. Passive

- My go to active tools are:
  - Naabu
  - Masscan
- Passive port discovery:
  - Smap (shodan port enumeration)
- What data (ports) you collect is up to you



- Some ideas:
  - 9200 (elasticsearch)
  - 2375 (docker)
  - 10250 (kubernetes API)
  - 8069 (odoo manager)
  - 6379 (redis)
  - etc.

Webservers & Endpoints



Scanning the web

- Use outputs from DNS resolution or port scans to trim input
- Don’t get blocked or banned!
  - Slow scanning
  - User agent
  - Proxy services
  - Lambdas (shadowclone)
- One GET request can get you a LOT of data
  - Title, status code, redirects, headers (content-length, CSP, cookies, etc.), and more
- Remember every request has a chance to incur WRATH!

Endpoints

- I use passive sources first here again
  - GAU / waymore
- Spidering/crawling is hard to do at scale
  - Blocks/bans/blacklisting
- I only kick off spidering on sites that look interesting
  - What looks interesting?
- Headless seems to work “the best”
- This is the next thing I personally am going to deep dive and optimize :)


WE SHOULD TAKE THIS DATA
AND MOVE IT SOMEWHERE ELSE
makeameme.org

Data

Subdomains vs. IPs

- Subdomains and IPs are both extremely useful!
- Subdomains
  - Virtual hosts (especially with caduceus data)
  - proxies/gateways work with subdomains
- IPs
  - Matching multiple domains to an IP
  - Makes port scanning simpler vs port scanning domains
  - Mapping a targets real internet presence

Data Engineering

- Our main goal is data => bugs
- Collecting data is not what our end goal is
  - Collect data you know you want to use
- Data engineering and data tooling is HEAVILY underused in bug bounty
  - There is more than flat files or postgres :)
- Start with DBs, caching, and data visibility/usability

- Two good reads
  - Designing Data-Intensive Applications
  - Fundamentals of Data Engineering
- Topics to look into:
  - ETL, data streaming, batch processing, orchestration, event-driven architecture, etc.

Automation's Advantage

- CDC
  - Change data capture
- Steps:
  - 1. Change Detection
  - 2. Change Capture
  - 3. Change Delivery
  - 4. Change Application
- This is automation’s BIGGEST advantage!
- Running a script once a month gives less benefits than running it every hour for THIS reason.
- Snapshot vs. continuous recon

WHAT I DO

- Apex domains
  - Follow steps and aggregate manually to reduce false positives
- Subdomains
  - Use my favorite sources with my own version of “subfinder”
  - Github subdomains
  - Caduceus & RDNS IPv4 scan twice a month, cloud IPs multiple times a day, Known target IPs multiple times a day
  - Gungnir constantly running
- Resolving
  - My own DNS resolution stack
  - Two lists (attempts set, timestamped set of resolving domains)
  - Use data to trigger vuln checks

- Port scanning
  - Smap every IP I get from DNS resolution
  - Every IP I see gets scanned with Naabu ~Once a month
- Web fingerprinting
  - Fingerprinting with simple GET requests
  - Store everything in visible format
  - gau / waymore every webserver for endpoints
  - Manually kick off spiders on really interesting domains
- Data
  - Lots of caching!
  - Backend storage for data aggregation
  - Easy no code dashboards for visualization (hoping to continue to upgrade this)
  - New asset and heavy change detection
- HUNT :)

THANKS TO THE HOMIES!

- Jason Haddix
  - And the squad of homies!
- Justin Gardner
- Sean Yeoh
- Shubham Shah
- And soooooo many more!

THANK YOU

