Improving Internet Wide Scanning with Dynamic Scanning

Team CIRCL
https://www.d4-project.org/

FIRSTCON21

Alexandre Dulaunoy

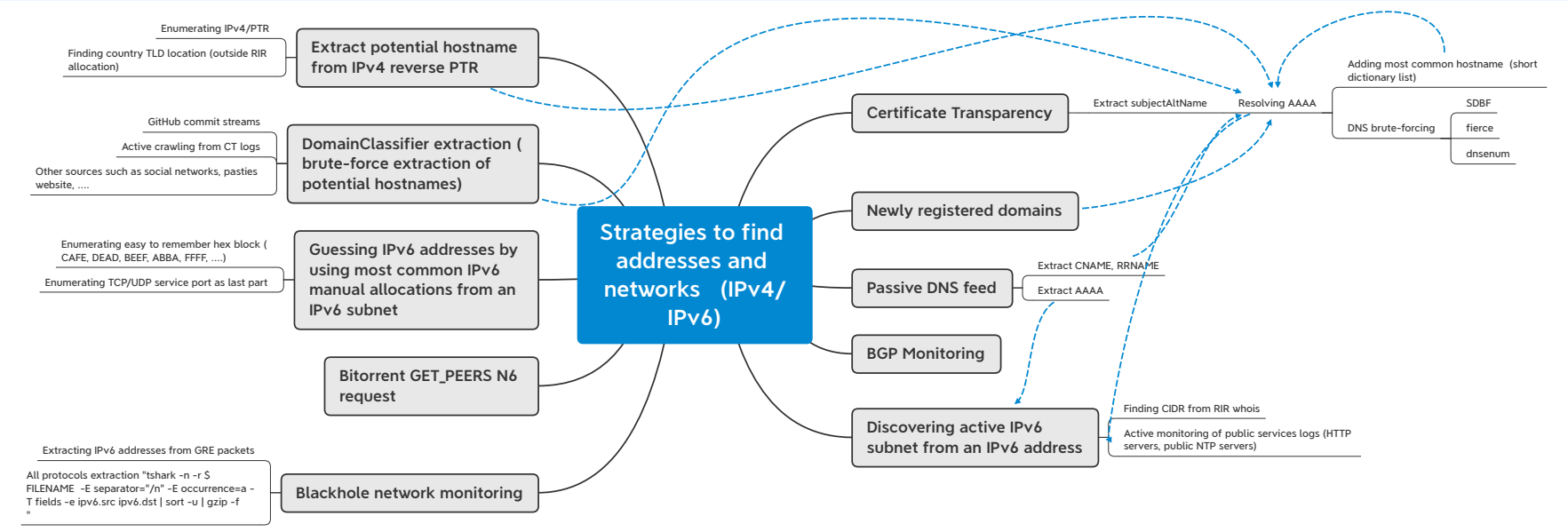
Problem statement

- Finding vulnerable devices can be challenging for CSIRTs (**waiting for the next scan** in Shodan, Censys).
- Finding the scope of the scan (regional versus global, wrong IRR allocation).
- Discovering newly devices exposed without scanning the whole IPv4 space.
- Discovering **named-based services**.
- Discovering **newly exposed devices** or services using IPv6 addresses.

Documenting our Journey

- The goal of the talk is to **summarize the techniques discovered, tested and used** in the past years by CIRCL.
- Showing the **advantages and disadvantages** of each discovery techniques/tools.
- We won't be exhaustive but covering the applicable strategies for CSIRTs.

Overview


Strategies to find addresses and networks (IPv4/ IPv6)
Extract potential hostname from IPv4 reverse PTR
Enumerating IPv4/PTR
Finding country TLD location (outside RIR allocation)
DomainClassifier extraction (brute-force extraction of potential hostnames)
GitHub commit streams
Active crawling from CT logs
Other sources such as social networks, pasties website, ...
Guessing IPv6 addresses by using most common IPv6 manual allocations from an IPv6 subnet
Enumerating easy to remember hex block (CAFE, DEAD, BEEF, ABBA, FFFF, ...)
Enumerating TCP/UDP service port as last part
Bitorrent GET_PEERS N6 request
Blackhole network monitoring
Extracting IPv6 addresses from GRE packets
All protocols extraction "tshark -n -r $ FILENAME -E separator="/n" -E occurrence=a -T fields -e ipv6.src ipv6.dst | sort -u | gzip -f "
Certificate Transparency
Extract subjectAltName
Resolving AAAA
Adding most common hostname (short dictionary list)
DNS brute-forcing
SDBF
fierce
dnsenum
Newly registered domains
Passive DNS feed
Extract CNAME, RRNAME
Extract AAAA
BGP Monitoring
Discovering active IPv6 subnet from an IPv6 address
Finding CIDR from RIR whois
Active monitoring of public services logs (HTTP servers, public NTP servers)

Certificate Transparency

- Certificate Transparency provides a **continuous stream of hostnames** in logs and X.509 certificate.
- Extracting the X.509 certificate and associated `subjectAltName`.
- **Resolving A and AAAA** records for each hostname seen.
- Storing the records[1] last-seen and stream to the scanner the newly seen IP addresses.

 → The enrollment of a certificate doesn't mean that a service is up-and-running.

[1] `https://github.com/D4-project/ct-scrutinize/blob/main/bin/ct-dns-resolver.py`

Brute forcing hostnames

- In all the techniques shown, **the brute forcing hostnames technique is reused** in many steps (from the CT logs extraction, newly registered domains...).
- Brute forcing can be slow and have significant scalability issues.
- A good balance is to have **a minimal dictionary** of the most common hostnames (e.g. a top 20 global or regional).
- More advanced techniques such as Markov Chain Models[2] can be used to recover the hostnames.

[2] SDBF: Smart DNS Brute-Forcer, https://hal.archives-ouvertes.fr/hal-00748792/document

IPv6 finding new hosts

- If you find a specific IPv6 address from other techniques (CT logs), finding IPv6 allocated subnet is easier.
- **IPv6 manual allocation** can be discovered by all the specific tricks such as
  - Compressed IPv6 address plus additional decimal (e.g. prefix::1);
  - Common hex block used (e.g. DEAD, BEEF, ABBA, CAFE, FFFF);
  - Service port in the address (e.g. ::53, ::443).

Finding randomized IPv6 addresses

```
** IPv6 General Address Analysis **

Total IPv6 addresses: 1222
Unicast:              1222 (100.00%)     Multicast:              0 (  0.00%)
Unspec.:                 0 (  0.00%)

** IPv6 Unicast Addresses **

Loopback:                0 (  0.00%)     IPv4-mapped:            0 (  0.00%)
IPv4-compat.:            0 (  0.00%)     Link-local:             0 (  0.00%)
Site-local:              0 (  0.00%)     Unique-local:           0 (  0.00%)
6to4:                    3 (  0.25%)     Teredo:                 3 (  0.25%)
Global:               1216 ( 99.51%)

+ IPv6 Unicast Interface Identifiers +

Total IIDs analyzed: 1222
IEEE-based:             15 (  1.23%)     Low-byte:              95 (  7.77%)
Embed-IPv4:             18 (  1.47%)     Embed-IPv4 (64):       39 (  3.19%)
Embed-port:              0 (  0.00%)     Embed-port (r):         0 (  0.00%)
ISATAP:                  0 (  0.00%)     Teredo:                 0 (  0.00%)
Randomized:           1042 ( 85.27%)     Byte-pattern:          10 (  0.82%)
```

Public service and active scanning

- Providing public and accessible services can help **to collect randomized IPv6 addresses**.
- Such service can be public web services, network services (NTP, STUN, DNS services).
- Bittorrent trackers[3] can be also a source of IPv6 addresses (GET_PEERS N6 request).

[3] Analysis of Bandwidth Attacks in a BitTorrentSwarm - https://openaccess.city.ac.uk/id/eprint/16158/1/Adamsky,%20Florian.pdf

Using IPv4 reverse PTR

```
1 {"timestamp":"1622039107","name":"104.248.18.217","value":"observium.fairit.de","type":"
      ptr"}
2 {"timestamp":"1622056049","name":"104.248.182.44","value":"gitlab.koshigaya.de","type":"
      ptr"}
3 {"timestamp":"1622003182","name":"104.248.19.147","value":"777-slots.de","type":"ptr"}
4 {"timestamp":"1622067602","name":"104.248.19.241","value":"rokmd.de","type":"ptr"}
5 {"timestamp":"1622004046","name":"104.248.19.43","value":"wordpress-
      haftpflichtversicherungdrohne.de","type":"ptr"}
6 {"timestamp":"1621988771","name":"104.248.19.95","value":"forum.sofacoach.de","type":"ptr
      "}
```

- An easy way to find **country allocation outside RIR whois allocation**.
- Extracting the domains and hosts can be used to **feed the DNS bruteforcer**.

Extracting potential hostnames from unstructured text

- **Stream of git commit messages** from GitHub or similar services is a gold mine for potential hostnames.
- Analysing the text to find for any potential domains or hostnames (e.g. DomainClassifier[4] library).
- Valid domains can be then forwarded to the DNS brute-forcer.
- Such technique can be applied on **any unstructured data source** (e.g. social networks, forums).

[4] https://github.com/adulau/DomainClassifier

Newly registered domains and Passive DNS stream

- Feed of **newly registered domains** (some can be downloaded from ICANN[5]).
- Newly registered domains can be then forwarded to the DNS brute-forcer.
- **Passive DNS streams** provide another way to gather recently seen domains but also directly IPv4 (A records) and IPv6 (AAAA records) addresses.

 → Feed of newly registered domains can be costly.

[5] https://czds.icann.org

BGP monitoring

- **Monitoring BGP messages** in real time can be used to **order the priority of scanning** while doing Internet-wide or regional scans.
- Finding new CIDR blocks in IPv6 or IPv4 including stable or unstables networks (e.g. installing new services).
- BGP feeds can be collected from existing BGP sessions or even via the Routing Information Service Live[6] from RIPE.

[6] https://ris-live.ripe.net/

Network telescope and black-hole monitoring

```
1   tshark -n -r \$FILENAME  -E separator="/n" -E occurrence=a -T fields -e ipv6.src ipv6.dst
        | sort
```

- **Unsolicited network traffic** can be analysed to feed the DNS brute force or the IPv4/IPv6 addresses can be extracted.
- IPv6 extraction from encapsulation protocols such as GRE can be provide some IPv6 addresses.
- Source IP addresses can be used as a priority mechanism for scanning (e.g. SSH scanner are more likely to be a vulnerable host).


→ The volume of IPv6 addresses seen can be very low compared to CT log monitoring.

Contact

- Get in touch if you want to share some experiences and you can even do a pull-request on the GitHub repository
- Contact: info@circl.lu
- Slides and notes: `https://github.com/adulau/active-scanning-techniques`
- @adulau @circl_lu @d4_project