DPU TDC 463

Scanning, Probing, and Surveying for Internet Hosts and Services

The Probing Challenge

To quickly, periodically, safely and accurately discover the services of interest, and details about them, on the public Internet through active and passive probing, while minimizing alarms and complaints.

Address Space Considerations

- An entire IPv4 address space scan is feasible
- An entire IPv6 address space scan is impractical
  - “smart” scanning may be feasible
  - IPv6 addresses not considered further here
- Optimization considerations
  - bogon, unused, unallocated addresses
  - parallel probing
  - speed versus noise

Performance Considerations

- Socket creation and usage
- Separate packets sent from response processing
- Timers (time-out and retransmission)
- State-based firewalls
- OS process limits
- Unseen limitations (e.g. VM, network)

Complaint Considerations

- Address selection (randomness) and frequency
- Probe host DNS names (A/AAAA and PTR)
- Probe host web page (with HTTPS and valid cert)
- Public disclosure
- Opt out mechanism
- Template response email

DNS Open Resolver Probing

- Periodically generate random addresses to probe
  - remove bogons, unused and unallocated
  - remove do_not_probe prefixes
- Periodically re-probe known open resolvers
- Generate a unique probe event ID
- Start up a pcap
- Blast out probes, let pcap get responses
  - and do not return ICMP port unreachable
- Parse and process pcap

A Unique DNS Probe Every Time

- One-time unique queries of the form, e.g.:

  `a1234567890p12345i12345.d0123456789.t12345.dnsresearch.cymru.com`
- a1234567890 = 'a' + IPv4 probed address as an integer
- p12345 = 'p' + probe UDP source port
- i12345 = 'i' + DNS ID
- d0123456789 = 'd' + probe group event ID
  - default is the UTC date/time YYYYMMDDHHSS
- t12345 = 't' + DNS TTL of the answer

Open NTP Server Probing

- The strategy is the same as the open resolver strategy
- We have fewer options for unique IDs however
- Default NTP source port based on days since 1970
  - note: Open resolver uses random from 1024-65535
- NTP payload sequence number = source port
- We verify sequence = source port in pcap processing

fpdns: DNS Server Fingerprinting

- Regularly fingerprint the open resolvers
- fpdns.pl state required to read and interpret results
- Fork X number of processes at a time
- Mostly works, with occasional bad data and errors
- fpdns.pl is third party code
  - and not that well maintained anymore

X.509 Certificate Collection

- Can't just blast probe packets and forget them
- Timers and parallel probes key to speed
- Do we just probe based on IP address?
  - use Alexa, DNS or passive DNS data for selection?
  - then we have to worry about DNS look up time too
- There are some 3rd party SSL certificate sources
  - e.g. EFF Observatory
- Just TCP port 443?

TCP Probing

- Generate TCP probes like UDP probes
- Do NOT complete 3-way handshake, return RST
- Those that respond, follow up with openssl connection
  - parse responses pcap (like previous methods)
- Retrieve certificate if available, parse, store

isatap and wpad query collection

- Register “interesting” and popular domain names
- Collect the queries you see
  - DNS query logging
  - pcap
- What answer do you provide?
- Leverage EDNS0 client subnet option?
- Additional info in SOA and TXT RRs

Custom Tooling

- Process control scripts, calls others
- Random address generator
  - Remove bogons
  - Remove DO NOT PROBE systems
- Packet prober (e.g. synscan)
- Background tcpdump collection
- Pcap processor and database inject

Miscellanea

- IGMP/DVMRP probing
- .edu zone reconstruction and evaluation
- zmap is good tool in the tool box now-a-days
- Running tcpdump in cron and capturing in intervals
- Fast versus slow probing