Experiences In Maintaining Long-term Data Collections

Edward Lewis

DINR 2023
22 February 2023

Why do this presentation?

- Many research activities are “one-shot”, one-time data collections, one-time analysis, one-time paper, and perhaps multiple presentations

- The purpose of research is to locate things that need attention or to show that changes have the desired impact

- It’s important to either repeat one-shot research or turn the efforts into long-term data collections or analysis
  - Were the observations “true” or ”one crazy day”
  - Or, were changes made to “fix” the situation?

- One-shot research and long-term research are not exactly the same, there considerations for long-term research related to “long-term”

One-shot vs. Long-term

- In a one-shot effort, a mindset of research is appropriate
  - Given an activity needing measurement or perhaps a large pre-collected data set, what can we find? Perhaps there is a hint of a goal, but often the art is in the originality, the free-form process of finding something interesting
  - This yields a good view of one moment in time
  - Repeating a one-shot a year later, for instance, needs a new data set, of the same format and quality and hopefully the same code is able to run. But sometimes the code has portions that were dependent on the older data, or perhaps on an old OS version

- Shifting to a long-term effort requires a operations mindset
  - Research and operations are very different
  - For a long-term effort, researchers have to anticipate changes to the computing and personnel environment, have to account for long-term storage of data, and backups

My long-term projects

- TLD Apex History
  - An effort to detect how DNSSEC is deployed across top-level domain registries
  - The goal is to look at configurations, not performance, not “service level agreements”
  - This effort has daily data since June 2011 despite changing computing environments, programming languages, and data storage systems

- DNS Core Census
  - An effort to collect meta-data about zones in the global public DNS that are close to the root (top-level domains and more)
  - A goal is to make is easier to categorize zones, categorizations used in other studies
  - Another goal is to catch route origin security (RPKI/ROA) details for zones
  - This effort is still evolving, the current form has been running for two-three years

Pipeline Architecture

- Common to the two efforts is the notion of a pipeline of data
  - Thinking in terms of a pipeline is the first of two mind shifts from one-shot to long-term
  - Three main components of a pipeline
    - Collection
      - Ingest the data in a reliable manner
      - Robust to eliminate most momentary errors
      - Resilient against manageable failures
    - Processing/Reduction
      - Efforts to convert raw data into what is desired
      - Collapse multiple raw data collections into daily (or appropriate) time units
      - Store the data
    - Publication
      - Put the data where it can be accessed by the intended audience
- Each of these components may have many sub-sections, but these three are distinct

Pipeline Resilience

- Common to the two efforts is the management of a pipeline of data
  - Thinking in terms of a *managing* a pipeline is the second of two mind shifts
  - Three main components of a pipeline
    - Collection
      - Activity benefits from multiple vantage points
      - Resilience comes from simply repeating the activity more times than is needed
      - Activity must be managed to avoid “polluting” or flooding the network
    - Processing/Reduction
      - These efforts many be central but be able to move if there is a failure
      - Alerting a human is an art, enough to indicate work is done on time but not overload
    - Publication
      - This is very simple but separate because publication tools change often
      - Approved communities may change, as well as data sharing policies

TLD Apex History

- Collection
  - Many different machines have been used, different operating systems over the years
  - Currently 5 virtual machines are rented, each running the collection 4 times per day
  - The collectors each keep logs and send email upon completion (but this often ignored)
- Processing/Reduction
  - Once a day, one (physical) machine
  - If processing is missed or interrupted, the next scheduled run will fix the problem
  - A failed collection doesn’t matter as data is consolidated daily (20 runs-> 1 day)
  - Email is sent once a day, at a specific time, marking the run and any errors
- Publication
  - Run once a day after the processing script, upon failure it can be run manually
- Originally in shell script, now in python, on MacOS, now various *nix machines
- Data originally stored in text files, now in a relational database

DNS Core Census

- Began in 2019, still evolving, always in python and on a Centos/Debian/Ubuntu base
- Collection
  - One machine, runs every few hours unless there’s been a successful run earlier
  - Only one machine means that some data takes duplicating the TLD Apex History are not as resilient (something for me to fix)
- Processing/Reduction
  - One machine, once per day, one day ”late” because it needs to pull in other data from internally resources and combine
  - Data is maintained “per run” not “per day” but that needs to change to scale into the future
- Publication
  - Internally in a relational database, externally in compressed json and csv per-run files
- Still learning, still evolving, need to incorporate lessons from TLD Apex History (I had forgotten)

Discussion

- Slides done, time for discussion, questions, etc.

Engage with ICANN

Thank You and Questions

Visit us at **icann.org**
Email: edward.lewis@icann.org

- @icann
- facebook.com/icannorg
- youtube.com/icannnews
- flickr.com/icann
- linkedin/company/icann
- slideshare/icannpresentations
- soundcloud/icann
- instagram.com/icannorg