Towards an Immune System for the Internet

Motivation

Internet constantly under attack

Cybercrime damages: est. $6T by 2021

Extremely powerful adversaries funded by nation states

Key problem: lack of cooperation, decentralization, lack of information sharing

Everyone is addressing cybersecurity on their own

Towards an Internet "Immune System"

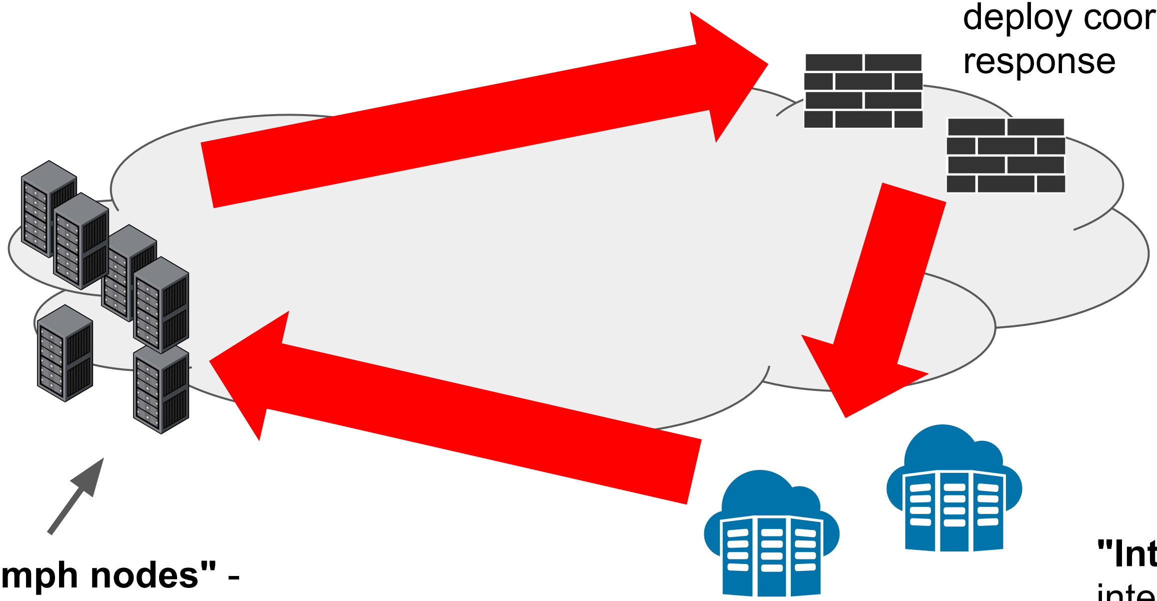
Human body takes a very different approach to threat

- Coordinated, system-wide information sharing
- Coordinated, system-wide learning
- Coordinated, system-wide response

Single, cooperative system maximizes visibility and response to threat

What if we could do this in the Internet?

Approach

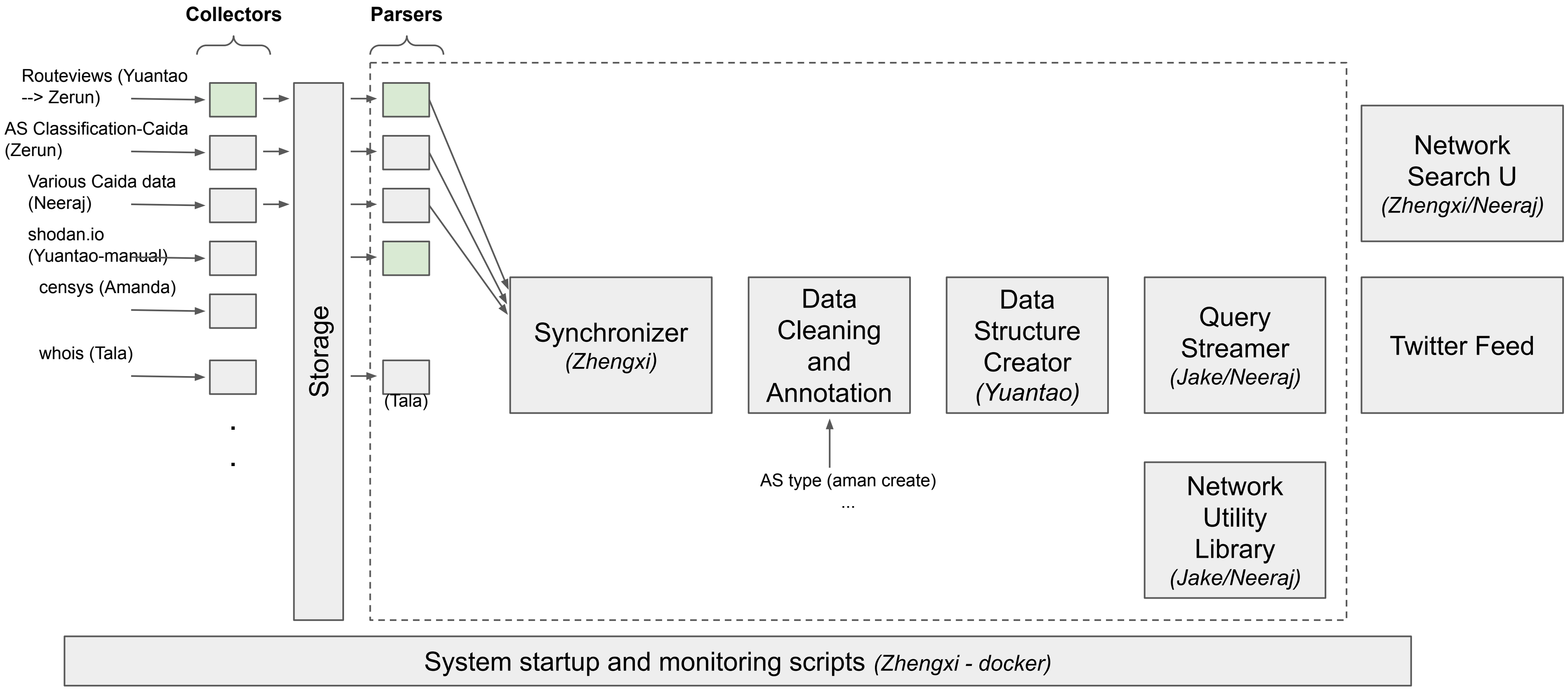


"Internet B Cells" - deploy coordinated response

"Internet Lymph nodes" - dedicated traffic scrubbing facilities monitor and detect infections from many vantage points

"Internet T Cells" - intelligent and resilient repositories for information about attacks - stores immunological memory

System Architecture



Collectors

Parsers

Routeviews (Yuantao --> Zerun)

AS Classification-Caida (Zerun)

Various Caida data (Neeraj)

shodan.io (Yuantao-manual)

censys (Amanda)

whois (Tala)

. .

Storage

(Tala)

Synchronizer *(Zhengxi)*

Data Cleaning and Annotation

AS type (aman create)
...

Data Structure Creator *(Yuantao)*

Query Streamer *(Jake/Neeraj)*

Network Utility Library *(Jake/Neeraj)*

Network Search U *(Zhengxi/Neeraj)*

Twitter Feed

System startup and monitoring scripts *(Zhengxi - docker)*

Internet Search Engine (need better name)

Google maps for the internet

Users can type in queries and get results

Visualization

Think about what to visualize/show

- "Shame on you" lists
- Graphical stuff using D3
- Search - try to do better than regexs
    - Does there exist a path going from AS X to AS Y going through AS Z -- good regex

Data sources

routeviews.org

https://www.caida.org/data/overview/

shodan.io - free api?

censys

GreyNoise (greynoise.io)

Example Streaming Queries

https://docs.google.com/document/d/12IyEyjGTz6q_BD85frU-QpShQ1zEDjDFIHqApTQqEIE/edit

Are enterprise hosts more or less secure than university hosts?

- internet.select_time(Jan 1 2019).ASset().enterprise().hosts().get_insecure(signature).size([single query]

Produce a graph showing how many enterprise hosts with DNS name including the string "news" arrive every day

- internet.ASset().enterprise.hosts().select_dnsname("*news*").registercallback(&mycount)
  - mycount(event e): output[e.day]++
- ~~internet.ASset().enterprise.hosts().registercallback(&mycount~~)
  - ~~mycount(event e): if (e.host.dnsname("*news*")) output[e.day]++~~

Show me the trend of number of IP prefix reallocations in the 10/8 supernet over time, show a monthly count.

- internet.IPprefixset().select_prefix("10.0.0.0/8").events(Jan 2016).reallocations()[single query]
- internet.IPprefixset().events().select_prefix("10.0.0.0/8").registercallback(&count_monthly[streaming]
  - count_monthly(event e): output[e.month]++;
  - internet.start_run()

Amanda's ideas

multiresolution, time based-graph - nodes annotated w time

nlp

what should intents be?

Example Realtime Queries

[move towards maintaining compact data structures that can be interactively queries]

Key tasks

1. Figure out list of data sources we can get
2. Start writing collectors for them and start them running so we can start collecting data
3. Start writing parsers for them
4. Figure out what data sources we can create (eg manual AS classification)
5. Start implementing other components and scripts
6. Start implementing library of network data utilities

Initial Query List

1. How much activity is there from a particular AS today? (how many prefix withdrawals/advertisements have that AS as its origin - make sure it's withdrawn from all vantage points, compute intersection)

2. How many prefix hijacks are there today? (get list of most important prefixes, and then note when they are hijacked) - give me a list of prefix hijacks (a prefix is hijacked if it's advertised by more than one AS)

3. How bad are route leaks today? (get # of prefixes each AS advertises, report if anomalous - get list of ASes in 99th percentile)

TODO

- script to log memory usage

- catch signals to debug if run out of memory

- patricia trie structure to quickly look up prefixes, hosts within prefix, etc