This material is based upon work supported by the Defense
Advanced Research Projects Agency (DARPA) under Contract No.
HR001119C0102. The opinions, findings, and conclusions stated
herein are those of the authors and do not necessarily reflect
those of DARPA.

Capturing data using a Network Telescope

Introduction

A network telescope is a system that collects packets observed in blocks of unused IP addresses. In our telescope, we used a passive system where our network infrastructure statically routed specific blocks of unused address space to a single switch port, and we collected all of the packets forwarded to that port.

Capturing data

We use `tcpdump` to capture packets. Wireshark or other utilities can be used in a similar manner. Some fast NICs also have special utilities that can be used to capture packets at very high rates. For modest-sized telescopes (up to a few

tens of thousands of addresses), ordinary hardware is more than adequate for packet collection. Large telescopes, monitoring millions of addresses, can require sophisticated engineering to ensure that the collection system can keep up, but this document only describes our tools that are adequate for modest telescopes.

For tcpdump, we use the following command:

```
sudo tcpdump -i IFACE -s0 -w DATANAME-%F-%H.pcap -G 3600 ip
```

Where IFACE is the name of the network interface from which to capture packet, and DATANAME is the name of the telescope. (See the documentation on the parameters, in `../../params`, for more information.)

The shell script `pcap-capture.sh` in this directory provides a wrapper for this command, which makes sure that the capture starts exactly at the beginning of an hour. This script is described in the next section.

As long as the system remains online and no errors cause the command to terminate, this tcpdump process command will continue to run on the system, rotating file captures every 3600 seconds and capturing the full packet. At the start of the next hour, the previous hour's file is closed and a new file is created for the new hour of data. Filenames are named and distinguished using the date and the hour of collection. File sizes are manageable taken in these increments. Refer to the documentation for tcpdump to get more information on the various options.

In retrospect, the timezone should have been set to UTC. Although, running the above command line above with setting the TZ environment variable to UTC can get around using the local timezone on the system.

The command would be:

```
TZ=UTC sudo tcpdump -i IFACE -s0 -w DATANAME-%F-%H.pcap -G 3600
ip
```

Having collected over a long period of time, there were periods were data failed to be collected or data was lost. Some were due to human errors, but others were caused by hardware failures as a disk going bad or the network being disconnected. Because we only had one NIC, data was transferred semi-weekly from the system manually with an external disk used as a data mule. A second NIC connected to the system would alleviate having to transfer data by hand as that would be the conduit to get the data to an analysis system. The external monitoring process to check whether the telescope was online helped with knowing the network was disconnected or something happened to the collection system.

Disk failures were addressed by having a backup of the data being collected. A Unix cron job was eventually written that would copy the last file closed and transfer it to a second disk on the system. Only about a month's worth is backed up to that second disk due to space limitations. Ideally, it would be good to keep as much as possible on the backup disk instead of just a subset of the data files.

Using pcap-capture.sh

The `pcap-capture.sh` takes its parameters from a parameter file, or from the environment (including assignments on the command-line)

There are three required and two optional parameters:

- *IFACE* - the name of the interface from which to capture packets (e.g., `eth0` or `eno0`)
- *PCAPDIR* - the path to the directory where the new pcaps are stored
- *DATANAME* - the name of the telescope, which is used as a part of filenames created by the capture
- *PCAPRAWDIR* (optional) - the path to the directory where the new pcaps are stored

  If both PCAPDIR and PCAPRAWDIR are provided, PCAPRAWDIR takes precedence.
- *TIMEZONE* (optional) - the timezone to use for the timestamps in the filenames, expressed in the simple string form (e.g. "EST" or "UTC")

The usual way to run `pcap-capture.sh` is to define these parameters in a parameter file (since some of these parameters are used by other tools, and the same parameter file can be used with those tools as well). Alternatively, the values can be passed as environment variables, or defined explicitly on the command-line. For example:

```
IFACE=eth0 PCAPDIR=/u1/ DATANAME=palomar TIMEZONE=UTC ./pcap-capture.sh1
```

Note that `pcap-capture.sh` waits until the beginning of an hour to begin capturing data – it usually does not begin capturing immediately.

The names of the output files, created in `$PCAPDIR` (or `$PCAPRAWDIR`) have the form `$DATANAME-YYYY-MM-DD-HH.pcap` where YYYY is the year, MM is the month of the year (01-12), DD is the day of the month (01-31), and HH is the hour of the day (00-23), relative to TIMEZONE (or local time, if no TIMEZONE is given).

Lessons learned and advice for the future

Our initial plan was to collect data for a few weeks, or maybe a month. The initial data was interesting enough that we continued, however, and at this point our data collection has gone on for much longer than planned. If we'd anticipated this, we would have spent more time designing our collection system in order to:

- *Use redundant hardware, with a UPS*

  We lost data due to hardware issues: power outages, the failure of a power supply in our collection host, and the failure of a disk drive. We should have built more redundancy into our collection system.

- *Use a watchdog monitor to detect failures*

  We also lost data due to human error: network administrators breaking the routing tables or disconnecting cables, and lab managers turning off parts of our infrastructure that they thought were unused.

  A watchdog monitor will send alerts if collection system loses connectivity with the network, allowing problems to be detected and corrected without delay.

  We had a network monitor that watched for loss of connectivity to our capture host, but it turned out to be less reliable than expected, which meant that a disconnected cable could take several hours to detect, and sometimes the network monitor also failed – which could turn this into days.

- *Use a better and more reliable source of time information*

  Make sure that your capture host is running a properly-configured time synchronization client, and that your host can reach the time servers it needs. Our capture host was kept in a network sandbox, because the capture was done outside our company firewall, and this complicated access to time servers. Because of this, our initial system was not able to reliably sync its clocks, and many hosts have surprisingly bad internal clocks, which can drift by large amounts if they are not corrected.

- *Use UTC timestamps instead of local time*

  Local time gets awkward when the switch to/from daylight savings time occurs.

  Comparing data from different telescopes is also awkward if the timestamps in the filenames are expressed relative to different time zones. It's better to use a single timezone for all your collections, and UTC is the obvious choice.

Our telescope

Our collection system runs Ubuntu 20.04 and it has a single gigabit Ethernet network interface card (NIC). External USB disks were added for storing captured packets, and for some temporary data storage redundancy. An outside watchdog monitor process was set up to check that the collection system was online.

Our telescope contained different subnets at different times. At its largest, our telescope contained 37 /24 subnets, for a total of 9472 different addresses.

From the networking point of view, a more ideal setup would have been a system that could support two NICs, one for the data collection and the other one as a way to retrieve the data collected by the system, access the time servers, be used by the monitoring system, etc.

From a storage point of view, an ideal setup would have been at least two large disks with a lot of storage. External disks worked well enough, but having the internal storage would be more desirable. A large telescope can quickly gather enough data to make data management an issue; many large disk drives should be on hand to support storage.

Note that efforts to conserve disk space (such as gzip'ing the pcap files) required a lot of computation (to compress and decompress the files). If you cannot obtain enough storage for the raw data, try using lz4 instead of gzip for compression – it does not shrink the data as much, but it runs an order of magnitude more quickly.

Related reading

- R. Pang, V. Yegneswaran, P. Barford, V. Paxson,and L. Peterson. Characteristics of Internet Background Radiation. In Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC), Oct 2004
  - https://www.researchgate.net/publication/2946564_Characteristics_of_Internet_Background_Radiation
- Wustrow, Eric, et al. "Internet background radiation revisited." Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. 2010
  - https://www.researchgate.net/publication/220269699_Internet_Background_Radiation_Revisited
- Iglesias and T. Zseby, "Pattern Discovery in Internet Background Radiation," in IEEE Transactions on Big Data, vol. 5, no. 4, pp. 467-480, 1 Dec. 2019, doi: 10.1109/TBDATA.2017.2723893.
  - https://ieeexplore.ieee.org/document/7970183
- Karyn Benson, Alberto Dainotti, kc claffy, Alex C. Snoeren, and Michael Kallitsis. 2015. Leveraging Internet Background Radiation for Opportunistic Network Analysis. In Proceedings of the 2015 Internet Measurement Conference (IMC '15). Association for Computing Machinery, New York, NY, USA, 423–436.
  - https://www.merit.edu/wp-content/uploads/2016/01/Leveraging_Internet_Background_Radiation.pdf
- Iglesias Vázquez, Félix & Zseby, Tanja. (2015). Entropy-Based Characterization of Internet Background Radiation. Entropy. 17. 74-101. 10.3390/e17010074.
  - https://www.researchgate.net/publication/274020470_Entropy-Based_Characterization_of_Internet_Background_Radiation
- Samuel Oswald Hunter. 2010. A Network Telescope Visualization Framework
  - https://www.cs.ru.ac.za/research/g07h3314/oldsite/resources/thesis.pdf