Provenance Histogram Visualization

Peer review 2

Junfeng and Michael, CSPC 547 Information Visualization

Background: Provenance

Data explains from origin to a certain state
It is used to track history of documents, artworks and scientific researches.

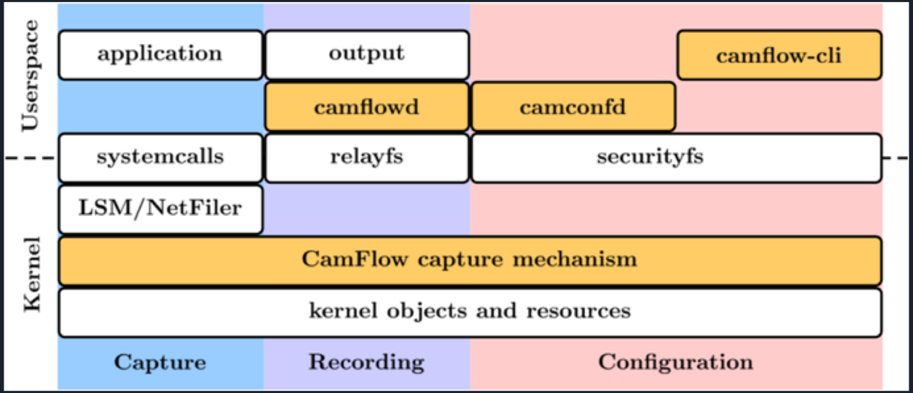
Background: Provenance

Imagine we could know what Linux does for every ‘**system call level**’

Research Reproducibility ..
Data Loss Prevention ..
Digital Forensic ..

“We are focusing on an Anomaly Detection”

Background: Provenance



Userspace

Kernel

application

output

camflow-cli

camflowd

camconfd

systemcalls

relayfs

securityfs

LSM/NetFiler

CamFlow capture mechanism

kernel objects and resources

Capture

Recording

Configuration

University of BRISTOL

Bristol Cyber Security Group

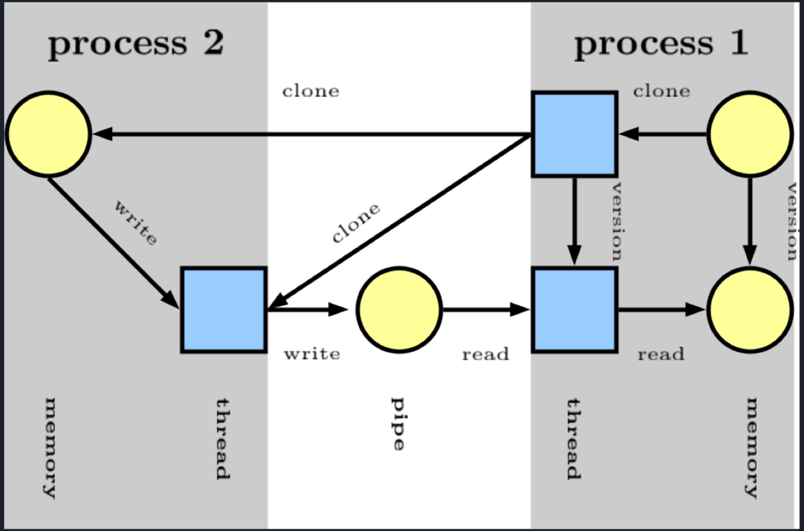
THE UNIVERSITY OF BRITISH COLUMBIA

UNIVERSITY OF CAMBRIDGE

HARVARD UNIVERSITY

- We will use a framework named Camflow to extract system provenance
- The framework captures the system provenance at system call level using built-in Linux framework. (Linux Security Model / NetFilter)

Background: Provenance



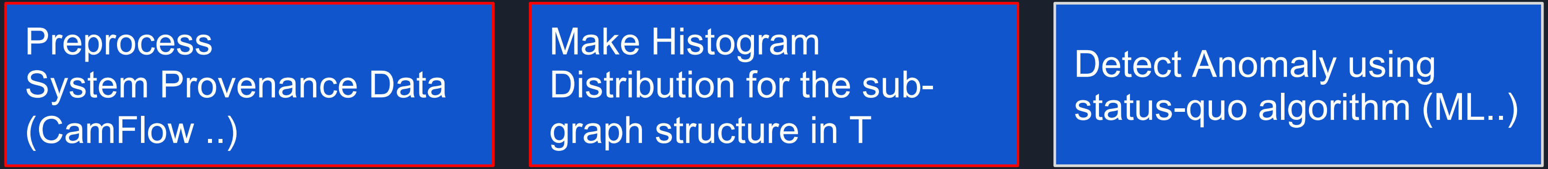
Directed Acyclic Graph
The following is a example of a provenance record

"activity":
{
	"cf:AQAAAAAAECn2QMAAAAAABcAAAAeO5YAAAAAAAAAAAA=":
	{"cf:id":"252327","prov:type":"task","cf:boot_id":23,"cf:machine_id":"cf:9845534","cf:version":0," …
"cf:rbytes":"0","cf:wbytes":"0","cf:cancel_wbytes":"0","prov:label":"[task] 0"}
}

# of Node Attributes: 27 (Task, Socket, File ..)
https://github.com/CamFlow/camflow-dev/blob/master/docs/VERTICES.md
# of Edge Attributes: 96 (Read, Write, Version..)
https://github.com/CamFlow/camflow-dev/blob/master/docs/RELATIONS.md

Background: UNICORN

- A research to detect an anomaly by analyzing the provenance data (Not published yet, thus we don't describe the algorithm part in detail)
- It abstracts the edge information using a hash operation. Which makes possible to generalize provenance data from several framework including Camflow
- UNICORN consists of following steps, and **we want to help provenance researchers to visualize the histogram and provenance data**



Preprocess
System Provenance Data
(CamFlow ..)

Make Histogram
Distribution for the sub-graph structure in T

Detect Anomaly using
status-quo algorithm (ML..)

Data Abstraction: Log Form

After preprocessing provenance data (from Camflow, in our case) UNICORN output

```
0 1 56888932933 56888932933 56888932933 2
```

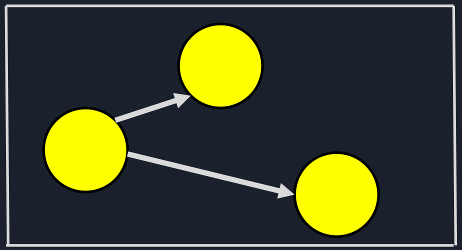
It abstracts an edge information, from left side,

- Source Node ID
- Destination Node ID
- Source Node Type: Hash value of four fields * “prov:type, cf:secctx, cf:mode, cf:name”
- Destination Node Type: Same hash value arguments with the source node type
- Edge Node Type: * “prov:type, cf:flag”
- Relative Timestamp

* We didn’t specify the cardinality of each field for this slides for simplicity. But we have those information in Camflow website, W3C website (World Wide Web Consortium specified the format of provenance, and Camflow uses it), and Camflow log itself.

Data Abstraction: Histogram

UNICORN process a list of edges (Edge information in previous slide), and it abstracts the sub graph structure as a hash value



*It hashes the sub-graph structure several time, not just once, hopping neighbor nodes

```
20172839394839 -> 1
29399404094949 -> 3
...
```

Left value is a hash value represents the sub-graph structure, and Right value is the count for the each element. We are discussing that how much we could refer the UNICORN mechanisms

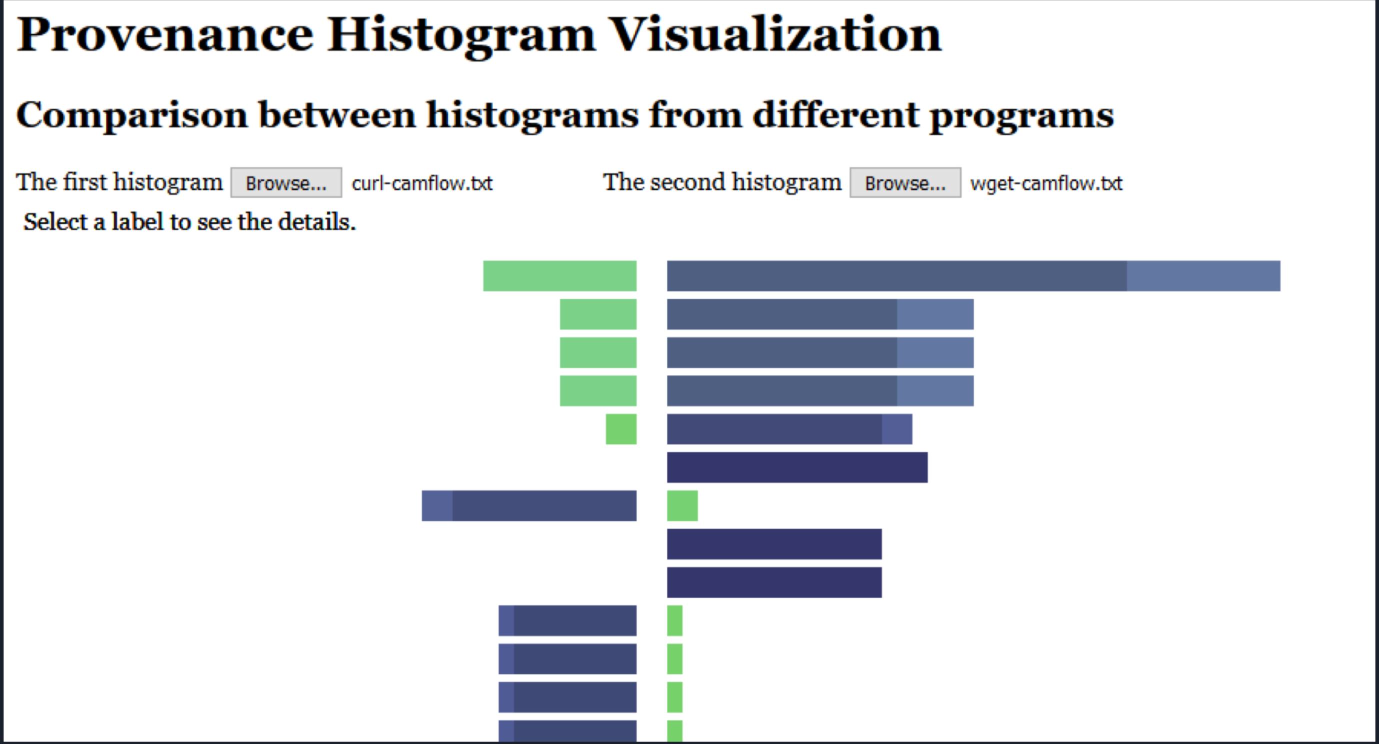
Domain Specific Task Abstraction

- Finding and analysing anomalies that triggered UNICORN.
- Exploring the `shape’ of the histograms of different programs
- Identify the overall trends in time-series histograms.
- Comparison of histograms
    - Either between histograms at **different timestamps**;
    - or between histograms of **different programs**.

Vis Implementation

- Custom web application built upon D3.js
- All data preprocessed

Rudimentary Draft: For Comparison



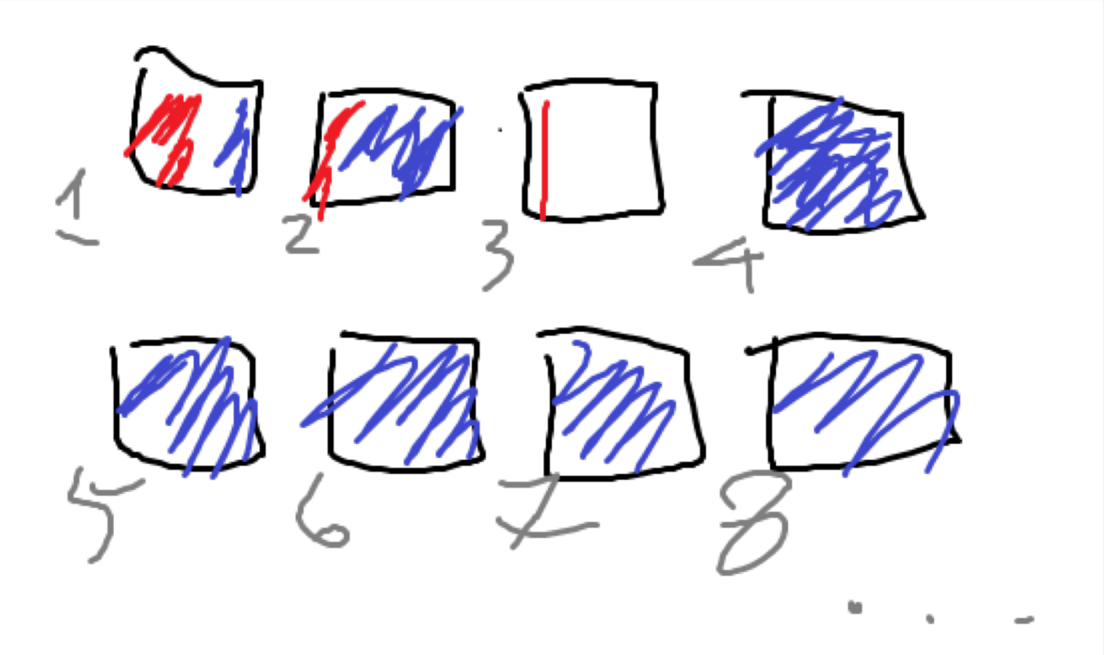
Provenance Histogram Visualization

Comparison between histograms from different programs

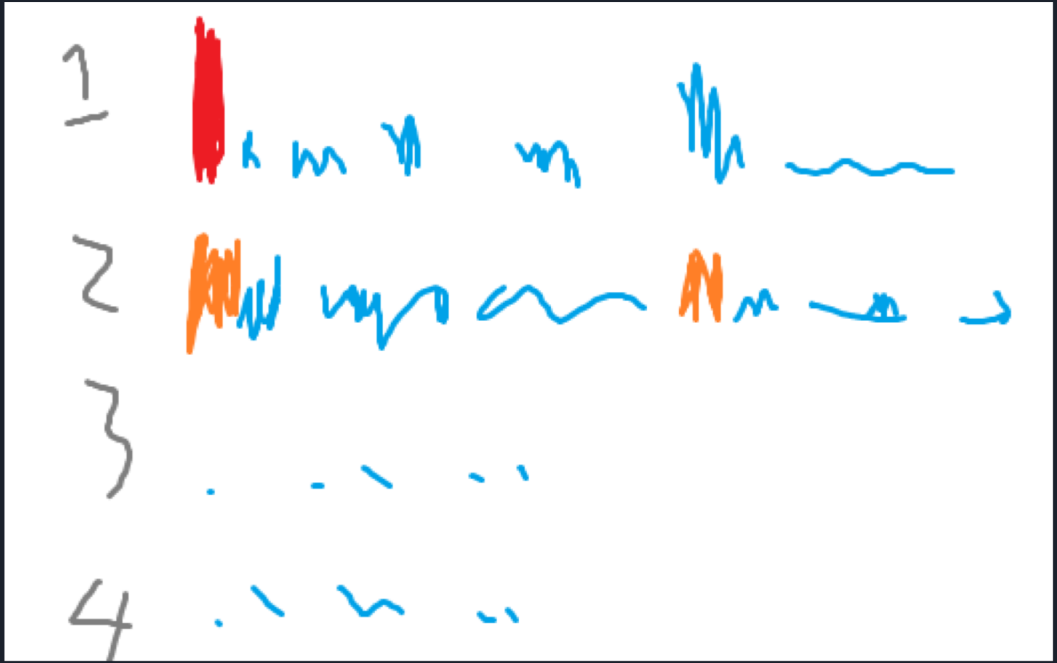
The first histogram Browse... curl-camflow.txt The second histogram Browse... wget-camflow.txt

Select a label to see the details.

Idea One: Time series vis



Idea Two: Time series vis



Challenge: Time series vis

- Showing the entire histogram is not possible for time series view.
- Need to reduce the number of features to be shown
- Possible candidates:
  - The most prominent histogram bins?
  - Some statistics (e.g. distribution of the bins? Total number of items?)
  - A `derivative’ from the last time stamp?
  - Some other derived data that somehow represents the `importantness’ of a certain timestamp?

Evaluation Plan

- Interview with domain experts who has good understanding of the UNICORN system.
- Systematic evaluation not feasible due to short time.

Comments Received

Question: Is time an axis in the data? Answer: Time is a part of the data, but we mainly focus on the structure of the graph.

Reflect: May need more description regarding colours.

It is important to be able to see the difference of the difference. Perhaps put two histograms bars on the same side. Perhaps add a switch: both sides and interleaved. Can use D3 transition - can keep track of where everything goes.

Encoding the difference twice might be confusing.

Add scales.

Time series - 80% of what exists in one frame get carried to the next one, so the unimportant ones can be dropped.