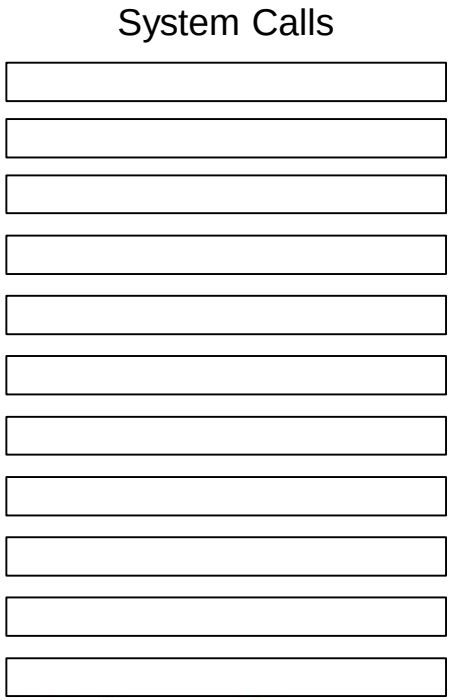
University of BRISTOL

Building a provenance-based IDS
and the questions we ask ourselves

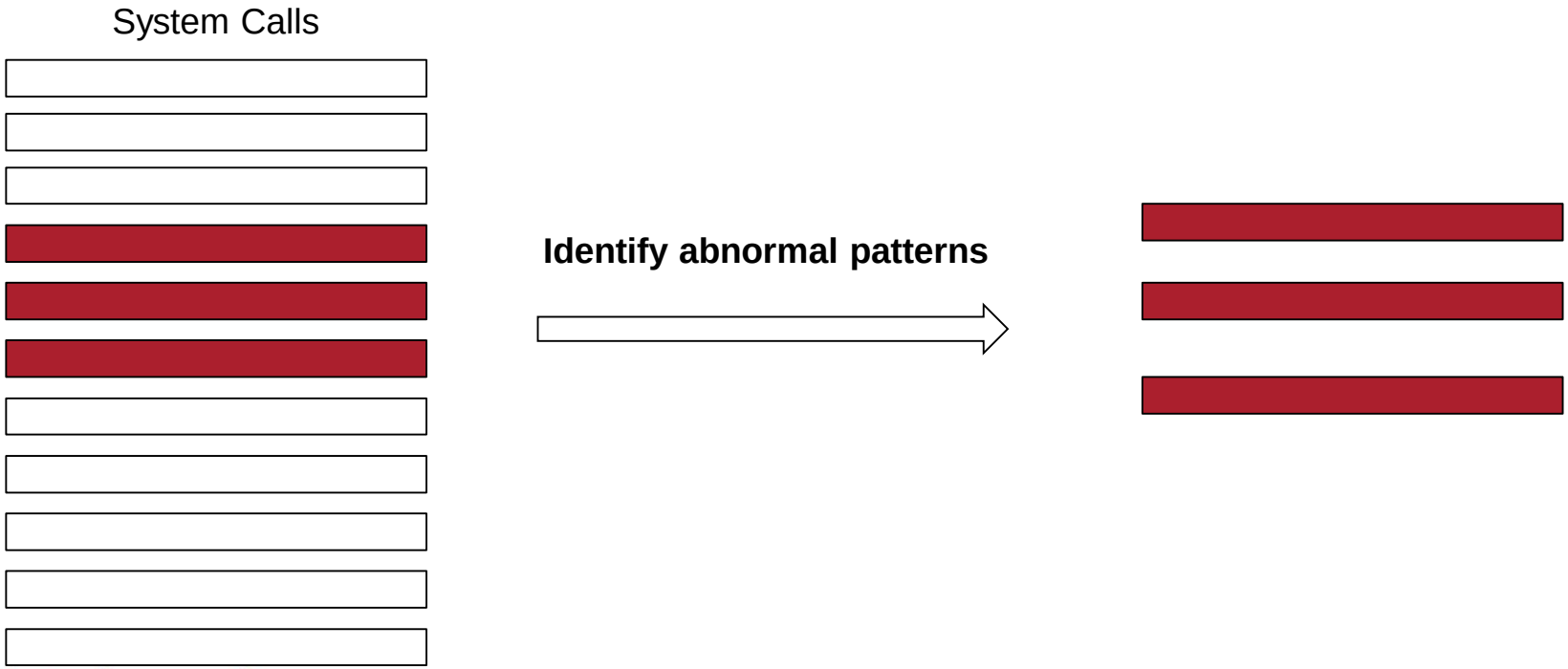
Thomas Pasquier
11/11/2019, Alan Turing Institute

bristol.ac.uk

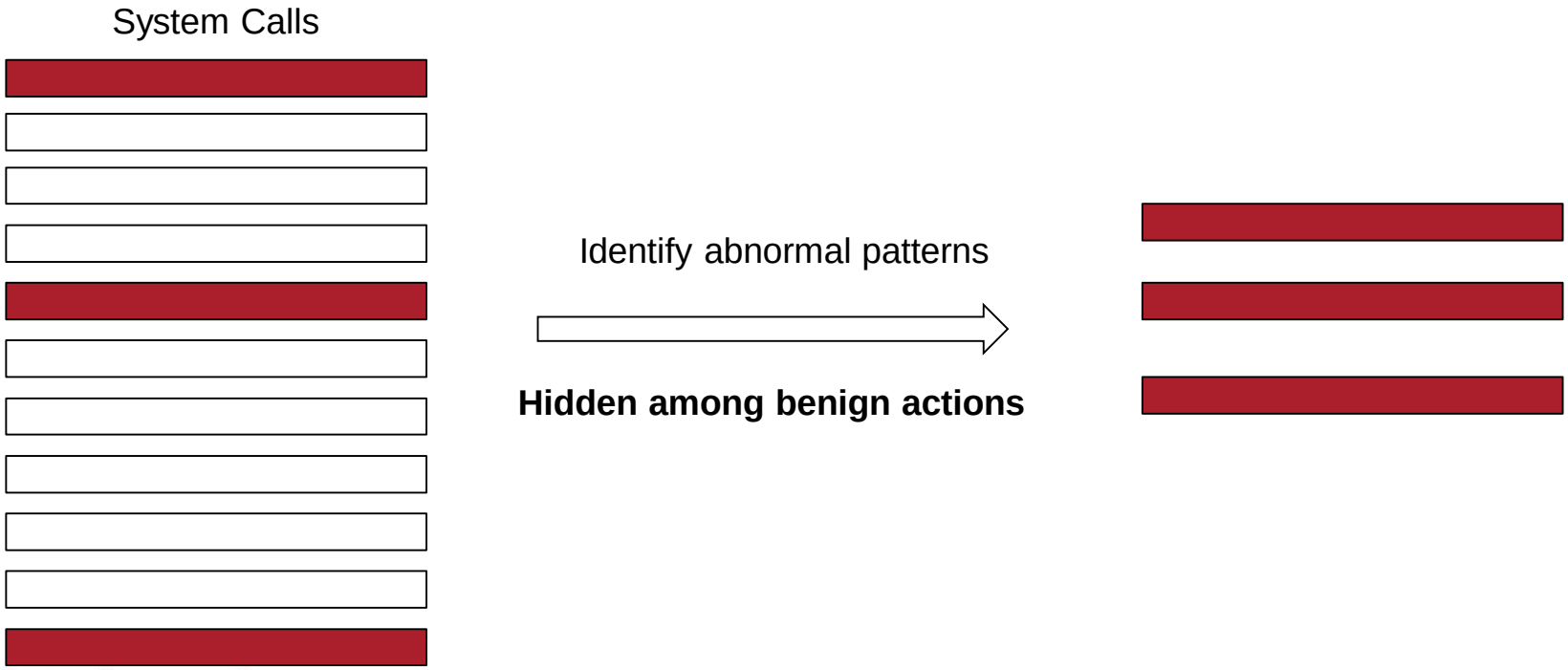
Host-based intrusion detection



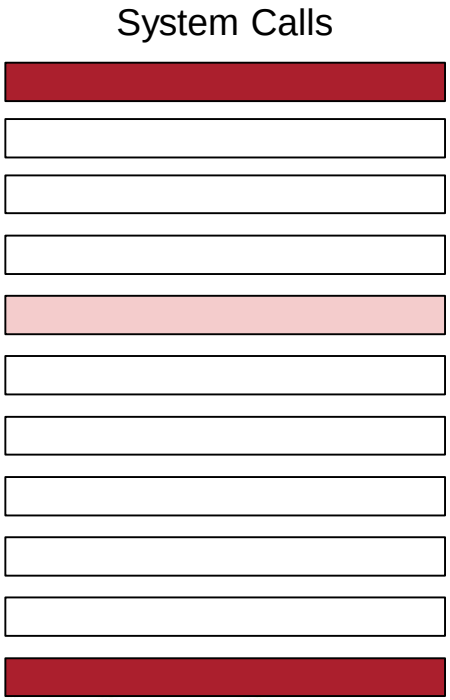
Host-based intrusion detection



Host-based intrusion detection



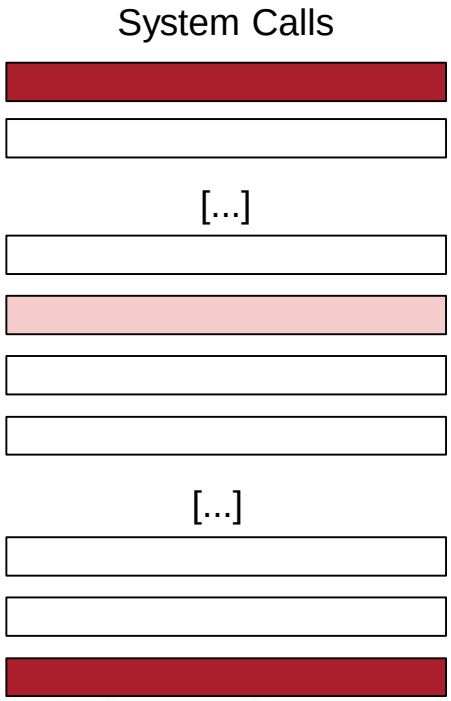
Host-based intrusion detection



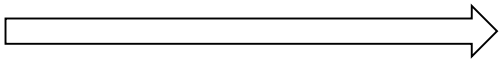
Identify abnormal patterns

Hidden among benign actions
Masquerading as bening action

Host-based intrusion detection

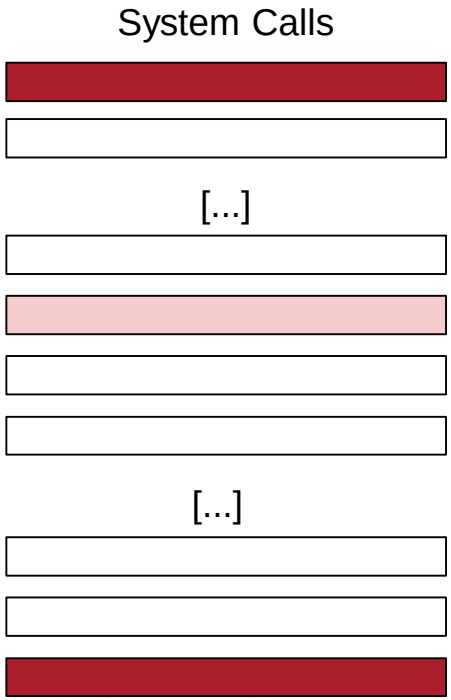


Identify abnormal patterns



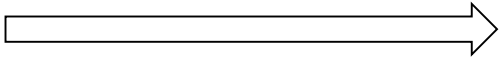
Hidden among benign actions
Masquerading as bening action
Over a long period of time

Host-based intrusion detection



Harder and harder to detect pattern

Identify abnormal patterns



Hidden among benign actions
Masquerading as bening action
Over a long period of time

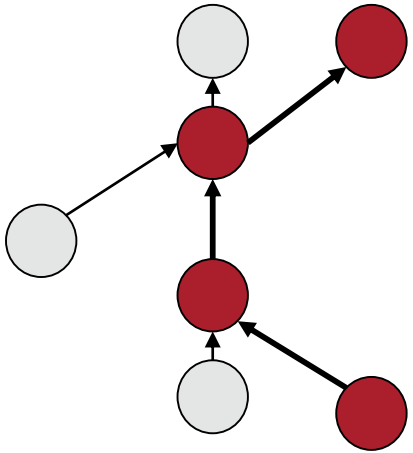
University of BRISTOL

What to do?

bristol.ac.uk

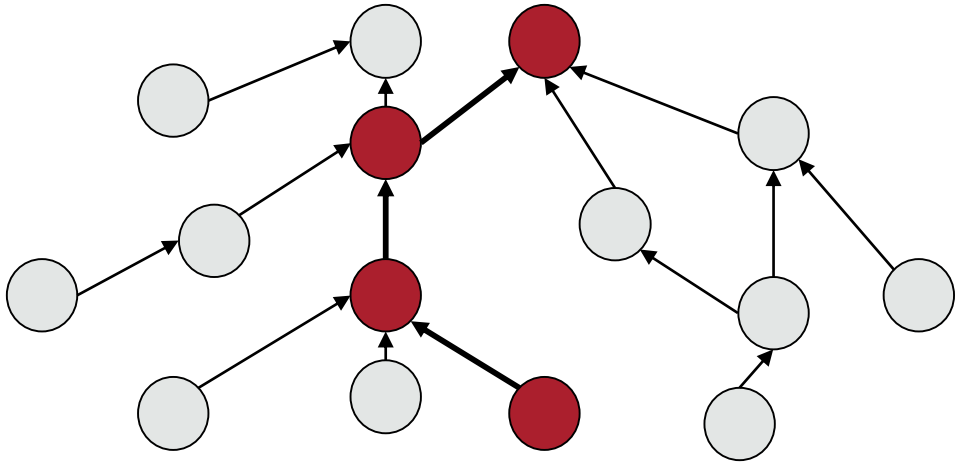
Provenance-based intrusion detection

- **Intuition**: provenance graph **exposes causality relationships** between events



Provenance-based intrusion detection

- **Intuition**: provenance graph **exposes causality relationships** between events



What is provenance in an operating system?

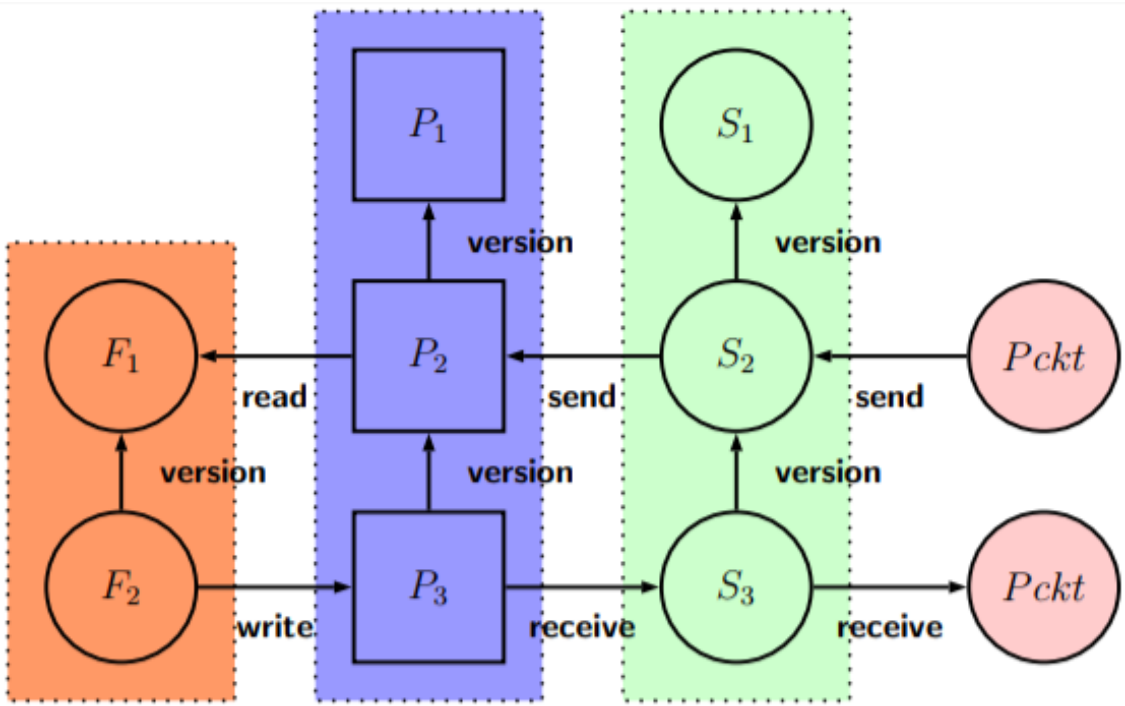
- Represent interactions between system objects
- Represented as a **directed acyclic graph**
- Information Flows
- **Relationship** between **kernel object states**
- History of a system execution

What is provenance in an operating system?

- Represent interactions between system objects
- Represented as a **directed acyclic graph**
- Information Flows
- **Relationship** between **kernel object states**
- History of a system execution

Or that is what we did with CamFlow… more on this later

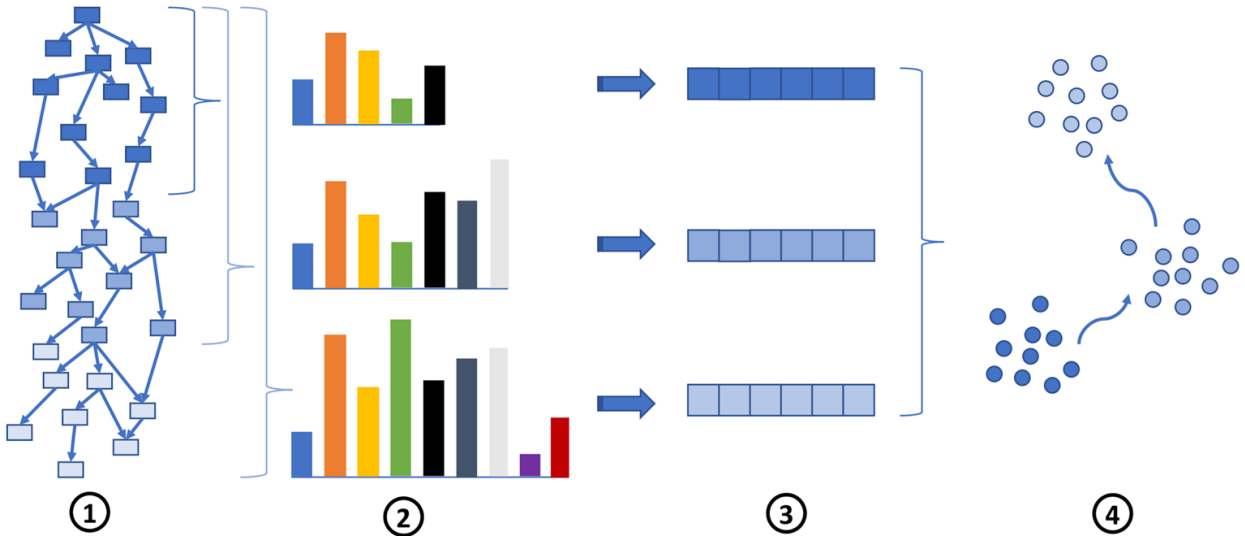
Provenance graph example



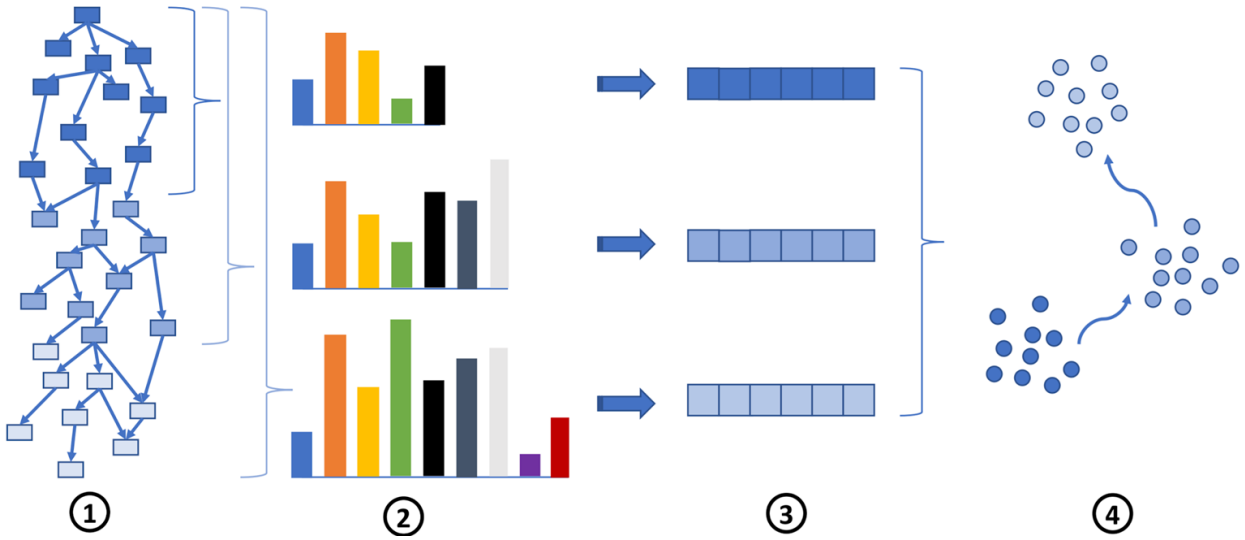
Provenance-based Intrusion Detection

- We target environment with minimal human intervention
  - Relatively well defined behaviour
  - In particular CI/CD pipelines
- Build a model of system behaviour (unsupervised, batch training)
  - in a controlled environment
  - from a representative workload
- **Detect deviation from the model**
- Several approaches being explored…

Detecting intrusion (an example)

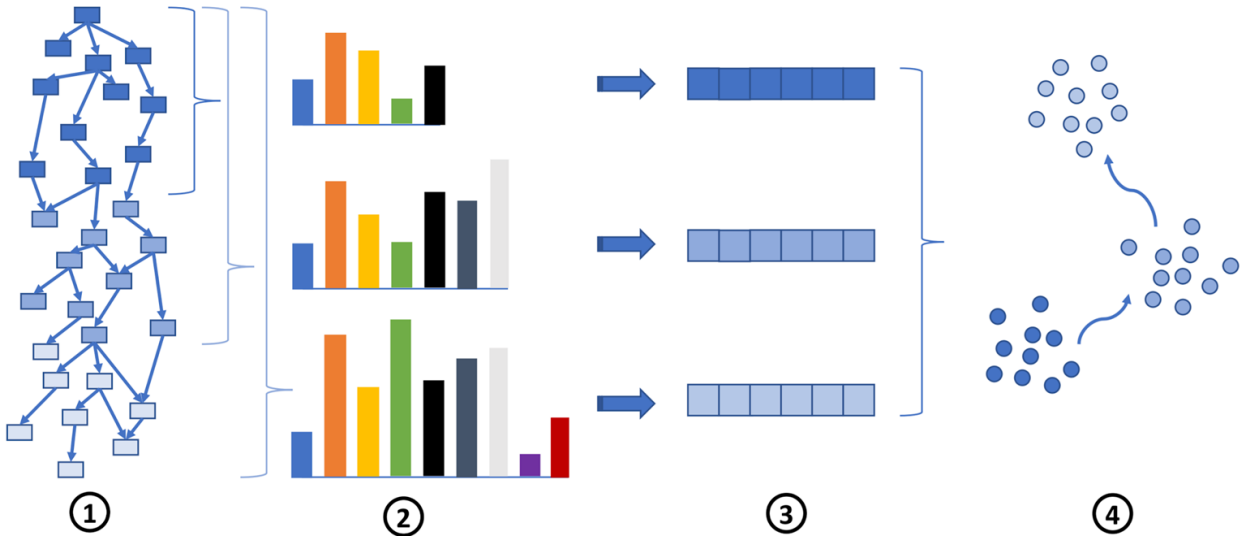


Detecting intrusion (an example)



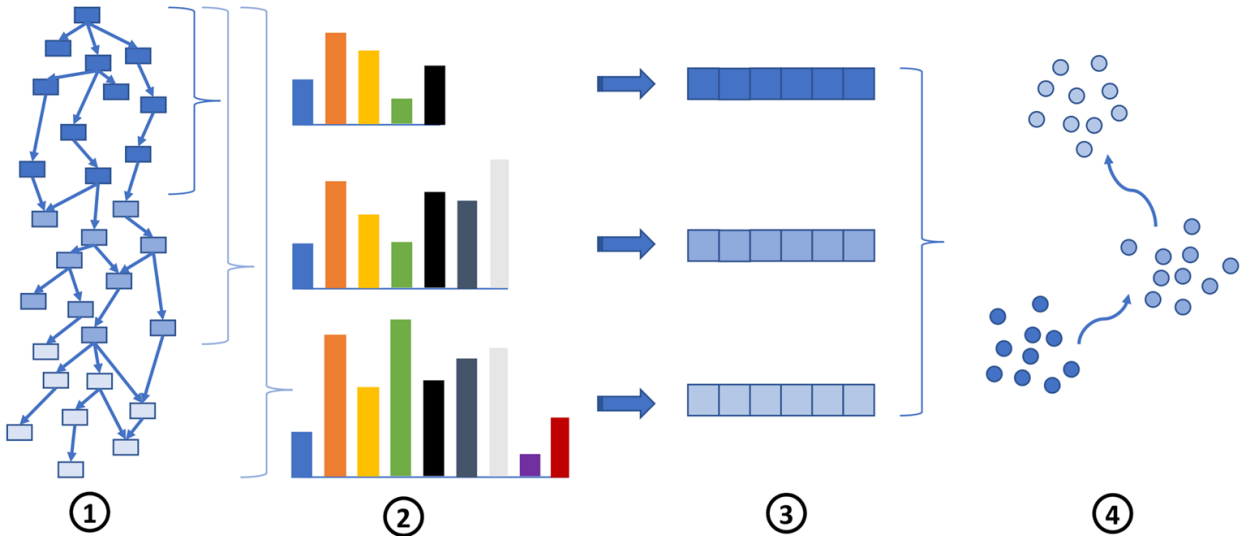
1) Graph streamed in, converted to histogram, labelled using struct2vec

Detecting intrusion (an example)



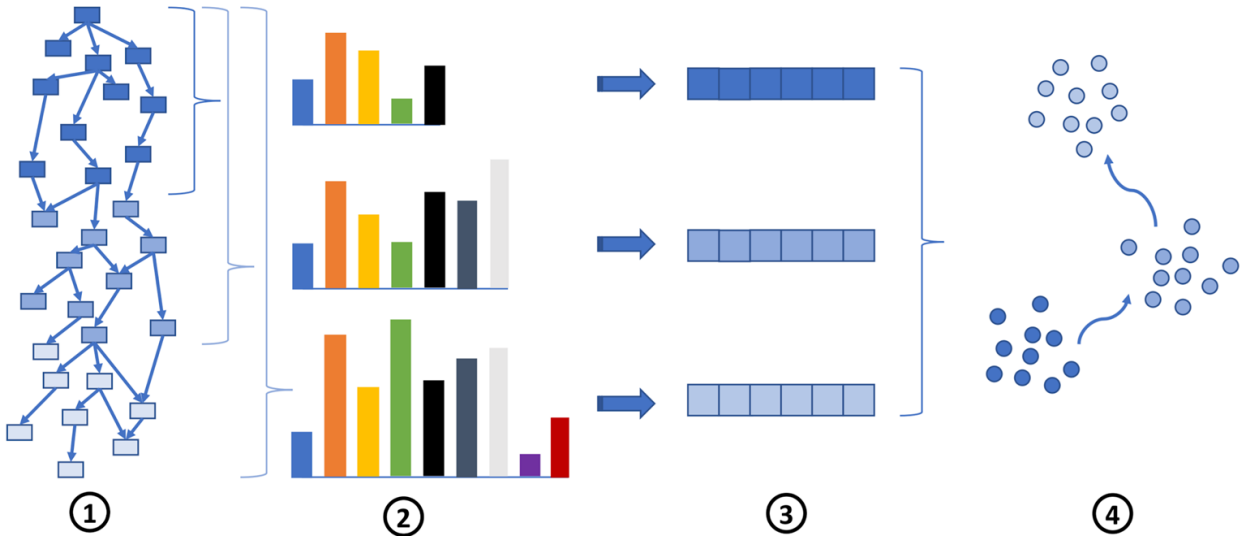
2) At regular interval, histogram converted to a fixed size vector using locality-sensitive hashing

Detecting intrusion (an example)



3) Feature vectors are clustered

Detecting intrusion (an example)



4) Cluster forms “meta-state”, transitions are modelled

In deployment, anomaly detected via clustering and “meta-state” model

Can we interpret the results?

- Sort of…
  - Previous scheme
  - Han et al., **FRAPpuccino: Fault-detection through Runtime Analysis of Provenance**, HotCloud 2017
  - Manzoor et al., **Fast Memory-efficient Anomaly Detection in Streaming Heterogeneous Graphs**, KDD 2016
- "There is something wrong in this **arbitrary subgraph**"
- **Need further analysis tool to understand**

Does provenance-based IDS work?

Experiment	Precision	Recall	Accuracy	F-Score
StreamSpot (baseline)	0.74	N/A	0.66	N/A
$R = 1$	0.51	1.0	0.60	0.68
$R = 3$	0.98	0.93	0.96	0.94

TABLE II: Comparison to StreamSpot on the StreamSpot dataset. We estimate StreamSpot's average accuracy and precision from the figure included in the paper [85], which does not report exact values. They did not report recall or F-score.

Experiment	Precision	Recall	Accuracy	F-Score
DARPA CADETS	0.98	1.0	0.99	0.99
DARPA ClearScope	0.98	1.0	0.98	0.99
DARPA THEIA	1.0	1.0	1.0	1.0

TABLE V: Experimental results of the DARPA datasets.

How do we build datasets?

Experiment	Precision	Recall	Accuracy	F-Score
StreamSpot (baseline)	0.74	N/A	0.66	N/A
$R = 1$	0.51	1.0	0.60	0.68
$R = 3$	0.98	0.93	0.96	0.94

TABLE II: Comparison to StreamSpot on the StreamSpot dataset. We estimate StreamSpot's average accuracy and precision from the figure included in the paper [85], which does not report exact values. They did not report recall or F-score.

Experiment	Precision	Recall	Accuracy	F-Score
DARPA CADETS	0.98	1.0	0.99	0.99
DARPA ClearScope	0.98	1.0	0.98	0.99
DARPA THEIA	1.0	1.0	1.0	1.0

TABLE V: Experimental results of the DARPA datasets.

Experiment	Precision	Recall	Accuracy	F-Score
SC-1	0.85	0.96	0.90	0.90
SC-2	0.75	0.80	0.77	0.78

TABLE VII: Experimental results of the supply-chain APT attack scenarios.

Attack looks like "normal workload"

How do we evaluate provenance-based IDS?

We never got the algorithm to work with SPADE (auditd) data

Experiment	Precision	Recall	Accuracy	F-Score
StreamSpot (baseline)	0.74	N/A	0.66	N/A
$R = 1$	0.51	1.0	0.60	0.68
$R = 3$	0.98	0.93	0.96	0.94

TABLE II: Comparison to StreamSpot on the StreamSpot dataset. We estimate StreamSpot's average accuracy and precision from the figure included in the paper [85], which does not report exact values. They did not report recall or F-score.

Experiment	Precision	Recall	Accuracy	F-Score
DARPA CADETS	0.98	1.0	0.99	0.99
DARPA ClearScope	0.98	1.0	0.98	0.99
DARPA THEIA	1.0	1.0	1.0	1.0

TABLE V: Experimental results of the DARPA datasets.

Experiment	Precision	Recall	Accuracy	F-Score
SC-1	0.85	0.96	0.90	0.90
SC-2	0.75	0.80	0.77	0.78

TABLE VII: Experimental results of the supply-chain APT attack scenarios.

What is provenance in an operating system?

- Represent interactions between system objects
- Represented as a **directed acyclic graph**
  - … or not?
- Information Flows
- **Relationship** between **kernel object states**
  - … or not?
- History of a system execution

What is provenance in an operating system?

- Multiple capture levels
- Share similar syntax…
- … but different semantic
- OPUS
- PASS
- CamFlow
- SPADE (auditd)

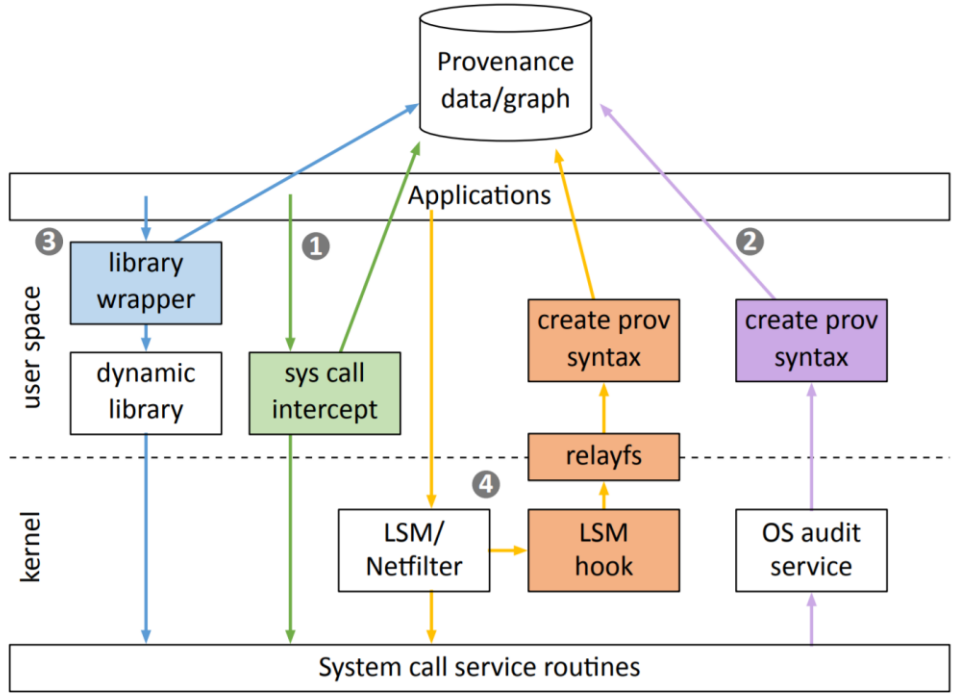


Figure 1 Provenance capture architectures.

Can we analyse provenance at runtime?

- Performance?
  - Pasquier et al., **Runtime Analysis of Whole-System Provenance**, CCS 2018
- Except CamFlow semantic …
- … does not work as well with others
  - cycles
  - ordering properties
  - etc…

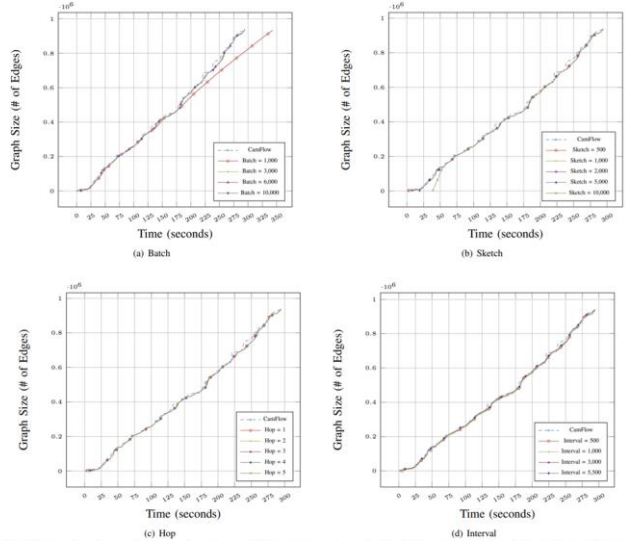


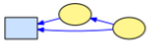
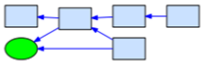
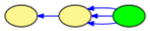
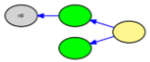
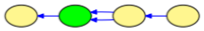
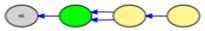
Fig. 4: Total number of processed edges over time (in seconds) in the SC-1 experimental workload with varying batch sizes (4(a)), sketch sizes (4(b)), hop counts (4(c)), and intervals of sketch generation (4(d)). Dashed blue line represents the speed of graph edges streamed into BABAR for analysis. Triangle maroon baseline has the same configurations as those used in our experiment and indicates the values of the controlled parameters (that remain constant) in each figure.

Can we be sure that capture is accurate and complete?

- Chan et al., **ProvMark: A Provenance Expressiveness Benchmarking System**, Middleware 2019
  - **Dynamic** provenance benchmark
  - Compared 3 systems (CamFlow, SPADE (auditd), OPUS)
- Pasquier et al., **Runtime Analysis of Whole-System Provenance**, CCS 2018
  - **Static** analysis of Linux Kernel
  - Generated model for CamFlow
  - Manual verification

Can we be sure that capture is accurate and complete?

- Chan et al., **ProvMark: A Provenance Expressiveness Benchmarking System**, Middleware 2019
  - **Dynamic** provenance benchmark
  - Compared 3 systems (CamFlow, SPADE (auditd), OPUS)

	open	read	write	dup	setuid	setresuid
SPADE				Empty		
OPUS		Empty	Empty			Empty
CamFlow				Empty		

Can we be sure that capture is accurate and complete?

- Chan et al., **ProvMark: A Provenance Expressiveness Benchmarking System**, Middleware 2019
  - **Dynamic** provenance benchmark
  - Compared 3 systems (CamFlow, SPADE (auditd), OPUS)
- Pasquier et al., **Runtime Analysis of Whole-System Provenance**, CCS 2018
  - **Static** analysis of Linux Kernel
  - Generated model for CamFlow
  - Manual verification

Given an event the generated provenance is **function of the application state and the system state** (e.g. open if file exist or not)

The questions we ask ourselves

- What is provenance in an operating system?
- Does provenance-based IDS work?
- Can we interpret the results?
- How do build datasets?
- How do evaluate provenance-based IDS?
- Can we evaluate provenance at runtime?
- Can we be sure that capture is accurate and complete?

The questions we ask ourselves

- What is provenance in an operating system?
- Does provenance-based IDS work?
- Can we interpret the results?
- How do build datasets?
- How do evaluate provenance-based IDS?
- Can we evaluate provenance at runtime?
- Can we be sure that capture is accurate and complete?
- **Capture and detection cannot be considered separately**

University of BRISTOL

Thank you! Questions?

More info online: http://camflow.org

Slides at: http://tfjmp.org

Looking for **collaborations**:
evaluation/datasets

bristol.ac.uk

Questions

- What system calls does CamFlow support?
  - See documentation!
  - https://github.com/CamFlow/camflow-dev/blob/master/docs/COVERAGE.md