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

Thomas Pasquier
11/11/2019, Alan Turing Institute
Host-based intrusion detection
Host-based intrusion detection

System Calls

Identify abnormal patterns
Host-based intrusion detection

System Calls

Identify abnormal patterns

Hidden among benign actions
Host-based intrusion detection

System Calls

Identify abnormal patterns

Hidden among benign actions
Masquerading as benign action
Host-based intrusion detection

System Calls

Identify abnormal patterns
Hidden among benign actions
Masquerading as benign action
Over a long period of time

bristol.ac.uk
Host-based intrusion detection

- System Calls
- Harder and harder to detect pattern
- Identify abnormal patterns
- Hidden among benign actions
- Masquerading as benign action
- Over a long period of time

bristol.ac.uk
What to do?
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

bristol.ac.uk
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

bristol.ac.uk
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?

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>StreamSpot (baseline)</td>
<td>0.74</td>
<td>N/A</td>
<td>0.66</td>
<td>N/A</td>
</tr>
<tr>
<td>$R = 1$</td>
<td>0.51</td>
<td>1.0</td>
<td>0.60</td>
<td>0.68</td>
</tr>
<tr>
<td>$R = 3$</td>
<td>0.98</td>
<td>0.93</td>
<td>0.96</td>
<td>0.94</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARPA CADETS</td>
<td>0.98</td>
<td>1.0</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>DARPA ClearScope</td>
<td>0.98</td>
<td>1.0</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td>DARPA THEIA</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

TABLE V: Experimental results of the DARPA datasets.
How do we build datasets?

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>StreamSpot (baseline)</td>
<td>0.74</td>
<td>N/A</td>
<td>0.66</td>
<td>N/A</td>
</tr>
<tr>
<td>$R = 1$</td>
<td>0.51</td>
<td>1.0</td>
<td>0.60</td>
<td>0.68</td>
</tr>
<tr>
<td>$R = 3$</td>
<td>0.98</td>
<td>0.93</td>
<td>0.96</td>
<td>0.94</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARPA CADETS</td>
<td>0.98</td>
<td>1.0</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>DARPA ClearScope</td>
<td>0.98</td>
<td>1.0</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td>DARPA THEIA</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

TABLE V: Experimental results of the DARPA datasets.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC-1</td>
<td>0.85</td>
<td>0.96</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>SC-2</td>
<td>0.75</td>
<td>0.80</td>
<td>0.77</td>
<td>0.78</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>StreamSpot (baseline)</td>
<td>0.74</td>
<td>N/A</td>
<td>0.66</td>
<td>N/A</td>
</tr>
<tr>
<td>$R = 1$</td>
<td>0.51</td>
<td>1.0</td>
<td>0.60</td>
<td>0.68</td>
</tr>
<tr>
<td>$R = 3$</td>
<td>0.98</td>
<td>0.93</td>
<td>0.96</td>
<td>0.94</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARPA CADETS</td>
<td>0.98</td>
<td>1.0</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>DARPA ClearScope</td>
<td>0.98</td>
<td>1.0</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td>DARPA THEIA</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

TABLE V: Experimental results of the DARPA datasets.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC-1</td>
<td>0.85</td>
<td>0.96</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>SC-2</td>
<td>0.75</td>
<td>0.80</td>
<td>0.77</td>
<td>0.78</td>
</tr>
</tbody>
</table>

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…
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)

<table>
<thead>
<tr>
<th></th>
<th>open</th>
<th>read</th>
<th>write</th>
<th>dup</th>
<th>setuid</th>
<th>setresuid</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPADE</td>
<td><img src="chart1.png" alt="Diagram" /></td>
<td><img src="chart2.png" alt="Diagram" /></td>
<td><img src="chart3.png" alt="Diagram" /></td>
<td>Empty</td>
<td><img src="chart4.png" alt="Diagram" /></td>
<td><img src="chart5.png" alt="Diagram" /></td>
</tr>
<tr>
<td>OPUS</td>
<td><img src="chart6.png" alt="Diagram" /></td>
<td>Empty</td>
<td>Empty</td>
<td><img src="chart7.png" alt="Diagram" /></td>
<td><img src="chart8.png" alt="Diagram" /></td>
<td>Empty</td>
</tr>
<tr>
<td>CamFlow</td>
<td><img src="chart9.png" alt="Diagram" /></td>
<td><img src="chart10.png" alt="Diagram" /></td>
<td><img src="chart11.png" alt="Diagram" /></td>
<td>Empty</td>
<td><img src="chart12.png" alt="Diagram" /></td>
<td><img src="chart13.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>
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**
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