

**PROVENANCE GRAPH GENERATION FOR INTRUSION DETECTION**

**PERI ADHITYAN**

**SUUPERVISOR: A/P KE YIPING KELLY**

**COLLEGE OF COMPUTING AND DATA SCIENCE**

Submitted in Partial Fulfillment of the Requirements for the Bachelor of Computing (Computer Science) Nanyang Technological University

**2025**

Table of Contents

[Abstract 3](#_Toc180660228)

[Acknowledgements 4](#_Toc180660229)

[List of Figures 5](#_Toc180660230)

[List of Tables 6](#_Toc180660231)

[Introduction 7](#_Toc180660232)

[Background 7](#_Toc180660233)

[Objectives 7](#_Toc180660234)

[Literature Review 8](#_Toc180660235)

[Provenance Capture Systems 8](#_Toc180660236)

[Provenance Graph Generation 8](#_Toc180660237)

[CamFlow 9](#_Toc180660238)

[Flurry Framework 10](#_Toc180660239)

[Scenarios 11](#_Toc180660240)

# Abstract

Cyber intrusion has been a growing issue for anyone with a digital footprint from individuals to companies to countries. This gives rise to the dire need for urgent and robust intrusion detection systems to pre-empt and mitigate cyber security incidents before major damage can be done. Data provenance graphs are being researched and utilized in auditing and intrusion detection for cyber security. Provenance graphs can depict the entirety of system execution and assist in gathering information regarding the origin of data, the current state and the entities that acted upon it. This project aims to setup existing provenance capture systems to generate provenance graphs during benign system execution and simulated attack scenarios. Additionally, the generated graphs will be used to train and test models in order to develop intrusion detection systems that provide semantic rich information for real-world application.

# Acknowledgements

I would like to express my gratitude for my final year project supervisor, Associate Professor Ke Yi Ping, Kelly for her mentorship, guidance and assistance throughout this project.

Additionally, I would like to thank PHD Student Mr. Li Wei for taking the time to answer queries and offer advice with regards to this project.

# List of Figures

# List of Tables

# Introduction

## Background

The world we live in today is one that relies on the virtual cyber space as much as it does on the physical space, some might say even more so. But the exponential growth in cyber activity has led to the direct increase in cyber-attacks as well. An increased digital footprint gives way to an array of attack vectors for threat actors to use as an entry point. Year by year we see an increase in cyber-attacks, with 2023 showing a 72% increase in data breaches surpassing the previous record high in 2021 [[Forbes]](https://www.forbes.com/advisor/education/it-and-tech/cybersecurity-statistics/).

To counteract these breaches and detect intrusion, researchers have turned to the usage of a provenance graph-based system. Provenance graphs are directed acyclic graphs used to determine relations between entities such as sockets, files and users, and actions such as the flow of data between them. These give an insight as to what benign activity might look like as compared to when a threat actor has breached the system for malicious intent. Researchers from Harvard and Cambridge Universities have pointed out the extensive capture of security sensitive kernel operations, the explicit relations it depicts between objects, that intrusions result from unexpected relations and the robustness of graphical representation [[Paper]](https://www.seltzer.com/assets/publications/Provenance-Based-Intrusion-Detection.pdf).

Provenance graphs show great advantages when it comes to intrusion detection. Since provenance graphs show system execution by displaying relations between system objects, simple audit files that ae unstructured and hard to read can be converted to provenance graphs. Secondly, provenance graphs are hard for attackers to replicate or forge as they are rich in semantics. They take into consideration spatial and temporal information which allow security analysts to conduct thorough and effective investigations. Finally, provenance graphs store all the execution history which aid analysts in investigating Advanced Persistent Threats (APTs). APTs are known for their long term embedding in systems and stealth in being undetected. The complete history of system execution provided by provenance graphs can easily aid analysts in the event of APTs [[Paper]](https://www.sciencedirect.com/science/article/abs/pii/S0167404821001061).

However, these discoveries and techniques do not come with their downsides. Recent Provenance Graph Based Intrusion Detection Systems (PIDS) used embedding techniques that incur high computational resource cost. Furthermore, as these systems take inputs from graphs, there are detection delays. Finally, these systems output uninterpretable results that do not give much detail other than the fact that they have been flagged out due to the fact that they deviate from normal system operation. [[Paper]](https://arxiv.org/pdf/2404.14720).

## Objectives

This study aims to setup a provenance capture system to capture whole system provenance. It will leverage CamFLow, an open-source project to bring observed provenance collection to the Linux Operating System that complies with W3C PROV-DM standard. These provenance graphs will be generated by implementing the FLURRY framework, an end-to-end data pipeline which simulates cyberattacks captures provenance data from these attacks into data provenance graphs and incorporates this data with a framework for training deep neural models that supports preconfigured or custom-designed models for analysis in real-world resilient systems[[paper]](https://arxiv.org/pdf/2203.02744) Finally, it will utilize these generated graphs to train and test models and gather statistics for benchmarking and further improvement.

# Literature Review

## Provenance Capture Systems

## Provenance Graph Generation

# CamFlow

# Flurry Framework

# Scenarios