NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

School of Computer Science and Engineering
College of Engineering

Provenance Graph Generation

For Intrusion Detection

Student: Chew Jie Ying Perlyn Supervisor: Associate Professor Ke Yiping, Kelly

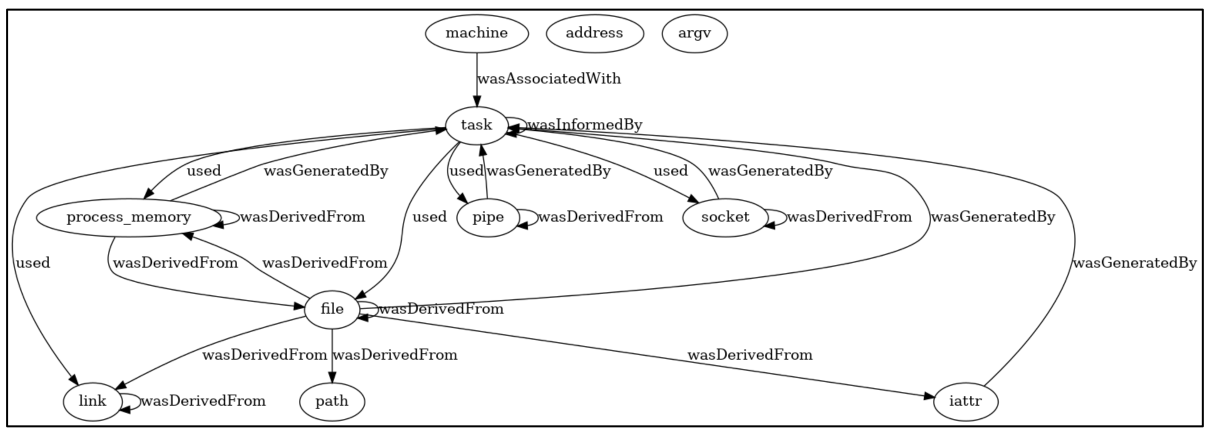
Project Objectives

With the increase in complexity of cyberattacks, traditional intrusion detection systems are struggling to identify sophisticated threats such as zero-day attacks or Advanced Persistent Threats (APTs). Provenance graphs emerge as a promising data source for modern intrusion detection by capturing comprehensive information on both malicious and benign system activities. These graphs present complex dependencies and relationships in the form of a directed acyclic graph that has potential for analysis using machine learning methods.

Recognising the potential of provenance graphs, this research implements Flurry, an end-to-end Framework built upon CamFlow, to improve the generation and capture of provenance graphs for intrusion detection. Intrusion scenarios will be designed then simulated on multiple security- sensitive applications across various operating systems. Extensive datasets of provenance graphs were produced via dynamically executing various attacks on Fedora and Ubuntu, then used to train and validate state-of-the-art graph-based models, to evaluate their effectiveness and accuracy. Specifically, a Graph Convolution Network (GCN) was implemented to perform graph classifications on provenance graphs.

Provenance Graphs

Provenance is generally defined as a "record that describes how entities, activities, and agents have influenced a piece of data". Specifically, in the context of intrusion detection, Provenance describes the history or lineage of an object that explicitly represents the dependency relationship between the damaged files and the intrusion processes, rather than the underlying system calls, to detect and analyse intrusions



Provenance Graph from the execution of a Brute Force Attack

Methodology

Generation of Graphs through Dynamic Execution of Attacks via DWVA

Hyperparameter Tuning of GCN model

		Size of Hidden Dimension	
		128 Nodes	256 Nodes
Number of Layers	1 Layer	accuracy: 0.655 precision: 0.524 recall: 0.743 F1 macro: 0.651	accuracy: 0.66 precision: 0.581 recall: 0.718 F1 macro: 0.659
	2 Layer	accuracy: 0.67 precision: 0.562 recall: 0.747 F1 macro: 0.668	accuracy: 0.69 precision: 0.590 recall: 0.765 F1 macro: 0.688
	3 Layer	accuracy: 0.685 precision: 0.590 recall: 0.756 F1 macro: 0.684	accuracy: 0.695 precision: 0.619 recall: 0.756 F1 macro: 0.694

Results and Conclusion

Statistics of Graph Datasets Generated

	Scenario Type	Number of Nodes	Number of Edges	Number of Relation Types	Accuracy
Brute-Force	Benign Attack	2110 4006	5289 11032	20 21	99.75
CL-Injection	Benign Attack	3164 1196	8195 2775	20 20	84.63
SQL-Injection	Benign Attack	1257 1031	3103 2294	19 18	54.75
XSS-DOM	Benign Attack	196 1295	462 2995	18 20	72.83
XSS-Reflected	Benign Attack	4610 3643	13828 9640	19 21	78.58
XSS-Stored	Benign Attack	2811 8847	7120 23433	22 23	98.75

Results of a 3-layer GCN with Hidden Dimension 256

	Accuracy	Precision	Recall	F1	False Alarm Rate
Brute-Force	99.75 ± 0.25	99.49 ± 0.50	100.0 ± 0.00	99.75 ± 0.25	0.49 ± 0.50
CL-Injection	84.63 ± 2.30	83.01 ± 4.25	85.81 ± 7.95	84.17 ± 1.65	15.88 ± 2.68
SQL-Injection	54.75 ± 1.75	62.72 ± 15.78	53.63 ± 0.62	57.20 ± 6.68	42.99 ± 3.73
XSS-DOM	72.83 ± 1.38	71.24 ± 4.12	74.25 ± 2.96	72.60 ± 0.70	27.98 ± 4.52
XSS-Reflected	78.58 ± 1.63	79.73 ± 0.21	77.81 ± 3.35	78.65 ± 1.63	20.58 ± 0.36
XSS-Stored	98.75 ± 0.87	98.47 ± 1.82	99.00 ± 0.07	98.73 ± 0.94	1.45 ±1.61

The effectiveness of GCN can be seen in performing graph classification on provenance graphs datasets. The strong performance of state-of-the-art graph models in anomaly detection exemplifies the potential of provenance graphs as an ideal data source for modern intrusion detection systems.