MagProv: Enhancing Fidelity of System Provenance via Asynchronous Taint Tracking

Hidden

Abstract—**Growing concerns over privacy issues in server-based social networks challenge the centralized architecture where users' personal data are concentrated at the service provider. We propose MAGPROV, a decentralized infrastructure to host a private, in-browser social network. Users are able to reach MAGPROV just with the browser, free from any kind of installation effort, then embrace the privacy brought by the purely decentralized architecture. For hosting a decentralized, yet usable and scalable, social network, specifically we design, implement and evaluate 1) a signaling and friendship exchange service that offers end-to-end data privacy/security and light-weight deployment with no computing requirement at all; 2) a client only logic including a flooding-based post delivery overlay network with which messages can reach both online and offline users with low latency. Evaluation results show that with minimal infrastructure deployment and flexible bootstrapping methods, MAGPROV is able to effectively accommodate a social network with several millions of concurrent users.**

Introduction

introcutions.

Provenance Deficiency

show the deficient provenance due to syscall level capturing. userspace tracking is also needed. But the performance is too bad.

have one simple example.

Overview

Overview of our system

Replay-able Logging

Explain the logging at runtime. explain the whole system logging (hooking syscalls, and particularly execve for whole system recording, maybe whitelisting.)

Provenance Construction Stage I

Explaint the coarse grained provenance construction. this can be brief.

Provenance Refinement

The provenance is refined by taint tracking. We first identify the source and sinks. This is basically to find the address space interleaving between two processes.

Source/Sink Identification

Taint Tracking

file to file, socket to file, file/socket to eip.

Implementation

Implementations.

Evaluation

Evaluation focus on the performance numbers: runtime overhead, taint overhead.

User Scenario

We have an APT running example to show the refined provenance. Have a chart to show the amount of enriched provenance. (number of nodes/edges)

Discussions

Limitations.1) side channel; 2) data race;

Related Work

1. Provenance works, protracer, lpm, recprov... 2. Record and replay, es, panda, vmware, kvm 3. Taint tracking.

Conclusion

conclude.

References