L-Università ta' Malta | Research Expo 2024

Christian Colombo

Department of Computer Science

Managing Cybersecurity of Software Monitors

Project brief

Aims & objectives

While software monitors can help assure that sensitive software systems are safe, one still needs to prepare for different scenarios of compromise, including those where the system and its security monitors are incrementally taken over by attackers:

1. *The monitor is run in a container, as separate as possible from the rest of the system.*
2. *Monitoring logs are stored as a tamper-evident file where data authenticity is verifiable.*

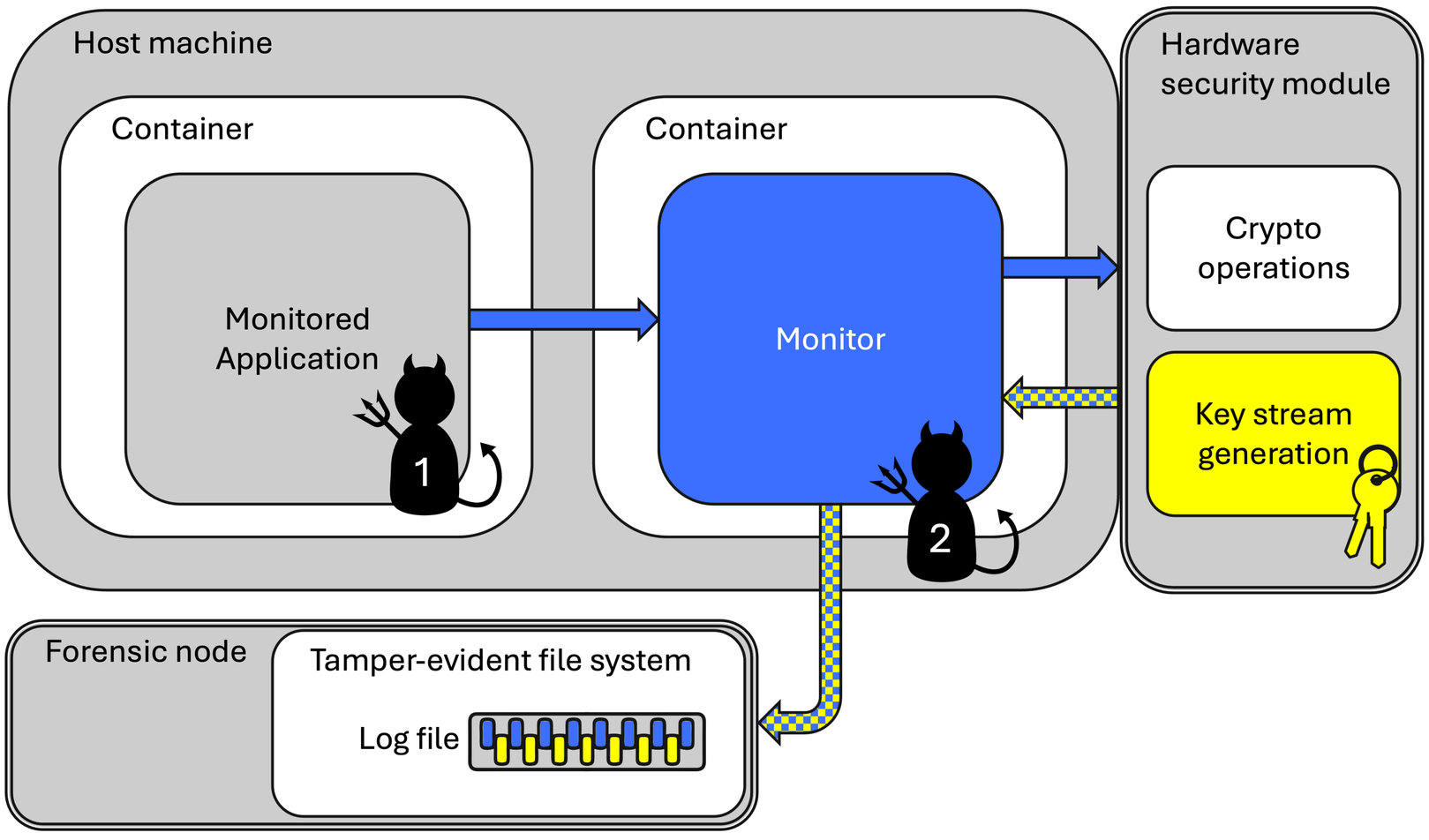
1. Monitor Execution Isolation

Making it harder for attackers to reach

Keeping the monitor running in isolation from the monitored application protects it if the application is taken over by malicious actors (**Attack scenario 1**).

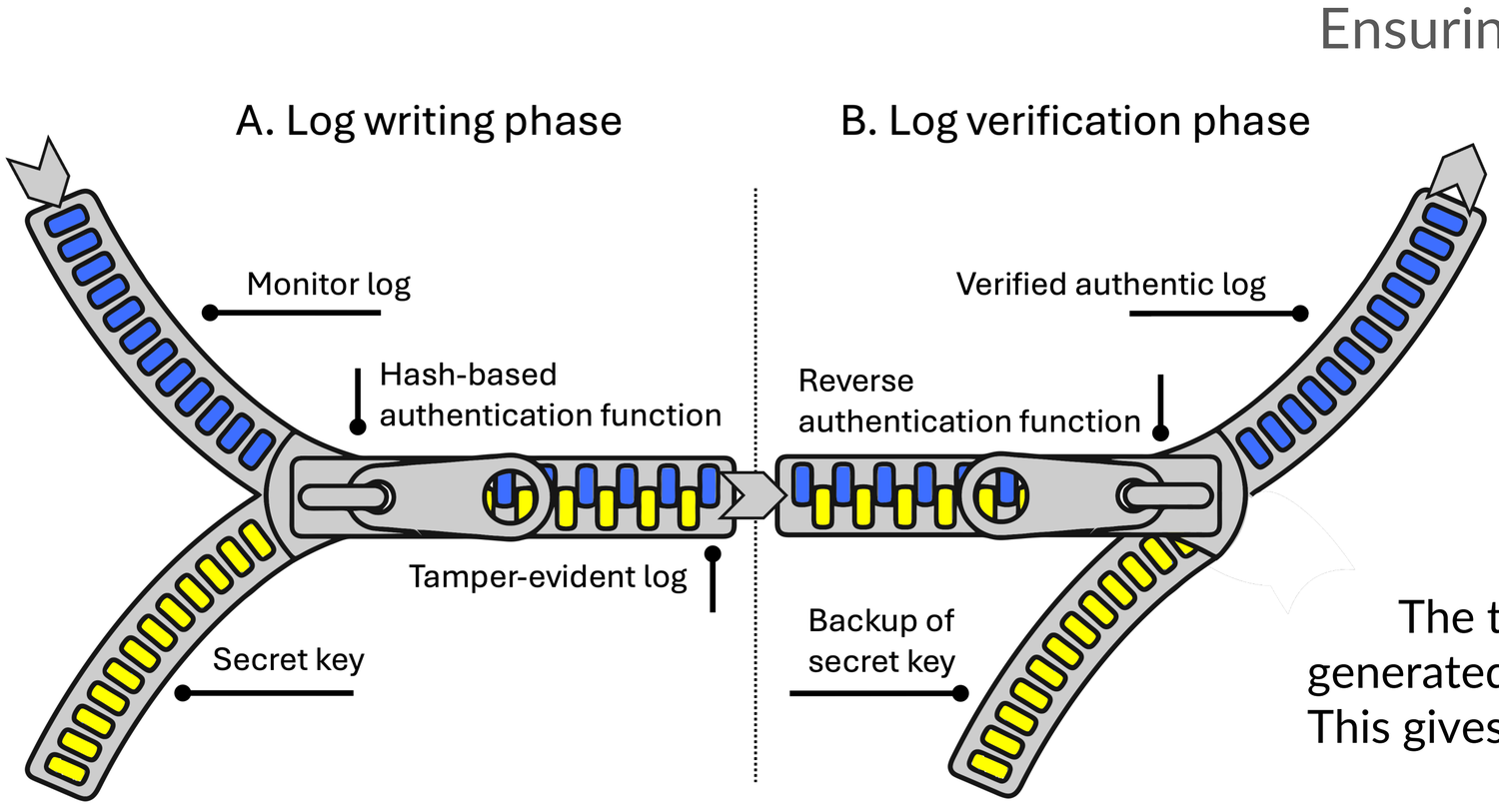
At the same time, this brings challenges with it due to extra layers of communication and reduced visibility of internal operations.

Yet, if attackers manage to gain full control of the host machine, there is little that can be done to stop them from corrupting the monitoring system as well (**Attack scenario 2**). Another layer of protection is needed…



2. Tamper-Evident Log Data

Ensuring evidence collected is authentic



We store the monitoring system output within a tamper-evident file system on a forensic node (**Phase A**). If all else fails and the attacker modifies the data, our file system is still able to distinguish between authentic and tampered monitor output.

How is this even possible?
The trick is to use cryptography: A secret key is generated and intertwined with the monitor output. This gives us the ability to audit the monitoring data for authenticity at a later stage (**Phase B**).