RUNTIME HOST-BASED INTRUSION/ANOMALY DETECTION FOR SAFETY-CRITICAL SYSTEMS

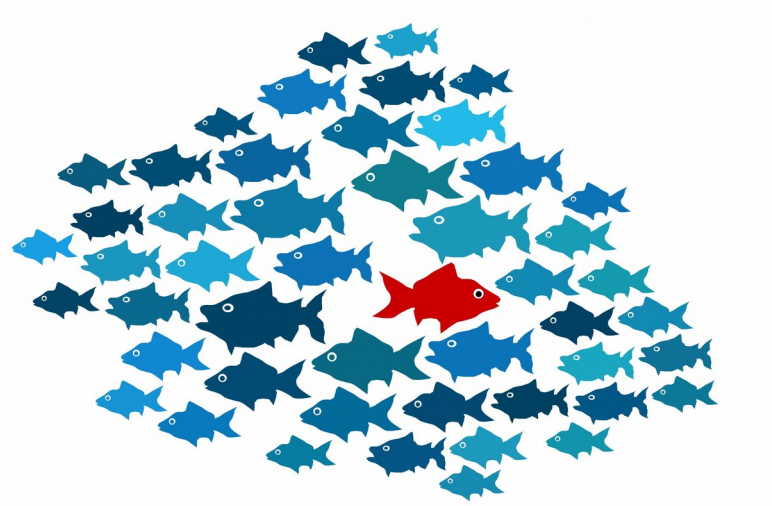
October 13, 2015

MOTIVATION

- Security “walls” have proven their inefficiency (Crypto, Firewalls, policies, etc.)
- Systems are a complex aggregate of interconnected systems. Integrators receive black box components from manufacturers.
- Human error (Incorrect/non-existent security policies, incomplete software requirement gathering, implementation and verification)
- Attack surfaces are ever changing.
- Zero-day attacks on safety-critical systems can be catastrophic. Traditional security mechanisms are not enough.
- Anomaly detection is intended to augment other security measures.

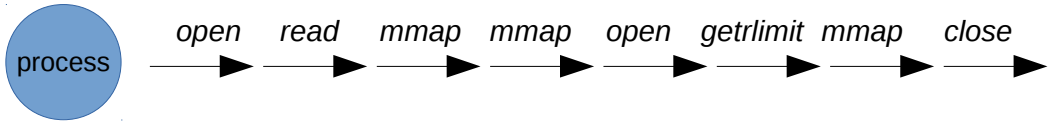
ANOMALY DETECTION – THREE MAIN COMPONENTS

- Defining normal by building a database/profile of normal behavior (training). This is done during testing and/or for a preset duration in the field.
- Monitoring behavior at runtime and detecting anomalous patterns. All abnormal behavior is detected including legal abnormalities such as lack of disk space.
- Reacting if necessary.



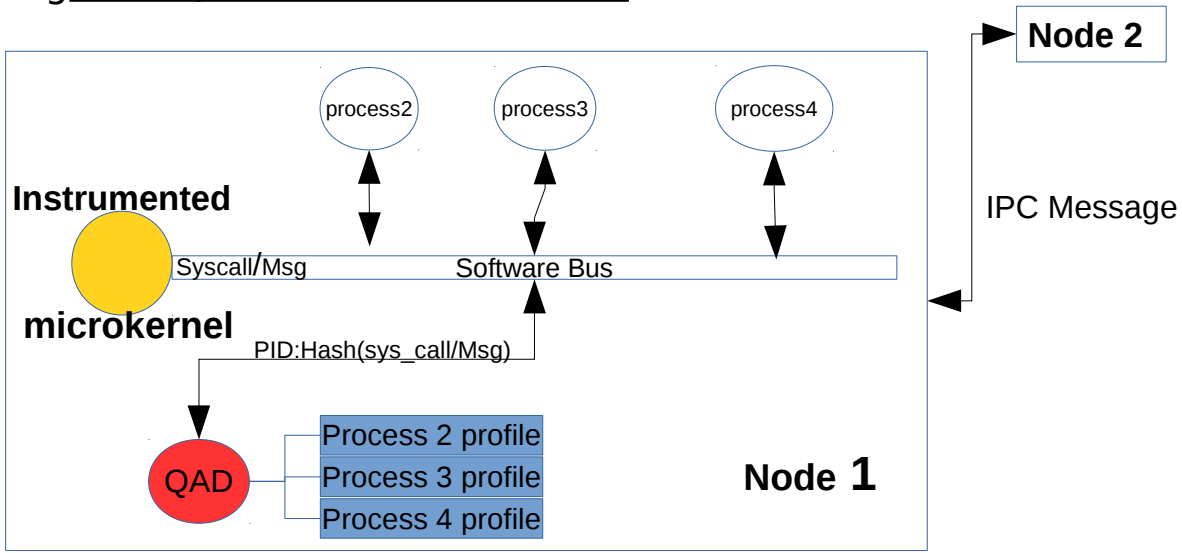
USING SYSTEM CALLS AS A BEHAVIORAL MODEL

- A system process is treated as a black box that emits signals and can be watched
- Evidence suggests that short-sequences of system calls are a good discriminator between abnormal and normal behavior of a process.
- Unique sequences of system calls generated by a process generate a stable signature of normal behavior
- The signature has low variance over a wide range of normal operating conditions and has a high probability of being perturbed when abnormal activities occur
- Very simple and light weight, suitable for an on-line, runtime IDS.
- Profiles are distributed and independent. Each process has its own profile.
- No change on the development side is required. It is completely transparent.



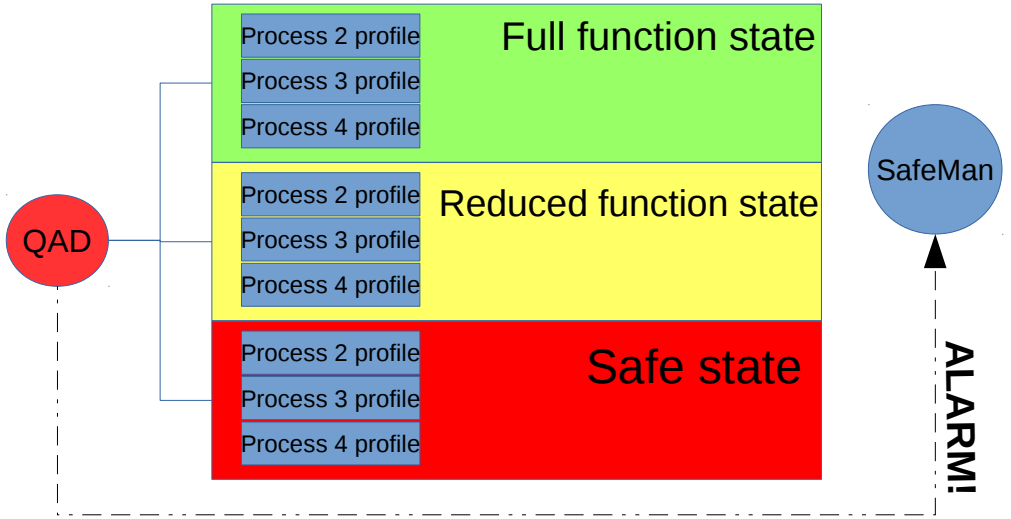
QNX SYSTEM CALLS

- Are mostly message sends (uniquely identified by PID sender,PID receiver, CHID, NID, Message head). The normal profile is build using their hash.
- This allows for detecting anomaly not just for “system calls” going to the kernel, but also anomaly over interprocess messaging.
- This includes messages going over QNET to a remote node.



REACTION FRAMEWORK

- Safety specialists/customers decide on reaction to an anomalous event. Process ID and anomaly threshold are parameters they could use to decide on appropriate action.
- A transition between safe states might be appropriate. Better safe than sorry!
- Processes' behaviors are profiled within different safety level contexts



THANK YOU.