Data Protection Compliant Real Time Anomaly Detection for End-to-End Encrypted Infrastructure

The obvious challenge of anomaly detection in complex systems is the question that comes up immediately: What is “normal”? After all, that is what defines an anomaly: not being normal. Often the answer seems obvious – for humans. Take the case of a municipal utility’s network being controlled from the accountant’s cubical. Humans notice that something is wrong here. Computer networks have a much harder time figuring that out, since if this control connection is not wanted, why was it configured or allowed in the first place? Are you not happy to have control?[1]

At the root of this discrepancy lies a knowledge-gap – and for a change that’s one where the human has the advantage over the networked computer. Humans know what trust relations to expect. We intuitively know that trusting accounting with operating the city’s power grid is not normal, nor is the alternative of the power people using payroll office computers. That’s an anomaly, and usually bad news.

Computer systems and networks, however, never had the means to express such trust relations. Locked up in a small box, the only outside contact being a few sensors and a network protocol that only allows to say “I’m hot!”[2] means that every piece has to fend for itself. Teamwork was neither planned nor paid for and situational awareness is something not even available to the brass. Sounds familiar?

Trinity Technology allows systems to trust each other exactly as far as those individuals responsible for them would trust each other, starting with: “Trust? For what?”, i.e. being specific. Occasionally, one may also verify whether that foundation the trust is built on still validates, or what a third party claiming to know about misbehavior or malice has to say – basically finding out if trust has been misused.

Trinity employs a cryptographically reinforced peer-to-peer network allowing systems to publish precisely defined trust to one another, or to withdraw it, in real time, just like people do. Multiple copies of the respective data are kept within the network, allowing the creation of cross-checked environments without the need for sensors.

Firewall and IDS/IPS rules attempt to model trust using a hierarchical static description of trivially spoofed indicators, not to mention their cost of operation. Humans don’t do that. Like with some personal firewalls, we decide once if an interaction should take place. If we change our mind or if the interaction took place regardless of what we decided, our trust model needs be adjusted accordingly and the argument enforced a bit more, may this be on software or a web site.

This is Trinity Technology.

In the context of anomaly detection, it means that finally having a good answer to the question of “What is normal?” Normal is any communication that has a corresponding trust relationship. Anything else is either communication that is not trusted, or a trust not intended to be there in the first place (i.e. nobody wants to be responsible). All of it can be verified in real time, in parallel, and by as many nodes as needed.

Trusted communication does not need to be inspected, so it should be encrypted end-to-end. For all other communication, the need for interception can be shown, e.g. in order to obtain a search warrant, which will hence be similarly context dependent.

Trinity Technology provides classification and detection of communication anomalies.

[1] Which by itself is a valid argument in the age Windows 2 (10b) nag screens, iWalledGardens and symbioted infested Linux.
[2] The protocol would, however, require accepting software modifications received and demand their immediate application to permanent storage, without any records (regulatory reasons), or cryptographic protections (usability reasons).