Evidence collection for critical infrastructure

Michael Sonntag

Institute for Information Processing and Microprocessor Technology (FIM)

Johannes Kepler University Linz, Austria

michael.sonntag@jku.at

Attacks on critical infrastructure

- First priority: Prevent them!
- But: Can we be sure that we will always succeed in this?
- Example (no final report available yet):
  - Austrian power grid almost collapsed in May 2013, which would probably have led to a European-wide blackout
    - Reason: A command for reading natural gas meters “escaped” into the electricity system, causing an avalanche of responses/replies/error messages
    - Regulating the grid was done by phone (and sending persons to power stations to regulate them manually if necessary, e.g. on the danube)

Therefore: Collect evidence now to be later able to detect source of problem and how it expanded, was it an attack, who was culprit, plug hole, …

Tracer boxes

- Basic idea: Black box recorders of airplanes
  - Copy **all** data (not: selected, extract, …) and retain it for later analysis
  - If the storage is full, delete oldest elements (ringbuffer; always almost full)
  - Numerous cheap appliances distributed through the network
  - No analysis of data, no passing it on to central stations
- Two kinds of loggers: For the network and for systems
  - Network: Copy all network data passing in either direction
    - Do not participate in network, just monitor (taps, mirroring ports etc.)
  - Systems: Store logging data
    - Separate from the (potentially hacked) system, no load for storing it
    - One-way connection: No feedback to system

Requirements for tracer boxes (1)

- Completely passive: If a system is hacked, the attacker can modify logs to keep his traces hidden. Completely passive → No attack possible!
  - Drawback: No commands possible
  - Solution: Separate network for commands. This can be tiny/trivial and secure, e.g. with a public key in the box and all commands must be signed by the private key
    - Examples: Serial lines, separate network, mobile communication
- No data analysis: Keep boxes simple & cheap; no management/updates needed
  - Also prevents DoS attacks or hacking through malformed content
  - Universally usable with little configuration
  - Drawback: When to stop recording?
  - Solution: See command channel above; physical switch on appliance

Requirements for tracer boxes (2)

- Rolling data storage: Data should be kept as long as possible
  - E.g. 100 Mbit/s ≈ 1 TB/day → Usable with large disks (50% usage; duplex; 100MB); industrial control systems (SCADA) have typically much lower communication rates!
- Stopping recording: Problem! No analysis → External command only
  - See above: Separate network and/or physical switch; depends on storage time too
- Liveness check: Some notification that the system is still working (and has not been stopped) is necessary
  - Separate communication network with send-only functionality (duplex → commands)
    - Requires some secret data in tracer box to prevent replay attacks
  - Potentially allows mapping attacks (=locating critical infrastructure)
    - Mobile phone → Difficult; reliability if critical infrastructure (e.g. phones!) fails?

Requirements for tracer boxes (3)

- No remote updates: Updates to the software should not be possible remotely
  - The “incoming” direction on the control channel must be as small and trivial as possible to prevent attacks (signing might not help here, e.g. copy&stop + replay during an attack)
- Encryption: Data should only be stored encrypted
  - If someone steals the box, the data cannot be used to gain any information
  - Private key is located at a central system which performs an analysis (if it is needed)
    - Physically gathering the box and replacing it with a new one (or disk only)
- Physical properties: Simple, cheap, robust, secure
  - No moving elements (but: harddisk!), completely enclosed (cooling!), steel casing (water, acid…; breaking it open; destroying it), securely attached to monitored system

Potential problems

- Different protocols: Many systems use Ethernet, but SCADA often use other busses/protocols (only an adapter problem - no content inspection!)
- Not usable for normal monitoring: Too dangerous to get hacked, overloaded, …
- No direct commercial benefit: But when a problem/bug occurs, even a single one could costs much more than many of these tracer boxes!
- Encrypted communication: Today still rare for industrial control systems, but this will definitely change in the future.
  - Potential solution: Decryption box in front of network tracer box, encryption reversible with a master key (requires new protocols; e.g. not with TLS!), no encryption (e.g. within a factory; separate network for control systems)
- Liveness signal: Separate communication network needed; management of numerous tracer boxes
- System tracer boxes: System must at least generate numerous log messages
  - Even this (not storing them, just sending them out on a separate interface) might be a lot of work and potentially too much for simple/weak systems

Communication with tracer boxes

- Tracer boxes can be mass-produced, but need at least some individualization
  - Unique identification of the tracer box (could be a serial number)
    - To be included in any stop command/liveness signal
- Keys required for such a setup on the tracer box:
  - Tracer box individual private key 1: To sign “alive” signals
    - Registry of public keys of all tracer boxes at the central system
  - Tracer box individual public key 2: To encrypt the collected data
    - Registry of private keys of all tracer boxes at the central system
    - This could also be the same public key for all tracer boxes
  - Public key of central system: To verify signature of command for stopping
    - Could also be an individual key for each tracer box (replay!)

Current state of implementation

- Worked on by a student as part of his thesis
- Demonstration only, not a full product
- Implementation details:
  - Raspberry mini-computer with Linux
  - External USB hard disk for storage
  - Small additional hardware for stop-switch, and “self-kill” switch based on command
  - Separate command/alive network by USB2Ethernet converter
    - Mobile phone/WLAN would also be easily possible
  - Daemons: Monitoring network as small log files; deleting old files until enough space
    - Problem: Not losing data on change to new log file; tools only support change by time (suboptimal when traffic is widely differing)

Conclusions

- Small and cheap tracer box allows gathering data for later forensic investigations
  - Specific production would be simple and cheap (waterproof & explosion protected → all locations including open air, little power required, simple to securely attach)
  - Very little customization needed (choice of second interface; individualization)
- Little possibility of misuse because of encryption and geographic distribution
- Will not help in evaluation, only provides the basis to do it
- Current main problem: SSD are too expensive and small, hard disks too fragile
  - Ideally they should store several weeks of data, so physical retrieval would be easy
- Could easily be prescribed by law, similar as in airplanes

Thank you for your attention!

Michael Sonntag

Institute for Information Processing and Microprocessor Technology (FIM)

Johannes Kepler University Linz, Austria

michael.sonntag@jku.at