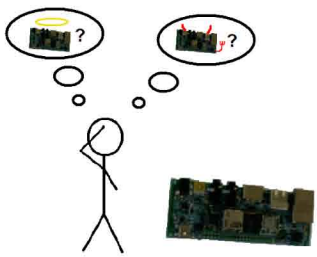
TCIPG

Trustworthiness Enhancement Tools for SCADA Software and Platforms

Sergey Bratus, James Oakley, Jason Reeves, and Sean W. Smith

Goals

- Develop tools to help maintain the *trustworthiness* of devices operating within the power grid.
- Make these tools flexible enough to support the wide range of systems present within the grid.
- Ensure that any tools we design are mindful of the constraints of both the power industry and embedded devices in general, and do not keep the devices they protect from performing their primary duties.



Fundamental Questions/Challenges

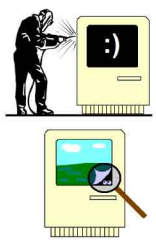
- What constraints are present for embedded devices in the power grid?
  - Resource restrictions.
    - Less memory, slower processors, etc.
  - Task requirements.
    - Ex. exchanging data within small time windows [3].
  - Industry demands.
    - Availability reigns supreme.
- *How do we build security tools to accommodate these?*

The "Standard" Approach To Security

- *"Thou shalt apply patches to your systems on a regular basis."*
  - ...except that we can't just pull an embedded power system out of service to patch/reboot it...
  - ...and installing these patches may break legacy-but-mission-critical applications that are vital to the system's operation.
- *"Thou shalt employ an intrusion detection system (IDS) to monitor your system for suspicious behaviors."*
  - ...except that many current intrusion detection solutions are based on using *virtualization*, which imposes too much overhead on embedded devices to be used within the power grid.

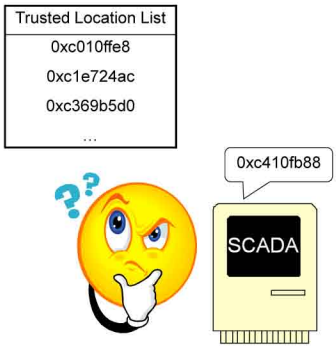
A Different Security Paradigm

- Protecting devices in the power grid requires us to reconsider our approach to security.
- What if we could build...
  - A patching system that could keep system software up-to-date without taking the system offline?
  - An IDS that could operate without a hypervisor, and instead leverage certain pieces of the system to protect itself?



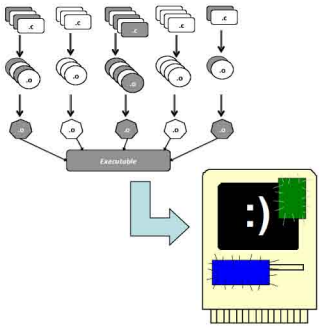
Project #1: Autoscopy Jr.

- **Lightweight IDS for resource-constrained embedded control systems.**
- Protects against control-flow alterations caused by malware.
- Operates in-kernel (no hypervisor required).
- Uses Kprobes [4] to monitor indirectly called functions.
- Flexible mediation scope.
- Imposed less than 5% overhead in testing on a non-embedded system (after profiling).



Project #2: Katana

- **Hot-patching system for critical systems that must be constantly available.**
- Allows patching without rebooting or losing state.
- Leverages information in the DWARF [1] debugging framework.
- Provides a basis for reasoning about the consequences of applying a patch.



Current Status

- **Katana:** Prototype has been released (http://nongnu.org/katana/).
  - Project currently dormant (due to graduation of project lead).
- **Autoscopy Jr.:** Program tested on non-embedded, non-power systems.
  - Currently working with Schweitzer Labs to test on actual power hardware (and potentially incorporate it into their product line).

Future Efforts

- Identify a new project lead for Katana.
- Continue our partnership with Schweitzer, and transition Autoscopy Jr. from a prototype into a production-quality tool.
- Attempt to integrate Autoscopy Jr. with only in-kernel security solutions, such as grsecurity [2].

For More Information

- **Katana:** http://nongnu.org/katana/
- **Autoscopy Jr.:** http://www.cs.dartmouth.edu/reports/TR2011-704-rev1.pdf

References

1. The DWARF Debugging Standard. The DWARF Standards Committee. *http://dwarfstd.org/.*
2. grsecurity. Open Source Security, Inc. *http://grsecurity.net/.*
3. IEEE standard communication delivery time performance requirements for electric power substation automation. IEEE Standard 1646-2004. *http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1405811.*
4. Jim Keniston, Prasanna S. Panchamukhi, and Masami Hiramatsu. Kernel probes (Kprobes). The Linux Kernel Archives. *http://www.kernel.org/doc/Documentation/kprobes.txt.*

All clipart from Microsoft Corporation.