Available online at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/ijcip

# Intrusion detection for resource-constrained embedded control systems in the power grid

*Jason Reeves*[a,*], *Ashwin Ramaswamy*[a], *Michael Locasto*[b], *Sergey Bratus*[a], *Sean Smith*[a]

[a] *Department of Computer Science, Dartmouth College, Hanover, New Hampshire 03755, USA*
[b] *Department of Computer Science, University of Calgary, Calgary, Alberta T2N 1N4, Canada*

ARTICLE INFO

*Article history:*
Received 16 April 2011
Accepted 25 January 2012
Published online 10 February 2012

*Keywords:*
Embedded control systems
Power grid
Intrusion detection

ABSTRACT

The power grid depends on embedded control systems or SCADA systems to function properly. Securing these systems presents unique challenges—in addition to the resource restrictions inherent to embedded devices, SCADA systems must accommodate strict timing requirements that are non-negotiable, and their massive scale greatly amplifies costs such as power consumption. Together, these constraints make the conventional approach to host intrusion detection – using a hypervisor to create a safe environment from which a monitoring entity can operate – too costly or impractical for embedded control systems in the critical infrastructure.

This paper discusses the design and implementation of Autoscopy, an experimental host-based intrusion detection mechanism that operates from within the kernel and leverages its built-in tracing framework to identify control-flow anomalies, which are most often caused by rootkits that hijack kernel hooks. The paper presents the concepts underlying the original Autoscopy prototype, highlights some of the issues that arose from it, and introduces the new system, dubbed Autoscopy Jr., which addresses the issues. Tests on non-embedded systems demonstrated that the monitoring scope could be managed to limit Autoscopy Jr.'s performance impact on its host to under 5%. The paper also describes the use of an optimized probe framework to reduce overhead and the test results obtained for a hardened kernel. The results demonstrate that Autoscopy Jr.'s design and effectiveness render it uniquely suited to intrusion detection for SCADA systems.



## 1. Introduction

The world's critical infrastructure has become increasingly dependent on embedded control systems—computers implanted in larger devices to serve as controllers and perform many of their important tasks. The power grid has not been immune from this trend. One study [1] predicts that the number of smart electric meters deployed worldwide – and by extension the embedded control systems inside these meters – will increase from 76 million in 2009 to roughly 212 million by 2014.

The need to secure software that expresses complex process logic is well understood, and is particularly important for devices operating as part of a SCADA system, where this logic applies to the control of potentially hazardous physical processes such as power generation. Any failure to secure these important devices can have grave consequences, as demonstrated by Stuxnet [2]. As a general exploit alone, Stuxnet's credentials are frighteningly

* *Corresponding author.*
E-mail address: Jason.O.Reeves.GR@dartmouth.edu (J. Reeves).

1874-5482/$ - see front matter 
doi:10.1016/j.ijcip.2012.02.002

impressive. The program attempted to subvert targets using four zero-day vulnerabilities and two compromised digital certificates, and incorporated rootkit functionality that enabled it to hide its behavior. However, rather than attacking systems indiscriminately, Stuxnet specifically targeted devices within an industrial control system. In particular, the program looked for Windows computers used to configure programmable logic controllers that operate uranium-enriching centrifuges [3]. Because industrial control systems are often found within critical facilities such as power plants, the consequences of such sabotage could be severe and potentially life-threatening. Therefore, ensuring the integrity of these devices and others within the critical infrastructure is essential.

A number of malware protection proposals (see, e.g., [4–9]) address the issue of device integrity using virtualization, relying on a hypervisor to create a trusted space to use for monitoring the potentially-compromised system. These proposals, however, fail to account for some key attributes of embedded control systems used in the power grid:

- The space and storage constraints of embedded devices may render the use of a hypervisor impractical. For example, Petroni and Hicks [7] found that running the Xen hypervisor on a test platform (a laptop with a 2 GHz dual-core processor and 1.5 GB RAM) imposes an overhead of nearly 40%.
- Embedded systems in the power grid must deal with strict application timing requirements, some of which require a message delivery time of no more than 2 ms [10].
- The extra costs associated with security computations (i.e., computations performed solely to achieve device security goals) do not scale well in a power grid environment. For example, LeMay and Gunter [11] note that, in a planned rollout of 5.3 million electric meters, including a trusted platform module (TPM) with each device would incur an added power cost of more than 490,000 kWh per year, even assuming that the TPMs sit idle at all times.

On the whole, the collective price (in terms of maintenance, patching, energy, etc.) [12] of hypervisor-based approaches obviates their use in industrial control environments. However, this conclusion leaves the door open for non-virtualized alternatives. A particularly promising approach is to use a kernel protection mechanism that resides at the same privilege level as the kernel to defend against malware. Such approaches have proven to be effective in the past. For example, kernel hardening efforts (e.g., grsecurity/PaX [13] and OpenWall [14]) that implement a variety of security mechanisms in the code of the Linux kernel itself (by creatively leveraging the MMU hardware of x86 and other architectures and the ELF binary format features) are successful at reducing the kernel attack surface without resorting to a separate implementation of a formal reference monitor.

This paper describes the Autoscopy system, which we developed as a prototype in-kernel intrusion detection mechanism [15] and recently refined to protect embedded control systems [16]. Instead of being separated from its host via a hypervisor, Autoscopy demonstrates the possibilities of an intrusion detection system working inside the operating system kernel to reduce the overhead on its host [15]. To do so, the system leverages Kprobes [17,18], a tracing framework included in the Linux kernel, to place probes in indirectly-called functions within the kernel to dynamically monitor the control flow of running programs for anomalies [15].

In tests run on a standard laptop system, Autoscopy was able to detect every one of the published control-flow hooking rootkit techniques it was tested against, while imposing an overhead of 5% or less on a wide range of performance benchmarks [15]. Our second iteration of the program, dubbed Autoscopy Jr., includes a system profiler, which permits the location and removal of "heavy" probes that generate too much overhead, helping balance security with performance and allowing the customization of the mediation scope to keep the overhead under the 5% limit [16]. These results indicate that, unlike virtualized intrusion detection solutions, Autoscopy's design and performance make it well-suited to the task of protecting embedded control devices, including those that are used within the critical infrastructure.

## 2. Background

This section discusses embedded systems in the power grid, frames the debate between virtualized and in-kernel security solutions, and introduces the tracing framework used by Autoscopy for monitoring its host. The section also provides details about one of the more successful projects in the area of kernel hardening.

### 2.1. *Embedded control systems in the power grid*

The electrical grid contains a variety of intelligent electronic devices (IEDs), including transformers, relays and remote terminal units. The capabilities of these devices can vary widely. For example, the ACE3600 RTU sports a 200 MHz PowerPC-based processor and runs a VX-based real-time operating system [19], while the SEL-3354 computing platform has an option for a 1.6 GHz processor based on the ×86 architecture and can support operating systems such as Windows XP or Linux [20].

In addition to the issues that arise from restricted resources, embedded control systems in the power grid are often subject to strict timing requirements when passing data in a network. For example, IEDs within a substation require a message delivery time of less then 2 ms to stream transformer analog sampled data, and must be able to exchange event notification information for protection within 10 ms [10]. Given these small timing windows, introducing even a small amount of overhead could affect a device so that it cannot meet its message latency requirements, prohibiting it from performing its task—an outcome that may well be worse than a malware infection. Therefore, it is vital to limit the amount of overhead imposed on a device, especially as its availability takes precedence over its security.

Another important issue is the evolution of the technologies used to power critical embedded systems. While companies have historically used customized proprietary products in SCADA and other critical systems, the current trend is to deploy commercial off-the-shelf (COTS) products, including operating systems, applications and communication protocols [21]. (The COTS trend was confirmed in a conversation