Chapter 16

# REMOTE FORENSIC ANALYSIS OF PROCESS CONTROL SYSTEMS

Regis Friend Cassidy, Adrian Chavez, Jason Trent and Jorge Urrea

**Abstract** Forensic analysis can help maintain the security of process control systems: identifying the root cause of a system compromise or failure is useful for mitigating current and future threats. However, forensic analysis of control systems is complicated by three factors. First, live analysis must not impact the performance and functionality of a control system. Second, the analysis should be performed remotely as control systems are typically positioned in widely dispersed locations. Third, forensic techniques and tools must accommodate proprietary or specialized control system hardware, software, applications and protocols.

This paper explores the use of a popular digital forensic tool, EnCase Enterprise, for conducting remote forensic examinations of process control systems. Test results in a laboratory-scale environment demonstrate the feasibility of conducting remote forensic analyses on live control systems.

**Keywords:** Process control systems, digital forensics, live forensics, EnCase

## 1. Introduction

The personal computer that sits on a desk at work or at home is similar to the systems that are used to operate many critical infrastructure components. Power plants, oil and gas pipelines, and other large infrastructures that once mainly employed legacy systems with proprietary technologies have adopted commodity computer systems, software and networking technologies, including Internet connectivity.

While commodity computer systems and the Internet are efficient, cost effective solutions for operating critical infrastructure components, they introduce vulnerabilities in addition to those that are invariably present in specialized process control systems. Mechanisms should be in place for incident response in the event of an attack or system failure. Security specialists need to investigate the root cause of the problem, resolve the issue, and mitigate current and future threats while minimizing or eliminating the downtime of control systems. This paper explores the application of a commercial software solution to perform secure, remote forensic analysis of process control systems while they are operating.

## 2. Process Control Systems

The United States critical infrastructure includes approximately 28,600 networked financial institutions, two million miles of pipeline, 2,800 power plants, 104 nuclear power plants, 80,000 dams, 60,000 chemical plants, 87,000 food processing plants, and 1,600 water treatment plants [3]. The National Strategy for Homeland Security states that these systems are so vital that their destruction "would have a debilitating impact on security, national economic security, national public health or safety, or any combination of [these] matters" [4].

Process control systems are responsible for the safe, reliable and efficient operation of many critical infrastructure components. One example is a supervisory control and data acquisition (SCADA) system, which performs tasks such as monitoring switches and valves, controlling temperature and pressure levels, and collecting and archiving field data. SCADA systems are required to maintain 24/7 availability and provide real-time (or near real-time) response. Estimated downtime costs in certain sectors range from $1 million to $4 million per hour [3].

The security of process control systems is a major concern. This is because current systems often use commodity hardware and commercial off-the-shelf software (operating systems, databases and applications) and because of increased network connectivity. Proprietary protocols have been replaced with Ethernet/IP-based protocols allowing for inexpensive, efficient solutions, but these expose process control systems to common network attacks. It is also increasingly common for process control systems to connect to enterprise networks (e.g., corporate IT networks), which are typically connected to the Internet. Figure 1 shows an industrial network that accesses real-time data from control systems for tasks such as statistical analysis, trending and budget analysis [3]. Even when control systems are isolated in their own internal networks, they are still vulnerable to attacks by malicious insiders or insiders who unwittingly introduce malicious code via removable media.

## 3. Digital Forensics

It is critical to implement security and auditing mechanisms for process control systems to combat vulnerabilities introduced by the underlying technology. Beyond network firewalls, monitoring tools and intrusion detection systems, there is a need for utilities that offer timely incident response and forensic analysis when protection systems fail. Identifying the problem and discovering the root cause of a system compromise or failure is important to mitigate its negative effects as well as to secure control systems from future breaches.

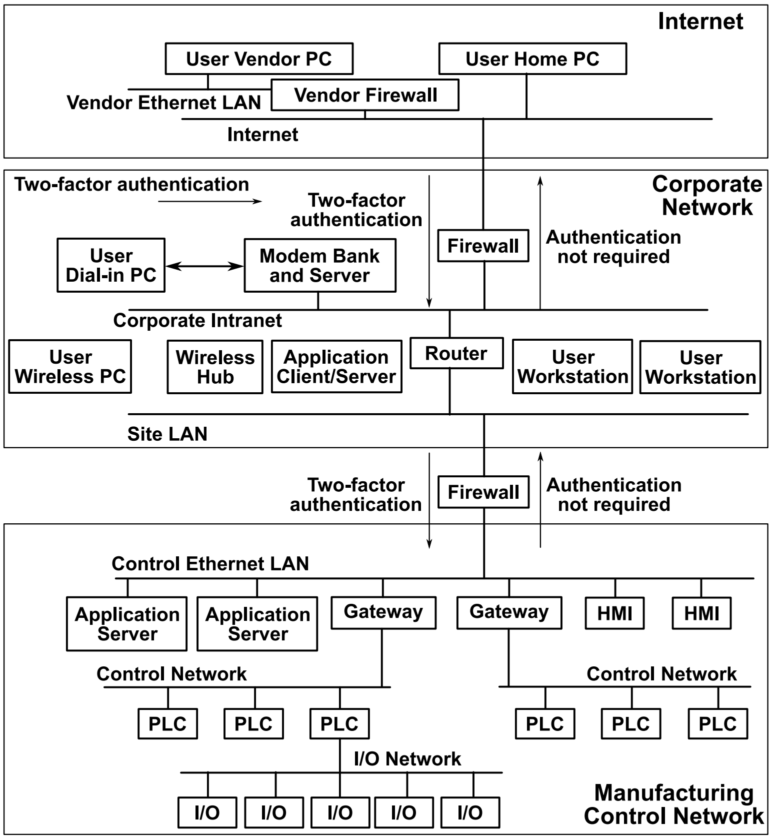

*Figure 1.* Process control network with Internet connectivity [3].

## 3.1 Forensic Analysis Requirements

This section lists the main requirements for performing forensic analysis on control systems.

**Availability** During a digital forensic investigation, a computer system is typically shut down and taken back to a laboratory where a specialist conducts a forensic examination. A process control system cannot be taken offline for forensic analysis, especially if it is monitoring or controlling plant operations. It is, therefore, necessary to conduct *in situ* analysis of a control system while it is operating.

**Remote Analysis** Control systems are widely distributed, often located hundreds of miles away from the control center and in hard-to-reach locations (e.g., offshore rigs). When responding to a security incident, it may not be feasible to wait for an examiner to travel to a distant site. It is, therefore, necessary to conduct remote analysis of control systems from a central location.

**Secure Analysis** Proper forensic analysis requires full access to a system and, if performed remotely, the access and retrieval of forensic data should be performed securely. The protocols and applications used for remote forensic analysis should have high assurance.

**Custom Analysis** Most digital forensic tools are designed for analyzing common workstations in an IT environment. They provide features that are optimized for file recovery, web history analysis, email analysis and keyword search. The forensic analysis of process control systems requires techniques and tools that can accommodate proprietary or specialized hardware, software, applications and protocols. Forensic tools for control systems should, therefore, be customizable and enable extensions, e.g., using plug-ins.

## 3.2 EnCase Enterprise

EnCase Enterprise from Guidance Software [1] is one of the most widely used digital forensic tools. It has sophisticated network-based forensic examination features, and satisfies all the requirements listed above. In particular, EnCase Enterprise is a multi-threaded scalable platform that provides immediate response, forensic analysis and proactive monitoring of large-scale enterprise networks. According to Guidance Software, EnCase Enterprise is designed for "anytime, anywhere" investigations – this makes it an attractive tool for conducting forensic analyses of process control systems.

EnCase Enterprise has three components that make remote analysis of a system possible (Figure 2). The Examiner component, which is the primary system used by a forensic examiner, houses the interface to EnCase forensic tools and applications. The Servlet component is a highly specialized service that runs on a target node and provides bit-level access to its hard drives. The SAFE (secure authentication for EnCase) component implements secure communications between the Examiner and Servlet components; it authenticates EnCase users and controls network access to remote servlets. SAFE is typically installed on a security-hardened server.

## 4. Customizing EnCase with EnScripts

EnCase's EnScript technology provides sophisticated customizable features for forensic examiners. It is a C++ based scripting language for interacting with live systems and analyzing volatile data and storage media.

Using the EnScript editor, scripts can be created to automate forensic processes and analyze large data sets that would be impractical to perform manually. Since EnCase Examiner provides bit-level access to a target system, EnScript's capabilities are limited only by the programming abilities and creativity of the script writer.

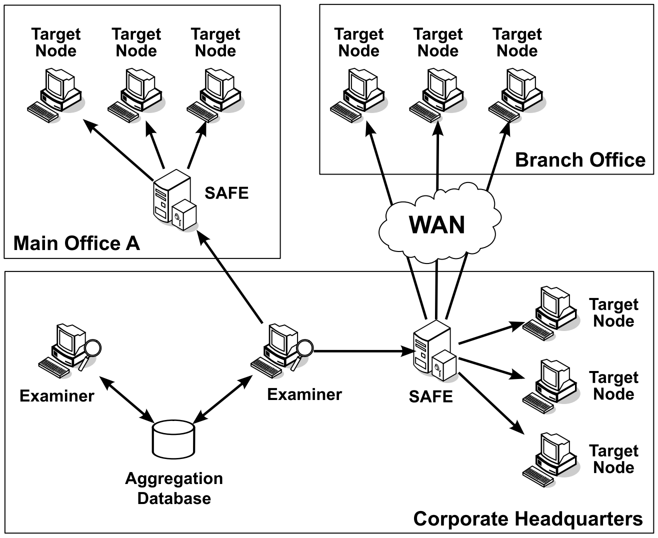

*Figure 2.* EnCase Enterprise architecture [2].

*Table 1.* HMI system threats.

| **Threat** | **Description** |
|---|---|
| User Account Tampering | HMI software should implement a user authentication policy with role-based permissions. User permissions could be modified maliciously or unauthorized user accounts could be created. |
| Project File Tampering | Project files make up the graphical interface used by an operator to monitor and control devices in a process control network. Configurations specified in these files could be modified maliciously. |
| Trojaned HMI Processes and Services | HMI software typically has multiple running processes and services. Services have open network connections. A Trojan could cause unwanted processes to execute and also cause anomalous behavior by the HMI. A Trojan could also send malicious communications using network ports that should not be open. |

Our research involved the analysis of human machine interface (HMI) software from two vendors. An HMI provides a graphical user interface for monitoring sensors and configuring programmable logic controllers (PLCs) in a process control network. Table 1 lists three main threats identified for HMI systems. EnScripts were written to identify the corresponding exploits.

*Table 2.* EnScripts for monitoring HMI system threats.

| **Threat** | **EnScript Description** |
|---|---|
| User Account Tampering | This EnScript accesses the user account configuration file on a live HMI system. User account information is extracted from the binary file and displayed for the examiner to review. Individual user accounts are checked for validity along with the permissions associated with the accounts. User account files may be copied to a forensic examiner's system for further analysis. |
| Project File Tampering | Two EnScripts assist in detecting tampering of HMI project files. The first script enables an examiner to review the contents of binary log files for HMI software. Changes should be properly logged when configuration changes are made to an HMI project. The examiner can use the script to verify that timestamped changes to configuration files are recorded in the log files. The script also extracts warnings and error events from the log files.<br>The second EnScript enables an examiner to create and compare hash values of selected files. For example, initial hash values can be computed for files in an HMI project that is known to be in a safe, operable and correct state. The hash values are periodically re-computed and compared with the known good set of hash values. Any discrepancy in hash values produces an alert that is recorded for further analysis. |
| Trojaned HMI Processes and Services | This EnScript provides detailed information about running processes and can be adjusted to focus on processes associated with HMI software. A timeline is given for the process execution order, and a visual process tree (or hierarchy) is constructed and displayed. A forensic examiner can study the instances, open files, dlls and open network ports associated with a process of interest. |

The EnScripts described in Table 2 were written to identify exploits corresponding to the threats listed in Table 1. They are illustrative of the range of scripts that may be written to support forensic analyses of process control systems.

## 5. Benchmark Testing

Due to the critical nature of its operations, a process control system must run continuously without any downtime or delay in transmitting control signals and process data. Consequently, the load placed on the running control system during forensic analysis must be minimized. Furthermore, forensic analysis

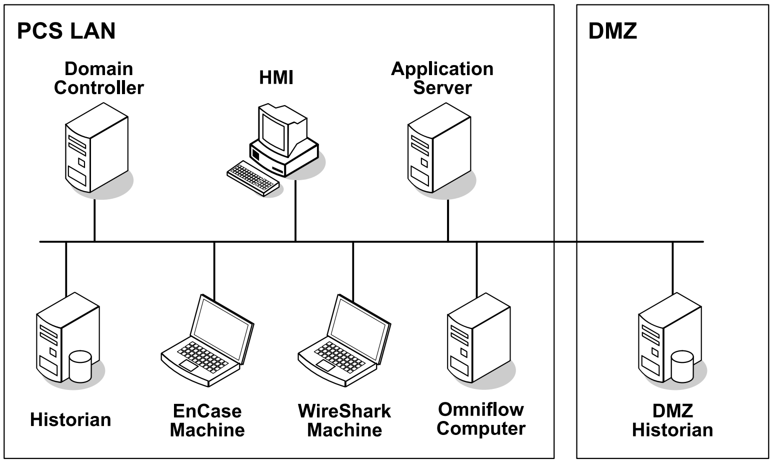

*Figure 3.* Benchmark test laboratory.

should be completely transparent to all control operations and should not incur additional overhead or maintenance on the part of an operator.

Benchmark testing was performed on the test systems to determine whether or not the forensic analysis would: (i) impact control system behavior, (ii) perform CPU and memory intensive procedures, and (iii) be detectable by monitoring CPU utilization, memory utilization, disk quotas, etc. As discussed above, forensic analysis should have little, if any, impact on the behavior of the control system. It is also important that forensic techniques that are CPU- and memory-intensive (e.g., imaging a hard drive) be performed without disturbing the normal mode of operation. Furthermore, it is desirable that forensic analysis be undetectable to the extent possible. If an adversary (or malicious operator) gained access to a control system and noticed high CPU or memory utilization, he/she might use the task manager to identify suspicious processes and stop the live forensic analysis.

EnCase Enterprise was used to perform forensic analysis in the benchmark tests. The tests used custom EnScripts that were written to analyze control systems running Wonderware software. Complete forensic images were acquired from the hard drives of the control system platforms. Several performance measures were computed while the forensic analysis was being performed. The laboratory setup, system specifications, procedures used, and results of the benchmark tests are described in the following sections.

## 5.1 Laboratory Seutp

The laboratory setup for the benchmark tests is shown in Figure 3. It incorporates several control system components.

- **Domain Controller:** This Windows domain controller manages access to resources in the control network.

- **HMI:** This system runs a graphical user interface that is used to monitor the status of flow rate data from the OmniFlow computer.

- **Application Server:** This system contains development tools to configure the control system, and holds the configuration data for the laboratory setup. Additionally, it provides computing resources for process control objects and drivers that communicate with external devices such as the OmniFlow computer.

- **Historian:** This system runs an SQL server that stores historical data. It also provides real-time data to other programs as needed.

- **DMZ Historian:** This historian resides in a network DMZ. It mirrors some of the data in the regular historian so that a corporate network can access the data without connecting to the control system network.

Auxiliary systems included an EnCase machine and a Wireshark network protocol analyzer. An OmniFlow computer was used to generate the values monitored by the laboratory control system.

## 5.2 System Specifications

Five systems were benchmarked to ascertain the impact of the two EnScripts and hard drive imaging executed on control system hardware and software. The specifications of the systems used in the benchmark tests are summarized in Table 3.

Two auxiliary machines were used in the benchmark tests, the EnCase machine and a machine that ran Wireshark network protocol analyzer to evaluate the impact of hard drive imaging on alarm propagation. The specifications of the auxiliary systems are summarized in Table 4.

## 5.3 Test Procedure

Microsoft's Performance Monitor Wizard (Version 1.1.3) was used to log performance data during the execution of EnCase EnScripts. The following six tests were conducted for each control system component:

- Benchmark (no servlet and scripts).

- Servlet only.

- Servlet and hash script.

- Servlet and process tree script.

- Servlet and both scripts (executed sequentially).

*Table 3.* System specifications.

| **Function** | **Processor and Speed** | **Memory** | **Operating System** | **Control Software** |
|---|---|---|---|---|
| Domain Controller | Pentium 4 (3 GHz) | 1 GB | Microsoft Windows Server 2003 (Standard Edition) – Service Pack 1 | NA |
| HMI | Pentium 4 (3 GHz) | 1 GB | Microsoft Windows XP (Professional) – Service Pack 2 | Wonderware |
| Application Server | Pentium 4 (3 GHz) | 1 GB | Microsoft Windows Server 2003 (Standard Edition) – Service Pack 1 | Wonderware |
| Historian | Pentium 4 (3 GHz) | 1 GB | Microsoft Windows Server 2003 (Standard Edition) – Service Pack 1 | MySQL |
| DMZ Historian | Pentium 4 (3 GHz) | 1 GB | Microsoft Windows 2000 (5.00.2193) – Service Pack 4 | NA |

*Table 4.* Auxiliary system specifications.

| **Function** | **Processor and Speed** | **Memory** | **Operating System** | **Software** |
|---|---|---|---|---|
| EnCase Machine | Intel T2600 (2.16 GHz) | 2 GB | Microsoft Windows XP (Professional) – Service Pack 2 | EnCase |
| Wireshark Machine | Pentium 4 (1.7 GHz) | 512 MB | Linux Kernel 2.6.15 | Wireshark |

- Servlet and complete imaging of hard drive.

The following performance data was recorded for each test:

- Percentage of committed memory bytes in use.
- Pages swapped per second.
- Bytes received at the network interface per second.

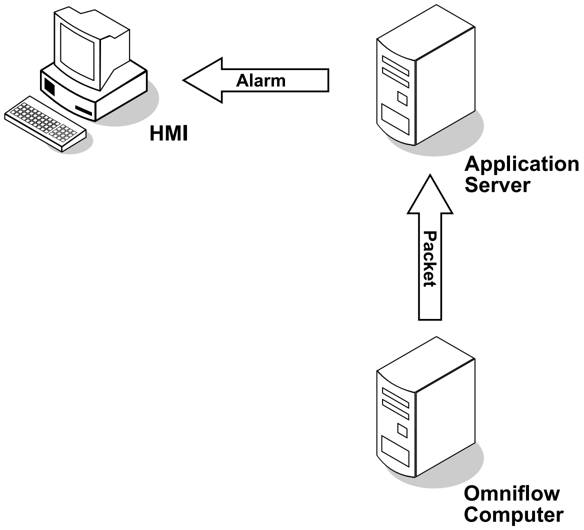

*Figure 4.* Alarm propagation path.

- Bytes sent from the network interface per second.
- Percentage of CPU time used.

Test data was collected for each performance measure for each control system component. For example, data collected from the application server can be used to compare the pages swapped per second for the baseline system, servlet only, servlet and hash script, servlet and process tree script, servlet and both scripts, and servlet and hard drive imaging. This data helps evaluate the impact of forensic analysis (in terms of a performance metric) on the control system component being investigated.

Tests were conducted to investigate if imaging a hard drive increased the time taken for an alarm to propagate from the OmniFlow computer to the HMI. The tests considered the alarm propagation path (Figure 4). Note that a packet containing an alarm triggering value is generated by the OmniFlow computer and sent to the application server. Next, the server recognizes the alarm value and sends an alarm to the HMI. Upon receiving the alarm, the HMI displays it on the operator's computer screen.

The first step was to arrange for an alarm to fire in the application server when the flow rate exceeded 1,500 bbl/hr. Next, the test machines and the Wireshark machine were linked to a Network Time Protocol (NTP) server (domain controller) to synchronize their clocks. Once this was done, the Wireshark machine was configured to capture packets going from the OmniFlow computer to the application server. The flow setting was then changed to a value below 1,500 bbl/hr.

When the HMI displayed a flow rate less than 1,500 bbl/hr, the flow setting was changed to a value above 1,500 bbl/hr. It was then possible to measure the time interval between the OmniFlow computer sending the first response packet with a value above 1,500 bbl/hr and the alarm being fired by the HMI. Knowing which packet contained a value above 1500 bbl/hr was simple because the application server only polled the OmniFlow computer every ten seconds; therefore, when the OmniFlow monitor value exceeded 1,500 bbl/hr, the next packet sent by the OmniFlow computer would be the alarm triggering packet. Unfortunately, the alarm settings on the application server had a resolution in seconds, so the measurement accuracy was limited to one second.

This test was performed three times each for three different cases. The first case was a control measurement with the system operating in its native state without any running EnCase processes. The second was the system running while a hard drive image of the application server was being taken by EnCase. The third was the system running while a hard drive image of the HMI was being taken.

## 6. Test Results

Using EnCase for remote forensic analysis of a live process control system was successful. It was also possible to create a bit-for-bit copy of a process control system's hard disk while it was in continuous operation. Custom forensic analysis of an HMI running different process control software was possible using EnCase's EnScript feature. Benchmarking was performed on a process control system to determine processor, memory and network resources required during remote analysis and imaging.

### 6.1 Custom EnScript Results

The two EnScripts (hash and process tree scripts) were executed on the domain controller without any major performance penalties. They produced a slight increase in memory use and a few spikes in network traffic and CPU utilization.

Similar results were obtained for the HMI, application server, historian and DMZ historian. The scripts were only applied to subsets of files and directories specific to process control software on the HMI, application server and historian; they took about one minute to execute. In the case of the domain controller and DMZ historian, the scripts were executed on a few chosen files and directories because these two machines did not run any process control software. In general, the custom scripts worked well in all instances and had little impact on the machines in the laboratory setup. The servlet also had no noticeable impact on system operation in all the tests.

The memory utilization benchmarks exhibited several anomalies. The most significant was that the baseline and servlet alone used more memory than some of the script executions. However, since the memory utilization in the benchmark tests differed by less than 0.5% (less than 5 MB of memory on the test systems), it is within the margin of experimental error. The memory utilization in the case of the DMZ historian had a similar anomaly with the benchmark tests being within approximately 2% of each other. However, the benchmark tests do show that forensic analysis does not have a detrimental effect on the available system memory.

The hash script had a significant impact on system operation, but should still be tolerable for most systems. The script produced one or two CPU utilization spikes to about 40% utilization, each lasting no more than five seconds. However, many of the hash script executions also resulted in CPU loads of 20% over a 40 second period. This 20% utilization did not impact the test systems, but should be scaled to the CPU resources available in control systems. The hash script also triggered bursts of page swapping when hash values were calculated. Increased network activity due to the hash script must also be considered, but this did not negatively impact system operation.

The process analysis script generally resulted in one or two CPU utilization spikes to about 50% utilization, each lasting less than five seconds. However, it is possible to scale the utilization according to processor speed. Increased network activity due to the process analysis script produced several small spikes, but these were close to those observed during normal system operations.

### 6.2 Imaging Results

In the case of the domain controller, the most noticeable performance decreases were seen in the numbers of bytes received and sent, and CPU utilization. While the hard drive was being imaged, peak values of 220,000 bytes/s received, 650,000 bytes/s sent and 45% CPU utilization were observed, each lasting about 30 seconds, 1 minute and 3 minutes, respectively. Similar results were obtained for the HMI, application server, historian and DMZ historian.

A future version of EnCase could permit users to set thresholds on the numbers of bytes sent and received per second, and on CPU utilization while forensic analysis is being performed. With this functionality, forensic analysis could be conducted without "evident" decreases in performance.

### 6.3 Alarm Propagation Results

The test results reveal that, even under the stress of imaging the entire hard drive of a control system component, there is no significant delay in alarm propagation. The time taken for a packet containing an alarm triggering value to manifest itself as an alarm in the HMI was not affected by the imaging process. In particular, the alarm delay was no more than one second, which was the resolution of HMI alarm timestamps.

## 7. Conclusions

Standard monitoring and forensic practices can enhance security in large-scale process control systems. The targeted areas include user authentication and permissions, configuration and log files, active processes, and open network connections. Process control software has begun to implement user-based and role-based authentication, and it is necessary to determine if these features have been circumvented. Analysis of log files, timestamps and hash values helps identify tampering of configuration files and other key files. It is also important to determine if process control software is behaving as intended by tracing process activity and monitoring open network connections.

Our test results demonstrate that EnCase Enterprise is an effective tool for conducting remote forensic examinations of live process control systems. Forensic processes, in particular, hashing, process analysis and hard drive imaging, did not strain CPU, memory and network resources to levels that impacted control system behavior or functionality. Furthermore, Encase Enterprise's scripting features make it possible to customize forensic techniques for proprietary hardware, software, applications and protocols used in process control networks.

## References

[1] Guidance Software, EnCase Enterprise (www.guidancesoftware.com/products/ee_index.asp), 2006.

[2] Guidance Software, How it works – EnCase Enterprise (www.guidancesoftware.com/products/ee_HowItWorks.asp), 2006.

[3] A. Miller, Trends in process control systems security, *IEEE Security and Privacy*, vol. 3(5), pp. 57–60, 2005.

[4] Office of Homeland Security, The National Strategy for Homeland Security, The White House, Washington, DC (www.whitehouse.gov/homeland/book/nat_strat_hls.pdf), 2002.