

A Survey of Industrial Control System Testbeds

Hannes Holm^(✉), Martin Karresand, Arne Vidström, and Erik Westring

Swedish Defence Research Agency (FOI), Olaus Magnus väg 42, Linköping, Sweden
{hannes.holm,martin.karresand,arne.vidstrom,erik.westring}@foi.se

Abstract. Conducting security tests such as vulnerability discovery within Industrial Control Systems (ICS) help reduce their vulnerability to cyber attacks. Unfortunately, the extreme availability requirements on ICS in operation make it difficult to conduct security tests in practice. For this reason, researchers and practitioners turn to testbeds that mimic real ICS. This study surveys ICS testbeds that have been proposed for scientific research. A total of 30 testbeds are identified. Most of these aim to facilitate vulnerability analysis, education and tests of defense mechanisms. Testbed components are typically implemented as simulation models. Testbed fidelity is rarely addressed, and at best briefly discussed.

Keywords: Industrial Control Systems · Testbed · IT security · Cyber security · Systematic literature review

1 Introduction

Our society depends on various critical services such as electricity, water purification and transportation to properly function. Not long ago, the Industrial Control Systems (ICS) that supervised and controlled most of these critical services were realized by specially constructed isolated devices. Along with the rest of our society, ICS have evolved and are now often delivered by complex interconnected IT solutions including commercial-off-the-shelf (COTS) technologies that in one way or another are connected to the Internet. The main reasons behind this evolution are increased functionality and increased effectiveness, as well as reduced costs. For example, IP-based remote control of railroad signaling and interlocking systems has increased the level of control of the railroad system. The benefits of using IT for critical infrastructure applications are thus clear.

However, the trend of interconnectivity and COTS has also brought about problems. Issues that are common in regular IT architectures, such as malware and misconfigurations, do now occur in ICS systems as well. Reduced availability due to such issues might be acceptable in regular IT architectures, but are generally completely unacceptable for IT that supports critical infrastructure services. For instance:

- Computers along railway tracks in Sweden send continuous data regarding the state of the track to remote railway operators. If there are more than 15 seconds between two points of data for a device, the corresponding track is considered faulty and all trains designated to traverse it are blocked [37].

- In the Energy Sector, digital protective relays are used to trip circuit breakers when power faults are detected – an event that can cause significant product damage and personnel harm. This function needs to be executed within a few milliseconds of the power fault to be of use.

To understand and manage the complexity of an IT architecture, e.g., to discover and mitigate security vulnerabilities within it, technical audits such as penetration tests are carried out. While technical audits often are considered an effective security solution, they can disrupt system services when they are conducted. This is particularly evident for ICS IT solutions – these are often not able to withstand even the most basic scanning tools. For example, a study involving Programmable Logic Controllers (PLC) and the vulnerability scanner Nessus showed that the 18% of the tested PLCs crashed as a result of a scan [32]. As a consequence, technical audits are generally thought of as (at best) difficult for IT architectures that support critical infrastructure services.

To study the vulnerability of IT architectures that are difficult to technically audit without compromising their reliability and performance, many researchers attempt to copy them in isolated environments, also called testbeds, where experiments safely can be performed. Creating a test bed however comes with various challenges, in particular: (i) it can be difficult to obtain a realistic test bed scale, and (ii) it can be difficult to achieve a realistic test bed configuration.

There are a number of approaches that can be used to implement components and configurations in testbeds. The most obvious approach is to include real hardware and software configured as they are configured in practice. This naturally provides a very high degree of fidelity. However, it is difficult to reconfigure and maintain real hardware and software in a testbed, especially given the presence of software exploits that have the potential to damage systems; not to mention reach a valid testbed scale due to the costs involved. An alternative is to employ simulation, to develop a new application or model that operate similarly to a desired solution [39][46]. Simulation models are generally easy to reconfigure, maintain and can provide an extensive testbed scale. However, it is difficult to obtain high fidelity from simulation models, especially when software exploits need be considered as these often only work given a specific code-base and configuration.

A third more attractive means of obtaining a large-scale realistic testbed is through virtualization. Virtualization is a technology which concerns isolating computer software in a means that enables layers of abstraction, both between different software and between software and hardware. For example, a virtual private network adds a layer on top of a computer network that isolates its users from others on the network; the Comodo antivirus uses operating system-level virtualization to create a sandbox for isolated web browsing; VMware and VirtualBox use hardware virtualization to enable guest operating systems to interface with software and hardware; the Quick Emulator (QEMU) use instruction set virtualization to provide a complete emulation of computer hardware in software. Virtualizing a testbed is attractive for several reasons, for example:

- It enables running multiple systems in parallel on single computer hardware.
- It enables quickly reconfiguring systems and networks using software scripts.
- It enables isolating the activity in the testbed from the physical systems as well as external systems.
- It enables using actual software and protocols rather than simulated equivalents.

In other words, virtualization can potentially allow low-cost, replicable and safe security studies of IT architectures that have configurations valid to those of real ICSs. An overview of virtualization approaches is given by Nanda and Chiu [38]. Of the approaches discussed by the authors, hardware virtualization is especially attractive for testbeds as it enables high-performance execution of real applications in virtual containers. Emulation also enables execution of real applications, but is generally slower than virtualization as all instructions need to be trapped by the emulator.

1.1 Research Questions

This study surveys existing ICS testbeds that have been proposed for scientific research and tries to answer the following four research questions (RQs):

- *RQ1*: Which ICS testbeds have been proposed for scientific research?
- *RQ2*: Which research objectives do current ICS testbeds support?
- *RQ3*: How are ICS components implemented in current ICS testbeds?
- *RQ4*: How do existing ICS testbeds manage requirements?

These RQs are addressed to gain an understanding of how previously constructed ICS testbeds for scientific research have been designed.

1.2 Outline

This paper is structured as follows. Section 2 describes related work. Section 3 describes the method of the systematic literature review. Section 4 describes the outcome of the systematic review. Finally, Section 5 concludes the paper and presents possible future research directions.

2 Related Work

To the authors' knowledge, there are as of yet no articles that focus on surveying ICS testbeds. That said, most articles that describe specific testbeds also briefly compare these testbeds to a few others that are deemed similar in scope. A recent such example is the article by Siaterlis and Genge [50], who compare the testbed EPIC to eight other current ICS testbeds. They use a loosely defined scale from one to three to compare the testbeds according to six main criteria (fidelity, repeatability, measurement accuracy, safety, cost effectiveness and multiple critical infrastructures) and two sub-criteria (cyber and physical).

There are however articles that focus on surveying network and software testbeds for other domains than critical infrastructures and ICSs. This section describes such surveys. Harwell and Gore [25] provide an overview of cyber ranges (a type of network and software testbed) and their usage and note that there are more than 100 active in the United States alone.

Davis et al. [13] present a survey of cyber ranges and categorize these in three categories: (i) modelling and simulation (where models of each component exist), (ii) ad-hoc or overlay (running tests on production network hardware with some level of test isolation provided by a software overlay) and (iii) emulation (mapping a desired experimental network topology and software configuration onto a physical infrastructure). In addition to these categories, they discuss capture the flag competitions such as DefCon, which use their own cyber ranges for their events. The authors also categorize the cyber ranges according to their supporting sector: academic, military or commercial. They found that the objective of most cyber ranges was training, and that most cyber ranges used either simulation or emulation.

Gluhak et al. [19] provide a survey on testbeds for experimental internet of things (IoT) research and identify a total of 23 testbeds. These testbeds have a different scope than the cyber ranges surveyed by Davis et al. [13] in the sense that they focus on specific networking technologies such as Wireless Sensor Networks. This scope in effect requires that the testbeds to a greater extent employ real hardware in front of virtualization.

Leblanc et al. [31] provide a snapshot of different tools and testbeds for simulating and modeling cyber attacks as well as defensive responses to those. The authors note that there is a considerable interest in the topic and that significant progress have been made; however, they also observe that there appears to be very little coordination and cooperation behind this progress.

3 Review Protocol

The RQs were investigated using the standard systematic literature review approach described by Kitchenham [29]. The review began with unstructured searches related to the topic with the purpose of identifying relevant keywords for systematic searches. A set of preliminary keywords were then used to query Scopus¹ for articles published between January 2010 and the 20th of November 2014 with the chosen keywords within their titles, keywords or abstracts, yielding a total of 123 matches. The result of this query was deemed too narrow; thus, the keywords were extended to be more inclusive. During the 18th of December 2014, a final set of keywords² was used to query Scopus. This query identified 1335 articles.

¹ A database that contains conference and journal articles from all major publishers, including IEEE, ACM, Springer, Elsevier and Wiley.

² (scada OR ics OR mtu OR plc OR rtu OR io OR “embedded device” OR “embedded system”) AND ((virtuali* OR simulat* OR emulat* OR hypervi* OR vmm OR “virtual machine” OR “dynamic recompilation”) OR (testbed OR “test bed” OR “cyber range”)).

The relevance of a subset of the 1335 articles (the 123 articles identified during the pre-study) was independently judged based on titles and abstracts by randomly chosen pairs of researchers. Redundant judgments were used to enable measuring the group’s internal agreement with the statistical metric Cohen’s Kappa [10]. The results showed strong agreement (a Kappa of 0.88 on a scale from 0 [no agreement] to 1 [complete agreement]), which is a sign that the group shares the same view on the project scope. Due to the strong agreement, each of the remaining 1212 articles was read by no more than one researcher. Out of the 1335 articles, 63 were judged as relevant and read in detail. Of these articles, 40 both concerned ICS testbeds and were deemed relevant after the more detailed review. The results from this literature review are presented in the following sections.

To answer the RQs, the following data were extracted from each article: (i) the objectives of the testbed, (ii) the configuration choices of the testbed and (iii) how the testbeds fidelity is ensured.

4 Results

The systematic literature review identified a total of 40 articles. These concerned 30 ICS testbeds that were planned or currently operational at the time of the present study. An overview of these testbeds is described by Table 1.

As can be seen, almost half of the identified testbeds were located in the USA. Five testbeds were only planned ([8], [15], [18], [28] and [58]), while the remaining 25 were claimed to be operational to an extent that facilitated technical studies related to their stated purposes. It should be mentioned that there are various other testbeds, such as DETER [5] and the U.S. National SCADA testbed, that were not directly identified by the systematic review. There are two explanations behind this: (i) they had either not published their results in forums indexed by Scopus or (ii) did not specifically concern ICS. The U.S. National SCADA testbed corresponds to the first explanation; DETER is not a testbed that has been designed for the purpose of ICS tests and thus corresponds to the second explanation. The testbeds that employ DETER, such as the testbed at the Technical Assessment Research Lab in China [17], view DETER as a tool that help realize an ICS testbed (similar to Matlab, OPNET or VirtualBox). The present study views DETER and other similar testbeds (e.g., Emulab, GENI and PlanetLab) in the same fashion as the ICS testbeds that use them.

4.1 Objectives of ICS Testbeds

An overview of the objectives that the creators of the testbeds present is given in Table 2. The most commonly mentioned objective is to use a testbed for vulnerability analysis, with education and tests of defense mechanisms on a split second place. These objectives highlight the fact that most testbeds focus on cyber security rather than, for instance, performance issues due to UDP packet loss.

Table 1. Overview of ICS testbeds.

ID	University/Organization	Country	References
1	American University of Sharjah	Abu Dhabi	[11]
2	Queensland University of Technology	Australia	[30]
3	RMIT University	Australia	[2],[40]
4	Research Institute of Information Technology and Communication	China	[58]
5	Technical Assessment Research Lab	China	[17]
6	Tsinghua University of Beijing	China	[9]
7	University of Zagreb	Croatia	[28]
8	Queen’s University Belfast	Ireland	[61]
9	University College Dublin	Ireland	[51]
10	European Commission Joint Research Centre	Italy	[20],[50]
11	European Commission Joint Research Centre	Italy	[16]
12	Ricerca sul Sistema Energetico	Italy	[14]
13	American University of Beirut	Lebanon	[44]
14	University Kuala Lumpur	Malaysia	[47],[48]
15	TNO	Netherlands	[8]
16	ITER Korea	South Korea	[54]
17	Case Western Reserve University	USA	[34]
18	Iowa State University	USA	[22],[23]
19	ITESM Campus Monterrey	USA	[43]
20	Lewis Research Center	USA	[4]
21	Mississippi State University	USA	[35],[36],[41],[42],[57]
22	Ohio State University	USA	[21]
23	Pacific Northwest National Laboratory	USA	[15]
24	Sandia National Laboratories	USA	[56]
25	Tennessee Technological University	USA	[52]
26	The University of Tulsa	USA	[24]
27	UC Berkeley	USA	[18]
28	University of Arizona	USA	[33]
29	University of Illinois at Urbana-Champaign	USA	[6],[7],[12]
30	University of Louisville	USA	[26]

These objectives are in general described on a very superficial level. For example, the type of vulnerability analysis that is proposed is typically described with generic statements such as “*It is imperative to analyze the risk to SCADA systems in terms of vulnerabilities, threats and potential impact*” [8] and “*An evaluation of the security of SCADA systems is important*” [2]. However, as stated by Davis et al. [12], the complex hardware and software interactions that must be considered makes vulnerability analysis a difficult task. Thus, there is a need to break it down into more tangible topics in order to yield useful testbed requirements. The same reasoning applies for other objectives, such as education and tests of defense mechanisms.

Table 2. Objectives of testbeds.

Objective	Testbeds
Vulnerability analysis	16
Education	9
Tests of defense mechanisms	9
Power system control tests	4
Performance analysis	1
Creation of standards	1
Honeynet	1
Impact analysis	1
Test robustness	1
Tests in general	1
Threat analysis	1

4.2 Implementation of ICS Testbed Components

Based on NIST 800-82 [53], an ICS testbed should consider four general areas: the control center, the communication architecture, the field devices and the physical process itself. This section describes how components concerning these areas are implemented in the 30 surveyed testbeds. An overview of the results is described by Table 3. More detailed descriptions are provided in the following sections.

Table 3. Number of articles assessing different areas and methods of implementation (virtualization, emulation, simulation and hardware).

Area	Covered	Virtualization	Simulation	Emulation	Hardware
Control center	20	4	9	1	11
Communication architecture	22	6	10	3	11
Fields devices	23	0	14	0	14
Physical process	12	0	12	0	0

The Control Center concerns the servers and operator stations that are used to remotely observe and control field devices, such as MTUs and data historians. Approximately two thirds of all testbeds contain descriptions regarding how their control center components are incorporated. Of these, most utilize simulations (30%) and/or hardware (37%). It is interesting that so few (13%) testbeds choose to virtualize the control system components, something which to a large extent is possible as they typically involve COTS operating systems such as Windows and Linux. The virtualization solutions that are mentioned concern DETER,

Emulab, GENI, PlanetLab and VirtualBox. Simulation-based approaches concern LabVIEW, Mathworks Simulink, HoneyD in combination with IMUNES (FreeBSD jails), the RINSE network simulator and custom Python scripts. The emulation approach involves RINSE (it combines emulation and simulation). Hardware concerns standard x86-based computers such as CitectSCADA 6.1 on Windows XP (used as OPC server and HMI).

The Communication Architecture involves components that realize communication within ICS, for instance, routers, switches and modems. 73% of all testbeds contain descriptions regarding how their communication architecture is incorporated. Of these, most utilize simulations (33%) and/or hardware (37%). As for control systems, many kinds of communication architectures are possible to easily virtualize. For example, Ethernet is commonly used within ICS and is easily virtualized through e.g. VirtualBox. Thus, it is interesting that few testbeds (20%) choose to do so. Virtualization is proposed using DETER, GENI, Emulab or Virtualbox. Simulation is proposed using OPNET, SITL communication network simulator, Iperf (for background traffic), RINSE, OMNET++, PowerWorld simulator, Mathworks Simulink, the Inet framework, NS-2, Networksim, the c2windtunnel framework, IMUNES, and custom Python scripts. Emulation is proposed using CORE (in combination with OpenVZ) and RINSE. Hardware generally involves Ethernet devices such as routers and switches.

Field Devices concern the components that link the physical world to the digital world, for instance, a PLC or an RTU. 77% of the testbeds contain descriptions on how field devices are incorporated – a higher number than for the control system, the communication architecture or the process. None of the testbeds contain virtualized or emulated field devices. An explanation for this result is that ICS field devices generally are based on specialized, sometimes proprietary, hardware and software that are unsupported by common virtualization and emulation tools. Simulation (47% of all testbeds) and hardware (47% of all testbeds) are used instead. Used simulation tools include STEP7 (of Siemens S7 PLCs), RSEmulate (by Allen-Bradley), LabVIEW, Scadapack LP PLC, Modbus Rsim, Soft-PLC, Python scripts with CORE, OpenVZ, PowerWorld server, and HoneyD in combination with IMUNES (FreeBSD jails). Hardware includes, for example, Allen Bradley Control Logix PLC, National Instruments NI-PXI, Omron PLC CJ1M-CPU11-ETN, CompactRIO from National Instruments, ABB 800F, Siemens OpenPMC, Siemens S7 PLC, Emerson Ctrl MD, and GE FANUC Rx3i.

The Physical Process concerns the physical reality that the ICS observe and control. Less than half of the testbeds describe how the process is implemented. In all cases, implementation builds on simulation models (rather than actual physical processes). The simulation approaches build on Matlab, Mathworks Simulink, Power Hardware-in-the-Loop (OPAL-RT), LabVIEW, PowerWorld, AnyLogic and EZJCOM, ANSYS, real time digital simulators, an Abacus

solar array simulator, a library file (.dll) for EPANET, OMNET, and a custom application written in Java.

Various Components and Protocols on different levels of abstraction are mentioned in the articles describing the 30 analyzed testbeds. The most commonly mentioned types of components are RTU (mentioned by 12 testbeds), MTU (8 testbeds), PLC (8 testbeds), HMI (7 testbeds) and IED (4 testbeds). Other product types that are mentioned by a single testbed each are DAQ, Data aggregator, HDBMS, OPC server/client, PDC, PMU, Relay and SCADA server/client. 13 testbeds do not mention any product types. It is worth mentioning that these definitions are rather vague, especially to practitioners. For example, the Swedish railroad has Siemens S7 PLCs that are connected to switchgear. The purpose of these PLCs is to package/unpackage the proprietary data that the switchgear sends and receives by the MTU. For this reason, the Siemens S7 PLCs are denoted as RTUs by operators of the Swedish railroad (as they have a specific purpose).

There are several components in NIST 800-82 [53] that are not explicitly mentioned for any testbed. In particular, the data historian, IO server and control server are not mentioned. The articles do not describe why this is the case. An explanation could however be that these components are thought of as integrated with the MTU.

Of the communication protocols described for the testbeds, Modbus (Modbus ASCII, Modbus TCP or Modbus RTU, mentioned by 13 testbeds) and DNP3 (12 testbeds) are by far the most commonly mentioned. OPC (5 testbeds), IEC 60870 (4 testbeds, including e.g. IEC 104), IEC 61850 (3 testbeds) and Profibus (2 testbeds) are also mentioned for more than one testbed. Fieldbus, FINS, GOOSE, ICCP, IEEE C37.118, CIP, RJ45, DeviceNet and Genius are mentioned for a single testbed each. Nine testbeds do not discuss any communication protocols. According to the American Gas Association's AGA-12 standard [1], there are between 150 and 200 SCADA protocols. There are thus a plethora of protocols that are not covered by current testbeds. How common these protocols are in practice is however unknown to the authors of this article.

4.3 Managing Testbed Requirements

Siaterlis et al. [49] describe four overall requirements that cyber security testbeds should fulfill:

- *Fidelity*: Reproduce as accurately as possible the real system under study.
- *Repeatability*: Repeating tests produces the same or statistically consistent results.
- *Measurement accuracy*: Observing tests should not interfere with their outcome.
- *Safe execution of tests*: Cyber security tests often involve adversaries that exploit systems using malicious software. As it can be difficult to know the

outcome of these activities beforehand, tests must ensure that the activity within the testbed is isolated.

Of these requirements, repeatability and measurement accuracy generally depend on activities outside of the technical scope of a testbed. For example, it is difficult to ensure that adversaries act in the same way during consecutive tests. For this reason, repeatability and measurement accuracy are excluded from the scope of the present pre-study. Safe execution of tests has been a focus area for most testbeds for cyber security analyses; for this purpose, it is arguably less interesting to study than fidelity.

Ensuring testbed fidelity, i.e., that a testbed accurately reflects the desired real environment(s), is a critical task as the quality of any data produced from interaction with the testbed otherwise is uncertain. More than half (63%) of the testbeds are not discussed at all regarding fidelity (see Table 4). The remaining testbeds are analyzed in respect to fidelity in two different means: practical experiences and/or standards. The fidelity of 23% of the testbeds is argued based on real data gathered by the authors: either from quantitative data gathered from ICS systems in operation and/or from qualitative personal experiences or discussions with ICS manufacturers, providers and operators. For instance, “*Based on discussions with some industry partners and on our own experience*” [2] and “*In order to capture real image of the power network, a small part of power network was taken*” [11]. The remaining 13% that discuss fidelity base their testbed designs on standards developed by NIST (e.g., the NIST 800-82), ISA (e.g., the ISA-99) or IEC (e.g., the IEC Smart Grid Standardization Roadmap).

Table 4. Testbed fidelity.

Fidelity	Testbeds
Not covered	19
Study of real systems	7
Based on standards	4

Of the testbeds that are discussed in terms of fidelity, two provide specific metrics that can be used to replicate their results with some degree of accuracy. The first is Reaves and Morris [41] (a testbed at the Mississippi State University), who describe 11 metrics involving Modbus traffic (e.g., byte throughput, master-to-slave inter-arrival time, error count and packet size). These metrics were chosen based on the rule sets of model-based intrusion detection systems. The authors also compare the result from attacks against testbed components (which in this case are simulated) to attacks against real components. The second is Siaterlis and Genge [50], who compare the execution time of their testbed to the required execution time of seven physical processes. Their results show that they fulfill the execution time for everything but the IEEE 118 bus model

(the testbed has an execution time of 155ms and the IEEE bus system has a requirement of 24ms).

An important aspect of testbed fidelity concerns what data should be collected in order to recreate a valid testbed design. For example, how a network topology or machine configuration best should be captured. Of all testbeds, the Iowa State University testbed is the only one that discusses this topic [22]. Hahn and Govindarasu [22] discuss how different data collection tools are able to fulfill the NIST 800-115 [45] methodology and the NERC critical infrastructure protection requirements. They used Wireshark to analyze network traffic, The Open Vulnerability Assessment Language (OVAL) Interpreter for analyzing machine configurations, Nmap and Sandia’s Antfarm for network and service discovery, Firewall and the access policy tool (APT) for firewall rule set discovery, and Nessus for vulnerability scanning. The results showed that these tools overall had excellent support for regular IT solutions such as Windows operating systems, but poor support for ICS specific components such as PLCs. For instance, *“there appeared to be numerous communications employing proprietary protocols which Wireshark was unable to identify”* and *“Nmap was not able to identify 53 out of 157 the open ports utilized in the network. This occurrence is a result of the heavy utilization of proprietary and SCADA specific protocols which are not recognized by Nmap”*. The analysis by Hahn and Govindarasu [22] is also limited as it does not study the potential to collect configuration data through agent based software, which is a common ICS industry practice.

5 Conclusions and Future Work

This study examined what ICS testbeds currently exist (RQ1), what ICS objectives these propose (RQ2), how ICS components are implemented within them (RQ3) and how they manage testbed requirements (RQ4).

The study identified 30 different ICS testbeds. The most common objectives of these testbeds are to facilitate vulnerability analysis, education and tests of defense mechanisms. These three objectives are described on a very superficial level for all existing testbeds. In order to be able to relate these objectives to actual testbed design decisions, there is a need to break them down and make them more tangible. One means to make them more tangible is to employ taxonomies, e.g., the taxonomy for ICS vulnerability assessment which is presented by NIST 800-82 [53]. This taxonomy employs three topics (policy and procedure vulnerabilities, platform vulnerabilities and network vulnerabilities) containing a total of 71 more concrete types of vulnerability assessments that can be used to create better requirements for ICS testbeds. For instance, if one wishes to analyze the presence of the platform vulnerability buffer overflow, there is a need for real software to be in place. This would preferably involve hardware, and at worst virtualization or emulation - simulation simply would not be sufficient as the software codebase would differ.

ICS components within the control center and communication architecture should generally be possible to virtualize without too many technical issues but

are still typically simulated by the testbeds. The technical difficulty of implementing field devices (e.g., a PLC or an RTU) depends on the kind of device that is considered. Modern field devices are often based on architectures and firmware that have current virtualization and/or emulation support. The same applies for field devices that manufacturers have created emulation software for (it is however not certain that manufacturers would want to share such technology). Older or proprietary field devices (such as the Siemens S7 series) are however not supported by any current virtualization or emulation approach. As a field device can be used for up to 40 years [55], there is bound to be a plethora of such devices in operation. Thus, it would be beneficial to construct emulators for these old and/or proprietary devices. There have been some research regarding virtualization of embedded systems [62][60][3][63]. Unfortunately, these works deal with performance issues such as the resource scheduling in hypervisors rather than how to virtualize specific existing field devices such as Siemens S7-1200. We are aware of but a single research project concerning this topic: an ongoing study by Idaho National Laboratory [27] proposes using the emulator QEMU in combination with the compiler LLVM to emulate field devices. This is a non-trivial task due to the extensive undocumented functionality in these devices. An example of the difficulty of reversing undocumented PLC code is given by Vidström [59], who present the results from reversing models in the Siemens S7 series. Due to this difficulty, a reasonable solution for field devices that are unsupported by current virtualization and emulation technologies could be simulation or implementation using real hardware. Of these two approaches, simulators are sufficient for most testbed purposes, with the exception of software and hardware vulnerability discovery.

What fidelity requirements that are posed on testbeds, and how these requirements are fulfilled, are rarely addressed by the studied articles. This is troublesome given the difficulty of validating cyber security results in general: if the validity of the testbed that facilitates tests of cyber security solutions is uncertain, any results produced by it are uncertain as well. To sum up, to accommodate high-fidelity security analyses, future ICS testbeds should:

- Clearly state the objectives of the testbed and relate these objectives to the configuration of the testbed.
- Employ virtualization or emulation in front of simulation and hardware approaches.
- Provide empirical results describing how the testbed fulfills its stated requirements.

For the third task, there is a need for a comprehensive evaluation framework that can be used to compare the fidelity of a testbed over time as well as compare it to other testbeds. As there currently is no “gold standard” available for this purpose, future work should focus on creating a standard framework for fidelity analyses of ICS testbeds.

Finally, there are various limitations to this work. First, the chosen search criteria have likely left out testbeds. Second, the data extraction formulary was

iteratively developed based on the results from a pre-study and the opinion by the group researchers. Even though the group was shown to share the same general mindset, a different set of researchers would certainly have amounted to different results.

References

1. (AGA), A.G.A.: Cryptographic protection of scada communications - retrofitting serial communications. Tech. rep., American Gas Association (AGA) (2006)
2. Almalawi, A., Tari, Z., Khalil, I., Fahad, A.: Scadavt-a framework for scada security testbed based on virtualization technology. In: 2013 IEEE 38th Conference on Local Computer Networks (LCN), pp. 639–646. IEEE (2013)
3. Åsberg, M., Forsberg, N., Nolte, T., Kato, S.: Towards real-time scheduling of virtual machines without kernel modifications. In: 2011 IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA), pp. 1–4. IEEE (2011)
4. Beach, R., Kinnach, G., Jett, T., Trash, L.: Evaluation of power control concepts using the pmad systems test bed. In: Proceedings of the 24th Intersociety Energy Conversion Engineering Conference, IECEC 1989, pp. 327–332. IEEE (1989)
5. Benzel, T.: The science of cyber security experimentation: the deter project. In: Proceedings of the 27th Annual Computer Security Applications Conference, pp. 137–148. ACM (2011)
6. Bergman, D.C.: Power grid simulation, evaluation, and test framework (2010)
7. Bergman, D.C., Jin, D.K., Nicol, D.M., Yardley, T.: The virtual power system testbed and inter-testbed integration. In: CSET (2009)
8. Christiansson, H., Luijff, E.: Creating a european scada security testbed. In: Goetz, E., Sheno, S. (eds.) Critical Infrastructure Protecti. IFIP, vol. 253, pp. 237–247. Springer, Boston (2008)
9. Chunlei, W., Lan, F., Yiqi, D.: A simulation environment for scada security analysis and assessment. In: 2010 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), vol. 1, pp. 342–347. IEEE (2010)
10. Cohen, J.: Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin* **70**(4), 213 (1968)
11. Darwish, K.W., Dhaouadi, R., et al.: Virtual scada simulation system for power substation. In: 4th International Conference on Innovations in Information Technology, IIT 2007, pp. 322–326. IEEE (2007)
12. Davis, C., Tate, J., Okhravi, H., Grier, C., Overbye, T., Nicol, D.: Scada cyber security testbed development. In: Proceedings of the 38th North American power symposium (NAPS 2006), pp. 483–488 (2006)
13. Davis, J., Magrath, S.: A survey of cyber ranges and testbeds. Tech. rep, DTIC Document (2013)
14. Dondossola, G., Garrone, F., Szanto, J.: Cyber risk assessment of power control systems-a metrics weighed by attack experiments. In: 2011 IEEE Power and Energy Society General Meeting, pp. 1–9. IEEE (2011)
15. Edgar, T., Manz, D., Carroll, T.: Towards an experimental testbed facility for cyber-physical security research. In: Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, p. 53. ACM (2011)
16. Fovino, I.N., Masera, M., Guidi, L., Carpi, G.: 2010 3rd Conference on An experimental platform for assessing scada vulnerabilities and countermeasures in power plants. In: Human System Interactions (HSI), pp. 679–686. IEEE (2010)

17. Gao, H., Peng, Y., Dai, Z., Wang, T., Jia, K.: The design of ics testbed based on emulation, physical, and simulation (eps-ics testbed). In: 2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 420–423. IEEE (2013)
18. Giani, A., Karsai, G., Roosta, T., Shah, A., Sinopoli, B., Wiley, J.: A testbed for secure and robust scada systems. *ACM SIGBED Review* **5**(2), 4 (2008)
19. Gluhak, A., Krco, S., Nati, M., Pfisterer, D., Mitton, N., Razafindralambo, T.: A survey on facilities for experimental internet of things research. *IEEE Communications Magazine* **49**(11), 58–67 (2011)
20. Guglielmi, M., Nai, I., Perez-Garcia, A., Siaterlis, C.: A preliminary study of a wireless process control network using emulation testbeds. In: Chatzimisios, P., Verikoukis, C., Santamaría, I., Laddomada, M., Hoffmann, O. (eds.) *MOBILIGHT 2010. LNICST*, vol. 45, pp. 268–279. Springer, Heidelberg (2010)
21. Guo, F., Herrera, L., Alsolami, M., Li, H., Xu, P., Lu, X., Lang, A., Wang, J., Long, Z.: Design and development of a reconfigurable hybrid microgrid testbed. In: 2013 IEEE Energy Conversion Congress and Exposition (ECCE), pp. 1350–1356. IEEE (2013)
22. Hahn, A., Govindarasu, M.: An evaluation of cybersecurity assessment tools on a scada environment. In: 2011 IEEE Power and Energy Society General Meeting, pp. 1–6. IEEE (2011)
23. Hahn, A., Kregel, B., Govindarasu, M., Fitzpatrick, J., Adnan, R., Sridhar, S., Higdon, M.: Development of the powercyber scada security testbed. In: Proceedings of the Sixth Annual Workshop on cyber Security and Information Intelligence Research, p. 21. ACM (2010)
24. Haney, M., Papa, M.: A framework for the design and deployment of a scada honeynet. In: Proceedings of the 9th Annual Cyber and Information Security Research Conference, pp. 121–124. ACM (2014)
25. Harwell, S.D., Gore, C.M.: Synthetic cyber environments for training and exercising cyberspace operations. *M&S Journal*, 36–48 (2013)
26. Hieb, J., Graham, J., Patel, S.: Security enhancements for distributed control systems. In: Goetz, E., Sheno, S. (eds.) *Critical Infrastructure Protection*. IFIP, vol. 253, pp. 133–146. Springer, Boston (2008)
27. (INL), I.N.L.: Control system automated vulnerability assessment study. Tech. rep., Idaho National Laboratory (INL) (2013)
28. Jurisic, B., Holjevac, N., Morvaj, B.: Framework for designing a smart grid testbed. In: 2013 36th International Convention on Information & Communication Technology Electronics & Microelectronics (MIPRO), pp. 1247–1252. IEEE (2013)
29. Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University **33**(2004), 1–26 (2004)
30. Kush, N., Clark, A.J., Foo, E.: Smart grid test bed design and implementation (2010)
31. Leblanc, S.P., Partington, A., Chapman, I., Bernier, M.: An overview of cyber attack and computer network operations simulation. In: Proceedings of the 2011 Military Modeling & Simulation Symposium, pp. 92–100. Society for Computer Simulation International (2011)
32. Lüders, S.: Cern tests reveal security flaws with industrial network devices. *The Industrial Ethernet Book 35*(CERN-OPEN-2006-074), pp. 12–23 (2006)
33. Mallouhi, M., Al-Nashif, Y., Cox, D., Chadaga, T., Hariri, S.: A testbed for analyzing security of scada control systems (tasscs). In: 2011 IEEE PES Innovative Smart Grid Technologies (ISGT), pp. 1–7. IEEE (2011)

34. Moore, D., Murray, J., Maturana, F., Wendel, T., Loparo, K., et al.: Agent-based control of a dc microgrid. In: 2013 IEEE Energytech, pp. 1–6. IEEE (2013)
35. Morris, T., Srivastava, A., Reaves, B., Gao, W., Pavurapu, K., Reddi, R.: A control system testbed to validate critical infrastructure protection concepts. *International Journal of Critical Infrastructure Protection* **4**(2), 88–103 (2011)
36. Morris, T., Vaughn, R., Dandass, Y.S.: A testbed for scada control system cyber-security research and pedagogy. In: Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, p. 27. ACM (2011)
37. Mossberg Sonnek, K., Holm, H., Lindgren, J., Lindgren, F., Westring, E.: Foi-r-4029-se, ncs3 - informations- och styrsystem inom spårbunden trafik, en kartläggning. Tech. rep., Swedish Defence Research Agency (FOI) (2014)
38. Nanda, T.C., Chiueh, S.: A survey on virtualization technologies. RPE Report, pp. 1–42 (2005)
39. Pegden, C.D., Sadowski, R.P., Shannon, R.E.: Introduction to simulation using SIMAN. McGraw-Hill, Inc. (1995)
40. Queiroz, C., Mahmood, A., Tari, Z.: Scadasim-a framework for building scada simulations. *IEEE Transactions on Smart Grid* **2**(4), 589–597 (2011)
41. Reaves, B., Morris, T.: An open virtual testbed for industrial control system security research. *International Journal of Information Security* **11**(4), 215–229 (2012)
42. Reddi, R.M., Srivastava, A.K.: Real time test bed development for power system operation, control and cyber security. In: 2010 North American Power Symposium (NAPS), pp. 1–6. IEEE (2010)
43. Salazar, E., Macías, M.E., et al.: Virtual 3d controllable machine models for implementation of automations laboratories. In: 39th IEEE Frontiers in Education Conference, FIE 2009, pp. 1–5. IEEE (2009)
44. Sayegh, N., Chehab, A., Elhajj, I.H., Kayssi, A.: Internal security attacks on scada systems. In: 2013 Third International Conference on Communications and Information Technology (ICCIT), pp. 22–27. IEEE (2013)
45. Scarfone, K.A., Souppaya, M.P., Cody, A., Orebaugh, A.D.: Sp 800–115. technical guide to information security testing and assessment (2008)
46. Schriber, T.J.: Introduction to simulation. In: Proceedings of the 9th Conference on Winter Simulation, vol. 1, p. 23. Winter Simulation Conference (1977)
47. Shahzad, A., Musa, S., Aborujilah, A., Irfan, M.: A new cloud based supervisory control and data acquisition implementation to enhance the level of security using testbed. *Journal of Computer Science* **10**(4), 652 (2014)
48. Shahzad, A., Musa, S., Aborujilah, A., Irfan, M.: Secure cryptography testbed implementation for scada protocols security. In: 2013 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), pp. 315–320. IEEE (2013)
49. Siaterlis, C., Garcia, A.P., Genge, B.: On the use of emulab testbeds for scientifically rigorous experiments. *IEEE Communications Surveys & Tutorials* **15**(2), 929–942 (2013)
50. Siaterlis, C., Genge, B.: Cyber-physical testbeds. *Communications of the ACM* **57**(6), 64–73 (2014)
51. Stefanov, A., Liu, C.C.: Cyber-power system security in a smart grid environment. In: 2012 IEEE PES Innovative Smart Grid Technologies (ISGT), pp. 1–3. IEEE (2012)
52. Stites, J., Siraj, A., Brown, E.L.: Smart grid security educational trainingwith thundercloud: A virtual security test bed. In: Proceedings of the 2013 on InfoSecCD 2013: Information Security Curriculum Development Conference, p. 105. ACM (2013)

53. Stouffer, K., Falco, J., Scarfone, K.: Guide to industrial control systems (ics) security. NIST Special Publication **800(82)**, 16–16 (2007)
54. Suh, J., Oh, J., Choi, J., Goff, J., Tao, J., Song, E., Fu, P., Lee, G., Eom, K.: Korean r&d on the converter controller for iter ac/dc converters. In: 2011 IEEE/NPSS 24th Symposium on Fusion Engineering (SOFE), pp. 1–5. IEEE (2011)
55. Sun, Y., Ma, T., Huang, B., Xu, W., Yu, B., Zhu, Y.: Risk assessment of power system secondary devices for power grid operation. In: 2012 China International Conference on Electricity Distribution (CICED), pp. 1–5. IEEE (2012)
56. Urias, V., Van Leeuwen, B., Richardson, B.: Supervisory command and data acquisition (scada) system cyber security analysis using a live, virtual, and constructive (lvc) testbed. In: Military Communications Conference, MILCOM 2012, pp. 1–8. IEEE (2012)
57. Vaughn, R.B., Morris, T., Sitnikova, E.: Development & expansion of an industrial control system security laboratory and an international research collaboration. In: Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop, p. 18. ACM (2013)
58. Wang, Y.F., Zhang, T., Ma, Y.Y., Zhang, B.: An information security assessments framework for power control systems. In: Advanced Materials Research, vol. 805, pp. 980–984. Trans. Tech. Publ. (2013)
59. Widström, A.: Foi-r-4029-se, möjligheter och problem vid analys av fientlig kod riktad mot siemens s7-serie. Tech. rep, Swedish Defence Research Agency (FOI) (2012)
60. Xi, S., Xu, M., Lu, C., Phan, L.T., Gill, C., Sokolsky, O., Lee, I.: Real-time multi-core virtual machine scheduling in xen. In: 2014 International Conference on Embedded Software (EMSOFT), pp. 1–10. IEEE (2014)
61. Yang, Y., McLaughlin, K., Sezer, S., Littler, T., Im, E.G., Pranggono, B., Wang, H.: Multiattribute scada-specific intrusion detection system for power networks. *IEEE Transactions on Power Delivery* **29**(3), 1092–1102 (2014)
62. Yoo, S., Park, M., Yoo, C.: A step to support real-time in virtual machine. In: 6th IEEE Consumer Communications and Networking Conference, CCNC 2009, pp. 1–7. IEEE (2009)
63. Zamorano, J., De La Puente, J., et al.: Design and implementation of real-time distributed systems with the assert virtual machine. In: 2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–7. IEEE (2010)