Pains, Gains and PLCs: Ten Lessons from Building an Industrial Control Systems Testbed for Security Research

Benjamin Green, Anhtuan Le, Rob Antrobus, Utz Roedig, David Hutchison, Awais Rashid
Security Lancaster Institute
Lancaster University
United Kingdom
{*b.green2, a.le, r.antrobus1, u.roedig, d.hutchison, a.rashid*}*@lancaster.ac.uk*

Abstract

Recent years have seen a number of cyber attacks targeting Industrial Control Systems (ICSs). Reports detailing the findings from such attacks vary in detail. Hands-on experimental research is, therefore, required to better understand and explore security challenges in ICSs. However, real-world production systems are often off-limits due to the potential impact such research could have on operational processes and, in turn, safety. On the other hand, software-based simulations cannot always reflect all the potential device/system states due to over-simplified assumptions when modelling the hardware in question. As a result, laboratory-based ICS testbeds have become a key tool for research on ICS security. Development of such a testbed is a costly, labour- and time-intensive activity that must balance a range of design considerations, e.g., diversity of hardware and software platforms against scalability and complexity. Yet there is little coverage in existing literature on such design considerations, their implications and how to avoid typical pitfalls. Each group of researchers embarks on this journey from scratch, learning through a painful process of trial and error. In this paper we address this gap by reflecting on over 3 years of experience of building an extensive ICS testbed with a range of devices (e.g., PLCs, HMIs, RTUs) and software. We discuss the architecture of our testbed and reflect on our experience of addressing issues of diversity, scalability and complexity and design choices to manage trade-offs amongst these properties.

1 Introduction

Industrial Control Systems (ICSs) play an important role in the monitoring, control, and automation of critical infrastructure such as water, gas, oil, and electricity [1]. First generation (monolithic) and second generation (distributed) ICSs typically used proprietary and closed- source components and standards, with limited connectivity to non-ICS systems. In contrast, contemporary third generation (networked) ICSs frequently use open technologies, while connecting to and communicating over other (potentially non-ICS) networks [2]. This openness has come at the cost of new points of ingress and attack vectors [3]. This is evidenced by a number of attacks targeting ICS [4, 5, 6, 7]. The attack surface of future generations of ICSs is likely to increase further with developments such as Industrial Internet of Things [8].

Consequently, there is increasing interest in the security research community to study security issues in ICSs and propose effective countermeasures that have been rigorously designed and evaluated. However, such research faces two key challenges. Firstly, experimentation on real-world ICSs is hardly possible due to the inherent risks and impact of failures arising from replicating an attack. Secondly, using software simulations has several disadvantages such as not being able to reflect all the potential system states, unavailable modelling of hardware, or over-simplified assumptions about the ICS [9]. As a result, we have seen the development of physical ICS testbeds, utilising real-world devices and systems within a laboratory setting [9, 10, 11].

Development of such testbeds is a costly, labour- and time-intensive activity that must balance a range of design considerations. For instance, *diversity*, in terms of a range of devices and software, is essential to replicate real-world scenarios—recent industry reports, e.g., [12] have shown that there are more than a hundred vendors that provide hardware and communication services to ICSs hosted in 170 countries. Such diversity comes at a cost, not only in financial terms but also with regards to scalability and complexity of the experimental infrastructure. Yet there is little coverage in existing literature of first-hand experience of such design considerations, their implications and how to avoid typical pitfalls. Each group of researchers, essentially, embarks on this journey from scratch – learning through a painful process of trial and error. We address this gap by reflecting on over 3 years of experience of developing such a testbed.

The novel contributions of this paper are as follows:

- We present the testbed design and architecture, including how it supports diversity of devices, as a blueprint for future efforts in this area.
- We distill ten lessons learnt from tackling issues of diversity, scalability and complexity. This includes challenges arising from the integration of a diverse range of devices and design considerations related to Hardware-in-Loop (HIL), simulation, and virtualisation.
- We evaluate our design choices against ICS testbed functionality recommended in literature [9, 10, 13] and contrast our testbed against four other physical testbeds.

The remainder of this paper is structured as follows. Section 2 introduces typical ICS architectures and provides an overview of the devices, network and software within our testbed. Section 3 discusses lessons learnt from our experiences to date. Section 4 evaluates our design choices against functionality recommended in literature, comparing our testbed against other similar efforts. Section 5 concludes the paper.

2 Background and Testbed Architecture

2.1 Typical ICS Architecture

The implementation of ICSs varies from sector to sector, from the overarching system architectures down to individual devices and network protocols. To account for these variations, reference models are often applied to their descriptions, providing a generic platform and terminology on which further discussions can begin. Arguably the most widely adopted model is the Purdue Enterprise Reference Architecture (PERA) [14]. As can be seen in Fig. 1, an ICS can be separated into six levels across four zones [14]:

- The *Safety Zone* includes systems and devices used to manage the safety functions of an ICS.
- The *Manufacturing Zone* includes systems and devices used for the monitoring, control and automation of physical processes within an operational site. These systems and devices are geographically located in close proximity to the physical process.
- The *Demilitarised Zone* provides a "buffer zone" where data can be shared between the Manufacturing and Enterprise zones. This allows for data to move beyond the geographical constraints described in the Manufacturing Zone.
- The *Enterprise Zone* provides more conventional non-ICS specific devices and systems to utilise the data fed in from the Manufacturing Zone (via the Demilitarised Zone) to perform supervisory and planning functions across the entire ICS estate.

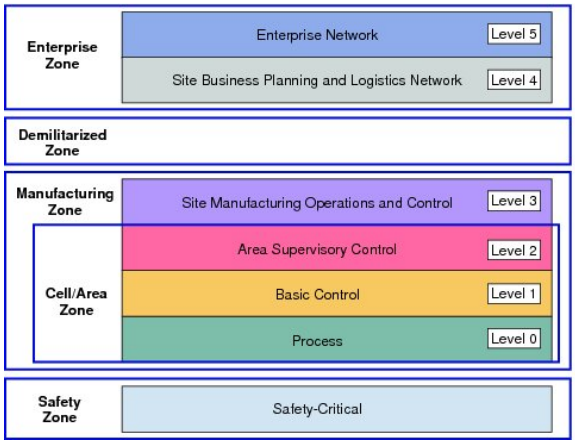

Figure 1: Purdue Enterprise Reference Architecture [14]

2.2 Architecture of Lancaster's Testbed

Fig. 2 provides a high-level view of Lancaster's ICS testbed based on the PERA reference model. Currently split across six Manufacturing Zones, an ICS Demilitarised Zone, and an Enterprise Zone (with its own separate Demilitarised Zone), all equipment in the testbed is physical (unless otherwise noted as *Virtualisation Platform* in Fig. 2). It is important to note that, within Lancaster's testbed, we have focused on the development of systems and devices across Levels 1, 2, 3, DMZ and 4. We next discuss the testbed in more detail in terms of its network architecture, the devices and the software.

2.2.1 Network

For each of the core zones, a standard private /24 block of IP addressing is applied. This decision is based on discussions with our industry partners, to fulfil existing requirements, and provide adequate network space to grow. Unique VLANs are allocated to each of these address ranges (their use is discussed in more detail in Section 2.2.3). As all the networking equipment used is physical, it resides in a data centre, interconnected with standard cat5. OSPF is used to provide a routing platform between each zone. There are exceptions to this, however, as can be seen for External Connectivity in Figure 2. Our external connectivity includes two manufacturing zones, one operating over 3G, the other over 4G or Satellite. In addition it provides an entry point for researchers external to Lancaster to utilise the testbed. The 3G, 4G and Satellite communications operate over standard private /24 blocks of IP addressing, privately routed (no public internet) via our telecommunication provider's lease-line infrastructure. Currently, external researchers are provided with a standard private /28 block of IP addresses, entering via the public internet with Cisco IPSec software clients. Networking space has been limited here to reduce the number of clients able to utilise remote connectivity at any one time.

Underpinning the network infrastructure described and depicted here is a management network. This op-

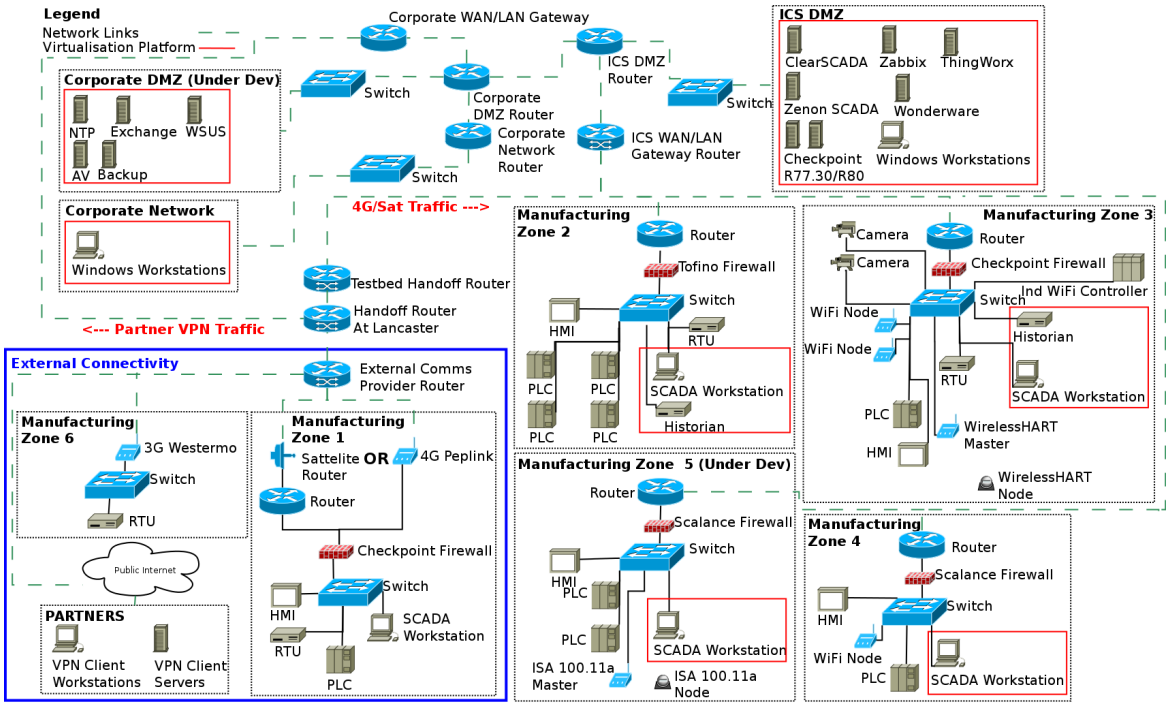

Figure 2: Network Diagram of Security Lancaster's ICS Testbed

erates over a single private /24 block of IP addressing. As with all other zones, a unique VLAN is allocated to this network (see Section 2.2.3 for more details on its use). Furthermore, all zones route via a designated switch. This provides a single point by which a VLAN Trunk can be established, providing network access to the virtualisation platform, and the testbed room (outside of the data centre) housing all physical ICS devices.

When considering the segregation and access between all networked zones, including all external connectivity, no rule sets are in place by default. This creates what could be described as a flat network. All firewall capability is placed in an ANY ANY state, allowing the flow of all traffic. While this is not how most ICSs will be configured, for a testbed it provides a starting point with no restrictions preventing systems and devices from operating. Each experiment can build up rule sets across the testbed as and when required, reverting back to the ANY ANY baseline post-completion of experimental work.

2.2.2 Devices

While physical devices can be easily re-configured to reside in any manufacturing zone, they are all designated a zone upon initial installation. This designation is derived from interactions with organisations operating ICS, and how they have designed and added to their infrastructures over time. For example, one manufacturing zone contains Siemens-only devices, another a blend of Siemens and Allen Bradley devices.

Where legacy devices have been selected, originally operating over serial based communications, these have been upgraded to IP. Again, this is a practice representative of real-world scenarios due to a range of observed benefits [15]. In addition, it reduces the complexity of initial configuration/re-configuration and compatibility within the testbed. However, should research objectives dictate the requirement for non-IP based communications, it is possible to revert this configuration, making use of legacy serial based communications.

At the lowest level of the Manufacturing Zone (Level 0) are a series of sensors and actuators. As previously noted, our focus has not been on the development of Level 0. Rather than replicate a full operational process as in [16], we have opted for a more simplistic process, designed to provide sensory data and controllable function, with little to no prerequisite knowledge for its configuration/operation. Figure 3 provide a piping and instrumentation diagram (PID) of this setup. There are four instances of this across the testbed. A standard wiring scheme is applied to all sensors and actuators, and is, therefore, interchangeable with any controller. To allow for this interchangeability, we opted for a series of RJ45 connections. To connect or disconnect a controller from one of the four sets of level 0 equipment takes a few seconds, and does not require devices to be powered down.

2.2.3 Software

Software deployed for the monitoring, control and automation of ICS is becoming more diverse and com-

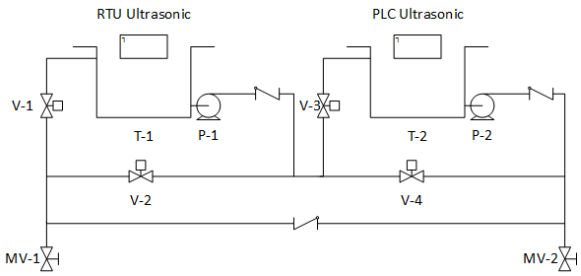

Figure 3: Level 0 PID

plex. To account for this changing landscape we opted to deploy a VMWare vSphere server. All desktop/server based software applications run inside this vSphere implementation as virtual machines. As noted above, VLANs are allocated to the various testbed zones, trunked back into this server. When building a virtual machine one can select the appropriate VLAN for its implementation (the zone in which it will reside). Thus all network traffic generated is handled via the physical network implementation without the requirement for vast numbers of physical desktops/servers. Access to these is achieved via the previously described Management Network, with every virtual machine assigned a secondary network interface for use on this network. Inside the testbed laboratory are five workstations, which reside on the management network, offering RPD or terminal access into each of the virtual machines.

The selection of relevant software – similar to selection of devices – is derived from interactions with organisations operating ICS, and how they have designed and added to their infrastructures over time. To provide a degree of future proofing in the initial stages of development we opted for KEPServerEX, providing us with support for over one hundred and fifty ICS protocols [17], meaning we are able to communicate with a vast number of ICS devices from all major vendors.

To account for the requirements posed by IIoT, rather than provide outbound connections to the public internet, we currently deploy relevant software packages inside virtual machines. This can be seen in Fig. 2 with ThingWorx and Wonderware inside the Demilitarised Zone.

3 Lessons Learnt

As noted above, an effective ICS testbed must include a diverse range of devices while providing scalability and keeping the complexity of the experimental infrastructure in check – essential requirements for ICS security research. Furthermore, where researchers may not be intune with the current state-of-the-art of such systems in a real-world context, it must be integrated as part of the baseline build. As researchers' skill sets vary, usability must also be considered. We next present the lessons learnt from our first hand experiences in realising diversity, providing scalability and managing complexity.

3.1 Realising Diversity

An effective testbed should be able to mimic a variety of ICSs setups. We faced several challenges when realising diversity in our testbed namely, how to select devices and protocols for inclusion, providing different configurations of devices/manufacturers typical in ICS settings and balancing device and protocol diversity against other requirements, such as the implementation of the physical process itself. Fig. 4 depicts a summarised (i.e. does not specify quantities/variations) view of the devices and systems currently implemented in Lancaster's testbed, highlighting existing levels of diversity.

Lesson 1: Device and technology selections should be market-driven. Our initial testbed designs were based around limited visibility of other facilities, focusing on one manufacturer's latest products. This, for example, limited the number of network protocols we were able to research and restricted any comparison between legacy and modern equipment. Over the last three years, we have worked with a wide range of industry organisations through research projects, meetings and workshops. These interactions have provided rich insights into ICS deployments from legacy, contemporary and future perspectives. Furthermore, it is important to reflect industrial practices in a testbed as, otherwise, the research will most likely be unsuitable for practical applications. Taking a *market-driven* perspective has led to our testbed implementing a wide range of industrial network protocols, these include Modbus/TCP, ISO-TSAP/S7, DNP3, OPC, EthernetIP and WirelessHART, operating on a variety of hardware and software. Example vendors include Siemens, Schneider, Westermo, and Allen Bradley. This allows us to create a range of experimental setups, mimicking a variety of typical ICS environments.

Lesson 2: Homogeneity and heterogeneity in field sites. Manufacturing zones can be homogeneous or heterogeneous. Some may deploy devices (PLCs, HMIs, etc.) from a single vendor while others may utilise and combine hardware from a range of vendors. Our manufacturing zones are, therefore, configured in a similar way. For example, Manufacturing Zone 3 is made up of Siemens-only Programmable Logic Controllers (PLCs), whereas Manufacturing Zone 2 comprises Siemens and Allen Bradley PLCs. This same discussion relates to legacy and non-legacy components, with some operators upgrading all devices simultaneously, and others upgrading as and when required. This, again, is reflected in our default Manufacturing Zone configuration. However, to account for the requirement of variations to the default setup, all controllers are located in close proximity to one another, and can be reconfigured in a matter of minutes to provide an alternative configuration, leaning towards or away from homogeneity/heterogeneity or legacy/non-legacy. Without this capability a complete and meaning-

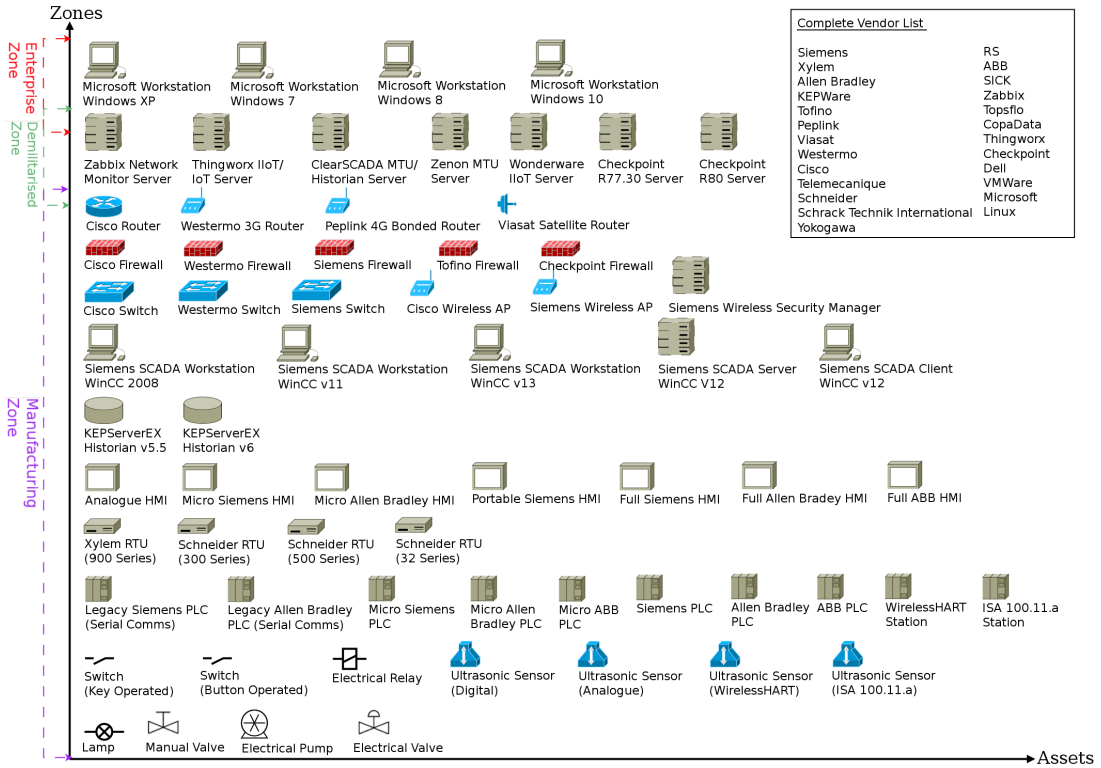

Figure 4: Overview of Devices and Software in the Security Lancaster ICS Testbed

ful security analysis may not be possible. In contrast, our original testbed setups were more static, and the ability to move devices heavily restricted based on their physical installation.

Lesson 3: Process diversity is not always crucial. Some testbeds replicate complex physical processes such as the six stages of purifying raw water, including the management of chemical dosing such as pH, chlorine, etc. [16], or the robotic assembly enclave [18]. These implementations reflect closely the processes used in real-world settings. While highly relevant to ICS security research, they represent a single process and manufacturing zone. Therefore, they cannot, for example, be easily reconfigured to represent different ICS architectures, use alternative vendors devices, mix and match legacy/non-legacy devices, etc. In contrast, we implement a simple water tank process as shown in Fig. 3. This enables us to maximise the diversity of devices and configurability of the testbed – achieved through hot-swap functionality using standardise wiring schemes – but at the expense of the ability to model stealthy attacks that exploit physical aspects of the process [19].

3.2 Providing Scalability

With the procurement of ICS hardware and software comes significant cost. Balancing cost and scalability while supporting diversity presents a significant challenge. For instance, a common method for scaling a testbed is utilising Hardware-In-the-Loop (HIL) mathematical representation for simulating the physical plants. On the other hand, software simulators such as Scilab and Scicos can be used to add to the number of field devices, while virtual machines are good for emulating the number of users/attackers [9]. Here we provide three lessons learnt around such options for scalability.

Lesson 4: Hardware-in-the-Loop (HIL) is not essential in the Manufacturing Zone. There is a lack of exact mathematical models for representing the behaviours of sensors and actuators used in monitoring and controlling the physical devices, or other factors that affect simulation accuracy such as noise [20]. We, therefore, currently discount HIL as a viable option for scalability. Furthermore, as previously discussed, process diversity is not crucial, hence our decision to use real devices. The hot-swap capability allows for a level of scalability with sensors and actuators, moving them between devices as and when required.

Lesson 5: Simulations in the Manufacturing Zone are not favoured. The replication of Manufacturing Zone devices is not our first choice. Software does not provide simulations of many essential types of devices, i.e. from different vendors or same vendor but distinctive versions, while the accuracy and reliability of such simulations in mimicking real-life operations remain an issue. Therefore, while the cost of physical equipment can be a limiting factor, the benefits it can bring in relation to experi-

mental rigour is an overriding constraint.

Lesson 6: Virtualisation and VLANs provide ease of integration and scaling. Deploying server and workstation instances across physical hardware is time consuming and costly. The use of virtualisation in conjunction with VLANs has provided us with an easy and cost effective way to integrate new systems, and scale up existing instances. The ability to deploy virtual machines and allocate network connections across all ICS zones, reduces not only the technical knowledge required when scaling up experiments, but also provides clean backups of known good systems should damage be caused during experimentation.

3.3 Managing Complexity

In search of diversity and scalability, one inevitably reaches a point where general testbed management becomes a challenge in itself. This can be exacerbated by the goals of the research and the researcher's experience. To begin operating and re-configuring various aspects of the testbed requires some basic knowledge, for instance, which configuration software packages are required. Where more complex research goals are proposed, additional devices may be required. For instance, capturing network traffic centrally across all manufacturing zones would require the configuration and implementation of additional network components. Here we provide four lessons learnt from managing this complexity, based on experience of experimental work utilising the testbed, examples of which can be seen in [21, 22, 23, 24]. These works also provide insights into the type of experimentation facilitated by the testbed.

Lesson 7: Employ a Management Network. The complexity of an experimental layer can be reduced by providing one central point by which connections to all software based applications and research tools can be achieved. Giving researchers a single network (called a Management Network in our case) on which all activities can be conducted reduces pre-requisite knowledge. However, it relies on all required research tools being in place within the existing infrastructure. This is a challenge we are currently facing and have begun to scale up with the inclusion of a more diverse set of virtual machines, including popular security-focused Linux distributions (e.g., Kali and SamuraiSTFU). Furthermore, the inclusion of appropriate data capture points also needs to be considered. For example, we lack the capacity to capture network traffic from every network zone into one centralised location. This is currently being addressed through mirrored ports trunked back into the server, with an appropriate service applied to its collection and analysis residing within the Management Network.

Lesson 8: Setup Multiple Manufacturing Zones. The concept of dividing devices up into discrete Manufacturing Zones was never intended as a means to tackle complexity, more to replicate real-world scenarios as previously discussed. However, it has proven valuable in the context of concurrent research activities. Discrete separation, mixed with the ability to quickly duplicate virtual machines anywhere in the network, means researchers can often conduct their activities simultaneously (dependent upon their objectives), without disrupting others. The ability to move devices in/out of manufacturing zones can also aid in their management, with non-relevant devices quickly removed during research.

Lesson 9: Comprehensively document as you build. Whilst a somewhat obvious lesson, this is nevertheless critical. Our testbed was built organically and initially utilised by those designing and implementing it. Hence, they had comprehensive knowledge of the infrastructure. However, as the number of users grew – with undergraduate, Master's as well as external collaborators utilising the testbed – this presented a major challenge. We had to go through a laborious and painful process of documenting the system – something which could have been avoided had we documented systematically as we built. We not only documented the communication and control processes within the testbed but also all known vulnerabilities in the devices and software currently deployed. This saves new researchers considerable time in understanding possible attack vectors, and composing plausable scenarios. However, keeping this documentation up-to-date is an on-going and substantial effort, one others building testbeds may want to explicitly budget.

Lesson 10: Optimise data logging for security purposes. Open ICS testbed datasets [25] often record and offer as many features as they can. Currently the collection and distribution of data from our testbed is limited as it involves a manual process requiring time and resource. One of our previous works [23] discusses the requirement for granular data flows and human-device interaction points, as a prerequisite to security control selection and implementation. Principles derived from this work were validated when applied in intrusion detection scenarios [21], which reduced the number of logged variables to just 3, yet still achieve high detection accuracy of 99% for passive attacks. However, such understanding cannot be obtained automatically in our current testbed design, something future developments must look to incorporate.

4 Evaluating Testbed against Research Criteria

The previous sections have described Lancaster's testbed, along with lessons learnt throughout its evolution. Here we offer a comparison against four other existing testbed infrastructures [16, 18, 26, 27]. Note that, while numerous virtualised ICS testbeds exist, e.g., [28,

29], we limit our comparison to a subset of similar physical testbeds reported in literature. We analyse their, and our own, compliance against a set of recommended testbed functionalities described in literature [9, 10, 13]. These works recommend ten categories of functionality for a testbed: (1) Physical device diversity (PD): Supports a wide range of physical devices; (2) Industrial protocol diversity (ID): Supports a wide range of industrial communication protocols; (3) Process diversity (PC): Supports more than one type of physical operational process; (4) Flexibility (FX): Supports multiple configurations; (5) Scalability (SC): Replicates the scale of the ICSs when needed; (6) Fidelity (FD): Mimics as close and accurate as possible a real ICS; (7) Simulation Support (SS): Offers simulations for field devices or process; (8) Software to support security analysis (SA): e.g., parsing tools for sniffed packets; (9) Optimisation for monitoring (OM): Supports optimising data logging to reduce the impact of security on general operation; (10) Openness (OP): Supports remote access or data openness.

Table 1 presents the results of our analysis. Black dots represent that a testbed supports a given function while grey dots represent that it acknowledges the future need for such a function. Our design decisions driven by the need to realise diversity while providing scalability and managing complexity enable our testbed to deliver on eight out of ten recommended functionalities. We note, however, that openness is partially supported in that we provide remote access for other researchers. The automation of collection and distribution of data from large-scale experiments is currently limited. Lack of process diversity and simulation are less of an issue as these are conscious design choices. Optimisation of data logging will, therefore, be a major goal for the future.

Table 1: Comparison of functions

Work	PD	ID	PC	FX	SC	FD	SS	SA	OM	OP
[16]	●	●	●	●		●		●		
[18]	●	●	●	●		●	●	●		
[25]	●	●	●	●		●	●	●		
[26]	●	●	●			●		●		
LAN	●	●		●	●	●		●	●	●

5 Conclusion

This paper discussed the ten most important lessons learned during the development of the Lancaster testbed for ICS security research. It can be seen that assuring diversity and scalability are crucial for reflecting a real-world ICS closely for a reliable study, while managing complexity is essential for research efficiency. Through the brief evaluation, we have demonstrated current benefits and limitations against other testbed implementations. We note that making the testbed more open for researchers external to Lancaster University, and therefore extending its usability, is of high importance. Activities including local access and demo capabilities (via a mobile demo unit), or remote access via VPN, are ongoing, while other targets such as maintaining logged data for forensic study and offering a platform for global all-inclusive connectivity will be addressed in our future work. Finally, we note that the testbed is also being used for research into the resilience of ICS and utility networks, investigating the detection and mitigation of intrusions across various layers of the infrastructure.

Acknowledgement. This work is partly supported by EPSRC/CHIST-ERA grant DYPOSIT: Dynamic Policies for Shared Cyber-Physical Infrastructures under Attack (EP/N021657/1).

References

[1] Eric D Knapp and Joel Thomas Langill. *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems*. Syngress, 2014.

[2] Hosny A Abbas. Future SCADA challenges and the promising solution: the agent-based SCADA. *International journal of critical infrastructures*, 10(3-4):307–333, 2014.

[3] Thomas Kropp. System threats and vulnerabilities [power system protection]. *IEEE Power and Energy Magazine*, 4(2):46–50, 2006.

[4] Bill Miller and Dale Rowe. A Survey SCADA of and Critical Infrastructure Incidents. In *Proc. 1st Annual Conference on Research in Information Technology*, RIIT '12, pages 51–56. ACM, 2012.

[5] Kelly Jackson Higgins. Latest Ukraine Blackout Tied To 2015 Cyberattackers, 2017.

[6] Robert M Lee, Michael J Assante, and Tim Conway. Analysis of the Cyber Attack on the Ukrainian Power Grid. Technical report, SANS, 2016.

[7] Robert M Lee, Michael J Assante, and Tim Conway. German Steel Mill Cyber Attack. Technical report, SANS, 2014.

[8] Amin Hassanzadeh, Shimon Modi, and Shaan Mulchandani. Towards effective security control assignment in the Industrial Internet of Things. In *2nd World Forum on IoT*, pages 795–800, 2015.

[9] Stephen McLaughlin, Charalambos Konstantinou, Xueyang Wang, Lucas Davi, Ahmad-Reza Sadeghi, Michail Maniatakos, and Ramesh Karri. The cybersecurity landscape in industrial control systems. *Proceedings of the IEEE*, 104(5):1039–1057, 2016.

[10] Hannes Holm, Martin Karresand, Arne Vidström, and Erik Westring. A survey of industrial control system testbeds. In *Secure IT Systems*, pages 11–26. Springer, 2015.

[11] William Knowles, Daniel Prince, David Hutchison, Jules Ferdinand Pagna Disso, and Kevin Jones. A survey of cyber security management in industrial control systems. *International journal of critical infrastructure protection*, 9:52–80, 2015.

[12] Oxana Andreeva, Sergey Gordeychik, Gleb Gritsai, Olga Kochetova, Evgeniya Potseluevskaya, Sergey I. Sidorov, and Alexander A. Timorin. Industrial control systems and their online availability. Report, 2016.

[13] BéLa Genge, Christos Siaterlis, Igor Nai Fovino, and Marcelo Masera. A cyber-physical experimentation environment for the security analysis of networked industrial control systems. *Computers & Electrical Engineering*, 38(5):1146–1161, 2012.

[14] Paul Didier, Fernando Macias, James Harstad, Rick Antholine, Scott A Johnston, Sabina Piyevsky, Mark Schillace, Gregory Wilcox, Dan Zaniewski, and S Zuponcic. Converged plantwide ethernet (cpwe) design and implementation guide. *Cisco Systems and Rockwell Automation*, 2011.

[15] Rene Midence, Roger Moore, and Glenn Allen. The migration of serial to Ethernet communications-Why bother? In *20th Int'l Conf. on Electricity Distribution-Part 1 (CIRED)*, pages 1–4. IET, 2009.

[16] Aditya P Mathur and Nils Ole Tippenhauer. Swat: a water treatment testbed for research and training on ics security. In *Cyber-physical Systems for Smart Water Networks (CySWater), 2016 International Workshop on*, pages 31–36. IEEE, 2016.

[17] PTC Inc. KEPServerEX - Solving Your Communications Challenges, 2017.

[18] Richard Candell, Timothy Zimmerman, and Keith Stouffer. An industrial control system cybersecurity performance testbed. *National Institute of Standards and Technology. NISTIR*, 8089, 2015.

[19] David I Urbina, Jairo A Giraldo, Alvaro A Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. Limiting the impact of stealthy attacks on industrial control systems. In *Proc. ACM SIGSAC Conference on Computer and Communications Security*, pages 1092–1105. ACM, 2016.

[20] E. J. M. Colbert and A. Kott. *Cyber-security of SCADA and Other Industrial Control Systems*. Advances in Information Security. Springer, 2016.

[21] William Jardine, Sylvain Frey, Benjamin Green, and Awais Rashid. Senami: Selective non-invasive active monitoring for ics intrusion detection. In *Proc. 2nd ACM Workshop on Cyber-Physical Systems Security & Privacy*, pages 23–34. ACM, 2016.

[22] Rob Antrobus, Sylvain Frey, Benjamin Green, and Awais Rashid. Simaticscan: Towards a specialised vulnerability scanner for industrial control systems. In *Proceedings of the 4th International Symposium for ICS & SCADA Cyber Security Research 2016*. BCS Learning & Development Ltd., 2016.

[23] Benjamin Green, Marina Krotofil, and David Hutchison. Achieving ics resilience and security through granular data flow management. In *Proc. 2nd ACM Workshop on Cyber-Physical Systems Security & Privacy*, pages 93–101. ACM, 2016.

[24] Jeremy Simon Busby, Benjamin Green, and David Hutchison. Analysis of Affordance, Time, and Adaptation in the Assessment of Industrial Control System Cybersecurity Risk. *Risk Analysis*, 2017.

[25] Wei Gao, Thomas Morris, Bradley Reaves, and Drew Richey. On scada control system command and response injection and intrusion detection. In *eCrime Researchers Summit (eCrime), 2010*, pages 1–9. IEEE, 2010.

[26] Thomas Morris, Anurag Srivastava, Bradley Reaves, Wei Gao, Kalyan Pavurapu, and Ram Reddi. A control system testbed to validate critical infrastructure protection concepts. *International Journal of Critical Infrastructure Protection*, 4(2):88–103, 2011.

[27] Igor Nai Fovino, Marcelo Masera, Luca Guidi, and Giorgio Carpi. An experimental platform for assessing scada vulnerabilities and countermeasures in power plants. In *3rd Conf. on Human System Interactions (HSI)*, pages 679–686. IEEE, 2010.

[28] David C. Bergman, Dong Jin, David M. Nicol, and Tim Yardley. The virtual power system testbed and inter-testbed integration. In *USENIX Workshop on Cyber Security Experimentation and Test*, 2009.

[29] Antoine Lemay, José M. Fernandez, and Scott Knight. An isolated virtual cluster for SCADA network security research. In *1st International Symposium for ICS & SCADA Cyber Security Research, ICS-CSR*, 2013.