Oops I Did it Again: Further Adventures in the Land of ICS Security Testbeds

Joseph Gardiner
Bristol Cyber Security Group, University of Bristol
Bristol, UK
joe.gardiner@bristol.ac.uk

Barnaby Craggs
Bristol Cyber Security Group, University of Bristol
Bristol, UK
barney.craggs@bristol.ac.uk

Benjamin Green
Security Lancaster Institute, Lancaster University
Lancaster, UK
b.green2@lancaster.ac.uk

Awais Rashid
Bristol Cyber Security Group, University of Bristol
Bristol, UK
awais.rashid@bristol.ac.uk

ABSTRACT

Research efforts in the security of Industrial Control Systems (ICS) have dramatically increased over the past few years. However, there is a limiting factor when work cannot be evaluated on real-world systems due to safety and operational reasons. This has led to multiple deployments of ICS testbeds covering multiple sectors including water treatment, power distribution and transportation networks.

Over the last five years, we have designed and constructed ICS testbeds to support cyber security research. Our prior work in building testbeds culminated in a set of design principles and lessons learnt, formulated to support other researchers in the design and build of their own ICS testbeds. In the last two years we have taken these lessons and used them to guide our own greenfield large-scale, complex and diverse process security testbed affording a rare opportunity to design and build from the ground-up—one in which are have been able to look back and validate those past lessons and principles.

In this work we describe the process of building our new ICS and IIoT testbed, and give an overview of its architecture. We then reflect on our past lessons, and contribute five previously unrecognised additional lessons based this experience.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Industrial Control Systems, ICS, SCADA, Operational Technology, OT, IIoT, CPS, Cyber Security, Testbeds

CPS-SPC '19, November 11, 2019, London, UK

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

ACM Reference Format:
Joseph Gardiner, Barnaby Craggs, Benjamin Green, and Awais Rashid. 2019. Oops I Did it Again: Further Adventures in the Land of ICS Security Testbeds. In *CPS-SPC '19: ACM Workshop on Cyber-Physical Systems Security & Privacy, November 11, 2019, London, UK*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Industrial Control Systems (ICS) form the backbone of modern day infrastructure, responsible for the delivery of services considered critical from a societal perspective [3]. Due to their criticality, the EU recently imposed new legislation in the form of the Network and Information Systems directive (NIS), mandating operators of critical national infrastructure (CNI) conform to a set of baseline principles. This acknowledgement of the threat posed to ICSs from a cyber security perspective comes after several years of high-profile attacks [15, 17], and an increasing number of identified vulnerabilities in common components and software [16].

The National Cyber Security Centre (NCSC), established by the UK government, for example, currently advises operators on their journey towards NIS compliance [1]. This is where academia can play the strongest role, conducting research to provide feedback into regulations and associated guidance, enhancing operators' capability to defend against attacks. However, this does not come without its challenges. Due to the critical nature of these systems access is highly restricted, presenting a roadblock when one seeks to engage in practical research activities [12]. As information surrounding the infrastructure can be deemed highly sensitive, access could be forbidden. As a system failure could have catastrophic impact, deployment of experimental infrastructure into live systems is not acceptable without extensive prior evaluation. This forms a key requirement for the use of testbeds, supporting practical research within a safe, controlled environment.

Over the last five years, we have designed and constructed ICS testbeds to support cyber security research. Our initial concepts, built out of Lancaster University [10], formed a starting point for the exploration of vulnerability scanners [1], intrusion detection systems [14], process comprehension [11], etc. This culminated in a set of design principles and lessons learnt, formulated to support other researchers in the design and build of their own ICS testbeds [12].

[1] https://www.ncsc.gov.uk/collection/nis-directive

Our work has progressed across the last two years, developing a greenfield testbed at the University of Bristol. This new facility has heeded advice from our existing design principles and lessons learnt, resulting in the rapid deployment of familiar technologies, and additional expansion towards the construct of a comprehensive resource pool.

The Bristol testbed is far larger in scale than our previous efforts. Rather than focus on a single physical process as before we incorporate multiple physical processes (see Section 3), and the associated physical infrastructure. This is backed up by a training and prototyping setup, as well as a mobile demonstration unit which also can be remotely integrated to form part of the main testbed. The testbed has also dramatically increased in complexity, with the addition of technologies to explore issues of convergence between operational technology (OT) and information technology (IT), and also where industrial internet of things (IIoT) and building management systems (BMS) interplay with traditional OT.

Building this second testbed from the ground up has provided us with a rare opportunity to evaluate our previous design principles and lessons. Throughout this process we also identified a set of new lessons, and extension of existing design principles. This paper revisits our existing work and highlights where lessons remain valid, where they are now considered less critical, and which (in practice) are a challenge to follow. We show that diversity, scalability and complexity are key principles for a testbed. However, data capture and safety are a matter of context. Through the combination of our old and new lessons we hope that groups who embark on their own ICS testbed projects can gain from our experience and this blueprint for building testbeds.

The remainder of this paper is structured as follows. Section 2 provides an overview of our research aims and design requirements for the testbed. Section 3 provides an overview of the new testbed. Section 4 revisits our existing lessons learnt, reflecting upon them with our recent experiences. Section 5 proposes a set of additional lessons through the design and build of our new testbed. Section 6 discusses a deployed use case within the new testbed and provides reference back to key design features of testbed and where our latest explorations fit within that landscape.

2 MOTIVATION

Our previous work set out a preference and justification for the development of large-scale physical testbed infrastructure that is capable of closely replicating real-world scenarios [12]. Whilst approaches that adopt the use of device and system simulation exist, ultimately they are limited in terms of the credibility offered during a wider range of research activities, and often result in the interconnection of physical devices into the simulated environment [7]. To highlight the possible research areas one may seek to explore within an ICS context, we present a set of use cases. From these we derive core testbed requirement/design principles, subtly extending viewpoints comprehensively described in our earlier work [10, 12].

2.1 Use Cases

The following use cases have been highlighted as areas in need of further study and have been derived through engagement with government, industry, and the academic community. This is not an exhaustive list, however we focus on these five areas. Together these afford a high-level viewpoint from which testbed requirements/design principles can be formed.

Convergence. One of the key areas that requires further examination is the convergence of ICS/OT technology with other instances of OT, as well as the convergence with other technologies such as Industrial Internet of Things (IIoT). For example, what happens when a physical process is located alongside a building management system, which is more likely to be connected to the internet, or insecure off-the-shelf IoT devices are used? Introducing IIoT devices and concepts into the OT environment can also introduce new methods of exploitation, which need to be explored.

Security Analysis of ICS Devices. Security vulnerabilities are constantly being found within ICS devices, and with new devices and protocols being released to the market the need to be studied to identity any potential unknown vulnerabilities.

Intrusion Detection. Intrusion detection with ICS environments is still an ongoing area of research. There is a large amount of scope for exploration of novel approaches to intrusion detection within ICS environments, in particular in converged environments wherein behaviour may be more complex and, hitherto, unknown.

Dataset Generation. There are limited datasets available for ICS security research, especially at scale. A particular goal of ours is to produce datasets that can be made public, including attacks against the ICS environments. As configuring testbeds for data collection can be a time intensive process, we aim to keep this as straightforward as possible and have designed this in from the start.

Human Factors. Further exploration is needed in the area of human factors in ICS security. This both includes how operators react under pressure and the implications that can have for security, as well as looking to build novel interfaces for security-related interactions with ICS equipment.

2.2 Requirements for an IIoT and ICS testbed

Using the aforementioned use cases and our existing work [6, 10, 12] as a base, the following five high-level design principles/requirements are formed, supporting core research challenges.

Requirement 1: Diversity. The testbed should contain a range of devices and software, from multiple manufacturers, covering both legacy and non-legacy deployments. This allows the testbed to replicate a variety of real-world deployments—where organic growth is the norm rather than greenfield deployment—with a high degree of accuracy.

Requirement 2: Scalability. With diversity of equipment, including both physical processes and control devices, comes both a monetary and time cost. ICS devices are expensive, and it can be a time intensive process to install and configure new devices. On the other hand, building to scales similar to those found in industry can prove useful for experimentation. The testbed should both be able to support multiple devices and processes at scale, but also provide methods to increase scalability with a reduced cost, for example through simulation and virtualisation.

Requirement 3: Complexity. With increased scale and diversity comes complexity, both in managing the testbed, and deploying experiments. This is amplified by the requirement of specialist knowledge for many aspects of working with ICS equipment, such as logic programming and OT specific communication protocols. Measures should be taken to reduce such complexity for both testbed maintainers, and researchers.

Requirement 4: Data capture. The testbed should be capable of appropriate data capture for experimentation, including when the system is under (simulated) attack.

Requirement 5: Safety. The testbed should be designed such that it poses minimal risk in terms of safety to researchers and engineers. As well as safety of the individuals, the testbed itself should be safe from outside influence, including unwarranted attack.

3 TESTBED OVERVIEW

Next we provide an overview of our testbed infrastructure. Figure 1 offers a high-level view of the testbed architecture, this can be used as a reference point in the identification of critical components and their position within the testbed as a whole. Each core category is tied back to use cases and associated testbed requirements outlined in Section 2.1, affording clear links between category attributes and the requirements they support. These links are summarised in Table 1.

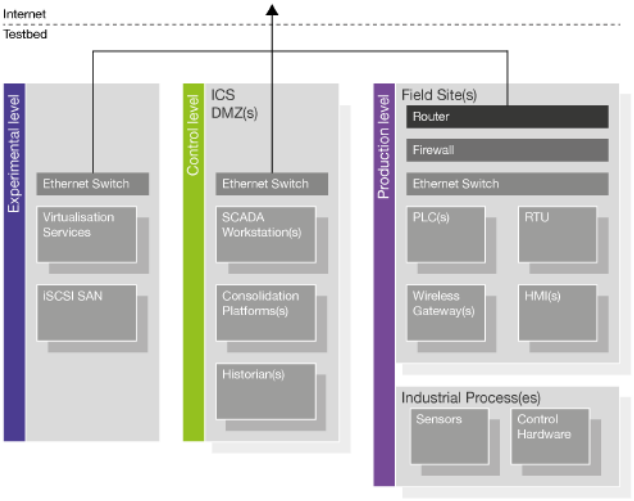


Figure 1: The testbed as a subset of the Reference Architecture for IIoT and Industrial Control Systems Testbeds [6]

3.1 Physical processes

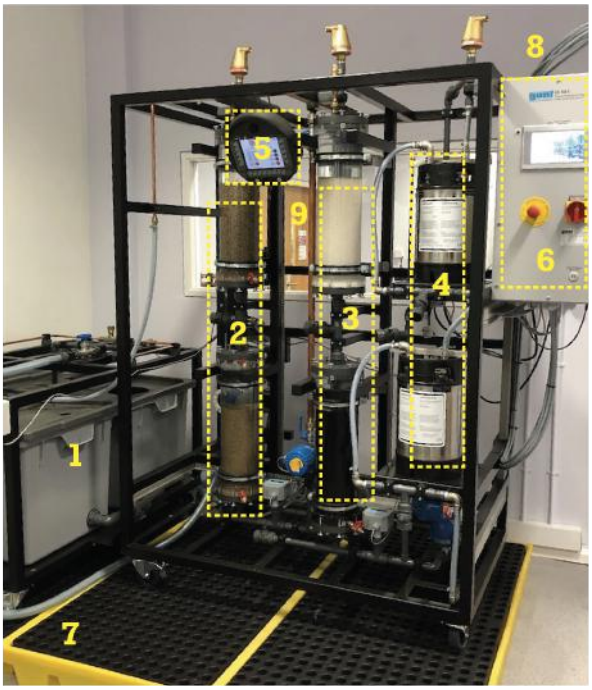
3.1.1 Water treatment plant. We integrated an off-the-shelf water treatment training rig from Gunt [2], a German manufacturer of industrial training equipment, into our testbed. The Gunt CE-581 water treatment plant consists of a three stage filtration, absorption and ion exchange process, used to deliver a training program focused on the chemical processes of cleaning and testing water. The rig uses a single pump to deliver *dirty* water through the system, four electro-mechanical valves for selecting which of the three stages are utilised, and a range of digital and analogue sensors including flow rate, temperature and pressure sensors. Our deployment of the Gunt CE-581 was customised by the manufacturer to our specifications for security experimentation. We had a removable copper pipe inserted into the system to allow for the easy inclusion of additional sensors, and a further safety overflow system added to the process in case of over-pressurisation. The CE-581 allows us to meet a number of our requirements - due to its multi-stage process it meets our complexity and scalability requirements (and can be expanded with new sensors through our modification), and supports diversity in terms of both the control architecture and process itself (with multiple sensor and control outputs to consider). The water treatment plant, in it's entirety, can be used across all five of our use cases.

The Gunt CE-581 water treatment plant comes with a control cabinet controlled by an Eaton programmable logic controller (PLC). This we replaced this with a custom-built control board, which can be seen in Figure 3. In our testbed, the primary control PLC for the CE-581 is a Siemens S7-1500 coupled with a Siemens ET-200S PLC for pump control, representing further diversity and complexity more akin to real-world deployment. The PLCs are connected to the water rig by custom cable harnesses routed through swappable terminal blocks within the water rig control cabinet. The swappable terminal blocks allow us to revert the CE-581 back to the original control system for maintenance. The board also features a Schneider ScadaPack32 remote terminal unit (RTU). A more detailed description of the control board design in provided in Section 3.2. Networking on the board is provided by an 16 port Westermo industrial layer 2 managed routable ethernet switch, complimented with a Checkpoint firewall which can act both as a firewall and as a data tap within the field site, as shown in Figure 4.

	Use cases					Requirements				
Testbed aspect	Convergence	Sec analysis	IDS	Datasets	Human factors	Diversity	Scalability	Complexity	Data capture	Safety
Water plant		✓	✓	✓	✓	✓	✓	✓	✓	✓
Factory		✓	✓	✓		✓	✓		✓	
BMS	✓	✓	✓	✓	✓	✓	✓		✓	
Control board						✓	✓	✓		✓
Comms	✓		✓	✓		✓	✓	✓	✓	✓
Security					✓	✓	✓	✓		✓
Software	✓		✓	✓	✓	✓	✓	✓	✓	✓
Industrial IIoT	✓	✓		✓	✓	✓	✓	✓	✓	
Mobile testbed	✓	✓			✓	✓	✓	✓	✓	
Training and prototyping	✓	✓			✓	✓	✓	✓		✓

Table 1: Summary of testbed aspects and how they relate to use cases and requirements

[2] https://www.gunt.de/en/products/process-engineering/water-treatment/multistage-water-treatment/water-treatment-plant-1/083.58100/ce581/glct-1:pa-148:ca-255:pr-57



Key:
1 Input (dirty) & output (clean) water tanks
2 Filtration tanks
3 Absorption tanks
4 De-ionisation tanks
5 Wireless HMI
6 Original control panel, replaced by field site board
7 Safety bunds
8 IO cabling to field site board

Figure 2: Water treatment process

3.1.2 Model factory. Our second process is a model factory from Fishertechnik [3]. Designed to train ICS engineers, the factory consists of four highly interconnected and dependant processes - picking, processing, sorting and storing. The factory contains a large amount of both analogue and digital interfaces (IO), with multiple sensors and motors to control. This presents a high degree of complexity in synchronising the four processes, allowing us to run experimentation on a highly complex and large scale process whilst being deployable within a lab by virtue of being physically small enough to fit on a table.

3.1.3 Building management system (BMS). To study further convergence issues, our building management system was custom designed and built for the lab. The BMS is of a dual interconnected PLC design consisting of a primary controller, the Trend IQ4e, which is connected to a smaller sub-field site Trend IQ3 PLC. Together these represent both current and legacy deployments, again as one is likely to find in the real world. The main PLC cabinet contains a number of controls, environmental sensors and actuators to represent typical mechanical & engineering (M&E) scenarios such as heating, cooling, lighting etc. Additionally, this cabinet also contains a number of gateway devices, such as the North Commander and a Phillips Hue Bridge, to allow for the onward deployment of both commercial and consumer IoT devices as one might find in evolving commercial settings. The sub-field site provides additional sensor / actuator space in a controlled environment. As Trend Controls limit programming and maintenance access to their PLCs in live deployments, our deployment is made in a similar manner such that the PLCs themselves can be viewed as *black boxes*. This is more representative of how they would be deployed in the wild.

[3]https://www.fischertechnik.de/en/products/simulating/training-models/536634-sim-factory-simulation-24v-simulation

3.2 Control board design

For each field site, we make use of a standardised control board to which all of the ICS devices for that site are mounted. An example of this board, used to control the water treatment plant, can be seen in Figure 3. As well as the ICS devices themselves, the board also features 24 volt power distribution and networking.

The board is designed in such a way that the PLCs are mounted on removable plates, and connected to removable terminal blocks on the board rather than directly to the process. All wires are individually numbered, and documented, allowing for the relatively easy swapping of PLCs on the board. The board is designed to hold 2 larger PLCs, 2 RTUs and associated equipment, though could hold multiple smaller devices if required. This design result in meeting our complexity & diversity requirement.

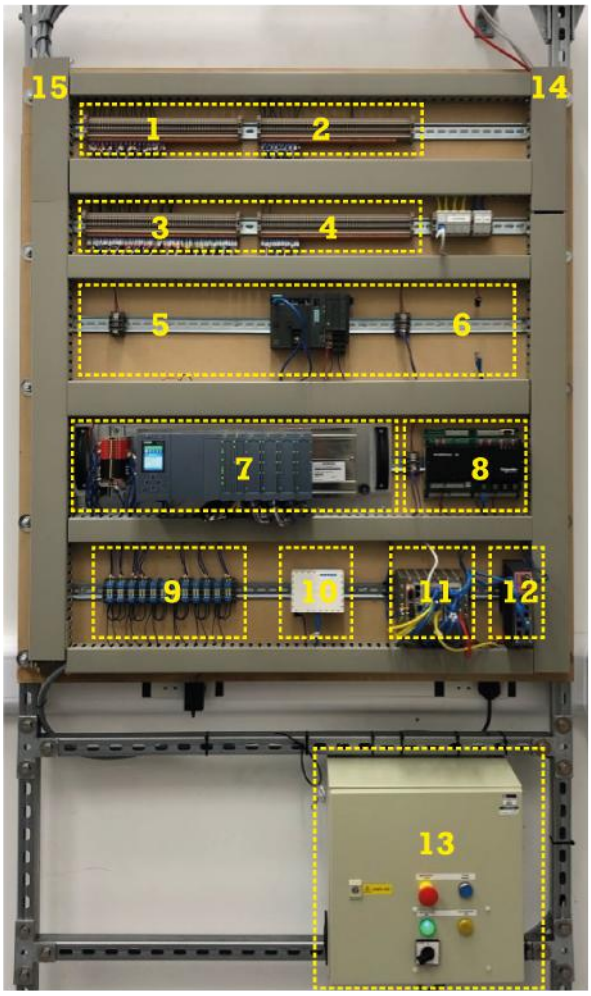
To meet safety requirements, the 240VAC to 24VDC power supplies for the board are housed inside a secured box below the control board. This box also feature an emergency stop button to shut power down to the board. By isolating the 240v supply, all exposed wiring on the control board is limited to a safe 24v. As the control boards are located within a secured room with limited access, the board can remain open rather than in a closed cabinet ala the main BMS.

We currently have one board in operation, and are in the process of building two further boards utilising the same design, the first to cater for the factory simulator, and then further physical processes.

3.3 Communication and networking

The design of the network is largely the same as in our previous testbed [12]. The network is split into three parts: OT, IT and Experimental. The architecture of this network can be seen in Figure 4. Each network is allocated a /24 address space, as well as a virtual local area network (VLAN). VLANS are connected to our virtualisation server (see Section 3.5), allowing us to easily connect different virtual machines to the various networks. The network is designed to be scalable, whilst minimising the complexity of configuring the network for experimentation. As well as providing networking for ICS devices, the network is also configured to allow us to connect other types of devices, such as our BMS system and IIoT devices, to facilitate convergence use cases.

Our testbed network is isolated from the wider university network to provide minimal chance of outside influence (for example, through attack). The only connection to the outside world is through a managed virtual private network (VPN) router which



Key:

1 & 2	Digital Inputs\Outputs (32 each)
3 & 4	Analogue Inputs\Outputs (16 each)
5 & 6	Secondary PLC\RTU Housing
7	Primary Programmable Logic Controller (PLC)
8	Primary Remote Terminal/Telemetry Unit
9	24VDC Distribution
10	WiFi Access Point
11	L3 Managed Ethernet Switch
12	Firewall
13	240V AC to 24V DC Power Supplies
14	Ethernet Back-haul to Core Network Infrastructure
15	IO Cabling to Physical Process

Figure 3: Field Site Control Board

sits on the gateway and only allows authorised partners access to the testbed. Similarly, a 4G connection from the mobile testbed can be tunnelled through the gateway in order to link the mobile setup into the main testbed network as it's own field site. This external connection allows for those partners to access varying levels of the testbed—from a single process through to the entire system as required for experimentation. It also caters for the federation of other testbeds into ours, thus providing another route for extensibility and complexity.

In order to facilitate data capture, every VLAN has a spanning port allowing us to capture full network data from any part of both the IT and OT networks.

Whilst most of the network is contained within a single server rack within a secured room, the management and security operations centre (SOC) networks are physically routed into our operations centre. SOC machines are directly connected to this network, within which there is a managed ethernet switch allowing for machines to be routed directly to different VLANs as required.

We also maintain a discrete experimental wireless network, separated from our testbed network, which is granted direct access to the wider internet through the university network. This network is for working with IoT devices which cannot be connected to the main testbed network, or to the main university network, but require an active internet connection.

3.3.1 Software-defined networking (SDN). We are currently in the process of building a software-defined version of our testbed network both in a physical setup utilising commercial SDN switches, as well as a fully virtual environment using OpenVSwitch[4] virtual switches. The physical switches support the Openflow SDN protocol, and so can be used with many different controller architectures. This will allow us to explore the potential security impact of introducing software-defined networking into OT / IT environments.

3.4 Security (Cyber and Physical)

As part of our safety requirement, we require controls to be put in place around the testbed environment to minimise risk of damage to the testbed, or indeed those operating it.

The testbed network is isolated from the wider university network. This allows the testbed to run with a lessoned risk of security breach from the internet or elsewhere within the university (and prevents attacks from leaking out of the testbed), however has the disadvantage that machines within the network do not have internet access (which makes updating software a difficult process). External access is through a certificate based VPN connection.

Access to virtual machines located within the testbed is managed by Active Directory (AD), with different levels of access based on requirement. Testbed admins have full administrator rights across all machines, whilst other researchers who need to simply access machines have standard user access.

All network points into the testbed are located within physically secured, and restricted access, rooms. For devices that sit within the communal areas (such as the water treatment plant and BMS system), IO and networking cables are routed to inside the process cabinets which remain locked. This minimises the risk of unauthorised devices being connected to the network.

In order to prevent contamination of the testbed, we operate a *"clean"* machine policy for connecting to the testbed. Apart from the dedicated SOC desktop machines, only a small number of designated maintenance laptops are connected to the testbed through the

[4] http://www.openvswitch.org

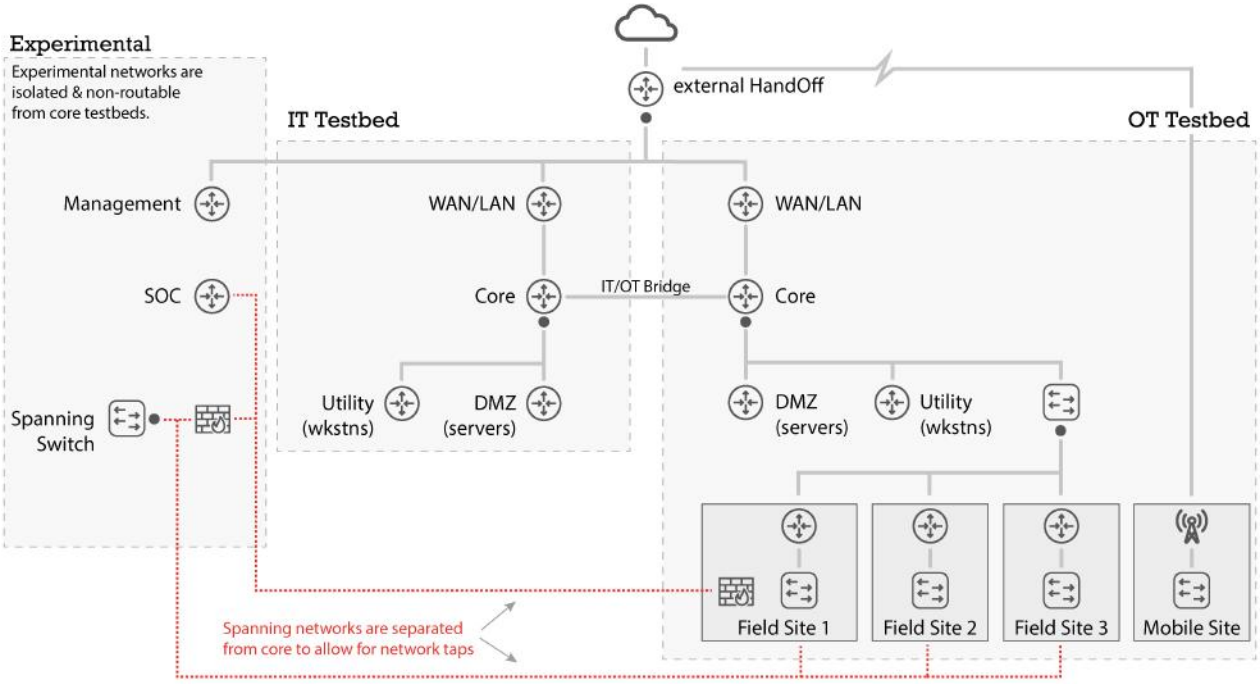


Figure 4: Testbed network diagram

management network. These machines are regularly wiped clean, and only connected to the internet when absolutely necessary. Live bridging between the testbed and internet accessible networks is not permitted.

3.5 Software

The software deployment within the testbed is critical for all of our use cases and testbed requirements, and broadly splits into one of two categories. The first is software used for testbed operations (the experimental level within the architecture presented in Figure 1, including applications for security and data collection. The second category is software that would be considered part of the operational environment either at the control or production levels of the architecture.

To ease maintenance and deployment, all software is installed into virtual machines running inside VMWare vSphere. As well as specialist ICS software, such as ClearSCADA, we also maintain virtual machines with other useful software installed. For example, we utilise a Windows virtual machine, connected to the OT LAN VLAN, containing all the software necessary for configuring Siemens devices.

The virtualisation server includes multiple base VM images, covering multiple windows and Linux variants, which can be used when building new virtual machines on the testbed. Wherever possible, software in installed by mounting clean USB sticks containing installation files into the virtual machines. Where installing updating software or operating system is only possible with an active internet connection, VMs are downloaded to one of our maintenance laptops, disconnected from the internet, the appropriate actions performed, and then uploaded to the virtualisation server.

3.5.1 Data storage. Directly connected to the virtualisation server is a high-capacity data storage array. The storage array allows for multiple virtual partitions to be created, which can be directly attached to virtual machines within vSphere using iSCSI connections. Partitions are used for hosting template virtual machines, as well as backups of deployed machines. Further partitions are used to provide high capacity data storage for individual VMs for data including telemetry and network traces to satisfy our data capture requirement.

3.6 Industrial IoT (IIoT)

As one of our use cases is issues around convergence of OT and IoT, we make use of a number of diverse hardware and software variants of IIoT. IIoT hardware includes multiple WirelessHART sensors and transmitters, along with WirelessHART gateways. In order to reduce complexity, we also have bespoke internal projects to produce IIoT sensors with near *hot-swappable* wireless protocols, for example through the use of Arduino devices which can convert to WirelessHART, Zigbee or HTTP(s) as needed.

We also make use of IIoT software solutions within the testbed environment. For example, we use KEPServerEX from Kepware, which provides data aggregation capabilities for ICS devices from multiple manufacturers. Thingworx, a cloud IIoT platform, is deployed within the IT DMZ network. A more detailed explanation and example use case of this software is provided in Section 6.

3.7 Mobile testbed

To satisfy a need to both understand the integration of mobile datacomms into the testbed for remote field sites, and to provide

a portable demonstrator we have built a mobile field site. This consists of a full control system (including 2 PLCs, a RTU and HMI), and a simple small scale physical process. This box is partly used as an outreach tool, allowing demonstration at events and meetings without having to rely on a remote connection to the main testbed. It also serves as a self contained miniature testbed that can be used for research projects isolated from the main testbed. When required, this mobile setup can be connected to the main testbed via a 4G radio connection, appearing as its own field site. It can then use this connection to provide access to the software services that make up the main testbed. A breakdown of the mobile demo box can be seen in Figure 5.

3.8 Training and prototyping

Whilst building the testbed it became apparent that the complexity, and value, of the testbed created a need to train new students and inexperienced researchers to be able to practice ICS concepts without the risk of damaging the testbed itself. Similarly, for research projects that result in direct access to the testbed through software, it is safer to develop and test outside of the main testbed before deployment for data gathering. Jumping straight into deployment on the production testbed can be complex, and so by moving through the three stages with increasing complexity researches can gradually build up approaches.

A two tier setup is used outside of the testbed for training. The first is a virtual environment utilising FactoryIO [5], which allows for the simulation of multiple physical processes with full 3D rendering. This software can be controlled by both virtual PLCs, certain physical Siemens and Allen Bradley PLCs via ethernet connections alone, or can be physically wired to other PLCs using USB data interface devices. This is useful as a first stage training process with minimal risk of damaging equipment. The second tier is a tabletop physical process consisting of a multi-conveyor sorting process produced by LJCreate [6]. The device consists of multiple sensors and actuators providing a detailed model of a real-world process. This comes pre-configured with Siemens S7-1200 PLCs, but can be controlled by other devices if desired.

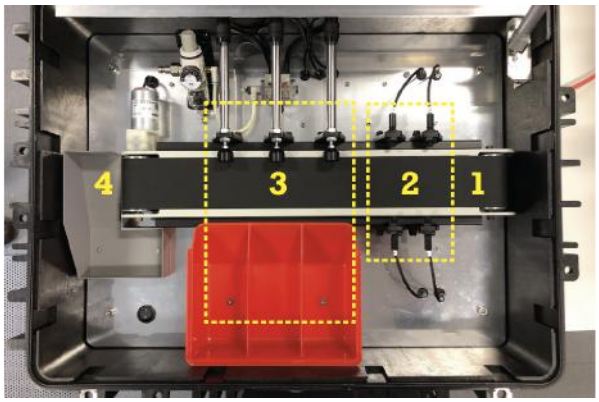
We require research projects to be tested on these setups, or the mobile demonstration box, before deployment to the main testbed.

4 REVISITING PAST LEARNT LESSONS

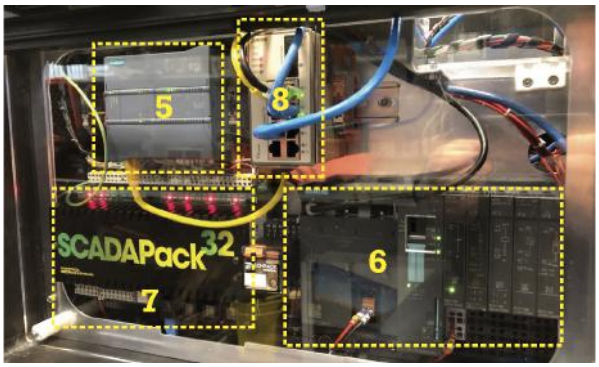
In our previous work, we provided a list of ten lessons learned in building an ICS testbed [12]. Across the following sections we revisit each of these, and describe how well they stand-up against two years of additional experience. We first provide a summary of the ten lessons, and then arrange these past lessons into three groups - those we found to be valid, those we no longer consider to be critical, and those that made us go *oops!* (i.e. valid yet challenging to follow in practice). From these a set of five additional lessons are derived and described in Section 5.

[5] https://factoryio.com
[6] http://www.ljcreate.com/uk/programs/engineering/control-and-instrumentation/hardware/318/industrial-control-teaching-set-siemens-detail

Process box



Control box



Key:

1	Conveyor Belt
2	Object Detection & Measurement Sensors
3	Pneumatic Sorters
4	Reject Bin
5	Secondary Programmable Logic Controller
6	Primary Programmable Logic Controller
7	Remote Terminal/Telemetry Unit
8	L3 Managed Ethernet Switch
9	Wireless Hart Transceiver
10	4G Radio (for backhaul to testbed)
11	Human Machine Interface (HMI)

Figure 5: Mobile conveyor/sort process & field site

4.1 Summary of old lessons

Lesson 1: Device and technology selections should be market-driven (OL1) . When building an ICS testbed, it is important to ensure that design choices are led by industry, including the selection of devices and protocols in use.

Lesson 2: Homogeneity and heterogeneity in field sites (OL2). There should be exploration of device selections that are both homogeneous, where manufacturing sites use devices from a single manufacturer, and heterogeneous, where devices from different manufacturers are mixed. This also includes legacy vs non-legacy devices, as would be the norm in non-greenfield sites.

Lesson 3: Process diversity is not always crucial (OL3). Having a single, simple process is preferred to process diversity, as it allows for easier diversity of the control architecture. A complex process means that swapping control devices is much harder, for example due to rewiring requirements.

Lesson 4: Hardware-in-the-Loop (HIL) is not essential in the Manufacturing Zone ((OL4). Due to a lack of accurate mathematical models for representing the behaviour of sensors and actuators, HIL is not viable for increasing scalability within a testbed. As process diversity is not crucial, real devices can be used and so HIL is not necessary.

Lesson 5: Simulations in the Manufacturing Zone are not favoured (OL5). Whilst software simulations are cost effective, they do not provide accuracy and reliability in mimicking real-world operations. Whilst using physical devices is far more expensive, the cost is acceptable for the experimental accuracy gained.

Lesson 6: Virtualisation and VLANs provide ease of integration and scaling (OL6). Making use of virtualisation and VLANs is easier, and cheaper, to deploy than physical hardware and allows for easier integration of new systems, and expansion.

4.1.1 Lesson 7: Employ a Management Network (OL7). By utilising a management network, which gives researchers a single access point to applications and tools located anywhere within the testbed, the complexity of the experimental layer is reduced.

Lesson 8: Setup Multiple Manufacturing Zones (OL8). Separating devices into discrete manufacturing zones, on top of providing a more real-world scenario, allows for multiple researchers to conduct their activities simultaneously.

Lesson 9: Comprehensively document as you build (OL9). To ensure accuracy and avoid extra time and cost in documenting a testbed after it is built, ensure that documentation is written during the build process, and budget for it if necessary.

Lesson 10: Optimise data logging for security purposes (OL10). Collecting and distributing data from the testbed is a complicated manual process. Steps should be taken to try and optimise and automate the data collection process for specific security use cases.

4.2 Valid lessons

The following seven lessons were found to remain valid during the design and build process of our new testbed.

Lesson 1: Device and technology selections should be market-driven (OL1). From the start of the design process, our testbed was developed to be as close to a real-world deployment as possible, both drawing on our previous experience and further interaction with industry experts. We cover legacy and non-legacy hardware from vendors including Siemens, Allen Bradley, Schneider, Yokogawa, Delta, Honeywell, Trend, Emerson, Westermo and Checkpoint.

Lesson 2: Homogeneity and heterogeneity in field sites (OL2). We have designed the testbed such that changing device selections is an easy process. As we have built the control board to be readily swappable, we can with minimal effort choose different devices to incorporate into each manufacturing site. This is backed up by our large selection of devices from different manufacturers (legacy and non-legacy).

Lesson 4: Hardware-in-the-Loop (HIL) is not essential in the Manufacturing Zone ((OL4). As before, we still believe HIL is not essential in the manufacturing zone. Whilst we argue below for the use of software-based simulation for training and prototyping purposes, using HIL for scalability with the production testbed is not desirable over actual physical processes. As we have adopted process diversity to explore issues of convergence, we have a large volume of sensor and actuator data to utilise without the need for simulation. Further, we are also in the process of designing a wireless sensor system to allow for the easy installation on non-wireless sensors across the different physical processes, allowing for further ease of deployment.

Lesson 6: Virtualisation and VLANs provide ease of integration and scaling (OL6). As in our previous testbed, we exclusively use virtualisation for deploying systems within the testbed. All virtual machines are hosted on a single, high-powered server. This includes an array of base virtual machine images for different operating systems, allowing the easy deployment of new workstations. When combined with the use of VLANs, we can install a system anywhere within the testbed from the virtualisation management interface.

Lesson 7: Employ a Management Network (OL7). We replicated the use of a management network in our testbed. Within our operations centre, we maintain a physical switch that allows devices to connect to the management network. Through this connection, remote desktop connection can be started to all virtual machine

Lesson 8: Setup Multiple Manufacturing Zones (OL8). We follow the same principle on our testbed. We treat each control board (as shown in Figure 3) as one production-level manufacturing zone. Each board supports a full control architecture for one or more physical processes, with each board supplied with it's own field site network allowing for experimentation to be run on each zone independently.

Lesson 10: Optimise data logging for security purposes (OL10). We still consider that for certain security use cases, steps should be taken to simplify and optimise data capture, as well as our research around IDS systems and dataset generation. This is reflected by one the major requirement for data collection.

4.3 Less critical lessons

The following two lessons were found to be less applicable during the design and build process of our new testbed.

Lesson 3: Process diversity is not always crucial (OL3). We make two observations about this lesson. First, having a more complex physical process allows for a testbed that more closely resembles a

real world environment. Whilst we agree this can mean more work to swap control devices, there are approaches to mitigate this (see Sec. 3.2). Secondly, we also argue that when looking at issues around convergence, process diversity does matter. As an example, in a real world environment the building management system and a factory process may well be on the same network, having a convergence influence over each other. A power distribution process can have a direct impact on a smart factory floor process. Having multiple physical process allows the exploration of issues around this.

Lesson 5: Simulations in the Manufacturing Zone are not favoured (OL5). We maintain that within the production testbed, software simulations are not favourable. However, software simulations can prove to be a valuable asset with the wider testbed environment. For example. software simulations of both processes and control devices are a useful training tool for researchers to become more familiar with concepts, such as PLC programming. Simulations can also be used as a prototyping stage when preparing experiments to be run on the production testbed, reducing the risk of damaging expensive physical equipment (see Sec. 3.8).

4.4 Oops! Lesson

The following lesson made us go *Oops!* during the build process of our new testbed. Whilst we feel this is a valid critical lesson, in practice it is challenging to follow when working at pace.

Lesson 9: Comprehensively document as you build (OL9). On commencement of the building of the testbed, this was one of the lessons that we attempted to follow as closely as possible. This was successful for a short period, however as more complex portions of the testbed were developed, and time-frames shortened, the documentation process was almost entirely forgotten. This has led to a requirement to document post-build, which will undoubtedly take far longer than if done in progress and posses additional challenges to both recollection and accuracy.

5 NEW LESSONS LEARNED

5.1 Refinements on old lessons

The following lessons are refinements of our previous lessons.

Refined Lesson 3: Build swappable capability into the control architecture (RL3). In old lesson 3, we argued that one of the benefits of a single, simple process is that the allows for effective swappability of control equipment through simple wiring and configuration. We found that process diversity is actually useful. However, a slightly different, and more proactive, approach needs to be taken to still maintain swappability of devices. Due to the nature of the physical infrastructure for control equipment, it can be a time intensive procedure to swap out hardware across different processes. For example, when replacing a PLC, you also have to replace the associate wiring to the physical process. Therefore, the physical design of how control boards are assembled should be such that swappability is made as simple as possible. The control board we use, as seen in Figure 3 and discussed in Section 3.2, has been designed to allow for relatively easy swapping of control equipment. As all wiring directly connected to the control equipment is contained within the board (and is well labelled), a new PLC can be wired in with minimal effort. The PLCs are also mounted on removable plates allowing for easy mounting and removal.

Refined Lesson 2: Temporality matters (RL2). In old lesson 2, we cover that it is important to have heterogeneity in manufacturing zones. We feel it is important to bring out the heterogeneity of devices from different time periods into a separate lesson. Whilst it is tempting to focus on the newest models of ICS equipment, in the real world installations will often have a mix of newer and older, legacy devices. When evaluating the security of these systems it is important to take into account differing security levels of different generations of devices, even from the same manufacturer. For example, a current generation PLC from a manufacturer may use the same protocol as an older generation model, but incorporate extra features to fix known vulnerabilities. Both devices could be installed within a real world scenario due to the common protocol, however if the older equipment is not taken into account then potential security issues could be missed.

5.2 New lessons

New Lesson 1: DIY vs Off-the-shelf vs Hybrid testbeds (NL1). One one hand, buying off-the-shelf can be a far quicker way to get a testbed up and running, with the added benefit that the product will almost certainly be built to a higher standard than achieved through a DIY approach. However, it is likely to be more expensive to purchase (excepting labour) than an equivalent DIY solution, and out-of-the-box be less configurable than a DIY approach. Whilst a DIY approach may be cheaper, it it far more time consuming and requires particular skill-sets not readily available. In our deployment, we use an off-the-shelf approach for physical processes with appropriate manufacturer customisations, but we replace the off-the-shelf control system with our own DIY control architecture, see Section 3.1.1 for more detail. When purchasing off-the-shelf hardware to be attacked, ensuring there are sufficient safety mechanisms in place is also critical.

Whilst a process can usually be considered safe during normal operation (and have suitable measures to ensure safety), while the system is under attack and the process is pushed past its limits, potentially with safety mechanisms disabled, safety can no longer be guaranteed. Measures should be taken during the design stage to provide non-digital backups to safety mechanisms, which should then be built into the off-the-shelf hardware.

Another disadvantage of off-the-shelf hardware is that it is often not easily extendable. It is expected that during the lifetime of the testbed, new hardware such as additional sensors may need to be added. As an example, the following modifications were made to our off-the-shelf water treatment process by the manufacturers in order to provide safety and extensibility:

- Three pressure release valves, set to 0.2 bar above the default safety cutoff of the system, were installed to release pressure in the system in the case where the pressure alarm is disabled due to attack.
- On the default configuration, the pipe connecting the filtration and absorption stages is fixed. This was replaced with a removable copper pipe, to allow us to introduce extra sensors into the system with relative ease. This can be seen in Figure 2.

New Lesson 2: Maintenance needs to be considered from the design stage (NL2). Introducing physical processes also introduces maintenance requirements, which grow with the size and complexity of testbeds. There is a time and cost involved with keeping physical processes operational, which needs to be considered from the outset of design. Our water treatment plant has had two major maintenance issues that have required intervention. Firstly, as the plant handles water (which sits at around 22°c), there is a risk of Legionella bacteria developing if the water is left to stagnate (as it would if the system is not run for longer periods of time). To combat this, a schedule of regular cleaning (including water replacement), and running to prevent stagnation, was introduced to minimise risk. Secondly, valves used to remove air from the system contain a filter (to prevent filtration material leaving the system) which rusted over, preventing the system from filling properly. This required periodic inspection and replacement as necessary.

New Lesson 3: Develop a reference architecture (NL3). Our reference architecture [6] aims to provide guidance in order to allow readers to produce a realistic, functional ICS testbed. Producing this reference architecture required a considerable amounts of work, across multiple universities, drawing on past experience of testbed building, and significant discussion with industry. There were three main benefits to producing a reference architecture:

(1) Further expansions to the testbed (for example, the addition of manufacturing zones or field sites) is made simpler, as the reference architecture can be followed.
(2) The reference architecture provides guidance to groups building their own testbeds, who may not necessarily have the experience or industrial knowledge to design a realistic architecture.
(3) If multiple testbeds are built following the same reference architecture, it is far easier to provide the functionality to connect different testbeds, even across different institutions, to provide a larger scale test environment.

New Lesson 4: Humans in the loop (NL4). It is important to utilise subject specialists when designing and building an ICS testbed. From the design stage, one needs to ensure that what is being purchased matches the requirements of the testbed and presents a realistic set to real-world usage. During the building phase, specialist skills may be required, outside of the remit of a computer scientist, to actually connect parts together. As an example, connecting our own PLCs to the water treatment plant required a large amount of specialised wiring. To ensure this was done correctly, and to a high standard, we utilised the university's engineering technician team to build the control board and carry out all wiring activities.

New Lesson 5: Provide infrastructure for training and prototyping (NL5). Working with, and on, industrial control systems requires specialist skill-sets. Rather than let inexperienced users have direct access to a primary testbed, it is important to provide infrastructure for training and practice, either through software simulation or through the use of (relatively) low-cost training hardware. Similarly, providing infrastructure for prototyping approaches can provide a simpler setup for rapid prototyping, and a proving ground before deployment on the primary testbed.

Lesson	OL1	OL2	OL3	OL4	OL5	OL6	OL7	OL8	OL9	OL10	RL3	RL2	NL1	NL2	NL3	NL4	NL5
Valid	✓	✓		✓		✓	✓	✓		✓							
Less crit			✓		✓												
Refined		✓	✓								✓	✓					
Oops									✓								
New											✓	✓	✓	✓	✓	✓	✓

Table 2: Summary of old and new lessons

5.3 Overall lessons learnt

Table 2 provides a summary of the old, refined and new lessons. We believe that both sets of lessons should be taken as a whole. Even the old lessons that we have adapted can still stand, depending on the aims of a testbed building project. Whilst we found that old lesson 3 (process diversity) is no longer accurate, in particular for a testbed looking at issues around convergence, for smaller testbed deployments this lesson can still be useful. Similarly, whilst we still maintain as per old lesson 5 that simulation of processes is not desirable within the production testbed, software simulation has a valuable use for training and prototyping..

6 DISCUSSION

The security testbed was built based on prior lessons and against a reference architecture for ICS & IIoT security testbeds [6]. By way of a use case for the testbed, we developed a demonstrator for the PETRAS National Centre of Excellence for IoT Systems[7] to highlight potential security vulnerabilities that manifest as a result of poorly considered convergence of operational technology with the industrial internet-of-things (IIoT). The "Securing IoT in Critical National Infrastructure" (SecCNIoT) demonstrator builds directly upon the security testbed through the inclusion of commercial IIoT solutions to accurately reflect a real-world deployment. demonstrator also covers multiple aspects from our research areas. The demonstrator exploits known ICS device security vulnerabilities, and also incorporates elements of active failure (human error) [4, 5].

6.1 The SecCNIoT demonstrator

The overall conceptual architecture of the demonstrator can be seen in Figure 6. Data aggregation from the testbed is performed by the KEPServerEX [8] (henceforth referred to as Kepware) data aggregation platform. Kepware collects data from multi-vendor ICS devices in the testbed. In our deployment, Kepware resides within a Microsoft Windows 7 VM, located within the OT DMZ network on the field site for the water treatment process—*field site 1*—communicating directly with the devices on its related control board.

The IIoT cloud platform for the demonstrator was provided by Thingworx [9], which supports the development of web-based applications utilising IIoT data. The manufacturers of Thingworx (PTC) acquired Kepware in 2016, and since have marketed Kepware and

[7] https://www.petrashub.org/petras-demonstrators-bringing-research-into-the-real-world
[8] https://www.kepware.com/en-us/
[9] https://www.ptc.com/en/products/iiot

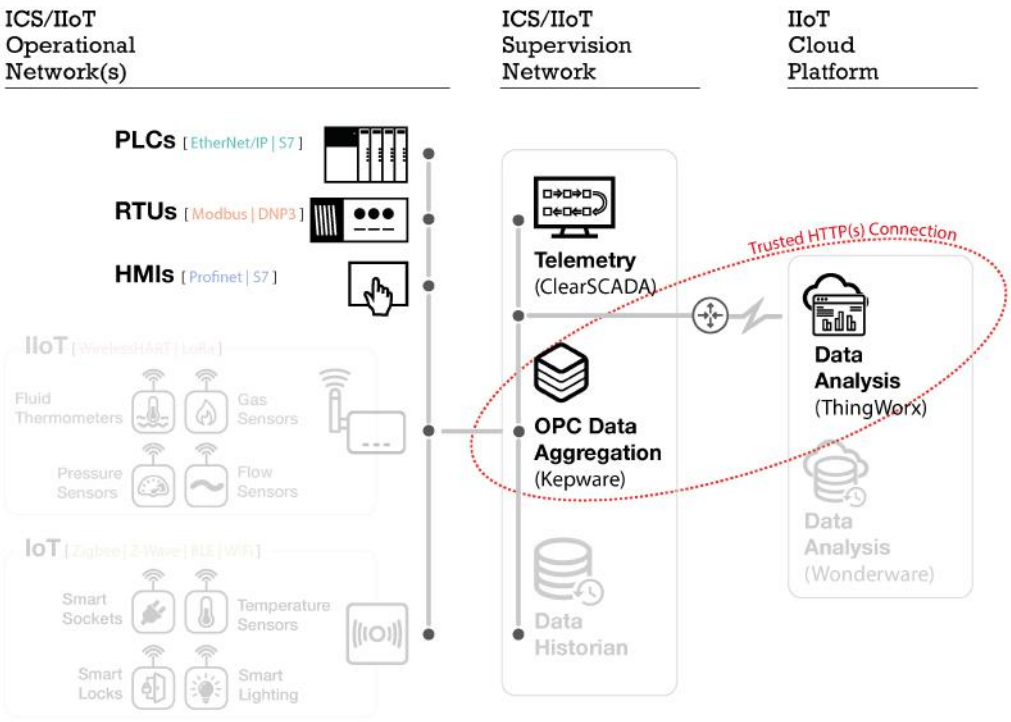


Figure 6: Testbed environment for SecCNIoT demonstrator

Thingworx as an IIoT solution, with EPServerEX providing data inputs to the Thingworx platform.

Thingworx is deployed on top of an Ubuntu virtual machine (supplied prebuilt by PTC) and uses Apache Tomcat 8.5 as its underlying platform. Our deployment operates Thingworx within a virtual cloud (i.e. inside our closed testbed environment). A trusted communication link between KEPServerEX and Thingworx is achieved by way of a default, pre-configured, HTTP connection.

6.2 Compromising the SecCNIoT demonstrator

The demonstration takes advantage of a number of aspects of this IIoT deployment, and is achieved in three primary steps. Prior to the attack we replace the KepServerEx manual, an Adobe PDF held on the Kepware server, with one which we have modified to contain a malicious payload. Whilst we take the liberty of doing this manually for the purposes of the demonstration, processes by which such a file might make it to a server, or an engineer's trusted workstation, are many and varied including USB drives, internet download, an injection into the supply chain (akin to the 2019 malware attack on Asus [10]) or a direct hack of the workstation itself.

Step 1—the Thingworx cloud platform is compromised by exploiting one or more of the well known CVE vulnerabilities published for Tomcat 8.5 [11] in the pre-configured state as shipped. In this state we are able to load our attack script to the Thingworx VM which, when executed, terminates the Thingworx process resulting in a loss of communication with Kepware. At this point we create an HTTP listener on the VM, ready for outbound connections from Kepware.

Step 2—as the Kepware workstation reports a communication error, resolution is sought by the engineer who opens the malicious manual. The payload displays the manual as expected, however it also re-establishes the trusted outbound communication to the Thingworx VM, only now this is intercepted by the listener created in *Step 1* above. As the connection is made, the attack script creates a reverse proxy in the VM providing remote access directly to the Kepware server.

Step 3—with a trusted connection established, the attack script undertakes reconnaissance of the ICS devices that are feeding data to the Kepware server, and provides a conduit for a number of predefined attacks upon PLCs, HMIs and RTUs. Through this process we are able to disable the over-pressure safety system for the water treatment plant and increase pump speeds to generate an unsafe operating state.

[10] https://www.symantec.com/blogs/threat-intelligence/asus-supply-chain-attack
[11] https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-887/version_id-199711/Apache-Tomcat-8.5.0.html

Work	PD	ID	PC	FX	SC	FD	SS	SA	OM	OP
[18]	●	◦	◦	◦		●		◦		
[2]	●	●	●	●		●	●	●		
[8]	●	●	●	●		●	●	●		
[20]	●	◦	●			●		◦		
LAN [12]	●	●		●	●	●		●	●	●
BRIS	●	●	●	●	●	●	●	●	●	●

Key: Black dots indicate that a testbed supports a feature, grey dots indicate that the work acknowledges a requirement of a feature

Table 3: Comparison of testbed functions.

6.3 Situating the testbed in the landscape

To evaluate our testbed, and through the use of the SecCNIoT demonstrator, we adopt the approach used in our prior testbed work [12]. In this we utilise the ten categories of testbed functionality as described in [9, 13, 19], being:

(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 3 presents our comparison, for the new testbed, in a the same format as used previously for ease of reference. Where the lack of process diversity and simulation acted as a conscious design choice in our previous testbed, through our adapted viewpoint of existing lessons, and inclusion of new lessons, we have been able to design and build our new testbed to meet all functionality categories. While we acknowledge some of these are still in their infancy, the testbeds fundamental construct is designed to support and manage their growth.

7 CONCLUSION

Our prior work in building an ICS security testbed resulted in the generation of ten key lessons useful for anyone designing and building their own testbed. In this paper we present how—when building a new greenfield testbed of significantly greater scale and extensibility—those ten guiding lessons played out, offering an opportunity to evaluate their usefulness. We found that, whilst most were still valid, two lessons around physical process diversity and the use of software simulations were no longer entirely valid when building a testbed at scale. A further lesson to document whilst you build, whilst valid, was in practice all but impossible to follow due to the pace of development and need to work through implementation challenges.

This paper contributes our attempt to valid our prior work, and in doing so brings a further seven additional lessons to the community. Two are refinements of the prior, and five are wholly new. Together these now fifteen lessons represent a guide to others who wish to embark of their own ICS security testbeds, offering sound and sage advice based on theory, practice and a lot of mistakes. Far better to learn from others than remake them.

ACKNOWLEDGMENT

This work has been funded by Lloyd's Register Foundation and the UK Engineering and Physical Science Research Council (EPSRC) as part of PETRAS: Cybersecurity of the Internet of Things Research Hub, grant no EP/N023234/1.

REFERENCES

[1] R. Antrobus, S. Frey, B. Green, and A. Rashid. Simaticscan: Towards a specialised vulnerability scanner for industrial control systems. BCS, 2016.
[2] R. Candell, T. Zimmerman, and K. Stouffer. An industrial control system cybersecurity performance testbed. Technical Report 8089, National Institute of Standards and Technology, NISTIR, 2015.
[3] Centre for the Protection of National Infrastructure. Critical National Infrastructure, 2017.
[4] B. Craggs. A just culture is fundamental: Extending security ergonomics by design. In *2019 IEEE/ACM 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)*. IEEE, 2019.
[5] B. Craggs and A. Rashid. Smart cyber-physical systems: Beyond usable security to security ergonomics by design. In *Proc. of the 3rd Int. Workshop on Software Engineering for Smart Cyber-Physical Systems*, SEsCPS '17, pages 22–25. IEEE Press, 2017.
[6] B. Craggs, A. Rashid, C. Hankin, R. Antrobus, O. Şerban, and N. Thapen. A reference architecture for iiot and industrial control systems testbeds. In *2nd Conference on Living in the Internet of Things 2019*, 12 2018.
[7] C. Davis, J. Tate, H. Okhravi, C. Grier, T. Overbye, and D. Nicol. Scada cyber security testbed development. In *2006 38th North American Power Symposium*, pages 483–488. IEEE, 2006.
[8] W. Gao, T. Morris, B. Reaves, and D. Richey. On scada control system command and response injection and intrusion detection. In *2010 eCrime Researchers Summit*, pages 1–9, Oct 2010.
[9] B. Genge, C. Siaterlis, I. Nai Fovino, and M. Masera. A cyber-physical experimentation environment for the security analysis of networked industrial control systems. *Comput. Electr. Eng.*, 38(5):1146–1161, Sept. 2012.
[10] B. Green, S. A. F. Frey, A. Rashid, and D. Hutchison. Testbed diversity as a fundamental principle for effective ics security research. *SERECIN*, 2016.
[11] B. Green, M. Krotofil, and A. Abbasi. On the significance of process comprehension for conducting targeted ics attacks. In *Proc. of the 2017 Workshop on CPS Security & Privacy*, pages 57–67. ACM, 2017.
[12] B. Green, A. Lee, R. Antrobus, U. Roedig, D. Hutchison, and A. Rashid. Pains, gains and plcs: ten lessons from building an industrial control systems testbed for security research. In *10th {USENIX} Workshop on Cyber Security Experimentation and Test*, 2017.
[13] H. Holm, M. Karresand, A. Vidström, and E. Westring. A survey of industrial control system testbeds. In S. Buchegger and M. Dam, editors, *Secure IT Systems*, pages 11–26, Cham, 2015. Springer International Publishing.
[14] W. Jardine, S. Frey, B. Green, and A. Rashid. Senami: Selective non-invasive active monitoring for ics intrusion detection. In *Proc. of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, pages 23–34. ACM, 2016.
[15] B. Johnson, D. Caban, M. Krotofil, D. Scali, N. Brubaker, and C. Glyer. Attackers Deploy New ICS Attack Framework "TRITON" and Cause Operational Disruption to Critical Infrastructure, 2017.
[16] Kaspersky Lab ICS CERT. Threat Landscape for Industrial Automation Systems. Technical report, 2019.
[17] R. M. Lee, M. J. Assante, and T. Conway. Analysis of the cyber attack on the Ukrainian power grid. Technical report, 2016.
[18] A. P. Mathur and N. O. Tippenhauer. Swat: a water treatment testbed for research and training on ics security. In *2016 Int. Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, pages 31–36, April 2016.
[19] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A. Sadeghi, M. Maniatakos, and R. Karri. The cybersecurity landscape in industrial control systems. *Proc. of the IEEE*, 104(5):1039–1057, May 2016.
[20] T. Morris, A. Srivastava, B. Reaves, W. Gao, K. Pavurapu, and R. Reddi. A control system testbed to validate critical infrastructure protection concepts. *International Journal of Critical Infrastructure Protection*, 4(2):88 – 103, 2011.