Everything is Awesome! Or is it? Cyber Security Risks in Critical Infrastructure

Awais Rashid[0000−0002−0109−1341]†, Joseph Gardiner[0000−0003−4748−4228]†, Benjamin Green[0000−0002−1013−9933]‡, and Barnaby Craggs[0000−0002−6706−9745]†

†Bristol Cyber Security Group, University of Bristol, Bristol, UK; ‡Security Lancaster Institute, Lancaster University, UK**
†{awais.rashid,barney.craggs,joe.gardiner}@bristol.ac.uk;
‡b.green2@lancaster.ac.uk
http://www.bristol.ac.uk/engineering/research/cyber-security/

Abstract. Industrial Control Systems (ICS) play an important role in the monitoring, control and automation of critical infrastructure such as water, gas, oil and electricity. Recent years have seen a number of high profile cyber attacks on such infrastructure exemplified by Stuxnet and the Ukrainian Power Grid attacks. This naturally begs the question: how should we manage cyber security risks in such infrastructure on which the day-to-day functioning of societies rely? What are the complexities of managing security in a landscape shaped by the often competing demands of a variety of stakeholders, e.g., managers, control engineers, enterprise IT personnel and field site operators? What are the challenges posed by the convergence of Internet of Things (IoT) and critical infrastructure through the so-called Industrial Internet of Things (IIoT)? In this paper, we discuss insights from a multi-year programme of research investigating these issues and the challenges to addressing them.

Keywords: Cyber security · Industrial control systems · Critical infrastructure · Industrial IoT · Cyber risk decisions.

1 Introduction

Critical infrastructure systems, e.g., water, power, etc. are increasingly being connected to other enterprise systems for a variety of reasons. These range from the need to remotely update and maintain systems, reducing the effort, time and cost of visiting remote, hard to access facilities through to the desire to gain real-time business intelligence in order to optimise processes and improve efficiencies. Consequently, the past assumption that such systems are air-gapped from wider networks, and the Internet, is increasingly being proven to no longer be valid. Unintentional connections are also an increasing issue, with poorly configured controls allowing outside connections to individual control devices. For instance,

** Current affiliation. The author conducted the work while employed by the Bristol Cyber Security Group.

a browse though the Shodan search engine of Internet connected devices shows a large number of programmable logic controllers (PLCs) from various manufacturers. As we note above, increasingly, these connections to external systems, and the Internet, are being introduced intentionally for remote connectivity and configuration capabilities or other business needs. This results in a number of risks to critical infrastructures as a consequence of an increased cyber-attack surface.

As we build more and more complex large, connected environments, the scale of connectivity and complexity of such enrivonments will only increase further—resulting in increased scale of attacks and impact. The problem is compounded by the fact that often critical infrastructures have multiple organisations that collectively contribute to the environment, e.g., power producers and power distribution networks are often owned by different organisations not to mention the large number of organisations that form the supply chain. The infrastructures remain in operation over a long lifespan and their make-up (devices, software, communication protocols) changes over time, often resulting in a large number of legacy and non-legacy systems working in conjunction to deliver critical services to citizens.

For instance, we are seeing the emergence of multiple products and services on the market that allow for data from industrial environments to be sent to the cloud for processing, allowing remote monitoring of processes, under the banner of Industrial Internet of Things (IIoT) and Industry 4.0. The latest push is to move onto SCADA-in-the-cloud, where more and more control functions are moved into the remote cloud environment. This transition results in ICS equipment becoming Internet-of-Things devices, with indirect connections to IT networks and the wider Internet—providing a route for attackers to gain access to these devices through compromise of the cloud environment.

Cyber security risks, therefore, need to be managed in the face of these new attack vectors and increased attack sophistication whereby highly resourced adversaries may disrupt critical services to large parts of the population. Managing such risks is, however, non-trivial for three reasons:

- Risk is a socio-technical construct—requiring not only an understanding of the technical threat landscape but also organisational and human dimensions of risk and risk decision-making.
- Risks arising from both legacy and non-legacy systems need to be understood especially those that emerge from the convergence of the two.
- Such understanding needs to be developed through direct engagement with the stakeholders and experimentation on realistic infrastructures. The former is often challenging due to the sensitive nature and confidentiality cultures within critical infrastructure organisations. The latter cannot be achieved through experimentation on production environments (as this can lead to disruption to the infrastructure), hence requiring realistic testbeds that enable modelling and experimentation of non-trival attack scenarios.

In this paper, we discuss insights from a multi-year programme of research investigating both social and technical dimensions of cyber security risks in critical

infrastructures. We first summarise (Section 2) insights from our prior work [4, 5] to discuss the human and organisational dimensions of cyber risks. We then move onto the technical aspects of the problem and particularly the risks arising from convergence of IIoT and ICS environments. We describe the Bristol Cyber Security Group (BCSG) testbed (Section 3) followed by detailed discussion of an attack (Section 4) against an ICS environment (set up using our testbed) which utilises a cloud provider. We exploit the connection from the operational network to the cloud to provide a tunnel through which we can gain access to the control equipment. Through this access we force the physical process, a water treatment plant, to enter an unsafe state by disabling a pressure alarm and increasing he speed of the primary pump.

2 Human and Organisational Dimensions of Cyber Risk

In prior work [5], we analysed a number of high profile attacks against industrial control systems (including those impacting critical infrastructures) and highlighted that perception errors play a key part in attack success. These perception errors relate to four dimensions (Figure 1):

System qualities Operators may have incorrect perceptions of particular qualities of the system, e.g., confidentiality, integrity, availability, resilience, etc. This may lead them to think that the system can withstand particular faults or recover gracefully when this may not be the case in reality.

System boundaries Operators may have incorrect perceptions a system's isolation (physical or virtual) from other systems. This, in turn, may lead the operators to assume that lateral movement across systems or from less critical to mission critical systems is not possible when this may not be the case in reality.

Observability Operators may assume that, at a particular point in time, their observation of system behaviour is accurate and complete.

Controllability Operators may assume that, even under attack conditions, they maintain control of the system especially safety-critical components.

However, these perception errors often arise due to *latent design conditions* [9] – improper specification of system qualities, borders, observability and controllability – during conception, design, implementation or evolution. These latent design conditions are further exacerbated when the attacker *actively tampers* with the observation and control link between the operators and the system.

A number of stakeholder decisions shape security within such a system, and by consequence, may lead to latent design conditions. Previously, we designed a game, Decisions & Disruptions[1] and analysed the decision-making processes of three different stakeholder groups: managers, computer science/IT experts and security experts [4]. The game (see Figure 2) charges a team of players to defend a hydro-electric power producer (represented by a Lego® board) against

[1] `https://www.decisions-disruptions.org`

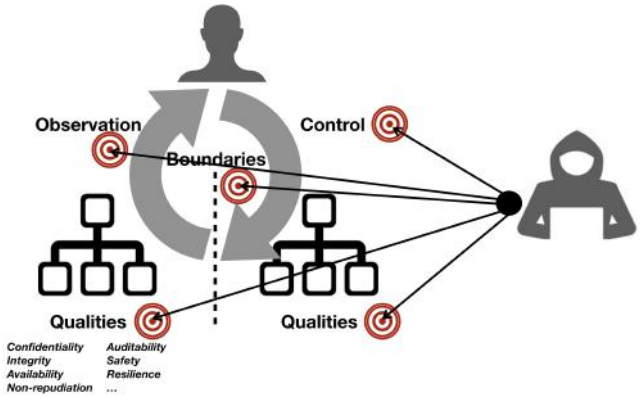


Fig. 1. Dimensions of cyber risk perception

cyber attacks. Players invest in various defences over four game rounds. Attacks occur during each round and their success depends on the defences in which the players have invested. However, the players have a limited budget in each round (unspent budget carries over to the next round) so they must prioritise and choose amongst the different types of defences. These range from basic defences such as firewalls, anti-virus and security training to more advanced network monitoring solutions. Players can also seek to gain intelligence by paying for a threat assessment and an asset audit. The game is available under a CC-BY-NC license and all the cards, a Bricklink model the Lego® and the rule book are all available on the website.

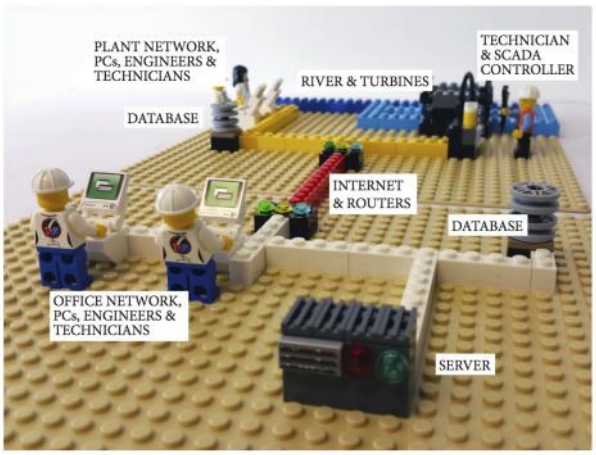


Fig. 2. An overview of the Decisions & Disruptions game board

Our analysis of twelve teams representing the three stakeholder groups: managers, computer scientists/IT personnel and security experts from academia and industry has highlighted a number of interesting patterns. We briefly summarise these below. Readers are referred to [4] for a detailed discussion:

- *Security experts* favoured advanced cyber security protection measures, e.g., the advanced network monitoring product (effectively a very expensive intrusion detection system) and often deprioritised basic protections (e.g., firewalls, anti-virus and patching) and intelligence gathering (e.g., threat assessment or asset audits). Their discussions were *scenario-driven* (e.g., what if ...) and they had *high confidence* in their decisions – even when these decisions led them to catastrophic outcomes.
- *Computer scientists*, by contrast, focused more substantially on intelligence gathering and human factors (e.g., security training) and deprioritised advanced cyber security protection measures. However, they also deprioritised data protection – a fundamental requirement for information security. They used *diverse* strategies during decision-making but expressed *low confidence* in their decisions even those that subsequently successfully deflected attacks.
- *Managers*, on the other hand, did prioritise data protection along side basic cyber security protection whilst also favouring advanced cyber security technologies. However, they paid less attention to human factors. Their decisions were very much *intuition-driven* (e.g., I like the firewall) and they too exhibited *low confidence* in their choices.

The analysis also showed that security experts did not necessarily always perform the best. There were also a number of good, bad and ugly decision-making patterns that stakeholders should look out for when making decisions about cyber seurity in an organisation. These are detailed in [4].

Having summarised human and organisational aspects of cyber security in critical infrastructure settings from prior work, in the rest of the paper, we discuss a concrete scenario emerging from the convergence of tranditional control systems (so-called *operational technology (OT)*) and Industrial IoT platforms and devices and demonstrate how this leads to an increased attack surfance and successful attacks.

3 Testbed

Before we discuss the details of the attack, we provide a brief overview of the testbed environment that is under attack. For a more detailed discussion of the testbed we refer the reader to [6].

3.1 Physical process

The primary physical process of the BCSG testbed, and the target of the described attack, is the Gunt CE581 water treatment plant[2], as seen in Figure 3. The CE581 system, designed for the training of water treatment engineers, consists of a three-stage filtration, absorption and ion-exchange process. The CE581 physical aspect is largely off–the-shelf, however we had the unit customised with safety valves to release system pressure when under attack, and added a removable copper pipe between the filtration and absorption stages to allow for the easy installation and removal of additional sensors.

The CE581 initially utilised a small Eaton PLC for control. By using swappable terminal blocks, we replaced the control equipment with our own control architecture. This allows us to utilise equipment that more closely represents real-world scenarios. This architecture is discussed in Section 3.2. The original control equipment can be made operational with minimal effort if required for maintenance.

The system has five controllable elements: the pump and 4 electronically operated valves to control the flow of water though the three stages. The system also has a number of sensors, including a pressure alarm, differential pressure across the filtration tanks and temperature sensors.

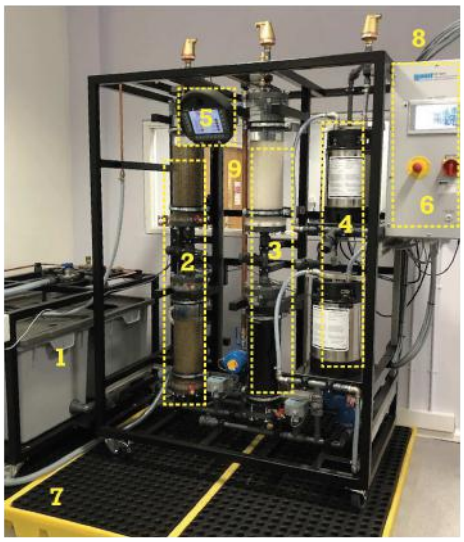
Normal operation Under normal operation, the system operates with the pump set at 80% speed utilising all three stages of the treatment process. Under these conditions, the system runs with around 1.55 bar of internal pressure. If the pump speed is increased past 90%, the system pressure increases above 1.6 bar. If this occurs, the pressure sensor sends an alarm signal to the PLC, and the logic deactivates the pump. After a few seconds, the pressure will drop below 1.3 bar and the alarm will terminate, allowing the pump to restart. If the pump speed is not reduced, the system will repeatedly exceed the pressure limit, shut down and then restart,

3.2 Control Equipment

We now describe the different pieces of control equipment that control the water treatment process. The control equipment is connected via ethernet to a Westermo industrial switch, which provides communication between devices and to other services.

Programmable logic controllers (PLCs) The primary PLC for the water treatment plant is a modern Siemens S7-1500. This unit controls the electronic valves and pump speed, as well as receiving inputs from the various sensors on the plant. A second, much older PLC, a Siemens ET200S, is used to control the

[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
9 Removable copper pipe for sensor installation

Fig. 3. Water treatment process

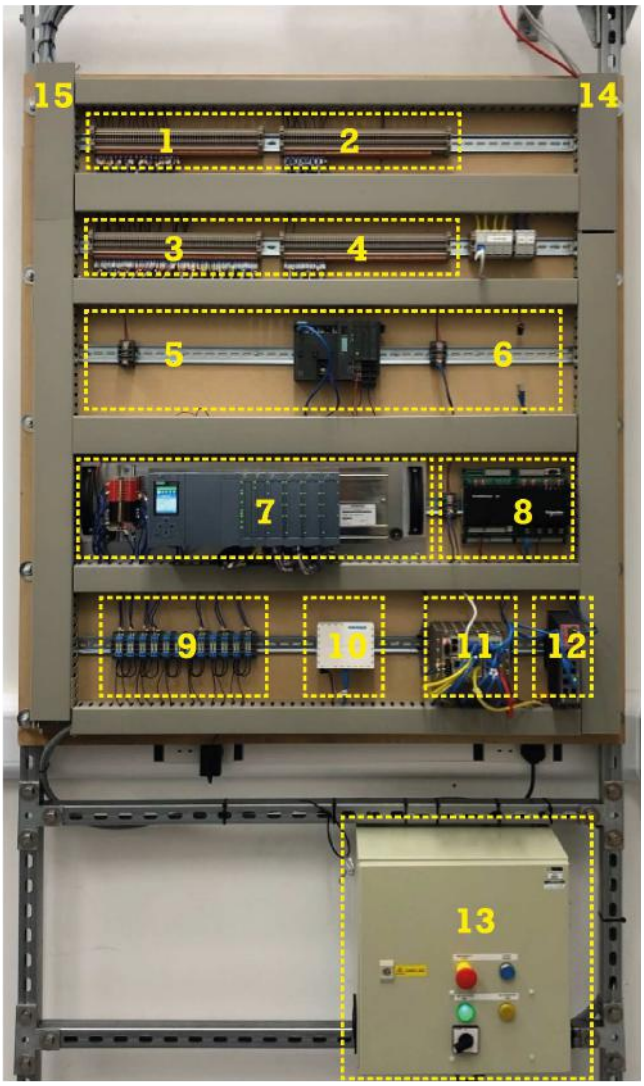
pump on-off state. This mix of older (more vulnerable) and newer devices allows us to examine the convergence of old and new devices.

Human machine interfaces (HMIs) The operator HMI is a wireless Siemens panel (MobilePanel 277 IWlAN V2). This is connected to the field site network over WiFi using a Mikrotik access point. The screen can be seen in Figure 5a.

Remote telemetry unit (RTU) Telemetry is provided through a SCADA-pack32 from Schneider, which communicates with the ClearSCADA software.

3.3 Networks

An overview of the network architecture is presented in Figure 6. The network is divided into the IT and operational (OT) networks. These are, in turn, divided



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 240VAC to 24V DC Power Supplies
14 Ethernet Back-haul to Core Network Infrastructure
15 IO Cabling to Physical Process

Fig. 4. Field Site Control Board

(a) HMI (b) ClearSCADA

Fig. 5. HMI and ClearSCADA screens

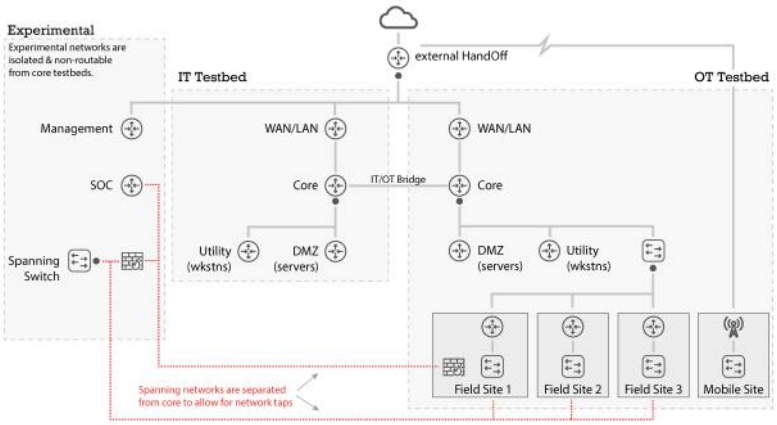


Fig. 6. Testbed network diagram

into multiple subnets, with each assigned its own /24 address space. Alongside the IT and OT networks there exists an experimental network which has a primary function of allowing researchers access to the testbed for experimentation and maintenance. For example, all virtual machines have a connection to the management network to allow for easy RDP access. Subnets are mapped to VLANs for allowing access by virtual machines. Each individual field site (physical process and associated control equipment) is assigned its own field site network.

The network is configured with multiple spanning networks to allow for network captures across any of the individual subnets within the network.

3.4 Software

All software is installed within virtual machines on a single server running VMWare vSphere. Each piece of software is installed within its own virtual

machine, which is, in turn, attached to an appropriate set of VLANs to insert the machine into the network architecture.

ClearSCADA from Schneider[3] is used to display telemetry data to the operations centre. ClearSCADA communicates with the Scadapack32 RTU over the Modbus and DNP3 industrial protocols and sits within the OT-Utility network. A basic example display can be seen in Figure 5b.

KEPServerEX from Kepware[4] (henceforth referred to as Kepware) is a data historian which sits inside individual field site networks with direct access to the control hardware. It can read and write data to these devices, and send it on to higher level services, including public cloud services. Kepware is installed within a Windows 7 virtual machine with an installation deployed within each field site network.

The primary cloud service that we use is Thingworx[5] from PTC (who now also own Kepware), a cloud-based IIoT platform. Through a connection to Kepware, apps can be developed to run on the Thingworx platform with data from the control devices. Thingworx is deployed within an Ubuntu virtual machine running Tomcat 8.5, as configured by the supplier. An example Thingworx application for the water process can be seen in Figure 8a.

4 Attack Overview

The physical process under attack is the water treatment plant as seen in Figure 3, with the control equipment mounted onto an specially designed board, as seen in Figure 4. The goal of the attack is to cause the water treatment process to enter into an unsafe state.

The overall conceptual architecture for the attack demonstration can be seen in Figure 7. Data aggregation from the testbed is performed by the Kepware data aggregation platform. In the deployment for the attack scenario, Kepware resides within a Microsoft Windows 7 VM, located within the OT DMZ network on the field site for the water treatment process, communicating directly with the devices on its related control board.

The IIoT cloud platform for the demonstrator is Thingworx, 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 Thingworx as an IIoT solution, with Kepware 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 Kepware and Thingworx is achieved by way of a default, pre-configured, HTTP connection.

[3] https://www.se.com/uk/en/product-range-presentation/61264-clearscada/
[4] https://www.kepware.com/en-us/
[5] https://www.ptc.com/en/products/iiot

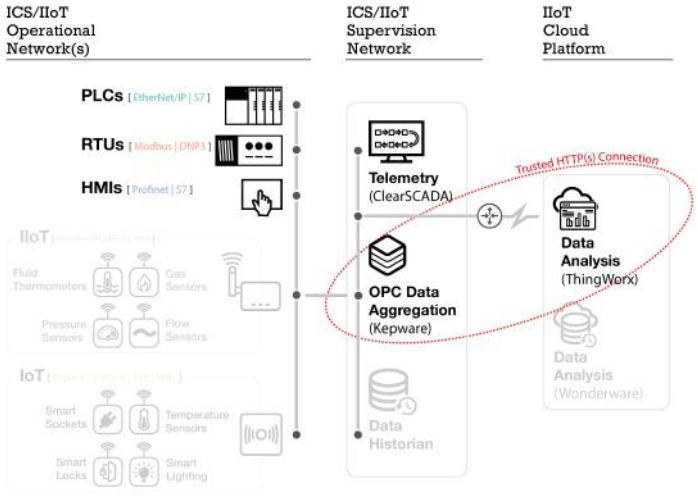
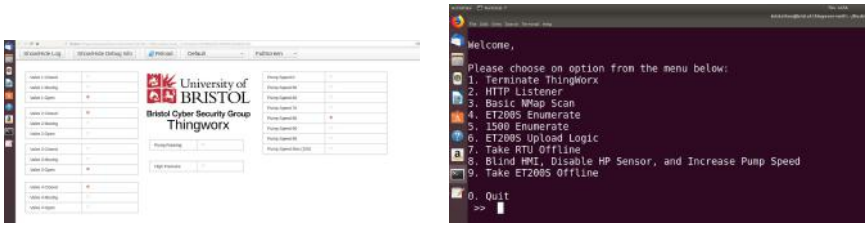


Fig. 7. Testbed environment for SecCNIoT demonstrator



(a) Thingworx (b) Attack script

Fig. 8. Thingworx and attack script

4.1 Anatomy of the attack on the OT-IIoT environment

We now discuss the anatomy of the attack we have implemented on this converged OT-IIoT environment as modelled within our testbed. It is an evolution of our previous work on attacks in ICS environments [7]. The flow of the attack can be seen in Figure 9.

(1) Compromise Thingworx The first step is to compromise the cloud machine hosting Thingworx. As mentioned, Thingworx was delivered pre-installed inside an Ubuntu virtual machine running on Tomcat 8.5. Tomcat is well known for having multiple security vulnerabilities that can be easily exploited to gain access to the host. Once access to the host is gained, through a vulnerability in Tomcat or other process, then the attacker loads a copy of the attack script

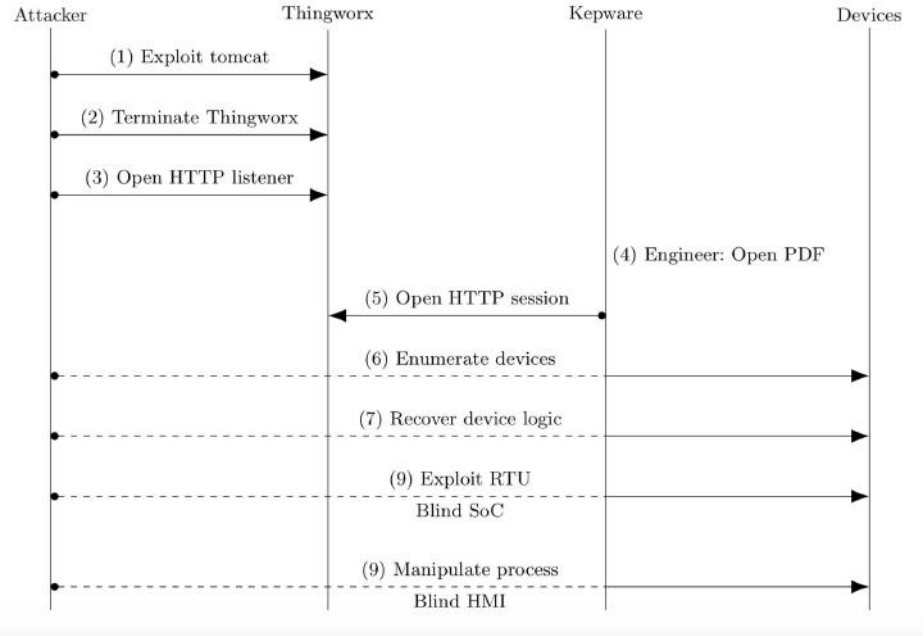


Fig. 9. Attack process. - - -indicates communication over the proxy

onto the host. The attack script, as seen in Figure 8b, is a simple python script utilising command line tools such as nmap and the Step7 python libraries.

(2) Terminate Thingworx On gaining access to the machine, the attacker first takes Thingworx offline by terminating the Tomcat process. This will mean Kepware can no longer communicate with the Thingworx server and will display an error.

(3) Setup HTTP listener The attacker now starts an HTTP listener on the Thingworx host, ready for receiving a connection from within the OT network in the next step.

(4) Compromise Kepware host The attacker now requires an engineer to open a connection back to the Thingworx host from the Kepware host. As Kepware is now throwing error messages due to a failed connection to Thingworx, the engineer is likely to inspect the machine. Prior to the attack, the attacker would have replaced the Kepware manual, an Adobe PDF held on the Kepware host, with one which they 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[6]) or a direct hack of the workstation itself.

In trying to fix the communication error, the engineer opens the malicious PDF file which, whilst appearing to open as normal for the engineer, executes the attacker's payload.

(5) Open reverse HTTP connection and set up proxy On opening the malicious PDF, the payload opens a reverse HTTP connection back to the Thingworx host, connecting to the listener started in step (3). As Kepware already communicated with the Thingworx host over HTTP, this connection is trusted and allowed through any firewalls. The attacker then. sets up a proxy connection through this HTTP link, and pipes all actions from their attack through this proxy, allowing them access to the control devices accessible from the Kepware host.

(6) Enumerate devices Once access has been gained to the ICS devices, the first stage is to enumerate these devices. In the first instance nmap can be used to perform a port scan across the field site network. This should give the attacker a list of devices, and open ports. From the open ports the attacker can gain some insight into the manufacturers of each device. For example, devices with port 102 open indicate Step7, which is the primary protocol in use by Siemens devices.

Specific tools can then be used to gain precise information about device hardware and software. As en example, PLCScan[7] can be used to gain details on older Step7 and modbus devices. For newer devices, there are nmap scripts as well as vulnerability scanners, e.g., [1, 2], available for collecting the same information. Returned information includes the hardware and firmware versions, and device name (as set by the operator). Knowing the hardware and firmware version allows the attacker to find publicly known vulnerabilities in the devices.

At this stage the attacker could use a known exploit to carry out, for example, a denial-of-service attack. In this scenario we go further to show how an attacker could have a controlled impact on the physical process.

(7) Recover device logic In order to craft attacks which can manipulate the process in a specific way, the attacker first needs to gain a copy of the logic running on the PLCs. This will allow them to know which memory addresses need to be attacked. For the Siemens ET200S, this is a simple case of directly reading a function block from the device using the Step7 protocol. This block will then need to be reverse engineered to recover the logic. For newer Siemens devices, such as the S7-1500, this no longer works. However, if the attacker has

[6] `https://www.symantec.com/blogs/threat-intelligence/asus-supply-chain-attack`

[7] `https://github.com/meeas/plcscan`

network access to the device and knows the exact hardware variant, the logic can be recovered from the device using the Siemens TIAPortal software.

Whilst reverse engineering the logic may prove difficult, especially if the attacker has no knowledge of the underlying physical process, it is possible to craft attacks using the recovered logic [8].

(8) Exploit RTU to blind control room To exploit the RTU, we developed a zero-day exploit and followed standard disclosure practices for vendor notification. Details can be found in a Schneider report [3]. Through this exploit, we load a new configuration to the device in order to change the devices IP address. This causes ClearSCADA to lose its connection to the device, meaning that the control room no longer has a view of the system.

(9) Manipulate process and blind HMI The final step is to directly interact with the PLC to interfere with the physical process, and cause the HMI to display incorrect information. This is achieved by repeatedly writing to specific memory addresses within the PLC logic, which overwrites variables used by the logic and specific tags on the PLC. The targeted elements are the state of the pressure alarm, and the pump speed. The pump speed is increased to 95% (above the standard operation of 80%). We also overwrite the variable that is read by the HMI to display the pump speed. Writing is achieved using the S7 protocol, with packets sent repeatedly at a high rate, faster than the PLCs cycle time.

This has the effect that the pump speed increases to 90% which causes the system pressure to exceed the usual safety limit, however the pressure alarm does not trigger the safety cutoff and the pump continues to run. Meanwhile, the HMI displays no pressure alarm and continues to indicate a pump speed of 80%.

5 Conclusion

In this paper, we have discussed insights from a multi-year programme of research studying cyber security risks in critical infrastructures from a socio-technical perspective—taking into account both technical vulnerabilities and human & organisational factors. Our work, to date, has shown that effective assessment and management of such risks requires an understanding of how stakeholders make security decisions and how latent design conditions manifesting *down the line* as a consequence of these decisions impact the cyber security of such infrastructure. Our work has also shown that one must consider the complexity arising from the melting pot that represents the devices and systems that are deployed within critical infrastructures. Given their long lifespan the infrastructure is a combination of devices, platforms and protocols from diverse manufacturers. Not only so, there are a range of legacy and non-legacy devices and systems in operation in conjunction within each other. The need for business efficiencies and remote maintenance and updates is leading to the infrastructure becoming more

highly connected to external systems and newer devices and platforms such as IIoT. This leads an increased attack surface that can be exploited by attackers to compromise security and hence safety as well as impact critical services to large swathes of the population. Studying such issues – from a *socio-technical perspective* – is a key part of our on-going and future work.

Acknowledgements.

This work is funded by EPSRC Grant "Mumba: Multi-faceted Metrics for ICS Business Risk Analysis" (EP/M002780/1), part of the Research Institute on Trustworthy, Interconnected, Cyber-Physical Systems (RITICS) and Lloyds Register Foundation grant "Securing IoT in Critical National Infrastructure", part of the UK Research Hub on Cyber Security of IoT (PETRAS). The work is also supported by Rashid's Fellowship from the Alan Turing Institute.

References

1. Rob Antrobus, Sylvain Frey, Awais Rashid, and Benjamin Green. Simaticscan: Towards A specialised vulnerability scanner for industrial control systems. In *4th International Symposium for ICS & SCADA Cyber Security Research 2016, ICS-CSR 2016, 23 - 25 August 2016, Queen's Belfast University, UK*, 2016.
2. Rob Antrobus, Benjamin Green, Sylvain Frey, and Awais Rashid. The forgotten i in iiot: a vulnerability scanner for industrial internet of things. In *IET Conference on Living in the Internet of Things*. IET, 2019.
3. Schceider Electric. Security Notification - Modicon Controllers and SCADA-Pack (V3.0). `https://www.schneider-electric.com/en/download/document/SEVD-2017-065-01/`, 2019.
4. Sylvain Frey, Awais Rashid, Pauline Anthonysamy, Maria Pinto-Albuquerque, and Syed Asad Naqvi. The good, the bad and the ugly: A study of security decisions in a cyber-physical systems game. *IEEE Trans. Software Eng.*, 45(5):521–536, 2019.
5. Sylvain Frey, Awais Rashid, Alberto Zanutto, Jerry S. Busby, and Karolina Follis. On the role of latent design conditions in cyber-physical systems security. In *Proceedings of the 2nd International Workshop on Software Engineering for Smart Cyber-Physical Systems, SEsCPS@ICSE 2016, Austin, Texas, USA, May 14-22, 2016*, pages 43–46, 2016.
6. Joe Gardiner, Barnaby Craggs, Benjamin Green, and Awais Rashid. Oops I did it again: Further adventures in the land of ICS security testbeds. In *ACM Workshop on Cyber-Physical Systems Security and Privacy (CPS-SPC)*. ACM Press, 2019.
7. Benjamin Green, Marina Krotofil, and Ali Abbasi. On the significance of process comprehension for conducting targeted ICS attacks. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy, Dallas, TX, USA, November 3, 2017*, pages 57–67, 2017.
8. Stephen McLaughlin and Patrick McDaniel. Sabot: Specification-based payload generation for programmable logic controllers. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 439–449, New York, NY, USA, 2012. ACM.
9. James Reason. *Managing the risks of organizational accidents*. Ashgate, 1997.