

From Plant to Lab: Industrial Emulation Tools for Real-World Security Testing in Industrial Control Systems

Argiro Anagnostopoulou¹^a, Thomas Papaloukas², George Stergiopoulos²^b
and Dimitris Gritzalis¹^c

¹*Dept. of Informatics, Athens University of Economics and Business, 76 Patision Ave, Athens, Greece*

²*Dept. of Information and Communication Systems Engineering, University of the Aegean, GR-83200 Samos, Greece*

Keywords: Industrial Control System, Industrial Process, Emulation Software, Comparative Analysis.

Abstract: The industry and research communities have introduced a variety of approaches and algorithms that require evaluation of their security and safety in industrial settings. However, conducting such assessments is challenging, given the need to maintain operational availability of these infrastructures. Consequently, scientists often capture datasets from authentic industrial environments, but assess attacks on such data in emulated or simulated environments. To replicate proper industrial conditions within controlled and simulated environments, testbeds need to be able to assess the effectiveness of implemented attacks and provide results usable to real-world systems. In this study, we present and compare such tools that aim to emulate or simulate industrial control systems by establishing eight criteria. The objective of our work is to address concerns surrounding the selection of an appropriate emulation tool based on specific needs.


1 INTRODUCTION


Industrial Control System (ICS) is a term used to describe industrial automation systems that are usually responsible for visualization and control of industrial processes and data acquisition. These systems are mainly used in industrial sectors or critical infrastructures (Mattioli and Moulinos, 2015). Indicative examples of ICS are the Supervisory Control and Data Acquisition (SCADA), the Programmable Logic Controllers (PLC), the Distributed Control Systems (DCS), and the Industrial Automation Systems (IAS). These systems are employed in environments where automation and monitoring are required, such as powerplants, water facilities, nuclear plants etc. (Stouffer and Falco, 2006), (Drias et al., 2015).


There is an increased interest of both research community and industries to reduce the attack surface on industrial environments. Researchers are constantly proposing algorithms and models to secure the industrial sector. However, these approaches cannot be evaluated on real-time industrial environments since

such infrastructures must be always up and running. For this reason, there are alternatives that may assist in the testing of the approaches. The first option is to develop, physically or virtually, a testbed in order to emulate the business processes of such infrastructures. However, in this alternative there is a dilemma: whether someone chooses an emulation software instead of physically designs a testbed on a research lab. The answer is not clear and depends on the purposes that the testbed need to be designed. Additional factors to consider may be the cost of equipment, the scale of the testbed, the downtime, or the risk of failure. A great advantage of preferring the emulation tools is that they are able at a great extend to emulate the conditions and setup of industrial sector at a low or even zero cost.

Another option is to find a publicly available dataset, and evaluate an approach using these data. This is a difficult task, since information regarding the business processes and transmitted values are quite protected. Sources that drive to real-world datasets are limited, because such information describe the operation of a critical infrastructure. Moreover, using a da-

^a <https://orcid.org/0000-0003-4199-6257>

^b <https://orcid.org/0000-0002-5336-6765>

^c <https://orcid.org/0000-0002-7793-6128>

taset limits the researcher's capabilities, since it would not be able to launch an attack scenario.

In this work, we present nine available tools that emulate industrial environments. We compare these tools regarding specific factors, such as complexity, pricing, accessibility of documentation, ease of configuration, and scalability.

1.1 Motivation and Contribution

There is a plethora of software available for emulating a SCADA system. The question is why a researcher chooses to use an emulation software instead of physically design and implement a testbed. There are multiple factors that we should consider before we decide, including pricing, downtime, flexibility, or risk of failure (Queiroz et al., 2011).

The primary obstacle for physically developing a testbed is its cost. An industrial environment is a quite complex infrastructure with various types of equipment. For example, SCADA is composed of multiple distributed devices. Each device may be included to a different type. Regarding the aim and the type of the device, there is a different cost and effort for its configuration. Thus, when we decide which device will be included in our testbed, for instance PLC or RTU, we should bare in mind the cost and the expertise needed (Queiroz et al., 2011).

In case that someone wants to launch attacks on the infrastructure, it is crucial such action not to happen in productive environment. Attacks like distributed denial of service (DDoS) cause downtime issues, because attackers overwhelm the resources of the infected system which is not designed to handle such amount of traffic. Such activities can either increase the response time, or even shutdown the whole service. Impacting the provided services for testing purposes, especially in SCADA systems, is not tolerable (Queiroz et al., 2011). Moreover, testing malevolent scenarios in real-time systems raises a great risk since the tested attacks aim to bypass any security measures that are in place. This may have a great impact on the business processes of the critical infrastructure (Queiroz et al., 2011).

In order to tackle such issues, the option of developing a testbed using an emulation software is quite appealing. Despite the fact that some emulation tools may be an expensive choice, on average we can reduce the implementation cost. Moreover, we can be more flexible since we are able to load saved projects, revert a virtual machine to a previous state, or eliminate the risk factor of causing impact on the operation of a real-world critical infrastructure. Our work aims to help research community to pinpoint which of the

available emulation tools can be suitable for corresponding purposes, budget and we attempt to address any concerns researchers may have.

1.2 Structure of the Work

The remainder of the paper is structured as follows. In Section 2 we outline the methodology used to carry out this work. In Section 3 we provide an overview of the industrial control systems, protocols and standards. In Section 4 we present nine tools used for the emulation of an ICS, while in Section 5 we thoroughly compare these tools. Finally, the paper ends with a few concluding remarks.

2 RESEARCH METHODOLOGY

For conducting a transparent and reproducible overview of the scientific literature regarding emulation tools for industrial environments, we utilize certain features of the PRISMA statement (Page et al., 2021). PRISMA approach is consisted of four steps: (a) define the work protocol, (b) identify studies based on targeted searches, (c) evaluate the selected studies, and (d) extract data, synthesize the main findings, and report the results.

2.1 Research Objectives and Strategy

As mentioned earlier, during the first phase of our study we defined our research questions, that helped us find publicly available information about emulation tools for industrial environments. Based upon our properly formulated research questions (see Table 1),

Table 1: Research questions and objectives of the work.

Research Question	Objectives
RQ1: What are the reasons to support the use of emulation software tools?	The objective here is to better understand the problem and identify why it will be helpful for research community to use emulation software tools.
RQ2: What are the available software tools that emulate ICS systems?	The aim is to identify the emulation software that focus on ICS systems.
RQ3: What are the emulation tools that can be used for research purposes?	The objective is to identify and present this software that can be used by research community.

we conducted a systematic literature review from January 2023 to August 2023. To retrieve relevant scientific literature, we used widely known academic search systems, including Google Scholar, Scopus, and Web of Science. Moreover, Google's search engine was used to extract relevant standards and best practices (grey literature). Table 2 shows the queries we used at all search systems.

Table 2: Keywords used during the search phase.

Scientific literature	Grey literature
("ICS" OR "SCADA") AND ("emulation software" OR "emulation tool") AND ("free of charge" OR "free" OR "open source") AND ("SCADA" OR "PLC" OR "RTU")	SCADA emulation, ICS emulation, industrial environment emulation

After evaluating the initial 200 results obtained from Google, we identified the available grey literature. We limited the inclusion to this number because, beyond this threshold:

- The Google query produced numerous irrelevant and low-quality results with minimal impact, as outlined in our exclusion criteria;
- not all actual results were accessible due to broken or inactive hyperlinks.

Google searches served as a supplementary search strategy, while Scopus was our main source. The quantity of documents retrieved from Google was relatively small compared to the bibliography obtained from Scopus. In order to effectively handle the vast volume of relevant literature and ensure that our review is thorough and conforms to the high standards of academic integrity, we created a list of inclusion and exclusion criteria which are applied at several stages.

The inclusion criteria on the first stage referred to whether the title is aligned with the research focus, while on the second stage we focused on how useful and relevant each study was regarding on both the abstract and the introduction. At the final stage we considered how applicable each publication was after comprehensive full-text reading.

The exclusion criteria on the first stage referred to research papers, book chapters and scientific articles that lacked peer review, publications not written in English, and studies lacking abstracts or introductions. On the second stage we eliminated articles that seemed relevant but were out-of-scope upon closer review, reports from organizations lacking recognized national or international status, and publications by authors not affiliated with reputable scientific communities or lacking citations and references.

Papers from repositories like arXiv are not excluded, because even though they may not meet all publication standards, some studies are considered credible (Xarhoulacos, 2021).

2.2 Selection of Studies and Analysis

According to PRISMA statement (Page et al., 2021), the four stages for the selection of literature are: (a) identification, (b) screening, (c) eligibility, and (d) inclusion. In Figure 1 we present the number of documents we retained on each stage. On the identification stage, we gathered 65 documents in the academic field. However, we excluded 13 of them since those were written in languages we could not parse. During the screening stage, we removed duplicates, and afterwards we evaluated the rest documents. A total of 20 papers were removed based on their title and abstract. Finally, on the eligibility stage 13 papers were rejected after reading the complete text body.

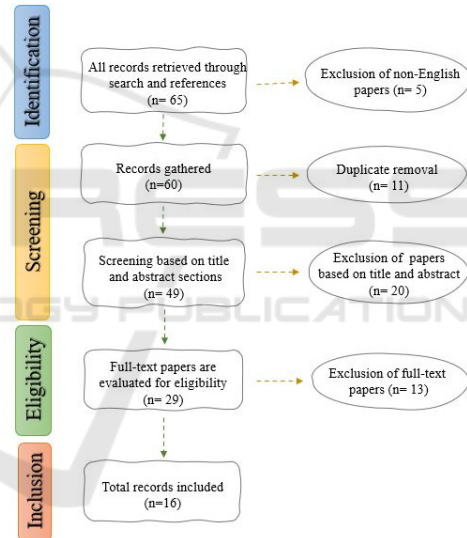


Figure 1: Flowchart of the search strategy.

Overall, in the main body of our literature review we included 16 papers. We also used additional literature for peripheral information presented in the manuscript, but they were not considered on the total number of included files as they did not address our research questions. The predominant challenge we faced was that the majority of the Google search results directed us to the websites of companies which sell emulation software.

3 INDUSTRIAL CONTROL PROTOCOLS & STANDARDS

We provide an overview of the ICS, along with the available communication protocols and standards that such systems utilize.

3.1 ICS Overview

ICS is mostly used for the mechanism of data gathering from various endpoint devices, in order to partially or fully automate the production process. Indicative types of ICS include the Process Control Systems (PCS), DCS, and SCADA (Macaulay and Singer, 2011). SCADA systems are mainly found in critical infrastructures and industrial sectors, such as water distribution systems, waste-water collection systems, or oil and gas pipelines (Macaulay and Singer, 2011). SCADA collects data from remote devices and controllers and send this information to a centralized system (Control Center), where an authorized user can connect, physically or remotely, in order to monitor and control the values in real-time.

The main components of SCADA are the control center, the distributed devices (e.g. PLC, RTU, etc.), the Human-Machine Interface (HMI), and the Master Terminal Unit (MTU). In details, HMI is located in the Control Center, along with the Data Historian, and MTU. HMI provides user with an environment that displays the monitored values, and the available actions for the configuration and control of the remote devices. MTU is the server that processes and stores the acquired data from the RTUs and PLC. The Data Historian provides a centralized database located in the control system, that supports analytics based on statistical process control techniques. Finally, Remote Terminal Unit (RTU) and Programmable Logic Controller (PLC) collect data, monitor and control sensors and actuators (Macaulay and Singer, 2011).

3.2 Communication Protocols

3.2.1 RS-485

RS-485 bus standard is used in the physical layer. It transfers small blocks of data over long distances (up to 1219m) at high speeds (≤ 10 Mbps) (Marais, 2008) (Axelson, 1999). The main advantage of this standard is the ability for communication in electrically noisy environments, and the support of multiple devices on the same bus (CUI Devices, 2020). The protocol was designed years before the development of Ethernet, and thus the security aspect was not a priority.

3.2.2 Modbus

Modbus is a publicly available protocol. A central master sequentially requests status information from each connected device (Modbus Organization, 2006), (OPS Telecom, 2023). Modbus is implemented in application layer, used for real-time communication and monitoring. Modbus was designed for serial communication and was later extended to run over TCP (Drias et al., 2015). Modbus provides two types of communication: (1) query/response between a master and a slave, and (2) broadcast communication, where the master sends a command to its slaves (Fovino et al., 2009). Its main advantage is that it can be included in a wide range of device types from any equipment vendor (OPS Telecom, 2023). The drawback of Modbus is the lack of security features, as it was originally developed without considering security aspects (Stouffer, 2006).

3.2.3 DNP3

Distributed Network Protocol (DNP3) is another protocol implemented in application layer. It is primarily designed to simplify the communication across various types of data acquisition and control systems. Three types of communications are defined in DNP3 protocol: i) unicast transaction, ii) broadcast transaction, and iii) unrequested responses from remote devices (East et al., 2009). In unicast transaction, the master can make a request to a targeted destination slave, and the slave device responds back with a message. In broadcast transaction, the master broadcasts a request towards all its slaves within the network, and in this instance, the slaves do not reply. The last type of communication is typically periodic, unrequested (by the master) updates or alerts are sent from the remote devices to the master (East et al., 2009).

3.2.4 HART

Highway Addressable Remote Transducer (HART) protocol is mostly applied to transmitters located in hazardous environments (petrochemical, pharmaceutical, chemical industries). Some of the published products provide a Bluetooth HART modem for the transmitters to connect, in order to remotely configure the transmitters while being in a dangerous environment, without the need of physical access to configure them (Yu et al., 2018). HART is a combination of analog and digital industrial automation protocol.

WirelessHART was later released to enhance wireless capabilities to HART technology while maintaining compatibility with pre-existing HART devi-

ces. WirelessHART is designed to use mesh networking technology. In a mesh network, each device can serve as a router for messages coming from other devices. Thus, a device can forward a message to the next closest one instead of communicating directly with a gateway. This extends the range of the network and provides redundant communication routes to increase reliability (Song et al., 2008).

3.2.5 ICCP/TASE 2.0

Inter-control Center Communications Protocol, also referred as Tele-control Application Service Element (TASE 2.0), is a protocol that allows communication and data exchange between different control centers over Local Area Networks and Wide Area Networks (Cunha et al., 2004), (Ilgner et al., 2021). TASE 2.0 relies on Manufacturing Message Specifications in order to transfer data and monitor network nodes. TASE 2.0 does not support authentication or encryption; instead, it implements an ingrained security suite of underlying TCP/IP stack (Ilgner et al., 2021).

3.2.6 CIP

Common Industrial Protocol (CIP) is an object-oriented protocol that transfers data between communication objects. An object is consisted of attributes, services, connections, and behaviors. The protocol also includes an extensive library that supports (a) typical automation functions (e.g. analog and digital input/output devices, HMI, motion control and position feedback etc.), (b) general purpose network communications, and (c) network services (e.g. file transfer etc.) (Weehuizen et al., 2007), (ODVA, 2006).

3.2.7 BACnet

Building Automation & Control Network (BACnet) is a protocol that was initially designed for heating, ventilation, and air conditioning (HVAC) systems, and was later extended to support further functionalities (Esquivel-Vargas et al., 2017). BACnet solves interoperability issues among devices from different vendors, by modelling exchanged information with object-oriented representations (Tang et al., 2020).

An object in BACnet is defined as a collection of information related with the functionality of a BACnet device (Newman, 2013). Although BACnet was not primarily designed with security in mind, there are available options to secure BACnet, such as through IP security solutions, IPsec, etc. (Peacock et al., 2017), (Peacock, 2019).

3.3 Communication Standards

We introduce the standards encountered during the communication of different components within an Industrial Control System (ICS).

3.3.1 IEC 60870-5-104 (IEC 104)

IEC 104 was developed on top of the serial communication standard IEC 60870-5-101 (IEC 101).

Standard IEC was originally developed to enable basic tele-control messages between a control station and outstations, over a communication link between them (e.g. telephone network, modem circuit) (Mai et al., 2019). IEC 104 is implemented in application layer and is based on TCP/IP. However, this makes it prone to attacks since it carries all the security issues of TCP/IP. Moreover, the lack of encryption on the transmitted data in the application layer is another security concern, making it vulnerable to Man-in-the-Middle (MITM) attacks (Radoglou, et al., 2019).

3.3.2 IEC 61850

IEC 61850 is an international standard that defines a comprehensive framework for communication between devices and systems used in substations (e.g., control systems). One of its goals is to improve the interoperability and integration of devices and systems in substation automation systems. It can support both Ethernet and serial communication protocols, and thus it provides an easier integration between various devices across different vendors. It is commonly implemented in electric power industries (Mackiewicz, 2006). An important aspect of IEC 61850 is the security guidelines that are included and focus on: i) authentication and access control, ii) data integrity and confidentiality, iii) network security, and iv) risk assessment and management (Hussain et al., 2019). On the other hand, IEC 61850 proposed SNTTP protocol that has less accuracy, while in time-critical applications (such as in industrial sector) there is need for accurate protocols to be applied (Sidhu et al., 2008).

3.3.3 ISO 15745

ISO 15745 proposes a framework for the development of communication profiles for industrial automation systems. The standard outlines the conditions for the creation and application of communication profiles, which are used in industrial automation environments to guarantee interoperability between devices and systems. A communication profile is a set of rules and protocols in order to describe how devices communicate in a system (Kosanke, 2006). Although

this standard provides rules and profiles as a framework for the development of communication protocols (which could be designed to meet security requirements), it does not explicitly deal with security requirements of industrial automation systems.

3.3.4 EN 62443

EN 62443 is a set of industrial cybersecurity standards developed by the International Electro-technical Commission (IEC) and the International Society of Automation (ISA). EN 62443 was designed to provide a comprehensive framework for securing industrial automation and control systems against cyber threats. The standard includes criteria for cybersecurity management systems, risk assessment, and incident response plans, along with a list of recommendations and best practices for safeguarding the network and devices of industrial automation and control systems. Due to restrictions on cost and resources, it is difficult to effectively address all security issues as required by IEC 62443 (Maidl et al., 2018).

4 EMULATION SOFTWARE

In this section we make an analysis of nine software tools that can be used to realistically emulate complex industrial processes or entire plants.

4.1 IEC Server & QTester104

IEC Server is a software, written in Java, that emulates field devices (such as an RTU) and SCADA, implementing a telecontrol message protocol specified in the IEC 60870-5 (Parcharidis, 2018). The software is free of charge, and it is simple to operate. The tool comes as a portable executable and does not need to be installed, reducing the configuration time compared to the other emulation tools. A drawback of this software is that the documentation is not publicly available because it is hosted on a website that is currently unavailable. Despite the lack of a manual, the tool includes several interesting actions, such as add IEC a 60870-5-104 command from a short list, pause or resume the emulation, start or stop the server that listens to a specified port, or even save a preset of a given configuration.

QTester104 is a software designed to receive data from a field device, based on the IEC 60870-5-104 communication protocol, in a SCADA system (Parcharidis, 2018). This software is also free of charge and can be compiled on Linux and Windows platforms.

There is adequate documentation that explains both the tool functionalities and the user interface.

4.2 OMNeT++

OMNeT++ (Objective Modular Network Testbed in C++) is an extensible, modular, component-based C++ simulation framework. It is primarily used for building network simulations. OMNeT++ is an open-source tool with extensive documentation. It can be used for free and for non-commercial purposes, such as for academic institutions or teaching (Ahmad and Durad, 2019). The OMNeT++ simulation kernel is standard C++ and runs on all platforms where C++ compiler is available. The simulation IDE requires Windows, Linux, or macOS. A project in OMNeT++ consists primarily of three files. A NED file that contains the topology of the network, an INI file that contains the configuration of the simulation, and the source code file which is written in C++ and manages the simulation (Ahmad and Durad, 2019). The project is compiled using the NEDC compiler, included in the OMNeT++ (Varga, 2005).

The greatest advantage of this software is its extensive documentation, provided both by researchers and the community. One negative point is that the configuration needed for the creation of a new project can be overwhelming, since there are many components that have to be included and users must have adequate knowledge of the C++ language.

4.3 RINSE

RINSE (Real-time Immersive Network Simulation Environment) is used to conduct real-time emulation for network-security purposes (Liljenstam et al., 2005). RINSE can emulate large networks, along with a great number of attacks and defensive measures (Davis et al., 2006). RINSE is composed of five basic elements: i) iSSFNet network simulator, ii) simulator database manager, iii) database, iv) data server, and v) client-side network viewers. The iSSFNet is a network simulator that runs over the Scalable Simulation

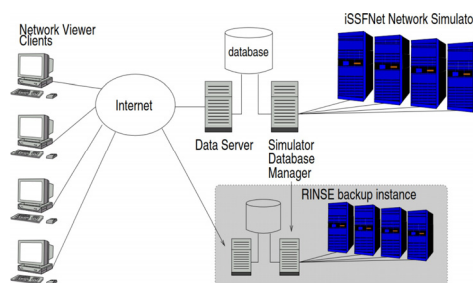


Figure 2: RINSE architecture.

Framework (iSSF), an Application Programming Interface (API) which is responsible for the synchronization and the functionality support. The iSSFNet is running on parallel machines and supports the simulation of large-scaled networks. The simulator database manager is responsible for supervising the data gathering from the simulation nodes, in order to store them in the database. These data are later delivered to the simulator. The data server allows users to monitor and control the simulated network. Finally, the network viewers are running on the clients, allowing users to gain a local view of the network (Liljenstam et al., 2005). RINSE architecture is depicted in Figure 2.

Through the network view clients, the users can execute basic commands that modify the simulation. Such commands can be categorized, based on their functionality:

- **Attack:** Commands used for launching attacks (such as DDoS, worms).
- **Defense:** Commands used for applying countermeasures (such as packet filters).
- **Diagnostic networking tools:** Commands for basic networking communication.
- **Device control:** Commands for the control of the devices (such as restart, reboot).
- **Simulator data:** Commands sent to the simulator in order to modify the output.

Although RINSE appears to be a promising simulation software tool (highly scalable, with multiple functionalities etc.), it has limited options to the commands that can be used for attacks. Most of the attack commands revolve around denial of service.

4.4 GRFICS

GRFICS (Graphical Realism Framework for Industrial Control Simulations) is a simulation tool designed specifically for ICS. It can be customized to meet user needs. GRFICS helps users understand ICS protocols through virtual network environments. This accessibility aims to improve community knowledge in ICS security. Users can simulate and observe cybersecurity attacks like command injection, man-in-the-middle attacks, and buffer overflows through 3D visualizations. Finally, the tool offers the chance of practicing defensive strategies by deploying proper firewall and intrusion detection rules within the virtual network. (Slatman H).

GRFICS allows users to swap components (such as PLCs, HMI and any I/O module) with real ICS devices. In order to emulate its physical processes, the software utilizes: i) the emulation backend, ii) the emulation API, iii) the 3D visualization, and iv) the I/O

modules (Formby et al., 2018). In regard to the visualization of the PLC, developers used a modified version of the OpenPLC. OpenPLC is an open-source software for virtualizing controllers, which supports multiple communication protocols (e.g., IEC 61131-3, Modbus/TCP or DNP3). For the HMI's visualization, developers implemented Advanced HMI, an open-source software allowing HMI virtualization.

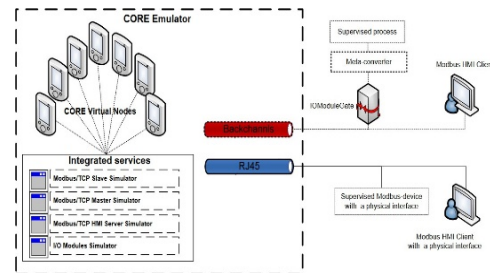


Figure 3: SCADA architecture.

Overall, GRFICS is a great framework that combines all the essential components: a virtualized network of PLC, HMI, router, and a workstation.

It is an ideal tool for researchers that want to conduct basic security attacks and defensive countermeasures at no cost. GRFICS can be downloaded from the GRFICS Git-Hub repository. Also, there is available an adequate documentation, as well as video tutorials on how to configure the virtual machines.

4.5 SCADA VT

SCADA VT is a security framework that is targeted to security experts of SCADA systems (Almalawi et al., 2013). It was developed on top of the CORE emulator (Ahrenholz, 2010). CORE is a network emulator, similar to OMNET++, OPNET, QualNet, NetSim, SSF Net, NS2 (NetWork Simulator 2) and NS3 (Network Simulator 3) (Ahmad and Durad, 2019), (Pan and Jain, 2008). Since the CORE emulator does not support the commonly used SCADA protocols, the creators of the testbed developed three essential components, which were integrated as services within the CORE emulator. These components are:

- **Modbus/TCP Simulators of Master/Slave:** Utilizes the master-slave architecture essential for SCADA systems. SCADA VT supports the Modbus protocol, and the modes of master-slave are integrated into the CORE emulator, using the Modbus library and python scripts.
- **Modbus/TCP Simulator of HMI Server:** Acts as the communication medium between the HMI client and MTU, facilitating a two-way communication for command exchanges.

- I/O modules Simulator: Acts as a server, and is in charge of receiving input data, from the external environment, and sending it to the requesting nodes.

In order to utilize the SCADA framework, the user should install the following components: i) the CORE emulator, ii) a third party publicly accessible Modbus library, iii) a Python interpreter, iv) a security tool (such as hping3), and v) the integration Python scripts, developed by the creators (Ahrenholz, 2010). The architecture of SCADA framework is shown in Figure 3. Because the CORE emulator is a GUI-friendly solution that does not require code to configure a network topology, the absence of the python scripts used to implement the required SCADA components into the emulator shifts the burden of developing new python scripts to end-users.

4.6 TASSCS

TASSCS (Testbed for Analyzing Security of SCADA Control Systems) is a testbed developed at the NSF Center for Autonomic Computing at the University of Arizona, aiming to help the security research, by providing innovative protection techniques for SCADA systems (Mallouhi et al., 2011). TASSCS uses three tools: (i) OPNET, a long-used by the industry commercial network simulator (Pan and Jain, 2008), (ii) PowerWorld that simulates the operations of the electrical power grid, and (iii) Autonomic Software Protection System (ASPS) whose role is to protect SCADA system and its network from the tested attacks (Mallouhi et al., 2011). Modbus RSim is a tool that helps users emulate PLC devices (Modbus server). The Modbus server is combined with the PowerWorld server and OPNET in order to listen incoming requests (Mallouhi et al., 2011). In Figure 4 we depict how the components of TASSCS are connected.

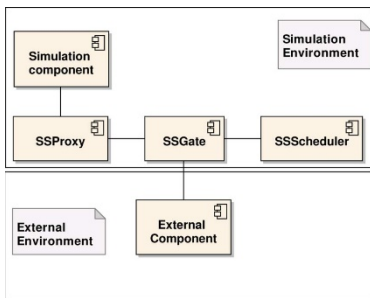


Figure 4: TASSCS architecture.

The TASSCS architecture has three components: (i) Control HQ, (ii) WAN, and (iii) Energy Field. Through the Control HQ, users can control all the available resources within the emulated environment,

along with the provided services. This component allows the presentation and storage of the collected data (e.g. historical data from the devices and sensors). Moreover, through Control HQ the end-users can manage the grid’s resources. The WAN component consists of multiple emulated sensors, such as PLC, RTU etc. These devices provide SCADA with the required data and execute the requested commands from the control center through the HMI. Finally, the Energy Field serves as the electrical grid controlled by the SCADA system. Through this component, developers would showcase the effectiveness of the ASPS, as it prevents the launched attacks, and minimizes the impact on the operations of the grid.

TASSCS has a great potential as it allows users to test various attack scenarios and has defensive capabilities which permit users to study the detection and prevention aspect, through the ASPS.

4.7 SCADASim

SCADASim is a framework used for emulating SCADA systems, developed at the Royal Melbourne Institute of Technology in Australia. SCADASim is an all-in-one, plug and go emulator, that utilizes the OMNET++ discrete event simulation engine (Qassim et al., 2017). SCADASim was developed with three key requirements (Queiroz et al., 2011): (a) no need for programming skills, (b) connectivity to multiple external devices (both hardware and software), and (c) support of multiple industry standard protocols (e.g. Modbus/TPC, DNP3).

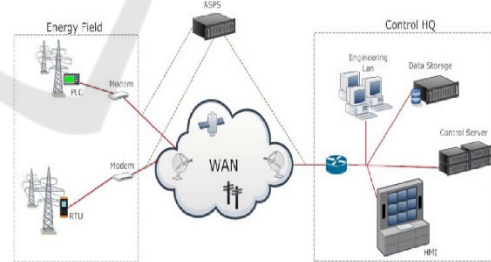


Figure 5: SCADASim architecture.

As depicted in Figure 5, SCADASim’s architecture consists of three essential components: i) SSScheduler, ii) SSGate, and the iii) SSProxy. The SSScheduler is a real-time scheduler, that allows users to add new schedulers to the OMNET++ simulator. Users can control and synchronize the messages that receive from the external environment. SSScheduler manages the SSGate instances, which are responsible to send and receive messages from the external environment. SSGate is the communication link to the ex-

ternal environment, where external SCADA components are allowed to connect, through a supported communication protocol.

Currently SCADA Sim supports three types of gates: ModbusGate, DNP3Gate, and HTTPGate. SS-Proxy is a representation of a real device or an external application with-in the emulated environment, and it communicates with the emulated objects (e.g. PLC, RTU, MTU) through the SSGate which routes their messages (Queiroz et al., 2011). SCADASim allows many attacks to be launched on the emulated environment, such as: Denial of Service, Man in the Middle, Spoofing, and Eavesdropping etc. Overall, SCADA Sim has an easy configuration, with adequate documentation provided.

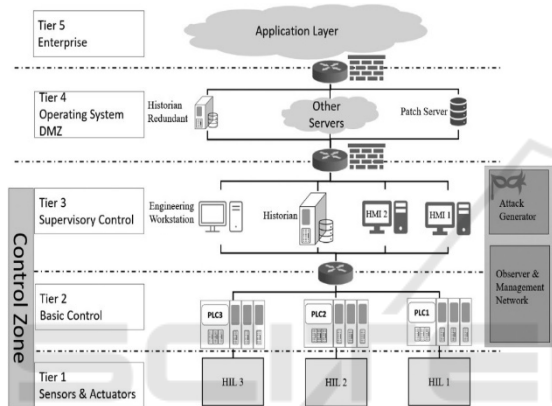


Figure 6: ICSSIM reference architecture.

4.8 ModbusPal & Rodbus

ModbusPal is a free, open-source emulation software written in Java, which supports both natively TCP/IP communication and serial communication as well. The user can include up to 247 slaves, and each slave can hold both registers and coils. ModbusPal can dynamically generate missing resources (slaves, registers and coils) as it receives requests from the master. ModbusPal supports automations, where an automation is defined as a generator that creates the values (through Linear, Random, and Sine generators) with a predefined step, and these values can be bound with a register or a coil. ModbusPal by default listens on port 502, which can be easily changed through the user interface. A slave can have its own IP address in order to be identified in the network. Finally, a project can be saved in an XMPP format so that the user can load it again on the emulator.

Rodbus (Rust and Modbus) is a Rust implementation of the Modbus protocol. It is a command-line tool written in Rust and can be easily installed using Car-

go. Rodbus provides in simple syntax all the essentials commands that could be used by the master, such as reading coils and registers, writing both single and multiple coils or registers, etc.

4.9 ICSSIM

ICSSIM is a python-based emulation testbed, created for security-research purposes. Several components of the emulation are created as Docker containers that may either share resources (such as a shared SQLite database for “hardwired” communication) or communicate across the network using configurable, private IP addresses. It supports Modbus TCP and includes classes that might be extended to support new protocols. The ICSSIM is capable of supporting both hardware and emulated ICS components, such as a PLC (Dehlaghi-Ghadim et al., 2023). Figure 6 presents the ICSSIM reference architecture.

The Attack Generator, a key component of this testbed, enables for the emulation of an adversary within the environment, assuming that the attacker already has a foothold. Because this Attack Generator is a Docker running Kali Linux, attackers can launch a variety of attacks. This emulation software does not require highly technical skills to operate, but basic containerization understanding is required. Finally, programming skills would help end-users to increase the emulation capabilities of the tool.

5 EMULATION SOFTWARE COMPARISON

In this section, we undertake a comparison of the emulation tools based on the criteria outlined in Table 3. Due to the software's modularity, most emulation tools can model any protocol as needed by researchers. The term "modularity" here refers to a software design approach that divides the program's functionality into independent, interchangeable modules. As indicated in Table 4, half of the tools have the capability to simulate any protocol requested by the user.

Specifically, SCADA-VT, TASSCS, ModbusPal & Rodbus, and ICSSim are dedicated to the Modbus/TCP protocol, with TASSCS also incorporating DNP3. On the other hand, IEC Server & QTester104 exclusively support the IEC 60870-5 protocol.

Table 3: Description of comparison criteria.

Criterion	Description
Modularity	Modularity of the software defines whether an emulation software could model any protocol that a researcher desires.
Sector	The sectors that an emulation tool is focused.
Attack variety	How flexible is to emulate several attacks or is specified to a single attack.
Open source	The capability of a tool and its source code to be used, altered, or distributed to anyone and for any purpose.
Free of charge	Whether a researcher should purchase the emulation tool.
Complexity	Whether the use of an emulation tool requires higher technical knowledge (e.g. programming skills) from users.
Scalability	Factors that indicate scalability of a tool: number of emulated devices, whether Firewalls, Intrusion Detection Systems etc., could be used in the emulation.
Easy to configure	How easy is for a researcher, may not be familiar in a great extend with programming and engineering
Flexibility	The degree of flexibility is determined by how configurable the environment is, the variety of attacks that can be launched.
Documentation	A tool offers a good experience if the user can easily find the details of how to configure and use it.

Pricing and accessibility of software are crucial aspects in comparison. Some tools are proprietary and intended for use by specific organizations. Fortunately, a significant number of tools are freely available and open-source. As shown in Table 4, IEC Server & QTester104, OMNeT++, GRFICS, SCADASim, ModbusPal & Rodbus, and ICSSim are all examples of tools that are both free of charge and open-source. Additionally, we considered specific sectors applicable by these tools. According to Table 4, most tools are not sector-specific. Only two tools have a concentrated focus on specific areas. GRFICS targets the chemical sector, while TASSCS is designed for use in the chemical and energy sectors.

A feature that is beneficial to users is the presence of adequate documentation. Unfortunately, only four tools have accessible documentation, which are OMNeT++, RINSE, ModbusPal & Rodbus, and ICSSim. Furthermore, it is important the tools to be kept up to date. This is because when the tools integrate updates, then enhance existing features, fix bug issues, or improve their performance. Based on Table 5, the most up-to-date tools are the IEC Server, the OMNeT++, the Rodbus, and the ICSSim. The GRFICS, SCADA VT, and TASSCS are no longer available.

Researchers aim to emulate environments not only for testing but also for launching attacks and drawing conclusions without impacting real-world infrastructures. Therefore, we evaluate the presented software based on the criterion of attack variety. Most of the tools meet this criterion, except IEC Server & QTester104, RINSE, and SCADA VT. In terms of flexibility and scalability, as indicated in Table 4, IEC

Server & QTester104, SCADA VT, and ModbusPal & Rodbus are identified as less scalable tools. Specifically, IEC Server & QTester104 and SCADA VT are noted as the least flexible among the tools.

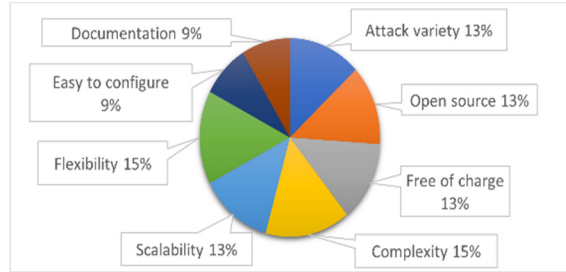


Figure 7: Number of criteria per emulation software.

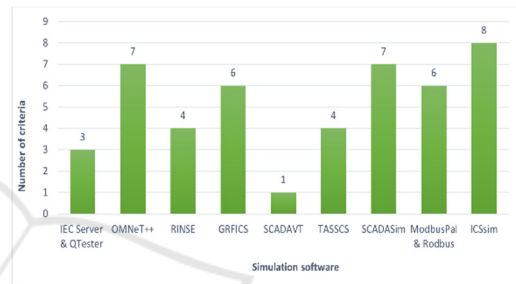


Figure 8: % of tools that incorporate each criterion.

Figure 7 indicates how many emulation tools incorporate a particular criterion. We notice that the larger percentage is attributed to complexity and flexibility. Conversely, the lowest percentage suggests that developers are not concerned with the ease of configuration and the documentation provision. Based on the above observations, we can assume that software developers focus enough on adding new features, but they sacrifice the ease of configuring the tool. Have in mind that the user, in our case a researcher, may not be familiar to a great extend with programming and engineering and customization adds complexity to execution.

Figure 8 depicts the number of criteria that each of the presented tools incorporates. We notice that only the ICSSim meets all the requirements. The second place, with 7 out of 8 incorporated criteria, belongs to the OMNeT++ and the SCADASim. The tool that satisfies only one requirement is SCADA VT.

Researchers are flexible to choose emulation software based on their expertise, specific emulation needs, and budget constraints. This analysis suggests ICSSim as a recommended choice that meets all specified criteria. Alternatively, researchers may consider either OMNeT++ or SCADASim, both satisfying 7 out of 8 criteria. The choice between these two tools depends on individual needs. OMNeT++ offers extensive

Table 4: Emulation Software Comparison.

Emulation software	Modularity	Sector	Attack variety	Open source	Free of charge	Complexity	Scalability	Flexibility	Easy to configure	Documentation
IEC Server & QTester104	IEC 60870-5	Generic	-	✓	✓	-	-	-	✓	-
OMNeT++	Any protocol	Generic	✓	✓	✓	✓	✓	✓	-	✓
RINSE	Any protocol	Generic	-	-	-	✓	✓	✓	-	✓
GRFICS	Any protocol	Chemical	✓	✓	✓	✓	✓	✓	-	-
SCADA VT	Modbus/TCP	Generic	-	-	-	✓	-	-	-	-
TASSCS	Modbus/TCP, DNP3	Energy, Chemical	✓	-	-	✓	✓	✓	-	-
SCADASim	Any protocol	Generic	✓	✓	✓	✓	✓	✓	✓	-
ModbusPal & Rodbus	Modbus/TCP	Generic	✓	✓	✓	-	-	✓	✓	✓
ICSSim	Modbus/TCP	Generic	✓	✓	✓	✓	✓	✓	✓	✓

documentation despite being challenging in configuration, while SCADASim is easy to configure but lacks accompanying documentation.

Table 5: Software Latest Updates.

Emulation software	Release date	Latest update
IEC Server	February, 2018	June 27, 2023
QTester	April, 2016	November 10, 2022
OMNeT++	December 2018	July 5, 2023
GRFICS	N/A	N/A
RINSE	May, 2018	December 14, 2020
SCADA VT	N/A	N/A
TASSCS	N/A	N/A
SCADASim	December, 2018	November 9, 2020
ModbusPal	March, 2009	February 20, 2018
Rodbus	August, 2019	April 27, 2023
ICSSim	April, 2022	February 2, 2023

6 CONCLUSIONS

Our work aimed to identify and present tools that can be used for the emulation of complex ICS infrastructures and systems, such as SCADA. We made a thorough analysis and comparison among these tools based on a number of criteria regarding the accessibility to the tool, flexibility in terms of the emulation components, configuration complexity, latest updates of the software, pricing, scalability, as well as the accessibility to documentation and manuals.

For an effective tool selection, a user should first define requirements per use-case and specific needs of the facility that requires the emulation tool. This includes understanding the type of ICS in use (e.g., SCADA, PLC, DCS), critical processes and components that must be emulated, along with the overall objectives of using the tool (e.g., training, vulnerability testing, system analysis).

At a minimum, plant users should choose based on (i) protocol support regarding the communication protocols used in the existing ICS (e.g., Modbus, IEC 61850), (ii) actuator and sensor emulation support to

simulate the types of equipment used in the system, and (iii) material mapping capabilities to be relevant to the processes (e.g., chemical properties in a processing plant).

Some of the simulation software provide high flexibility regarding the simulation components. This may cost the ease of using and configuring the software and leads to an increased level of complexity, requiring higher technical knowledge (e.g. programming skills) from users. For example, OMNeT++, offers simulation of any desired component within an industrial control system, supporting any communication protocol, due to its modularity. However, it requires considerably higher technical skills, compared to other simulation solutions. The pricing is another concern that we have in mind. Most of the emulation software is free of charge, where others are either not publicly available, or require a license.

Finally, we aimed at addressing the challenging dilemma of whether someone chooses an emulation software instead of physically designs a testbed on a research lab. The objective of our analysis is to help research community identify which of the existing emulation tools can be suitable for corresponding purposes and budget.

REFERENCES

- Ahmad, Z., Durad, M. (2019). Development of SCADA simulator using omnet++. In 2019 16th International Bhurban Conference on Applied Sciences and Technology, pg. 676–680. IEEE.
- Ahrenholz, J. (2010). Comparison of core network emulation platforms. In 2010 Military Communications Conference, pg. 166–171. IEEE.
- Almalawi, A., Tari, Z., Khalil, I., Fahad, A. (2013). Scadavt-a framework for scada security testbed based on virtualization technology. In 38th Annual IEEE Conference on Local Computer Networks, pg. 639–646.
- Axelsson, J. (1999). Designing rs-485 circuits. *Circuit Cellar*, 107:20–24.

- Coffey, K., Smith, R., Maglaras, L., Janicke, H. (2018). Vulnerability analysis of network scanning on SCADA systems. *Security and Communication Networks*, 2018.
- CUI Devices (2020). RS-485 serial interface explained.
- Cunha, C., Rein, O., Jardini, J., Magrini, L. (2004). Electrical utilities control center data exchange with iccp and cim/xml. In *2004 IEEE Transmission and Distribution Conference & Exposition*, pg. 260–265. IEEE.
- Davis, C., Tate, J., Okhravi, H., Grier, C., Overbye, T., Nicol, D. (2006). SCADA cyber security testbed development. In *2006 38th North American Power Symposium*, pg. 483–488. IEEE.
- Dehlaghi-Ghadim, A., Balador, A., Moghadam, M., Hansson, H., Conti, M. (2023). Icssim - a framework for building industrial control systems security testbeds. *Computers in Industry*, 148:103906.
- Drias, Z., Serhrouchni, A., Vogel, O. (2015). Taxonomy of attacks on industrial control protocols. In *2015 International Conference on Protocol Engineering*, pg. 1–6. IEEE.
- East, S., Butts, J., Papa, M., Sheno, S. (2009). A taxonomy of attacks on the dnp3 protocol. In *International Conference on Critical Infrastructure Protection*, pg. 67–81. Springer.
- Esquivel-Vargas, H., Caselli, M., Peter, A. (2017). Automatic deployment of specification-based intrusion detection in the bacnet protocol. In *Proc. of the 2017 Workshop on Cyber-Physical Systems Security and Privacy*, pg. 25–36.
- FieldComm Group. Wirelesshart.
- Formby, D., Rad, M., Beyah, R. (2018). Lowering the barriers to industrial control system security with {GRFICS}. In *2018 USENIX Workshop on Advances in Security Education*.
- Fovino, I., Carcano, A., Masera, M., Trombetta, A. (2009). Design and implementation of a secure modbus protocol. In *International conference on critical infrastructure protection*, pg. 83–96. Springer.
- Hussain, S., Ustun, T., Kalam, A. (2019). A review of iec 62351 security mechanisms for iec 61850 message exchanges. *IEEE Transactions on Industrial Informatics*, 16(9):5643–5654.
- Ilgner, P., Cika, P., Stusek, M. (2021). Scada-based message generator for multi-vendor smart grids: Distributed integration and verification of tase. 2. *Sensors*, 21(20):6793.
- Incorporated, A. (2005). BusWorks® 900EN Series: 10/100M Industrial Ethernet I/O Modules w/Modbus.
- Kosanke, K. (2006). Iso standards for interoperability: A comparison. In *Interoperability of enterprise software and applications*, pg. 55–64. Springer.
- Liljenstam, M., Liu, J., Nicol, D., Yuan, Y., Yan, G., Grier, C. (2005). Rinse: The real-time immersive network simulation environment for network security exercises. In *Workshop on Principles of Advanced and Distributed Simulation*, pg. 119–128. IEEE.
- Macaulay, T., Singer, B. L. (2011). *Cybersecurity for industrial control systems: SCADA, DCS, PLC, HMI, and SIS*. CRC Press.
- Mackiewicz, R. E. (2006). Overview of iec 61850 and benefits. In *2006 IEEE Power Engineering Society General Meeting*, pg. 8–10. IEEE.
- Mai, K., Qin, X., Silva, N., Cardenas, A. (2019). Iec 60870-5-104 network characterization of a large-scale operational power grid. In *2019 IEEE Security and Privacy Workshops*, pg. 236–241. IEEE.
- Maidl, M., Kroselberg, D., Christ, J., Beckers, K. (2018). A comprehensive framework for security in engineering projects-based on iec 62443. In *2018 IEEE International Symposium on Software Reliability Engineering Workshops*, pg. 42–47. IEEE.
- Mallouhi, M., Al-Nashif, Y., Cox, D., Chadaga, T., Hariri, S. (2011). A testbed for analyzing security of SCA-DA control systems. In *ISGT 2011*, pg. 1–7. IEEE.
- Marais, H. (2008). Rs-485/rs-422 circuit implementation guide. AN-960 Analog Devices.
- Mattioli, R., Moulinos, K. (2015). Analysis of ICS-SCADA cyber security maturity levels in critical sectors.
- Modbus Organization (2006). *Modbus messaging on TCP/IP Implementation Guide V1*.
- Journal on Wireless Communications and Networking*, 2012(1):1–17.
- Newman, H. (2013). Bacnet explained. *ASHRAE Journal*.
- ODVA (2006). *Common Industrial Protocol (CIP)*.
- OPS Telecom (2023). *Understanding modbus protocol - rtu vs tcp vs ascii*.
- Page, M., et al. (2021). The Prisma 2020 statement: An updated guideline for reporting systematic reviews. *88: 105906*.
- Pan, J., Jain, R. (2008). A survey of network simulation tools: Current status and future developments. *2(4):45*.
- Parcharidis, M. (2018). Simulation of cyber-attacks against scada systems.
- Peacock, M. (2019). Anomaly detection in bacnet/ip managed building automation systems.
- Peacock, M., Johnstone, M., Valli, C., Camp, O., Mori, P., Furnell, S. (2017). Security issues with bacnet value handling. In *ICISSP*, pg. 546–552.
- Qassim, Q., et al., (2017). A survey of SCADA testbed implementation approaches. *Indian Journal of Science and Technology*, 10(26):1–8.
- Queiroz, C., Mahmood, A., Tari, Z. (2011). Scada-sim - A framework for building scada simulations. *IEEE Transactions on Smart Grid*, 2(4):589–597.
- Radoglou-Grammatikis, P., Sarigiannidis, P., Giannoulakis, I., Kafetzakis, E., Panaousis, E. (2019). Attacking iec-60870-5-104 SCADA systems. In *2019 IEEE World Congress on Services*, vol. 2642, pg. 41–46. IEEE.
- Slatman H. A curated list of resources related to Industrial Control System (ICS) security.
- Sidhu, T., Kanabar, M., Parikh, P. (2008). Implementation issues with iec 61850 based substation automation systems. In *15th National Power Systems Conference*, pg. 473–478.
- Song, J., Han, S., Mok, A., Chen, D., Lucas, M., Nixon, M., Pratt, W. (2008). *Wirelesshart: Applying wireless technology in real-time industrial process control*. In

- 2008 IEEE Real-Time and Embedded Technology and Applications Symposium, pg. 377–386. IEEE.
- Stouffer, K., Falco, J. (2006). Guide to supervisory control and data acquisition (SCADA) and industrial control systems security. NIST.
- Tang, S., Shelden, D. R., Eastman, C. M., Pishdad-Bozorgi, P., Gao, X. (2020). Bim assisted building automation system information exchange using bacnet and ifc. *Automation in Construction*, 110:103049.
- Varga, A. (2005). Omnet++ discrete event simulation system, ver. 3.2, User Manual.
- Weehuizen, F., Brown, A., Sunder, C., Hummer, O. (2007). Implementing IEC 61499 communication with the cip protocol. In 2007 IEEE Conference on Emerging Technologies and Factory Automation, pg. 498–501. IEEE.
- Xarhoulacos, C. G., Anagnostopoulou, A., Stergiopoulos, G., & Gritzalis, D. (2021). Misinformation vs. situational awareness: The art of deception and the need for cross-domain detection. *Sensors*, 21(16), 5496.
- Yu, Y.-S., Chen, C.-H., Cheng, K. (2018). Design and implementation of a remote hart configurator. In 2018 IEEE International Conference on Applied System Invention, pg. 510–512. IEEE.

