Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS USA 2025 - Selected Papers from the 25th Annual Digital Forensics Research Conference USA

Out of Control: Igniting SCADA investigations with an HMI forensics framework and the ignition forensics artifact carving tool (IFACT)

LaSean Salmon [a,b,*], Ibrahim Baggili [a,b]

[a] Baggil(i) Truth (BiT) Lab, Center of Computation & Technology, Baton Rouge, LA, USA
[b] Division of Computer Science & Engineering, Louisiana State University, Baton Rouge, LA, USA

ARTICLE INFO

Keywords:
SCADA
PLC
HMI
Digital forensics
SCADA forensics
Forensic tools

ABSTRACT

In the modern industrial landscape, Programmable Logic Controllers (PLCs) and Supervisory Control and Data Acquisition (SCADA) systems serve as critical components in the automation and control of various industrial processes. While their widespread availability and overall efficiency are crucial, the increasing integration of these systems with networked environments has exposed them to a growing array of cyber threats. Meanwhile, the rapid growth and deployment of SCADA systems worldwide pose increasing challenges to managing their security effectively. We explore the value of HMI-focused digital forensics within SCADA environments, emphasizing the unique challenges in their evaluation and the information contained in digital artifacts. We present a comprehensive forensic analysis of Ignition: a popular SCADA software platform developed by Inductive Automation. We also develop a generic forensic analysis framework that can be used when conducting a forensic investigation on an HMI environment. Our investigative process is supported with the creation of IFACT: an HMI Forensic Analysis Tool created to streamline the process of parsing system information presented in Ignition HMI-sourced forensic data. The data recovered from memory, network, and disk forensic investigations provides insight into the state of the SCADA system, including tag and PLC utilization and configurations. Using IFACT, we investigate how long this data persists in volatile memory and how its lifetime is variable.

1. Introduction

Supervisory Control and Data Acquisition (SCADA) systems are crucial for the management of Industrial Control Systems (ICS) that drive our daily lives, including food and beverage processing, manufacturing plants, transportation systems, energy facilities and wastewater treatment plants. As industrial control operations have advanced, the need for resilient, efficient, and scalable technology to support critical infrastructure has grown significantly. This demand has driven the development of Industrial Internet of Things (IIoT) SCADA systems, which are networked architectures designed for ICS.

As SCADA systems evolved to operate in networked environments, the integration of security features, such as authentication and encrypted communication, lagged behind. Despite this, the global interconnectivity of SCADA systems and the use of industry-standard software have led to a new wave of cyber attacks. Attackers exploit remote access to target insecure system configurations, standardized SCADA protocols, and component vulnerabilities, contributing to a significant rise in widespread, severe, and marketable attack techniques (Javed Butt et al., 2019). The increasing frequency and sophistication of attacks targeting SCADA systems makes advances in forensic technology essential. The growing complexity of SCADA platforms and devices highlights the need for tools and methodologies that can help analysts quickly identify and attribute attacks. Our primary objective is to establish a framework for guiding SCADA Human–Machine Interface (HMI) forensics investigations, create a reference for analyzing data traveling throughout large-scale industrial systems efficiently, and evaluate the efficacy of system fingerprinting through forensics. We accomplish this through the forensic evaluation of the Ignition SCADA system software, and the development of a tool to support investigations. Additionally, we will explore the potential for using SCADA HMI forensics to analyze multiple layers of the ICS architecture. This is essential in supporting investigations and the cross-analysis of SCADA HMI data. To achieve these goals, we defined the following research questions:

[*] Corresponding author. Baggil(i) Truth (BiT) Lab, Center of Computation & Technology, Baton Rouge, LA, USA.
E-mail addresses: lsalmo1@lsu.edu (L. Salmon), ibaggili@lsu.edu (I. Baggili).

https://doi.org/10.1016/j.fsidi.2025.301933

RQ1: Can a generic framework be developed for the collection of forensic evidence from a centralized SCADA HMI system?
RQ2: To what extent can system data be extracted from artifacts collected during a SCADA HMI forensics investigation?
RQ3: Are forensic artifacts collected from a centralized SCADA HMI environment sufficient to gather system-wide data and identify anomalous behavior across multiple layers of the SCADA architecture?
RQ4: What is the lifetime of Ignition system data in memory, and what implications does this have for forensic investigations?

In addressing these research questions, we make the following contributions:

- Development of a forensic artifact acquisition framework focusing on SCADA HMI data.
- Evaluation of SCADA HMI forensic data, demonstrating how it defines system behavior and reveals signs of cyber attacks.
- Creation of a tool for extracting artifacts from Ignition forensic data.
- Investigation of system conditions and their impact on the efficacy of SCADA HMI memory forensics.

The remainder of this paper is organized as follows: Background, Testbed Design and Setup, SCADA HMI Forensic Acquisition Framework, Forensic Acquisition, Forensic Artifact Analysis, Ignition Forensic Artifact Carving Tool, Evaluation, Discussion, Related Work, Conclusion and Future Work.

2. Background

SCADA systems rely on various components and design structures to maintain smooth and efficient functionality. This section elaborates on these key components and examples of how they can be exploited to compromise SCADA systems.

2.1. Programmable Logic Controllers

The introduction of computers in the 1970s revolutionized industrial systems by replacing inflexible, hard-wired relays with reconfigurable Programmable Logic Controller (PLC)s (Antón et al., 2017). PLCs function by reading input data from devices such as sensors and switches, which is then processed using conditional logic to produce outputs. Output data is then sent to devices such as relays and actuators. This I/O data is commonly referred to as PLC tags. This execution cycle enables real-time evaluation of physical conditions, allowing PLCs to automate industrial machinery. Their proximity to electromechanical processing components emphasizes their criticality in ICS, making the uninterrupted operation of PLCs essential to maintaining the reliability and safety of these systems.

2.2. SCADA system architecture

SCADA systems comprise both hardware and software components designed to remotely manage and optimize industrial processes. These systems are expansive networked architectures that provide centralized control over low-level components, including PLCs, distributed across large physical areas. The scalability of SCADA systems has grown alongside industrial demands, accommodating an increasing number of devices and complex system requirements (O.V et al., 2024).

The gradual standardization of communication protocols has bridged the gaps between SCADA system layers. Compatibility and scalability have been further enhanced by the adoption of internet-based protocols, such as Modbus TCP and OPC UA (Antón et al., 2017). These protocols have facilitated seamless communication between devices and allowed SCADA-specific software to efficiently monitor and control large-scale industrial operations.

2.3. Human machine interfaces and ignition

HMIs enable the remote management, monitoring, and maintenance capabilities of SCADA systems. Traditionally, these interfaces collect data from field devices and present it to operators through graphical displays, allowing real-time interaction with system data. Operators use this data to perform system diagnostics, adjust processes, or generate reports. The standardization of communication protocols and system architectures has led to the development of widely-used SCADA HMI software (Doumanidis et al., 2023). This evolution has also enabled the incorporation of advanced data monitoring technologies, including system alarms, historical data capture and trend analysis, while reducing operational costs.

Advanced HMI software, like Inductive Automation's Ignition, introduce software Gateways. These Gateways are systems capable of collecting and processing vast amounts of SCADA system data, including running processes and PLC I/O. Multiple SCADA HMI Gateways can operate concurrently to segment complex industrial networks or serve as backups. They seamlessly exchange data and enable easy access through remote web clients.

2.4. SCADA system exploitation

While the convenience and scalability of SCADA systems have driven rapid industrial development and improvements in safety and efficiency, the interconnectivity of these systems also introduces significant vulnerabilities. Initially, SCADA and ICS components were designed to operate on heavily restricted, small-scale networks. As a result, many components were developed with inherent security flaws, which persist in modern-day systems (Ayub et al., 2023). Decreased physical separation within systems has increased their exposure to cyber threats, enabling attacks to infiltrate and propagate both vertically and laterally across layers of the ICS architecture. This risk is exacerbated by the growing standardization within SCADA environments, allowing attackers to reuse malware with minimal modification, as seen in high-profile cases like Stuxnet. While standardized software simplifies deployment, its perceived ease of use can lead to exploitable misconfigurations. Exploits aimed at these systems include unauthorized code execution, data extraction, and denial of service (DoS) attacks, all of which can significantly disrupt operations and compromise system integrity.

3. Testbed Design and Setup

In this section, we present an overview of the SCADA system testbed developed for our forensic investigation. Table 1 outlines the hardware and software components utilized in our study. The interaction and configuration of the testbed components are illustrated in Fig. 1.

3.1. Hardware setup

Modern industrial-scale SCADA systems rely on a variety of specialized hardware components. To simulate a diverse range of real-world scenarios, our testbed includes an assortment of PLCs from different manufacturers. These devices include:

- SIMATIC S7-1200 by Siemens
- Micro850 by Allen–Bradley
- ClickPLC by Automation Direct

As depicted in Fig. 1, the ④ Siemens SIMATIC S7-1200 was coupled with an ③ 8 position I/0 simulator, enabling hardware-controlled input simulation for Remote Terminal Unit (RTU) emulation. We included an Analog Input Expansion Module to provide two additional inputs to the ⑤ Allen–Bradley Micro850, diversifying the format of system data. The ⑥ Automation Direct ClickPLC operated standalone. All PLCs and

Table 1
Apparatus table depicting the hardware and software utilized throughout the experiment.

Hardware/ Software	Use	Company	Software/Model Version
Micro850 PLC	Testbed Field Control	Allen Bradley	2080-L50E-24QBB
ClickPLC	Testbed Field Control	Automation Direct	C0-12DD2E-2-D
Siemens S7-1200 PLC	Testbed Field Control	Siemens	6ES7 212-1BE40-0XB0
Ignition	SCADA HMI Platform	Inductive Automation	8.1.33
Windows Server 2022 Virtual Machine	Ignition Host	Microsoft Corporation	Windows Server 2022 21H2 build 20348-2031
Windows Server 2022 Virtual Machine	SQL Server	Microsoft Corporation	21H2 build 20348-2031
Proxmox Virtual Environment	Host VMs	Proxmox Server Solutions GmbH	PVE 5.15.102–1
Analog Input Expansion Module	Testbed Field Control	Allen Bradley	2080-IF2
Arduino Uno	Field Device Simulation	Arduino	Rev3
24VDC Optocoupling Relay	Field Device Simulation	ANMBEST	
8 position Input Simulator	Field Device Simulation	Siemens	6ES7274-1XF30-0XA0
16-port gigabit Unmanaged Switch	Testbed Network Switch	Netgear	
24VDC Power Supply	Testbed Power	MEAN WELL	NDR-120-24
CLICK Programming Software	Control Logic Programming	Automation Direct	3.43
Connected Components Workbench	Control Logic Programming	Allen Bradley	22.00.00
Simatic STEP 7 Basic	Control Logic Programming	Siemens	17
DB Browser for SQLite	View SQLite Database	DigitalOcean, LLC	3.13.99
Wireshark	View PCAP Files	The Wireshark team	4.0.1
Autopsy	View Disk Files	SLEUTH KIT LABS	4.21.0

relevant hardware components were powered by 24V DC power supplies. To supply system inputs to the Micro850 and ClickPLC, we constructed a simple hardware input simulator using an ② Arduino UNO R3 and a ① relay module with octocoupler high/low level triggers. This allowed us to supply simulated digital and analog inputs to the Micro850 and the ClickPLC in the absence of physical field devices. The Arduino was programmed to provide a timed sequence of predefined inputs to each PLC, allowing for controlled emulation of normal system operation and states.

The SCADA workstation and backend database storage systems were configured using two Windows Virtual Machine (VM)s. We hosted our VMs using Proxmox, an open-source virtualization platform that facilitates the management of virtual machines, storage, and network configurations. All relevant hardware is connected to the testbed network through ethernet and a network switch.

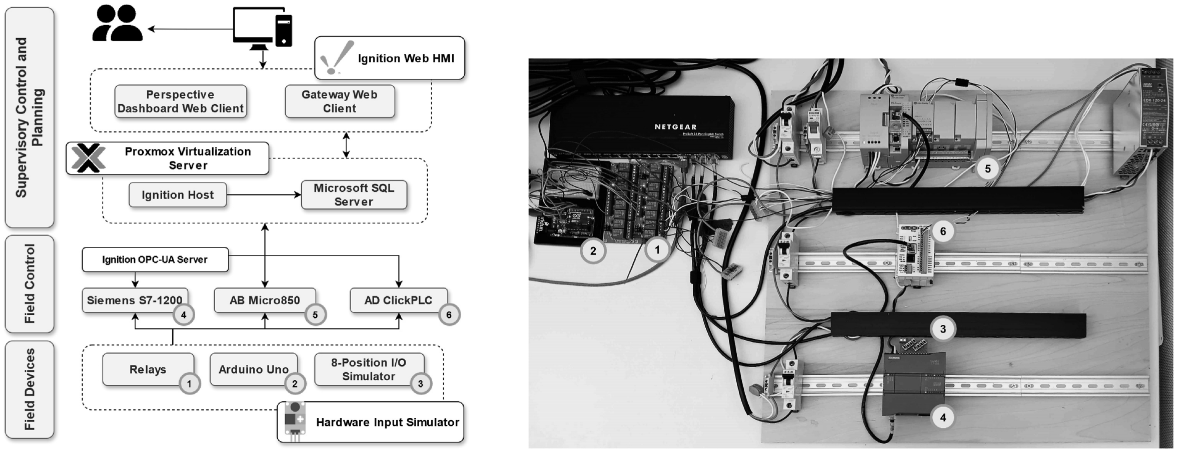
3.2. PLC control logic

SCADA systems are utilized across diverse industries, managing numerous critical processes. Reflecting this diversity, the control logic implemented in the testbed consists of simple ladder logic programs tailored to the specific hardware available to each PLC. Each program simulates small-scale industrial control operations representative of real-world applications.

The SIMATIC S7-1200 control logic emulates aspects of a bottle sanitization sequence used in food and beverage processing. The Micro850 control logic is based on mechanisms that are found in water and wastewater treatment facilities. Lastly, the control logic of the ClickPLC represents a simple traffic control system for a four-way intersection, mimicking infrastructure management processes. Combined, all three PLCs feature processes indicative of systems critical to maintaining large-scale industrial operations while generating a variety of tag data. Additionally, randomized, scripted values (within predefined margins) are written to Ignition tags to emulate dynamic data observed in real-world systems.

3.3. HMI and SCADA system infrastructure

The SCADA HMI environment in our testbed uses the Ignition platform to host the Gateway, providing interactive dashboards that display tag data and device states, accessible via a web client. The HMI also includes OPC UA and Modbus TCP servers to facilitate communication with PLCs and the SQL bridge. A separate Windows VM hosts the SQL server, which stores tag history and alarms. Tag data from PLCs is accessed through address mapping to PLC memory, with predefined



(a) The experimental SCADA Testbed configuration organized in relation to standard SCADA Architecture found in Critical Infrastructure.

(b) The SCADA Testbed hardware setup.

Fig. 1. Comparing the SCADA Testbed hardware setup to a standard SCADA Architecture.

access privileges. Outputs and holding registers are read/write accessible for control, while tag data updates are transmitted through periodic network broadcasts.

4. SCADA HMI Forensic Acquisition Framework

The SCADA HMI Forensics Acquisition Framework addresses the central HMI Gateway workstation and three distinct modes of communication: (1) Gateway-to-Gateway, (2) Gateway-to-Client and (3) Gateway-to-Field Controllers (PLCs). These communication modes represent the flow of data throughout a SCADA system, encompassing not only the information stored in PLCs from lower-layer hardware, but also the processes and communication essential for accurate and efficient system operation and monitoring. Our framework defines these modes for data attribution in a comprehensive, structured investigation. One of the most recognized models for defining the hierarchical architecture of ICS is the Purdue Enterprise Reference Architecture (PERA), introduced to systematically organize ICS data with a focus on system security (Williams, 1994). We used this model, referred to commonly as the Purdue ICS Model, as reference throughout our work. In developing our SCADA HMI Forensics Acquisition Framework, we considered not only how forensic data could be soundly collected to reveal insights into the aforementioned communication modes, but also how the data aligns with the traditional ICS Architecture. This section defines our proposed solution.

4.1. The SCADA HMI gateway

Modern web-based industrial systems are often managed through SCADA HMI Gateways. This powerful system software is offered by many vendors including Inductive Automation, Eaton, and General Electric, among others. We define SCADA HMI Gateways as:

- Workstation(s) collecting multi-component SCADA system data for process control and management
- Hosts for Gateway and HMI web servers, providing real-time data visualization and interaction
- Providing management features such as alarming and data historians, enhancing convenience and system oversight.

While multiple Gateways can exist within a SCADA system to facilitate workload distribution and redundancy, they generally function as centralized access points for SCADA control and monitoring. These systems exist on Layer 2 of the Purdue ICS Model.

4.2. Gateway-to-Gateway

SCADA HMI Gateway software is responsible for hosting and managing a wide range of system data related to various projects and operations. Gateway-to-Gateway communication consists of the data transferred within the SCADA HMI Gateway itself. This includes the storage of system configuration data, access control profiles, web server configurations for hosting project dashboards and views, and historian and alarming configurations. The information stored and exchanged through these Gateways provides critical insights into the SCADA system's architecture and functionality. They communicate key information about the devices they are collecting data from, where data is stored, and how data is being manipulated. Understanding Gateway-to-Gateway communication is essential for forensic investigations, as it reveals the underlying structure and operation of the SCADA system.

4.3. Gateway-to-Client

Remote access to ICS data is facilitated by Gateway-to-Client communication, where remote clients connect to the web servers hosted by the central Gateway workstation. These connections allow operators to manage and supervise industrial processes from remote locations. Through this communication channel, remote operators can interact with field controller data and manage system operations in real time. Any modifications made by remote clients are efficiently relayed back to the Gateway, ensuring seamless updates and synchronization across the ICS. This interaction is critical for enabling operators to monitor, control, and respond to system states, enhancing operational flexibility and oversight.

4.4. Gateway-to-field controller

The most fundamental communication in our framework is the transfer of data between field controllers and the Gateway. Field controllers represent the lower-layer intelligent hardware devices directly responsible for the manipulation of data for specific industrial processes within an ICS. These devices reside in Layer 1 of the Purdue ICS model, where PLCs operate. The Gateway initiates requests for tag data from field controllers over communication servers. These requests are transmitted directly to the field controllers, which then return the requested data to the Gateway for further processing and monitoring. This communication typically occurs over standardized protocols such as Modbus TCP or device-specific protocols like S7COMM.

Field controllers process data provided by lower-layer input hardware, such as sensors and switches, which reside in Layer 0 of the Purdue ICS model. Notably, the Gateway can access and modify input data directly in PLC memory, enabling real-time interaction with field devices. This direct access highlights the critical role of Gateway-to-Field Controller communication in ensuring reliable and accurate system operation.

4.5. The SCADA HMI Forensic Acquisition Framework application

The operation of a SCADA HMI system can be broken down into the fundamental modes of communication described in the framework: Gateway-to-Gateway, Gateway-to-Client, and Gateway-to-Field Controllers. Our testbed incorporates each of these communication modes, aligning with the framework's structure. Additionally, we referenced the Purdue ICS model to determine the architectural placement of each testbed component, as illustrated in Fig. 1a.

An analysis of the HMI forensic data we acquired and how it is reflected in our framework is described in Forensic Artifact Analysis. Fig. 2 presents an overview of the methodology we used to apply this framework to an investigation. Forensic data was collected from the testbed's Ignition Gateway host, serving as the target SCADA HMI system. To facilitate data extraction for analysis, we developed the Ignition Forensic Artifact Carving Tool (IFACT). IFACT was designed to streamline recovery of Ignition HMI forensic artifacts from network traffic, memory, and disk, offering a comprehensive view of system data and device communication in one application.

5. Forensic Acquisition

After finalizing the testbed, a comprehensive forensic investigation of the SCADA HMI system was conducted. This section outlines our methodology used for collecting forensic images and data. We also define the feasibility of our methodology in real-world SCADA HMI environments.

5.1. Network forensics

Network packet dumps of traffic coming to and from the Ignition Gateway were collected using Wireshark, enabling the capture of communication between the Gateway and the three PLCs. With direct access to network hardware, analysts could also employ port mirroring to duplicate network traffic to a monitoring port, providing an additional layer of visibility for forensic analysis without installing

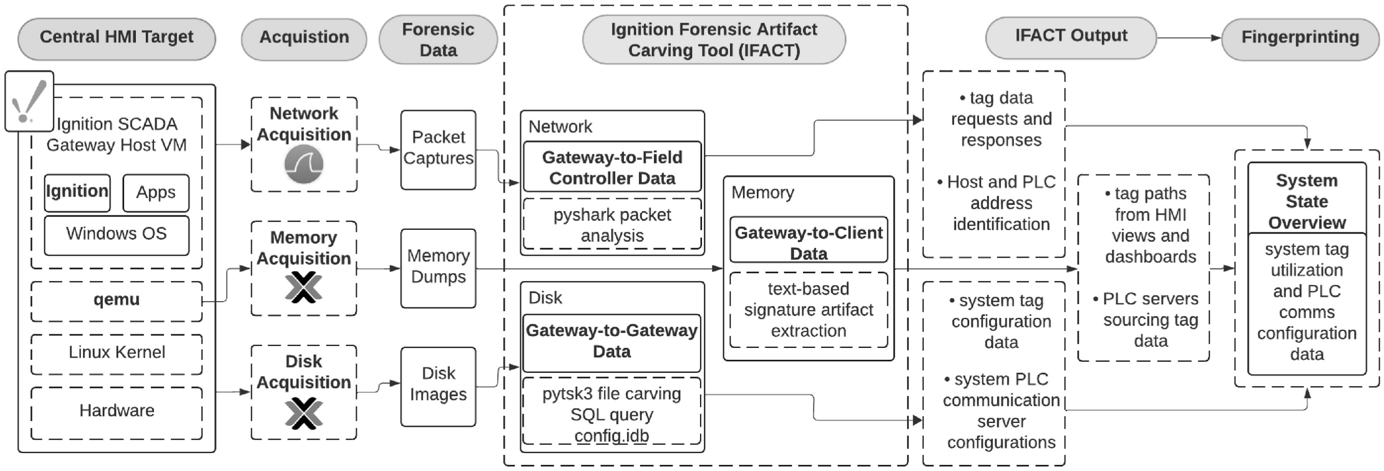

Central HMI Target
Acquistion
Forensic Data
Ignition Forensic Artifact Carving Tool (IFACT)
IFACT Output
Fingerprinting
Ignition SCADA Gateway Host VM
Ignition
Apps
Windows OS
qemu
Linux Kernel
Hardware
Network Acquisition
Memory Acquisition
Disk Acquisition
Packet Captures
Memory Dumps
Disk Images
Network
Gateway-to-Field Controller Data
pyshark packet analysis
Memory
Gateway-to-Client Data
text-based signature artifact extraction
Disk
Gateway-to-Gateway Data
pytsk3 file carving SQL query config.idb
• tag data requests and responses
• Host and PLC address identification
• system tag configuration data
• system PLC communication server configurations
• tag paths from HMI views and dashboards
• PLC servers sourcing tag data
System State Overview
system tag utilization and PLC comms configuration data

Fig. 2. The methodology for implementing the SCADA HMI Forensic Acquisition Framework in an investigative scenario on the SCADA testbed. Forensic network, memory, and disk data is collected from the testbed's Ignition host workstation. Forensic artifacts are recovered by IFACT and analyzed for SCADA system fingerprinting.

additional packet capturing software on systems.

5.2. HMI workstation disk and memory forensics

Disk and memory forensic evaluations were performed on the workstation hosting the Ignition HMI software. Disk images were created by accessing the storage disk for the Ignition host VM and copying the contents to a separate disk image file. For collecting memory dumps, we similarly used Proxmox memory dumping features through the qemu monitor.

While tools exist for forensic data collection on physical servers, the hosting of SCADA systems through virtualization is becoming increasingly common for numerous reasons. When implementing modern SCADA systems, virtualization offers a reduction in costs, increased scalability, and easier maintenance. Additionally, improving the lifespan of legacy systems is often accomplished by moving to virtual environments, rather than assuming the costs of physical upgrades for integration and scaling. Virtualization also makes the implementation of redundant systems more practical, which is particularly desirable for core HMI Gateways. This redundancy mitigates some risk in having to shut down Gateway server VMs, interrupting system operation, for disk image acquisition. With these considerations, our forensic collection methodology for both disk and memory is feasible and can be similarly deployed in modern virtualized SCADA environments with limited system impact.

5.3. SQL database forensics

In industrial control networks, SCADA HMI systems like Ignition are commonly paired with a SQL database to facilitate long-term data storage. This database is integral for maintaining historical records and operational backups. In our testbed, the SQL database was locally hosted on the Ignition Gateway workstation, but it may also be hosted remotely. During our forensic investigation, the SQL database was successfully recovered as part of the disk acquisition process. This recovery allowed us to directly access and analyze the stored data, providing insights into the system's configuration and functionality. Details of the database analysis are discussed further in the Ignition Forensic Artifact Carving Tool section.

6. Forensic Artifact Analysis

This section details the Ignition-relative artifacts that we were able to recover from each forensic data source along with our analysis of these artifacts. We describe how this data is categorized by the fundamental modes of communication described in our framework for the purpose of organizing and attributing data during an investigation.

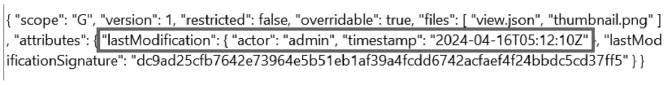
6.1. Network artifacts

Network traffic analysis provided valuable insights into Gateway-to-Field Controller communication. This traffic revealed the streams of messages between the HMI workstation and the PLCs. SCADA HMI data requests reveal information about the device being probed including its IP for communication server configuration, its manufacturer, model, and the communication protocol being utilized between them. Packet data also reveals the values of each tag as it's read from PLC memory. Other packet data reveals communication with data historians.

As with traditional system fingerprinting techniques, this network data was instrumental in identifying field devices communicating with the central HMI Gateway over the network, the frequency of transmissions, and the raw data being sent. Frequency of data transmission from the PLCs means even a relatively small timeline of network traffic can reveal the devices, tags, and tag data for a large system. With the nature of tag transmission, a disproportionate amount of responses and requests can reveal anomalous system behavior.

6.2. Disk artifacts

Disk artifacts recovered from the Ignition system primarily revealed system configuration data and resources for HMI views and dashboards. These include the DLLs and executables used to run the HMI processes. Correlations between tag data and view elements is stored. Resource files, as seen in Fig. 3, also assert the most recent modification timestamp and the actor that performed it. We were able to recover the local Ignition 'config.idb' SQL database including backups. The Ignition database stores the full tag configuration data. This includes the communication servers, data types, tag names, history providers, and alarming configurations. Also featured is the individual PLC communication server information. Additionally, Ignition HMI view projects maintain a logging database that records system errors, such as tag historian storage errors as seen in Fig. 4. These logs can serve as a crucial



```
{ "scope": "G", "version": 1, "restricted": false, "overridable": true, "files": [ "view.json", "thumbnail.png" ]
, "attributes": { "lastModification": { "actor": "admin", "timestamp": "2024-04-16T05:12:10Z" }, "lastMod
ificationSignature": "dc9ad25cfb7642e73964e5b51eb1af39a4fcdd6742acfaef4f24bbdc5cd37ff5" } }
```

Fig. 3. Disk resource.json data for Designer project files.

resource for identifying system misconfigurations or failures in data storage and retrieval processes.

Data on disk reflects mainly Gateway-to-Gateway communication, revealing to investigators anomalies in the Gateway's configuration, data propagated between Gateways, and Gateway operation failures.

6.3. Memory artifacts

Much of the Ignition system data found in memory reflects information about resources being utilized. This includes modules, DLL names and versions, PLC drivers, and Ignition functions and classes. Gateway server information including port numbers, the Gateway role, its status, and name was also recovered from memory. For the HMI web-clients, hosted resources and web-page artifacts can be recovered.

The most notable artifacts in memory reflect the Gateway-to-Client communication featured as messages being sent throughout the SCADA system. This includes alarm messaging that reveals whether alarms are currently active, the current timestamp, the relevant tag value at the time, and whether they were acknowledged or cleared, which is depicted in Fig. 5. System access and modification data alongside the performing actor and timestamps can also be recovered. Tag historian memory buffer errors can be found in memory alongside tag data read responses, revealing whether tag history data is reliably being stored and read. Additionally, tag path artifacts recovered from client web views can be recovered. As depicted in Fig. 6, memory also held the names of the PLC communication servers.

Much of this data can be cross-reference with other forensic data for assessing anomalies when they are not properly logged in the Gateway disk. Moreover, memory serves as a valuable resource for fingerprinting system devices and analyzing project tags utilized in the SCADA environment.

7. Ignition Forensic Artifact Carving Tool

To support the implementation of our framework and our investigation, we developed IFACT. This tool was designed to parse valuable artifacts from forensic images and data collected from an Ignition SCADA HMI system running on Windows. IFACT streamlines forensic artifact recovery and high-level analysis, providing insights into how PLCs and their associated tag data are communicated and utilized throughout the system without any preliminary knowledge of its structure. We designed this tool for the purpose of efficient cross-analysis of data through a single application, and to showcase the range of data that SCADA HMI forensics can provide in a preliminary investigation.

IFACT is a Windows web-based application with a Python3 back-end for processing and an HTML/JavaScript front-end for visualizing forensic artifacts. The tool interface is depicted in Fig. 9 and Fig. 10 of the Appendix: IFACT. The code for the tool is available at the time of publication and can be accessed through the following link: https://github.com/LaSeanS/IFACT.

Network captures are filtered by IFACT to produce condensed PCAP files featuring communication between the Ignition Gateway host and the PLCs in the system. Our testbed featured both Modbus TCP and S7COMM communication for reading tag data from PLC memory. Dissection and contextual analysis of protocol data is used to distinguish the addresses of devices behaving as PLC field devices and Ignition Gateways.

Disk images are carved by IFACT utilizing pytsk3, the sleuthkit Python wrapper, to recover files from the Ignition directory and extract them to a traversable local directory for the user. The file paths, sizes, and SHA256 hash signatures are logged. These files include the Ignition Gateway's 'config.idb' database file. IFACT queries this database to provide critical system configuration data to the user.

Memory images are parsed by IFACT to produce readable Ignition artifacts to the user using predefined text-based signatures, revealing a range of volatile data being actively used in the system. This includes web artifacts that are used to display HMI views and dashboards. Examples of tag path data artifacts from memory are depicted in Fig. 7.

8. Evaluation

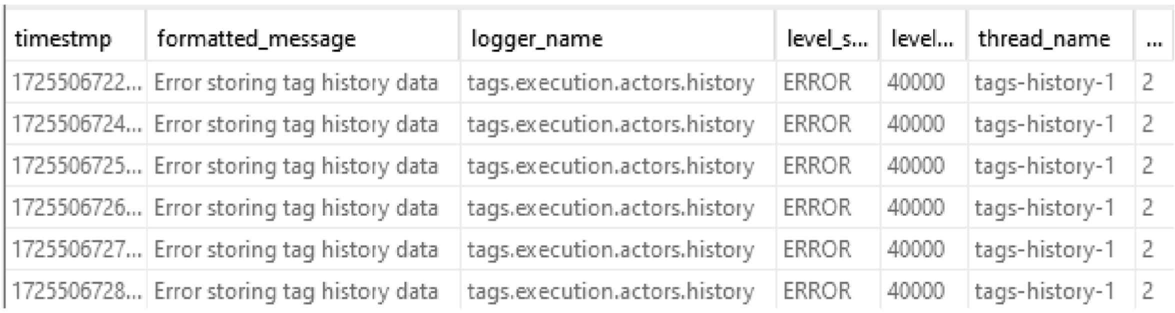
We evaluated the efficacy of HMI forensic investigations for fingerprinting SCADA systems by assessing IFACT's ability to recover PLC connections and tag data from network, disk, and memory.

8.1. Network

Network artifact recovery by IFACT focuses on the lowest layer of communication in our SCADA HMI Forensic Acquisition Framework: Gateway-to-Field Controller. Requests for PLC tag data from the Ignition Gateway are performed at a constant rate over the network. Packet captures of traffic from the network interface between the Gateway workstation and the PLCs were used in two tests. Network evaluation validated whether IFACT was capable of identifying the addresses of each unique device: the Gateway, and the three PLC devices. The results confirmed IFACT's effectiveness in recovering and analyzing network artifacts, highlighting its ability to pinpoint device addresses and reliably parse data from communication packets. The results of tests on two network packet captures is depicted in Table 2.

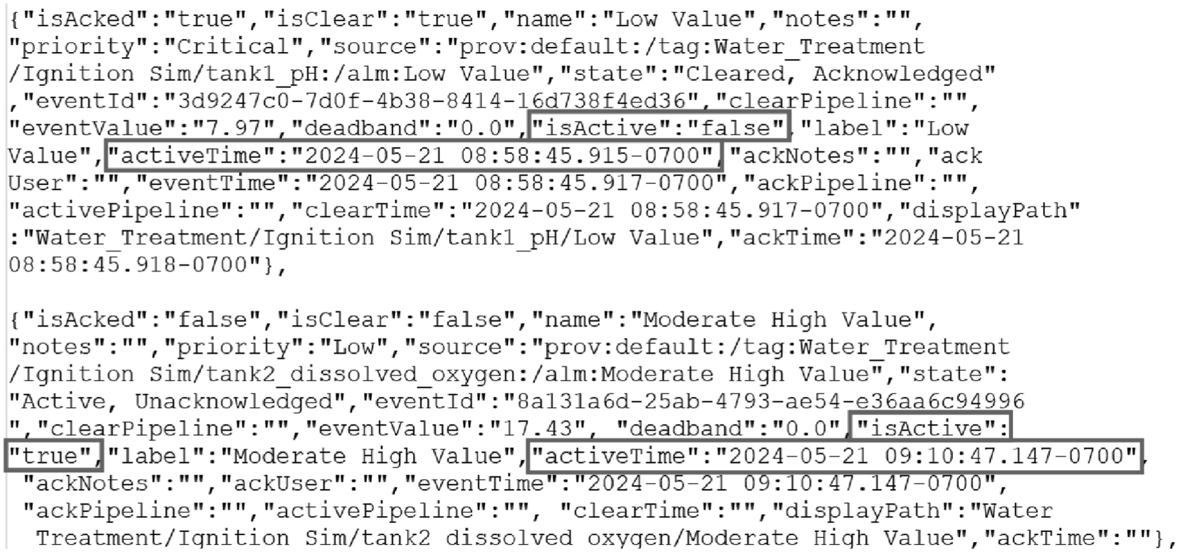
8.2. Disk

Evaluation of IFACT's performance in disk artifact recovery considered the PLC device communication server and tag configuration data that is stored in the system. Our controlled testbed has both PLC sourced tag and global scripting tag data that is stored for use in Ignition projects as atomic tags. In a SCADA system, this information would be propagated as Gateway-to-Gateway data. We tested IFACT's ability to parse this information from a raw disk image recovered from our testbed. IFACT's performance was compared to the baseline data stored in the Gateway for the testbed designer project. Table 3 presents IFACT's



timestmp	formatted_message	logger_name	level_s...	level...	thread_name	...
1725506722...	Error storing tag history data	tags.execution.actors.history	ERROR	40000	tags-history-1	2
1725506724...	Error storing tag history data	tags.execution.actors.history	ERROR	40000	tags-history-1	2
1725506725...	Error storing tag history data	tags.execution.actors.history	ERROR	40000	tags-history-1	2
1725506726...	Error storing tag history data	tags.execution.actors.history	ERROR	40000	tags-history-1	2
1725506727...	Error storing tag history data	tags.execution.actors.history	ERROR	40000	tags-history-1	2
1725506728...	Error storing tag history data	tags.execution.actors.history	ERROR	40000	tags-history-1	2

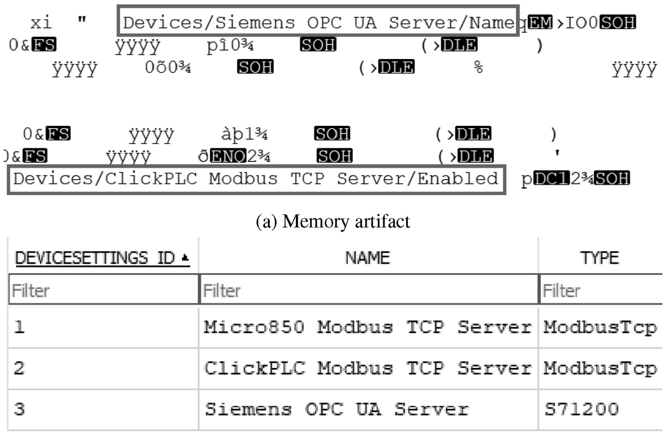
Fig. 4. Ignition database system_logs.idb for system logging found in disk.



```
{"isAcked":"true","isClear":"true","name":"Low Value","notes":"",
"priority":"Critical","source":"prov:default:/tag:Water_Treatment
/Ignition Sim/tank1_pH:/alm:Low Value","state":"Cleared, Acknowledged"
,"eventId":"3d9247c0-7d0f-4b38-8414-16d738f4ed36","clearPipeline":"",
"eventValue":"7.97","deadband":"0.0","isActive":"false","label":"Low
Value","activeTime":"2024-05-21 08:58:45.915-0700","ackNotes":"","ack
User":"","eventTime":"2024-05-21 08:58:45.917-0700","ackPipeline":"",
"activePipeline":"","clearTime":"2024-05-21 08:58:45.917-0700","displayPath"
:"Water_Treatment/Ignition Sim/tank1_pH/Low Value","ackTime":"2024-05-21
08:58:45.918-0700"},

{"isAcked":"false","isClear":"false","name":"Moderate High Value",
"notes":"","priority":"Low","source":"prov:default:/tag:Water_Treatment
/Ignition Sim/tank2_dissolved_oxygen:/alm:Moderate High Value","state":
"Active, Unacknowledged","eventId":"8a131a6d-25ab-4793-ae54-e36aa6c94996
","clearPipeline":"","eventValue":"17.43", "deadband":"0.0","isActive":
"true","label":"Moderate High Value","activeTime":"2024-05-21 09:10:47.147-0700",
 "ackNotes":"","ackUser":"","eventTime":"2024-05-21 09:10:47.147-0700",
 "ackPipeline":"","activePipeline":"", "clearTime":"","displayPath":"Water
  Treatment/Ignition Sim/tank2 dissolved oxygen/Moderate High Value","ackTime":""},
```

Fig. 5. Memory artifacts depicting active and inactive alarms.



```
  xi  "   Devices/Siemens OPC UA Server/NameqEM›IO0SOH
0&FS     ÿÿÿÿ    pî0¾     SOH        (›DLE     )
    ÿÿÿÿ    0ŏ0¾     SOH        (›DLE     %          ÿÿÿÿ


  0&FS      ÿÿÿÿ     àþ1¾     SOH        (›DLE      )
)&FS      ÿÿÿÿ     ðENQ2¾    SOH        (›DLE      '
Devices/ClickPLC Modbus TCP Server/Enabled  pDC12¾SOH
```

(a) Memory artifact

DEVICESETTINGS_ID ▴	NAME	TYPE
Filter	Filter	Filter
1	Micro850 Modbus TCP Server	ModbusTcp
2	ClickPLC Modbus TCP Server	ModbusTcp
3	Siemens OPC UA Server	S71200

(b) Ignition SQL database entries for PLC connection server settings

Fig. 6. Memory vs Disk artifacts revealing PLC connections active in the SCADA system testbed.

ability to recover all atomic tag and PLC data.

8.3. Memory

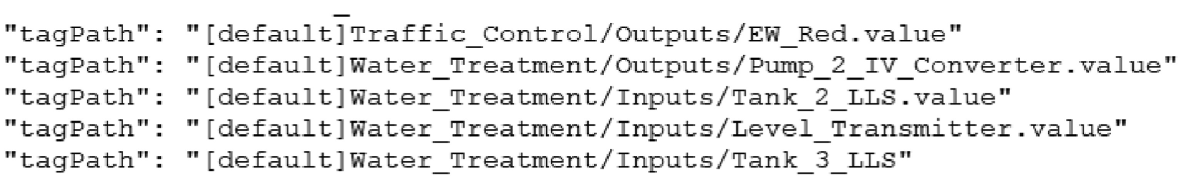
While evaluating memory, we considered that the most notable memory artifacts consisted of data being used for HMI web views and dashboards in Gateway-to-Client communication. Our evaluation focused on IFACT's ability to recover tags being actively utilized in Ignition projects to present HMI views and the lifetime of this data. For initial validation, we created 'test memory dumps' with controlled data to serve as our baseline. To create these, memory artifacts were extracted from actual SCADA testbed memory dumps to create controlled memory dumps for evaluation. The results of this experiment are captured in Table 4. IFACT was able to identify each of the three PLC connections in both tests. For recovering tag data, we first measured the total number of tag path artifacts each test was able to recover. Then, we measured the total number of unique tag paths that were contained in

Table 2
IFACT network data recovery evaluation test measuring the ability of IFACT to identify the gateway and PLC addresses alongside each tag data request and response between the Gateway and PLCs. A checkmark denotes IFACT successfully recovering and identifying the address attributed to the corresponding device.

PCAP 1	Gateway	Micro850	ClickPLC	S7-1200
Baseline	✓	✓	✓	✓
IFACT Output	✓	✓	✓	✓
	Tag Requests	**Tag Responses**		
Baseline	8338	8335		
IFACT Output	8338	8335		
PCAP 2	**Gateway**	**Micro850**	**ClickPLC**	**S7-1200**
Baseline	✓	✓	✓	✓
IFACT Output	✓	✓	✓	✓
	Tag Requests	**Tag Responses**		
Baseline	2709	2709		
IFACT Output	2709	2709		

Table 3
The communication server and Atomic tag data recovered during IFACT disk evaluation testing. A checkmark denotes IFACT successfully recovering server data attributed to the corresponding device.

	Micro850	ClickPLC	Siemens S7-1200	Atomic Tags
IFACT	✓	✓	✓	42
Ignition	✓	✓	✓	42



```
"tagPath": "[default]Traffic_Control/Outputs/EW_Red.value"
"tagPath": "[default]Water_Treatment/Outputs/Pump_2_IV_Converter.value"
"tagPath": "[default]Water_Treatment/Inputs/Tank_2_LLS.value"
"tagPath": "[default]Water_Treatment/Inputs/Level_Transmitter.value"
"tagPath": "[default]Water_Treatment/Inputs/Tank_3_LLS"
```

Fig. 7. PLC tag paths as they appear carved from views in memory.

Table 4
The PLC device communication and tag data recovered during IFACT memory evaluation testing.

Test 1	PLCs	Total Tag Paths	Unique Tag Paths
Baseline	3	695	26
IFACT Output	3	695	26
Test 2	**PLCs**	**Total Tag Paths**	**Unique Tag Paths**
Baseline	3	715	26
IFACT Output	3	715	26

these artifacts.

We evaluated the lifetime of data in memory through two scenarios to assess their impact on SCADA HMI memory forensics. In Scenario 1, 3 min after initial memory acquisition, all HMI client views and dashboards were closed. In Scenario 2, the Ignition Gateway application was also terminated. Memory dumps were collected every 2 min over a 12-min period. The results, shown in Fig. 8, reveal that while tag data continued to populate memory in scenario 1, scenario 2 saw an immediate degradation in artifact recovery. Alongside this decline in tag data recovery, one of the testbed PLCs could no longer be identified. These findings suggest that as long as the Ignition application remains operational, web views and associated tag data can be reliably recovered for forensic analysis. However, once the Ignition process is stopped, even on specialized systems, the availability of memory artifacts declines sharply, significantly impacting forensic investigations.

9. Discussion

In the following section, we address the research questions presented in the Introduction. Here, the purpose of our research and tool along with the overall value of SCADA HMI forensic analysis will be articulated.

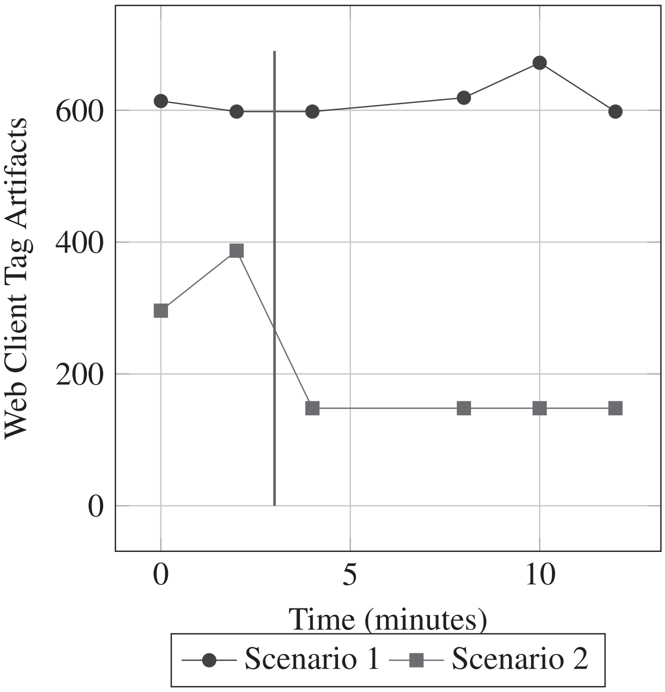


Fig. 8. Comparison of the lifetime of Ignition web client data in memory through two testing scenarios. Between the second and third memory dumps (3 min mark), two scenarios were executed. Scenario 1: all web client views and dashboards are closed. Scenario 2: all web client views and dashboards are closed and the Ignition Gateway application is terminated.

9.1. RQ1: developing a SCADA HMI forensics framework

Can a generic framework be developed for the collection of forensic evidence from a centralized SCADA HMI system? The SCADA HMI Forensic Acquisition Framework defines the nature of data communicated within web-based SCADA HMI systems. Our acquisition process reflects a methodology for collecting forensic data that can provide insight to investigators on each fundamental mode of communication. With the expansive nature of these systems, our framework guides investigators in structured analysis of central SCADA HMI workstation data for system fingerprinting and revealing anomalous or inconsistent data throughout the system. Utilizing our framework provides contextual insight to system data where inconsistencies could indicate if and how system tampering occurred.

9.2. RQ2: system data extracted from HMI forensic artifacts

To what extent can system data be extracted from artifacts collected during a SCADA HMI forensics investigation? Our findings indicate that HMI forensic analysis provides a comprehensive view of the SCADA system's state, offering investigators a solid frame of reference. This analysis can be leveraged to gather a wide range of information, enabling insight into the core operations of vastly distributed SCADA systems. Through our framework, we are also able to assert that HMI forensics can be leveraged to analyze data across multiple layers defined in the Purdue ICS Model.

We propose that with fingerprinting performed through the central HMI of a SCADA system, such as Ignition, you can concisely profile the system including the amount and manufacturers of remotely operating devices, system configuration, active users, and the operation of devices reflected through tags, alarming, and HMI views. This fingerprinting capability is especially valuable when considering how vastly distributed modern SCADA systems are. Maintaining an accurate profile of a system is essential for maintenance, security monitoring, and attack attribution analysis. When HMI analysis is leveraged, it can assist investigators in achieving an accurate image of the devices in a system and how they interact as a whole down to the individual process level.

9.3. RQ3: SCADA architecture forensics and anomalies

Are forensic artifacts collected from a centralized SCADA HMI environment sufficient to gather system-wide data and identify anomalous behavior across multiple layers of the SCADA architecture? SCADA HMI monitoring and alarming are vital for the continuous operation of a SCADA system. However, blindly trusting and relying on the output of static ruling systems can be as harmful as disregarding their use entirely. Proper centralized HMI forensics, even standalone, can reveal a variety of artifacts indicating system fallout after an attack. Forensic artifacts can be used to identify how system data is affected, and how dynamic cross-validation can be used to develop real-time mitigation applications. For example, identifying network artifacts of SCADA HMI Gateway-to-Client or Gateway-to-Gateway communication could reveal evidence of MitM attacks targeting Gateways through the websocket: a viable attack strategy that has been explored through investigation of zero-days in the past. Cross-analysis of network and memory view data could reveal evidence of attacks through system inconsistencies even if the data may otherwise circumvent static alarming.

Overall, SCADA HMI Forensics can reveal anomalous data resulting from numerous attack types and vectors and serves as a good reference point for investigators identifying attacks in the face of system failure. Contextual insight using our framework could be used to dynamically validate a system's operation. Dynamic detection strategies are becoming increasingly valuable in ensuring critical infrastructure maintains sound operation in the face of sophisticated zero-days, and forensic investigations on SCADA HMI systems can reveal how tooling operating alongside SCADA HMI Gateway software can be developed for modern system resilience.

9.4. RQ4: the efficacy of SCADA HMI memory forensics

What is the lifetime of Ignition system data in memory, and what implications does this have for forensic investigations? Through our experiments, we determined that whether or not SCADA HMI Gateway applications are running at the time of memory acquisition can have drastic effects on the fingerprinting capabilities of forensic analysis and overall artifact recovery. Even on a specialized Gateway host with few processes running, we found that data loss occurs almost immediately. In the presence of a cyber attack, we suggest that investigators acquire memory dumps prior to terminating the Gateway process whenever possible.

10. Related Work

In this section, we review literature on PLC and HMI Forensics followed by SCADA System Forensics and Anomaly Detection.

10.1. PLC forensics

As the lowest layer of control, much of the research in the field of SCADA and ICS forensics focuses on the creation of tools and methodologies for the specialized hardware of PLCs. Work has been conducted to explore how network and device level PLC forensics can be used to recover information about attacks executed on ICS environments. They also describe the notable challenges for device-level PLC forensics including restricted access to hardware, the obstacle of proprietary firmware, and insufficient logging capabilities native to PLCs (Ahmed et al., 2017). Since many attacks on PLCs aim to disrupt or rewrite control logic programs or device firmware, research explores detection of these changes. Some researchers utilized pre-existing PLC debugging tools to acquire and analyze control logic from devices (Wu and Nurse, 2015). Network artifact analysis has also been used to accomplish this task through the extraction of process operations from PLCs to detect anomalies in their processes in real-time (Hadžiosmanović et al., 2014).

Other works explore how traditional network forensic analysis techniques can be utilized to conduct remote PLC memory acquisition, circumventing the need for risky direct access to device hardware. Research emphasizes how access to industrial networks can be leveraged for both investigations and attacks (Denton et al., 2017; Zubair et al., 2022). This technique has also been used to detect control logic changes by passing memory variables through defined rules for specific programs (Yau, 2015). The forensic analysis of artifacts collected from PLC memory is incredibly device specific, but researchers have established methodologies and frameworks for the analysis process as well as identification of the nature of information you can collect from this memory. There have also been tools created to automate the extraction and analysis of forensically relevant data. This is valuable due to the tedious nature of manual memory analysis (Awad et al., 2023; Rais et al., 2022).

10.2. HMI forensics

Related research supports the importance of considering control systems as a valuable source of data for SCADA environment incident response and forensic investigations. Works evaluating modern forensic techniques describe how investigators can apply knowledge of forensic analysis of regular IT systems to SCADA control systems. While existing tools can be leveraged, they are still lacking for efficiently collecting data for these systems. Additionally, a gap exists between the knowledge of forensic investigators and the complexities of SCADA control systems, inciting a need for methodologies to evolve proportionally (van der Knijff, 2014).

Researchers have created collections of existing tools for forensic analysis of a SCADA system during attack diagnosis. They point to some notable artifacts that should be considered during analysis of HMI workstations. However, these works generally lack any tooling and methodology to help speed up analysis of large SCADA systems (Mason and Zhou, 2021; Eden et al., 2016). Considering the feasibility of collecting forensic data from process control systems, research has been done to explore remote forensic analysis. This work details the effectiveness of remote acquisition techniques and the impacts imaging has on the system (Cassidy et al., 2008).

Overall, most work in this area is limited in depth of defining the nature of data that can be recovered in forensic analysis of SCADA systems with a focus on the HMI. Critically, they don't provide sufficient evidence of how it can jumpstart a forensic investigation by helping quickly perform system fingerprinting or effects of an attack on a system. There is little existing tooling for making the analysis process more efficient and effective.

10.3. SCADA System Forensics and Anomaly Detection

A number of surveys have been conducted exploring the field of ICS and SCADA forensics tooling and methodologies and how it has evolved over the years. Research supports the importance of capturing both online and offline analysis of forensic data to diagnose SCADA systems. Notably, collecting the appropriate data to identify tampering between central control systems and PLCs is valuable. As securing hardware is a significant challenge in large, distributed industrial systems, practical and efficient solutions for conducting forensic investigations and diagnosing systems is a necessity. Researchers note that many generalized digital forensics frameworks for SCADA systems lack practical evaluation and tooling to support them (Awad et al., 2018).

A comprehensive analysis of the state of digital forensics concerning ICS was conducted by (Cook et al., 2023). Researchers drew comparisons between IIoT and Internet of Things (IoT) forensics, most notable being the nature of specialized devices and propriety technology and software. At the time of their research, they noted how the introduction of cloud-based computing in ICS may be an area of concern for IIoT in the future. This projection and the lack of research in the area is a reality today. The difficulty of analyzing forensic artifacts along with a general lack of guidance and standards in the field motivates the development of more broadly applicable, open-source tools and techniques for conducting forensic investigations (Eden et al., 2017).

To establish forensic readiness for complex SCADA systems, it is crucial to retain knowledge of all devices and the large volumes of data circulating on the industrial network, a task made harder as these systems evolve. Monitoring network data between PLCs and workstations is essential for this purpose (Eden et al., 2016). Recent work has also explored the link between ICS anomaly detection and SCADA forensics (Cook et al., 2023). Collecting and analyzing I/O data aids in building anomaly detection models, which is critical for identifying insider threats and covert attacks that bypass access controls (McParland et al., 2014). Behavior-based anomaly detection models also rely on broad system data to detect anomalies, indicating the importance of developing novel anomaly identification techniques.

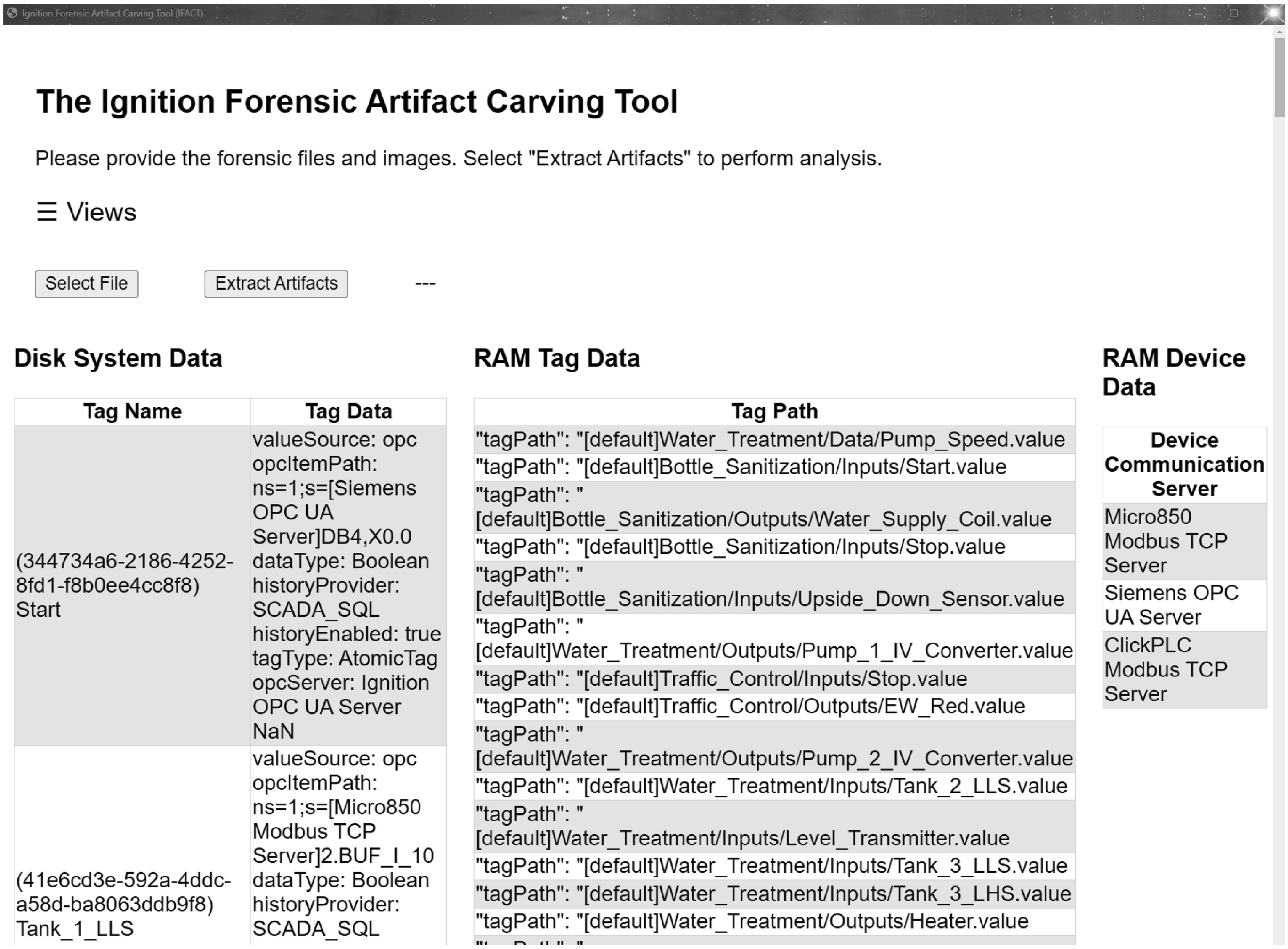
11. Conclusion

Our research has explored the practicality of HMI-focused forensic investigations in providing insight to SCADA system investigators for reference on multiple layers of the Purdue model for evaluating ICS security. We also categorized how data collected in an HMI forensics investigations can help fingerprint systems and identify anomalies and inconsistencies in SCADA systems. The SCADA HMI Forensic Acquisition Framework was created to assist investigators in determining the nature of data that can be collected through a thorough HMI forensic investigation and contextualizing it. Our framework centered around the SCADA HMI Gateway, a popular component used in modern web-based SCADA systems. We also categorize the fundamental forms of communication that can be investigated: Gateway-to-Gateway, Gateway-to-Client, and Gateway-to-Field Controller. This framework was supported by the evaluation of an Ignition SCADA testbed based on real-world SCADA environments and an accompanying analysis tool, IFACT. We also provide insight into how investigators can obtain optimal results from memory forensic investigations on SCADA systems.

12. Future Work

Our analysis of the HMI environment in SCADA systems and the creation of a general SCADA HMI Forensic Acquisition Framework provides a baseline for incident response investigations. Future work will explore dynamic data evaluation for anomaly-based threat detection in SCADA systems. We hypothesize that tooling can be developed to integrate with HMI software for advanced security monitoring through cross-analysis of workstation data. Lastly, IFACT can be expanded to support a wider variety of PLC and Gateway configurations for further research and investigations.

Appendix. IFACT



The Ignition Forensic Artifact Carving Tool

Please provide the forensic files and images. Select "Extract Artifacts" to perform analysis.

Disk System Data

Tag Name	Tag Data
(344734a6-2186-4252-8fd1-f8b0ee4cc8f8) Start	valueSource: opc opcItemPath: ns=1;s=[Siemens OPC UA Server]DB4,X0.0 dataType: Boolean historyProvider: SCADA_SQL historyEnabled: true tagType: AtomicTag opcServer: Ignition OPC UA Server NaN
(41e6cd3e-592a-4ddc-a58d-ba8063ddb9f8) Tank_1_LLS	valueSource: opc opcItemPath: ns=1;s=[Micro850 Modbus TCP Server]2.BUF_I_10 dataType: Boolean historyProvider: SCADA_SQL

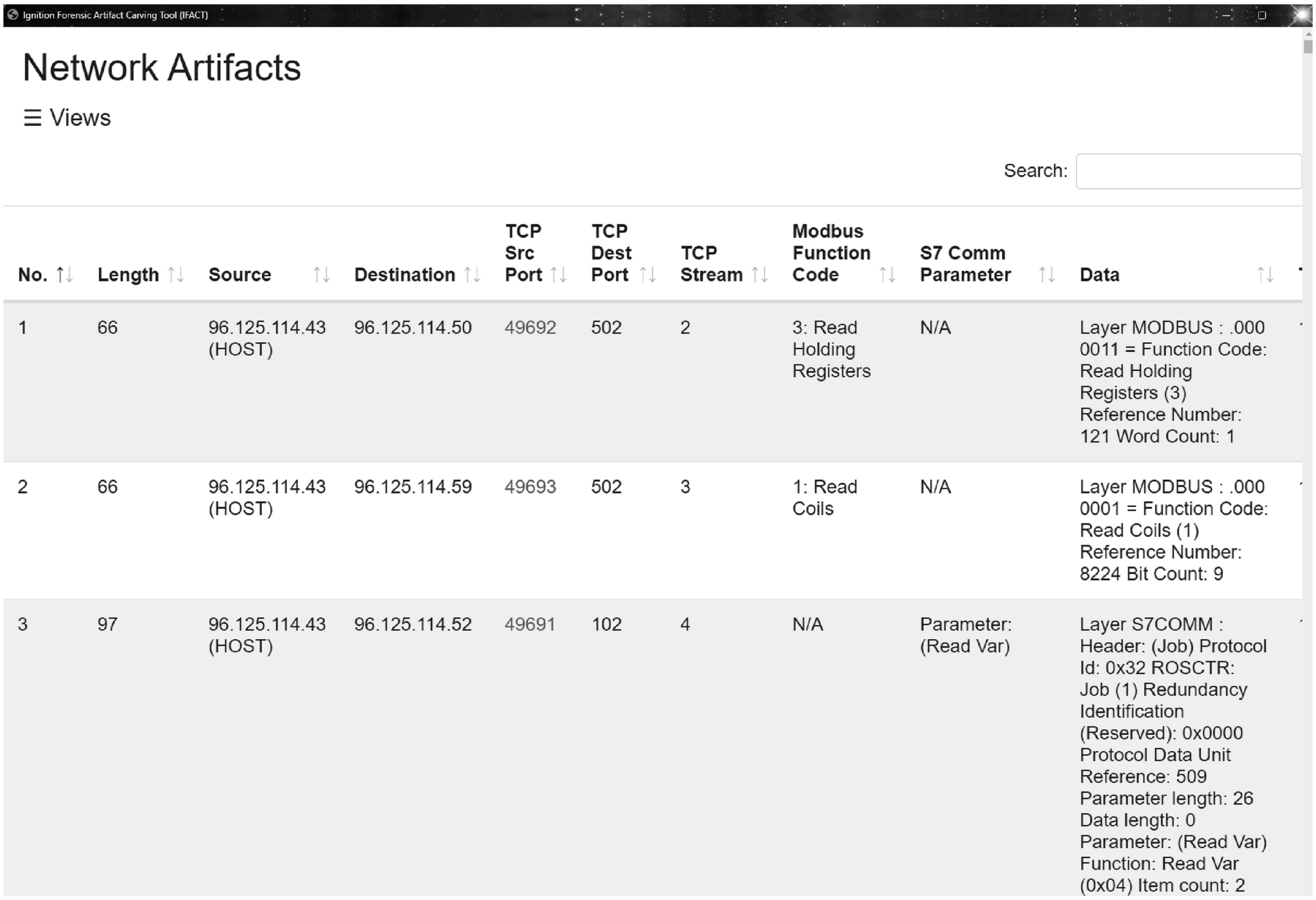
RAM Tag Data

Tag Path
"tagPath": "[default]Water_Treatment/Data/Pump_Speed.value
"tagPath": "[default]Bottle_Sanitization/Inputs/Start.value
"tagPath": "[default]Bottle_Sanitization/Outputs/Water_Supply_Coil.value
"tagPath": "[default]Bottle_Sanitization/Inputs/Stop.value
"tagPath": "[default]Bottle_Sanitization/Inputs/Upside_Down_Sensor.value
"tagPath": "[default]Water_Treatment/Outputs/Pump_1_IV_Converter.value
"tagPath": "[default]Traffic_Control/Inputs/Stop.value
"tagPath": "[default]Traffic_Control/Outputs/EW_Red.value
"tagPath": "[default]Water_Treatment/Outputs/Pump_2_IV_Converter.value
"tagPath": "[default]Water_Treatment/Inputs/Tank_2_LLS.value
"tagPath": "[default]Water_Treatment/Inputs/Level_Transmitter.value
"tagPath": "[default]Water_Treatment/Inputs/Tank_3_LLS.value
"tagPath": "[default]Water_Treatment/Inputs/Tank_3_LHS.value
"tagPath": "[default]Water_Treatment/Outputs/Heater.value
"tagPath": "

RAM Device Data

Device Communication Server
Micro850 Modbus TCP Server
Siemens OPC UA Server
ClickPLC Modbus TCP Server

Fig. 9. IFACT front page interface after analysis



Network Artifacts

☰ Views

Search:

No.	Length	Source	Destination	TCP Src Port	TCP Dest Port	TCP Stream	Modbus Function Code	S7 Comm Parameter	Data
1	66	96.125.114.43 (HOST)	96.125.114.50	49692	502	2	3: Read Holding Registers	N/A	Layer MODBUS : .000 0011 = Function Code: Read Holding Registers (3) Reference Number: 121 Word Count: 1
2	66	96.125.114.43 (HOST)	96.125.114.59	49693	502	3	1: Read Coils	N/A	Layer MODBUS : .000 0001 = Function Code: Read Coils (1) Reference Number: 8224 Bit Count: 9
3	97	96.125.114.43 (HOST)	96.125.114.52	49691	102	4	N/A	Parameter: (Read Var)	Layer S7COMM : Header: (Job) Protocol Id: 0x32 ROSCTR: Job (1) Redundancy Identification (Reserved): 0x0000 Protocol Data Unit Reference: 509 Parameter length: 26 Data length: 0 Parameter: (Read Var) Function: Read Var (0x04) Item count: 2

Fig. 10. IFACT network artifact view

References

Ahmed, I., Obermeier, S., Sudhakaran, S., Roussev, V., 2017. Programmable logic controller forensics. IEEE Security Privacy 15, 18–24. https://doi.org/10.1109/MSP.2017.4251102.

Antón, S.D., Fraunholz, D., Lipps, C., Pohl, F., Zimmermann, M., Schotten, H.D., 2017. Two decades of scada exploitation: a brief history. In: 2017 IEEE Conference on Application, Information and Network Security (AINS), pp. 98–104. https://doi.org/10.1109/AINS.2017.8270432.

Awad, R.A., Beztchi, S., Smith, J.M., Lyles, B., Prowell, S., 2018. Tools, techniques, and methodologies: a survey of digital forensics for scada systems. In: Proceedings of the 4th Annual Industrial Control System Security Workshop. Association for Computing Machinery, New York, NY, USA, pp. 1–8. https://doi.org/10.1145/3295453.3295454.

Awad, R.A., Rais, M.H., Rogers, M., Ahmed, I., Paquit, V., 2023. Towards generic memory forensic framework for programmable logic controllers. Forensic Sci. Int.: Digit. Invest. 44, 301513. https://doi.org/10.1016/j.fsidi.2023.301513. https://www.sciencedirect.com/science/article/pii/S2666281723000148. selected papers of the Tenth Annual DFRWS EU Conference.

Ayub, A., Jo, W., Qasim, S.A., Ahmed, I., 2023. How are industrial control systems insecure by design? a deeper insight into real-world programmable logic controllers. IEEE Security Privacy 21, 10–19. https://doi.org/10.1109/MSEC.2023.3271273.

Cassidy, R.F., Chavez, A., Trent, J., Urrea, J., 2008. Remote forensic analysis of process control systems. Critical Infrastructure Protection 1, 223–235. Springer.

Cook, M., Marnerides, A., Johnson, C., Pezaros, D., 2023. A survey on industrial control system digital forensics: challenges, advances and future directions. IEEE Communications Surveys Tutorials 25, 1705–1747. https://doi.org/10.1109/COMST.2023.3264680.

Denton, G., Karpisek, F., Breitinger, F., Baggili, I., 2017. Leveraging the srtp protocol for over-the-network memory acquisition of a ge fanuc series 90-30. Digit. Invest. 22, S26–S38. https://doi.org/10.1016/j.diin.2017.06.005. https://www.sciencedirect.com/science/article/pii/S1742287617301925.

Doumanidis, C., Xie, Y., Rajput, P.H., Pickren, R., Sahin, B., Zonouz, S., Maniatakos, M., 2023. Dissecting the industrial control systems software supply chain. IEEE Security Privacy 21, 39–50. https://doi.org/10.1109/MSEC.2023.3266775.

Eden, P., Blyth, A., Burnap, P., Cherdantseva, Y., Jones, K., Soulsby, H., Stoddart, K., 2016. Forensic readiness for scada/ics incident response. In: 4th International Symposium for ICS SCADA Cyber Security Research 2016 (ICS-CSR), Science Open.

Eden, P., Blyth, A., Jones, K., Soulsby, H., Burnap, P., Cherdantseva, Y., Stoddart, K., 2017. SCADA System Forensic Analysis within IIoT. Springer International Publishing, Cham, pp. 73–101. https://doi.org/10.1007/978-3-319-50660-9_4.

Hadžiosmanović, D., Sommer, R., Zambon, E., Hartel, P.H., 2014. Through the Eye of the Plc: Semantic Security Monitoring for Industrial Processes. Association for Computing Machinery, New York, NY, USA, pp. 126–135. https://doi.org/10.1145/2664243.2664277.

Javed Butt, U., Abbod, M., Lors, A., Jahankhani, H., Jamal, A., Kumar, A., 2019. Ransomware threat and its impact on scada. In: 2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3), pp. 205–212. https://doi.org/10.1109/ICGS3.2019.8688327.

Mason, T., Zhou, B., 2021. Digital forensics process of an attack vector in ics environment. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 2532–2541. https://doi.org/10.1109/BigData52589.2021.9671986.

McParland, C., Peisert, S., Scaglione, A., 2014. Monitoring security of networked control systems: it's the physics. IEEE Security Privacy 12, 32–39. https://doi.org/10.1109/MSP.2014.122.

O.V, G.S., Karthikeyan, A., Karthikeyan, K., Sanjeevikumar, P., Karapparambil Thomas, S., Babu, A., 2024. Critical review of scada and plc in smart buildings and energy sector. Energy Rep. 12, 1518–1530. https://doi.org/10.1016/j.egyr.2024.07.041. https://www.sciencedirect.com/science/article/pii/S2352484724004670.

Rais, M.H., Awad, R.A., Lopez, J., Ahmed, I., 2022. Memory forensic analysis of a programmable logic controller in industrial control systems. Forensic Sci. Int.: Digit. Invest. 40, 301339. https://doi.org/10.1016/j.fsidi.2022.301339. https://www.sciencedirect.com/science/article/pii/S2666281722000087. selected Papers of the Ninth Annual DFRWS Europe Conference.

van der Knijff, R., 2014. Control systems/scada forensics, what's the difference? Digit. Invest. 11, 160–174. https://doi.org/10.1016/j.diin.2014.06.007. https://www.sciencedirect.com/science/article/pii/S1742287614000814. special Issue: Embedded Forensics.

Williams, T.J., 1994. The purdue enterprise reference architecture. Comput. Ind. 24, 141–158. https://doi.org/10.1016/0166-3615(94)90017-5. https://www.sciencedirect.com/science/article/pii/0166361594900175.

Wu, T., Nurse, J.R., 2015. Exploring the use of plc debugging tools for digital forensic investigations on scada systems. Journal of Digital Forensics, Security and Law 10, 79–96. https://doi.org/10.15394/jdfsl.2015.1213. URL: https://commons.erau.edu/jdfsl/vol10/iss4/7.

Yau, K., 2015. Plc forensics based on control program logic change detection. Journal of Digital Forensics, Security and Law. https://doi.org/10.15394/jdfsl.2015.1211.

Zubair, N., Ayub, A., Yoo, H., Ahmed, I., 2022. Pem: remote forensic acquisition of plc memory in industrial control systems. Forensic Sci. Int.: Digit. Invest. 40, 301336. https://doi.org/10.1016/j.fsidi.2022.301336. https://www.sciencedirect.com/science/article/pii/S2666281722000051. selected Papers of the Ninth Annual DFRWS Europe Conference.