University of Glasgow | School of Computing Science

Emulation of Real-World Cyber attacks on ICS infrastructure for Provenance Dataset Generation

Mustafa Onur Cay

Abstract

As Industry 4.0 becomes commonplace the attack surface for OT and ICS infrastructure increases. Legacy OT systems become more exposed through the use of internet connected devices. While this makes the systems much easier to manage and operate it also allows state sponsored attackers and organised groups to exploit them, as they present a valuable target to the missions of other nations and play a big role in a given country's political dynamics. Securing these devices presents a unique challenge because they are often old and filled with vulnerabilities. An approach is the use of system monitoring and provenance logging on top of existing security best practices. Provenance logging allows defenders to see a complete view of the network and monitor multiple different data points, which can be used to understand how attacks originate, propagate and impact critical infrastructure, giving defenders time to react before availability is lost. In this work we build a Purdue-segmented ICS testbed with physical Siemens PLCs controlling a scaled hydroelectric process and emulate a two-day APT campaign against it using MITRE CALDERA. Telemetry is collected at every Purdue layer alongside CALDERA operation reports as attack ground truth, and we demonstrate that the captured data supports cross-layer provenance graph construction and attack chain recovery.

1 Introduction

Industrial Control Systems (ICS) present unique cybersecurity challenges due to their role in critical infrastructure and their exposure to Advanced Persistent Threats (APTs)[23]. Unlike conventional cyberattacks, APTs targeting ICS often employ multi-stage campaigns that traverse IT networks, Operational Technology (OT) systems, and Programmable Logic Controllers (PLCs). A prime example of this is Stuxnet, where the attack starts at a workstation and ultimately affects PLCs[12]. These attacks can remain undetected for extended periods, as individual actions appear benign when examined in isolation[19].

Provenance-based Intrusion Detection Systems (PIDS) offer a promising solution by tracking causal relationships between system entities across attack stages, allowing us to reduce the number of false alarms compared to traditional IDS by using context which comes from the causal relationships[36]. However, research in this area faces a critical obstacle: the lack of comprehensive datasets that document realistic attack scenarios spanning IT-OT-PLC boundaries.

This research addresses this gap by developing a methodology to generate and collect high-fidelity provenance-capable data from emulated APT scenarios. We design and implement a hybrid testbed spanning Purdue Model levels L0 through L4, using physical Siemens PLCs controlling a hydro water plant process alongside virtualised IT and OT infrastructure. Attack scenarios are derived from the analysis of nine major ICS malware families and executed using MITRE CALDERA as the adversary emulation framework. The collected data captures attack progression across IT->OT->PLC boundaries and is validated by demonstrating that attack chains can be reconstructed through backward tracing.

The contributions of this work are:

- A systematic analysis of 9 major ICS attacks, extracting 69 unique MITRE ATT&CK for ICS techniques and categorising them by attack lifecycle stage.
- The design and implementation of a hybrid ICS testbed spanning L0-L4 of the Purdue Model with physical PLCs and comprehensive provenance collection.
- A CALDERA-based attack emulation methodology that automates multi-stage APT campaigns across IT-OT-PLC boundaries.
- A provenance-capable dataset capturing complete attack chains from initial IT compromise through to physical process manipulation.

2 Background

2.1 ICS/OT and the Purdue Model

ICS represent the integration of computers, electrical, and mechanical devices used to monitor and control physical industrial processes [8]. These systems form the backbone of critical infrastructures including power plants, water treatment facilities, and manufacturing industries, where they manage complex operational processes through automated control mechanisms.

ICSs are fundamentally composed of two interconnected domains: Operational Technology (OT) and Information Technology (IT). The OT network encompasses the hardware and software dedicated to monitoring and managing industrial equipment, assets, processes, and events, including Programmable Logic Controllers (PLCs), sensors, and actuators that directly interface with physical processes. While OT systems contain computing elements, they differ fundamentally from traditional IT in their priorities. OT emphasises availability, real-time operation, and safety over the confidentiality and data integrity priorities of IT systems. Conversely, the IT network contains traditional computing infrastructure such as workstations, databases, and enterprise systems used for information processing and business operations [8].

Historically, IT and OT networks operated in isolation from one another. However, the phenomenon of IT/OT convergence has led to their interconnection, driven by the digitisation of industrial processes and the need for integrated data analytics. While this convergence enables enhanced operational efficiency and real-time decision-making capabilities, it simultaneously exposes previously air-gapped OT systems to cyber threats originating from IT networks [8]. The reference architecture for ICSs is the Purdue Model, which divides the system into hierarchical zones with distinct functions. At the foundation lies the Control Zone (OT network), subdivided into multiple levels: Level 0 contains sensors and actuators directly interfacing with physical processes, Level 1 includes intelligent devices such as PLCs and Remote Terminal Units (RTUs), Level 2 encompasses Human Machine Interfaces (HMIs) and control room workstations, and Level 3 contains manufacturing operations systems. Above this

sits the Enterprise Zone (IT network) handling business logistics and enterprise operations. Critically, a Demilitarised Zone (DMZ) serves as the intermediary layer controlling data exchange between the Control and Enterprise zones [8].

2.2 Advanced Persistent Threats

APTs are prolonged and targeted cyber attacks against systems and organisations. They employ multi-stage methodologies and can remain undetected within target environments for extended periods, often months, maintaining stealthy access to achieve objectives such as cyber-espionage, sabotage, or subversion [36]. APTs are often state-sponsored actors due to their complexity and the resources required to execute such attacks, as well as their military style of attack structures. They use multi-stage techniques compared to common cyber criminals. These stages are known as the APT life cycle and are often highly detailed and planned. While there are many different interpretations of these stages, they generally share ideologies.

State-sponsored APTs generally aim to either sabotage the targeted system or perform espionage by extracting data. These attacks are rarely monetary in nature, with some exceptions such as LAZARUS[15]. In the case of ICS attacks, the final phase is almost always about disruption and/or destruction rather than data exfiltration. APTs are generally identified and named by security companies such as Dragos, Mandiant, ESET, and Symantec who analyse malware found in compromised systems[24].

We draw our definition for ICS attacks from the meaning of ICS malware as defined by Dragos[17]: software that is ICS capable (able to speak ICS-specific protocols, communicate with PLCs, modify project files, or interact with engineering workstations), designed with malicious intent, and has the ability for adverse effects on OT networks. We expand this definition to include attacks which are impactful in ICS and OT systems, as this provides a broader view into the attacks and behaviours of APTs.

2.3 Provenance and Existing Datasets

Provenance-based intrusion detection systems (PIDS) track causal relationships between system entities such as processes, files, network connections, and sensors, to reconstruct attack chains and identify malicious activities [36]. Unlike traditional approaches that analyse events in isolation, PIDS model system activity as directed provenance graphs where nodes represent entities and edges capture causality relationships [9]. This graph-based representation enables backward and forward tracing from detected anomalies, making PIDS particularly effective against sophisticated multi-stage APTs.

2.3.1 IT Provenance Datasets. The DARPA Transparent Computing (TC) programme produced five engagement datasets (E1-E5, 2016-2019) as benchmark datasets for provenance-based intrusion detection research [36]. These datasets capture multi-stage APT scenarios on enterprise IT infrastructure using the Transparent Computing Common Data Model (TCCDM) format. The DARPA Operationally Transparent Cyber (OpTC) dataset (2020) extended this work, collecting over 17 billion events from 1,000 Windows 10 hosts [9]. Despite their scale, these datasets focus exclusively on enterprise IT environments with no coverage of operational technology or physical processes.

2.3.2 ICS Attack Datasets. Three prominent ICS security datasets demonstrate sophisticated attack generation capabilities while highlighting the critical gap in provenance-based data collection. The Secure Water Treatment (SWaT) testbed dataset from iTrust contains both normal operational data and records of 36 cyber-physical attacks collected over 11 days [3]. The Water Distribution (WADI) dataset extends SWaT, capturing 127 sensor measurements over 16 days including 15 attack types [2]. The Hardware-in-the-Loop Augmented ICS (HAI) dataset captures 59 sensor measurements with 28 stealthy attacks [29]. However, none of these captures system-level provenance data and instead provide only sensor measurements, network traffic, and operational logs. This prevents researchers from performing root cause analysis or cross-domain causality tracking.

2.3.3 CICAPT-IIoT. The CICAPT-IIoT dataset (2024) provides the first provenance-based APT dataset for Industrial Internet of Things environments [13]. Using MITRE CALDERA for attack emulation, it captures over 20 attack techniques with labelled provenance data. However, it uses software PLC emulation rather than physical hardware, focuses on IoT edge devices without enterprise IT infrastructure, and lacks properly segmented DMZ architecture. Most critically, it does not capture complete IT->OT->PLC attack chain traversal. CICAPT-IIoT does validate the CALDERA-based methodology, supporting the feasibility of our approach. Table 8 summarises the architectural coverage of these datasets alongside our own.

3 Methodology

3.1 ICS Attack Analysis

To inform the design of our attack scenarios, we conducted a systematic analysis of nine major ICS attacks spanning from 2010 to 2024 (Figure 1). We selected these attacks for their recency, impact on target systems, or architectural complexity. Our analysis drew on technical reports from Dragos, ESET, Symantec, and Mandiant, as well as academic literature and the MITRE ATT&CK for ICS framework[25]. MITRE ATT&CK organises adversary behaviour hierarchically: tactics represent the adversary's objectives, techniques describe specific methods used, and procedures detail implementation specifics [18]. The framework includes a specialised ICS matrix which we use throughout this work.

Miller et al. (2021) [23] conducted a comprehensive historical analysis of 43 ICS cyberattacks spanning from 1988 to 2021, demonstrating a fundamental shift in ICS threats around 2009. Prior to 2009, attacks were predominantly perpetrated by individuals with motivations centred on personal grievances or financial gain. From 2009 onwards, organised groups and nation-state actors became the dominant threat agents, with attacks concentrated in politically contested regions and targeting critical infrastructure[23].

Figure 1 places our nine selected attacks in this post-2009 window and records their attribution, target sector, and dominant industrial protocol. Three features stand out. Attribution is heavily state-linked: four of the nine (Havex, BlackEnergy, Industroyer, Industroyer2) are widely reported in open-source analyses as associated with suspected Russian GRU or FSB units, with others similarly suspected of state involvement, US/Israel on Stuxnet, Chernovite on PIPEDREAM, and Russian actors on TRISIS and FrostyGoop; only Fuxnet, associated with the BlackJack group, is not suspected of direct state backing. The cadence has also accelerated, with four of the nine occurring in 2022 or later and two pairs (Industroyer2 with PIPEDREAM in 2022, FrostyGoop with Fuxnet in 2024) clustering within months. Protocol

Dataset	Year	IT	OT	PLC	Purdue	Duration	Provenance	Attack Types	Testbed Type
DARPA OpTC	2020	✓	×	×	L5 Only	14 Days	✓(TCCDM)	Multi-stage APT	Virtual
SWaT	2016	×	✓	✓	L2-L0	11 Days	×	36 Attacks	Physical
WADI	2017	×	✓	✓	L2-L0	16 Days	×	15 Attacks	Physical
HAI	2021	×	✓	✓	L2-L0	4 Days	×	28 Attacks	Hybrid
CICAPT-IIoT	2024	✓	✓	Partial[a]	L3-L1	2 Phases	✓(CSV)	20+ TTPs	Hybrid
Our Work	**2026**	✓	✓	✓	**L4-L0**	**4 Days**	✓(Raw)	**53 TTPs**	**Physical**

Table 1: Comparison of ICS security datasets showing architectural coverage and data collection characteristics. [a]CICAPT-IIoT uses Raspberry Pis with OpenPLC to simulate PLCs yet they have physical sensors. Our work is to be relased in raw format but used as Prov-DM for the demo.

coverage is heterogeneous, spanning Profinet/S7, OPC Classic, IEC 60870-5-104, TriStation, Modbus TCP, and Meter-Bus, which limits how far any single protocol-specific defence can transfer across the set. These observations directly inform the scenario design in Section 3.3: the campaign must exercise state-actor tradecraft rather than opportunistic scripting, and it must span multiple Purdue layers rather than concentrate at one. From this analysis, the nine selected attacks can be categorised into three architectural paradigms that reflect the evolution Miller describes.

3.1.1 Monolithic, Purpose-Built Malware. Monolithic malware represents the first generation of ICS-specific cyber weapons. They are highly integrated and single-purpose tools engineered for particular industrial targets. These attacks are characterised by tightly coupled components, extensive pre-operational intelligence requirements, and limited adaptability beyond original design parameters.

Stuxnet exemplifies this paradigm as the first confirmed malware designed to cause physical damage to industrial processes. Discovered in mid-2010, it utilised four zero-day vulnerabilities and two stolen certificates, targeting five Iranian organisations[12]. Initial compromise occurred through removable media exploiting the LNK vulnerability (CVE-2010-2568), which was also used for propagation and lateral movement[12]. Upon identifying Siemens WinCC/PCS7 systems, Stuxnet replaced legitimate DLLs to control PLC-Windows traffic, installing rootkits using stolen certificates for persistence[19]. Stuxnet then changed the states of the PLCs to interfere with operation while displaying fake data to the operators.

TRISIS demonstrates even narrower targeting specificity. Discovered in 2017 at a Middle Eastern petrochemical plant with attackers present since 2014, this compiled Python script specifically targeted Schneider Electric Triconex safety systems[11, 19, 23]. Communicating via TriStation protocol, TRISIS exploited a controller configured in PROGRAM mode during operation to write firmware memory and upload modified ladder logic altering safety set points[11, 19]. The attack exemplifies monolithic scalability limitations, as each SIS requires unique process knowledge, making deployment across victims difficult without significant rework[11].

Havex, deployed in the Dragonfly campaign from 2011, represents a monolithic reconnaissance framework compromising over 2,000 sites in energy and petrochemical sectors[10, 31]. Employing spear-phishing, watering hole attacks, and supply chain compromise of three ICS software vendors[26, 31], Havex deployed an ICS-specific plug-in exploiting the OPC protocol to systematically map industrial equipment[10, 26].

Industroyer2, discovered and prevented by ESET researchers on April 8, 2022 during Russia's invasion of Ukraine, represents a strategic pivot toward monolithic design despite descending from modular malware[27]. Unlike Industroyer(1), which we discuss below, Industroyer2 emerged as a streamlined tool focusing exclusively on IEC 60870-5-104 protocol with all attack parameters hardcoded directly into the executable[27, 33]. This traded adaptability for operational speed, requiring complete recompilation for different environments. Deployments were accompanied by multiple wiper components (CaddyWiper, ORCSHRED, SOLOSHRED, AWFULSHRED), and the attack was attributed to Sandworm/ELECTRUM[27].

3.1.2 Modular, Framework-Based Malware. Modular frameworks prioritise adaptability and component reusability over rigid single-purpose design, employing plug-in architectures and protocol-specific payloads enabling adaptation to diverse industrial environments without complete redevelopment.

BlackEnergy's evolution illustrates this transition. Version 1 functioned as a simple DDoS tool, while Version 2 added plug-in support. BlackEnergy V3 represented significant capability expansion with virtual machine detection, memory-resident operation, and critically, plug-in architecture supporting modular capabilities[19]. The 2015 Ukraine attack demonstrated this modularity through spear-phishing, driver-based persistence, RAT deployment, and credential harvesting enabling VPN-based OT network access[19, 28]. Before execution, attackers deployed the KillDisk plug-in to wipe computers and prevent remote recovery[19, 28].

Industroyer(1), also known as CRASHOVERRIDE, disrupted Kiev's power on December 17, 2016 and marked the first framework designed to attack electric grids[10, 23]. Its core strength was a launcher orchestrating four separate protocol modules[7]: IEC 104 (masquerading as master controller), IEC 61850 (circuit breaker state manipulation), OPC DA (denial of visibility), and IEC 101[7, 10]. Destruction mechanisms included data wipers and DoS tools exploiting CVE-2015-5374 in Siemens SIPROTEC relays[7, 10].

PIPEDREAM, discovered in early 2022 before deployment in any victim environment, represents the seventh-known ICS-specific malware[16, 22]. Five components target different aspects: EVILSCHOLAR (Schneider CODESYS PLCs), BADOMEN (Omron controllers and servo drives), MOUSEHOLE (OPC-UA servers), DUSTTUNNEL (host reconnaissance and C2), and LAZYCARGO (rootkit installation)[16, 22]. Executing 36 ICS attack techniques (46% of MITRE ATT&CK for ICS)[16], PIPEDREAM demonstrates the most comprehensive ICS attack capability observed to date.

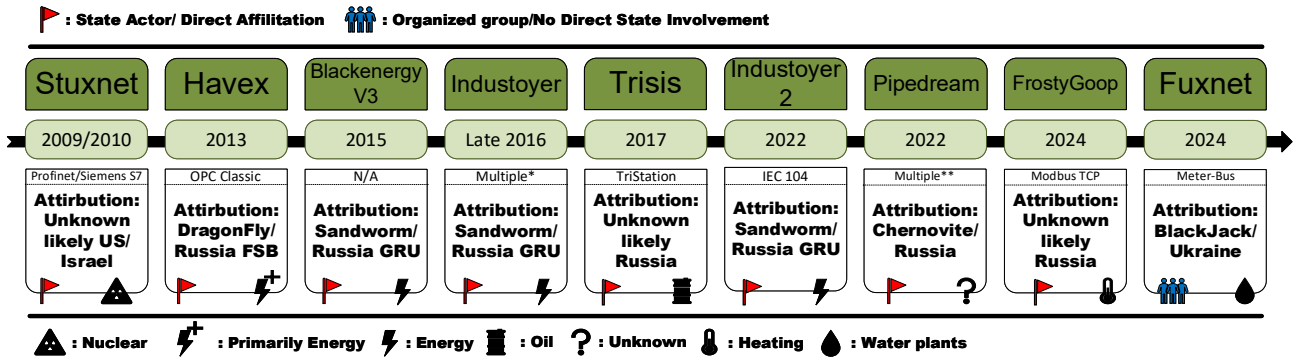

Figure 1: Timeline of the nine ICS attacks analysed in Section 3.1, showing year, target sector, protocol, and attribution. State-sponsored or state-linked actors dominate after 2010.

3.1.3 Living-off-the-Land and Protocol Abuse. Living-off-the-land attacks minimise custom malware, relying on legitimate industrial tools and protocols while exploiting inherent system functionality. This challenges traditional detection by blurring malicious and legitimate activity, requiring behavioural analysis rather than signatures.

FrostyGoop, the ninth ICS-specific malware, is the first to leverage Modbus TCP for physical impact[1]. Deployed January 2024 against Ukrainian energy infrastructure, it disrupted heating to over 600 apartments for two days[1]. Written in Golang with openly available libraries[4], FrostyGoop implemented three standard Modbus functions[1, 4], operations indistinguishable from legitimate communications. Attackers manipulated ENCO firmware, downgrading to versions lacking monitoring while manipulating temperature and pressure readings[1].

Fuxnet, deployed April 2024 by Blackjack against Moskollektor, targeted Moscow's underground infrastructure monitoring, compromising approximately 2,659 sensor gateways with 1,700 rendered inoperable[21, 32]. Initial access exploited default passwords requiring no custom exploits[32]. Following ten months of access establishment[32], Fuxnet executed destructive actions including disabling remote services, deleting file systems, corrupting NAND flash, and implementing Meter-Bus fuzzing flooding sensors with malformed packets[21, 32].

Both FrostyGoop and Fuxnet demonstrate how living-off-the-land attacks leverage legitimate protocol characteristics to achieve disruption with minimal custom development. The key differentiator from monolithic malware is the abuse of native protocol functionality rather than the exploitation of implementation vulnerabilities. Yet both confirm that even generic protocol abuse requires extensive pre-attack reconnaissance.

3.1.4 TTP Extraction and Evolutionary Trends. MITRE ATT&CK is a publicly available knowledge base of adversarial techniques based on real-world observations [25]. The framework organises adversary behaviour hierarchically: tactics represent the adversary's tactical objectives, techniques describe specific methods used to achieve those objectives, and procedures detail implementation specifics [18]. The framework includes a specialised ICS matrix which we use throughout this work.

We mapped the nine selected attacks against MITRE ATT&CK for ICS techniques, extracting 69 unique techniques organised by attack lifecycle stage. The analysis reveals that the impact stage is the most heavily populated, as ICS attacks often differentiate in the methods they use to disrupt physical processes. Reconnaissance and discovery techniques are evenly utilised across attacks, reflecting the crucial role of pre-operational intelligence. The persistence and C2 stages show particular density in modular malware like PIPEDREAM, which requires ongoing communication with command infrastructure.

ICS attacks have evolved significantly in sophistication. Stuxnet (2010) marked a shift from individual actors to nation-state actors with political goals. Architecturally, attacks progressed from monolithic designs requiring extensive target-specific development, through modular frameworks enabling component reuse, to living-off-the-land approaches leveraging existing infrastructure and protocols to minimise detection. This progression informed our scenario design: rather than emulating a single attack, we drew techniques from across these paradigms to construct a representative multi-stage campaign spanning the full IT-OT-PLC attack chain.

3.2 Testbed Architecture

The testbed implements a Purdue Model segmented ICS environment controlling a physical hydro water plant (Figure 3). The Enterprise (L4) and Engineering (L3) zones run as virtual machines on two separate hypervisor hosts, providing clean physical separation between IT and OT, and are bridged by a jumpbox serving as the L3.5 DMZ. Zone segmentation is enforced by a pair of OPNsense routers, one per zone, through which all inter-zone traffic must traverse. Levels 1 and 0 are physical: Siemens S7 PLCs and HMIs connected via an unmanaged switch whose mirror port feeds the packet capture pipeline. It is also important to note that in order to make the use of Caldera feasible defender exclusions were added. Caldera's default agent Sandcat is profiled extensively by Windows Defender. This means that without turning off certain features of defender like realtime protection and adding exclusions it is not possible to use Sandcat agent. In a real attack APTs would likely have their own agent that is specifically compiled for the attack and might even have stolen certificates for signing their binaries. This means they handle evasion before the attacks even start.

3.2.1 Attacker and C2 Infrastructure. Two Raspberry Pi devices sit outside the enterprise perimeter. The first is the attacker host. It runs CALDERA 5.3.0 on port 8888 alongside an HTTP staging server on port 5555 for delivering pre-staged binaries to implanted agents. An internal port redirect forwards inbound traffic on port 80 to the CALDERA listener, so agents check in against a standard HTTP port rather than the non-standard

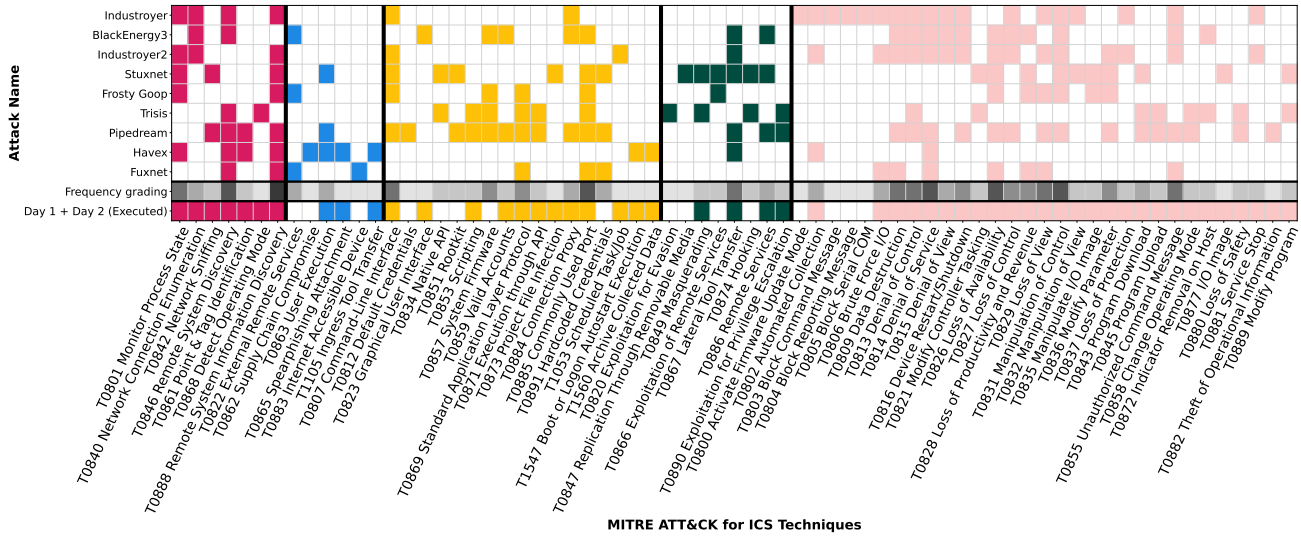

Figure 2: Heatmap characterising the capabilities of nine major ICS malware families and the emulated scenario (bottom row) against the MITRE ATT&CK for ICS framework. Red (left) indicates Reconnaissance and Discovery techniques, Blue represents Initial Access and Incursion, Yellow denotes Persistence and C2, Green signifies Lateral Movement and Evasion, and Pink (right) highlights Data Extraction and Impact techniques resulting in physical or operational consequences.

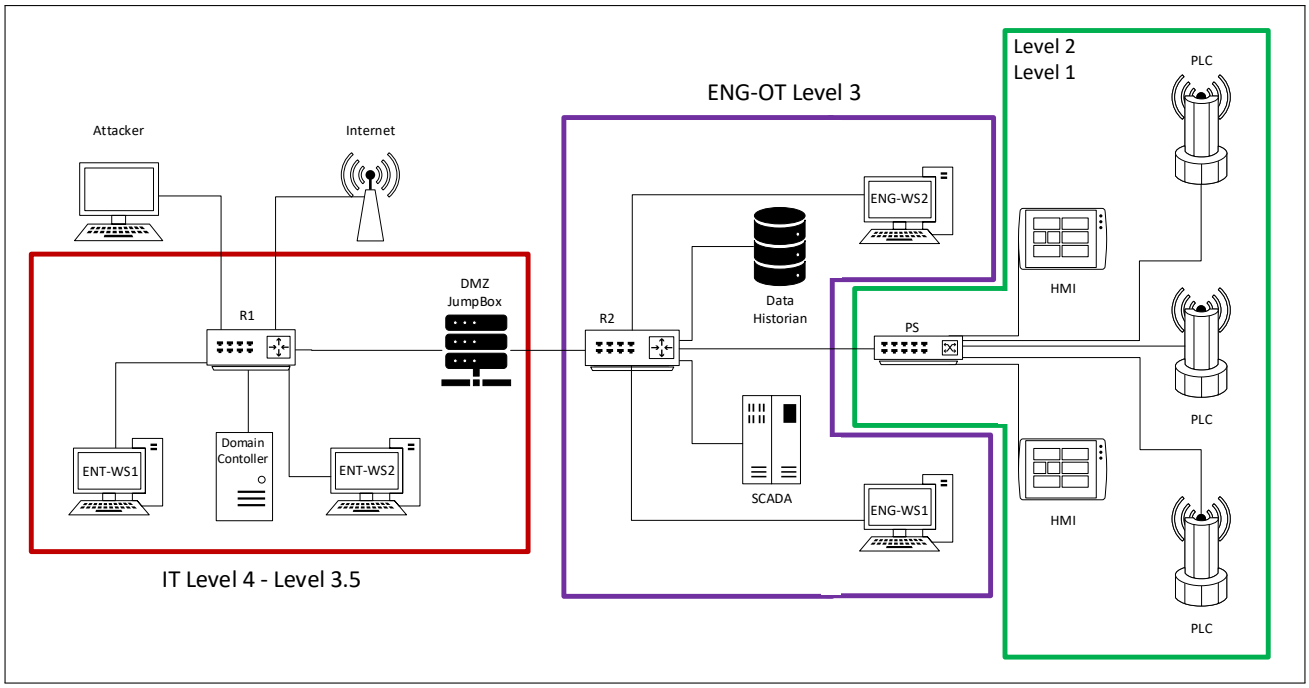

Figure 3: Testbed architecture spanning Purdue Model levels L0 to L4. Zones are coloured: Enterprise (red), Engineering (purple), physical process (green).

CALDERA default. This machine is not connected to the internet; it only receives traffic from the testbed. The second Pi is a harmless network bridge. It connects to `eduroam` Wi-Fi and performs a double-NAT that exposes the uplink to the Enterprise router, providing the sole internet path consumed by the two user workstations. It has no attacker role. The Enterprise router additionally rewrites DNS responses for `caldera-server.com` to the attacker Pi, so agents resolve a plausible external hostname rather than a bare IP. The `eduroam` network enforces client isolation, preventing direct connectivity between devices on the wireless segment, so the uplink cannot be pivoted, along with the router VM this stops any attempts to reach the attacker Pi from other hosts on the same network.

3.2.2 Enterprise Zone. The Enterprise zone is a small Active Directory environment (`Enterprise.local`) with a domain controller and two user workstations. The DC additionally serves as the zone's DNS resolver, so every name lookup from a workstation traverses it, and hosts an SMB share used for routine

Host	Role
DC	Windows Server 2019, AD + zone DNS
ENT-WS1, ENT-WS2	Windows 10 user workstations
Jumpbox	Windows 10, dual-homed IT/OT gateway
R1	OPNsense zone router

Table 2: Enterprise zone hosts (192.168.1.0/24, Level 4 and the L3.5 DMZ).

file exchange between enterprise users; access is restricted to domain-joined IT hosts. A dual-homed jumpbox sits on both the IT and OT subnets and is the only sanctioned crossing between zones; it is domain-joined but deliberately has no internet access, mirroring the common industrial pattern where operators RDP from the enterprise network into an OT-facing gateway. No machine below the jumpbox is domain-joined; authentication in the OT zone is local only. Internet for the two user workstations is provided through R1 via the dedicated Raspberry Pi uplink (Section 3.2.1).

Host	Role
ENG-WS1, ENG-WS2	Windows 10, Siemens TIA Portal
SCADA	Windows Server 2019, Siemens WinCC +TIA portal
Historian	Windows Server 2019, Node-RED + InfluxDB + Grafana
R2	OPNsense zone router

Table 3: Engineering zone hosts (192.168.3.0/24, Level 3).

3.2.3 Engineering Zone. The Engineering zone houses the OT side supervisory infrastructure. The two engineering workstations (ENG-WS1, ENG-WS2) run Siemens TIA Portal and hold local working copies of the PLC project files, while the authoritative copy and final compile-and-download step reside on the SCADA host. SCADA exports the project directory over SMB as `ENG_Shared_Projects`; the two engineering workstations mount the share via matched local accounts so authentication stays within the zone, and the historian takes no part in the share. The share is not reachable from the Enterprise zone. SCADA itself also runs Siemens WinCC and provides the primary operator view of the process. The data historian runs a Node-RED flow that polls the PLCs over S7Comm, persists process variables into InfluxDB, and surfaces them through Grafana dashboards, a plausible OT-side telemetry stack [20]. Authentication is local on every host, and none of the machines are domain-joined, so a compromise in the Enterprise zone does not translate directly to OT access. No host in this zone has internet connectivity, so all attack tooling must either be pushed manually to ENT-WS1/2 then to Jumpbox and finally to OT machines, or use Caldera agents built-in payloads feature which handles the transfer automatically over implants.

3.2.4 Physical Process. The physical layer is the Glasgow Cyber Defence Lab (GCDL) GREENS testbed [14], a scaled hydroelectric generation environment comprising two generators, associated pumps, a sump and reservoir tank along with a safety interlock line. Generators are turned on and off using an HMI screen. Then a bigger pump comes on periodically to pump the water back up to the reservoir tank keeping the physical process going. The safety interlock prevents the top tank from overflowing. Control is distributed across three physical Siemens PLCs: two S7-1200 units (192.168.3.5, 192.168.3.7) drive the generator and pump logic, while a legacy S7-300 (192.168.3.9) handles the safety-critical interlock. The larger water return pump is driven through a variable speed drive (VSD) that takes its speed setpoint from PLC memory. Two physical HMIs (Level 2) provide operator visibility and control. All Level 1 and Level 0 traffic speaks S7Comm and S7CommPlus over TCP/102 along with other Layer 2 traffic and crosses a physical 24-port unmanaged switch; a configured mirror port on this switch is the sole observation point for process layer traffic.

3.2.5 Installed Software and Baseline Activity. Enterprise workstations run a standard office software stack: LibreOffice for document editing and spreadsheets, Google Chrome for web browsing, Adobe Acrobat Reader for PDFs, Notepad++ for text editing, and the Proton Mail desktop client (and web interface) for email. The jumpbox and domain controller were left as near-default Windows installations with no additional productivity software. The engineering workstations and SCADA host run Siemens TIA Portal for PLC project development and monitoring; SCADA additionally runs Siemens WinCC for process visualisation. The historian runs Node-RED, InfluxDB, and Grafana for process data collection and dashboarding. All hosts run Sysmon for system-level event logging.

Semi-structured benign activity was performed on all data collection days to ensure realistic mixed traffic in the dataset. In the enterprise zone, workers created and edited documents in LibreOffice, read PDFs, sent and received email, and browsed the web including Siemens support pages and general news sites. Workers operated under a distracted-worker premise, with activity occurring in intermittent bursts rather than continuously. Access to the engineering zone followed the standard operational path: enterprise users connected to the jumpbox via RDP using saved credentials, then from the jumpbox into OT hosts. Engineering activities such as opening TIA Portal, going online with PLCs to inspect tags, and reviewing process dashboards were sometimes performed remotely through this RDP chain rather than at the local console. On the second dedicated benign day, these activities also included an update to the water-control PLC via TIA Portal, which produced a legitimate PLC download within the benign record that is retained in the dataset as a realistic example of routine engineering activity rather than excluded. On SCADA, the operator HMI was monitored through WinCC, and on the historian, Grafana dashboards provided process visibility. At the physical layer, operators periodically monitored the HMI panels and visually inspected the running process. On attack days, this same benign activity continued in parallel with attack operations, providing a realistic background of legitimate operations against which attack traffic can be distinguished.

3.3 Attack Scenario Design

Figure 2 summarises the technique coverage of the emulated scenario against the nine historical attacks. The bottom row of the heatmap shows the 53 techniques executed across both days of the scenario. These were selected through a prioritisation process balancing three criteria: feasibility within the testbed topology, precedent in the historical attack corpus, and ensuring that every Purdue Model layer from L4 down to L0 is exercised by at least some techniques. Of the 69 techniques observed in the historical corpus, 16 were not executed. Four are structurally

inapplicable to the testbed: Block Serial COM requires serial links but all PLC communication uses TCP/102, External Remote Services requires internet-exposed remote access services such as VPNs into the OT network which the testbed does not have, Internet Accessible Device does not apply as no PLC or OT host has internet connectivity, and Default Credentials were not present in the testbed as all accounts were configured with non-default passwords. Six fall outside the scope of this work: Activate Firmware Update Mode and System Firmware target controller firmware rather than the program logic level targeted by our attacks, Rootkit, Native API, and Hooking require kernel-level driver manipulation or direct Win32/NT API calls and DLL import table interception which were unnecessary given that PowerShell and Python provided sufficient execution capability, and Exploitation for Evasion was not applicable as endpoint protection was disabled on all VMs to reflect the reality that APT groups typically employ custom-compiled tooling with unique signatures, stolen code-signing certificates, and obfuscation techniques that render signature-based detection ineffective. Four corpus techniques were addressed through related but distinct methods rather than direct replication: Block Command Message and Block Reporting Message involve intercepting or dropping specific protocol messages between operator stations and controllers, typically through network-layer techniques such as ARP poisoning or injecting TCP resets. Our scenario did not perform packet-level interception; instead, connection-slot exhaustion prevented new S7Comm sessions from being established, which denies operators the ability to issue new commands or receive new reports but does not affect already-established sessions. Supply Chain Compromise involves distributing a trojanised artefact through a trusted channel for a victim to consume unknowingly, but in our scenario the attacker drove the TIA Portal download directly via RDP rather than waiting for an engineer, so the file staging is better captured by Project File Infection. Exploitation of Remote Services was unnecessary as all lateral movement used valid harvested credentials over WinRM rather than exploiting a remote service vulnerability. The remaining two were not exercised as adversary choices: Hardcoded Credentials were present in the testbed but were deliberately not used as simply reading known passwords from configuration files would generate minimal provenance data compared to active credential harvesting through DPAPI and LSASS, and Replication Through Removable Media was not pursued as the engineering effort to integrate USB propagation into the automated attack chain was not justified given the existing P2P agent path.

The scenario follows a single two-day structure. Day 1 is a reconnaissance and exfiltration campaign drawing inspiration from the Havex/Dragonfly espionage model. Day 2 is a return engagement with offensive payloads crafted from Day 1 intelligence, drawing from TRITON (safety override), Stuxnet (PLC logic manipulation), and Industroyer (coordinated disruption and denial of view). This mirrors real APT tempo: the intelligence collected during espionage directly enables the precision of subsequent destructive operations. Exfiltrated TIA Portal project files are modified offline to carry malicious logic, and PLC memory maps identifying which addresses control safety interlocks and physical outputs allow Day 2 attacks to target exact process variables.

Day 1 begins with a spearphishing email delivering a macro-enabled document to an enterprise workstation. The macro establishes a command-and-control agent that checks in over standard HTTP. Local privilege escalation exploits PrintNightmare to create an administrative account, and a scheduled task is registered to re-launch the agent under the new account at user logon. This logon-triggered task is weak persistence in practice: the backup administrator account is not used interactively, so the trigger is unlikely to fire. Credential harvesting from DPAPI and CMDKEY stores yields domain credentials, enabling lateral movement to the jumpbox via PowerShell Remoting using domain account authentication, with agent deployment executed through WMI process creation on the remote host. From the jumpbox, a peer-to-peer agent relay extends C2 reach into the OT zone without requiring internet connectivity on any OT host. Since no OT machine is domain-joined, lateral movement into the engineering zone uses a different mechanism: agents are deployed to the historian, SCADA server, and engineering workstations over WinRM using harvested local OT credentials. Once inside the OT zone, the attacker conducts host enumeration, discovers Siemens engineering software installations, and performs passive S7Comm network sniffing to identify active PLC traffic. Active PLC reconnaissance follows: CPU information, operating modes, full memory dumps, and program block captures are collected from all three PLCs. The day concludes with collection and exfiltration. TIA Portal project files from the engineering share, InfluxDB process historian data, and Node-RED configuration are compressed, staged, and exfiltrated through the P2P relay chain back to the C2 server. Critically, all PLC interaction on Day 1 is read-only: no writes, no mode changes. The attacker observes and collects, producing four key outputs: complete TIA project files for offline modification, PLC memory address maps linking specific bits to safety interlocks and physical outputs, OT credentials for re-entry, and sufficient process understanding to inform targeted attacks.

Day 2 opens with C2 presence re-established across the IT-to-OT path. The first offensive action deploys a shadow controller, a Python overlay process on an engineering workstation that writes attacker-defined speed and timing values to PLC memory while SCADA and the HMIs continue displaying plausible but incorrect readings. This runs for an extended period to generate subtle manipulation traffic before escalation. The attacker then stages a modified TIA Portal project on the engineering share, kills TIA Portal processes on all engineering workstations and the SCADA host, and uses a proxied RDP session to open TIA Portal on SCADA and download the malicious hardware and software configuration to all three PLCs. During this process, the attacker changes PLC access passwords, disables the web servers, and changes HMI web interface credentials on every controller. The HMIs can no longer authenticate, TIA Portal cannot reconnect to stop the CPUs, and the web diagnostic interfaces are no longer accessible. The only remaining operator action is to pull physical power.

The attack then escalates to direct bus-level disruption. Safety interlock bits are cleared, disabling flood-protection logic. Brute-force I/O rapidly toggles physical outputs at 100 ms intervals. A connection-slot exhaustion attack locks all three PLCs against new S7Comm sessions, though already-established HMI connections are not severed. On the legacy S7-300, function blocks are deleted and the CPU is stopped and hot-restarted, causing it to fault with no executable logic. In parallel, local TIA Portal backups on the engineering workstations are destroyed, the engineering share is wiped, InfluxDB data is partially deleted, and all persistence mechanisms (Run keys, scheduled tasks, staging artefacts) are removed. The combined effect is that operators

lose control of all three PLCs through different mechanisms: password lockout and malicious logic on the S7-1200s, program deletion and CPU fault on the S7-300. Engineering recovery paths are destroyed with no project backups remaining, and process telemetry is degraded.

Three operational constraints shaped the execution. First, agents deployed to the OT zone over WinRM died after about two hours, likely the WinRM idle session timeout on the remote side. A registry Run key was written on the historian and SCADA hosts as a fallback, but the HKCU entry only fires on a new logon, and users stay logged in rather than logging out and back in during an operation. In practice this was handled by re-deploying the agent over WinRM before the original died, which restarted the clock. Both the original and re-deployed agents ran exfiltration tasks in parallel until the original was killed manually; on Day 1 this manual kill happened before the two-hour timer would have elapsed, so the visible agent lifetimes in the reports are shorter than the observed timeout. Second, the multi-hop P2P relay chain constrains bandwidth, so large transfers such as the TIA project files required splitting and compression before dropping the payload, this meant that the historian InfluxDB data was stolen in small 4 hour chunks. Third, TIA Portal operations such as opening projects, compiling, and downloading to PLCs require graphical interaction and cannot be automated through CALDERA abilities. These steps were performed via a proxied RDP session and recorded as manual log entries rather than automated C2 links.

3.4 Configuration, Automation and Attack Tooling

The attack scenario was orchestrated through CALDERA 5.3.0, an open-source adversary emulation framework developed by MITRE. CALDERA executes attack chains through *abilities*, individual technique implementations written as shell commands, which are grouped into *adversary profiles* representing sequential attack stages. A REST API and the MAGMA web interface allow operators to launch operations, inject manual commands(links), and inspect agent output in real time. Rather than running the entire scenario as a single monolithic operation, the attack was divided into stage-based adversary profiles (e.g. “Stage 2: Discovery”, “Stage 7: Lateral Movement to Jumpbox”) that were executed sequentially through a Python runner script. This design allowed each stage to be restarted independently if an agent died or a command failed, and produced per-stage operation logs that simplify provenance labelling after the fact.

The Day 1 tooling mixes living-off-the-land techniques with pre-staged executables: built-in Windows commands and PowerShell drive most of the reconnaissance and lateral-movement steps, with custom binaries reserved for capabilities that have no native equivalent such as credential extraction and protocol-specific PLC interaction. Initial access delivers a macro-enabled document whose macro downloads and executes a PowerShell script from the C2 staging server to establish the first agent. Discovery and enumeration use native commands such as `whoami`, `ipconfig`, `net user`, and `Get-Process`. Domain reconnaissance uses PowerView, an offensive PowerShell module from the PowerSploit framework that provides Active Directory enumeration functions such as `Get-NetDomainController` and `Get-NetComputer`. Privilege escalation exploits PrintNightmare to create a local administrator account, after which a fodhelper-based UAC bypass script downloads and launches an elevated agent running under the new account. Credential access uses Mimikatz (staged as `m.exe`) to extract DPAPI master keys and decrypt credentials from Windows Credential Manager and CMDKEY stores. Lateral movement from the enterprise zone to the jumpbox uses PowerShell Remoting for session establishment with WMI for remote process creation, while movement into the OT zone uses WinRM with harvested local credentials. On the historian, Node-RED credential store, flow definitions, and settings file (which contains the encryption key) are copied for offline decryption then a Python script (`influx_export.py`) uses the API token to export InfluxDB process data in four-hour chunks. PLC reconnaissance uses two Python scripts deployed to the engineering workstations: `plc_cpu_info.py` enumerates CPU model, firmware version, and operating mode across all three PLCs, and `plc_mem_rw.py` performs full memory area dumps and program block extraction.

All PLC interaction scripts use the Snap7 library, an open-source S7Comm protocol implementation that communicates over TCP port 102. On the S7-1200 controllers (192.168.3.5 and 192.168.3.7), Snap7 can read and write memory areas when the PUT/GET access level is enabled in the hardware configuration, but cannot upload or delete program blocks as the firmware rejects these operations. The S7-300 (192.168.3.9) imposes no such restriction, permitting full block-level operations including deletion.

Day 2 deploys several purpose-built attack tools to the engineering workstations, spanning the monolithic and modular paradigms introduced in Section 3.1. The *shadow controller* (`plc_shadow_ctrl.py`) takes the monolithic approach, written specifically for this hydroelectric process with the target’s memory layout and control logic hardcoded throughout. It is a Python overlay process that connects to the water and generator PLCs and takes over the pump cycle. It first sets an internal marker flag (M0.5, referred to as MASTER_OFF) that disables all function blocks in the PLC program, giving the attacker exclusive control. In Siemens S7 PLCs, I-registers hold input values from field sensors, M-flags serve as internal markers and working variables for the control logic, and Q-registers hold output values that drive physical actuators. The shadow controller writes an attacker-defined speed setpoint to MD600 (a marker double-word), which is the internal variable that the variable speed drive reads to set the actual pump speed. Meanwhile, Q-registers QW200 and QW250 are written with the normal expected display values so that SCADA and the HMIs continue showing plausible readings. The controller also recomputes normalised flow values from the real sensor input (IW96) and writes them back to I-area addresses, maintaining consistent flow displays. Speed and cycle timing parameters are read from plain-text configuration files (e.g. `shadow_single_speed.txt`, `shadow_dual_on.txt`), allowing the attacker to adjust the manipulation in real time. A separate kill flag (`shadow_kill.flag`) triggers a clean shutdown and restores original PLC values.

The *bus controller* (`plc_dos_bus.py`) takes the modular approach: it targets the S7-300 and combines denial of service with destructive actions under file-flag control, reusing a stand-alone connection-flood component rather than bundling that capability inline. It launches a connection-flood subprocess (`plc_dos.py`) that saturates the PLC’s S7Comm connection slots, blocking new connections from operators and HMIs which use the same S7Comm protocol on port 102. A dedicated worker connection retained by the bus controller then performs actions triggered

by writing `true` to flag files: `safety.flag` clears the safety interlock bits and zeros all physical outputs, `brute.flag` rapidly toggles outputs at 100 ms intervals, and `delete.flag` triggers a one-shot sequence that stops the CPU, deletes function blocks FC1 and FC3, and hot-restarts the PLC with no executable logic. The same connection-flood script (`plc_dos.py`) was also used independently earlier in the attack sequence, launched against all three PLCs in parallel to lock out operators while TIA Portal processes were killed and the RDP session was prepared. It was then stopped before the malicious project download so that TIA Portal could connect to the PLCs for the malicious project download.

The malicious TIA Portal project was prepared offline using the files exfiltrated on Day 1. Modifications were applied per-PLC: the flood-control PLC received access protection at level 4 (full protection) with a new password and its web server was disabled; the generator PLC received identical protection settings along with altered fan timing parameters; and the water PLC received protection plus modified pump speed and timing values. The project was compressed, split into parts for transfer over the bandwidth-constrained P2P relay, and reassembled in a staging directory on the SCADA host. Since downloading the project to the PLCs requires the TIA Portal GUI, a `netsh interface portproxy` rule was configured to tunnel RDP traffic through the C2 agent chain, allowing the attacker to interact with TIA Portal on the SCADA host via a proxied remote desktop session and acting as another example of LOTL style of attacks. During cleanup, InfluxDB data on the historian was partially wiped using a REST API delete call with the same token used for exfiltration.

Taken together, the scenario spans the three paradigms identified in Section 3.1. Day 1's reconnaissance and lateral movement lean on living-off-the-land, the style most visible in recent deployments such as FrostyGoop and Fuxnet, while the Day 2 PLC tooling splits between the monolithic approach of Stuxnet and TRISIS and the modular approach of BlackEnergy and PIPEDREAM. The paradigms are a design space rather than a strict timeline, as Industroyer2's 2022 return to monolithic design illustrates, and treating them that way let each stage be built in whichever style suited it best.

3.5 Provenance Data Collection

Observing an attack chain that spans from an enterprise phishing email all the way down to a physical actuator requires telemetry at every Purdue layer. No single sensor provides this coverage, so the testbed deploys four complementary data sources: host event logging on every Windows VM, attack reports from the C2 framework, network capture at two sources, and physical process telemetry from the controllers. Since data collection collided with the UTC to BST time shift, it is worth nothing that all the collected data is in UTC. The remainder of this section describes each source and what layer of the attack it makes visible.

3.5.1 Host Telemetry (L4–L2): Sysmon. Every Windows VM in the testbed, from enterprise workstations through to SCADA and the historian, runs Microsoft Sysmon. Sysmon is a kernel-mode service that writes detailed process, network, file, registry, and image-load events into the Windows event log. The testbed uses a customised configuration derived from the SwiftOnSecurity community baseline [30], extended for ICS-aware monitoring. The extensions add network-connection includes for the entire OT subnet, S7Comm and Modbus ports, and Siemens engineering binaries (TIA Portal, WinCC, S7 services), so that every inbound or outbound connection touching the process network is recorded regardless of the initiating process. File-create rules are extended to flag writes to TIA Portal project files (`.ap16`–`.zap18`), PLC source files (`.awl`, `.scl`), and the Node-RED configuration directory. Process-access events targeting `lsass.exe` are retained with credential-access relevant `GrantedAccess` rights, and named-pipe monitoring is extended to match the CALDERA Sandcat and Manx agent pipe names and the PsExec service pipe. Noise from VMware Tools and multicast service discovery is explicitly suppressed so that the dataset reflects guest activity rather than virtualisation artefacts. One gap in this configuration is that the Sysmon `FileDelete` rule group (event ID 23) was not populated, so file deletions are not recorded in the host event log and must be inferred from other signals, for example the absence of files that were previously observed being written. Raw Sysmon events are exported from each host as EVTX files and preserved verbatim as the authoritative L4–L2 host record.

3.5.2 Attack Ground Truth: CALDERA Operation Reports. Each stage-based CALDERA adversary produces a JSON operation report recording every ability invocation: the agent it ran on, the exact command, the timestamp, and the captured stdout. Because the scenario is decomposed into small per-stage adversaries (Section 3.4), these reports pin every automated attack action to a known host and time window, providing the ground truth needed to label Sysmon events and network traffic against specific techniques.

3.5.3 Network Capture (L4–L0): Hypervisor and Mirror Port. Network traffic is captured at two complementary locations. Both hypervisor hosts run packet capture on their virtual switches, recording all inter-VM traffic within each zone, including flows that never leave the physical machine. The 24-port unmanaged switch interconnecting the physical PLCs, HMIs, and OT-zone uplink has a configured mirror port feeding a second capture, which is the only observation point for purely physical traffic such as HMI-to-PLC S7Comm exchanges. The two sources are intentionally overlapping rather than disjoint: traffic between OT VMs and physical PLCs appears in both, allowing cross-validation between the virtual and physical capture paths. All network data is stored as pcap. Process-variable telemetry comes from two sources at the controller layer. The Node-RED flow on the historian polls the PLCs over S7Comm and persists pump speed, flow rate, generator state, and safety-line status into InfluxDB, which is exported as CSV. Independently, each PLC maintains an on-device diagnostic buffer recording CPU mode transitions, hardware and I/O faults, and communication events. These buffers are dumped at the end of each day and provide a controller-side record of state changes such as remote stop commands and the runtime faults that follow a malicious download or block deletion.

4 Dataset and Evaluation

4.1 Dataset Composition

The collection campaign produced five days of telemetry, roughly 26 GB after trimming: one dry run to validate the data collection methods, two benign baseline days, and two attack days. Each day provides four primary data sources (network pcap, Sysmon EVTX, historian InfluxDB exports, and text dumps of the PLC on-device diagnostic buffers), supplemented on attack days by CALDERA operation reports. Capture windows sum to just over

Day	Dur.	pcap Size	pcap Pkts	Sysmon Size	Sysmon Evts	Hist. Size	CALDERA	PLC
Day 1 Dry Test	3h 00m	0.7 GB	5.4 M	45 MB	27,582	258 MB	N/A	–
Day 2 Benign	6h 45m	4.9 GB	17.4 M	127 MB	83,149	600 MB	N/A	8
Day 3 Benign	7h 00m	14.5 GB	28.2 M	179 MB	128,779	614 MB	N/A	4
Attack Day 1	4h 40m	2.9 GB	12.5 M	112 MB	75,406	421 MB	114	12
Attack Day 2	2h 15m	1.3 GB	5.4 M	44 MB	29,311	128 MB	95	54

Table 4: Composition of the collected dataset per capture day. Sizes are the on-disk sizes of the trimmed artefacts; counts are the records contained within them. M denotes millions. pcap Pkts is the total packet count summed across all capture points and therefore counts packets observed in more than one capture multiple times. CALDERA columns report the number of stage ground-truth links on the attack days, and PLC columns report entries retained from the on-device diagnostic buffer exports within the trim window. Day 1 Dry Test predates PLC-side diagnostic collection.

twenty-three hours of observed activity, and coverage is complete across the declared window for every source on every day, with one exception flagged below that is itself an attack signal rather than a collection artefact. Table 4 summarises the composition.

The pcap traces dominate on-disk size. The historian carries continuous process-variable telemetry from all three PLCs in the testbed: 22 tag series polled from the Water PLC (HydroControl), 22 from the Generator PLC (HydroGenerator), and 7 from the Flood PLC (HydroFlood), sampled by the Node-RED flow at approximately five samples per second per series across the full capture window. PLC diagnostic events are rare because the on-device buffer is a fixed-size ring that only records system-level occurrences such as CPU mode changes, security logons, faults, and configuration modifications; normal tag reads and writes do not populate it. The fifty-four entries retained for Attack Day 2 therefore represent fifty-four distinct controller-side state changes caused by the destructive phase, a density that the raw count understates. CALDERA reports are small but load-bearing on attack days: they are the authoritative ground-truth label for every automated attack action.

The benign days are deliberately longer than the attack days so that the baseline covers a full working-day rhythm. An important consequence of this, visible in the table, is that raw volume does not distinguish attack days from benign days. Day 3 Benign produces more pcap packets and Sysmon events than either attack day, because web browsing and document work generate volume at rates the attacks do not. This ordering is also partly a function of how actively simulated workers used the IT zone on a given day; a productive benign day is likely to outweigh a restrained attack day by raw volume, and a quiet benign day can fall below attack-day activity. The distinguishing signal is therefore not how much data a day produces but the presence of specific controller-side state changes. Attack Day 2 shows this most clearly with fifty-four PLC diagnostic entries retained against four to eight on benign days, reflecting the destructive phase's direct manipulation of the controllers; this motivates the structural validation approach taken in Section 4.2.

Application-layer protocol composition is similar across all OT-active days. TCP dominates by bytes and S7Comm is the signature industrial-layer protocol on every OT-active day. The captures are deliberately overlapping along predictable paths. Traffic between an OT VM and a physical PLC or HMI appears both on that VM's hypervisor tap and on the mirror port off the physical switch; traffic between two OT VMs on the same hypervisor host is forwarded directly by the virtual switch and appears only in the two VM taps; and purely physical exchanges such as PLC-to-PLC and HMI-to-PLC travel only within the physical switch and appear only in the mirror. The OT-zone router sees no intra-subnet traffic: the OPNsense captures contain zero S7Comm or COTP frames across all five days, so the router's observations are restricted to cross-zone flows, primarily RDP and TLS between the enterprise workstations, the jumpbox, and OT hosts. We have not performed a packet-level deduplication study across the capture points; the overlap description above is therefore derived from the network architecture and expected routing behaviour rather than from packet-level measurement.

Three dataset-level observations are worth calling out. First, the historian for Attack Day 2 stops populating the two S7-1200 measurement series at 15:12 UTC while the S7-300 series continues until 15:49 UTC, shortly before the operator cut physical power to the controllers. This asymmetry is the data-side fingerprint of the Level-4 password activation driven by the malicious TIA project download (T0862, T0843): client authentication against the two S7-1200s fails from that moment onwards, but the S7-300 remains readable because its password protection only covers code upload and download. The early stop is kept as-is in the historian export, not back-filled or padded to the window, because it is itself the data-side signature of the lockout. Second, the `GA_POWER` and `GB_POWER` fields declared in the InfluxDB schema are empty across every day because the Node-RED flow never writes to them; generator electrical output remains observable through the populated `GA_VLT_OUT` and `GB_VLT_OUT` voltage-out series, so no process-state information is lost. Third, the Day 1 Dry Test Sysmon record is anomalously sparse compared to the later days, this is due to a change in the Sysmon config used.

4.2 Provenance Validation

Validation of the dataset is carried out as two parallel checks. This section covers the first, which is structural: the pipeline must produce a graph in which the attack chain from initial access down to PLC state changes is captured as a single connected subgraph, and that subgraph must be traversable end-to-end using only the PROV-DM relations. The second check, in Section 4.3, runs a supervised graph-learning algorithm on the same graph to confirm that it is in a form a standard node classifier can consume. The two checks are independent. The structural check uses no classifier and the classifier is trained on labels derived from a source (the CALDERA operation reports) that the structural check never consults. The rest of this section describes how the graph is built, then walks through a hand-compiled reference subgraph that traces the attack across every Purdue layer that the data collection reaches.

4.2.1 Graph Construction and Enrichment. The graph is built from three of the four data sources described in Section 3.5: Sysmon EVTX from every Windows host, the network pcaps, and the historian CSV export. The CALDERA operation reports are not read during graph construction. They are kept as a separate ground truth overlay and used only at training time in the ML section, so the graph itself stays a record of the collected logs.

The base builder reads ten Sysmon event types and writes a PROV-DM document [5] with five node types and seven edge relations. The events and their mappings are:

- **Event 1 (Process create)**: activity node; populates `startTime`, image, command line, user and parent GUID.
- **Event 5 (Process terminate)**: `endTime` on the matching activity.
- **Event 11 (File create)**: file entity with a `wasGeneratedBy` edge to the writing activity.
- **Event 3 (Network connection)**: flow entity, `wasGeneratedBy` if outbound and `used` if inbound.
- **Events 8 and 10 (CreateRemoteThread, ProcessAccess)**: `wasInformedBy` edge between two activities on the same host.
- **Events 2, 12, 13, 15 and 22**: folded into attributes on the parent activity (file time change, registry writes, file stream writes, DNS queries).

Registry writes, DNS queries, file stream writes and a few smaller classes are folded into attributes rather than kept as separate entities. The result is a graph of five node types (processes, files, network flows, PLC variables and one agent hub per PLC) linked by the seven PROV-DM relations, which appear throughout the enrichment passes described next.

Five enrichment passes then join the per-host graphs into a single view. Table 5 lists them in the order they run. Together they add the image-to-activity link that captures which file on disk each process was launched from; cross-host binary identity by hash, which is how the replicated CALDERA agent binary on every host the attacker reaches gets tied together; a single flow entity per network session so that a connection observed by both endpoints is not double-counted; a fallback for WinRM and DCOM connections where only the sender or receiver logged the Sysmon event but the hypervisor or mirror pcaps confirmed the conversation; and finally the PLC variable nodes and per-PLC agent hub that expose physical process changes on the graph.

The PLC agent hub is the structural element that links the OT side of the graph to the physical process. Every host side flow whose destination is a PLC, and every `plcvar` produced from that PLC's historian tags, points at the same hub. No direct edge is drawn from a host process to a `plcvar` so a host process and the physical state change it caused are connected through the hub in two steps rather than one. While there are ways to connect this specific graph better because S7Comm has `function codes` which allow us to inspect network packets and determine the purpose of the connection there is no single rule reliable across S7Comm(Plus), Modbus and other similar protocols used by other PLCs. After enrichment, Day 1 contains 62,895 nodes and 80,825 edges. Day 2 contains 24,385 nodes and 36,524 edges.

4.2.2 Structural Traceability. Structural coherence is checked by hand-compiling a reference subgraph for each attack day that traces the attack chain across every layer it reaches, and confirming that every node in the subgraph is reachable from every other using only PROV-DM relations on the enriched graph. Figure 4 shows the Day 1 subgraph. It contains thirty-four nodes across seven clusters: the C2 flows to and from the attacker server, the phishing macro shell on ENTWS1 and the PowerShell download that drops the first agent binary, the privilege escalation chain that runs fodhelper for a UAC bypass and launches the elevated agent, the PSRemoting step that pivots onto the Jumpbox, the Jumpbox receiver chain where `wsmprovhost.exe` and `WmiPrvSE.exe` drop and execute the replicated agent binary, followed by a scheduled task that persists the agent across sessions, the second PSRemoting step onto the historian and the same drop and execute pattern there, and finally the PowerShell loop that invokes `python plc_cpu_info.py` against the PLC subnet, with the resulting S7Comm flow on port 102 closing the chain at L1. Every edge in the figure is one of the seven PROV-DM relations emitted by the pipeline so no edge is synthesised by hand for the figure. The ability to draw this chain from the full graph using only provenance edges is evidence that the pipeline captures the Day 1 attack end to end.

Pass	What it adds
Image bridge	`used` edge from every activity to the file it was launched from, matched by image path on the same host.
Hash	SHA-256, MD5 and imphash attributes on every process and file; `wasDerivedFrom` edge between identical files on different hosts, which is how the CALDERA agent binary reuse across the Purdue stack appears in the graph.
Cross-host flow	Matches Sysmon event 3 on the sender host against the matching event 3 on the receiver host by 5-tuple and timestamp, merges them into one flow entity, and adds a `wasInformedBy` edge from the receiver activity to the sender activity.
Pcap bridge	Handles the case where only one host logged the event: links the flow to its matching conversation in the pcap and, for WinRM (ports 5985 and 5986) and DCOM (port 135), bridges the sender to the nearest handler process on the destination host (`wsmprovhost.exe` or `WmiPrvSE.exe`) inside a five second window.
Historian	Adds a `plcvar` entity for each change on an attack-related InfluxDB tag, debounced to half a second to collapse rapid toggles, and a per-PLC agent hub linked by `wasAttributedTo` to every `plcvar` from that PLC and to every flow whose destination is that PLC.

Table 5: The five enrichment passes in run order. Each is additive; together they stitch the per-host graphs into a single cross-layer view.

The Day 2 reference subgraph is shown in Figure 5 and traces the staged denial-of-service and block-delete attack against the S7-300 flood controller. It picks up where the Day 1 chain ends, at the attacker's surviving Jumpbox agent, and runs down to the `MasterFlood` PLC variable changes that the historian captures during the destruction sequence. A PowerShell `PSSession` from the Jumpbox agent uses the engineering credentials harvested on Day 1 to copy `SecurityService.exe` onto ENGws1 over WinRM, `wsmprovhost.exe` on ENGws1 receives the connection and the new CALDERA agent is dropped into `C:\Windows\Temp` and executed. That agent launches a PowerShell one-liner which

creates three flag files, one each for the safety-bypass, brute-force and block-delete phases, and starts `plc_dos_bus.py`. Then the orchestrator script in turn spawns `plc_dos.py` against the flood controller at 192.168.3.9. The two Python processes hold long-running S7Comm sessions on port 102, each logging over thirty minutes of traffic to the PLC. Three later PowerShell invocations flip the flag files from `false` to `true` at staged intervals, escalating the attack through its successive phases, and each phase is followed by `MasterFlood` value changes on the flood controller that the historian captures as `plcvar` entities attributed to the PLC agent hub. As with Day 1, every edge in the subgraph is one of the PROV-DM relations produced by the pipeline.

4.3 Machine Learning Potential

The structural check in Section 4.2 shows that the graph captures the attack chain as a connected subgraph spanning every layer of the collection. The complementary question is whether the same graph is in a form that a supervised graph classifier can consume, and whether that classifier reaches meaningful precision and recall on test nodes. Established pipelines from the provenance intrusion detection literature, including Flash [34], Kairos [6], ThreaTrace [35] and the survey by Zipperle et al. [36], target enterprise side with thousands of benign baseline traces and do not transfer directly to an ICS dataset of this size, however they are all models trained to work with graph data and build from provenance basics. So we run a small supervised GraphSAGE pilot/test rather than reproducing a detection benchmark.

The enriched PROV-JSON is loaded into a PyTorch Geometric graph. Edges keep their PROV semantics but edge type identifiers are not fed to the model. Each node carries a 69-dimensional feature vector made of two parts. The first five dimensions are a one-hot indicator of the node type (process, file, flow, PLC variable or PLC agent). The remaining 64 dimensions are a hashed embedding of the node's most identifying text attribute: the command line for processes, the path for files, the destination IP and port for flows, and the field name for PLC variables. The attribute string is split on whitespace into tokens, each token is hashed into one of 64 buckets using scikit-learn's `FeatureHasher`, and the counts form the second part of the vector. The same technique is used in Kairos to avoid vocabulary blow-up on unseen command lines [6]; we use the flat rather than hierarchical variant.

Labels come from the CALDERA operation reports. Every link in a report records the host, plaintext command and decision timestamp. Links are matched to process nodes inside a window of (−15 s, +30 s) around the decide time, first by executable stem, then by a shell intent hint (PowerShell, cmd, python), and as a last resort by a word overlap check between the CALDERA command and the Sysmon command line, accepting a match when more than half of the distinct words are shared. The timing window allows us to account for clock sync issues and P2P chain latency. Then matched processes become seeds, and the label is propagated forward by one edge along `wasStartedBy` and `wasGeneratedBy` to child processes and to generated flows; generated files are propagated only when the extension is an executable or archive. Service discovery flows (mDNS, SSDP, LLMNR, NetBIOS, and multicast and link local addresses) and PowerShell script policy temporary files are filtered out to stop incidental sweep-in. PLC-variable and PLC-agent nodes are never labelled positive because the enrichment pipeline deliberately does not draw a `wasGeneratedBy` edge from a host process to a `plcvar` (Section 4.2.1), so forward propagation never reaches them and neither type appears in the per-class F1 table below. The positive set contains 546 nodes on Day 1 (0.87% of the graph) and 1,113 on Day 2 (4.57%). Day 2 has a higher propagation ratio because three of its major stages (the engineer's manual RDP walkthrough, the shadow controller loop and the Python denial of service stage) run without CALDERA orchestration and therefore do not seed directly; the positive label instead arrives through forward propagation from whatever orchestrated ancestor sits above them in the process tree, or is missed entirely where no such ancestor exists.

The model is a two layer GraphSAGE classifier. Each message passing layer aggregates information from a node's immediate neighbours, so two layers give every node access to its one hop and two hop neighbourhood in the graph; this depth matches the one forward edge used during label propagation and is also what ThreaTrace settles on for a similar task [35]. Nodes are split 70/30 stratified by label. The loss is binary cross entropy weighted so that the minority positive class is not drowned out at a sub one percent positive rate. Training runs on CPU and completes in under a minute on both days. Results are reported on the held-out 30% test split at two thresholds: the default 0.5, and a threshold chosen to maximise F1 on the training split (both days land at 0.9).

	Day 1	Day 2
Nodes / Edges	62,895 / 80,825	24,385 / 36,524
Positive rate	0.87%	4.57%
All-zeros F1	0.000	0.000
All-ones F1	0.017	0.087
Model F1 at threshold 0.5	0.783	0.974
Model at 0.9, precision	0.739	0.960
Model at 0.9, recall	0.982	0.997
Model at 0.9, F1	0.843	0.978
Per-class F1, `proc`	0.831	0.868
Per-class F1, `flow`	0.960	0.998

Table 6: Precision, recall and F1 on the 70/30 test split for each day, alongside the two constants (predicting all nodes benign, which gives F1 = 0, and predicting all nodes malicious, which gives F1 = $2p/(1+p)$ for positive rate p). Per-class F1 is reported at threshold 0.9.

		Day 1		Day 2	
		Pred. ben.	Pred. mal.	Pred. ben.	Pred. mal.
thr = 0.5	benign	18,617	88	6,965	17
	malicious	2	162	1	333
thr = 0.9	benign	18,648	57	6,968	14
	malicious	3	161	1	333

Table 7: Confusion matrices on the 70/30 test split at the default (0.5) and tuned (0.9) thresholds. Rows are ground truth, columns are model predictions.

Table 6 reports performance alongside the two constant baselines. Both days clear the baseline(constant predictions) F1 comfortably and reach high recall at meaningful precision. Table 7 breaks the errors into their raw components at both operating points. Missed attack nodes stay in the low single digits on both days (two or three on Day 1, one on Day 2), and the precision

differences are driven by a small pool of false positives rather than by the model overlooking the attack. Moving from the 0.5 threshold to 0.9 trades about thirty false positives for one extra missed node per day. This is the indicitive of a calibrated model whose default cutoff is already close to optimal. Day 2 is the easier of the two in absolute terms because the capture window is shorter, so the graph is smaller overall (24k nodes against 63k) and the positive rate is higher (4.57% against 0.87%). The Day 2 flow F1 of 0.998 should, however, be read with care. The flow class is dominated by one attack pattern, with 934 of the 954 positive flows being S7Comm sessions from ENGws1 to the PLCs on port 102, most of them from the Python denial-of-service stage. Each `snap7` call opens its own TCP socket, so one attacker process pulls hundreds of near-identical flow nodes into the positive set through `wasGeneratedBy`, and the model's main job on Day 2 is to pick up this fan-out rather than to generalise across flow types. The `proc` F1 (0.87 on Day 2, 0.83 on Day 1) is the more informative figure. Taking the induced subgraph over the predicted positive nodes at threshold 0.5, Day 1 breaks into 40 connected components, the largest two with 546 and 238 nodes, and the remaining 38 are small (two nodes or fewer). Day 2 breaks into 59 components, the top five sized 497, 437, 39, 39 and 22. This suggest that there is a subgraph that can potentially be discovered better by a better model.

Two further caveats bound these results. First, labels come only from CALDERA links. Manual attack steps, for example the engineer's RDP walkthrough that precedes the TIA project upload on Day 2, have no operation report and are missing from the seed set. They appear in the graph as unlabelled negatives, which lowers precision even though the activity is malicious. This gap is documented with the published dataset so that downstream users can add those labels by hand. Second, the evaluation is within each day rather than across days. However with more work done on label engineering and a different methodology for the graphing, which could allow 2 days to be linked, a version of the graph that can be used to train a more comprehensive model can be achieved.

4.4 Summary of Validation Results

Together the two checks validate that the dataset supports the graph-based analysis it was built for. The reference subgraphs in Section 4.2.2 show the attack chain is recoverable from the full graph using only PROV-DM relations, end-to-end on Day 1 and over the destruction sub-chain on Day 2. On the machine-learning side, a supervised GraphSAGE model trained on labels derived from the CALDERA operation reports clears both constant-prediction baselines comfortably on both days, confirming that the graph is usable with standard provenance-ML approaches. The CALDERA reports also give future users a ready-made source of labels that can be reused, extended, or replaced when they build their own graph from the released telemetry, which should make the dataset straightforward to build on.

5 Limitations and Future Work

5.1 Host Telemetry Limitations

The most consequential limitation sits at the intersection of the Sysmon configuration used and the attack design that CALDERA constrained us to. The Day 2 payloads are autonomous host-resident scripts that run once launched and are driven by flag files written into `C:\Windows\Temp`, a pattern adopted because CALDERA does not support interactive operator sessions and a fully self-contained payload with the fine-grained control the scenario needed would have been prohibitively expensive to build. Our Sysmon ruleset does not match the `.flag` extension in its `FileCreate` rules and does not enable the `FileDelete` rule group at all, so neither the creation of the flag files that coordinate phase transitions nor the cleanup-phase deletions at the end of Day 2 appear in the host log. Both gaps are configuration choices rather than capability limits, and a better Sysmon configuration would meaningfully improve the dataset on its next iteration.

Even with those rules populated, Sysmon would not make the payload itself fully visible. Once a payload has opened its persistent S7Comm socket to the PLC, every application-layer action thereafter, including every S7Comm request, DB read, and DB write, travels inside that existing connection, and event 3 fires only on socket creation. The same model gap applies to the flag files themselves: when a payload reads a flag to pick up a new phase parameter or reloads a configuration value, Sysmon logs nothing, because it has no file-read event type, so the exact moment of each behaviour change is not recoverable from the host log alone. The Day 2 shadow controller ran for roughly forty minutes of continuous PLC manipulation yet produced a single process node and two outbound flow nodes on the graph; the per-packet S7Comm traffic it generated is still present in the pcap, though the flag-file reads that drive its phase changes have no host-side timestamp at all. Recovering that activity would require either closer integration of the pcap with the host graph or Windows Security Log file-system auditing on the flag-file paths. An attack mechanism built around shorter-lived, repeated processes rather than one long-running one would have been more naturally captured by Sysmon, but would have sacrificed the continuous-loop control the payloads needed for the scenario.

5.2 Dataset Caveats

A second structural limitation is that the Day 1 and Day 2 graphs are treated as separate inputs in both the structural and machine-learning checks. The VMs are snapshot-restored between the two attack days, which re-randomises process GUIDs, P2P listener ports, and the agent binaries dropped at runtime. The Day 2 attacker is therefore only connected to the Day 1 attacker through on-disk artefacts such as the scheduled tasks and registry keys left in place on Day 1 and the replicated Sandcat binary identified by its SHA-256 hash. A related break appears within Day 1: the scheduled-task persistence on the jumpbox launches its children under `taskeng.exe` rather than `schtasks.exe`, so the host-event record contains no parent-child edge to the post-restart agent despite the tasks themselves being linked by name and command line. Extending the enrichment pipeline to bridge these handoffs via non-GUID anchors would yield a single connected graph across the full campaign and is the most useful improvement for downstream ML.

Three smaller dataset caveats follow. First, the Day 1 Dry Test should be read as a test day rather than a clean baseline; it predates the finalised Sysmon configuration and lost coverage to host sleep on SCADA and the two engineering workstations. It is retained in the release so downstream users can see the collection pipeline evolving, but is not directly comparable to the other four days. Second, the ENG-WS2 pcap on Day 2 Benign has a gap from 16:36 to 17:05 UTC produced by a known RDP-related lockup that required a restart of the VM from the hypervisor panel, which in turn invalidated the virtual switch port and silently dropped the ENG-WS2 hypervisor capture until it was restarted by hand.

Sysmon on ENG-WS2 came back up with the VM and recorded the window in full, and the hypervisor captures on adjacent machines together with the router and jumpbox taps carry all cross-host flows involving ENG-WS2 during the gap. The only traffic that cannot be reconstructed is intra-OT communication that terminates at ENG-WS2 without reaching another captured tap, which on this benign day, with no domain membership and no attack activity, amounts to little beyond ARP chatter. Third, the class distribution on Day 2 is shaped by the reverse of the effect described in the opening paragraphs. The denial-of-service stage opens a fresh TCP socket for each `snap7` call, so one long-running attacker intent turns into hundreds of near-identical flow nodes on ENGws1 that all propagate the attack label through `wasGeneratedBy`. The Day 2 positive rate of 4.57% (against 0.87% on Day 1) and the `flow` per-class F1 of 0.998 are therefore both inflated by this fan-out. The `proc` per-class F1 is a more stable indicator and is comparable on the two days (0.868 on Day 2, 0.831 on Day 1), suggesting that the process-level signal itself is of similar difficulty on both. Day 1 is the more conservative headline figure for what a minimal GraphSAGE baseline can do on this graph.

5.3 Scope and Future Work

Finally, the scope of the testbed bounds what can be claimed from these results. All three controllers are Siemens S7 units speaking S7Comm on TCP/102, and the physical process is a single scaled hydro-electric plant; findings about which protocol messages generate which host-level artefacts do not transfer directly to Rockwell, Schneider, or Omron stacks, nor to power-grid, chemical, or oil-and-gas processes with different safety-interlock topologies. A related point applies at the TTP level and is in truth a general limitation of any TTP-labelled dataset rather than one specific to this work. The MITRE ATT&CK technique classification is implementation-agnostic, but a dataset necessarily captures one concrete procedure per technique: host discovery in this collection is a Python script chained to built-in Windows commands, whereas the same technique can equally be implemented through pure living-off-the-land tooling, a custom PowerShell module, or an off-the-shelf offensive framework, and the host-level artefacts of each look quite different. A detector trained on the procedures executed here will recognise those procedures more reliably than it will generalise across every way of performing the same technique. The attack campaign is also limited to two days, which excludes the multi-week to multi-year dwell characteristic of real APT activity; features that only manifest at that timescale, such as adversary adaptation in response to defender discoveries, are absent from the dataset by construction.

On top of the future work mentioned in this section other considerable additions could be standardising the graph structure. With more emphasis on the ICS data the graph can be better integrated for example; utilising the PLC logs in graph building, by extending the PLC agent nodes with this information or in labelling to help pin point the malicious project download to the PLC. Moreover, the model proposed in this is very basic, and it can be carried forward greatly. Different approaches in labelling and encoding the data like changing the propagation behaviour or including temporal and/or spatial information better will yield better and more accurate results.

6 Conclusions

Drawing on an analysis of nine major ICS attacks spanning 2010 to 2024, we built a Purdue-segmented testbed with physical Siemens PLCs controlling a real hydroelectric process and emulated a two-day APT campaign across it, automated through CALDERA 5.3.0 with per-stage operation reports as ground truth. The scenario spans an enterprise-side phishing stage, lateral movement into the operational network, and Day 2 PLC manipulation against three physical controllers, exercising 53 of the 69 MITRE ATT&CK for ICS techniques seen across the nine attacks.

Logs were collected from every Purdue layer at once: Sysmon on every Windows VM, packet capture at both hypervisors and at the mirror port of the physical PLC switch, InfluxDB process data from the historian, and on-device diagnostic buffers from each controller. We hand-traced an example attack chain across each day to show that a cross-layer path can be recovered from the graph using only PROV-DM relations, and a basic GraphSAGE pilot trained on CALDERA-derived labels comfortably beat the constant-prediction baselines on both days.

The value of the dataset is not in any single layer but in the combination. The physical controllers produce real S7Comm traffic and real controller-side state changes that software emulation cannot easily reproduce, and the CALDERA reports provide a ready-made source of labels, tying host events back to specific attack steps without needing to be labelled by hand. That combination is what neither existing IT-only provenance datasets nor sensor-only ICS datasets currently provide.

The dataset is released alongside the raw CALDERA operation reports and the on-device PLC logs, so future work can rebuild the graph from the underlying sources rather than inherit the choices made here.

Acknowledgments

I would like to thank my supervisor Dr. Marco Cook for his guidance, support and excellent topic selection. He and this topic has taught me so many new things that I find genuinely interesting. I also would like to thanks Deniz for listening to me explain provenance and why it's important, how to do privilege escelation and various other things she has no interest in.

References

[1] Carolyn Ahlers and Kyle O'Meara. 2024. Impact of FrostyGoop ICS Malware on Connected OT Systems. https://hub.dragos.com/hubfs/Reports/Dragos-FrostyGoop-ICS-Malware-Intel-Brief-0724_.pdf

[2] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur. 2017. WADI: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks* (Pittsburgh, Pennsylvania) *(CySWATER '17)*. Association for Computing Machinery, New York, NY, USA, 25–28. doi:10.1145/3055366.3055375

[3] Md Raihan Ahmed, Jainta Paul, Levi Taiji Li, Luis Garcia, and Mu Zhang. 2025. ICSTracker: Backtracking Intrusions in Modern Industrial Control Systems. In *2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (2025-06). 193–207. doi:10.1109/DSN64029.2025.00031 ISSN: 2158-3927.

[4] Asher Davila and Chris Navarrete. 2024. *FrostyGoop's Zoom-In: A Closer Look into the Malware Artifacts, Behaviors and Network Communications.* https://unit42.paloaltonetworks.com/frostygoop-malware-analysis/

[5] Khalid Belhajjame, Reza B'Far, James Cheney, Sam Coppens, Stephen Cresswell, Yolanda Gil, Paul Groth, Graham Klyne, Timothy Lebo, Jim McCusker, Simon Miles, James Myers, Satya Sahoo, and Curt Tilmes. 2013. *PROV-DM: The PROV Data Model.* W3C Recommendation. World Wide Web Consortium. https://www.w3.org/TR/prov-dm/

[6] Zijun Cheng, Qiujian Lv, Jinyuan Liang, Yan Wang, Degang Sun, Thomas Pasquier, and Xueyuan Han. 2024. Kairos: Practical Intrusion Detection and Investigation using Whole-system Provenance. In *2024 IEEE Symposium on Security and Privacy (SP)* (2024-05). arXiv:2308.05034 [cs.CR] doi:10.1109/SP54263.2024.00005

[7] Anton Cherepanov. 2017. A new threat for industrial control systems(ESET INDUSTROYER). (2017). https://web-assets.esetstatic.com/wls/2017/06/Win32_Industroyer.pdf

[8] Mauro Conti, Denis Donadel, and Federico Turrin. 2021. A Survey on Industrial Control System Testbeds and Datasets for Security Research. 23, 4 (2021), 2248–2294. doi:10.1109/COMST.2021.3094360

[9] Feng Dong, Shaofei Li, Peng Jiang, Ding Li, Haoyu Wang, Liangyi Huang, Xusheng Xiao, Jiedong Chen, Xiapu Luo, Yao Guo, and Xiangqun Chen. 2023. Are we there yet? An Industrial Viewpoint on Provenance-based Endpoint Detection and Response Tools. arXiv:2307.08349 [cs] doi:10.48550/arXiv.2307.08349

[10] Dragos Inc. 2017. Dragos Crashoverride Analyzing the Threat to Electric Grid Operations. https://nsarchive.gwu.edu/sites/default/files/documents/3869008/Dragos-CRASHOVERRIDE-Analyzing-the-Threat-to.pdf

[11] Dragos INC. 2017. TRISIS Analysis of Safety System Targetted Malware. https://infocon.org/mirrors/vx%20underground%20-%202025%20June/Papers/ICS%20SCADA/Triton/2017-12-14%20-%20TRISIS%20Malware%20-%20Analysis%20of%20Safety%20System%20Targeted%20Malware.pdf

[12] Nicolas Falliere, Liam O Murchu, and Eric Chien. 2011. Synmantec Stuxnet. http://large.stanford.edu/courses/2011/ph241/grayson2/docs/w32_stuxnet_dossier.pdf

[13] Erfan Ghiasvand, Suprio Ray, Shahrear Iqbal, Sajjad Dadkhah, and Ali A. Ghorbani. 2024. CICAPT-IIOT: A provenance-based APT attack dataset for IIoT environment. arXiv:2407.11278 [cs] doi:10.48550/arXiv.2407.11278

[14] Glasgow Cyber Defence Lab. [n. d.]. GCDL Testbeds. https://www.gla.ac.uk/research/az/cyber-defence-lab/gcdltestbeds/ University of Glasgow, Cyber Defence Lab.

[15] NCC GROUP. 2022. Lazarus. https://www.nccgroup.com/the-lazarus-group-north-korean-scourge-for-plus10-years/

[16] Dragos Inc. 2022. *Pipedream: Chernovite's Emerging Malware Targetting Industrial Control Sustems.* Technical Report. Dragos. https://hub.dragos.com/hubfs/116-Whitepapers/Dragos_ChernoviteWP_v2b.pdf?hsLang=en

[17] Dragos Inc. 2025. Understanding ICS Malware: Defining a Credible Threat to Industrial Infrastructure. (2025). https://dragos.brightspotcdn.com/96/80/40971d1647ffa71269fc32b0dd79/understanding-ics-malware-whitepaper-06-25.pdf

[18] Yuning Jiang, Qiaoran Meng, Feiyang Shang, Nay Oo, Le Thi Hong Minh, Hoon Wei Lim, and Biplab Sikdar. 2025. MITRE ATTCK Applications in Cybersecurity and The Way Forward. arXiv:2502.10825 [cs.CR] https://arxiv.org/abs/2502.10825

[19] Rajesh Kumar, Rohan Kela, Siddhant Singh, and Rolando Trujillo-Rasua. 2022. APT attacks on industrial control systems: A tale of three incidents. 37 (2022), 100521. doi:10.1016/j.ijcip.2022.100521

[20] Milica Lekić and Gordana Gardašević. 2020. Supervisory Control and Data Acquisition Approach in Node-RED: Application and Discussions. *IoT* 1, 1 (2020), 56–78. doi:10.3390/iot1010005

[21] Bryce Livingston and Sam Hanson. 2024. Strategic Overview of the Fuxnet Malware. (2024). https://hub.dragos.com/hubfs/Reports/Dragos_SB_Intel_Fuxnet_ICSMalware.pdf?hsLang=en

[22] MANDIANT. 2022. *INCONTROLLER: New State-Sponsored Cyber Attack Tools Target Multiple Industrial Control Systems | Mandiant | Google Cloud Blog.* https://cloud.google.com/blog/topics/threat-intelligence/incontroller-state-sponsored-ics-tool/

[23] Thomas Miller, Alexander Staves, Sam Maesschalck, Miriam Sturdee, and Benjamin Green. 2021. Looking back to look forward: Lessons learnt from cyber-attacks on Industrial Control Systems. *International Journal of Critical Infrastructure Protection* 35 (Dec. 2021), 100464. doi:10.1016/j.ijcip.2021.100464

[24] Mitre Groups [n. d.]. https://attack.mitre.org/groups/ Groups that are tracked by Mitre and the security community with their alternative names.

[25] Mitre Main Page [n. d.]. https://attack.mitre.org/ Main page of mitre attack, talking about realworld threats..

[26] Julian Rrushi, Hassan Farhangi, Clay Howey, Kelly Carmichael, and Joey Dabell. 2015. A Quantitative Evaluation of the Target Selection of Havex ICS Malware Plugin. (2015). https://api.semanticscholar.org/CorpusID:36706910

[27] Luis Salazar, Sebastián R. Castro, Juan Lozano, Keerthi Koneru, Emmanuele Zambon, Bing Huang, Ross Baldick, Marina Krotofil, Alonso Rojas, and Alvaro A. Cardenas. 2024. A Tale of Two Industroyers: It was the Season of Darkness. In *2024 IEEE Symposium on Security and Privacy (SP)* (2024-05). 312–330. doi:10.1109/SP54263.2024.00162 ISSN: 2375-1207.

[28] SANS and E-ISAC. 2016. Sans and E-ISAC BlackEnergy Case. https://nsarchive.gwu.edu/sites/default/files/documents/3891751/SANS-and-Electricity-Information-Sharing-and.pdf

[29] Hyeok-Ki Shin, Woomyo Lee, Jeong-Han Yun, and HyoungChun Kim. 2020. HAI 1.0: HIL-based Augmented ICS Security Dataset. arXiv:2005.10676 [cs.CR] doi:10.48550/arXiv.2005.10676

[30] SwiftOnSecurity. 2021. *sysmon-config: A Sysmon configuration focused on default high-quality event tracing.* https://github.com/SwiftOnSecurity/sysmon-config

[31] Symantec. 2014. *Dragonfly: Cyberespionage Attacks Against Energy Suppliers.* Technical Report. Symantec. https://docs.broadcom.com/doc/dragonfly_threat_against_western_energy_suppliers

[32] Team82. 2024. *Unpacking the Blackjack Group's Fuxnet Malware.* https://claroty.com/team82/research/unpacking-the-blackjack-groups-fuxnet-malware

[33] Giannis Tsaraias and Ivan Speziale. 2022. *Industroyer vs. Industroyer2: Evolution of the IEC 104 Component.* Technical Report. Nozomi Networks. https://www.aisp.sg/document/publication/Nozomi-Networks-WP-Industroyer2-A4.pdf

[34] Mati Ur Rehman, Hadi Ahmadi, and Wajih Ul Hassan. 2024. FLASH: A Comprehensive Approach to Intrusion Detection via Provenance Graph Representation Learning. In *2024 IEEE Symposium on Security and Privacy (SP)* (2024-05). 3552–3565. doi:10.1109/SP54263.2024.00139

[35] Su Wang, Zhiliang Wang, Tao Zhou, Hongbin Sun, Xia Yin, Dongqi Han, Han Zhang, Xingang Shi, and Jiahai Yang. 2022. threaTrace: Detecting and Tracing Host-based Threats in Node Level Through Provenance Graph Learning. 17 (2022), 3972–3987. arXiv:2111.04333 [cs.CR] doi:10.1109/TIFS.2022.3208815

[36] Michael Zipperle, Florian Gottwalt, Elizabeth Chang, and Tharam Dillon. 2023. Provenance-based Intrusion Detection Systems: A Survey. 55, 7 (2023), 1–36. doi:10.1145/3539605

A Cross-Layer Attack Subgraphs

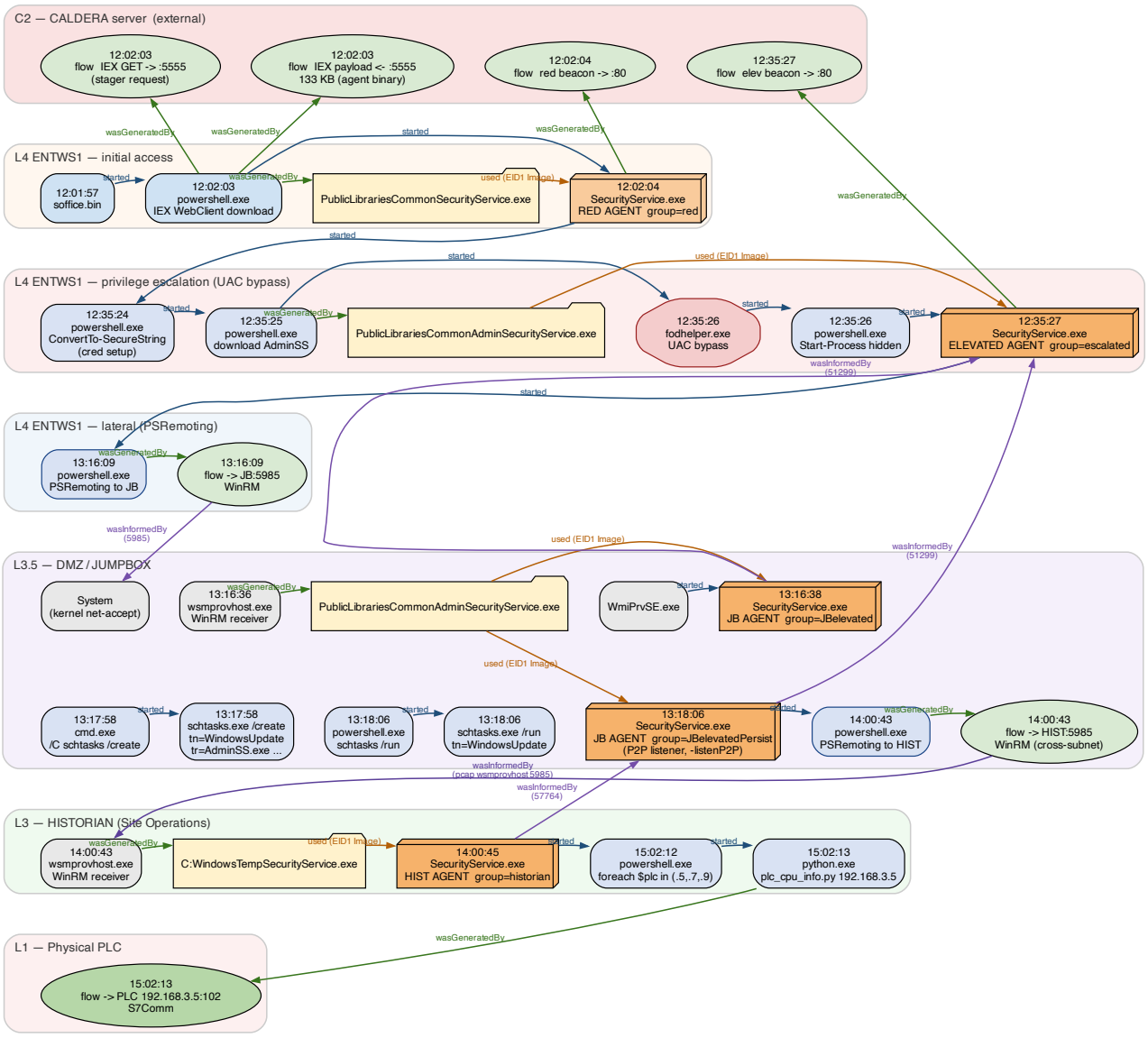

Figure 4: Cross-layer attack subgraph traced from Attack Day 1, showing initial entry at ENTWS1 propagation of the attacker laterally and theft of physical process information.

Dataset	Year	IT	OT	PLC	Purdue	Duration	Provenance	ATT&CK Coverage	Testbed Type
DARPA OpTC	2020	✓	×	×	L5 Only	14 Days	✓(TCCDM)	~28	Virtual
SWaT	2016	×	✓	✓	L2-L0	11 Days	×	~5[b]	Physical
WADI	2017	×	✓	✓	L2-L0	16 Days	×	~6[b]	Physical
HAI	2021	×	✓	✓	L2-L0	4 Days	×	~5[b]	Hybrid
CICAPT-IIoT	2024	✓	✓	Partial[a]	L3-L1	2 Phases	✓(CSV)	21	Hybrid
Our Work	**2026**	✓	✓	✓	**L4-L0**	**4 Days**	✓(Raw)	**53**	**Physical**

Table 8: Comparison of ICS security datasets showing architectural and adversarial coverage.
[a]CICAPT-IIoT uses Raspberry Pis with OpenPLC to simulate PLCs yet has physical sensors. Our work is to be released in raw format and used as PROV-DM for the demo.
[b]The original SWaT, WADI and HAI papers do not provide MITRE ATT&CK mappings; counts shown are derived from the documented sensor/actuator manipulation attacks.

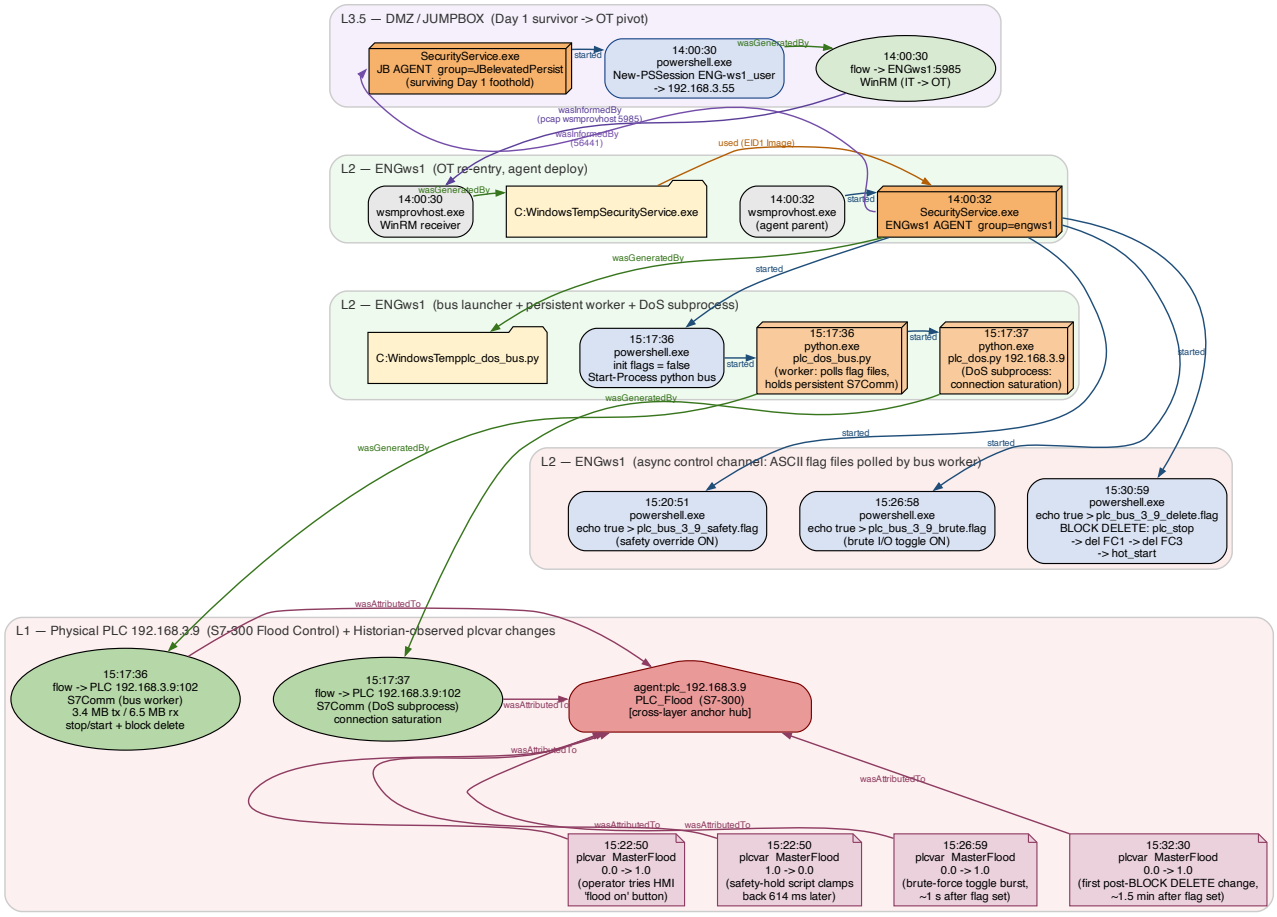

Figure 5: Cross-layer attack subgraph traced from Attack Day 2, showing OT re-entry through the Day 1 Jumpbox foothold and the block-delete destruction sequence against the S7-300 flood controller.