ABSTRACT

SINGH, GURKIRAT. Adaptive Defensive Deception in IT/OT Networks. (Under the direction of Dr. Munindar P. Singh).

Traditionally, Operational Technology (OT) has been isolated from the IT networks. This has now changed due to the improvement in business and engineering operations that IT/OT convergence offers. However, the convergence comes at the cost of increased vulnerability, as it extends the attack surface to include the potential sophisticated cyber-physical attacks. Furthermore, classical cyber defense approaches are ineffective against complex threats targeting physical processes. There is a need for interdisciplinary security approaches to provide cyber-physical security. This paper presents a modular adaptive defensive deception framework that adds a layer of security to OT networks. The framework takes advantage of the characteristics of the OT network and uses risk analysis to manipulate the number of defensive honeypot resources. We evaluated the framework in a lab setting with test cases specific to our threat model to demonstrate its effectiveness.

Adaptive Defensive Deception in IT/OT Networks

by
Gurkirat Singh

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh, North Carolina
2022

APPROVED BY:

Dr. Ruozhou Yu	Dr. William Enck
Dr. Samudra Vijay External Member	Dr. Munindar P. Singh Chair of Advisory Committee

BIOGRAPHY

The author received his B.Tech. in Computer Science and Engineering from Manipal University in India, in 2016. After working in the networking industry with Citrix Systems for 5 years as a Software Developer, he started his Master's degree in Computer Science from North Carolina State University. During the course of his graduate degree, he worked as a Research Assistant under the guidance of Dr. Munindar P. Singh, for an year, working on developing a framework for enhancing security in the OT networks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

INTRODUCTION

Operational technology (OT) refers to programmable systems that manage devices interacting with the physical environment. OT is a general term that encompasses several types of control systems, such as Industrial Control Systems, fire control systems, and physical access control mechanisms. OT components like programmable logic controllers (PLCs), human-machine interfaces (HMIs), supervisory control and data acquisition (SCADA) servers, and industrial networks were designed to be used in isolation, resulting in silos of specialized devices. However, advances in the technologies in the IT industry, such as communication protocols, storage, and analytics, have motivated organizations to break down these silos to take advantage of the improvement in business and engineering processes. For example, the traditional industrial network communication protocols such as Common Industrial Protocol (CIP), Modbus, and Profibus are now modified to work with standard IT network protocols like TCP/IP. New protocols like Ethernet/IP, Modbus TCP, and Profinet enable remote monitoring and control of underlying processes. Proprietary software and operating systems are also replaced by commodity software and operating systems. The IT/OT convergence has dramatically benefitted the OT industry and has led to cyber-physical systems becoming increasingly common in the field.

However, while convergence offers many advantages, these benefits come at the cost of security. Specific characteristics of OT systems make them vulnerable to an extensive array of attacks. Industrial protocols fail to provide authentication or encryption as most do not support SSL or TLS (Jingran et al. 2020). Also, OT systems have to be highly available, and therefore patching a vulnerability does not get priority over maintaining uptime in most cases. OT systems might be in place decades longer than typical IT systems, hence making them more vulnerable over time. Moreover, third-party vendors often have access to the monitoring systems of an organization in order to streamline engineering processes. The traditional cyber security measures, such as Intrusion Detection Systems (Morris et al. 2012) (Paridari et al. 2018) and Firewalls, often fail to stop sophisticated cyber attacks such as APTs (Chen et al. 2014). About 90% of organizations reported at least one OT system intrusion in 2020 (Fortinet 2020).

Other characteristics of OT networks are not taken advantage of in traditional cyber security measures but can be very effective. For example, OT infrastructure exhibits comparatively simple network dynamics. Since the devices have a specific work process, they use a fixed topology. The user population is also stable. They also use only a limited number of protocols with regular communication patterns.

In this work, we develop a modular adaptive defensive deception framework that takes advantage of OT network characteristics to detect, stall, and contain a threat. The primary goal of the framework is to provide business continuity and minimize downtime. Three components make up the framework: risk management, honeypot management, and isolation management. The architecture can be used along with already existing defense mechanisms, adding another layer to the defense. In addition, the modular design gives the ability to enhance functionality by adding more components and modifying existing components.

We evaluate the technique in a lab setting against a test suite consisting of test cases ranging from denial-of-service attacks to insider attacks. We weigh the benefits of the technique over other techniques used in the defense of OT networks.

This paper makes the following contributions:

1. Presents a framework to detect cyber-physical threats.

2. Researches existing industrial honeypots and develops a methodology to use them in the defense framework.

3. Introduces a mechanism to isolate the affected subnetwork to prevent lateral propagation of attack.

The remainder of this document is organized as follows. Chapter 2 describes the attacker model used in the work. Chapter 4 gives an overview of the defensive architecture. Chapter 5 evaluates our solution; describes simulated data, and evaluates the performance of the proposed framework in terms of attack detection and resiliency against attacks. Chapter 6 discusses the limitations of the work, and a few suggestions for how it can be improved. Chapter 7.1 summarizes the work. Possibilities for future work are discussed in Chapter 7.2.

CHAPTER

2

ATTACKER MODEL

We describe the reference attacker model in this section. We assume that the attacker's goal is to compromise the normal functioning of the enterprise. For example, it may want to shut down the functional units, corrupt the data and backups, or steal intellectual property during the attack (e.g., production plans, chemical formulae). We assume the attacker is an insider or has already breached the perimeter defenses. Once connected to the OT network, the attacker performs reconnaissance to fingerprint the target systems and obtain basic system information, such as the number of devices, their addresses, their port status, and the type of industrial protocol. We assume that the attacker has advanced knowledge of industrial protocols. The attacker may also be familiar with underlying physical processes and control logic. After reconnaissance, the attacker identifies a target and starts interacting with the target as a malicious device. It may do the following actions: modify a resource (e.g., change the HMI process tags, or change the PLC register values to corrupt the functionality), delete a resource (e.g., ladder logic programs from PLC), use the resource to control/attack other resources (e.g., send malicious commands or malformed packets to PLC from HMI).

The cyber kill chain (CKC) is a classic cybersecurity model to better understand the stages required to execute an attack. The model was developed by the computer security

incident response team (CSIRT) at Lockheed Martin. The purpose of the model is to come up with solutions to stop the attack in each of the stages. We define the CKC in our attacker model as following:

1. Intrusion: At the intrusion stage, the attacker tries to compromise the perimeter defences to gain access to the OT network. The attacker can be an insider or have an insider accomplice who already has access to bypass the perimeter defenses. They can also do so by using social engineering attacks, using a compromised system or account, or exploiting a gap in security.

2. Reconnaissance: At the reconnaissance stage, the attacker fingerprints the target systems and obtains basic system information. This is the first stage where our framework can identify the presence of an attacker in the network. If the interaction of the attacker is an outlier to the baseline, the framework identifies the attacker's presence and takes appropriate actions. The attacker can also go unnoticed in this stage if the attacker actions do not raise enough alarms to trigger defensive actions.

3. Exploitation: Once the attacker gets information about the network, it can then use this information to identify vulnerabilities in the systems and exploit those in the exploitation stage. Any action taken by the attacker can trigger further alarms that will help the defensive framework to identify it's presence in the network.

4. Privilege Escalation: In the privilege escalation stage, the goal of the attacker is to escalate their privileges in order to compromise more important assets. They try to

5. Lateral Movement: The attacker tries to propagate the attack to additional sub-networks in order to meet their goal in the lateral movement stage. This is the most critical stage as the attacker can cause major harm to normal operations of the organisation, if they succeed in doing so. Our framework is designed to isolate the affected sub-network when the presence of the attacker is determined in the sub-network. This stops the attacker to propagate the attack to additional sub-networks.

6. Obfuscation: In the obfuscation stage, the attacker tries to cover it's tracks to hide their actions in the previous stages. They can do so by changing the logs, or tampering the timestamps in the system. This raises further alarms that helps the framework to locate the attacker.

7. Denial of Service: One of the goals of the attacker is to disrupt the organization's operations, and it can do so in the denial of service stage. The goal of the defense framework is to not let the attacker reach this stage.

8. Exfiltration: If the attacker finds some sensitive data, they copy the data outside the organisation in the exfiltration stage. The frameworks prevents the exfiltration of significant amount of data by isolating the affected sub-network. Although, there is a possibility that the attacker can exfiltrate small amount of data and remain unnoticed.

CHAPTER

3

APPROACH OVERVIEW

We lay out the overview of the approach in this section. The defense framework learns the baseline behavior of the network and then monitors the network for any deviation from the expected behavior. We do so by maintaining a risk score for each type of resource in the network. Depending on the risk score of a type of resource in the system, the defender takes the following actions:

Table 3.1: Defender actions

Risk Score	**Action**
Low	No Action / Decrease the number of honeypot devices
Medium	Increase the number of honeypot devices
High	Isolate the affected subnetwork

To achieve this, the defense framework we define has the following properties:

- Adaptive: This property keeps the overhead of memory and computation low when there is low risk of an attack. When the risk level is low, we can have a minimum

number of honeypot resources running and ramp up the number as the risk level goes up. This also keeps the overhead on the network low since there is minimal communication between the honeypot devices when the network is not under attack. While the system might be less secure initially, but as soon as the risk level goes up, and more honeypots are instantiated, the likelihood of detecting an attacker increases.

- Virtual honeypots: Having virtual honeypots enables the defense framework to be plugged into an existing network architecture of an organization. It also makes the adaptive property more practical, as physical devices need not be kept on standby. The virtual honeypots can be created and destroyed as the risk level changes in the system.

- Modular: The defense framework is designed to be modular and each module focuses on one specialized task. This enables the framework to be updated easily and more modules can be added to improve functionality. The modular feature also allows the framework to be easily integrated with an existing network architecture, as custom modules can be added to pre-process or post-process the data according to individual needs.

We make the important assumption that the defense framework is not compromised during an attack. To make it more probable that the defense framework is not compromised, the modules of the framework communicate over a separate internal network. Modbus TCP/IP is one of the most commonly used protocols in industrial networks, and we use the same in our implementation. The framework can however work with any plain text protocol. The architecture is defined in detail in the next section.

CHAPTER

4

ARCHITECTURE

In this section, we define the architecture of our framework. The framework overcomes the limitations of traditional cyber defense approaches by implementing modular plugins that are specifically designed for OT networks. These plugins are supported by sub-modules that come together to perform a specific function in the framework.

4.1 Architecture Overview

Fig.4.1 shows a high-level topology of a typical OT network. There are four levels represented in the topology. Actuators and sensors constitute Level-0, and they interact with the physical environment. Level-1 devices (PLCs, RTUs) control Level-0 devices and directly communicate with them. Level-1 devices handle automating the processes in the OT network. Supervision and control devices(HMIs, MTUs) constitute Level-2. Devices in Level-2 provide control over the lower-level devices and the interfaces to do that. Finally, Level-3 provides management and storage solutions and acts as a DMZ between the enterprise and operational networks. For the sake of simplicity, we assume that a single network switch

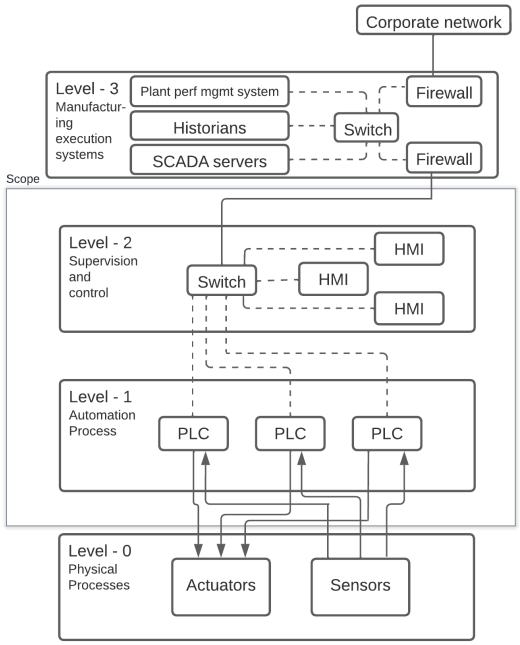

Figure 4.1: High-level topology of a typical OT network.

handles the traffic from Level-1 and Level-2 devices. However, the approach will work the same with multiple switches.

Since all the physical operations are managed and controlled by devices Level-1 and Level-2, we focus our defense architecture around these. Specifically, we identify HMI and PLC as critical assets. Fig.4.2 represents a high-level overview of the defense architecture and where it fits in the OT network. The three components that comprise the architecture are:

1. Risk management: It is composed of four sub-modules. It is responsible for identifying the base state of the system, monitoring the system for anomalies, calculating the risk score of assets based on the state of the system, and reporting the risk score to concerned components when certain conditions are met.

2. Honeypot management: It is responsible for creating and deleting the honeypot devices and managing honey traffic to replicate physical plant processes and control routines.

3. Isolation management: It is responsible for aiding the human operator in making

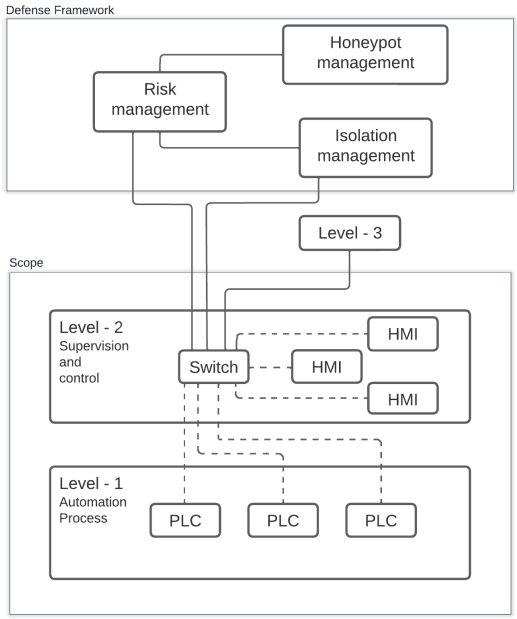

Figure 4.2: Defense framework.

decisions in case of an attack. It can take a backup of the current system state, restore the system from a backup and isolate the critical subnetwork from the compromised network to prevent lateral propagation of the attack.

Each component can operate independently on its own and interacts with other components through APIs over a separate private network. We assume the adversary cannot compromise this private network in our implementation. In the subsequent subsections, we provide a high-level design of the components. Fig. 4.3 shows the high level of interactions between the components and the OT network.

4.2 Risk management

Risk management involves monitoring the risk of the devices in the system and determining if there is a threat in the network. Our implementation is done in two stages, with the help of four sub-modules. As described in Fig. 4.4, Stage 1 is to establish a baseline of the system's network traffic under the conditions when a threat is not present in the network. The Baseline Generation Module (BGM) is responsible for performing this task. The generated

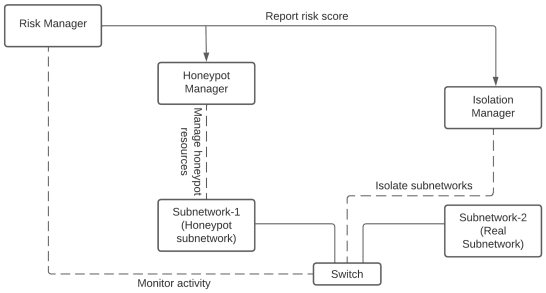

Figure 4.3: Components overview.

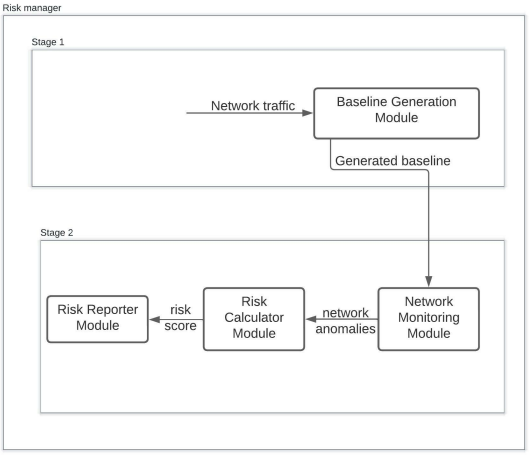

Figure 4.4: Risk Manager.

baseline is then utilized in Stage 2, where the Network Monitoring Module (NMM) uses it to monitor and report any anomalies in the network traffic. In addition, NMM works with the Risk Calculator Module (RCM) and Risk Reporter Module (RRM) to alert other plugins of the presence of any threat in the network.

Since a learning phase is involved before the active detection phase, we assume that the system is in a safe configuration during the learning phase and while setting up the modules that use the data acquired in this phase. The risk management plugin submodules are described below.

4.2.1 Baseline Generation Module

BGM operates in the learning phase and uses deep packet inspection to create rules that classify network packets as malicious or benign. The rules take advantage of the OT network's fixed topology and record the devices' communication patterns. If the topology changes, the administrator triggers the BGM again, which creates a new set of rules. The rules are made based on the following parameters:

1. 4-tuple: Since the topology is fixed, each device in the network communicates with a fixed number of OT devices. This allows us to create an allowlist that documents all IP-port combinations of devices communicating with each device.

2. Memory-address (PLC specific): A memory-address allowlist documents the addresses (coils and registers) accessed via reads or writes over the network. The module reads the network packets, identifies the target, and records the memory location the packet is trying to read/modify.

3. Throughput: All devices' read and write throughput is captured in a time series. It documents the traffic throughput a device is expected to see at a given time of day.

We assume that the traffic pattern remains similar, and there are only slight variations during operation over a significant duration of time. If that requirement is not fulfilled, the rule-based approach will not be ideal. The module can then use statistical and machine-learning-based techniques to identify patterns.

4.2.2 Network Monitoring Module

Once BGM identifies the normal network behavior in Stage 1, NNM uses it in Stage 2 to identify any anomalies in the traffic. First, it computes the specified parameters with the live traffic and compares them against the normal network behavior. If it finds any network packets for which calculated parameters do not match the baseline, it reports the event to the Risk Calculator Module. The NMM also considers the uncertainty value in the case of throughput parameters to avoid False Positives. It only reports an event to the RCM when the following condition is met at any particular time:

$$\alpha(i) > \beta(i) + \delta(i)$$

where $\alpha(i)$ is the i-th throughput parameter value computed in the live traffic, $\beta(i)$ is the i-th throughput parameter value in the baseline, and $\delta(i)$ is the uncertainty value chosen for that parameter.

4.2.3 Risk Calculator Module

The risk calculator module keeps track of the risk score of the devices in the OT network. While there are many risk computation models specific to industrial systems (Cherdantseva et al. 2016) (Peng et al. 2013) (Shi et al. 2018) (Chen et al. 2015), we use the NIST guide for risk assessment (Gallagher and Blank 2012) for risk score computation in our work. It defines adversarial risk within the cybersecurity environment as:

$$R_i = T_i * V_i * I_i * L_i$$

where T is threat, V is vulnerability, I is impact, L is the likelihood and i is the device being assessed. Threat is defined as a function of opportunity, capability and intent. We use the following representation of threat in our work:

$$Threat = Capability * Intent * Opportunity$$

From the above equations, we set our risk equation to be:

$$R_i = \sqrt[6]{(C_i * In_i * O_i * V_i * I_i * L_i)}$$

where C_i is the capability of an attacker on an individual system, In_i is the attacker's intent on an individual system, and O_i is the attacker's opportunity on an individual system. We use all values on a scale of 10, and the final risk score is calculated on a scale of 10, which is then labeled as low(0-3), medium(4-6), or high(7-10) for taking decisions. We use the assessment scales described in appendices of NIST guide (Gallagher and Blank 2012) to calculate the risk score in our implementation. The same guide can be used to choose parameter values in future implementations. Since there is a chance that an incorrect value for a parameter can be selected, as the transition boundaries are not very well defined, we categorize the score into buckets to minimize the impact of incorrect parameter values.

On receiving an event from the NMM, the RCM computes the risk score for the device the event was reported for, considering new information it received. The calculated risk

score is forwarded to the Risk Reporter Module. If no event is received from the NMM, the risk score for devices with elevated risk is computed after a fixed interval and forwarded to RRM to act.

4.2.4 Risk Reporter Module

The Risk Reporter Module communicates the risk score to the concerned module. It reports the risk score to the honeypot manager and isolation manager if the risk score falls into categories that need reporting.

4.3 Honeypot management

Honeypot management involves managing the number of honeypot resources of a particular type in the network, depending on the risk score of that type of resource. In our implementation, the honeypot manager takes care of creating or removing virtual honeypots from the network with the change of risk score. There has been much recent development in virtual honeypots for Industrial networks. Some solutions offer a complete network of ICS devices based on honeynets that the user can configure (Serbanescu et al. 2015), and other solutions let the user configure individual devices (Alves and Morris 2018) (López-Morales et al. 2020). Since, in our implementation, we need control over individual devices, we use openPLC for a virtual honeypot for PLC and ScadaBR from the same project (openplcproject) as a virtual honeypot for HMI.

As the honeypot manager receives the risk score for a specific device (PLC or HMI), it performs the following actions:

1. If the risk score is low, reduce the number of specific honeypot device. (until it reaches configured minimum number of honeypot devices)

2. If the risk level is moderate, create a new honeypot device and set up communication with other honeypot devices.

After the creation of a honeypot resource, the honeypot manager establishes communication between the newly created honeypot resource and its already existing counterparts. For example, if the honey manager receives a high risk score for PLC, it will create a new instance of honeypot PLC, connect with already existing honeypot HMI and start a dummy

task to initiate honey traffic. In our implementation, we use the traffic files captured by the BGM to initiate honey traffic. More sophisticated traffic generation techniques using machine learning or GAN (Zolbayar et al. 2022) can be used in future work.

4.4 Isolation management

Isolation management is responsible for containing an attacker to a subnetwork, to prevent lateral propagation of an attack. In our implementation, isolation manager periodically receives risk scores from the Risk Reporter Module. If the risk score of any device crosses a predefined threshold, the isolation manager isolates the particular subnetwork to contain the threat. It also notifies the system administrator about the threat. The system administrator can then decide to restore the system from a backup, or isolate the critical subnetwork until the threat is identified and terminated.

CHAPTER

5

EVALUATION

We set up an experimental testbed to perform possible attacks. The simulated attacker tries to access the file system of the HMI and manipulate the User Program to be uploaded to the PLC in a series of tests. The attacker also tries to manipulate PLC coils and registers directly over the network in some of the tests. The testbed implements our defender model. Based on how our defender reacts to the introduced attacks, we assess it's viability using metrics from other research models implemented for the security of OT networks.

5.1 Attack Response Experiment

We assume that the attacker has breached the perimeter defenses and is already inside the enterprise network. We deploy our defender model that should be able to detect the presence of attacker inside the network, creating and deleting honeypot resources for the purpose of the same and report it to the isolation manager with a risk score.

We use Airgap as our isolation manager. Airgap is a patent pending device, that is designed to isolate networks and backups in case of an attack. It is connected to the switch

which connects our real subnetwork and the honeypot subnetwork as shown in Fig. 5.1.

5.1.1 Experimental Setup

Since we identified HMI and PLC as our critical assets, our experimental setup comprises two identical VMs running windows 10, which we call host machines. Each of the host machines is connected to a NAS, which stores state information of the HMI and PLC, and can be used to restore the state of the system if it ever goes down. One of the host machines hosts our HMI, SIMATIC WinCC Explore. The WinCC software has some well-known vulnerabilities, some of which were exploited by the famous Stuxnet worm. This HMI is connected to a Siemens S7-1200 PLC, which is made to perform a dummy operation like monitor the temperature through a connected thermometer. The PLC raises an alarm in the HMI if the temperature is outside a certain range.

The other system hosts a ScadaBR HMI, which can be instatiated through a script, connected to OpenPLC PLC, which can be created and deleted by the defender as required. Our defender is an Airgap agent process running in a different VM in the same hypervisor, communicating with the risk manager, and honeypot manager running in the host machines.

The risk management plugin monitors any changes in the critical subassets of the PLCs. On every action, it determines if the system is at risk due to the performed action. It calculates the risk score, and communicates that to the Airgap agent process through the virtual network interface. The Airgap agent process receives the risk score from the risk manager plugins. Depending on the received risk scores, the Airgap agent performs the following actions:

1. Isolate the backup from the network if risk is medium to high.

2. Isolate the compromised subnetwork from rest of the network if the risk if high to critical.

3. Shut down the current operations, and restart the processes by loading the safe states from the backup.

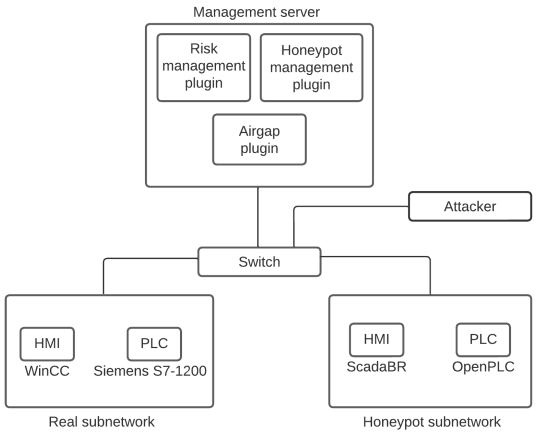

Figure 5.1: Experimental setup.

5.1.2 Expected Results

The defender should identify the presence of the attacker once the attack has started, and should perform Action 1 and Action 2. As mentioned earlier, the cost of shutting down an OT network is significantly higher than that of an IT network. Therefore, Action 3 is not desirable if the threat can be contained by performing Action 1 and Action 2. But if both the subnetworks are compromised, Action 3 must be performed. The actions taken are based on the risk score calculated.

We use the following terms in this section for calculating the accuracy of our defense framework:

- TP: This is the number of True Positive cases. These are all the cases where an ongoing attack was correctly identified by the defense framework.
- FP: This is the number of False Positive cases. These are all the cases where benign traffic is incorrectly identified as attacker traffic.
- TN: This is the number of True Negative cases. These are all the cases where the defense framework correctly identifies that there is no attacker present in the network.
- FN: This is the number of False negative cases. These are all the cases where the attacker traffic is incorrectly identifies that there is an attacker present in the network, when there is not attacker.

5.1.3 Results

We evaluate the results for different attack scenarios and calculate the following values using the labels described above:

1. Sensitivity: This is the true positive rate.

$$Sensitivity = TP/(TP+FN)$$

2. Specificity: This is the false positive rate.

$$Specificity = FP/(FP+FN)$$

3. Accuracy: This is the fraction of correct classification of cases.

$$Accuracy = (TP+TN)/(TP+TN+FP+FN)$$

The results of the experiment are as below (ROC curve shown in 5.2):

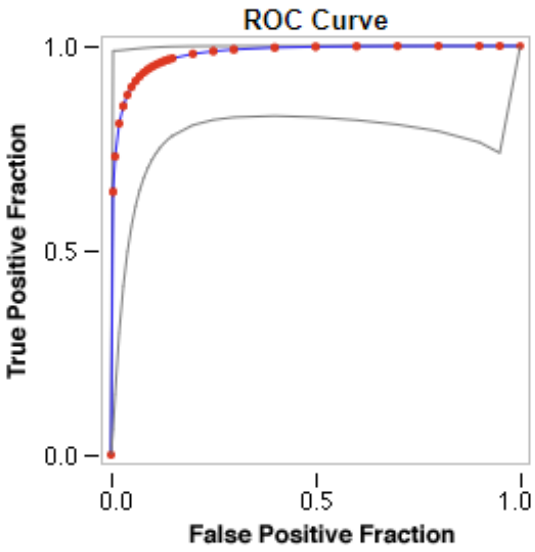

Figure 5.2: ROC curve - Experiment 1
x-axis: False Positive fraction y-axis: True Positive fraction

Number of Cases:	40
Number Correct:	36
Accuracy:	90%
Sensitivity:	88.5%
Specificity:	92.9%
Pos Cases Missed:	3
Neg Cases Missed:	1
Fitted ROC Area:	0.975
Empiric ROC Area:	0.973

5.2 Adaptive Honey Resources

The goal here is to maintain the number of honeypot resources based on the risk level of the resources. In addition to the risk manager, the host machines has another module, for creating and deleting honeypot resources. This module queries the risk score from the Airgap agent process and performs the following actions:

1. If the risk score is low, delete all honeypot PLCs.
2. If the risk level is low to moderate, create a honeypot PLC and connect it with the HMI.

The risk manager takes into consideration the newly created honeypot resources in the risk calculation and give them a higher priority as a legitimate user will be unaware of the new resources' existence, so any action performed on it will more likely be by an attacker.

5.2.1 Experimental Setup

Using the same experimental setup as in subsection 5.1.1. Additionally, a honeypot manager will be installed on the host machines. The Airgap agent now performs the following actions:

1. Alert the honeypot management plugin if the risk level is low to medium.
2. Isolate the backup from the network if risk is medium to high.
3. Isolate the compromised subnetwork from the rest of the network if the risk if high to critical.

4. Shut down the current operations, and restart the processes by loading the safe states from the backup.

5.2.2 Expected Results

The defender should identify the presence of the attacker once the attack has started, and should successfully perform Action 1 along with the actions defined in the previous section. The actions are taken based on the risk score calculated. We evaluate the results for different attack scenarios and calculate the true-positive rate, the false-positive rate, and the area under Receiver Operating Characteristic curve.

5.2.3 Results

We evaluate the results for different attack scenarios and calculate the following values using the labels described above(ROC curve shown in 5.3):

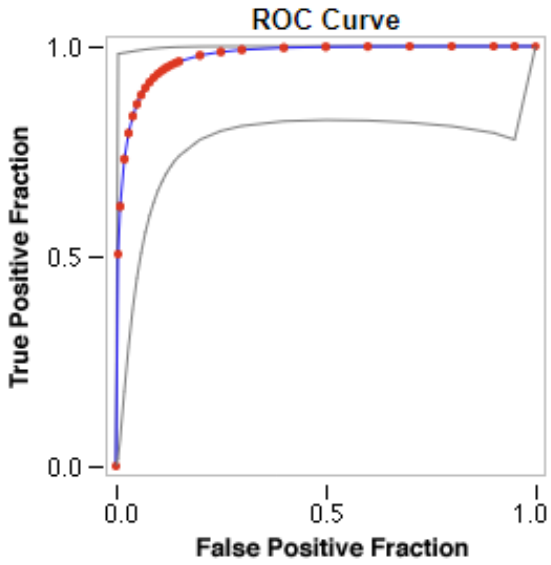

Figure 5.3: ROC curve - Experiment 2
x-axis: False Positive fraction y-axis: True Positive fraction

Number of Cases	40
Number Correct	37
Accuracy	92.5%
Sensitivity	92.3%
Specificity	92.9%
Pos Cases Missed	2
Neg Cases Missed	1
Fitted ROC Area	0.981
Empiric ROC Area	0.977

On observing the results of the above experiments, Experiment 2 certainly performs better than Experiment 1 as is evident from accuracy. Experiment 2 does have performance overhead as it does create additional honeypot PLCs in the VM, but that overhead gives a good ROI in terms of locating the attacker and notifying the Airgap agent. This overhead is considerably less than the actual PLCs and HMIs deployed in the existing techniques both in terms of number of resources and traffic overhead on the network. Therefore, we can conclude that this technique when used along with IDS, gives an extra layer of protection for the OT networks with minimal performance overhead, as the new virtual resources are only created when the attacker has actually breached the perimeter defences, unlike the other deception implementations using static honeypot resources, which have a performance overhead regardless of the attack.

CHAPTER

6

LIMITATIONS

We discuss the key assumptions and limitations of our approach in this section.

- The goal of the defense framework is to ensure Integrity and Availability. Confidentiality is a secondary goal, and while the framework will raise the alarm for the exfiltration of a significant amount of data, it may not be able to stop the exfiltration of small amounts of data.
- We assume that the defense framework modules cannot be corrupted by the attacker. Even if the attacker is aware of the framework, it cannot manipulate it without giving away its presence. We try to achieve this by implementing separate communication channels over which the modules of the framework communicate. This can be further ensured by implementing strict communication protocols so that only minimal entities can interfere with the operations of the defense framework.
- For the defense framework to identify the presence of an attacker in the network, the attacker has to interact with the devices in the network that are being monitored.

If the attacker sits in the network passively without any network trace, the defense framework will not know about it's presence.

- The parameter values used to calculate the risk score are chosen from the assessment scales described in the NIST guide (Gallagher and Blank 2012). Since the transition boundaries are not very well defined, there is a possibility for selecting incorrect values for some of the parameters. To minimize the impact of incorrect parameter value selection, we categorize the score into low, medium and high buckets. We then take actions based on what bucket the risk score falls into. For example, using the risk formula in 4.2.3 consider these two scenarios:

Table 6.1: Risk score calculation: Case 1

Parameter	**Value**
C_i	5
In_i	5
O_i	5
V_i	5
I_i	5
L_i	5
Risk Score	**5**

Table 6.2: Risk score calculation: Case 2

Parameter	**Value**
C_i	5
In_i	5
O_i	5
V_i	5
I_i	7
L_i	7
Risk Score	**5.59**

In both cases, the risk score falls into the medium category and hence the action taken in both cases is the same.

- Isolating the affected subnetwork can hinder the organization's operations temporarily. An improvement will be giving a choice to the administrator to take appropriate action in the situation, but that involves manual intervention. In our implementation, we choose to isolate the network automatically, and the critical operations can be carried out manually. The isolation disables the attacker from further manipulation and also prevents the lateral propagation of the attack into other subnetworks.

CHAPTER

7

CONCLUSIONS

7.1 Conclusion

The lack of cybersecurity approaches to help private companies prevent and recover from threats via the internet threatens the reliability of the services provided by the critical infrastructure sectors. Today, hackers, terrorist groups, and nation states have more opportunities to penetrate the critical infrastructure because of increased automation and interconnectivity to the internet. There is a wide range of cyber threats, cyberwarfare being the worst case scenario.

The traditional protection mechanisms focus only on perimeter defences, which we see become weaker and weaker. This paper explores the possibility of using virtual honeypot resources which are believable enough that the attacker is lured to them without realising that they are not actual resources. While the use of virtual honeypots is seen a lot in the IT networks, it was easy to identify a virtual honey resource in an industrial setup using the honeyscore. But due to recent advancements in the area, more and more honey resources are now available for the industrial network as well, which an attacker cannot tell apart

from an actual resource. This is particularly beneficial for the OT networks as setting up real honey resources adds to the cost overhead of the corporation, and it is static in nature. This opens up a new dimension for implementing defensive deception in the industrial network, with a lot of scope for further research. This work, by demonstrating that these techniques can actually work in a physical setup, provides a path to such further research.

7.2 Future Work

We made important assumptions about the attacker. The most important assumption is that the attacker has breached the perimeter defences and is already in the network. While our focus was on HMIs and PLCs, future work include other critical assets. Monitoring every resource in an OT network will provide a greater level of security as the attacker can then be detected as soon as it gets into the network, through any of the endpoints.

We assumed that a corporate network will be divided into subnetworks, and this assumption is central to our implementation, as the idea is to contain the attacker in subnetworks and prevent the lateral movement of the attack. This is however not true in many small to medium sized corporations as they operate flat networks. Although there are guidelines and policies defined by NIST (Stouffer et al. 2015), CISA (CISA), and ISA (ISA), they are not often followed in practice. Due to this, they are even more vulnerable to attacks, as a compromised corporate network automatically means a compromised operational network. As future work, techniques can be developed to protect the critical infrastructure in networks like these, where the attacker cannot be contained once the perimeter defences have been breached.

REFERENCES

T. Alves and T. Morris, "Openplc: An iec 61,131–3 compliant open source industrial controller for cyber security research," *Computers & Security*, vol. 78, pp. 364–379, 2018, accessed: July 21, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404818305388

P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *Communications and Multimedia Security*, B. De Decker and A. Zúquete, Eds., 2014.

Q. Chen, R. Abercrombie, and F. Sheldon, "Risk assessment for industrial control systems quantifying availability using mean failure cost (mfc)," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 5, pp. 205–220, 09 2015.

Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, and K. Stoddart, "A review of cyber security risk assessment methods for scada systems," *Computers & Security*, vol. 56, pp. 1–27, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404815001388

CISA, "Cisa - securing industrial control systems," accessed: 03/15/2022. [Online]. Available: https://www.cisa.gov/publication/securing-industrial-control-systems

CSIRT, "The cyber kill chain (ckc)," accessed: 11/4/2022. [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Fortinet, "2020 state of operational technology and cybersecurity report," Fortinet, Tech. Rep., 2020. [Online]. Available: https://www.fortinet.com/content/dam/fortinet/assets/analyst-reports/report-state-of-operational-technology.pdf

P. D. Gallagher and R. M. Blank, "Guide for conducting risk assessments," NIST, Tech. Rep., 2012. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final

ISA, "Isa99 - industrial automation and control systems security," accessed: 03/27/2022. [Online]. Available: https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa99

W. Jingran, L. Mingzhe, X. Aidong, H. Bo, H. Xiaojia, and Z. Xiufang, "Research and implementation of secure industrial communication protocols," in *2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS)*, 2020.

E. López-Morales, C. Rubio-Medrano, A. Doupé, Y. Shoshitaishvili, R. Wang, T. Bao, and G.-J. Ahn, "Honeyplc: A next-generation honeypot for industrial control systems," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications*

Security, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 279–291. [Online]. Available: https://doi.org/10.1145/3372297.3423356

T. Morris, R. Vaughn, and Y. Dandass, "A retrofit network intrusion detection system for modbus rtu and ascii industrial control systems," in *2012 45th Hawaii International Conference on System Sciences*, 2012.

K. Paridari, N. O'Mahony, A. El-Din Mady, R. Chabukswar, M. Boubekeur, and H. Sandberg, "A framework for attack-resilient industrial control systems: Attack detection and controller reconfiguration," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 113–128, 2018.

Y. Peng, T. Lu, J. Liu, Y. Gao, X. Guo, and F. Xie, "Cyber-physical system risk assessment," in *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2013, pp. 442–447.

A. V. Serbanescu, S. Obermeier, and D.-Y. Yu, "A scalable honeynet architecture for industrial control systems," in *ICETE*, 2015.

L. Shi, Q. Dai, and Y. Ni, "Cyber–physical interactions in power systems: A review of models, methods, and applications," *Electric Power Systems Research*, vol. 163, pp. 396–412, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779618302086

K. Stouffer, S. Lightman, V. Pillitteri, M. Abrams, and A. Hahn, "Guide to industrial control systems (ics) security," NIST, Tech. Rep., 2015. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

B.-E. Zolbayar, R. Sheatsley, P. McDaniel, M. J. Weisman, S. Zhu, S. Zhu, and S. Krishnamurthy, "Generating practical adversarial network traffic flows using nidsgan," 2022. [Online]. Available: https://arxiv.org/abs/2203.06694

APPENDIX

APPENDIX

A

ACRONYMS

A summary of all acronyms is documented in Table A.1.

Table A.1: A summary of acronyms used in alphabetical order.

Acronym	Abbreviation
Cybersecurity and Infrastructure Security Agency	CISA
Generative adversarial network	GAN
Human Machine Interface	HMI
National Institute of Standards and Technology	NIST
Operational Technology	OT
Programmable Logic Controller	PLC