IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 24, 2021, accepted September 6, 2021, date of publication September 9, 2021,
date of current version September 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3111420

Last Line of Defense: Reliability Through Inducing Cyber Threat Hunting With Deception in SCADA Networks

ABDUL BASIT AJMAL[1], MASOOM ALAM[1], AWAIS ABDUL KHALIQ[1], SHAWAL KHAN[1], ZAKRIA QADIR[2], AND M. A. PARVEZ MAHMUD[3]

[1]Cyber Security Laboratory, Department of Computer Science, COMSATS University Islamabad, Islamabad 44500, Pakistan
[2]School of Computing Engineering and Mathematics, Western Sydney University, Penrith, NSW 2751, Australia
[3]School of Engineering, Deakin University, Geelong, VIC 3216, Australia

Corresponding author: Abdul Basit Ajmal (abdulbasitajmal7@gmail.com)

This work was supported by the Higher Education Commission Pakistan, Technology Development Fund Grant 206.

ABSTRACT There exists a gap between existing security mechanisms and their ability to detect advancing threats. Antivirus and EDR (End Point Detection and Response) aim to detect and prevent threats; such security mechanisms are reactive. This approach did not prove to be effective in protecting against stealthy attacks. SCADA (Supervisory Control and Data Acquisition) security is crucial for any country. However, SCADA is always an easy target for adversaries due to a lack of security for heterogeneous devices. An attack on SCADA is mainly considered a national-level threat. Recent research on SCADA security has not considered ''unknown threats,'' which has left a gap in security. The proactive approach, such as threat hunting, is the need of the hour. In this research, we investigated that threat hunting in conjunction with cyber deception and kill chain has countervailing effects on detecting SCADA threats and mitigating them. We have used the concept of ''decoy farm'' in the SCADA network, where all attacks are engaged. Moreover, we present a novel threat detection and prevention approach for SCADA, focusing on unknown threats. To test the effectiveness of approach, we emulated several SCADA, Linux and Windows based attacks on a simulated SCADA network. We have concluded that our approach detects and prevents the attacker before using the current reactive approach and security mechanism for SCADA with enhanced protection for heterogeneous devices. The results and experiments show that the proposed threat hunting approach has significantly improved the threat detection ability.

INDEX TERMS Threat hunting, indicators of compromise (IOC), Industrial Internet of Things (IIoT), supervisory control and data acquisition (SCADA), cyber deception, honeypots, decoys.

I. INTRODUCTION

SCADA system is a network of different components, which are responsible for the reliable and accurate working of crucial industrial processes. SCADA system gathers and organizes data from different actuators for real-time monitoring. SCADA consists of components, such as PLC (programable logical controller), HMI (human-machine interaction), MTU (Master terminal unit), Historian, and RTU's (Remote terminal unit). They combine and build a complete network. PLC's communicate with HMI through RTU and MTU. Example is given in Figure 1. Heterogeneity of devices used by SCADA makes it more difficult for defenders to counter threats [1].

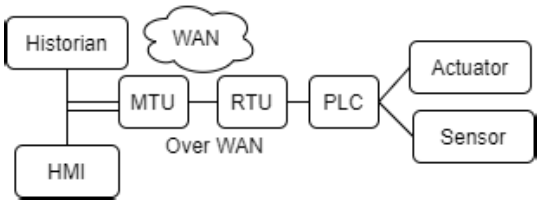

FIGURE 1. Abstract SCADA network.

Most of the security tools are less interactive, working on specific logic, for example: watching a specific gateway and searching for specific threats. This approach is totally

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

based on actions performed by an adversary that will invoke the security system. This approach does not help to foresee threats. Cybercriminals are well aware of such approaches and know-how to deal with them [2]. For example, polymorphous malware.

Many organizations and SCADA networks rely on vulnerability assessment and partially reactive security solutions. This approach is suitable for known adversaries, but that is not the case all the time. Adversaries are motivated to employ new, better, and improved attack methods and techniques. Such threats are categorized as ''unknown threats''. Recent research did not talks about unknown threats [3]–[5] and focuses on reactive approaches using machine learning and attack graphs but problem is still there for unknown file-less stealthy attacks. We need a approach to detect and prevent unknown threats for SCADA systems. While detection, learning new techniques from adversary is crucial to foreseen threats. If we look at Stuxnet, it stayed undetected in the network for a long time and exploited around 20 zero-days in Siemens step7. It might be possible that nuclear plant was protected from all visible, known threats but attackers exploited invisible, unknown threats. Here comes threat hunting. It uncovers new TTP's (techniques, tactics, and procedures) to forecast threats.

We are presenting a novel threat hunting approach for SCADA to detect and mitigate unknown threats. Approach uses ''decoy farm'' where all attacks are engaged and threat hunters collect IOC's from decoy farm and learn new TTP's. Decoy farm is a collection of several decoys connected through each other. This Proactive approach will help to increase the threat detection and mitigation ability of SCADA network. We used threat hunting in conjunction with cyber deception and kill chain to detect and mitigate unknown threats. For this purpose, we used specially crafted lures and PLC decoys in SDN (software deined newtork) to achieve our approach implementation. From past and recent research, it has proved that deception has countervailing effects on improving defense [6].

In order to understand this, we will review threat hunting methodology and process. In the following Section IV, we have discussed experiment results with depth analysis of the threat hunting approach.

The paper is organized into five major sections. Section II describes the related work, Section III discusses threat hunting while analyzing its methodologies and the whole process involved in it. Also, we will analyze how to gather intelligence and its data sources in detail. Section IV discusses threat hunting novel approach for SCADA networks with practical (including results and analysis). Section five discusses the analysis of the experiment and future work. Section VI, sums up the paper with a conclusion.

II. RELATED WORK

Research article ''SCADA systems: Vulnerability assessment and security recommendations'' [3] and other related work [7], [8] describes a variety of common vulnerabilities for SCADA networks. Furthermore, they have provided recommendations for each vulnerability. This research article has generally considered ''Existing Vulnerabilities'' and corresponding ''Known Threats.'' This raises a question about unknown threats exploiting zero-days and targeting SCADA networks. Such as ''Stuxnet'' appeared again in recent years [9].

Previous work on SCADA security such as [4], [5] considers the limited scope for securing SCADA, such as dll injections for windows and securing windows host only. The major flaw with this approach is; this scenario only fits where the attacker has already got initial access and trying to load the actual payload in memory using dll injection. Due to a lack of realism for attack evaluation on approach, SCADA is still vulnerable. Bypassing dll injection detection mechanism is not difficult for the real-world adversary. The authors do not consider unknown threats. Moreover, approaches to predict attacks such as [10] is a good approach. However, it uses several static preferences for each node which is static in nature. However, in reality, attacker deals with uncertainty while launching attacks so, in such cases attack prediction vs. actual threat model can be different while keeping the current attack surface same. In [11] authors have used a decentralized approach for preventive threats. We have taken this approach in the ''kill chain scenario'' and integrated it with cyber deception to deceive attackers at each kill chain phase (details are in later sections).

Specifically, our approach covers HMI, PLC, and Endpoints (Windows or Linux) threats, including network-based attacks. Moreover, we tested our approach against adversary-inspired attacks and successfully detected and prevented threats. Our ''Novel Threat Hunting Approach'' has addressed all these issues and has provided an approach for the SCADA threat detection and prevention. In this research, we have used the following tools in Table 2.

Likewise, approaches for detecting intrusions at the network level such as [12], [13] emphasize on reactive approach. While experimenting with our presented approach, we were able to evade such IDS using traffic manipulation/Impersonation using ''Malleable C2''. In our proposed approach, we focus on a proactive approach to enhance threat detection and prevention ability.

Approach we are using in this research for launching attacks and conducting threat hunting is a continuation of our previous research [18] in SCADA context.

The introduced information is likewise agnostic of the investigative strategies utilized throughout the hunting process, enabling the model adaptability to work with any hunting tool or system, such as Stateful examination [19]. The paper [20] describes in detail intrusions and their identification. The diamond model is represented in Figure 2. Such model is used to do following activities [20]:

1) Characterize organized threats
2) Consistently track them as they evolve
3) Sort one from another
4) Figure out ways to counter them

TABLE 1. Some known SCADA threats.

Threat	Tools used	Mitigation
Mass Scanning, Banner Grabbing	Shodan.io	Use proxy
Modbus Register/Coils modify	Metasploit	Access control and isolate slave
Process and Command Injection	Metasploit	Protect process
Parameter manipulation	Metasploit	Authenticate parameters
Botnet & DDoS	Custom tools	IDS/Firewalls
Web Based HMI Attacks	BurpSuite, ZAP	Web App firewall
S7 [14]	Metasploit/custom tool	Patching
Modbus Client sniffer	Metasploit	Authentication protocol hardening
Modbus Slave scanner	Metasploit	Use authentication
Modbus Unit ID scanner	Metasploit	Limit response and authenticate
Remote code execution	Metasploit	Update systems
Masquerading	Traffic Dump	Use PVLAN [15]
Phishing Attacks	Custom tools	Training and Packet-filtering of spam
SQL Injection [16]	OWASP	Query sanitizing [17]
HMI Directory Traversal Attack	Dirbuster	Limit access and parameter check

TABLE 2. Open source tools used.

Tool Name	Purpose	Utility
Mininet, Open vSwitch and Ryu	Environment for building SDN	Can quickly simulate network
Docker	Encapsulate	Reliable environment
CuckooBox	Malware Analysis	Analyzing Unknown
Zeek	Inspecting Network	Gives deep insight to hunters
Maltrail	Inspecting malicious trails	Help analyzing network
Canary Token	Lure	Detect attack at different stages
Conpot	Simulate SCADA	Can simulate PLC's
Honeyd , NOVA	Simulate windows system	Supports variety of services

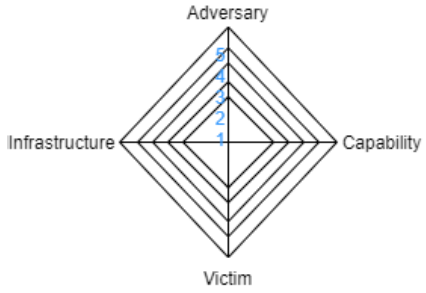

FIGURE 2. Diamond model for intrusion analysis.

A. KILL CHAIN ANALYSIS

The paper with the title "Kill chain for industrial control system" [21] explains adversary actions and techniques with the help of the kill chain. We have derived kill chain methodology for SCADA from [21] and integrated it with cyber deception, and presents kill chain and deception approach for SCADA as shown in Figure 4. This will help us to identify attack behaviors at each phase of an attack.

B. SCADA SIMULATION AND ATTACK SIMULATION

For SCADA simulation, we took help from a paper "Simulating Industrial Control Systems using Mininet" [22], and for attacks "Simulation of cyber-attacks against SCADA systems" [23] we will be discussing them in later sections.

C. THREAT HUNT MODEL

We have used the threat hunt model with slight modifications before hypothesis, details are present in our previous work [18]. SANS defined the formal model for threat hunting that many hunters adopt. Threat hunting is briefly discussed here [24]. Identifying the area of the hunt, including all related equipment; like systems and used protocols and then building a hypothesis, Validating and verifying hypothesis are discussed here [25].

D. SOFTWARE DEFINED NETWORK AND DECOYS

Open source mininet [26] which provides flexible and scale-able SDN which can be integrated with the actual network to supports a wide variety of controllers such as Ryu. We used mininet for SCADA network simulation and launched adversary-inspired attacks to perform threat hunting. We have chosen the "Ryu" controller to route and filter traffic inside SDN, and rules can be set for enhanced security [27]. The controller is configured in a way that attacks will be diverted to SDN and engage the attacker with decoys.

For decoys, projects like Honeynet, Nova, honeyd are used for windows system simulation. We have extracted ''honeyd'' fingerprint data and used it to simulate different devices within SDN (at mininet nodes). Conpot [28] is SCADA (IIoT) honeypot, and it can simulate the majority of SCADA protocols and components, including HMI, with features enough for slowing down the attacker and capturing their activity. Honeypot detection tools can easily detect conpot on the basis of fingerprint data. To avoid fingerprinting decoys, we will be using a customized version of conpot by editing XML files, changing banner, customizing protocol details for a new fresh look.

E. ATTACK ANALYSIS

To quickly analyze the behavior of files, we have used the open-source sandbox ''Cuckoo Sandbox.'' This is a leading open-source sandbox that provides a detailed dynamic analysis of malware. We will be using these sandboxes for analyzing unknown threats. For malicious traffic detection and threat hunitng we used RITA (Real Intelligence Threat Analytics) [29], Zeek Bro, Maltrail [30]. To capture unknown threats present at endpoint, we used ''lures'' in the form of fake active directory files, documents. One example of such lure is canary token. Such lures work as indicator of compromise at endpoint as well as network level. For threat hunting such lures are vital.

F. OPENCTI

This is an open-source database of threats. Organizations use this project to manage their threat data. We have used threat intelligence to keep threat hunters updated about the latest threats and their techniques.

III. INVESTIGATING THREAT HUNTING

Many organizations are unaware that their confidential data is being compromised [31] by an adversary. This happens because security mechanisms lack proactive searching of threats. For example, the firewall is watching a specific gateway. What if the threat has already bypassed them and lurking in your network. Attackers rely on the living of the land techniques to bypass security mechanisms. The use of next-generation firewalls is effective, but it depends on the data set they trained. The attacker might exploit a machine-learning algorithm to teach ''known bad as good'' over a period of time and then launch an attack on SCADA. For example, Microsoft Tay chatbot [32]. Proactive approach is need of hour to counter threats.

The threat hunting process is cyclic Fig. 3 in nature. It consists of four processes. The first one is about the creation of a hypothesis; the second is about verifying and validating the hypothesis. This process also includes further investigation for any proof with the help of tools and techniques. The next process explains new TTP's and patterns. The final process includes enrichment. It is informing the incident response team about new TTP's.

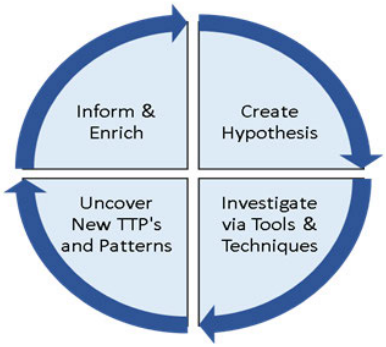

FIGURE 3. Threat hunting cycle.

A. SMALL CASE STUDY

The first step is to create a hypothesis; for example, an employer informs that his system got infected by email-based malware. After some investigation, we found that malware bypassed the sandbox.

1) Hypothesis: Malware has bypassed the sandbox. Sandbox was unable to detect it.
2) Investigation via tools and techniques: We found that malware was embedded inside a pdf with different headers. We also noticed reverse connections to malicious staging servers. Traffic analysis gave us a clear view of the presence of beacons.
3) Uncover New TTP's: Encoded headers with cross-compile capability placed in fragments at different points in pdf. Staging servers used DNS tunneling for connection with the target. [33].
4) Enrichment: This is done by the Incident Response team, which updates new threat definitions and categorizes threats using YARA [34] and STIX. Building actionable intelligence (STIX2) from intelligence feeds, and hunting data is one strategy for enrichment. The second strategy; use shared intelligence across different organizations; this approach is good for saving resources. For shared intelligence, ''OpenCTI'' is used.

B. EXPLANATION

1) PROBLEMS ASSOCIATED WITH EXISTING SECURITY MECHANISM

Typically organizations and SCADA have different layers of defense. Such as, at the network level, they use IDS, and at endpoints, they use EDR for advanced threat detection and prevention. For logging, they use sysmon. All logs are fed to SIEM. The problem with this approach is, there are hundreds of events that SIEM generates. For operators, it is challenging to look into each event. Intelligent solutions have solved this issue, but the problem is still there. APT attacks can still evade them and can stay stealthy for the maximum possible time.

We conducted an attack simulation over such a secure environment and were able to evade IDS and EDR's. We used ''Malleable C2'' to evade network-based security mechanisms. At endpoints, we used fileless and kernel-level exploits to evade security mechanisms. A proactive approach can

increase the likelihood of detecting threats in the early stages of the attack.

We used ''Zeek'' [35] (open-source threat hunting tool) to inspect traffic and found HTTP packets that were spoofed and were using impersonated SSL certificates. In a proactive approach, we detected the threat in the early stage. In this case, we can use a kill chain to sabotage attacker intents. Moreover, we can expect that attacker has the capability of performing advanced attacks (such as zero-day). For such cases, we must aim to detect and divert attacks.

In our presented approach, we have focused on early threat detection as well as engaging attacker in SDN decoy farm (isolate attacker from the actual network). Where threat hunters can learn TTP's effectively by collecting IOCs. We have concluded that this approach has countervailing effects in detecting threats and protecting the actual SCADA network.

C. CONTRIBUTION AND HUNTING APPROACH

The main idea behind the approach is to build a simulated decoy SCADA network which can be used as a target environment to divert and record attacker activities in an isolated environment. Moreover, integrating cyber deception, kill chain, and threat hunting in decoy networks. We are presenting a threat hunting approach in conjunction with deception and kill chain, ensuring early detection and prevention attacks. This approach uses a ''decoy farm'' where attacks are engaged, analyzed, and provides intelligence from each phase of the attack cycle (kill chain) in advance, which facilitates the hunt team to build hypotheses quickly and efficiently to hunt unknown threats. For that purpose, the objective can be stated as follows:

- Keep the attacker engaged and delay malicious activities.
- Record attacker activities and learn new techniques.
- Prevent attack; Keep attacker isolated in a simulated environment.

We used mininet to quickly build SDN. In addition to this, we attached docker containers at each SDN node in Figure 6. Each container is there for a distinct purpose. Node 3 is equipped with special threat hunting tools for capturing and monitoring network protocols and traffic. We have named this node as ''Orchestrating analytic node.'' Each container is configured to send logs periodically to node 3. If the attack is detected, it will generate alerts and forward them to the admin node. Even a network scan or HTTP request will alert the system. Admin can deploy new SCADA decoys to keep attackers engaged using NOVA and conpot. For that purpose, we used scripts to quickly deploy new SCADA honeypots (Inside mininet SDN).

Admin node can manage other containers. Node 1 has running customized conpot, gaspot (SCADA honeypot), which is simulating seven different SCADA protocols such as DNP, modbus, FTP, TFT. In addition to this, HMI and historians are also attached. There is another container (Node 2), which is a purpose-based container. If an attacker tries to extend its activity attacker will be diverted towards this docker, and the admin can extend the network using mininet (mn –topo = single,5 –mac –controller remote –switch ovsk). For the time being, the admin is doing all this manually.

We have named this all simulated environment as *''Deception & Hunting Unit''*. This deception unit can be attached with SCADA at different data points, Modbus slaves. If PLCs are at remote locations, then each deception unit can be deployed with each PLC. Whenever there is a network scan hit on a simulated environment, it will quickly alert node 3. The likelihood of detecting an attack at a very early stage becomes high. Figure 4 explains our deception and kill chain approach.

Inside two attack engagement decoys (node 1 and 2), we have placed lures and breadcrumbs. Sysmon is installed on windows system, simulating HMI using SCADA BR inside node 1. When the attacker sees a complete SCADA system that is vulnerable, this approach will divert the attacker towards the decoy farm and keep the attacker engaged as long as possible.

We can analyze network packets with tools like open-source Zeek bro, which categorizes traffic based on protocols, or use Wireshark to capture network traffic for threat hunting. And then extract metadata, look for anomalies or any beaconing if exists [36].

Brief threat *hunting process proposed in this paper* is in the Figure 5. Actual threat hunt process starts from *step 4-Hunt process* in Figure 5. Pre-hunt activities are taken from [37] and integrated into our approach. At the network level for effective threat hunting, protocols and network logs must be analyzed.

D. THREAT INTELLIGENCE AND DATA SOURCES

Threat hunting uses information from different sources, like Endpoint Detection and Response (EDR), Threat Intelligence (TI), Past Incidents, or Over Dark web [38]. The evolution of threat intelligence is briefly explained here [39]. To build an understanding of new attack techniques, threat intelligence is necessary. For our approach, we focus on publicly available sources, including malware intelligence frameworks like MISP (Malware Information Sharing Platform and Threat Sharing), CISCO Talos TI, Open CTI (for threats), and threat research blogs, websites, and threat reports. We are more focused on the FireEye threat research blog, ATT&CK, MALPEDIA, PT Security, Bleeping, Maloverview, WeLiveSecurity by Node32, VirusRadar, MalPipe, AlienVault Threat Exchange, and UNIT 42 by Palo Alto. We can also use the latest YARA rules to see malicious instincts of the latest threats from Github. PT ESC Threat Intelligence (PT Security), has a dedicated team who release threat reports after in depth analysis of different threats including APT's.

For our proposed approach threat hunters will collect IOC's from ''node 3'', and start building hypothesis and listing data sources in CMF (collection management framework). If there

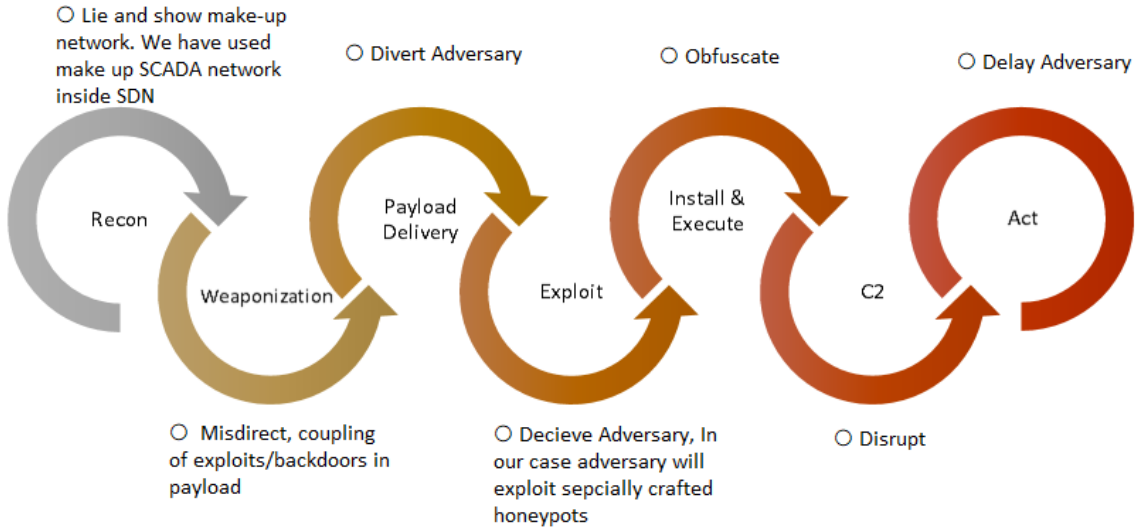

FIGURE 4. Kill chain and deception.

are threats that are unknown, in this case hunters will use dynamic analysis tools present inside node 3 or use REST API of hybrid-analysis.com, malwr.com or virustotal.com.

IV. EXPERIMENT

We will be conducting several SCADA-based attacks, including APT attacks mentioned in Table 4, on a simulated SCADA decoy farm.

A. APPROACH

We have used docker containers running ubuntu on each node connected with the OVS switch in mininet (SDN). Each node is defined with a different purpose, as shown in Figure 6: Lures and decoys are strategically placed so that they can provide intelligence for each ATT&CK phase. Decoy services like ssh, telnet are also running inside node 1. Decoys are deployed inside each container (Node 1 and 2) to detect any network scanning on the network and will provide us with time intelligence. Such IOC sensors are even capable of detecting threats that are lurking inside the network. Canary tokens are used in the form of documents and fake active directories. If an attacker successfully deceives all security measures and tries to ex-filtrate or open docs, it will alert the attacker's location, time zone, MAC, and IP address. All used decoys and IOC sensors are modified form of these open source projects. We used canary-tokens, conpot, honeyd, nova, artillery, hornssh.

1) TOPOLOGY

OVS is our main switch, and port one is connected with Node 1. HMI (we used conpot PLC and scadabr as HMI) and other essential SCADA components are present inside node1. Node 2 is connected to switch on port 2 as in Figure 6. Conpot is used to simulate the whole SCADA network with all necessary protocols and services like Modbus, DNP3, FTP, HTTP server, TFTP, and SSH. Maltrail, Zeek bro, cuckoo

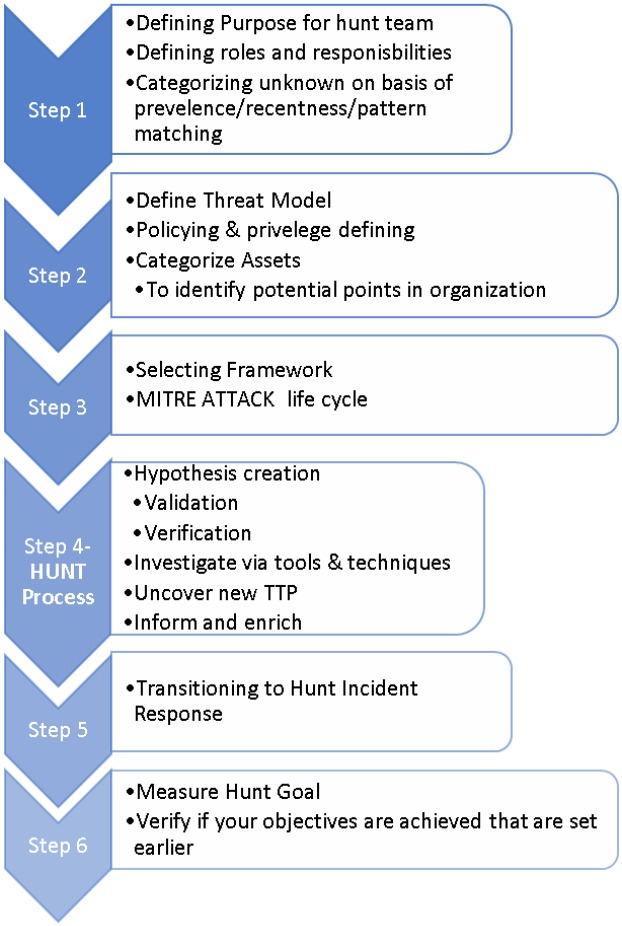

FIGURE 5. Threat hunting process from scratch.

sandbox are deployed on Node 3 and directly connected with switch on port 3. Inside node 1, there is a nested network windows machine running combined HMI with Sysmon and procmon installed on it. All 3 Nodes are connected with the

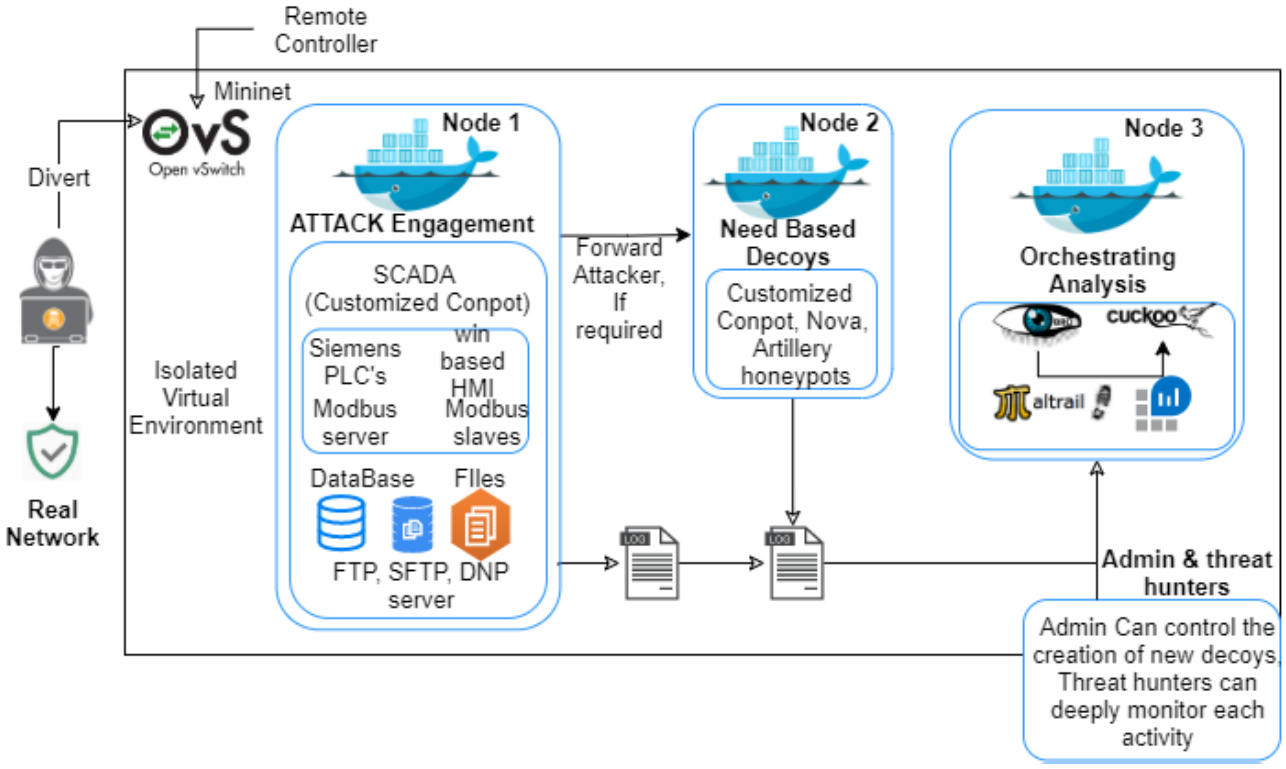

FIGURE 6. Approach for SCADA decoy farm.

base switch, and each node has a running docker container. The basic command we used to quickly deploy SDN (using mininet) inside node 2;

"mn –topo = single,5 –mac –controller remote –switch ovsk".

Following is the target environment details:

TABLE 3. Lab environment details.

Platform	OS	Clock Speed	RAM
Docker Node 1	Ubuntu	2.4Ghz	6GB
Docker Node 2	Ubuntu	2.8Ghz	6GB
Docker Node 3	Ubuntu	2.4Ghz	6GB
Attack Machine	Debian	2.4Ghz	6GB

2) STRATEGY

We customized the SCADA honeypot as it is mimicking a real PLC by modifying the XML template as shown in Figure 7 from the conput configuration directory. By default, the conpot can easily be detected by the attacker if it is a honeypot or not, as the fingerprint database has fingerprints of the conpot. For example, the Metasploit module can easily rate honeypot, and it can detect conpot as well. So we are customizing conpot fingerprint instincts, which will be different from its default values as shown in Figure 7. In this way, we can save decoys from detection. Figure 8 shows PLC scan result of conpot customized deployment. When an attacker scans or sees a whole vulnerable network, it will eventually divert attackers towards vulnerable devices. We are diverting attackers from the actual network into a decoy farm (simulated SCADA in SDN). Whole attack engagement is monitored from real-time logs, and logs will be transmitted using log beat in a structured form STIX2 (*Extracting TTP's from log files*: Using open-source https://github.com/vlegoy/rcATT tool) for attack hypothesis generation.

TABLE 4. Attack details on simulated environment.

Attacks	Mitigation
Stuxnet (Figure 10)	Divert & Neutralized
TriStation Malware [40]	Divert & Neutralized
Hatman Malware [41] [42]	Divert & Prevent
Headless http server	Divert & Executed
HMI [43]	Divert & Executed
HTTP DoS [44]	Divert & Neutralized
FTP anony login	Divert & Executed
DNP [45] & Modbus	Divert &Neutralized
Controller-attacks [46]	Dynamic rules [47] [48]

3) EVALUATION

Our objective for experiments is to evaluate our threat hunting approach and threat mitigation strategy for SCADA, and we are evaluating it against real-world simulated attacks. We executed some known APT threats with slight modifications in our simulated environment as present in Table 4; our simulated network is designed to react to every threat and entertain it within isolation to stop its propagation. During this process,

TABLE 5. Attack techniques used.

Techniques	Objective
Hide Artifacts [49]	Bypass detection
Hijack execution flow	Defense evasion
Non-standard C2 [50]	Custom C2
Non-Standard encoding [51]	Custom encoder [52]
Modified known trojans	Obfuscation
Modify Authentication process	Evade security
Process Injection	Execute binary
Securityd Memory T1555.002	Credential access

```
<modbus enabled="True" host="0.0.0.0" port="5020">
<device_info>
    <VendorName>Siemens</VendorName>
    <ProductCode>SIMATIC</ProductCode>
    <MajorMinorRevision>S7-200</MajorMinorRevision>
</device_info>
<mode>serial</mode>
<delay>100</delay>
<slaves>
    <slave id="0">
        <blocks>
        <block name="memoryModbusSlave0BlockA">
            <type>COILS</type>
            <starting_address>1</starting_address>
            <size>128</size>
            <content>memoryModbusSlave0BlockA</content>
            </block>
            <block name="memoryModbusSlave0BlockB">
            <type>DISCRETE_INPUTS</type>
            <starting_address>10001</starting_address>
```

FIGURE 7. Customizing MODBUS slaves in XML file.

threat hunters can record new attack patterns using static and dynamic networks and forensic artifacts analysis.

Following are the details (in Table 5) of different techniques used to evade defense mechanisms and execution of malicious binaries. Details related to adversary emulation are here [18].

B. ATTACK AND THREAT HUNTING

We scanned the network with namp (Aggressive scan, slow scan, ping scan, delay scan, T5 scan, syn scan). For the plc scan, we used an open-source plc scanner. NSE scripts in Nmap also provide the ability to scan MODBUS. By using Metasploit, we tried uploading HTTP shell on the headless HTTP server. We launched some known attacks as specified in Table 5 and 4. We used evasion techniques as specified in Table 4 to wrap and compile beacons, then upload them on the FTP server. One of our beacons successfully bypassed the firewall and IDS using malleable C2, and we got the shell. For example, one of the payloads we used from Metasploit was *Linux/x86/shell/reverse* as shown in Figure 9.

1) GATHERING INTELLIGENCE AND THREAT HUNTING

We got the presence of anomalies on the network from maltrail web portal as well as Zeek logs co-relation also gave threat hunters a clear view of the attack, using RITA (Real Intelligence Threat Analytics) [53]. Network scanning inside

```
Module                    :  v.0.0
Name of the PLC           : SIMATIC 200(1)
Name of the module        : Siemens, SIMATIC, S7-200
Plant identification      : Nexlinx
Copyright                 : Original Siemens Equipment
Serial number of module   : S C-F3U398112015
Module type name          : IM151-8 PN/DP CPU
OEM ID of a module        :
Location designation of a module:
```

FIGURE 8. Scanned result of PLC decoy.

```
msf5 > use exploit/multi/http/apache_mod_cgi_bash_env_exec
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RHOST [redacted]
RHOST => [redacted]
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set TARGETURI /
TARGETURI => /
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set PAYLOAD linux/x86/shell/reverse_tcp
PAYLOAD => linux/x86/shell/reverse_tcp
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set LHOST 172.20.16.125
LHOST => 172.20.16.125
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > show options
```

FIGURE 9. HTTP shell from metasploit.

SDN will cause decoys to generate logs, such as connect requests to FTP, TFTP, or PLC. These all logs are monitored to find potential threats which can lead to building hypotheses. After analysis, it yielded the presence of beacons. Aggregating incoming and outgoing traffic from Zeek bro gave us deep insight into the details of outgoing malicious connections. Our decoys (customized conpot) acted as IOC and helped threat hunters populate the CMF table; there were 47+ attacks on the HTTP and Modbus servers. No device is compromised as this whole network is just a simulation and running on a proxy layer between the actual system.

There is a "phenomenon" in which malware tries to contact the long-dead C&C server's domain, lacking any DNS resolution continuously in case of old malware presence on internal PCs of the organization. In this case, already lurking threats can be detected as well. Following is the initial intelligence Table 6, which is enough for building and validating the hypothesis. Many malware tries to access the victim's Internet IP address by using ipinfo requests. Thus each request was deeply analyzed in case of consistent occurrences.

In the situation in which heuristic mechanism is detected by different connection attempts to a substantial amount of various TCP ports, customized conpot and other windows (HMI) decoys triggered alert whenever they get a syn request. Further co-relating these logs with previously detected techniques, threat hunters can populate the collection management framework table. After successfully conducting threat hunting, hunters utilize a collection management framework to manage the data collected to be used in validation. Threat hunters consider the different dimensions of threats that are likely to happen or already exist. In Table 7 there are some artifacts which can be used for evidence collection at endpoints.

What do we achieve from this? Preventing attacks to our actual network, at the same time recording TTP from the actual adversary and uncovering new TTP. We were able to detect unknown threats from analyzing logs from decoys and other sensors of maltrail. Existing threats are also detected that were sending requests to dead C2 servers.

TABLE 6. Discovered techniques.

Technique	Procedure	Sub-technique
HTTP command injection	Pass in query	Shell Injection
Modbus Coil Modification	Used Metasploit	Coil scanning, Random data injection
AMSI Bypass	load files reflectively	String encoding, string obfuscation, load string & script from trusted source
Reverse Shell	Initial access	PowerCat
Protocol Encapsulation	C2	DNS over HTTPS
PLC Takeover	Manipulation	Session Hijacking
Dll Injection	Partial fileless	Reflective Injection
Traffic mainupulation	Positive c2	Malleable C2

TABLE 7. Endpoint artifacts for evidence collection.

Artifact Name	Details
Logs	HTTP, FTP, Modbus server logs
Attacker connected to system	Vssadmin events, VSS shadow copy event
LSASS dump	Check IOC using YARA
Proc dump	Registry keys, dump activity
Psexesvc	Shimcache results, psexescvc file in directory

TABLE 8. CMF for threat hunters.

Collection Management Framework (CMF)	Source	Source Detail
Location	Endpoint	Syetem T1
Kill chain step	Detection evasion (endpoint)	Process Injection
Data type	Raw log	Sysmon log
Collection method	Telemetry log sharing	Log beat
Storage duration	60 minutes	Completed 4 phase cycle

V. RESULTS

All known and unknown threats are detected, and their activities are recorded across different sensors. From initial scanning to code execution and payload downloading from C2 is recorded in our sensors. During ex-filtration, canary tokens generated alerts with their activity. Our Hunting approach decreased the duel time between attacker and defense mechanism.

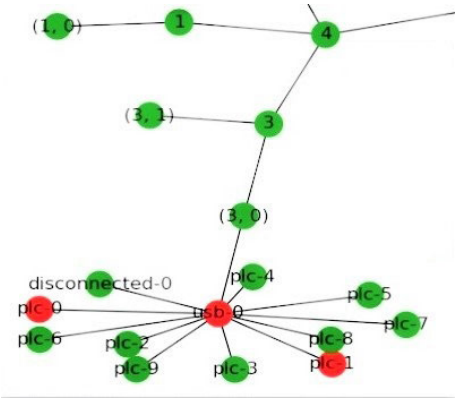

FIGURE 10. Stuxnet neutralized successfully and executed kill switch, Red: infected nodes.

We ran different variants of Stuxnet in our simulated environment; Siemens step7 was also installed on windows based system. Following are the results, kill switch triggered after 10 waves of execution and was not able to penetrate even from one node.

We tried the same attacks on the same network with a traditional AV/EDR, firewall (Palo Alto), and IDS/IPS (Zeek bro). In this case, security mode was reactive. Some malware and reverse connections were blocked by a firewall. So we tried attacks with unknown signatures with end-to-end encryption. It bypassed the firewall and IDS, and even windows defender was not able to detect any suspicious activity. Moreover, we tried endpoint attacks and was able to evade Symantec and Fortinet EDR. After that, we started pulling files slowly in multi-thread mode with DNS requests to see if IDS detects it. IDS was able to identify something happening not good but did not know what to do with this. We tried HTTPS beacon with impersonating certificates that attack was successful without giving an immediate indication. After analyzing logs from firewalls and IDS, the information from logs was not enough to build and support the hypothesis or decide whether it was a threat.

Above radar diagram, 11 defines what we are trying to prove by experiment. The time taken by traditional security measures to detect, identify and respond to threat approximately 60 percent less as compared to the threat hunting approach. We tried more than 30 different types of attacks that are already existed (from Metasploit). Remaining of the attacks are modified attacks and can easily bypass static and dynamic analysis. If we consider a scenario where all attacks which are filed, in such case firewall and IDS performs very well. But now, what we are dealing with, are file-less unknown attacks in such a scenario endpoint security, IDS and firewalls do not perform well. This comparison is with and without threat hunting approach in the same threat environment. The time taken by threat hunting approach to detect, identify, record, and respond 60 percent faster. As in hunting, we aggressively keep on searching for threats (known and

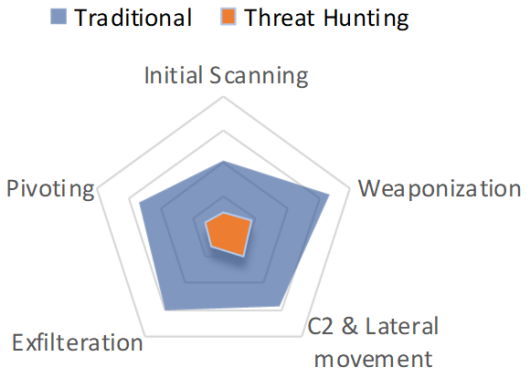

FIGURE 11. Detection time gap between both approaches.

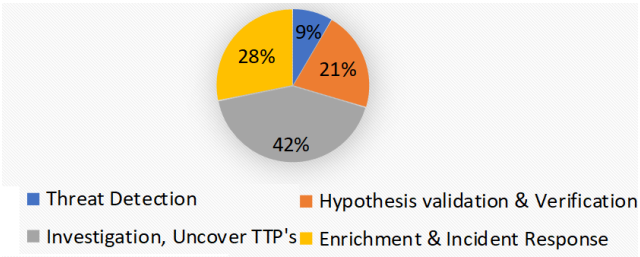

FIGURE 12. Division of each phase in whole process.

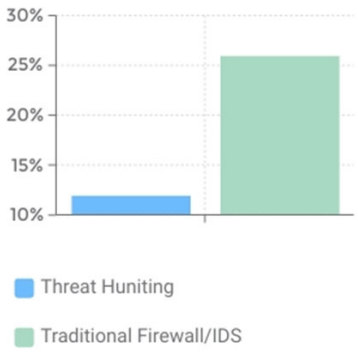

FIGURE 13. False positive percentage.

unknown) that are lurking in our network. Figure 12 shows the division of process and time from threat detection and mitigation. Threats are detected from sensors and manual searching of threats; once threats are detected, 21 percent of the time is utilized to verify and validate a hypothesis. The majority of time is consumed in investigating threats.

In traditional security mechanisms for file-less attacks, it is difficult to identify threats as they reside in RAM; integrating such attacks with a living of the land binaries can boost the evasion capabilities at endpoints. If the firewall/IDS or EDR is next-generation, it still takes more time to detect as it learns by time with behavior and updates its rules. The false-positive ratio is high in the case of file-less attacks. The below graph has drawn in the context of the experiment we did.

In case of an unknown attack, we tried fooling firewalls and EDR with the help of different approaches, such as mimicking legitimate traffic using Malleable C2, which caused an increase in false-positive for firewalls. False-positive do exist

TABLE 9. Discovered techniques mapped on ATT&CK.

Tactic	Technique
Defense Evasion	Obfuscated Information T1027
Discovery	Query Registry T1012, T1082
Execution	Shared Modules T1129, T1059.003
Exfiltration	Covert C2 Channel T1041
Persistence	Create or Modify Process T1543.003

in the threat hunting approach, but it is quite less. It occurred because of different stealthy techniques we employed during the attack. Eventually, this leads us to build a hypothesis on wrong assumptions. If we give more time to log analysis, different trails of false positives can be minimized. In Table 9 there are discovered techniques by threat hunting after adversary emulation, which are known to ATT&CK but with unknown sub-techniques.

VI. CONCLUSION

Due to the change of threat landscape, reactive approaches are ineffective in detecting and reacting in time, resulting in no detection or increasing duel time between incident response and attack. Proactive approaches in conjunction with deception and threat intelligence are an effective way of detecting and preventing threats quickly and using SCADA decoy farm to engage in attack and record its activity by providing IOC's to threat hunters. Hence we concluded that the threat detection ability of SCADA is increased using the threat hunting approach against real-world attacks as compared to traditional security mechanisms. For future directions, Our future work includes "Adversary simulation" on networks to mature our threat hunting teams with regular adversary exercises.

REFERENCES

[1] B. Miller and D. Rowe, "A survey SCADA of and critical infrastructure incidents," in *Proc. 1st Annu. Conf. Res. Inf. Technol.*, 2012, pp. 51–56.
[2] N. Ismail. (2017). *The Rise of Cybercrime Continues to Accelerate*. [Online]. Available: https://www.information-age.com/rise-cyber-crime-continues-accelerate-123467629/
[3] D. Upadhyay and S. Sampalli, "SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101666.
[4] J.-M. Lee and S. Hong, "Host-oriented approach to cyber security for the scada systems," in *Proc. 6th IEEE Congr. Inf. Sci. Technol. (CiSt)*, Jun. 2020, pp. 151–155.
[5] J.-M. Lee and S. Hong, "Keeping host sanity for security of the SCADA systems," *IEEE Access*, vol. 8, pp. 62954–62968, 2020.
[6] A. Bushby, "How deception can change cyber security defences," *Comput. Fraud Secur.*, vol. 2019, no. 1, pp. 12–14, Jan. 2019.
[7] K. Coffey, R. Smith, L. Maglaras, and H. Janicke, "Vulnerability analysis of network scanning on SCADA systems," *Secur. Commun. Netw.*, vol. 2018, pp. 1–21, Mar. 2018.
[8] T. Cruz, L. Rosa, J. Proença, L. Maglaras, M. Aubigny, L. Lev, and J. Jiang, "A cybersecurity detection framework for supervisory control and data acquisition systems," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2236–2246, Dec. 2016.

[9] S. Al-Rabiaah, "The 'Stuxnet' virus of 2010 as an example of a 'APT' and its 'Recent' variances," in *Proc. 21st Saudi Comput. Soc. Nat. Comput. Conf. (NCC)*, 2018, pp. 1–5.

[10] Z. Tian, W. Wu, S. Li, X. Li, Y. Sun, and Z. Chen, "A security model of SCADA system based on attack tree," in *Proc. IEEE 3rd Conf. Energy Internet Energy Syst. Integr. (EI2)*, Nov. 2019, pp. 2653–2658.

[11] A. O. Gomez Rivera and D. K. Tosh, "Towards security and privacy of SCADA systems through decentralized architecture," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2019, pp. 1224–1229.

[12] S. V. B. Rakas, M. D. Stojanovic, and J. D. Markovic-Petrovic, "A review of research work on network-based SCADA intrusion detection systems," *IEEE Access*, vol. 8, pp. 93083–93108, 2020.

[13] S. N. Mohan, G. Ravikumar, and M. Govindarasu, "Distributed intrusion detection system using semantic-based rules for SCADA in smart grid," in *Proc. IEEE/PES Transmiss. Distrib. Conf. Expo. (T&D)*, Oct. 2020, pp. 1–5.

[14] E. Biham, S. Bitan, A. Carmel, A. Dankner, U. Malin, and A. Wool, "Rogue7: Rogue engineering-station attacks on S7 Simatic PLCs," in *Proc. Black Hat USA*, Las Vegas, NV, USA, Jun. 2019.

[15] S. Sudhakaran, R. K. Shenoy, A. Sarada, and J. J. McCann, "VLAN to VXLAN translation using VLAN-aware virtual machines," U.S. Patent 10 476 699, Nov. 12, 2019.

[16] G. Tsochev, R. Yoshinov, and O. Iliev, "Key problems of the critical information infrastructure through scada systems research," *SPIIRAS Proc.*, vol. 18, no. 6, pp. 1333–1356, Nov. 2019.

[17] Q. Li, F. Wang, J. Wang, and W. Li, "LSTM-based SQL injection detection method for intelligent transportation system," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4182–4191, May 2019.

[18] A. B. Ajmal, M. A. Shah, C. Maple, M. N. Asghar, and S. U. Islam, "Offensive security: Towards proactive threat hunting via adversary emulation," *IEEE Access*, early access, Aug. 11, 2021, doi: 10.1109/ACCESS.2021.3104260.

[19] D. Karev, C. McCubbin, and R. Vaulin, "Cyber threat hunting through the use of an isolation forest," in *Proc. 18th Int. Conf. Comput. Syst. Technol.*, Jun. 2017, pp. 163–170, doi: 10.1145/3134302.3134319.

[20] S. Caltagirone, A. D. Pendergast, and C. Betz, "The diamond model of intrusion analysis," Diamond Model Intrusion Anal., Tech. Rep., Jul. 2013. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA586960

[21] X. Zhou, Z. Xu, L. Wang, K. Chen, C. Chen, and W. Zhang, "Kill chain for industrial control system," in *Proc. MATEC Web Conferences*, vol. 173, 2018, p. 1013.

[22] M. Parcharidis, "Simulation of cyber attacks against SCADA systems," M.S. thesis, Int. Hellenic Univ., Thermi, Greece, 2018. [Online]. Available: https://repository.ihu.edu.gr//xmlui/handle/11544/29209

[23] O. Masset and B. Taburiaux, "Simulating industrial control systems using mininet," M.S. thesis, Dept. Ecole Polytechnique de Louvain, Univ. Catholique de Louvain, Ottignies-Louvain-la-Neuve, Belgium, 2018. [Online]. Available: http://hdl.handle.net/2078.1/thesis:14706

[24] A. Bhardwaj and S. Goundar, "A framework for effective threat hunting," *Netw. Secur.*, vol. 2019, no. 6, pp. 15–19, Jun. 2019.

[25] D. Gunter and M. Seitz. (2018). *A Practical Model for Conducting Cyber Threat Hunting*. [Online]. Available: https://www.sans.org/reading-room/whitepapers/threathunting/paper/38710

[26] *Emulator for Rapid Prototyping of Software Defined Networks*. Accessed: Feb. 2, 2021. [Online]. Available: https://github.com/mininet/mininet

[27] P. Raghav and A. Dua, "Enhancing flow security in ryu controller through set operations," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2017, pp. 1265–1269.

[28] Github. *ICS/SCADA Honeypot*. Accessed: Jan. 13, 2021. [Online]. Available: https://github.com/mushorg/conpot

[29] Activecm. *Activecmrita: Real Intelligence Threat Analytics (Rita) is a Framework for Detecting Command and Control Communication Through Network Traffic Analysis*. Accessed: Mar. 23, 2021. [Online]. Available: https://github.com/activecm/rita

[30] *Malicious Traffic Detection System*. Accessed: Feb. 2, 2021. [Online]. Available: https://github.com/stamparm/maltrail

[31] Ponemon Institue. (2016). *Cost of Data Breach Study, Global Analysis*.[Online]. Available: https://www.datasheetarchive.com

[32] P. Bright, "Microsoft terminates its Tay AI chatbot after she turns into a Nazi," *Ars Technica*, vol. 24, pp. 110–116, Mar. 2016.

[33] T. Kakumaru, "Launching threat hunting from almost nothing," SANS Threat Hunting & IR Summit, London, U.K., Tech. Rep. 1536354142, 2018. [Online]. Available: https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1536354143.pdf

[34] Y. Awad, M. Nassar, and H. Safa, "Modeling malware as a language," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6, doi: 10.1109/ICC.2018.8422083.

[35] *zeek Bro, Network Threat Hunting Utility*. Accessed: Feb. 2, 2021. [Online]. Available: https://www.zeek.org/

[36] B. H. Klimkowski, "Using bro to hunt persistent threats," United States Mil. Acad., Sep. 2017. Accessed: Dec. 4, 2020. [Online]. Available: https://old.zeek.org/brocon2017/slides/persistent_threats.pdf

[37] P. E. Devon Kerr. (2018). *The Endgame Guide to Threat Hunting Practitioners*. [Online]. Available: https://cyberforensicator.com/2018/06/05/the-endgame-guide-to-threat-hunting-practitioners-edition/

[38] P. Mundas. (2019). *Core of Threat Hunting*. [Online]. Available: https://www.peerlyst.com/posts/core-of-threat-hunting-prasanna-b-mundas

[39] R. Brown and R. M. Lee, "The evolution of cyber threat intelligence (CTI): 2019 SANS CTI survey," SANS Inst., Bethesda, MD, USA, Tech. Rep., Feb. 2019.

[40] S. Mansfield-Devine, "Critical infrastructure: Understanding the threat," *Comput. Fraud Secur.*, vol. 2018, no. 7, pp. 16–20, Jul. 2018.

[41] A. Di Pinto, Y. Dragoni, and A. Carcano, "TRITON: The first ICS cyber attack on safety instrument systems," in *Proc. Black Hat USA*, 2018, pp. 1–26.

[42] K. Hemsley and R. Fisher, "A history of cyber incidents and threats involving industrial control systems," in *Int. Conf. Crit. Infrastruct. Protection*. Cham, Switzerland: Springer, 2018, pp. 215–242, doi: 10.1007/978-3-030-04537-1_12.

[43] A. Kleinmann, O. Amichay, A. Wool, D. Tenenbaum, O. Bar, and L. Lev, "Stealthy deception attacks against SCADA systems," in *Computing Security*. Cham, Switzerland: Springer, 2017, pp. 93–109, doi: 10.1007/978-3-319-72817-9_7.

[44] R. R. Zebari, S. R. M. Zeebaree, and K. Jacksi, "Impact analysis of HTTP and SYN flood DDoS attacks on apache 2 and IIS 10.0 web servers," in *Proc. Int. Conf. Adv. Sci. Eng. (ICOASE)*, Oct. 2018, pp. 156–161.

[45] I. Darwish and T. Saadawi, "Attack detection and mitigation techniques in industrial control system-smart grid DNP3," in *Proc. 1st Int. Conf. Data Intell. Secur. (ICDIS)*, Apr. 2018, pp. 131–134.

[46] T.-H. Nguyen and M. Yoo, "Analysis of link discovery service attacks in SDN controller," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2017, pp. 259–261.

[47] S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in IoT using SDN," in *Proc. 27th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2017, pp. 1–6.

[48] D. Santos, A. de Sousa, and C. M. Machuca, "Robust SDN controller placement to malicious node attacks," in *Proc. 21st Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, Feb. 2018, pp. 1–8.

[49] R. Colelli, S. Panzieri, and F. Pascucci, "Exploiting system model for securing CPS: The anomaly based IDS perspective," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2018, pp. 1171–1174.

[50] K. Nagendran, S. Balaji, B. A. Raj, P. Chanthrika, and R. G. Amirthaa, "Web application firewall evasion techniques," in *Proc. 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2020, pp. 194–199.

[51] K. Monnappa, *Learning Malware Analysis: Explore the Concepts, Tools, and Techniques to Analyze and Investigate Windows Malware*. Birmingham, U.K.: Packt, 2018.

[52] R. Mainieri and C. A. Hastings, "Methods and systems for encoding computer processes for malware detection," U.S. Patent 9 860 262, Jan. 2, 2018.

[53] *Activecm: Real Intelligence Threat Analytics (rita) is a Framework for Detecting Command and Control Communication Through Network Traffic Analysis*. Accessed: Feb. 2, 2021. [Online]. Available: https://github.com/activecm/rita

ABDUL BASIT AJMAL is currently pursuing the master's degree in information security with COMSATS University Islamabad, Pakistan. His current research interests include securing industrial grade systems, threat hunting, threat replication, adversary simulation, and risk assessment.

MASOOM ALAM received the Ph.D. degree in computer sciences from the University of Innsbruck, Austria. He is currently an Associate Professor with the Department of Computer Sciences, COMSATS Institute of IT, Islamabad, Pakistan. His research interests include access control systems, model-driven architecture, and workflow management systems.

AWAIS ABDUL KHALIQ received the B.Sc. degree in information security from The University of Azad Jammu & Kashmir, Muzaffrabad, Pakistan, in 2014. He is currently pursuing the M.S. degree in information security with COMSATS University Islamabad, Pakistan. His research interest includes data privacy in smart city.

SHAWAL KHAN received the bachelor's degree in computer science from Shaheed BB University, Upper Dir, Khyber Pakhtunkhwa, Pakistan. He is currently pursuing the master's degree in information security with COMSATS Institute of IT, Islamabad, Pakistan. His research interests include access control, cryptography, and network security.

ZAKRIA QADIR received the M.Sc. degree in sustainable environment and energy systems from Middle East Technical University, Turkey, in 2019. He is currently pursuing the Ph.D. degree in wireless communication and cloud computing with Western Sydney University, Australia. His research interests include sustainable cities, artificial intelligence, machine learning, optimization techniques, wireless communication, the IoT, renewable energy technology, and cloud computing.

M. A. PARVEZ MAHMUD received the B.Sc. degree in electrical and electronic engineering, the M.Eng. degree in mechatronics engineering, and the Ph.D. degree. After the successful completion of his Ph.D. degree with multiple awards, he worked as a Postdoctoral Research Associate and an Academic with the School of Engineering, Macquarie University, Sydney, NSW, Australia. He is currently Alfred Deakin Postdoctoral Research Fellow with Deakin University, where he is also involved in the supervision of eight Ph.D. students. He is also a Key Member of Deakin University's Advanced Integrated Microsystems (AIM) Research Group. Apart from this, he is actively involved with different professional organizations, including Engineers Australia and IEEE.