A Novel IDS Securing Industrial Control System of Critical Infrastructure Using Deception Technology

Shaobo Zhang, Beijing Institute of Technology, China

Yuhang Liu, Peking University, China

Dequan Yang, Beijing Institute of Technology, China*

https://orcid.org/0000-0002-2551-1945

ABSTRACT

The industrial control system (ICS) has become the key concept in the modern industrial world, enabling process monitoring and system control for general industrial systems and critical infrastructures. High-skilled hackers can invade an imperfect ICS by existing vulnerabilities without much effort. Conventional defenses (such as encryption and firewall) to keep invaders away are getting less effective when an attack is carried out by exploiting an array of particular vulnerabilities. Under this circumstance, a new-type intrusion detection system (IDS) based on deception strategy using honeypot technique is proposed, which is of dramatic effectiveness in protecting ICSs of critical infrastructures. In this honeypot-based model, the authors capture malicious internet flows and system operations. They analyze the collected data before alerting and preventing the intrusion alike when it affects the system in the future. This paper deals with the model's concept, architecture, deployment, and what else can be achieved in the field of critical infrastructure cybersecurity (CIC).

KEYWORDS

Critical Infrastructure, Honeypot, Industrial Control System, Intrusion Detection System

INTRODUCTION

Over the past decade, various modern technologies such as the Internet of Things (IoT), Big Data, and Cloud Computing have been terrifically advanced. These significant improvements bring abundant opportunities for industry development and become essential drivers of innovation in industry. As a result, a new industry concept has emerged, the Fourth Industrial Revolution (Industry 4.0) (Schwab, 2017).

As the so-called "Fourth Industrial Revolution" evolving further, industry today has become more intellectual and connective than any other era in history. As a result, Industrial Control Systems (ICSs), designed to focus on system functions rather than the Internet connection and remote distribution, are now migrating from their original isolated networks (usually LANs) to some public environment, such as the Internet. By utilizing the powerful ability of interconnection, ICSs, including Supervisory Control and Data Acquisition Systems (SCADAs) (Gaushell & Darlington, 1987), Distributed Control System (DCS), and other control system configurations like Programmable Logic Controllers (PLC)

(Stouffer et al., 2011), can conduct remote control and instant supervision of the target industrial systems nowadays.

Unfortunately, a variety of cybersecurity challenges has been emerging due to exposing of vital services of ICSs. (Ani et al., 2017) Tight connection of devices and components of ICSs results in high risk in security. Communication of devices perennially exchange vast quantity of safety-critical data through the open air and constantly attract various types of attack. Attackers may manipulate the whole industrial Internet by exploiting vulnerabilities in front-end equipment and sensors, communication networks, and back-end of IT systems (Kumar & Patel, 2014). Once attackers gain the whole or partial access control privileges of the system, there is no doubt that informational and economic loss will be immeasurable. Even people's life security will be in great jeopardy in some severe cases. On December 23, 2015, the Ukrainian Kyivoblenergo, a regional electricity distribution company in Ukraine, reported customer service outages. The outages were due to a third party's illegal entry into the company's computer and SCADA systems: some crucial substations were disconnected for three hours. Later statements indicated that the cyberattack impacted additional portions of the distribution grid and forced operators to switch to manual mode. It is said that a foreign attacker remotely controlled the SCADA distribution management system. The outages were initially thought to have affected approximately 80,000 customers. However, later it was revealed that three different distribution companies were attacked, resulting in several outages that caused approximately 225,000 customers to lose power across various areas. (Case, 2016) A typical domino effect of attacks on ICSs has been seen in the event. (Arief et al., 2020)

Intrusion detection systems (IDSs) are software or hardware systems that automate the process of monitoring the events occurring in a computer system or network, analyzing them for signs of security problems. As cybersecurity issues have become the primary concern of ICSs, IDSs have become a necessary addition to the security infrastructure of most organizations (Bace & Mell, 2001). Intrusion detection methodologies are classified into three major categories: Signature-based Detection (SD), Anomaly-based Detection (AD), and Stateful Protocol Analysis (SPA). Every one of these types is utilized in different situations. Using the IDSs, the authors can make monitoring the whole security system a much more relaxing work with higher efficiency. In the industrial Internet field, the use of IDSs is commonplace owing to their high quality and low cost.

Along with IDS, the honeypot is another helpful method to perceive unknown attacks and intrusions. As the honeypot is commonly defined as "an information system resource whose value lies in unauthorized or illicit use of that resource", the honeypot is the security resource deception frame up to act like a decoy whose importance resides in getting probed, attacked, or compromised. It contains no sensitive data; however, it pretends to be a valuable portion of the network (Oza et al., 2019). With the help of the deception function of the honeypot, IDSs can be used to collect valuable data from zero-day attacks and other unknown malicious actions in SCADA systems or DCSs. Thus, the authors may be about to stop attacks alike outside our critical infrastructures next time.

To avoid ICS being intruded and make it have confidentiality, availability, non-repudiation, identification, integrity, and logging specifications, necessary actions must be done. As ICS and other industrial Internet is a resource-constrained communication network which largely relies on low-bandwidth channels for communication among lightweight devices regarding CPU, memory, and energy consumption (Heer et al., 2011), traditional security mechanisms such as secure protocols (Gubbi et al., 2013), lightweight cryptography (Cole & Ranasinghe, 2008) and privacy assurance (Pöhls et al., 2014) are no longer suitable to protect the complicated industrial Internet.

In this study, a novel IDS using deception technology solution for ICS and critical infrastructures is proposed to solve the difficulties listed before. The authors deploy a self-developed honeypot system onto several Industrial Control Computers to simulate an actual ICS environment and expose the experiment system to the external Internet. With this well-designed honeypot, the authors successfully collect a large quantity of malicious data generated by different attackers. The system retains the collected data safely and then analyzes it. For our system administrators, they can easily view the

health status of the whole system. Furthermore, the authors can identify existing attacks and prevent future attacks by utilizing analyzed data.

The rest of this paper is organized as follows: the authors will roughly introduce the background information of this field, including relative vulnerabilities and corresponding solutions, and some related work done by previous researchers in BACKGROUND. Then, in METHODOLOGY AND APPLICATION, our system's overall design and architecture will be presented in detail. Then, the authors are about to conduct an experiment simulating a realistic industrial network environment to test this system in EXPERIMENT before drawing the conclusion and making a future decision in CONCLUSION.

BACKGROUND

This section will present background information about ICSs' architecture of today's critical infrastructures and security vulnerabilities they are facing. The authors will also discuss essential solutions to these security issues. Finally, some related work using deception and other cybersecurity strategies which inspired us to construct our deception-based IDS is reviewed at the end of the section.

MODERN ICS AND SCADA SYSTEM

A typical ICS with a SCADA system is also combined with several different devices. One of these devices is Supervisory Computer which is responsible for communicating with field controllers and devices by sending commands to these devices and fetching necessary status information back to the ICS. (Turner, 1974) PLCs provide basic logic interfaces between SCADA systems and sublayer physical devices. (Bolton, 2015) Similarly, Remote Terminal Units (RTUs) can aggregate scattered data from all sensors and field devices. Meanwhile, a Human Machine Interface (HMI) using a Graphical User Interface (GUI) can help administrators interact with industrial hardware directly and efficiently.

As a crucial part of critical infrastructures, modern ICSs are widely used in critical industry fields such as electrical, water and wastewater, oil and natural gas, chemical, transportation, pharmaceutical, pulp and paper, food and beverage, and manufacturing. (Gaushell & Darlington, 1987) To achieve the goal of controlling and monitoring these industrial systems, a standard ICS usually consists of different types of components and controllers, which can ensure the proper operation of the whole industrial system. (Gaushell & Darlington, 1987) To be specific, most modern ICSs are designed and deployed with SCADA systems to enable functions of data acquisition and monitoring the production environment. Compared with DCSs, SCADEs can help administrators of critical industrial systems to control their production sites with a centralized control terminal in both isolated and remote locations.

In addition, aiming at establishing robust, covert, and safe infrastructures, many SCADA systems have historically used several kinds of special-designed and private communication protocols to transfer data between devices. Using different protocols guarantees isolation and protection among different systems. Distributed Network Protocol (DNP) designed for automation and control (Segall, 1983), Serial Modbus protocol, and Inter-control Center Protocol are three representative protocols used in numerous SCADA system implementations. However, against the initial goal of implementing unique protocols, standardization of these private protocols is becoming the mainstream in recent studies and development. (Li et al., 2012)

CYBERSECURITY VULNERABILITIES

With the rapid development of modern information systems and cutting-edge technologies, more and more ICSs have been upgraded from the legacy proprietary and stand-alone architecture to a transparent and standardized environment. However, because of the tight interconnection with public networks, many vulnerabilities and security issues have emerged to affect the holistic stability and availability of industrial systems. According to (Alcaraz, & Zeadally, 2013), attackers, including state-sponsored attackers involved in cyber espionage activities and attackers driven by religious

and political beliefs, can invade or break down an ICS. The vulnerabilities attackers exploit usually lies in these related fields:

- Legacy components and layering architecture: As described in previous sections, many ICSs have been operating for several decades before upgrading in the production environment, full of old and legacy equipment and components. These out-of-date components are always too weak to resist external attacks, leading to the breakdown of the entire system. Additionally, the layering design of ICS architecture makes extending and implementing an ICS in various scenarios an easy job, but in the meantime, once one of the layers is torn down by attackers, the whole system is out of control, no matter how robust the other layers are. In other words, the attack surface is expended due to the limitation of the existing system and architecture.
- HMIs, crucial servers, and historical databases: The access control (authentication) is not always effective and appropriate. Traditional credentials like the "username & password" tuple are no longer unbreakable. For example, attackers may use social engineering tricks (Hadnagy, 2010) or just conduct a brute-force attack to get the "key to the treasure cave". Additionally, as most credentials are stored in online databases, crucial data is facing great threats without sufficient protection like SSL encryption and VPN technology.
- Existing defense software and hardware: Although defense software like firewalls, IDS, and IPS has been already installed in many ICSs, the lack of accurate and strict configuration may still lead to leaks of sensitive data and intrusion of the system. As most IDSs using the Signature-based Detection (SD) mode, it is of incredible difficulty to generate proper rules to cover the whole ICS and SCADA, and many defense software only protects external attacks while internal ones are sometimes omitted. (Matoušek et al., 2020) Furthermore, proprietary and custom protocols used in the communication process among devices will also make detecting malicious network flows a more challenging task.
- Field devices and embedded systems: Field devices are usually designed just to meet their purpose's demand, so there is no additional computing power for protecting themselves. In most critical infrastructures, terminal industrial devices like RTUs and PLCs are not designed to be authentication-sensitive, which means access to these devices asks for weak or even no authentication. Besides, these kinds of devices are also vulnerable to DoS attacks, so that attackers can easily make sensor nodes and process controllers go offline by conducting a DoS or even DDoS.
- Education level and security awareness: Although cybersecurity is becoming the first priority in both educational and productional environments, unfortunately, most students and even professionals are not able to access proper education resources or are not aware of securing the cyber systems. Besides, even if sometimes educational training is conducted, the efficiency and outcome of the specific training method is hard to evaluate, and the general agreement of best practice of cybersecurity training has never been reached. (Chowdhury & Gkioulos, 2021)

BASIC SOLUTIONS

Since more and more vulnerabilities were found by commercial organizations and research groups, a wide range of solutions and theories have been developed to tackle this worsening situation. According to (Asghar et al., 2019), methodologies of cybersecurity solutions of ICSs and critical infrastructure can be classified into three main directions: security evaluation tools, intrusion detection and prevention technologies, ICS risk management. Using security evaluation tools that can efficiently and effectively simulate the actual control systems and field devices in an isolated and virtualized environment makes analyze the typical ICS, test its robustness and dig out potential flaws much more conveniently and safely. By comparison, detection and prevention techniques both focus on protecting running ICSs from malicious behavior in the production environment. In addition, cryptography

theory is also utilized to establish credible communication channels where only privileged entities can understand messages flowing through. Unlike the other two methods, ICS risk management, including access control management and risk management, provides a methodology and guideline that sets standards for securing the ICS environment. Furthermore, a variety of security metrics are practical measurements that can present the security level the system has achieved.

To be specific, many researchers have already designed and implemented defense frameworks to address crucial cybersecurity challenges in ICSs and critical infrastructure. This existing work inspires us with its outstanding innovation and novel strategies.

Kuipers and Fabro (2006) not only describe the risks that most contemporary control system architectures are faced with but also propose a unique defense strategy called "Defense-in-Depth". In the strategy, companies that need to maintain various field devices or industrial-level process systems, access to facilities via remote connection, and provide public network services for customers and partners can divide their network architecture into a multi-level (or multi-zone) one. Each of the levels just needs to focus on specific security concerns with specific defense strategy in-depth, which offers administrators more opportunities for information and resource control and introduces cascading countermeasures that will not impose a negative effect on business services' operation.

The Software Defined Network (SDN) and Network Function Virtualization (NFV) are used by researchers to design an automatic incident-response mechanism for ICSs. (Piedrahita et al., 2017) This combination of SDNs and NFV is a promising solution for critical infrastructure with the "Defense-in-Depth" strategy. In their design, the system will automatically reconfigure the network by rerouting the malicious traffic to an ICS honeypot in which administrators are able to supervise the behavior of attackers. In contrast, attackers cannot obtain any valuable information about the system. Moreover, as the hybrid architecture of SDN and NFV can implement different detection technologies, this system can be easily used in different attack scenarios. As a result, the real processes and services are protected, and the impact of attacks is mitigated.

DECEPTION TECHNOLOGIES IN ICS

Compared with traditional cybersecurity strategies like intrusion detection and prevention, deception is a much more active method to fight against hidden attackers. (Denning, 2014) Generally speaking, almost all the different deception models attempt to confuse the attackers and stop them from accessing the real systems by offering invalid information to attackers or leading attackers to some traps. According to (Pawlick et al., 2019), standard taxonomy of defensive can be presented as follows:

- Perturbation: This method can limit leakage of sensitive information by combining the noise with true data, making attackers hard to understand the meaning of messages.
- Moving Target Defense: With the help of randomization and reconfiguration of the networks, assets, and defense tools, this type of deception can limit the effectiveness and value of reconnaissance conducted by attackers before the actual attack.
- Obfuscation: By directing attackers to decoy targets instead of real ones, the obfuscation method is helpful to waste attackers' time and resources in order to protect valuable information.
- Mixing: This strategy focuses on the "isolation" of interconnected parties. Building mixed networks and mixed zones can reduce linkability, which will make recognizing the actual topology of the overall network a tough task.
- Honey-x: Administrators can draw attackers toward specific honeypot hosts or networks by disguising these systems as valuable network resources. Honeypot technology is of excellent efficiency to obtain attacker's information while protecting the real assets.
- Attacker Engagement: The feedback technique is used to influence attackers over an extended period dynamically. By doing so, not only are attacker's time and resources wasted, but also, administrator can gather intelligence about them.

In our novel design, the authors utilize the honeypot technology to achieve our goal of detecting and alerting attacks occurring in ICS and critical infrastructure environment.

METHODOLOGY AND APPLICATION

Inspired by previous work, the authors design and build our own IDS architecture which equips several special features aiming at overcoming challenges in the ICS environment:

- C/S Model: The whole system is running based on the Client/Server (C/S) model, which means the client hosts unnecessarily need a high level of computing power, and most of the computation-intensive job is assigned to the central server host.
- Lightweight and Dockerlized: All of the software utilized in this system is lightweight and requires little hard disk space, RAM capacity, and CPU power. Additionally, benefited from the advantages of Docker technology, deploying this system is not a demanding task.
- IPv6 Supported: As the IPv4 addresses have been exhausted, more and more companies and products turn to use IPv6 addresses to transport information. To meet the needs of using IPv6 addresses in ICSs, the authors design and construct our system with IPv6-supported techniques from the very first.

As this novel architecture of honeypot system in the environment of ICSs and critical infrastructure, all the related software is organized to adapt to this complex application scenario. As shown in Figure 1, the whole system can be divided into three parts:

- Honeypots: Specially designed honeypots with many ports and services opened to attract the attackers will monitor and collect attackers' information when they conduct attacks. After collecting valuable, honeypots can transmit data (including system logs and system call information) to the core server instantly.
- Core Server: Receive data from honeypots, processing and saving data, and provides system administrators with a series of tools to manage honeypot information and make further decisions.
- Monitoring & Analysis: Check whether honeypot is attacked and analyze the attacker's malicious behavior with the processed data captured by honeypots.

Figure 1. Overview of the proposed architecture
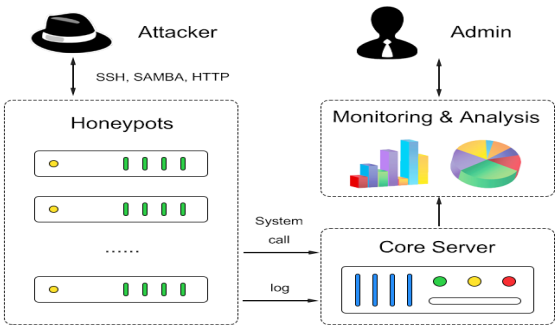
As a result of the fact that the IPv6 address space is so vast that it is hard for intruders to compromise other hosts after dominating the honeypot, Because the high-interaction honeypot can provide more details about the malicious behavior of attackers, it is more appropriate to place honeypot as high-interaction. However, the intruder might detect the core server by capturing the IPv6 packets transmitted between the honeypot and the core server, so it is recommended to deploy related applications in Docker containers to minimize the risks. In addition, this architecture does not identify attacks by identifying the characteristics of behaviors; as a result, it is useful for arbitrary attacks, including those against 0-day vulnerabilities. More details about how to deploy this system and how this system works are as shown below, divided into three relating parts.

HONEYPOTS

Multiple honeypots are supported in this architecture. Each honeypot consists of two parts (Decoy Service, Data Collection), as shown in Figure 2. The operating system of the honeypot must be Linux since some of the related applications can just run in Linux-based operating systems.

- Decoy Service: It consists of a series of vulnerable or seemingly valuable processes, which are mainly used to attract attackers to attack. It is highly recommended to deploy some applications that will be used as common servers. Deploying them in the Docker container is also a good choice since containerized deployment is becoming more and more widespread.
- Data Collection: By analyzing the relevant logs, including the syslog and the logs of the program with vulnerabilities, administrator can quickly find out the abnormal behaviors in the honeypot and preliminarily analyze these malicious behaviors. Moreover, the collection of system call events is beneficial to restore malicious behaviors. Here is the related software:
  - Filebeat: The Filebeat application can collect logs and transfer them to the Elasticsearch application, and the logs will be processed and stored.
  - Falco: Falco is a behavioral activity monitor designed to detect anomalous activity in applications. It can not only detect some operations that should not occur in the honeypot, such as writing below some sensitive directories like "/root" and network activity of system processes but also especially support Docker container. Furthermore, Falco provides a set of predefined abnormal behavior monitoring rules, and it also supports necessary customization. All Falco alerts will be sent via syslog.
  - Sysdig: Sysdig is a universal system visibility tool with native support for containers. In this architecture, Sysdig is mainly used to collect the system calls information and find the critical information related to the attack from the sea of system calls through a newly developed tool, which will be discussed in the section "Monitoring & Analysis". Sysdig outputs system calls line by line as events, so it is necessary to configure it to print data buffers in base64. Unnecessary processes need to be shut down to avoid too much data from Sysdig. All captures will be placed in a specific folder and mounted to an exclusive directory of the core server through the SMB protocol.

Figure 2. Detailed honeypot's structure
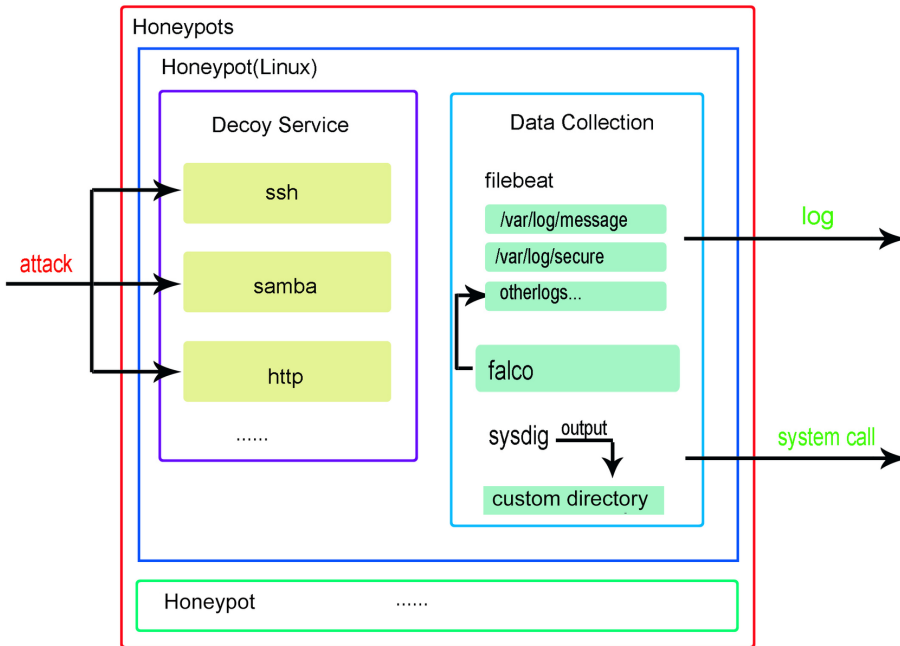
High-interaction honeypot does not expect to be accessed by non-malicious users, so even if a large number of services are running to disguise the honeypot as a regular server, it will not generate a large number of irrelevant logs and system calls to drown out the malicious behavior. Furthermore, once the system is intruded on, it is easy to separate these particular contents from the logs.

The attacker may destroy the data collection after getting into the honeypot. Therefore, to ensure the security of the core server, configuring security in Elasticsearch running on the core server is very necessary, such as restricting Filebeat to document creation only. Besides, the collected data should be processed and stored immediately to minimize possible data loss owing to network failure or other potential attacks.

CORE SERVER

The primary role of the core server is to process data from honeypots and provide Web service for the system administrators to monitor and analyze the attacker's behavior. The core server must use Linux OSs to support the software required for this architecture. Figure 3 shows that the data processing of the core server can be divided into two streams.

- Data stream for monitoring: The principal purpose of monitoring is to discover the intrusion immediately and provide preliminary judgment so it deals with logs.
- In this architecture, Filebeat sends the transactions directly to Elasticsearch by using the Elasticsearch HTTP API. Elasticsearch uses the "pipeline" (a definition of a series of processors to process the received raw data) installed by Filebeat to processing logs of syslog format, and

it can also be modified through the Elasticsearch API. It is necessary to modify the pipeline to parse JSON because the log of Falco contains some of it. In addition, Elasticsearch is a distributed document storage that stores complex data structures serialized as JSON documents.

- Kibana is an open-source analytics and visualization platform which uses Elasticsearch as a data source and provides a set of tools to visualize logs. Since it can be easily operated through the web browser, administrators can use Kibana to visually monitor the honeypot with some dashboards and powerful management tools.

- Data stream for analysis: Monitoring does not provide sufficient details about malicious operations. For example, it can only tell the system administrator that the attacker has created a file but cannot recover the file data. Therefore, the main purpose of this part is to provide the administrator with two essential functions named "Shell Capture" and "Trace File" for detailed analysis. As stated previously, all the system call information captured by honeypot will be attached to the corresponding folder on the core server through the SMB protocol. In addition, a novel PHP CLI script named "sysdigscanner" has been implemented, which scans all captures every 10s, and Sysdig is used as a tool to process these captures, which are modified during the interval, to separate shell commands and file operations. These data will eventually be put into MySQL. HPViewer is a novel web application developed with PHP programming language, which is built based on Docker image "mattrayner/lamp" (the base image along with a LAMP stack (Linux, Apache, MySQL, and PHP) (Lawton, 2005) all in one handy package). The administrator can use HPViewer through web browsers to analyze the behavior of attackers in detail. Shell commands and the basic information of read/write system call are retrieved from MySQL, updated by sysdigscanner every 10s while the file read/write data comes from the source file captured by sysdig in honeypots.

In this architecture, once anomalous behavior occurs in the honeypot, monitoring data can be observed instantly, and the administrator can analyze it within 10s. Moreover, based on the fact that Sysdigscanner only processes the modified captures, to improve the processing efficiency, it is recommended to adjust the command line flag of Sysdig to increase the maximum number of captures and reduce the size of a single one.

Using Docker technology simplifies the deployment of the core server greatly. As HPViewer can be created as a layer from the "mattrayner/lamp" Docker image, to make it work, just mount the folder which stores the Sysdig captures into the container by Docker volume. Besides, the official images of Elasticsearch and Kibana can be pulled from Docker Hub and deployed directly.

Figure 3. Detailed core server structure. "Sysdigscanner" is a scheduled PHP CLI script. "HPViewer" (HP means honeypot) is a web application based on LAMP.
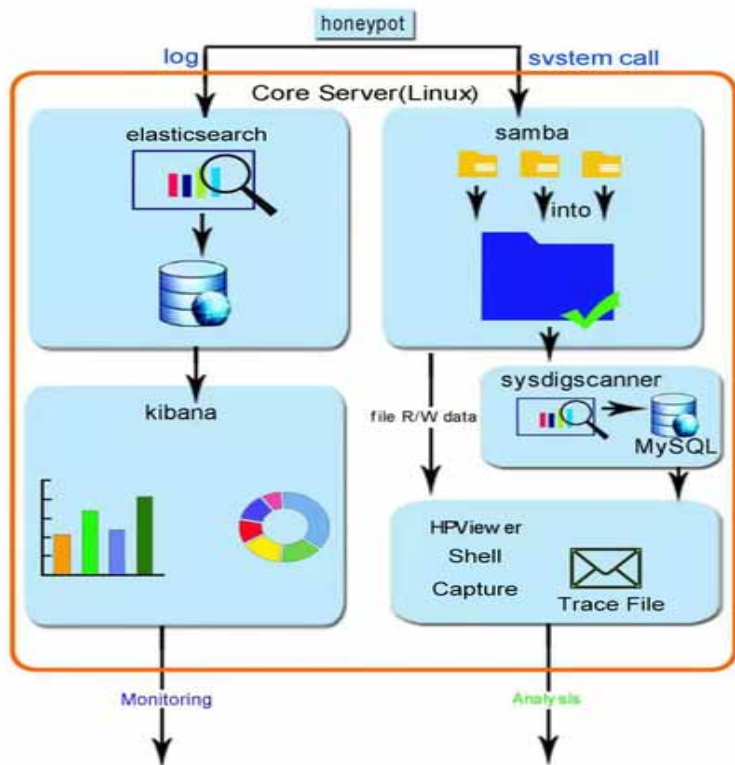
MONITORING & ANALYSIS

This part provides the administrator with two graphic user interface (GUI) platforms, one for monitoring honeypots, another for analyzing malicious behaviors detected in honeypots.

With many options for creating charts and dashboards to showcase data, Kibana is used for monitoring. Some dashboards required for this architecture have been added by Filebeat through Kibana HTTP APIs, such as "SSH logins". Besides, "Falco logs" and "Falco specific monitor" are two newly created dashboards using Kibana visualization tools, which provide more comprehensive monitoring of honeypots. Table 1 shows the important charts and the dashboards, respectively. "SSH login attempts" displays SSH login requests as a date histogram (splits a date field into buckets by interval and counts requests of each bucket), and different colors represent different types of login responses. "SSH users of failed login attempts" shows which user name appears the most in those failed logins (the larger the font size is, the more it was tried). "Syslog events hostnames and program" shows the number distribution of syslogs generated by each program of each hostname as a double doughnut chart. "Falco log events by hostname" clarifies which honeypots have suspicious events by displaying the Falco logs of each host as a date histogram. "Falco log events by container ids" is designed to provide monitoring for Docker containers. "Falco log of event rules" shows which rule is triggered the most. "Falco specific monitor" provides monitoring for rules of particular concern, such as starting a shell within a container. Moreover, the administrator can use the filter provided by Kibana to filter out the logs of interest and view the log details in log data tables.

A web application named HPViewer was built in order to analyze malicious behaviors. For this purpose, it has three menus in its interface. These menus are "Overview" that can manage (view, download, delete) the captures, "Shell Capture" that can examine captured shell commands, and "Trace File", where the reading or writing data of the specified file can be downloaded. Figure 4 shows the role of each part of the "shell capture" page; honeypot can be selected in the left list; shell commands are displayed in a tree structure, with the most recent shell placed at the top; once "sysdigscanner" finishes processing the data and inserts it into MySQL, the administrator can see the latest commands in this page. The page for tracking the file read/write operations in the honeypot is shown in Figure 5. To begin with, specify the hostname, the program that uses one of these system calls and the substring of file path to search for valuable files, then click the blue cloud icon to download the read/write data of the corresponding file. The data will be decoded from base64, and it will be the same as the raw data of the system call.

The malicious behaviors of the attacker will be pronounced in the monitoring because the honeypot is not designed to interact with non-malicious users, and the "Falco log" dashboard only displays the graph when there is an anomalous operation in honeypots. Moreover, "Falco log" can show many specific anomalous behaviors, including their details such as writing file below "/root" or "/etc", deleting or renaming shell history, updating package repository, and launching sensitive mount container (user, command, hostname, and additional information for each specific event will be shown). "SSH logins" is useful to monitor SSH brute-force attacks. The "Syslog" dashboard sometimes gives auxiliary analysis since it is abnormal for a program to generate a large number of logs suddenly, and the administrator can view the syslog in detail from the log data table of this dashboard. "Falco specific monitor" is useful when combined with custom Falco rules—for example, making a rule that can only run Nginx in the Nginx container and monitoring the startup of other programs. Once administrator get the alert from monitoring, use HPViewer to see what commands have been executed and downloading related files. This feature is handy since many things are considered a Linux file, including directories, terminals, and network sockets.

Table 1. Representative dashboards and charts provided for monitoring. Log data tables are omitted because of containing too many texts.

Dashboard	Vital Chart
SSH Logins	SSH login attempts
	SSH username of failed login attempts
Syslog	Syslog events hostnames and programs
Falco log	Falco events by hostname
	Falco log events by container id
	Falco log events by rule
Falco Specific monitor	A series of bar charts for specific Falco rules

Figure 4. "Shell Capture" page of HPViewer

Figure 5. "Trace File" page of HPViewer

EXPERIMENT

After designing and building our system, the authors conduct some simulation experiments to test the functionality and efficiency of the system. Finally, to fulfill the experiment goal, the authors construct an emulation environment simulating an ICS environment in the real world, in which several microcomputers used for industrial control are deployed.

In this simulation, the authors deploy honeypots to these microcomputers as decoys to attract attackers to intrude, so as to collect valuable information of attackers and send data to the core server; the authors also maintain a central server outside the emulated "ICS" network, which plays a key role to gather information from all decoy hosts before conducting further analysis and detection. With this special topology and architecture, this emulation system can now be called an implementation of honeynet in the ICS. (Spitzner, 2003) Detailed information on configuring the two parts is listed as follows:

- Core server: To make our IDS more swift and more robust, the authors decide to deploy the core server onto a "high-performance" computer (compared with honeypot hosts). With 4GB RAM and a Quad-core CPU of 2.0GHz, this server is capable of running key server applications of the ID (like Docker, Samba, Elasticsearch container, Kibana container, HPViewer container, and other related libraries and tools). Once the installation of the core server is done, honeypot hosts can be turned on to monitor the whole emulation environment.
- Honeypot hosts: This kind of microcomputer is a low-power (lower than 48W) integrated machine, with low RAM capacity (no more than 2GB), limited hard disk space (no more than 32GB), and low CPU power (such as Intel N2805, Intel N2830 and Intel J1900). The authors install minimal CentOS 7 Linux distribution (whose ISO image size is only 973 MB) onto these microcomputers and install our honeypot application software (as mentioned in previous sections, including Filebeat, Falco, and Sysdig). After successfully configured with essential arguments and environment, honeypot hosts can connect to the core server, and they are about to go online.

When the construction of the emulation system is completed, the authors expose the "ICS" to the external network, where numerous attackers exist. Thus, while honeypots are constantly intruded and attacked, the central server behind the honeynet is able to receive valuable data, and the system administrator can analyze the behavior of attacks and take appropriate prevention action.

After deploying the emulation system, information collected from honeypots by Sysdig tool is shown in Figure 6, Figure 7 and Figure 8 via accessing to HPViewer service, and Kibana service presents detailed data generated by Filebeat and Falco to the administrator with visual graphs and charts like Figure 9, Figure 10 and Figure 11.

Figure 6. Overall data collected by honeypot hosts and transported with SMB protocol

Figure 7. Shell commands logs collected by honeypot hosts

Figure 8. File read/write logs within the whole file system collected by honeypot hosts

These three figures illustrate the basic functions of the HPViewer, where administrators can trace the system shell execution as well as the file I/O activities via the Web GUI. Furthermore, administrators can easily access different honeypot hosts on the single Web page, which accelerate the management of the whole system.

In Kibana, administrators are able to view the syslog and other crucial information efficiently with the help of graphical charts and dashboards, shown in the following figures.

Figure 9. System logs collected by Filebeat running on honeypot hosts

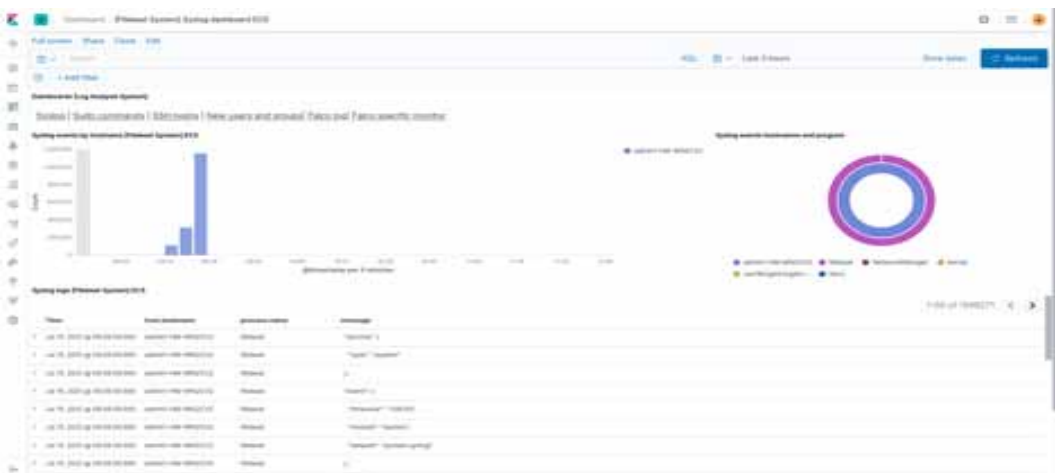**Figure 10. Syslog data collected by honeypot hosts**

Figure 11. Logs of shell commands executed by "sudo" command collected by honeypot hosts

Furthermore, as shown in "Figure 12" and "Figure 13", there are also various special dashboards provided by Kibana service used to monitor specific applications and services running as targeted decoys on honeypot hosts, including AWS, Azure, Apache, Nginx, MySQL, MongoDB, Redis, and so forth. The system administrator can install these specific services onto honeypot hosts to tempt attackers with different interests.

Figure 12. Special dashboards provided by Kibana for specific applications

Figure 13. More special dashboards provided by Kibana for specific applications

[Filebeat Logstash] Slowlogs ECS	Overview of Logstash Slowlogs	
[Filebeat MISP] Overview	Overview dashboard for Filebeat MSIP module.	
[Filebeat MongoDB] Overview ECS	Filebeat MongoDB module overview	
[Filebeat MySQL] Overview ECS	Overview dashboard for the Filebeat MySQL module	
[Filebeat NATS] Overview ECS	Overview of NATS server statistics	
[Filebeat Netflow] Autonomous Systems	Autonomous systems Netflow	
[Filebeat Netflow] Conversation Partners	Netflow conversation partners	
[Filebeat Netflow] Flow Exporters	Netflow exporters	
[Filebeat Netflow] Flow records	Netflow flow records	
[Filebeat Netflow] Geo Location	Netflow geo location	
[Filebeat Netflow] Overview	Overview of Netflow	
[Filebeat Netflow] Top-N	Netflow Top N flows	
[Filebeat Netflow] Top-N Flows	Top N network flows	
[Filebeat Netflow] Traffic Analysis	Netflow traffic analysis	
[Filebeat Nginx] Access and error logs ECS	Dashboard for the Filebeat Nginx module	
[Filebeat Nginx] Overview ECS	Dashboard for the Filebeat Nginx module	
[Filebeat Osquery] Compliance pack ECS	Dashboard for visualizing the data collected by the Osquery compliance pack.	
[Filebeat Osquery] OSSEC rootkit pack ECS	This dashboard shows data collected by the OSSEC rootkit pack from osquery	
[Filebeat PANW] Network Flows ECS	Palo Alto Networks PAN-OS Networks Overview.	
[Filebeat PANW] Threats Overview ECS	Palo Alto Networks PAN-OS Threats Overview	

In these figures, it can be seen that sensitive information about any interaction with the honeypot hosts has been successfully recorded and transported by effective methods. Thus, after receiving valuable data about the potential attacks, system administrators can take timely and appropriate action to mitigate the influence of these attacks.

CONCLUSION

In previous sections, the authors introduced our novel IDS architecture, which aims to secure ICSs and solve the security challenges of critical infrastructure. With the help of system monitor tools like Filebeat, Falco, and Sysdig, and data visualization tools like Elasticsearch, Kibana, and HPViewer, the authors build up an IDS using the deception strategy of excellent timeliness, effectiveness, and user-friendliness. In our experiment, the timeliness and effectiveness of the system are verified, which means our system is ready to conduct further experiments and deploy in a real ICS environment.

In the future, the authors plan to dig into the interaction among honeypot hosts, which will definitely make full use of the power of honeynet, and it is valuable for our system to take more early action when encountering sudden attacks. In addition, implementing some Machine Learning

or Deep Learning algorithms to automatically detect and classify different types of attacks based on data collected by honeypot hosts is also scheduled in our roadmap.

ACKNOWLEDGMENT AND FUNDING AGENCY

REFERENCES

Alcaraz, C., & Zeadally, S. (2013). Critical control system protection in the 21st century. *Computer*, *46*(10), 74–83. doi:10.1109/MC.2013.69

Ani, U. P. D., He, H., & Tiwari, A. (2017). Review of cybersecurity issues in industrial critical infrastructure: Manufacturing in perspective. *Journal of Cyber Security Technology*, *1*(1), 32–74. doi:10.1080/23742917.2016.1252211

Arief, R., Khakzad, N., & Pieters, W. (2020). Mitigating cyberattack related domino effects in process plants via ICS segmentation. *Journal of Information Security and Applications*, *51*, 102450. doi:10.1016/j.jisa.2020.102450

Asghar, M. R., Hu, Q., & Zeadally, S. (2019). Cybersecurity in industrial control systems: Issues, technologies, and challenges. *Computer Networks*, *165*, 106946. doi:10.1016/j.comnet.2019.106946

Bace, R. G., & Mell, P. (2001). *Intrusion detection systems*. Academic Press.

Bolton, W. (2015). *Programmable logic controllers*. Newnes. doi:10.1016/B978-0-12-802929-9.00001-7

Case, D. U. (2016). Analysis of the cyber attack on the Ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC), 388*.

Chowdhury, N., & Gkioulos, V. (2021). Cyber security training for critical infrastructure protection: A literature review. *Computer Science Review*, *40*, 100361. doi:10.1016/j.cosrev.2021.100361

Cole, P. H., & Ranasinghe, D. C. (2008). *Networked RFID systems and lightweight cryptography*. Springer., 10. doi:10.1007/978-3-540-71641-9

Denning, D. E. (2014). Framework and principles for active cyber defense. *Computers & Security*, *40*, 108–113. doi:10.1016/j.cose.2013.11.004

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. doi:10.1016/j.future.2013.01.010

Hadnagy, C. (2010). *Social engineering: The art of human hacking*. John Wiley & Sons.

Heer, T., Garcia-Morchon, O., Hummen, R., Keoh, S. L., Kumar, S. S., & Wehrle, K. (2011). Security Challenges in the IP-based Internet of Things. *Wireless Personal Communications*, *61*(3), 527–542. doi:10.1007/s11277-011-0385-5

Kuipers, D., & Fabro, M. (2006). Control systems cyber security: Defense in depth strategies (No. INL/EXT-06-11478). Idaho National Laboratory (INL).

Kumar, J. S., & Patel, D. R. (2014). A survey on internet of things: Security and privacy issues. *International Journal of Computers and Applications*, *90*(11).

Lawton, G. (2005). LAMP lights enterprise development efforts. *Computer*, *38*(9), 18–20. doi:10.1109/MC.2005.304

Li, Z., Yang, F., & Ishchenko, D. (2012, July). *The standardization of distribution grid communication networks. In 2012 IEEE Power and Energy Society General Meeting*. IEEE.

Matoušek, P., Ryšavý, O., Grégr, M., & Havlena, V. (2020). Flow based monitoring of ICS communication in the smart grid. *Journal of Information Security and Applications*, *54*, 102535. doi:10.1016/j.jisa.2020.102535

Oza, A. D., Kumar, G. N., Khorajiya, M., & Tiwari, V. (2019). Snaring Cyber attacks on IoT devices with Honeynet. In *Computing and Network Sustainability* (pp. 1–12). Springer. doi:10.1007/978-981-13-7150-9_1

Pawlick, J., Colbert, E., & Zhu, Q. (2019). A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy. *ACM Computing Surveys*, *52*(4), 1–28. doi:10.1145/3337772

Piedrahita, A. F. M., Gaur, V., Giraldo, J., Cardenas, A. A., & Rueda, S. J. (2017). Leveraging software-defined networking for incident response in industrial control systems. *IEEE Software*, *35*(1), 44–50. doi:10.1109/MS.2017.4541054

Pöhls, H. C., Angelakis, V., Suppan, S., Fischer, K., Oikonomou, G., Tragos, E. Z., . . . Mouroutis, T. (2014, April). RERUM: Building a reliable IoT upon privacy-and security-enabled smart objects. In 2014 IEEE Wireless Communications and Networking Conference Workshops (WCNCW) (pp. 122-127). IEEE.

Schwab, K. (2017). *The fourth industrial revolution*. Currency.

Segall, A. (1983). Distributed network protocols. *IEEE Transactions on Information Theory*, *29*(1), 23–35. doi:10.1109/TIT.1983.1056620

Spitzner, L. (2003). The honeynet project: Trapping the hackers. *IEEE Security and Privacy*, *1*(2), 15–23. doi:10.1109/MSECP.2003.1193207

Stouffer, K., Falco, J., & Scarfone, K. (2011). Guide to industrial control systems (ICS) security. *NIST Special Publication, 800*(82), 16-16.

Turner, E. B. (1974). Computer based supervisory control systems. *IEEE Transactions on Industry Applications*, *IA-10*(2), 305–315. doi:10.1109/TIA.1974.349149

Shaobo Zhang is currently an undergraduate studying at Beijing Institute of Technology, with great interest in Cybersecurity, Computer Network, and other IT-related technologies.

Yuhang Liu received a bachelor's degree in Internet of Things Engineering from Beijing Institute of Technology in 2020. In the same year, he entered Peking University to study for a master's degree in Computer Application Technology and is currently engaged in the research of network information security and natural language processing.

Dequan Yang received his Ph.D. in Control Science from the Beijing Institute of Technology in 2013. In 2013 he was hired at the Beijing Institute of Technology. His research interest includes cloud data center and cyberspace security.