Central Power Research Institute

Vol. 16(2)/153-162, July-December 2020
ISSN (Print) : 0973-0338
DOI : 10.33686/pwj.v16i2.155568

Power Research

Hybrid SCADA Security Testbed as a Service

Rajesh Kalluri, Reddi Hareesh, M. V. Yeshwanth*, R. K. Senthil Kumar and B. S. Bindhumadhava

Center for Development of Advanced Computing (C-DAC), Knowledge Park, No 1, Old Madras Road Byappanahalli, Bangalore – 560038, Karnataka, India; rajeshk@cdac.in, reddihareesh@cdac.in, mvyeshwanth@cdac.in, senthil@cdac.in, bindhu@cdac.in

Abstract

Supervisory Control and Data Acquisition (SCADA) systems are deployed for control and management of critical infrastructures (power, oil, gas, water, etc.), industries (manufacturing, production, etc.) and public facilities (airport, ships, transport etc.). With the evolution of the technologies in communication, SCADA systems are connected to different networks using heterogeneous communication infrastructure. Thus, SCADA systems became vulnerable to threats of connected systems along with its legacy threats. A security assessment is required to understand the security posture of the system. However, it is not possible to simulate and analyze attacks on a real SCADA system. Hence, a testbed is needed to conduct any security assessment by modeling the architecture on the SCADA testbed. In this paper, we will discuss the need for testbeds, hybrid testbeds, how we established a hybrid testbed, simulation and impact analysis of attacks on the hybrid testbed and the process of providing the testbed as a service.

Keywords: Hybrid Testbed, SCADA Security, Simulation of Attacks, SPADE, Testbed, Testbed as a Service

1. Introduction

SCADA systems are widely used in critical infrastructure industries, manufacturing industries and public facilities. Earlier, SCADA systems were deployed with legacy systems, proprietary protocols, maintained air gaps between networks and were considered to be secure. Due to the increased connectivity to the internet and corporate network, SCADA networks are no longer immune to cyber-attacks[1]. Along with the legacy vulnerabilities, SCADA systems inherited the vulnerabilities of the connected networks. Key security components while addressing the security of SCADA system are confidentiality, integrity and availability. Availability is the top priority for SCADA systems whereas confidentiality is the main concern for IT systems.

Practically, there are many possible attack scenarios based on the SCADA architecture. There are 4 key attack scenarios on the control systems based on the criticality level, i.e. attack from the field network, attack from the corporate network, physical attacks and attacks on devices exploiting communication channels. Communication protocols used in SCADA are plain text protocols[4,5] and are not provided with enough security features.

In order to better understand how to protect SCADA systems, it is important to analyse the security risk, attack impacts on the system and develop appropriate security solutions to protect them[2,3]. A SCADA testbed can be used to model a SCADA system to be tested with the additional benefit of testing real attacks on the system and analyze the impact of the attacks. Building a SCADA testbed is important since it is not possible to conduct security experiments on a real SCADA system considering the cost, downtime and risk on the system.

Typically, SCADA testbeds are classified as physical, virtual/emulated and hybrid testbeds. The physical testbed consists of same physical components that are used in the real SCADA system, virtual testbed consists of virtualized components and hybrid testbed consists of a combination of physical and virtual components. In this paper, we will discuss setting up of hybrid testbed, modeling different networks of a typical SCADA network, simulation of different attacks targeting the control networks and the procedure of providing the testbed as a service to scale it to any SCADA architecture.

The rest of the paper is organized as follows: Section 2 discusses the hybrid testbed, Section 3 discusses setting

*Author for correspondence

up the testbed, Section 4 discusses the simulation of attacks on the testbed and Section 5 discusses providing the testbed as a service.

2. Hybrid Testbed

Testbeds of SCADA systems can provide effective support for analyzing and assessing the vulnerabilities and security of SCADA systems.

SCADA testbeds can be used for different purposes such as threat and vulnerabilities identification, simulation of attacks and their impact analysis. These SCADA testbeds can be typically classified into physical testbed, virtual testbed, simulation testbed and hybrid testbed[4].

Physical testbed: Physical SCADA testbed is the process of setting up the architecture of utility with replication of existing physical components[15]. National SCADA testbed[5] is one of the best examples of the physical testbed. Physical testbed is good in understanding the system vulnerabilities and the attack impact with better accuracy with live examples. But considering the cost factor, this testbed is not suitable for scaling up. In case any component got damaged during the simulation of attacks, it is very difficult and expensive to restore back.

Simulation testbed: Simulation testbed models the whole architecture using simulated software that provides similar functions and behaviors of the SCADA system being modeled. Scalability, reconfiguration of the components and maintenance of the testbed can be easily achieved. Simulation testbeds do not provide high fidelity as physical testbeds. SCADASim framework developed at the Royal Melbourne Institute of Technology, Melbourne, Australia, provides predefined modules for building SCADA simulations[6].

Virtual testbed: Virtualization is the concept of executing software in an environment that minimizes or eliminates the software's dependence on the hardware on which it runs[7]. Virtual testbed is used to overcome the limitations between physical and simulation testbed. Virtualization is the concept of executing software in an environment that minimizes or eliminates the software's dependence on the hardware on which it runs[8]. Typically, Virtual testbed[9] can be used to model all layers of the SCADA infrastructure. The testbed uses the PowerWorld simulation system to simulate the operations of segments of the electrical power grid and OPNET tool to simulate computer networks.

Hybrid testbed: In order to utilize the advantages of each approach, a hybrid testbed has been proposed[10,11].

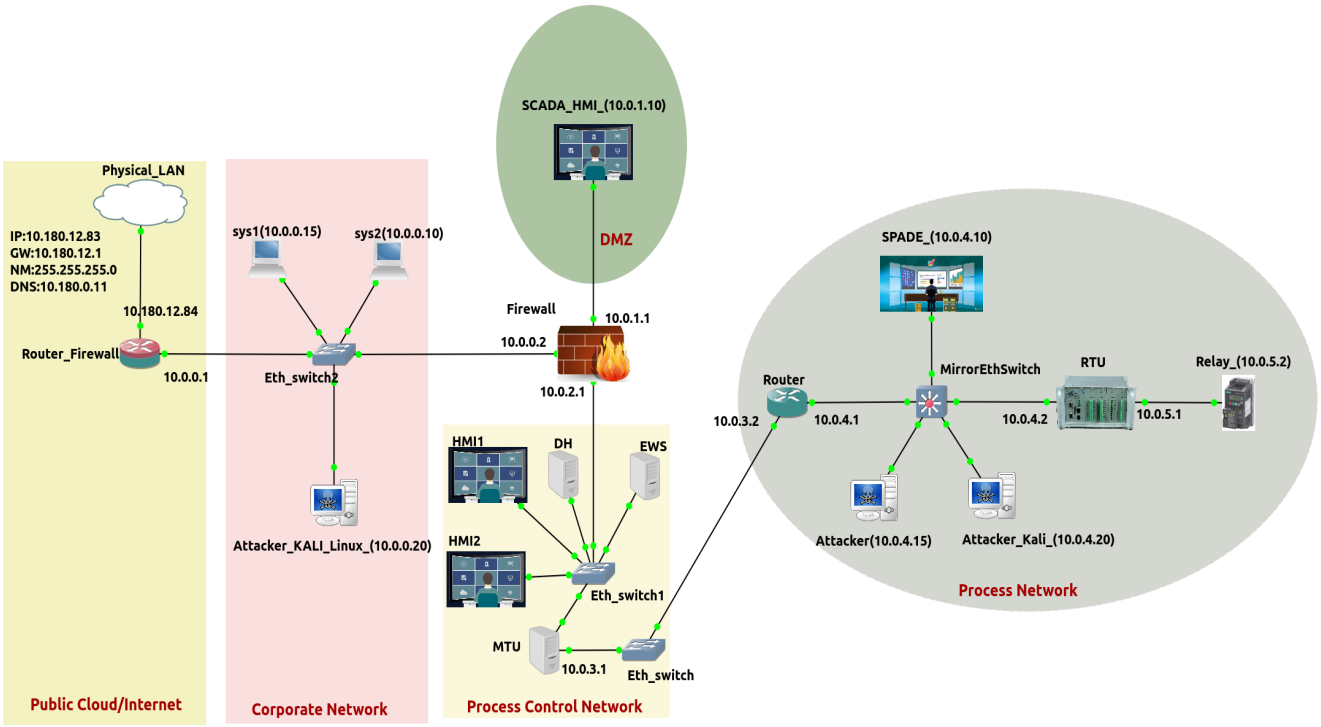

Figure 1. Configurable Hybrid SCADA testbed Architecture.

A hybrid testbed consists of a combination of physical, virtual and simulated components. This testbed provides a cost effective and high-fidelity model. In this paper, hybrid test bed has been modeled with the typical networks involved in SCADA systems such as corporate network, demilitarized zone, process control network and process network.

3. Setting up the Testbed

The Hybrid testbed has been set up using a network software emulator called the GNS3[17]. GNS3 allows to run a small topology consisting of only a few devices or simulate the network of a whole organization on a single computer/remote-server.

In the hybrid testbed, corporate network is connected to the internet through an emulated router called "Vyos". The emulated router is interfaced with a physical router using the host machine's ethernet port. The corporate network contains emulated systems using the docker. One of the emulated systems is kali linux which helps in attacking the MTU (Master terminal unit). We can import software-images of several network devices to emulate the physical network devices with the same configurations. These network components will be able to communicate with physical network devices. Similarly, dockers, VMware, Vbox virtual machines can be imported to emulate different operating systems to be connected into the network. Any number of physical systems can be added to the GNS3 network making it horizontally scalable. All the networks are separated using an emulated firewall. All the components are easily configurable and can simulate attacks using an emulated C-DAC SCADA Threat Analyzer and Security incident and event management tools[13].

The hybrid testbed is a composition of simulated, emulated and physical components spread across the different network layers namely the corporate network, process control network zone and the process network connected as shown in Figure1. Brief details about these networks are explained as below:

- **Corporate Network (CN)**

Corporate networks consist of a variety of emulated as well as physical computer systems connected to the internet. These systems and networks are connected to the internet and are more vulnerable to attacks. The Corporate network is connected to a three-legged emulated firewall router Vyos which separates the DMZ (Demilitarized Zone) and the PCN (Process Control Network). The corporate network is connected to the process control network through demilitarized-zone.

- **Demilitarized Zone (DMZ)**

Demilitarized zone (DMZ) is used to isolate the process control network from corporate networks. DMZ hosts servers that collect and aggregate data received from SCADA servers in the process control network. Typically, outside looking services such as HMI (Human machine interface) and data historians are deployed in DMZ. The DMZ in the hybrid testbed is created using Vyos router which isolates the PCN from the most vulnerable corporate network. Any interaction between the CN and PCN is through the services deployed in DMZ.

- **Process Control Network (PCN)**

Process control network consists of different components such as the MTU, HMI, database server, application server. PCN consists of one or multiple Master terminal units and HMIs.

Master terminal unit (MTU): Master terminal unit interfaces multiple RTUs (Remote terminal unit) to the HMI. MTU connects to the RTU to acquire data and sends it to the HMI. Multiple emulated or physical MTUs can be connected for fault tolerance.

Human machine interface (HMI): It acquires measured and indication data and displays in various formats like single line diagram, tabular diagrams. Multiple emulated or physical HMI may be added to the network to improve availability.

- **Process Network**

Process network typically consists of all field devices such as Remote Terminal Units (RTU), SCADA Protocol Anomaly Detection Engine (SPADE), field devices such as relays, circuit breakers, etc. This network consists of multiple remote terminal units, simulation components of power systems and field devices.

Remote Terminal Units (RTU): It's an electronic device that interfaces field devices in the physical world to a distributed control system or SCADA system by transmitting telemetry data to a Master Terminal Unit (MTU), and by using messages from the master terminal unit to control connected objects. Field devices to the RTUs are connected as physical components as well as using Hardware In the Loop (HIL) technology. For the same, we have used the CPRI real time digital simulator[12].

SPADE: SPADE is used to detect anomalies with the communication channel between RTU and MTU adhering to IEC 870-5-104. SPADE is a passive monitoring device and does not disturb the communication between RTU and MTU. It works by capturing and performing the Deep Packet Inspection (DPI) and the deep content inspection (DCI) in real-time on the packets that are entering and leaving the RTU. It uses the mirrored-port to capture all the RTU traffic. Visualization of alerts or events captured by SPADE can be visualized using SCADA Vision as shown in Figure 2. This solution is indigenously developed by C-DAC.

- **Simulated Component of the Power System**

For simulating the field devices, power system network has been modeled in scilab[18] and the electrical inputs have been provided to the SCADA testbed and the corresponding control action from the SCADA testbed has been sent to the simulation environment where the necessary suggested action has been implemented on the power system network forming a closed-loop feedback system. The matrix based computations required for the power system modeling can be performed efficiently. Power system components like transmission lines, loads and generating sources are mathematically modeled and the desired characteristics of the system components have been realized using scilab. The power system parameters like voltages, voltage angle, real and reactive powers at various nodes and the line current flow, circuit breaker status for each line has been provided as an input to the SCADA testbed. As a case study Western electricity coordinating council (WECC)–3 machine 9-bus and the IEEE 14 bus 5 machine system have been modeled and corresponding inputs are given to the SCADA system. Any control actions such as line removal and restoration can be implemented on the system. The corresponding system behavior like voltage, power, and rotor angle variations of the generators with respect to the changes in power system configuration has been studied.

4. Simulation and Impact Analysis of Attacks

Using the testbed we can establish several attack scenarios that can compromise the integrity and availability of the process network[14,16]. CDAC's multi agent framework (CMAF), SPADE, C-DAC SCADA threat analyzer (STA)[13] and an emulated Kali-Linux based system with pre-configured tool is introduced to the testbed network as shown in Figure 1. These will help conduct several attacks that are unsafe for the real SCADA environment.

Practically we can simulate numerous attacks using the testbed and targeting different network segments in the SCADA network. However, considering the criticality of the system, in this paper we are covering the simulation of attacks on the process network. Attacks simulation has been divided into two categories i.e. attack on process network components through process control network and attack on process network components through communication channels.

4.1 Attacks on Process Network Components through the Process Control Network

In this section, CDAC's multi agent framework (CMAF), C-DAC SCADA threat analyzer (STA)[13] and an emulated Kali-Linux based system with pre-configured tools will be used for simulation and impact analysis of attacks.

4.1.1 DoS (Denial of Service) Attack

To simulate the DoS attack, the attacker initiates attack through compromised MTUs in the network. Using the STA, the attacker could select a type of DoS attack from the repository for installation. On selection of a DoS type, an installation agent will be sent to the compromised MTUs to install and run the particular DoS tool. This will flood the targeted RTU(s) with the large-sized packets or malicious packets to crash the RTU service, or flood the RTU with packets to overwhelm the RTUs resources or the network bandwidth, etc.

Once the DoS attack is successful, the target RTU becomes unavailable for essential data acquisition and to perform supervisory control actions, and MTU will not be able to update the data on servers. Hence the critical tasks of control systems such as load management, load scheduling, and load forecasting will be affected, this intern can jeopardize the process being monitored by the targeted SCADA system.

4.1.2 Using Malwares to Compromise PCN Systems

The mobile agents of CDAC's CMAF tool are used to perform malware attacks across the SCADA testbed. The

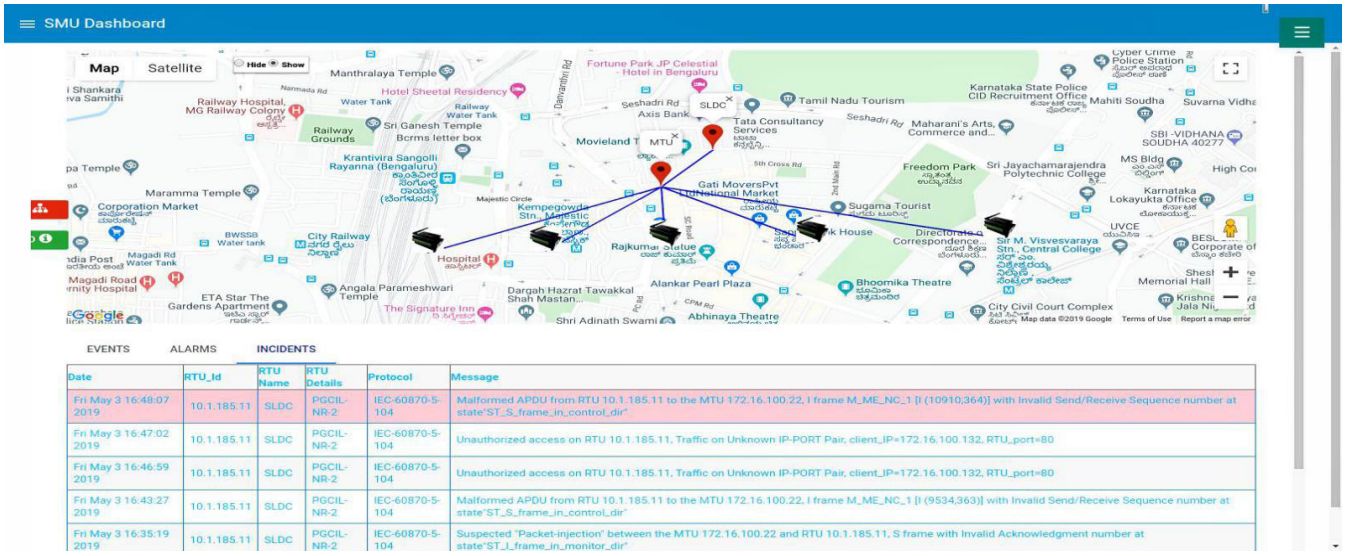

Figure 2. SPADE SCADA-vision dashboard.

CMAF mobile agent is chosen for an attack environment as it enables easier communication over SCADA through code migration and node to node communication that is necessary for analyzing malware characteristics and behavior. Using STA, malware attacks can be initiated in the following steps.

- Select the target system
- Select the malware from the repository
- Inject the malware into the target system using the installation agent

These malwares can reprogram the host system communication logic or data processing logic to suppress alarms and reports, to hide the malicious activity, modify the data being reported, damage the equipment, and DoS attack by crashing down the service running on the host system such as MTU or RTU.

4.2 Attacks on Process Network through Communication Channel

The SCADA testbed uses a CDAC developed DPI and DCI based passive security monitoring solution called SCADA Protocol Anomaly Detector.

(SPADE) to detect these attacks and generate the alerts with an associated impact severity level. Since active monitoring can introduce overhead to the SCADA network which is sensitive to unexpected traffic on the network, SPADE uses passive monitoring with DPI and DCI based white-list rules to effectively detect security incidents on the SCADA networks. The attacks on the process network are simulated by introducing a Kali-Linux attack machine with several attack tools configured, to the process network as shown in Figure 1.

The following are some of the attacks simulated on the process network. SPADE is used to detect the attack and to assess the system response to the attack. And to visualize the results, SPADE uses a graphical UI as shown in Figure 1. Figure 2 shows a typical snapshot of the SCADA-Vision dashboard listing the latest alerts.

4.2.1 Unauthorized Control Command on RTU

To simulate the attack, CDAC has developed the MITM (Man In the Middle Attack) tool to intercept the communication between the target RTU and MTU. Pretending to be a legitimate MTU, the attacker sends a C_SC_NA digital control command to perform on/off action on a circuit breaker connected at an unknown device address as shown in Figure 3. This attack can result in unexpected and dangerous behavior in the process being monitored by the targeted RTU.

When the attacker spoofs his identity to a legitimate MTU IP and MAC address, the command is detected by communication pattern rules based on the packet's insignificance in the present state of communication between the actual MTU and RTU. As the attacker may not be having the knowledge of device (sensor) addresses, the attack will trigger an alert based on a violation of a proposed signature rule based on the white-listed List of Device Addresses (IOA).

Figure 3. Unauthorized do control-command attack on RTU.

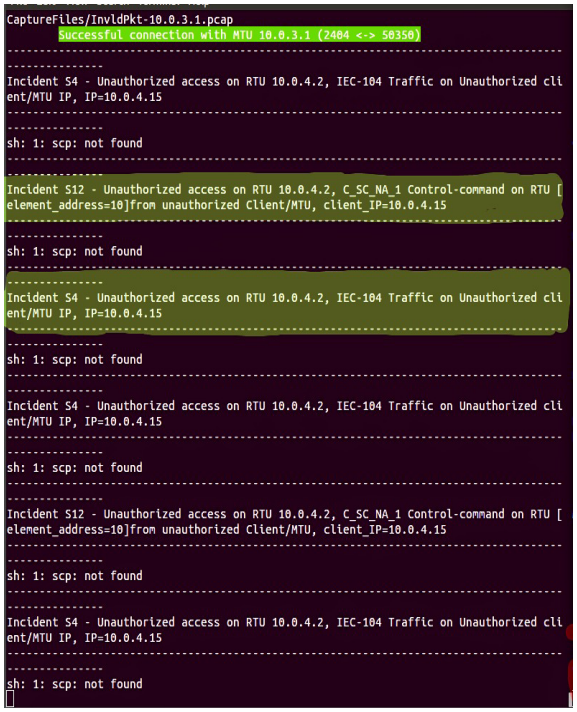

Figure 4. Incident-report for unauthorized control-command.

Each rule violation resulted in the generation of an alert with an associated impact/risk level. Behavior profiling on these sets of alerts is done to find the type of attack, and hence unauthorized control command attack is successfully detected and visualized by SPADE as shown in Figure 4.

4.2.2 Data Modification Attack

The attacker uses the Ettercap tool on Kali-Linux to perform an ARP-Poisoning attack between the target RTU and MTU and intercepts a legitimate analog control command (C_SE_NA) sent by the MTU and modifies the contents of the packet by injecting some extra bytes to the frame before sending it to the RTU to disrupt the normal operation of the system as shown in Figure 5. This attack can also result in unexpected and dangerous behavior in the process being monitored by the targeted RTU. As the injected bytes violate the SPADE white-list protocol-behavior-model-rules that verifies the protocol compliance, along with the violation of SPADE white-list signatures on MAC-IP address pairs, a set of alerts are generated. Behavior profiling on these sets of alerts is performed and the data modification attack is successfully detected and visualized by SPADE as shown in Figure 7.

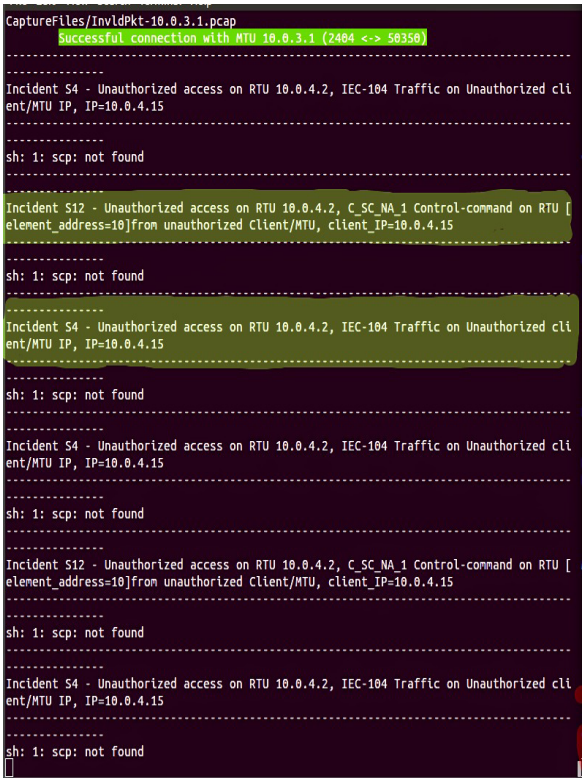

Figure 5. AO control-command with data-modification attack.

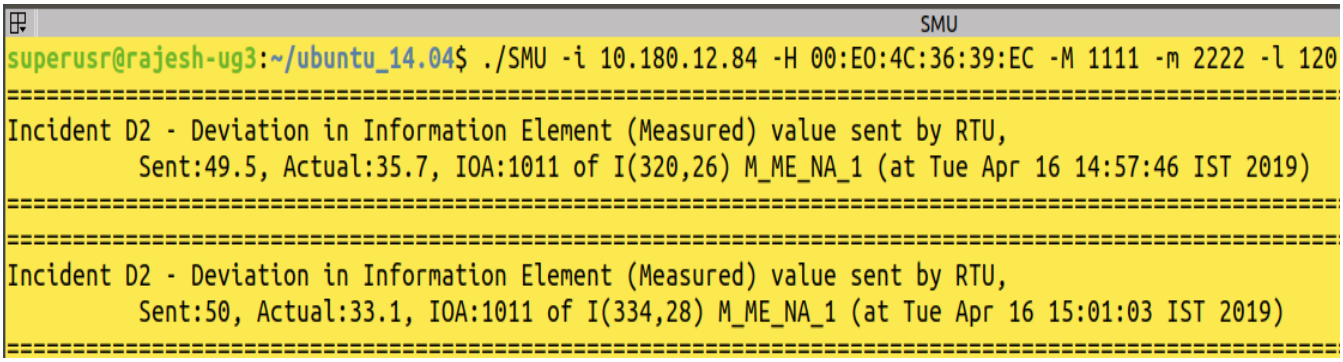

Figure 6. Incident-report for malicious behavior of RTU.

4.2.3 Malicious Behavior of RTU (Malware on RTU)

An attacker injects a malicious RTU configuration file to intercept the RTU data processing logic with an intention to modify the sensor values in the message which carries data from RTU to MTU. The attack is to mislead the operator with wrong information about the process being monitored. The malicious code is written to intercept M_ME_NA which carries measured data of analog channels, and modify the measured analog value at frequency sensor with sensor address <X> to always a safe value, though the actual value is at an unsafe level. The Impact of this attack will be severe as it can induce damage to the system, equipment and life. For detection, as soon as the malformed spontaneous data packet is captured by the SPADE, the values sent in the message for the particular analog sensor address <X> is correlated with the sensor data from the same device (obtained through a redundant port of the sensor). As the value sent by the RTU is deviating from the actual value, an alert is successfully triggered to indicate abnormal behavior of the RTU and visualized by SPADE as shown in Figure 6.

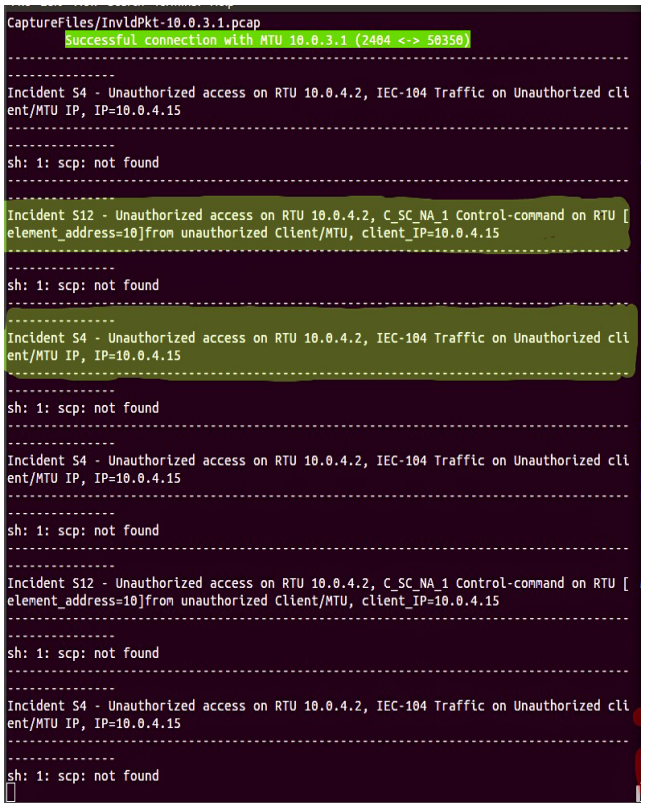

Figure 7. Incident-report for data modification attack.

4.2.4 MITM Attack

The attacker uses the Ettercap tool on Kali-Linux to perform an ARP-Poisoning MITM attack between the target RTU and MTU to intercept himself between them as shown in Figure 8. Attackers can perform data modification attacks to mislead the operator, packet injection attack to cause unauthorized control actions or disrupt the RTU with an attack such as buffer overflow.

SPADE successfully detects the MITM attack using white-list signature rules, as the ARP poisoning violates the MAC-IP address pair rule. The alert is finally visualized to the operator as shown in Figure 9.

4.2.5 SYN-flood, ICMP-flood, UDP-flood DoS Attack

The attacker passively listens to the network to read the network addresses used by the authorized hosts, spoofs the identity of an authorized MTU in the network and uses the hping packet generator tool to flood the RTU with TCP-SYN, ICMP-Echo or UDP packets as shown in Figure 10 to overwhelm the RTU resources or the network bandwidth to make the RTU inaccessible for the intended operations.

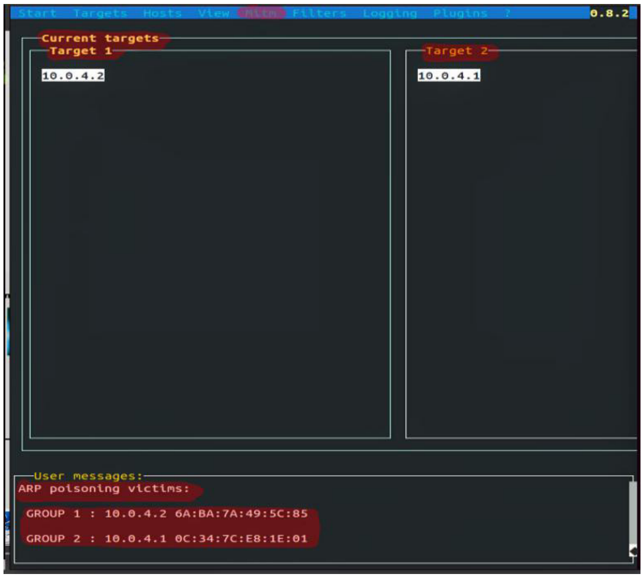

Figure 8. MITM attack between RTU and MTU.

Though the attacker spoofs his identity, SPADE successfully detects the attack using a set of white list rules, such as the white-listed rule that restricts network throughput at the RTU to a safe level, the rules that monitors the number of half-opened sessions and the connected sessions, rules that keeps track of ports that are opened on RTU and the rules that puts host based restriction, that limits the hosts that can connect to particular port on the RTU. SPADE generates a set of alerts based on the rules that are violated and by performing decision making on these alerts the flooding attack is detected and visualized as shown in Figure 11.

Figure 9. Incident-report for MITM attack.

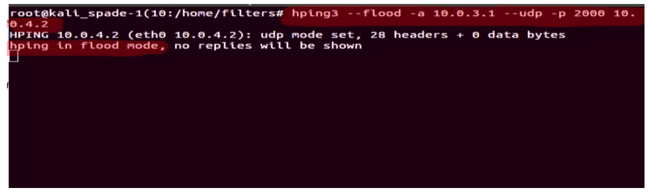

Figure 10. Simulation of UDP-flood dos attack.

4.2.6 Unauthorized Port-scan on RTU

The attacker uses nmap tool to perform port scan on RTU to find the opened ports on RTU as shown in Figure 12, and uses that information to scan for vulnerabilities in the system to perform sophisticated zero-day attacks. SPADE detects the attack successfully, as the attack violates a set of white-list rules such as the restriction on number of TCP connection on a TCP port, rule that doesn't expect a connection request on a closed port and a rule that restricts the hosts that can connect to particular port on RTU. Based on the rules violated the attack is detected and visualized to the operator as shown in Figure 13.

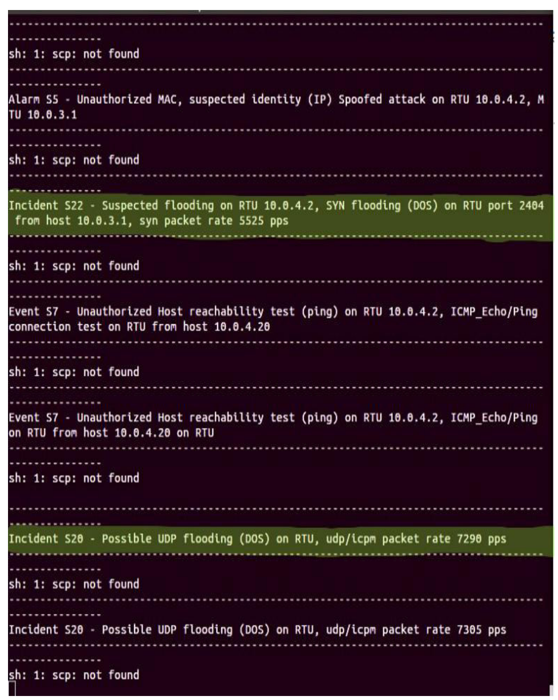

Figure 11. Incident-report for UDP-flood dos attack.

Figure 12. Unauthorized port-scan on RTU.

5. Providing Test Bed as a Service

Any utility who wants to simulate attacks on a testbed consists of steps such as setup a testbed, model the network, and simulate the attacks, analyzing the impact results, maintaining the testbed, updating the components of the testbed. However, this requires capital expenditure, operational expenditure and the right skill set. In order to overcome the difficulties, this hybrid testbed is provided as a service to utilities. There are various phases involved in the process to provide this testbed as a service i.e. modeling phase, simulation of attacks, vulnerability analysis phase.

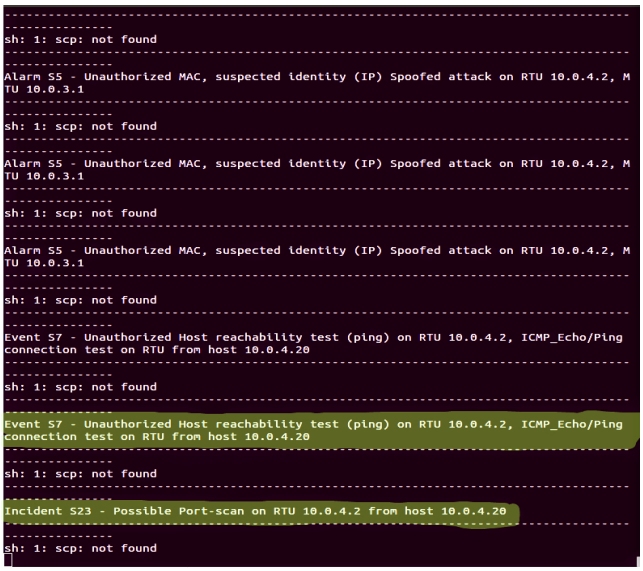

Figure 13. Incident-report for unauthorized port scan.

The Modeling phase consists of modeling the architecture on the testbed, the simulation attacks phase consists of simulating different attacks as required by the user and generating a report. The Vulnerability analysis phase consists of analyzing the vulnerabilities of the whole system.

Steps in modeling phase:

1. Every User needs to create an account before starting to evaluate their system through the portal.
2. The User can send an online request for the architecture setup by uploading the architecture diagram
3. The architecture will be modeled by C-DAC on the testbed and replied through email to verify.
4. The user after verifying will be prompted to add/upload the corresponding application which should be running on every system.
5. Sequence of Initiation/Starting of Systems and the application should be defined by the User online.
6. Any configuration for individual systems and misbehavior can be defined by the User.
7. Any fallbacks and issues should be handled as per the User verification message.
8. Once the network component architecture and corresponding applications are set up, the field devices and the data format can be configured and simulated.
9. The user is ready for Security Assessment.
10. The user can initiate "Startup Testbed".
11. Any fallbacks and issues should be handled as per the User verification message in a redundant manner.

Steps in simulation of attacks phase:

Now the user will be prompted with a list of attack scenarios against which the SCADA System can be assessed. For Example, Various DoS attacks, MiTM can be performed.

1. The User can select an attack first like "DoS attack".
2. Next any sub attack category can be selected like "Syn flood", "Buffer Overflow" etc.
3. Any known attack can be "Previewed in Action" for a sequence of events which take place during the attack to get an understanding what it is all about.
4. The propagation of the attack over the network can also be visualized in the Dashboard.
5. The detection chart for the attack will also be shown.
6. Detailed report will be generated after successfully simulating various attacks.

Vulnerability analysis phase consists of the following steps:

1. The user can run a SCADA System Health Monitor over the entire SCADA or a part of the SCADA System.
2. This uses a repository of known vulnerability databases to scan through the system under consideration.

3. Now the SCADA System can be checked against these selected vulnerabilities if it is existing in the current SCADA system under consideration.
4. If it exists, the system will tell the probability of an attacker exploiting the same.
5. The system will also mention the appropriate mitigation steps for overcoming the vulnerability like patch fix-up, upgrade etc.

6. Conclusion

To protect SCADA systems from cyber threats, it is important to assess the security of the systems by simulating attacks, studying the impact analysis of attacks to plan countermeasures. Since it is impractical to simulate attacks on real time systems, a testbed is required. A hybrid testbed provides high fidelity and cost-effective solution compared to the physical, simulation and virtual testbeds. In this paper, we have set up the hybrid testbed using the GNS3 network simulator with physical, virtual and simulated components. Various attacks have been simulated targeting the process network with RTUs and field devices. Attacks on RTU are detected using a SCADA protocol anomaly detector. Attacks have been simulated targeting components in the process control network. This testbed has been provided as a service to users since it is difficult to set up a whole testbed in their premises.

7. References

1. Hemsley, Kevin E, Fisher E. History of industrial control system cyber incidents. No. INL/CON-18-44411-Rev002. Idaho National Lab. (INL), Idaho Falls, ID (United States); 2018.
2. Abrams M, Weiss J. Malicious control system cyber security attack case study - Maroochy Water Services, Australia. Technical Report, Mitre.org.; 2008. p. 1–16.
3. Babu B, *et al.* Security issues in SCADA based industrial control systems. 2017 2nd International Conference on Anti-Cyber Crimes (ICACC). IEEE; 2017. https://doi.org/10.1109/Anti-Cybercrime.2017.7905261. PMid: 29199662. PMCid:PMC5750623
4. Qassim Q, *et al.* A survey of SCADA testbed implementation approaches. Indian Journal of Science and Technology. 2017; 10(26).
5. Hahn A, *et al.* Development of the PowerCyber SCADA security testbed. Proceedings of the sixth annual workshop on cyber security and information intelligence research; 2010. https://doi.org/10.1145/1852666.1852690
6. Queiroz C, Mahmood A, Tari Z. SCADASim-A framework for building SCADA simulations. IEEE Transactions on Smart Grid. 2011; 2(4):589–97. https://doi.org/10.1109/TSG.2011.2162432
7. Daniels J. Server virtualization architecture and implementation. Crossroads. 2009; 16(1):8–12. https://doi.org/10.1145/1618588.1618592
8. Reaves B, Morris T. An open virtual testbed for industrial control system security; 2012. https://doi.org/10.1007/s10207-012-0164-7
9. Mallouhi M, *et al.* A testbed for analyzing security of SCADA control systems (TASSCS). ISGT 2011. IEEE; 2011. https://doi.org/10.1109/ISGT.2011.5759169
10. McLaughlin S, Konstantinou C, Wang X, Davi L, Sadeghi A, Maniatakos M, *et al.* The cybersecurity landscape in industrial control systems. Proceedings of the IEEE. 2016 May 5; 104:1039–57.
11. Holm H, Karresand M, Vidström A, Westring E. A survey of industrial control system testbeds. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Buchegger S, Dam M, (Eds), Springer; 2015. 9417. https://doi.org/10.1007/978-3-319-26502-5_2
12. Amaraneni A, *et al.* Transient analysis of cyber-attacks on power SCADA using RTDS. Power Research. 2015; 11(1):79–92.
13. Rao MS, Kalluri R, Kumar RKS, Prasad GLG, Bindhumadhava BS. Impact analysis of attacks using agent-based SCADA testbed. In ISGW 2017: Compendium of Technical Papers, Springer, Singapore; 2018. p. 41–54. https://doi.org/10.1007/978-981-10-8249-8_4
14. Stranahan J, Soni T, Heydari V. Supervisory control and data acquisition testbed vulnerabilities and attacks. SoutheastCon, Huntsville, AL, USA; 2019. p. 1–5. https://doi.org/10.1109/SoutheastCon42311.2019.9020436
15. Stranahan J, Soni T, Heydari V. Supervisory control and data acquisition testbed for research and education. 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA; 2019. p. 85–9. https://doi.org/10.1109/CCWC.2019.8666482
16. Rosa L, Cruz T, Simões P, Monteiro E, Lev L. Attacking SCADA systems: A practical perspective. 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon; 2017. p. 741–6. https://doi.org/10.23919/INM.2017.7987369. PMid:28246669
17. Available from: www.gns3.com.
18. Available from: www.scilab.org.