SECURITY ISSUES IN SYSTEM DEVELOPMENT LIFE CYCLE OF SMART GRID

Parminder Kahlon
B.S., Lovely Professional University, India, 2006

PROJECT

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

at

CALIFORNIA STATE UNIVERSITY, SACRAMENTO

SPRING
2011
SECURITY ISSUES IN SYSTEM DEVELOPMENT LIFE CYCLE OF SMART GRID

A Project

by

Parminder Kahlon

Approved by:

__________________________________, Committee Chair
Isaac Ghansah, Ph.D.

__________________________________, Second Reader
Ahmed Salem, Ph.D.

__________________________________
Date
Student: Parminder Kahlon

I certify that this student has met the requirements for format contained in the University format manual, and that this project is suitable for shelving in the Library and credit is to be awarded for the project.

______________________, Graduate Coordinator
Nikrouz Faroughi, Ph.D. Date

Department of Computer Science
Abstract

of

SECURITY ISSUES IN SYSTEM DEVELOPMENT LIFE CYCLE OF SMART GRID

by

Parminder Kahlon

Smart Grid includes critical hardware and software applications that can be misused by an unauthorized person or a hacker. Failures can bring critical parts of the system to a halt. The destruction is not only limited to the monetary losses but also human loss due to a disgruntled action. There have been reports including one from United States Department of Homeland Security that cyber spies have managed to inject malicious software into the electric grid, water, sewage, and other infrastructure control software. This software could enable hackers or unauthorized users to take control of key facilities or networks via the Internet, causing power outages and tremendous damage to all sectors of the economy. As the grid becomes more central to our energy infrastructure, it will become more important to ensure its security. Smart Grid systems create a link between physical and software systems, both of which can fail.

There is a strong need of building a secure and intelligent system that can handle all the exceptions and results in consistency of information flow. The System Development Life Cycle (SDLC) of Smart Grid should involve the tactics and techniques to address the Cyber-Security issues of the Grid. Cyber-Security comprises maintaining the
confidentiality, integrity and availability of the Smart Grid system. Security threats which arise due to improper security requirements, malicious code, Denial of Service (DOS), malfunctioning device, lack of security testing etc. can be tackled in the SDLC of Smart Grid.

This Project demonstrates how different security practices can be used in the SDLC to enhance the security of the Smart Grid. This research focuses on the security practices and controls for each phase of the SDLC to secure the Smart Grid by Design, Deployment and Default so that if somehow it fails it can fail securely. As Smart grid system is a System-of-Systems (SoS), it includes different software and applications which are acquired from different vendors or parties and outsourced teams. These outsourced and Commercial-of-the-shelf (COTS) hardware and software components increase the risks of compromised software or third party tampering in the supply chain process. This project also includes some of the key practices and controls which can help to reduce the security vulnerabilities due to Supply Chain in Smart Grid Environment.

____________________________
Isaac Ghansah, Ph.D.

__________________________________, Committee Chair

____________________________
Date
DEDICATION

This project is dedicated to my wonderful parents, Sukhwinder Kaur Kahlon and Lakhbir Singh Kahlon, who have raised me to be the person I am today. I appreciate the hard work and sacrifice they put and made me capable to reach this far. I express my deepest gratitude and love for their dedication and the many years of support during my earlier studies that provided the foundation for this work. Thank you for all the unconditional love and support which provide me with the confidence that I am capable of doing anything I put my mind to. Thank you for everything. I love you!
ACKNOWLEDGMENTS

The success of this project depends on the encouragement and guidance I got from many individuals. I take this opportunity to express my gratitude to them in my humble acknowledgment.

I would like first and foremost thank Dr. Isaac Ghansah for his direction, assistance and guidance. His Network Security class motivated me to undertake this project. I appreciate Dr. Ghansah’s time, patience, comments and technical advice for my project. I extend my heartfelt gratitude to Dr. Ahmed Salem for agreeing to be the second reader of my project.

I thank my husband Mr. Ravi Gill for his love, support and advice at each and every step of my life. I owe him for keeping my spirits up. I am indebted to him more than he knows. Thank you for understanding and supporting me all the way through it.

Special thanks should be given to my parents not only the ones who gave birth to me but my Parent-in-laws, without their support and sacrifice I would not have gone so far. Finally, words alone cannot express the thanks I owe to my brother, sisters (Ramneek, Aman and Ramn) and whole family for their encouragement and support.

Last but not the least; I thank Almighty God for giving me Strength, Determination, Guidance and Knowledge to accomplish this project.
TABLE OF CONTENTS

Dedication........................................................................................................................................ vi
Acknowledgments................................................................................................................................ vii
List of Tables ......................................................................................................................................... x
List of Figures ....................................................................................................................................... xi
Chapter
1. INTRODUCTION .......................................................................................................................... 1
  1.1 Background..................................................................................................................................1
  1.2 Statement of Problems ..............................................................................................................3
  1.3. Project Objectives..................................................................................................................4
  1.4. Report Organization ..............................................................................................................5
2. SMART GRID .................................................................................................................................. 7
  2.1 Introduction ...............................................................................................................................7
  2.2. Main Components of Smart Grid ..........................................................................................8
    2.2.1. Advanced Metering Infrastructure ...............................................................................9
    2.2.2. Demand Response ........................................................................................................11
    2.2.3. Supervisory Control And Data Acquisition (SCADA) system .....................................13
    2.2.4. Plug In Hybrid Electric Vehicles..................................................................................15
3. PHASE 1: TRAINING AND INITIATION ................................................................................. 19
  3.1. Employees and Customers .....................................................................................................19
  3.2. Development Team ...............................................................................................................21
4. PHASE 2: ACQUISITION AND DEVELOPMENT ....................................................................... 26
  4.1. Best Practices for Acquisition and Development .................................................................26
    4.1.1. Identify Security requirements ......................................................................................26
4.1.2. Conduct Risk Assessment ................................................................. 31
4.1.3. Supply Chain Risk Analysis ................................................................. 32
4.1.3. Define Third Party Code Licensing Security Requirements ................. 34
4.1.3. Design and Evaluate Security Architecture ........................................... 35
4.2. Different Techniques for Acquisition and Development for Smart Grid Components ........................................................................................................37
4.2.1. Attack Trees for AMI .............................................................................. 37
4.2.2. Threat Modeling for Demand Response ................................................. 41
4.2.3. RAIM Framework for SCADA .............................................................. 52
4.2.4. UML/OCL for Plug In Electric Vehicles ................................................ 58
5. PHASE 3: IMPLEMENTATION AND ASSESSMENT ............................................. 65
5.1. Security Implementation Practices ............................................................. 66
5.2. Security Assessment Practices ................................................................. 70
6. PHASE 4: OPERATIONS AND MAINTENANCE ........................................... 79
7. PHASE 5: DISPOSAL .................................................................................. 91
8. CONCLUSION .............................................................................................. 97
8.1. Project Outcomes .................................................................................. 97
8.2. Future Work ......................................................................................... 98
Glossary ........................................................................................................ 99
References .................................................................................................... 102
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1 Requirement Analysis by Attack Trees for AMI</td>
<td>38</td>
</tr>
<tr>
<td>Table 2 Threat Table</td>
<td>51</td>
</tr>
<tr>
<td>Table 3 Common security vulnerability Techniques</td>
<td>71</td>
</tr>
<tr>
<td>Table 4 Hot Spots</td>
<td>75</td>
</tr>
<tr>
<td>Table 5 Sanitizing Methods</td>
<td>92</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1 Concept of Smart Grid ................................................................. 1
Figure 2 Vulnerability removal filters ......................................................... 2
Figure 3 AMI components ........................................................................... 11
Figure 4 Generic Open ADR Interface Architecture .................................... 12
Figure 5 Conventional view of Plug in Hybrids .......................................... 16
Figure 6 V2G concept ................................................................................ 17
Figure 7 Smart Grid = Electric Grid + Intelligence .................................... 20
Figure 8 Potential Software Supply-Chain Paths (Initiation and Development) ....33
Figure 9 Architecture and design review approach .................................. 36
Figure 10 AMI Attack tree ........................................................................... 38
Figure 11 Iterative Threat Modeling process from Microsoft ..................... 45
Figure 12 General Automated Events Architecture with Standalone DRAS ....47
Figure 13 DRMS High-Level Functional Decomposition .......................... 48
Figure 14 Primary Use Case diagram ............................................................ 49
Figure 15 Anatomy of Threat activity ......................................................... 53
Figure 16 SCADA Security Framework: RAIM Framework ....................... 56
Figure 17 Procedure to Calculate Vulnerability Indices ............................ 58
Figure 18 Expected Charge Locations ........................................................ 60
Figure 19 Class Model for RBAC- entity Classes ....................................... 62
Figure 20 Sanitization and Disposition Decision Flow .............................. 95
1.1 Background

Smart Grid consists of a large number of embedded systems (home appliances, cell phones, PDAs, sensors, Smart Cards) and other network devices connected to each other by communication channels. These systems are interactive with the physical world and thus considered as critical infrastructure. Figure 1 illustrates the concept of Smart Grid. These independent systems are going to be collaborated under the Smart Grid umbrella.

Currently, these systems are migrating from the proprietary solutions to open standard and from standalone systems to networked environment which increases the concern of security threats in these systems and as a whole the smart Grid System. Therefore, there is a need to build – in security into the Smart Grid system instead of treating security as.
an add-on. Incorporating security into the System Development Life Cycle (SDLC) of Smart Grid will lead to stronger security and cost reduction as most of the vulnerabilities are going to be resolved in the early phases of the Development Life Cycle. It will help in the efficient decision making, earlier risk assessment and mitigation. Figure 2 below provides a basis for the security development steps and how different vulnerabilities can be discovered and mitigated in each phase of SDLC [3].

Microsoft has created the Security Development Lifecycle (SDL) [13] as a security assurance process that focuses on the software development. SDL aims to reduce the number of vulnerabilities in software by introducing security and privacy practices throughout all the phases of development process. These security and privacy practices are based on the Microsoft technologies like SQL server, ASP.NET, Microsoft windows.
These practices can be used for the secure development of applications and software based on Microsoft technologies but in general the concepts can be used for other technologies as well. Similar to this NIST described security considerations for the SDLC [54] which describes the key security roles and responsibilities needed in the development cycle. The document describes different security control gates, documents and artifacts required during each phase of the development lifecycle which act as the key decision controls for the security and privacy of an information system. NIST discusses the high level processes and management roles for building security into the SDLC of an Information System. Based on these above described works a Security Development Lifecycle for Smart Grid is created in this project which includes the key practices for each phase which helps in building security into the system.

1.2 Statement of Problems

It has been reported by Microsoft that over 70% of the attacks happen through the application layer and because 75% of the organizations do not carry cyber-security insurance it costs a huge amount of money if the application got compromised. Other vulnerabilities occur in the network layer\(^1\). The network layer can be attacked by the spoofing the network packets etc. Some protocols like the Internet protocol (IP) do not have any method that validates the authenticity of the packet’s source. This implies that

\(^1\)Rocky Heckman “Microsoft Application Threat Modeling”, Microsoft Security Seminar 2006, Australia.
the attacker can forge the source address to any other address [55]. If a vulnerability gets exploited the associated costs with it is:\(^2\):

- Cost of application being down (lost sales, etc.)
- Cost of deploying incident response team
- Cost of developing patch
- Cost of testing patch
- Potential regulatory fines
- Risk of litigation
- Reputation risk to company

Smart Grid contains different web applications, network and hardware devices which if compromised can cost billions of dollars to the organization and even life to the users or customers. Therefore, in order to avoid this huge cost and reputation lost for an organization it is essential to include security practices early in the SDLC of the Smart Grid.

1.3 Project Objectives

One of the main objectives of the project is to study the best practices to incorporate in the SDLC for Smart Grid. These practices can help in early analysis and mitigation of the risks introduced at different stages of the lifecycle. Some of these practices or techniques

ensure secure Supply chain of the Smart Grid System. This project will not discuss all the best practices for the secure development of the system but only some of the important ones that can be followed for building stronger security.

1.4 Report Organization

This report is divided into eight chapters.

Chapter 2 provides introduction about Smart Grid and its components that make it a critical infrastructure.

Chapter 3 discusses the first phase i.e. Training and Initiation and security practices need to be followed this phase.

Chapter 4 discusses the Acquisition and Development phase of the Smart Grid SDLC. This phase includes security practices like threat modeling, attack trees and Real time Monitoring, Anomaly Detection, Impact Analysis and Mitigation Strategies (RAIM) framework for the initial requirement analysis and development.

Chapter 5 Implementation and Assessment discusses practices like secure coding, Fuzz testing to implement and validate for security.

Maintenance and operations phase in chapter 6 contains essential security practices required to secure the maintenance and operations of the Smart Grid applications and components.
Chapter 7 deals with Disposal phase which is the last phase of the SDLC and which concludes the lifecycle of an application by sanitizing different media and securing the decommissioned systems.

Finally, Chapter 8 concludes the project by providing the overview of the project outcomes and future work need to be done.
Chapter 2

SMART GRID

2.1 Introduction

Till now our traditional electric system was serving us well. We were satisfied with the facilities provided by our electric system. Think of electricity at different places like home, hospitals, railway stations, banks, streets etc. Can we think about what would have happened or where we could be without the electricity? As the requirements increase, the electric infrastructure keeps on changing with the more efficient devices and controls. Nowadays, everyone is concerned about renewable energy, green house effect etc. Today electricity is being generated by coal, nuclear compounds and other mechanical processes but we cannot use these resources on a wider scale until they are somehow connected to each other or communicate with each other. In order to connect different sources of electricity and their production centers, a significant amount of software and hardware devices are required. Smart Grid has additional features which can revolutionize the electric industry. According to the Department of Energy [1],

“The electric industry is poised to make the transformation from a centralized, producer-controlled network to one that is less centralized and more consumer-active. The move to a smarter grid promises to change the industry’s entire business model and its relationship with all stakeholders, involving and affecting utilities, regulators, energy service providers, technology and automation vendors and all consumers of electric power.”

With the inclusion of Internet and other communication protocols to the electric industry will help to turn the concepts like “Prices to devices” into reality and the help in two way
communications [1]. Some of the benefits that Smart Grid can bring to the electric industry are [2]:

(i) Improve the grid reliability: Smart grid will reduce the duration of blackout and power outages.
(ii) Reduce the price of the electricity: It will reduce the price of electricity by providing an interaction between the consumer and the supplier.
(iii) Offer new Products and Services.
(iv) Improved operational efficiency.
(v) Improved Security and Safety: If implemented correctly, Smart Grid can improve the security and safety by reducing the vulnerability of the grid and uncertainty of the hazards.
(vi) Promote Environmental Quality by deploying the Renewable Sources.

2.2 Main Components of Smart Grid

Smart Grid companies are trying to accommodate four major infrastructures into Smart Environment. These are: Advanced Metering Infrastructure (AMI), Demand Response (DR), Supervisory Control And Data Acquisition (SCADA) and Plug in Electrical Vehicles (PHEVs). A brief introduction of these components is provided in the next part of the chapter.
2.2.1 Advanced Metering Infrastructure

AMI is the key element in the Smart Grid Infrastructure. According to Advanced Metering Infrastructure (AMI) Security (AMI-SEC) Task force [4], “It is the convergence of the Power Grid, the communications infrastructure and the supporting information infrastructure.”

AMI refers to the systems that measure, collect, analyze energy usage from advanced devices such as electricity meters, gas meters, and water meters through various communication media on request or on a pre-defined schedule. This Infrastructure includes hardware, software, communications, customer associated, systems and Meter Data Management (MDM) software [5].

The Network between the measurement devices and business systems allows collection and distribution of information to customers, suppliers, utility companies and service providers. This enables these businesses to either participate in or provide, demand response solutions, products and services. By providing information to customers, the system assists a change in energy usage from their normal consumption patterns, either in response to changes in probe or as incentives designed to encourage lower energy usage at times of peak-demand periods or higher wholesale prices or during periods of low operational systems reliability.
Components of AMI

- **Smart Meter**: The Smart Meter is the source of the metrological data as well as other energy related information. Smart Meter can provide interval data for customer loads and distributed generation. The four basic functions of Smart Meter [4] are: (a) the monitoring and recording of demand (b) the logging of power relevant events e.g. outages (c) the delivery of usage and logging information to the upstream utilities, and (d) delivering and receiving of control messages, e.g. controlling smart appliances, remote disconnect, etc.

- **Customer Gateway**: This acts as an interface between the AMI network and customer systems and appliances within the customer facilities, such as a Home Area Network (HAN) or Building Management System (BMS). It can be placed away from the smart meter.

- **Communication Network**: This is the weakest point of the Smart Grid. The information flows from meter to the AMI head end through this network.

- **Head End Software**: This System manages the information exchange between the external systems and the AMI network [6].

The figure 3 shows the components of AMI connected to other Smart Grid components like distributed energy resources, Load control devices etc.
2.2.2 Demand Response

In layman terms, demand response means active participation of the customer in the electricity markets, seeing and responding to the prices as they change during the day. Currently, the prices of the electricity are based on the flat rate determined by the utilities, no specific pricing for the particular duration of the day. According to Department of Energy [56]:

"Changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized."

One of the goals for incorporating demand response into the smart grid is to make the grid price responsive with the interval based meter rates. One of the efforts of demand response is to achieve an Open Automated Demand Response (ADR) communication
model. Open ADR is a communication data model designed to interact with Demand Response signals by automated DR actions from Energy-Management and Control Systems (EMCS), which are pre programmed, at electric consumer’s sites. Open ADR facilitates the internet based energy pricing with preprogrammed control strategies to optimize the energy use of a site without manual intervention. OpenADR architecture depicted in Figure 4 consists of a Demand Response Automation Server (DRAS) and a DRAS Client. A server provides signals corresponding to DR events to notify customers and a client at the customer’s site listens to the signals and automates signals to pre-programmed control systems.

The information flow in the OpenADR system involves [7]:

- The utility – defines the DR event and price signals for the Demand Response Server (DRS).
• DRS client – requests event information from the DRS. DRS client can be a client and logic with integrated relay or a web based service for a control system.
• Pre-programmed DR strategies determine action based on event and price.
• EMCS carries out load shed based on the DR events and strategies.

2.2.3 Supervisory Control And Data Acquisition (SCADA) System

SCADA systems are used for automation in the Smart Grid. Incorporation of the SCADA system makes the whole technology automated with a single data center for the entire control of electricity distribution and transmission. In Smart Grid, SCADA helps in increasing the operational efficiency and reliability of electricity distribution and transmission as well as cutting costs for the end users. Some of the benefits of SCADA to Smart Grid are [63]:

• Accommodates wide variety of power generation options – central, distributed, intermittent and dispatchable.
• Self healing – anticipates and instantly reports to system in order to avoid or mitigate power outages.
• Empowers the customer – interconnects with energy management systems in smart buildings to enable customers to manage energy use and reduce the energy costs.
• Optimize assets by minimizing the operations and maintenance costs

• Enables competitive energy markets – real time information, lower transaction costs and available to everyone.

This system is used to gather data from sensors and other instruments located at remote sites and to transmit and display this data at a central site for either control or monitoring purposes. SCADA systems help in the automation of different processes. The data collected from sensors and instruments is viewed on different host computers at the central site. Analog signals contain levels of temperature, humidity, pressure, flow rate and motor speed. Another layer of equipment between the remote sensors and instruments and the central computer on the remote side connects the sensors and field instruments. Sensors typically have digital or analog I/O and these signals are not in a form that can be easily communicated over long distances. The intermediate equipment is used to digitize then packetize the sensor signals so that they can be digitally transmitted via an industrial communications protocol over long distances to the central site.

The SCADA host is usually a PC running sophisticated SCADA Man Machine Interface or Human Machine Interface software. This software is used to poll the remote sites and store the collected data in its centralized database. The Plant equipment is controlled by the SCADA Host software logic and that control can be initiated by a Human or a Machine. The SCADA host is used to gather real time data from the remote locations which can be:

(i) Analog Data which is used for trending and reports generation
(ii) Digital Data which is used for alarming

(iii) Pulse which signifies the revolution of some sort of meter in the network

Due to the remoteness of the systems communication can take place through wireless or wired media [8]. Sometimes, more than one communication channels are required for backup in case one communication channel fails to send the signals from the fields to the SCADA host. One of the biggest challenges in the SCADA systems is the secure and timely actionable information sharing.

2.2.4 Plug in Hybrid Electric Vehicles

A Plug-in Hybrid Electric Vehicle (PHEV) is hybrid electric vehicle with a large battery pack which can be recharged from the electricity grid. PHEV has the ability to use electricity as an alternative fuel. Figure 5 shows the conventional view of plug in hybrids. The combination of efficiency and fuel switching capability have made PHEVs and other electric drive vehicles an important part of the transportation future. The PHEV has the qualities of the existing hybrid vehicles and it offers the cost savings and energy benefits of electric vehicles. The integration of PHEVs in the Smart Grid infrastructure could diversify the fuel sources used for automobiles, and enhance energy security by reducing dependence on foreign oil [9].
The concept of V2G (Vehicle to Grid) in Smart Grid as shown in Figure 6 offers an additional advantage of transferring energy back to the grid from a vehicle. Being a vital component for both the utility and the customer, it will allow withdrawing power from each other as needed. “Peak Load leveling is the concept of vehicle to grid technology that allows V2G vehicles to provide power to help balance loads by "valley filling" (charging at night when demand is low) and "peak shaving" (sending power back to the grid when demand is high).
It can enable utilities new ways to provide regulation services (keeping voltage and frequency stable) and provide spinning reserves (meet sudden demands for power). In future development, it has been proposed that such use of electric vehicles could buffer renewable power sources such as wind power, for example, by storing excess energy produced during windy periods and providing it back to the grid during high load periods, thus effectively stabilizing the intermittency of wind power. Some see this application of vehicle-to-grid technology as a renewable energy approach that can penetrate the baseline electric market”. [10]
Moreover, PHEVs will help in the reduction of the carbon-dioxide and other green house emissions which will help in reducing the effects of global warming.

Above discussion highlighted some of the key functions and the benefits of the Smart Grid components. The Next chapter discusses the Secure Smart Grid System Development Life Cycle. It highlights some of the security practices that need to be followed in different phases of the SDLC of Smart Grid in addition to the common security practices for the information systems.
Chapter 3

PHASE 1: TRAINING AND INITIATION

One of the biggest issues in the SDLC is the lack of security training and knowledge among the developers, testers and system designers. Poor design decisions made by the development teams can lead to overly complex design of the application. Training and Education is one of the important phases in Smart Grid development to provide software assurance training. Not only the development team but the employees at the utilities and the customers need to be a part of the Smart Grid Software Assurance training.

3.1 Employees and Customers

In order to make the developed Smart Grid system work efficiently, it is really important for the employees and the customers to know the difference between the traditional Grid and the Smart Grid infrastructure. Smart Grid is going to use the two way digital communication technologies to deliver electricity from utility to the customer locations and vice a versa. Unlike, the traditional system it is not a unidirectional flow of electricity but a bidirectional flow from homes to utilities as well as shown in figure 7.
Smart Meters installed at the customer locations monitors the customer appliances and dynamically communicate the electricity usage back to the utility for billing, demand response and scheduling and load control. Therefore, the customer and employees (who are going to use the Smart Grid infrastructure) should know how to secure these devices not only physically but other kinds of malicious attacks. One of the tactics used by hackers to get the personal identification information is “Social Engineering”.

“Social Engineering is the act of manipulating people into performing actions or divulging confidential information, rather than by breaking in or using technical cracking techniques. While similar to a confidence trick or simple fraud, the term typically applies to trickery or deception for the purpose of information gathering, fraud, or computer
An attacker interacts with employees and either tries to gain information from them or tries to trick them into doing something they should not do. For example, An attacker might be able to convince an employee to turn off the security system to get unauthorized access to the system. The same tactic can be used by a hacker to get the information about a device or a controller from an innocent customer or an employee to hack the home area network system. The utilities should provide training to convey among the customers what kind of information is sensitive and can be used as a tool by the hackers to perform an attack. Similarly, the employees should know what kind of information should be released to other person, another employee etc. The customer security training should be based on social engineering and physical security of the AMI infrastructure and can be carried out by sending brochures, information materials explaining different scenarios and consequences of releasing information to a third party. Another way of global communication is television or radio.

3.2 Development Team

The Members of the Smart Grid development team should get the appropriate security training based on their role in the development process. This training should provide information about recent attacks and threats due to different technologies and/or software vulnerabilities. Other concepts that need to be included in the training are:
Analysts: Analysts should know how to create threat models in order to help the developers and testers to understand the vulnerabilities and threats. The Training should also include the following key areas:

- **Attack surface reduction**: The attack surface of an application includes code, interfaces, services, protocols, and practices available to all users [58], with the consideration of what is accessible to the unauthenticated users. Attack surface reduction is a compromise between the perfect safety and the unmitigated risk that minimizes the code exposed to untrusted users. Attack surface reduction in addition to the code quality can help in producing more secure software [58].

- **Defense in depth**: Defense in depth is an approach to defend a system against an attack. It is a layering tactic approach to information and electronic security [59]. The term Defense in depth is taken from the military strategy that seeks delay, rather to prevent an attack. But in computer networks [59] the defense in depth should not only prevent the breach but also help in detecting and responding to an attack.

- **Principle of least privilege**: Principle of east privilege means giving a user only those privileges which are absolutely essential to do his/her work. For example [60], a backup user does not need to install software; hence the backup user has rights only to run backup and backup-related applications. Any other privileges like installing software etc. are blocked.
• Secure defaults: It means that the default configuration settings are the most secure settings possible, which are not necessarily the most user friendly settings [61]. In many cases, security and user friendliness is waged based on both risk analysis and usability tests.

• Software Supply Chain Risks Analysis for Commercial Off The Shelf (COTS) applications

• Procurement language for the COTS applications: National Institute of Standards and Technology (NIST) created a procurement language for the information systems that can be used for defining terms and conditions for the procurement of COTS applications.

Developers: Developers should get training about secure coding practices and know about the common vulnerabilities in different programming languages [13] such as the following:

• Buffer overruns: It occurs when the external input is treated as the trusted data. Copying this data using operations such as CopyMemory, strcat, strcpy, or wcscpy can cause unexpected results that may lead to system corruption.

• Integer arithmetic errors: Integer arithmetic errors like integer overflow occur when an arithmetic operation attempts to create a numeric value that is larger than can be represented within the available storage space.
• Cross-site scripting: Cross site scripting in web applications is a vulnerability that allows hackers to inject client side script into the web pages. This vulnerability is used by the attackers to bypass the access controls and gain elevated access privileges.

• SQL injection: SQL injection attack is inserting SQL query via the input data from the client to the application. The attacker can read, modify or delete data from the database leading to repudiation and data integrity issues.

• Weak Cryptography: An attacker can find a weakness in a code, cipher, cryptographic protocol or key management scheme of a Weak cryptographic module and can lead to an attack. This process is also called "cryptanalysis".

• Unmanaged code issues: Unmanaged code runs by itself, calls and uses routines in the operating system. Hackers can exploit the unmanaged code and run the malicious code to gain access to an application.

Testers: Testers’ training involves the practices and techniques to test the security of the application. Testers should know how to test the software based on threat models [13].

• Security functionality testing: Functional security testing ensures that software behaves as it should. If one of the security requirements of a system states that the length of user input should be certain characters long. Then functional testing is the part of the process that this requirement is implemented and works correctly.
• Risk assessment: For testers risk assessment deals with the identification of risks and applying appropriate mitigation strategies to the identified vulnerabilities. Test cases are then written based on the risks identified.

• Test methodologies and automation: Testers should know about different security testing methodologies such as risk driven testing, penetration testing, fuzz testing, regression testing and compliance testing in order to test different security features of an application.
Chapter 4

PHASE 2: ACQUISITION AND DEVELOPMENT

In this phase of SDLC, the project team concentrates on identifying the systems requirements (functional and non functional requirements including the security requirements). Overall need and the goals of the Smart Grid components are identified in this phase.

4.1 Best Practices for Acquisition and Development

4.1.1 Identify Security Requirements

The analysts identify all the security requirements and security goals of the application, verify and validate them in terms of feasibility and ambiguity. In addition to that the teams are heavily involved in identifying the threats related to different components and detailed risk assessment and exploring design alternatives. Some components in the Smart Grid system need to be acquired from different vendors. Therefore, different acquisition and procurement requirements need to be defined for the system.

Based on different activities involved in this phase the security team analyses and identifies different requirements that may lead to security breach or violates any security principles (i.e. Confidentiality, Availability and Integrity). These different types of requirements which may involve internal or external security threats are:

- Identification Requirements: These types of requirements identify the extent to which a Smart Grid application or its component identifies its externals before interacting with them. Some of the examples are: An application should identify
its client applications before allowing them to use its capabilities. The data center shall identify all the personnel before allowing them to enter.

- **Authentication Requirements**: These requirements specify the extent to which a Smart Grid application or its component shall verify the identity of its externals before interacting with them. Examples are: The application shall verify the identity of all the users before allowing them to update their user information. The application shall verify the identity of the user before accepting the credit card payment from the user.

- **Authorization Requirements**: These are the requirements that specify the access and usage privileges of authenticated users and the client applications. For example: The application shall allow each customer to obtain access to their personal account information.

- **Immunity Requirements**: These requirements specify the extent to which a Smart Grid application or its components shall protect itself from infection by unauthorized undesirable programs (computer viruses, worms and Trojan horses). Examples are: The application shall protect itself from infection by scanning entered or downloaded data and software for known computer viruses, worms, Trojan horses and other similar malicious software.

- **Integrity Requirements**: These requirements specify the extent to which Smart Grid application or its components shall ensure that its data and communications are not intentionally or unintentionally corrupted via unauthorized creation,
modification or deletion i.e. the application shall prevent the unauthorized corruption of emails.

- Intrusion Detection Requirements: These requirements specify the extent to which a smart Grid application or component shall detect and record attempted access or modification by unauthorized individuals. Examples are: The application shall detect and record all the attempted accesses that fail identification, authentication, or authorization requirements.

- Non-Repudiation Requirements: These requirements specify the extent to which an application or component shall prevent a party to one of its interactions from denying having participated in all or part of the interaction. Examples are: The application shall make and store tamper-proof records of the contents changed in the transaction, date and time of the transaction, identify the customer involved in the transaction.

- Privacy Requirements: These requirements specify the extent to which application shall keep its sensitive data and communications private from unauthorized users and programs. Examples are: Personal Identification should not be stored, should have communication privacy, and data storage privacy.

- Security Auditing Requirements: These requirements specify the extent to which a business, application or a component shall enable security personnel or an application to audit the status and use of its security mechanisms. Examples are:
The application shall collect, organize, summarize, and regularly report the status of its security mechanisms including:

- Identification, Authentication, and Authorization.
- Immunity.
- Privacy.
- Intrusion Detection.

- Survivability Requirements: These requirements specify the extent to which an application or component shall survive the intentional loss or destruction of a component. The application either fails gracefully or continues to function even though certain components have been damaged or destroyed. Examples are: The application shall not have a single point of failure. The application shall continue to function (possibly in degraded mode) even if a data center is destroyed.

- Physical Protection Requirements: These requirements specify the extent to which an application or component shall protect itself from physical assault. Examples are: The data center shall protect its hardware components from physical damage, destruction, theft, or surreptitious replacement. The data center shall protect its personnel from death, injury, and kidnapping.

- System Maintenance Security Requirements: These requirements specify the extent to which an application or a component shall prevent authorized modifications from accidentally defeating its security mechanisms. Examples are: The application shall not violate its security requirements as a result of the
upgrading of a data, hardware, or software component. The application shall not violate its security requirements as a result of the replacement of a data, hardware, or software component. [14]

- Procurement Requirements: These requirements specify the extent to which a third-party application should follow the security policies and procedures designed for an application or infrastructure. These requirements define the security procedures to follow and minimum security requirements need to be fulfilled by any COTS hardware or software product before its procurement. These requirements include identification of the development process management, architecture and design of the COTS application, software development, component assembly, testing, installation and acceptance, Built-In software defenses, software manufacturing and packaging, support, operating environment for services and SLAs (Service Level Agreements), security monitoring and timeliness of vulnerability mitigation requirements for the third party involved in the development of the Smart Grid components. These requirements act as the instructions to the suppliers how to incorporate security in the SDLC at their end. [15].

- Cryptographic Requirements: These requirements specify the key encryption and decryption levels, keys sharing mechanisms or algorithms for an application. These requirements identify the cryptographic modules and assign them different levels of security based on the information they receive and send.
4.1.2 Conduct Risk Assessment

Risk Assessment is fundamental to the security of Smart Grid. It ensures that the controls and expenditures are fully commensurate to the risks to which the Smart grid organization is exposed. It ensures that the product is in compliance with the security policies and procedures. Different methods are being used in the industry for the security risk analysis. One of the important aims of the risk assessment is to put the process of determining appropriate and cost effective controls to objective basis. There are two different approaches for risk analyses are: Quantitative Analysis and Qualitative Analysis.

Qualitative Analysis can be performed by the understanding the organization and identifying the people and the assets at risk. Then the loss risk events/vulnerabilities can be identified e.g. crime related events, consequential events (events that can impact the security in other components or software even if they are not directly related to each). Then a probability of loss risk and frequency of events is calculated. For example, a business that has a history of criminal activity both at and around its property will likely have a greater probability of future crime if no steps are taken to improve security measures and all other factors remain relatively constant (e.g., economic, social, political issues).

Quantitative Analysis can be performed by calculating the loss event probability of a particular event. The higher the probability more loss is associated with that event. Based
on the probability the events can be characterized as virtually certain, highly probable, less probable or probability not known. [16]

4.1.3 Supply Chain Risk Analysis

Due to the complexity and diversity of software and hardware in the Smart Grid System, it is almost impossible to build everything in house. Different components can be acquired from different outsourced teams, organizations, private companies etc. The procurement of hardware components includes supply chain risks such as manufacturing and delivery interruptions and substitution of different substandard components. Similarly, Software supply chain also includes the risks of tampering and introduction of software defects. In this phase of the Smart Grid SDLC the security analysts’ team should frame different policies and procedures to identify and overcome supply chain risks. Supply chain process is global and involves different entities at different steps. Defects can be introduced to the applications at any step of the supply chain which may not be visible to others. Various ways in which software and hardware security risks are introduced into the supply chain include:

- Coding and design defects during development
- Improper control of access to a product or service when it is transferred between organizations
- Insecure deployment or configuration and distribution
- Use of product or application that introduce coding defects or configuration changes that allow security compromises

Therefore, it is important to take the supply chain risks into consideration early in the development cycle. In this initial phase the analysts may start with analyzing different paths a software or hardware components can take and analyze the risks associated with each path. This would help to identify potential vulnerabilities. These vulnerabilities can be taken care of in the next phases of the development cycle e.g. it can help in designing test cases and compliance requirements of the application. Figure 8 shows the different supply chain paths an application or component can go through. Each layer can be responsible for inserting different defects for future exploitability.

![Figure 8: Potential Software Supply-Chain Paths (Initiation and Development)](image-url)
Design and coding defects in the operations may need special security monitoring and patches to avoid future security issues. Supply chain security issues can be reduced by controlling the way by which security risks can be introduced into the application or product during the acquisition and development phase:

- Supplier’s capability: Suppliers should have good security and management practices in place.
- Product security: Risk Assessment of the product should be performed for critical security compromises and mitigation requirements.
- Product Logistics: While in transit the product should have proper access control.

The analysis of supply chain risks should go beyond the product and the supplier assessment and also include the deployment and operations. [17]

4.1.4 Define third party code licensing security requirements

All third party contracts for different products and services of the Smart Grid should include a provision for licensing contract that requires any third party to provide proof of compliance with the security policies and procedures. It should include the requirements such as verification of the code from independent party, security tools output and logs, and source code itself for verification and validation [13]. This Process also includes framing policies and regulations for system certification and accreditation. The third party should be provided with terminology used in the Procurement documents. A
standard procurement language should be used so that terms and conditions are unambiguous and pertains same meaning for different users of the document. Department of Homeland Security (DHS) created a procurement language for the control systems which can be used in the procurement of different Smart Grid services and Products [18].

According to DHS [18],

“Over the years, these systems have gone from proprietary, stand-alone systems, to those that use commercial off-the-shelf (COTS) hardware and software components. With the increase of more commonly used hardware and software, comes the potential for information technology (IT) vulnerabilities to be exploited within the control systems environment............. The goal is for federal, state, and local asset owners and regulators to obtain a common control systems security understanding; using these procurement guidelines will help foster this understanding and lead to integration of security into control systems.”

4.1.5 Design and Evaluate Security Architecture

An architecture and Design review should be performed in order to identify the security vulnerabilities that may cause cascading impact on the later phases of the SDLC. The Security architecture review should include different aspects of security. Figure 9 shows three major aspects to consider while conducting an architecture and design review for security are:
• Deployment and Infrastructure: The design of the application should be reviewed with respect to the target deployment environment and its associated security policies and procedures. The evaluation should also consider the constraints imposed by the underlying infrastructure-layer security and the operational security practices being used by the application or system.

• Security frame: Critical areas of the application need to be reviewed. An effective way to review the critical areas is to focus on the set of categories that have the most impact on security, particularly at an architectural and design level, and where mistakes are most often made. The security frame aspect of the design review describes these categories. They include authentication, authorization, input validation, exception management, and other areas. Frame can be used as a
roadmap to perform review consistently in order to ensure that all areas are covered during the review.

- Layer-by-layer analysis: Logical layers [19] of all the applications must be reviewed and security should be evaluated in the presentation, business and data access logic.

4.2 Different Techniques for Acquisition and Development for Smart Grid Components

Based on different characteristics each Smart Grid component possesses unique methodologies that can be applied for the acquisition and development of these components. Some of them are discussed below:

4.2.1 Attack Trees for AMI

One of the important concerns with AMI infrastructure is energy theft. The security modeling technique of attack trees is really effective in understanding the strategies for energy thefts in AMI. The attack tree recursively breaks the malicious user’s intent or goal into the sub goals. The root node identifies the single goal of all attacks in the tree. The leaves having no descendents represent the specific attacks that must take place for the goal to be achieved. The manipulation of the paths to the root goal with the logical operators AND and OR as shown in figure 10 which determine whether one or all of the children in a given internal node need be completed in order to achieve the goal [20].
The figure depicts the attacks which are required for committing the energy theft. There are three main ways an attack can be performed for the energy theft.

First, the branch of the figure shows that the theft can take place before the meter makes a demand measurement.

Second, before/after demand values are stored in the meter and

Third, after measurements and logs have left the meter in transmission to the utility [20]. Therefore, AMI threats can be easily identified by the attack trees because of different qualities of these trees:
• Individual attack trees can be composed to achieve specific goals: E.g. an adversary that attempts to cause rolling blackouts may have a sub goal of forging energy demand at distribution substation meters. Attack trees also provide a useful information for identifying both the root causes of attacks as well as the “low-hanging fruit” that is likely to be exploited.

• Easily identifies the parties involved in the theft such as customers, organized crime, utility insiders, Nation States etc.

• Helps in identifying the mitigating tactics for the attack or theft

• Helps to reduce attack surface

• Identify the vulnerabilities in the network, e.g. in the figure 10 there are three ways in which the demand data can be tampered: (a) while it is recorded (via electromechanical tampering) (b) while it is rest at the meter and (c) in the flight across the network.

• Helps in prioritizing the requirements based on the risk associated with the threat

Table 1 summarizes a brief analysis of AMI components based on the types of attacks or threats they may have and the requirements that can identified by the attack trees.
Table 1: Requirement Analysis by Attack Trees for AMI

<table>
<thead>
<tr>
<th>AMI Component</th>
<th>Types of Attack or Threat</th>
<th>Types of Requirements identified by Attack Trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meter</td>
<td>Viruses</td>
<td>Immunity requirements, Intrusion detection requirements, Privacy requirements.</td>
</tr>
<tr>
<td></td>
<td>Trojan Horses</td>
<td>Non-repudiation requirements, Privacy requirements, Authentication and Authorization requirements, survivability requirements, Immunity requirements.</td>
</tr>
<tr>
<td></td>
<td>Trapdoor</td>
<td>Physical protection requirements</td>
</tr>
<tr>
<td></td>
<td>Service spoofing</td>
<td>Confidentiality requirements, Integrity requirements, security Auditing requirements, physical protection requirements.</td>
</tr>
<tr>
<td></td>
<td>Theft</td>
<td>Confidentiality Requirements</td>
</tr>
<tr>
<td></td>
<td>Tampering</td>
<td>Confidentiality requirements, Integrity requirements, Availability requirements, Non-Repudiation requirements, Immunity requirements, Security Auditing requirements, Privacy requirements, System Maintenance Security requirements.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical Protection requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Non repudiation requirements, Integrity requirements.</td>
</tr>
<tr>
<td>Network and communication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eavesdropping</td>
<td></td>
<td>Confidentiality Requirements</td>
</tr>
<tr>
<td>Traffic Analysis</td>
<td></td>
<td>Confidentiality requirements, Integrity requirements, Availability requirements, Non-Repudiation requirements, Immunity requirements, Security Auditing requirements, Privacy requirements, System Maintenance Security requirements.</td>
</tr>
<tr>
<td>EM/RF Interception</td>
<td></td>
<td>Physical Protection requirements</td>
</tr>
<tr>
<td>Media Scavenging</td>
<td></td>
<td>Non repudiation requirements, Integrity requirements.</td>
</tr>
</tbody>
</table>
4.2.2 Threat Modeling for Demand Response

Demand response involves communication between different components such as the Energy Management System, Demand Response server, Metering infrastructure and the Distribution Generation. It is basically an information flow from one component to another with different events involved in the process. Type of information that flows through the Demand Response Systems include:

- Price based information
- Event based information
- Bidding information

Therefore, one of the critical security requirements of the demand response infrastructure is to secure the communication among its components.

Firstly, the information sent between the entities needs to be confidential and protected from unauthorized access as it can lead to the leaking of the customer information to the adversary.

Secondly, the authentication of the different devices is required in order to provide timely response to the demand response event signals. If one of the components of the demand response infrastructure fails to authenticate with the DR control services, it will not be able to connect to or respond to the DR event signals in order to protect from the unauthorized devices to communicate the DR system, such as hijacking of the meter.

Thirdly, data integrity is critical to this system in order to avoid unauthorized manipulation of the demand information and the control signals. An unauthorized user
can manipulate the control signals to manage devices and control the usage of the meter by inducing an inappropriate response e.g. turning on or off the electrical devices or shutting down DR operation. This can lead not only to financial impacts to the utility but can put the customer’s life at risk.

Fourthly, availability and accountability of the information is another aspect of demand response to deal with. Real time pricing and real time load use being the major part of the demand response infrastructure needs to be available all the time to the customer and the utility. Unavailability of the system may lead to the wrong or delayed actions to different events and can impact the customer and the markets. Failure to hold account of the event or action taken by the communicating parties because of the invalid meter, energy management system, or DR services information may result in dispute among different parties.

Some of the high level requirements of the Demand Response (DR) System are:

- Information transmitted must be protected from unauthorized access, inspection, and modification from unintended users.
- Information transmitted either to or from the Demand Response Automated Server (DRAS) must maintain confidentiality and integrity from third parties.
- The DRAS must provide accountability for the Prices received by participants, DR events received by participants and their history and Bids
submitted by participants. It should maintain confidentiality of participants, utilities and ISOs.

Proper access control to the information stored on the DRAS must be provided so that only authorized users can modify it [21].

In order to overcome all the shortcomings of the events and fulfill all the security requirements of the demand response system, it is really important to include threat modeling technique in the SDLC of the Smart Grid for the demand response infrastructure.

Threat modeling is a security analysis methodology that can be used to identify requirements and risks, and guide subsequent design, coding, and testing decisions. The methodology is mainly used in the earliest phases of a project, using specifications, architectural views, data flow diagrams, and activity diagrams. But it can also be used with detailed design documents and code. In the requirements analysis phase it helps in identifying the security requirements and the mitigation strategies for the threats. Threat modeling addresses those threats which have the potential of causing the maximum damage to an application [22].

Threat modeling can be used for demand response infrastructure to identify the security requirements and the risks associated with them. This type of modeling can be used at different levels of abstraction. It can be used at the component level as well as the application level. The distributed nature of the applications and systems in demand response environment makes it really hard to study as a whole. Different components of
the demand response are either acquired or commercial-off-the-shelf applications. Communication among different components can be carried through the web services using the web services description Language (WSDL) and the eXtensible Mark-up Language (XML) languages. It is really important to identify the threats at the component level as well as at the application level and the web services level. Threat modeling helps in dividing the system into different domains, components, assets, applications etc. and analyzes the threats in that domain or area and any other dependencies from the outer domains. General approaches to threat modeling are:

- Asset Centric
- Attacker Centric
- Software Centric

It is an iterative process, shown in figure 11, which can lead to threat at the root or the basic level. The process goes through iterations until an indecomposable threat task is achieved.
Important steps involved in the threat modeling process are:

- Identify assets: This entails understanding every component of the demand response system and its interconnections, defining usage scenarios, and identifying assumptions and dependencies. The analyst must have the thinking of the adversary in order to fully utilize the concept of threat modeling in demand response. Based on a thorough understanding of the process and the systems, the
next task is to identify the critical assets in the system those can potentially be attacked by the hackers or the adversary. An access point is the point of entry to the system that can be used by the adversary to enter and exploit the system. Therefore, it is really important to identify all the components of the systems and identify the critical assets and the access points.

- Create an architecture overview: The next step is to create different architectural views for the system based on their functionalities and types of communication between the components. These different architectural views will help in identifying the trust boundaries of the system. “A trust boundary is a boundary across which there is a varied level of trust. For example, the network may form a trust boundary, as anyone can gain access to the Internet, but not everyone should have access to the enterprise system. Related to trust boundaries are trust levels. Trust levels indicate how much trust is required to access a portion of the system. For instance, if a user is an administrator, they are trusted to do more than normal users [24]. As the Demand Response system consists of diverse applications and platforms in which they are designed it is critical to identify the technologies involved so that the major vulnerabilities can be considered in the threat analysis phase of the requirement analysis of the Demand Response [23].

Figure [12] illustrates different boundaries and events that can take place in a specific boundary. The boundaries can be the utility, participant site and the Demand Response Automation server (DRAS). These boundaries helps in
identifying the different events hosted in a specific boundary and the type of information flowing through different components. Figure 12 shows different events associated with the DRAS, participant site and the utility.

Figure 12: General Automated Events Architecture with Standalone DRAS [25]

- Decompose the application: In order to do a top down analysis of the demand response system should be decomposed into components. The top down analysis of the demand Response system is really important as the whole system is composed of really complex components such as pricing, scheduling, bidding process, control systems etc. Each component is unique and requires further analysis. Each component has different security and privacy requirements e.g.
systems at the utility level require different access privileges than the systems at the customer or participant site. Moreover, these systems have different forms of authentication and authorization requirements. System context diagrams and functional diagrams can be used to point out the different core functionalities and supported functionalities of the different parts of the demand response flow [25]. Figure 13 shows a high level functional decomposition of the demand response management system.

Figure 13: DRMS High-Level Functional Decomposition [25]

Therefore, one of the important parts of the threat modeling process is to define components and their functionalities well in order to identify all the threats associated to these unique components.

- Identify and categorize the threats: One of the important outputs of the threat modeling process is the identification of the threats and the causes of threats. In
this task, demand response applications are thoroughly verified with the help of data flow diagrams, system context diagrams etc. to identify the threats to those systems, system components, or the information flowing through those systems or communication channel. During this activity identify all the network threats, Host threats and application threats. Threat modeling also helps in categorizing the threats from the misuse cases. Misuse cases or abuse cases can be drawn from the use cases of the application or component. Misuse cases often triggers a chain reaction for a threat which makes it easier to identify non functional requirements. Figure 14 shows the primary use case of the Demand Response System.

Figure 14: Primary Use Case diagram [25]
To identify the threats to an application from the use cases, the analyst should have the qualities like critical thinking and thinking like an adversary. These misuse cases can be used to identify the threats to the Demand Response System. Some of the common threats can be:

- Attacker eavesdrops on unencrypted communication between company server and customer.
- Attacker obtains valid login credentials to demand manager company network.
- Attacker uses credentials to access Company’s network and customer information database.
- Attacker exploits a web application vulnerability to access customer data via the Internet [25].

These models help to identify the types of attackers and in turn allow furnishing policies or strategies to avoid security issues created by these different attacker profiles. The identified threats should have an effect related to it. The threat effect value can be one of the STRIDE values (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privileges) [13].

- Rank or prioritize the threats: Using the DREAD categories (Damage potential, Reproducibility, Exploitability, Affected users, and Discoverability), the threats can be ranked as the highest, medium or lowest priority. Table 2 categorizes threats in different priorities based on the DREAD categories. While assigning the priorities and ranking Table 2 can be used:
Table 2: Threat Table [13]

<table>
<thead>
<tr>
<th>Rating</th>
<th>High (3)</th>
<th>Medium (2)</th>
<th>Low (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D Damage potential</td>
<td>The attacker can subvert the security system; get full trust authorization; run as administrator; upload content.</td>
<td>Leaking sensitive information</td>
<td>Leaking trivial information</td>
</tr>
<tr>
<td>R Reproducibility</td>
<td>The attack can be reproduced every time and does not require a timing window.</td>
<td>The attack can be reproduced, but only with a timing window and a particular race situation.</td>
<td>The attack is very difficult to reproduce, even with knowledge of the security hole.</td>
</tr>
<tr>
<td>E Exploitability</td>
<td>A novice programmer could make the attack in a short time.</td>
<td>A skilled programmer could make the attack, then repeat the steps.</td>
<td>The attack requires an extremely skilled person and in-depth knowledge every time to exploit.</td>
</tr>
<tr>
<td>A Affected users</td>
<td>All users, default configuration, key customers</td>
<td>Some users, non-default configuration</td>
<td>Very small percentage of users, obscure feature; affects anonymous users</td>
</tr>
<tr>
<td>D Discoverability</td>
<td>Published information explains the attack. The vulnerability is found in the most commonly used feature and is very noticeable.</td>
<td>The vulnerability is in a seldom-used part of the product, and only a few users should come across it. It would take some thinking to see malicious use.</td>
<td>The bug is obscure, and it is unlikely that users will work out damage potential.</td>
</tr>
</tbody>
</table>

Ranking and prioritization of the threats also help in decision making during the process of Capital Planning and Investment Control (CPIC), as the funds are not available at a time to implement all the requirements at once.

- **Specify Security Requirements:** The security requirements are then defined in order to match the risks to the protection costs. All these requirements must map to the identified misuse cases. [Source: eliciting security requirements with misuse cases]. “This process of identifying critical assets, threats and security requirements is cyclic. On the one hand, the critical assets defined in the larger process drive the identification of threats in the security process. On the other hand, the threats identified in the security process drive the definition of new security requirements which, when implemented, may create new vulnerable assets.” [26].
4.2.3 RAIM Framework for SCADA

Being the central nerve system of the Smart Grid, SCADA Systems are identified as one of the critical infrastructures. Other risks which make these systems more vulnerable to attacks are:

(i) Greater complexity which increases the exposure, potential attackers and unintentional errors

(ii) Linked networks introduce common vulnerabilities

(iii) “Denial of Service” – type attacks

(iv) Increased number of entry points and paths

(v) Compromise of data confidentiality or customer privacy

(vi) Disruption of IT equipment by Electro Magnetic Pulse (EMP) and Geo-Magnetically induced [27]. A variety of communication systems are deployed inside the SCADA systems for monitoring and control of different processes. The analog, digital and status data acquired by the SCADA systems is used by the Energy Management System (EMS) to perform a wide range of functions which also includes the Real Time Control at the control center. Failure of a communication channel due to the component failure or communication delay in the operational environment could lead to power failures or control other facility operations [28].

Figure 15 shows different types of attack agents that can perform an attack on the SCADA systems and its subsystems. An attack can be performed by a knowledgeable
person who knows the inside out of these systems or by a disgruntled Employee. Some of the characteristics which make these systems vulnerable to attacks are:

Time critical, Embedded, Distributed, Intelligent, Large, Open, Heterogeneous

Figure 15 also explains the threat activity on a SCADA system. Like others, this system has insider and outsider threats which can cause attack on the system, by the system and through the system. In order to overcome all the limitations of the SCADA systems, a thorough study of the system is required. To study these systems we need to know all its operations and real time functionalities. The key is to identify the system properties and understand the threats and impact they can make.

Figure 15: Anatomy of Threat activity [29]
Chee-Wooi Ten, Govindarasu Manimaran and Chen-Ching Liu [30] developed a security framework for the SCADA systems that can be used in the requirement analysis phase for these systems to identify all the security requirements at the component level. This framework can be used in the iterative system development lifecycle models in which the requirements are analyzed and coded as the system is built and analyzed. Each iteration consists of security requirements analysis, design, coding and testing. The results from the assessment and testing phase leads to the next level of the iterations. Therefore, this framework can be used for the agile development model. Interdependence modeling is used as a key to identify the threats and the impacts/mission of the attacks. This Framework is also useful in economic impact analysis which helps to identify the appropriate measures that mitigate risks at the pivotal network nodes. The framework consists of four components, shown in the figure below [30]:

(i) Real Time Monitoring: A variety of information networks are related to SCADA which involves control center, intelligent electronic devices at substations, distributed sensors that measure the electrical and other quantities and a variety of communication links between the control center and the substations. In this process all the real time processes and information is analyzed which includes all the security logs, event logs, and file integrity logs. Other physical aspects of these real time devices are also studied which involves the critical alerts and system health messages. Then information is gathered and potential evidences are extracted from the results. E.g. The denial of service attacks is critical to these
systems as they can lead to the resource exhaustion. Packet flooding is one of the typical resource exhaustion attacks. Denial of service attack can cause serious effect on the SCADA system and its critical functions like state estimation, alarm processing and preventive controls etc. Therefore, all the events and logs are monitored and evaluated for the evidences and for the further study of the evidences in the Anomaly Detection phase. These evidences are also used to follow different preventive action and mitigation strategies to overcome the attack or threat.

(ii) Anomaly detection: Correlation techniques are used for the Anomaly detection to systematically establish the relationship between statistical data sets from various sources like cyber related event logs and other audit data from power instruments. Categories of event correlation that can be used in the power infrastructures are: (a) Temporal correlation (b) Spatial correlation and (c) Hybrid correlation. The combination of these correlations identifies threats that may capture local or global abnormality. Different system logs are required for correlation for anomaly detection. Some of the systems are:

Communication systems: The logs from the communication systems consist of status of communication server to all IEDs such as communication link failure or degradation of the expected throughput, irregular frequency, and volume of usage on a specific application etc.
Computer systems: These types of logs include the alarms of intrusion attempts with respect to the attempt frequency to each system, number of rests, shutdowns or stopping etc.

The Anomaly detection phase provides an output for formulating a hypothesis and different remedial actions to follow to mitigate the threat [30]. Figure 16 illustrates all the four parts of the RAIM framework and techniques associate with each phase.

Figure 16: SCADA security Framework: RAIM Framework [30]
(iii) Impact Analysis: In this phase the intrusion behavior of the cyber-attacks on SCADA system are analyzed and their respective consequences are evaluated. If a cyber-security of SCADA systems is compromised it can lead to serious damage by causing loss to load or equipment damage. An integrated risk modeling technique that includes both power control system vulnerabilities and the resulting impacts on the real time operation of the power system is used. This technique has four key steps:

- **Cybernet**: Network that incorporates combinations of intrusion scenarios into the SCADA systems.

- **Power flow simulation**: Captures the steady state behavior of the power system under a CyberAttack. The compromised system is isolated from the network and is evaluated.

- **Vulnerability index calculation**: Vulnerability index for a scenario is calculated by multiplying steady state intrusion probability (obtained from cybernet analysis) and impact factor (obtained from power flow simulation analysis). Figure 17 shows the procedure to calculate the vulnerability indices with attack tree modeling:
Figure 17: Procedure to Calculate Vulnerability Indices [30]

- Security improvements: based on the vulnerability assessment results, security improvements to the SCADA systems are done.

(iv) Mitigation Strategies: The Output from the above three phases can be used as a guideline for framing different mitigation techniques for different threats found in the analysis. The nature of mitigation techniques depends on the following:
  - Intrusion Attempts
  - Intruded Scenario
  - Ongoing Denial of Service Attack [30]

4.2.4 UML/OCL for Plug in Electric Vehicles

Incorporating PHEV into the Smart Grid will provide:
  - Freedom to travel and fuel as needed without boundaries
Ability to use easy, convenient, and familiar systems to purchase electrical energy (cash, credit cards or pre-paid)

Reasonably priced energy

Convenient charging locations

Obtain charge within a reasonable time period [31].

Therefore, it raises the question of providing appropriate access to an authorized user and availability of the Smart Grid system for billing and charging locations. Some of the capabilities that a Smart Grid Infrastructure should possess in order to successfully accommodate Plug In vehicles are [32]:

- Coordinating energy transfers with financial transactions
- Authentication –consumer identification; transaction security
- Authorization –consumer eligibility to buy or sell electricity
- Accounting –amounts of energy bought or sold in transactions
- Location validation –consumer, vehicle, utility, and specific connection
- Multiple dynamic pricing regimes
- Multiple single-utility and multiple-utility connections
- Micro-clearing between utilities and consumers, as well as third parties (parking garages, restaurants, malls, office buildings, hotels, etc.)
- Payment convenience – multiple methods: more than credit card “at the pump”.

Different users require access to electricity at different locations with appropriate privileges. Figure 18 shows different locations where a user needs access to charging infrastructure:

![Charging Locations Diagram]

Figure 18: Expected Charge Locations [31]

It can overload the system at a particular location during the peak hours. For instance, if an employee at a department wants to charge their vehicles from 8:00 am in the morning to 12:00 pm, the electric utility has to meet the requirement of the customers. This creates the issue of privacy of movement and communication. Most of the issues associated with the PHEVs are based on authentication and authorization requirements. One of the most useful techniques that can be used for specifying and identifying the security requirements of the PHEV communication in Smart Grid Role-based access by UML.
Authorization policies can be articulated and enforced with the help of UML and OCL. Role Based Access Control (RBAC) is an approach to restricting system access to authorized users. RBAC turned out to be one of the greatest access control mechanism for variety of systems. The advantage of the RBAC is that it introduced the concept of separation of duty (SoD) rules in access policies which can be implemented in a natural way [33]. RBAC authorization constraints are an important means for laying out the higher level access policies. [34]. OCL and UML can be used to specify different access policies. These two languages had been used for specifying several classes of role based access policies. “Moreover, owing to the fact that OCL has proved its applicability in several industrial applications, OCL is a good means for such a practically relevant process like the design of access policies. Therefore, for designing the access policies and specifying those formally for the PHEVs UML/OCL are one of the efficient methods” [19]. RBAC has received considerable popularity as an alternative to the traditional discretionary and mandatory access control in the industry. Being simple and easier, in RBAC the permissions are assigned to the users according to their roles/ functions. For example, in the PHEV infrastructure the customers and the employees need different kind of access to different information. The utilities are going to provide access to their information and resources to their customers depending on their relationship with them. The explicit representation of roles simplifies the security management and allows using security principles like separation of duty and least privileges. RBAC has the following main components:
Users, Roles, P, S (sets of users, roles, permissions, activated sessions)

- **UA** _Users × Roles_ (user assignment)
- **PA** _Roles × P_ (permission assignment)
- **RH** _Roles × Roles_ is a partial order also called the role hierarchy or role dominance relation. P is a set of ordered pairs of operations and objects. Applying it to the security and access control all the resources accessible in an IT system e.g. files, databases, tables, websites etc. are referred as object. On the other hand, an operation is an active process applicable to the available objects (e.g. read, write, delete, update, insert, append). The relation PA assigns each role a subset of P. PA determines the operations that each role may execute and the objects to which the operations are applicable for a given role [35].

These access policies can be expressed in UML as shown below in the figure. Figure 19 illustrates RBAC model expressed in the unified modeling language. Classes in UML are defined in terms of their attributes and relationships.

![Figure 19: Class Model for RBAC- entity Classes](image)

Figure 19: Class Model for RBAC- entity Classes [36]
Other Important advanced concept of the RBAC model is the Authorization Constraints. Similarly, OCL can also be used to specify the access policies and the security requirements formally to avoid the ambiguity and confusion due to different interpretation. The OCL syntax is similar to the set theory with some defined terms. It describes constraints on the object oriented models and also heavily used for object oriented analysis and design. Each OCL expression is written in the context of a specific class. The reserved word “self” is used to refer to a contextual instance. The context keyword is used for the type of the context instance. The label inv: declares the constraint to be an invariant. Below is an example of the OCL statement:

```oclick
context Role inv: self.user->size()<2
```

The statement below shows another example describing that a user can be assigned to a role r2 only if she is already a member of role r1 [37].

```oclick
class User inv:
  self.role_->includes('r2') implies self.role_->includes('r1')
```

Therefore, the combination of UML and OCL is really a strong security specification techniques especially for the systems where the role based access is the key. Accommodating this technique in the security requirement analysis of the PHEVs allow determining and specifying different access types, levels and the users of the system. Specifying security requirements formally reduces the ambiguity.

An Extension of UML for Security specifications (UMLsec): “UML sec is the extension of UML that allow to express the security relevant information within the
diagrams in a system specification. UML sec is defined in form of a UML profile using the standard UML extension mechanisms. In Particular, the associated constraints give criteria to evaluate the security aspects of a system design, by referring to a formal semantics of a simplified fragment of UML.” [38]

UML sec is a useful tool for business analysts, systems analysts and developers who do not possess background in security. It is a formal security specification standard that can be used to communicate the security aspects of a system among teams having different technical background. UML sec can efficiently express the general security requirements such as fair exchange, confidentiality and secure communication and information flow etc. for the distributed systems. UML includes activity diagrams, state charts, sequence diagram, and deployment diagrams. which can formally specify the security requirements of subsystems. The UMLsec extension has three main mechanisms: stereotypes, tagged values and constraints. Stereotypes are the new types of modeling elements. Tagged values are the name-value pair for the modeled elements. Detailed information about UML sec can be found in the paper “UMLsec: Extending UML for Secure Systems Development”, written by Jan Jurjens. UMLsec is also used to model the dynamic security behavior of a system [38].
Chapter 5

PHASE 3: IMPLEMENTATION AND ASSESSMENT

Implementation phase ensures the secure coding of the Smart Grid applications and components. The Assessment part of this phase includes: verification of the functionality and security through user acceptance testing, Quality assurance testing, fuzz testing, penetration testing, compliance testing and load testing to ensure that each component of the Smart Grid is secured and working as expected. The Smart Grid System or component is then integrated into the production environment once the verification is done [39].

The Carnegie Mellon Software Engineering Institute’s CERT program created a list of secure coding practices for different programming languages like java, C, C++ etc. According to Seacord from SEI CMU [40] says,

”C and C++ were selected because a large percentage of critical infrastructures are developed and maintained using these programming languages. C and C++ are popular and viable languages although they have characteristics that make them prone to security flaws. Without secure coding practices, software vulnerability reports are likely to continue on an upward trend”

These secure coding practices should be used in the Smart Grid environment as most of the legacy and new systems are developed or going to be developed using these programming languages [40].
5.1 Security Implementation Practices

Implementation phase should include the following practices:

(i) Secure coding practices: Most of the security vulnerabilities arise due to the insecure coding practices. From the coding perspective, vulnerabilities that occur weaken the security of an application or a system. Developers should proactively guard against these vulnerabilities. Some of these caused by [41]:

- Inherent vulnerabilities of the programming languages
- Security through obscurity
- Buffer overflow and format string vulnerabilities
- Race conditions locking problems
- Trusting user inputs and not validating parameters
- Poor exception handling
- Complexity
- Control granularity
- Excessive privilege

Smart Grid Developers should follow some security guidelines and rules when writing code. Some of these are:

a) Banned APIs: Banned APIs should not be used in old or new code. An attacker can control the incoming buffer, and the code uses data from the buffer to determine the maximum buffer length to copy.
b) Secure methods to access databases: An arbitrary query can be executed by a hacker using string concatenation while creating dynamic queries. This vulnerability can allow an unauthorized user to get interactive logon to the SQL server that may lead to the execution of malicious commands leading to the update, deletion, disclosure of the operating system or any user data. Therefore, to access the databases any combination of the parameterized queries, LINQ, or stored procedures can be used. Ad-hoc SQL queries should be avoided in order to avoid SQL injection attacks.

c) Input Validation: All inputs (user input, environment variables, file input etc.) to the system should be validated. It helps to protect application even if the component is reused or moved elsewhere. All online services must follow input data validation and encoding requirements in order to avoid vulnerabilities like cross site scripting.

d) Exception handling: Exceptions are a powerful way to handle run time errors but can also be abused by masking errors that can help the attackers to compromise systems. Exceptions should be handled at the appropriate level in the code and provide the sufficient information about the exception. Global exception handlers should be avoided.

e) Random numbers: The random numbers should be nondeterministic in nature.
f) Strong Cryptography: Industry Standard ciphers and algorithms should be used, avoiding the in-house or proprietary cryptographic solutions [41].

(ii) Aspect Oriented Programming (AOP) for coding the Security: Several security cross-cutting concerns are scattered into the application logic. In a real application logging code probably replicated in several class files. Some other security cross-cutting concerns those are replicated across the application code:

- Logging
- Access Control
- Error Handling
- Transaction Management
- Session Management
- Input/Output Validation

AOP is a new paradigm that promotes the separation of concerns. Implementing security with the help of AOP helps in encoding the security information into an independent piece of code. Security of most of the applications in the Smart Grid infrastructure can be enhanced by this programming paradigm as security can be implemented without changing the exiting code. It allows security policies to be separated from the actual code and helps to separate the roles of the developers and the security expert. Developers can write the code and the security expert can specify the security properties. AOP can be used to secure insecure function calls and has other important uses [41]:
• Automatically perform error checking on security-critical calls.

• Can be implemented in the stack guard technique of buffer overflow protection, inserting special code at function entry and exit.

• Automatically log data that may be relevant to security.

• Can be used to replace generic socket code with SSL socket code.

• Automatically insert code at startup that goes through a set of “lock down” procedures that most programmers would not add to their programs.

• Can be used to specify privileged sections of a program and automatically request and return privileges when appropriate.

Below is an example of security incorporated into AOP [42]:

```java
class MyFirstClass {
    public void aMethod (String bar) {
        Logger.doLoggingBefore();
        // business logic goes here
        Logger.doLoggingAfter();
    }
}
class MySecondClass {
    public void bFunction (Object arg) {
        Logger.doLoggingBefore();
        // business logic goes here
        Logger.doLoggingAfter();
    }
}
```

Here are two classes in which logging functionality is duplicated. With the AOP LogInterceptor aspect can be created and injected before and after any call to a class. Both the classes have no dependency on the LogInterceptor aspect [42]. Therefore, centralizing security policies with the AOP can reduce the amount of code. These policies can be used for different applications and components like SCADA, Demand Response, AMI etc. in the Smart Grid as similar security
features (logging, session management, and transaction management) are required across various applications.

(iii) Information sharing in the Error messages and Output: Error message are really important for the user information. Error messages are one of the important components in the user experience as it directs the user to the following path. These messages help the user to react differently under different conditions. Error messages are critical as far the security of an application is concerned. These should be specific and have minimum information disclosure but enough information so that the user can fix the problem. Errors Messages should not disclose any application level information so that a hacker can use it to compromise the Smart Grid system. All the exceptions should be handled internally and errors messages should be customized to present it to the user based on the type of application or component [13].

5.2 Security Assessment Practices

(i) Security Functionality Testing: Functional security testing ensures that software behaves as it should. If one of the security requirements of a system states that the length of user input should be certain characters long. Then functional testing is the part of the process that this requirement is implemented and works correctly. This type of testing emphasizes on what an application should not do instead of what it should do. Functional security testing is one of the important forms of testing for the Smart Grid in order to verify some critical features
spread throughout the System [43]. These features include encryption, user
identification, logging, confidentiality, authentication, etc. Testing these features
will highlight the effectiveness (or ineffectiveness) of such security features
which have been implemented. Some of the common Security functionality
testing techniques are shown in the table below:

Table 3: Common security vulnerability Techniques [43]

<table>
<thead>
<tr>
<th>Technique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad-hoc and explanatory testing</td>
<td>Testing based on the experience of the tester.</td>
</tr>
<tr>
<td>Requirements based testing</td>
<td>Test cases are built around the ambiguities in the requirements</td>
</tr>
<tr>
<td>Robustness and fault tolerance testing</td>
<td>Test cases are chosen outside the domain in order to test program robustness to unexpected and erroneous inputs. useful for probing fault tolerance and error handling.</td>
</tr>
<tr>
<td>State based testing</td>
<td>Force transitions that do not appear in higher level design artifacts, since vulnerabilities often appear when software enters an unexpected state.</td>
</tr>
<tr>
<td>Control flow testing</td>
<td>Aim at covering all statements, classes, or blocks in a program. Decision/condition coverage is one example.</td>
</tr>
<tr>
<td>Data flow testing</td>
<td>Used to test interfaces between subsystems</td>
</tr>
<tr>
<td>Code based testing</td>
<td>This technique is a superset of control flow testing and data flow testing.</td>
</tr>
<tr>
<td>Boundary value analysis</td>
<td>Choose test cases on or near the boundaries of the input domain of variables, with the rationale that many defects tend to concentrate near the extreme values of inputs. A classic example of boundary-value analysis in security testing is to create long input strings in order to probe potential buffer overflows.</td>
</tr>
</tbody>
</table>
Fault based testing  | Introduce faults during testing to probe program robustness and reliability.
---|---
Load and performance testing  | Load and stress testing exercise a system to the maximum design load and beyond it. Stressful conditions can expose vulnerabilities that are otherwise hard to see, and vulnerabilities can also be caused by the mechanisms that software uses to try to deal with extreme environments.
Run time verification  | Run-time verification seeks to validate that an application conforms to its security requirements and specifications by dynamically observing the application’s behavior in a test environment. Requirements such as “all authentication credentials must be encrypted while in transit” can thus be dynamically verified through observation.

Security Testing of Smart Grid should follow a list of checks for the applications for comprehensive tests. The list must include the following checks:

- Authorization checks
- Authentication checks
- Data Validation Checks
- Session management checks
- Buffer Overflows
- Injection Attacks
- Error Handling Checks
- Configuration Checks: Some of the configuration checks are:
  - a) Default accounts present
  - b) Missing updates and patches
  - c) Debugging and trace enabled
d) Unhardened operating system

e) Sample or demo apps

f) Directory traversal enabled

g) Mis-configured SSL

h) Backup or old files present in the system.

- Data Encryption Checks
- Data Storage Checks

(ii) Risk Driven testing: Risk-Driven testing deals with the identification of risks and applying appropriate mitigation strategies to the identified vulnerabilities. It is a process of document findings from the architectural risk analysis in much greater detail which includes details on how to actually carry out an attack. It reveals different hidden aspects of the application or product which are not easily identified by any other type of testing technique. The system is thoroughly analyzed in the earlier phases of the SDLC. Priorities are identified for the risks by calculating the probability of the occurrence of the risk, cost associate with it, impact of the risk etc. Test cases are then written based on the risks identified.

Unfortunately, risk driven testing is an art as the efficiency of risk driven testing is based on the experience and knowledge of the tester. A tester must know the system and its environment, interactions with the environment and should be able to understand the assumptions of the developers. Threat models are one of the helpful tools for this type of testing. But in security there is no appropriate
level of abstraction. Any manageable level of abstraction hides something and that hidden piece can be the exploitation of the system [43].

(iii) Regression Testing: Smart Grid system is huge and practically unmanageable as a whole, therefore, regression testing is an important phase of testing in Smart Grid as only some selected and appropriate test cases are used to explore the affects of changes to the system. This type of testing is helpful for identifying the error conditions which are not triggered in the normal usage of the system. Regression testing for the Smart grid applications helps to identify not only the bugs introduced in the maintenance phase but also helps in exploring other system vulnerabilities by triggering them. It may involve running the program under low memory conditions, insufficient permissions and privileges, interrupting a transaction before its completion or disabling the connectivity to basic network services like the DNS. Self healing is one the characteristics of this System, therefore, in regression testing if some unexpected errors or unhandled exceptions are triggered by these actions, it means that there are some unexpected conditions which cannot be handled by the system or application itself.

(iv) Code Review: Code review is technique of verifying the security controls to ensure that they work as they are intended to be and they are invoked at the right place and the right event. Code review should be done for both the in- house applications and COTS applications. Code review in combination with manual
penetration testing and automated code scanners is one of the effective techniques to find security flaws. Code Scanners are helpful in the hardware firmware code analysis of the Smart grid components like Smart meters etc. [44]. Code flow analysis and data flow analysis techniques should be used while reviewing the code e.g. If conditions, try catch blocks, input and output points, inputs and outputs should be properly analyzed. Table 4 describes the “Hotspots” table created by Microsoft which describes what a developer or a tester should look for while performing the Code review [45].

<table>
<thead>
<tr>
<th>Hotspot</th>
<th>What to look for</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL injection</td>
<td>A SQL injection attack occurs when untrusted input can modify the semantics of a SQL query in unexpected ways. As you review the code, make sure that the SQL queries are parameterized and that any input used in a SQL query is validated.</td>
</tr>
<tr>
<td>Cross Site Scripting</td>
<td>Cross-site scripting occurs when an attacker manages to inject script code into an application so that script code is echoed back and executed in the security context of the application. This can allow an attacker to steal user information, including forms data and cookies. This vulnerability can be present whenever a Web application echoes unfiltered user input back to Web content.</td>
</tr>
<tr>
<td>Data Acess</td>
<td>Look for improper storage of database connection strings and proper use of authentication to the database.</td>
</tr>
<tr>
<td>Input/data validation</td>
<td>Look for client-side validation that is not backed by server-side validation, poor validation techniques, and reliance on file names or other insecure mechanisms to make security decisions.</td>
</tr>
<tr>
<td>Authentication</td>
<td>Look for weak passwords, clear-text credentials, overly long sessions, and other common authentication problems.</td>
</tr>
<tr>
<td>Authorization</td>
<td>Look for failure to limit database access, inadequate separation of privileges, and other common authorization problems.</td>
</tr>
</tbody>
</table>
Sensitive data | Look for mismanagement of sensitive data by disclosing secrets in error messages, code, memory, files, or the network.
---|---
Unsafe code | Pay particularly close attention to any code compiled with the `/unsafe` switch. This code does not have all of the protection that normal managed code has. Look for potential buffer overflows, array out of bound errors, integer underflow and overflow, as well as data truncation errors.
---|---
Unmanaged code | In addition to the checks performed for unsafe code, also scan unmanaged code for the use of potentially dangerous APIs such as `strcpy` and `strcat`. Be sure to review any interop calls as well as the unmanaged code itself to make sure that bad assumptions are not made as execution control passes from managed to unmanaged code.
---|---
Hard coded secrets | Look for hard-coded secrets in code by looking for variable names such as "key", "password", "pwd", "secret", "hash", and "salt".
---|---
Poor error handling | Look for functions with missing error handlers or empty `catch` blocks.
---|---
Web.config | Examine your configuration management settings in the Web.config file to make sure that forms authentication tickets are protected adequately, that the correct algorithms are specified in the `machineKey` element, and so on.
---|---
Code access security | Search for the use of asserts, link demands, and `allowPartiallyTrustedCallersAttribute` (APTCA).
---|---
Code that uses cryptography | Check for failure to clear secrets as well as improper use of the cryptography APIs themselves.
---|---
Undocumented public interfaces | Most undocumented interfaces should not be in your application', and they are almost never given the same level of design and test scrutiny as the rest of the code.
---|---
Threading problems | Check for race conditions and deadlocks, especially in static methods and constructors.
---|---

All these hotspots mentioned in the table are areas of concern with the Smart grid applications. All these hotspots are based on the implementation mistakes that may result in common application vulnerabilities.
(v) Fuzz Testing: Fuzz testing is a technique by which the applications are forced to consume corrupted data and the results are observed. Properly coded applications will not crash as they reject the improper data. Fuzz testing is done by attacking an application’s data interfaces, file system, network, libraries, registry and Graphical User Interface (GUI). This type of testing provides very deterministic results by focusing on one area of concern. In Smart Grid most of the applications are consuming data from network servers of various types. Fuzzing these network streams can expose lot of vulnerabilities in the network, data stream, and input data. Special tools are available to effectively Fuzz test an application or a system by creating a large set of files. These tools also help in Fuzz testing a specific part of the network or system [46].

(vi) Compliance Testing: Compliance testing is another important form of testing which is basically done for a specified criterion. It is should be done for both hardware and software components of the Smart Grid. Compliance Testing ensures that a Smart Grid system meet some specific standards which have been developed for the security, efficiency, and interoperability. It should be performed by external organization preferably standards body to have greater guarantee of compliance. In this form of testing the architecture and design, databases, transactions, configurations, security policies, network components, development policies, and operating systems etc. are tested to ensure that all the aspects of the organization are compliant to the security standards. Once this
testing is completed, the system or product is awarded with a certification for being complying with the standards [47].

(vii) Verify Assembly Process: The final step of this phase is to verifying the decisions along the way by re-examining the assembly from bottom up. This step ensures that the risks are mitigated to an acceptable level. Asset defenses are checked to make sure that all the security measures are taken care of. The inputs to this process are the mapped business functions which are classified based on their likelihood and associate consequences, set of security requirements which are applied to each domain, set of security requirements for each element of the candidate architecture and a set of solution components. The output of this process is the assurance of the proper assembly and design [48].
Chapter 6

PHASE 4: OPERATIONS AND MAINTENANCE

The operations and maintenance phase in the SDLC includes all the activities related to keep the system working as intended. In this phase each component of the Smart Grid System should work according to what has been specified in the requirements. The maintenance of the hardware is necessary due to the wear and tear of the components and patch management or upgrades are required for software components to ensure proper and secure functioning. This is the critical phase for the Smart Grid Infrastructure as most of the security breaches happen in this phase. It is important to take a snapshot of the systems configuration before the system enters into the production environment for evaluation. Changes in the configuration and security goes hand in hand, if there are some changes to the configurations before production, then some modifications need to happen in the security tools, policies and procedures. All the components e.g. AMI, SCADA, demand response and Plug in hybrids required to be secure during their operations. Operational security contains all day to day activities performed by the information security groups such as: incident reporting, managing the firewall/IDS/security devices, patch management, and vulnerability management [39].

Operations Security includes the following aspects [49]:

- Physical and Environment Protection
- Production
- Input/Output Controls
• Emergency and Contingency Planning
• System and Data Backup
• Software Maintenance Control
• System Documentation
• System Change Management

Before putting the Smart Grid System or any of its components into production the security team must ensure that the servers have the latest security updates from the software manufacturer. Latest version of the antivirus software should be running on the servers and actively scanning all the files and shared directories. The security team should implement the secure auditing and the logging of the objects and files which contain sensitive information [13].

6.1 Security Practices for the Operations and Maintenance

As the Smart Grid system is most vulnerable to attacks in the operations phase, strict practices should be followed in order to have the secure operations. Some of the practices that can be followed by Security and Maintenance team of Smart Grid in order to make this implement security in this phase are:

• Use of Firewall: a firewall can be used to detect attacks like the SQL injections and Cross- Site Scripting. The firewall might not catch all the attacks but it could be useful in the cases when the organization does not have control on the code (Third Party Control) and code cannot be fixed easily as an emergency measure.
• Whitelist of the Commands: In order to prevent the operating system command injections, a whitelist of the commands should be created as run time policy enforcement measure. Any other command should not be allowed to run.

• Server Configuration: Configure the servers to use encrypted channels like Secure Socket Layer (SSL) and other secure protocols.

• Control capabilities of operating system for Access Control: Access Control capabilities of the operating system can be used to define the ACLs properly. “Default Deny” policy should be used when defining the ACLs.

• Permission Assignment for Critical resource: All the configuration files, executable files, and other libraries should default to readable and writable by the Software Administrator. Do not assume that the Administrator should assign permissions for each and every file manually.

• Taint Propagation Environment: Taint propagation is to follow untrusted data and identify points where they are misused. In order to secure the Smart Grid System against the attacks such as code generation and operating system command injections, make sure that the code runs in the environment that performs the automatic taint propagation and prevents any command that uses the tainted variables from running. This helps in forcing the application to remove taint from the commands and prevent their execution [50].

Department of Homeland Security (DHS) created a catalog of the controls systems security which are then adopted by the AMI – SEC taskforce for the AMI security [51].
These controls can be used for all other components of the Smart Grid System to ensure the proper operations and maintenance of the system as these controls contain different aspects to reduce and handle the cyber security issues in the critical infrastructures during their operations and maintenance phase. These controls help in ensuring secure communication, information integrity, Incident handling etc. Some of the important activities or controls that need to be included in the operations and maintenance phase of the Smart Grid are:

- **Training and Security Roles**: Training to the employees must be provided demonstrating their roles and responsibilities in their respective domains. Security roles and responsibilities for the Smart Grid users and employees should be defined, implemented and operated based on the criticality of information handled by the component. Defining these roles and responsibilities for users helps in alignment with the security requirements [51].

- **Information and Document Management Controls**: Information and documentation management is one of the important tasks in the operations and maintenance phase of the smart Grid. The operations data, analyses, business impact studies, risk tolerance profiles etc. are important to protect besides the requirements, design and implementation artifacts, which contains the sensitive company information. In addition to these security measures, philosophy, implementation strategies, business conditions change and studies need to be protected and appropriate versions need to be retained. Following are some
controls that can be used to protect the information and documentation about the Smart Grid system:

   a) Information and document management policies and procedures: An organization should frame, distribute and update periodically the policies and procedures about:

       1. Purpose, scope, roles, responsibilities, management commitment, and coordination among organizational entities, and compliance

       2. Implementation, maintenance and upgradation of the Smart Grid system components, services, etc.

   The information and document management policy and procedures should be consistent with the applicable policies, procedures or regulations and laws. Uncontrolled and unauthorized access to information such as billing, electrical consumption, SCADA operations, Demand response activities or other aspects of utility programs expose the utility and its customers to potential fraud and theft. Therefore, the document policies are necessary for protecting the organization and the customers from frauds and misuse.

   b) Information and documentation Retention: An organization should have information and documentation retention policies in order to protect the electronic and paper data related to Smart Grid. The
organization should manage the data based on the formal assigned roles and responsibilities. Information and documentation retention is important for the forensics studies in case of system failure, fraud, misuse or other security issues investigation [51].

c) Information and documentation handling, classification and exchange:
Proper policies and procedures should be updated and implemented for proper information handling classification and exchange. The information related to the Smart Grid should be classified according to the latest industry standards, sensitivity and its consequences. Classification of information into different protection categories e.g. “Confidential”, “secret”, or “top secret” is important to be implemented as the other blanket policies are not effective for handling information. Moreover, legally binding contracts and policies between the sender and the receiver of information are required avoid the deliberate and unintentional use of that information [51]. With this, both the parties know the protection level for their information or documentation copying, sharing, transmittal, and distribution.

- System Maintenance Controls

  (i) Legacy system upgrades, Monitoring and Evaluation: Some systems in the Smart Grid are easy for the attackers to get control as they were implemented decades ago when security was not a concern. These legacy
systems need to be replaced or updated by the organization. Moreover, other systems also need continuous monitoring and evaluation for the security upgrades to address the identified vulnerabilities.

(ii) Back Up and Recovery: All systems and applications in the Smart Grid are required to work under adverse conditions, natural disasters etc. Therefore, it is important in the maintenance phase that the organization should secure the backup and recovery system in order to restore the system to the normal conditions after natural disaster or any other calamity. The backups should be separated from the operational environment as they can be a reason for the security attack to the operational environment. The backup and recovery processes should be secure and done periodically.

(iii) Unplanned, Periodic and Remote system Maintenance: Any type of maintenance should be properly documented which includes information about the maintenance activity. Maintenance record should include necessity, impact, authorized users, equipment or component information, testing information and the ripple effects of the change if any. Untracked activities in the Smart Grid can lead to poor decision making and ignoring the malicious changes that can cost billions of dollars to organization. Any component or system under maintenance requires rigorous testing before putting back into operations. All COTS
components should have special maintenance policies and procedures to ensure that only desired maintenance of the system is completed without the addition of other malicious components. Remote maintenance should also be handled with proper care. All the sessions and connections should be terminated securely and properly after the maintenance. Remote maintenance in systems such as SCADA, Smart Meters or other AMI components requires highly secure environment and session termination technology to avoid breach into the network after session termination.

(iv) Tools: All Tools like diagnostic and test equipment should be properly secured. Proper encryption techniques should be used in order to protect the integrity of the data saved in those tools. All the tools should be checked for the malicious code before their usage. Security monitoring tools like Nmap, Nessus, Metasploit should be properly protected and special care should be taken to avoid the leakage of data from the tools. The data should be disposed properly after running those tools for monitoring and validating the security of the systems. Similarly, reports from other password cracking tools like John the ripper, Cain and Abel should be secured from unauthorized users and other employees [51].

- Incident Response Controls: The Organization should have a well established Software Security Incident Response Process. The software security incident response team should have a team of engineers and communication
professionals. During the incident response activity engineers can investigate and develop the solution and the communication team can provide required information and guidance to the users and the customer [52]. The organization should train the incident response team for specific roles and responsibilities to avoid confusion.

- Access Control: Due to the diversity of the users in the Smart Grid environment, the organization requires strict procedures and policies to categorize different users and their access levels. Aspect Oriented programming is one of the techniques that can be used for separation of concerns in the software development. Access control policies should base on the following criteria for different users:

(i) Connection type.

(ii) Update access.

(iii) Time of day.

(iv) Existence.

(v) Cascading authorization.

(vi) Global permissions.

(vii) Combination of privileges.

On the other hand the authentication of the users should based on the combination of different authentication strategies such as passwords, physical security device e.g.
smart card, PIN (personal identification Number etc. and Biometric identification e.g. retinal scan, palm identification, voice verification etc. [51]

- Audit and Accountability Controls [51]: Auditing and logging of the different components of the Smart Grid is necessary to validate that the security mechanisms present in the assessment and implementation phases are working as desired in the operations phase and after the system maintenance. Security audits are helpful in ensuring system’s records and activities are in compliance with the security policies and procedures. It also helps in finding security breaches or its attempts into the system. Moreover, Logging is also helpful in anomaly detection and forensics analysis after an incident occurs. Especially in the Smart Grid environment Auditing and logging is critical, as there are other threat vectors which may try every second to get through the holes in the infrastructure to create mass destruction. The Smart Grid components should at least record the following three events:

(i) Security events

(ii) Control events

(iii) System/ Device Configuration changes

Auditing and Logging of these should happen at different levels of abstraction which includes the packet level logging to analyze the traffic coming from the network. The audit activities should generate the records for:

- Startup and shutdown of the audit functions
- Successful and failed logins
• Failed authentications of signed or encrypted requests
• Change in access control or privilege
• Changes to security settings
• Creation, deletion, or modifications of users, password, tokens, and security keys
• Triggering of tamper sensors
• System Warnings and error messages

System logging must take into account numerous daily network/users/devices activities which contain the following information about them [51]:

• Date and time of the event
• The component of the AMI system (e.g., software or hardware component) where the event occurred
• Type of event
• User/subject/device identity
• The operational consequences in the case of an operational event

“These valid activities need to be distinguished from activities that appear suspicious. For this reason, an effective clipping mechanism should be in place. This mechanism, which includes setting clipping levels to define acceptable system activities, acts as a baseline for determining system violations. The goal of monitoring, auditing, and clipping levels is to discover problems before major damage occurs, and to be alerted
when a possible attack is underway. Theoretically, when the clipping mechanism detects that the baseline has been exceeded, an alarm is generated and the system records further information regarding the detected changes in activity” [49]. In other words, as soon as system detects that activities are occurring that fall outside of the predefined acceptable threshold, it notifies the security administrator via e-mail or pager, and generates a log of further activity. This log can then be used to investigate the suspicious activity. In addition, the Smart grid security team should analyze audit reporting, securing audit information and Tools.
Chapter 7

PHASE 5: DISPOSAL

Due to the enormous nature of the Smart Grid system, different types of media, web services and systems are used to transfer information from one place to another. This media and services often decommissioned or retired due to the system upgrades, maintenance or Configuration upgrades. All these different services, media and systems requires special handling, care and sanitization as they contain critical information about the internal infrastructure, customers, policies and procedures, employees etc. This information can be used by a malicious user or an authorized person to hack the infrastructure. In order to avoid the inadvertent release of this critical information or software, all types of media (hard copy and electronics media), systems and the services should follow strict disposal and sanitization policies. Disposal phase in the Smart Grid environment ensures the disposal or destruction of the information or documentation based on the organization policies and procedures framed during the operations phased based on the criticality of the information or documentation.

Major steps in this phase involve:

1. Preserve information
2. Sanitize Media
3. Document disposal activities and decisions pertaining to different media.

Proper disposal and sanitization of the media and software should occur throughout the SDLC of Smart Grid.
According to NIST [53],

“Media sanitization is one key element in assuring confidentiality. In order for organizations to have appropriate controls on the information they are responsible for safeguarding. They must properly safeguard used media. An often rich source of illicit information collection is either through dumpster diving for improperly disposed hard copy media, acquisition of improperly sanitized electronic media or through keyboard and laboratory reconstruction of media sanitized in a manner not commensurate with the confidentiality of its information. Media flows in and out of organizational control through recycle bins in paper form, out to vendors for equipment repairs, and hot swapped into other systems in response to emergencies. This potential vulnerability can be mitigated through proper understanding of where information is location, what that information is and how to protect it.”

NIST created guidelines for the disposal and sanitization of the media based on the type of information it contains. Table 5 below shows four different categories of media sanitization:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposal</td>
<td>Disposal is the act of discarding media with no other sanitization considerations. This is most often done by paper recycling containing non-confidential information but may also include other media.</td>
</tr>
<tr>
<td>Clearing</td>
<td>Clearing information is a level of media sanitization that would protect the confidentiality of information against a robust keyboard attack. Simple deletion of items would not suffice for clearing. Clearing must not allow information to be retrieved by data, disk, or file recovery utilities. It must be resistant to keystroke recovery attempts executed from standard input devices and from data scavenging tools. For example, overwriting is an acceptable method for clearing media. There are overwriting software or hardware products to overwrite storage space on the media with non-sensitive data. This process may include overwriting not only the logical storage location of a file(s) (e.g., file allocation table) but also may include all addressable locations. The security goal of the overwriting process is to replace written data with</td>
</tr>
</tbody>
</table>
random data. Overwriting cannot be used for media that are damaged or not writeable. The media type and size may also influence whether overwriting is a suitable sanitization method.

Studies have shown that most of today’s media can be effectively cleared by one overwrite.

<table>
<thead>
<tr>
<th>Purging</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purging information is a media sanitization process that protects the confidentiality of information against a laboratory attack. For some media, clearing media would not suffice for purging. However, for ATA disk drives manufactured after 2001 (over 15 GB) the terms clearing and purging have converged.</td>
</tr>
<tr>
<td>A laboratory attack would involve a threat with the resources and knowledge to use nonstandard systems to conduct data recovery attempts on media outside their normal operating environment. This type of attack involves using signal processing equipment and specially trained personnel.</td>
</tr>
<tr>
<td>Executing the firmware Secure Erase command (for ATA drives only) and degaussing are examples of acceptable methods for purging.</td>
</tr>
<tr>
<td>Degaussing of any hard drive assembly usually destroys the drive as the firmware that manages the device is also destroyed.</td>
</tr>
<tr>
<td>Degaussing is exposing the magnetic media to a strong magnetic field in order to disrupt the recorded magnetic domains. A degausser is a device that generates a magnetic field used to sanitize magnetic media.</td>
</tr>
<tr>
<td>Degaussers are rated based on the type (i.e., low energy or high energy) of magnetic media they can purge. Degaussers operate using either a strong permanent magnet or an electromagnetic coil. Degaussing can be an effective method for purging damaged media, for purging media with exceptionally large storage capacities, or for quickly purging diskettes.</td>
</tr>
<tr>
<td>Degaussing is not effective for purging nonmagnetic media, such as optical media [compact discs (CD), digital versatile discs (DVD), etc.].</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destroying</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destruction of media is the ultimate form of sanitization. After media are destroyed, they cannot be reused as originally intended. Physical destruction can be accomplished using a variety of methods, including disintegration, incineration, pulverizing, shredding, and melting.</td>
</tr>
<tr>
<td>If destruction is decided upon due to the high security categorization of the information or due to environmental factors, any residual medium should be able to withstand a laboratory attack.</td>
</tr>
<tr>
<td>•  <strong>Disintegration, Incineration, Pulverization, and Melting.</strong> These sanitization methods are designed to completely destroy the media. They are typically carried out at an outsourced metal</td>
</tr>
</tbody>
</table>
destruction or incineration facility with the specific capabilities to perform these activities effectively, securely, and safely.

- **Shredding.** Paper shredders can be used to destroy flexible media such as diskettes once the media are physically removed from their outer containers. The shred size of the refuse should be small enough that there is reasonable assurance in proportion to the data confidentiality level that the information cannot be reconstructed.

Optical mass storage media, including compact disks (CD, CD-RW, CD-R, CD-ROM), optical disks (DVD), and magneto-optic (MO) disks must be destroyed by pulverizing, crosscut shredding or burning. Destruction of media should be conducted only by trained and authorized personnel. Safety, hazmat, and special disposition needs should be identified and addressed prior to conducting any media destruction.

Other factors that can impact the sanitization and disposal decisions in Smart Grid environment are:

- Type and size of media
- Criticality of information which in turn influences the decisions of outsourcing the media for disposal or in house processing
- Level of training of personnel with the disposal tools
- Cost of Sanitization.

The information sanitization and disposition decision depends on the security and confidentiality of the information but not on the type of media. After the determination is made, proper disposal techniques best for that media is determined to achieve sanitization.

The flowchart in figure 20 shows the sanitization and disposition process based on the level of confidentiality. Different organizations can use this flowchart to determine
different disposition techniques required for the Smart Grid media based on the confidentiality of the information [53].

![Sanitization and Disposition Decision Flow](image)

**Figure 20: Sanitization and Disposition Decision Flow [53]**

Secure Decommissioning of the web services: Disposal phase for the web services should be secure. First, while creating the Web services proper versioning standard should be in order to avoid the confusion. The latest version of the web service should be used at all
required places. If more than one version of a web service is being used there should be a proper documentation to capture this information. Outdated versions should be decommissioned securely and removed from all the application interfaces. Multiple versions of a service can be one of the sources of the attack.
Chapter 8

CONCLUSION

8.1 Project Outcomes

The project provided an overview of the security practices which should be followed in the System Development Life Cycle (SDLC) of Smart Grid. These practices help to overcome different vulnerabilities which tend to be overlooked by the project team. Security practices discussed in this project could be helpful in improving the COTS application acquisition process as well as the testing of the software and hardware components of Smart Grid. Including training in the beginning of the SDLC improves understanding and in turn can be helpful in eliminating some of threats imposed by employees and customers. Some of the other techniques such as threat modeling and risk assessment discussed in the project may help to address different security issues and enforce security policies for potential threats. By following the above discussed security practices each phase of SDLC acts as a filter for the vulnerabilities. A percentage of vulnerabilities can be detected and removed in each phase which in turn helps to make the system more secure. Last but not the least this project supported the principle of building in security in the system and not treating it as an add-on at the end of system development.
8.2 Future Work

The project discussed the security practices which need to be followed in different phases of the SDLC of Smart Grid in order to identify vulnerabilities and their impact in the earlier phases of the development lifecycle. Studied security practices will help in stronger security and cost reduction of the Smart Grid system. Techniques discussed such as threat modeling, RAIM framework, fuzz testing, compliance and conformance testing will help in building security into Smart Grid system early in its System Development Life Cycle (SDLC). There are other related components not discussed in this report such as Software metrics and the Capability Maturity Model (CMM) for Smart Grid environment. There is a need to frame out different metrics for each phase of SDLC to measure the probability of an attack and risk related to vulnerability found in a particular phase. This will help in determining the cost of a vulnerability based on which phase of SDLC it is found and mitigated. Other research issues include analysis of different secure architecture styles for the Smart Grid.
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL</td>
<td>Access Control List</td>
</tr>
<tr>
<td>ADR or AutoDR</td>
<td>Automated Demand Response</td>
</tr>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>AMI</td>
<td>Advanced Metering Infrastructure</td>
</tr>
<tr>
<td>AOP</td>
<td>Aspect Oriented Programming</td>
</tr>
<tr>
<td>AP</td>
<td>Application Server</td>
</tr>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>AS</td>
<td>Authentication Server</td>
</tr>
<tr>
<td>CEC</td>
<td>California’s Energy Commission</td>
</tr>
<tr>
<td>CERT</td>
<td>Computer Emergency Response Team</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial- Off – the –Shelf</td>
</tr>
<tr>
<td>CPP</td>
<td>Critical-Peak Pricing</td>
</tr>
<tr>
<td>CPUC</td>
<td>California Public Utility Commission</td>
</tr>
<tr>
<td>DAC</td>
<td>Discretionary Access Control</td>
</tr>
<tr>
<td>DEA</td>
<td>Data Encryption Algorithm</td>
</tr>
<tr>
<td>DER</td>
<td>Distributed Energy Resources</td>
</tr>
<tr>
<td>DES</td>
<td>Data Encryption Standard</td>
</tr>
<tr>
<td>DHS</td>
<td>Department of Homeland Security</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name Server</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of Service Attack</td>
</tr>
<tr>
<td>DRAS</td>
<td>Demand Response Automation Server</td>
</tr>
<tr>
<td>DR</td>
<td>Demand Response</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>DREAD</td>
<td>Damage Potential, Reproducibility, Exploitability, Affected Users, Discoverability</td>
</tr>
<tr>
<td>DSA</td>
<td>Digital Signature Algorithm</td>
</tr>
<tr>
<td>EMCS</td>
<td>Energy Management and Control Systems</td>
</tr>
<tr>
<td>EMP</td>
<td>Electro-Magnetic Pulse</td>
</tr>
<tr>
<td>EMS</td>
<td>Energy Management System</td>
</tr>
<tr>
<td>HAN</td>
<td>Home Area Network</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hyper Text Transfer Protocol Secure</td>
</tr>
<tr>
<td>IFC</td>
<td>Integer Factorization Cryptography</td>
</tr>
<tr>
<td>ISO</td>
<td>Independent System Operator</td>
</tr>
<tr>
<td>LANs</td>
<td>Local Area Networks</td>
</tr>
<tr>
<td>KDC</td>
<td>Key Distribution Center</td>
</tr>
<tr>
<td>LBNL</td>
<td>Lawrence Berkeley National Laboratory</td>
</tr>
<tr>
<td>MAC</td>
<td>Mandatory Access Control</td>
</tr>
<tr>
<td>MAC</td>
<td>Message Authentication Code</td>
</tr>
<tr>
<td>MDM</td>
<td>Meter Data Management</td>
</tr>
<tr>
<td>MITM</td>
<td>Man-In-The-Middle</td>
</tr>
<tr>
<td>NAN</td>
<td>Neighborhood Area Network</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>OCL</td>
<td>Object Constraint Language</td>
</tr>
<tr>
<td>OpenADR</td>
<td>Open Automated Demand Response</td>
</tr>
<tr>
<td>PCT</td>
<td>Programmable Communicating Thermostat</td>
</tr>
<tr>
<td>PG&amp;E</td>
<td>Pacific Gas and Electric Company</td>
</tr>
<tr>
<td>PHEV</td>
<td>Plug In Electric Hybrid</td>
</tr>
<tr>
<td>Abbr</td>
<td>Full Form</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
</tr>
<tr>
<td>RAIM</td>
<td>Real Time Monitoring, Anomaly Detection, Impact Analysis, Integration techniques, Mitigation Strategies</td>
</tr>
<tr>
<td>RBAC</td>
<td>Role-Based Access Control</td>
</tr>
<tr>
<td>RTP</td>
<td>Real-Time Pricing</td>
</tr>
<tr>
<td>SAML</td>
<td>Security Assertion Markup Language</td>
</tr>
<tr>
<td>SCADA</td>
<td>Supervisory Control And Data Acquisition</td>
</tr>
<tr>
<td>SDLC</td>
<td>System Development Life Cycle</td>
</tr>
<tr>
<td>SEI</td>
<td>Software Engineering Institute</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Socket Layer</td>
</tr>
<tr>
<td>STRIDE</td>
<td>Spoofing, Tampering, Repudiation, Information Disclosure, Elevation of Privileges</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol/Internet Protocol</td>
</tr>
<tr>
<td>ToU</td>
<td>Time-of-Use Pricing</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>V2G</td>
<td>Vehicle-To-Grid</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Service Description Language</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
<tr>
<td>WS-Security</td>
<td>Web Services Security</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Mark-up Language</td>
</tr>
</tbody>
</table>
REFERENCES


[20] Stephen McLaughlin, Dmitry Podkuiko, and Patrick McDaniel,” Energy Theft in the Advanced Metering Infrastructure, Systems and Internet Infrastructure Security Laboratory (SIIS), Pennsylvania State University, University Park, PA.


[60] “Principle of least privilege”, Wikipedia. Available:

[61] “Secure by Default”, Wikipedia. Available: