Summer 2018

Analysis of Bulk Power System Resilience Using Vulnerability Graph

Md Ariful Haque
Old Dominion University

Recommended Citation

Haque, Md A.. "Analysis of Bulk Power System Resilience Using Vulnerability Graph" (2018). Master of Science (MS), Thesis, Computational Modeling & Simulation Engineering, Old Dominion University, DOI: 10.25777/fqw2-xv37
https://digitalcommons.odu.edu/msve_etds/14

ANALYSIS OF BULK POWER SYSTEM RESILIENCE

USING VULNERABILITY GRAPH

by

Md Ariful Haque
B.S. December 2006, Bangladesh University of Engineering and Technology
M.B.A. June 2016, IBA, University of Dhaka, Bangladesh

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

MODELING AND SIMULATION

OLD DOMINION UNIVERSITY
August 2018

Approved by:

Sachin Shetty (Director)

Yuzhong Shen (Member)

Hong Yang (Member)

ABSTRACT

ANALYSIS OF BULK POWER SYSTEM RESILIENCE USING VULNERABILITY GRAPH

Md Ariful Haque
Old Dominion University, 2018
Director: Sachin Shetty

Critical infrastructure such as a Bulk Power System (BPS) should have some quantifiable measure of resiliency and definite rule-sets to achieve a certain resilience value. Industrial Control System (ICS) and Supervisory Control and Data Acquisition (SCADA) networks are integral parts of BPS. BPS or ICS are themselves not vulnerable because of their proprietary technology, but when the control network and the corporate network need to have communications for performance measurements and reporting, the ICS or BPS become vulnerable to cyber-attacks. Thus, a systematic way of quantifying resiliency and identifying crucial nodes in the network is critical for addressing the cyber resiliency measurement process. This can help security analysts and power system operators in the decision-making process. This thesis focuses on the resilience analysis of BPS and proposes a ranking algorithm to identify critical nodes in the network. Although there are some ranking algorithms already in place, but they lack comprehensive inclusion of the factors that are critical in the cyber domain. This thesis has analyzed a range of factors which are critical from the point of view of cyber-attacks and come up with a MADM (Multi-Attribute Decision Making) based ranking method. The node ranking process will not only help improve the resilience but also facilitate hardening the network from vulnerabilities and threats.

The proposed method is called MVNRank which stands for Multiple Vulnerability Node Rank. MVNRank algorithm takes into account the asset value of the hosts, the exploitability and impact scores of vulnerabilities as quantified by CVSS (Common Vulnerability Scoring System). It also considers the total number of vulnerabilities and severity level of each vulnerability, degree centrality of the nodes in vulnerability graph and the attacker's distance from the target node. We are using a multi-layered directed acyclic graph (DAG) model and ranking the critical nodes in the corporate and control network which falls in the paths to the target ICS. We don't rank the ICS nodes but use them to calculate the potential power loss capability of the control center nodes using the assumed ICS connectivity to BPS. Unlike most of the works, we have considered multiple vulnerabilities for each node in the network while generating the rank by using a weighted average method. The resilience computation is highly time consuming as it considers all the possible attack paths from the source to the target node which increases in a multiplicative manner based on the number of nodes and vulnerabilities. Thus, one of the goals of this thesis is to reduce the simulation time to compute resilience which is achieved as illustrated in the simulation results.

This thesis is dedicated to my father who used to say, “Knowledge is the best ornament of a human being.”

ACKNOWLEDGMENTS

The thesis work and materials presented here are based on work supported by the Department of Energy under Award Number DE-OE0000780 and the Office of the Assistant Secretary of Defense for Research and Engineering (OASD (R&E)) agreement FA8750-15-2-0120. I would like to thank CYBER RESILIENT ENERGY DELIVERY CONSORTIUM (CREDC) for their contribution towards the thesis.

There are many people who have contributed to the successful completion of this thesis. I would like to express my heartiest gratitude to my thesis supervisor Dr. Sachin Shetty for his supervision and untiring efforts throughout the work of this thesis. I would also like to thank the committee members Dr. Yuzhong Shen and Dr. Hong Yang for their helpful guidance and advice regarding the completion of this thesis. I would also like to extend my thanks to the department chair Dr. Rick Mckenzie.

My heartiest thanks to my spouse Mst. Mohosina Khatun for her continuous support and motivation in my life. Finally, thanks to the Almighty Allah for giving me strength to carry out this thesis and for His infinite blessings in my life.

NOMENCLATURE

BPS	Bulk Power System
ICS	Industrial Control System
SCADA	Supervisory Control and Data Acquisition
NVD	National Vulnerability Database
CVSS	Common Scoring System Vulnerability
A_V	Access Vector
A_C	Access Complexity
A_A	Access Authentication
I_I	Impact on System Integrity
I_C	Impact on System Confidentiality
I_A	Impact on System Availability
$G(N,e)$	Vulnerability Graph with *N* nodes and *e* edges
R	Resilience
K_e	Critical Functionality
$c(e)$	Path Cost Value
A_i	Asset Value of Node *i*
D_i	Shortest Path Distance of Node *i* from target *t*
ES	Exploitability Score
IS	Impact Score
R_i	Ranking Value of Node *i*
RC_i	Relative Criticality of Node *i*
$C_d(i)$	Degree Centrality of Node *i*
Vuln	Vulnerability

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

The ever-increasing instances of cyber-attacks on critical infrastructures have motivated researchers to analyze and develop methods to safeguard infrastructures. The North American Electric Reliability Corporation (NERC) has defined Bulk Power System (BPS) [4] as “facilities and control systems necessary for operating an interconnected electric energy supply and transmission network”. NERC recommends using the term “BPS” when referring to the interconnected network or power grid. ICS and SCADA are integral parts of BPS. ICS is also critical components facilitating operations in different important industries such as electricity, water, oil and gas, transportation and manufacturing. Adverse events in ICS may be caused today by not only natural disasters but also by smart cyber-attackers. Any adverse event on the ICS of BPS may disrupt the critical services and may result in safety risks to people and the environment[1].

The North American Reliability Corporation (NERC) is working along with the Department of Defense (DoD) in standardizing and developing processes to prevent and minimize the threats and impacts of cyber-attacks on the BPS. Typical cyber-security actions primarily focus on intrusion detection techniques to detect threats and take necessary measures based on detected threats such as introducing and implementing patches. Though intrusion detection is an important security task, there is always a need for methods and techniques to make the BPS resilient and lessen cyber vulnerabilities. In the past, the most common threats faced by the ICS or BPS were in the physical domains with adverse events such as physical attacks, failures and natural disasters. As a consequence, a lot of efforts have been made to

[1] IEEE Transactions and Journals style is used in this thesis for formatting figures, tables, and references.

analyze the resilience and survivability of ICS in the presence of such threats, but some recent events show that those systems are at more risk due to cyber-attacks which normally takes time to identify and respond to. There have been numerous approaches developed to quantify the resilience of BPS. In one recent work, Sachin et al. [5] have quantified the resilience metrics for BPS by taking into consideration path costs and critical functionality constructed by the vulnerabilities presented in different hosts using a graph model. Computation of resilience metrics is highly time-consuming. This thesis uses the resilience metric defined in the above article and proposes a ranking algorithm that can help reduce the simulation time to calculate the resilience.

This thesis has analyzed different resilience frameworks of BPS in scholarly articles. The proposed ranking approach is to help network analysts make the BPS network security measures harder to exploit. The proposed method has been supported by necessary simulation results and a comparison with a previously published conference paper.

1.1 Motivation

Some recent reports on cyber-attacks, claiming a number of incidents that have occurred in the national infrastructure, confirm that the US energy sector, especially the power grid and SCADA systems are constantly under cyberattack [6]. According to the report, during the fiscal year 2014, there were 79 hacking incidents where energy companies were the targets, and between April 2013 and 2014, threat actors hit 37% of energy companies, making the energy sector one of the most critical industries under cybersecurity perspective. Some of those targeted attacks on the energy sectors in recent times are discussed in detail in chapter 2. There is no way we can ignore the need to have more research to safeguard the US energy sector and other critical sectors.

The purpose of this thesis work is to analyze different ways to make the electric power system network, specifically the BPS/ICS networks to harder to exploit. One BPS security measure is to analyze the resilience frameworks currently in place. The major goal of this thesis is to quantify the resilience metrics of BPS and develop an approach to suggest which network elements can be influential in the resilience improvement process considering the network structure that already exists. Knowing which systems or system components is important for improving resilience of the network against cyber-attacks and software vulnerabilities can give the network analyst a proper direction. This can facilitate the power system operators in the network hardening and optimization process. This thesis sheds light on the network elements which can be considered critical for improving the resilience of target systems by ranking them.

1.2 Problem Description

For this thesis work, a multi-layered directed acyclic graph (DAG) model derived from NIST SP 800-82 has been used. For real bulk power systems, the corresponding DAG may have several tens of thousands of nodes at the different layers such as the corporate layer and the control system layer. In addition, one single node may have multiple active vulnerabilities. As a consequence, when the vulnerability graph is being generated by using some tools or programs, the size and complexity exceed human capability to visualize and analyze it. Therefore, it is important to identify relevant nodes of the graph which are critical from a resilience standpoint to facilitate the network analyst. One of the ways to identify and rank the critical nodes in the vulnerability graph is to develop a ranking algorithm. Given a ranked attack graph, the regulator or the system administrator can focus on relevant nodes to figure out where to start deploying security measures. Ranking algorithms for attack graphs have been proposed before in scholarly articles reviewed in detail in chapters 3 and 4, but in those works, which considered graph theory, nodes represents system states and edges represents conditions for transitions between states. Some papers used the Bayesian probability-based approach. Most of the works that are currently in place do not consider multiple vulnerabilities and the combined effects on the nodes to be ranked and the attacker behavior, thus resulting in a partial scenario from a cybersecurity perspective.

1.3 Method and Procedure

To develop a node ranking algorithm that can improve bulk power system resilience in a systematic manner, this study reviews the literature on related subjects such as resilience, risk, SCADA security, cyber-physical system security of ICS and SCADA network, graph theory, attack-graph in analyzing cyber-security and vulnerability assessment frameworks, etc. There are a lot of factors that need to be considered when trying to identify and rank the critical nodes in a vulnerability graph model. Some targets may be lucrative to attackers because of the potential damage impact on the network by exploiting the target. To reach a target, in a vulnerability graph, the attacker may exploit several intermediate nodes in the corporate and control network layers before reaching to the target physical system layer. The number of intermediate nodes may not be the same for different target nodes. That is why the ranking of the nodes should not be the same each time; rather, the ranking should be changed dynamically based on the target node.

This thesis is proposing an MADM approach-based algorithm which considers several crucial factors respective to the vulnerability graph model. The algorithm considers asset value of the intermediate nodes, the exploitability and impact scores of vulnerabilities as quantified by CVSS (Common Vulnerability Scoring System). It also takes into account the total number of vulnerabilities each node has and the severity level of each vulnerability. Other factors considered important are degree centrality and the attacker's position in the shortest paths to the target node. Unlike most of the works, we have considered multiple vulnerabilities for each node in the network while generating the rank because only considering the most exploitable vulnerability may not give a complete analysis of resilience as because the most exploitable vulnerability may have the least possible impact on the network and vice versa. For verification purposes, we have used a sample network and construct a database that extracts most of the vulnerability information from the National Vulnerability Database (NVD). For validation, as we are lacking real system data, we have used statistical analysis and comparison of our rank with a previously published paper.

1.4 Contribution of The Thesis

This thesis contributes in many ways to the analysis of the resilience of the bulk power system. Some of the major contributions of this thesis are highlighted below.

- Most of the existing works on ranking either rank the vulnerability separately to patch or rank the nodes by considering the most exploitable vulnerability. Some of the attack graph analysis considers some pre and post-conditions for an attacker to be successful to exploit a target. In a real system network, although it is necessary to consider the pre and post-conditions, the attacker is able to penetrate the network bypassing those pre and post-conditions. This work considers multiple vulnerabilities that each host may have, which means the work not only considers the most exploitable vulnerability but also it considers the comprehensive effects of all the exploitable vulnerabilities and their impacts. As we know, the most exploitable vulnerability may have the least possible impact, so an attacker will always want to consider the benefits that he may achieve by exploiting a vulnerability. Thus, the ranking formulation that we are proposing is a comprehensive approach.

- Most of the works formulate ranking of network nodes based on the exploit metrics only. This work considers both exploit and impact metrics. The work expands its analysis by considering some other critical factors such as asset value of the hosts, degree centrality of the node in the graph and attacker relative position in the network. Thus, the ranking is robust.
- The resilience equation that is being used in the thesis work is highly time-consuming if the network size is large. The ranking provides a systematic approach to consider the most critical nodes sequentially and thus help reduce the computation time of resilience. Using the ranking reduces the resilience computation time by nearly 50% which can be useful for large networks' resilience computation.
- The thesis can help power system operators to evaluate their corporate and control network security measures and thus assist them in hardening network security.

CHAPTER 2

BACKGROUND OF THE THESIS

Critical Infrastructures are the lucrative target of cyber-attacks by cyber-criminals. Although the ICS within the critical infrastructures are not directly vulnerable to security flaws themselves due to their proprietary technologies and isolated commands and control methods of operation, the extensive use of Information and Communication Technologies (ICT) in ICS and integration of ICT in ICS make them vulnerable to cyber-attacks. The same is true for the electric power grid SCADA network. There have been a lot of research activities to develop resilience framework specific to BPS (Bulk Power System), some of which will be discussed in this chapter. In this chapter, some of the recent cyber-attacks on the energy sectors have been presented in detail to focus our attention on protecting the energy sector's control network from smart cyber-criminals. This chapter also presents the existing frameworks and a comparison analysis between them and definitions of resilience and its quantification process. The chapter also addresses NIST SP 800-82 on which the DAG model has been developed which has been used in node ranking algorithm discussed in chapter 4.

2.1 Interdependencies of Critical Infrastructures on SCADA or ICS

Critical infrastructures often have interdependencies between various industrial sectors as well as interconnections between other business entities. Critical infrastructures are highly interconnected and mutually dependent in complex ways where information and communications need to be done between different hosts using the IT network systems. Thus, an incident in one infrastructure can affect other infrastructures either directly or indirectly through cascading and escalating failures. The electric power generation and distribution companies use distributed

SCADA control systems for its operation and monitoring. For example, some SCADA systems are used to monitor and control electricity distribution by collecting data from remote terminal units and issuing commands to remote field location devices from a centralized control center.

As electric energy is necessary for running almost all other sectors, it is often considered to be one of the most prevalent sources of disruptions of interdependent critical infrastructures. For example, if an attacker successfully penetrates any network element in the Corporate Network which is being used to communicate with enterprise networks and outside stakeholders, then he can gain access to other hosts in the same network and propagate to the next level. Taking advantage of the system components' vulnerabilities, an attacker may reach and exploit some control system devices and can shut down a large generation or distribution unit. This would lead to loss of power at a transmission substation. The loss of power of this substation may cause a major imbalance in the power grid, triggering a cascading failure. Sometimes, because of loss of monitoring, the attacker may disrupt several power stations which could result in large blackouts. As almost all the industries are dependent on electricity, that could potentially affect oil and natural gas production, refinery operations, water treatment systems, wastewater collection systems, and pipeline transport systems and many other businesses. Thus, it can lead to a major national crisis.

2.2 Cyber Incidents in Critical Infrastructures

The energy sector, being a critical infrastructure always has a high attack risk from cyber-criminals highly skilled in hacking technologies and procedures. Some of the global cyberattack incidents that have been taken place in the energy sector, nuclear power plants, etc. are explained in the following subsections.

2.2.1 Cyber-Attacks on the Ukrainian Power Grid

There have been numerous events where a cyber-attack has been reported as the cause of the failure of the critical infrastructure system. One such attack was on the electric power grid of Ukraine in December 2015. [7-9]. According to the reports, a regional electricity distribution company of Ukraine had reported service outages to customers on December 23, 2015. The outages were due to a third party's illicit passage into the company's PC and SCADA systems networks. Due to the cyber-attacks on the SCADA systems, seven 110 kV and twenty-three 35 kV substations were out of service for three to six hours. Later statements indicated that the cyber-attack impacted additional portions of the distribution grid and operators needed to switch to manual mode to restore the grid. The outages were originally thought to have affected approximately 80,000 customers. Later it was revealed that three different energy distribution companies were attacked, resulting in other outages that caused approximately 225,000 domestic and industrial customers to live without power across various zones within the country. A detailed analysis of the electric grid failure due to cyber-attack was done in [10, 11]. The analysts have identified that the intruders used spear phishing emails to plant the malware trojan named "BlackEnergy3". Intruders exploited Microsoft Office vulnerabilities and got control of those document files that contained the malware which gave them a foothold into the Information Technology (IT) networks of the electricity companies. The intruder stole important credentials from the business networks and utilized virtual private networks (VPNs) to enter the ICS network and finally exploited and utilized the existing remote access tools within the SCADA environment for issuing commands directly from a remote station which is like an operator HMI (Human Machine Interface). A flow diagram is given below.

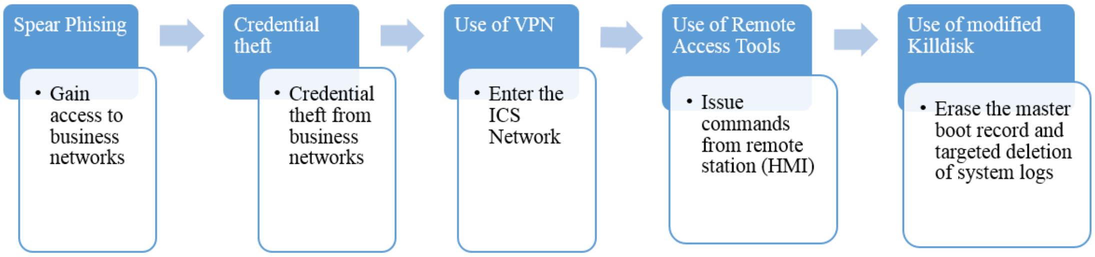

Spear Phising
Gain access to business networks
Credential theft
Credential theft from business networks
Use of VPN
Enter the ICS Network
Use of Remote Access Tools
Issue commands from remote station (HMI)
Use of modified Killdisk
Erase the master boot record and targeted deletion of system logs

Fig. 1. Flow Diagram of process followed by cyber attackers in Ukrainian Power grid compromise.

2.2.2 Stuxnet worm attack on Iranian Nuclear Plant

The Stuxnet Worm first emerged in summer 2010. Stuxnet was a 500-kilobyte computer worm that could infiltrate numerous computer systems. According to several reports and analysis [12-14], more than fifteen Iranian facilities were attacked and infiltrated by the Stuxnet worm. It is believed that this attack was initiated through a USB drive of one of the workers. One of the affected industrial facilities was the Natanz nuclear facility. The details of the events have been analyzed by researchers and experts. One such report [15] has mentioned that Stuxnet specifically made programmable logic controllers (PLCs) as the target, which allows the automation of electromechanical processes such as those used to control machinery on factory assembly lines and centrifuges for separating nuclear material. Stuxnet exploited four zero-day vulnerabilities [16] and Stuxnet functions by targeting machines using the Microsoft Windows operating system and networks, then seeking out Siemens Step7 software. Stuxnet reportedly compromised Iranian PLCs, collecting information on industrial systems and causing the fast-rotating centrifuges to rip themselves apart.

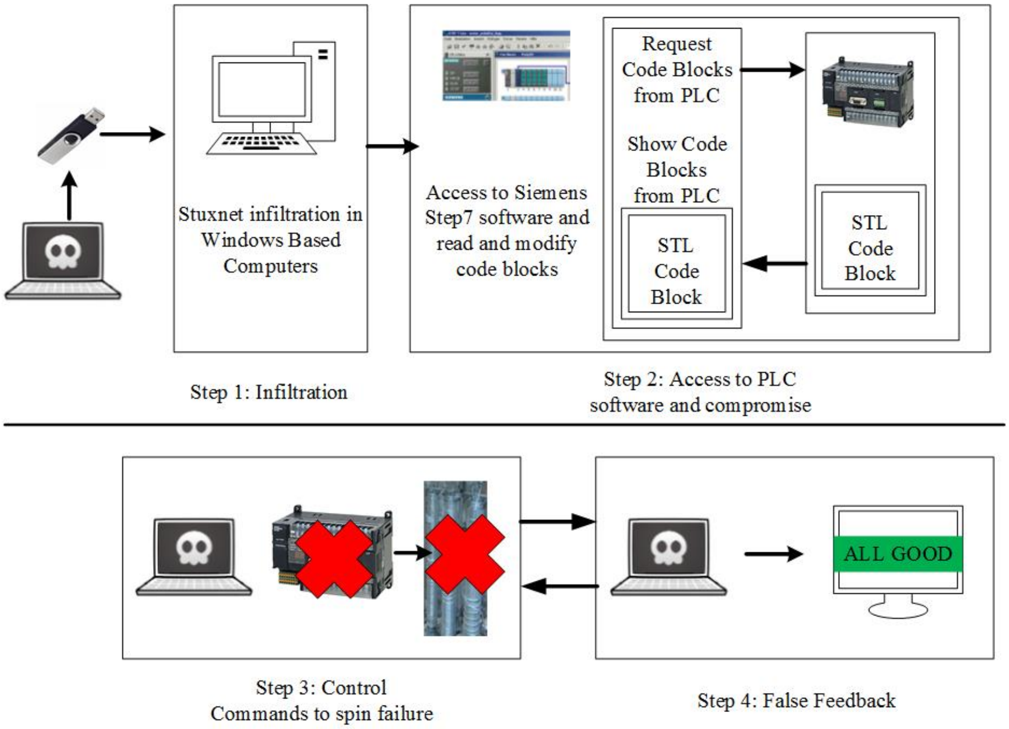

Stuxnet infiltration in Windows Based Computers
Step 1: Infiltration
Access to Siemens Step7 software and read and modify code blocks
Request Code Blocks from PLC
Show Code Blocks from PLC
STL Code Block
STL Code Block
Step 2: Access to PLC software and compromise
ALL GOOD
Step 3: Control Commands to spin failure
Step 4: False Feedback

Fig. 2. Flow Diagram of Stuxnet attack on Iranian Nuclear Centrifuges.

Initially, the cause of the failures of centrifuges in the nuclear plant was not discovered, but in June 2010 Iranian authorities contracted computer security specialists in Belarus to examine their computer systems [17]. The security firm finally discovered multiple malicious files on the Iranian computer systems and identified that these malicious files were the Stuxnet worm. Although Iran has not published specific details regarding the impacts of the attack, it was estimated that the Stuxnet worm destroyed 984 uranium-enriching centrifuges which are nearly one-fifth of Iran's nuclear centrifuges. A flow diagram of the attack steps is depicted in Fig. 2. As a protective measure after the Stuxnet attack, Siemens released a detection, quarantine and removal tool for Stuxnet. Siemens recommends contacting customer support if an infection is detected and advises installing Microsoft updates for security vulnerabilities and prohibiting the use of third-party USB flash drives [18]. Siemens also advises immediately upgrading password access codes [19]. The worm's ability to reprogram external PLCs may complicate the removal procedure which is also alarming for the electric power grid SCADA system. In another report [20], an SAP researcher discussed the impact of Stuxnet on electric grid monitoring and control systems such as SCADA/DCS. The analysis mainly focused on the assessment of existing security considerations and posed some thoughts on the next generation SCADA/DCS systems from a security perspective.

2.2.3 Dragonfly

"DragonFly", the scandalous hacking group that has been in operation since at least 2011 is interested in targeting the United States and European companies in the energy sector [21]. Security firm Symantec is warning that a series of recent cyber-attacks not only compromised US and European energy sector companies but also pointed out that the intruders are increasingly gaining hands-on access and knowledge to the power grid operations – which Symantec professionals thought to be enough to control the ICS and may lead to potential outages in North America and Europe [22].

The Dragonfly group, also popularly known as "Energetic Bear", are suspected to have been running their operation since at least 2011. As reported, Dragonfly initially focused and targeted defense and aviation companies in the US and Canada before shifting its focus mainly to US and European energy firms in early 2013 [23]. During the past several years, including 2014, Dragonfly malware infiltrated hundreds of business computers in a regular successful endeavor to gather information on the industrial control systems across the United States and Europe [24]. During analysis by security firm Symantec it was found that the attack was performed in an organized way over an extended period and used infection methods that were difficult to detect. The malware gathered data and information that are imperative to the operation of the impacted systems across the energy and pharmaceutical sectors. It has been identified that the operational mode of "Dragonfly" involves targeting the victim and its purpose is to steal information. The malware is often treated as an Advanced Persistent Threat (APT), meaning that the malware is designed to become occupant on the victim's system to collect information over an extended period without being identified. APT execution of automated industrial processes requires expert knowledge of both information technology and the use of particular industrial systems [25].

According to the report published by SANS Institute [24] in February of 2013, Symantec identified a spear-phishing email effort that appeared to target specific organizations to seek organizations' confidential information. The email assault continued until June 2013. Around that time, the attack started utilizing the watering-hole technique which involved redirection of website addresses to those maintained and controlled by the Dragonfly group. Software with malicious contents was secretly kept on those sites, and it was found that victims themselves were transferring those malicious contents to various company networks unknowingly. The cyber-attackers also began to utilize websites hosted by ICS product vendors to insert the malware directly into software that would be downloaded and used by the professionals working with ICS systems in those companies.

There are other cyber-attack events worth mentioning that turned professionals' attention towards the security measures of their networks but those are not being presented here because of the scope of this thesis.

2.3 Literature Review of Resilience Definitions and Resilience Frameworks

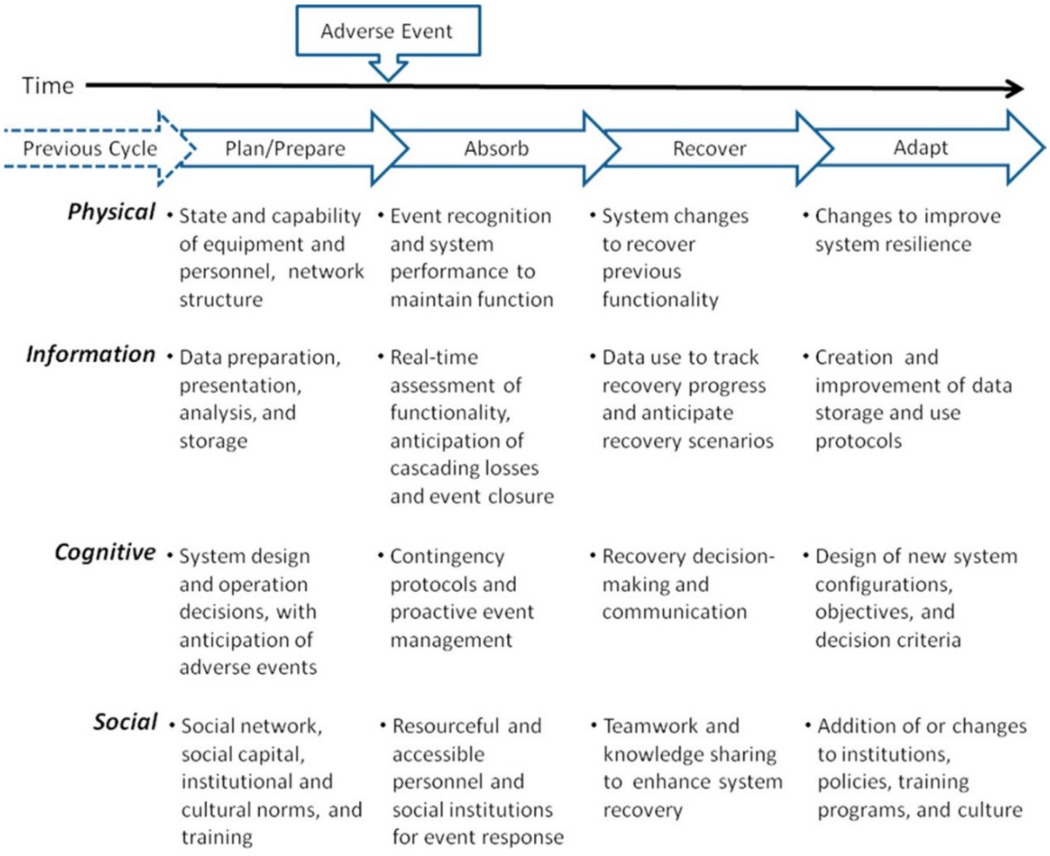
Industrial Control Systems (ICS) are critical components facilitating operations in vital industries such as water, electricity, oil and gas, transportation and manufacturing [26]. For that reason, any adverse events in ICS may both cause critical services to fail and may hamper or create risk to people and the community. That's why there has been research and analysis on the formal definitions of resilience and much effort has been given by scholars all around the world to formulate a framework for resilience. In this literature review section, some of those efforts are summarized with the critical analysis on those frameworks applicability towards safeguarding electrical energy systems or ICS systems in BPS.

2.3.1 NAS Definition of Resilience

The National Academy of Science (NAS) in their report "Disaster Resilience: A National Imperative", has defined the term resilience as "The ability to prepare and plan for, absorb, recover from, or more successfully adapt to actual or potential adverse events" [27]. The definition was created to keep in mind the perspective of natural disasters, the definition seems to have good application towards the energy systems strength or ICS network strength in protecting its critical resources from diverse attacks and failures. Although it is a broad definition, but it doesn't focus on quantitative aspects of the resilience; rather, it mainly focuses on the qualitative aspects.

2.3.2 Measurable Resilience for Actionable Policy Framework

Igor Linkov et al. [28] have identified two challenges for quantification of resilience in complex systems. The first challenge is to relate quantitative risk assessment to the resilience of the system. In most of the systems, quantitative risk assessment is treated as the dominant paradigm for system design and management, but in many cases, such as if sudden disastrous conditions happen, the traditional risk analytical approach may become ineffective. Therefore, according to the authors, it needs to treat resilience measurement as a distinguishable novel analytical approach separate from the traditional risk assessment, but resilience and risk may be complementary to each other. The second challenge is related to the fragmentation of resilience into separate disciplines, including engineering infrastructure, environmental management, and cybersecurity.



	Plan/Prepare	Absorb	Recover	Adapt
Physical	• State and capability of equipment and personnel, network structure	• Event recognition and system performance to maintain function	• System changes to recover previous functionality	• Changes to improve system resilience
Information	• Data preparation, presentation, analysis, and storage	• Real-time assessment of functionality, anticipation of cascading losses and event closure	• Data use to track recovery progress and anticipate recovery scenarios	• Creation and improvement of data storage and use protocols
Cognitive	• System design and operation decisions, with anticipation of adverse events	• Contingency protocols and proactive event management	• Recovery decision-making and communication	• Design of new system configurations, objectives, and decision criteria
Social	• Social network, social capital, institutional and cultural norms, and training	• Resourceful and accessible personnel and social institutions for event response	• Teamwork and knowledge sharing to enhance system recovery	• Addition of or changes to institutions, policies, training programs, and culture

Fig. 3. Resilience Metrics proposed by Igor Linkov et al. [28].

In the proposed resilience metrics, as shown in Fig. 3, mapping of system domains across an event management cycle of resilience function has been presented where the cells of the metrics provide guidelines that need to be researched and combined into the overall system resilience measurement process.

The authors have considered four separate domains for information distribution across the networks: physical, information, cognitive and social. In their proposed resilience metric, each cell is representing what is important for developing quantitative and qualitative measures according to the NAS four stage definition of resilience as provided in section 2.3.1. Based on the framework presented earlier, a detailed resilience metric framework has been proposed in [29] by the authors with the guidelines explained in details. Also, Roege et al. in their article "Metrics for energy resilience" [30] explained the resilience metric framework in terms of energy systems. Their study synthesizes previously proposed metrics and other emergent resilience literature to provide a multi-dimensional model intended for use by the practitioners and others associated with energy delivery systems.

2.3.3 R4 Resilience Framework

According to Kathleen and Bruneau [31], the term "Resilience" is emphasized to improve the capacity of physical and human systems to respond to and recover from adverse events. A more Resilient system should have fewer probabilities of failure. It should also reduce the consequences of failure and take less time for recovery.

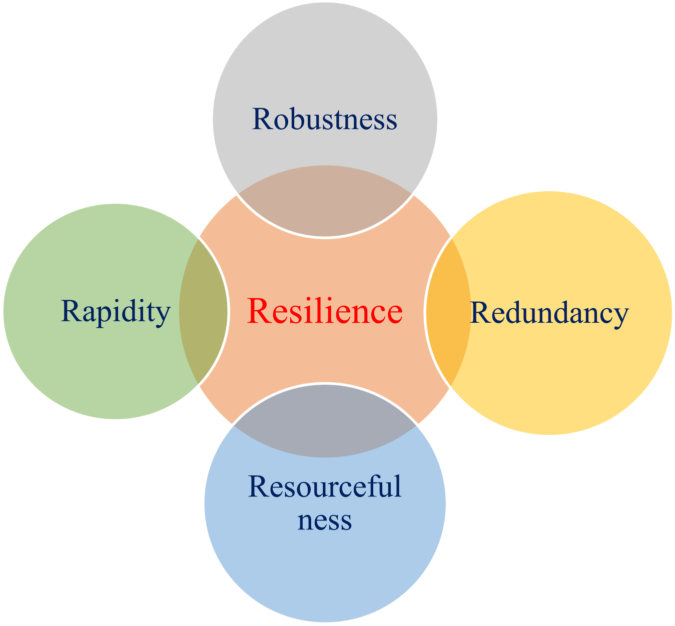


Fig. 4. R4 Resilience Framework [31].

In other words, resilience can be measured by the functionality of a complex system after an adverse event and by the time it takes for the system to return to normal operational levels of performance as the system was performing before the adverse event occurs. The Multidisciplinary Center for Earthquake Engineering Research (MCEER) investigators proposed an R4 framework for resilience, where the researchers have identified four main characteristics of a system to be resilient from adverse events which are depicted in Fig. 4. The interpretation of the four components of resilience as defined in [31] is given below:

- **Robustness:** "The ability of the system, system elements and other units of analysis to withstand disaster forces without significant degradation or loss of performance".
- **Redundancy:** "The extent to which systems, system elements or other units are substitutable, i.e., capable of satisfying functional requirements, if significant degradation or loss of functionality occurs".

- **Resourcefulness:** “The ability to diagnose and prioritize problems and to initiate solution by identifying and mobilizing material, monetary, informational, technological and human resources”.
- **Rapidity:** “The capacity to restore functionality in a timely way, containing losses and avoiding disruptions”.

The framework has been discussed based on the perspective of the transportation system as an example in the referenced paper, but this can be equally applicable in the power system domain. From the perspective of Bulk Power Systems, robustness refers to the ability of the ICS or SCADA systems to withstand the adverse impact caused either by a disaster event or cyber-attacks. In other terms, how much the system can maintain its most critical services and availability during an adverse event. Redundancy can be thought of as the availability of critical functionalities through alternate ICS systems such as redundant RTU, PLC, etc. Resourcefulness refers to the system and human capacity to diagnose the problem, identify and application of the best solution with shorter loss of the functionality and services. Rapidity refers to the time duration within which the system can be restored to its pre-adverse events conditions and provide normal services without loss or reduction in service availability.

2.3.4 Conceptual Framework for Urban Energy Resilience

Ayyoob Sharifia and Yoshiki Yamagatab [32] have proposed a definition of resilience with respect to energy systems by reviewing extensive technical literature as claimed by the authors. Their study defines energy resiliency as a range of preparation, absorption, recovery, and adaptation measures that ensure availability, accessibility, affordability, and acceptability of energy supply, transmission, and distribution over time. The authors also emphasized the need for considering the criteria and indicators that can address both mitigation and adaptation aspects of urban climate changes related to energy resiliency.

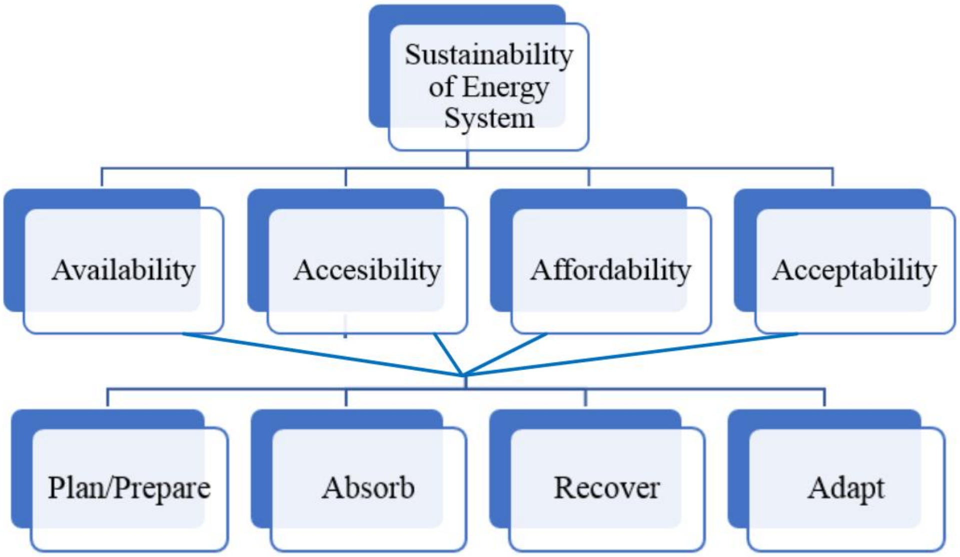


Fig. 5. Conceptual Framework of Urban Energy Resilience [32].

In the above referenced article, the authors proposed an energy resilient urban system framework able to ensure availability, accessibility, affordability, and acceptability of energy supply, under varying conditions, through enhancing its ability to plan/prepare for disaster, absorb its initial shocks, recover rapidly and adapt and self-organize. The conceptual diagram illustrating the relationship between the concepts discussed in the article is being depicted here in Fig. 5.

2.3.5 Resilience Frameworks for Engineered and Infrastructure Systems

In the article, “A metric and frameworks for resilience analysis of engineered and infrastructure systems” [33], Royce Francis and Behailu Bekera have proposed a resilience analysis framework and a metric for measuring resilience. The proposed analysis framework consists of system identification, resilience objective settings, vulnerability analysis, and stakeholder engagement which are closely related to each other.

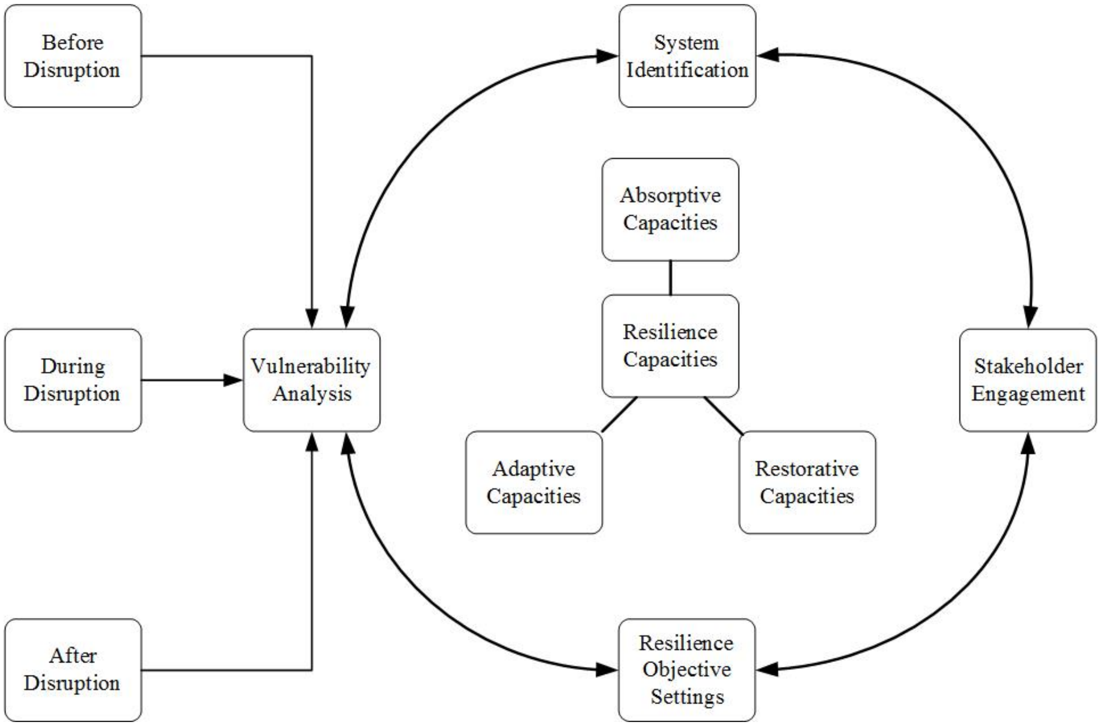


Fig. 6. Resilience framework [33].

The implementation of this framework is focused on the achievement of three resilience capacities: adaptive capacity, absorptive capacity, and recoverability. The proposed resilience analysis framework is presented in Fig. 6. This framework depicted in Fig. 6 consists of five components.

1. system identification
2. vulnerability analysis (before, during and after disruption)

3. resilience objective setting (identifying goals such as normal performance or basic identity to be achieved or sustained)
4. stakeholder engagement (coordination, cooperation& information sharing) and
5. resilience capacities.

The proposed resilience approach emphasizes an assessment of the system’s ability to (i) identify and absorb potential disruptions; (ii) develop adaptive means to manage and accommodate changes within or around the system; and (iii) establish response behaviors targeted at either building the capacity to withstand the disruption or recover as quickly as possible after an impact. These capacities are in line with the definitions provided by NAS in section 2.3.1.

2.4 NIST SP 800-82 Guidelines

This section presents the SCADA systems overview and NIST CSSP Defense-In-Depth recommended architecture for ICS security. This architecture is used as a base in our model and algorithm development. That is why we need a thorough understanding and discussion of the important guidelines and recommendations as proposed in NIST SP 800-82.

2.4.1 SCADA Systems Overview

SCADA systems are used to control dispersed assets. Centralized data acquisition and monitoring and control are some of the important functionalities of the control system network where the SCADA system can be hosted. SCADA systems integrate data acquisition systems with data transmission systems and HMI software to provide a centralized monitoring and control system for numerous process inputs and outputs [34]. SCADA systems are designed to collect field information, transfer it to a central computer facility, and display the information to the operator graphically or textually, thereby allowing the operator to monitor or control an entire system from a central location in near real time. Typical field devices include PLC (programmable logic controller), RTU (Remote Terminal Unit), IED (Intelligent Protective Device) such as the relay and modem, etc.

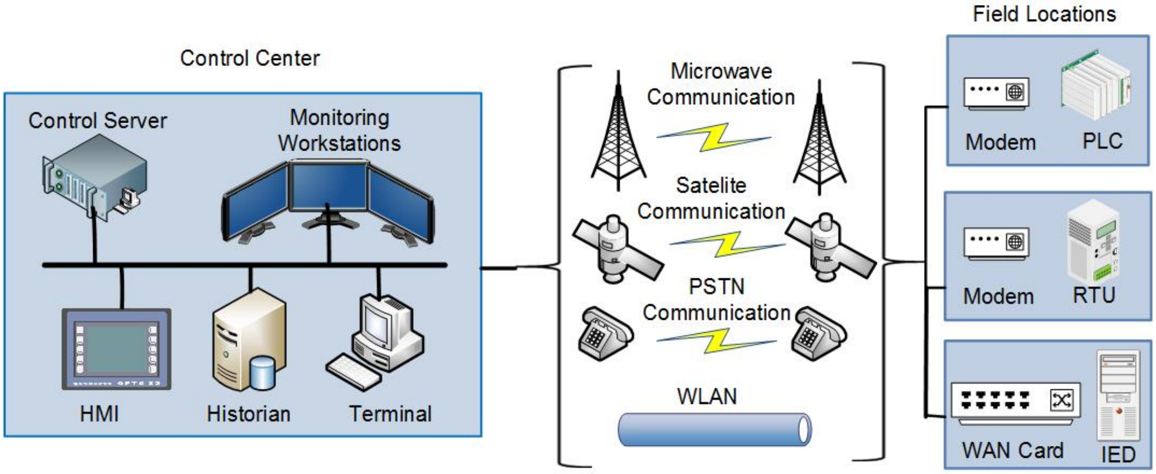


Fig. 7. Generic SCADA and ICS communication network in energy sector.

Fig. 7 shows a generic SCADA and ICS communication network for the energy sector. The control center is normally hosted with control servers, historians, HMI (Human Machine Interface), operator terminals and monitoring workstations, etc. The filed locations consist of the ICS such as PLC, RTU, and IED where communication is established with the control center through using some modem or WAN cards using communication networks such as WLAN communication.

2.4.2 NIST ICS Security Architecture

The National Institute of Standard and Technology (NIST) has provided several guidelines for designing ICS network architecture in its “Guide to Industrial Control Systems (ICS) Security” [35]. In general, the ICS network should be separate from the corporate network in terms of traffic flow. Internet, FTP, email, WAP and remote access services are normally permitted in the Corporate network for business communications, but the same services should not be allowed on the ICS network. If ICS network traffic is not separated and being carried with the corporate network traffic, the ICS networks can be subjected to DoS (Denial of Service) or Man-in-the-Middle attacks. If the two networks are not separated, then the security flaws present in the corporate network due to software vulnerabilities can affect the security of the ICS or control system network. Often practical issues, such as the cost of maintaining separate traffic by creating a homogenous network, can lead to the connection between the ICS and the corporate network, which can have a high-security risk. NIST has suggested that in those cases, it should be protected by boundary protection devices such as the use of a firewall and DMZ. A DMZ is a separate network segment that connects directly to the firewall. Servers containing the data from the ICS that needs to be accessed from the corporate network are put on the DMZ. The ICS-CERT recommended practices working group provides additional guidance as recommended practices [36].

ICS network should be segregated. This can be done by implementing logical network separation such as using VLAN (Virtual Local Area Networks), Encrypted Virtual Private Networks (VPNs), IP filtering, maintenance of whitelists instead of blacklists and used of boundary protections. Boundary protection devices can control the flow of information between interconnected security domains to protect the ICS against malicious cyber adversaries.

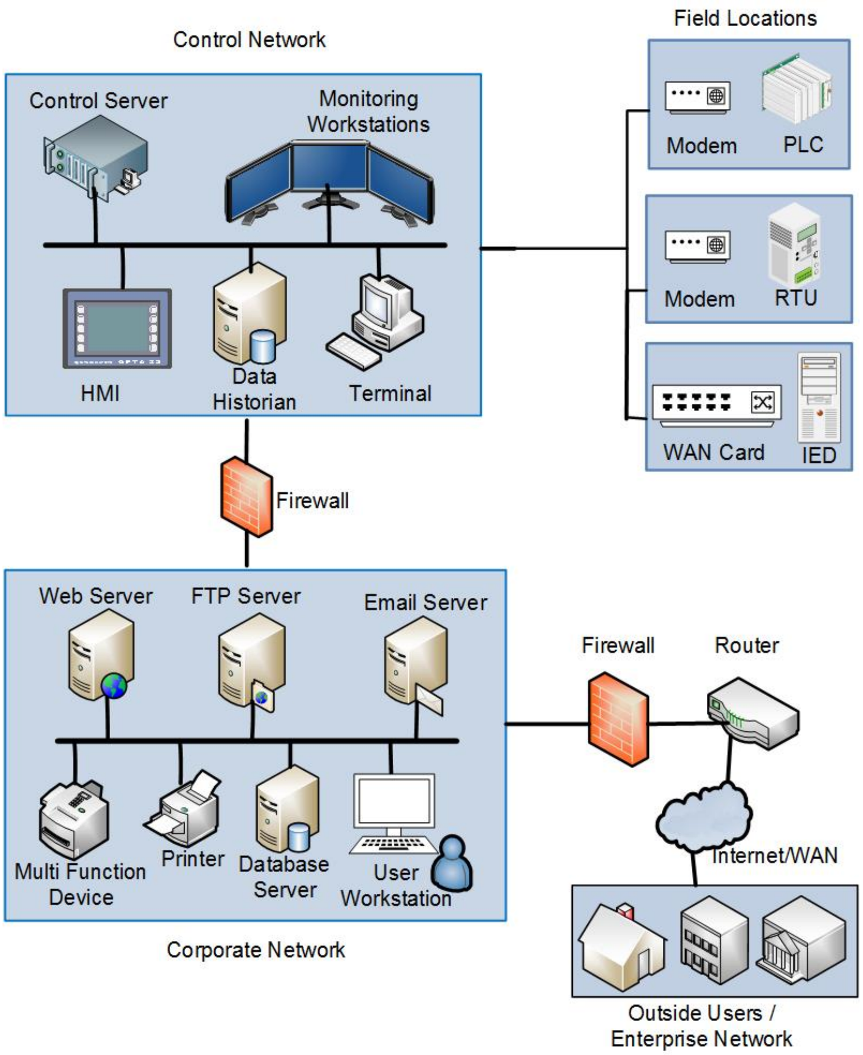


Fig. 8. Corporate Network and Control Network Separation Example Using Firewall [35].

Fig. 8 shows an initial consideration by NIST, where the corporate network and control network are being separated by using a firewall in between them. Outside users and enterprise networks communicate with the corporate network through the firewall, router and traffic passing through the internet or WAN. Any communication that comes from the outside world through the internet or WAN can pose threats to the ICS, so those traffic communications should be separated using firewalls between the corporate and control network so that the unnecessary communication between these two layers can be controlled. An illustration of this network is given here. Let us say that the data historian resides on the control network and no firewall between the corporate and control layer is present to control the traffic communication. The enterprise world or the outside users may communicate with the data historian and normally this communication occurs at the application layer as Hypertext Transfer Protocol (HTTP) request. Flaws in the historian's application layer code could result the data historian being compromised. Once the historian is compromised, the remaining network elements on the control network becomes vulnerable. Thus, a firewall rule must exist that controls the traffic from the corporate layer to the control layer.

Although the above network segregation improves network security, it doesn't diminish the communication between the corporate and control layer; thus, security threats to the ICS network controlled by the control layer remain. Thus, another separation can be implemented using a DMZ, where the DMZ can host the data historian. If the DMZ hosts the data historian being separated from the control layer using a firewall, then the Corporate network need not communicate with the control layer; rather, it needs to communicate with the DMZ layer. An illustration is given in Fig. 9.

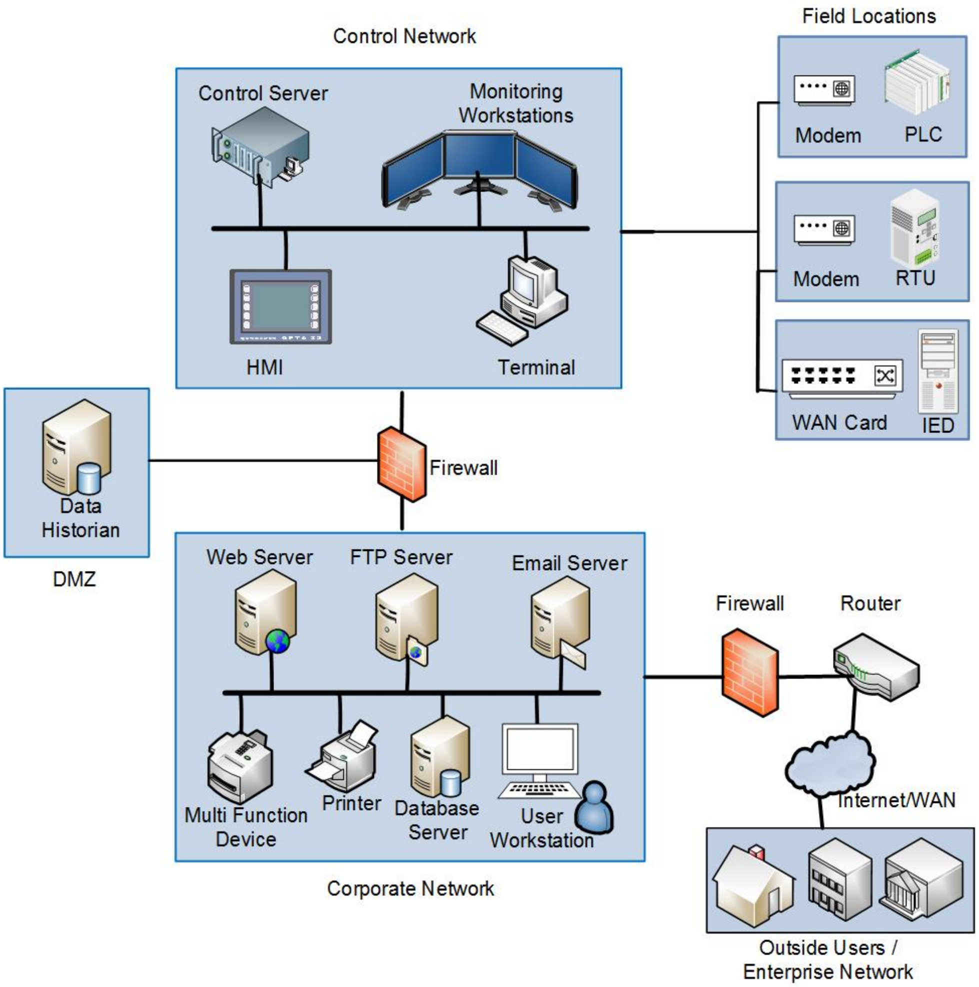


Fig. 9. Corporate Network and Control Network Separation Example Using DMZ and Firewall [35].

According to NIST, the use of firewalls with the ability to establish a DMZ between the corporate and control networks can offer a significant improvement in the security of ICS. Each DMZ holds one or more critical components, such as the data historian, the wireless access point, etc. In effect, the use of a DMZ- capable firewall allows the creation of an intermediate network. To create a DMZ, the firewall requires offering three or more interfaces. One of the interfaces is connected to the corporate network, the second interface is connected to the control network, and the remaining interfaces are connected to the shared or insecure devices such as the data historian server or wireless access points on the DMZ network. NIST recommends that the firewall rulesets should be such that only connections between the control network and DMZ that are initiated by control network devices are permitted.

2.4.3 NIST CSSP Recommended Defense-In-Depth Architecture

In general, any single security solution cannot be adequate for protecting the ICS. That is why a multiple layer strategy involving multiple overlapping security mechanisms is being recommended by researchers so that the impact of a failure in any mechanism can have minimal impact on the ICS. This technique is also known as the defense-in-depth security strategy. Fig. 10 shows the ICS defense-in-depth architecture strategy. The strategy has been developed by the DHS Control Systems Security Program (CSSP) NCCIC/ICS-CERT Recommended Practices committee.

The CSSP Defense-In-depth strategy for ICS has been described in "Control Systems Cyber Security: Defense in Depth Strategies" [37]. The strategy includes the use of firewalls, DMZs, and intrusion detection capabilities throughout the ICS architecture. The use of multiple demilitarized zones in the architecture provides the added capability to separate functionalities and access privileges and has proved to be very effective in protecting large system architectures comprised of networks with different functional operations.

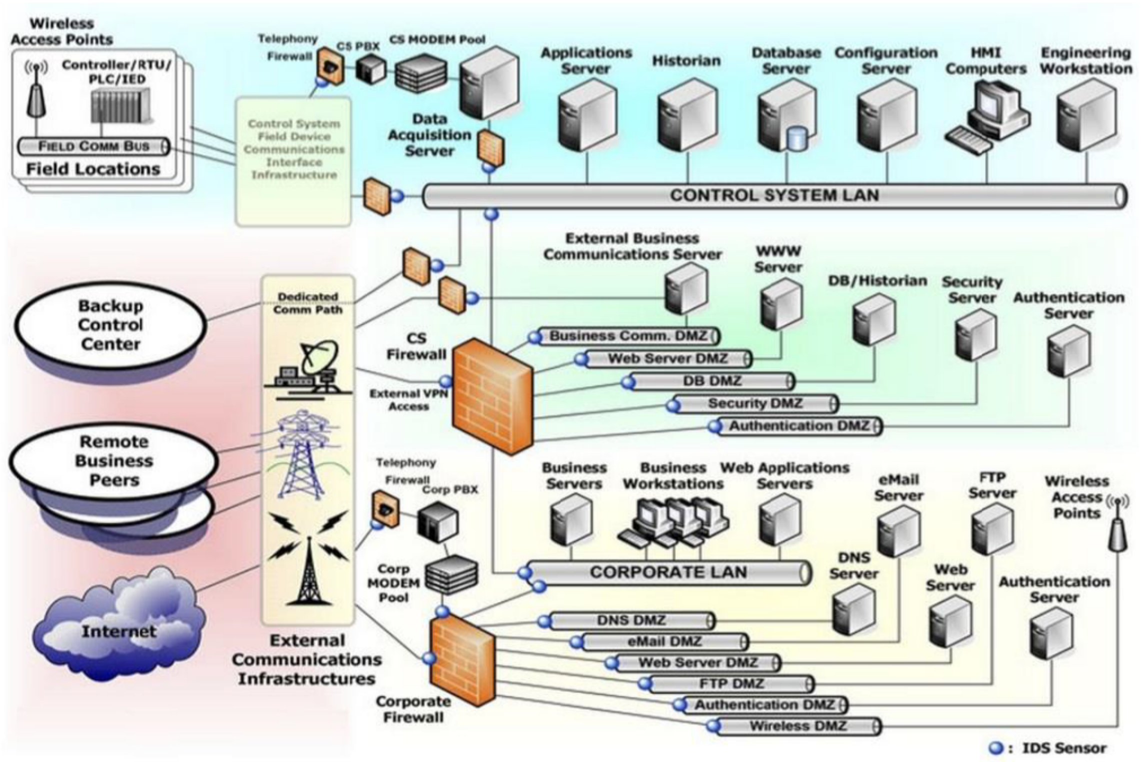


Fig. 10. NIST CSSP Recommended Defense-In-Depth Architecture [37].

The thesis provides the details of NIST in this report because the work has followed the strategy and used the strategy to generate a directed acyclic graph (DAG) model for resilience calculation of bulk power systems and the generation of node ranking algorithms based on the DAG model. The vulnerability graph model discussion is provided in Chapter 3.

CHAPTER 3

VULNERABILITY GRAPH ANALYSIS

Graph theory-based analysis of network security has been done for years. Graphs consist of nodes, links and a mapping function that defines how nodes connect to one another [38]. The most widely used form of the graph in security analysis is Attack Graph. Attack graphs model the possible paths and path combinations that a potential attacker may use to exploit a target network or network element. Attack graph generation is a process that includes vulnerability information processing, collecting network topology and application information, determining reachability conditions among network hosts and applying the core graph building algorithm [39]. A vulnerability graph is a form of attack graph where the vulnerabilities present in the hosts or nodes form the edges between them. In this chapter, a literature review of different scholarly work on graph-based analysis of bulk power system security is presented. The chapter also includes the Directed-Acyclic-Graph model derived from NIST 800-82 SP "Guide to industrial control systems (ICS) security" guidelines, mathematical definition of Resilience and its components. The CVSS base metrics have been presented here to explain how the scores have been used in our calculation. The rationale for using some of the factors for deriving the node ranking algorithm presented in chapter 4 is discussed at the end of the chapter.

3.1 Literature Review on Graph Based Analysis of Network Security

Attack Graphs or vulnerability graphs have been in use in network security analysis for a long time and are playing an increasingly key role in network security analysis. For decades, attack graph analysis has also been used in the analysis of critical infrastructure security such as ICS security, Smart Grid security and computer network security analysis. This section reviews some of the scholarly articles that have used the attack graph in different forms to analyze the security perspectives in different domains especially ICS and power system domain.

Phongphun Kijsanayothin and Rattikorn Hewett [40] have proposed an analytical approach to attack graph analysis for network security. In the paper, the authors have discussed how to use an attack graph to better protect the network and take cost-effective countermeasures by statistically analyzing the attack graphs using reasoning mechanisms based on logical expressions and conditional preference networks. Thaier Hamid and Carsten Maple [41] have proposed a graph theoretical approach to network vulnerability analysis and countermeasures. The automatic formation of vulnerability information has been troublesome. The paper proposed a cost metric based on Markov's model using combinations of vulnerabilities score from CVSS and ranking algorithms. For each host, the authors developed a cost rank Markov's model to reduce the complexity of the attack graph visibility. Cynthia Phillips and Laura Painton Swiler [42] have discussed a graph-based system for network vulnerability analysis. The paper has presented a method for risk analysis of computer networks which is based on the idea of the attack graph having attack states and transitions between the states. The attack graph can be used to identify attack paths that are most likely to succeed. Kerem Kaynar [39] has done a systematic study of the methods applied in each phase of the attack graph generation process which includes the usage of attack graphs for network security. Zhang et al. [43], have proposed an effective method of attack graph generation by modeling the network security status considering the host computer, devices link relation and characteristics of attacks and used a forward-search, breadth-first and depth-limited algorithm to produce the attack routes. Sheyner et al. [44], have presented an automated generation and analysis technique of attack graph, where the authors have used symbolic model checking algorithms to construct the graphs and FSM (Finite State Model) to analyze the graph. Xinming Ou and Anoop Singhal [45], have discussed network security issues based on attack graph techniques where the researchers rely on the MulVAL [46] logical attach graph generation tool for generating the graph. The authors have also considered the vulnerabilities as reported in NVD by scanning the network using MulVAL (Multihost, multistage, Vulnerability Analysis). Zhiming Liu et al. [47] have presented complex network security analysis based on an attack graph model where the authors have considered the scalability problem of attack graph generation when the network size grows. The authors have proposed an attack graph generation method for the complex network by analyzing the network framework and key nodes and then using algorithms to combine greedy policy, forward exploration and backward searching to generate the attack graph. Nayot Poolsappasit [48] has discussed the process of dynamic security risk management using Bayesian attack graphs. The author has proposed a risk management framework using Bayesian networks that enable a system administrator to quantify the chances of network compromise at various levels and has shown how to use this information to develop a security mitigation and management plan using the Bayesian attack graph. Williams et al. [49] have presented GARNET(graphical attack graph and reachability network evaluation tool) to facilitate attack graph analysis. The tool provides a simplified view of critical steps that can be taken by an attacker and of host-to-host network reachability that enables the exploits. The proposed tool also includes zero-day attacks and allows users to perform “what-if” experiments. Marcel Frigault and Lingyu Wang [50, 51] have proposed methods of measuring network security using Bayesian network based attack graphs. The authors focused on measuring the combined effects of the vulnerabilities instead of considering individual vulnerability effect and tried to capture the scenario where exploiting one vulnerability may make it easier for the attacker to exploit the second vulnerability. The work is based on the Bayesian network based probabilistic model to compute security metrics. Another work by Marcel Frigault et al. [52] have also used a dynamic Bayesian network based model and incorporates temporal factors such as the availability of exploit codes or patches. Kyle Ingols et al. [53] have proposed a practical attack graph generation method for network defense. The work has used a multiple-prerequisite attack graph where the authors have used readily available source data to automatically compute network reachability, classify vulnerabilities and recommend actions to improve the network security. Paul Ammann et al. [54] have discussed several issues on the scalable graph-based network vulnerability analysis. The authors have argued that attack graph represents more explicit information than is necessary for the network analyst and thus proposed a more compact and scalable representation of attack graph. S. Jha, O. Sheyner and J. Wing [55] have used attack graphs for network security analysis and proposed two formal analyses of attack graphs. The authors have presented a minimization analysis technique that allows analysts to decide which minimal set of security measures would guarantee the safety of the system. The authors have also provided a formal characterization and presented a greedy algorithm with provable bounds. Barbara Kordy et al. [56] have presented a discussion of attack and defense modeling techniques based on Directed-Acyclic-Graphs (DAGs) where the authors have summarized and compared existing methodologies and their features. Steven Noel et al. [57] have also used attack graphs for measuring the network security risk by using Markov modeling and Bayesian networks. Vivek Shandilya et al. [58] have discussed the use of attack graphs in security systems where the authors have presented a survey and critical study of state-of-the-art technologies in attack graph generation used in the security system and have also identified the challenges and direction of the current research in using attack graphs. Peng Xie et al. [59] have also used Bayesian Networks for Cybersecurity analysis of computer systems. The work is centered around near real-time security analysis and presents the efforts to identify the important types of uncertainty by using Bayesian networks to capture the security analysis. Sebastian et al. [60] have used attack graphs for intrusion detection using vulnerability information. The work mainly focuses on the integration of the attack graph workflow with an Intrusion Detection System (IDS) to improve alert and correlation quality. The vulnerability and system information are considered for the prioritizing of IDS alerts. Chunlu Wang et al. [61] have proposed a novel comprehensive network security assessment approach that supports automatic attack graph generation based on the correlated vulnerability database and quantitative vulnerability assessment utilizing Bayesian attack graphs. Yong Wang et al. [62] have discussed network vulnerability analysis to protect network security based on attack capability transfer using attack graphs. Based on the attack capability transfer, the authors have presented a new method for construction of an attack graph where the authors have considered network vulnerability quantitative analysis and security hardening method based on approximate greedy algorithm. Mohammed Alhomidi and Martin Reed [63] have presented risk assessment and analysis through population based attack graph modeling. The proposed attack graph-based risk assessment model helps organizations and decision makers make appropriate decisions in terms of security risks.

In the power system domain, there have been numerous works that directly consider the attack graph-based analysis for measuring the cybersecurity of power sectors, specifically the control system of the power sectors. Chee-Wooi Ten et al. [64] have used attack-tree based analysis for attack and defense modeling of control systems used in the power sector. The researchers have proposed a SCADA framework and an attack-tree based methodology for impact analysis for the power system control networks. Nian Lie et al. [65] have proposed

MCDM (Multiple Criteria Decision Making) approaches for security assessment of communication networks of power control systems using an attack graph. The authors have decomposed the overall security assessments in two parts; one is the security analysis model for a power control system using an attack graph that includes a construction algorithm and vulnerability function, and another one is based on the quantification of the security degree in each control step which is a hybrid MCDM approach integrated with an analytic hierarchy process (AHP). Saman Zonouz et al. [66] have presented a security-oriented cyber-physical state estimation (SCPSE) for power grid critical infrastructures which at each time instant identifies the compromised set of hosts in the cyber network and the maliciously modified set of measurements obtained from power system sensors. The authors have used an attack graph template (AGT) for the analysis. Yichi Zhang et al. [67] have considered SCADA cybersecurity in relation to the power system reliability. Reliability of the power system can be impacted by various cyber-attacks. The paper considered four attack scenarios for cyber components in networks of the SCADA systems where the authors have used two Bayesian attack graph models to illustrate the attack procedures and evaluated the probabilities of successful cyber-attacks. Ceeman et al. [68] have proposed CPIndex, a security-oriented stochastic risk management technique that calculates cyber-physical security indices to measure the security level of the underlying cyber-physical settings. The proposed CPIndex implements belief propagation algorithms on the created stochastic models combined with a novel graph-theoretic power system indexing algorithm to calculate the security-level of the system’s current cyber-physical state. The authors have used a dependency graph in modeling the CPIndex.

Most of the above works related to the cybersecurity analysis of either the computer networks or electrical smart grid or SCADA systems have used the graph theory such as

Bayesian attack graph or Directed Acyclic graph or Markov process-based graph. In this thesis, the DAG model has been used because of the resilience equation that has been developed based on the DAG model. Also, the Bayesian attack graphs can illustrate the probabilistic approach by using the exploitability, but using Bayesian networks don’t incorporate impact caused by each vulnerability.

3.2 DAG Model

In an earlier work, Sachin et al. [5], developed a Directed Acyclic Graph (DAG) model based on the CSSP Defense-In-Depth Architecture presented in chapter 2. In this thesis, we have used the same DAG model for ranking critical nodes and thus used the DAG model for resilience calculation. The following subsections discuss the derivation of the DAG model and how the model is being used for resilience formulation.

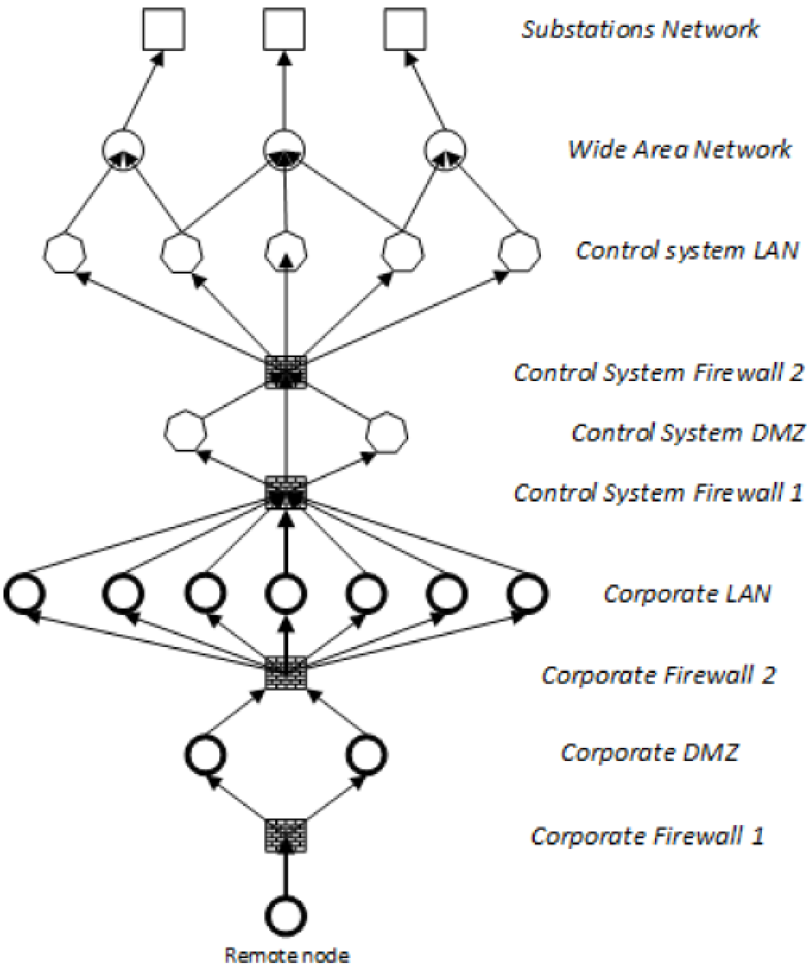


Fig. 11. Multi-layered DAG Model for Bulk Power System [5].

The graph is organized into 10 layers which correspond to security domains implementing security policies and protocols as described in NIST 800-82 report. Nodes in the higher layers are more critical than nodes in lower layers. For example, to ensure operational resilience, the nodes in the control system LAN are more critical than the nodes in the corporate LAN. Some of the layers have been discussed in the following subsections:

1. **Substation layer:** This top layer corresponds to the substations of bulk power systems. There are three types of substations: power generation nodes or generators that generate the power, transmission nodes or transmitters that transmit the power among high voltage transmission lines, and distribution nodes that distributes the power to local distribution grids and end users. The power nodes are not directly attacked by the cyber-attackers; rather, the field location devices which control the power nodes are being targeted by the intruders. That is why field location devices are also being considered as nodes belonging to the substation layer. Typical components that can be found at field location nodes are Remote Terminal Unit (RTU) or Programmable Logic Controller (PLC) which control through some actuators/relays the process running on the node, some sensors which measure the physical state of the process, some communication equipment and Intelligent Electronic Devices (IED) for connection with the control server. This layer is the most important as an attack here can have an impact on the whole network.
2. **Control System LAN:** The Control System Local Area Network (CS LAN) corresponds to the control center which includes equipment like control servers, communications routers, engineering workstations, Human Machine Interfaces (HMI), Application Servers and Historian databases which are all connected to a LAN. This layer is for monitoring alarms, performance data collections and reporting, and configuration change of the substation's network, etc.
3. **Communication Wide Area Network:** Substations need to communicate with the control system. Sensors at the field locations need to send their measurements to the control center, and the control center needs to process the received data with a control algorithm and send some commands to the actuators located at field locations. The control center needs to be

highly connected to the field location network and a Wide Area Network is needed as field locations are distributed over the whole country.

4. **Corporate Local Area Network:** The corporate local area network is a group of hosts connected together for the business communications and outside enterprise communication. It consists of business workstations, web servers, email servers, DNS and application servers. Users of this layer don't have in general the skills and the experience to operate on the Control System LAN. In addition, the Corporate LAN and the Control System LAN have very different traffic.
5. **Firewalls and DMZ layers:** Ideally the Control System LAN should be physically isolated from the Corporate LAN to have a more secure network. However, in practice, some considerations like installation cost or network homogeneity make necessary a connection between both networks. Therefore, as guided in CSSP Defense-In-Depth architecture, the two networks should be at least logically separated by a boundary device like a Control System firewall and establish Demilitarized Zones (DMZ) between the Control System LAN and the Corporate LAN which can significantly reduce the chances of successful attacks because of traffic segregation. By creating a DMZ, no direct communication paths are required from the Corporate LAN to the Control System LAN.

The DAG model presented above has been elaborated in the following diagram by showing the nodes in each layer and the connectivity between them based on the CSSP Defense-In-Depth Architecture.

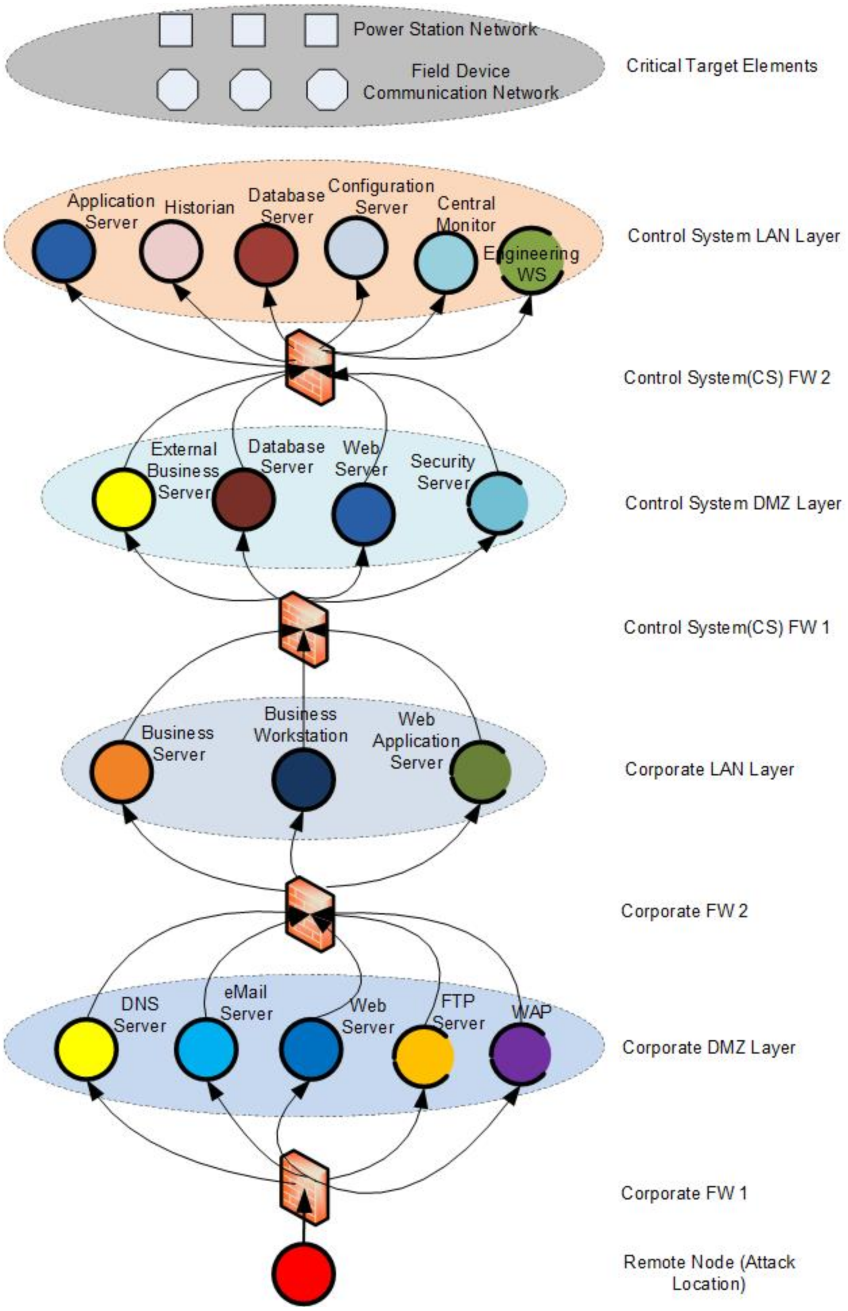


Fig. 12. Elaborated DAG Model for Bulk Power System.

3.3 Resilience Quantification From DAG

This section presents the necessary definitions for the quantification of resilience for the BPS. Some of the most important topics to discuss are node vulnerability, channel vulnerability cost, critical functionality and resilience.

3.3.1 DAG definition

Generally, a graph is a 3-tuple defined by the set equation $G = [N, L, f]$, where N is the set of nodes, L is the set of links or edges between nodes and f is a mapping function where $f\colon L \rightarrow N \times N$, which maps links into pairs of nodes [38].

3.3.2 Node Vulnerabilities and Edge Cost Definition

Each node in the vulnerability graph consists of one or more vulnerabilities associated with its product configuration. Such as, if the node is a Microsoft XP computer, then this host has the vulnerabilities of the Microsoft XP product. According to the DAG, there are as many edges between a source and a destination node as the number of vulnerabilities possessed by the destination node. That means, edges between node i and j, where i is the source node connected with destination node j, the number of edges between (i, j) belongs to a set of all the vulnerabilities of node j and being defined as $e_{ij} \in Vuln_j$, where $Vuln_j = \{vuln_{j1}$, $vuln_{j2}, vuln_{j3}, \dots., vuln_{jn}\}$. As each vulnerability is scored by the base metrics of CVSS, the edge between two nodes has the quantitative values of Access Vector, Access Complexity, Access Authentication and Confidentiality Impact, Integrity Impact and Availability Impact. Thus, if an edge is being constructed between node (i, j) by a vulnerability $vuln_{j1}$, then its Access Vector is denoted as $A_V(vuln_{i,j1})$. Similarly, Access Complexity and Access

Authentication are denoted as $A_C(vuln_{i,j1})$ and $A_A(vuln_{i,j1})$. The edge cost corresponding to the vulnerability $vuln_{j1}$ is defined by the following equation [5]:

$$W_{vuln_{i,j1}} = 10 - 20 \times A_C(vuln_{i,j1}) \times A_V(vuln_{i,j1}) \times A_A(vuln_{i,j1}) \qquad (1)$$

The edge cost definition restricts the value as $0 \leq W_{vuln_{i,j1}} \leq 10$. The lower the cost, the more exploitable the vulnerability.

3.3.3 Channel Vulnerability Path Definition

A channel vulnerability path e is a sequence of vulnerabilities among N nodes $i_1, i_2, \ldots\ldots i_N$ by exploiting $N-1$ vulnerabilities $vuln_{i_1 i_2}, vuln_{i_2 i_3}, vuln_{i_3 i_4}, \ldots\ldots, vuln_{i_{N-1} i_N}$. The cost of the channel vulnerability path by exploiting the $N-1$ vulnerabilities between nodes $i_1, i_2, \ldots\ldots i_N$ is given by:

$$c(e) = \sum_{k=1}^{N-1} W_{vuln_{i_k i_{k+1}}} \qquad (2)$$

The lower the cost of the channel vulnerability path, the easier it can be exploited by the attacker.

3.3.4 Critical Functionality Definition

Critical functionality (CF) is referred to as the functionality function [69] and performance of the system [70] and quality of the system [71]. One example of CF can be the percentage of nodes that are functioning [72]. From the perspective of the DAG graph, the critical functionality is evaluated by the level of availability of a given target node. The CVSS base metrics consider that a vulnerability can impact the availability of a node by none, partial and complete level. The impact of a vulnerability is calculated as follows.

$$IS = 10.41 \times \left(1 - (1 - I_C) \times (1 - I_I) \times (1 - I_A)\right) \quad (3)$$

From the base metrics definition, the critical functionality for each vulnerability can be defined as:

$$K_e(t) = \begin{cases} 1 - 0 = 1, if\ impact\ is\ none \\ 1 - 0.275 = 0.725, if\ impact\ is\ partial \\ 1 - 0.660 = 0.340, if\ impact\ is\ complete \end{cases} \quad (4)$$

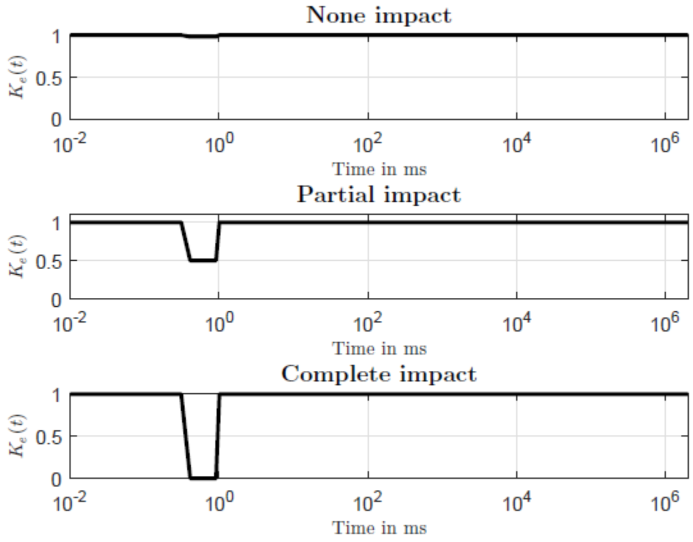


Fig. 13. Critical functionality based on availability impact [5] .

Fig. 13. shows the critical functionality of the system for the none, partial and complete impact over a time period of T. The area encircled by the curves is a measure of the resilience of the system. Based on the DAG model and the definitions provided in the above sections, Resilience of a target node is defined as [5]:

$$R = \frac{1}{|E|} \sum_{e \in E} \frac{c(e)}{c_{max}} \times \left[\frac{1}{T} \int_0^T K_e(t) dt\right] \quad (5)$$

where, $|E|$ is the cardinality of E, $c(e)$ is the individual path cost, c_{max} is the maximum value of the cost of all the paths, $K_e(t)$ is the critical functionality based on the impact of vulnerabilities.

3.4 CVSS Metrics

Identification and assessment of vulnerabilities are one of the crucial factors in network security. This section focuses on the Common Vulnerability Scoring System and our choice of using CVSS Base Metrics for the edge weights calculation.

3.4.1 Different Vulnerabilities Scoring Systems

In computer and network security, a vulnerability refers to a weakness or a bug or an exposure of a software or hardware application, system, device or service which allows an attacker to exploit the system and possibly lead to loss of confidentiality, integrity, and availability. In most cases, vendors or manufacturers of hardware and software keep tracks of the vulnerabilities associated with their products in their own way. Over the past several years, some large computer security vendors and not-for-profit organizations have developed, promoted, and implemented procedures to rank information system vulnerabilities [73], such as National Vulnerability Database, US-CERT [74], ISS X-Force, Symantec, Microsoft, Sun, Redhat, and so on. Some of those are discussed below.

The NVD is the U.S. government repository of standards based vulnerability management data represented using the Security Content Automation Protocol (SCAP) [75]. Along with the other contents, the NVD includes databases of security-related software flaws and exploit and impact metrics. The NVD supports both Common Vulnerability Scoring System

(CVSS) v2.0 and v3.0 standards [76]. The NVD database provides CVSS base metrics which gives the quantitative score of each vulnerability.

The Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) collaborates with international and private sector Computer Emergency Response Teams (CERTs) to share control systems-related security incidents and mitigation measures [77].

The Symantec Security Response Threat Severity Assessment [78] evaluates computer threats (viruses, worms, Trojan horses and macros) and classifies them into clearly defined categories of risk for computer users [79]. Each threat is ranked high, medium or low severity based on the number of impacted computer systems. The three major threat components that are considered by Symantec to determine the severity rating are the extent to which a malicious program is un-noticeable, the damage that a malicious program causes if encountered and the rate at which a malicious program spread.

Microsoft has its own vulnerability severity rating system [80] to help its customers understand the risk associated with each vulnerability. Their ratings are classified and defined as given in TABLE 1 below:

TABLE 1

MICROSOFT SECURITY SEVERITY RATING SYSTEM [1]

Rating	Definition

Critical	A vulnerability whose exploitation could allow code execution without user interaction. These scenarios include self-propagating malware (e.g. network worms), or unavoidable common use scenarios where code execution occurs without warnings or prompts. This could mean browsing to a web page or opening email. Microsoft recommends that customers apply Critical updates immediately.
Important	A vulnerability whose exploitation could result in compromise of the confidentiality, integrity, or availability of user data, or of the integrity or availability of processing resources. These scenarios include common use scenarios where the client is compromised with warnings or prompts regardless of the prompt's provenance, quality, or usability. Sequences of user actions that do not generate prompts or warnings are also covered. Microsoft recommends that customers apply Important updates at the earliest opportunity.
Moderate	Impact of the vulnerability is mitigated to a significant degree by factors such as authentication requirements or applicability only to non-default configurations. Microsoft recommends that customers consider applying the security update.
Low	Impact of the vulnerability is comprehensively mitigated by the characteristics of the affected component. Microsoft recommends that customers evaluate whether to apply the security update to the affected systems.

The Common Weakness Enumeration (CWE) [81] is a formal list of software weaknesses which serves as a common language for describing software security weaknesses and provides a common baseline standard for weakness identification, mitigation, and prevention efforts.

According to MITRE [82], Software acquirers want assurance that the software products that are being obtained are reviewed for known types of security flaws, and the acquisition groups in large government and private organizations are moving forward to use these types of reviews as part of future contracts.

3.4.2 Why CVSS Metrics?

There are a number of other vulnerability "scoring" systems managed by both commercial and non-commercial organizations as described in the previous section. They each have their way of scoring the severity; thus, they differ by what they measure. As the Information Technology management system requires identification and assessment of vulnerabilities across a broad range of hardware and software platforms manufactured and marketed by different vendors, it requires some common standards to quantify and evaluate the vulnerabilities associated with hardware or software products. The Common Vulnerability Scoring System (CVSS) is an open framework that provides quantitative scores of each of the vulnerabilities and is used by many other organizations such as NVD, ICS-CERT, MITRE, etc.

The Forum of Incident Response and Security Teams (FIRST) has categorized CVSS into 3 groups: Base, Temporal and Environmental [83]. Each group produces a numeric score ranging from 0 to 10. Each vulnerability has a textual representation that corresponds to a quantitative measure between 0 and 1. As defined by the FIRST, the Base group represents the intrinsic or inherent qualities of a vulnerability, the Temporal group reflects the dynamic characteristics of a vulnerability that change over time and the Environmental group represents the characteristics of a vulnerability unique to a specific user's platform and environment. Security-related professionals and researchers can benefit by using CVSS for scoring IT vulnerabilities. Thus, this thesis uses the CVSS base metrics for calculating edge costs in the vulnerability graphs using the metrics under Base metric groups such as Access Vector, Access Complexity, Access Authentication, Confidentiality Impact, Integrity Impact and Availability Impact.

3.4.3 Elaboration of CVSS Metrics

As the CVSS Base metrics are being used for the calculation of resilience and developing of the ranking algorithm, so it is needed to discuss the Base Metrics elaborated on here.

TABLE 2

CVSS ACCESS VECTOR DESCRIPTION [2]

Metric Value	Description
Local (L)	A vulnerability that can be exploited only if the attacker has local or physical access to the system such as if he has a local account to login to the system. Examples of attacks using Local access are Firewire/USB attacks or local privilege escalations (by using sudo in Linux/Unix).
Adjacent Network (A)	A vulnerability that is exploitable with adjacent network access such as that which requires the attacker to have access to either the broadcast or collision domain of the vulnerable software. Examples can be access to IP subnet, Bluetooth, IEEE 802.11, and local Ethernet segment.
Remote Network (N)	A vulnerability that is exploitable without having local network access or local access and can be performed remotely. An example of a network attack is an RPC (Remote Procedure Call) buffer overflow.

3.4.3.1 Access Vector (Av)

This metric reflects how the vulnerability is exploited. The more remote an attacker can be to attack a host, the greater the vulnerability score. The possible value of the Access Vector and corresponding definition are given below in TABLE 2.

3.4.3.2 Access Complexity (Ac)

FIRST defines the “Access Complexity” metric as “a measure of the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system” [2]. In other words, it is the extent of technical hurdles that the attacker needs to overcome to launch a successful attack. There are vulnerabilities to which the attacker may gain access with less effort (e.g., in case of buffer overflow attack in an Internet service, the attacker needs to find the target system location; after that, the attacker can launch an exploit at his convenience). There are other vulnerabilities which require victims’ involvement in additional steps to successfully exploit the vulnerability, such as a vulnerability in an email client is only exploited after the user downloads and opens a malicious file attached to email. The lower the required complexity, the easier it is for the attacker to exploit and, thus, the higher the vulnerability score. TABLE 3 below lists the possible values of the access complexity metric.

TABLE 3

CVSS ACCESS COMPLEXITY DESCRIPTION [3]

Metric Value	Description

High (H)	Specialized conditions exist, such as a race condition with a narrow window, or a requirement for social engineering methods that would be readily noticed by knowledgeable people.
Medium (M)	There are some additional requirements for access, such as a limit on the origin of the attacks, or a requirement for the vulnerable system to run with an uncommon, non-default configuration.
Low (L)	There are no special conditions for access to the vulnerability, such as when the system is available to a large number of users, or the vulnerable configuration is ubiquitous.

3.4.3.3 Access Authentication (A_A)

FIRST defines "Access Authentication" as a "measure of the number of times an attacker has to authenticate to a target in order to exploit a vulnerability". This metric doesn't reflect the strength or complexity of the authentication process; it only gives an idea about the required number of authentications to provide credentials by an attacker before an exploit may occur. The fewer authentication instances that are required, the easier it is to exploit and the higher the vulnerability score. The possible values for this metric are listed in TABLE 4.

TABLE 4

CVSS ACCESS AUTHENTICATION DESCRIPTION [2]

Metric Value	Description

Multiple (M)	Exploiting the vulnerability requires that the attacker authenticate two or more times, even if the same credentials are used each time. An example is an attacker authenticating to an operating system in addition to providing credentials to access an application hosted on that system.
Single (S)	The vulnerability requires an attacker to be logged into the system (such as at a command line or via a desktop session or web interface).
None (N)	Authentication is not required to exploit the vulnerability.

3.4.3.4 Confidentiality Impact (I_C)

The confidentiality impact (I_C) metric has been defined by FIRST as the impact on the confidentiality of data processed by the system. Confidentiality refers to limiting information access and disclosure to only authorized users, as well as preventing access by, or disclosure to, unauthorized ones. Increased confidentiality impact increases the potentiality for loss of information and thus increases the vulnerability score. The possible values for this metric are listed in TABLE 5.

TABLE 5

CVSS CONFIDENTIALITY IMPACT DESCRIPTION [3]

Metric Value	Description
None (N)	There is no impact to the confidentiality of the system.

Partial (P)	There is considerable informational disclosure. Access to some system files is possible, but the attacker does not have control over what is obtained, or the scope of the loss is constrained. An example is a vulnerability that divulges only certain tables in a database.
Complete (C)	There is total information disclosure, resulting in all system files being revealed. The attacker is able to read all of the system's data (memory, files, etc.)

3.4.3.5 Integrity Impact (I_C)

According to FIRST, the Integrity Impact measures the impact to the integrity of a successfully exploited vulnerability. Integrity refers to the trustworthiness and guaranteed veracity of information. Increased integrity impact increases the vulnerability score. The possible values for this metric are listed in TABLE 6.

TABLE 6

CVSS INTEGRITY IMPACT DESCRIPTION [3]

Metric Value	Description
None (N)	There is no impact to the integrity of the system.
Partial (P)	Modification of some data or system files is possible, but the scope of the modification is limited.
Complete (C)	There is total loss of integrity; the attacker can modify any files or information on the target system.

3.4.3.6 Availability Impact (I_A)

This metric measures the impact to the integrity of a successfully exploited vulnerability. Integrity refers to the trustworthiness and guaranteed veracity of information. Increased integrity impact increases the vulnerability score. The possible values for this metric are listed in TABLE 7.

TABLE 7

CVSS AVAILABILITY IMPACT DESCRIPTION [3]

Metric Value	Description
None (N)	There is no impact on the availability of the system.
Partial (P)	There is reduced performance or loss of some functionality.
Complete (C)	There is total loss of availability of the attacked resource.

CHAPTER 4

MULTIPLE VULNERABILITY NODE RANK (MVNRANK) ALGORITHM

This chapter focuses on the ranking algorithm. The node ranking algorithm that has been developed as part of the thesis has been called MVNRank which stands for Multiple Vulnerability Node Rank. As stated in previous chapters, MVNRank is an MADM (Multi-Attributes Decision Making) based ranking algorithm. The node ranking process will not only help to improve the resilience but also facilitate the way to harden the network from vulnerabilities and threats by identifying critical nodes and provide suggestions for removing and reducing the exploitable paths. The proposed MVNRank (Multiple Vulnerability Node Rank) algorithm takes into account asset value, the exploitability and impact scores of vulnerabilities as illustrated in CVSS (Common Vulnerability Scoring System), the total number of vulnerabilities each product has and severity of each vulnerability, degree centrality of the nodes and also the relative closeness of the intermediate nodes in the shortest paths to target node. The next sections focus on some of the ranking algorithms based on graph theory and then the details of MVNRank in the subsequent sections.

4.1 Literature Review on the existing ranking Algorithms

There have been numerous approaches developed to deal with the ranking of critical assets in a network or in other applications. One of the widely used ranking algorithms is Google’s PageRank Algorithm. Google's PageRank algorithm [84, 85] sorts the results of a query by the most relevant or important pages that match a given search string so that indexed pages can be listed in order of importance, making it easier for the user to find pages relevant to their search parameters. PageRank assesses the importance of a web page by the number of pages linking to it as well as the importance of these pages. Xing et al. [86] have worked on the PageRank algorithm and proposed a weighted PageRank algorithm. The page rank algorithm approach is not directly applicable to the node ranking of this thesis because it does not consider the edge weights in the form of exploitability and impact scores of the associated vulnerabilities, but the basic concept of PageRank algorithm has been used in a different way while assessing the dynamic asset value based on the dependency relationship of the nodes which is one of the factors considered for the ranking of critical assets. Kijsanayothin et al. [87] have conducted exploit based analysis using Markov computational process and ranked nodes in the attack model in order of their likelihoods of being compromised. Pengfei Li and Xiaofeng Qiu [88] proposed an algorithm named NodeRank which is based on state enumeration attack graphs where the rank value of the nodes shows the likelihood of an intruder reaching this state. Xia Yang et al. [89] proposed DBRank for ranking vulnerabilities to patch in computing networks. DBRank prioritizes vulnerabilities based on the diffusibility and benefit of vulnerability exploitation. This is interesting because the authors have considered the impact as benefits of the attacker. Paul Barford et al., in their book "Cyber Situational Awareness" [90], have explained two different methods to determine and prioritize critical assets: Analytic Hierarchy Process (AHP) and Decision Metric Analysis (DMA). Initially, this thesis considered some of the ways explained in DMA, but these methods don’t incorporate the exploit and impact base metrics which are critical in our case as we are using the network topology derived from NIST CSSP defense-in-depth architecture and the resilience formula depends on those exploitability and impact base scores. Sawilla, Reginald E., and Xinming Ou. [91] discussed the process of identifying critical attack assets in dependency attack graphs. Miura-Ko, Reiko Ann and Nicholas Bambos [92] proposed SecureRank which prioritizes vulnerabilities and network nodes based on the percentage of time a random attacker would spend while trying to exploit them. SecureRank takes into account the network topology and potential node interactions in calculating their relative risk and priority. The algorithm is good for prioritizing vulnerabilities but doesn't take into consideration all the possible path combinations that the vulnerabilities of the intermediate nodes may offer to the attacker. Mehta, Vaibhav et al. [93] proposed a way of ranking attack graphs where the authors considered nodes as system states and edges are transitions between states.

4.2 MVNRank Algorithm Factors and Formulation

To determine the cybersecurity metrics of a complex bulk power system network, it is not sufficient to consider only the effects of exploiting each individual vulnerability and the impact caused by that vulnerability. It is a must for the analyst to take into consideration all the possible attack intrusion and attack scenarios where an attacker may combine several exploits and launch a multi-stage multi-host attack to compromise the security of the ICS system [94]. The same is also necessary to improve node resilience. The MVNRank takes into account asset value, the exploitability and impact scores of each of the vulnerabilities, the total number of vulnerabilities each product has and severity of each vulnerability, and also the relative closeness of the intermediate nodes in the shortest paths to the target node. Considering the intermediate nodes in the shortest paths allows MVNRank to catch up with the multi-stage multi-host attack optimization. Also, the resilience equation used here takes into account all the possible path combinations due to the vulnerabilities of the intermediate nodes and their edge exploitability and impact.

4.2.1 Asset Value and Node Importance in Vulnerability Graph

In general, the Asset value refers to the importance of the files and data stored in a host or server. This is directly related to how much damage an attacker can do to a network by compromising the asset. For example, if an attacker can compromise a database server the information he can steal and thus the damage he can do to the network or organization is much greater than the damage done by compromising a workstation. Asset value is also a measure of the contribution of the node in the overall network compared to its peer nodes. For example, if an attacker can penetrate a workstation probably only the user or application of that workstation is compromised and at risk but no other machine in the network may be affected because they may not have application dependency on that workstation, but if the attacker can successfully compromise a server, then the users or workstations connected to that server may not be able to run the applications on that server. Hence, all the users connected to that server or dependent on the server's application are impacted and are unable to get the service. That is why the server is more important than the workstation here. In our network topology based on the NIST 800-82 CSSP defense-in-depth architecture, the nodes in Control LAN layer such as Historian Servers or Application Servers are more critical in terms of asset value than the nodes in Corporate DMZ layers such as FTP Server, Email Server or Web Server, because if somebody can penetrate the control LAN system machine, he would have more access to the application of the power station network consisting of the Remote Terminal Units (RTU) and Industrial Control Systems (ICS). Normally experienced network administrators assign an asset value from 0 to 1 to the network elements, where 1 is the highest asset value and 0 is no asset value. So far, we know that Google's PageRank algorithm can put importance on the webpage based on the number of links pointing to this page as well the number of links pointing to those linked pages. For determining the asset value of a node in our network topology, we are using almost a PageRank centrality approach in a different way by considering the number of predecessors each node has and the number of predecessors of the present node's predecessors' predecessor nodes until the attack origin node. We have assumed that every predecessor node has some application or service dependency on the successor nodes. Even if they don't have a dependency, there exists some sort of communication channel which can be exploited from the predecessor nodes to the successor node. Thus, the successor node's asset value is dependent on the predecessor's asset value.

Let us consider, the graph is denoted as $G(V, E)$, where V is a set of vertices or nodes and E is the set of edges between nodes. $IntermediateNodes$ is the set of all the nodes that fall in all the possible paths from the source *s* to target *t*. So, if the set of nodes V is defined as $V = \{v_1, v_2, v_3, \ldots\ldots v_{n-2}, v_{n-1}, v_n\}$ and there exists paths from source v_1 to target v_n, which pass through all the other nodes, then $IntermediateNodes = \{v_2, v_3, \ldots, v_{n-1}: v_1 = s \text{ } and \text{ } v_n = t\}$. There are some nodes that may not fall in the paths from the specific source node to target node, hence, $IntermediateNodes \subset V$. Asset value of Node v_i is denoted by A_{v_i} and can be found by considering the number of predecessor nodes as given below.

$$A_{v_i} = N_{PR}(v_i) + \sum_{v_j \in PR(v_i)} N_{PR}(v_j) + \sum_{v_k \in PR(v_j)} N_{PR}(v_k) + \ldots\ldots + \sum_{\substack{v_n \in PR(v_m) \\ v_n = s}} N_{PR}(v_n) \tag{6}$$

where, A_{v_i} is the asset value of Node v_i where $v_i \in IntermediateNodes$, $N_{PR}(v_i)$ is the number of predecessors' of node v_i has, $PR(v_i)$ is the set of all predecessors of node v_i, $v_n = s$ is the attack origin node such as for our case the Corporate Firewall 1. Thus, A_{v_i} is the summation of the number of all the predecessor nodes plus the predecessors of the predecessor nodes until the starting point of attack (node 0).

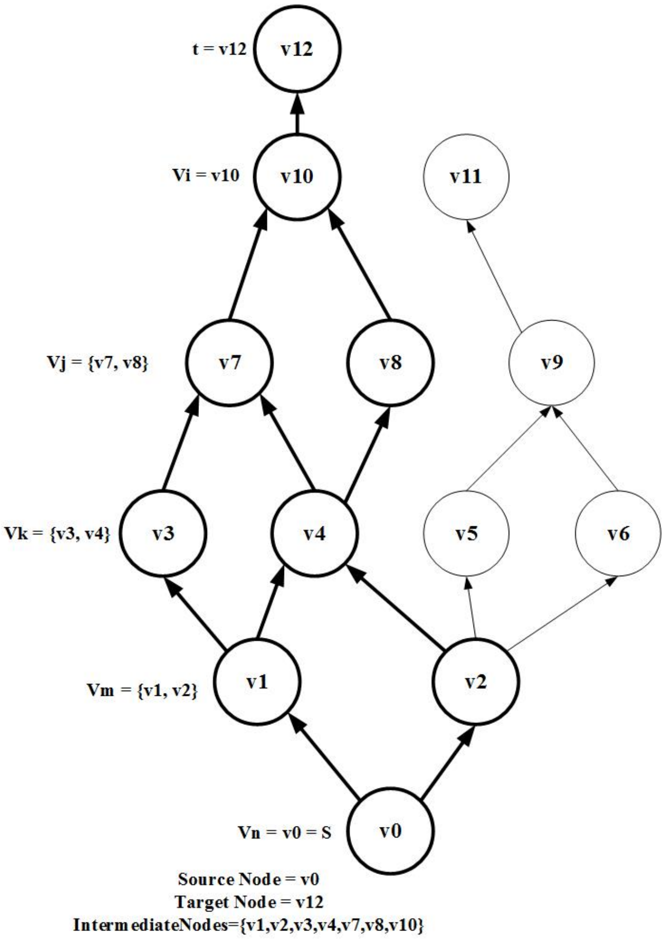


Fig. 14. Asset Value Illustration.

Fig. 14 demonstrates the illustration of the asset value equation. Let us say that we have source node $s = v0$ and target node $t = v12$. Then $IntermediateNodes =$

$\{v1, v2, v3, v4, v7, v8, v10\}$. Now, if we are interested to find the asset value of node $v10$, then $vi = v10, vj = \{v7, v8\}, vk = \{v3, v4\}, vm = \{v1, v2\}, vn = v0 = s,$ $N_{PR}(v_i) = N_{PR}(v10) = 2$ and $PR(v_{10}) = \{v7, v8\}$. Now, we can add weights on the predecessor's value based on the distance of those predecessors from node v_i. Because only the direct predecessors can be directly dependent on successor's services or applications, the rest of the nodes are indirectly dependent on the current node's services and application and thus have reduced weight. The modified form of equation (6) including the distance-based weight method is given below:

$$\begin{aligned} A_{v_i} = N_{PR}(v_i) &+ \sum_{v_j \in PR(v_i)} \frac{1}{D_{(v_j, v_i)}} \times N_{PR}(v_j) \\ &+ \sum_{v_k \in PR(v_j)} \frac{1}{D_{(v_k, v_i)}} \times N_{PR}(v_k) + \dots\dots \\ &+ \sum_{\substack{v_n \in PR(v_m) \\ v_n = s}} \frac{1}{D_{(v_n, v_i)}} \times N_{PR}(v_n) \end{aligned} \tag{7}$$

The above equation can be rewritten in short form as below:

$$A_{v_i} = N_{PR}(v_i) + \sum_{v_j \in IN(v_i)} \sum_{v_k \in PR(v_j)} \frac{1}{D_{(v_k, v_i)}} \times N_{PR}(v_j) \tag{8}$$

where, $IN(v_i)$ is the set of all intermediate nodes through which node v_i is reachable from source node, $PR(v_j)$ is the set of immediate predecessors of node v_j and $N_{PR}(v_j)$ is the number of predecessors of node v_j. It is to be noted that A_{v_i} is not the same for the same host or node in the graph; it is different in the case of a different target node, because the intermediate nodes falling to the paths to the target are different when the target node is changed.

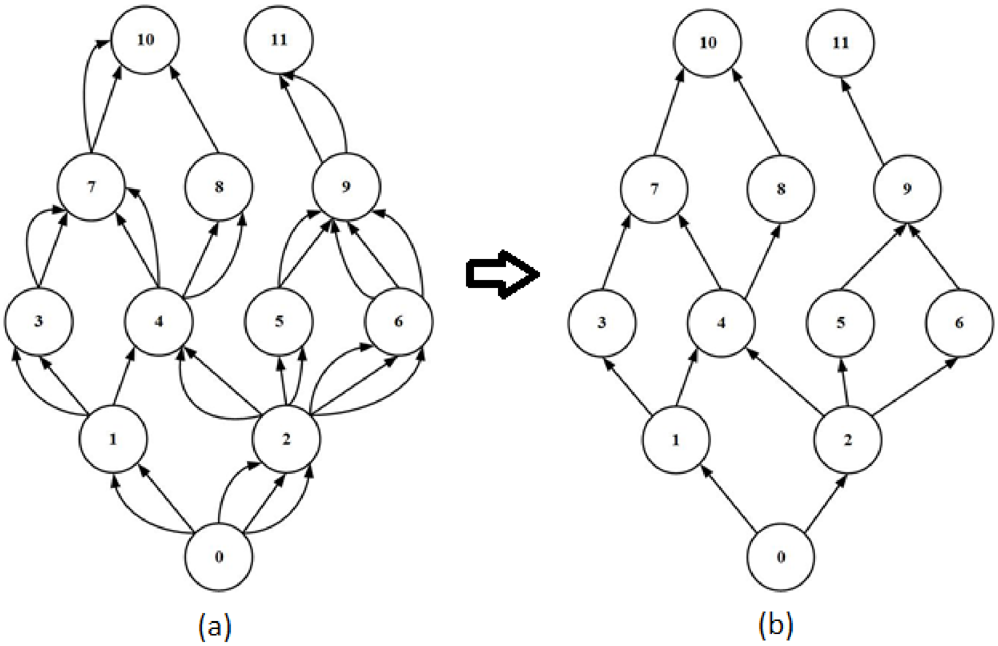


Fig. 15(a). Vulnerability Graph Example for determining Asset Value (b) Simplified form of Fig. 15(a) for asset value determination.

In Fig. 15(a), there can be multiple vulnerabilities possessed by each node and the number of edges between the two nodes is equal to the number of vulnerabilities possessed by the destination node. For example, in (a), node 1 has 2 vulnerabilities, so there are two directed edges from node 0 to node 1. Similarly, node 6 has three vulnerabilities, that is why there are three edges from node 2 to node 6. To determine the asset value of each node as per the formula given in the equation, there is no need to consider the number of vulnerabilities or edges between the nodes, only the number of predecessors and distance from the asset value determinant node are important. Thus, the figure in (a) has been simplified in (b) by removing multiple edges and keeping only one edge between each node for asset value calculation demonstration purposes.

Let us first take node 10 as our target node from an attacker's perspective. The attack origin node is thought to be node 0 for this example. Node 10 can be reachable from node 0 through multiple paths such as 0→1→3→7→10, 0→1→4→7→10, 0→2→4→7→10 and 0→2→4→8→10. Thus, the intermediate nodes that fall in the way of node 0 to node 10 are 1, 2, 3, 4, 7 and 8. Thus, we are interested in determining the asset value of node 1, 2, 3, 4, 7 and 8 which values will be used in ranking of the nodes calculation. Node 1 and 2 have an asset value of 1 as they have only the origin node as the predecessor node, so the second component of the equation is zero for nodes 1 and 2.

TABLE 8

ASSET VALUE DETERMINATION EXAMPLE (NODE 7)

Nodes	Number of Immediate Predecessors	Distance from Asset value determinant node
1, 2	1	2
3	1	1
4	2	1
7	2	0 (asset value determinant node)

Thus, the asset value of node 7 can be calculated using the equation; the demonstration is given below:

$$A_7 = N_{PR}(7) + \left[\frac{1}{D_{(1,7)}} \times N_{PR}(3) + \frac{1}{D_{(1,7)}} \times N_{PR}(4) + \frac{1}{D_{(2,7)}} \times N_{PR}(4)\right] + \left[\frac{1}{D_{(0,7)}} \times N_{PR}(1) + \frac{1}{D_{(0,7)}} \times N_{PR}(2)\right]$$

$$A_7 = 2 + \left[\frac{1}{2} \times 1 + \frac{1}{2} \times 2 + \frac{1}{2} \times 2\right] + \left[\frac{1}{3} \times 1 + \frac{1}{3} \times 1\right]$$

$$A_7 = 2 + (0.5 + 1.00 + 1.00) + (0.33 + 0.33) = 5.16$$

Thus, node 7 has an asset value of 5.16 when the target node is node 10. Similarly, for node 4, the asset value can be determined by using the equation and an illustration is given below:

$$A_4 = N_{PR}(4) + \left[\frac{1}{D_{(0,4)}} \times N_{PR}(1) + \frac{1}{D_{(0,4)}} \times N_{PR}(2)\right] A_4$$

$$A_4 = 2 + \left[\frac{1}{2} \times 1 + \frac{1}{2} \times 1\right]$$

$$A_4 = 2 + (0.5 + 0.5) = 3.00$$

Thus, node 4 has an asset value of 3.00 when the target node is node 10. This is also analogous with our basic concepts that the lower layer's nodes (such as nodes belonging to Corporate DMZ or Corporate LAN) should have less asset value than the upper layer nodes (such as Control DMZ or Control System LAN layer). Also, nodes 1 and 2 have an asset value of 1.00 each as shown here: $A_1 = N_{PR}(1) = 1, A_2 = N_{PR}(2) = 1$.

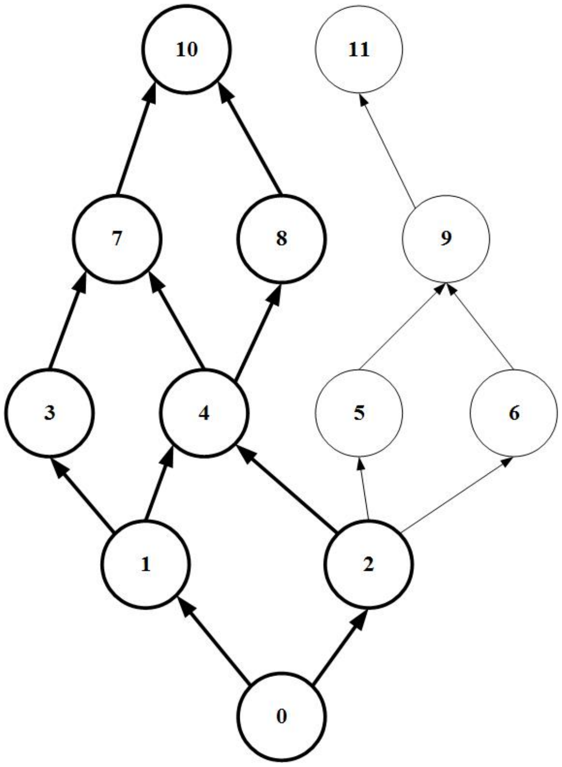


Fig. 16(a). Intermediate nodes (bold) when target is node 10.

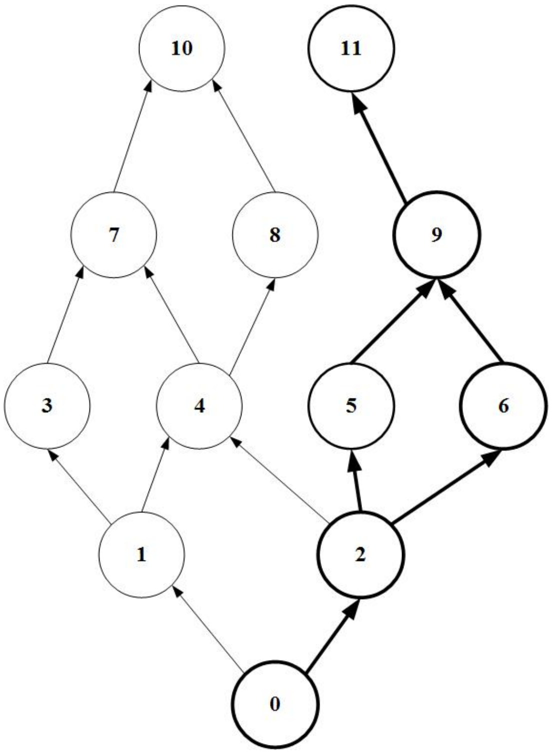


Fig. 16(b). Intermediate nodes when target is node 11.

Referring to Fig. 16(a) and (b), when we shall try to optimize a different target such as node 11 instead of node 10, as a separate list of nodes such as in this case nodes 2, 5, 6 and 9 will be the nodes to calculate the asset value, because the paths from node 0 to node 11 are 0→2→5→9→11 and 0→2→6→9→11. Thus, if the target node is changed, then the asset value is also changed. For example, if the target is 11, then we do not need to consider nodes 7 or 8, as they don't fall on the paths to reach to target node 11 and they will have 0 asset value this time.

4.2.2 Distance and Reachability in Vulnerability Graph

One of the crucial factors to be considered for node ranking is how far is the attack launch node from the target node. This gives an idea about the reachability of the attacker to the target node.

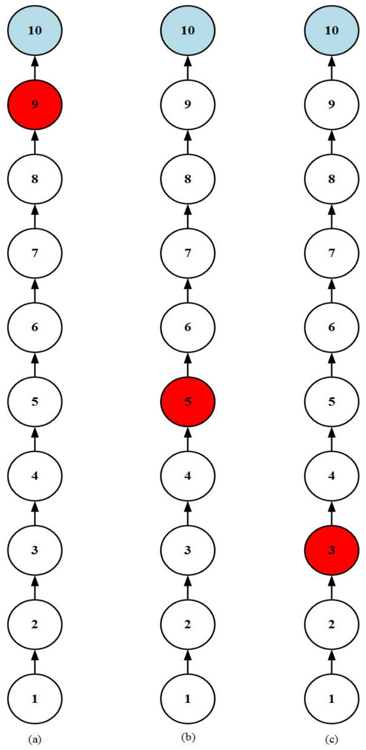


Fig. 17(a). Node Distance (Target Node is 10, attacker's current position in node 9).

Fig. 17(b). Node Distance (Target Node is 10, attacker's current position in node 5).

Fig. 17(c). Node Distance (Target Node is 10, attacker's current position in node 3).

For example, as in the above figure, if there are 10 nodes from 1 to 10 connected in direct hierarchy such as 1→2→3...→10, then if the attacker is now in node 9 as in Fig. 17(a), he is just 1 distance away from the target, so the attacker can make greater damage than if he is now in node 5 as in Fig. 17(b) which is having a distance of 5 yet to reach to the target or if the attacker is now in node 3 as in Fig. 17(c) which is having a distance of 7 yet to reach to the target. Which means, the attacker needs to put forth more effort and exploit more intermediate nodes vulnerabilities to reach to a certain target depending on his current position. In graph theory, eccentricity is defined as the maximum graph distance between two nodes. We need the shortest distance between two nodes. Distance D_i is equal to the number of intermediate nodes in the shortest path to pass through to reach the target. Distance is defined as below:

$$D_i = d(v_i, t) \tag{9}$$

where $d(v_i, t)$ is the shortest path length between node v_i and target t. Relative Closeness is defined by the following equation (10):

$$d_i = \frac{1}{D_i} \tag{10}$$

This means the greater the distance of the attacker's current position from the target node, the smaller is the relative closeness value for that node which is an indication about the capability of the attacker to cause damage or exploit the target node.

4.2.3 Degree Centrality in Vulnerability Graph

The degree centrality of a node v is the fraction of nodes connected to it. It usually refers to the number of links incident upon a node or the number of ties the node has in the graph. For a given graph $G := (n, e)$ where $|n|\ and\ |e|$ denotes nodes and edges respectively, the degree centrality of a node v_i in the graph G is defined as:

$$C_d(i) = \deg(v_i) \tag{11}$$

We are considering the degree centrality to consider the reachability through the nodes.

4.2.4 Exploit Factor (EF) and Impact Factor (IF) of edges in Vulnerability Graph

Each Vulnerability is associated with CVSS base metrics which are Access vector (A_V), Access complexity (A_C), Access authentication (A_A), Confidentiality Impact (I_C), Integrity Impact (I_I) and Availability Impact (I_A). The inputs taken from NVD database to process the calculation of Resilience and Node Ranking Metric are given in TABLE 9 and TABLE 10.

TABLE 9

CVSS EXPLOITABILITY BASE METRICS [2]

Base Metrics Category	Value	Score
Access vector (A_V)	Local (L)	0.395
	Adjacent Network (A)	0.646
	Remote Network (N)	1.0
Access complexity (A_C)	High (H)	0.35
	Medium (M)	0.61
	Low (L)	0.71
Access authentication (A_A)	Multiple (M)	0.45
	Single (S)	0.56
	None (N)	0.704

TABLE 10

CVSS IMPACT BASE METRICS [2]

Base Metrics Category	Value	Score
Confidentiality Impact (I_C)	None (N)	0.0
	Partial (P)	0.275
	Complete (C)	0.660
Integrity Impact (I_I)	None (N)	0.0
	Partial (P)	0.275
	Complete (C)	0.660
Availability Impact (I_A)	None (N)	0.0
	Partial (P)	0.275
	Complete (C)	0.660

Each vulnerability represents an edge in the graph which has the weights that represents the exploitability and impact as calculated from the CVSS. Edge exploitability and impacts are the most important quantitative parameters that we have in quantification of resilience of a network or node, so we have taken into consideration both the exploitability and impact metrics of the vulnerabilities associated with the nodes while ranking them. In TABLE 9 and TABLE 10, we have represented the base exploitability and impact metrics as given in CVSS. The exploitability and impact have been calculated in CVSS by the below equations [2]:

$$\begin{gathered} Exploitability\ Score, ES = 20 \times Access\ Vector \\ \times Access\ Complexity \times Access\ Authentication \end{gathered} \tag{12}$$

or,

$$ES = 20 \times A_V \times A_C \times A_A \tag{13}$$

and,

$$\begin{gathered} Impact\ Score, IS = 10.41 \times \\ \Big(1 - \big((1 - ConfImpact) \times (1 - IntegImpact) \times (1 - AvailImpact)\big)\Big) \end{gathered} \tag{14}$$

or,

$$IS = 10.41 \times \big(1 - (1 - I_C) \times (1 - I_I) \times (1 - I_A)\big) \tag{15}$$

Both exploitability score and impact score are in the scale of 0 to 10. To use weighted average method, we have scaled down the ES and IS to the scale 0 ~ 1 and called them exploitability factor (EF) and impact factor (IF).

$$EF = \frac{ES}{10} \tag{16}$$

and,

$$IF = \frac{IS}{10} \tag{17}$$

The question here is that not all the vulnerabilities have the same exploitability and same impact on the network. Some of the vulnerabilities are highly critical with a CVSS base score of 7.0 to 10.0 while others are moderately critical with a base score of 4.0 to 6.9 and others that are below 4.0 fall in the category of low severity. Thus, treating each of them on the same scale would give us a non-appropriate result. That is why we have introduced a weighted average method to weight each vulnerability based on its exploitability and impact score separately. The highly severe vulnerabilities have a weight of 0.5, while moderately severe vulnerabilities have been given a weight of 0.3 and low severity vulnerabilities are given a weight of 0.2. The summation of these three category weights are equal to 1.0 as needed. TABLE 11 illustrates the weighted ranges for exploit factor and TABLE 12 illustrates the weighted range values for impact factor.

TABLE 11

EXPLOITABILITY FACTOR WEIGHT

EF	Severity	Weight
0.7 ~ 1.0	High	0.5
0.4 ~ 0.69	Medium	0.3
0.0 ~ 0.39	Low	0.2

TABLE 12

IMPACT FACTOR WEIGHT

IF	Severity	Weight
0.7 ~ 1.0	High	0.5
0.4 ~ 0.69	Medium	0.3
0.0 ~ 0.39	Low	0.2

The most interesting part of working with vulnerabilities is that a vulnerability may have a high exploit factor, but it may have a very low impact factor or vice versa. That is why using the above weighted approach, we have come up with the following two formulae for considering the exploitability and impact of the vulnerabilities along with their severity level.

As the node v_i, has vulnerabilities from 1 to n, so EF_i and IF_i are an array of exploitability factor and impact factor for all the vulnerabilities of node v_i and they are being defined as:

$$EF_i = \left[EF_{i_1}, EF_{i_2}, EF_{i_3}, \dots \dots, EF_{i_n}\right]$$

$$IF_i = \left[IF_{i_1}, IF_{i_2}, IF_{i_3}, \dots \dots, IF_{i_n}\right]$$

Weighted Exploitability Factor for node v_i having n vulnerabilities is defined by:

$$EF_{w_i} = \frac{\sum_{j=1}^{n} \left(w_{i_j} \times EF_{i_j}\right)}{\sum_{j=1}^{n} w_{i_j}} \tag{18}$$

Weighted Impact Factor for node v_i having n vulnerabilities is defined by:

$$IF_{w_i} = \frac{\sum_{k=1}^{n} \left(w_{i_k} \times IF_{i_k}\right)}{\sum_{k=1}^{n} w_{i_k}} \tag{19}$$

Both the weighted exploitability factor EF_{w_i} and weighted impact factor IF_{w_i} have a value from 0 to 1.

4.2.5 Physical Impact Factor of nodes in Vulnerability Graph

In section 4.2.3, we have considered the impact factor based on the software vulnerability, but the same vulnerability may cause a different impact when it belongs to Corporate DMZ layer than when it belongs to the control system layer which controls ICS

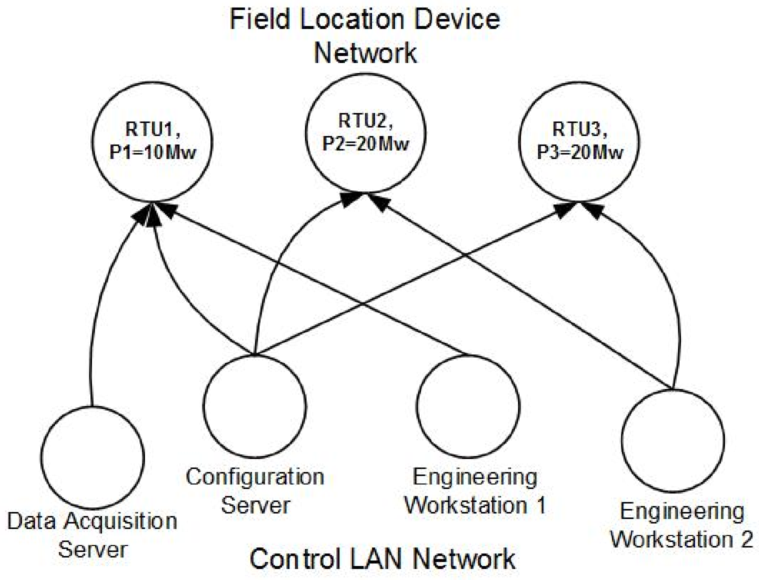


Fig. 18. Sample Network Connectivity Between Control System Lan Network and Field Device Network.

(Industrial Control System) or SCADA network for electrical power generation and distribution system network. Mostly in the power system domain, the physical loss is quantified by the amount of power outage that can be caused by the attacker by exploiting a vulnerability. That is why it is important to consider the physical impact of the vulnerability.

In this part, by connectivity we mean logical connections, an example of which can be any application or software already installed in the Configuration Server to operate the RTU's remotely or in the Data Acquisition Server to get real-time data from the RTU's to monitor its performance. Based on the logical connections we assumed in Fig. 18, Data acquisition server has logical connections with RTU1 and RTU1 is controlling 10 MW power distribution unit. Configuration Server is logically connected to all three RTU's, Engineering Workstation 1 is logically connected to RTU1 only and Engineering Workstation 2 has a logical connection or access to RTU2 and RTU3. RTU2 and RTU3 are controlling 20 MW power distribution unit each. If RTU1 is compromised, there is the chance of losing 10 MW power. Similarly, if RTU2 and RTU3 are compromised, then 20 MW of power is possible to be made unavailable by the attacker. Thus, each RTU has a fraction of power loss chance. If RTU1 is compromised then 10/(20+20+10) or 20% of the total power is lost. Now, if Data Acquisition server is being compromised, it has the potential to impact 20% of the total power loss by compromising RTU1. Similarly, if Engineering workstation 2 is comprised, it has the potential to impact (20+20)/(10+20+10) or 80% of the total power loss. Thus, a network element in the Control system LAN network can have a physical power loss ranging from 0 (no power outage) to 1 (maximum power outage), so, the expected fractional power loss can be defined as below.

$$Expected\ Power\ Loss\ Factor, EPLF_i = \frac{\sum_{m=1}^{u} B_{im} P_m}{\sum_{m=1}^{U} P_m} \tag{20}$$

Here, B_{im} is a binary quantity (1,0) which means whether node v_i is having an application or software installed that can access the RTU m i.e., whether node v_i has some logical connections to reach RTU m or not. If the application is in place already, there are logical connections between node v_i and RTU m and thus, $B_{im} = 1$, otherwise $B_{im} = 0$. Here,

$$\sum_{m=1}^{U} P_m = Total\ Power\ Delivery\ Capacity\ of\ Substation$$

$$\sum_{m=1}^{u} P_m = Total\ Power\ Controlled\ by\ all\ the\ RTU$$

$$which\ can\ be\ accessible\ from\ node\ m$$

$U = Total\ Number\ of\ RTUs\ available\ in\ the\ physical\ power\ system\ network$ and, $u = Total\ Number\ of\ RTUs\ having\ logical\ connections\ with\ node\ v_i$.

Mostly in the power system domain, people are concerned about the availability impact and integrity impact rather than the confidentiality impact, because the physical layer devices such as PLC or RTU are hard coded and their information exposure is not treated with as much importance as the availability and integrity are treated. Because if the attacker can modify the existing PLC code, he can easily try to throw the system out of normal operation and perform unexpected operations, as we saw in the Stuxnet attack on the Iranian Nuclear Power plant, where the attacker was able to modify the code and destroy the centrifuges. Here, Impact factor using the integrity impact and availability impact is modified for the case of physical power loss calculation as below:

$$IF_{PL} = 1.1307 \times \left(1 - (1 - I_I) \times (1 - I_A)\right) \tag{21}$$

The factor 1.1307 comes from the calculation to make IF_{PL} equal to 1.0. The maximum value of I_I *and* I_A is 0.660. By putting those values on the right-hand side of the equation and considering $IF_{PL} = 1$ on the left-hand side, the constant value comes to 1.1307, so considering the physical power loss using the same weighted approach as before as in TABLE 11, we can compute the weighted average power loss impact factor for node v_i as below:

$$IF_{PL_{W_i}} = EPLF_i \times \frac{\sum_{l=1}^{n} w_{i_l} \times IF_{PL_{i_l}}}{\sum_{l=1}^{n} w_{i_l}} \tag{22}$$

The full form of $IF_{PL_{W_i}}$ is found in equation (23) as below.

$$IF_{PL_{W_i}} = \frac{\sum_{m=1}^{u} B_{im} P_m}{\sum_{m=1}^{U} P_m} \times \frac{\sum_{l=1}^{n} w_{i_l} \times IF_{PL_{i_l}}}{\sum_{l=1}^{n} w_{i_l}} \tag{23}$$

4.2.6 MVNRank Algorithm Formula

In the node ranking, we have considered the above properties which are asset value, relative closeness and the weighted exploitability factor, weighted general impact factor, weighted physical impact factor and the total number of vulnerabilities and the degree centrality of the node. Node Ranking Value of v_i is found by the below equation:

$$R_i = A_{v_i} \times d_i \times C_d(i) \times N_i \times EF_{w_i} \times \left(IF_{w_i} + IF_{PL_{w_i}}\right) \tag{24}$$

The extended form of node ranking equation is given in equation (25) as below.

$$\begin{aligned} R_i = A_{v_i} \times d_i \times C_d(i) \times N_i \times \frac{\sum_{j=1}^{N_i} w_{i_j} \times EF_{i_j}}{\sum_{j=1}^{N_i} w_{i_j}} \\ \times \left(\frac{\sum_{k=1}^{N_i} w_{i_k} \times IF_{i_k}}{\sum_{k=1}^{N_i} w_{i_k}} + \frac{\sum_{m=1}^{u} B_{im} P_m}{\sum_{m=1}^{U} P_m} \times \frac{\sum_{l=1}^{N_i} w_{i_l} \times IF_{PL_{i_l}}}{\sum_{l=1}^{N_i} w_{i_l}} \right) \end{aligned} \tag{25}$$

Here, $C_d(i)$ is the degree centrality of node v_i and N_i is the number of vulnerabilities of node v_i. For the IT domain network except for the control system LAN, the nodes would not have the second impact factor component in equation 25, because they don't have any potential power loss which means they have a 0 value for the $IF_{PL_{w_i}}$. Only the Control System LAN network nodes have this physical impact factor, so those nodes will have non-zero $IF_{PL_{w_i}}$ which would give them some priority over other nodes. To normalize we need to find out the maximum value of the ranking value of all nodes, which is defined by:

$$Maximum\ Node\ Rank\ Value,\ R_{max} = \max_{\forall i | \exists v_i \in IN(v_t)} [R_i] \tag{26}$$

Relative criticality of each node is found by dividing the node's rank value by the maximum node rank value R_{max}. Thus, relative criticality of node v_i has been found by dividing the corresponding R_i by R_{max}. Thus,

$$RC_i = \frac{R_i}{R_{max}} \tag{27}$$

Here, $0 \leq RC_i \leq 1$. Based on the relative criticality, it is possible to form the Node Rank metric which is sorted from highest RC_i towards lowest RC_i based on sorting. For example, if we have 4 nodes A, B, C and D and after calculating the relative criticality value we have found that $RC_A = 1.00, RC_B = 0.45, RC_C = 0.69, RC_D = 0.75$, then the Node Ranking Metric (decision metric) is going to look like:

$$\begin{bmatrix} A & 1.0 \\ D & 0.75 \\ C & 0.69 \\ B & 0.45 \end{bmatrix}$$

Here rank of the nodes are chronological values where the top node is the highest rank node and so on downwards.

4.3 Flow Diagram of Ranking Metric Implementation

Fig. 19 shows the flow diagram of the ranking metric which gives an overview of how the ranking metric is used for resilience improvement of a particular network element or the network itself.

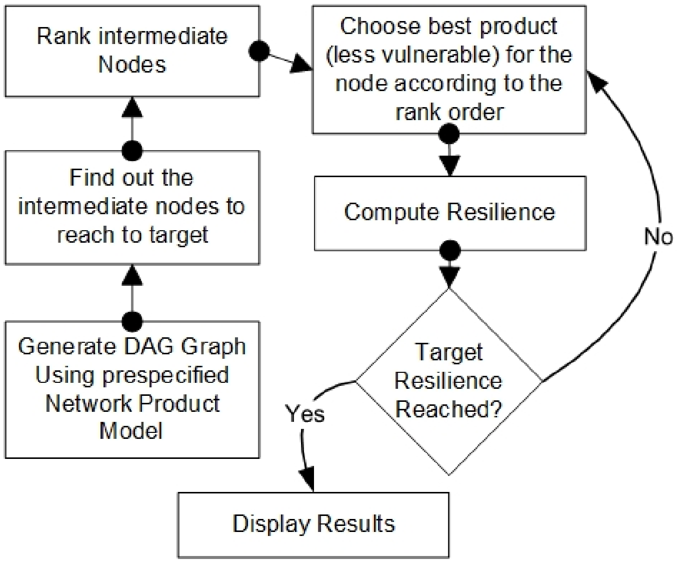
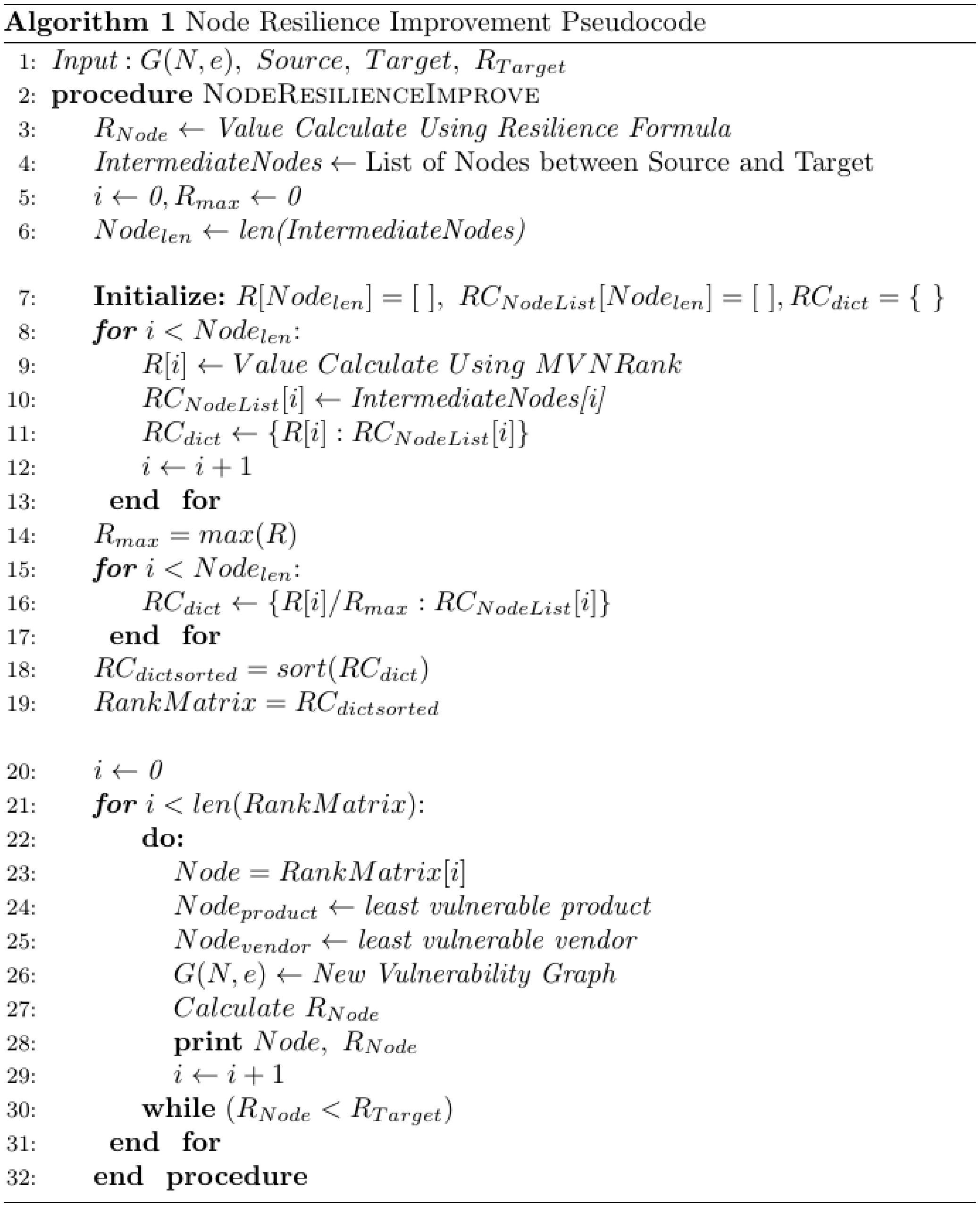

Rank intermediate Nodes
Choose best product (less vulnerable) for the node according to the rank order
Find out the intermediate nodes to reach to target
Compute Resilience
No
Generate DAG Graph Using prespecified Network Product Model
Target Resilience Reached?
Yes
Display Results

Fig. 19. Flow Diagram for Resilience Improvement Using Ranking Metric.

As illustrated, the intermediate nodes are ranked and according to the order of the node the best product with either possible zero vulnerability or less vulnerability than the previous product is used to simulate and compare resilience value. The process repeats until the target resilience is reached.

4.4 MVNRank Algorithm Implementation Pseudocode

The ranking algorithm can be used for individual node resilience improvement and network resilience improvement. To improve the network resilience, it is necessary to provide the list of the target nodes and then it loops the node resilience algorithm with the number of target nodes that are given as input as the target node. The ranking algorithm pseudocode is given in Fig. 20 and Fig. 21.



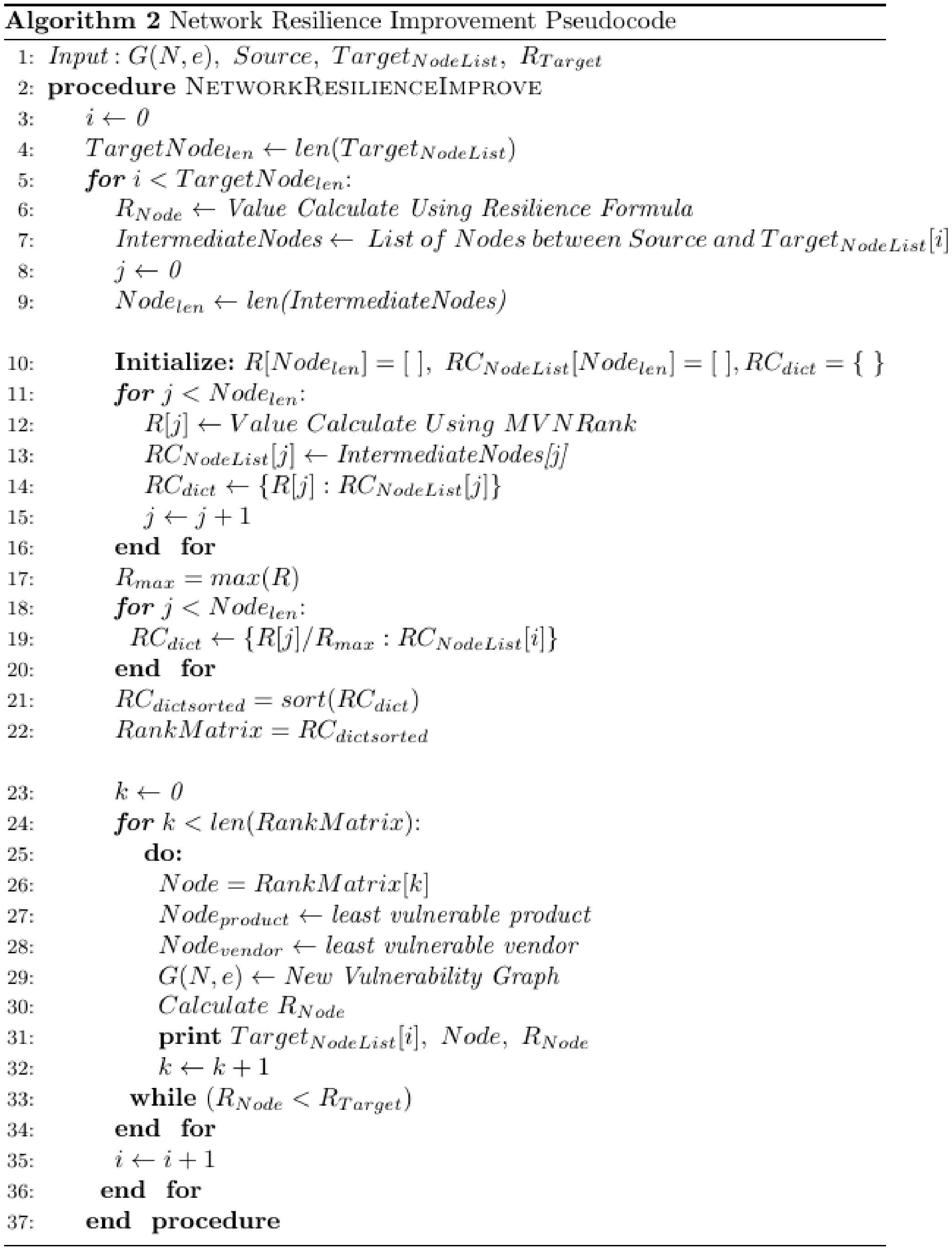
Algorithm 1 Node Resilience Improvement Pseudocode

```
$Input : G(N, e),\ Source,\ Target,\ R_{Target}$
procedure NodeResilienceImprove
    $R_{Node} \leftarrow$ Value Calculate Using Resilience Formula
    $IntermediateNodes \leftarrow$ List of Nodes between Source and Target
    $i \leftarrow 0, R_{max} \leftarrow 0$
    $Node_{len} \leftarrow len(IntermediateNodes)$

    Initialize: $R[Node_{len}] = [\ ],\ RC_{NodeList}[Node_{len}] = [\ ], RC_{dict} = \{\ \}$
    for $i < Node_{len}$:
        $R[i] \leftarrow Value\ Calculate\ Using\ MVNRank$
        $RC_{NodeList}[i] \leftarrow IntermediateNodes[i]$
        $RC_{dict} \leftarrow \{R[i] : RC_{NodeList}[i]\}$
        $i \leftarrow i + 1$
    end for
    $R_{max} = max(R)$
    for $i < Node_{len}$:
        $RC_{dict} \leftarrow \{R[i]/R_{max} : RC_{NodeList}[i]\}$
    end for
    $RC_{dictsorted} = sort(RC_{dict})$
    $RankMatrix = RC_{dictsorted}$

    $i \leftarrow 0$
    for $i < len(RankMatrix)$:
        do:
            $Node = RankMatrix[i]$
            $Node_{product} \leftarrow$ least vulnerable product
            $Node_{vendor} \leftarrow$ least vulnerable vendor
            $G(N, e) \leftarrow$ New Vulnerability Graph
            Calculate $R_{Node}$
            print $Node,\ R_{Node}$
            $i \leftarrow i + 1$
        while $(R_{Node} < R_{Target})$
    end for
end procedure
```

Fig. 20. Node Resilience Improvement Pseudocode.



Algorithm 2 Network Resilience Improvement Pseudocode

```
$Input: G(N, e),\ Source,\ Target_{NodeList},\ R_{Target}$
procedure NetworkResilienceImprove
    $i \leftarrow 0$
    $TargetNode_{len} \leftarrow len(Target_{NodeList})$
    for $i < TargetNode_{len}$:
        $R_{Node} \leftarrow$ Value Calculate Using Resilience Formula
        $IntermediateNodes \leftarrow$ List of Nodes between Source and $Target_{NodeList}[i]$
        $j \leftarrow 0$
        $Node_{len} \leftarrow len(IntermediateNodes)$

        Initialize: $R[Node_{len}] = [\,],\ RC_{NodeList}[Node_{len}] = [\,], RC_{dict} = \{\ \}$
        for $j < Node_{len}$:
            $R[j] \leftarrow$ Value Calculate Using MVNRank
            $RC_{NodeList}[j] \leftarrow IntermediateNodes[j]$
            $RC_{dict} \leftarrow \{R[j] : RC_{NodeList}[j]\}$
            $j \leftarrow j + 1$
        end for
        $R_{max} = max(R)$
        for $j < Node_{len}$:
          $RC_{dict} \leftarrow \{R[j]/R_{max} : RC_{NodeList}[i]\}$
        end for
        $RC_{dictsorted} = sort(RC_{dict})$
        $RankMatrix = RC_{dictsorted}$

        $k \leftarrow 0$
        for $k < len(RankMatrix)$:
            do:
              $Node = RankMatrix[k]$
              $Node_{product} \leftarrow$ least vulnerable product
              $Node_{vendor} \leftarrow$ least vulnerable vendor
              $G(N, e) \leftarrow$ New Vulnerability Graph
              Calculate $R_{Node}$
              print $Target_{NodeList}[i],\ Node,\ R_{Node}$
              $k \leftarrow k + 1$
          while $(R_{Node} < R_{Target})$
        end for
        $i \leftarrow i + 1$
      end for
    end procedure
```

Fig. 21. Network Resilience Improvement Pseudocode.

4.5 Rational Behind Factors Selection

As this thesis is using a graph-based approach for resilience calculation and node ranking, some of the important properties of graph theory have been considered crucial factors in the ranking process. Then those properties have been found either useful or not useful based on their application towards the goal of this thesis. The properties that have been considered are the degree, in-degree, out-degree, closeness centrality, betweenness centrality and eigenvector centrality. Below is a short definition of all those properties from the graph-based theory followed by their application toward our algorithm generation.

A. Degree, In degree, Out Degree

The number of links—directed or undirected connecting a node to the graph is called the degree of the node; thus, degree denotes the number of links a node has with other nodes in the graph. When the graph is directed the out-degree of a node is equal to the number of outward-directed links, and the in-degree is equal to the number of inward-directed links. In other words, out-degree is the number of tails, and in-degree is the number of heads connected to a node. For directed graphs, the node degree is the sum of in-degree & out-degree. In the directed acyclic graph model, the in-degree directly depends on the number of vulnerabilities the node has, and the out-degree is the number of vulnerabilities of the nodes where the node is a direct predecessor of those nodes. To evaluate a node based on the number of vulnerabilities, the in-degree is more appropriate in the directed acyclic graph model.

B. Degree centrality

The degree centrality of a node v is the fraction of nodes connected to it. It usually refers to the number of links incident upon a node or the number of ties the node has in the graph [95].

C. Betweenness centrality

Betweenness centrality is a measure of a node within a graph which quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. The betweenness centrality metric helps us understand how important a node is while considering the overall network attack scenarios.

D. Closeness centrality

Closeness centrality of a node is the average length of the shortest path between the node and all other nodes in the graph [95]. Thus, the more central a node is, the closer it is to all other nodes. The closeness of a node is the distance to all other nodes in the graph or in the case that the graph is not connected to all other nodes in the connected component containing that node. Closeness centrality is normalized by the minimum distance possible. Higher values of closeness indicate higher centrality.

E. Eigenvector centrality

Eigenvector centrality is a measure of the influence of a node in a network. It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. Google's PageRank is a variant of the eigenvector centrality [96].

To find out which of the above metrics would contribute more in our goal to rank the nodes of a network based on their importance and thus to improve the resilience of the network, we have used a sample hypothetical network with the following data to get the vulnerabilities extracted from the NVD database. The data is given in TABLE 13.

TABLE 13

PRODUCT MODEL SPECIFICATION FOR GRAPH PROPERTIES RELATION

Node	Vendor	Product	Number of Vulnerabilities Considered
APPLICATION_SERVER	Microsoft	Windows Server 2003	45
AUTHENTICATION_SERVER	Microsoft	Windows Server 2003	45
BUSINESS_SERVER	Microsoft	Windows Server 2003	45
BUSINESS_WORKSTATION	Microsoft	Windows Server 2003	45
CONTROL_FW1_SERVER	Cisco	ASA5500	5
CONTROL_FW2_SERVER	Cisco	ASA5500	5
CORPORATE_FW1_SERVER	Cisco	ASA5500	5
CORPORATE_FW2_SERVER	Paloalto	PANOS 7	15
CS_DB_SERVER1	Microsoft	SQL_SERVER	11
DB_SERVER1	Microsoft	SQL_SERVER	11
EMAIL_SERVER1	Microsoft	Windows Server 2003	45
ENG_WORKSTATION1	Microsoft	Windows XP	151
FTP_SERVER1	Microsoft	Windows Server 2003	45
HISTORIAN_SERVER1	Microsoft	SQL_SERVER	11
HMI_COMPUTER1	Microsoft	Windows XP	151
SECURITY_SERVER1	Microsoft	SQL_SERVER	11
WEBAPPLICATION_SERVER1	Microsoft	Windows Server 2003	45
WEB_SERVER1	Microsoft	Windows Server 2003	45
WWW_SERVER1	Microsoft	Windows Server 2003	45

Based on the above data we constructed a network graph using the NetworkX module in python and compared the metrics presented earlier. Here the X axis is different nodes that have been given in column 1 of TABLE 13. From Fig. 22 and Fig. 23, it is seen that In-degree is almost equivalent to the number of vulnerabilities a node has and degree centrality is proportional to In-degree curve whereas eigenvector centrality, betweenness centrality, and closeness centrality are not directly proportional to the vulnerabilities that we are considering.

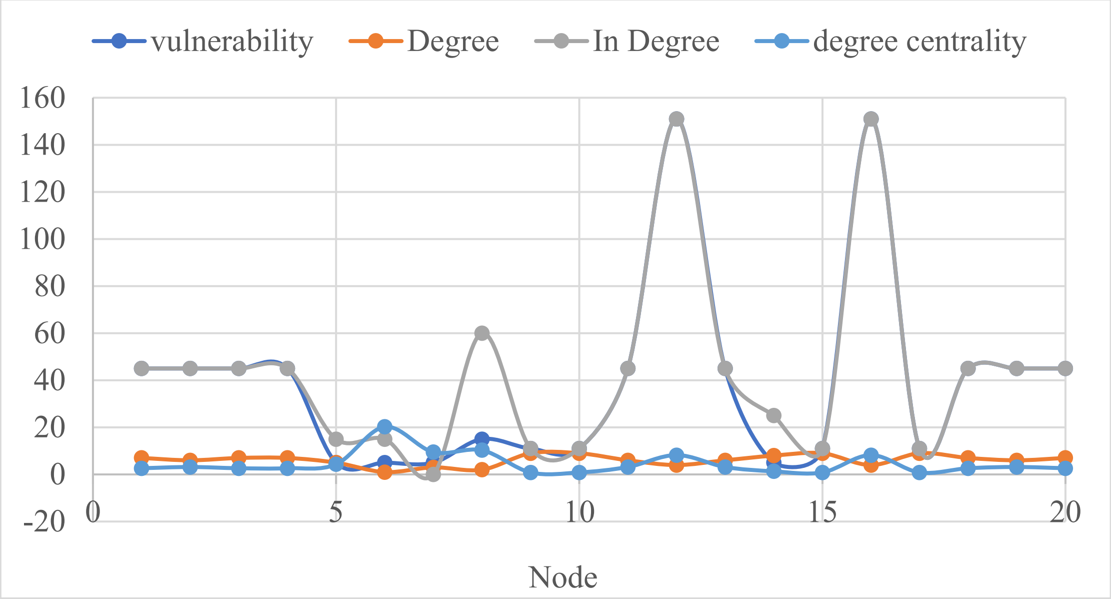


Fig. 22. Relation between Vulnerability Numbers, Degree, In Degree and Degree Centrality derived from the sample network DAG Model.

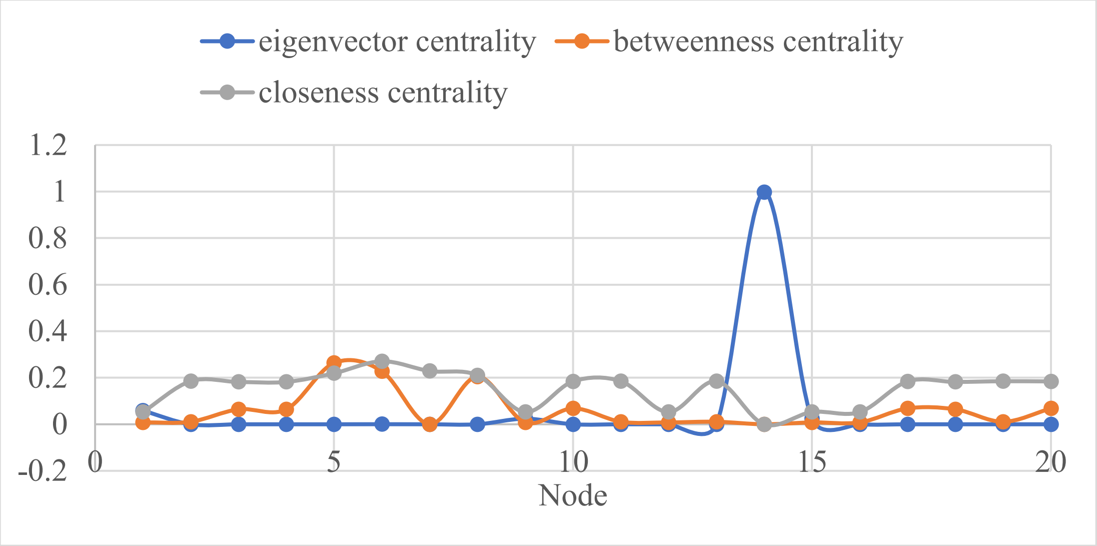


Fig. 23. Relation between Eigenvector Centrality, Betweenness Centrality and Closeness Centrality derived from the sample network DAG Model.

As I have given the complete mathematical expression for node ranking and relative criticality as in equations (25) and (26), I shall give some of the examples of using the software vulnerabilities and the rationale against choosing relative closeness as one of our parameters in determining the node criticality rankings in TABLE 14.

TABLE 14

RANKING VALUE WITH NODE DISTANCE

Product	Vuln No	R_i (D_i=1)	R_i (D_i=2)	R_i (D_i=3)	R_i (D_i=4)
Microsoft XP	152	130.7	-	-	-
Microsoft Vista	483	422.3	211.1	140.7	105.6
Cisco ASA5500	5	4.24	-	-	-
Paloalto PANOS 7	12	10.06	5.03	3.35	-

As we can see from TABLE 14, Microsoft XP and Microsoft Vista operating systems have 152 and 483 vulnerabilities from 2013-2017. If both are 1 distance away from the target node where we are interested in improving resilience, we should consider replacing the Microsoft Vista node with some other node with fewer vulnerabilities and impact as calculated by equation (7); thus, Microsoft vista gets preference in node ranking over Microsoft XP. The reason behind this is as Microsoft Vista has 483 vulnerabilities it can offer more channel vulnerability paths for the attacker to exploit this node than Microsoft XP does. If the same Microsoft Vista installed host is 4 distances away and the Microsoft XP installed host is still one distance away from the target node, then their relative values come to 105.56 and 130.68 respectively in this case, even if Microsoft Vista has more than 3 times the number of vulnerabilities than Microsoft XP, we should consider the Microsoft XP host because the node ranking value is greater considering both the number of vulnerabilities and the relative closeness. The same thing applies to our second example, where the CISCO ASA5500 product has 5 vulnerabilities and Paloalto PANOS has 12 vulnerabilities. If both of those FW's are 1 distance away from the target node, we would surely need to consider PALOALTO PANOS over CISCO ASA5500 because of its higher node ranking value than CISCO ASA5500, but if the PANOS is 3 distances away and ASA5500 is 1 distance away, then we should give priority to ASA5500 because its value is still 4.24 while the PANOS value drops to 3.35. This way we can make important decisions by considering the current node's distance from the target node.

TABLE 15

RANKING VALUE WITH CVSS EDGE WEIGHTS

Product	Vuln No	R_i (D_i=1)
Microsoft Windows Server 2003	45	37.06362
Microsoft Windows Server 2016	45	39.26116

In TABLE 15, we have two distinct products with a similar number of vulnerabilities from 2017-2013, they both have 45 vulnerabilities. Based on the criticality of the vulnerabilities, windows_server_2016 has a node ranking value 39.26 which is greater than windows_server_2003's node ranking value 37.06. That means we shall consider windows_server_2016 to be prioritized in node ranking over windows_server_2003 while replacing the nodes for improving the resilience.

CHAPTER 5

VERIFICATION AND VALIDATION

The analytical framework of the ranking algorithm has been verified using a small network setup. For this verification, we have used python NetworkX [97] module and a database that has been built using the XML files from NVD and MITRE. The NetworkX module integrated into python provides some means of graph calculations such as drawing of the graphs using Graphviz [98] or PyGraphviz [99] module called from NetworkX module. After the graph is constructed with the necessary nodes and edge parameters with weight values, it is possible to calculate the paths between nodes, shortest paths between nodes, degree parameters such as degree, degree centrality, closeness centrality and lots of other metrics, etc. For the simulation purpose, we have used multi-digraph [100] which facilities multiple parallel directed edges between nodes where each edge represents a vulnerability. For other calculations, customized procedures and functions have been defined to produce necessary outputs such as resilience calculation, node ranking value calculation, and graphs generation. All the output shown here are generated using python and custom-made functions. The following sections are going to discuss the network setup for the verification, different outputs to demonstrate the applicability of the node ranking algorithm based on the graph. Finally, the result has been compared with a previously published scholarly article to validate it against known results and check the ranking consistency.

5.1 Simulation Setup

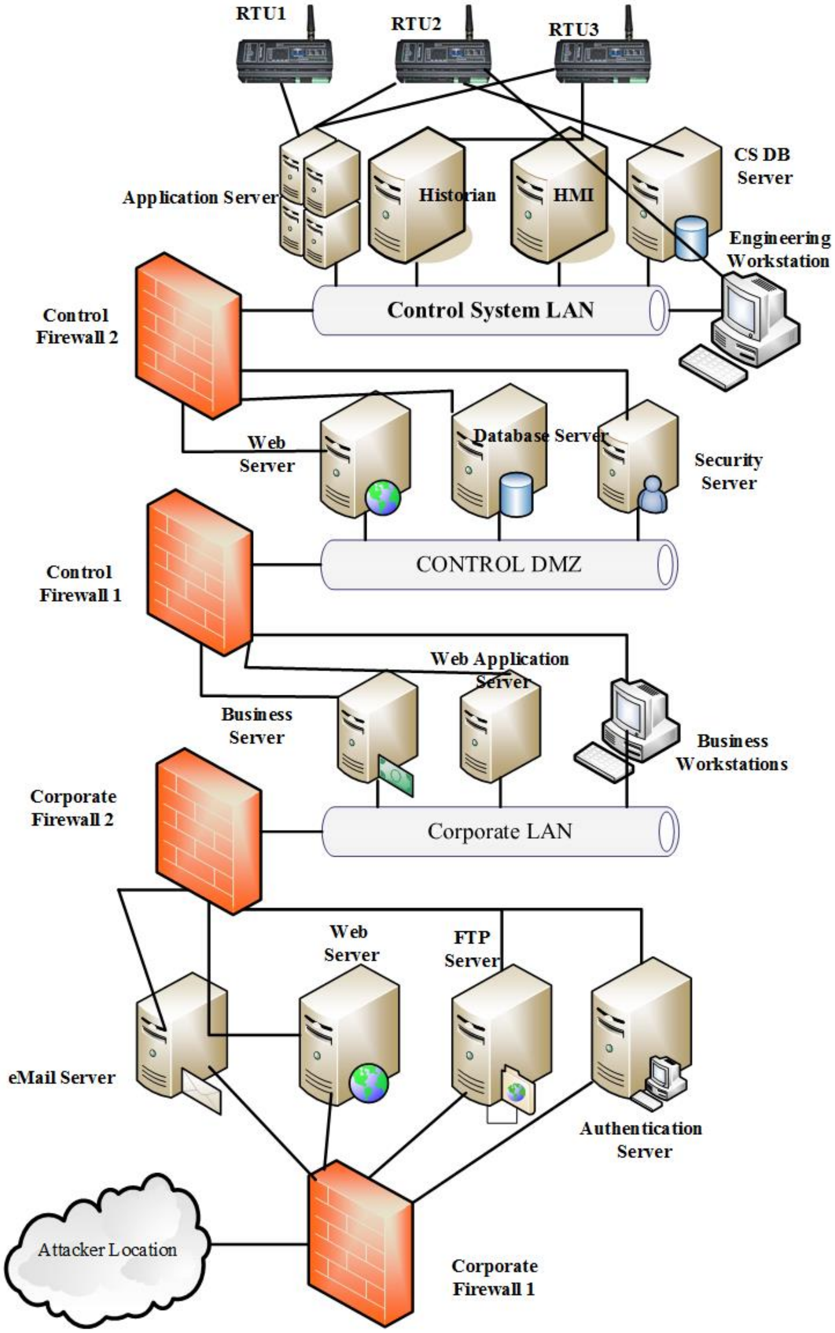


Fig. 24. Sample Network Considered for Simulation Purpose.

Fig. 24 shows the sample network considered for simulation of the node ranking algorithm. The sample network consists of the minimum number of levels as represented by the DAG model. Each layer corresponds to the NIST-CSSP defense-in-depth architecture. The nodes in the Corporate DMZ layer are eMail Server, Web Server, FTP Server and Authentication Server. The Corporate LAN layer consists of Business Server, Web Application Server, and Business Workstation. The Control DMZ consists of Web Server, Security Server, and Database Server. The Control System LAN layer consists of Application Server, Historian, HMI, CS DB Server and Engineering Workstation. For this simulation, each node has been considered only once but in general, there can be hundreds of similar nodes or other different nodes that the network may consist of.

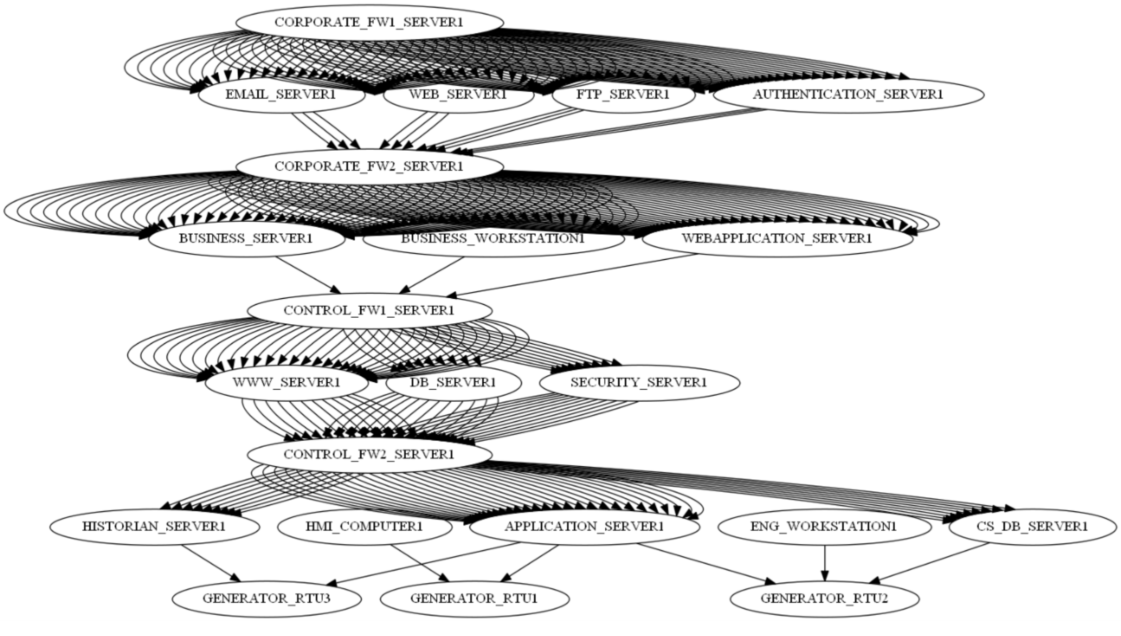


Fig. 25. Directed Acyclic Vulnerability Graph Generated from Sample Network Using NetworkX.

Fig. 25 shows the corresponding vulnerability graph which has been derived from the simulation in NetworkX with the specific product as in subsection. If a node has multiple vulnerabilities, then there are multiple directed edges towards that node, which means we have considered the possible number of vulnerabilities as reported in the NVD database corresponding to the vendor and product model. Each vulnerability constructs an edge in the vulnerability graph and each edge has the values of cost, impact, A_V, A_C, A_A, I_I, I_C and I_A as its weight.

5.2 Simulation Parameters

TABLE 17 shows the product vendor and model used for simulation purposes. TABLE 16 shows the simulation environment. Here, each product is associated with the last three years of vulnerabilities as defined in NVD database. Models for Generator Remote Terminal Units are not provided because they are thought to be nodes with power capacity only and not directly vulnerable if they can be isolated from the Corporate LAN and Corporate DMZ Layers.

TABLE 16

SIMULATION ENVIRONMENT SPECIFICATION

Environment	Specification
Model	DELL Latitude E5570
Processor	Intel Core i7 2.7GHz
Memory	16 GB (15.7 GB Usable)
Operating System	Windows 7 6d Bit

TABLE 17

SIMULATION MODEL PRODUCT AND VENDOR SPECIFICATION

Node	Vendor	Product	Number of Vulnerabilities
Corporate FW1	Cisco	ASA5500	1

Email Server1	Microsoft	Windows Server 2003	30
FTP Server1	Microsoft	Windows Server 2003	30
Web Server1	Microsoft	Windows Server 2003	30
Authentication Server1	Microsoft	Windows Server 2003	30
Corporate FW2	Juniper	SRX210	3
Business Server1	Microsoft	Windows Server 2016	45
Business Workstation1	Microsoft	Windows Server 2003	30
Web Application Server1	Microsoft	Windows Server 2016	45
Control FW1	Cisco	ASA5500	1
WWW Server1	Microsoft	Windows Server 2003	30
DB Server1	Microsoft	SQL Server	9
Security Server1	Microsoft	SQL Server	9
Control FW2	Paloalto	PANOS 7.1	12
Application Server1	Microsoft	Windows Server 2003	30
CS DB Server1	Microsoft	SQL Server	9
Eng Workstation1	Microsoft	Windows XP	0
Historian Server1	Microsoft	SQL Server	9
HMI Computer1	Microsoft	Windows XP	0

5.3 Simulation Outputs and Analysis

The simulation complexity generates when there are a lot of paths from a source node to the target node. The more multi-host multi-stage vulnerabilities there are, the greater the computation time required to calculate the path cost. Before going to the ranking demonstration, we are presenting here some of the path cost histogram to understand the network complexity in terms of computation. Path cost histograms of Security Server1, WWW Server1, CS_DB_Server1, and Application Server1 are given below.

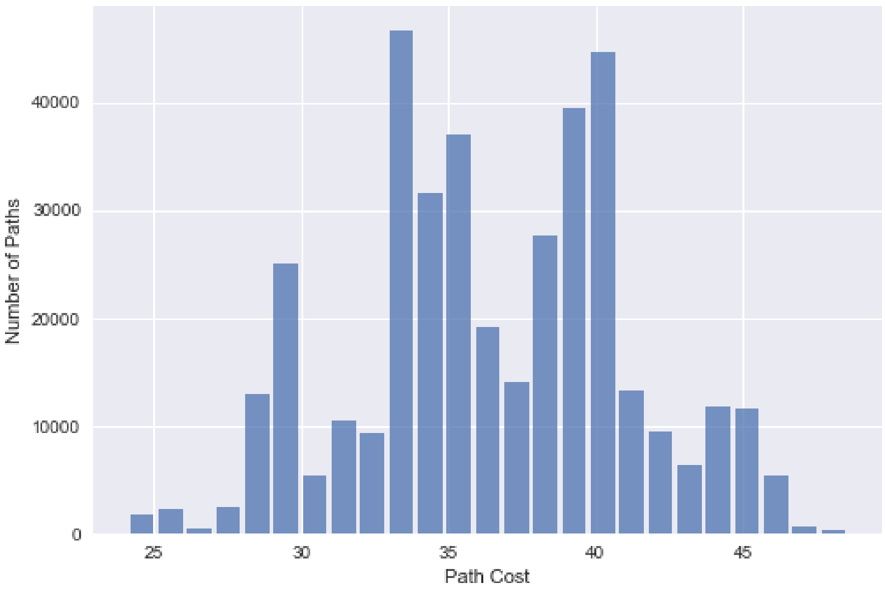


Fig. 26. Path Cost Histogram of Security Server 1.

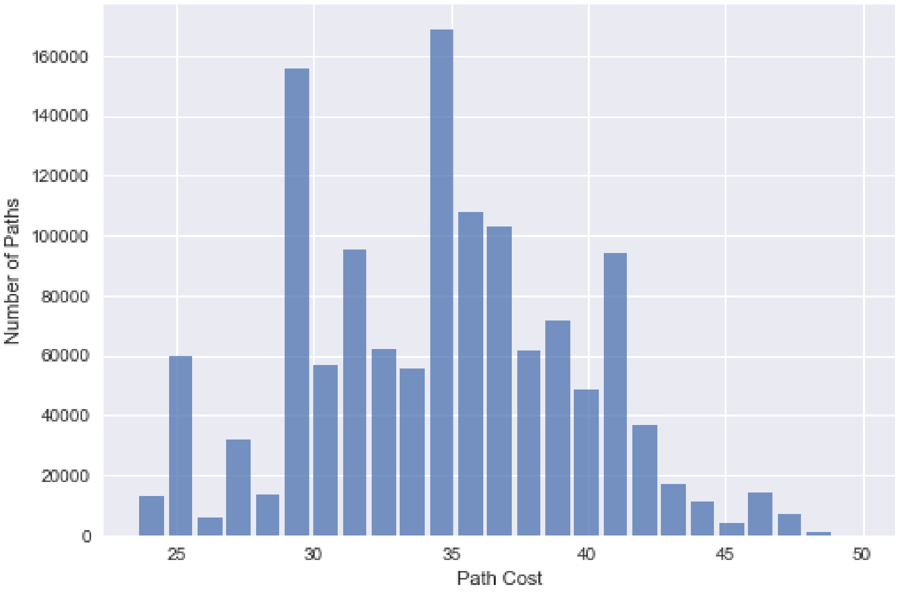


Fig. 27. Path Cost Histogram of WWW Server 1.

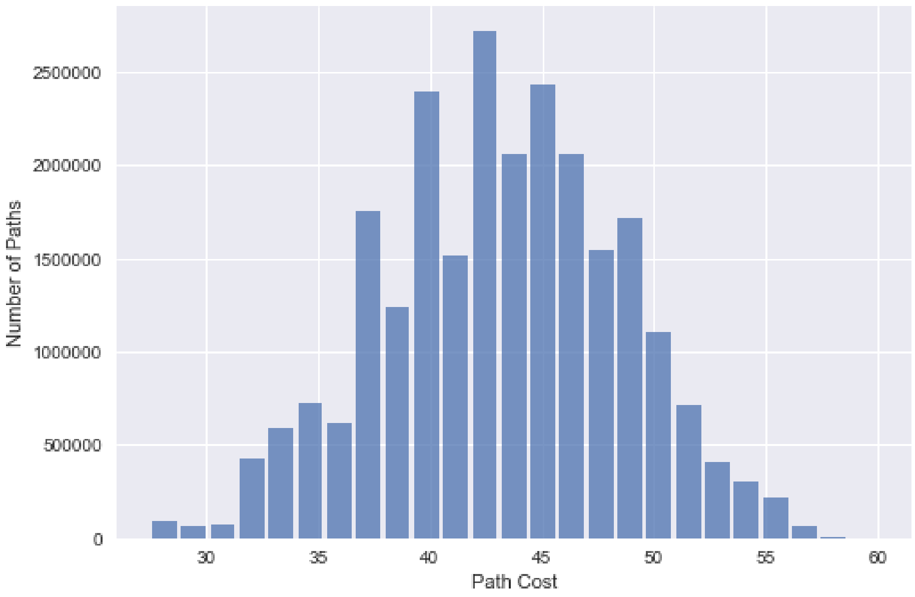


Fig. 28. Path Cost Histogram of CS DB Server 1.

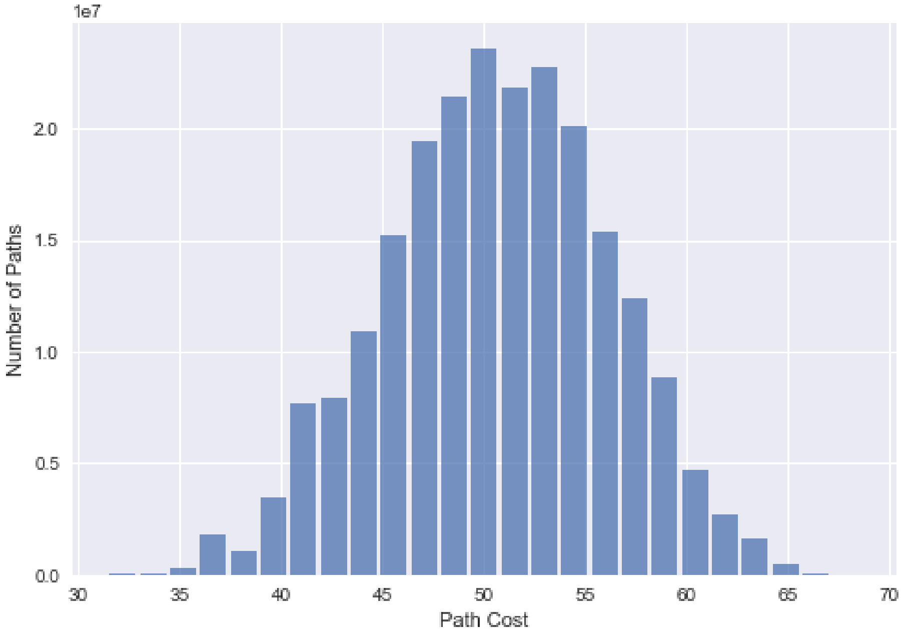


Fig. 29. Path Cost Histogram of Application Server 1.

Almost all the path cost histograms follow the normal distribution and their paths increases with the distances from the source. Between two nodes an edge can have a maximum cost metric of 10. The maximum cost 70 means there are 7 levels of nodes from the source to the target node. The degree distribution of the nodes for this simulation is given below.

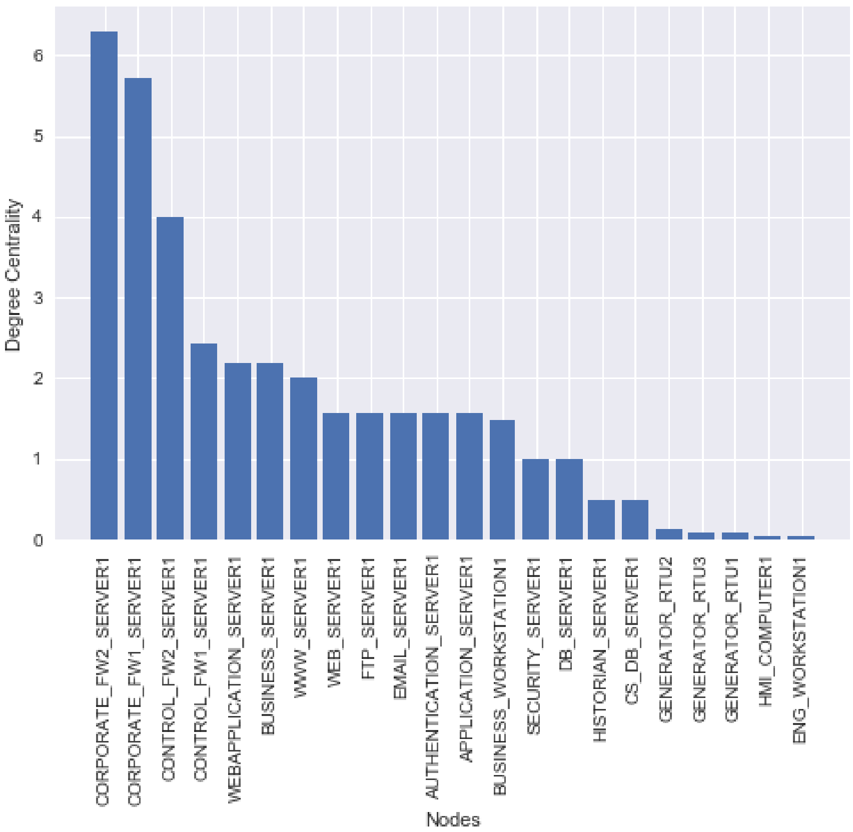
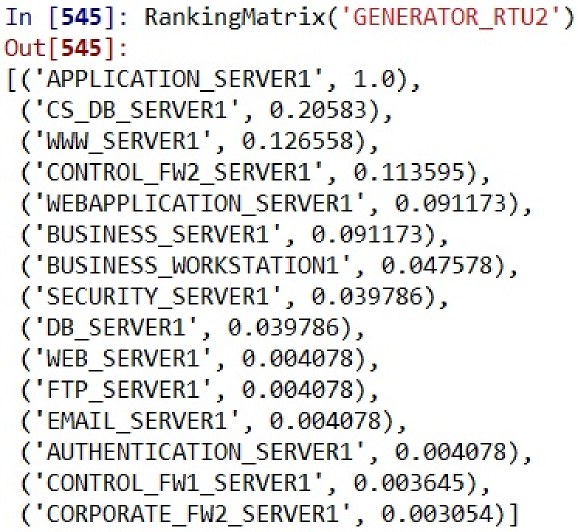


Fig. 30. Degree centrality of the nodes in simulation network.

There are two major objectives of the simulation that we are more concerned about:

1. Ranking algorithm demonstration
2. Demonstration of resilience improvement using the algorithm.

To demonstrate the ranking algorithm, two nodes have been chosen here, “Generator RTU2” and “Application Server1”. As stated before, the node ranking algorithm ranked the intermediate nodes that fall on the path to the target node, so for this case, “Generator RTU2” and “Application Server1” are two different targets for the demonstration of the node rank. Below is the output of the relative criticality value for the above two nodes as given in Fig. 31 and Fig. 33.



```
In [545]: RankingMatrix('GENERATOR_RTU2')
Out[545]:
[('APPLICATION_SERVER1', 1.0),
 ('CS_DB_SERVER1', 0.20583),
 ('WWW_SERVER1', 0.126558),
 ('CONTROL_FW2_SERVER1', 0.113595),
 ('WEBAPPLICATION_SERVER1', 0.091173),
 ('BUSINESS_SERVER1', 0.091173),
 ('BUSINESS_WORKSTATION1', 0.047578),
 ('SECURITY_SERVER1', 0.039786),
 ('DB_SERVER1', 0.039786),
 ('WEB_SERVER1', 0.004078),
 ('FTP_SERVER1', 0.004078),
 ('EMAIL_SERVER1', 0.004078),
 ('AUTHENTICATION_SERVER1', 0.004078),
 ('CONTROL_FW1_SERVER1', 0.003645),
 ('CORPORATE_FW2_SERVER1', 0.003054)]
```

Fig. 31. Python Output of Ranking Metric when Target Node is “GENERATOR RTU2”.

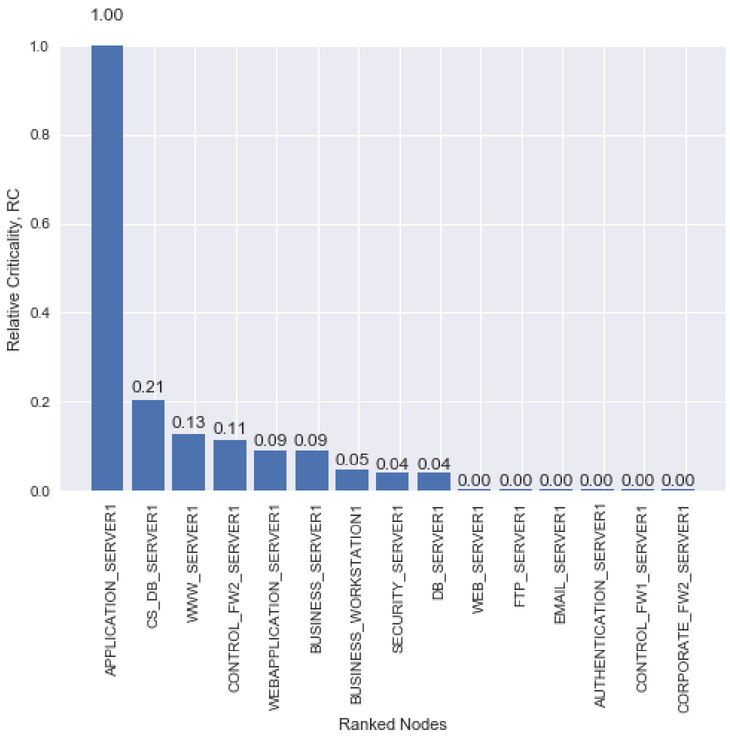
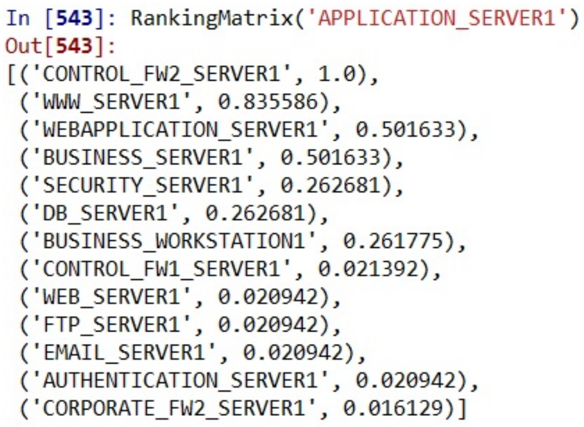


Fig. 32. Ranking Metric Output when Target Node is “GENERATOR RTU2”.

Fig. 32 demonstrates that “APPLICATION_SERVER1” has the rank order 1 when the target node is “Generator RTU2”. Similarly, “CS_DB_SERVER1” has rank order 2. This means when considering improvement of resilience for “Generator RTU2”, the network analyst should consider “APPLICATION_SERVER1” first and then “CS_DB_SERVER1” for patching or removing vulnerabilities.



```
In [543]: RankingMatrix('APPLICATION_SERVER1')
Out[543]:
[('CONTROL_FW2_SERVER1', 1.0),
 ('WWW_SERVER1', 0.835586),
 ('WEBAPPLICATION_SERVER1', 0.501633),
 ('BUSINESS_SERVER1', 0.501633),
 ('SECURITY_SERVER1', 0.262681),
 ('DB_SERVER1', 0.262681),
 ('BUSINESS_WORKSTATION1', 0.261775),
 ('CONTROL_FW1_SERVER1', 0.021392),
 ('WEB_SERVER1', 0.020942),
 ('FTP_SERVER1', 0.020942),
 ('EMAIL_SERVER1', 0.020942),
 ('AUTHENTICATION_SERVER1', 0.020942),
 ('CORPORATE_FW2_SERVER1', 0.016129)]
```

Fig. 33. Python Output of Ranking Metric when Target Node is "APPLICATION_SERVER1".

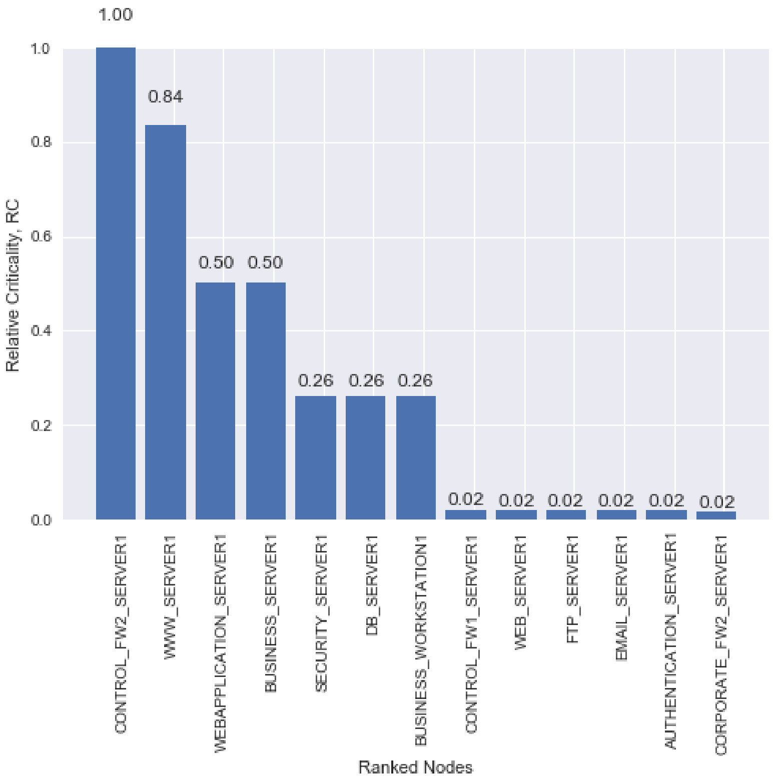


Fig. 34. Ranking Metric Output when Target Node is "APPLICATION_SERVER1".

Fig. 33 represents the relative criticality value when the target node is "APPLICATION_SERVER1". The same has been plotted in bar chart format in python in Fig. 34. Here, when the target node is "APPLICATION_SERVER1", the "CONTROL_FW2_SERVER1" has rank order 1 and "WWW_SERVER1" has the rank order 2, so these nodes need to be patched in order to improve the resilience of "Application Server1" node. The nodes that are ranked in two different target cases are different; thus, the algorithm is dynamic because it doesn't give the same rank for the same node all the time.

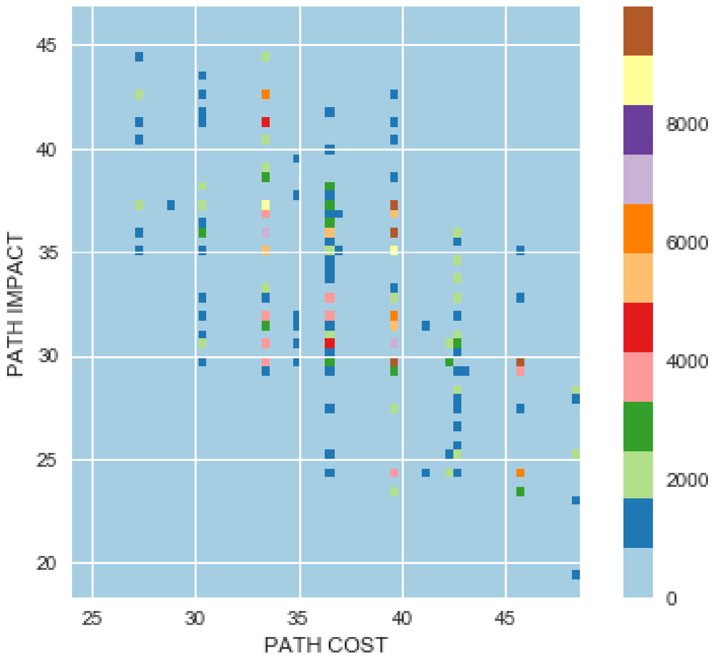


Fig. 35. Path cost Vs Path Impact Heatmap of Security Server.

Heatmap of channel vulnerability path cost vs path impacts is shown in Fig. 35. The same heatmap is shown for Control Firewall 2 in Fig. 36. The heatmaps reveal that most of the channel vulnerability paths are in the center with medium ranges of path costs and medium ranges of path impacts, so we can't say that lower path cost leads to high impacts or vice versa.

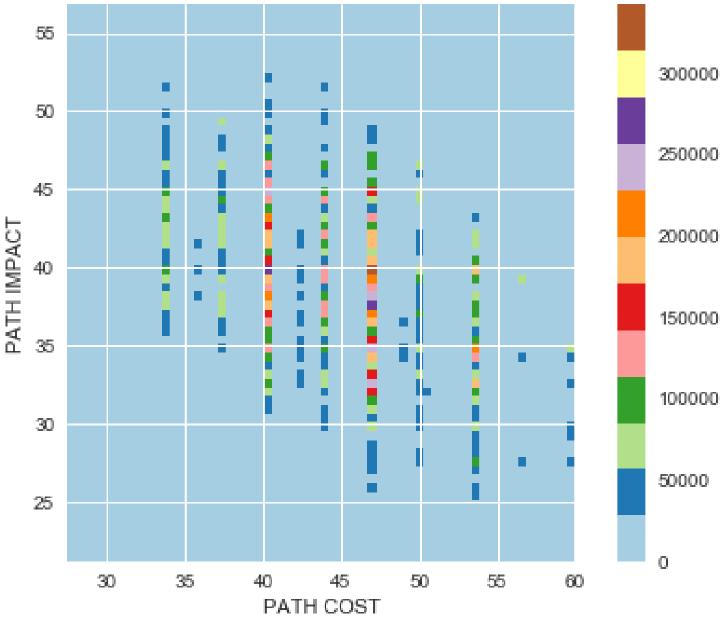


Fig. 36. Path cost Vs Path Impact Heatmap of Control Firewall 2.

One of the major concerns of the simulation of this type of network is that each node may have millions of incoming paths combinations from the original entry point as stated before. For example, in our simulation Email Server1 has only 30 incoming paths, but Security Server1 has 129600 incoming paths which is the combination of different exploitable vulnerabilities from the origin node as shown in Fig. 37. Again, Control Firewall2 has 8294400 (8.29M) paths and Application Server1 has 248832000 (248.8M) paths as in Fig. 38. The more total paths, the longer it takes to simulate the resilience of that node.

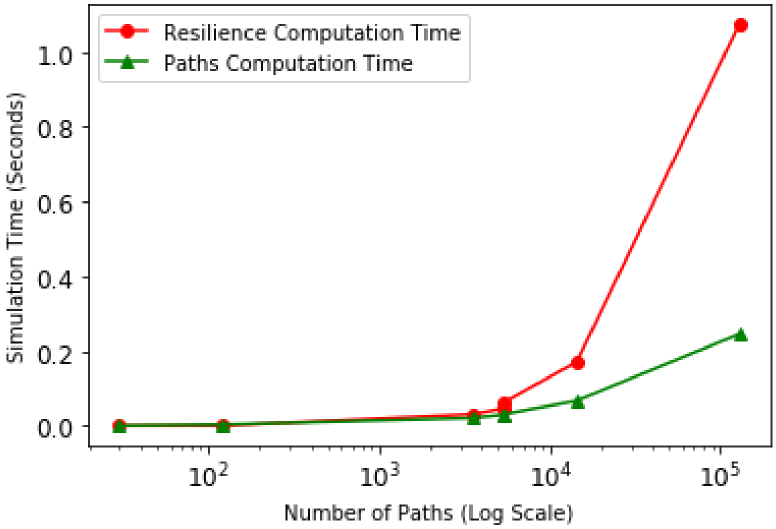


Fig. 37. Simulation Time (Maximum Paths Number ≤ 129600).

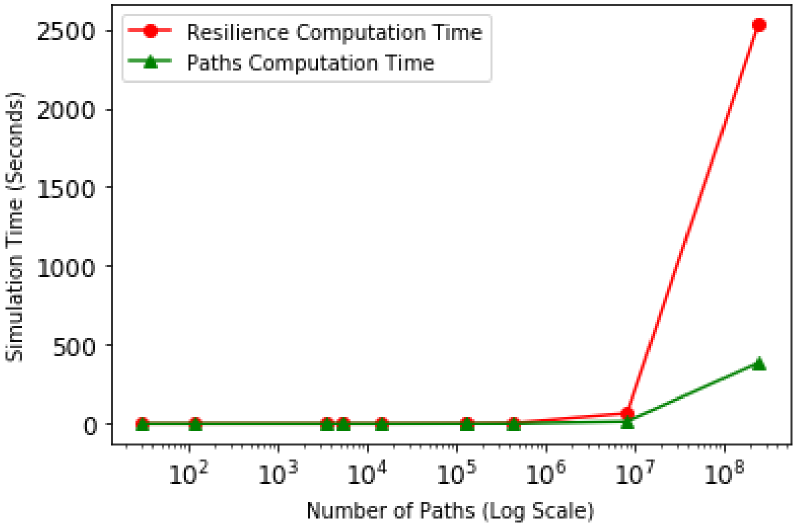


Fig. 38. Simulation Time (Maximum Paths Number ≤ 248832000).

In Fig. 37 and Fig. 38, we have found that the resilience computation time sharply increases when the number of paths increases. While Fig. 37 shows simulation time for resilience and paths computation up to 0.13M paths, Fig. 38, shows the simulation time for the same up to

248.8M paths. Application Server1 has 248.8M paths and it takes 2537.342 seconds (42.23 min) to compute the resilience and 387.924 seconds (6.47 min) to compute the total number of paths in the vulnerability graph.

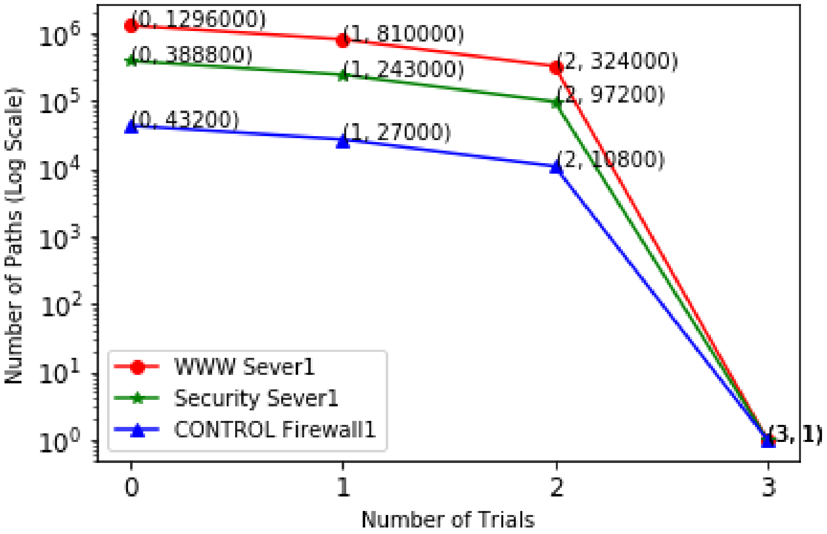


Fig. 39. Number of Exploitable Paths Reduction Over Simulation Trials for Three Selected Nodes (WWW_Server1, Security_Server1, Control Firewall1).

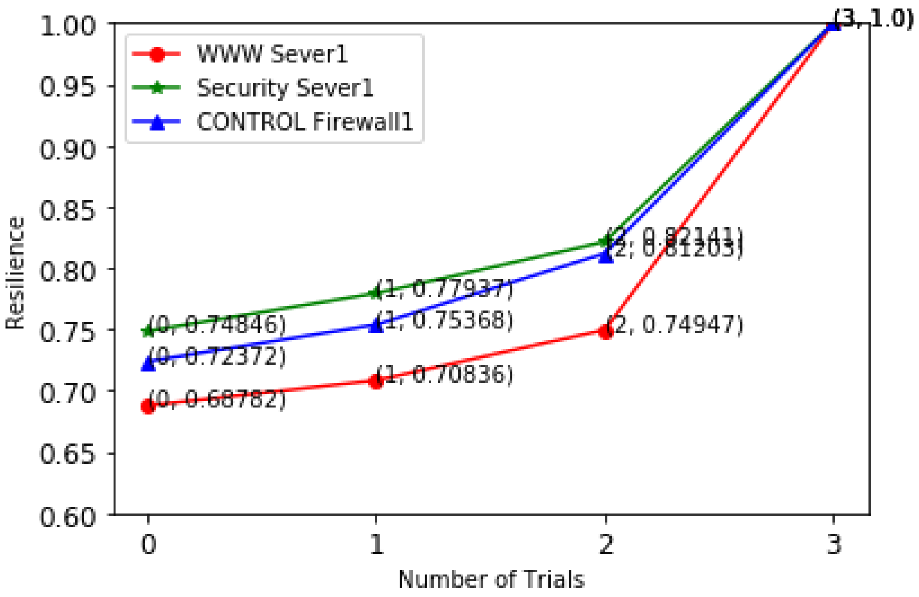


Fig. 40. Resilience Over Simulation Trials for Three Selected Nodes (WWW_Server1, Security_Server1, Control Firewall1).

For demonstration of the ranking algorithm, we have selected three nodes that reach resilience value 1.0 after 3 trials; that means after 3 trials no path exist from the source node towards the target node. Fig. 39 shows the path reductions over simulation trials for the three selected nodes and Fig. 40 shows the resilience value over simulation trials where nodes have been selected as per rank order. As after three trials no path exists from source to the target node, so the resilience reaches 1.0. For number of paths log scale has been used for proper demonstration and the actual value is shown in (x,y) co-ordinate format. After 3rd trials the value of paths become 0, but due to use of log scale we have taken it as 1.

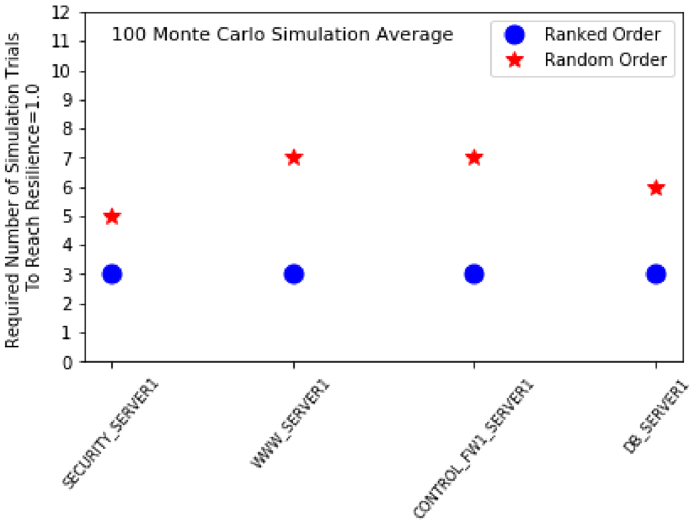


Fig. 41. Comparison of Number of Simulation Trials Needed to Reach Resilience Value 1.0 for Four Selected Nodes (Secrutiy_Server1, WWW_Server1, Control Firewall1, DB_Server1) using 100 Monte Carlo Simulation Average.

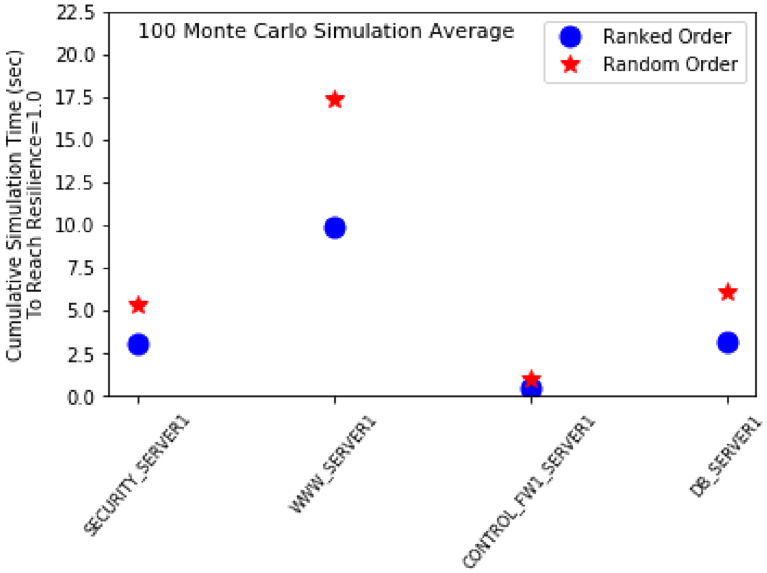


Fig. 42. Comparison of Simulation Time Required to Reach Resilience Value 1.0 for Four Selected Nodes (Secrutiy_Server1, WWW_Server1, Control Firewall1, DB_Server1) using 100 Monte Carlo Simulation Average.

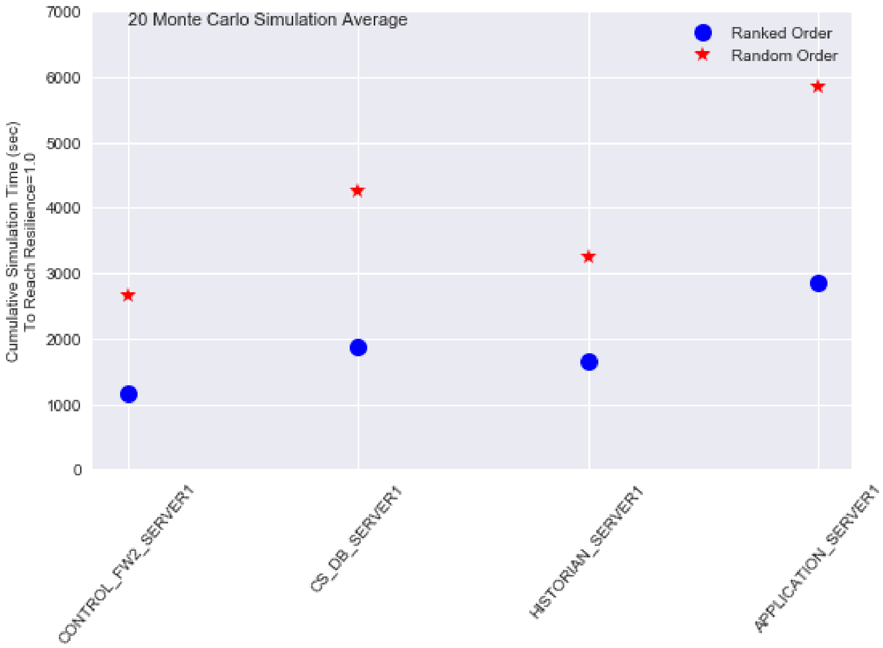


Fig. 43. Comparison of Simulation Time Required to Reach Resilience Value 1.0 using 20 Monte Carlo Simulation Average.

Fig. 41 shows a comparison of the required number of trials to reach a resilience value of 1.0 for the four selected nodes. Using the ranking order, we can reach to target resilience value 1.0 for those nodes in 3 trials but using random order it needs more simulation trials such as up to 7 in some cases. Similarly, Fig. 42 and Fig. 43 show two different comparisons of the total simulation time required to reach the resilience value of 1.0 for some nodes using ranking and random order. Simulation time using the ranking order is less in most of the cases as shown by the blue dots. For demonstration purposes, we have chosen nodes other than power station nodes because resilience computation time is much higher for power station nodes and needs high computing resources, but the results are equally applicable to power station layer nodes also.

With the previous simulation results, the improvement in simulation time is demonstrated, but we don’t know how much time is improved. To know that, we have done some comparison between the collected two sets of data and done some regression analysis as given below.

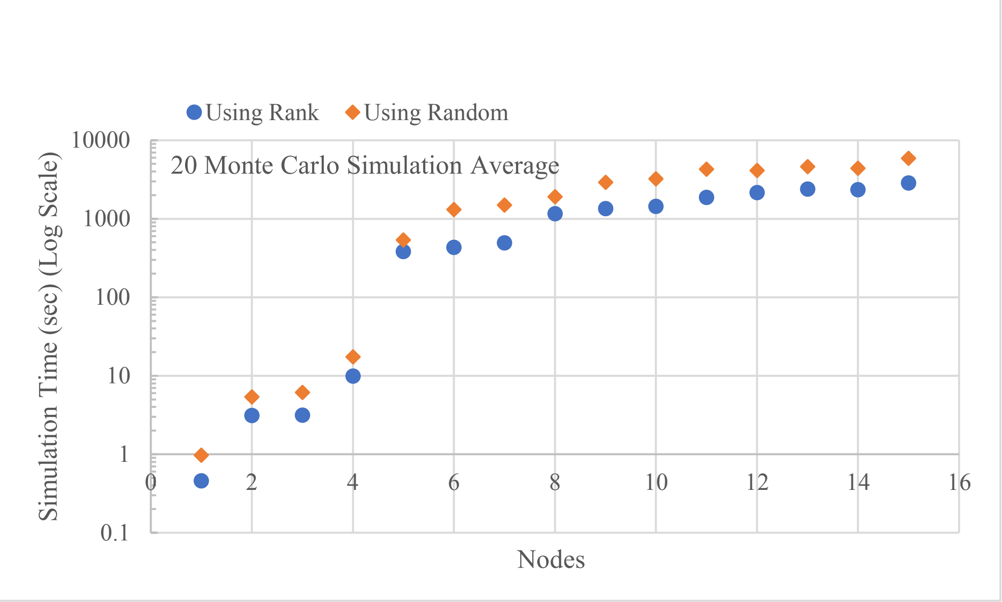


Fig. 44. Simulation time comparison using rank order and without rank order.

The regression analysis shows that almost 50% time can be reduced by using the ranking algorithm as given below.

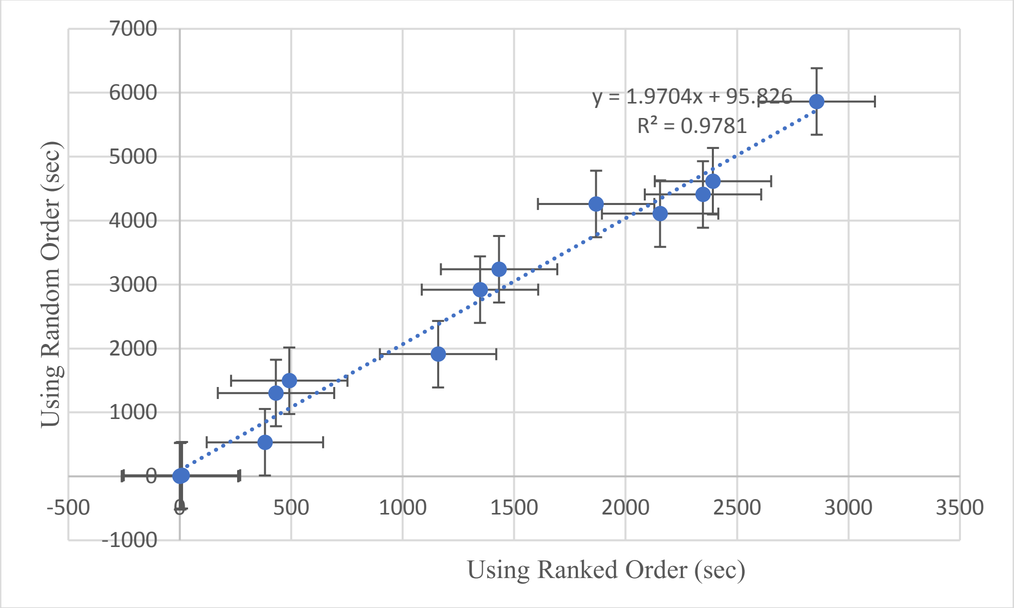


Fig. 45. Regression Analysis.

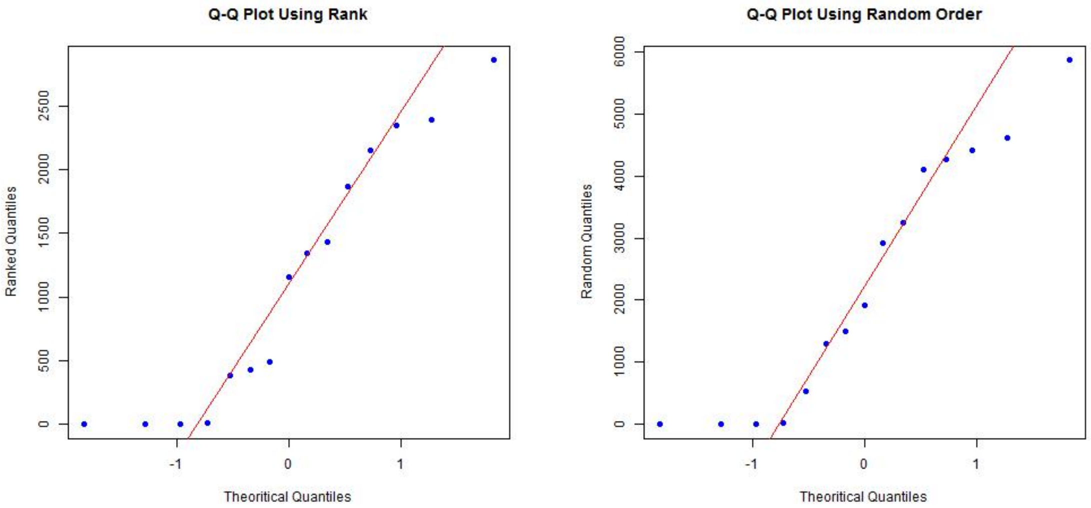


Fig. 46. Q-Q Plot of Simulation Times to Check Normality.

The Q-Q plots of the simulation times don’t show strongly that the simulation time data are normally distributed. This may be because of insufficient data. There always remains the concern regarding the scaling issue. That is why we have taken some other simulation outputs for the network with nodes numbered from 20 to 200. The simulation outputs of 20 Monte Carlo averages are given in the below figures.

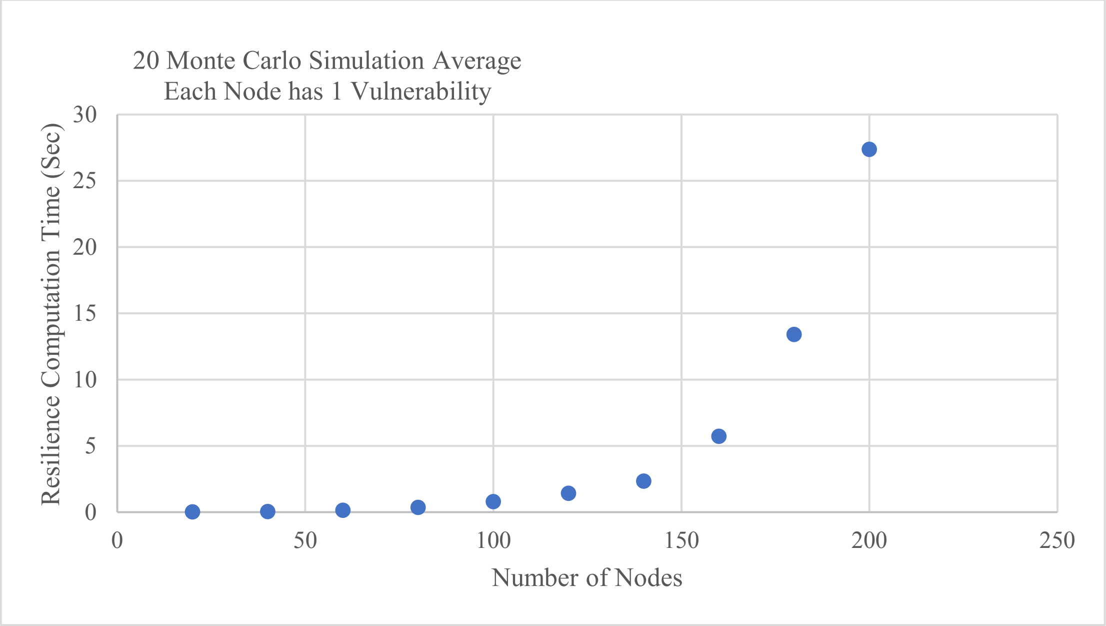


Fig. 47. Number of Network Nodes Vs Resilience Computation Time (Nodes are having 1 Vulnerability).

Fig. 47 shows the resilience computation time for a node (Security Server1) in Control DMZ layer for a different number of nodes. In this case nodes have only 1 vulnerability. Fig. 48 shows the resilience computation time for a node (Security Server1) where each node is having 3 vulnerabilities. The computation time is much less when there is only 1 vulnerability per host. The computation time is much higher with increment in number of nodes and when each host has 3 vulnerabilities. Fig. 49 shows a comparison of both Fig. 47 and Fig. 48. Log scale is used to facilitate the comparison.

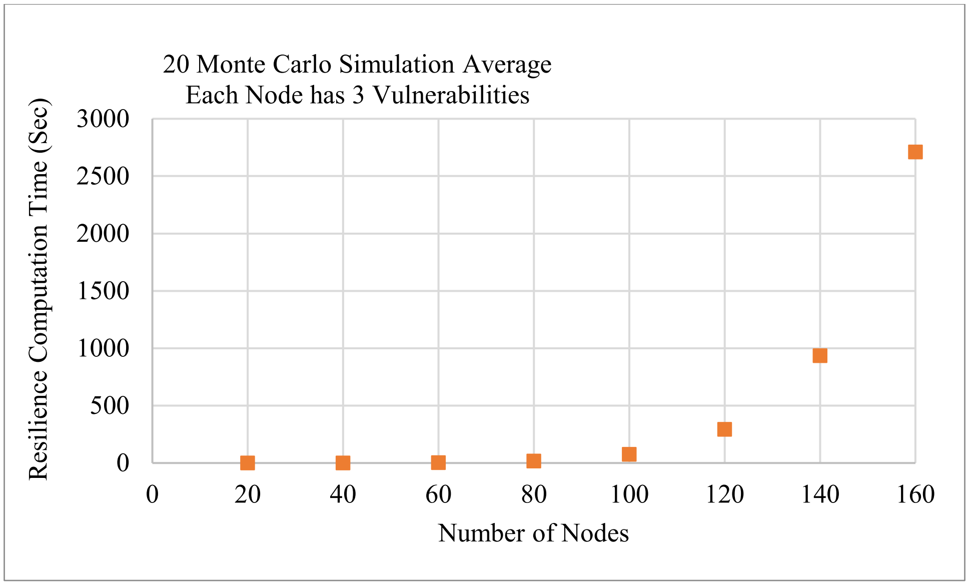


Fig. 48. Resilience Computation Time Vs Number of Network Nodes (Nodes are having 3 Vulnerabilities).

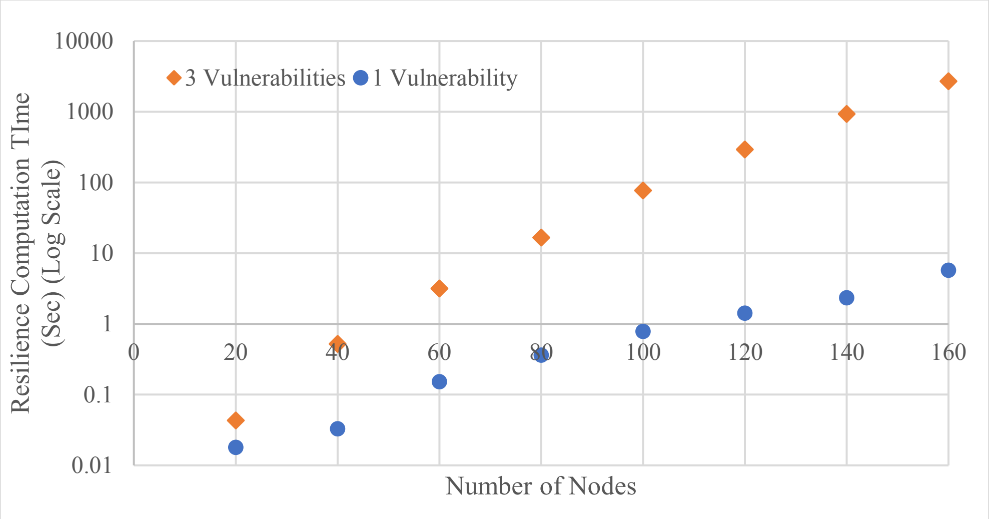


Fig. 49. Resilience Computation Time Comparison Vs Number of Vulnerabilities.

It is necessary to evaluate the computation time for the ranking metrics. This is given in Fig. 50 and Fig. 51.

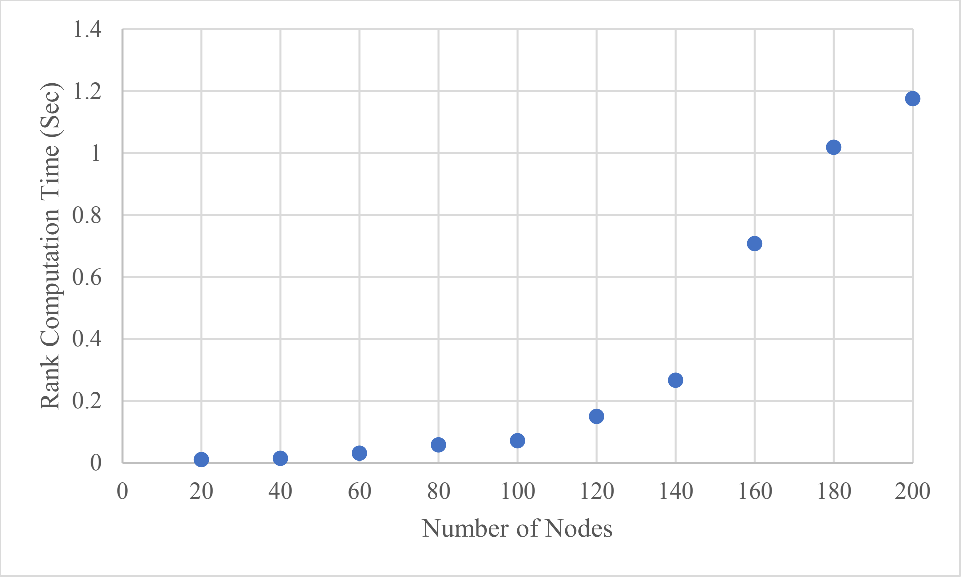


Fig. 50. Rank Computation Time Vs Number of Nodes.

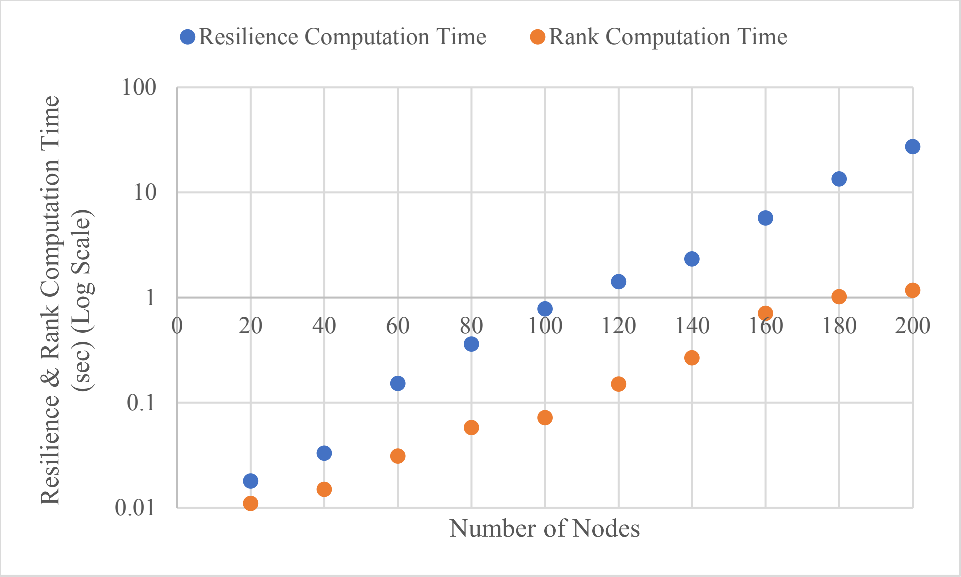


Fig. 51. Resilience and Rank Computation Time Vs Number of Nodes.

5.4 Validation

This section is going to present the validation of the ranking algorithm that we have proposed. Because of the restrictions of information disclosures followed by the power sector companies, there is no formal data available regarding the network structures and rankings of network elements which can be compared with our simulation data. Also, the direct quantification of resilience has not been conducted by the scholarly research to our knowledge. Thus, we have taken a model presented in [87] which also works for node ranking based on the exploitability metrics where they have followed Markov states transition probabilities for the ranking calculation. The state transition graph is given in Fig. 52.

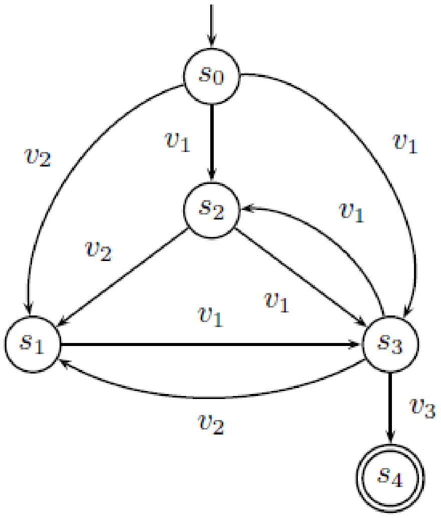


Fig. 52. Sample State Transition Graphs [87].

For validation purposes, the same state transition graph has been considered where the states are considered as nodes and transitions between states are considered edges between nodes. Additionally, the dummy source and target node have been considered so that the states which are nodes in this case, can be ranked using the ranking algorithm. The DAG model form of the graph as drawn from the figure is depicted in Fig. 53.

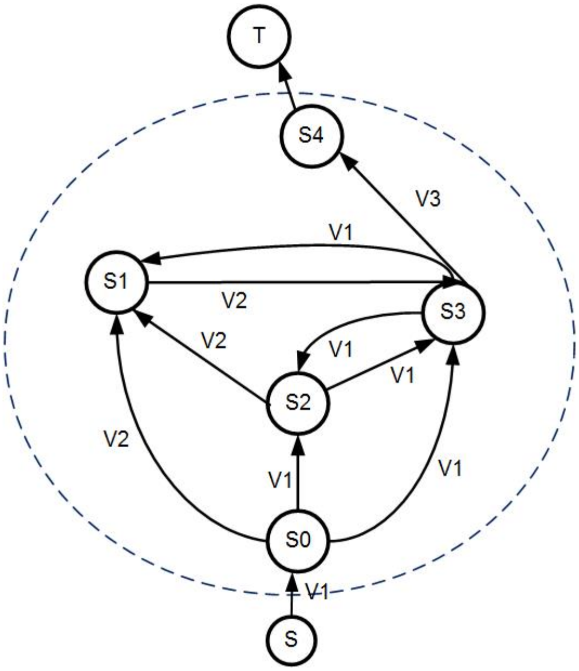


Fig. 53. DAG Model for Fig. 52**.**

The vulnerabilities that have been considered in the work by Kijsanayothin et al. are presented here with their CVE id and descriptions in TABLE 18 and the scores of those vulnerabilities are presented in TABLE 19.

TABLE 18

VULNERABILITIES FOR SAMPLE NETWORK

Vulnerability	CVE Number	Details

V1	CVE-2006-5794	Unspecified vulnerability in the sshd Privilege Separation Monitor in OpenSSH before 4.5 causes weaker verification that authentication has been successful, which might allow attackers to bypass authentication [101].
V2	CVE-2006-5051	Signal handler race condition in OpenSSH before 4.4 allows remote attackers to cause a denial of service (crash), and possibly execute arbitrary code if GSSAPI authentication is enabled, via unspecified vectors that lead to a double-free [101].
V3	CVE-2004-0148	wu-ftpd 2.6.2 and earlier, with the restricted-gid option enabled, allows local users to bypass access restrictions by changing the permissions to prevent access to their home directory, which causes wu-ftpd to use the root directory instead [101].

TABLE 19

SAMPLE NETWORK VULNERABILITY SCORES

Vulnerability	Exploitability	Impact	A_C	A_V	A_A	I_C	I_I	I_A
V1	10	6.4	1.0	0.71	0.704	0.275	0.275	0.275
V2	8.6	10	1.0	0.61	0.704	0.660	0.660	0.660
V3	3.9	10	0.395	0.71	0.704	0.660	0.660	0.660

TABLE 20 below demonstrates the calculations using the MVNRank algorithm where the nodes are ranked based on the relative criticality (RC_i) value.

TABLE 20

RANKING VALUE CALCULATION FOR SAMPLE NETWORK

States/Nodes	A_i	d_i	N_i	$C_d(i)$	EFw_i	IFw_i	R_i	RC_i	Rank
S_0	1	0.33	1	0.75	1.0	0.64	0.1584	0.0183	5
S_1	5.5	0.33	3	1	0.86	1.0	4.257	0.493	2
S_2	4	0.33	2	1	1.0	0.64	1.6896	0.196	3
S_3	6	0.5	3	1.5	1.0	0.64	8.64	1.0	1
S_4	6	1	1	0.25	0.39	1.0	0.585	0.0677	4

The ranking order of Mehta et al. and Kijsanayothin et al. is presented in TABLE 21 and compared with our ranking approach.

TABLE 21

RANKING VALUE COMPARISON FOR SAMPLE NETWORK

States/Nodes	Mehta et al.'s approach	Mehta et al.'s Rank Order	Kijsanayothin et al.'s Approach	Kijsanayothin's et al.'s Rank Oder	RC_i	Rank
S_0	0.150	3	0.150	4	0.0183	5
S_1	0.145	4	0.1287	5	0.493	2

S_2	0.102	5	0.1658	3	0.196	3
S_3	0.209	2	0.2548	2	1.0	1
S_4	0.394	1	0.394	1	0.0677	4

Both Mehta et al. and Kijsanayothin et al. have used the Markov states transition model. In those works, the authors found that S_4 is the highest value ranked state because it seems to be the target. In our ranking, S_4 is not the highest ranked node because it only contains one low exploitability (V1, exploitability 3.9) vulnerability. Thus, the nodes that can help (S_3, S_2, S_1) reach this S_4 node should a higher rank than S_4 which is found in our rank order. In our rank, S_3 is the highest ranked node because it has direct incoming paths (vulnerabilities) from S_0, S_1, S_2 and it has direct outgoing paths to S_4, no other nodes in the graph have direct paths to S_4 except S_3. In our approach, S_1 has the second highest rank because S_1 has direct incoming paths from S_0, S_2, S_3 and outgoing paths to S_3 from which S_4 is reachable. Thus, the ranking order seems more reasonable than the other two approaches.

To validate our results, we have compared them statistically with the approaches of Mehta and Kijsayanothin using a non-parametric test method which is a Wilcoxon signed ranked test. TABLE 22 presents the comparison of rank values between our approach compared with Mehta et al.'s approach and TABLE 23 shows rank values of our approach compared with Kijsayanothin et al.'s approach.

TABLE 22

RESULTS COMPARISON OF MVNRANK AND MEHTA APPROACH

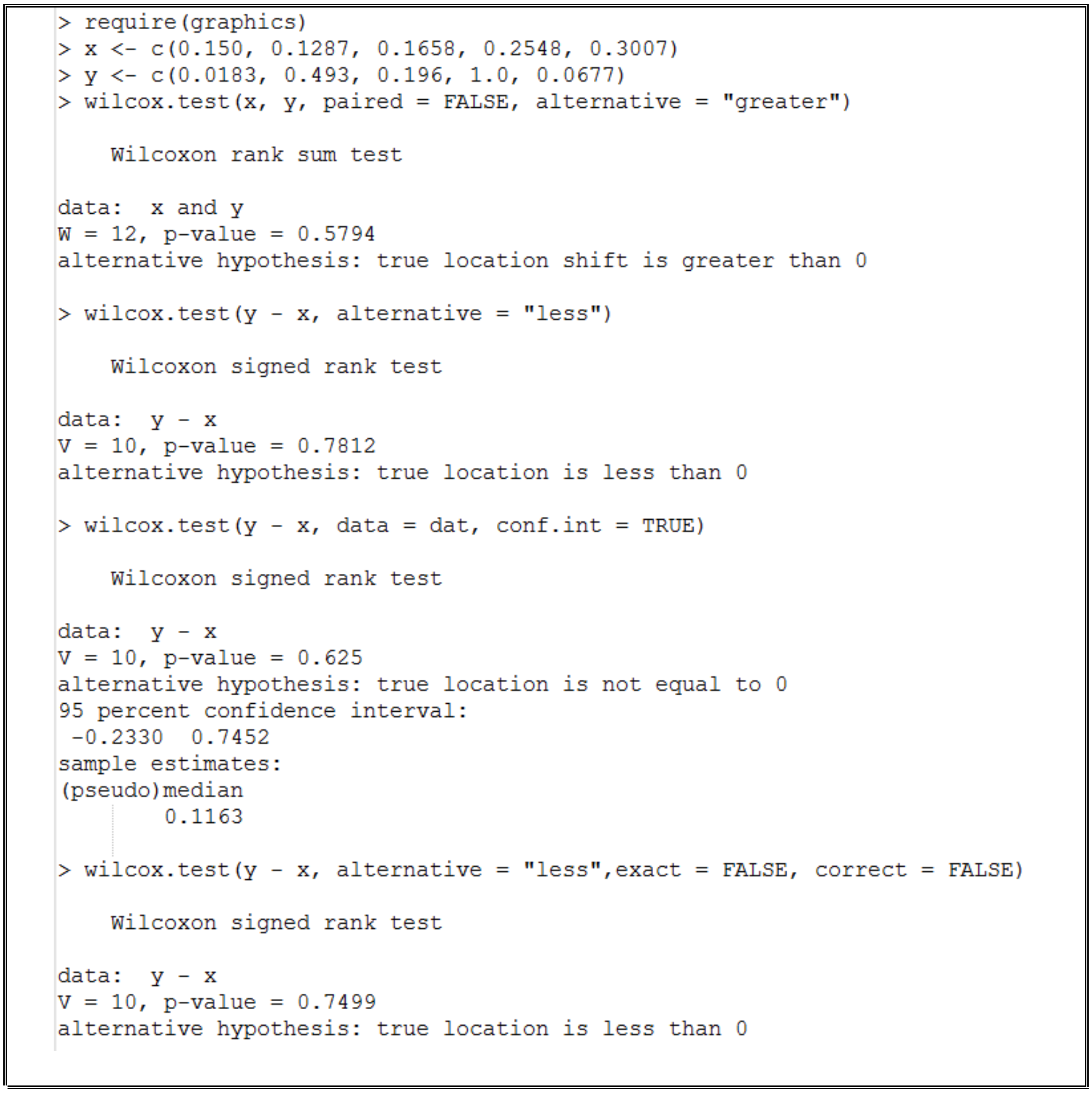
States/Nodes	Mehta et al.'s X_{1i}	Our Approach X_{2i}
S_0	0.150	0.0183
S_1	0.145	0.493
S_2	0.102	0.196
S_3	0.209	1.0
S_4	0.394	0.0677

TABLE 23

RESULTS COMPARISON OF MVNRANK AND KIJSAYANOTHIN APPROACH

States/Nodes	Kijsayanothin et al.'s X_{1i}	Our Approach X_{2i}
S_0	0.150	0.0183
S_1	0.1287	0.493
S_2	0.1658	0.196
S_3	0.2548	1.0
S_4	0.3007	0.0677

We have used the R software for generating the outputs of comparison with the Kijsayanothin approach and our approach. Below are the results shown from the R software output in Fig. 54.



```
> require(graphics)
> x <- c(0.150, 0.1287, 0.1658, 0.2548, 0.3007)
> y <- c(0.0183, 0.493, 0.196, 1.0, 0.0677)
> wilcox.test(x, y, paired = FALSE, alternative = "greater")

    Wilcoxon rank sum test

data:  x and y
W = 12, p-value = 0.5794
alternative hypothesis: true location shift is greater than 0

> wilcox.test(y - x, alternative = "less")

    Wilcoxon signed rank test

data:  y - x
V = 10, p-value = 0.7812
alternative hypothesis: true location is less than 0

> wilcox.test(y - x, data = dat, conf.int = TRUE)

    Wilcoxon signed rank test

data:  y - x
V = 10, p-value = 0.625
alternative hypothesis: true location is not equal to 0
95 percent confidence interval:
 -0.2330  0.7452
sample estimates:
(pseudo)median
        0.1163

> wilcox.test(y - x, alternative = "less",exact = FALSE, correct = FALSE)

    Wilcoxon signed rank test

data:  y - x
V = 10, p-value = 0.7499
alternative hypothesis: true location is less than 0
```

Fig. 54. Wilcoxon Signed Rank Test Output Using Kijsayanothin and Our Approach

The results show the values in the two samples are close with probability values greater than 0.5 for each of the cases. The 95% confidence interval is -0.2330 ~ 0.7452, which indicates a strong similarity of the median of the two data sets. The corresponding boxplot is given in Fig. 55 below.

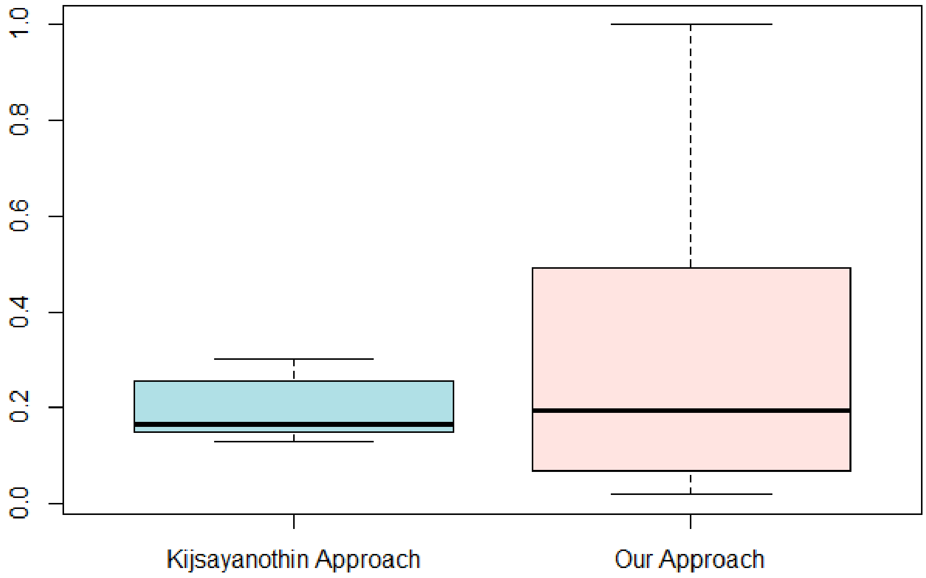


Fig. 55. Boxplot of Ranking Value of Kijsayanothin Approach and Our Approach

The above boxplot shows that the data of the two approaches are from two different distributions, but their median is almost similar. We have also plotted the boxplots of the Mehta approach, Kijsayanothin approach and our approach in the below diagram to have a comparison which is presented in the below.

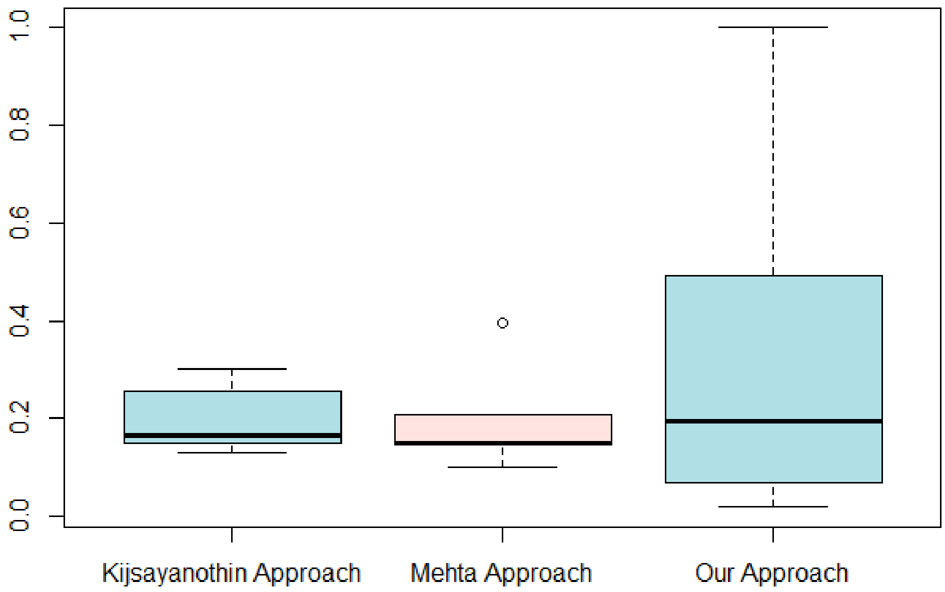


Fig. 56. Boxplot of Rank Value of Kijsayanothin, Mehta and Our Approach

Again, the boxplots show that although the distributions are different in three approaches, but their median is almost similar which validates our approach with other approaches.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

Improving resilience of network element or the network is not a straightforward process. It accounts for the combination of the effects by changing different parameters for the intermediate nodes as ranked by MVNRank. The proposed ranking method addresses the total exploitable path combinations that are offered to the attacker due to existing vulnerabilities in the network elements. The factors considered in calculating relative criticality are important from a network security analysis perspective as described in previous chapters. The simulation time is very high with a large number of nodes, so we have demonstrated the results using a small-scale simulation setup, but this ranking algorithm can be implemented in a large-scale network to rank and prioritize the network elements while trying to make particular network elements more resilient to cyber-attacks. MVNRank considered the possibility of potential physical damage in terms of electric power loss that can be accomplished by an attacker while ranking the nodes. This highlights the importance between traditional IT network elements and control system LAN elements. This node ranking algorithm can be useful in large power system network cyber resilience optimization processes.

6.1 Limitations & Challenges

This thesis has some limitations and challenges. Those are discussed below.

a) Firewall and security policies: The research doesn’t consider the firewall and security policies; rather, it considers only the vulnerabilities of the firewall product models. Our idea is to integrate the policies in future works.

b) High memory consumption and higher simulation time: As the network grows larger, memory consumption increases due to exponential increment in the number of paths in the graph model and computation of the path costs and impacts required for resilience calculation take a long simulation time. That is why this resilience improvement method may not be suitable for online simulation purposes; rather, because of its high simulation time it is being considered as an offline method of simulation.

c) Scaling: Although scaling is not a problem using the model, it will increase the simulation time and will need high-performance computing resources.

d) Power system physical connectivity consideration: This thesis mainly focuses on the ranking of the cyber nodes which can help reach a particular ICS in the bulk power system network. The study doesn't consider the physical bus system-based calculation of power loss capability of the individual generators or associated RTU's. In future, we plan to include the IEEE standard bus systems-based calculation for power loss potentials by the field location devices.

6.2 Future Works

As stated in the limitation and challenges section, we don't yet consider the physical bus systems to compute the power damage capability of each RTU and the firewall security policies. In the future, we plan to include firewall policies and physical bus system-based calculation of potential power damage of RTU to resemble real and robust network scenarios. The inclusion of firewall policies will provide some future research goals to be achieved by the ranking algorithms and resilience improvement process. Physical bus-system based power damage capability calculations can give a robust cyber-physical network scenario which would also lead to some other research objectives.

Overall, this thesis captures some of the analytics to be addressed in the bulk power system resilience improvement process within the scope of graduate study. There are opportunities to improve the ranking algorithm proposed in this thesis, and there is a long way to go with the research of bulk power system resiliency research.

REFERENCES

[1] M. S. TechCenter. "Security Update Severity Rating System," https://technet.microsoft.com/en-us/security/gg309177.aspx.

[2] P. Mell, K. Scarfone, and S. Romanosky, "A complete guide to the common vulnerability scoring system version 2.0." p. 23.

[3] Wikipedia. "Common Vulnerability Scoring System," https://en.wikipedia.org/wiki/Common_Vulnerability_Scoring_System.

[4] S. C. NERC, "MEMORANDUM-Use of 'Bulk Power System' versus 'Bulk Electric System' in Reliability Standards," 2012.

[5] S. Shetty, G. Kamdem, K. Brijesh, and N. David, "Cyber Resilience Metrics for Bulk Power System," *Risk Analysis Journal*, 2018.

[6] P. Paganini, "The US energy industry is constantly under cyber attacks," *Securityaffairs*, Novermber 19, 2014.

[7] A. Greenberg. "'Crash Override': The Malware That Took Down a Power Grid," June 13; https://www.wired.com/story/crash-override-malware/.

[8] J. Summers, and M. Walstrom. "Cyberattack on Critical Infrastructure: Russia and the Ukrainian Power Grid Attacks," https://jsis.washington.edu/news/cyberattack-critical-infrastructure-russia-ukrainian-power-grid-attacks/.

[9] K. Zetter. "Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid," https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/.

[10] D. U. Case, “Analysis of the Cyber Attack on the Ukrainian Power Grid,” 2016.

[11] FireEye. "Cyber Attacks on the Ukrainian Grid: What You Should Know," https://www.fireeye.com/content/dam/fireeye-www/global/en/solutions/pdfs/fe-cyber-attacks-ukrainian-grid.pdf.

[12] M. Holloway. "Stuxnet Worm Attack on Iranian Nuclear Facilities," http://large.stanford.edu/courses/2015/ph241/holloway1/.

[13] M. B. Kelley, “The Stuxnet attack on Iran’s nuclear plant was ‘far more dangerous’ than previously thought,” *Business Insider,* vol. 20, 2013.

[14] J. Grayson, “Stuxnet and Iran's Nuclear Program,” *Physics 241*, 2011.

[15] D. Kushner, “The real story of stuxnet,” *ieee Spectrum,* vol. 50, no. 3, pp. 48-53, 2013.

[16] R. Naraine, E. Protalinski, and D. Danchev, “Stuxnet attackers used 4 windows zero-day exploits,” *ZDnet Blog*, 2010.

[17] B. Kesler, “The vulnerability of nuclear facilities to cyber attack,” *Strategic Insights,* vol. 10, no. 1, pp. 15-25, 2011.

[18] Siemens, "SIMATIC WinCC / SIMATIC PCS 7: Information about Malware / Viruses / Trojan horses," 2010.

[19] T. Espiner, “Siemens warns Stuxnet targets of password risk,” *CNet. Retrieved November,* vol. 8, pp. 2015, 2010.

[20] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security." pp. 4490-4494.

[21] M. Kumar. "Dragonfly 2.0: Hacking Group Infiltrated European and US Power Facilities," September 07, 2017; https://thehackernews.com/2017/09/dragonfly-energy-hacking.html.

[22] A. Greenberg. "Hackers Gain Direct Access to US Power Grid Controls," September 6; https://www.wired.com/story/hackers-gain-switch-flipping-access-to-us-power-systems/.

[23] S. S. Response, "Dragonfly: Western Energy Companies Under Sabotage Threat," June 30, 2014.

[24] N. Nelson, “The impact of dragonfly malware on industrial control systems,” *SANS Institute*, 2016.

[25] G. Wangen, “The Role of Malware in Reported Cyber Espionage: A Review of the Impact and Mechanism,” *Information,* vol. 6, no. 2, pp. 183-211, May 18, 2015.

[26] ICS-CERT. "Abstract: Defense-in-Depth RP," https://ics-cert.us-cert.gov/Abstract-Defense-Depth-RP.

[27] N. R. Council, "Disaster resilience: a national imperative," Washington, DC: The National Academies Press, 2012.

[28] I. Linkov, D. A. Eisenberg, M. E. Bates, D. Chang, M. Convertino, J. H. Allen, S. E. Flynn, and T. P. Seager, "Measurable resilience for actionable policy," ACS Publications, 2013.

[29] I. Linkov, D. A. Eisenberg, K. Plourde, T. P. Seager, J. Allen, and A. Kott, “Resilience metrics for cyber systems,” *Environment Systems and Decisions,* vol. 33, no. 4, pp. 471-476, 2013.

[30] P. E. Roege, Z. A. Collier, J. Mancillas, J. A. McDonagh, and I. Linkov, “Metrics for energy resilience,” *Energy Policy,* vol. 72, pp. 249-256, 2014.

[31] K. Tierney, and M. Bruneau, “Conceptualizing and measuring resilience: A key to disaster loss reduction,” *TR news*, no. 250, 2007.

[32] A. Sharifi, and Y. Yamagata, “A conceptual framework for assessment of urban energy resilience,” *Energy Procedia,* vol. 75, pp. 2904-2909, 2015.

[33] R. Francis, and B. Bekera, “A metric and frameworks for resilience analysis of engineered and infrastructure systems,” *Reliability Engineering & System Safety,* vol. 121, pp. 90-103, 2014.

[34] S. A. Boyer, *SCADA: supervisory control and data acquisition*: International Society of Automation, 2009.

[35] K. Stouffer, J. Falco, and K. Scarfone, “Guide to industrial control systems (ICS) security,” *NIST special publication,* vol. 800, no. 82, pp. 16-16, 2011.

[36] ICS-CERT, "Recommended Practices for ICS."

[37] D. Kuipers, and M. Fabro, *Control systems cyber security: Defense in depth strategies*, Idaho National Laboratory (INL), 2006.

[38] T. G. Lewis, “Network science: theory and practice,” pp. 23, 2009.

[39] K. Kaynar, “A taxonomy for attack graph generation and usage in network security,” *Journal of Information Security and Applications,* vol. 29, pp. 27-56, 2016.

[40] P. Kijsanayothin, and R. Hewett, "Analytical approach to attack graph analysis for network security." pp. 25-32.

[41] T. Hamid, and C. Maple, “A Graph theoretical approach to Network Vulnerability Analysis and Countermeasures,” *IJCA Special Issue on Network Security and Cryptography NSC*, 2011.

[42] C. Phillips, and L. P. Swiler, "A graph-based system for network-vulnerability analysis." pp. 71-79.

[43] T. Zhang, M.-Z. Hu, D. Li, and L. Sun, "An effective method to generate attack graph." pp. 3926-3931.

[44] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs." pp. 273-284.

[45] X. Ou, and A. Singhal, "Attack graph techniques," *Quantitative Security Risk Assessment of Enterprise Networks*, pp. 5-8: Springer, 2012.

[46] X. Ou, S. Govindavajhala, and A. W. Appel, "MulVAL: A Logic-based Network Security Analyzer." pp. 8-8.

[47] Z. Liu, S. Li, J. He, D. Xie, and Z. Deng, "Complex network security analysis based on attack graph model." pp. 183-186.

[48] N. Poolsappasit, R. Dewri, and I. Ray, “Dynamic security risk management using bayesian attack graphs,” *IEEE Transactions on Dependable and Secure Computing,* vol. 9, no. 1, pp. 61-74, 2012.

[49] L. Williams, R. Lippmann, and K. Ingols, "GARNET: A graphical attack graph and reachability network evaluation tool," *Visualization for Computer Security*, pp. 44-59: Springer, 2008.

[50] M. Frigault, and L. Wang, *Measuring network security using bayesian network-based attack graphs*: IEEE, 2008.

[51] M. Frigault, “Measuring network security using Bayesian Network-based attack graphs,” Concordia University, 2010.

[52] M. Frigault, L. Wang, A. Singhal, and S. Jajodia, "Measuring network security using dynamic bayesian network." pp. 23-30.

[53] K. Ingols, R. Lippmann, and K. Piwowarski, "Practical attack graph generation for network defense." pp. 121-130.

[54] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph-based network vulnerability analysis." pp. 217-224.

[55] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs." pp. 49-63.

[56] B. Kordy, L. Piètre-Cambacédès, and P. Schweitzer, “DAG-based attack and defense modeling: Don’t miss the forest for the attack trees,” *Computer science review,* vol. 13, pp. 1-38, 2014.

[57] S. Noel, S. Jajodia, L. Wang, and A. Singhal, “Measuring security risk of networks using attack graphs,” *International Journal of Next-Generation Computing,* vol. 1, no. 1, pp. 135-147, 2010.

[58] V. Shandilya, C. B. Simmons, and S. Shiva, “Use of attack graphs in security systems,” *Journal of Computer Networks and Communications,* vol. 2014, 2014.

[59] P. Xie, J. H. Li, X. Ou, P. Liu, and R. Levy, "Using Bayesian networks for cyber security analysis." pp. 211-220.

[60] S. Roschke, F. Cheng, and C. Meinel, "Using vulnerability information and attack graphs for intrusion detection." pp. 68-73.

[61] C. Wang, Y. Wang, Y. Dong, and T. Zhang, "A novel comprehensive network security assessment approach." pp. 1-6.

[62] Y. Wang, X. Yun, Y. Zhang, S. Jin, and Y. Qiao, "Research of network vulnerability analysis based on attack capability transfer." pp. 38-44.

[63] M. Alhomidi, and M. Reed, "Risk assessment and analysis through population-based attack graph modelling." pp. 19-24.

[64] C.-W. Ten, G. Manimaran, and C.-C. Liu, "Cybersecurity for critical infrastructures: Attack and defense modeling," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans,* vol. 40, no. 4, pp. 853-865, 2010.

[65] N. Liu, J. Zhang, H. Zhang, and W. Liu, "Security assessment for communication networks of power control systems using attack graph and MCDM," *IEEE Transactions on Power Delivery,* vol. 25, no. 3, pp. 1492-1500, 2010.

[66] S. Zonouz, K. M. Rogers, R. Berthier, R. B. Bobba, W. H. Sanders, and T. J. Overbye, "SCPSE: Security-oriented cyber-physical state estimation for power grid critical infrastructures," *IEEE Transactions on Smart Grid,* vol. 3, no. 4, pp. 1790-1799, 2012.

[67] Y. Zhang, L. Wang, Y. Xiang, and C.-W. Ten, "Power system reliability evaluation with SCADA cybersecurity considerations," *IEEE Transactions on Smart Grid,* vol. 6, no. 4, pp. 1707-1721, 2015.

[68] C. Vellaithurai, A. Srivastava, S. Zonouz, and R. Berthier, "CPIndex: cyber-physical vulnerability assessment for power-grid infrastructures," *IEEE Transactions on Smart Grid,* vol. 6, no. 2, pp. 566-575, 2015.

[69] G. P. Cimellaro, A. M. Reinhorn, and M. Bruneau, "Framework for analytical quantification of disaster resilience," *Engineering Structures,* vol. 32, no. 11, pp. 3639-3649, 2010.

[70] M. Ouyang, and Z. Wang, "Resilience assessment of interdependent infrastructure systems: With a focus on joint restoration modeling and analysis," *Reliability Engineering & System Safety,* vol. 141, pp. 74-82, 2015.

[71] D. A. Reed, K. C. Kapur, and R. D. Christie, "Methodology for assessing the resilience of networked infrastructure," *IEEE Systems Journal,* vol. 3, no. 2, pp. 174-180, 2009.

[72] A. A. Ganin, E. Massaro, A. Gutfraind, N. Steen, J. M. Keisler, A. Kott, R. Mangoubi, and I. Linkov, “Operational resilience: concepts, design and analysis,” *Scientific reports,* vol. 6, pp. 19540, 2016.

[73] J. T. Chambers, and J. W. Thompson. "Common Vulnerabilty Scoring System: Final Report and Recommendations By The Council," https://www.first.org/cvss/cvss-dhs-12-02-04.pdf.

[74] US-CERT. "Vulnerability Bulletins," https://www.us-cert.gov/ncas/bulletins.

[75] NIST. "National Vulnerability Database," https://nvd.nist.gov/general.

[76] NIST. "National Vulnerability Database - Vulnerability Metrics," https://nvd.nist.gov/vuln-metrics/cvss#.

[77] ICS-CERT. "Industrial Control System Cyber Emergency Response," https://ics-cert.us-cert.gov/.

[78] Symantec. "Threat Severity Assessment," https://www.symantec.com/security_response/severityassessment.jsp.

[79] Symantec, *Severity Assessment, Threats, Events, Vulnerabilities, Risks*, 2006.

[80] M. S. TechCenter. "Microsoft Exploitability Index," https://technet.microsoft.com/en-us/security/cc998259.

[81] MITRE. "CWE Common Weakness Enumeration," https://cwe.mitre.org/.

[82] MITRE, "CWE Common Weakness Enumeration Details."

[83] FIRST. "CVSS," https://www.first.org/cvss/.

[84] S. Brin, and L. Page, “Anatomy of a Large-Scale Hypertextual Web Search Engine. 7th Intl World Wide Web Conf,” 1998.

[85] S. Brin, and L. Page, “Reprint of: The anatomy of a large-scale hypertextual web search engine,” *Computer networks,* vol. 56, no. 18, pp. 3825-3833, 2012.

[86] W. Xing, and A. Ghorbani, "Weighted pagerank algorithm." pp. 305-314.

[87] P. Kijsanayothin, and R. Hewett, "Exploit-based analysis of attack models." pp. 183-186.

[88] P. Li, and X. Qiu, "NodeRank: An Algorithm to Assess State Enumeration Attack Graphs." pp. 1-5.

[89] X. Yang, S. Shunhong, and L. Yuliang, "Vulnerability ranking based on exploitation and defense graph." pp. V1-163-V1-167.

[90] P. Barford, M. Dacier, T. G. Dietterich, M. Fredrikson, J. T. Giffin, S. Jajodia, S. Jha, J. H. Li, P. Liu, and P. Ning, “Cyber SA: Situational Awareness for Cyber Defense,” *Cyber Situational Awareness,* vol. 46, no. 1, pp. 3-13, 2010.

[91] R. E. Sawilla, and X. Ou, "Identifying Critical Attack Assets in Dependency Attack Graphs." pp. 18-34.

[92] R. A. Miura-Ko, and N. Bambos, "SecureRank: A risk-based vulnerability management scheme for computing infrastructures." pp. 1455-1460.

[93] V. Mehta, C. Bartzis, H. Zhu, E. Clarke, and J. Wing, "Ranking attack graphs." pp. 127-144.

[94] L. Lu, R. Safavi-Naini, M. Hagenbuchner, W. Susilo, J. Horton, S. Yong, and A. Tsoi, “Ranking attack graphs with graph neural networks,” *Information Security Practice and Experience*, pp. 345-359, 2009.

[95] Wikipedia. "Centrality," https://en.wikipedia.org/wiki/Centrality#Degree_centrality.

[96] D. Austin. "How Google Finds Your Needle in the Web's Haystack," http://www.ams.org/publicoutreach/feature-column/fcarc-pagerank.

[97] A. Hagberg, P. Swart, and D. Schult. "NetworkX," https://networkx.github.io/.

[98] "Graphviz," http://graphviz.readthedocs.io/en/stable/manual.html.

[99] "PyGraphviz," https://pygraphviz.github.io/.

[100] "NetworkX Muti-DiGraph," https://networkx.github.io/documentation/networkx-1.10/reference/classes.multidigraph.html.

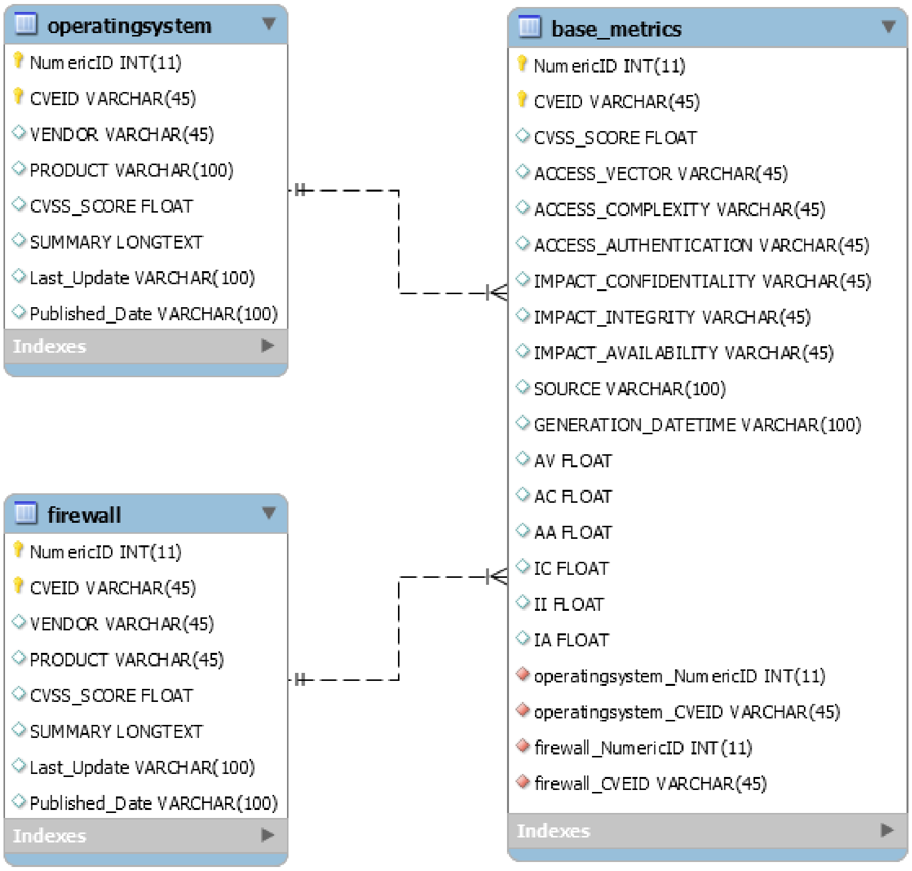
[101] MITRE. "Common Vulnerability Exposure," http://cve.mitre.org/cve/search_cve_list.html.

[102] NIST, "NVD Data Feeds," 2017.

APPENDICES

APPENDIX A: DATABASE SCHEMATIC FOR SIMULATION DESIGN

For simulation purposes, we have developed a database in MySQL Workbench. The database has been built using the xml file from the NVD website [102] and is used for simulation purposes. The database schema mainly consists of three tables: base_metrics, operatingsystem and firewall. The schematic diagram of the tables is given below in Fig. 57.



operatingsystem
NumericID INT(11)
CVEID VARCHAR(45)
VENDOR VARCHAR(45)
PRODUCT VARCHAR(100)
CVSS_SCORE FLOAT
SUMMARY LONGTEXT
Last_Update VARCHAR(100)
Published_Date VARCHAR(100)
Indexes

firewall
NumericID INT(11)
CVEID VARCHAR(45)
VENDOR VARCHAR(45)
PRODUCT VARCHAR(45)
CVSS_SCORE FLOAT
SUMMARY LONGTEXT
Last_Update VARCHAR(100)
Published_Date VARCHAR(100)
Indexes

base_metrics
NumericID INT(11)
CVEID VARCHAR(45)
CVSS_SCORE FLOAT
ACCESS_VECTOR VARCHAR(45)
ACCESS_COMPLEXITY VARCHAR(45)
ACCESS_AUTHENTICATION VARCHAR(45)
IMPACT_CONFIDENTIALITY VARCHAR(45)
IMPACT_INTEGRITY VARCHAR(45)
IMPACT_AVAILABILITY VARCHAR(45)
SOURCE VARCHAR(100)
GENERATION_DATETIME VARCHAR(100)
AV FLOAT
AC FLOAT
AA FLOAT
IC FLOAT
II FLOAT
IA FLOAT
operatingsystem_NumericID INT(11)
operatingsystem_CVEID VARCHAR(45)
firewall_NumericID INT(11)
firewall_CVEID VARCHAR(45)
Indexes

Fig. 57. Simulation Database Schematic.

The product vendor, model and associated vulnerabilities are stored in the firewall and operatingsystem table. For example, if the product is CISCO ASA5500, then the product is listed in the table firewall. Each vulnerability has a CVE id that is compared with the base_metrics table. The Base_metrics table has all the quantitative values for the vulnerabilities identified by the CVE id.

VITA

Md Ariful Haque is a graduate student in the department of Modeling Simulation and Visualization Engineering (MSVE) at Old Dominion University starting Fall’2016. He is currently working as Graduate Research Assistant under the supervision of Professor Dr. Sachin Shetty. Mr. Haque has received his bachelor’s degree in Electrical and Electronics Engineering (EEE) from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in November 2006. Mr. Haque has earned his Master’s in Business Administration (MBA) from Institute of Business Administration (IBA), University of Dhaka, Bangladesh in June 2016. Mr. Haque has 8 years of professional experience in the telecommunication sector, working at Grameenphone and Ericsson, where he conducted Core Network startup, testing, integration and before joining Old Dominion University as a graduate student. His research interests include modeling simulation, data analysis, cyber security, IoT security, and cloud computing.