

# TECHNIQUES FOR ENHANCED VULNERABILITY MANAGEMENT FOR INDUSTRIAL INTERNET OF THINGS

GEETA



AMAR NATH & SHASHI KHOSLA SCHOOL OF INFORMATION  
TECHNOLOGY

INDIAN INSTITUTE OF TECHNOLOGY DELHI

June 2022

# TECHNIQUES FOR ENHANCED VULNERABILITY MANAGEMENT FOR INDUSTRIAL INTERNET OF THINGS

by

GEETA

Amar Nath & Shashi Khosla School of Information Technology

Submitted

in fulfillment of the requirements of the degree of Doctor of Philosophy

to the



**Indian Institute of Technology Delhi**

**June 2022**

DEDICATED TO

*My family.*

Thank you for your eternal support.

# Certificate

This is to certify that the thesis titled **Techniques for Enhanced Vulnerability Management for Industrial Internet of Things** being submitted by **Ms. GEETA** for the award of **Doctor of Philosophy in Information Technology** is a record of bona fide work carried out by her under my guidance and supervision at the Amar Nath and Shashi Khosla School of Information Technology, Indian Institute of Technology Delhi. The work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma.

Kolin Paul  
Professor  
Department of Computer Science and Engineering  
Indian Institute of Technology Delhi  
New Delhi- 110016

# Acknowledgements

A Ph.D. is a ride with a unique track of insightful, enriching, frustrating, enlightening and memorable years, where the co-passengers are a source of motivation and reinvigoration. I wish to acknowledge all the people who helped me in their ways in the journey of making my dream a reality.

First, I would like to thank my supervisor, the most cherished person I have ever met, Prof. Kolin Paul, for his invaluable guidance, exciting research discussions, words of encouragement, unwavering faith in my capability and continuous feedback that has led to the work in this thesis. He always provided a healthy work environment that motivated me to achieve my goals. The work flexibility that he allowed; let me balance my personal and professional life. He helped me grow as an independent researcher, allowing me to explore a new domain and work on ideas on my terms and pace. He always provided practical insights and feedback on my thesis work. I also thank him for the constant feedback he provided on my research writing and presentations that have helped me refine the skills necessary to succeed in research and academia. I thank him for teaching me to always look on the positive side of the coin. To summarize, I would say that this thesis is a result of my pursuance and his influence.

I would also like to thank my research collaborator Dr. Alaa Allakany (Kyushu University, Japan), Prof. Koji Okamura (Kyushu University, Japan) and Dr. Praveen Gauravram (TCS Australia) for the valuable discussion that helped me to refine my research works. The critic and thought-provoking review, along with an appreciation for work, motivated me towards continuous improvements.

I would also like to thank my SRC members (*Ranjan*<sup>3</sup> of my SRC committee) Prof. Smriti Ranjan Sarangi, Prof. Priti Ranjan Panda and Prof. Ranjan Bose for their insightful questions and excellent suggestions.

I am grateful to my thesis examiners Prof. Sandeep Kumar Shukla (IIT Kanpur, India) and Prof. Nilanjan Banerjee (University of Maryland, Baltimore County, USA) for providing detailed comments and recommendations to mold my thesis in a more interesting way.

I would like to thank the lab in-charge Manju Chopra and the administrative staff at SIT and CSE departments for facilitating administrative matters. I am also grateful to my friends Vijay Kumar, Aruna Bansal, Himanshu Gandhi, Shubhani Gupta, Dr. Chandrika Bhardwaj, Srishti Kulshrestha, Britty Baby, Priyanka Chauhan and Pooja Jangir at IIT Delhi with whom I shared precious learning years of my life. The never-ending discussions, presentation rehearsals, manuscript reviews, experience-sharing sessions and wonderful lab environment kept me motivated and reinvigorated throughout the journey. I take this opportunity to express my earnest gratitude to Dr. Madhulika Mohanty and Dr. Rajesh Kedia for their professional guidance.

I would also like to thank my siblings Anju Yadav and Dr. Manish Yadav and sister-in-law Pooja Yadav for always being there whenever I needed to confide in them. I am also grateful to my mother-in-law and father-in-law for their help when I have to attend conferences. I am thankful to them for never imposing family responsibilities and understanding when I could not attend a family function. I also appreciate Komal and Anand for always being available and helping me in many ways.

Even imagining the thought of starting a Ph.D. with a kid in the womb is very challenging. I started both the motherhood and Ph.D. journey together. I could handle both things because of four strong pillars behind me: my mother, father, husband Kamal, and little munchkin Lakshya. I am blessed with their love and support throughout the journey. I would not have started and survived this journey without my husband's continuous efforts. A night goodbye with a smile and saying, 'Don't worry about me'. while wishing to sleep

in mom's lap was a challenging moment of the day for Lakshya. I owe all those hours to my son. I tried to assimilate my father's ambition, mother's autodidact, husband's dedication, and son's curiosity to walk this path. I dedicate my thesis to my family.

**Geeta**

# Abstract

Industrial Control Systems (ICS) are characterized by large numbers of tightly integrated, interdependent and heterogeneous components in a network. They act as a base system for safety and mission-critical Industrial Internet of Things (IIoT) applications such as smart grids, nuclear power plants, process control systems and robotics systems. The complex ICS, e.g., Supervisory Control and Data Acquisition (SCADA), consists of many interdependent subsystems. Modern SCADA systems are an amalgam of IIoT and legacy systems. IIoT is essentially a realization of advances in the connectivity of hardware and data networks that SCADA provides. Therefore, modern SCADA systems can be considered a use-case for IIoT-based systems. The modernization of the SCADA system, standardization of communication protocols and almost ubiquitous interconnectivity courtesy for IIoT has drastically increased the attack surface of the SCADA systems. Systematic Vulnerability Management (VM) of these attack surfaces minimizes risks and impacts associated with vulnerability exploitation. VM is a cyclical practice of identifying, analyzing, prioritizing and fixing/ monitoring possible exploitation of vulnerabilities in complex End-to-End (E2E) systems. State-of-art vulnerability discovery and real-time

exploitations monitoring approaches fail to discover and monitor Multi-Host Multi-Stage (MhMs) attacks since they rely on isolated systems testing and monitoring. The Central Monitoring System (CMS) needs each system log transmitted to the central server, hence having a significant network overhead. Moreover, the recent attacks demonstrate that whenever an adversary can not directly exploit a vulnerability, it exploits a series of vulnerabilities to reach the target node, generally referred to as MhMs attacks. In this thesis, we provide a three-level security solutions VM for MhMs attacks, including a detailed vulnerability analysis for SCADA with the help of various datasets, which clearly demonstrate the importance of the problem statement. We first discover all such vulnerabilities that lead to paths to the critical node, then prioritize the vulnerabilities for efficient patching, followed by monitoring the critical vulnerabilities.

We assess the SCADA system vulnerabilities leveraging the National Vulnerability Database (NVD) to understand the severity and characteristics of these vulnerabilities by using SCADA-related keywords such as SCADA, PLC, IIoT, ICS, etc. We analyze the extracted vulnerabilities by year-wise vulnerability count, attack-vector-wise analysis, impact analysis of the vulnerabilities on the CIA triad and CWE-count analysis. Based on this analysis outcome, we find answers to four critical questions regarding the SCADA system vulnerabilities. Next, we developed a possible MhMs attacks discovery framework *IoT-PEN*. This leverages the standalone system (discovered) vulnerabilities, network topology and target graphs. *IoT-PEN* also generates a security state report of each system and possible MhMs attacks. The framework evaluations demonstrate that the framework is efficient and scalable to larger networks.

To address the challenges in prioritizing patches in ICSs, we proposed two approaches: *SmartPatch* and *PatchRank*. *PatchRank* demonstrates that by incorporating domain-specific characteristics, a practical severity score is assigned to a vulnerability as compared to widely used scoring systems in industries, i.e., Common Vulnerability Scoring Systems (CVSS). *PatchRank* models the attacker-defender scenario as a two-player simultaneous non-cooperative game. However, patching is a repetitive process. Therefore, *PatchRank* is extended to *SmartPatch*. *SmartPatch* models the scenario as multiple attackers and multiple defenders repetitive games by considering the architectural features: functional dependencies, topological dependencies, vulnerabilities assessment features and patch dependencies. We validate the applicability of *SmartPatch* by considering the case study of an interdependent, complex SCADA chain in the smart grid system using the IEEE 5-Bus system. Our comparative analysis of the proposed approach with state-of-the-art approaches demonstrates that *SmartPatch* reduces Residual Impact Score (RIS) by a faster rate, i.e., after each iteration, the RIS value for *SmartPatch* is the least.

To detect ongoing attacks, we propose *GLoM* : A Global Monitor using SpatioTemporally Correlated Local Monitor that can detect ongoing MhMs attacks on the connected systems. It leverages deep learning-based algorithms to detect anomalies with high accuracy and attack graphs to map various anomalous behavior to detect MhMs attacks. *GLoM* is a two-stage deep learning-based model, where the workload is divided between Local Monitors (LM) and Global Monitors (GM). LMs use LSTM to detect the abnormal behavior of a system leveraging syslogs. Parallely, GM discovers possible vulnerabilities on the devices using vulnerability scanners in the network, followed by generating Possi-

ble Attack Graphs (*PAG*) by mapping the prerequisites and post-conditions required to exploit a vulnerability. The framework highlights MhMs possible attacks if there is an attack path from the source to the target node. In last, critical paths are identified using exploitability scores. The similarity index between possible attack-paths and evidence from logs identifies the most probable attack scenario an adversary may be following. Using LMs, network communication overhead decreased by 88% on the publicly available dataset OpenStack (Loghub). LSTM-based anomaly detection shows 99% accuracy in detecting the anomalous logs with an average anomalous log prediction overhead of 0.6 msec. We achieved 98% and 97% accuracy in generating the prerequisites and post-conditions of a vulnerability, respectively. We evaluated *GLoM* efficiency in MhMs attack detections using a local testbed.

In brief, this thesis aims at building techniques for efficient vulnerability management to discover MhMs possible attacks. It provides vulnerability prioritization techniques by considering the domain context. It also proposes a global monitoring system to detect ongoing multi-host exploitations using system logs.

# सार

औद्योगिक नियंत्रण प्रणाली (ICS) एक नेटवर्क में बड़ी संख्या में कसकर एकीकृत, अन्योन्याश्रित और विषम घटकों की विशेषता है। वे सुरक्षा और मिशन-महत्वपूर्ण औद्योगिक इंटरनेट ऑफ थिंग्स (IIoT) अनुप्रयोगों जैसे स्मार्ट ग्रिड, परमाणु ऊर्जा संयंत्र, प्रक्रिया नियंत्रण प्रणाली और रोबोटिक्स सिस्टम के लिए एक आधार प्रणाली के रूप में कार्य करते हैं। जटिल ICS, जैसे, पर्यवेक्षी नियंत्रण और डेटा अधिग्रहण (SCADA), में कई अन्योन्याश्रित उप-प्रणालियाँ शामिल हैं। आधुनिक SCADA सिस्टम IIoT और विरासत प्रणालियों का एक मिश्रण हैं। IIoT अनिवार्य रूप से SCADA द्वारा प्रदान किए जाने वाले हार्डवेयर और डेटा नेटवर्क की कनेक्टिविटी में प्रगति की प्राप्ति है। इसलिए, आधुनिक SCADA सिस्टम को IIoT-आधारित सिस्टम के लिए उपयोग-मामला माना जा सकता है। SCADA प्रणाली के आधुनिकीकरण, संचार प्रोटोकॉल के मानकीकरण और IIoT के लिए लगभग सर्वव्यापी इंटरकनेक्टिविटी शिष्टाचार ने SCADA सिस्टम की हमले की सतह को काफी बढ़ा दिया है। इन आक्रमण सतहों का व्यवस्थित भेद्यता प्रबंधन (VM) भेद्यता शोषण से जुड़े जोखिमों और प्रभावों को कम करता है। VM जटिल एंड-टू-एंड (E2E) सिस्टम में कमजोरियों के संभावित शोषण की पहचान, विश्लेषण, प्राथमिकता और निर्धारण/निगरानी का एक चक्रीय अभ्यास है। अत्याधुनिक भेद्यता खोज और रीयल-टाइम शोषण निगरानी दृष्टिकोण मल्टी-होस्ट मल्टी-स्टेज (MhMs) हमलों की खोज और निगरानी करने में विफल होते हैं क्योंकि वे पृथक सिस्टम परीक्षण और निगरानी पर भरोसा करते हैं। केंद्रीय निगरानी प्रणाली को केंद्रीय सर्वर को प्रेषित प्रत्येक सिस्टम लॉग की आवश्यकता होती है, इसलिए एक महत्वपूर्ण नेटवर्क ओवरहेड होता है।

इसके अलावा, हाल के हमलों से पता चलता है कि जब भी कोई विरोधी सीधे तौर पर भेद्यता का फायदा नहीं उठा सकता है, तो वह लक्ष्य नोड तक पहुंचने के लिए कमजोरियों की एक श्रृंखला का फायदा उठाता है, जिसे आम तौर पर MhMs हमलों के रूप में जाना जाता है। इस थीसिस में, हम MhMs हमलों के लिए तीन-स्तरीय सुरक्षा समाधान वीएम प्रदान करते हैं, जिसमें विभिन्न डेटासेट की सहायता से एससीएडीए के लिए विस्तृत भेद्यता विश्लेषण शामिल है, जो स्पष्ट रूप से समस्या कथन के महत्व को प्रदर्शित करता है। हम पहले ऐसी सभी कमजोरियों का पता लगाते हैं जो महत्वपूर्ण नोड तक ले जाती हैं, फिर कुशल पैचिंग के लिए कमजोरियों को प्राथमिकता देते हैं, इसके बाद महत्वपूर्ण कमजोरियों की निगरानी करते हैं।

हम SCADA से संबंधित कीवर्ड जैसे SCADA, PLC, IIoT, ICS, आदि का उपयोग करके इन कमजोरियों की गंभीरता और विशेषताओं को समझने के लिए राष्ट्रीय भेद्यता डेटाबेस (NVD) का लाभ उठाते हुए SCADA सिस्टम कमजोरियों का आसंजन करते हैं। हम वर्ष-वार द्वारा निकाली गई

कमजोरियों का विश्लेषण करते हैं। भेद्यता गणना, हमले-वेक्टर-वार विश्लेषण, CIA ट्रायड पर कमजोरियों का प्रभाव विश्लेषण और सीडब्ल्यूई-गिनती विश्लेषण। इस विश्लेषण के परिणाम के आधार पर, हमें SCADA सिस्टम की कमजोरियों के बारे में चार महत्वपूर्ण सवालों के जवाब मिलते हैं। इसके बाद, हमने एक संभावित MhM अटैक डिस्कवरी फ्रेमवर्क *IoT-PEN* विकसित किया। यह स्टैंडअलोन सिस्टम (खोजी गई) कमजोरियों, नेटवर्क टोपोलॉजी और लक्ष्य ग्राफ का लाभ उठाता है। *IoT-PEN* प्रत्येक सिस्टम और संभावित MhM हमलों की सुरक्षा स्थिति रिपोर्ट भी तैयार करता है। ढांचे के मूल्यांकन से पता चलता है कि ढांचा बड़े नेटवर्क के लिए कुशल और मापनीय है।

ICS में पैच को प्राथमिकता देने में चुनौतियों का समाधान करने के लिए, हमने दो दृष्टिकोण प्रस्तावित किए: *SmartPatch* और *PatchRank*। *PatchRank* दर्शाता है कि डोमेन-विशिष्ट विशेषताओं को शामिल करके, उद्योगों में व्यापक रूप से उपयोग किए जाने वाले स्कोरिंग सिस्टम, यानी कॉमन वलनरेबिलिटी स्कोरिंग सिस्टम (CVSS) की तुलना में एक व्यावहारिक गंभीरता स्कोर एक भेद्यता को सौंपा गया है। *PatchRank* हमलावर-डिफेंडर परिदृश्य को दो-खिलाड़ी एक साथ गैर-सहकारी खेल के रूप में मॉडल करता है। हालाँकि, पैचिंग एक दोहराव वाली प्रक्रिया है। इसलिए, *PatchRank* को *SmartPatch* तक बढ़ा दिया गया है। *SmartPatch* वास्तुशिल्प सुविधाओं पर विचार करके कई हमलावरों और कई रक्षकों के दोहराव वाले खेलों के रूप में परिदृश्य को मॉडल करता है: कार्यात्मक निर्भरता, टोपोलॉजिकल निर्भरता, कमजोरियाँ मूल्यांकन सुविधाएँ और पैच निर्भरता। हम IEEE 5-बस प्रणाली का उपयोग करते हुए स्मार्ट ग्रिड सिस्टम में एक अन्योन्याश्रित, जटिल एससीएडीए श्रृंखला के केस स्टडी पर विचार करके *SmartPatch* की प्रयोज्यता को मान्य करते हैं। अत्याधुनिक दृष्टिकोणों के साथ प्रस्तावित दृष्टिकोण का हमारा तुलनात्मक विश्लेषण दर्शाता है कि *SmartPatch* अवशिष्ट प्रभाव स्कोर (RIS) को तेज दर से कम करता है, अर्थात्, प्रत्येक पुनरावृत्ति के बाद, *SmartPatch* के लिए RIS मूल्य सबसे कम है।

चल रहे हमलों का पता लगाने के लिए, हम *GLoM* का प्रस्ताव करते हैं: SpatioTemporally सहसंबद्ध स्थानीय मॉनिटर का उपयोग करने वाला एक वैश्विक मॉनिटर जो कनेक्टेड सिस्टम पर चल रहे MhM हमलों का पता लगा सकता है। यह MhMs हमलों का पता लगाने के लिए विभिन्न विषम व्यवहारों को मैप करने के लिए उच्च सटीकता और हमले के ग्राफ के साथ विसंगतियों का पता लगाने के लिए गहन शिक्षण-आधारित एल्गोरिदम का लाभ उठाता है। *GLoM* एक दो चरणों वाला गहन शिक्षण-आधारित मॉडल है, जहाँ कार्यभार को स्थानीय मॉनिटर्स (LM) और ग्लोबल मॉनिटर्स (GM) के बीच विभाजित किया जाता है। LMs सिस्टम का लाभ उठाने वाले syslogs के असामान्य व्यवहार का पता लगाने के लिए LSTM का उपयोग करते हैं। समानांतर रूप से, GM नेटवर्क में भेद्यता स्कैनर का उपयोग करने वाले उपकरणों पर संभावित कमजोरियों का पता लगाता है, इसके बाद संभावित हमले के ग्राफ (पीएजी) उत्पन्न करके एक भेद्यता का फायदा उठाने के लिए आवश्यक पूर्वापेक्षाओं और बाद की स्थितियों का मानचित्रण करता है। यदि स्रोत से लक्ष्य नोड तक कोई हमला

पथ है, तो फ्रेमवर्क MhM के संभावित हमलों पर प्रकाश डालता है। अंत में, शोषकता स्कोर का उपयोग करके महत्वपूर्ण पथों की पहचान की जाती है।

संभावित हमले के रास्तों और लॉग से सबूत के बीच समानता सूचकांक सबसे संभावित हमले के परिदृश्य की पहचान करता है जिसका एक विरोधी अनुसरण कर सकता है। LM का उपयोग करते हुए, सार्वजनिक रूप से उपलब्ध डेटासेट OpenStack (Loghub) पर नेटवर्क संचार ओवरहेड 88% तक कम हो गया। LSTM-आधारित विसंगति का पता लगाना 0.6 msec के औसत विषम लॉग भविष्यवाणी के साथ विषम लॉग का पता लगाने में 99% सटीकता दिखाता है। हमने भेद्यता की पूर्वापेक्षाएँ और बाद की स्थितियाँ उत्पन्न करने में क्रमशः 98% और 97% सटीकता हासिल की। हमने स्थानीय टेस्टबेड का उपयोग करके MhMs हमले का पता लगाने में *GLoM* दक्षता का मूल्यांकन किया।

संक्षेप में, इस थीसिस का उद्देश्य MhMs संभावित हमलों की खोज के लिए कुशल भेद्यता प्रबंधन के लिए तकनीकों का निर्माण करना है। यह डोमेन संदर्भ पर विचार करके भेद्यता प्राथमिकता तकनीक प्रदान करता है। यह सिस्टम लॉग का उपयोग करके चल रहे बहु-होस्ट शोषण का पता लगाने के लिए एक वैश्विक निगरानी प्रणाली का भी प्रस्ताव करता है।

# Contents

Certificate	iii
Acknowledgements	v
Abstract	ix
Table of contents	xxiv
List of Figures	xxx
List of Tables	xxxiii
List of Acronyms	xxxv
List of Symbols	xxxix

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Vulnerability management . . . . .	9
1.1.1	Vulnerabilities discovery . . . . .	9
1.1.2	Vulnerability prioritization . . . . .	10
1.1.3	Vulnerability monitoring . . . . .	11
1.2	Challenges of IIoT-based systems . . . . .	11
1.3	Summary of Contributions . . . . .	12
1.3.1	Literature survey & Vulnerability assessment . . . . .	13
1.3.2	Research work . . . . .	14
1.4	Organization . . . . .	16
<b>2</b>	<b>Background &amp; Related work</b>	<b>19</b>
2.1	Basic terminology . . . . .	19
2.2	Vulnerability lifecycle . . . . .	21
2.3	Roadmap of Vulnerability Management in IIoT . . . . .	23
2.3.1	Vulnerability discovery . . . . .	24
2.3.2	Vulnerability analysis . . . . .	28

---

2.3.3	Vulnerability prioritization . . . . .	29
2.3.4	Vulnerability remediation . . . . .	34
2.3.5	Vulnerability verification and monitoring . . . . .	35
2.4	Database information . . . . .	39
2.5	SCADA Architecture . . . . .	42
2.5.1	SCADA Components . . . . .	43
2.5.2	IIoT-based SCADA . . . . .	45
2.5.3	Attacks on SCADA systems . . . . .	46
2.6	Nash equilibrium . . . . .	52
2.7	Summary . . . . .	54
<b>3</b>	<b>Assessment of SCADA System Vulnerabilities</b>	<b>57</b>
3.1	Related Work . . . . .	58
3.2	SCADA-specific Vulnerabilities Extraction . . . . .	59
3.3	SCADA vulnerabilities Assessment . . . . .	60
3.4	Discussion . . . . .	67
3.4.1	Q1. How the attacks on SCADA systems have been carried out? . . .	67

---

3.4.2	Q2. What is the most vulnerable subsystem of the complex SCADA system? . . . . .	69
3.4.3	Q3. What is the impact of current vulnerabilities on the SCADA system? . . . . .	70
3.4.4	Q4. Is CVSS a good vulnerability scoring system for SCADA vulnerabilities? . . . . .	71
3.5	Summary . . . . .	72
<b>4</b>	<b>An E2E Pentest-based Security Analysis Framework</b>	<b>73</b>
4.1	Problem Setup . . . . .	77
4.2	Background . . . . .	80
4.3	Design and implementation . . . . .	82
4.3.1	Design (Target Graph Structure) : . . . . .	83
4.3.2	Implementation . . . . .	85
4.3.2.1	Stage 1 (Pentesting setup installation): . . . . .	85
4.3.2.2	Stage 2 (Get current state information of each node) . . . . .	85
4.3.2.3	Stage 3 (Extract CPE from .xml file generated by Nmap) . . . . .	85

---

4.3.2.4	Stage 4 (Prerequisites and postconditions generation for all the reported vulnerabilities & target-graph generation)	86
4.3.2.5	Stage 5 (Analysis of Target-paths & Recommendations)	88
4.4	Performance & Scalability	94
4.4.1	Finding the critical target-path, critical node and critical vulnerability	99
4.5	Summary	103
<b>5</b>	<b>Patch prioritization frameworks</b>	<b>105</b>
5.1	<i>PatchRank</i>	107
5.1.1	Proposed method	108
5.1.1.1	CVSS scoring system	113
5.1.1.2	Node Rank	114
5.1.1.3	Vulnerability Rank	118
5.1.1.4	Patch Threshold	119
5.1.2	Evaluation	121
5.2	SmartPatch	124

---

5.2.1	Patch prioritization methodology . . . . .	125
5.2.1.1	Step 1: Architectural features identification . . . . .	127
5.2.1.2	Step 2: System vulnerability risk assessment . . . . .	128
5.2.1.3	Step 3: Vulnerability patch prioritization . . . . .	130
5.2.2	Case Study . . . . .	137
5.2.2.1	Need of patch prioritization in SCADA chain in the smart grid . . . . .	138
5.2.2.2	SmartPatch for SCADA chain in smart grid . . . . .	139
5.2.2.3	Evaluation . . . . .	141
5.2.2.4	Comparative analysis of <i>SmartPatch</i> . . . . .	152
5.3	Related Work . . . . .	156
5.4	Summary . . . . .	158
<b>6</b>	<b>Global Monitor using SpatioTemporally Correlated Local Monitors</b>	<b>161</b>
6.1	Introduction . . . . .	161
6.2	Methodology . . . . .	165
6.2.1	Role of LM and GM in offline phase (Figure 6.2) . . . . .	167

---

6.2.1.1	Role of LM . . . . .	167
6.2.1.2	Role of GM . . . . .	170
6.2.2	Role of LM and GM in online phase (Figure 6.5) . . . . .	173
6.2.2.1	Role of LM . . . . .	173
6.2.2.2	Role of GM . . . . .	174
6.2.3	Vulnerability Index . . . . .	177
6.3	Implementation & Evaluation . . . . .	178
6.3.1	Network overhead . . . . .	179
6.3.2	Generation of prerequisites and postconditions . . . . .	182
6.3.3	Case-study . . . . .	185
6.4	Summary . . . . .	189
<b>7</b>	<b>Conclusion and Future Directions</b>	<b>191</b>
7.1	Future Research Directions . . . . .	194
	<b>Bibliography</b>	<b>216</b>
	<b>List of Publications</b>	<b>217</b>

Biography

221

# List of Figures

1.1	SCADA Application Areas . . . . .	3
1.2	IIoT and SCADA . . . . .	4
1.3	Priority order for SCADA and General IT . . . . .	6
1.4	Generic vulnerability management lifecycle (*Extended stage) . . . . .	7
1.5	Summary of thesis contributions . . . . .	13
2.1	Vulnerability Lifecycle . . . . .	22
2.2	Correlation of CVE-CWE-CAPEC database . . . . .	42
2.3	Interrelation of SCADA system components . . . . .	43
2.5	Country vs. Attacks count for the duration [1982, 2015] . . . . .	50
2.6	Sector vs. Attack count for the duration [1982, 2015] . . . . .	50

---

2.7	Categorization of attacks for the duration [1982, 2015] . . . . .	51
3.1	Extracted Parameters from NVD . . . . .	59
3.2	Analysis of Year vs. Number of Vulnerabilities for the SCADA system. From 2010 to 2011, there is a drastic increase in the number of vulner- abilities reported due to the connectivity of SCADA systems to external networks over the Internet. After 2015, with IIoTization and Industry 4.0, the attack surface in the SCADA system increased rapidly. . . . .	60
3.3	Analysis of Attack vector vs. Number of vulnerabilities. The maximum number of vulnerabilities (319) have been reported on the network due to insecure communication protocols. . . . .	61
3.4	Availability impact score over the years. Availability impact is categorized into three classes (Complete, Partial and None). . . . .	62
3.5	Confidentiality impact score over the years. Confidentiality impact is cat- egorized into three classes (Complete, Partial and None). . . . .	63
3.6	Integrity impact score over the years. Integrity impact is categorized into three classes (Complete, Partial and None). . . . .	63
3.7	Count vs CWE-Number. The top-3 CWE reported over the years are CWE-119, CWE-20 and CWE-22. . . . .	64
3.8	Top-3 CWE analysis . . . . .	66

---

3.9	Severity analysis of vulnerabilities . . . . .	67
4.1	Overview of End-to-End IoT system . . . . .	75
4.2	Multi-host, multi-stage vulnerability exploitation example with pre and postcondition . . . . .	78
4.3	Target-paths for example setup shown in Figure 4.2 . . . . .	79
4.4	IoT-PEN stages . . . . .	82
4.5	Network node and edge template . . . . .	83
4.6	Vulnerability template . . . . .	84
4.7	Current state information collection using MQTT . . . . .	86
4.8	Running Time vs. Number of nodes (Number of vulnerabilities on each node = 5) . . . . .	96
4.9	Running time of IoT-PEN vs. Number of vulnerability ( Number of nodes = 5) . . . . .	97
4.10	Total Target-paths to the root node generated vs. Number of nodes ( Number of vulnerabilities on each node = 5) . . . . .	98
4.11	IIoT Example Network . . . . .	100
4.12	A part of Target graph of IIoT example network . . . . .	101

---

4.13	Various IoT topologies . . . . .	102
5.1	Viable system model for SCADA (Spyridopoulos et al., 2017) . . . . .	109
5.2	Graphical SCADA Model . . . . .	110
5.3	SCADA viable model with two PLC (S11, S12) and one MTU (S13), S2 (SCADA Server), S3 (HMI), S3* (Control system), S4 (R*D), S5 (Man- agement), E (Environment) with weighted interconnection . . . . .	121
5.4	SmartPatch framework . . . . .	126
5.5	Viable SCADA Model for SCADA supply chain in smart grid . . . . .	137
5.6	Case study for SCADA chain in Smart grid . . . . .	140
5.7	Functional dependency model for case study shown in Figure 5.6 . . . . .	142
5.8	External attacker’s strategies . . . . .	148
5.9	Expected impact variation with an increasing number of defenders for each stage . . . . .	149
5.10	Value of game variation against external attackers for smart grid case study	150
5.11	Expected impact variation for defending against internal attackers in smart grid case study . . . . .	151
5.12	Residual impact score variation of SmartPatch vs. Defender resources . . .	153

---

5.13	Residual impact score variation of SmartPatch vs. Round number by varying number of defenders . . . . .	154
5.14	Residual impact score variation of CVSS, SecureRank, VulCon and SmartPatch . . . . .	155
6.1	Temporal and spatial correlation. LM is responsible to detect anomalous behavior on a single system while GM combines LM reports and generates the current security state. . . . .	166
6.2	Role of local and global monitor in offline mode . . . . .	168
6.3	LSTM model used in GLoM . . . . .	169
6.4	Extraction of CVE-CAPEC-sigma-rules relationship. Generally, for a CVE there is a one-to-one mapping of CWE, while for a CAPEC there is one-to-many CWE mapping. Therefore, for a CVE, there can be multiple ways to exploit and we embed all the attack patterns to that vulnerability. . . .	172
6.5	Role of the local and global monitor in the online phase . . . . .	174
6.6	Logs to CVE mapping by applying two filters i.e. Application name, Attack vector . . . . .	175
6.7	CVE to syslog mapping for CVE-2020-25196 on a Linux system . . . . .	176
6.8	Accuracy and Recall over OpenStack database using FTW and SSW . . . .	181

6.9 a) Anomalous score of system logs time period over time perceived by  
Global Monitor (grouped over 1 min) b) AEI variation for S1 and S2. . . . 187

6.10 Real exploitation scenario generation using possible attack graph and Ev-  
idence list for the given case study . . . . . 188

# List of Tables

2.1	Vulnerability discovery: Summary of the Related Work. . . . .	27
2.2	Vulnerability prioritization: Summary of the Related Work. . . . .	33
2.3	Vulnerability monitoring: Summary of the Related Work. . . . .	38
2.4	Some of the Important Attacks during 1982-2016 . . . . .	47
3.1	Vulnerability to the SCADA systems . . . . .	65
3.2	Severity distribution using Base score . . . . .	66
4.1	Rules for generating prerequisites using locality, authentication, privilege, CPE - O (Operating System), A (Application) . . . . .	87
4.2	Rules for generating prerequisites using the description field (A-Application)	88

4.3	Rules for generating postconditions using the NVD dataset. CPE - O (Operating System), A (Application) , Impact score ( 5.9 (All CIA impact value), [3.4 - 5.9))) (Partial CIA impact), <3.4 (None CIA impact) ) . . . . .	89
4.4	Time elapsed in various IoT-PEN stages . . . . .	95
5.1	Adjacency matrix representation of SCADA ( $G_{adj}$ ) graphical model . . . . .	111
5.2	Strategy for selecting the most vulnerable node . . . . .	115
5.3	Selection of the most vulnerable node (Mixed Nash Equilibrium) . . . . .	116
5.4	Game theory strategy for selecting the vulnerability to patch . . . . .	118
5.5	Comparison of patching order using CVSS and <i>PatchRank</i> . . . . .	122
5.6	Vulnerability prioritization using the node importance . . . . .	123
5.7	Payoff matrix for $ S_D  = 2,  S_A  = 2$ . . . . .	135
5.8	Vulnerability metrics calculation for case study (Figure 5.6) (MU: Mitigation-utility, NV: $NV_{chrono}$ ), RS: $Risk_{score}$ , ST: $S_{TDep}^s$ , SF: $S_{FunDep}^s$ , V: $V_{chrono}$ . . . . .	143
5.9	External attacker's strategy for a case study in Figure 5.8 . . . . .	147
5.10	Defender's strategies for each SCADA system in smart grid (Without considering $S_{TDep}$ ) . . . . .	149

---

5.11 Defender’s strategies for each SCADA system in smart grid for defending against external attackers . . . . .	150
5.12 Defender’s strategies for each SCADA system in smart grid for defending against internal attackers . . . . .	152
5.13 Comparison with related work . . . . .	157
6.1 Time elapsed in various stages . . . . .	182
6.2 Accuracy of ML algorithms for generating prerequisites and postconditions	184
6.3 Confusion matrix for generating prerequisites and postconditions using RF	184
6.4 Vulnerabilities details . . . . .	186

# List of Acronyms

**AC** Access Complexity

**AEI** Actual Exploit Index

**Au** Authentication

**AV** Access Vector

**BC** Betweenness Centrality

**CAPEC** Common Attack Pattern Enumeration and Classification

**CI** Critical Infrastructure

**CMS** Central Monitoring Systems

**CPE** Common Platform Enumeration

**CVE** Common Vulnerability Enumeration

**CVE-ID** Common Weakness Enumeration Identifier

**CVSS** Common Vulnerability Scoring System

**CWE** Common Weakness Enumeration

**DER** Distributed Energy Resource

**DoS** Deniel of service

**EL** Evidence List

**FTW** Fixed Timing Window

**GM** Global Monitor

**GUI** Graphical User Interface

**ICS** Industrial Control System

**IDS** Intrusion Detection Systems

**IIoT** Industrial Internet of Things

**IoT** Internet of Things

**IT** Information Technology

**KNC** K-Neighbors Classifier

**LAN** Local Area Network

**LM** Local Monitor

**LCS** Lowest Common Subsequence

**LR** Logistic Regression

**LSTM** Long-Short Term Memory

**LVI** Local Vulnerability Index

**MhMs** Multi-host Multi-stage

**ML** Machine Learning

**MLP** Multi-layer Perceptron classifier

**NCC** Nearest Centroid Classifier

**NVD** National Vulnerability Database

**OT** Operational Technology

**PAG** Possible Attack Graph

**PR** Privilege Required

**RAPR** Repository for Attack Pattern Rules

**RF** Random Forest classifier

**RIS** Residual Impact Score

**RISI** Repository of Industrial Security Incidents

**Sc** Scope

**s.t.** such that

**SCADA** Supervisory Control and Data Acquisition

**SGDC** Stochastic Gradient Descent Classifier

**SSW** Session Specific Window

**TCP** Transport Layer Protocol

**TTC** Time-To-Compromise

**UDP** User Datagram Protocol

**UI** User Interaction

**VM** Vulnerability Management

**VSM** Viable System Model

**VSMS** Viable System Model for SCADA

**w.r.t.** with respect to

**WIVSS** Weighted Impact Vulnerability Scoring System

**WAN** Wide Area Network

# List of Symbols

$A^v$  Availability impact of vulnerability  $v$

$AC^v$  Attack complexity of vulnerability  $v$

$AV^v$  Attack vector of vulnerability  $v$

$B_{Score}^v$  Base score of vulnerability (Basescore parameter of CVSS)

$C_A^v$  Cost of attacking a vulnerability  $v$

$C_D^v$  Cost of defending a vulnerability  $v$

$C^v$  Confidentiality impact of vulnerability  $v$

$CVSS_{v_i}$  CVSS severity score of vulnerability  $v_i$

$EI_A(j, k)$  Effective impact of attack when attacker chose strategy  $j$  and defender chose strategy  $k$

$EPr_A(j, k)$  Effective profit of attack when attacker chose strategy  $j$  and defender chose strategy  $k$

$Vul_{exploit}^v$	Exploit score of vulnerability v
$ESC$	Exploitability subscore of vulnerability v
$G_{adj}$	Adjacency matrix of VSMS graphical model
$I_A^v$	Impact of successful exploitation a vulnerability v
$I^v$	Integrity impact of vulnerability v
$I$	Inspection space
$ISC$	Impact subscore of vulnerability v
$K$	Log Key set
$N_{rank}$	Node rank for patching
$N_{topology}$	Network topology
$N_A$	Number of attackers
$N_D$	Number of defenders
$N_V$	Number of vulnerabilities
$NV_{chrono}$	Normalised chronological age of vulnerability v
$P_A^j$	Probability of attacking systems using strategy j
$P_A^v$	Probability of successful attack on vulnerability v
$P_A$	Probability of successful attack

$P_D^k$  Probability of defending system using strategy k  
 $P_D^v$  Probability of defending vulnerability v  
 $P_D$  Probability of successful defend  
 $Payoff_A$  Payoff of attacker  
 $Payoff_A^i$  Payoff of the attacker in  $i^{th}$  iteration  
 $Payoff_D$  Payoff of defender  
 $Payoff_D^i$  Payoff of the defender in  $i^{th}$  iteration  
 $PO_A$  Expected payoff of attacker  
 $PO_D$  Expected payoff of defender  
 $Pr_A^v$  Profit gained by attacker by successful exploitation of a vulnerability v  
 $PR^v$  Privilege associated with vulnerability v  
 $R_{adj}^s$  Risk associated with a subsystem s  
 $R_{adj}$  Associated Risk  
 $R_A$  Resources available to attacker  
 $R_D$  Resources available to defender  
 $R$  Set of rules to detect a CVE exploitation  
 $Re_A^v$  Revenue generated by attacker by successful exploitation of a vulnerability v

$RIS^t$  Residual impact score after iteration  $t$   
 $S_{FunDep}^s$  Functional dependency score of subsystem  $s$   
 $S_{TDep}^s$  Topological dependency score of subsystem  $s$   
 $s_A^j$   $j^{th}$  strategy of attacker  
 $s_D^k$   $k^{th}$  strategy of defender  
 $Sc$  Set of attack scenarios  
 $Vul_{severity}^v$  Severity Score of vulnerability  $v$   
 $S1, S2, S3, S3^*, S4, S5$  System instances  
 $t_1, t_2, t_3$  Time instances  
 $UI^v$  User interaction of vulnerability  $v$   
 $V_{chrono}$  Chronological age of a vulnerability  $v$   
 $V_{rank}$  Vulnerabilities rank for patching  
 $v_i$  Vulnerability instance  $\in V$   
 $V$  Vulnerability set  
 $W$  Node severity matrix