CYBER 2024

The Ninth International Conference on Cyber-Technologies and Cyber-Systems

ISBN: 978-1-68558-186-2

September 29 - October 03, 2024

Venice, Italy

CYBER 2024 Editors

Tiago Gasiba, Siemens AG, Munich, Bavaria, Germany

Joshua Sipper, Air University, Air Command and Staff College, USA

CYBER 2024

Forward

The Ninth International Conference on Cyber-Technologies and Cyber-Systems (CYBER 2024), held between September 29th, 2024, to October 3rd, 2024, in Venice, Italy, continued a series of international events covering many aspects related to cyber-systems and cyber-technologies; it was also intended to illustrate appropriate current academic and industry cyber-system projects, prototypes, and deployed products and services.

The increasing size and complexity of the communications and the networking infrastructures are making difficult the investigation of resiliency, security assessment, safety and crimes. Mobility, anonymity, counterfeiting, are characteristics that add more complexity in Internet of Things and Cloud-based solutions. Cyber-physical systems exhibit a strong link between the computational and physical elements. Techniques for cyber resilience, cyber security, protecting the cyber infrastructure, cyber forensic, and cyber-crimes have been developed and deployed. Some new solutions are nature-inspired and social-inspired, leading to self-secure and self-defending systems. Despite the achievements, security and privacy, disaster management, social forensics, and anomalies/crimes detection are challenges within cyber-systems.

We take here the opportunity to warmly thank all the members of the CYBER 2024 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to CYBER 2024. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the CYBER 2024 organizing committee for their help in handling the logistics of this event.

We hope that CYBER 2024 was a successful international forum for the exchange of ideas and results between academia and industry for the promotion of progress related to cyber-technologies and cyber-systems.

CYBER 2024 Chairs

CYBER 2023 Steering Committee Chair
Steve Chan, Decision Engineering Analysis Laboratory, USA

CYBER 2024 Steering Committee
Carla Merkle Westphall, UFSC, Brazil
Barbara Re, University of Camerino, Italy
Rainer Falk, Siemens AG, Corporate Technology, Germany
Daniel Kästner, AbsInt GmbH, Germany
Anne Coull, Flinders University, Adelaide, Australia
Steffen Fries, Siemens, Germany
Sibylle Fröschle, TU Hamburg, Germany
Andreas Aßmuth, Fachhochschule Kiel, Germany

CYBER 2024 Publicity Chairs
Laura Garcia, Universidad Politécnica de Cartagena, Spain
Lorena Parra Boronat, Universidad Politécnica de Madrid, Spain

CYBER 2024
Committee

CYBER 2023 Steering Committee Chair

Steve Chan, Decision Engineering Analysis Laboratory, USA

CYBER 2024 Steering Committee

Carla Merkle Westphall, UFSC, Brazil
Barbara Re, University of Camerino, Italy
Rainer Falk, Siemens AG, Corporate Technology, Germany
Daniel Kästner, AbsInt GmbH, Germany
Anne Coull, Flinders University, Adelaide, Australia
Steffen Fries, Siemens, Germany
Sibylle Fröschle, TU Hamburg, Germany
Andreas Aßmuth, Fachhochschule Kiel, Germany

CYBER 2024 Publicity Chairs

Laura Garcia, Universidad Politécnica de Cartagena, Spain
Lorena Parra Boronat, Universidad Politécnica de Madrid, Spain

CYBER 2024 Technical Program Committee

Aysajan Abidin, imec-COSIC KU Leuven, Belgium
Shakil Ahmed, Iowa State University, USA
Cuneyt Gurcan Akcora, University of Manitoba, Canada
Oum-El-Kheir Aktouf, Grenoble Institute of Technology, France
Abdullah Al-Alaj, Virginia Wesleyan University, USA
Khalid Alemerien, Tafila Technical University, Jordan
Usman Ali, University of Connecticut, USA
Aisha Ali-Gombe, Towson University, USA
Anas AlSobeh, Southern Illinois University, USA
Alina Andronache, University of West of Scotland, UK
Abdullahi Arabo, University of the West of England, UK
A. Taufiq Asyhari, Coventry University, UK
Syed Badruddoja, University of North Texas, USA
Morgan Barbier, ENSICAEN, France
Samuel Bate, EY, UK
Vincent Beroulle, Univ. Grenoble Alpes, France
Clara Bertolissi, Aix-Marseille University | LIS | CNRS, France
Khurram Bhatti, Information Technology University (ITU), Lahore, Pakistan
Michael Black, University of South Alabama, USA
Davidson R. Boccardo, Clavis Information Security, Brazil
Felix Boes, University of Bonn, Germany
Ravi Borgaonkar, SINTEF Digital / University of Stavanger, Norway

Florent Bruguier, LIRMM | CNRS | University of Montpellier, France
Enrico Cambiaso, Consiglio Nazionale delle Ricerche (CNR), Italy
Nicola Capodieci, University of Modena and Reggio Emilia (UNIMORE), Italy
Pedro Castillejo Parrilla, Technical University of Madrid (UPM), Spain
Steve Chan, Decision Engineering Analysis Laboratory, USA
Christophe Charrier, Normandie Universite, France
Bo Chen, Michigan Technological University, USA
Mingwu Chen, Langara College, Canada
Lu Cheng, ArizonaState University, USA
Ioannis Chrysakis, FORTH-ICS, Greece / Ghent University, Belgium
Anastasija Collen, University of Geneva, Switzerland
Giovanni Costa, ICAR-CNR, Italy
Domenico Cotroneo, University of Naples, Italy
Anne Coull, Flinders University, Adelaide, Australia
Monireh Dabaghchian, Morgan State University, USA
Dipanjan Das, University of California, Santa Barbara, USA
João Paulo de Brito Gonçalves, Instituto Federal do Espírito Santo, Brazil
Vincenzo De Angelis, University of Reggio Calabria, Italy
Noel De Palma, University Grenoble Alpes, France
Luigi De Simone, Università degli Studi di Napoli Federico II, Italy
Jerker Delsing, Lulea University of Technology, Sweden
Patrício Domingues, Polytechnic Institute of Leiria, Portugal
Paul Duplys, Robert Bosch GmbH, Germany
Soultana Ellinidou, Cybersecurity Research Center | University Libre de Bruxelles (ULB), Belgium
Rainer Falk, Siemens AG, Corporate Technology, Germany
Omair Faraj, Internet Interdisciplinary Institute (IN3) | UOC, Barcelona, Spain
Yebo Feng, University of Oregon, USA
Eduardo B. Fernandez, Florida Atlantic University, USA
Steffen Fries, Siemens Corporate Technologies, Germany
Somchart Fugkeaw, Thammasat University, Thailand
Damjan Fujs, University of Ljubljana, Slovenia
Steven Furnell, University of Nottingham, UK
Gina Gallegos Garcia, Instituto Politécnico Nacional, Mexico
Tiago Gasiba, Siemens AG, Germany
Huangyi Ge, Purdue University, USA
Kambiz Ghazinour, SUNY Canton, USA
Konstantinos Giannoutakis, University of Macedonia, Greece
Uwe Glässer, Simon Fraser University - SFU, Canada
Ruy Jose Guerra Barretto de Queiroz, Federal University of Pernambuco, Brazil
Ekta Gujral, Walmart Global Tech, USA
Chunhui Guo, San Diego State University, USA
Amir M. Hajisadeghi, AmirkabirUniversity of Technology, Iran
Arne Hamann, Robert Bosch GmbH, Germany
Ehsan Hesamifard, University of North Texas, USA
Gahangir Hossain, West Texas A&M University, Canyon, USA
Mehdi Hosseinzadeh, Washington University in St. Louis, USA
Zhen Huang, DePaul University, USA
Maria Francesca Idone, University of Reggio Calabria, Italy

Christos Iliou, Information Technologies Institute | CERTH, Greece / Bournemouth University, UK
Shalabh Jain, Research and Technology Center - Robert Bosch LLC, USA
Kevin Jones, University of Plymouth, UK
Georgios Kambourakis, University of the Aegean - Karlovassi, Samos, Greece
Sayar Karmakar, University of Florida, USA
Dimitris Kavallieros, ITI-CERTH, Greece
Saffija Kasem-Madani, University of Bonn, Germany
Daniel Kästner, AbsInt GmbH, Germany
Basel Katt, Norwegian University of Science and Technology (NTNU), Norway
Mazaher Kianpour, Norwegian University of Science and Technology, Norway
Lucianna Kiffer, Northeastern University, USA
Sotitios Kontogiannis, University of Ioannina, Greece
Tanya Koohpayeh Araghi, University Oberta de Catalunya, Spain
Dragana S. Krstic, University of Nis, Serbia
Fatih Kurugollu, University of Derby, UK
Cecilia Labrini, University of Reggio Calabria, Italy
Ruggero Lanotte, University of Insubria, Italy
Rafał Leszczyna, Gdansk University of Technology, Poland
Eirini Liotou, National and Kapodistrian University of Athens, Greece
Jing-Chiou Liou, Kean University - School of Computer Science and Technology, USA
Hao Liu, University of Cincinnati, USA
Xing Liu, Kwantlen Polytechnic University, Canada
Qinghua Lu, CSIRO, Australia
Yi Lu, Queensland University of Technology, Australia
Mahesh Nath Maddumala, Mercyhurst University, Erie, USA
Jorge Maestre Vidal, Universidad Complutense de Madrid, Spain
Louai Maghrabi, Dar Al-Hekma University, Jeddah, Saudi Arabia
Yasamin Mahmoodi, Tübingen University | FZI (Forschungszentrum Informatik), Germany
David Maimon, Georgia State University, USA
Ivan Malakhov, Università Ca’ Foscari Venezia, Italy
Timo Malderle, University of Bonn, Germany
Mahdi Manavi, Mirdamad Institute of Higher Education, Iran
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Michael Massoth, Hochschule Darmstadt - University of Applied Sciences / CRISP – Center for Research in Security and Privacy, Darmstadt, Germany
Vasileios Mavroeidis, University of Oslo, Norway
Mohammadreza Mehrabian, University of the Pacific, USA
Weizhi Meng, Technical University of Denmark, Denmark
Carla Merkle Westphall, UFSC, Brazil
Massimo Merro, University of Verona, Italy
Caroline Moeckel, Open University, UK
Lorenzo Musarella, University Mediterranea of Reggio Calabria, Italy
Vasudevan Nagendra, Stony Brook University, USA
Roberto Nardone, University Mediterranea of Reggio Calabria, Italy
Niels Nijdam, University of Geneva, Switzerland
Klimis Ntalianis, University of West Attica, Greece
Jason Nurse, University of Kent, UK
Riccardo Ortale, Institute for High Performance Computing and Networking (ICAR) of the National

Research Council of Italy (CNR), Italy
Jordi Ortiz, University of Murcia, Spain
Richard E. Overill, King's College London, UK
Mohammad Zavid Parvez, Charles Sturt University, Australia
Antonio Pecchia, University of Sannio, Italy
Eckhard Pfluegel, Kingston University, London, UK
Mila Dalla Preda, University of Verona, Italy
Muhammad Haris Rais, Virginia Commonwealth University, USA
Paweł Rajba, Hitachi Energy / University of Wroclaw, Poland
Massimiliano Rak, Università della Campania, Italy
Alexander Rasin, DePaul University, USA
Danda B. Rawat, Howard University, USA
Barbara Re, University of Camerino, Italy
Leon Reznik, Rochester Institute of Technology, USA
Jan Richling, South Westphalia University of Applied Sciences, Germany
Giulio Rigoni, University of Florence / University of Perugia, Italy
Antonia Russo, University Mediterranea of Reggio Calabria, Italy
Peter Y. A. Ryan, University of Luxembourg, Luxembourg
Asanka P. Sayakkara, University of Colombo School of Computing (UCSC), Sri Lanka
Florence Sedes, Université Toulouse 3 Paul Sabatier, France
Abhijit Sen, Kwantlen Polytechnic University, Canada
Shirin Haji Amin Shirazi, University of California, Riverside, USA
Srivathsan Srinivasagopalan, AT&T CyberSecurity (Alien Labs), USA
Zhibo Sun, Drexel University, USA
Ciza Thomas, Government of Kerala, India
Zisis Tsiatsikas, Atos Greece / University of the Aegean, Greece
Tobias Urban, Institute for Internet Security - Westphalian University of Applied Sciences, Gelsenkirchen, Germany
Eric MSP Veith, OFFIS e.V. - Institut für Informatik, Germany
Mudit Verma, Arizona State University, Tempe, USA
Simon Vrhovec, University of Maribor, Slovenia
Stefanos Vrochidis, ITI-CERTH, Greece
James Wagner, University of New Orleans, USA
Gang Wang, Emerson Automation Solutions, USA
Qi Wang, Stellar Cyber Inc., USA
Ruoyu "Fish" Wang, Arizona State University, USA
Zhiyong Wang, Utrecht University, Netherlands
Zhen Xie, JD.com American Technologies Corporation, USA
Cong-Cong Xing, Nicholls State University, USA
Ping Yang, State University of New York at Binghamton, USA
Wuu Yang, National Chiao-Tung University, HsinChu, Taiwan
George O. M. Yee, Aptusinnova Inc. & Carleton University, Ottawa, Canada
Serhii Yevseiev, National Technical University - Kharkiv Polytechnic Institute, Ukraine
Kailiang Ying, Google, USA
Wei You, Renmin University of China, China
Yicheng Zhang, University of California, Irvine, USA
Piotr Zwierzykowski, Poznan University of Technology, Poland

Copyright Information

Table of Contents

Artificial Intelligence and Semiotics: Using Semiotic Learning to Bolster AI

Joshua A. Sipper
Air Command and Staff College
Air University
Maxwell AFB, AL, United States
Email: joshua.sipper.1@us.af.mil

***Abstract*— Artificial Intelligence (AI) continues to grow into areas such as Natural Language Processing (NLP) and other arenas where understanding the meaning behind speech, images, and symbols is of increasing importance, such as education. These realms of understanding are not merely semantic, but semiotic in scope, carrying with them the potential for AI to grow toward the understanding of the meaning behind what is being said, written, pictured, or symbolized. This growth necessitates the advancement of techniques in semiotic learning such as neural sketch learning, paradigmatic associations, and advanced heuristics. In this paper, semiotic learning will be defined and discussed. Additionally, some techniques and strategies for AI semiotic learning will be discussed and modeled including an AI accommodation/assimilation model. These strategies are ultimately useful for defensive cyber operations and offensive cyber operations through the use of semiotic context matching to improve defensive and offensive strategies.**

Keywords- cyber; artificial intelligence; semiotics; accommodation; assimilation; heuristics.

I. INTRODUCTION

As the world of Artificial Intelligence (AI) continues to expand exponentially, the need for these technologies to comprehend various types of discrete and esoteric information grows as well. AI is one of those terms that is often confusing, with many people inferring that it means only one thing. The reality, however, is that AI expresses itself in at least three major ways: semantic, semiotic, and singular. The semantic characterization deals primarily with data associations with some decision functions at a basic level, absent of the true "understanding" of how those data interrelate. Semiotic relationships include much of the data association methodologies in semantic AI but work toward helping neural networks to make idiomatic associations for a level of networked "meaning" regarding the data within a particular system or framework. The "singular" or "singularity" refers to machine consciousness with a full understanding of meaning, relationships, idioms, and feelings similar in scope to how human beings navigate stimuli, data, and personal emotions. This discussion will focus primarily on the semiotic realm of AI with some explanation of semantic AI for differentiation. These strategies are ultimately useful for defensive cyber operations and offensive cyber operations through the use of semiotic context matching to improve defensive and offensive strategies. The rest of the paper is structured as follows. In Section II, computational semiotics will be defined. Section III details neural sketch learning for semiosis. Paradigmatic associations will be discussed in Section IV to give a solid frame of reference for how to translate paradigms into code and algorithms. Section V deals with advanced heuristics or mental shortcuts and how these can be used to improve efficiency and understanding in algorithmic applications. The AI accommodation and assimilation model will be detailed in Section VI followed by the conclusion in Section VII.

II. DEFINING COMPUTATIONAL SEMIOTICS

Before one can embark upon sifting through the complex territory of semiotic support for AI, a full understanding of computational semiotics is necessary. Semiosis is the "study of meaning and communication processes…from the point of view of formal sciences, linguistics, and philosophy" including the situational control of logical systems to produce automatic control of systems [1]. This definition flows directly into the fundamental premise of AI to provide meaningful and intelligible information to humans for use in understanding and using information more rapidly while also making improved predictions concerning numerous systems and data. Semiotics is the study of signs and symbols, especially as a means of language or communication. It is a multidisciplinary perspective that incorporates the thinking and theory fragments of many different thinkers, including linguists, phenomenologists, and philosophers. The foundation of semiotic computational relationships was developed in the 1970s by Russian AI researcher D.A. Pospelov who connected AI theory to human reasoning through studying two models: a deductive "maze model" and an inductive "chess model" [2]. The "maze model" was an earlier concept developed in the 1950s drawing on the work of cognitive psychologists who based human thought on the premise of linear decision-making.

However, the maze hypothesis began to fall into dispute as it came under increased scrutiny, leading to the more inductive "chess model" which offered more probability across human thought and meaning construction. This divergence of human and computer thought is captured well by Deb Roy in her study concerning schema theory and

ISBN: 978-1-68558-186-2

semiotics. Human beings generally construct meaning through forming concepts around ideas in scaffolds called “schemas” which are used constantly to produce and express ideas between humans that are filled with meaning [3]. This leads to “a causal-predictive cycle of action and perception” where people create meaning and share complex relationships about events, places, and numerous other value associations [3]. While numerous methodologies for meaning-making have been explored in recent years, the use of automatic reasoning through pattern recognition has shown a great deal of promise [4]. This method of semiotic computing is based on systems that learn paradigms, which are then transformed into new symbols on which syntagmatic and paradigmatic analysis can be performed again [4]. This analysis and reanalysis of concepts, words, paradigms, and schemas leads to semiotic functions that can then be bolstered through iterative processing and syntactic connection scaffolding. This spiral model allows for the exponential reinforcement of syntax and paradigms similar to how humans learn and integrate knowledge and complex information through language using accommodation and assimilation. Accommodation can best be described as the changing of one’s knowledge schema to accommodate new information. Whereas assimilation is the changing of the information being adopted into one’s knowledge schema to more easily fit current information and understanding. This action of assimilation and accommodation is performed primarily through symbolic understanding.

Humans are very skilled at grouping information into semiotic symbolic databases within our schemata allowing us to rapidly understand and predict numerous, complex circumstances [5]. Generally, this process is seen as intuitive and related to human ability to make predictive decisions. This is commonly based on probabilistic decision-making similar to the processes found in quantum computers [6]. Taking this intuitive approach and combining it with symbolic reasoning, has the potential to transform semantic reasoning machines into semiotic reasoning machines capable of understanding symbolic meaning and constructing that meaning into higher order understanding and schemas [5]. If AI is to become the powerhouse it is meant to be, semantic constructions must be transformed into semiotic scaffolds. While Generative Pre-trained Transformers (GPTs) and Generative Adversarial Networks (GANs) will continue to be useful for general understanding and information construction, semiotic AI offers a bridge into a world where Artificial General Intelligence (AGI) is a reality. With this capability within the reach of human use and development, humanity is poised to see enormous leaps in rapidity of decision-making and predictive analysis. Through these predictive algorithms and with the use of more rapid decision-making as a result, cyber offense and defense will be transformed in rapidity, accuracy, and utility.

III. NEURAL SKETCH LEARNING FOR SEMIOSIS

If computational semiotics are the connection between human schematic symbolic understanding, neural sketch learning is the network of pathways mapped out in the most anthropomorphic sense. Human beings are naturally suited to making connections semiotically that allow us to reason and connect disparate data seamlessly. However, computer systems are not innately so capable of tying concepts together as humans. This is where the strategy of neural sketch learning can potentially undergird semiotic learning processes for AI systems. In the following discussion, learning compositional rules through neural program synthesis will be explored to gain insight into how neural sketch learning may be used to support symbolic information scaffolding. Also, neural models for Natural Language Processing (NLP) will be discussed in reference to how we might use neural sketches to bridge the gaps between semantic NLP and semiotic processing with symbolic sketch models. Next, learning to infer program sketches will be examined for information regarding inference-based programs and methods that can be used semiotically for higher-level computational understanding. Then, neural sketch learning for conditional program generation will be analyzed to reveal how conditional programs can be leveraged more robustly through neural sketch learning for symbolic understanding and reasoning in emerging semiotic processing. Finally, idiomatic synthesis and parsing will be defined and discussed to make connections between human idiomatic understanding and teaching semantic and semiotic processing platforms how to recognize and connect these difficult, esoteric linguistic obstacles.

When human beings reason and connect thoughts and patterns, we often tend to draw on connected meaning between numerous words and meaning constructs. For example, when two people talk about growing up in different places with different parents, and different siblings, they automatically form meaningful constructs around the people and places being represented by the other person in the conversation based on meaning constructs held within their own emotional, social, and cultural schemata. This act of meta-cognition or thinking about thinking, allows each person to accommodate and assimilate information about the other person’s experiences and emotions. In semantic relationships using NLP, these relationships, as complex as they are, are not present. One way to potentially address this lack of connection is through what can be referred to as “meta-grammar” and “meta-learning [7].” These learning methodologies for semantic systems have the propensity to learn entire rule systems from examples. In other words, schemas can be generated and then taught to these learning systems to promote and sustain semiotic connections to form meaning for semiotic reasoning. Meta-learning or learning about learning is supported through scaffolding and explaining informational attributes and connected concepts within semantic programs with NLP. This is further supported through meta-grammar or grammar about grammar that takes conceptual, holistic frameworks and teaches them to semantic systems to build and bolster discrete connections between words, phrases, and concepts [7].

Part of the roadblock to semiotic expression in computing frameworks has been the inability of these systems to parse and understand formal language that is not machine-friendly [8]. To avoid this issue, natural language interfaces have been developed to assist in linguistic connections between humans and machines. What machines often do not grasp is the underlying meaning of human expression, however, though symbolically modeling these expressions and building appliances in code using algorithmic means, a functionally semiotic interpretation could be manifested [8]. This method, overlaid with program synthesis [7] reveals promise toward semiosis due to its NLP underpinnings; namely the ability to take multiple complex concepts and scaffold them together to promote language accommodation and assimilation.

Another area inherently connected to semiotic processing is the ability for machine systems to infer meaning from data provided by human input. As humans, we are able to take large sets of disparate and convoluted data and draw inferences from that data to synthesize and explain new data. This complex cognitive computation is part of the non-linear capability human beings have developed through communication with other humans and experiences in the natural world that present a particular survival advantage. However, these inferences are not intrinsic to machine systems due to several reasons including complexity and the lack of imprint of human survival instincts on these systems. Nye, et. al., propose “a system which mimics the human ability to dynamically incorporate pattern recognition and reasoning to solve programming problems from examples or natural language specification [9]. It is this ability to recognize and respond to patterns and symbolic meaning that lends itself to semiosis. Through understanding and connecting information at the meta level, machines can address the gaps found in semantic processing where machines cannot understand the multiple levels of connected meaning innately found through human cognition.

Semantic programming through syntactical interpretation has traditionally been a roadblock to conditional program generation, but great strides have been made in recent years toward semantic and potential semiotic solutions. Through leveraging combinatorial and neural techniques, conditional programs could make the leap toward human-like decisions and predictions by creating and synthesizing language and patterns such that rapid processing of large and complex data sets will be tenable [10]. This is accomplished through neural sketches that combine numerous data attributes and connections allowing for program synthesis of data for program generation [10]. This capability can be further supported using programs like SKETCHADAPT, which are useful for data and program synthesis [9]. Ultimately, the goal is to flow together methods and tools that can bolster complex data combinations for potential semiosis.

One of the most interesting expressions of human thought and experience is our linguistic propensity for creating idioms. An idiom is usually a word or phrase that is directly tied to a social or cultural concept. For instance, in the United States people often use idioms taken from American sports like football and baseball. If someone says that you “hit a home run” or “knocked that one out of the park” they are complimenting you on a great success since these phrases are ties to the concept of great success in the game of baseball in America that allows the team to score points and win the game. However, terms like this are not just a potential language barrier between different cultures, but also present numerous difficulties for semantic AI. The idiomatic problem is in fact even more pronounced in machine systems since idioms are a peculiarly human way of communication. A potential solution for this issue is to incorporate high-level and low-level reasoning at every step of the linguistic translation process in semantic programming and NLP [11]. While most idiomatic interpretations in code language are accomplished through the use of tokens that symbolically represent common patterns, human beings use high-level and low-level reasoning to ascertain idiomatic language, formulate definitions and predictions, and select appropriate linguistic responses. Shin, et. al., suggest alleviating this issue by “interleav[ing] high-level idioms with low-level tokens at all levels of program synthesis, generalizing beyond fixed top-level sketch generation [11].” This method of neural sketch learning offers the advantage of incorporating complex levels of language interpretation and generation to assist in translating language semantically within AI systems. This level of understanding and meaning making has great potential in further supporting semiotic AI translation and generation.

Neural sketch learning offers numerous opportunities to support and grow AI toward semiosis by allowing for the synthesis and translation of large, complex, and esoteric datasets. Compositional rules are a first step toward forming a linguistic foundation for semiotic learning and processing. These rules can then be expanded into learning models using neural interfaces and sketches for cognitive modeling like how humans accommodate and assimilate information. The next level of semiotic understanding can be supported by the ability of semiotic machines to infer data and meaning from large and complex datasets for the purpose of understanding seemingly disparate information and making inferred connections for complex language processing and generation. From these large datasets, cyber defense and offense can also be bolstered through the rapid association of threats and vulnerabilities to identify and assess cyber risks and offensive cyber opportunities. Conditional program generation adds another layer to the foundation of semiosis using neural sketch learning for making conditional connections. Finally, idiomatic understanding and processing through low-level and high-level reasoning bolsters semiosis through supporting AI synthesis of language based on a wide array of human social and cultural understandings. All of these neural processing and learning schemas can be mutually beneficial for semiotic AI language and task processing.

IV. PARADIGMATIC ASSOCIATIONS

Paradigms hold great power in human estimation as they often define the zeitgeist surrounding historical, societal, and cultural events and experiences. The same is true at the logical later of coding, processing, and parsing information as paradigms are powerful representations of symbolic realities meant to be understood by the AI systems in question. As we explore paradigmatic associations in semiotic AI, the areas of structural linguistics, implicitly learned paradigmatic relations, syntagmatic and paradigmatic associations, computation of word associations, and a syntagmatic paradigmatic model will be discussed. As discussed earlier, linguistic structures are foundational to building meaning in semiotic AI as they provide the basis for understanding paradigms. Also, taking these linguistic approaches and applying them directly to implicit structures where paradigms can be associated discretely with each other adds another layer of potential semiosis. The process of relating syntactical paradigms across numerous levels of computational understanding is the next level of meaning making. This is undertaken through the association of words, syntax, symbols, and groups of related data to form complex computational models and relationships for building meaning for accommodating and assimilating information. Finally, modeling these syntagmatic and paradigmatic relationships is necessary to make the final meaning generation imperative for paradigmatic semiosis.

Structural linguistics have been used for human communication for as long as language has existed. As we navigate social and cultural relationships, human beings build immense cognitive databases for understanding linguistic structures such as symbols, paradigms, syntax, and numerous other linguistic relationships. In AI systems, these connections are extremely advantageous since semantic processing is central to the ability of GPTs to translate information into actionable syntagmatic content. The central proposition of structural linguistics rests on the ability to interpret the meaning of a word based on its paradigmatic and syntagmatic associations [12]. In other words, when a person reads the word “wet”, they automatically associate that word with multiple other words and build meaning maps for those terms. One might associate “wet” with rain, water, clouds, ocean, river, etc. This is a syntagmatic association model. However, when building a paradigm, all of these terms and perhaps even antonyms might be grouped together to form a central concept of the word “wet.” This understanding of word associations has led to the advent of synonymous models such as the Tensor Encoding (TE) model which can perform numerous semantic tasks including synonym judgement [12]. In AI semiotic systems, the ability to form paradigmatic and syntagmatic associations for meaning creation is at the center of the process of semiosis. Without the basic association of words and scaffolding of paradigmatic content, meaning creation and association would not be possible. With these semiotic strategies, however, query expansion, intrinsic linguistic synthesis and expression, and internal paradigmatic evaluation are made possible, contributing to the realization of semiosis.

Paradigmatic relationships are another important area associated with semiosis since these connections happen “within the same event, either simultaneously, immediately following each other, or separated by one or more other elements [13].” These close connections between words, phrases, occurrences, and other events make the construction of paradigms possible. The question posed by Yim, et. al. in their 2019 article is: “Can paradigmatic relations be learned implicitly?” This question has been posed numerous times in research as it deals with the central tenet of meaning-making around syntagmatic and paradigmatic structures. Interestingly, this research was performed on human subjects to grasp how humans might associate words syntagmatically to build paradigms implicitly. The research supported the implicit learning of paradigmatic relations where participants had strong syntagmatic connections [13] suggesting that paradigmatic relationships were implicitly possible. This is a promising finding for potential semiosis in machine constructs as well, since similar syntagmatic relationships have been noted in semantic processing and relationship building.

Syntagmatic and paradigmatic associations share many connected articulations with semiotic AI. Attributional and relational similarities form the central base for understanding semantic similarity in human and machine systems, supporting learning schemas at numerous levels [14]. Humans depend on semantic similarity for numerous communication and socialization circumstances. For instance, when someone says, “you wear that well” they might be referring to shoes, clothes, a particular social or work position, a smile, or many other semantic possibilities. As humans, we learn to draw on these semantic similarities to interpret meaning. These same meaning-making and semantic relationships are central to semiosis in machines as these are relationships that are idiomatic and require potentially massive amounts of context to understand and interpret. The simplest way to begin these associations is through forming semantic relationships between pairs of words [14] and subsequently building expanded paradigms and syntax around them. This action of paradigmatic and syntagmatic scaffolding can enrich the semiotic relationships necessary for semiosis in AI.

Computing word associations carries a heavy burden of the necessary capability to form and process semiotic AI scaffolds. The construction of paradigmatic algorithms for semiotic AI is tied directly to their relationship as either relationally paradigmatic or syntagmatic. “There is a syntagmatic relation between two words if they co-occur in spoken or written language more frequently than expected from chance and if they have different grammatical roles in the sentences in which they occur. Typical examples are the word pairs coffee – drink, sun – hot, or teacher – school. The relation between two words is paradigmatic if the two words can substitute for one another in a sentence without affecting the grammaticality or acceptability of the sentence. Typical examples are synonyms or antonyms like quick – fast, or eat – drink [15].” Given this example, semiosis is possible in systems (human and machine) with paradigmatic and syntagmatic connections that can be made, sustained, synthesized, and perpetuated incorporating hyper-assimilation and -accommodation. With these capabilities brought to bear, meaning-making and understanding at a basic level have the potential to support and develop semiotic AI capabilities. This is accomplished using “algorithms that automatically retrieve words with either the syntagmatic or the paradigmatic type of relationship from corpora [15] suggesting meaningful connections between semantic-layer information and semiotic-layer schemata. Through building scaffolds and schemata through associations, cyber defense systems such as Intrusion Prevention Systems (IPS) can make informed and accurate predictions concerning malicious programs, thereby adding virtually precognitive protections to networks, databases, and critical information.

In their work on sentence processing, Dennis and Harrington developed a Syntagmatic Paradigmatic (SP) model that makes associations across large linguistic frameworks, suggesting that semiotic AI schemas are potentially rooted in complex meaning generation and association. This is accomplished through characterizing sentence processing as retrieval from memory using distributed representations [16]. This ability to "generalize beyond the specific instances in memory" lends credence to the potential semiotic capacity of AI systems in that it is based more on large, complex attributes instead of more discrete increments. Another area of promise with the SP model is its ability to provide systematicity, making arbitrary relationships using relational representations [16]. This function of the model allows the system to make connections between numerous disparate data sets suggesting potential semiotic utility for AI. From a structural linguistic perspective [12], the use of sentence processing is the next level in syntagmatic and paradigmatic processing adding potential to the development of semiotic AI.

Semiosis is dependent on numerous, complex methods to construct meaning across systems, language, and algorithms. This is nowhere more evident than in the areas of syntagmatic and paradigmatic associations. With a view into how syntax and paradigms are constructed and used in language, logic, and processing, semiotic AI has the potential to draw together numerous complex threads algorithmically to support greater communication, understanding, and meaning generation. With the use of automatic query expansion for structuring language exponentially, AI systems can generate more meaningful responses and process larger, deeper data stores for increased meaning and context. Through implicitly learned paradigmatic associations, semiosis can be supported intrinsically to offer more endemic capability and data synthesis. Syntagmatic and paradigmatic word associations through understanding word similarity also undergirds semiosis as the connections between words can created syntax and paradigms necessary for meaning-making and algorithmic growth and synthesis. Additionally, computation of word associations adds another layer of capability to AI semiosis as the syntagmatic and paradigmatic capabilities inherent in these processes underscores the process of connectedness of language for making meaning. All of these methods and capabilities together construct potential sentence analysis and synthesis withing semiotic AI to further grow and expand understanding and meaning across the AI enterprise.

V. ADVANCED HEURISTICS

Heuristics or mental shortcuts, have been an item of study related to semantic AI for some time as they are focused on efficiency and rapidity of processing. However, the power of heuristics also have deep application within the realm of semiotic AI, since these mental shortcuts are directly related to the abstract symbolic thought necessary for the synthesis of meaning in human cognition. There are numerous ways to explore heuristic models, from the basis of efficient, parallel processing to the application of heuristics to cybersecurity. However, the following analysis will delve primarily into how heuristic capabilities can support cognitive and affective frameworks for semiosis. First, an examination of heuristics using rules-based algorithms for aggregation of common data sets will be employed. Next, metaheuristics for metacognition and information depth will be examined. Then, an empirical study related to heuristics will be explored to get a sense of the discretely scientific and mathematical processes being used to produce more complex heuristic models. Finally, two game-related heuristics models will be explored and related to how humans use semiosis when playing games.

Rules are the currency humans and machines use to understand and navigate data in complex systems. Algorithms contain numerous rule sets in their detailed instructions to ensure an AI is operating efficiently and correctly. Within heuristic frameworks, rules are also of critical importance as they help direct the processing and confluence of data for synthesis [17]. Heuristic frameworks operate on the premise of providing efficient, direct correlation of data within specific scaffolds to build schemas capable of assimilating and accommodating various types of information. This is a critical aspect of meaning making in semiotic frameworks since symbolic intricacies related to metacognition rest in understanding the information behind the information. This means that grouping parallel information and using primary, secondary, and tertiary rule sets to check and relate disparate data can lead to building efficient and effective mental shortcuts within machine systems that allow them to recognize and communicate effectively at the semiotic level [17]. This is generally accomplished using multiple running processing threads, load balancing, and granularity control to ensure the data is being sorted, related, and processed efficiently and rapidly [17].

The act of using rules to bring together disparate data to establish schemata relates directly to the next level of heuristic semiosis: metaheuristics. "Metaheuristics exploit not only the problem characteristics but also ideas based on artificial intelligence methodologies, such as different types of memory structures and learning mechanisms, as well as analogies with optimization methods found in nature [18]." Built on this methodology, mechanisms for information synthesis toward semiosis have potentially solid purchase. This method of forming data connections and relationships is akin to what has been observed for centuries in human metacognitive capability. The act of "thinking about thinking" carries with it the most foundational characteristics of building and creating meaning [4]. In machine systems, the act of metacognition and metaheuristics rests primarily on the ability of the system to correlate not only data, but complex sets of data that can be semiotically woven together. IPS and other mechanisms used form cyber defense use heuristics to make predictions and decisions concerning protective tactics for information networks. Tarantilis, et. al. suggests that, “a metaheuristic algorithm can also use one or various neighbor structures during the search process…or metaheuristic algorithm or a sophisticated combination of different metaheuristic concepts, a hybrid metaheuristic algorithm [18].” Using these metaheuristic structures, semiosis can be promoted by the interleaving of data and concepts toward the building of semiotic AI.

Kask and Dechter espouse an empirical framework in their study concerning mini-bucket heuristics [19]. The concept rests on “using a branch and bound search for finding the Most Probable Explanation (MPE) in Bayesian networks [19].” The use of Bayesian networks in this case can be leveraged for predictive analysis, which is critical for semiotic approaches. Part of the human ability to tie information together for the construction and synthesis of meaning is based on what most people would consider Bayesian or historical data. This is a foundational precept from the affective domain of learning as it draws on past information and experiences to develop cognitive, psychomotor, and affective linkages for accessing and creating meaning [3]. Through the use of elimination of mini-buckets through Bayesian analysis, Kask and Dechter found that, “search can be competitive with the best known approximation algorithms for probabilistic decoding such as Iterative Belief Propagation (IBP) when the networks are relatively small, in which case search solved the problems optimally [19]” indicating heuristic capabilities offer a way toward connecting information and efficiently processing and interrelating said data.

Games have been used as mechanisms toward meaning synthesis in the human experience from time immemorial. Games are a way to not only model human and machine learning, but to guide meaningful interactions that can be used to scaffold connections for semiosis. Ancient historical and cutting-edge modern context surround the eastern game of Go; a mainstay in China and one studied most recently through a contest between a Chinese Go champion and Google’s Deep Mind. Bergmark and Stenberg study the heuristic relationships surrounding Go as they examine heuristics using a Monte Carlo Tree Search (MCTS) model [20]. The outcome of their research gave insight into how decisions can be made and related to one’s opponent in a game situation; a central aspect of AI algorithmic representations in GANs. Interestingly, AI can use metaheuristic analysis to predict all possible moves to probabilistically select the best move; [20] a metacognitive advantage over the semantic capabilities of most humans. While this capability is more closely related to semantics, there are numerous foundational components present which allow for the construction of meaning based on the machine capability to “outthink” an opponent. Another study using the game of Connect Four, delves into teaching computers to “think [21].” The researchers in this case decided to build a “genetic algorithm” to “evolve” proper weight values in the systemic thought processes of the program. “A genetic algorithm is an optimization technique that uses a fitness function to attempt to find the best value for a variable over many iterations, in a manner that mimics natural selection [21].” Again, this is a more semantic representation of data, but has direct application to potentially semiotic AI due to the decision processes that leverage integrated meaning behind the thought processes employed.

Heuristic models and techniques lend themselves well to semiotic AI through their ability to groups information into mental shortcuts that can be used for the construction of meaning in machine systems. Rules-based systems are a natural starting point for establishing algorithmic properties for building meaning since it is those rule sets that make processing and interleaving of data practicable. This naturally leads to the process of metaheuristics where heuristic methods are further refined and based on analogies; another particularly meaning-based area of human thought in the affective domain. Bayesian analysis adds another important later to this framework as it is based on prior information that can be leveraged for semiosis, much the same way we as humans use memory. Of course, human experience and processing of information through gamification is an area of human practice that has existed for as long as humanity. Through gaming, machine systems have the opportunity to game out information and begin to build potential meaning scaffolds that may be used in assimilation and accommodation of data into future schemata.

VI. AI ACCOMMODATION/ASSIMILATION MODEL

The following model is descriptive of the aforementioned data concerning neural sketch learning, paradigmatic associations, semiotics, and advanced heuristics for semiotic AI scaffolding. As mentioned earlier, accommodation and assimilation are integral parts of learning theory. Accommodation “is where the new element does not and cannot fit the new schema and thus a process of transformation of both takes place, involving the original stimulus or object of learning and the schema that is attempting some form of accommodation with it [21].” As relates to semiotic machine systems, accommodation is the level at which an AI would necessarily have to make allowances for information not specific to its particular learning or knowledge schema. As a contrast, assimilation “is where a new element has to be addressed and made sense of by the individual, but this process is still essentially passive. The new elements are easily absorbed, indeed assimilated, into the existing schema of the individual [21].” Again, assimilation requires a semiotic AI to take information that may be unfamiliar or new and make sense of that information; an integral aspect of semiosis.

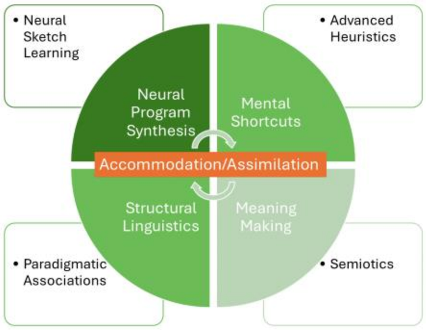


Figure 1. AI Accommodation/Assimilation Model

Figure 1 depicts the AI Accommodation/Assimilation Model, indicating the four areas discussed earlier and relating these areas to their functions and support to AI semiosis through inputs for accommodation and assimilation of information and meaning for semiotic AI. Neural Sketch Learning supports semiosis through neural program synthesis where structures mimicking human neural networks can provide the underlying layer of logical and semiotic pathways necessary for information construction and creation. Advanced Heuristics provide the mental shortcuts machines will need to take abstract concepts and disconnected data and provide inputs into the neural networks within a semiotic AI superstructure. Paradigmatic Associations are pivotal for creating the structural linguistics necessary to form words, phrases, sentences, etc. for the construction of meaning from a semantic standpoint. Paradigmatic Associations further carry the weight of bridging the linguistic infrastructure between neural networks and heuristic structures. Finally, Semiotics are the glue that binds all of the other structures together through the process of meaning making, allowing for the construction of meaning for accommodation and assimilation. All of these elements work together to form the necessary basis for AI to accommodate and assimilate new knowledge and meaning (understanding) for the synthesis necessary for basic semiosis.

VII. CONCLUSION

Semiotic AI, from a meaning-making perspective, is essentially the next level of artificial intelligence following the semantic AGI so many are striving for currently. If we are to reach this next echelon, several types of programming, learning theories, and linguistic structures must first be understood and modeled. Semiotics in general must first be understood as they are not merely about pure information, but also about the accommodation and assimilation of meaning. To get closer to this level of understanding and meaning synthesis, neural sketches must be considered as they provide components of information that can be leveraged across neural networks. Layered atop these network scaffolds are paradigmatic associations necessary for understanding the paradigmatic, syntagmatic, and idiomatic language and components necessary for semiosis. Also, advanced heuristics must be considered as they provide the mental shortcuts generally missing across current AI structures that could assist with the further construction and synthesis of semiotic meaning. Finally, a holistic model is presented above to suggest a way toward a confluence of these disparate methodologies into an apparatus for the accommodation and assimilation of information into and by semiotic AI. These strategies are foundational to cyber offensive and defensive operations through the use of predictive decision support to ensure advanced risk avoidance and mitigation. Altogether, the research and recommendations herein provide an overview of the potential tools and methods for machine semiosis.

REFERENCES

[1] R. Gudwin and J. Queiroz, "Towards an introduction to computational semiotics," International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 2005., Waltham, MA, USA, 2005, pp. 393-398, doi: 10.1109/KIMAS.2005.1427113.

[2] Pospelov, D.A., IJCAI'75: Proceedings of the 4th international joint conference on Artificial intelligence - Volume 1September 1975Pages 65–70.

[3] Roy, D. Semiotic schemas: A framework for grounding language in action and perception, Artificial Intelligence, Volume 167, Issues 1–2, 2005, Pages 170-205, ISSN 0004-3702, https://doi.org/10.1016/j.artint.2005.04.007.

[4] Valerio Targon, Toward Semiotic Artificial Intelligence, Procedia Computer Science, Volume 145, 2018, Pages 555-563, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.11.121.

[5] Nöth, Winfried. “Semiotic Machines.” Cybern. Hum. Knowing 9 (2002): 5-21.

[6] Agrawal, Paras M., and Ramesh Sharda. “OR Forum—Quantum Mechanics and Human Decision Making.” Operations Research 61, no. 1 (2013): 1–16. http://www.jstor.org/stable/23482069.

[7] Nye, Maxwell, Armando Solar-Lezama, Josh Tenenbaum, and Brenden M. Lake. "Learning compositional rules via neural program synthesis." Advances in Neural Information Processing Systems 33 (2020): 10832-10842.

[8] Dong, Li. “Learning natural language interfaces with neural models.” AI Matters 7 (2019): 14 - 17.

[9] Nye, M., Hewitt, L., Tenenbaum, J., Solar-Lezama, A., Proceedings of the 36th International Conference on Machine Learning, PMLR 97:4861-4870, 2019.

[10] Murali, Vijayaraghavan, Letao Qi, Swarat Chaudhuri and Chris Jermaine. “Neural Sketch Learning for Conditional Program Generation.” International Conference on Learning Representations (2017).

[11] Shin, Richard, Miltiadis Allamanis, Marc Brockschmidt and Oleksandr Polozov. “Program Synthesis and Semantic Parsing with Learned Code Idioms.” ArXiv abs/1906.10816 (2019).

[12] Symonds, M., Bruza, P., Zuccon, G., Koopman, B., Sitbon, L.,and Turner. I., 2014. Automatic query expansion: A structural linguistic perspective. J. Assoc. Inf. Sci. Technol. 65, 8 (August 2014), 1577–1596. https://doi.org/10.1002/asi.23065

[13] Yimm, H., Savic, O., Unger, L., Sloutsky, v., and Dennis. S., 2019. “Can Paradigmatic Relations be Learned Implicitly?” Australian Research Council’s Discovery Projects.

[14] Matsuoka, J. and Lepage. Y. 2014. Measuring Similarity from Word Pair Matrices with Syntagmatic and Paradigmatic Associations. In Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon (CogALex), pages 77–86,

Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

[15] Reinhard Rapp. 2002. The Computation of Word Associations: Comparing Syntagmatic and Paradigmatic Approaches. In COLING 2002: The 19th International Conference on Computational Linguistics.

[16] Dennis, Simon J. and Michael Harrington. “The Syntagmatic Paradigmatic Model: A distributed instance-based model of sentence processing.” (2002).

[17] Perri, Simona, F. Ricca and Marco Sirianni. “Advanced Heuristics for Parallel ASP Instantiation.” RCRA@AI*IA (2009).

[18] Tarantilis, Christos D.. “Advanced Heuristics in Transportation and Logistics.” (2005).

[19] Kask, K and Dechter. R. “Mini-Bucket Heuristics for Improved Search.” ArXiv abs/1301.6708 (1999): n. pag.

[20] Bergmark, Fabian and Johan Stenberg. “Heuristics in MCTS-based Computer Go : Can heuristics improve the performance of MCTS-based computer go?” (2014).

[21] Baly, K., Freeman, A., Jarratt, A., Kling, K., Prough, O. Prof. Greg Hume Research in Computer Science (CS 492) Spring, 2012.

[22] Scott, David. “Philosophies of Learning.” In On Learning: A General Theory of Objects and Object-Relations, 159–72. UCL Press, 2021. https://doi.org/10.2307/j.ctv1b0fvk2.17.

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

Graceful Degradation under Attack: Adapting Control Device Operation Depending on the Current Threat Exposure

Rainer Falk, Christian Feist, and Steffen Fries
Siemens AG
Technology
Munich, Germany
e-mail: {rainer.falk|christian.feist|steffen.fries}@siemens.com

***Abstract*—Cybersecurity includes preventing, detecting, and reacting to cyber-security attacks. Cyber resilience goes one step further and aims to maintain essential functions even during ongoing attacks, allowing to deliver an intended service or to operate a technical process, and to recover quickly back to regular operation. When an attack is carried out, the impact on the overall system operation is limited if the attacked system stays operational, even with degraded performance or functionality. Control devices of a cyber physical system typically monitor and control a technical process. This paper describes a concept for a control device that can adapt to a changing threat landscape by adapting and limiting its functionality. If attacks have been detected, or if relevant vulnerabilities have been identified, the functionality is increasingly limited towards essential functions, thereby reducing the attack surface in risky situations, while allowing the cyber physical system to stay operational.**

Keywords–cyber resilience; cyber physical system; industrial security; cybersecurity.

I. INTRODUCTION

A Cyber Physical System (CPS), e.g., an industrial automation and control system, contains control devices that interact with the real, physical world using sensors and actuators. They implement the functionality to control and monitor the operations in the physical world, e.g., a production system or a power automation system. A control device can be a physical device, e.g., an industrial Internet of Things (IoT) device, a Programmable Logic Controller (PLC), or a virtualized control device, e.g., a container or virtual machine executed on a compute platform.

Control devices communicate via data networks to exchange control commands and to monitor the CPS operation to realize different automation use cases. These use cases may comprise predictive maintenance or the reconfiguration of control devices for flexible automation and for optimizing operational systems (Industry 4.0), or specific line protection features in power system operation. The connectivity of control devices is thereby increasingly extended towards enterprise networks and towards cloud-based services, increasing the exposure towards attacks originating from external networks or the Internet [1].

Being resilient means to be able to withstand or recover quickly from difficult conditions [2][3]. It extends the focus of "classical" Information Technology (IT) and Operational Technology (OT) cybersecurity, which put the focus on preventing, detecting, and reacting to cyber-security attacks, to the aspect to continue to deliver an intended outcome despite an ongoing cyber attack, and to recover quickly back to regular operation. When an attack is carried out, the impact on the overall system operation is limited if the attacked system stays operational, even with degraded performance or functionality.

This paper describes a concept for a control device that can adapt to a changing threat landscape by adapting and limiting its functionality. If attacks have been detected, or if relevant vulnerabilities have been identified, devices can limit their functionality increasingly towards only essential functions, thereby reducing their attack surface in risky situations. Essential functions here relate to the contribution of the device to the intended operational use case and the embedding operational environment.

The remainder of the paper is structured as follows: Section II gives an overview on related work. Section III describes the concept of graceful degradation under attack, and Section IV presents a possible usage example in industrial automation systems. Section V provides a preliminary evaluation of the presented approach. Section VI concludes the paper and gives an outlook towards future work.

II. RELATED WORK

Cybersecurity for Industrial Automation and Control Systems (IACS) is addressed in the standard series IEC62443 [4]. This series provides a holistic security framework as a set of standards defining security requirements for the development process and the operation of IACS, as well as technical cybersecurity requirements on automation systems and the used components.

Cyber resilience gets increasing attention, as can be seen by recent security standards and the draft regulation of the Cyber Resilience Act (CRA) [5] and the Delegated Regulation for the Radio Equipment Directive (RED) [6]. Technical standards are currently developed addressing RED and CRA regulative requirements. The standard NIST SP800-193 [7] describes technology-independent guidelines for resilience of

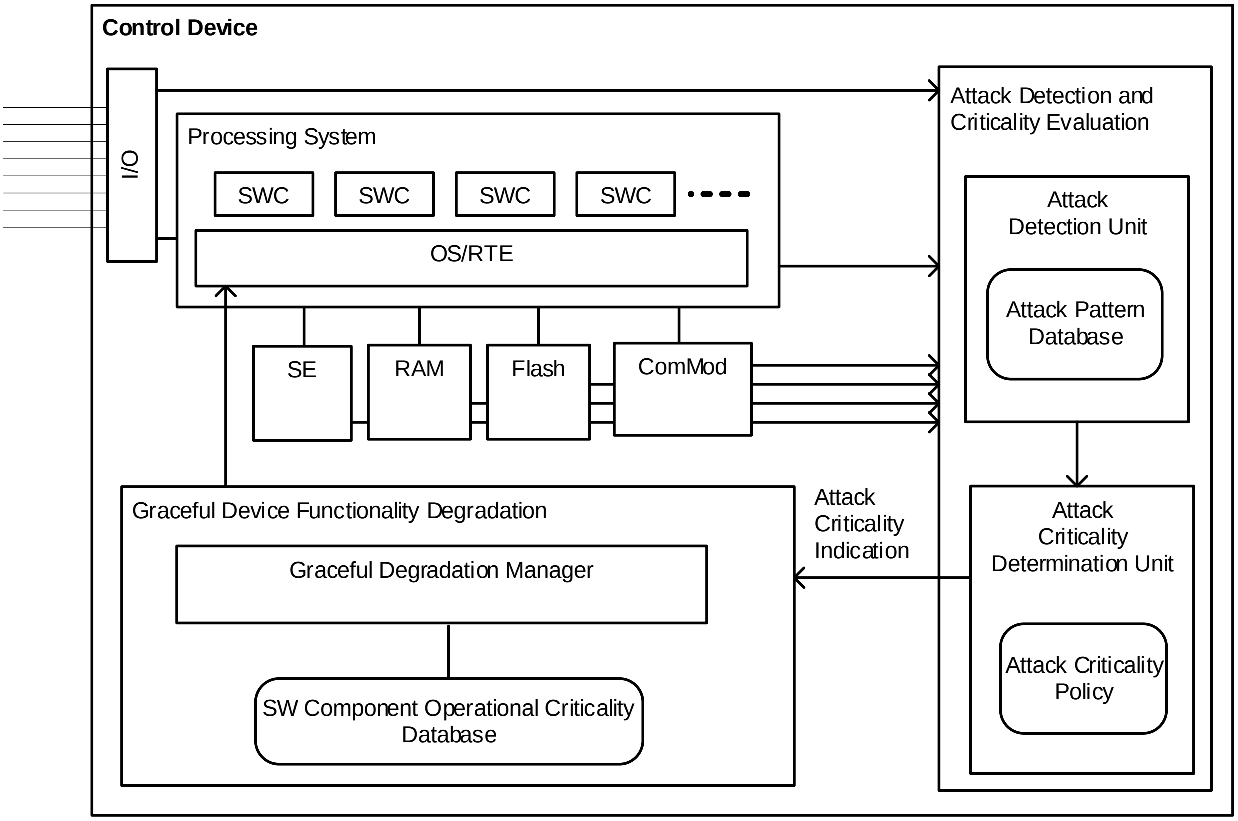

Control Device
I/O
Processing System
SWC
SWC
SWC
SWC
OS/RTE
SE
RAM
Flash
ComMod
Attack Detection and Criticality Evaluation
Attack Detection Unit
Attack Pattern Database
Attack Criticality Indication
Attack Criticality Determination Unit
Attack Criticality Policy
Graceful Device Functionality Degradation
Graceful Degradation Manager
SW Component Operational Criticality Database

Figure 1. Control Device with graceful degradation under attack.

platform firmware. Resilience-specific roots of trust are defined for update of platform firmware, for detection of a corrupted firmware, and for recovery from a compromised platform state. England et al. give a high-level overview of the Cyber Resilient Platforms Program (CyReP), describing hardware and software components addressing NIST SP800-193 requirements [9]. A working group on “cyber resilient technologies” of the Trusted Computing Group (TCG) is working on technologies to enhance cyber resilience of connected systems. Here, different building blocks for cyber resilient platforms have been described that allow to recover from a malfunction reliably back into a well-defined operational state [8]. Such building blocks enhance resilience as they allow to recover quickly and with reasonable effort from a manipulation. Basic building blocks are a secure execution environment for the resilience engine on a device, protection latches to protect access to persistent storage of the resilience engine even of a compromised device, and watchdog timers to ensure that the resilience engine can in fact perform a recovery.

The draft regulation of the Cyber Resilience Act (CRA) [5] includes in Annex I requirements related to maintain essential functions under attack, by the requirement “protect the availability of essential functions, including the resilience against and mitigation of denial of service attacks”. Furthermore, it is also required that devices “minimize their own negative impact on the availability of services provided by other devices or networks”. Specifically, the latter is also a prominently stated requirement of RED [6].

The NIST Cybersecurity Framework (CSF) 2.0 [10], which gives general guidance on managing risk, addresses resilience for normal and adverse situations. A further document from ETSI, EN 303 645 [11], describes specific requirements for the consumer device domain.

III. CONTROL DEVICE WITH GRACEFUL DEGRADATION UNDER ATTACK

Control devices of a cyber physical system monitor and control a technical process via sensors and actuators. The proposed enhanced control device can adapt to a changing threat landscape by adapting and limiting its functionality depending on the current threat landscape. If attacks have been detected, or if relevant vulnerabilities have been identified, the functionality of the device is increasingly limited towards essential functions. This graceful degradation under attack reduces the attack surface in risky situations, while maintaining essential functions of the device. This allows the cyber physical system, in which the control device is deployed, to stay operational even during attack.

Figure 1 shows the concept of a control device that is designed for graceful degradation under attack. The main functionality of the device is realized on its processing system by multiple SoftWare Components (SWC) that are executed

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

by an Operating System (OS) and/or an app RunTime Environment (RTE). Software components may, e.g., implement the control function and diagnostic functions. The components interact with the physical world via sensors and actuators that are connected via an Input/Output (I/O) interface. The processing system uses a Secure Element (SE) for secure key storage and cryptographic operations, a Random Access Memory (RAM), a flash memory, and a Communication Module (ComMod).

An attack detection and criticality evaluation module monitors the operation of these device components to detect unexpected device behavior, here by matching the detected monitoring events with an attack pattern database. It would also be possible to check the device monitoring data against reference states providing the expected behavior. Such a check could be done against static reference data, but could also be done in conjunction with a digital twin, providing a simulation of the ongoing process. If a suspicious device behavior is detected, a criticality is determined, and depending on that, the functionality of the device is adapted by the Graceful device functionality Degradation Manager (GDM). For example, a SWC implementing a simplified control function with reduced functionality can be activated instead of the regular control function, reducing the threat exposure.

This example shows a self-contained realization in which the attack detection and graceful degradation functionality is realized as part of the device. A distributed implementation involving also device-external components would be possible as well, but would require tight protection of all external interfaces to ensure a reliable operation even during ongoing attacks.

In industrial automation, the control functionality is usually not fixed, but is commissioned by the automation system operator, a machine builder, or an integrator. For this application domain, the need is therefore foreseen to allow also commissioning the graceful degradation functionality of a control devices, allowing to define the device resilience behavior under attack. This specifically relates to the definition of essential functions, depending on the application use case.

IV. Usage Example

This section describes the usage in an exemplary way, distinguishing software components of varying criticality from the perspective of maintaining the CPS operation under attack.

Figure 2 shows example software components that are grouped according to the operational criticality. The graceful degradation manager activates the software components of the respective functionality group depending on the current attack scenario. In this example, three sets of software components are defined, defining the software components that are active in full, reduced, and in minimum functionality mode.

To ensure cyber resilience, the functionality is reduced to a limited control functionality that can be less optimized and lead to reduced CPS performance, and to keep limited remote access. In more critical attack scenarios, a fail-safe operation mode is activated, i.e., if even the reduced functionality operation cannot be ensured reliably.

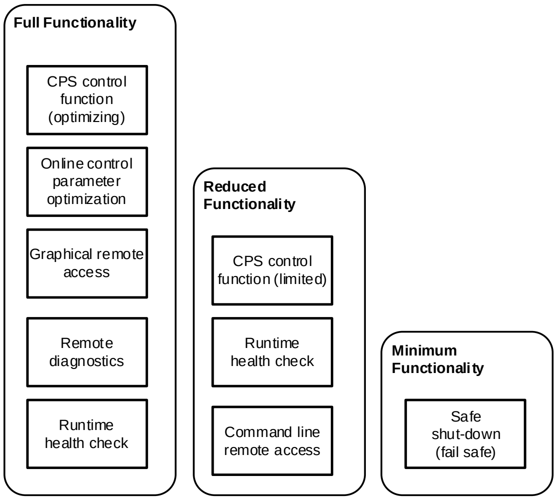

Full Functionality
CPS control function (optimizing)
Online control parameter optimization
Graphical remote access
Remote diagnostics
Runtime health check
Reduced Functionality
CPS control function (limited)
Runtime health check
Command line remote access
Minimum Functionality
Safe shut-down (fail safe)

Figure 2. Software components with different operational criticality.

As an industrial example, a protection device of a substation may be considered that is attacked via the network interface. In the extreme case, the network interface may be switched off for a limited time by the GDM, keeping the protection functionality based on local sensor readings and connected actors. That way, the protection device will not communicate its measurements to other substation devices in the substation, but it retains the local protection functionality and thus the safety of the connected power line.

V. Evaluation

This section gives a preliminary evaluation of the presented concept from different perspectives.

CPS availability perspective: Availability and the flexibility to adapt to changing production requirements are important requirements for OT operators [5]. The proposed approach allows to maintain CPS operation in a limited way even under ongoing attacks or in specific failure situations. A reliable CPS operation can be maintained, avoiding the need to shutdown the CPS operation completely. This is considered to be the main advantage of enhanced control device resiliency with graceful degradation under attack, as the availability of the CPS is improved.

CPS operational performance perspective: The limited function mode may lead to a reduced productivity and less efficiency of the CPS. The exact impact depends on the limitations of the limited control operation functionality.

Implementation perspective: Devices have to implement the functionality for attack detection and resilience management / graceful degradation in a highly protected execution environment that can be relied upon even if the main processing system of the control device should be attacked. The overhead depends on the specific technical implementation approach, e.g., requiring an additional protected hardware component, e.g., a secure microcontroller or a secured Field Programmable Logic Controller (FPGA). Both development effort and hardware costs are increased, which would have an impact in particular for cost-optimized control devices.

Engineering perspective: The graceful degradation functionality (attack criticality determination, as well as the definition of use case specific essential functions) has to be planned and defined so that it can be commissioned on the control device, leading to additional commissioning effort. It may be required that the same functionality has to be realized in different versions, e.g., in fully flexible, optimized operation mode and a limited operation mode. Blueprints that give practice-proven engineering examples can limit the required additional engineering effort.

Testing perspective: The graceful degradation functionality has to be tested carefully to ensure that relevant attack scenarios are reliably detected, and also to validate that the limited control operation mode is reliably activated and performs reliably even under the detected attack scenarios. Testing has to be performed both on device-level for a single control device, as well as on system level for a CPS that uses multiple control devices, where some may be enhanced with graceful degradation under attack. As testing attack scenarios in real-world operational systems is often not possible, simulation tools are essential that allow simulating the CPS operation realistically under various attack scenarios when the engineered graceful degradation functionality is in place. Testing can be performed not only during the planning and engineering phase, but also during regular CPS operation to test the impact of recent attacks.

Overall, implementing, engineering, and testing graceful degradation under attack implies additional effort that, in the end, has to be justified by the increased availability of the CPS. The benefit depends on the attacks observed in real-world operations. Simulation tools (like digital twins) can be used also for this purpose to determine key performance indicators of the real-world CPS for which resilience under attack is protected with control devices implementing the engineered graceful degradation functionality and comparing it with a simulated CPS using control devices *not* implementing the engineered graceful degradation functionality.

VI. Conclusion and Future Work

The proposed concept for cyber resilient control devices can enhance CPS availability even under ongoing attack scenarios. However, it comes with relevant additional effort for implementation, engineering, testing, training, and with overhead for the trusted execution environment required for resilience functionality that requires besides hardware support also specific security-focused implementation effort. However, cyber resilience requirements and technologies are increasingly defined in cybersecurity standards and regulations, and are adopted in real-world solutions, e.g., for data centers [12].

The additional effort needed for implementing cyber resilience for control devices has to be justified by the positive impact on CPS operation, allowing to maintain a reliable CPS operation during ongoing attacks. The CPS operation may relate to a business model focusing on providing a continuous service like energy provisioning or may focus on the preservation of a safety function, like the availability of a protection system. Simulation tools for CPS and their control devices allow investigating cyber resilience for CPS in both the planning and operation phases, reducing in particular the testing effort, and allowing to analyze the effectiveness for different types of attack.

References

[1] Platform Industrie 4.0, “Resilience in the Context of Industrie 4.0”, Whitepaper, April 2022. [Online]. Availabe from: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Resilience.html 2023.08.08

[2] R. Falk and S. Fries, “Enhanced Attack Resilience within Cyber Physical Systems”, Journal on Advances in Security, vol 16, no 1&2, pp. 1-11, 2023. [Online]. Available from: https://www.iariajournals.org/security/sec_v16_n12_2023_paged.pdf 2023.08.08

[3] R. Falk and S. Fries, “System Integrity Monitoring for Industrial Cyber Physical Systems”, Journal on Advances in Security, vol 11, no 1&2, July 2018, pp. 170-179. [Online]. Available from: www.iariajournals.org/security/sec_v11_n12_2018_paged.pdf 2023.08.08

[4] IEC 62443, “Industrial Automation and Control System Security” (formerly ISA99). [Online]. Available from: http://isa99.isa.org/Documents/Forms/AllItems.aspx 2023.08.08

[5] “Proposal for a Regulation of the European Parliament and of the Council on horizontal cybersecurity requirements for products with digital elements and amending Regulation (EU) 2019/10202, COM/2022/454 final, Document 52022PC0454, Sep. 2022. [Online]. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52022PC0454 2023.08.08

[6] “Directive 2014/53/EU of the European Parliament and of the Council of 16 April 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of radio equipment and repealing Directive 1999/5/EC Text with EEA relevance”, 10/2023. [Online]. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32014L0053 2023.08.08

[7] A. Regenscheid, “Platform Firmware Resiliency Guidelines”, NIST SP 800-193, May, 2018. [Online]. Available from: https://csrc.nist.gov/publications/detail/sp/800-193/final 2023.08.08

[8] TCG, “Cyber Resilient Module and Building Block Requirements”, V1.0, October 19, 2021. [Online]. Available from: https://trustedcomputinggroup.org/wp-content/uploads/TCG_CyRes_CRMBBReqs_v1_r08_13jan2021.pdf 2023.08.08

[9] P. England, et al., “Cyber resilient platforms”, Microsoft Technical Report MSR-TR-2017-40, Sep. 2017. [Online]. Available from: https://www.microsoft.com/en-us/research/publication/cyber-resilient-platforms-overview/ 2023.08.08

[10] NIST CSF, “The NIST Cybersecurity Framework (CSF) 2.0”, Feb., 2024. [Online]. https://doi.org/10.6028/NIST.CSWP.29 2023.08.08

[11] EN 303 645, “Cyber Security for Consumer Internet of Things: Baseline Requirements”, ETSI, V2.1.1 (2020-06), June 2020. [Online]. Available from: https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf 2023.08.08

[12] Intel Data Center Block with Firmware Resilience, Solution Brief. [Online]. https://www.intel.com/content/dam/www/public/us/en/documents/solution-briefs/firmware-resilience-blocks-solution-brief.pdf 2023.08.08

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

Analysing Cyber Challenges: Towards Enhancing Autonomous Vehicle Cybersecurity Resilience

Tanisha Soldini, Elena Sitnikova, Karl Sammut
College of Science and Engineering
Flinders University
Adelaide, Australia
e-mail: {tanisharose.soldini | elena.sitnikova | karl.sammut}@flinders.edu.au

***Abstract*—Autonomous vehicles' rise in society represents an important technological advancement in the transportation sector, promising improved financial investments, mobility, and efficiency. Connectivity to cloud-based or fifth generation cellular networks increases autonomous vehicles' exposure to cyber threats, compromising vehicle safety, privacy, and economic stability. Ensuring resilient cyber security measures are critical for safeguarding transportation's critical infrastructure. This paper presents a taxonomy of cyber attacks and mitigation mechanisms of autonomous vehicles. Analysis of recent literature reveals a diverse range of threats, from Global Positioning System spoofing to malware attacks, countered by mitigation mechanisms, such as cryptography, software and network security solutions. Examination of current challenges has identified several future research directions, such as architectural solutions and adversarial machine learning, presenting continuous opportunities for innovation and advancement in the transportation field. Developing robust cyber security mechanisms is essential to closing the gap in protecting autonomous vehicles and ensuring the integrity of transportation infrastructure.**

Keywords-autonomous vehicles; critical infrastructure; cyber security; cyber resilience; taxonomy.

I. INTRODUCTION

Autonomous Vehicles (AVs) represent a continuing technological innovation in the transportation sector. The evolution of AVs includes state-of-the-art sensor technology, computing power, and Artificial Intelligent (AI) Systems. Employing AVs in the urban transport landscape has the potential to improve efficiency, lower costs, reduce emissions, increase mobility and accessibility [1][2][3]. AVs function through their interconnected systems and communication protocols, increasing the potential attack surface. Consequences of such attacks include compromised safety, breaches of information, financial losses, and damage to reputation [4]. Current taxonomies for understanding and mitigating cyber attacks on AVs address various elements of cyber security but fails to include all critical areas of AV security. These gaps in knowledge poses risks of accidents, financial losses, and widespread transportation disruptions [5][6][7][8]. Following research questions are developed to address the gaps in securing AVs:

1) What types of cyber-attacks are most pertinent to AV systems, and how can they be categorised?
2) What are the effective mitigation mechanisms for these attacks, and how can they be systematically classified?

This paper proposes a new taxonomy of cyber attacks and mitigation mechanisms on AVs. The taxonomy will classify attack types and countermeasures, facilitating improved identification of system vulnerabilities and offer areas for future research to safeguard transportation infrastructure.

The remainder of the paper is as follows: Section II introduces AVs, discussing their architecture, and impact on society. Section III presents a taxonomy of cyber-attacks, categorising them based on attack types. In Section IV, mitigation mechanisms are investigated, labelling them as network security, software security, and cryptography. Section V provides an analysis of current challenges in securing AVs and Section VI outlines potential areas for future research in developing resilient cyber security measures.

II. INTRODUCTION TO AUTONOMOUS VEHICLES

First introduced in the 1980s, AVs integrate physical and computational processes to improve safety, mobility and efficiency [9].

A. Architecture of Autonomous Vehicles

AVs architecture can be separated into three distinct layers: perception and sensor integration, decision and control, and chassis [10][11].

Perception and Sensor Integration: AVs integrate various sensors, such as Radio Detection and Ranging (Radar), Light Detection and Ranging (LiDAR), Cameras (Image sensors), Global Positioning System (GPS) to perceive the vehicle's surroundings and position localisation.

Decision and Control: Processed sensor data is used to perform higher-level decision-making to outline pathways, predict actions, avoid obstacles, and control vehicle motion.

Chassis: The chassis layer interfaces with the decision and control layer, regulating vehicle mechanical components.

B. Level of Vehicle Autonomy

The Society of Automotive Engineers (SAE) defines vehicle automation into six levels, ranging from 0 (requiring full human control) to 5 (complete automation), dependent on the extent of human interaction necessary for operation [9][12]. The six levels are defined as follows:

- **Level 0:** All tasks accomplished by human drivers.
- **Level 1:** Human driver controls the vehicle, automation systems can assist.
- **Level 2:** Human driver controls driving process and monitors the environments with automated functions applied.

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

- **Level 3:** Automated vehicle with human operator prepared to assume command of the vehicle at any instance.
- **Level 4:** Under specific circumstances, automated driving occurs, otherwise the operator can assume control of the vehicle.
- **Level 5:** Under all conditions, automated driving occurs, and the operator can take control of the vehicle.

Societal impact of AVs is multifaceted and largely dependent on their autonomy level. These levels of autonomy have potential to increase independence and access to transportation, and improve road safety.

III. Taxonomy of Cyber Attacks

A. Methods of Cyber Attacks and Targeted Components

Several attack pathways in AVs exist. These include:

- **Remote Access and Control:** Exploitation of electronic control systems, gaining unauthorised access and critical functions control [6].
- **Sensor Manipulation:** AVs' reliance on sensor technology for manoeuvring in their surroundings, poses a significant threat. Attackers can spoof sensor data or jam signals, causing the system to misinterpret its environment. Such manipulation could result in hazardous driving and breach of privacy [13][14].
- **Wireless Networks:** Utilised by AVs facilitate communication with nearby vehicles and infrastructure, this reliance introduces vulnerabilities exploitable by attackers. By targeting these communication networks, adversaries can disrupt operations or inject false information into the vehicle's system. This interference can lead to confusion or incorrect decision-making by the vehicle's system, compromising its ability for safe and reliable operation [15]. These include vehicle-to-vehicle (V2V), vehicle-to-network (V2N), vehicle-to-infrastructure (V2I), and vehicle-to-everything (V2X) [16].
- **Software Vulnerabilities:** AVs operate as sophisticated computer systems using a variety of algorithms and software. Consequently, software vulnerabilities emerge as a prominent threat to vehicle safety and security. Malicious software, such as ransomware, poses a potential risk to AV operations by disrupting operations or extorting users for financial gain [17].
- **Hardware Vulnerabilities:** Hardware components, such as the Electronic Control Units (ECUs), On-Board Diagnostic Port (OBD) and Controller Area Network (CAN), can pose potential weaknesses to their physical components and systems. These vulnerabilities may be exploited by through tampering and unauthorised access [5]. Developing a taxonomy of cyber attacks and and identifying corresponding mitigation mechanisms is essential for protecting AVs.

B. Motivations and Perpetrators Behind Cyber Attacks

Cyber attacks on AVs are typically perpetrated by hackers, cyber criminals, and disgruntled individuals. Motivations behind these attacks can be divided into three principal objectives: operational disruptions, gaining vehicle control, and data theft [16].

- **Operational Disruptions:** Compromises critical AV components that are essential for driving functionality, rendering autonomous driving inoperative.
- **Gaining Vehicle Control:** Allows attackers to manipulate critical vehicular functionalities, such as route deviation, emergency braking, and speed modulation.
- **Data Theft:** Stealing data from AV systems, potentially fuelling subsequent cyber attacks.

Cyber criminals can infect the AVs' network with malware, disrupting system operations, harming users, their surroundings, and causing financial losses [18].

C. Cyber Attack Classification

Cyber attacks on AVs can be classified as follows:

- **Man-in-the-Middle (MITM) Attacks:** MITM attacks occur when attackers intercept and alter communications between two components, compromising the integrity and confidentiality of the data exchanged. Methods include intercepting and tampering with vehicle communications, impersonating legitimate entities, exploiting wireless interfaces, rerouting messages and attacking dynamic rerouting [19].
- **Infection Attacks:** Infection attacks involve injecting malicious code into a vehicle's systems, which can potentially compromise its functionality and safety. Methods include exploiting software vulnerabilities, violating wireless interfaces, supply chain attacks, infecting removable media, and compromising backend systems [20].
- **Tampering Attacks:** These attacks involve the unauthorised manipulation of data, software, or hardware components on AVs, potentially affecting their performance and safety. Methods include sensor data tampering, such as intercepting camera perception by physically obscuring its view, spoofing LiDAR signals, jamming or injecting noise into sensors [21]. Communication mechanisms can be tampered with by injecting malicious data or MITM attacks. Software/firmware tampering exploits vulnerabilities in the vehicle's ECUs by introducing malicious code into the in-vehicle infotainment system or compromising vehicle software through supply chain attacks. Physically tampering with AVs can grant access to the vehicle's internal networks and components. Rogue commands can be sent from the CAN bus through internal access, and actuators can be tempered with to control AV driving operations [5].

1) Identity-based:

- **Spoofing Attacks:** An attacker feeds false information to vehicle systems or sensors to disrupt their data. Spoofing can occur with sensors and communication systems [6].
- **Impersonation Attacks:** Attackers disguise themselves as legitimate entities to access or influence AV systems. Methods include spoofing vehicle identities, impersonating infrastructure, compromising wireless interfaces and cryptographic keys [21].

- **Sybil Attacks:** A single malicious entity creates multiple identities to gain influence. Methods include impersonating multiple vehicles, overwhelming legitimate entities, disrupting platoon operations, exploiting authentication vulnerabilities, and colluding with malicious insiders [6].
- **Replay Attacks:** Capturing and replaying valid data transmissions to bypass authentication. Replay attacks can target GPS signals, sensor data, and communication mechanism. Cryptographic and sensor fusion replay attacks exist [6].

2) Service-based:

- **Denial of Service (DoS) Attacks and Distributed Denial of Service (DDoS) Attacks:** DoS and DDoS attacks overwhelm systems with data to impair operations. Methods include flooding wireless communication channels, jamming sensor signals, exploiting software vulnerabilities, and targeting backend infrastructure [22].
- **Jamming Attacks:** Jamming attacks interfere with wireless communication channels. Methods include jamming sensor and wireless communications [16].
- **Routing Attacks:** Routing attacks disrupt routing protocols to create network instability. Methods include wormhole attacks, sinkhole attacks, black hole attacks, and grey hole attacks [6].

3) Software-based:

- **Malware Attacks:** Introduces malicious code to compromise systems. Methods include exploiting software vulnerabilities, compromising wireless interfaces, supply chain attacks, removable media infection, and compromising backend systems [7].

4) Data Privacy:

- **Location Trailing Attacks:** Monitors a vehicle's location without authorisation. Methods include exploiting localisation algorithms, compromising wireless communications, and exploiting GPS vulnerabilities [4].
- **Eavesdropping Attacks:** Intercepts and accesses private data transmissions. Methods include intercepting wireless communications, exploiting vulnerabilities in communication protocols, and compromising wireless access points [5].

In classifying each type of cyber attack, their characteristics, techniques, and goals are identified.

D. Potential Consequences of Cyber Attacks on Autonomous Vehicles

Cyber attacks on AVs can cause harm to their users and surroundings. Potential consequences are as follows:

- **Safety Risks:** Effective cyber attacks can cause accidents and endanger surroundings by allowing malicious actors to hijack critical vehicle functions, including path control, acceleration, and braking. Spoofing or jamming of the sensors can disrupt navigation. Based on disrupted collected data, incorrect decisions can lead to vehicle malfunctions. If vehicles used for public transport, emergency services or law enforcement are compromised, public safety can be endangered [5].
- **Loss of Vehicle Control:** Malicious attacks have the potential to gain unauthorised remote access to a vehicle's ECUs, paralysing the car or causing erratic behaviour. Software vulnerabilities allow attackers to compromise safety-critical functions [5].
- **Privacy and Data Breaches:** Sensitive data collected by AVs, such as location tracking, driver behaviour, and other personal information, could be exposed by cyber attacks. If this data is breached, it could enable stalking, identity theft, and other privacy violations [6].
- **Traffic Disruptions and Infrastructure Damage:** Compromised AVs could be rerouted or have their navigation systems manipulated, potentially causing major traffic jams, road blockages, and damage to infrastructure [21].
- **Financial Losses and Legal Liabilities:** Cyber attacks may lead to a loss of public trust, potentially resulting in costly vehicle recall and legal liabilities [6].

Securing AVs from cyber attacks is essential to the formulation of mitigation methods in harm prevention.

IV. Mitigation Mechanisms

Mitigation mechanisms are classified as follows:

A. Network Security

Network security mitigation mechanisms protect AV communication networks and systems from cyber attacks.

1) Intrusion Detection Systems: Intrusion detection systems (IDSs) are employed to detect and mitigate various network-based attacks. There are four main IDSs implemented to secure AVs [6][23]:

- **Signature-based IDS:** Functions by comparing observed behaviour against a database of known signatures.
- **Anomaly-based IDS:** Operates by recognising anomalies in a vehicle's behaviour that deviate from the normal or expected patterns.
- **Specification-based IDS:** Monitors a vehicle's behaviour against a set of predefined rules or specifications.
- **Hybrid-based IDS:** Combines the strengths of signature-based and anomaly-based detection methods to defend against a broader spectrum of cyber threats.

B. Malware Detection

Malware detection systems, an extension of IDSs, employ signature and behaviour-based techniques to mitigate cyber attacks. In addition to these, malware detection includes [7]:

- **Heuristic-based Techniques:** Employ heuristic rules and algorithms to identify potential malware based on characteristics or patterns associated with malicious code.
- **Cloud-based Techniques:** Leverages cloud computing services for efficient and scalable malware detection in AVs.

ISBN: 978-1-68558-186-2

1) Machine Learning and Deep Learning for Intrusion Detection: Network IDSs in AVs utilise Machine Learning (ML) and Deep Learning (DL) models for their fast detection and response times to cyber threats, and ability to leverage insights from data analytics. Models include k-nearest neighbour (KNN), decision trees, auto-encoders and long short-term memory (LSTM) networks [23].

These mechanisms can mitigate cyber attacks, such as spoofing, flooding, modifying hardware components, replay attacks, firmware attacks, and identifying unauthorised access [7].

C. Software Security

1) Machine Learning Algorithms: ML models are employed for various security tasks, including intrusion detection, malware analysis, and vulnerability assessment. Similar to ML for IDSs, ML models detect anomalies and deviations in normal software behaviour, identifying previously unseen attack vectors and zero-day exploits [6].

2) Software Analysis Techniques: Static and dynamic analysis methods are used to analyse AV software for potential vulnerabilities and malicious code [7]:

- **Static:** Examines code without executing it to identify potential vulnerabilities.
- **Dynamic:** Executes code in a controlled environment and monitors for anomalies.

Software techniques can address vulnerabilities like code injection and memory corruption, while ML models counter spoofing, flooding, and replay attacks [21].

D. Cryptography

1) Encryption Techniques: Encryption (symmetric and asymmetric) techniques are used to secure data transmissions and communications in AVs. Public-key cryptography is employed for secure key distribution and authentication in V2V/V2I communications [16].

- **Symmetric:** Encrypts data transmissions in V2V/V2I communications.
- **Asymmetric:** Secures key distribution and authentication in V2V/V2I communications.

Encryption helps mitigate attacks like eavesdropping, spoofing, and MITM attacks, safeguarding the privacy and integrity of transmitted data [16].

2) Authentication Technique:

- **Digital Signatures:** Authenticate the source and integrity of messages or data transmitted between vehicles and infrastructure.
- **Message Authentication Codes:** Provide data origin authentication and integrity verification for V2V/V2I communications.

3) Blockchain Technology: Blockchain (BC) technology is used to store and share information on an advanced database. Each dataset is stored in blocks, linked together in a chain. BC technology has gained popularity with its ability to prevent cyber attacks through its inherent security measures of decentralisation, transparency, encryption, and immutability [24].

These mitigation mechanisms can counter attacks like spoofing, replay attacks, and injection of fake data [16].

E. Comparison of Existing Literature

A systematic review of current literature on cyber attack taxonomies and mitigation mechanisms for AVs was conducted for comparison Table I presents a comparison of existing literature [6][8][16][17][23][25]. This method is performed to reduce bias in the literature. All literature presents a detailed classification of cyber attacks and mitigation mechanisms. Papers [6][8][17][23][25] are lacking in conveying the motivation behind cyber attacks and those responsible. While papers [8][16][17][23][25] do not detail AV architecture.

TABLE I. COMPARISON OF EXISTING LITERATURE

Literature	[17]	[16]	[6]	[23]	[25]	[8]	Analysis of Paper
Architecture of AV	✗	✗	✓	✗	✗	✗	✓
Motivation and Perpetrators of cyber attacks	✗	✓	✗	✗	✗	✗	✓
Cyber Attack Classification	✓	✓	✓	✓	✓	✓	✓
Target Components	✓	✓	✓	✓	✓	✓	✓
Potential Consequences on Society	✓	✓	✓	✗	✓	✗	✓
Mitigation Mechanisms	✓	✓	✓	✓	✓	✓	✓
Current Challenges and Future Work	✓	✓	✗	✓	✓	✓	✓

Comparison of literature reveals significant gaps in current research on AV cyber security taxonomies, highlighting the need for the proposed taxonomy.

V. DISCUSSION

A. Current Challenges in Securing Autonomous Vehicles from Cyber Attacks

The complexity of cyber attacks and securing AVs will continue to grow with advancements in technology. AVs are computers comprised of complex software algorithms, large amounts of data, and a multitude of electronic components, making them difficult to integrate into traditional security approaches. Their complex and interconnected nature allows for multiple potential entry points for attack points, making it a complicated task to secure all these vectors.

Real-time operation is critical for functioning AVs. High volumes of data must be processed and analysed to detect and combat cyber security risks in real-time. Advancements in high-speed computer processing systems and algorithms are imperative for such progress. AV architecture should be designed to manage system faults and scalability issues [5].

Securing vehicle communication is essential to successful functionality. Breaches in AVs have sequential impacts on critical infrastructure, vice versa. V2X technologies require robust protective methods and the implementation of secure communication networks to ensure reliability. Existing mitigation mechanisms against DDoS attacks in V2V communications are largely theoretical and require verification in a trusted testing environment [16].

AVs depend on ML algorithms to decipher real-time data and formulate operational decisions. These algorithms can be vulnerable to adversarial attacks, manipulating sensor data and leading to potentially hazardous decisions. Errors, such as misclassification, can be triggered by specifically crafted adversarial inputs in deep-learning models. Updating these models with new incoming data from the vehicle has the potential to leak private information [20][26].

These challenges in cyber security provide potential areas for future research.

VI. Conclusion and Future Work

A. Future Work

1) Securing Sensor Data: Securing sensor data is vital for AVs to accurately observe their surroundings, make informed decisions, and ensure safe performance. Research is required to improve sensor fusion techniques to combine data from various sensors, providing an accurate view of the environment.

2) Adversarial Machine Learning Algorithms: Restricted availability of standardised datasets has limited the development of Adversarial Machine Learning (AML) mitigation mechanisms against cyber attacks. Current datasets do not account for advancement in AML and adversarial attacks. Compiling an accessible, up-to-date dataset that represents different attack scenarios and network traffic diversity is critical for developing effective AML mitigation mechanisms [27].

3) Real time decision making: AVs require sophisticated algorithms and computing processors to effectively evaluate vast amounts of data, posing challenges in real-time decision making [5].

4) Securing Autonomous Vehicles with AI and BC Technologies: BC technologies demonstrate promise in preventing cyber attacks through their inherent security measures of decentralisation, transparency, encryption, and immutability. In comparison to traditional security approaches, AI has shown greater efficiency and a faster detection rate when addressing cyber threats. There is a lack of research concerning the interrelationship between BC and AI, and consequently, an understanding of AVs' security and privacy [24].

5) Communication Mechanisms: Challenges exist in implementing strong communication networks in AVs. These systems require networks that can manage low-latency communications, high volumes of complex data flows, and resilient connectivity in harsh environments. 5G cellular V2X products are still under development and security attacks have currently not been prominent. Security features are well defined and rely on authentication and encryption. However, their effectiveness needs to be tested before and after AV deployment [16] [23].

6) Architectural Solution: Future research has the potential to combine an architectural solution for AVs with Supervisory Control and Data Acquisition (SCADA) integration. A hierarchical self-aware architectural solution allows for real-time operational analysis, decision formulation, and integration with remote locations. The architecture includes four layers: monitoring, analysis, decision-making, and visualisation. A Security-specific In-vehicle Black Box (STCB) is employed to execute the security models and algorithms within a trusted environment [28]. This architectural solution provides security coverage through multiple hierarchical layers, enabling proactive and tailored security measures. SCADA has multiple integration points:

- Monitoring layer collects data from various sensors.
- Analysis layer uses machine learning techniques for anomaly detection.
- Decision layer determines the severity of incidents and triggers appropriate responses.
- Visualisation layer is a human-machine interface (HMI) module.

Integration of in-vehicle security components with an external Virtual Security Operation Centre (VSOC) facilitates coordinated responses across vehicle fleets. The STCB enhances situational awareness and timely mitigations through security-specific logging and analysis, real-time threat detection, and automated responses. Leveraging SCADA's capabilities, the hierarchical self-aware architecture can be implemented for unified security management of AVs.

B. Conclusion

The increasing reliance on connectivity in AVs has introduced vulnerabilities to cyber attacks, posing significant risks to safety, privacy, and economic stability. This paper has provided a taxonomy of cyber attacks, mitigation mechanisms, and future research areas. Analysis of current challenges reveals the potential for a multidisciplinary approach that integrates an architectural solution with SCADA systems to counter the diverse range of threats facing AVs. As the transportation sector continues to evolve, it is imperative that resilient cyber security methods are developed and implemented to safeguard AVs while maintaining the integrity of critical infrastructure. Future research should focus on addressing the identified challenges and developing innovative solutions to ensure secure and reliable operation of AVs.

References

[1] E. Kassens-Noor *et al.*, "Sociomobility of the 21st century: Autonomous vehicles, planning, and the future city", *Transport Policy*, vol. 99, pp. 329–335, Dec. 2020, ISSN: 0967070X. DOI: 10.1016/j.tranpol.2020.08.022.

[2] D. Bissell, T. Birtchnell, A. Elliott, and E. L. Hsu, "Autonomous automobilities: The social impacts of driverless vehicles", *Current Sociology*, vol. 68, no. 1, pp. 116–134, Jan. 1, 2020, Publisher: SAGE Publications Ltd, ISSN: 0011-3921. DOI: 10.1177/0011392118816743.

[3] R. Bennett, R. Vijaygopal, and R. Kottasz, "Willingness of people with mental health disabilities to travel in driverless vehicles", *Journal of Transport & Health*, vol. 12, pp. 1–12, Mar. 2019, ISSN: 22141405. DOI: 10.1016/j.jth.2018.11.005.

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

[4] A. Algarni and V. Thayananthan, "Autonomous vehicles: The cybersecurity vulnerabilities and countermeasures for big data communication", *Symmetry*, vol. 14, no. 12, p. 2494, Dec. 2022, Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2073-8994. DOI: 10.3390/sym14122494.

[5] A. Giannaros *et al.*, "Autonomous vehicles: Sophisticated attacks, safety issues, challenges, open topics, blockchain, and future directions", *Journal of Cybersecurity and Privacy*, vol. 3, no. 3, pp. 493–543, 2023, Publisher: MDPI AG, ISSN: 2624-800X. DOI: 10.3390/jcp3030025.

[6] K. Kim, J. S. Kim, S. Jeong, J.-H. Park, and H. K. Kim, "Cybersecurity for autonomous vehicles: Review of attacks and defense", *Computers & Security*, vol. 103, p. 102 150, Apr. 1, 2021, ISSN: 0167-4048. DOI: 10.1016/j.cose.2020.102150.

[7] S. Aurangzeb, M. Aleem, M. T. Khan, H. Anwar, and M. S. Siddique, "Cybersecurity for autonomous vehicles against malware attacks in smart-cities", *Cluster Computing*, Oct. 3, 2023, ISSN: 1573-7543. DOI: 10.1007/s10586-023-04114-7.

[8] S. K. Khan, N. Shiwakoti, P. Stasinopoulos, and Y. Chen, "Cyber-attacks in the next-generation cars, mitigation techniques, anticipated readiness and future directions", *Accident Analysis & Prevention*, vol. 148, p. 105 837, Dec. 1, 2020, ISSN: 0001-4575. DOI: 10.1016/j.aap.2020.105837.

[9] C. Morrison, E. Sitnikova, and S. Shoval, "A review of the relationship between cyber-physical systems, autonomous vehicles and their trustworthiness", in *International Conference on Cyber Warfare and Security*, Num Pages: 611-621,XV, Reading, United Kingdom: Academic Conferences International Limited, 2018, pp. 611–621, XV.

[10] W. Zong, C. Zhang, Z. Wang, J. Zhu, and Q. Chen, "Architecture design and implementation of an autonomous vehicle", *IEEE Access*, vol. 6, pp. 21 956–21 970, 2018, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2828260.

[11] M. Pipicelli *et al.*, "Architecture and potential of connected and autonomous vehicles", *Vehicles*, vol. 6, no. 1, pp. 275–304, Mar. 2024, Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2624-8921. DOI: 10.3390/vehicles6010012.

[12] "SAE levels of driving automation™ refined for clarity and international audience", [Online]. Available: https://www.sae.org/site/blog/sae-j3016-update (visited on 05/30/2024).

[13] Y. Wang, Q. Liu, E. Mihankhah, C. Lv, and D. Wang, "Detection and isolation of sensor attacks for autonomous vehicles: Framework, algorithms, and validation", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8247–8259, Jul. 2022, Conference Name: IEEE Transactions on Intelligent Transportation Systems, ISSN: 1558-0016. DOI: 10.1109/TITS.2021.3077015.

[14] Z. El-Rewini *et al.*, "Cybersecurity attacks in vehicular sensors", *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13 752–13 767, Nov. 2020, Conference Name: IEEE Sensors Journal, ISSN: 1558-1748. DOI: 10.1109/JSEN.2020.3004275.

[15] A. Al-Sabaawi, K. Al-Dulaimi, E. Foo, and M. Alazab, "Addressing malware attacks on connected and autonomous vehicles: Recent techniques and challenges", in *Malware Analysis Using Artificial Intelligence and Deep Learning*, M. Stamp, M. Alazab, and A. Shalaginov, Eds., Cham: Springer International Publishing, 2021, pp. 97–119, ISBN: 978-3-030-62582-5. DOI: 10.1007/978-3-030-62582-5_4.

[16] M. Pham and K. Xiong, "A survey on security attacks and defense techniques for connected and autonomous vehicles", *Computers & Security*, vol. 109, p. 102 269, 2021, Publisher: Elsevier BV, ISSN: 0167-4048. DOI: 10.1016/j.cose.2021.102269.

[17] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber threats facing autonomous and connected vehicles: Future challenges", *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2898–2915, Nov. 2017, Conference Name: IEEE Transactions on Intelligent Transportation Systems, ISSN: 1558-0016. DOI: 10.1109/TITS.2017.2665968.

[18] A. Seetharaman, N. Patwa, V. Jadhav, A. S. Saravanan, and D. Sangeeth, "Impact of factors influencing cyber threats on autonomous vehicles", *Applied Artificial Intelligence*, vol. 35, no. 2, pp. 105–132, Jan. 28, 2021, ISSN: 0883-9514, 1087-6545. DOI: 10.1080/08839514.2020.1799149.

[19] R. Gothwal, G. Dharmani, R. S. Reen, and E. G. AbdAllah, "Evaluation of man-in-the-middle attacks and countermeasures on autonomous vehicles", in *2023 10th International Conference on Dependable Systems and Their Applications (DSA)*, Tokyo, Japan: IEEE, Aug. 10, 2023, pp. 502–509, ISBN: 9798350304770. DOI: 10.1109/DSA59317.2023.00070.

[20] M. C. Chow, M. Ma, and Z. Pan, "Attack models and countermeasures for autonomous vehicles", in *Intelligent Technologies for Internet of Vehicles*, N. Magaia, G. Mastorakis, C. Mavromoustakis, E. Pallis, and E. K. Markakis, Eds., Cham: Springer International Publishing, 2021, pp. 375–401, ISBN: 978-3-030-76493-7. DOI: 10.1007/978-3-030-76493-7_12.

[21] A. Chowdhury, G. Karmakar, J. Kamruzzaman, A. Jolfaei, and R. Das, "Attacks on self-driving cars and their countermeasures: A survey", *IEEE Access*, vol. 8, pp. 207 308–207 342, 2020, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3037705.

[22] A. El-Ghamry and M. Elhoseny, "Detecting distributed DoS attacks in autonomous vehicles external environment using machine learning techniques", in *The 3rd International Conference on Distributed Sensing and Intelligent Systems (ICDSIS 2022)*, Hybrid Conference, Sharjah, United Arab Emirates: Institution of Engineering and Technology, 2022, pp. 292–308, ISBN: 978-1-83953-818-6. DOI: 10.1049/icp.2022.2479.

[23] T. Limbasiya, K. Z. Teng, S. Chattopadhyay, and J. Zhou, *A systematic survey of attack detection and prevention in connected and autonomous vehicles*, Aug. 5, 2022. arXiv: 2203.14965[cs].

[24] G. Bendiab, A. Hameurlaine, G. Germanos, N. Kolokotronis, and S. Shiaeles, "Autonomous vehicles security: Challenges and solutions using blockchain and artificial intelligence", *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 3614–3637, Apr. 2023, Conference Name: IEEE Transactions on Intelligent Transportation Systems, ISSN: 1558-0016. DOI: 10.1109/TITS.2023.3236274.

[25] X. Sun, F. R. Yu, and P. Zhang, "A survey on cybersecurity of connected and autonomous vehicles (CAVs)", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6240–6259, Jul. 2022, Conference Name: IEEE Transactions on Intelligent Transportation Systems, ISSN: 1558-0016. DOI: 10.1109/TITS.2021.3085297.

[26] M. Sadaf *et al.*, "Connected and automated vehicles: Infrastructure, applications, security, critical challenges, and future aspects", *Technologies*, vol. 11, no. 5, p. 117, Oct. 2023, Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2227-7080. DOI: 10.3390/technologies11050117.

[27] S. Mokhtari, A. Abbaspour, K. K. Yen, and A. Sargolzaei, "A machine learning approach for anomaly detection in industrial control systems based on measurement data", *Electronics*, vol. 10, no. 4, p. 407, Jan. 2021, Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2079-9292. DOI: 10.3390/electronics10040407.

[28] A. Adu-Kyere, E. Nigussie, and J. Isoaho, "Self-aware cybersecurity architecture for autonomous vehicles: Security through system-level accountability", *Sensors*, vol. 23, no. 21, p. 8817, Jan. 2023, Number: 21 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 1424-8220. DOI: 10.3390/s23218817.

Poisoning Attack Data Detection with Internal Coefficient Displacement for Machine Learning Based NIDS

Hajime Shimada
Information Technology Center, Nagoya University
Nagoya, Japan
Email: shimada@itc.nagoya-u.ac.jp

Takuya Kuwayama
Grad. Sch. Informatics, Nagoya University
Nagoya, Japan
Email: kuwayama@net.itc.nagoya-u.ac.jp

Hirokazu Hasegawa
Center for Strategic Cyber Resilience R&D, NII
Tokyo, Japan
Email: hasegawa@nii.ac.jp

Yukiko Yamaguchi
Information Technology Center, Nagoya University
Nagoya, Japan
Email: yamaguchi@itc.nagoya-u.ac.jp

***Abstract*—Due to the improvement of Machine Learning (ML) techniques, ML has been used extensively in the cyber security area and Machine Learning based Network-based Intrusion Detection Systems (ML-NIDS) is a one of those examples. However, arising methods to attack ML systems are becoming new threats to them. A poisoning attack is one of those threats and has adverse effects on the classification performance. As a threat on ML-NIDS, we are concerned about a threat where an attacker distributes manipulated traffic session data as a new dataset, aiming at a poisoning attack on ML-NIDS. In this paper, we try to identify whether newly added training data is poisoning attack data or not based on the displacement of an internal coefficient of a classifier. This research utilizes Support Vector Machine (SVM) as a classifier so that the internal coefficient vector is represented as a gradient coefficient vector of hyperplane in SVM classifier. We assumed that manipulated traffic session data for poisoning attack will largely confuse the internal coefficient vector. Thus, if the internal coefficient vector displaces largely after retraining with newly added data, we estimate that the newly added data is a poisoning attack data. We also propose a method to define a threshold value that distinguishes poisoning attack data and clean data. We evaluated our proposal with SVM based NIDS with an open traffic session dataset and poisoning attack with Biggio's SVM poisoning algorithm. We confirmed that our proposal can detect poisoning attack data and achieves 0.9838 F1 score at 8% poisoning rate (ratio of newly added poisoning attack training data to existing clean data), which is better performance compared to the existing poisoning attack data detection method.**

Keywords–poisoning attack detection; machine learning based NIDS.

I. Introduction

Due to the improvement of Machine Learning (ML) techniques, machine learning has been used extensively in the field of cyber security. Machine Learning based Network-based Intrusion Detection Systems (ML-NIDS) is a one of those examples. However, many people are proposing new attack methods to ML systems and they are becoming new threats in the cyber security area. There is a poisoning attack that has an adverse effect to the classification performance. The poisoning attack distributes malicious data in advance and that malicious data contaminate and pollute training data. As a threat on ML-NIDS area, we are concerned about a threat where an attacker distributes manipulated traffic session data as a new dataset with aiming poisoning attack, and some ML-NIDS system maintainers wrongly include them in training data.

Although several methods have been proposed to detect and exclude poison data from training data, there are few studies that evaluate the effectiveness of defense methods in machine learning-based NIDS. In this paper, we try to identify whether newly added training data is poisoning attack data or not based on the displacement of an internal coefficient of a classifier. This research utilizes Support Vector Machine (SVM) for a classifier so that the internal coefficient vector is represented as a gradient coefficient vector of hyperplane in SVM classifier. We assumed that manipulated traffic session data for poisoning attack will largely confuse the internal coefficient vector. Thus, if the internal coefficient vector displaces largely after retraining with newly added data, we estimate that the newly added data is a poisoning attack data. This method requires a threshold value to distinguish poisoning attack data and clean data. We also proposed how to define the threshold value from existing clean data. Our proposed method creates the threshold value by dividing existing clean data into baseline data, additional clean data, and additional local poisoning attack data which is generated by some poisoning attack algorithm from existing clean data. By comparing internal coefficient vector displacements between additional clean data learning result and additional local poisoning attack data learning result, we can obtain the threshold value. We assume that ML-NIDS system maintainer perform such an evaluation of additional data to exclude poisoning attack data comes from some attackers.

We evaluated our proposal with Kyoto 2016 Dataset [1][2] which is a traffic session dataset with malicious/benign ground truth label. We created SVM classifier and performed the poisoning attack with Biggio's SVM poisoning algorithm [3] for baseline. Then, we evaluated our internal coefficient displacement based detection with Euclidean distance and compared it with existing SVM poisoning attack detection method Curie [4]. We confirmed that our proposal can detect poisoning attack data effectively in many poisoning rate (ratio of newly added poisoning attack training data to existing clean data) and it achieves 0.9838 F1 score at 8% poisoning rate as a best score. On the other hand, Curie gives moderate performance because Curie evaluates in individual traffic session

ISBN: 978-1-68558-186-2

sample level so that it cannot exclude a poisoning attack traffic session sample that have quite similar characteristic to existing clean data samples.

The rest of the paper is organized as follows. Section II introduces related works about ML-NIDS, traffic datasets, poisoning attacks, and poisoning attack data detection. Section III introduces our proposal that distinguishes poisoning attack data with the displacement of the internal coefficient vector before and after retraining. We also propose a method to define the threshold value to distinguish poisoning attack data and clean data. Section IV shows evaluation setups, evaluation results of our proposal, and evaluation results of Curie as a comparison target. Finally, we conclude and introduce future works in Section V.

II. Related Work

A. Network-based Intrusion Detection Systems and Researches

Network-based Intrusion Detection Systems (NIDS) are widely used for observing malicious network traffic and some of them are working as Network-based Intrusion Protection Systems (NIPS). Nowadays, we equip NIDS not only at a border, like the Internet gateway, but also some observation points in intranet. The detection method is largely separated into signature based detection and behavior based detection and ML (ML-NIDS) is widely used for behavior based detection.

Researches on ML-NIDS start from comparatively ancient age. For example, Mukkamala et al. proposed ML-NIDS using Neural Network and support vector machine in 2002 [5]. Currently, there are too many successor researches in ML-NIDS. Nowadays, ML-NIDS is already used in commercial security appliances and many companies consider that ML technologies are effectively working in their security appliances. For example, Sophos Ltd. is applying deep learning protection in their Sandstorm UTM (Unified Threat Management) appliance or software [6].

B. Traffic Dataset

To promote NIDS research, we have to obtain traffic data including malicious/benign traffic data. However, it is hard for many researchers to create own experimental network which can generate both malicious/benign traffic so that many researcher use traffic dataset to promote their own research.

KDD (Knowledge Discovery and Data mining) Cup 1999 Data is one of the most famous traffic dataset [7]. It summarizes traffic data of 1999 DARPA Intrusion Detection Evaluation Dataset [8] into traffic session level so that it can list many traffic data with small file size. It is also famous for adding statistical data as a feature of a traffic session mainly derived from relationships among sessions.

Kyoto 2006+ dataset is a dataset which is generated by Song et al. [9]. The data source is honeypot at Kyoto University and they generate KDD Cup 1999 Data like traffic session level dataset with malicious/benign ground truth label given by security appliances. It equips not only statistical features existing in KDD Cup 1999 Data but also newly generated statistical features. Kyoto 2016 dataset [1][2] is an extension of Kyoto 2006+ dataset. The duration of Kyoto 2006+ dataset is 3 years, but the duration of Kyoto 2016 dataset is 9 years. Kyoto 2016 dataset also gives much more session data even in Kyoto 2006+ duration because PC and software advancement makes additional interpretation to pcap file which could not be interpreted in Kyoto 2006+ age.

C. Poisoning Attack Data Generation

Several researchers touch poisoning attack data generation and its performance.

Biggio et al. proposed SVM poisoning attack algorithm that generates poisoning attack data [3]. It is one of a gradient ascent methods and similar to a gradient descent method on Neural Network. An outlined algorithm is shown as Algorithm 1. It updates L to maximizing loss function in SVM. In this research, we use this algorithm for poisoning attack data generation.

Apruzzese et al. evaluated a poisoning attack to ML-NIDS in an experimental network with normal traffic and malware originated attack traffic [10]. They generated poisoning data with randomly increasing feature vector values from clean attack traffic data and confirmed dramatic degradation of True Positive Rate (TPR) in Random Forest, Multiple Layer Perception, K-Nearest Neighbor methods.

Algorithm 1 Biggio's SVM poisoning attack[3].

```
Input: training data $\mathcal{D}_{\mathrm{tr}}$, validation data $\mathcal{D}_{\mathrm{val}}$, feature vector
    and ground truth label of initial attack point $\{x_c^{(0)}, y_c\}$, step
    size $t$
Output: Feature vector of one adversarial training data $x_c$

Train SVM with $\mathcal{D}_{\mathrm{tr}}$
Current iterations $k \leftarrow 0$.
repeat
    Train SVM again with $\mathcal{D}_{\mathrm{tr}} \cup (x_c^{(k)}, y_c)$
    Calculate gradient of loss function $\frac{\mathrm{d}L}{\mathrm{d}u}$ with $\mathcal{D}_{\mathrm{val}}$
    Let $u$ to parallel vector to $\frac{\mathrm{d}L}{\mathrm{d}u}$
    Update adversarial training data by $k \leftarrow k+1$, $x_c^{(k)} \leftarrow x_c^{(k-1)} + tu$
until $L\left(x_c^{(k)}\right) - L\left(x_c^{(k-1)}\right) < \epsilon$
return: $x_c = x_c^{(k)}$
```

D. Countermeasure to Poisoning Attack Data

There are two directions on countermeasures to poisoning attack data. The one is a hardening a classifier training algorithm not to be affected by poisoning attack data and the other one is a method to detect and exclude poisoning attack data.

Zhou et al. have promoted research in the hardening the classifier training algorithm [11]. They proposed AD-SVM (ADversarial Support Vector Machine) which has additional constraints that is designed for considering poisoning attack data may try to maximize hinge loss. However, it gives some adversarial affect to classification performance because a typical SVM tries to minimize hinge loss.

There are several researches in the method to detect and exclude poisoning attack data. Steinhardt et al. have proposed a method to detect poisoning attack data by combining estimating a barycenter of data class and outlier detection algorithm

[12]. They also confirmed resistance to poisoning attack data. They find that MNIST-1-7 dataset and Dogfish data have high resistance to poisoning attack data but IMDB Sentiment dataset has low resistance. Taheri et al. have tried to estimate an original ground truth label which is flipped when generating poisoning attack data [13]. They utilized Neural Network for estimation and rated data as poisoning attack data if ground truth label is differed from an estimated label.

Laishram et al. proposed Curie that is an algorithm to exclude poisoning attack data generated by SVM poisoning attack [4]. An outlined algorithm of Curie is shown as Algorithm 2. Curie also exploits flipped label similar to Taheri's method. Curie firstly compresses training data to two dimensions and performs clustering with DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Then, Curie adds class label that is weighted with constant and treated as 3rd dimension. Fianlly, Cuirie calculates an average distance between samples in a same cluster. If a sample is a clean data, the distance tend to become small. If a sample is a poisoning attack data, distance tend to become large. Curie detects and excludes poisoning attack data with an above criterion.

Algorithm 2 Algorithm of Curie [4].

```
Input: $Data = (F, C)$ for inspection. ($F$ is a set of feature vector, $C$ is a set of class label)
Output: Set of vector $M$ which has excluded poisoning attack data
$PcaData \leftarrow$ PCA($Data.F$) {Compress data to two dimension}
$Clusters \leftarrow$ DBSCAN($PcaData$) {Clustering data by DBSCAN}
for $point \in Data$ do
    $point.F \leftarrow$ Append($point.F, point.C \times \omega$) {Add weighted ground truth label as 3rd dimension of feature}
    $cls \leftarrow$ GetCluster($point, Clusters$)
    $sample \leftarrow$ Sample($cls, count$)
    for $s \in sample$ do
        $s.F \leftarrow$ Append($s.F, s.C \times weight$)
        $d \leftarrow$ EucledianDistance($point.F, s.F$)
        $Dist.point \leftarrow Dist.point + d$
    end for
    $Dist.point \leftarrow Dist.point/$Size($cls$) {Calculate average distance from randomly selected 10 samples in same cluster}
    $Dist \leftarrow$ ZScore($Dist$) {Normalization with Z value}
end for
for $point \in Data$ do
    if $Dist.point \leq \theta$ then
        $Result \leftarrow$ Append($Result, point$)
    end if
end for{Choose samples that have over $\theta$ reliability}
return: $Result$
```

III. Proposal of Internal Coefficient Displacement Based Detection

A. Assuming Poisoning Attack Scenario

Figure 1 shows an assumed scenario of poisoning attack threat. A company working on security measures is working for updating a classifier of ML-NIDS to catch up with latest cyber attacks. To update classifier, the company gathers traffic data including malicious and benign traffic data from published traffic dataset. Some attacker considers attacking some organization that is using ML-NIDS of the company and the attacker try to weaken ML-NIDS to pass through attack traffic (considering backdoor attack). The attacker tries to generate poisoning attack data from an existing traffic dataset and publish it as a new traffic dataset. If the company includes the poisoning attack data to ones training dataset, the company creates classifier from polluted dataset and it may generate a polluted classifier that have a backdoor. If the polluted classifier has been published, an intention of the attacker has succeeded.

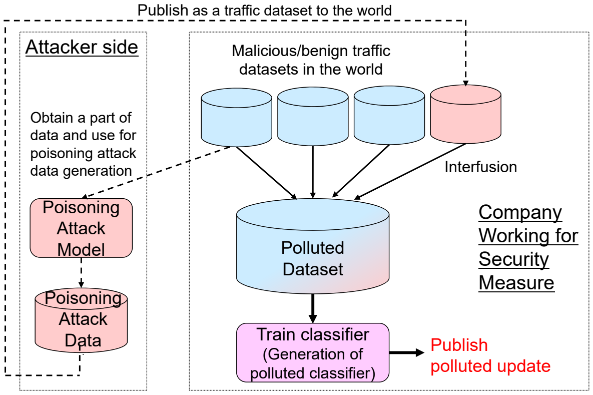

Publish as a traffic dataset to the world
Attacker side
Obtain a part of data and use for poisoning attack data generation
Poisoning Attack Model
Poisoning Attack Data
Malicious/benign traffic datasets in the world
Interfusion
Polluted Dataset
Company Working for Security Measure
Train classifier (Generation of polluted classifier)
Publish polluted update

Figure 1. Assuming scenario: perform poisoning attack by distributing poisoning attack data.

B. Idea of Internal Coefficient Displacement Based Detection

To detect a block of poisoning attack data, we assumed "Poisoning attack data are designed to confuse classification criterion (e.g., hyperplane) so that it may largely displaces an internal coefficient of a classifier" so that we considered to detect the poisoning attack data from the internal coefficient distance between before and after retraining with additional data. Based on above assumption, we also assumed "The internal coefficient distance between before and after retraining may become large value if the additional data contain poisoning attack data. On the other hand, if the additional data is only clean data, internal coefficient distance becomes moderate value." so that we created a following procedure to detect poisoning attack data.

Figure 2 represents a poisoning attack data detection method based on an above assumption. Firstly, we create a classifier from reliable datasets O. Then, we add some additional training data A to O. We obtain classifier C_O and C_{OA} from both data blocks and obtain internal coefficient vectors v_O and v_{OA} from them. Then, we calculate Euclidean distance D between v_O and v_{OA} and compare with threshold value D_{th}. If D is larger than D_{th} that means classifier has largely displaced, we judge the data block A as poisoning attack data or the data block A contains some poisoning attack data. Otherwise, the data block A becomes clean data.

As shown in Section IV, we choose SVM for a classifier so that the internal coefficient becomes a gradient coefficient vector of SVM classifier. So, we evaluated the distance of the

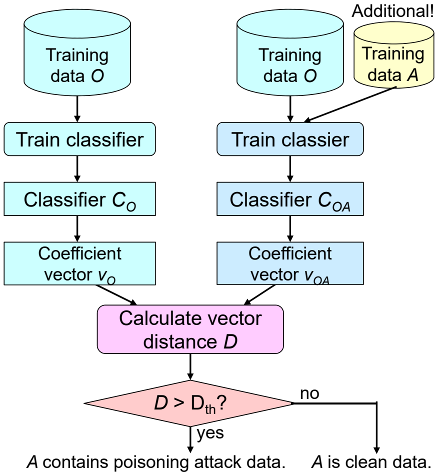


Figure 2. Proposal of internal coefficient distance based detection.

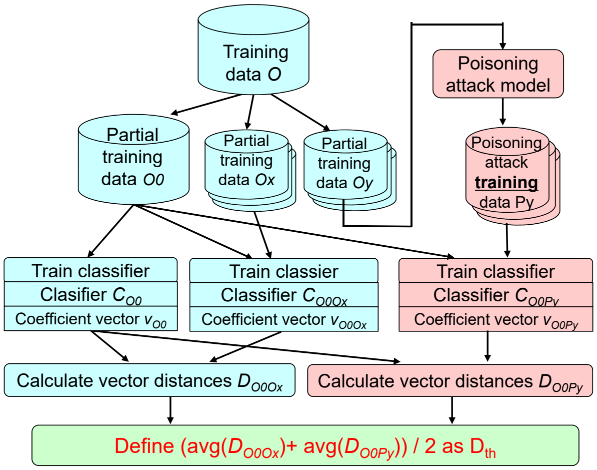

Training data O
Poisoning attack model
Partial training data O0
Partial training data Ox
Partial training data Oy
Poisoning attack **training** data Py
Train classifier
Clasifier C_{O0}
Coefficient vector v_{O0}
Train classier
Classifier C_{O0Ox}
Coefficient vector v_{O0Ox}
Train classifier
Classifier C_{O0Py}
Coefficient vector v_{O0Py}
Calculate vector distances D_{O0Ox}
Calculate vector distances D_{O0Py}
Define (avg(D_{O0Ox})+ avg(D_{O0Py})) / 2 as D_{th}

Figure 3. How to generate threshold value.

gradient coefficient vector in Section IV. But we think that many of ML algorithms have internal coefficient vectors so that this method can easily to be adopted into versatile ML algorithms.

C. Method to Define Threshold from Existing Data

To define threshold value D_{th} in Figure 2, we propose a method that generates D_{th} from existing clean data. To obtain displacement when we retrain the classifier with poisoning attack data, we generate poisoning attack training data from a part of the existing clean data.

Figure 3 shows an outlined flow of the proposal. Firstly, we divide existing clean training data O to following data blocks.

- Large size data block O_0 which is used for generating baseline classifier C_{O0}.
- Small size data blocks O_x which is used for generating classifier with additional clean data $C_{O0Ox}(x = 0, ..., n - 1)$.
- Small size data blocks O_y which is used for generating classifier with additional poisoning attack training data $C_{O0Py}(y = 0, ..., n - 1)$.

To generate poisoning attack training data, O_y is converted to P_y with an existing poisoning attack model in a poisoning attack detecting organization (e.g., security measure company).

The classifier generation part is similar to Figure 2. The baseline classifier C_{O0} is generated by training with training data O_0 and obtain coefficient vector v_{O0}. The classifier C_{O0Ox} is generated from merged data of training data O_0 and training data O_x and obtain coefficient vector v_{O0Ox}. The classifier C_{O0Py} is generated from merged data of training data O_0 and training data P_y and obtain coefficient vector v_{O0Py}. Then, we calculate distances between v_{O0} and v_{O0Ox}. We repeatedly calculate distances with different data blocks with varying x and y for n times and get an average. This average becomes an average coefficient perturbation when the classifier retrains with additional clean data. Similarly, we calculate distances between v_{O0} and v_{O0Py} and get average. This average becomes an average coefficient perturbation when the classifier retrains with poisoning attack data as additional data. We defined that the D_{th} is a middle of both averages.

In next section, we evaluate an adequacy of D_{th} definition with cross validation and compare it with an existing poisoning attack data exclusion method.

The method is partially introduced in domestic conference with malware binary feature classification [14]. This proposal extends the method with extending the method to NIDS including a time series update operation of the classifier.

IV. Evaluation

A. Experimental Setup

Before presenting the experimental results, we introduce an experimental setup. As a classifier to realize NIDS that classifies traffic session data into malicious or benign, we created a classifier with SVM. To generate poisoning attack data, we used Biggio's SVM poisoning algorithm [3] which is introduced Section II-C.

We used Kyoto 2016 [1][2] traffic dataset which is introduced Section II-B as a session level traffic dataset with malicious/benign ground truth label. We randomly picked up malicious/benign traffic sessions from November 2015 month of Kyoto 2016 dataset with keeping malicious:benign ratio to 1:1. We used 12 numeric parameters (duration, source bytes, destination bytes, same destination count, same service rate, SYN error rate, same service SYN error rate, same destination host count, same destination host and service count, same source port rate in same destination host count, SYN error rate in same destination host count, SYN error rate in same destination host and service count) of session data to generate SVM classifier.

To generate poisoning attack data, we choose 10,000 traffic session samples and generated poisoning attack data samples with Biggio's SVM poisoning algorithm [3] using Adversarial Robustness Toolbox library [15]. We confirmed that the generated poisoning attack data degrades SVM based ML-NIDS classification accuracy and classification accuracy degrades dramatically if poisoning attack data occupies more

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

TABLE I. HYPER-PARAMETER OF CURIE OBTAINED WITH BAYESIAN OPTIMIZATION.

Parameter	Value	Definition
$omega$	1030.6	Coefficient when adding ground truth to 3rd dimension (in step 4 of Algorithm 2)
$theta$	0.581	Threshold to distinct poison/clean based on average distance from cluster member (in step 16 of Algorithm 2)
$sample\_size$	36	Sampling amount from the same cluster (in step 6 of Algorithm 2)
eps	0.732	Distance to define the same cluster (in step 2 of Algorithm 2)
$min\_samples$	15	Distinct as noise if a number of samples in eps distance is smaller than this value (in step 2 of Algorithm 2)

TABLE II. RESULTS OF INTERNAL COEFFICIENT DISPLACEMENT METHOD.

Poisoning rate (num of data blocks)	Accuracy	Precision	Recall	F1 score
1% (303 blocks)	0.8861	0.8104	0.9551	0.8768
2% (147 blocks)	0.9592	0.9592	0.9592	0.9592
3% (96 blocks)	0.9430	0.9641	0.9250	0.9441
4% (72 blocks)	0.9510	0.9583	0.9446	0.9514
5% (57 blocks)	0.9421	0.9553	0.9308	0.9429
6% (45 blocks)	0.9500	0.9667	0.9355	0.9508
7% (39 blocks)	0.9558	0.9500	0.9611	0.9555
8% (33 blocks)	0.9841	0.9682	1.0000	0.9838
9% (30 blocks)	0.9825	1.0000	0.9662	0.9828
10% (27 blocks)	0.9639	0.9833	0.9465	0.9646
15% (15 blocks)	0.9500	0.9700	0.9327	0.9510
20% (12 blocks)	0.9625	0.9625	0.9625	0.9625

TABLE III. RESULTS OF CURIE METHOD.

Poisoning rate	Accuracy	Precision	Recall	F1 score
0.0%	0.7830	NaN	0.0000	NaN
2.5%	0.7902	0.8400	0.0905	0.1634
5.0%	0.7909	0.7692	0.1613	0.2667
7.5%	0.8039	0.8272	0.2528	0.3873
10.0%	0.8047	0.7838	0.3107	0.4450
12.5%	0.8074	0.7676	0.3682	0.4977
15.0%	0.8121	0.7614	0.4281	0.5481
17.5%	0.8152	0.7311	0.4814	0.5805
20.0%	0.8176	0.7400	0.5316	0.6187

than 20% of training data. These 10,000 poisoning attack data samples are treated as P_y. We choose different 12,960 traffic session samples as clean training data O. Data O is separated into 3,240 O_0 samples and 9,720 O_x samples. Data O_x and P_y are divided into 3 groups (around 3,000 samples per each group) to perform cross validation in our proposal. So, when we define threshold value D_{th}, from 1st group of O_x and P_y, we evaluate D_{th} with 2nd and 3rd group of O_x and P_y. Similarly, when we define D_{th} from 2nd and 3rd groups of O_x and P_y, we evaluate D_{th} with another groups. In this way, we achieve cross validation of D_{th} and obtain classification performance metric values.

As a comparison target to our proposal, we implemented Curie [4] from scratch. To tune hyper-parameter of Curie for traffic session samples, we used Bayesian optimization in Scikit-Optimization library [16]. Table I shows obtained hyper-parameters of Curie.

B. Evaluation results

Table II shows detection performance of our proposed internal coefficient displacement method under different poisoning rate. Poisoning rate means a rate of additional (poisoning) data block amount compared to original training data. If a number of original training data is 3,000 and additional (poisoning) data is 30, the poisoning rate becomes 1%. If additional training data is poisoning attack data and D_{th} distinguishes it as poisoning attack data, the result becomes True Positive. If additional training data is clean data and D_{th} distinguishes it as clean data, the result becomes True Negative. False Positive and False Negative classes are defined as similar. We can obtain multiple additional training data blocks from O_x and P_y so that we evaluate with many additional training data blocks as possible. For example, in 1% poisoning rate, we can create 101 additional data blocks from each group of O_x and P_y so that we evaluated with 303 times trial (101 additional data blocks per group times 3 groups) in 1% poisoning rate. The number of data blocks are noted beside individual poisoning rates in Table II.

As shown from Table II, our proposal achieves good performance because it achieves 88% accuracy even in 1% poisoning rate and achieves more than 94% accuracy at more than 2% poisoning rate. The performance of the proposal increases in proportion to the poisoning rate and it achieve quite high distinguish performance at 8% and 9% poisoning rate. At greater than 10% poisoning rate, one outlier dominated data block has generated and the outlier dominated data block degrades performance at greater than 10% poisoning rate area. We think that an advantage of our proposal comes from evaluating with data block level. In practical additional training, we add data at block level and not at individual sample level so that distinguishing at data block level may become a moderate assumption in practical viewpoint.

Table III shows detection performance of Curie under different poisoning rates. Poisoning rate 0% means “all data are clean data” so that precision becomes not a number due to both no True Positive and False Positive samples. Compared to our proposal shown in Table II, Curie increases its performance in proportion to the poisoning rate especially in a precision viewpoint, but an increment of accuracy is comparatively slow. This characteristic comes from that Curie perform detection in each sample granularity but our proposal performs detection in block of data granularity. So, there is a possibility that Curie increases performance if we treat each block of data as one sample (e.g., set a virtual averaged sample that represents the block of data). We also confirmed that Curie cannot distinguish a poisoning attack traffic session sample that have quite similar characteristic to existing clean data samples.

V. CONCLUSION AND FUTURE WORK

This paper proposes a method to identify whether newly added training data is poisoning attack data or not based on the displacement of the internal coefficient vector before and after retraining. We assumed that manipulated traffic session data for poisoning attack will largely confuse the internal coefficient vector which is an internal state of ML classifier. Thus, if the internal coefficient vector displaces largely after retraining with newly added data, we estimate that the newly added data is the poisoning attack data. We also proposed how to define the threshold value from the existing clean data.

We evaluated our proposal with SVM classifier, Biggio’s SVM poisoning algorithm, and Kyoto 2016 Dataset and

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

compered with the existing method Curie. By utilizing SVM for classifier, the internal coefficient vector is represented as the gradient coefficient vector of hyperplane in SVM classifier. We confirmed that our proposal can detect poisoning attack data effectively and achieves 0.9838 F1 score at 8% poisoning rate in best case. This performance may come from our method treat and evaluate additional training data at data block level. On the other hand, Curie gives moderate performance because Curie evaluates at individual traffic session sample level so that it cannot distinguish the poisoning attack traffic session sample that have quite similar characteristic to an existing clean data samples.

For the future extension, our proposal may give good performance in the other ML algorithms so that we want to evaluate this method with different ML-NIDS algorithms. Furthermore, we want to apply this method to some other cyber-security area such as malware detection/classification, spam detection/classification, process activity detection/classification, and so on.

ACKNOWLEDGMENT

This work was partially supported by JSPS KAKENHI Grant Numbers 23K28086 and 24K14959.

REFERENCES

[1] R. Tada, R. Kobayashi, H. Shimada, and H. Takakura, “Generating Kyoto 2016 Dataset for NIDS Evaluation (in Japanese)”, *Journal of Information Processing*, vol. 58, no. 9, pp. 1450–1463, Sep. 2017.

[2] “Traffic Data from Kyoto University’s Honeypots”, [Online]. Available: https://www.takakura.com/Kyoto_data/.

[3] B. Biggio, B. Nelson, and P. Laskov, “Poisoning Attacks against Support Vector Machines”, in *Proceedings of the 29th International Conference on Machine Learning (ICML ’12)*, Jul. 2012, pp. 1807–1814.

[4] R. Laishram and V. V. Phoha, “Curie: A method for protecting SVM Classifier from Poisoning Attack”, in *arXiv e-prints, arXiv:1606.01584*, Jun. 2016.

[5] S. Mukkamala, G. Janoski, and A. Sung, “Intrusion Detection Using Neural Networks and Support Vector Machines”, in *Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN’02)*, vol. 2, May 2002, pp. 1702–1707.

[6] Sophos Ltd., “Sophos UTM / Deep Learning Protection”, [Online]. Available: https://www.sophos.com/en-us/products/unified-threat-management.

[7] The UCI KDD Archive, “KDD Cup 1999 Data”, [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[8] MIT Lincoln Laboratory, “1999 DARPA Intrusion Detection Evaluation Dataset”, [Online]. Available: https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset.

[9] J. Song *et al.*, “Statistical Analysis of Honeypot Data and Building of Kyoto 2006+ Dataset for NIDS Evaluation”, in *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS ’11)*, Apr. 2011, pp. 29–36.

[10] G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti, “Addressing Adversarial Attacks Against Security Systems Based on Machine Learning”, in *Proceedings of 11th International Conference on Cyber Conflict (CyCon 2019)*, May 2019, pp. 1–18.

[11] Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and B. Xi, “Adversarial Support Vector Machine Learning”, in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2012, pp. 1059–1067.

[12] J. Steinhardt, P. W. Koh, and P. Liang, “Certified Defenses for Data Poisoning Attacks”, in *Proceedings of 31st Neural Information and Processing Systems (NIPS ’17)*, Dec. 2017, pp. 3520–3532.

[13] R. Taheri *et al.*, “On Defending Against Label Flipping Attacks on Malware Detection Systems”, *Neural Computing and Applications*, vol. 32, pp. 14 781–14 800, Jul. 2020.

[14] H. Shimada, S. Su, H. Hasegawa, and Y. Yamaguchi, “Gradient Variation based Poisoning Attack Data Detection for Poisoning Attacks Targeting SVM based Malware Detection (in Japanese)”, Information Processing Society Japan, Technical Reports 2022-CSEC-98, Jul. 2022, pp. 1–8.

[15] M.-I. Nicolae *et al.*, “Adversarial Robustness Toolbox v1.0.0”, Nov. 2019, [Online]. Available: https://github.com/Trusted-AI/adversarial-robustness-toolbox.

[16] T. Head, K. Kelly, and A. C. Mayes, “scikit-optimize: v0.5.1.”, Mar. 2018, [Online]. Available: https://github.com/scikit-optimize/scikit-optimize.

ISBN: 978-1-68558-186-2

Towards Automated Checking of GDPR Compliance

Pauline Di Salvo-Cilia
Aix-Marseille Univ., CNRS
Marseille, France
pauline.di-salvo-cilia@lis-lab.fr

Alba Martinez Anton
Aix-Marseille Univ., CNRS
Marseille, France
alba.martinez-anton@lis-lab.fr

Clara Bertolissi
Aix-Marseille Univ., CNRS
Marseille, France
clara.bertolissi@lis-lab.fr

***Abstract*—We propose a prototype for automating the General Data Protection Regulation compliance checking, in particular for consent-related principles. Our solution leverages provenance graphs to model compliance-related information. We present a prototype implementation of our model, based on Prolog.**

Keywords- Privacy; Data protection; GDPR compliance.

I. Introduction

In recent years, the quantity of personal data managed by systems has been growing steadily. In order to protect users and their data, the European Union (EU) has established the General Data Protection Regulation (GDPR) [4], which applies to European countries since 2018. Among the principles that are described [GDPR art.5], such as transparency, data minimization, consent, etc., we focus on four principles :

- consent compliance [GDPR art.6] : personal data is used only for purposes the user has given consent to.
- data access [GDPR art.15(1)]: a report is sent *in time* after a user request.
- data erasure [GDPR art.17] : personal data is erased *in time* after a user request.
- storage limitation [GDPR art.5(1)]: personal data must not be stored for *too long* after its last use.

Note that time intervals are specific to the system and must be adhered to without undue delay. The *data subject* (the owner of the personal data) should be informed of the status of his/her request within one month [GDPR art.12(3)], with a possible extension of up to two additional months, if necessary.

The idea of our approach is to automate GDPR compliance checking [1] [3] by storing system data and their dependencies in the form of a provenance graph [2] and specifying GDPR principles as paths to be retrieved in the graphs. Compliance checking is then realized by taking advantage of the efficient reasoning capabilities for path condition resolution provided by Prolog solvers. In Section II, we introduce the data model we use, as well as the specification of the GDPR principles to be checked. We present our prototype in Section III, and apply it on a use case in Section IV. We conclude in Section V.

II. Provenance graph model

In this work, we extend the Open Provenance Model (OPM) [2] with GDPR data. The system information is represented as a directed labelled acyclic graph, called provenance graph, where nodes and edges represent system data and their dependencies. The standard OPM model captures provenance entities called *artifacts*, *processes* and *agents*. Each dependency, with its timestamp(s), shows causality between entities: used (process on artifact) and wasGeneratedBy (artifact on process), where the timestamp indicates when the artifact was used (resp. generated); wasControlledBy (process on agent), where two timestamps give the beginning and the end of the process execution; wasDerivedFrom (artifact on artifact) and wasTriggeredBy (process on process), with a timestamp indicating when the first entity was created (resp. triggered). Note the dependencies may contain a role, used to further specify them. Timestamps are useful for compliance verification, where a total order between processes may be needed.

To reason about personal data and GDPR compliance, we have extended the nodes of the provenance graph with a list of attributes related to the GDPR context. In particular, artifacts that contain personal data are extended with an attribute *personal*, while processes are extended with an attribute *action*, identifying the purpose for which the process is executed. Consent is modeled as an artifact, generated by the consent giving process of the data subject. The consent artifact has an attribute *purposes*, specifying a list of consented purposes for the corresponding personal data. The consent artifact can be updated, thereby creating a new consent artifact (since artifacts are immutable pieces of data in OPM).

Figure 2 depicts a sample of a provenance graph representing an online forum application. User Bob creates an account before joining a group of discussion of interest to him. After creating his account, which implies to enter some personal information such as his phone number and email address, an identifier id_bob is automatically created by the system. User personal information and identifiers are represented as artifacts, with the attribute *personal* set to $True$ (for the sake of readability, node attributes are not depicted in the graph).

Bob is asked to provide his privacy preferences via the filling of a policy template provided by the system (e.g., a cookies acceptance policy). As a result, an artifact *consent_bob_v0* is created with the attribute *purposes*, whose value is a list of pairs (personal data, purpose of use). For instance, Bob gives consent for using his identifier for statistical analysis purposes, but not for sharing with third parties, i.e., purposes = [(id_bob, analysis)].

III. Prototype: Architecture and Implementation

We describe here the components and architecture of our prototype, as depicted in Figure 1.

a) Interface: Via the interface, an auditor can specify the system he wants to verify and optionally a subset of processes

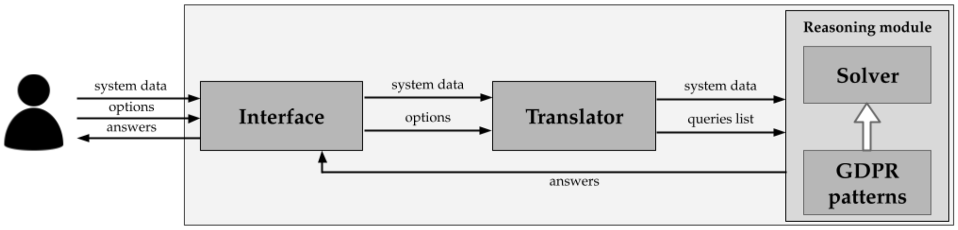


Fig. 1. Prototype Architecture.

(or a subset of personal data) he wants to focus on. The interface retrieves the system log files (in the form of a provenance graph), and possibly specific personal data, users or processes to check (by default, the whole system is audited). The auditor can also choose the GDPR principle(s) he wants to verify via the suitable menu, as depicted in Figure 3(a).

b) Translator: The translator module converts interface inputs into Prolog queries. If no option is specified via the interface, the module returns one query per GDPR principle to check, with no specific parameters (i.e., using variables that Prolog will instantiate by graph nodes). Otherwise, several queries can be returned, with parameters adequately instantiated to cover user selection. Queries and system data are sent to the Prolog solver, using the JPL library.

c) Reasoning module: The reasoning module contains the Prolog solver, which resolves path queries based on the obtained provenance graph, and the GDPR patterns specification. Each GDPR principle is encoded as a Prolog rule. Here is an extract of the pattern concerning consent compliance:

```
consent(DP, PU, T) : −
    wasControlledBy(P1, S, "owner", TB, TE),
    wasGeneratedBy(C, P1, "consent", T), isPurpose(PU, DP, C)
```

When queried, this rule verify if there exists an artifact consent `C` with the attribute ”purposes” containing the purpose `PU` for the personal data artifact `DP`. The variables T, TB, TE correspond to timestamps and $P1$ to the consent process controlled by the data subject S.

The solver returns all possible paths matching the query. In case of non-compliance, the user is given enough information to identify the issue (see Figure 3(b)).

IV. Demonstration

Consider an online forum platform, where users can, e.g., create accounts and join discussion groups. We are interested in user Bob and his activities in the system (a snapshot is depicted in Figure 2, where timestamps are expressed in minutes).

Let us suppose we want to check the compliance of Bob’s personal data processing w.r.t. Bob’s privacy preferences, as registered in his consent (see Figure 3(a)). The reasoning module receives the query, it looks for all processes using Bob’s personal data and checks if the purpose of the process has been consented by Bob.

In our example, at time $\mathtt{t} = 21$, Bob joins a group, which generates a marketing cookie using $\mathtt{DP} = id\_bob$. At time $\mathtt{t} = 26$, this cookie is used by the process $\mathtt{P} = sendCookie$, associated to a purpose $\mathtt{PU} = sendThirdParties$ (provided by the system). The solver tries to instantiate the predicate `consent(DP, PU, T)` with the previous values, however Bob has consented to use his identifier only for *analysis* purposes, i.e., `consent(id_bob, analysis, 17)`. As a result, the system is non-compliant. The solver returns the non-compliant process details, displayed in the interface (see Figure 3(b)).

V. Conclusion

We have presented an extension of the provenance model to automate GDPR compliance verification. We have also developed a proof-of-concept prototype to demonstrate the feasibility our approach. Future work includes automating provenance graph generation on various scenarios (e.g., social networks, public services, webstores) for more extensive testing. We also plan to extend the approach to other regulations, such as GDPR-UK or the United States Health Insurance Portability and Accountability Act (HIPPA).

References

[1] D. Basin, S. Debois, and T. Hildebrandt. On purpose and by necessity: Compliance under the gdpr. In *Financial Cryptography and Data Security*, pp. 20–37. Springer Berlin Heidelberg, 2018.

[2] L. Moreau, et al. The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems*, vol. 27, no. 6, pp. 743–756, June 2011.

[3] A. Tauqeer, A. Kurteva, T. Raj Chhetri, A. Ahmeti, and A. Fensel. Automated gdpr contract compliance verification using knowledge graphs. *Information*, vol. 13, no. 10, 2022.

[4] European Union. General data protection regulation, 2016. Accessed: 2024-08-23.

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

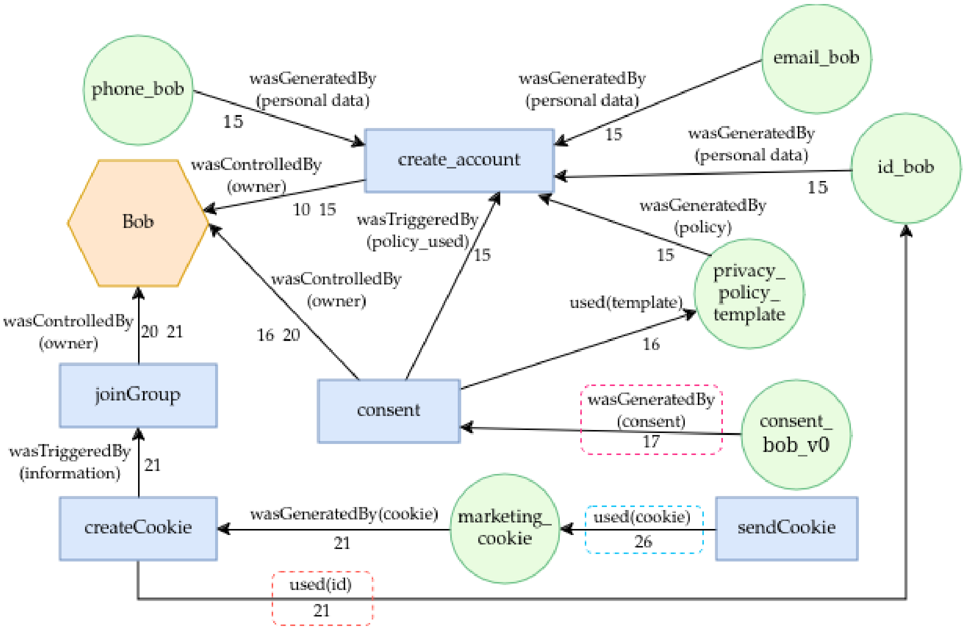
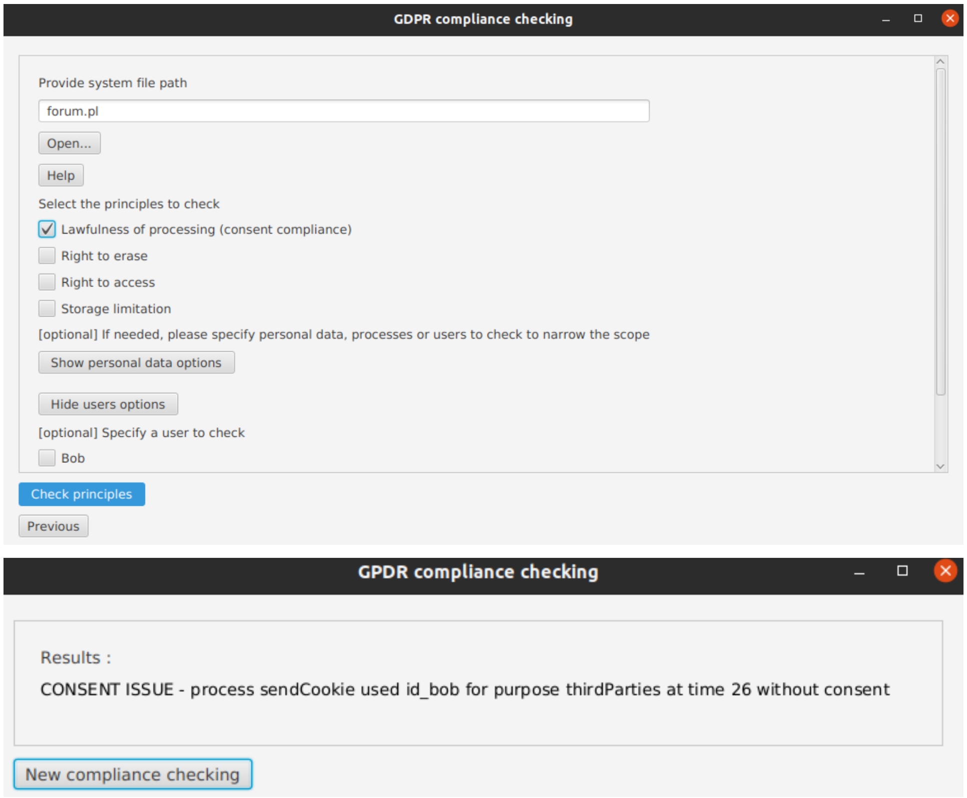


Fig. 2. Online forum application: provenance graph sample.


GDPR compliance checking

Provide system file path

forum.pl

Open...

Help

Select the principles to check

☑ Lawfulness of processing (consent compliance)

□ Right to erase

□ Right to access

□ Storage limitation

[optional] If needed, please specify personal data, processes or users to check to narrow the scope

Show personal data options

Hide users options

[optional] Specify a user to check

□ Bob

Check principles

Previous

GPDR compliance checking

Results :

CONSENT ISSUE - process sendCookie used id_bob for purpose thirdParties at time 26 without consent

New compliance checking

Fig. 3. Interface: (a) option and (b) results screens.

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

CAN Message Collision Avoidance Filter for In-Vehicle Networks

Uma Kulkarni
Institute for Secure Cyber-Physical Systems
Hamburg University of Technology
Hamburg, Germany
email: uma.kulkarni@tuhh.de

Sibylle Fröschle
Institute for Secure Cyber-Physical Systems
Hamburg University of Technology
Hamburg, Germany
email: sibylle.froeschle@tuhh.de

***Abstract*—Modern Vehicles are architecturally very complex with a large number of Electronic Control Units (ECUs) connected to each other by network buses such as Controller Area Network (CAN). When two messages with the same ID but different contents appear on the bus simultaneously, it results in a message collision. This can result in loss or delay of critical information. If done deliberately by malicious actors, it can lead to unsafe conditions. In this paper, the goal is to motivate the need for a solution against message collisions and propose a concept of a one time programmable CAN message collision avoidance filter that needs no updates or secure storage.**

Keywords-Controller Area Network(CAN); In-Vehicle Networks; Message Collisions; Filter.

I. Introduction and Related Work

Vehicles today have a large number of Electronic Control Units (ECUs) connected to each other by network buses such as Controller Area Network (CAN). CAN is one of the most widely employed network bus systems in the automotive domain due to its simplicity of implementation and low computational resource requirement. Since CAN's introduction, the vehicle has changed a lot: with more functions and more connectivity. But CAN still remains one of the preferred options. As a result, a need for add-on measures for filling in the gaps for which CAN was not designed (such as security) arises.

The in-vehicle network is commonly functionally divided into domains of grouped ECUs or nodes with gateways between the domains. Certain domains such as the powertrain CAN can be categorised as safety-critical. The nodes on a safety-critical section of CAN exchange important information that is crucial for a vehicle's seamless functioning and safety. Any missing information is undesirable and in certain cases, dangerous as e.g. pointed out in [1]. The present work summarises the problem of CAN message collisions and proposes a solution for the same. The main contributions are: (1) we present the need for a filter to avoid CAN message collisions. (2) we present a concept for one such filter with its operation and advantages.

The CAN specification [2] shows that when a transmitting node monitors a different bit value on the bus from the one it has sent, it interprets this as a bit error and the transmission is unsuccessful. It is clear that an error that is not due to a fault in the sender itself or in the bus will unfairly stop a transmission that would otherwise be successful. Furthermore, the security concerns arising from message collisions are shown in [3] and [1]. In [1], it has been shown how, based on collisions, an attacker can silence a target node. To carry out such attacks remotely, the attacker first has to pass through a gateway to disrupt a target safety-critical CAN. The idea being proposed in the present work is primarily motivated by [4] where a firewall is presented for blocking all unauthorised CAN frames from nodes categorised as high-risk. The firewall goes between every high-risk node and the internal CAN bus. It decides its actions based on a programmed pass list of acceptable message IDs. Our work aims at eliminating this list of IDs altogether and is in contrast only targeted at collision causing frames. As a result, our proposed solution does not need any updates after architectural changes. A statefull firewall is presented in [5] which also operates based on pass and block lists of frames and sequences of frames. Another closely related idea is implementing a filter in a secure CAN transceiver by NXP [6]. Its adoption would require all nodes, or at least all critical nodes, to include this secure transceiver. As opposed to this, as already stated, our approach eliminates the pass and block lists and can be a stand-alone addition to a complete critical section of CAN. In particular, in view of remote attacks, our approach exploits the directionality of attacks from gateway to the safety-critical CAN, enabling an overall security concept. The existing studies have elaborate solutions to broad frame filtering needs, but we believe that, for message collisions, our light solution will be sufficient and more efficient.

The rest of the paper is structured as follows. In Section II, we present the necessary preliminaries and our motivation for this work. In Section III, we present the filter concept and operation. Finally, we conclude and state future work directions in Section IV.

II. Preliminaries and Motivation

Some of the fields in a CAN frame that are relevant to the concept are described here: The Start Of Frame (SOF), a single dominant bit, is an indication of a node's wish to start transmission. The Identifier (ID) serves two purposes: identifying the information as well as the priority of the frame in case of simultaneous transmission attempts. The priority is decided by bitwise arbitration of the ID. Data Length Code (DLC) denotes the length of data in Data Field. CAN also offers fault confinement features. The errors that could possibly occur during a frame transmission are classified into five types: Bit, Stuff, Form, ACK and CRC of which bit error is relevant here. Every occurrence of an error results in the

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

observing node notifying this on the bus by sending error frames and adding to its Transmit Error Counter (TEC) or Receive Error Counter (REC) by following the rules of fault confinement. Successful transmissions and receptions result in the reduction of the count. The CAN nodes can be in one of the following states: (1) Error Active: TEC<127 and REC<127. The node can participate normally in communication on the bus and is capable of sending an Active Error Flag of 6 dominant bits. (2) Error Passive: $127 <$ TEC < 256 and $127 <$ REC. The node can only send a Passive Error Flag of 6 recessive bits. (3) Bus-off: TEC $\geq$ 256. A node no longer participates on the bus unless it gets out of bus-off state or is reset [2].

When two messages with the same ID appear on the bus simultaneously, it is non-consequential if the message contents are identical. But if the message contents differ, i.e., a message collision occurs, the first different bit results in a bit error. An error frame is sent and the sender's TEC increases. If this happens repeatedly, the sender acts according to the fault confinement rules of CAN and goes into error passive and eventually bus-off state. Such message collisions are undesired for multiple reasons, the most critical being: it results in unavailability of information that is critical for safety relevant functions. Also, it may be exploited by malicious actors intentionally causing denial of service by forcing a critical node into bus-off state.

The idea proposed in this work aims at preventing such collisions while causing negligible interruption to the CAN functioning. The advantages of the concept are: No changes to the existing CAN protocol. No pass list of permissible messages. So, there is also no need of storing such a list securely. It does not depend on the database of message IDs. So, there is no need for an update to the filter with every change in the architecture or message matrix. It can be implemented on a one time programmable chip making it tamper-proof and low-maintenance. It is stand-alone and does not need to be integrated with every ECU. This is favourable as the ECUs in a vehicle come from multiple manufacturers. Due to its characteristics: stand-alone, one time programmable and no-update, it will be an economic solution.

III. Filter Concept and Operation

The filter contains two transceivers, let us say transceiver-left and transceiver-right, with a microcontroller in between. As depicted in Fig. 1, on the left of transceiver-left is the gateway to the domain in question and beyond transceiver-right on the right is the CAN with critical and honest nodes (such as Powertrain CAN). The microcontroller is extremely lean with limited functions in order to keep the overhead of the filter to a minimum.

The operation of the filter can be explained with the help of four scenarios (see Fig. 2 for a part of the operation with standard CAN). Scenario 1: Bus is idle on both sides. In this case, the filter continues to monitor the bus on both sides. Scenario 2: An SOF bit is encountered by only one of the transceivers. The filter acts as a simple forwarding block. It replicates every bit as it is from the side it is received, say transceiver-right, to the other side, say left side. Dominant bits on the left side while the frame is being transmitted are forwarded to the right as they might be a part of an error flag. Scenario 3: An SOF bit is encountered on both sides of the filter simultaneously and the IDs are different. The filter does not interfere with the bitwise arbitration process. The filter compares the bits on both sides until the last but one bit of the ID. Temporary counter C counts the bits. As soon as different bits are monitored, the bit monitored by transceiver-left is forwarded on the right bus and the bit monitored by transceiver-right is forwarded on the left bus. After the arbitration phase, in normal conditions, the behaviour is the same as in Scenario 2, as shown in Step (3) in the figure. Scenario 4: An SOF bit is encountered on both sides of the filter simultaneously and the IDs are the same. The initial steps

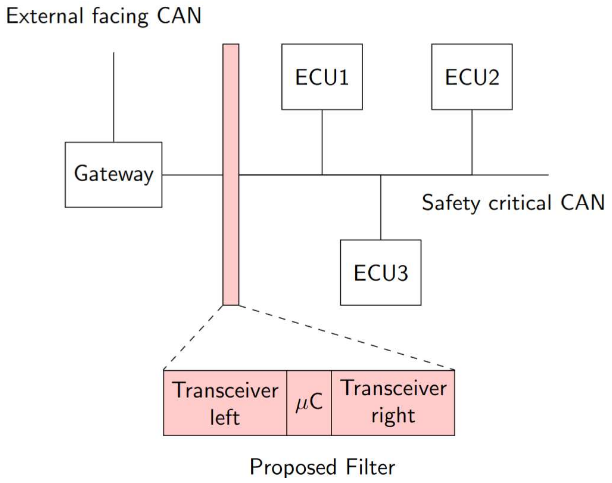


Figure 1. Example architecture of an in-vehicle CAN network with the proposed filter.

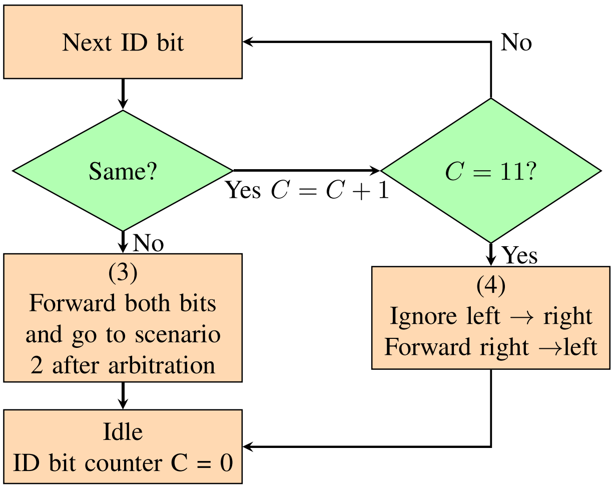


Figure 2. Operation after SOF received on both sides for a message with 11 bit ID

are the same as in Scenario 3. As soon as the last bit of the ID field is compared and it is observed that the entire ID was the same on both sides, the filter blocks the rest of the frame received by transceiver-left (non-critical side) and only forwards the remaining frame received by transceiver-right (critical side) to the other side, as shown in Step (4) in the figure. In normal cases, the filter will be faced with one of the Scenarios 1-3. In case of an impending collision, the filter behaves according to Scenario 4.

IV. Conclusion and Future work

We have proposed a lightweight filter for preventing CAN message collisions that comes with the following advantages: (1) It requires no change to the existing protocol. (2) It does not require secure storage for pass and block lists. (3) It is one-time programmable, and hence, it is tamper-proof. (4) It is stand-alone and does not need to be integrated into every ECU. These advantages make our solution economical and it will be sufficient and efficient for preventing CAN message collisions.

Future work includes implementing the filter and demonstrating its operation on a simulation platform and experimental work to analyse the delay introduced by the filter. We will also show how the filter provides a crucial component within an overall CAN security concept.

References

[1] S. Fröschle and A. Stühring, "Analyzing the capabilities of the CAN attacker," in *Computer Security–ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part I 22*. Springer, 2017, pp. 464–482.

[2] *CAN Specification, Robert Bosch GmbH, Postfach*, vol. 50, p. 15, 1991.

[3] K.-T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1044–1055.

[4] A. Humayed, F. Li, J. Lin, and B. Luo, "Cansentry: Securing can-based cyber-physical systems against denial and spoofing attacks," in *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*. Springer, 2020, pp. 153–173.

[5] T. Lenard and R. Bolboaca, "A statefull firewall and intrusion detection system enforced with secure logging for controller area network," in *Proceedings of the 2021 European Interdisciplinary Cybersecurity Conference*, 2021, pp. 39–45.

[6] *NXP Semiconductors. NXP TJA115x Secure CAN Transceiver Family*, 2019. [Online]. Available: https://www.nxp.com/docs/en/fact-sheet/SECURCANTRLFUS.pdf [last accessed 24 Sept 2024]

Prepare for the Worst, Rather Than Hope for the Best

Preparing to Recover From Major Cyber Security Incidents

Anne Coull
College of Science and Engineering
Flinders University
Sydney, Australia
email: anne.coull@proton.me

***Abstract*—As the threat landscape continues to escalate, organisational leaders are realising that they cannot prevent every cyber incident. The cyber security lens is shifting its focus toward the need for resilience, and the ability to recover from Major Cyber Security Incidents. Cyber incident recovery differs from every day IT incident recovery. The threat actors will have been in the systems domain establishing a foothold, installing malware, and exfiltrating data prior to their presence being noticed. Following the standard IT recovery playbooks will exacerbate the situation, causing confusion and delays. Preparation is the key to cyber incident and recovery readiness. This paper outlines a practical approach for IT and cyber operational teams to apply that will prepare them for major cyber events so that in the heat of an incident, they have the tools at hand, the confidence, and the capability to deal with the situation and the ability to recover within resilience appetite and tolerance.**

Keywords-cyber resilience; recovery; major cyber security incident; playbook.

I. INTRODUCTION

Cyber Security resilience describes the ability to protect against, respond to, and recover from cyber threats [16]. The preceding decade has seen significant focus and uplift in cyber security protection investment in Australian critical infrastructure. And as organisations realise that they cannot expect to prevent 100% of cyber incidents, they are shifting their attention toward preparing for them. This paper provides guidance on how organisations can move to a position where they can recover from significant cyber incidents, and that they are able do so within a reasonable timeframe. Cyber security is different from IT disaster recovery. Prior to the incident being raised, the cyber threat actor has already made-ready, ensuring they can maintain access even when discovered, finding and stealing valuable information assets, and compromising data, backups, configuration files, and applications throughout the environment. All this is achieved well before anything as visible as ransomware is triggered. Applying normal IT recovery processes will only exacerbate the problem, and extend the recovery times. In addition, cyber incident recovery necessitates teams from different areas to work together towards a shared outcome [19].

Preparation is the key [6][16]. Endeavouring to address system and environmental shortcomings whilst attempting to recover business critical systems in the heat of a major incident is not ideal. This will further delay, and may even inhibit the ability to recover. Well before the incident is experienced, the environment needs to be remediated to limit exposure of critical systems, slow lateral movement, close vulnerabilities, correct misconfigurations, tighten privileged user access, and reduce the blast radius of the incident. Section II outlines how cyber security incident recovery differs from normal IT recovery. Section III walks through the five elements that need to be addressed when developing the ability to recover from a Major Cyber Security Incident. Section IV explains the top six threat scenarios that teams should be prepared to recover from. Section V focuses on recovery testing and continuous improvement. Section VI addresses the organisational challenges of cyber resilience readiness. Section VII discusses the metrics use to uplift response performance, and the conclusion reiterates the need for early preparation, and closes the article.

II. CYBER RECOVERY IS DIFFERENT

It is common for the Information Technology (IT) technical support teams to assume that the approach for recovering from cyber security incidents is the same as that used for every-day incidents. This introduces the risk of the intended recovery actions actually making the situation worse and causing further delays. The difference with cyber security incidents is that, prior to being discovered the cyber threat actors have escalated their access privileges, established backdoors, moved laterally across the systems domain, deployed malware to compromise systems and data, altered configuration files and applications across the environment, exfiltrated valuable data and compromised or deleted backups. Their goal is to inflict the most damage on their victim, remove any opportunity of recovery, and maximise the likelihood that their target will be willing to pay, in the instance of ransomware, for example [2].

This means that cyber incident recovery differs. Backups are likely to be deleted or compromised over a period of time. The standard recovery approach exacerbates the situation by reinstalling the malware into the production environment; Restoring a backup compromised with

ISBN: 978-1-68558-186-2

malware will restart the attack process and further extend the recovery times [7].

A. *The Cyber Recovery Process*

NIST's Cyber Incident Response Lifecycle, NIST 800-61r3, incorporates the steps needed to prepare for and recover from a cyber incident (Figure 1) [16].

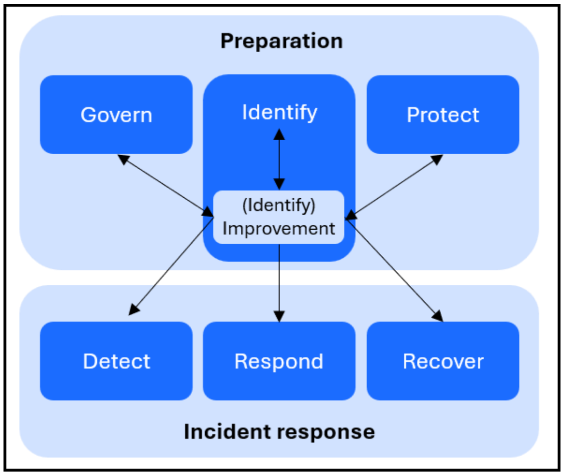


Figure 1. Incident response lifecycle model based on CSF 2.0 Functions [15][16].

In addition to the normal IT recovery steps, cyber recovery incorporates containment and eradication.

1) *Containment*

The purpose of containment is to prevent the threat spreading further across the environment, to reduce the extent of the immediate damage and the opportunity for further exfiltration. If third-parties are needed to help co-ordinate the recovery, or to provide technical guidance or hands-on-keyboard for the recovery actions, now is the time to engage them.

2) *Eradication*

During eradication, the Cyber Security Operations Centre (CSOC) and IT technical support teams work together to ascertain scale of the threat, and assess the extent of the damage or potential damage. Prior to deleting or rebuilding, snapshots of impacted systems, devices, and files need to be taken for the forensic analysts to review [7]. They may be able to map these against known threat actor Tactics, Techniques, and Procedures (TTPs), which will enable the recovery team to predict likely methods and next steps for the attacker. Compromised user accounts are disabled and credentials reset, any malware installed by the threat actor is erased, and vulnerabilities that were exploited during the attack are closed. Immutable backups are scanned to identify any missing or compromised components. Once a clean set of backups are identified, the recovery teams need to establish a clean recovery environment. The unspoiled backups are restored into this secure, clean environment, assured and tested to ensure production readiness. If preparation has not provided secured baseline configuration system-state backup from which to rebuild from, there may be a need to rebuild, reinstall and reconfigure platforms and environments from scratch. Once this has been achieved, patches will need to be installed, passwords updated, and security controls overlayed [16][18].

3) *Recovery*

Only when production readiness is assured, and the team are confident the threat has been contained and eradicated will they be ready to recover the system fully by switch it over to Production [16].

III. Preparing to Recover Rrom a Major Cyber Security Incident

A. *Pre-work*

Prior to working with any specific technology teams or systems, the first step, when preparing to recover from a Major Cyber Security Incident (MCSI), is to understand the organisation's resilience appetite and its tolerance for outages impacting customers [17]. This will provide the basis for system prioritisation, and targets for acceptable recovery times. The second step is to identify the critical IT systems that, if compromised, would have significant impact on the organisation's ability to continue to deliver services. These include access and identification control systems, such as Microsoft's Active Directory, connectivity products, such as VPN & Citrix Netscaler, and data storage, such as private and public cloud.

The IT teams supporting these critical systems will need to be heavily involved in preparing for cyber incident recovery, as they know these systems best. Support from their leaders is essential to ensuring resilience is a priority for these teams, amongst their usual workload. Top-down leader engagement is the most effective approach: Educating leaders and then their teams, ensuring objections are handled, resilience is prioritised, and skilled Subject Matter Expert (SME) resources are made available. Resilience and recovery preparation activities will need to be prioritised and incorporated into the teams' delivery plans or backlogs, to be appropriately allocated to sprints or epics, in line with other priorities in the team's backlogs.

B. *Elements contributing to recover-ability*

There are five elements to be addressed when developing the ability to recover efficiently from a MCSI impacting one or more of company's critical IT systems, in order to meet the organisation's resilience appetite and impact tolerance [17]. These are:

1) *Environmental Remediation*

Work with the IT support team, the cyber red team, and the cyber risk and controls' assurance teams to identify and prioritise environment remediation requirements, such as: vulnerabilities and control gaps;

risks and issues; insecure configurations and misconfigurations; lack of network, system and access segregations; excessive and inappropriate use of privileged access; and poor password management practices. Vendors may be able to provide scanning scripts and threat simulation capabilities to assist with identifying these remediation opportunities. But the platform support teams will have a list of items they know should be fixed. Capture and prioritise all these remediation items. Develop a plan to group them, map them to existing uplift and re-platforming projects and allocate resources to close them. This will significantly lower the initial risk of a cyber incident, slow down the threat actors' transgressions, reduce the blast radius, and streamline recovery. Funding will need to be considered for the big-ticket items, such as retiering or major platform upgrades.

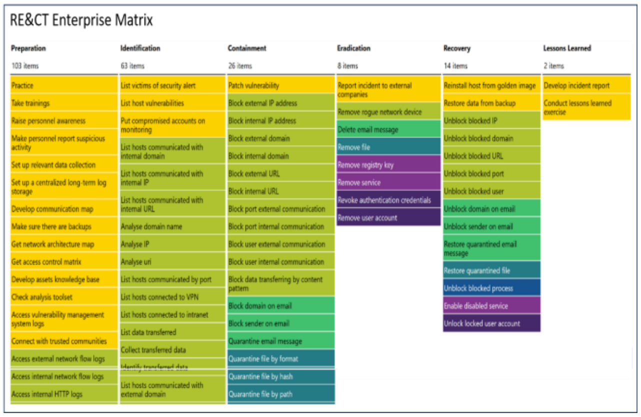
2) Recovery Preparation based on RE&CT

GitHub's RE&CT Enterprise Matrix provides comprehensive guidelines on how to prepare for efficient recovery from cyber security incidents. Based on the MITRE ATT&CK and D3FEND MATRICES, the GitHub RE&CT Enterprise Matrix outlines actions for all stages of the Cyber Incident Response Lifecycle [6][11]-[13][16]. Exhaustive examples are provided for multiple attack sequences in the RE&ACT Enterprise Matrix. They include practical actions that expose assumptions and facilitate comprehensive and complete preparation of capabilities to enable accelerated response. These include basic, but essential items, such as:

i) taking a system-state (golden) image; a snapshot of the baselevel system configuration on critical systems, and storing a clean copy of this in a secure offsite and offline location. This simple mitigation ensures the recovery team won't ever need to rebuild the entire system from scratch, by establishing immutable backups that will be accessible only to those who need them in times of crisis.

ii) building the capability to trigger bulk access revocation and re-enablement, and bulk password resets for compromised accounts, user groups and suppliers' accounts (Figure 2) [3][6].

Use the RE&CT Matrix as a guideline by reviewing each item listed, first determining if this item is relevant to the platform under review, and then assessing whether this has already been addressed, or needs to be actioned. All action items are added to the remediation list to be tracked through to completion.



RE&CT Enterprise Matrix

Preparation	Identification	Containment	Eradication	Recovery	Lessons Learned
103 items	63 items	26 items	8 items	14 items	2 items
Practice	List victims of security alert	Patch vulnerability	Report incident to external companies	Reinstall host from golden image	Develop incident report
Take trainings	List host vulnerabilities	Block external IP address	Remove rogue network device	Restore data from backup	Conduct lessons learned exercise
Raise personnel awareness	Put compromised accounts on monitoring	Block internal IP address	Delete email message	Unblock blocked IP	
Make personnel report suspicious activity	List hosts communicated with internal domain	Block external domain	Remove file	Unblock blocked domain	
Set up relevant data collection	List hosts communicated with internal IP	Block internal domain	Remove registry key	Unblock blocked URL	
Set up a centralized long-term log storage	List hosts communicated with internal URL	Block external URL	Remove service	Unblock blocked port	
Develop communication map	Analyse domain name	Block internal URL	Revoke authentication credentials	Unblock blocked user	
Make sure there are backups	Analyse IP	Block port external communication	Remove user account	Unblock domain on email	
Get network architecture map	Analyse uri	Block port internal communication		Unblock sender on email	
Get access control matrix	List hosts communicated by port	Block user external communication		Restore quarantined email message	
Develop assets knowledge base	List hosts connected to VPN	Block user internal communication		Restore quarantined file	
Check analysis toolset	List hosts connected to intranet	Block data transferring by content pattern		Unblock blocked process	
Access vulnerability management system logs	List data transferred	Block domain on email		Enable disabled service	
Connect with trusted communities	Collect transferred data	Block sender on email		Unlock locked user account	
Access external network flow logs	Identify transferred data	Quarantine email message			
Access internal network flow logs	List hosts communicated with external domain	Quarantine file by format			
Access internal HTTP logs		Quarantine file by hash			
		Quarantine file by path			

Figure 2. RE&CT Enterprise Matrix extract [6].

3) Response-Recovery Scenarios, including who does what.

Validated and assured recovery plans build capability and confidence for swift recovery.

a) Prepare cyber-specific response plans for the most common, and highest impacting threat scenarios.

The IT support teams understand the technology best and as such are major contributors to determining the most streamlined and comprehensive recovery process [16]. The playbooks need to be composed for the average person in the support team to follow, not the most knowledgeable or experienced SME. GitHub's RE&CT site and some vendor sites provides standard recovery playbooks for the common cyber incidents, such as ransomware [6][8]. Complete recovery plans and playbooks are developed by the cyber and IT teams walking through and documenting each cyber scenarios, identifying each step in the recovery process, who is doing that action, and what additional information and/ or materials are needed.

Together, the playbooks will provide end-to-end concise and easy to follow instructions for the team members to follow in the heat of a major incident. When dependencies are identified and accountabilities span more than one team, all the respective specialists need to be involved in developing the complete set of playbooks.

b) Table-top Test

Completeness is assessed through tabletop testing where the CSOC and IT support teams all work together, walking through the playbooks for the chosen scenario, with each person practicing performing their role in the playbook. Testing the set of playbooks end-to-end in conjunction with the CSOC will quickly highlight omissions and refinements needed. Tabletop testing should be repeated until all parties are willing to sign-off on the complete set of playbooks for that scenario.

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

4) *Vendor engagement to assist with MCSIM.*

Vendor involvement in cyber incidents will depend upon the organisation’s sourcing strategy, previous agreements and contractual arrangements. Specialist IT vendors, advisory, and cyber insurance companies offer services to assist with coordinating Major Cyber Security Incident Management (MCSIM) during the event. This will only be effective if they have a clear understanding of the environment, and/ or up-to-date architectural documentation, and recovery playbooks available [16]. Capacity to manage the situation during the event will be heavily reliant on the amount of preparation performed across all the relevant teams prior to the incident.

When vendors are active in supporting the applications and infrastructure, they will have a hands-on-key board role to play during the recovery preparation and incident recovery. In this instance, it is imperative that these personnel participate in the preparation and testing activities, and that arrangements are made during the preparation phase to ensure this assistance will be made available when it is needed. The supplier contracts may need to be updated to include this requirement.

5) *Full MCSI Simulation(s)*

Complex to plan and organise, but well worth the effort, full MCSI simulations enable teams to practice MCSIM and recovery in near-real circumstances. This builds confidence for both the participants as well as the stakeholders observing, and exposes gaps in the preparation plans and playbooks that would otherwise only be discovered during a real incident.

Early engagement with the Crisis Management Teams to garner support and establish communications and co-ordination plans will ensure the simulation is as near to real as it can be.

6) *Post Incident Review*

Whether a simulation or a real event, a Post Incident Review (PIR) provides opportunity for those involved to debrief, capture lessons learned, and apply these to improve the process for next time. The focus is on what worked well, what was needed that wasn’t easily available. Aspects to be addressed for next time should all be captured and actioned appropriately to ensure readiness improves with each instance.

IV. Recovery Scenarios

A practical, structured approach to building the recovery plans and playbooks will ensure greatest benefit for least effort.

A. Focus on common scenarios

Rather than attempt to develop playbooks that address every attack sequence, organisations can be prepared for the majority of potential situations by focusing on the most likely, largest scale and biggest impacting MCSI threat scenarios they will need to be able to recover efficiently from. By preparing for these scenarios, incident management and recovery teams will be in a strong position to recovery from most:

1) *Nation State Actor*

The cyber security intelligence team will be able to provide insight into the likely nation state threat actors and their typical TTPs. As an example, the People's Republic of China’s (PRC) Volt Typhoon has been active since at least 2021. This group has been observed targeting critical infrastructure organisations where it has been actively performing information gathering and espionage. More recently Volt Typhoon has been attributed as the cause of critical infrastructure outages across the United States. The Volt Typhoon TTPs are based around “stealth in operations using web shells, Living-Off-The-Land (LOTL) binaries, hands on keyboard activities, and stolen credentials” [12].

2) *Ransomware*

Ransomware is highly visible and noisy due to its direct impact on the business users. By the time the threat actor has triggered the ransom message, files will have been exfiltrated, encrypted, backups deleted or compromised, configuration files, applications and data sets across the domain infected with malware. This can take moments or weeks, but the recovery team can safely assume that they are dealing with a broadly compromised environment. Two-way communication with the impacted users will help ascertain the extent of the impact as well as providing confidence to the users that their data will be recovered, in some form.

Recovery steps will need to address every aspect of the infection in order to contain and eradicate it, including and well beyond the encrypted files and backups. Depending on the extent of the infection, systems may need to be recreated. Preparation will include implementing regular immutable system-state backups along with clean, secure recovery environments for the recovery of critical systems.

3) *Dormant Threat (Prestaging)*

Dormant threat is very similar to ransomware in the early phases. This is when the threat actor is discovered before they trigger the blast. Malware payloads will have been deployed across the environment, including backups. The recovery team can expect that data, systems, and configuration files are already compromised. Containment will be similar to ransomware, without the need to replace encrypted files or the corresponding noise from the impacted users.

4) *Third Party*

Ahead of the event, the recovery team will need to generate a list of critical third-party suppliers, to understand the services they provide and their methods for access. In the situation when a supplier is compromised, the recovery team needs to be able to remove connectivity between the organisations swiftly

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

and completely. Access accounts will need to be disabled in a seamless and timely manner while the supplier focuses on containing the threat in their environment. Business processes will need to be pre-prepared to ensure they can function independently through this period.

Detection testing should determine if the infection has spread from the suppliers into the primary organisation's environment, and appropriate containment actions taken if it has. When the supplier can confirm they have completely eradicated and recovered from the threat, only then should the connection between the two organsations be re-established, and the vendor's user accounts re-enabled.

5) *Zero Day – No patch available*

When a critical system is vulnerable to a threat actively exploiting an unpatchable zero day, then compensating controls need to be implement to protect the critical system. The simplest mitigation is to take the platform offline, but this may not be possible for key systems used by the business so alternatives need to be investigated and tested.

6) *Supply Chain*

The organisation needs to be prepared for when the operating system or software update has already been infected, as was the case in the SolarWinds compromise [10]. Preparing for a supply chain compromise involves a full review of the software deployment strategies to ensure software is always deployed in stages and tested prior to full deployment. This will ensure only limited platforms/ devices are impacted. Additionally, all deployments will need to have a rollback strategy. This may be a straight-forward as uninstalling the update, or may involve taking a backup prior to deployment to enable an immediate restore if/ when it is needed.

B. *Building recovery plans and playbooks*

In the heat of a major incident, the recovery teams will have limited capacity to guess the next step or make-it-up on the fly. They will need to have a complete set of logically structured recovery plans with comprehensive playbooks to address all the interdependencies and actions to be performed by each of the teams involved. Teams include the Crisis management team, to co-ordinate external communications and business continuity, the CSOC, who typically co-ordinate cyber incident response and recovery activities, the relevant IT support teams, any vendors involved as incident co-ordinators or extended support team members with hands-on-keyboards, business and technology leaders, communications specialists etc.

In addition to the playbooks, the recovery plans include fundamentals, such as: lists of responsible personnel for each system with contact details, and rosters for extended outagess; pre-established communications plans and virtual war-rooms and bridges for keeping the resolver teams and other stakeholders up to date; data capture methods for later forensics activities; and handover procedures to ensure progress continues smoothy for outages spanning more than twelve hours [16].

While basic playbooks can be sourced online through CISA [3] Microsoft [8], for example, these are very generic and do not relate to the specific environment in the organisation. Hence, their value is limited. Alternatively, if the organisation has engaged a vendor to lead their critical incident management, then this vendor may be leaned upon to provide the CSOC playbooks for the IT playbooks to plug into.

V. Recovery Testing and Continuous Improvement

Recovery plans are initially table-top tested with each person walking through their role in the playbook. Each IT team and the CSOC team bring and use their respective playbooks to ensure they fit together, end-to-end. Regular reviews, every 6-12 months, ensure the approach is continuously improved throughout the process. Once complete, these recovery plans need to be reviewed and updated, as part of operational readiness, whenever there is a change.

Once a set of recovery play books has been completed, a full MCSI simulation, with each person performing their role, is the most effective way to test the plans and identify any gaps ahead of the real event. If the MCSI simulation identifies significant gaps and delays, these need to be addressed and retested within 3 months. Once these simulations run more smoothly, MCSI simulation should be run every 6 months to ensure everyone knows their role in the event of a major incident, or cyber-initiated crisis. The cyber security intelligence team can provide guidance on suitable scenarios for MCSI simulations, based on what they are observing in the cyber threat landscape.

When an organisation has been significantly compromised, access to operational document storage systems may be hampered. It is recommended that an alternative site is established in which to store crisis management and MCSIM documentation. This will ensure these essential instructions are available when they are needed.

VI. Organisational Challenges

A. *Resistance and avoidance*

Both the CSOC and the IT personnel will be busy dealing with their day-to-day response and support activities. Leader intervention will be needed to direct them to prioritise these MCSI preparation activities. Fear of failure and lack of cyber awareness may trigger resistance in those who should be involved. Resistance comes in many forms. The most common is avoidance: "don't understand the ask," "don't have capacity to get involved" or "too busy". These will need to be actively and persistently addressed by leadership.

B. Lack of cyber knowledge

It cannot be assumed that the IT personnel have any understanding of cyber security, or any of the additional considerations and actions required when recovering from a cyber incident. This is best taught early during or before the preparation phase, rather than in the heat of a major incident. Education sessions need to start with the very basics. With explanations of how cyber attacks work, the fundamentals of TTPs, and the types of systems and data being targeted [9]. Compromise scenarios should be explained in detail so these IT professionals begin to understand what they will be facing in the event, and how they can be prepared.

VII. Metrics

Two sets of metrics need to be considered. Those relating to recovery readiness, and those that relate to the recovery times.

1) Recovery readiness

Recovery readiness is just that. How ready are the teams to recover from a MCSI. Readiness can be assessed by tracking the completeness and effectiveness of the preparation products outlined: Cyber incident recovery plans complete, end-to-end, for the top six cyber scenarios; Table-top testing completed within the last 6 months, and all shortcomings addressed; Remediation items addressed; Control gaps closed; React matrix preparation steps completed. Full crisis simulation testing completed within the previous 2 years for this technology platform; and Contact lists maintained.

2) Recovery times

MCSI are less frequent than IT major incidents. This limits the value of measuring Mean Time To Recover (MTTR) for a MCSI, but it does not detract from the ability to measure recovery times for full crisis simulations. By setting targets to reduce MTTR, teams identify and address delays, with the ultimate objective of aligning recovery times with the organisation's resilience appetite and impact tolerance.

VIII. Conclusion

Compromise of critical systems can cripple an organisation. Preparation will mean the difference between taking minutes or hours to recover from a MCSI rather than days, weeks or even months. A structured approach to prepare for and practice managing such incidents will build corporate capability and confidence in those responsible. This paper outlined a practical approach that organisations can apply when bringing together their IT and CSOC team to prepare for the worst, rather than hope for the best.

References

[1] ASD, "Essential eight maturity model," Australian Signals Directorate, Australian Cyber Security Centre, Available from: https://www.cyber.gov.au/resources-business-and-government/essential-cyber-security/essential-eight/essential-eight-maturity-model, accessed 24 Aug 2024.

[2] M. Benmalek, "Ransomware on cyber-physical systems Taxonomies, case studies, security gaps, and open challenges," [p.1, p.190, 2023], Available from: https://www.researchgate.net/publication/377211197_Ransomware_on_cyber-physical_systems_Taxonomies_case_studies_security_gaps_and_open_challenges, accessed 24 August 2024.

[3] CISA, "Federal Government Cybersecurity Incident & Vulnerability Response Playbooks," 2024, Available from: https://www.cisa.gov/sites/default/files/2024-03/Federal_Government_Cybersecurity_Incident_and_Vulnerability_Response_Playbooks_508C.pdf, accessed 15 July 2024.

[4] CISC, "Security of Critical Infrastructure Act 2018 (SOCI)," Available from: https://www.cisc.gov.au/legislation-regulation-and-compliance/soci-act-2018, accessed February 2024.

[5] J. Ford and H. S. Berry, "Leveling Up Survey of How Nation States Leverage Cyber Operations to Even the Playing Field," 2023, Available from: https://www.researchgate.net/publication/371084176_Leveling_Up_Survey_of_How_Nation_States_Leverage_Cyber_Operations_to_Even_the_Playing_Field, accessed July 2024.

[6] Github, "RE&CT Enterprise Matrix, MITRE ATT&CK® Navigator v2.3.2," Available from: https://atc-project.github.io/react-navigator/, accessed June 2024.

[7] G. Johansen, "Digital Forensics and Incident Response: Incident response tools and techniques for effective cyber threat response, " Packt Publishing, 2022.

[8] Microsoft Learn 2022, "Incident response in the Microsoft Defender portal - Microsoft Defender XDR," Available from: https://learn.microsoft.com/en-us/defender-xdr/incidents-overview#incident-response-workflow-example-in-the-microsoft-defender-portal, accessed 15 July 2024.

[9] Mandiant, "APT1: Exposing One of China's Cyber Espionage Units," Available from: https://www.mandiant.com/resources/reports/apt1-exposing-one-chinas-cyber-espionage-units, accessed July 2024.

[10] MITRE ATT&CK^1, "SolarWinds compromise, campaign C0024," Available from: https://attack.mitre.org/campaigns/C0024/, accessed February 2024.

[11] MITRE ATT&CK^2, "Matrix - Enterprise," Available from: https://attack.mitre.org/matrices/enterprise/, accessed February 2024.

[12] MITRE ATT&CK^3, "Volt Typhoon, BRONZE SILHOUETTE, Group G1017," Available from: https://attack.mitre.org/groups/G1017/, accessed April 2024.

[13] MITRE D3FEND 2023, "D3FEND Matrix," Available from: https://d3fend.mitre.org/, accessed July 2024.

[14] NIST^1, "Security and Privacy Controls for Information Systems and Organizations (nist.gov)," Available from: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf, accessed April 2024.

[15] NIST^2, "The NIST Cybersecurity Framework (CSF) 2.0," [p.15], Available from: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf, accessed February 2024.

[16] NIST^3, "NIST SP 800-61r3 initial public draft, Incident Response Recommendations and Considerations for Cybersecurity Risk Management: A CSF 2.0 Community Profile," [p.1, p.10. p.22], Available from: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r3.ipd.pdf, accessed 24 August 2024.

[17] PWC, "Operational resilience: hoe to set and test impact tolerances," Available from: white-paper-on-impact-tolerances-feb-2020.pdf (pwc.co.uk), accessed 27 August 2024.

[18] D. Schlette, M. Caselli and G. Pernul, "A Comparative Study on Cyber Threat Intelligence: The Security Incident Response Perspective" in IEEE Communications Surveys & Tutorials,

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

vol. 23, no. 4, pp. 2525-2556, Fourthquarter 2021, doi: 10.1109/COMST.2021.3117338. Available from: https://ieeexplore.ieee.org/document/9557787, accessed 24 August 2024.

[19] J. Steinke, B. Bolunmez, L. Fletcher, V. Wang, A. J. Tomassetti, K. M. Repchick, S. J. Zaccaro, R. S. Dalal, and L. E. Tetrick, “Improving cybersecurity incident response team effectiveness using teams-based research,” 2015, IEEE Security & Privacy, vol. 13, no. 4, pp. 20-29, July-Aug. 2015, doi: 10.1109/MSP.2015.71, Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7180274, accessed 24 August 2024.

GenAttackTracker: Real-Time SCADA-based Cyber Threat Detection Through Scoring and Bayesian Model Integration

Fatemeh Movafagh and Uwe Glässer
School of Computing Science, *Simon Fraser University*
British Columbia, Canada
Email: {fma44, glaesser}@sfu.ca

***Abstract*—The increasing sophistication and evolving nature of cyber threats pose significant risks to critical infrastructure systems. This research introduces GenAttackTracker, a novel algorithmic framework designed for real-time detection and interpretation of cyber threats in Supervisory Control and Data Acquisition (SCADA) systems. By integrating dynamic anomaly scoring with hierarchical Bayesian modeling, GenAttackTracker enhances situational awareness for identifying potential security breaches in operational technology environments. This robust mechanism contributes directly to enhancing cyber resilience by improving threat detection in critical infrastructure systems, an essential component of ensuring the continuity and security of mission-critical processes. The framework leverages primary data from SCADA systems and secondary contextual data sources, termed Suspicious Activity Markers (SAMs). Through Bayesian inference, the model continuously updates its understanding of the system's security status, allowing informed decision-making.**

Keywords*—*Cyber Resilience; Critical Infrastructure Security; Cyber-Physical Systems; Supervisory Control and Data Acquisition (SCADA); Online Threat Detection; Bayesian Inference, Anomaly Detection; Suspicious Activity Markers (SAMs); Machine Learning; Real-Time Cyber Threat Detection.

I. Introduction

Cyber threats continually evolve, exerting new capabilities and enhancing their metamorphic nature to evade detection by legacy antivirus products. With evermore sophisticated threats, such as malware-free intrusions and zero-day exploits targeting Critical Infrastructure (CI), cybersecurity breaches become more inevitable—leaving infrastructures on which we all depend at high risk of global threat activity [1]. This reality amplifies fears of catastrophic events tailored to incapacitate CI systems in key infrastructure sectors.

The research project presented here aims at enhancing CI protection by reinforcing security and resilience of mission-critical Operational Technology (OT) against advanced cyber threats. OT is vital to industrial process automation as used for many types of CI facilities, which are often highly interconnected, mutually dependent systems [2]. In manufacturing and production, process automation frequently hinges on OT to interoperate with the physical environment, where Industrial Control Systems (ICS) monitor and control physical processes, devices, and infrastructures. Most prominently, Supervisory Control and Data Acquisition (SCADA) architectures allow large-scale processes to span multiple sites and work over large distances. A SCADA-based OT system is a Cyber-Physical System (CPS) that enables supervisory process control by capturing real-time data of the infrastructure's operational status. Industry sectors using SCADA include manufacturing, oil and natural gas, electrical generation and distribution, maritime, rail, and utilities [3].

With the paradigm shift to Industry 4.0, intelligent process control aims at even tighter integration of digital control loops powered by AI, embedded computing, robotics, and Internet of Things (IoT) with technical processes in the physical environment. This trend inevitably increases fragility of process automation, making OT more vulnerable by amplifying the risk of cascading and escalating failures. Beyond extensive disruptions of critical services, highly orchestrated attacks can result in disastrous physical damage caused by triggering cascading malfunctions to overload mission-critical system components.

Considering that complete security of network technology may be unattainable, the focus shifts to risk mitigation and remediation. Our work aims at proactive measures that reduce the likelihood and the potential impact of severe cyberattacks. Traditional risk mitigation methods are often inadequate for addressing advanced cyber threats due to their highly sophisticated and evolving nature. The gravity of this situation calls for advanced analytical models and algorithmic methods to ensure that cyber situational awareness keeps pace with the evolving threat landscape. Artificial Intelligence (AI) is instrumental in detecting and interpreting abnormal OT system behavior by continuously analyzing supervisory control data streamed from system operations. Abnormal behavior patterns can signal imminent threat activity after a security breach. A timely response launching countermeasures is critical to contain any intrusion before it can spread laterally across wider networks. The dynamics and anatomy of intricate attack scenarios requires advanced analytical models and algorithmic methods for turning cyber situational awareness into actionable intelligence in real-time.

Research Question. For OT systems relying on supervisory control system architectures, such as SCADA, we consider the following research question: *how can contextual data and information from secondary threat intelligence sources substantiate evidence of changes in the system's security status derived from online analysis of control data?* Fusing data and information from a number of causally related events may arguably result in more accurate situational awareness as baseline for online inference and decision-making processes.

Methodology. Inevitable uncertainty due to lack of ground truth is problematic for the reliable detection and interpretation of unexpected behavior patterns relevant a system's security status and also increases the rate of false positives. A Bayesian modeling approach can significantly improve the outcome. Bayesian inference promotes frequent updating of conditional probabilities as new information becomes available, providing a dynamic perspective of security threat levels. Thus, the more technical question is: *how can Bayesian inference and the integration of contextual data and information, termed Suspicious Activity Markers (SAMs), enhance situational awareness of cyber threat activities targeting the operation of CI systems?*

Contribution. The novel contribution of this paper is GenAttackTracker, a generic analytical framework for online detection and interpretation of abnormal behavior patterns in supervisory control data streamed from a mission-critical OT system. Combining dynamic attack scoring with Bayesian inference to fuse results from control data analysis with real-time contextual information into actionable threat intelligence, the model uses an end-to-end pipeline for stream-based anomaly detection with three phases: behaviour prediction, inference and interpretation. Our earlier work [4] outlines the concept, while this work describes the technical realization and presents experimental results.

The remainder of the paper is organized as follows. Section II explains basic concepts and discusses related work, while Section III defines the technical problem. Next, Section IV describes the methodological development of the algorithmic framework and explains the core model of GenAttackTracker. Section V presents the experimental setup and the resulting insights, and Section VI concludes the paper.

II. Background and Related Work

Automation enables stable operation of OT: the devices and the machinery that monitor and control physical processes [2][3]; it enhances efficiency, quality of service delivery, productivity and safe operation of critical assets. Supervisory control of the cyber-physical system status is critical for issuing alerts and initiating an emergency shutdown operation when abnormal behavior patterns approach or violate defined safety margins.

A. Online Anomaly Detection

Supervisory control data is time series data to be interpreted as streamed real-value measurements taken at regular time intervals. Discordant patterns that do not match the expected normal system behavior but appear to occur "out of place" are called anomalies or outliers. Online detection of anomalous behavior in time series data streamed from the operation of an OT infrastructure can be a very challenging problem:

- Identifying anomalous behavior patterns requires learning normal behavior to train a robust machine learning model that not only fits previously observed data but also carries over to unobserved data. Developing such a model is usually not a trivial task.
- Anomalous patterns generally occur for various reasons, such as equipment failures, manual control intervention, and unauthorized tampering with control settings. Thus, an even more intricate problem is to differentiate the typically few anomalies of interest—above all, suspicious abnormal behavior indicating a potential security threat—from the vast majority of anomalies caused by noise, seasonality or other trends that are irrelevant to security.

Real-world physical processes are notoriously liable to difficult to predict "external" factors, such as fluctuations in demand and supply, technical instabilities, component failures et cetera. These phenomena result in hard to predict variance in the data—commonly referred to as "noise".

B. Suspicious Activity Markers

Indicators of Compromise (IOCs) are traditionally used in digital forensics to identify artifacts left behind by attackers, such as malware signatures, unusual traffic patterns, or file hashes. These are crucial in post-incident investigations, helping to trace and understand the extent of a breach [5][6]. IOCs are typically reactive, meaning they are often identified postmortem after the damage has already occurred [7].

In contrast, we present Suspicious Activity Markers (SAMs) as a concept aiming at real-time detection of threat activity before a cybersecurity compromise fully manifests. SAMs are akin to Indicators of Attack (IOAs), which have been promoted in industry contexts—originally by CrowdStrike—but with a distinct emphasis. IOAs generally focus on recognizing the Tactics, Techniques, and Procedures (TTPs) used by attackers. These indicators aim to detect cyberattacks at an early stage, potentially before significant harm is done and it is observable before the attack is fully unfolded. An IOA security strategy focuses on detecting the attacker's intent, enabling early intervention. Such indicators can assist security teams in intercepting even unknown types of attacks [7]–[9]. However, the definition of IOAs is vague and overlaps with IOCs, leading to potential confusion [10]. Examples of IOAs include but are not limited to [7]:

- Communication between public servers and internal hosts, indicating possible unauthorized data transfer;
- Connections through non-standard ports;
- User logins from multiple locations, potentially indicating stolen credentials.
- Unusual spikes in SMTP traffic;
- Internal hosts communicating with countries the business does not serve;
- Numerous honeytoken alerts from a single host.

Our specific focus here is on using SAMs as a secondary data source to corroborate findings from the primary source, i.e., supervisory control data. We define SAMs as follows:

Definition of SAM: A SAM is a contextual observation that provides additional insight into the operational security status of an SCADA system. SAMs are not intended to identify an attacker's intent directly, but rather to refine the understanding of potentially anomalous activities detected in supervisory control data. By integrating SAMs with the primary data source, we aim to reduce false positives and improve the accuracy of detecting anomalies of interest, i.e., cyber threats to mission-critical infrastructures.

In previous works like [11]–[15], the integration of secondary auxiliary metrics into anomaly detection frameworks has been explored. Our approach differs significantly though in terms of generalization and method integration. Our focus is on using SAMs as secondary data sources to dynamically update our belief systems. This approach allows for a more refined and contextually aware detection mechanism.

C. Bayesian Analysis

Integrating contextual information from multiple sources can improve the effectiveness of cyberattack detection [16]. Bayesian modeling offers effective solutions for security threat detection and information fusion under uncertainty [4]. These methods can integrate heterogeneous data sources, including sensor networks and soft information, to improve anomaly detection in cybersecurity [17][18]. Bayesian models are particularly suitable for cyberattack detection due to their ability to update probabilities as new evidence is observed in addition to incorporating uncertainty. The continuous updating process makes Bayesian analysis and inference ideal for dynamic environments where attack patterns evolve over time [17]. Hierarchical Bayesian models, in particular, are well-suited for this context as they allow for multi-level aggregation of information, which is crucial for systems with distributed components and diverse data sources [19].

D. AttackTracker Framework

Attack Tracker is a distributed analytic framework designed for real-time detection of cyber threat activities in supervisory control system data [20][21]. By employing a scalable hierarchical network of detector agents to monitor various levels of the control system, the orchestration of threat detectors naturally matches the organization of SCADA architectures. Local detectors focus on identifying anomalies within subsystems, while higher-level detectors aggregate this information to detect and assess threats in different system components.

The framework consists of several key components. The Behavior Predictor learns and predicts normal subsystem behavior to identify any deviations or anomalies. The Inference Engine processes observations, assigns attack scores, and aggregates results from lower-level detectors to enhance system-wide threat detection. The Dynamic Scoring method adjusts detection thresholds dynamically based on the current system state and historical data, effectively handling contextual noise and reducing false positives.

Attack Tracker has been successfully applied to the Secure Water Treatment (SWaT) testbed [22] (see also Sect. V-A), demonstrating its capability to detect a wide range of cyber threats in SCADA-based CI systems.

III. Problem Definition

This section defines the problem of identifying anomalous behavior linked to a cyberattack in the supervisory control data streamed from the operation of a SCADA-based OT system. Henceforth, SCADA data is simply referred to as control data.

A. Primary and Secondary Data Sources

We categorize available data and information sources into a primary source and multiple secondary sources:

- **Primary Source:** Control data, formally represented as a multivariate time series $X = (x_t)_{t=1}^{T}$, for $T \in \mathbb{N}$, consists of discrete multivariate measurements x_t from sensors and actuators monitoring and controlling the system at time t, where t refers to logical rather than physical time. This data is the foremost anomaly detection input, providing insights into the operational state of the infrastructure.
- **Secondary Sources:** A given collection of SAMs is characterized as a set of 3-tuples, $\text{SAM} = \{SAM_j\}_{j=1}^{N}$, where each SAM_j has three attributes, $(type_j, p_j, weight_j)$. Here, $type_j$ denotes the type of suspicious activity; $weight_j$ indicates the importance or impact of SAM_j; and p_j represents the probability value indicating the likelihood of an attack being in progress. SAMs provide contextual information that can enhance the detection capabilities by highlighting potential threat indicators.

In order to effectively utilize both primary and secondary sources, we must establish a systematic approach that integrates multiple data streams, allowing for a comprehensive assessment of potential threats within the SCADA system.

At any time step $i > l$, with $i, l \in \mathbb{N}$, the supervisory control data to be analyzed at time i is given by X_i, for $X_i = (x_{i-l}, \ldots, x_i)$, while the corresponding activity marker values to be considered at time i are given by SAM_i, with $SAM_i = \{(\text{type}_{i,j}, p_{i,j}, weight_{i,j})\}_{j=1}^{N}$. The invariable length of the sliding observation time window is $l + 1$.

Objectives:

- Calculate the Anomaly Score AS_i at step i for $\mathbf{X}_i$, relative to the estimated behavior $\hat{\mathbf{X}}_i$, to assess any deviations of the actually observed from the expected normal behavior:
$$\text{AS}_i = f(X_i, \hat{X}_i), \text{with } f : \mathbf{X} \times \mathbf{X} \mapsto \mathbb{R}^+,$$
where the real-valued function f quantifies the result.
- Update the posterior probability of an attack in progress, given the observed supervisory control data and the values of contextual markers (SAMs) at timestep i:
$$P(\text{Attack}_i | X_i, \text{SAM}_i) = \frac{P(X_i|\text{Attack}_i) \cdot \sum_{j=1}^{N} (p_{i,j} \times weight_{i,j}) \cdot P(\text{Attack}_i)}{P(X_i) \cdot P(\text{SAM}_i)} \tag{1}$$
where:
  - $P(X_i|Attack_i)$ is the likelihood of observing the control data given an attack, informed by the Anomaly Scores AS_i,
  - $P(Attack_i)$ is the prior probability of an attack,
  - $P(X_i)$ and $P(SAM_i)$ are the marginal probabilities of the control data and SAMs, respectively.

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

The challenge lies in effectively integrating control data and additional contextual information to provide a comprehensive and real-time assessment of the threat level. A key difficulty is determining the extent of deviation from normal behavior that should be considered indicative of an attack, rather than a benign anomaly. This challenge of selecting the appropriate threshold for deviation is critical, as setting it too low may result in false positives (incorrectly identifying normal behavior as an attack), while setting it too high could lead to missing true positives (failing to detect actual attacks). Hence, developing a methodology that accurately analyzes these deviations and updates the likelihood of an attack based on new data and contextual markers is essential for more reliable threat detection. Figure 1 provides an overview of such a methodological framework. It depicts how primary supervisory control data X_i are processed by a machine learning model to generate an anomaly score. This score is compared against a threshold, with secondary data SAM_i integrated via a Bayesian model to refine the final attack likelihood score. The Bayesian analysis and inference are explored in detail in Sections IV and V.

B. Levels of Abstraction

To ensure a clear and structured approach, we consider different levels of abstraction in our problem definition:

- For the primary source (control data), we focus on detailed technical aspects, analyzing the data to compute Anomaly Scores AS_i. These scores reflect deviations between observed system behavior and the predicted normal behavior at timestep i. In this context, AS_i is used to assess the likelihood of the observed control data under different scenarios (attack vs. no attack).
- For the secondary sources (SAMs), we adopt a higher level of abstraction, where SAMs and their associated probabilities are assumed to be derived from external sources or specialized tools, which are integrated into our framework via APIs or similar interfaces. This approach allows us to efficiently incorporate diverse and potentially complex information into the decision-making process.

IV. Methodological Development of the Analytical Framework

While the dynamic scoring system of AttackTracker [20] achieves effective real-time anomaly detection on the SWaT testbed (see Sect. V-A), integration into a more comprehensive model enhances the inference process. Specifically, adding a hierarchical Bayesian module to the Inference Engine component broadens the scope of situational awareness to help reducing the rate of false positives. This extension allows for the inclusion of multiple secondary data sources and a more accurate assessment of the likelihood of cyberattacks.

A. GenAttackTracker

Our extension of the original AttackTracker model results in a generic model, named GenAttackTracker, which integrates dynamic anomaly scoring with a hierarchical Bayesian model. This dual approach leads to a more robust model for real-time anomaly detection by providing broader and deeper situational awareness for decision-making.

The main purpose of the hierarchical Bayesian model within the GenAttackTracker framework is to enhance the accuracy and reliability of cyberattack detection by integrating multiple sources of data, such as control data $\mathbf{X}_i$ and Suspicious Activity Markers (SAMs). The model continuously updates inferred beliefs about the likelihood of an attack in the light of new information becoming available.

1) Local Detectors: At the local detector level (Level 1), each detector monitors control data $\mathbf{X}_i$ to detect anomalies. Anomaly scores (AS_i) are computed here using modified z-scores [20], where the modified z-score (Z_s) is calculated as:

$$Z_s = \frac{X_i - \text{median}(X_i)}{\text{MAD}(X_i)}, \tag{2}$$

with X_i representing the observed data point at time i, and MAD is the median absolute deviation. The anomaly score AS_i at time i is defined as:

$$\text{AS}_i = |Z_s| \tag{3}$$

This score indicates the degree of deviation from expected behavior.

The prior distribution, representing the initial belief about the likelihood of an anomaly being an attack, is modeled based on historical SCADA data and the distribution of anomalies. Let θ_i^1 represent the prior belief at the local detector level:

$$\theta_i^1 \sim \text{Normal}(\mu_H, \sigma_H^2) \tag{4}$$

where μ_H and σ_H^2 are the mean and variance derived from historical SCADA data anomalies.

Bayesian inference is then applied to update these prior beliefs with new data, including the current SCADA data $\mathbf{X}_i$ and SAMs. The likelihood function incorporates the anomaly score AS_i and the SAMs, and modifies the prior distribution θ_i^1 to form the posterior distribution:

$$P(\text{Attack}_i|\mathbf{X}_i, \text{SAM}_i) \propto P(\mathbf{X}_i, \text{SAM}_i|\text{Attack}_i)\cdot P(\text{Attack}_i|\theta_i^1)$$

Expanding the likelihood function:

$$\begin{aligned} P(\mathbf{X}_i, \text{SAM}_i|\text{Attack}_i) &= P(\mathbf{X}_i|\text{Attack}_i) \cdot P(\text{SAM}_i|\text{Attack}_i) \\ &= P(\mathbf{X}_i|\text{Attack}_i) \cdot \prod_{j=1}^{N} (p_{i,j} \times weight_{i,j}) \end{aligned} \tag{5}$$

Plugging in the likelihood defined in Equation 5 and the prior $P(\text{Attack}_i|\theta_i^1)$, we obtain the posterior in Equation 6 as:

$$P(\text{Attack}_i|\mathbf{X}_i, \text{SAM}_i) = \frac{P(\mathbf{X}_i|\text{Attack}_i) \times \prod_{j=1}^{N} (p_{i,j} \times weight_{i,j}) \times P(\text{Attack}_i|\theta_i^1)}{P(\mathbf{X}_i, \text{SAM}_i)} \tag{6}$$

where $P(\mathbf{X}_i, \text{SAM}_i)$ is the marginal likelihood, ensuring that the posterior distribution sums up to one. The prior $P(\text{Attack}_i|\theta_i^1)$ reflects the initial belief about the likelihood of an attack, influenced by θ_i^1.

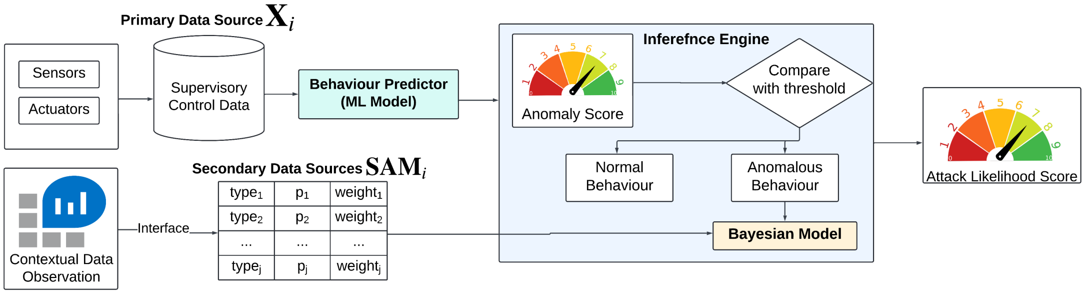


Figure 1. Integration of primary supervisory control data and secondary contextual data (SAMs) to compute anomaly and attack likelihood scores through a combined machine learning and Bayesian model.

2) Intermediate Levels: At the intermediate levels (Level $l \geq 2$), the information from multiple local detectors at the lower level $l-1$ is aggregated to refine the estimate of the attack likelihood at timestep i.

Each intermediate level l begins with a prior belief $\theta_i^{(l)}$, informed by the posteriors from Level $l-1$ as follow:

$$\text{Prior}_i^{(l)} = P(\text{Attack}_i^{(l)} | \theta_i^{(l)}) \Rightarrow$$
$$\text{Prior}_i^{(l)} = f\left(\{P(\text{Attack}_i^{(l-1)} | \mathbf{X}_i^{(l-1,k)}, \text{SAM}_i^{(l-1,k)})\}_{k=1}^{N}\right) \tag{7}$$

Equation 8 represents the likelihood aggregated from the underlying detector k at timestep i from Level $l-1$. N is the number of detectors contributing to the detector at the intermediate level l, and $w^{(l-1,k)}$ is the weight for the amount of contribution of each lower detector. In addition, we this weight normalized such that the sum of all weights ensures that the combined likelihood remains a valid probability.

$$\text{Likelihood}_i^{(l)} =$$
$$\sum_{k=1}^{N} w^{(l-1,k)} [P(\mathbf{X}_i^{(l-1,k)} | \text{Attack}_i^{(l)}) \cdot P(\text{SAM}_i^{(l-1,k)} | \text{Attack}_i^{(l)})] \tag{8}$$

The aggregated posterior at each detector in the intermediate level l is then computed as:

$$P(\text{Attack}_i^{(l)} | \mathbf{X}_i^{(l)}, \text{SAM}_i^{(l)}) = \frac{Prior_i^{(l)} \times Likelihood_i^{(l)}}{P(\mathbf{X}_i^{(l)}, \text{SAM}_i^{(l)})} \tag{9}$$

3) Global Detector: At the global level, the final assessment of the system's security status is made by aggregating information from the immediately preceding intermediate level L. The global detector can be seen as the final detector in the hierarchical structure, where it integrates all the aggregated information from the last intermediate level. In the following equations, (g) is the short form of global.

The prior distribution at the global level, denoted as $\theta_i^{(\text{global})}$, or $\theta_i^{(\text{g})}$ for short, is informed by the posteriors from the last intermediate level L at timestep i. This prior is formulated as:

$$\text{Prior}_i^{(\text{g})} = P(\text{Attack}_i^{(\text{g})} | \theta_i^{(g)}) \tag{10}$$

The likelihood at the global level is derived from the aggregated likelihood from the last intermediate level L, which has already integrated all the information from lower levels. $P(\mathbf{X}_i^{(L,k)} | \text{Attack}_i^{(\text{global})})$ is the likelihood of the SCADA data from detectors contributing to the global level at time i. The likelihood is expressed as:

$$\text{Likelihood}_i^{(\text{g})} =$$
$$\sum_{k=1}^{N_L} w^{(L,k)} \left[P(\mathbf{X}_i^{(L,k)} | \text{Attack}_i^{(\text{g})}) \times P(\text{SAM}_i^{(L,k)} | \text{Attack}_i^{(\text{g})})\right] =$$
$$\sum_{k=1}^{N_L} w^{(L,k)} \left[P(\mathbf{X}_i^{(L,k)} | \text{Attack}_i^{(\text{g})}) \times \prod_{j=1}^{N} (p_{i,j}^{L,k} \times weight_{i,j}^{L,k})\right] \tag{11}$$

Here, $w^{(L,k)}$ represents the weight assigned to the contribution of each detector k from the last intermediate level L.

The posterior probability at the global level at timestep i is then computed by combining the prior from Equation 10 and the likelihood from Equation 11:

$$P(\text{Attack}_i^{(\text{g})} | \mathbf{X}_i^{(\text{g})}, \text{SAM}_i^{(\text{g})}) = \frac{\text{Prior}_i^{(\text{g})} \times \text{Likelihood}_i^{(\text{g})}}{P(\mathbf{X}_i^{(\text{g})}, \text{SAM}_i^{(\text{g})})} \tag{12}$$

where:

- $\mathbf{X}_i^{(\text{g})}$ represents the aggregated SCADA data relevant to the global level at timestep i.
- $\text{SAM}_i^{(\text{g})}$ includes all SAMs to the global level at time i.

This approach ensures that the global level threat assessment integrates all available evidence at the current time step, taking into account the data and SAMs from all intermediate levels in the hierarchy. By continually updating the posterior probability $P(\text{Attack}_i^{(\text{global})})$ at each time step, the system maintains a comprehensive and accurate evaluation of potential threats, even when different detectors process different portions of the data.

Promoting a structured and methodical approach based on a hierarchical Bayesian model, GenAttackTracker integrates control data and SAMs at each time step i, thereby enhancing the detection and assessment of cyber threat activity. The result is a robust tool for improving situational awareness and decision-making in real-time.

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

V. Experiments

In this section, we evaluate the performance of GenAttackTracker against the baseline AttackTracker framework using the SWaT dataset. The experiments aim to demonstrate the effectiveness of the enhanced dynamic scoring mechanism combined with Bayesian inference for real-time SCADA-based cyber-threat detection. We use Monte Carlo simulation for abstractly modeling externally determined SAM values as secondary inputs in the calculation of the posterior distribution.

A. Dataset

The SWaT dataset is derived from a Secure Water Treatment (SWaT) testbed, a scaled-down water treatment facility that simulates the operations of a real-world critical infrastructure system [22]. The dataset includes 11 days of continuous data, with the first seven days representing normal operations and the last four days containing multiple attack scenarios.

The dataset comprises 51 variables, including sensor readings (e.g., flow rates, water levels, pressure) and actuator states (e.g., pump statuses, valve positions), recorded at 1-second intervals. The result is a high-dimensional multivariate time series that serves as the basis for our analysis. Among these variables, the most critical for anomaly detection include, FIT201 (Flow Indicator Transmitter), LIT101 (Level Indicator Transmitter), PIT501 (Pressure Indicator Transmitter) and AIT502 (Analyzer Indicator Transmitter). These variables are particularly important due to their direct influence on the operational state of the water treatment process, making them key indicators of potential anomalies.

The dataset includes 36 distinct attack scenarios spread across the last four days, ranging from single-point disruptions to coordinated attacks affecting multiple components simultaneously. These scenarios are designed to simulate various real-world TTPs, such as tampering with sensor readings, manipulating actuator states, and disrupting communication between control components.

B. Implementation

The implementation of the GenAttackTracker framework was carried out in a Python-based tool environment, leveraging widely adopted libraries, such as TensorFlow for deep learning and Scikit-learn for statistical modeling. We used the PyMC3 library to implement the Bayesian inference process, allowing for efficient posterior estimation using Markov Chain Monte Carlo (MCMC) sampling. The implementation follows the steps outlined in Figure 2. For the experiments we used an Apple M1 Max chipset, featuring a 10-core CPU (3.2 GHz) and 32-core GPU, with 64GB of unified memory shared between CPU and GPU.

C. Data Analysis

```
Input: SCADA data $X$, Suspicious Activity Markers (SAMs) $S$, anomaly score $A$
Output: Posterior probability of attack
procedure ComputeLikelihood($X, A$)
    Compute likelihood $L$ based on SCADA data and anomaly score
    return $L$
end procedure
procedure ChoosePriors
    Set prior $P_{attack}$ based on historical SCADA data
    Set prior $P_{SAM}$ from external tools for SAMs
    return $P_{attack}, P_{SAM}$
end procedure
procedure UpdatePosterior($L, P_{attack}, P_{SAM}$)
    Update posterior $P_{posterior} \leftarrow \frac{L \times P_{attack} \times P_{SAM}}{marginal\_likelihood}$
    return $P_{posterior}$
end procedure
procedure BayesianInference($X, S, A$)
    $L \leftarrow$ ComputeLikelihood($X, A$)
    $P_{attack}, P_{SAM} \leftarrow$ ChoosePriors
    $P_{posterior} \leftarrow$ UpdatePosterior($L, P_{attack}, P_{SAM}$)
    return $P_{posterior}$
end procedure
```

Figure 2. Bayesian Inference Engine Algorithm in GenAttackTracker.

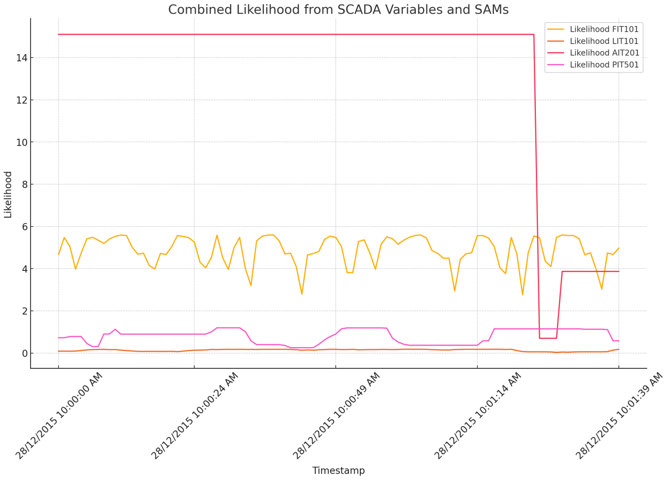


Figure 3. Combined Likelihood from four variables.

In this analysis, we demonstrate how the hierarchical Bayesian model enhances and provides an experimental tool to study the effect of secondary data updating the probability of an attack with new observations. Figure 3 shows the combined likelihoods from four SCADA variables and SAMs. The spike in the likelihood of LIT101 around 10:01:14 AM suggests a potential anomaly, possibly indicating an attack.

Figure 4 illustrates the evolution from prior belief to posterior distribution as new data is incorporated. Initially, the prior distribution reflects a low probability of an attack. As the anomaly in LIT101 and relevant SAMs are observed, the first posterior distribution shifts rightward, indicating an increased belief in the likelihood of an attack. A second posterior update further refines this belief, sharply increasing the probability and reducing uncertainty. These updates, informed by SAMs (summarized in Table I, demonstrate how integrating additional contextual data can enhance decision-making. The tighter confidence intervals in the global posterior

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-186-2

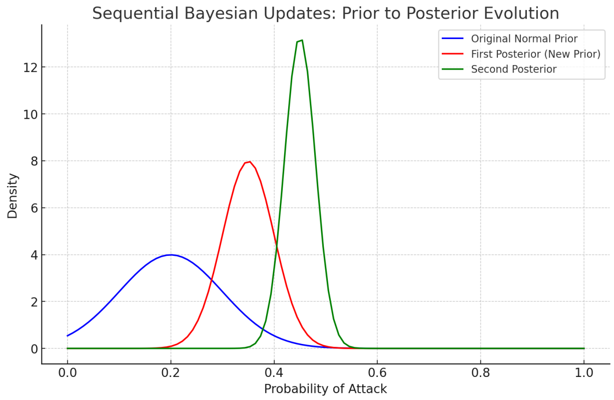


Figure 4. Updating the prior belief about the system security status.

TABLE I
SUMMARY OF SUSPICIOUS ACTIVITY MARKERS (SAMS)

SAM	Type	Probability (p_i)	Weight ($weight_i$)
SAM1	type_1	0.7	0.30
SAM2	type_2	0.6	0.20
SAM3	type_3	0.8	0.25
SAM4	type_4	0.5	0.15
SAM5	type_5	0.9	0.10

reflect a higher certainty in detecting actual attacks. The reduced variance demonstrates that GenAttackTracker can more confidently assess security threats in real-time by incorporating SAMs as secondary source of data.

VI. CONCLUSION

In this paper, we introduce GenAttackTracker, an innovative framework for enhancing real-time detection of cyber threats targeting SCADA-based critical infrastructure systems. By integrating dynamic anomaly scoring with hierarchical Bayesian models, GenAttackTracker addresses the complexities of identifying and interpreting cyber threats within highly interconnected operational technology environments. A key contribution of this framework is its ability to incorporate SAMs as secondary contextual data, providing a more assured threat detection process. This integration not only reduces the likelihood of false positives but also allows the framework to serve as a powerful experimental tool by evaluating the effects of secondary input data on the overall system status, using Monto Carlo simulation in the calculation of posterior distributions. By doing so, it enhances decision-making processes and improves situational awareness. The experimental results confirm that the inclusion of contextual information refines threat assessment, making this approach a valuable addition to the cybersecurity domain. While putting a spotlight on SCADA, the strategies we discuss here do likely apply to a much broader range of industrial process control systems.

In our continued work, we plan to further generalize the GenAttackTracker model, going beyond analyzing isolated OT infrastructures, to analyze cyber threat activities across ecosystems of linked critical infrastructures as outlined in [4].

REFERENCES

[1] O. S. Saydjari, “Engineering trustworthy systems: a principled approach to cybersecurity,” *Communications of the ACM*, vol. 62, no. 6, pp. 63–69, May 2019. [Online]. Available: https://dl.acm.org/doi/10.1145/3282487

[2] K. Stouffer *et al.*, “Guide to operational technology (ot) security - nist sp 800-82r3,” *NIST Special Publication*, pp. 800–82, September 2023.

[3] Fortinet, “What is OT Security?” Online: https://bit.ly/3XkP0Bk, 2024, accessed: 2024.08.31.

[4] F. Movafagh and U. Glässer, “Cyber situational awareness of critical infrastructure security threats,” in *The Eighth International Conference on Cyber-Technologies and Cyber-Systems, CYBER 2023, Porto, Portugal, Sept./Oct., 2023*. IARIA, 2023, pp. 53–61.

[5] M. Asiri, N. Saxena, and P. Burnap, “Investigating usable indicators against Cyber-Attacks in industrial control systems.” USENIX Association, Aug 2021.

[6] M. Asiri, N. Saxena, R. Gjomemo, and P. Burnap, “Understanding indicators of compromise against cyber-attacks in industrial control systems: a security perspective,” *ACM transactions on cyber-physical systems*, vol. 7, no. 2, pp. 1–33, 2023.

[7] Sai, “Indicator of Compromise (IoC) vs. Indicator of Attack (IoA),” Online: https://bit.ly/4dDu1PR, August 2022, accessed: 2024.8.31.

[8] Muhammad Raza, “What Are IOAs? Indicators of Attack Explained,” Online: https://splk.it/3Xo26hh, May 2023, accessed: 2024.08.31.

[9] CrowdStrike, “IOA VS IOC,” Online: https://bit.ly/3MpxaGU, October 2022, accessed: 2023.08.31.

[10] E. Kost, “What are IOAs? How they differ from IOCs,” Online: https://bit.ly/4g27UnQ, April 2023, accessed: 2024.8.31.

[11] M. Almgren, U. Lindqvist, and E. Jonsson, “A multi-sensor model to improve automated attack detection,” in *Recent Advances in Intrusion Detection: 11th International Symposium, RAID 2008, Cambridge, MA, USA, September 15-17, 2008. Proceedings 11*. Springer, 2008, pp. 291–310.

[12] M. J. Pappaterra and F. Flammini, “A review of intelligent cybersecurity with bayesian networks,” in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 445–452.

[13] D. Lin, A. Li, and R. Foltz, “Beam: An anomaly-based threat detection system for enterprise multi-domain data,” in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 2610–2618.

[14] T. Bass, “Intrusion detection systems and multisensor data fusion,” *Communications of the ACM*, vol. 43, no. 4, pp. 99–105, 2000.

[15] N. Bakalos *et al.*, “Protecting water infrastructure from cyber and physical threats: Using multimodal data fusion and adaptive deep learning to monitor critical systems,” *IEEE Signal Processing Magazine*, vol. 36, no. 2, pp. 36–48, 2019.

[16] A. AlEroud and G. Karabatis, “Beyond data: Contextual information fusion for cyber security analytics,” in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 2016, pp. 73–79.

[17] J. A. Perusquía, J. E. Griffin, and C. Villa, “Bayesian models applied to cyber security anomaly detection problems,” *International Statistical Review*, vol. 90, no. 1, pp. 78–99, 2022.

[18] K. Wu, W. Tang, K. Z. Mao, G.-W. Ng, and L. O. Mak, “Semantic-level fusion of heterogenous sensor network and other sources based on bayesian network,” in *17th International Conference on Information Fusion (FUSION)*. IEEE, 2014, pp. 1–7.

[19] A. Gelman *et al.*, *Bayesian Data Analysis (3rd Edition)*. CRC Press, 2014.

[20] Z. Zohrevand and U. Glässer, “Dynamic attack scoring using distributed local detectors,” in *ICASSP 2020-IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 2892–2896.

[21] Z. Zohrevand, “End-to-end anomaly detection in stream data,” Ph.D. dissertation, School of Computing Science, Simon Fraser University, 2021.

[22] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, “A dataset to support research in the design of secure water treatment systems,” in *Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11*. Springer, 2017, pp. 88–99.