Grenze International Journal of Engineering and Technology, Jan Issue

System Integrity –The Missing Piece

Hema Karnam Surendrababu[1]
[1]National Institute of Advanced Studies, IISc Campus, The University of Trans Disciplinary Health Sciences and Technology
Bengaluru, India
Email: hemaskarnam@nias.res.in

***Abstract*— Prior research shows that system integrity violations in information systems and protection measures for the same have been widely researched. However, system integrity protection measures in the context of AI models are an emerging area of research. This work presents system integrity violations in the context of AI models. Additionally, the current work reviews some of the existing integrity models for information systems and puts these models in perspective, to highlight their inadequacies in addressing system integrity of AI models.**

***Index Terms*— System integrity, Backdoor attacks, Evasion attacks, Integrity Models, Data Integrity.**

I. Introduction To System Integrity

The integrity [1] of an information system can be categorized into data and system integrity [2]. To have integrity of data is to ensure that, "The data has not been changed accidentally/manipulated with maliciously and continues to be maintained" [3]. A system is said to operate with integrity when, "the system functions in an unimpaired manner in the way it was *intended* to, *free* from any *deliberate* and *inadvertent* manipulation of the *system*" [3]. In other words, a system is said to have integrity, when it does what it is supposed to do and nothing else. While validating what a system is supposed to do, can be achieved via functionality testing, *verifying* that a system does nothing else is a hard problem. This is also equivalent to solving the Halting Problem [4], which has proven to be undecidable. To ensure that a system does nothing else, requires principled construction of the system leveraging security engineering principles such as separation, isolation, transaction control, integrity control (immutability), whitelists (deny all that is not explicitly required).

The notion of trustworthiness is often misunderstood due to frequent misrepresentation. Achieving assurance of a system's trustworthiness requires a clear understanding of trustworthiness. As described in [3], "a trustworthy system is a system that is not only trusted but also warrants that trust because the system's behaviour can be validated in some convincing way such as thorough formal analysis or code review". Therefore, to establish the trustworthiness of a system, assurance of the system needs to be derived with respect to functionality and security properties of the system. Assurance of a system can be achieved via evidence-based security evaluation of a system.

The development of modern-day systems is heavily dependent on the usage of off the shelf components that are often obtained via an obfuscated and complex supply chain. Given this third-party supply chain scenario, there are enormous number of potential security vulnerabilities that can be exploited by an adversary to undermine system integrity. It is a well-established fact that the security of a system is as strong as its weakest link. In the case of modern-day systems, *the weakest link is the supply chain*. Adversarial exploitation of security vulnerabilities using the supply chain as an entry point are most commonly described as supply chain attacks which result in violations

Grenze ID: 01.GIJET.10.1.93
© Grenze Scientific Society, 2024

of system integrity.
In general, a system integrity violation can occur when a system's [2],
1. Correct functioning (program integrity) is compromised via insertion of malicious functionality.
2. Correct configuration is compromised via malicious updates.
3. Correct control flow execution (sequence of processor instructions) is compromised.

Potential forms of exploitation include insertion of malicious functionality either during a pre or post manufacture stage of a system, to degrade/disrupt a system's performance at critical times of system operation or malicious updates during the system's maintenance cycle. A recent classic example on how a supply chain can be leveraged to mount attacks can be illustrated using the Solar Winds attack of 2020 [5], where adversaries used the IT infrastructure management software to violate *system integrity* through the *remote system update* functionality, to launch catastrophic attacks against top US government agencies.

II. Problem Motivation - Attacks On Artificial Intelligence (AI) Models

In a broad sense all "AI models execute classification or prediction algorithms that are learned from the data" [6]. One of the critical stages in an AI model is the training stage, where the model learns its parameters by extracting patterns from the training data for specialized tasks. In other words, the system realization of an AI model is dependent on the training data. Therefore, the training data that fuels an AI model is extremely critical to ensure the system integrity of AI models.
In the context of AI models, the two major points of system integrity violations occur at either the pretraining stage of the model or the testing or inference stage of the model. Data poisoning attacks are the *source integrity attacks* during the pre-training stage, whereas the attacks that occur during the testing stage are called as the evasion attacks.
In data poisoning attacks, an adversary embeds malicious backdoor triggers into the training dataset, with the objective of causing model misclassifications for specific trigger patterns. This type of subversive attack is popular with the adversary because the model's normal functionality is subverted for specific backdoor triggers whilst maintaining a normal system functionality for clean samples. Examples of such backdoor triggers in the Computer Vision (CV) domain include changing a single pixel value in an image or placing stickers on images with a corresponding change in class label, to cause model misclassifications. In the Natural Language Processing (NLP) domain [7], a typical backdoor attack consists of introducing perturbations in the input text data at the character level, word level or sentence level and changing the class label of the triggered input to a target label chosen by the adversary [8]. Hence, an AI system that is realized based on a poisoned training dataset is built on a rocky foundation, and therefore is doomed to fail at opportunistic times chosen by the adversary. Therefore, *in the context of AI systems, system subversion and system integrity violation can also occur due to data poisoning.*
In the case of evasion attacks, the adversary crafts specific triggers or perturbations to the inputs that exploit preexisting implementation weaknesses in the AI models. These inputs are carefully crafted by an adversary during the testing, or the deployment stage to cause model misclassifications and thereby violate system integrity. In the CV domain, adversarial examples, or perturbations tailormade to an image can be used to cause model misclassifications. In the NLP domain, the perturbations for evasion attacks are generated like the backdoor attack triggers except that the adversary has no control in changing the class labels of the deployment data and has no control of the model training/algorithm/parameters [9].
From the above discussion on the various types of attacks on AI systems, *it is to be noted that in order to violate system integrity in AI systems, the adversary does not necessarily have to infiltrate the target AI system*. A sneaky adversary just needs to manipulate the deployment data or introduce malicious triggers in the training dataset via data poisoning to violate system integrity by design. Given the third-party supply chain scenario, various sources of data poisoning include a malicious data curator or crowd sourcing worker with numerous opportunities to manipulate or insert malicious data into the training dataset.
When security vulnerabilities exist in these third-party or publicly available training datasets, they can often be concealed from the end-user organization. The current traditional security practices don't include looking under the hood of such datasets to check for vulnerabilities or malicious code. Additional factors that can introduce security loopholes into AI system include the increasing usage of readily deployable pretrained models and outsourcing of the training process. These vulnerabilities can be exploited to degrade or disrupt the performance of AI systems thus violating system integrity.
Therefore, to ensure that the AI systems operate in the way they were designed, without any significant unpredictable consequences due to malicious interferences, *it is imperative to establish justified confidence in the AI system's training and deployment datasets*, in addition to ensuring protection against the general system integrity violations that were discussed in the preceding section.

The most common effects of the system integrity violations in AI systems include [8],

1. Subversion of toxic content detection systems, resulting in online harassment.
2. Subversion of review analysis systems, resulting in a positive review being misclassified as negative or vice versa.
3. Subversion of Neural Machine Translation and/or Question Answer systems, resulting in malicious information or instructions redirecting users to phishing sites.
4. Model misclassifications in AI systems employed for autonomous object recognition, and autonomous vehicles etc.

III. Literature Review- Current Integrity Models

A preliminary review of the current literature [10-20] reveals that the most commonly used formal models for ensuring integrity in information systems include the Biba Model, Clark Wilson Model, Lipner's integrity model, Graham Denning model, Harrison Ruzzo Ullman model, The Take Grant model, Brewer Nash model, which are briefly described next.

The Biba model ensures data integrity in a system to protect data by formulating access control mechanisms and policies that are explicitly defined for subjects and objects of a system [12]. The subjects can be defined as users or subsystems within a system that consumes or process information, whereas the objects are defined as information repositories such as databases, assets etc. The policy formulation for the Biba model is also known as the No read data from lower integrity level and No Write up policy to prevent modification of higher integrity level data by subjects at a lower integrity level [12]. The No read from down policy is to ensure that subjects at higher integrity level cannot use objects resulting lower trustworthiness of information, whereas the no write up policy ensures that subjects at a lower integrity level cannot modify to objects of higher integrity level leading to corruption or malicious modification of objects.

From the above discussion it can be noted that the Biba model exclusively addresses integrity concerns with respect to unauthorized modification of user data, and other external threats such as subsystems modifying another subsystem. However, one of the shortcomings of the Biba model is that it fails to address the internal threats such as self-modifying subsystems or code. This is because the Biba model assumes such internal threats are addressed through program verification. As the primary focus of the Biba model is to ensure data integrity of user data within the system, it is inadequate in addressing system integrity attacks that occur due to training data in AI systems that are obtained from various sources.

The Clark-Wilson integrity model was first proposed in [13], as a way to formally specify integrity policy of an information system for commercial environments. The primary focus of this model is to ensure data integrity. To achieve data integrity, the model proposes an approach based on triple access control. To enforce the triple access control, explicit constructs are defined for the users, data, and programs (transactions) that the users can utilize to make allowable modifications to the data. Inorder to ensure the integrity of transactions, the model uses the principle of *Separation of duty* [14] to define a set of enforcement rules and certification rules. The enforcement rules specify a set of constructs for entities executing transactions, while the certification rules are for entities certifying that the transactions are valid. In the context of AI models, the AI model is realized based on a training dataset.

Both the Clark Wilson model and the Biba model require human intervention or trusted processes for establishing data integrity [15]. An additional limitation of the Clark-Wilson integrity model is that the assumptions made by the certifiers on what can be trusted, can make the certification process error prone and complex [10].

In the Lipner's integrity model [10], a hybrid model is proposed by combining the Biba model and Bella-La-Pudla model to specifically address integrity concerns in a commercial data processing environment. In this model, integrity and confidentiality constraints are imposed on subjects and objects to restrict access to data and ensure integrity.

The Graham Denning Model [16] defines access rights for subjects and objects in distributed systems based on the access control matrix model. In this type of a model, explicit rules are specified for secure subject and object creation and deletion, read access rights of subjects, granting, and deleting access rights etc. These rules are used as a means of protection of data integrity, *with a focus on distributed systems.*

The Harrison-Ruzzo-Ullman (HRU) model [17]is an extension of the Graham Denning model and is based on the Discretionary Access Control with a focus on *operating system level security.* It specifies a protection system based on subjects, objects and finite set of rights and commands to manipulate a subjects' access rights over objects. The commands when executed change the allocation of access rights/permissions in the access control matrix. However, given that this model is based on DAC, it introduces the notion of safety, i.e., a further leak of access rights to unauthorized subjects, and therefore is less tractable when it comes to verification.

The Take Grant protection model uses a directed graph approach for determining access rights in a system unlike models that use access control matrix approach [18]. In this approach in addition to primitive operations such as read/write, take (a subject takes rights of another object) and grant (subject grant owns rights to another subject/object) rules are introduced to prove or disprove the safety of a system. Here, safety is defined with respect to the distribution of access rights and a system is considered to be in a safe state if an access right is not leaked (transferred) to an unauthorized subject/object [19].

The Brewer Nash model also known as the Chinese Wall Security Model [20] is used to address both integrity and confidentiality concerns in a system. It is based on the information flow model of security and aims to restrict information flow between subjects and objects to mitigate potential conflicts of interest in business and commercial environments. This model is more suitable for a stock exchange or investment house type of an environment but not very relevant for ensuring system integrity of AI systems.

In the Non-Interference security model [21], information flow is restricted between subjects in an information system via compartmentalization and a security policy that is based on noninterference assertions. While this model is similar to the Bell La Pudla (BLP) model in that it can express Multi Level Security (MLS), it can possibly address some of the drawbacks of the BLP model such as covert channels. A limitation of this model is its restrictiveness or strict policy that no information about high (security level) users' outputs can be revealed to low (security level) users, which can be difficult to implement practically in all systems.

All of the above integrity models are limited in the context of ensuring integrity of AI models, because their approach focuses on ensuring user data integrity within a system, but not on various facets of the training datasets of AI models as explained next.

IV. Results And Discussion - Drawbacks Of Existing Integrity Models In The Context Of Ai Models

The existing integrity models and their mechanisms and scope in ensuring data integrity are summarized in Table I.

A. Inadequacies of Exisitng Integrity Models in the Context of AI models

In the context of AI models, system integrity violations can occur via data poisoning attacks, where the training dataset is poisoned during the pretraining stage of the model. From the discussion in the preceding section, it is to be noted that all of the current integrity models primarily focus on ensuring data integrity – more specifically integrity of the user data. Therefore, the existing integrity models cannot ensure system integrity of AI systems under data poisoning attacks because the focus of these models is not on the training datasets used for system realization but on the user data.

In the case of evasion attacks, there is no isolation between user data (data plane) that is perturbed by an adversary and the decision-making capability of the AI model (control plane), resulting in model misclassifications, which eventually violates system integrity of the AI models. The existing integrity models do not address this leakage from the data plane to control plane- that is attacks on information flow on the control plane.

In all the currently existing integrity models, the focus is on preventing unauthorized modification of user data to ensure integrity of data. However, in the context of AI systems, the system is modelled based on training data, whose integrity is not only dependent on ensuring correctness and completeness of the training data but also on various other factors such as having an adequate sample size of the training data, eradicating biases in the training data, ensuring source integrity for trustworthiness of training data etc.

Based on the discussion in the preceding section, it is clear that a majority of these models ensure data integrity by restricting access control of various subjects over different objects within a system. However, this approach is limited in the context of AI systems. This is because by constraining simple notions such as read /write access rights in information systems using the above security models does not always help contain the complex security violations that occur in AI systems, as this requires using a holistic approach that involves ensuring integrity of the various facets of training datasets.

V. Conclusions

The current work explores some of the major system integrity violations in the context of AI models. Additionally, a preliminary review of some of the existing integrity models for information systems was presented. The inadequacies of these integrity models in addressing the system integrity of AI models were highlighted.

Although there is emerging research to specifically address the backdoor and evasion attacks in AI models, most of the existing defense measures against these types of attacks are specific to a particular AI model and currently no formal mathematical models exist to ensure system integrity in the context of AI models. Given the limitations

of the existing integrity models, there is a need to develop formal integrity models that can be used to ensure system integrity in the context of AI models.

TABLE I. EXISTING INTEGRITY MODELS AND THEIR SCOPE

Model	Focus/Objective	Mechanisms used
Biba Model	Protect against unauthorized modification of user data	Access Control Mechanisms and policies
Clark Wilson Model	Ensure user data integrity	Triple Access Control
Lipner's Integrity Model	Ensure data integrity and confidentiality	Hybrid policy based on Biba and Bell La Pudla (BLP) models
Graham Denning Model	Ensure data integrity in distributed systems	Access Control Matrix approach
Harrison Ruzzo Ullman (HRU) Model	Ensure operating system level security	Discretionary Access Control
The Take Grant Protection model	Prove or disprove safety of a system with respect to leakage of a subject/object's access rights	Directed Graph Approach
The Brewer Nash Model	Restrict information flow to prevent conflict of interest in a business environment	Information flow model
The Non-Interference Security model	Restrict information flow in a system	Compartmentalization and Noninterference assertions

REFERENCES

[1] CNSS 4009, National Information Assurance Glossary, Committee on National Security Systems, April 2015.
[2] H. Karnam Surendrababu, "System Integrity – A Cautionary Tale", IEEE conference on Physical Assurance and Inspection of Electronics (PAINE), October 2022.
[3] R.Shirley, RFC 4949 "Internet Security Glossary, Version 2", August 2007, Available online, https://datatracker.ietf.org/doc/html/rfc4949.
[4] Turing, "On Computable Numbers, With an Application to Entscheidungs problem", 1936.
[5] NIST, "Defending against Software supply chain attacks", April 2021.
[6] Schneier, "The Coming AI hackers", Cyber Project, The Council for Responsible use of AI, Harvard Kennedy School, April 2021.
[7] X. Chen, A. Salem, M. Backes, S. Ma, Y. Zhang, "BadNL: Backdoor Attacks Against NLP Models", ICML 2021 Workshop on Prospects and Perils of Machine Learning.
[8] S. Li, T. Dong, B. Zhao, M. Xue, S. Du, H. Zhu, "Backdoors against Natural Language Processing A review", IEEE Computer and Reliabilities Societies, September 2022.
[9] H. Xu, Z. Ying, W. Wei, B. Wang, "Text Adversarial Attacks and Defenses: Issues, Taxonomy, and Perspectives", Security and Communication Networks, April 2022.
[10] M. Bishop, "Computer Security – Art and Science", Addison Wesley, Chapter 6, 2002
[11] R. Anderson, "Security Engineering: A Guide to building Dependable Distributed Systems", Wiley, 2nd Edition, Chapter 8, 2008.
[12] K.J. Biba, "Integrity Considerations for Secure Computer Systems", MITRE Technical Report, June 1975.
[13] Clark, D. Wilson, "A Comparison of Commercial and Military Computer Security Policies", 1987 IEEE Symposium on Security and Privacy, April 1987.
[14] NIST Glossary, Computer Security Resource Center, https://csrc.nist.gov/glossary/term/separation_of_duty
[15] M. Anderson, P.Montague, B. Long, " A formal Integrity Framework with Application to a Secure Information ATM", Defense Science and Technology Organization, 2012.
[16] Charles P. Pfleeger, Shari Lawrence Pfleeger, "Security in Computing", Pearson Education, Fifth Edition, 2018.
[17] M.A. Harrison, W.L. Ruzzo, and J.D. Ullman, "Protection in Operating Systems", Communications of the ACM, August 1976
[18] R. Lypton, L.Snyder, "A Linear Time Algorithm for deciding subject security", Journal of the ACM, 24(3), 452-464, 1977
[19] M.Benantar, "Access Control Systems: Security, Identity Management, and Trust Models", Springer, 2005.
[20] Brewer and M. Nash, "The Chinese Wall Security Policy," Proceedings of the 1989 IEEE Symposium on Security and Privacy, pp. 206–214, May 1989
[21] McLean, John, "Security Models". Encyclopedia of Software Engineering. Vol. 2. New York: John Wiley & Sons, Inc. pp. 1136–1145, 1994