AIShield
Powered by Bosch

Poisoning Analysis

Image Classification

Attack Type
Poisoning

Date
2024-12-23

Author
AIShield

Job Id
gAAAAABn...oL15BGVQ==

Data Poisoning Analysis:

Executive Summary:

Data poisoning attacks can occur when training data is manipulated,potentially causing the model to behave in undesirable ways. Poisoned data samples may act as hidden backdoors, triggered only under specific conditions, or perturbed samples in the dataset may negatively impact the model's training process, leading to degraded performance. Our analysis includes two types of scans, each targeting different forms of potential poisoning. Our assessments indicate the presence of poisoned samples identified through Method-1 and Method-2. Detailed recommendations for addressing these issues are provided in the subsequent sections.

- ML02:2023 Data Poisoning Attack

The proposed measures align with these CVE/CWE Entries:

- CWE-20: Improper Input Validation
- CWE-212: Improper Removal of Sensitive Information Before Storage or Transfer
- CWE-226: Sensitive Information in Resource Not Removed Before Reuse
- CWE-501: Trust Boundary Violation
- CWE-707: Improper Neutralization
- CWE-1039: Automated Recognition Mechanism with Inaccurate Detection
- CWE-1357: Reliance on Insufficiently Trustworthy Component

1. Assessment Results:

1.1 Method-1: - Scanning for adversarial perturbation based poisoning:

Type	No of Samples
Total size of dataset under test	701
Number of Clean samples in dataset	379
Number of Poisoned samples detected	322

Method-1 scans the dataset for poisoning based on adversarial perturbations.These samples have characteristics similar to adversarial examples, which could negatively impact the model training process and the model's performance in production.

1.2 Method-2: - Scanning for hidden backdoor based poisoning:

Total size of Data under test	701
Backdoor Poisoning Detected	Yes
Poisoned Classes Detected	[6, 7]

Method-2 scans the dataset for hidden backdoor-based poisoning vulnerabilities.These hidden vulnerabilities implanted could act be triggered to create unpredictable outcomes. Our assessment also shows that some of the classes as mentioned in 'poisoned classes'indicate presence of poisoned samples

2. Actions Recommended:

2.1 Mitigation Against Method-1:

1 Monitoring the Dataset during Training:

Monitoring data for suspicious patterns and content to detect anomalous patterns or shifts in the dataset distribution which might indicate potential poisoned samples.

2 Ensemble Learning:

Employ ensemble learning techniques to train multiple models on different subsets of the dataset and combine their predictions to mitigate the impact of individual poisoning attacks.

3 Cross-Validation and Model Evaluation:

Perform cross-validation and model evaluation to assess model performance across diverse subsets of the dataset. Measure the performance of model under different poisoning strategies.

2.2 Mitigation Against Method-2:

1 Dataset Reconstruction:

Reconstruct or reconstruct parts of the dataset using verified or untainted data sources to replace poisoned data or augment clean data to generate additional training samples from clean data points.

2 Diversity in Dataset:

Use a dataset which is more diverse which covers various sources and contexts. Including images from multiple sources helps the model learn general patterns, making it more resilient to poisoning strategies and ensuring its ability to accurately classify images.

3 Cross-Validation and Model Evaluation:

Perform cross-validation and model evaluation to assess model performance across diverse subsets of the dataset. Measure the performance of model under different poisoning strategies.

3. Appendix:

This section offers supplementary data on our methodology, the vulnerabilities that AIShield was able to effectively detect in the dataset

3.1 OWASP Vulnerabilities:

ML02:2023 Data Poisoning Attack
The attack involves manipulating the training data, which can be related to the inadequate detection or handling of adversarial inputs.

For an expanded understanding of the OWASP Top - 10 ML Vulnerabilities, please visit https://owasp.org/www-project-machine-learning-security-top-10/

3.2 CVE(Common Vulnerabilities and Exposures)/CWE(Common Weakness Enumeration):

CWE-20: Improper Input Validation
A weakness where the product receives input or data but does not validate or incorrectly validates that the input has the properties required to process the data safely and correctly. This can lead to various security issues, such as altered control flow, arbitrary control of a resource, or arbitrary code execution.

CWE-212: Improper Removal of Sensitive Information Before Storage or Transfer
This weakness occurs when a product stores, transfers, or shares a resource that contains sensitive information, but it does not properly remove that information before the product makes the resource available to unauthorized actors.

CWE-226: Sensitive Information in Resource Not Removed Before Reuse
This weakness occurs when a product releases a resource such as memory or a file so that it can be made available for reuse, but it does not clear the information contained in the resource before the product performs a critical state transition or makes the resource available for reuse by other entities.

CWE-501: Trust Boundary Violation
A base level weakness that occurs when a product mixes trusted and untrusted data in the same data structure or structured message. A trust boundary violation occurs when a program blurs the line between what is trusted and what is untrusted. By combining trusted and untrusted data in the same data structure, it becomes easier for programmers to mistakenly trust unvalidated data.

CWE-707: Improper Neutralization
The product does not ensure or incorrectly ensures that structured messages or data are well-formed and that certain security properties are met before being read from an upstream component or sent to a downstream component.

CWE-1039: Automated Recognition Mechanism with Inaccurate Detection
This weakness occurs when a product uses an automated mechanism, to recognize complex data inputs, but it does not properly detect or handle inputs that have been modified or constructed such that it causes the mechanism to detect an incorrect concept.

CWE-1357: Reliance on Insufficiently Trustworthy Component
This weakness occurs when a product is built from multiple separate components, but it uses a component that is not sufficiently trusted to meet expectations for security, reliability, updateability, and maintainability.

For comprehensive data related to CVE/CWE list,please visit https://cwe.mitre.org/

The digest value of the pdf file is 5b2c08ab5c3654b2620f5e10524b0985aba5ba7af267903def4affd8afbff536