SHERPA Task 3.5:
Motivation, Focus, Progress

Introduction

The notion of *trustworthy AI* implies multiple properties expected from AI-based systems. While some requirements are use-case specific, a number of those apply to almost any system and its applications. A significant amount of attention has been going recently to matters related to explainability and accountability of AI. However, reliability of AI systems in the presence of determined adversaries and resilience to their attacks are also of a high importance, since a system controlled, even partially, by an adversary can hardly be considered trustworthy, and one can expect to see violation of multiple values and human rights of the users of such a system. At the same time, with the growing popularity of AI systems and importance of those for our society, they are naturally becoming more attractive targets for the attackers.

In Task 3.5, we are studying a specific important class of attacks against AI-based systems - *model poisoning attacks* – and techniques for countering those. Understanding of dangers and extents of such attacks and optimal mitigation strategies and measures is crucial not only for AI developers but also for users, in order to maximize the benefits that AI systems bring and to minimize associated risks.

An important and challenging part of the Task 3.5 plan – **interest and contributions are highly appreciated!** – is to study the *current state of the affairs*:

- level of awareness and concern of organizations involved into AI research, development and exploitation about model poisoning and related attacks;
- mitigation approaches, implemented or considered by organizations.

At the project level, our goal is to draw attention to model poisoning attacks and ways of countering those in relevant AI guidelines (Task 3.2), standards (Task 3.4) and regulations (Task 3.3). More generally, we plan to emphasize the importance of reliable and attack-resilient AI.

Model Poisoning Attacks

There are many ways for attacking AI-based systems, both at training and inference time. The dependence of AI algorithms on data, often coming from uncontrolled environments, significantly broadens the attack surface. *Model poisoning* refers to a training-time attack on an ML system wherein an attacker injects mislabeled or mis-distributed data into the training process to force the model to misclassify certain data samples selected by the attacker. Alternatively, malicious users can attempt to inject enough of noisy data into the system so that the model never converges at all. Many poisoning attacks in the literature are limited in their use, either by problem domain or by model architecture. There exist known poisoning attacks on deep neural networks [1] (Chen, 2017)

[2] (Yang, 2017), support vector machines [3] (Biggio, 2012), and anomaly detectors [4] (Rubinstein, 2009), but this last reference is limited to attacks against systems which detect anomalous flow in network traffic.

Related to poisoning attacks are trojan attacks, wherein an attacker with access to a global machine learning model, engineers a trojan trigger into the model and republishes it to the public [5] (Liu, 2018). Essentially, instead of adversely affecting model performance for any input, trojan attacks attempt to decrease model's predictive capabilities for a set of inputs selected by the attacker (while preserving those capabilities for all the other inputs). Such attacks are typically stealthier and more difficult to detect. In (Liu, 2018) the authors give the example of a malicious party who has access to a neural network used in self-driving cars, which can be trojanised to behave erratically in the presence of very specific road signs. Similar trojan attacks can be found in [6] (Zou, 2018), and defences against some trojan attacks can be found in [7] (Chen, 2019) [8] (Gao, 2019), though this literature is largely limited to the domain of neural networks. Defence techniques often involve anomaly detection approaches used to identify and filter out training samples that appear significantly different from "typical" training data.

The fundamental privacy and security issues specifically related to Federated Learning stem from the fact that the local users have control over their local training process and have access to the global model [9] (Bagdasaryan, 2018). The first of these facts makes data poisoning attacks quite easy to carry out [10] (Bhagoji, 2018). Access to the global model makes possible the model replacement attacks of (Bagdasaryan, 2018), for which one cannot use anomaly detection as a defence in the presence of a secure aggregator [11] (Bonawitz, 2017), which can be considered a trade-off between higher user privacy and reliability of a global model. Giving individual users access to the global model also leaks valuable company IP to potential malicious parties.

Proposed Use Case of Task 3.5

The proposed focus of Task 3.5 is on studying model poisoning attacks against AI-based systems and designing methods of detecting and countering those attacks to increase the systems' reliability. Threats to reliability obviously have direct consequences for multiple ethical values in the use of SIS. Since Federated Learning and similar approaches to training Machine Learning models open new avenues for attacks and new needs in protection methods, an important part of the plan is to monitor actual development and application of the new model training approaches and attacks exploiting specific properties of those. As mentioned above, while such new approaches better protect privacy of the users, they also complicate poisoning detection.

Since the technical work in Task 3.5 is mainly being carried out by F-Secure, the main proposed use case is application of AI to dynamic detection of advanced cyberattacks. For dynamic attack detection systems, the most popular approach at the moment is to install sensors in an environment to be protected (customer's endpoints, servers and network equipment), stream relevant data from the sensors to a security backend, analyze the data there via a combination of machine learning methods, detection rules, and human expert analysis, and take decisions. Since parts of the customer environments may be controlled by attackers, the data coming to the security backend may be altered by the attackers (poisoned). Furthermore, the described attack detection approach requires transmitting large volumes of data and storing and processing those in a security provider backend, which clearly has serious data confidentiality and privacy implications for customers and data handling cost implications for security providers. Due to these challenges, moving parts of data processing and detection logic to customer endpoints looks highly appealing. That also includes the use of federated learning and similar approaches for training ML-based detection models. However, with all the cost-saving, confidentiality and privacy benefits, new risks are coming. With full access to the local sensor and detection logic, an attacker may be able to alter local model training to affect both local and global models so that chosen attacks are no longer detected. Exploring ways of dealing with these challenges is one of the key research directions in Task 3.5.

Progress Update (March 2020)

We chose to start with studying backdoor poisoning attacks, focusing on a common setting for data poisoning in which ML models, used to detect security-related anomalies in our case, (a) are trained online, and (b) use distributed training data. Distribution of the data means that a training dataset used to train an ML model is split into several parts owned by multiple parties. Online training means that an ML model is periodically and incrementally updated (e.g., on a daily basis) as the data owner parties collect and provide new training data. Online training using distributed data is a popular setup for training anomaly detection systems for two reasons. First, it enables leveraging the maximum amount of data (provided by multiple parties) in order to model comprehensively diverse normal (assumed to be benign in our case) behavior observed in multiple monitored systems. Second, updating a model on a regular basis enables adapting to changes and capturing the evolution of normal behavior of monitored systems.

We studied the vulnerability to backdoor poisoning attacks of a real anomaly detection system used to detect anomalous process launches in a computing system by modelling the distribution of typical process launch events and detecting deviations from this distribution. This model belongs to the class of statistical distribution models with thresholds. While simple, models of this class can be applied to many use cases, are easy to understand and, consequently, find numerous applications in the industry. Having carried out a systematic study, with careful definitions of attack surface and attacker’s capabilities, we identified three tactics of poisoning attacks against the anomalous process launch detection model and evaluated their effectiveness and efficiency considering natural adversarial goals. While aimed at a specific anomaly detection system, our three attack tactics, attempting (with varying levels of inventiveness) to distort the target model statistics or thresholds, can be generalized to essentially any statistical distribution model with thresholds. Based on the analysis of these tactics, we provide several mitigation recommendations, including (i) data format and range validation; (ii) normalization of local models prior to aggregation; (iii) strong client authentication; (iv) outlier detection for local models; (v) detection of abnormal evolution of local models over time.

Comments and Questions

… are most welcome and to be directed to:
alexey.kirichenko@f-secure.com

References

[1] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning, 2017. https://arxiv.org/abs/1712.05526.

[2] C. Yang, Q. Wu, H. Li, and Y. Chen. Generative poisoning attack method against neural networks, 2017. https://arxiv.org/abs/1703.01340

[3] B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against support vector machines, 2012. https://arxiv.org/abs/1206.6389

[4] B.I.P. Rubinstein, B. Nelson, L. Huang, A.D. Joseph, S. Lau, S. Rao, N. Taft, and J.D. Tygar. Antidote: Understanding and defending against poisoning of anomaly detectors, 2009. https://people.eng.unimelb.edu.au/brubinstein/papers/imc206-rubinstein.pdf

[5] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks, 2018. https://weihang-wang.github.io/papers/tnn_ndss18.pdf.

[6] M. Zou, Y. Shi, C.Wang, F. Li,W. Song, and Y.Wang. Potrojan: powerful neural-level trojan designs in deep learning models, 2018. https://arxiv.org/abs/1802.03043.

[7] H. Chen, C. Fu, J. Zhao, and F. Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks, 2019. https://cseweb.ucsd.edu/~jzhao/files/DeepInspect-IJCAI2019.pdf

[8] Y. Gao, C. Xu, D. Wang, S. Chen, D. Ranasinghe, and S. Nepal. Strip: A defence against trojan attacks on deep neural networks, 2019. https://arxiv.org/abs/1902.06531

[9] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning, 2018. https://arxiv.org/abs/1807.00459

[10] A.N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. Analyzing federated learning through an adversarial lens, 2018. https://arxiv.org/abs/1811.12470

[11] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H.˜ B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy-preserving machine learning, 2017. https://eprint.iacr.org/2017/281.pdf

This project has received funding from the
European Union's Horizon 2020 Research and Innovation Programme
Under Grant Agreement no. 786641