Poisoning Attacks on Federated Learning-based IoT Intrusion Detection System

Thien Duc Nguyen, Phillip Rieger, Markus Miettinen, Ahmad-Reza Sadeghi
Technical University of Darmstadt, Germany
(ducthien.nguyen, markus.miettinen, ahmad.sadeghi)@trust.tu-darmstadt.com, phillip.rieger@cysec.de

***Abstract*—Federated Learning (FL) is an appealing method for applying machine learning to large scale systems due to the privacy and efficiency advantages that its training mechanism provides. One important field for FL deployment is emerging IoT applications. In particular, FL has been recently used for IoT intrusion detection systems where clients, e.g., a home security gateway, monitors traffic data generated by IoT devices in its network, trains a local intrusion detection model, and send this model to a central entity, the aggregator, who then computes a global model (using the models of all gateways) that is distributed back to clients. This approach protects the privacy of users as it does not require local clients to share their potentially private IoT data with any other parties, and it is in general more efficient than a centralized system. However, FL schemes have been subject to poising attacks, in particular to backdoor attacks.**

In this paper, we show that FL-based IoT intrusion detection systems are vulnerable to backdoor attacks. We present a novel data poisoning attack that allows an adversary to implant a backdoor into the aggregated detection model to incorrectly classify malicious traffic as benign. We show that the adversary can gradually poison the detection model by only using compromised IoT devices (and not gateways/clients) to inject small amounts of malicious data into the training process and remain undetected. Our extensive evaluation on three real-world IoT datasets generated from 46 IoT devices shows the effectiveness of our attack in injecting backdoors and circumventing state of the art defenses against FL poisoning. Finally, we discuss shortly possible mitigation approaches.

I. Introduction

The market of Internet-of-Things (IoT) devices is booming as more and more users leverage wireless connectivity and intelligent functionality to access various services. However, many of these devices are riddled with security problems due to inadequate security designs and insufficient testing. Consequently, security vulnerabilities are exploited in various attack scenarios as shown recently by, e.g., ”IoT Goes Nuclear” [28], attacks against Honeywell [10], or a set of Z-Wave devices [11] as well as crucial large scale DDoS attacks [2], [35], [13], [36], [25]. Given that increasingly IoT devices are entering the market and a general security standard is missing, one can expect that many insecure devices continue to be deployed in many application domains. Patching IoT devices against known attacks is not effective due to the diversity of vulnerabilities and attacks. Hence, it is reasonable not to make many assumptions about the security architectures and features on IoT devices and rather counter security threats arising from attacks compromised devices, in particular, against unknown attacks. To detect compromised devices, network-based intrusion detection systems (NIDSs) can be deployed in end-user networks [23], [9], [26]. An NIDS passively monitors and analyzes device communications (network traffic) in order to detect if the network is under attack. A compelling NIDS approach that has the potential to detect previously unknown attacks is based on *anomaly detection*. It consists of training a model characterizing normal device behavior and using this model for detecting ”anomalous” behavior that deviates from the normal model. In this context *Federated Learning* (FL) seems to be an adequate tool, as FL is an emerging solution for the distributed training of machine learning models utilized in various application areas. It can provide benefits with regard to communication requirements and privacy of training datasets, which is why recently a number of FL-based systems have been proposed, e.g., for word prediction [20], [19], medical applications [32], [14], [8], as well as for IoT [23], [27], [30], [31], [18]. In FL, each local client participating in the system uses its private local training dataset to train a *local model*, which is sent to a central aggregator. The aggregator then uses a *federated averaging* algorithm to aggregate the local models to a *global model* which is then propagated back to the local clients. Especially for applications targeting IoT settings, FL can provide significant privacy benefits, as it allows local clients to participate in the system without the need to expose their potentially privacy-sensitive local training datasets to others. This is particularly important if behavioral data of IoT devices are used, since information about the usage and actions of IoT devices may allow to profile the behavior and habits of their users, thus potentially violating user privacy. Another benefit that FL provides in IoT settings is that the aggregation of locally trained models makes it possible to obtain accurate models quickly even for devices that typically generate only little data (e.g., simple sensors or actuators). Relying only on data available in the local network would require a lot of time to collect sufficient training data for an accurate model.

However, recent research shows that FL can be a target of *backdoor attacks*, a type of poisoning attack in which the attacker corrupts the resulting model in a way that a set of specific inputs selected by the attacker will result in incorrect predictions as chosen by the attacker. There are currently several backdoor attacks on image classification [33], [3], [12] and word prediction [3].

Workshop on Decentralized IoT Systems and Security (DISS) 2020
23-26 February 2020, San Diego, CA, USA
ISBN 1-891562-64-9
https://dx.doi.org/10.14722/diss.2020.23003
www.ndss-symposium.org

Goals and contributions. In this paper, we present backdoor attacks on FL-based IoT anomaly detection system, in which the attacker aims at poisoning the training data by stealthily injecting malicious traffic into the benign training dataset. Consequently, the resulting model would incorrectly classify malicious traffic as benign and fail to raise an alarm for such attack traffic patterns. We show that compromised IoT devices can be utilized by the attacker to implant the backdoor. We evaluate the effectiveness of our attack on a recent proposal for FL-based IoT anomaly-detection in [23]. In the anomaly detection scenario, a backdoor corresponds to malicious behavior generated by the attack, e.g., IoT malware that would be accepted as normal by the anomaly detection model.

Our main contributions as follows:

- We introduce a new attack approach that circumvents IoT intrusion detection system using Federated Learning (FL). In this attack, the attacker indirectly attacks FL-based IoT anomaly detection systems by controlling IoT devices to gradually inject malicious traffic. Contrary to existing poisoning approaches, our attack does not require the attacker to compromise clients [3], [12].
- We provide an extensive evaluation using *three timely real-world IoT datasets* related to a concrete FL-based IoT anomaly detection system to demonstrate the impact of our attack, showing that it can bypass existing defenses.

II. System and Threat model

In traditional anomaly detection settings, the model is learned based on training data originating from the objects to be modeled. The IoT setting, however, poses challenges for this approach. For one, IoT devices, being typically single-use appliances with limited functionality, do not generate significant quantities of data, making training of a model purely on data collected from the local network of a user challenging, as it may take a long time to aggregate sufficient data for training a stable model. This mandates an approach in which training data from several different users is aggregated into a joint training dataset, making it possible to learn a stable model faster.

On the other hand, however, it is not desirable to aggregate training data centrally, as the data obtained from the communication of IoT devices potentially can reveal privacy-sensitive behavioral information about users. To enable effective learning of detection models by making use of several user's training data while maintaining the privacy of individual users' datasets, *federated learning* can be applied. In contrast to a fully centralized learning architecture, in an FL setting each client (user) trains a *local model* based on its locally available training data instead of sending its data to a central entity. The locally trained client models are then aggregated by the central entity to a *global model*, which can then be distributed back to the clients to be used in anomaly detection locally, or used as a basis for subsequent training iterations for refining the model further. The advantage of this approach is that clients can benefit from information contributed by other clients while not having to share their detailed training datasets and thereby better protect the privacy of local users.

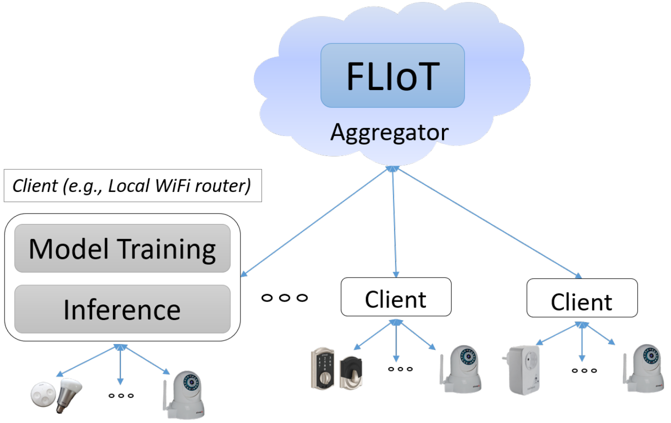

Fig. 1: Overview of the FL-based IoT intrusion detection system [23]

In a typical IoT scenario, the FL setting would be implemented by having in each local private IoT network (e.g., the smart home of a user) a dedicated security gateway (SGW) aggregating a training dataset from devices in the local network and training local detection models for those devices [23], [31]. The intelligent nodes of local networks would then share their local models with a central server aggregating the models and generating a global model from them. Similar learning setups have been successfully implemented, e.g., for device-type-specific intrusion detection [23].

A. System Model

We consider a setting in which FL is used to realize an anomaly detection-based intrusion detection system for IoT devices, as we have kindly received access to a number of real-world datasets (Sect. IV-A1) of IoT devices and IoT malware. We adopt the system setting, DÏoT, proposed by Nguyen *et al.* [23], in which neural network-based models are used to detect compromised IoT devices in local networks. The system is based on training a model with packet sequences in a device's communication flows and detecting abnormal packet sequences (e.g., generated by IoT malware) that are not consistent with the normal communications of the device in question. The overall system set-up is shown in Fig. 1. It consists of a number of local *Security Gateway*s, which collaborate with an *Aggregator* to train anomaly detection models based on GRUs (Gated Recurrent Unit, a type of Recurrent Neural Network (RNN)) [7] for detecting anomalous behavior of IoT devices. The *Security Gateway*s act as the local WiFi routers in end-user networks, so that all IoT devices, e.g., in the smart home of a user connect to the Internet through the *Security Gateway* over WiFi or an Ethernet connection. In this way, the gateway is able to monitor all communications of IoT devices in its network. To train detection models used for anomaly detection, an FL approach is applied: *Security Gateway*s locally train detection models, which they then send to the *Aggregator*, who will aggregate them to a global model and propagate this global model back to the *Security Gateway*s. Therefore, each *Security Gateway* can benefit from training data contributed by all participating gateways. During the operation of the system, the training of the detection model is iteratively repeated in order to gradually increase the accuracy of the model, as more training data become available. This repeated training process is performed either routinely, or, until the global model reaches a specific level of convergence, i.e., when the model doesn't improve significantly anymore. We formalize the operation of the system in two phases, training phase and detection phase, as follows:

Training phase. The global model G_k is trained through many training iterations $t = 1, 2, \ldots$ to learn the normal communication patterns of a specific device type k. In the training iteration t, the *Security Gateway*s receive global model G_k^{t-1} from previous training round $t-1$ then each *Security Gateway* i uses its own data D_{ik} generated by device type k to train G_k^{t-1} to achieve a local model W_{ik}^t as formalized in Eq. 1:

$$W_{ik}^t \leftarrow LocalTrain(G_k^{t-1}, D_{ik}) \tag{1}$$

Then those local models $(W_{1k}^t, W_{2k}^t, \ldots, W_{nk}^t)$ are sent to the *Aggregator*, in which it aggregates them into the global model G_k^t using FedAvg, a widely used aggregation algorithm proposed by McMahan *et al.* [19], as formalized in Eq. 2:

$$G_k^t = \frac{\sum_{i=1}^n n_{ik} W_{ik}^t}{n_k} \tag{2}$$

Where n_{ik} is the number of data points that the *Security Gateway* i has for device type k and n_k is total data points for device type k: $n_k = \sum_{i=1}^n n_{ik}$.

Detection phase. The intrusion detection is performed by the *Security Gateway*s by identifying the device communication behaviors that do not match with the trained global model corresponding to the device's type and raising an alarm, if such traffic is detected. It also collects training data for the training process mentioned above.

B. Threat Model

Attacker's goal. The attacker aims at corrupting the global model G so that G will provide incorrect predictions but attacker's chosen outputs C: $G(x) = c \in C$ for targeted inputs $x \in T$, while performing normally on benign inputs $x \notin T$. In IoT intrusion detection case, the attacker's goal is to manipulate the global anomaly detection model used for intrusion detection in a way that adversarial inputs like packet sequences of malicious traffic from, e.g., IoT malware like *Mirai* malware [2], [23] are incorrectly deemed normal by the detection model. Consequently, the model would not detect that malicious traffic as anomalous.

Attacker's capability. We assume that the attacker compromise and control a number of IoT devices in different local networks. The attacker can also use his own devices to connected to the *Security Gateway*s. The attacker has full control of the compromised IoT devices, e.g, controlling those devices to inject arbitrary traffic. For simplicity, we assume that the attacker has full control of d IoT devices in m *Security Gateway*s and m models trained from those *Security Gateway*s are poisoned models.

No compromised *Security Gateway*s required. In our attack, the attacker does not need to compromised *Security Gateway*s (clients) while current poisoning attacks often require compromising clients [3], [4], [6]. It makes our attack more practical than requiring *Security Gateway*s to be compromised because *Security Gateway*s are strictly secured devices.

Full knowledge of the targeted system but no control. The attacker has full knowledge of the operations and parameters of the intrusion detection system, e.g., DÏoT as described in Sect. II-A, but has no control of the system. He can only control compromised IoT devices.

Assumptions. The attacker can control one or more IoT devices in each of m *Security Gateway*s. Since compromising IoT devices takes considerable effort, we assume that m is at most less than half of the total number of clients n, i.e., $m < \frac{n}{2}$.

III. Our Data Poisoning Attack

In general, in order to evade an anomaly detection system, the attacker adjusts his attack in the way that the malicious behaviors are close to benign behaviors. However, a challenge of this approach is that the attacker has to modify malicious traffic in the way that the semantic of attack retains. Recent defenses, e.g., DÏoT show that by precisely modeling network communication of devices with the help of deep learning, their system can detect malicious traffic effectively. We, therefore, propose a different attack vector. Instead of trying to adjust attack traffic patterns, e.g., changing malware' behaviors to directly evade the anomaly detection, our attack targets the data collection state. Our intuition is that by injecting a small amount of malicious traffic into the benign traffic, this traffic will be not detected as anomalous, and will be learned as normal traffic in the training phase, i.e., the model is gradually backdoored. Consequently, the model will not detect that "backdoored" traffic as malicious in the detection phase. To do that, the attacker can use the compromised IoT devices to implant malicious traffic which will be used as training data by the *Security Gateway* running on the local network gateway (cf. Fig. 1).

We formalize our attack process as follows: In the absence of the attacker, a *Security Gateway* i monitors the communications of an IoT device type k and uses the communication data as training data D_{ik} for a behavioral model W_{ik} of the device type k as shown in Eq. 1. However, if an attacker $\mathcal{A}$ manages to compromise and take control over device d which has device type k, it can cause the device to inject specific malicious network traffic patterns $D_{\mathcal{A}}$ that will be captured by the *Security Gateway* i and used along with the benign traffic data D_{ik} as training data for training a local model W_{ik} for device type k, i.e., $D_{\mathcal{A}}$ is an additional input for the Eq. 1 which is generalized (without specific training round t) as in Eq. 3 as follows:

$$W'_{ik} \leftarrow LocalTrain(G_k, D_{ik} + D_{\mathcal{A}}) \tag{3}$$

Where W'_{ik} illustrates that model W_{ik} is poisoned with $D_{\mathcal{A}}$. Using this approach the attacker can thus poison the training dataset used to train model W_{ik} and can thereby introduce a backdoor in the model, as the attack network traffic patterns contained in $D_{\mathcal{A}}$ will be erroneously used by *Security Gateway* i as benign training data and consequently incorporated in model W'_{ik}.

The challenges of our attack. The challenges of our attack are two folds: It has to evade (1) the traffic anomaly detection of G_k [23] and (2) the model anomaly detection of the aggregator [33], [3]. For the former, the proportion of malicious traffic data injected by the compromised IoT devices must be small enough compared to the benign data so that it retains undetected by G_k. Otherwise, that malicious data will be excluded from the training data. One way to tackle this challenge is that the attacker can inject malicious traffic at the time that the compromised device d is generating a high volume normal traffic. For the later, if a model anomaly detection approach is deployed in the aggregator to detect anomalous models deviated from normal models (see Sect. V-A), the attacker has to also control the amount of poisoned data in the way that it does not make the poisoned model W'_{ik} deviated from the benign models, i.e., W'_{ik} retains undetected. To tackle this challenge, the attack can control poisoned data rates as explained as follows:

Controlling Poisoned Data Rate (PDR). In our attack the attacker $\mathcal{A}$ can control the ratio of poisoned traffic $D_{\mathcal{A}}$ it injects in the network with respect to the benign traffic D_{ik} the compromised device generates. We denote this ratio *Poisoned Data Rate (PDR)*:

$$PDR = \frac{|D_{\mathcal{A}}|}{|D_{ik}|}$$

By controlling the amount of malicious data $D_{\mathcal{A}}$ injected by the compromised device the attacker controls the PDR of its attack. Attacker $\mathcal{A}$ will seek to select a PDR that would provide an optimal balance between effectiveness and stealthiness of the backdoor attack, in which the higher the PDR used is, the better the accuracy of the backdoor will be. However, this will also make the poisoned model W_{ik} is more deviated from the benign models lead to be easier to detect. We will prove the effectiveness of this attack strategy in the Sect. IV-B and Sect. V.

IV. Evaluation

A. Experimental Setup

1) Datasets: To be comparable to the work of DÏoT, we use the same datasets. Moreover, we have also obtained the dataset from paper authors Sivanathan *et al.* [34]. In total, we evaluate our attack on three real-world datasets generated by 46 commodity IoT devices and infamous IoT malware, Mirai [2]. Here is the list of our used datasets:

- DÏoT-Benign: The IoT traffic has been generated from 18 IoT devices deployed in a real-word smart home [23].
- UNSW-Benign: The IoT traffic has been generated from 28 IoT devices in an office for 20 days [34].
- DÏoT-Attack: The attack traffic has been generated by 5 IoT devices infected by the Mirai malware which has 13 attack types, e.g., infection, scanning, SYN flood, HTTP flood, etc. [23].

Following the work from DÏoT described in Sect. II, we separate benign datasets for each device type resulting in 24 device-type-specific datasets in total and each of these datasets will be evaluated independently. To simulate FL setting, the dataset of each device type will be divided among clients, in which each client has an independent data of approximately 3000 packets.

2) Experimental implementation and metric: We implemented our attacks using the PyTorch framework [1] and conducted all experiments on a server with 20X Intel Xeon CPU cores, 64GB RAM, 4X NVIDIA GeForce GPUs (with 11GB RAM each), and Ubuntu 18.04 LTS OS.

To measure the effectiveness of our attack, we use for parameters as follows:

- **Backdoor Accuracy** (**BA**)**.** It refers to how good a poisoned model is in the backdoor task, i.e., it is the fraction of malicious samples that the system falsely classifies as normal samples to the total malicious samples.
- **Main Task Accuracy** (**MA**)**.** It indicates how good the anomaly detection model correctly detects benign traffic, i.e., it is the fraction of normal samples that the system correctly classifies as normal traffic to the total normal samples.
- **Poisoned Data Rate** (**PDR**) as defined in Sect. III).
- **Poisoned Model Rate** (**PMR**)**.** It is the fraction of the number of the gateways that have compromised IoT devices to the total number of the gateways, i.e., $PMR = \frac{m}{n}$.

B. Experimental Results

Malicious Data Injection. We conduct an experiment to find out how much data that the attacker can gradually inject in the network so that it is still under the radar of the system. We took 7,688 windows of data (250 samples per window) from five randomly chosen device types (AmazonEcho, DlinkCam, DLinkType05, EdimaxPlug, NetatmoWeather). We gradually increase the amount of malicious data injected in each window until the system detects this window is anomalous. As expected, the attacker needs only inject $PDR = 36.7 \pm 6.5\%$ malicious data in average to make the system incorrectly identifying the malware traffic as malicious.

Attack accuracy. To evaluate the accuracy of our attack in which controlling PDR strategy is applied in the condition of different poisoned model rates (PMR), we conducted an experiment on the Netatmo data (100 clients) with wide-range of PDR and PMR. As shown in Fig. 2, our attack achieves $BA = 100\%$ for all PMRs except for $PMR \leq 5\%$. For example, with $PDR = 35\%$ the adversary can poison the global model successfully $BA = 100\%$ with only $PMR = 20\%$, i.e., it requires only 20% of *Security Gateway*s has compromised IoT devices.

V. Defenses

A. Generalized clustering approach

To evaluate the effectiveness of our attack against state-of-the-art defenses based on distinguishing malicious and benign models, e.g., [33], [6], [3], we generalize those approaches as a clustering-based approach, in which we use k-means to cluster models into two groups, in which the bigger cluster

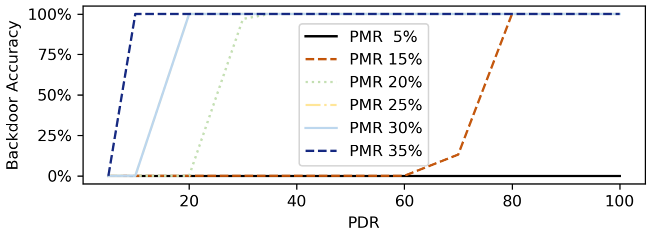

Fig. 2: Backdoor accuracy for different PDRs and PMRs. Note that MA is 100% for all attacks

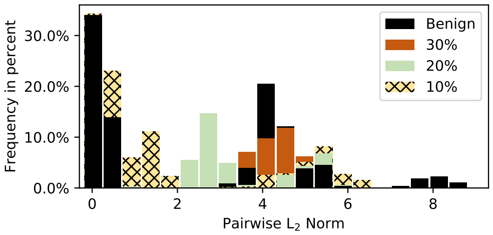

Fig. 3: Distribution of pairwise L$_2$-norms for different PDRs

is considered to be benign and the models of the smaller cluster are discarded, i.e., the models from the smaller group are assigned as malicious, e.g., [33]. We run an experiment with 100 client models in total, where 75 were benign and 25 under adversaries control, using our attack with a PMR of 25%. We preset k-means with number of clusters as two and use the pairwise L$_2$-norm distances as input. Figure 3 illustrates the distribution of these distances. As the figure shows, our attack imitates the L$_2$-norms of benign clients well. However, for higher PDRs the distances increase significantly. While for the majority of the benign clients, the malicious clients for all PDRs, have almost the same distances, as they are all in the first bar of the histogram, to another group of benign clients, the distances increases for higher PDRs. This is confirmed by the BAs and the clustering results. After 3 rounds of training, the BA reaches 100% for a PDR of 20%. However, the experiment also shows that a PDR of 30% is too high since here, the clustering is effective and filters the malicious clients out. Therefore, the attacker can select $10\% < PDR \leq 20\%$ to launch attacks successfully ($BA = 100\%$) without being detected.

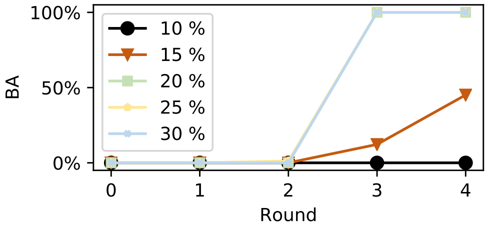

Fig. 4: BAs for different PDRs after multiple rounds

B. Differential Privacy

Another state-of-the-art defense is based on a differential privacy approach, as it was proposed by McMahan *et al.*[21] and discussed in detail in Sect. VI, in which all updates are downscaled, if they are higher than a fixed clipping boundary and Gaussian noise is added to all parameters after the aggregation. The rational behind this approach is that it reduces the attack impact by averaging out poisoned updates. Figure 4 shows the BA for different PDRs after multiple rounds. The setup is the same as in Sect. V-A. We chose a clipping boundary of two, based on the update norms for the benign clients. We evaluated a wide-range of noise levels (standard deviation of Gaussian noise) as from 0.0001 to 0.0075. Figure 4 shows the BAs for the highest noise level. It shows that the attack is successful against the clipping and noising defenses since it reaches a BA of 100% after 4 rounds with a PDR of at least 20%. However, on the other side, the defense also causes a drop in the MA. For the highest noise level, the MA goes down to 94.2%, while our attack is still successful. For the lower noise levels, BA still retains 100%. Therefore, this defense is not effective to mitigate our attack unless a scarifying MA, i.e, this defense is not practical.

C. Possible Mitigation Approaches

As discussed above, existing defenses are ineffective to mitigate our attacks, this highlights the need for new defense approaches. In this section, we discuss possible solutions that are potential for future works. In general, there are three possible directions to find a solution to mitigate our attacks as follows:

- **New poisoning FL defenses on the sever-side**. The first direction is to introduce a new FL poisoning defense deployed on the aggregator, e.g., finding better features and clustering algorithms to identify poisoned model updates. Since this approach only investigates the model updates, it does not require changes in the IoT intrusion detection part.
- **Filtering or tolerating poisoned data on the client-side.** Since clients, e.g., *Security Gateways* has full control of the training data, introducing a poisoned data filtering or tolerating approach is a potential solution. For example, we can investigate the possibility of utilizing existing poisoning defenses for centralized learning, e.g, noisy data tolerance [17], [22] or outlier data mitigation [29].
- **Identifying malicious traffic injection.** Another solution is to improve the sensitivity IoT intrusion detection systems to be able to identify even small amount of malicious traffic at the injection state and discard this traffic data.

VI. Related Work

Poisoning attacks against machine learning models originally targeted centralized training settings [5]. These attacks intend to modify decision boundaries (or cause concept drift) by manipulating training inputs to the model [15]. This goal is achieved by modifying training input labels [5], by crafting synthetic inputs meant to produce a slow concept drift [16] or by injecting noise in samples before feature extraction [24]. Mitigating such attacks can be addressed using training methods tailored to deal with noisy data [17], [22].

In FL settings, a number of poisoning attacks have been proposed [12], [33], [6], [3]. These attacks focus only on benchmark datasets of text prediction or image classification applications. In these attacks, the attacker has full control of compromised clients, i.e., the adversary can arbitrarily change training datasets or training algorithms. For example, Bagdasaryan *et al.* [3] propose a *constrain-and-scale* attack that the attacker can fully manipulate the training process: data, algorithms, and parameters. We however consider a different application setting as IoT and introduce a new attack approach. Moreover, our attack does not require attackers to compromise clients, i.e., it requires weaker attackers' capability, making our attack more practical.

To tackle poisoning attacks in FL, several defenses have been proposed [12], [33], [6]. The main idea of those defense approach is to identify malicious updates that deviated from the benign updates. For example, Auror [33] tries to cluster model updates into two classes based on indicated features, in which the smaller class will be identified as the malicious class and filtered out. Blanchard *et al.*introduces Krum [12], in which the client that has the smallest sum of distances to other $n-m-2$ clients will be selected as the global model. We generalize these clustering-based approaches and show that it is not effective to defend our attack (see Sect. V-A). Another defense approach is to reduce the attack impact by scale down models that have high update amplitudes or add noise to average out poison updates [3], [21]. However, this approach is not practical because it also damages the performance of the model in the main task as shown in Sect. V.

VII. Conclusion

In this paper, we introduce a novel backdoor attack on FL-based IoT intrusion detection system, in which the attacker can gradually inject poison data via compromised IoT devices (and not gateways/clients compared to existing approaches). Our extensive evaluation on three real-world IoT datasets generated from 46 IoT devices and infamous IoT malware, Mirai, shows that our attack is effective and bypasses current defenses. This raises the need for new defense mechanisms against our attacks on FL-based IoT intrusion detection system.

Acknowledgment

This work was supported by the Intel Collaborative Research Institute for Collaborative Autonomous & Resilient Systems (ICRICARS) and the Deutsche Forschungsgemeinschaft (DFG) – SFB 1119 – 236615297.

References

[1] "Pytorch," 2019, https://pytorch.org.

[2] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, 2017, pp. 1093–1110.

[3] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," *CoRR*, vol. abs/1807.00459, 2018. [Online]. Available: http://arxiv.org/abs/1807.00459

[4] M. Baruch, G. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," *CoRR*, vol. abs/1902.06156, 2019. [Online]. Available: http://arxiv.org/abs/1902.06156

[5] B. Biggio, B. Nelson, and P. Laskov, "Support vector machines under adversarial label noise," in *Asian Conference on Machine Learning*, 2011, pp. 97–112.

[6] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems, NIPS*. Curran Associates, Inc., 2017, pp. 119–129.

[7] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014, http://arxiv.org/abs/1412.3555.

[8] T. M. Deist, A. Jochems, J. van Soest, G. Nalbantov, C. Oberije, S. Walsh, M. Eble, P. Bulens, P. Coucke, W. Dries, A. Dekker, and P. Lambin, "Infrastructure and distributed learning methodology for privacy-preserving multi-centric rapid learning health care: eurocat," *Clinical and Translational Radiation Oncology*, vol. 4, pp. 24 – 31, 2017.

[9] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," *CoRR*, vol. abs/1804.04159, 2018. [Online]. Available: http://arxiv.org/abs/1804.04159

[10] D. Fisher, "Pair of bugs open honeywell home controllers up to easy hacks," https://threatpost.com/pair-of-bugs-open-honeywell-home-controllers-up-to-easy-hacks/113965/.

[11] B. Fouladi and S. Ghanoun, "Honey, i'm home!!, hacking zwave home automation systems," black Hat USA. [Online]. Available: https://cybergibbons.com/wp-content/uploads/2014/11/honeyimhome-131001042426-phpapp01.pdf

[12] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *CoRR*, vol. abs/1808.04866, 2018. [Online]. Available: http://arxiv.org/abs/1808.04866

[13] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and analysis of hajime, a peer-to-peer iot botnet," in *26th Annual Network and Distributed System Security Symposium, NDSS, San Diego, California, USA, February 24-27, 2019.*

[14] A. Jochems, T. M. Deist, I. E. Naqa, M. Kessler, C. Mayo, J. Reeves, S. Jolly, M. Matuszak, R. T. Haken, J. van Soest, C. Oberije, C. Faivre-Finn, G. Price, D. de Ruysscher, P. Lambin, and A. Dekker, "Developing and validating a survival prediction model for nsclc patients through distributed learning across 3 countries," *International Journal of Radiation Oncology*Biology*Physics*, vol. 99, no. 2, pp. 344 – 352, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360301617308258

[15] M. Kearns and M. Li, "Learning in the presence of malicious errors," *SIAM Journal on Computing*, vol. 22, no. 4, pp. 807–837, 1993.

[16] M. Kloft and P. Laskov, "Online anomaly detection under adversarial impact," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 405–412.

[17] N. Manwani and P. Sastry, "Noise tolerance under risk minimization," *IEEE transactions on cybernetics*, vol. 43, no. 3, pp. 1146–1151, 2013.

[18] S. Marchal, M. Miettinen, T. D. Nguyen, A. Sadeghi, and N. Asokan, "Audi: Towards autonomous iot device-type identification using periodic communication," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2019.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pp. 1273–1282.

[20] B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," 2017. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[21] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning

differentially private language models without losing accuracy," *Sixth International Conference on Learning Representations*, 2018. [Online]. Available: http://arxiv.org/abs/1710.06963

[22] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 1196–1204.

[23] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A. Sadeghi, "DÏoT: A federated self-learning anomaly detection system for iot," in *The 39th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2019, 10.1109/ICDCS.2019.00080.

[24] R. Perdisci, D. Dagon, W. Lee, P. Fogla, and M. Sharif, "Misleading worm signature generators using deliberate noise injection," in *Security and Privacy, 2006 IEEE Symposium on*. IEEE, 2006, pp. 15–pp.

[25] Radware, "BrickerBot results in PDoS attack," https://security.radware.com/ddos-threats-attacks/brickerbot-pdos-permanent-denial-of-service/.

[26] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Hyperspherical cluster based distributed anomaly detection in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 74, no. 1, pp. 1833–1847, 2014.

[27] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing-supported internet of things," *IEEE Access*, vol. 7, pp. 69 194–69 201, 2019.

[28] E. Ronen, A. Shamir, A. Weingarten, and C. O'Flynn, "Iot goes nuclear: Creating a zigbee chain reaction," *IEEE Security Privacy*, vol. 16, no. 1, pp. 54–62, 2018.

[29] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-h. Lau, S. Rao, N. Taft, and J. Tygar, "Antidote: understanding and defending against poisoning of anomaly detectors," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*. ACM, 2009, pp. 1–14.

[30] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Federated Learning for Ultra-Reliable Low-Latency V2V Communications," *Global Communications Conference, 2018*, 2018.

[31] J. Schneible and A. Lu, "Anomaly detection on the edge," in *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, 2017, pp. 678–682.

[32] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas, "Federated learning: Collaborative machine learning without centralized training data," 2018. [Online]. Available: https://www.intel.ai/federated-learning-for-medical-imaging/

[33] S. Shen, S. Tople, and P. Saxena, "Auror: Defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32Nd Annual Conference on Computer Security Applications*, ser. ACSAC '16. ACM, 2016, pp. 508–519.

[34] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, 2018.

[35] S. Soltan, P. Mittal, and H. V. Poor, "Blackiot: Iot botnet of high wattage devices can disrupt the power grid," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, 2018, pp. 15–32.

[36] T. Yeh, D. Chiu, and K. Lu, "Persirai: New internet of things (IoT) botnet targets IP cameras," TrendMicro, 2017, https://blog.trendmicro.com/trendlabs-security-intelligence/persirai-new-internet-things-iot-botnet-targets-ip-cameras/.