IoTGAN: GAN Powered Camouflage Against Machine Learning Based IoT Device Identification

Tao Hou[†], Tao Wang[‡], Zhuo Lu[†], Yao Liu[†] and Yalin Sagduyu[§]
[†]University of South Florida, Tampa, FL, USA, {taohou@, zhuolu@, yliu@cse.}usf.edu
[‡]New Mexico State University, Las Cruces, NM, USA, taow@nmsu.edu
[§]Intelligent Automation Inc, Rockville, MD, USA, ysagduyu@i-a-i.com

***Abstract*—With the proliferation of IoT devices, researchers have developed a variety of IoT device identification methods with the assistance of machine learning. Nevertheless, the security of these identification methods mostly depends on collected training data. In this research, we propose a novel attack strategy named *IoTGAN* to manipulate an IoT device's traffic such that it can evade machine learning based IoT device identification. In the development of IoTGAN, we have two major technical challenges: (i) How to obtain the discriminative model in a black-box setting, and (ii) How to add perturbations to IoT traffic through the manipulative model, so as to evade the identification while not influencing the functionality of IoT devices. To address these challenges, a neural network based substitute model is used to fit the target model in black-box settings, it works as a discriminative model in IoTGAN. A manipulative model is trained to add adversarial perturbations into the IoT device's traffic to evade the substitute model. Experimental results show that IoTGAN can successfully achieve the attack goals. We also develop efficient countermeasures to protect machine learning based IoT device identification from been undermined by IoTGAN.**

***Index Terms*—Internet of Things, Device Identification, IoT Security, Machine Learning, Generative Adversarial Network**

I. Introduction

The Internet of Things (IoT) refers to the network of physical devices that are embedded with sensors, chips, operating systems and other technologies, collecting and exchanging data over Internet [1]. The popularization of universal computer chips and the ubiquity of wireless networks enable the revolution to transform traditional devices into smart devices as a part of the IoT. These IoT devices can be extensively deployed for different purposes including consumer (e.g., smart home, health care), commerce (e.g., transportation, manufacturing, agriculture), and military (e.g., battlefield equipment, autonomous reconnaissance). However, the heterogeneity of these devices also imposes security challenges to the management of IoT networks.

For a network containing different kinds of IoT devices, it is vital to identify the type of each IoT device before applying fine-graded security policies. Other than managing different kinds of IoT devices locally by the entity which the devices belong to, knowing the type of the device can enable a global management for security purpose from the level of the whole network, in order to permit or prohibit IoT device's specific behavior. For example, in a military base, the network should keep the geographical information confidential and forbid the surveillance camera transferring video data to the outside. Another example is that an organization may have different permissions for its personnels to access different smart devices (e.g., the maintenance staff can adjust the air conditioner unit; the security guard can view the monitor video; and any person should be able to control the smart bubble). More importantly, IoT device identification can facilitate detecting vulnerable IoT devices and preventing malicious rogue IoT devices.

Researchers have proposed various methods for IoT device identification. A simple way is using identifiers (e.g., MAC addresses, IP addresses, Bluetooth ID, Zigbee ID) to identify IoT devices. However, various identifier spoofing attacks [2], [3] have been exploited to deceive the identification. It is necessary to develop new methods that can avoid using these identifiers for IoT device identification. Recent progresses including using statistical features which can reflect the behavior of a specific IoT device are developed for identification [4]. However, due to the limitation of mathematical model specific analysis, it is hard for these methods to keep a steady accuracy rate when applying to different real-world scenarios. Moreover, these methods usually introduce a high overhead and fail when the traffic is encrypted.

To overcome these weaknesses, researchers have introduced multiple machine learning based methods to assist IoT device identification or network traffic fingerprinting [5]–[14]. Specifically, these machine learning based methods can successfully identify IoT devices even when the identifier is spoofed, meanwhile they can achieve a high accuracy rate no matter whether the traffic is encrypted. These methods can be usually formalized as feature-based machine learning classification problems. k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), Random Forest and Neural Networks are the most frequently used approaches to build the machine learning model. Through learning on traffic features on packets (e.g., packet size, header information, even encrypted content in data payload [15]) or flows (e.g., packet interval, packet count in a time window), these machine learning models can achieve an accuracy rate as high as 99% in IoT device identification.

Nevertheless, machine learning based IoT device identification methods are designed with the aim for high performances, but lacks any security guarantee. We discover that there exists a subtle attack surface that can undermine the machine learning based IoT device identification. Specifically, these methods rely on learning from the features which are obtained from network traffics completely or partly controlled by IoT devices. Thus, a rogue IoT device can maliciously alter its network traffic to evade the machine learning based identification. In this research, we aim to investigate current machine learning based IoT device identification methods, reveal the potential attacks, and derive corresponding countermeasures to protect the IoT device identification from been undermined.

To this end, we develop an attack strategy, named *IoTGAN*, which can efficiently disturb the machine learning based IoT device identification. IoTGAN is inspired by Generative Adversarial Network (GAN) [16] and it allows rouge IoT devices to manipulate its traffic to camouflage themselves from been identified. We implement IoTGAN as a practical system to launch this attack. However, IoTGAN can not be simply implemented by directly using GAN. Two major technical challenges must to be addressed to achieve the malicious goal of evading IoT device identification: (i) How to obtain the discriminative model in black-box settings, and (ii) How to add perturbations to IoT traffic through the manipulative model (i.e., the generative model in GAN) to evade the identification while not influencing the functionalities of IoT devices.

In addition to the attack strategy design and analysis, we develop an effective defense approach, named *Device Profiling*, to protect the machine learning based IoT device identification from been undermined by IoTGAN. *Device Profiling* utilizes the raw wireless signals emitted from IoT devices, which exhibits inherent hardware features and cannot be manipulated by IoT devices, to mitigate the effect of IoTGAN.

We conduct experiments on real-world IoT devices with different machine learning based identification methods to evaluate the effectiveness of IoTGAN. The experimental results show that IoTGAN can evade all the identification methods with a successful rate higher than 90%. We also conduct experiments to evaluate the defense approach and observe that the attack successful rate significantly drops to nearly zero after the deployment of Device Profiling.

The remainder of this paper is as follows. In Section II, we introduce preliminaries. In Section III, we investigate the existing machine learning base IoT device identification methods, and state our attack strategy of IoTGAN. In Sections IV and V, we introduce the two core technical contributions in the implementation of IoTGAN. We discuss potential countermeasures to improve security in Section VI and present the experiment results in Section VII. Finally, we conclude the paper in Section VIII.

II. Preliminaries

We introduce the preliminaries in this section, including the architecture of IoT and general IoT devices.

A. *IoT Architecture*

Organizations like The IEEE Standards Association, International Electrotechnical Commission, and American National Standards Institute are working on developing the standards for IoT. Generally, IoT follows a multi-layer architecture.

- **Physical layer:** This layer includes the low level hardware components, such as sensors, actuators and RFIDs. IoT relies on these essential components to perform the fundamental functionalities (e.g., monitor the environment, collect information, manage operations).
- **MAC/link layer:** This layer connects different devices to a network, thus to enable transmitting or exchange of the data which is collected from physical layer. The connectivity can be achieved through different kind of protocols, including WiFi, NFC, Bluetooth, ZigBee and cellular networking.
- **Network layer and above:** The network layer and above connect IoT devices together via networking and provide application-level services to the end user. There are more than hundreds of applications in the IoT ecosystems, such as smart home, smart transportation, and smart city.

B. *IoT Devices*

An IoT network may be connected with heterogeneous devices for different applications. We introduce the mainstream IoT products in current marketplace, with the application domain they belong to and the adopted communication protocols in Table I. We can find that WiFi and bluetooth are commonly used in smart home applications. While applications of smart transportation and smart city prefer cellular networks for long distance connection.

TABLE I
Mainstream IoT products.

Name	Application	Communication Protocol
Google Nest	Smart Home	Bluetooth, WiFi
Apple AirTag		Bluetooth, UWB, NFC
Amazon Echo		Bluetooth, WiFi, Zigbee
Samsung SmartThings		WiFi, Zigbee, Z-Wave
Drone	Smart Transportation	WiFi, Cellular, Telemetry
Smart car		Bluetooth, WiFi, Cellular
Rail detector		LoRa, Cellular
Smart trash	Smart City	WiFi, Zigbee
Weather station		WiFi, LoRa, Cellular
Smart street light		Cellular
Gunshot detector		WiFi, Cellular

III. Evading ML-based IoT Device Identification

To evade identification, it is essential for the attacker to have the knowledge of the target machine learning models. Therefore, in this section, we first investigate existing machine learning based IoT device identification methods, then we introduce the attack model to achieve the adversarial goal.

A. *Existing IoT Device Identification Methods*

Remote service creates a subtle attack surface to infer the identity of different IoT devices. As most IoT devices request remote service via the RESTful APIs, which adopt the uniform interface to improve the visibility of interactions, the attackers are able to learn their identities by exploring unique features extracted from uniform headers of the service request. The work in [5] shows that the ML classifiers (i.e., SVM, and Logistic Regression) can reach high accuracy when the header features are considered (e.g., port numbers, domain names, and cipher suites).

Nevertheless, modeling remote service requests may not always yield sufficient information for accurate device identification, especially when the device is communicating with an anonymous service provider. More information can be gathered by learning the traffic/data flow patterns during the interaction. In particular, multiple works have been done to learn the spatial and temporal patterns of traffic/data flow to improve the identification accuracy. The work in [6] has collected and characterized the statistical attributes of traffic traces over 20 types of IoT devices and achieved a detection rate of 95%. The work of [7] models the periodic communication traffic using fingerprints extracted from frequency domain and adopts the k-NN classifier with the detection rate of 98.2%. The authors in [8] automate the process of feature extraction via genetic algorithm and deploy various machine learning algorithms (i.e., DecisionTable, J48 Decision Trees, OneR, and PART) to increase the detection rate. The authors in [11] develop a multi-stage meta classifier that explores the flow-level attributes to further improve the classification accuracy based on the network traffic analysis. Deep learning models (e.g., CNN and RNN) have also been applied to achieve the fine-grain device identification. The work in [12] converts the network payloads into image representation to fully capture the traffic details and achieves over 99% overall average detection accuracy. The authors in [13] propose a hybrid supervised and unsupervised deep learning approach to enable the refined classification for both known and unknown device types.

Important Features Used in IoT Device Identification: Feature extraction is a key component in machine learning to have an accurate device identification. Here we summarize the most commonly used features in IoT device identification and place them into two categories: (1) Remote service features, including service request interval, service volume, service type (e.g., NTP, DNS, Storage), service domain name, and service active/sleep cycles. (2) Network packet/flow features, including local port, remote port, local address, remote address, encryption algorithm, packet size, packet interval, and communication protocol.

B. Attack Model

Machine learning based IoT device identification systems usually are hosted on the network administration side. For the attacker, the target system is a black box without disclosing the knowledge of the model internal structure and features used for identification. In this research, we aim to develop a camouflage attack that can help IoT devices to evade the machine learning based identification in a black-box setting. Though the black-box setting makes it more challenging to launch the attack, it will promote the attack to be more practical for real-world scenarios.

Specifically, we proposed a system, named IoTGAN, to achieve this attack goal. IoTGAN is inspired by GAN [16]. GAN is comprised by two models: the discriminative model and the generative model. The generative model is trained to generate new samples by adding noise to the input data, while the discriminative model is used to distinguish between generated samples and real samples. Following a two-player minimax game, the generative model will be continually updated until the game achieves equilibrium.

As Figure 1 shows, IoTGAN includes the discriminative model and the manipulative model. The discriminative model has the same functionality as in GAN to help improve the manipulative model until the manipulated IoT device traffic can evade the identification. While the manipulative model works as the generative model in GAN to manipulate the traffic of IoT devices by adding perturbations. In next two sections, we introduce our core technical contributions in the implementation of IoTGAN: (i) How to obtain the discriminative model in a black-box setting, and (ii) How to add perturbations to IoT traffic through the manipulative model to evade the identification without affecting the functionality of IoT devices.

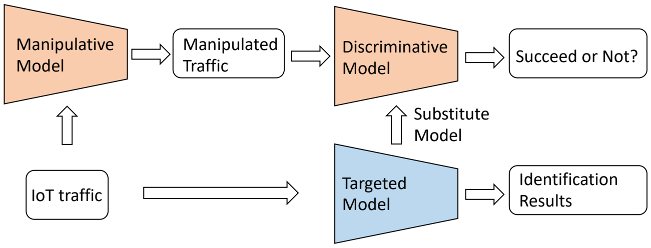

Fig. 1. The architecture of IoTGAN.

IV. Obtaining The Discriminative Model in A Black-box Setting

In this section, we introduce our design of the discriminative model in IoTGAN. The discriminative model should have the same population distribution as the target IoT device identification model, i.e., for the same input, it should have the same identification results. We aim to make it practical and can be broadly applied to a generic situation (i.e., a black-box setting in which the attacker has no knowledge of the target system). In what follows, we first present a general formula of IoT device identification. Then, we design our algorithm to obtain the discriminative model.

According to the literature, IoT device identification algorithms are either statistical or machine learning based methods given users' traffic features. We denote the target black-box identification algorithm as a multi-class classifier M. The inputs of M are K traffic features of a IoT device denoted as $\mathcal{H} = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_K\}$, where $\mathbf{h}_i$ is the i^{th} traffic feature of the device. Accordingly, the identification results $\mathcal{S}$ can be written as

$$\mathcal{S} = M(\mathcal{H}). \tag{1}$$

The output is the identified IoT class denoted as $\mathcal{S} = \{S_i\}, i \in [1, 2, ..., N]$, where N is the total classes of IoT devices that can be identified. The identifier aims to estimate the likelihood probability $P(S|H)$, which is the probability of the IoT class S given the traffic input H. As shown in Figure 2, our goal is to establish a substitute model, namely, $\hat{M}$, adapted to the input-output relation of the target black-box model M, such that $\hat{M}$ and M have the same identification results for the same group of IoT devices. This problem in fact falls into the area of model transferability [17] in machine learning. When labeled training datasets are collected from the same population distribution of the target black-box model, it is feasible to train a substitute model even when it has different internal structures [18]. Without loss of the generality, we adopt multi-layer full-connected neural network to learn the target black-box identifier. Sigmoid function is applied at the last layer and the class associated with the maximum probability will be selected as the output.

A. Training Data Collection

Due to the broadcast nature of wireless communications, we assume an attacker can eavesdrop on the traffic of IoT devices and observe the identification results from the network administrator. We then extract the features from the IoT traffics and treat them as the inputs for the substitute model $\hat{M}$. The identification results from the network administrator will be treated as the output. As discussed, the attacker has no knowledge of traffic features used in the black-box model. To address the issue, we surveyed existing ML based IoT device identification methods and build a feature pool that contains all common traffic features used in these models. The feature pool will be used as the start point of model training.

B. Substitute Model Training

Our model training takes two steps, 1) to obtain a substitute model that can yield the same results as the target identifier, 2) to mitigate the performance overhead by selecting a refined subset from the feature pool.

The process to obtain the accurate substitute model can be stated as finding an $\hat{M}$ that minimizes the empirical loss L over the training dataset. Specifically, the process can be formulated as

$$\hat{M} = \arg\min L(\mathcal{S}, \hat{\mathcal{S}}) = \arg\min_{\hat{M}} L(\mathcal{S}, \hat{M}(\mathcal{H})), \tag{2}$$

where $\hat{\mathcal{S}}$ is the predicted identification result yielded by the substitute model $\hat{M}$ given input $\mathcal{H}$. The obtained substitute model will work as the discriminative model in IoTGAN.

The first step focuses on building an accurate substitute model, but it may incur considerable amount of computational overhead as hundreds of features are used in model training. To this end, IoTGAN carefully shrinks the parameter space by selecting a refined subset where only features that indeed affect accuracy will be maintained from the pool. In particular, we adopt a weight training algorithm to obtain weights for all features in the pool. Then, we create the subset by selecting top L weighted features and retrain the labeled data. In order to determine the optimal size of the subset, we further define a metric Performance Gain to evaluate the prediction results of the substitute model given different values of L.

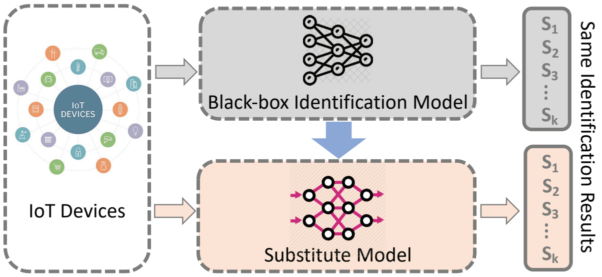

Fig. 2. Obtaining a substitute model as the discriminative model in IoTGAN.

$$\text{Performance Gain} = \frac{(r_c - r_p)/r_c - (c_c - c_p)/c_c}{(c_c - c_p)/c_c}.$$

where r_c and r_p represent the accuracy rate for current and previous subset size of L respectively, and c_c and c_p denote the computational overheads for current and previous subset size of L respectively. In particular, the computational overhead is defined as the time required to identify the class of IoT device using the substitute model. A positive Performance Gain indicates that the accuracy grows faster than the computation overhead with the increasing size of the subset. In IoTGAN, L is chosen empirically according to the performance gain. Specifically, we aim to keep a refined feature subset that can obtain a substitute model with high accuracy rate but at the same time impose negligible time overhead on device identification.

V. Manipulating the Traffic of IoT Device to Evade Identification

In this section, we design the manipulative model in IoTGAN. The manipulative model is used to manipulate IoT device's traffic such that the discriminative model can not successfully identify the IoT devices. As we introduced in aforementioned content, the discriminative model in IoTGAN is a substitute model of the target identification model, therefore IoTGAN just needs to manipulate IoT device traffic to evade the obtained substitute model. In particular, the manipulate model is trained following the strategy of training the generative model in GAN. This training procedure minimizes the probability of correct identification in the discriminative model. The training process of the manipulative model in IoTGAN is shown in Algorithm 1.

The manipulative model takes IoT device traffic $\mathbf{h}$ and a noise $\mathbf{s}$ as input. The noise is a vector of traffic features. Specifically, we only consider to manipulate the features that do not affect the functionality of IoT devices. For example, the content in data payload of a packet will not be changed to avoid disturbing the communication of IoT devices. As most features used in IoT identification (e.g.,service request

Algorithm 1: Training process of the manipulative model in IoTGAN.

```
Input  : Original IoT device traffic feature vector $\mathcal{H}$
Output: Trained manipulative model
while not satisfy (2) do
    Learning on original traffic $\mathcal{H}$;
    Update the substitute model $\hat{M}$;
/* Obtain the substitute model.                */
while Not converging do
    Initialize the multiplier factor $\mathbf{r}$;
    Set the adversarial perturbation $\mathbf{s} = \mathbf{rh}$;
    /* Get the perturbation.                   */
    Generate the manipulated traffic $\mathbf{h}' = G_\theta(\mathbf{h}, \mathbf{rh})$;
    Label $\mathbf{h}'$ using the substitute model $\hat{M}$;
    Update the manipulative model's parameter set $\theta$.
/* Train the manipulative model.               */
```

interval, service volume, local port, remote port) does not associate with the content of a message, it is feasible to fool the discriminative model by manipulating the traffic features. In IoTGAN, the manipulative model G generates the manipulated traffic $\mathbf{h}'$ by

$$\mathbf{h}' = G_\theta(\mathbf{h}, \mathbf{rh}), \tag{3}$$

where $\mathbf{r} = \{r_1, r_2, ..., r_n\}$ is the multiplier factor for generating noise $\mathbf{s}$, n is the number of elements in $\mathbf{h}$. For $i \in 1, 2, ..., n$, r_i is 0 if the corresponding feature may influence the functionality of IoT device; otherwise, r_i a random number sampled from a uniform distribution of [0, 0.1]; and θ is the parameter set of G. Since we consider the discriminative model as a multi-class identifier, we cannot simply adopt existing training procedure of GAN which only works for the classifier with binary decisions. In this paper, we propose a refined training process with two operation modes 1) device misidentification, 2) identity spoofing.

A. Device Misidentification

In device misidentification, the attacker aims to manipulate the traffic features such that the discriminative model will mislabel the given datasets. The process to achieve the purpose is to find a manipulative model G that can maximize the empirical loss L between the original and modified outputs over the training dataset. Specifically, the process can be formulated as

$$G_\theta = \arg\min L(\hat{\mathcal{S}}, \hat{\mathcal{S}}_{G_\theta}) = \arg\max_{G_\theta} L(\hat{M}(\mathcal{H}), \hat{M}(G_\theta(\mathbf{h}, \mathbf{rh}))), \tag{4}$$

where $\hat{\mathcal{S}}_{G_\theta}$ is the predicted identification results of $\hat{M}$ given the modified input $G_\theta(\mathbf{h}, \mathbf{rh})$. After training, the attacker can hide the identify of IoT devices from been identified by the network administrator.

B. Identity Spoofing

The misidentification can hide the identify of different IoT devices but cannot camouflage them as any specified classes for malicious purpose. For example, a surveillance camera may want to pretend as a device with no sensitive data to circumventing the rigorous export control policy. Towards this objective, identity spoofing aims to generate modified features that can fool the target model and yield the specified identification outputs designated by the attacker.

The process of identify spoofing can be formulated as finding a manipulative model G to minimize the empirical loss L between the specified class and outputs given the modified training dataset. Specifically, it is described as

$$G_\theta = \arg\min_{G_\theta} L(S_{spoof}, \hat{M}(G_\theta(\mathbf{h}, \mathbf{rh}))), \tag{5}$$

where s_{spoof} is the class specified by the attacker. After training, the attacker is able to deceive the discriminative model and pretend as any type of IoT device.

VI. Countermeasures

As discussed in Section III, traffic based features can be subtly manipulated by IoT devices to evade machine learning based IoT device identification. To defend against IoTGAN, we aim to identify different IoT devices using features that cannot be easily manipulated. The proposed classifier is complementary to existing IoT identification methods and can be easily integrated with them to improve identification accuracy.

We observe that raw wireless signals emitted from different IoT devices can exhibit inherent hardware features that cannot be forged by common users. In particular, manufacturing imperfection existing in IoT devices may impose a substantial change on the transmit signal waveforms, yielding unique features for device identification. We also note that wireless signals experience distinct channel distortions when they travel through different propagation paths [19]–[21]. We aim to take advantage of these nonlinear characteristics of radio channels to fingerprint different devices at various locations.

Specifically, we propose a method named *Device Profiling* to distinguish different IoT devices. The method includes two components: 1) feature profiling, which statistically describes the nonlinear characteristics of the transmit signals from IoT devices; 2) device fingerprinting, which builds a neural network based multi-stage classifier to learn the feature patterns for accurate and efficient device identification, even when the traffic based features are manipulated.

In feature profiling, we profile different IoT devices using features extracted from radio frequency signals in four perspectives: amplitude attenuation, phase shift, frequency offset, and arrival angle. In particular, amplitude attenuation and phase shift can be extracted from channel estimation at the receiver. Both features indicate the channel distortions caused by the internal hardware imperfection and the distinct progation paths. Frequency offset can be estimated by the maximum likelihood algorithm to derive the frequency deviation caused by transmitters's internal imperfections. The arrival angle can be measured via the multiple-input and multiple-output (MIMO) technique and exhibits the location-specific information of propagation channels.

In device fingerprinting, we build a multi-stage classifier which combines CNN and multi-class decision tree for accurate and efficient device identification. The input of the clas-

TABLE II
THE IDENTIFICATION RATE FOR DIFFERENT TARGET MODELS AND TRAINED DISCRIMINATIVE MODELS.

Model	Target Model		Discriminative Model	
	Training Dataset	Test Dataset	Training Dataset	Test Dataset
Random Forest	97.62%	97.32%	96.36%	97.10%
Decision Tree	92.20%	93.25%	92.12%	90.89%
SVM	97.89%	96.58%	96.50%	95.52%
k-NN	93.11%	92.33%	90.12%	91.56%
Neural Networks	98.56%	98.86%	98.00%	97.92%

sifier is the profiled features and the channel state information estimated at the receiver, while the output is the mapping results of different IoT devices (i.e., a possibility associated with each specific device is generated to indicate how likely the input belongs to the device). In particular, the CNN is used to learn the subtle differences between different IoT devices that cannot be fully captured by the features extracted in feature profiling. The multi-class decision tree breaks down the whole dataset into different levels and make decisions step by step, in that way we only travel through the branches with high possibility, reducing the unnecessary searching space.

Unlike traditional link signature based identification [22], [23], where a location-specific metric is extracted from radio channels to localize devices at different positions, the proposed approach considers both internal hardware imperfection and external location-specific features to distinguish devices with different identities (e.g., types, locations). In addition, the deep learning based classifier is expected to better fuse different features and achieve a higher accuracy.

Our experiment shows that the propose Device Profiling can achieve fast and accurate IoT device identification even in the presence of the manipulated traffic flows.

VII. Experimental Evaluations

In this section, we evaluate the attack performance of IoTGAN and the defense effectiveness of Device Profiling.

A. Experimental Setup

The dataset used in evaluation is obtained from UNSW IoT Trace Data [24]. The dataset is collected in real-world environment deployed with 28 different IoT devices. These IoT devices include smart cameras (e.g., Samsung SamrtCam, Belkin camera, TP-Link Cloud camera, and Ring door bell), smart hubs (e.g., Amazon Echo, Samsung Smart Things), smart switches (e.g., Belkin motion detector, and TP-Link smart plug) and healthcare devices (e.g., Withings Smart scale, Blipcare blood pressure meter). In the evaluation, we split the IoT trace dataset into two parts. The first part contains 80% of the data as the training dataset, while the remaining 20% is used as the test dataset. In order to validate the performance of IoTGAN, we adopt several typical machine learning based IoT device identification models as the target black-box models, including Random Forest [25], Decision Trees [5], SVM [5], k-NN [7], and Neural Networks [13].

B. The Evaluation of IoTGAN

We following the references to implement the five typical machine learning based IoT device identification models. Then we use the UNSW IoT Trace Data to train and test these models. We use a metric named identification rate to evaluate the performance of IoTGAN, it is defined as

$$\text{Identification Rate} = \frac{\text{Correct Identification Count}}{\text{Total Identification Count}}.$$

The evaluation results for the five target machine learning based IoT device identification models without the attack of IoTGAN is shown in the left two columns of Table II. We can observe that all these five typical identification models can achieve a high identification rate larger than 92% for IoT device identification. For the Neural Networks based model, the identification rate can even achieve 98.86%.

The evaluation results of Table II indicate that the five selected typical identification models are able to successfully identify IoT devices when there is no malicious interference. Therefore, they can work as good baselines to evaluate the effectiveness of IoTGAN. If their identification rates are significantly dropped under the attack of IoTGAN, our proposed attack strategy will be proved effective in camouflaging IoT devices.

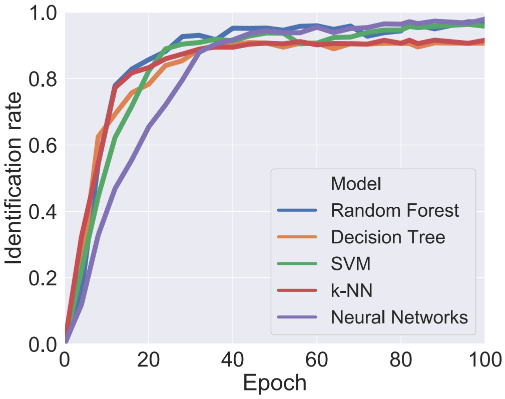

Fig. 3. The identification rates for the discriminative model.

1) The Evaluation of Discriminative Model: We train discriminative models and manipulative models for all five target models with the UNSW IoT Trace Data. The same training and test datasets are used to obtain the discriminative models. Since discriminative models are the substitutes of the target identifiers, they should have the similar identification rate. The right two columns of Table II demonstrate the performance of the discriminative models. As shown, our substitute models can achieve similar identification rate as the target ones. The results indicate that our proposed training process can effectively learn the target identifiers and generate the same identification results.

Figure 3 shows the identification rates of the discriminative models as the number of epochs increases. As shown, the discriminative models for all target identifiers can reach the stability after 40 epochs and approach an identification rate higher than 90%. In particular, the discriminative model for random forest identifier can achieve a high identification rate of 92.10% after 40 epochs.

2) The Evaluation of Manipulative Model: We then use IoTGAN to attack these target models and calculate the new identification rate for both the training dataset and test dataset. We first mount the attack of the device misidentification to reduce the identification rate of the target model. The evaluation results are shown in Table III. We can observe that IoTGAN is able to decrease the identification rates to almost zero for all five target models. This indicates that IoTGAN can successfully manipulate the IoT traffic to evade machine learning based IoT device identification.

TABLE III
THE IDENTIFICATION RATE FOR DIFFERENT MACHINE LEARNING BASED IOT DEVICES IDENTIFICATION MODELS UNDER THE ATTACK OF IOTGAN.

Model	Training Dataset	Test Dataset
Random Forest	0.15%	0.13%
Decision Tree	0.05%	0.06%
SVM	0.11%	0.08%
k-NN	0.21%	0.23%
Neural Networks	0.08%	0.06%

Next, we launch the attack of identity spoofing to camouflage IoT devices with specified types. In UNSW IoT Trace Data, we have four main types of IoT devices (i.e., smart cameras, smart hubs, smart switches and healthcare devices.). We conduct an experiment of the identify spoofing among all these types of IoT devices and evaluate their performance. In particular, we define a metric named Spoofing Rate that indicates the successful rate of the spoofing attack. It is described as the ratio between the count of successful spoofing identifications and the total count of identifications,

$$\text{Spoofing Rate} = \frac{\text{Successful Spoofing Identification Count}}{\text{Total Identification Count}}.$$

Table IV demonstrates the performance of the identify spoofing. As shown, we can achieve more than 90% successful rate when spoofing the IoT identify between **smart camera** ⇔ **smart hub**; **smart camera** ⇔ **healthcare device**; **smart hub** ⇔ **healthcare device**; **smart switch** ⇔ **healthcare device**. Such results indicate that the attack can effectively hide the identity of IoT devices and designate them new specified ones. However, we also find two exceptions (i.e., **smart camera** ⇔ **switch**, **smart hub** ⇒ **switch**) that can only achieve the spoofing rate around 70%. This may because the traffic pattern of the smart camera is quite different from the smart switch that can hardly be spoofed. Specifically, the smart camera usually enables a real-time video transmission that demands a high bandwidth and privileged wireless channel. On the other hand, the data from the smart switch is relatively static and sparse that only occupies a very limited bandwidth. Due to the nature difference of the traffic pattern between the camera and switch, it's hardly to hide some trivial features without affecting its functionalities.

C. The Evaluation of Device Profiling

We implement the Device Profiling following the design in Section VI. It mainly includes two components: the feature profiling, which uses the nonlinear characteristics of the transmit RF signals for IoT device profiling; and the device fingerprinting, which build a neural network based multi-stage classifier to learn the feature patterns for accurate and efficient device identification even when the traffic based features are manipulated. The system is built with Intel WiFi Wireless Link 5300 802.11n MIMO radio tool [26] to collect data. Device Profiling is deployed in a real-world environment with different IoT devices (e.g., Eufy security camera, Eufy smart lock, and Ecobee thermostat) for evaluation. We then use IoTGAN to attack this Device Profiling system. The evaluation results is shown in Figure 4

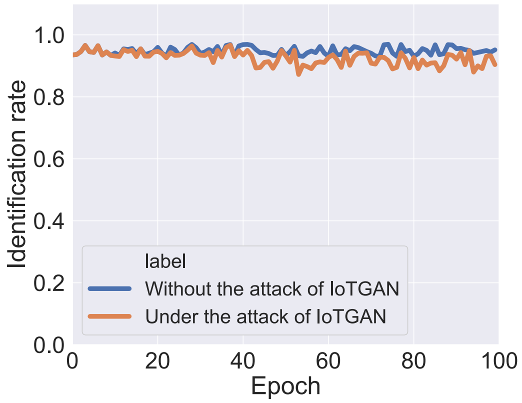

Fig. 4. The identification rates for Device Profiling in the scenario: (1) without the attack of IoTGAN, and (2) under the attack of IoTGAN.

From Figure 4, we can see when there is no attack of IoTGAN, Device Profiling can achieve an identification rate around 98%. When IoTGAN is deployed to launch attacks, there is nearly no effect on the identification rate of Device Profiling when the epoch of training the manipulative model is less than 30. This indicates that the generative model needs approximate 30 epochs of training to approach the stable state. After that, the Device Profiling can still maintain an identification rate around 95%. Therefore, we can conclude

TABLE IV

THE SPOOFING RATE FOR DIFFERENT TYPES OF IDENTITY SPOOFING ATTACK (**camera** ⇔ **hub** INDICATES THE IDENTIFY SPOOFING BETWEEN THE SMART CAMERAS AND SMART HUBS. **camera** ⇒ **hub** INDICATES THE ATTACKER AIMS TO CAMOUFLAGE THE IDENTIFY OF SMART CAMERA TO THE SMART HUB.)

Identity Spoofing	**camera ⇔ hub**		**camera ⇔ health**		**camera ⇔ switch**		**hub ⇔ health**		**hub ⇔ switch**		**switch ⇔ health**	
Model	⇐	⇒	⇐	⇒	⇐	⇒	⇐	⇒	⇐	⇒	⇐	⇒
Random Forest	92.11%	91.59%	85.12%	89.13%	77.12%	68.31%	91.13%	90.12%	91.87%	75.63%	90.34%	91.31%
Decision Tree	93.19%	92.21%	90.23%	89.35%	70.79%	69.21%	89.23%	92.49%	92.85%	71.25%	91.13%	92.95%
SVM	89.17%	91.75%	89.72%	91.89%	72.65%	70.12%	92.54%	92.13%	90.15%	69.92%	93.56%	92.79%
k-NN	91.57%	91.12%	88.21%	90.92%	68.15%	69.78%	91.45%	90.68%	89.74%	70.54%	91.51%	90.36%
Neural Networks	93.67%	93.98%	89.10%	87.88%	73.64%	71.23%	91.65%	93.68%	91.20%	71.99%	92.55%	93.79%

that Device Profiling can successfully defend against the attacks launched by IoTGAN.

VIII. CONCLUSIONS

In this paper, we investigate the security of machine learning based IoT device identification methods. We propose a novel attack strategy named IoTGAN to manipulate the IoT devices' traffic such that it can evade machine learning based IoT device identification. In IoTGAN, a substitute model is designed to fit the target identification model in a black-box setting, and a manipulative model is trained to add adversarial perturbation into IoT devices' traffic to evade the substitute model. Experimental results show that IoTGAN can effectively disrupt device identification. We also develop countermeasures complementary to existing methods to further protect machine learning based IoT device identification.

ACKNOWLEDGMENT

The work at USF was supported in part by NSF CNS-1553304 and ECCS-2029875.

REFERENCES

[1] "Internet of things," https://en.wikipedia.org/wiki/Internet_of_things, 2021.

[2] Y. Chen, W. Trappe, and R. P. Martin, "Detecting and localizing wireless spoofing attacks," in *IEEE Communications Society Conference on sensor, mesh and ad hoc communications and networks*, 2007.

[3] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Generative adversarial network in the air: Deep adversarial learning for wireless signal spoofing," *IEEE Transactions on Cognitive Communications and Networking*, 2020.

[4] N. Ammar, L. Noirie, and S. Tixeuil, "Network-protocol-based iot device identification," in *IEEE FMEC*, 2019.

[5] N. Yousefnezhad, A. Malhi, and K. Främling, "Automated iot device identification based on full packet information using real-time network traffic," *Sensors*.

[6] A. Sivanathan, D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Characterizing and classifying iot traffic in smart cities and campuses," in *IEEE Conference on Computer Communications Workshops (IEEE INFOCOM WKSHPS)*, 2017.

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

[8] A. Aksoy and M. H. Gunes, "Automated iot device identification using network traffic," in *IEEE International Conference on Communications (ICC)*, 2019.

[9] T. Hou, Z. Qu, T. Wang, Z. Lu, and Y. Liu, "Proto: Proactive topology obfuscation against adversarial network topology inference," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1598–1607.

[10] T. Hou, T. Wang, Z. Lu, and Y. Liu, "Combating adversarial network topology inference by proactive topology obfuscation," *IEEE/ACM Transactions on Networking*, 2021.

[11] A. Sivanathan, H. Habibi Gharakheili, and V. Sivaraman, "Managing iot cyber-security using programmable telemetry and machine learning," *IEEE TNSM*, 2020.

[12] J. Kotak and Y. Elovici, "Iot device identification using deep learning," in *Computational Intelligence in Security for Information Systems Conference*. Springer, 2019.

[13] J. Bao, B. Hamdaoui, and W.-K. Wong, "Iot device type identification using hybrid deep learning approach for increased iot security," in *International Wireless Communications and Mobile Computing*, 2020.

[14] S. Leroux, S. Bohez, P.-J. Maenhaut, N. Meheus, P. Simoens, and B. Dhoedt, "Fingerprinting encrypted network traffic types using machine learning," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018.

[15] T. Hou, T. Wang, Z. Lu, and Y. Liu, "Smart spying via deep learning: inferring your activities from encrypted wireless traffic," in *IEEE GlobalSIP*, 2019.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, 2014.

[17] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[18] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[19] Y. Sagduyu, Y. Shi, and T. Erpek, "Adversarial deep learning for over-the-air spectrum poisoning attacks," *IEEE Transactions on Mobile Computing*, 2019.

[20] T. Wang, Y. Liu, Q. Pei, and T. Hou, "Location-restricted services access control leveraging pinpoint waveforming," in *ACM Conference on Computer and Communications Security*, 2015.

[21] T. Wang, Y. Liu, T. Hou, Q. Pei, and S. Fang, "Signal entanglement based pinpoint waveforming for location-restricted service access control," *IEEE Transactions on Dependable and Secure Computing*, 2016.

[22] Y. Liu and P. Ning, "Enhanced wireless channel authentication using time-synched link signature," in *IEEE INFOCOM*, 2012.

[23] N. Patwari and S. K. Kasera, "Temporal link signature measurements for location distinction," *IEEE Transactions on Mobile Computing*, vol. 10, no. 3, pp. 449–462, 2011.

[24] 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 TMC*, 2018.

[25] R. K. Singh, S. Dalal, V. K. Chauhan, and D. Kumar, "Optimization of far in intrusion detection system by using random forest algorithm," in *International Conference on Advanced Computing and Software Engineering*, 2019.

[26] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11 n traces with channel state information," *ACM Computer Communication Review*, 2011.