# GUIDE: Graphical User Interface Fingerprints Physical Devices

Qiang Li [*], Xuan Feng [†], Zhi Li [†], Haining Wang [‡], Limin Sun[†]
[*] School of Computer and Information Technology, Beijing Jiaotong University, China
[†] Institute of Information Engineering, Chinese Academy of Sciences, China
[‡] Department of Electrical and Computer Engineering, University of Delaware, USA

***Abstract*—Nowadays, the number of visible physical devices exposed on the Internet is dynamically increasing and they play a crucial role for bridging between the cyber space and the physical world, such as network printer, Webcam, and industrial control devices. Discovering these devices brings about the deep understanding on these devices' characteristics and help secure device security in the cyber space. A device fingerprint is a prerequisite of device discovery in the Internet. However, today's online device search depends on keywords of packet head fields and the keyword collection is done manually. This impedes an accurate and large-scale device discovery, due to high human efforts and inevitable human errors, as well as the difficulty of keeping keywords complete and updated. To address this problem, we propose *GUIDE*, a framework to automatically generate device fingerprints based on webpages embedded in these devices. In order to demonstrate how *GUIDE* works, we also develop its prototype system and provide a case study which discover surveillance devices in the cyber space.**

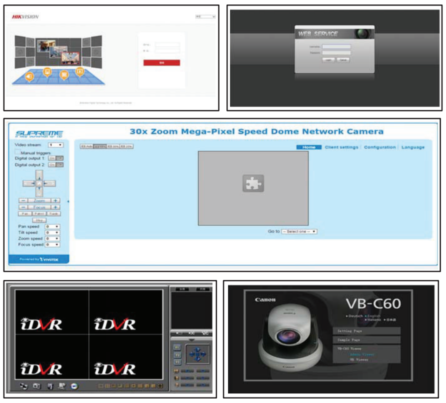

Fig. 1: Webpage appearances of embedded devices.

## I. Introduction

We are on the brink of a new era in the development of Internet of Things (IoT), where the number of physical devices with computing and communication capabilities is rising rapidly. Those devices are crucial in the physical world, such as routers, net-printers, webcams, switches, bridges and work stations.

Discovering these devices online exposure brings about many benefits. One advantage is to help system administrators with security auditing, reveal new kinds of vulnerabilities and preserve device security in the cyber-space. Another is to shed light on distributed online devices and network measurement statistics. However, today's commercial search engine, Shodan [1], and academic engine, Censys [2], discover these Internet-connected devices using keywords by manual collection. They send application-layer requests to online devices and recognize them by comparing the field values with pre-defined keywords. This method is currently a manual process which is arduous, incomplete, and makes it difficult to keep up-to-date with the numerous new devices and new version updates.

In this paper, we have been motivated by an intuitive observation that many embedded devices have a webpage which used to access and configure device conveniently. There are two characteristics for embedded webpages: *invariant* and *distinct*. *Invariant* characteristic indicates that webpages are written into the firmware and keep stable along with time. *Distinct* characteristic is that webpage appearance is capable of acting as a signature of device, distinguishing from other devices in the cyber space. Figure 1 shows an example, that webpage appearances play a distinct role to identify physical devices. In this paper, we propose *GUIDE*, a framework that automatically generate fingerprints based on webpages and use it to discover online embedded devices.

*GUIDE* use HTML parser to extract the primitive data from the webpages and natural language processing to extract content. After pre-processing, *GUIDE* proposes a feature selection algorithm to find general features for embedded devices . A supervised learning model in *GUIDE* trains a classification model for recognize device type. In order to demonstrate how *GUIDE* works, we also develop its prototype system and provide a case study that discover surveillance devices in the cyber space.

## II. GUIDE: Design and Implementation

In this section, we explain how *GUIDE* works and describe the details of implementation.

### A. Design

The overall architecture of *GUIDE* is shown in the Figure 2. There are three parts: the pre-processing module, the

[†] Zhi Li is corresponding author.

978-1-5090-3281-5/16/$31.00 ©2016 IEEE

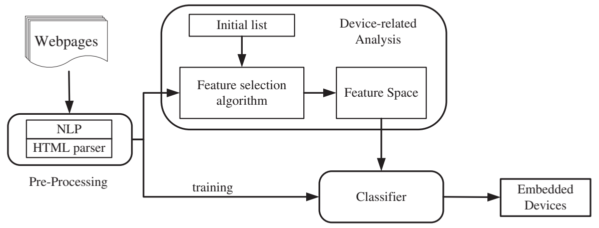

Fig. 2: The framework of *GUIDE*

device-analysis module, and the classifier module. In the pre-processing module, the input is webpages and the output is the content list for the next two stages. We use the HTML (XML) documents parser to extract the primitive data from the webpages. Then natural language processing (NLP) is used to finish jobs like word splitting, stemming and redundant content removal. Each webpage would be transformed into a content list. In the device-related analysis module, we extract general features for physical devices. We manually pick up a small initial set of words and it can be automatically expanded via a specific feature selection algorithm. The device-related modular would generate a general list for the classifier of devices. The expanded features are used as feature space for every webpages of physical devices. In the classifier module, we use a supervised machine learning approach to train a classifier based on a set of training data. Each webpage is transformed into a feature vector in the pre-processing module, and the training set is represented as training matrix, where columns are feature spaces and rows are the number of training data. It is noted that we should train the the classification model which is according with the device. If network router is the device need to be identified, the device-related modular focuses on the routers and the classifier uses the training data of routers. Based on the *GUIDE*, for any given webpages, our approach can tell whether it is that type of device.

### *B. Implementation*

We have implemented a prototype system as a self contained piece of software based on open source libraries. Our core classification processing and scrapping algorithms are mainly written in Python and go language. We use the open-source Python library "BeautifulSoup" to extract the content of each webpage. We extract a primitive word list from each webpages and use the Natural Language Toolkit (NLTK) to acquire the content list. Every webpage is transformed into a feature vector for classification. We use the open-source skit-learn with SVM classifier to train a classification model. The classifier determine whether the webpage is an embedded device. We validate the framework of automatical fingerprint generation on a Cloud computing server, Amazon EC2.

## III. A case study

Based on *GUIDE*, we validate its effectiveness through generating fingerprints of surveillance devices. The surveillance device is typical embedded device, it is a type of digital video device commonly employed for monitoring surrounding environments, e.g. Webcam, Network Virtual Record (NVR) and Digital Virtual Record (DVR). Surveillance devices play a vital important rule for bridging between the cyber space and the physical world.

TABLE I: The overall accuracy for four classification models.

| Classifier | PR | RC | F1-mean |
|---|---|---|---|
| SVM | 99.00 | 95.22 | 96.98 |
| KNN | 99.06 | 96.65 | 97.84 |
| Naive Bayes | 91.18 | 88.75 | 89.95 |
| Decision Tree | 98.86 | 87.94 | 93.09 |

To validate our proposed method, we have collected the dataset of surveillance devices in the cyber space which included 42000 webpages. We manually label these webpages via watching their graphical interface in the browser. If it is surveillance device, we labeled it as a "Surveillance" tag, otherwise as "Non-surveillance" tag. We divide the datesets into two parts, a 20,000 size part for training and a 22,319 size part for test process.

We choose four type typical classification models, respectively, Support vector machine (SVM), k-nearest neighbors (KNN), Multinomial Naive Bayes, and Decision Tree. They are typical supervised learning algorithm in pattern recognition. We also use the choose $Chi^2$ as feature selection and rank top 100 as our feature space. Table. I shows their performance amongst these four classification models. The result show that SVM and KNN models have the best performance, and the Naive Bayes performs worst. It is obvious that automatically generating fingerprint via the *GUIDE* has achieved a promising result. *GUIDE* could be also used to identify other physical devices, such as industrial control systems [3]

## IV. Conclusion

In this paper, we proposed propose *GUIDE*, a novel framework for automatic and accurate identifying physical devices. The core of our approach is to use a webpage embedded in a device as its fingerprint. We used natural language processing and machine learning to generate a device fingerprint automatically. Furthermore, we use a case study to demonstrate the *GUIDE*. The experimental results show that *GUIDE* can achieve 96% recall and 98% precision in the classification.

## V. Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities (K15RC00060) and the National Defense Basic research Program of China (No. JCKY2016602B001).

## References

[1] The search engine for internet-connected devices. [Online]. Available: https://www.shodan.io/

[2] Censys, the internet-wide search engine. [Online]. Available: https://censys.io/

[3] X. Feng, Q. Li, H. Wang, and L. Sun, "Characterizing industrial control system devices on the internet," *24th IEEE International Conference on Network Protocols (ICNP 2016)*, 2016.