Automated Hidden Sensor Detection in Sensor-Rich Spaces

Brent Lagesse*, Kevin Wu†, Jaynie Shorb‡ and Zealous Zhu§
Computing and Software Systems
University of Washington Bothell
Bothell, WA USA
Email: *lagesse@uw.edu, †kevinw9@uw.edu, ‡jaynies@uw.edu, §zealousz@uw.edu

***Abstract*—The emerging ubiquity of IoT devices with monitoring capabilities has resulted in a growing concern for user privacy. Our work focuses on detecting IoT devices that are being used to spy on users' activity. We have developed a framework for affecting the physical environment to induce a corresponding signal in the digital environment in order to determine what devices are observing our activity.**

We will demonstrate an implementation of this framework focused on video streaming. We will demonstrate that our approach produces both high true positives and low false positives despite a variety of changing environmental factors such as background movement and placement of the IoT device. Our demo is implemented on a stock Android phone and only requires a second wireless adapter if the phone is not rooted.

I. Introduction

The manufacturing and deployment of Internet of Things (IoT) devices is rapidly growing and introducing more pervasive and useful services to users. Unfortunately, these devices raise privacy concerns since they may be placed without a person's knowledge[1]. In addition to maliciously placed devices, devices might be used by an attacker who has compromised the device[2].

Consider the following scenarios. In the first case, an attacker has placed a hidden web cam in a room that you are visiting. Such a web cam is designed to blend in with the surroundings and can easily occupy less than 5 cm^2 of surface area. The camera can operate on an encrypted channel that cannot be directly read by the user, so its presence is not easily distinguishable from the dozens of other Wi-Fi devices that are frequently visible in many houses, hotels, and apartments. In the second case, the many camera-enabled devices that surround us every day such as web cams, TVs, laptops, and cell phones are potential sources of spying. Such devices stream information about the user to remote sources because the user does not understand the functionality provided by the manufacturer or application. In the third case, an attacker has exploited a vulnerability and compromised the device [3]. In this case, as in the first, it is difficult for the user to detect that a device has been compromised, and prior work in the area of detecting hidden cameras focuses on detecting unknown hidden cameras rather than maliciously manipulated cameras that the user knows are present.

Previous research in privacy in IoT spaces has generally focused on protecting the user from an attacker that discovers their activities [4]. The focus of our work examines an attacker model where attacker attempts to record information about a user without the user's knowledge. In this case, the user wants to identify that they are being monitored.

Previous work in detecting devices that are surreptitiously recording user activity has largely focused on cameras. In particular, this work [5] focuses on identifying IR used in night vision or the presence of unexplained Wi-Fi devices in order to determine locations to manually search for cameras.

Our work augments these existing approaches and provides a framework that is more widely applicable to IoT devices that could be used to surveil users. To this end, our work contributes the following advancements:

- A framework for manipulating the physical environment to induce detectable signals in the digital environment
- An implementation of this framework for Wi-Fi streaming cameras
- A computationally efficient system that runs on a mobile device with minimal additional hardware
- An analysis of the parameter space of the system to demonstrate its efficacy for detecting Wi-Fi streaming cameras

Additionally, our system addresses parts of the problem that previous approaches do not address. Previous work helps to identify the presence of hidden cameras, but not their status of actively streaming video of the user. Previous work relies on the hidden cameras being in night vision mode to detect RF signals. Furthermore, techniques that rely on intercepting the data and reconstructing it are not effective if the camera is using an encrypted connection that the user does not have the key to.

II. Design

A. System Model

Our work is focused on pervasive IoT systems that primarily operate indoors in houses and offices. These systems contain large numbers of resource constrained devices with rich abilities to sense and distribute information with some capability of processing information. These devices are typically wirelessly connected. In particular, this system affects our design considerations in that there are limitations on the amount of data that can be stored on the individual devices, so we expect that wireless data streaming will be prevalent.

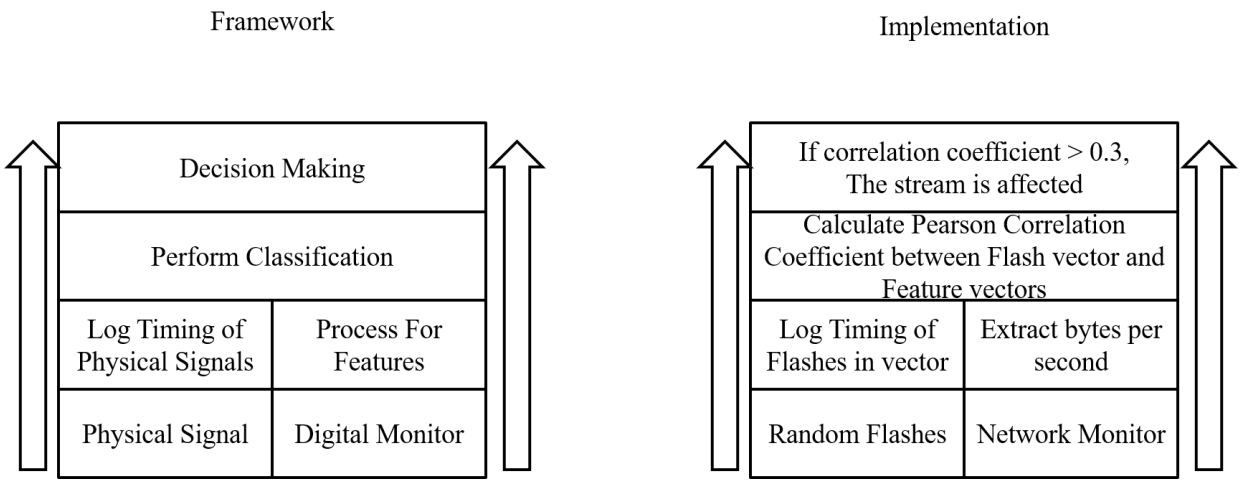

Fig. 1. Framework and Implementation

B. Attacker Model

Our approach addresses three different attacker models. We refer to these models as the hidden device model, the compromised device model, and the unexpected recording model. Our system is useful in detecting attackers utilizing any of the three models.

1) Hidden Device Model: In the hidden device model, an attacker has hidden a monitoring device in the space in which a user will occupy. Examples of this may include a short-term rental [1] or could be a compromise of a person's own home. In this model, the user is not initially aware that the device exists. This is the model that most previous work [5] focuses on detecting.

2) Compromised Device Model: In the compromised device model, an attacker utilizes a user's own device to record their actions. This model has been documented[2] as a problem with IoT devices.

3) Unexpected Recording Model: In the unexpected recording model, the attacker is the device or application manufacturer. This model has been acknowledged multiple times as companies are frequently recording information from IoT devices for either improving functionality or targeting products and advertising [6].

C. System Design

Figure 1 shows our generic framework and the specific implementation used in our demonstration. Our design consists of a physical and a digital side. In the physical side, a signal is induced that is designed to cause IoT devices to respond digitally. On the digital side, our system records digital outputs that occur during this time. The digital output is processed to extract features necessary for classification, and then the features and the information about the physical signal are processed by a classification algorithm (in our implementation, we only record whether or not the physical signal was active; however, other information such as intensity or frequency of the signal could also be recorded). The output of the classification algorithm is then used to decide which streams were recording the user, if any.

III. Prototype Implementation and Demonstration

In the implementation we present in this demo, we use the flash on a camera and log a value of 1 in a vector if a flash occurred in a particular second and 0 if a flash did not occur. Simultaneously, we record network traffic in promiscuous mode and create a vector of the number of bytes transmitted each second for each MAC address. We then calculated the Pearson correlation coefficient between the flash vector and the bandwidth vectors. If the correlation coefficient is greater than 0.3 (this value was chosen experimentally as it optimizes true positive vs false positive rates), we classify the MAC address as recording the subject. Since we only record bandwidth, we are able to perform this calculation on encrypted data. Likewise, we also do not need to store the data recorded for more than one second, so we do not consume a significant amount of memory. Currently, this approach results in a true positive rate of about 90% and a false positive rate of approximately 5% as shown in figure 3.

We implemented our system on several Android phones including a Moto Z running Android version 6.0.2, and an HTC M7 running Android 5.0.2. The devices are connected with a Sabrent NT-WGHU Wireless adapter based on the 8187L chipset via USB OTG cable to capture the Wi-Fi network traffic. The Wireless adapter allowed us to capture any traffic in promiscuous mode, which unrooted Android does not allow us to. Any network traffic will be saved in PCAP files that are generated by the Kismet Wi-Fi PCAP capture 2012.12.1. The Sensor Detection App we have implemented will further read those PCAP files and calculate their Pearson correlation coefficient between flashes and the bandwidth to classify streams into spying and non-spying devices.

A. Showcase Plan

In the demo session, we plan to demonstrate our automated hidden sensor detection system with multiple sensors carrying

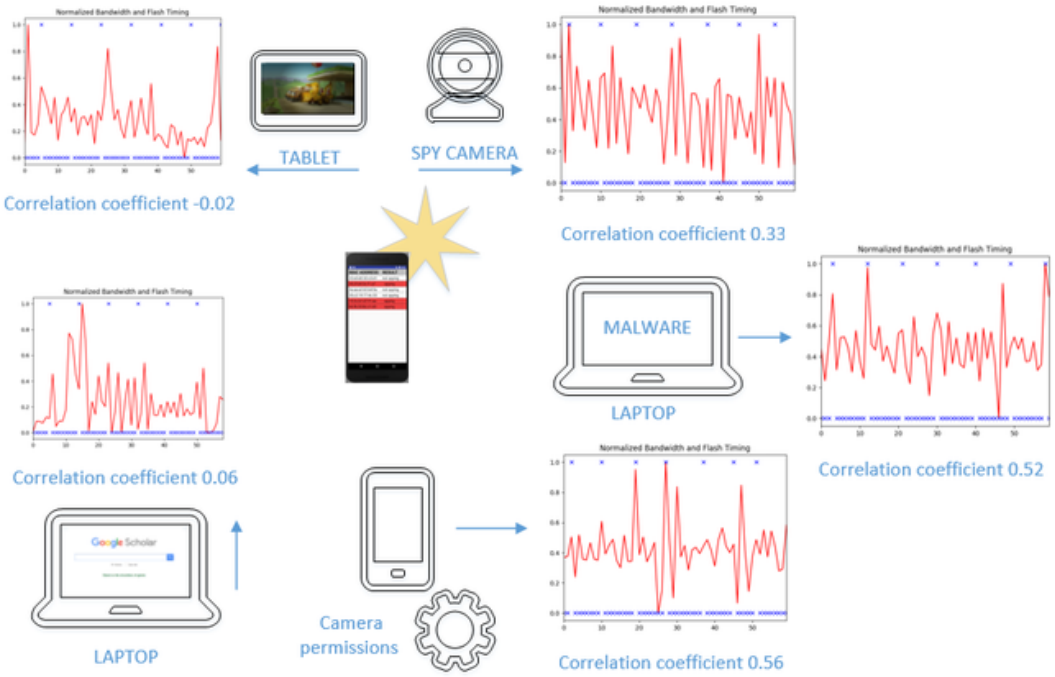

Fig. 2. Demo Setup. The user holds the phone in the middle of a room full of devices and it flashes while recording network traffic. Classification occurs on the phone and the phone outputs a list of MAC addresses and highlights those that are believed to be spying.

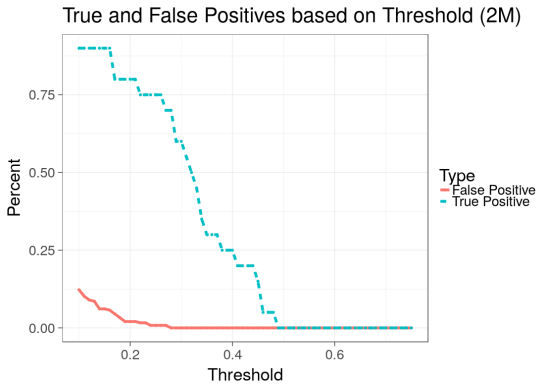

Fig. 3. True and False Positives as Function of Threshold Acceptance

the three attacker models we have discussed earlier (note that we will only be simulating malware and not running actual malware) and as shown in figure 2. The Android based mobile phone will capture the Wi-Fi network traffic while manipulating the physical environment with flashlights. The system will further calculate confidence level of each captured network streams and present the results onto the front-end of the mobile phone. We will bring our own hardware and will required steady Wi-Fi signal to carry out the demo. This demo will also require an area which we can placed at least six sensors in a distance of two meters away from the Android mobile phone.

IV. Conclusion

The main idea we want visitors to our demo to take away is that it is possible affect the digital world with actions in the physical world and detect hidden IoT devices that are violating a users' privacy with devices that are commonly owned by the users. We have demonstrated the feasibility of manipulating the physical world in a way that produces a digital footprint that can be detected by eavesdropping on wireless communications, even if those communications are encrypted and we do not have permission to join the network. We have demonstrated that we can accomplish this without the burden of expensive or cumbersome additional hardware.

References

[1] "Yvonne Edith Maria Schumacher vs Airbnb, Inc., a foreign corporation, and Fariah Hassim and Jamil Jiva." [Online]. Available: https://cdn2.vox-cdn.com/uploads/chorus_asset/file/5398067/1-main.0.pdf

[2] Y. M. Pa Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: Analysing the Rise of IoT Compromises | USENIX." Usenix, 2015. [Online]. Available: https://www.usenix.org/conference/woot15/workshop-program/presentation/pa

[3] R. M. Ogunnaike and B. Lagesse, "Toward consumer-friendly security in smart environments," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2017, pp. 612–617.

[4] N. Apthorpe, D. Reisman, and N. Feamster, "A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic," *arXiv:1705.06805 [cs]*, May 2017, arXiv: 1705.06805. [Online]. Available: http://arxiv.org/abs/1705.06805

[5] P. Polstra, "Am I Being Spied On? Low-tech Ways Of Detecting High-tech Surveillance," Las Vegas, NV, Aug. 2014.

[6] S. Burke, "Google admits its new smart speaker was eavesdropping on users," Oct. 2017. [Online]. Available: http://money.cnn.com/2017/10/11/technology/google-home-mini-security-flaw/index.html