IoT Network Monitor

IoT Home Inspector Challenge Submission

Wildcard Submission

Prepared by: Rohan Doshi, Gudrun Jonsdottir, Daniel Wood
Advised by: Nick Feamster, Noah Apthorpe, and Dillon Reisman
Center for Information Technology Policy (CITP)
Princeton University

May 19, 2017

ABSTRACT

We present a wildcard submission to the FTC's IoT Home Inspector Competition: IoT Network Monitor. IoT Network Monitor is an intuitive and user-friendly interface for consumers to visualize vulnerabilities of their smart home IoT devices. Running on a Raspberry Pi v3 which is configured as a WiFi access point, IoT Network Monitor analyzes the traffic of connected devices in three ways. First, it detects devices with default passwords exploited by previous attacks such as the Mirai botnet, changes the passwords to randomly generated 12 character strings, and reports the new passwords to the user. Second, it conducts deep packet analysis on the data stream of each device and notifies the user if potentially sensitive personal information is being transmitted without encryption. Last, it detects denial-of-service (DoS) traffic originating from IoT devices connected to the network and instructs the user to disconnect devices that have been hacked. The IoT Network Monitor will allow homeowners to understand the security of their home networks and learn what actions are appropriate when security vulnerabilities are detected. Adoption of this tool will make consumers' homes more secure and protect the global Internet against threats from compromised IoT devices.

CONTENTS

MOTIVATION

When consumers purchase IoT devices for their home, whether an Amazon Echo, wireless smart scale, or Nest thermostat, they currently have no visibility into the traffic that their devices are sending across the Internet or how vulnerable their devices are to attack. IoT devices have access to some of our most personal and private information, including financial data, location, health, and behavior—data that users want to prevent from being shared with adversaries and network observers. The only visibility consumers might have into how a device respects privacy is from smartphone or web apps associated with their devices; they don't have the time or capability to closely scrutinize their outgoing data with network analysis tools like Wireshark. We therefore present IoT Network Monitor: an intuitive and user-friendly WiFi hub that visualizes the security and privacy vulnerabilities of IoT devices in a consumer's home through an easy-to-use interface.

FRAMEWORK

Home IoT devices are usually connected to home networks behind a firewall or network address translator. This allows the devices to directly communicate with other hosts on the home network and can pose a variety of security threats. For example, if an IoT device has open ports or insecure passwords, it can be hijacked, reveal sensitive information, conduct denial of service attacks on the local network, join an IoT botnet, or do any number of harmful things. However, consumers don't have a simple, intuitive interface to understand and monitor these threats. Even if they are aware of a security attack, consumers often are typically unsure what action to take in response. Our IoT Network Monitor is designed to combat these issues.

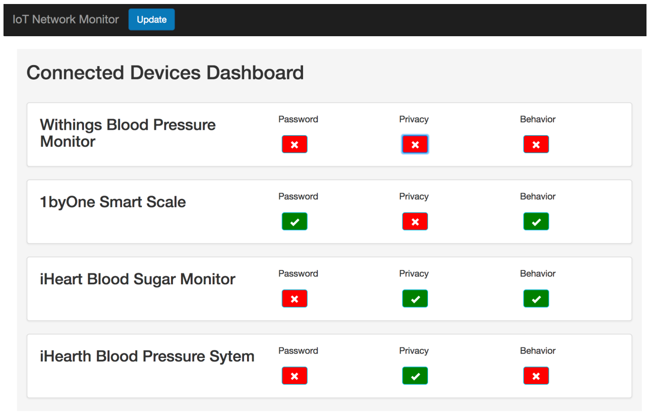
The IoT Network Monitor includes a user-friendly web interface that integrates with a network middlebox that can sniff traffic on the LAN. The IoT Network Monitor collects packet captures as PCAP files throughout the day (e.g., for the first ten minutes of each hour) and performs the following analyses to identify 3 particularly relevant security and privacy vulnerabilities:

1. **Default Passwords**: Scan LAN for IoT devices with open ports 22 (SSH) and/or 23 (Telnet), attempt login with well-known default passwords, create a new password if necessary.
2. **Unencrypted Private Information**: Check for unencrypted packet contents containing personal identifiers and other forms of sensitive data (e.g. medical data).
3. **Anomalous Network Traffic**: Detect DoS traffic originating from an IoT device on the LAN using the FITBOT framework.

IMPLEMENTATION

The IoT Network Monitor is designed to be deployed on a Raspberry Pi 3, which is automatically configured to act as a WiFi access point. Consumers simply connect their IoT devices to the Raspberry Pi's WiFi network as they normally would any WiFi network. While the consumer uses their devices, the IoT Network Monitor software on the Raspberry Pi will automatically begin its security and privacy analyses on the connected devices. The IoT Network Monitor includes a built-in web app that displays the results of these analyses in a user-friendly format. All code and documentation for the IoT Network Monitor can be found on Github at https://github.com/RohanDoshi2018/IoTSecurityHub.

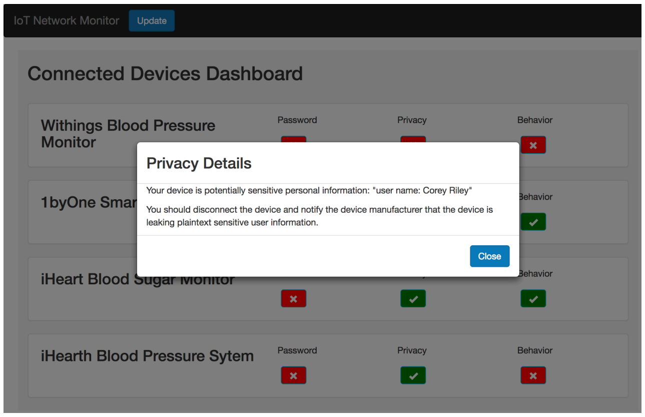
The IoT Network Monitor web interface notifies users of security and privacy threats with clear colors and relevant text (e.g., sample of leaked sensitive information or a proactive suggestions for a new password). The interface also provides straightforward instructions to help users understand what to do in response to a threat.



Connected Devices Dashboard

	Password	Privacy	Behavior
Withings Blood Pressure Monitor	✗	✗	✗
1byOne Smart Scale	✓	✗	✓
iHeart Blood Sugar Monitor	✗	✓	✓
iHearth Blood Pressure Sytem	✗	✓	✗

Figure: Main screen of the dashboard includes state of connected devices



Privacy Details

Your device is potentially sensitive personal information: "user name: Corey Riley"

You should disconnect the device and notify the device manufacturer that the device is leaking plaintext sensitive user information.

Figure: When the user clicks on the status button for a device, details are provided

PASSWORD INSPECTION

The IoT Network Monitor scans the home network for devices that have been shipped with a default username and password combination on an open port, either port 22 (SSH) or port 23 (Telnet). This vulnerability of IoT devices – when multiple devices are shipped with the same default logon credentials – enabled the construction of the Mirai botnet. When such devices are detected on a home network, our tool changes the password of the open port to a randomly generated 12 character string and notifies the user of the new password on the user-interface.

This process involves the following steps:

1. Nmap, a network scanner, is used to scan the home network for all devices listening on ports 22 and 23. It scans a range of 256 private IP-addresses based on the IP address of the WiFi access point running the IoT Network Monitor. This range should be sufficient to capture all devices on a regular home network while still running efficiently.

2. Usernames and passwords from a list of 62 default login credentials included in the Mirai botnet source code are used for login attempts on the devices found by Nmap. If either a SSH or a Telnet connection is successfully established, a default username and password combination for the specific device has been found.

3. Detected default passwords are changed (using the bash *passwd* command) to randomly generated 12-character strings of ASCII-characters and digits. New randomly generated passwords are returned to the user on the web interface of the IoT Network Monitor.

4. If all 62 login attempts to a particular device are unsuccessful, it is assumed that the device does not come with a well-known default password, and the device is labeled secure on the web interface.

PRIVACY ANALYSIS

The IoT Network Monitor intercepts packet data from connected IoT devices and detects potentially sensitive information about the user that is being sent to the Internet unencrypted. If the IoT Network Monitor discovers that a device is leaking personally identifiable information or medical information, it alerts the user of this behavior.

The feature is implemented using a five step approach:

1. **Separate packet streams:** The IoT Network Monitor separates connected devices on the network into individual packet streams for analysis. This can be done by classifying packet conversations by device MAC addresses or the external IP address of the server communicating with the device.

2. **Separate by protocol:** Once a device's packet stream has been identified, the IoT Network Monitor begins the process of deep packet inspection by separating the stream by network protocol. It separates HTTP and TCP packets for further analysis.

3. **Remove headers:** Next, packet payloads are separated from the packet headers so that the IoT Monitor can examine the application data and screen it for plaintext content.

4. **Find unencrypted packets:** Packets are then classified as either plaintext or encrypted using payload entropy as an indicator of encryption. By training the IoT Network Monitor on a test set of payloads, we observed a threshold entropy value below which we were able to consistently identify plaintext packets. Entropy analysis is conducted using the Shannon Entropy Test, which calculates the quantitative measure of the variability in the frequency of the different possible characters in a payload. While encrypted strings have very high entropy, unencrypted plaintext English strings exhibit very low entropy, and are thus fairly easy to identify.

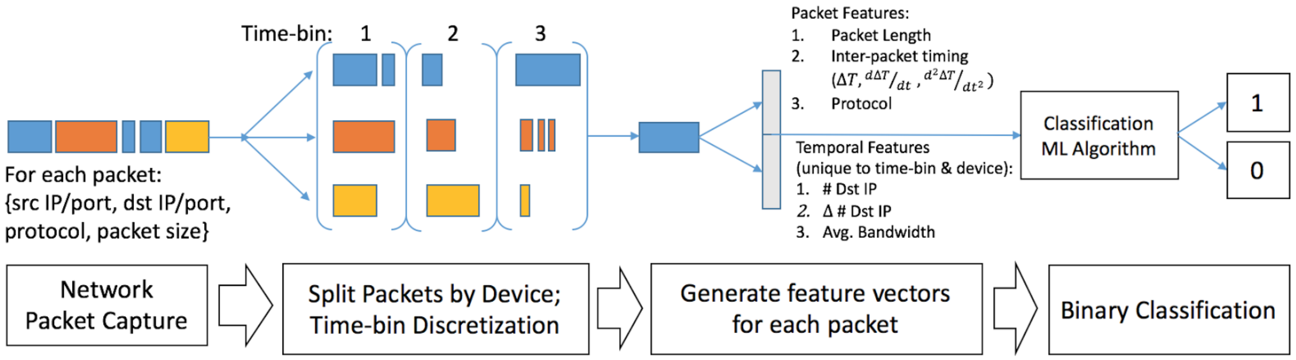
5. **Search for sensitive information:** Lastly, identified plaintext packets are further analyzed for potentially sensitive information. This is implemented by a straightforward dictionary search; plaintext packets are searched for the presence of common first names, medical terms, and personally identifying information. If the monitor discovers instances of these terms in the unencrypted packets coming from an IoT device, it alerts the user on the IoT Network Monitor web interface.

ANOMALY DETECTION

The IoT Network Monitor detects anomalous behavior of any devices connected to the hub. Many IoT devices are fundamentally insecure, exposing the Internet to a variety of attacks. In particular, IoT botnets, like the Mirai botnet, leverage insecure IoT devices to conduct some of the strongest distributed denial of service (DDoS) attacks ever recorded.

This has spurred interest in developing network mechanisms, particularly at the local level, for detecting attack traffic from compromised IoT devices. Despite the growing threat of IoT botnet attacks, little research has been done to adapt these techniques specifically to consumer home networks. The key challenge lies in determining which features capture the unique statistical properties of IoT traffic and help differentiate it from anomalous IoT attack traffic. Our analysis indicates that IoT traffic has many unique behaviors which can be leveraged to enhance machine learning accuracy.

We propose FITBOT (Find IoT Botnets), a novel packet-level machine learning anomaly detection framework for detecting botnet attack traffic in consumer Internet of Things (IoT) device networks. FITBOT leverages IoT-specific device behaviors to improve upon previous anomaly detection algorithms, since IoT devices (e.g. we take advantage of IoT devices' limited number of endpoints, regular internal time intervals between packets, small size packet payloads for attacks, etc.).



Time-bin: 1 2 3
For each packet:
{src IP/port, dst IP/port, protocol, packet size}
Packet Features:
1. Packet Length
2. Inter-packet timing
$(\Delta T, {}^{d\Delta T}/_{dt}, {}^{d^2\Delta T}/_{dt^2})$
3. Protocol
Temporal Features
(unique to time-bin & device):
1. # Dst IP
2. Δ # Dst IP
3. Avg. Bandwidth
Classification ML Algorithm
1
0
Network Packet Capture
Split Packets by Device; Time-bin Discretization
Generate feature vectors for each packet
Binary Classification

Figure: The proposed FITBOT Framework.

The FITBOT framework involves four key steps:

1. **Network Packet Capture:** The IoT Network Monitor samples network traffic and generates a packet capture files (PCAP) that contains a variety of flow-based attributes; no deep-packet inspection is required. For each packet, we record the following flow-based attributes: source IP address, source port, destination IP address, destination port, packet size in bytes, and the time elapsed since the last packet arrived.

2. **Split Packet By Device and Time-bin:** In order to generate temporal features on a per-device basis, the packets must be split by their source IP address. This way, we can quantify and represent whether specific devices are behaving anomalously in a given time period. Using the timestamps of the packet capture, we split packets into continuous non-overlapping time-bins. The length of the time-bin may be modified to capture different types of temporal features.

3. **Generate Feature Vectors for Each Packet**: We generate packet-level features and temporal features for each packet and concatenate all these features into a feature vector. Packet-level features are normally derived almost directly from the flow-based attributes; this includes packet length, inter-packet timing velocity and acceleration, and protocol. And, temporal features are unique to a specific device and time-bin; these tend to be more IoT-device specific, such as the number of distinct destination IP addresses, the change in that metric between time bin, and the average bandwidth.

4. **Binary Classification**: We formulate this anomaly detection problem as a classified binary classification task. A variety of classifiers, such as Support Vector Machines, K-Nearest-Neighbors, Random Forests, and Decision Trees, can be used at this step to output a final packet-level prediction.

In our research, we experimentally generated normal and attack network traffic on a simulated consumer IoT device network. By applying our framework on the local router, we identified normal and attack traffic with an accuracy above 99%; we found that random forest and K-Nearest-Neighbors classifiers were particularly effective. Another key advantage of our framework is that it is flow-based, stateless, and protocol-agnostic; therefore, it can be implemented on most types of LAN middleboxes (e.g. routers, network switches, firewalls). The IoT Network Monitor's WiFi access point actively runs the FITBOT framework.

CONCLUSION

Along with the quality of the framework, we put specific emphasis on the user-friendliness of the proposed IoT Network Monitor. As we envision regular homeowners adapting this product, many of which are not very tech-savvy, it is very important that the user-interface is be implemented in a straightforward and an intuitive way. To allow for a wide adoption, it is vital that the IoT Network Monitor is easy to use and can provide homeowners a clear and a simple view of the security of their home network.

We believe that the three security and privacy analyses performed by the IoT Network Monitor are foundational to ensure a secure smart-home. Allowing users to easily monitor the security of sensitive personal information being sent on their home network will hopefully reduce the likelihood of cyberattacks on consumer IoT devices.

Furthermore, adoption of the IoT Network Monitor would place pressure on manufacturers to design IoT devices with a focus on security by encouraging consumers to discontinue use of vulnerable devices. In the meantime, the IoT Network Monitor will change default passwords of devices, raising the bar for constructing Mirai-based botnets and making some classes of large DDoS attacks infeasible. However, if it so happens that a device on a home network does become a part of a botnet, the IoT Network Monitor will advise the user to disconnect the device and discontinue its usage, limiting the impact of such large-scale attacks. Therefore, the wide adoption of the IoT Network Monitor would not only increase the security of specific homes, but get us much closer to a more secure Internet-of-Things.