IoT Network Monitor

Name 1
Princeton University
netid@princeton.edu

Name 2
Princeton University
netid@princeton.edu

Name 3
Princeton University
netid@princeton.edu

Noah Apthorpe
Princeton University
apthorpe@princeton.edu

Nick Feamster
Princeton University
feamster@princeton.edu

ABSTRACT

IoT Network Monitor is an intuitive and user-friendly interface for consumers to visualize vulnerabilities of IoT devices in their home. Running on a Raspberry Pi v3 which is configured as a router, the IoT Network Monitor analyzes the traffic of connected devices in three ways. First, it detects devices with default passwords exploited by previous attacks such act he Mirai Botnet, changes the passwords to a randomly generated 12 character string, and reports the new password to the user. Secondly, it conducts deep packet analysis on the data stream of each device and notifies the user of potentially sensitive personal information that is being transmitted in the clear. Lastly, it detects IoT Botnet DoS traffic originating from an IoT device connected to the network and instructs the user to disconnect the device if it has been hacked. This user-friendly IoT Network Monitor will enable homeowners to maintain a good understanding of the security of their home network and understand better what actions are appropriate when a certain security vulnerability is detected. A wide adoption of this tool will not only make consumersâĂŹ homes more secure, but it will get us a step closer to a more secure IoT network.

KEYWORDS

Internet of Things, security, privacy, default passwords, anomaly detection

1 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 behaviorâĂŞâĂŞdata that users want to prevent from being shared with adversaries and network observers. The only visibility consumers have into the privacy of their information is what they see on the apps associated with their devices on their phones; they donâĂŹt have the time or capability to closely scrutinize their outgoing data via Wireshark. Thus we present IoT Network Monitor: an intuitive and user-friendly router to which consumers can connect their devices and visualize the vulnerabilities of IoT devices in their home.

2 RELATED WORK

3 PROPOSED ARCHITECTURE

As consumers continue to adopt IoT devices, consumers will continue to place insecure IoT devices within networks in positions of trust (e.g. smart home networks). Behind the firewall, these IoT devices will have access to other network devices and can pose a variety of security threats. For example, if the device has open ports or insecure passwords, these IoT devices can be hijacked, reveal sensitive information, conduct denial of service attacks on the local network, join an IoT botnet, etc. Yet, 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 not sure what action to take in response. We propose the IoT Security Hub to help combat these issues.

Our framework for the IoT Security Hub includes a user-friendly web interfaceÂăthat integrates with a network middlebox that can sniff traffic on the LAN. The IoT Security Hub collects packet captures as PCAP files throughout the day (say, for the first ten minutes of each hour). The Hub software then runs a variety of scripts to parse and analyze the latest PCAP files. It looks for vulnerabilities in the following three dimensions:

- Password Vulnerability: Scan IoT devices on the network, looking for open ports 22 and 23, and offering a new password if the password is easy to crank.
- Privacy Analysis: Check for unencrypted payloads in packet captures containing personal identifiers and other forms of sensitive data (e.g. medical data).
- Anomaly Detection: Detect IoT Botnet DoS traffic originating from an IoT device on the LAN using the FITBOT framework.

4 IMPLEMENTATION

We implemented the web app as a Flask app to help integrate the front-end interface with the python scripts analyzing the network. This software, as is, is written to be deployed on a Raspberry Pi 3 and relies on the dumpcap utility for packet capture. The implementation code can all be found on Github at: https://github.com/RohanDoshi2018/IoTSec

One thing to note are the challenges associated with designing the user experience with the web platform. Apart from notifying users of security threats with clear colors and relevant text (e.g. sample of leaked sensitive information or a proactive suggestions for a new password), we always provided a call to action to help users understand what to do next in response to a threat.

5 PASSWORD INSPECTION

The first feature of 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 or port 23 (the SSH port or the Telnet port, respectively). This vulnerability of IoT devices âĂŞ when multiple devices are shipped with the same default logon credentials âĂŞ 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.

The following steps were taken in order to achieve this goal:

- First, 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 IP-addresses based on the IP address of the device that the software is running on. This range should be sufficient to capture all devices on a regular home network, while still maintaining a fast runtime.
- The same list of 62 default logon credentials as was used in the Mirai botnet source code is then used in an attempt to log on to the devices found by Nmap. If either a SSH or a Telnet connection is successfully established, the correct username and password combination for the specific device has been found.
- Once all devices with default passwords have been detected, the tool logs on to the devices using the newly found logon credentials. Then, it uses the bash-command passwd to change the password to a randomly generated string, consisting of 12 characters that can be upper or lower case ASCII-characters and digits 0-9.
- Lastly, when vulnerable devices are detected, the new randomly generated password is returned to the user on the user-interface of the IoT network monitor. However, if all 62 logon attempts to a particular device are unsuccessful, it is assumed that the device does not come with a default password, and is labeled secure on the user interface.

6 PRIVACY ANALYSIS

The second feature of IoT Network Monitor intercepts packet data from connected IoT devices and detects potentially sensitive information about the user that is being sent in the clear from the device to the router and on to the server. If 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:

- Separate packet streams: IoT Network Monitor separates connected devices on the network into separate packet streams for analysis. This can be done by classifying packet conversations by the external IP address of the communicating with the device, since the gateway router typically makes it difficult to divide outgoing traffic into sets based on IP addresses of the origin device.
- Separate by protocol: Once a deviceâĂŹs packet stream has been identified, 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.
- Remove headers: Next, packet payloads are separated from the packet headers so that IoT Monitor can examine the application data and screen in for plaintext leaks.
- Find unencrypted packets: Packets are then classified as either plaintext or encrypted using payload entropy as an indicator of encryption. By training 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.
- Search for sensitive information: Lastly, plaintext packets which have been identified 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 Connected Devices Dashboard.

7 ANOMALY DETECTION

The third feature of IoT Network Monitor is a feature that detects anomalous behavior of any devices connected to the hub. Many IoT devices are fundamentally insecure, exposing the Internet to a variety of new attacks. In particular, IoT botnets, like the Mirai botnet, are leveraging these insecure IoT devices to conduct some of the strongest distributed denial of service (DDoS) attacks ever recorded. These type of IoT botnet attacks are growing in frequency and magnitude, restricting the ability of citizens to access the Internet freely and reliably. 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 IoT device 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. After all, 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. The key contributions in feature engineering leverage observations of IoT-specific device behaviors (e.g. limited number of endpoints, regular internal time intervals between packets, small size packet payloads for attacks, etc.).

The FITBOT framework proposes four key steps:

- Network Packet Capture: The middlebox will sample network traffic and generate 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.

- 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.
- 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, interpacket 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
- 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. âĂĺ

In our research, we experimentally generate normal and attack network traffic on a simulated consumer IoT device network. By applying our framework on the local router, we identify 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).

8 EVALUATION

9 CONCLUSIONS

Along with the quality of the framework, we put specific emphasis on the user-friendliness of the proposed IoT Network Inspector. As we envision regular homeowners adapting this product, many of which are not very tech-savvy, it was very important to us that the user-interface would be implemented in a straightforward and an intuitive way. To allow for a wide adoption of a product such as this one, it is vital that the product 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 features that the IoT Network Monitor focuses on are the foundational factors of a secure smart-home. By allowing users to easily monitor the security of sensitive personal information being sent by their devices, the likelihood of cyberattacks (such as ransom attacks) on users of IoT devices would decrease.

Furthermore, more pressure would be put on manufacturers as more users would discontinue the use of vulnerable devices, forcing manufacturers to take action and ensure proper encryption. Moreover, as this tool changes default passwords of devices, a wide adoption of this device would disable the construction of Mirai based botnets, therefore, eliminating some large DDoS attacks. However, if it so happens that a device on a home network does become a part of a botnet, our tool will advise the user to disconnect the device and discontinue its usage, further decreasing the risks of large attacks of such kind. Therefore, a wide adoption of this device would not only increase the security of specific homes, but get us much closer to a more secure worldwide IoT network.

10 FUTURE WORK

REFERENCES