Threat Modeling Executive Document for Mycroft, a open-source voice assistant (revised September 2018)

Jay Chow, Kevin Hamilton, and Yuqing Wang, MSSI graduate students, JHUISI

Abstract - This document outlines our project, which was, to develop a threat model for Mycroft, an open-source voice assistant. Threat modeling is a technique meant to improve the effectiveness of a security assessment. We specifically looked at Mycroft to ensure none of the tenets of security could be broken including: confidentiality, integrity, availability, authenticity, accountability, non-repudiation, and reliability. Ideally, threat modeling should be conducted before the implementation of a software application. By thinking from the perspective of the attacker, security analysts can identify components in a system that would have the highest risk of attack. These components should have priority in security assessments. This document covers our project goals, project execution highlights, project accomplishments, and what we learned. The appendix includes tables and other information.

1 Project Overview

Our goal as a group was to identify threats to Mycroft-core, an open-source voice assistant. We chose Mycroft-core because we found that voice assistants are only going to become more and more prevalent moving forward into the future. In the media, there has been a lot of speculation about the security and privacy regarding the contents of the conversations people have with voice assistants. With the huge increase of IoT devices, security has become a more complex and serious issue. We wanted, as a group, to take a deep dive to see if an open-source tool not tied to big tech companies would be a safer, more secure solution for people who want to have a customizable and highly configurable voice assistant.

1.1 Mycroft Overview

Mycroft-Core can be installed on Linux, built on a Raspberry Pi, or run on Mycroft's own specially made hardware with built-in audio systems. Mycroft-Core has very few supported voice commands. Mycroft-Core allows for voice command extensions via skills. These skills are downloaded separately by the user to extend the functionality of Mycroft. For the scope of this project, we focused exclusively on just Mycroft-Core without any added skills.

Mycroft-Core has a long list of functionality which can be viewed in table format in the appendix of this document. Mycroft-Core has configuration options but its default is to connect to Mycroft's own backend. This project focuses on building a threat model using the default configuration. Our architecture diagram helps illustrate the process from when a user gives a voice command through to Mycroft's response.

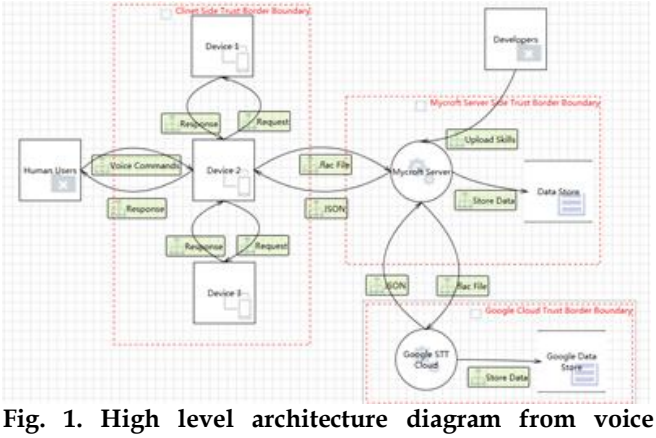

Fig. 1. High level architecture diagram from voice command and response

1.2 Mycroft Architecture

Mycroft uses a client-server architecture to turn the user's voice command into text format. The user starts by prompting Mycroft to start listening by using the wake word, which is set to the default "Hey, Mycroft". When this wake word is heard, Mycroft starts recording an audio file. When no more voice is captured, the file is closed and sent via Mycroft's API, which can be found in the source code. This .flac file, the audio file, is sent to Mycroft's own servers outside of the user's local network using HTTPS. Mycroft's privacy policy states they store a user's voice recordings for the purposes of testing, developing, and improving voice-related technology. We

believe that at this stage Mycroft stores a copy of this voice file in their own internal database. Mycroft then sends this audio file to Google's Speak to Text cloud service, anonymizing where the audio file came from in terms of the original device IP. While this process is outside our view we have reason to believe that communication between Mycroft servers and Google's Speak to text cloud will also utilize HTTPS. Once the text file has returned back to the client, it is parsed by the available commands to determine a response.

2 PROJECT EXECUTION

In this project, we urge the readers to think about the potential security and privacy concerns of Mycroft. To our knowledge, security and privacy threats of Mycroft have not received enough attention. To investigate the security of Mycroft, we begin by consulting relative documents and codes. We focus on Mycroft on the Linux platform and test the main functions. Based upon the understanding of the functions in light of the product, we create a threat model to analyze the architecture of the system. Then, since Mycroft handles the data in a relatively unsafe way, we extend the analysis to include user privacy and asset safety. We decompose the application to identify, document and rate the threats of each part. By doing so, we unveil useful details about the Mycroft system and gain an understanding of the overall system.

On a high level, we describe the way we conducted threat modeling for Mycroft.

Threat Modeling Process:

1) Identify the valuable assets that the system must protect.
2) Use tables and the Microsoft threat modeling tool to document the architecture of the application, including subsystems, trust boundaries and data flow.
3) Decompose the architecture of the application, including the underlying network and host infrastructure design, to create a security profile for the application. The aim of the security profile is to uncover vulnerabilities in the design, implementation or deployment configuration of the application.
4) Identify the threats that could affect the application, keeping the goals of an attacker in mind, and with knowledge of the architecture and potential vulnerabilities of the application.
5) Document each threat using a common threat template that defines a core set of attributes to capture for each threat.
6) Rate the threats to prioritize and address the most significant threats first. These threats present the biggest risk, and the rating process weights the probability of the threat against damage that could result should an attack occur.

3 ACCOMPLISHMENTS

We followed the official tutorials to install Mycroft on a virtual machine. Mycroft must have a python environment since it needs the support of many python libraries. The simplest way to install Mycroft for Linux is to clone the mycroft-core repo to your system and run a shell script, which will install all dependencies. You can get all the resources by running command "git clone https://github.com/MycroftAI/mycroft-core.git" and then set up the environment by executing the command "bash dev_setup.sh".

We tested Mycroft for Linux on the Ubuntu platform because it's the most stable platform for Mycroft. The device must have a built-in microphone and speakers and must be connected to the Internet since the client side will connect to the server side.

The process of testing the Mycroft client on our own virtual machine helped us better understand how Mycroft actually works and deals with the voice commands. We even discovered some obvious threats during the test. For example, Mycroft can't identify the user's voice which can be a high level threat since it may allow malicious voice commands from attackers and may lead to dangerous outcomes. We determined that doing this light dynamic analysis of Mycroft's execution would give us a better understanding of the attack surfaces.

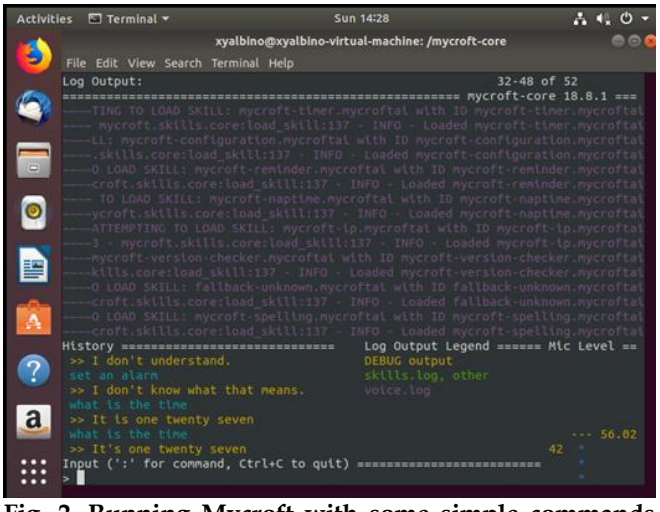

Fig. 2. Running Mycroft with some simple commands on Ubuntu platform.

To use Mycroft, we have to pair the device to a Mycroft home account (it can be a github or google account) by way of a 6-digit code. The interesting thing is the only way to get the 6-digit pair code is to ask Mycroft via a voice command. This can be a high risk threat since Mycroft doesn't have any function to identify the correct voice which means anyone could pair your device to another account without your permission. We think this is a huge design mistake. More importantly, Mycroft

won't provide users much protection for the connection though it will alert users to set up firewalls for the socket (8081 by default) on their own. This may be a problem since not every user is used to such settings.

4 LESSONS LEARNED

One of the major takeaways from this project we learned is just how challenging developing a threat model can be. This lesson is reinforced by the volume of vulnerabilities that are introduced with the introduction of new software. Some of the top companies with tremendous amounts of human talent are still prone to letting threats bypass their close watch. To add to the challenge of this project, we were doing a threat model on software that has already been developed. We had to rely on documentation and thousands of lines of code to determine the functionality, trust boundaries, and threats to the system. While we found this project challenging, the more time we spent trying to understand the technology and common attack patterns, we could start to see how these common attack patterns could be used to pose a threat to Mycroft-Core.

We found some threats for Mycroft during the test and analysis, both high-risk and low-risk. For example, Mycroft can't identify a user's voice which might be a high-risk threat because it may allow others to perform malicious commands. And the threat that attackers may get the voice should be a low-risk threat since it's not very easy to intercept the packages or hack in the device. We also considered outside of the box threats, a example of this would be a man in the middle attack. The communication between the device and Mycroft's servers is encrypted but the packet headers contain source and destination IPs. A malicious actor could listen on the wire and capture these packets creating a model of when the user is at home by the usage of their voice assistant. This model may give the malicious actor intervals of time in which they have confidence the user is not home leaving them vulnerable to a robbery.

Along with the nonconventional attacks, we also believe that commonly used attacks pose a threat to the security tenets. We believe that a DDoS attack on Mycroft's backend servers might be enough to disrupt the availability for users. We also believe if a malicious actor had admin rights on a Mycroft enabled system they could add python code to fork the .flac file to not only be sent to Mycroft's backend servers but to a their own command and control system as well. This attack is one of high importance because most users would not be checking the integrity of the Mycroft code to ensure that it hasn't been injected with malicious code or checking to see if an abnormal connection has been opened. This attack is low noise meaning it would not appear to the user that behavior of the system has changed.

Another important issue we learned from the task is how dangerous the virtual voice assistant can be. The technology may be new to some users, but they are in the marketplace today. The work we did referenced some famous products such as Google Home, Apple's Siri and Amazon's Alexa. All these products have always been susceptible to accidental hijack. To our knowledge, security and privacy threats of this technology have not yet received enough attention.

REFERENCES

[1] Mycroft AI, Inc., "Mycroft.AI Documentation," Myecroft Documentation, https://mycroft.ai/documentation. 2015.

[2] M. Scholefield, "Mycroft Core the Mycroft Artificial Intelligence platform," https://github.com/MycroftAI/mycroft-core. 2018.

[3] Mycroft AI, Inc., "Mycroft for Linux ", Mycroft Documentation, https://mycroft.ai/documentation/linux. 2015.

[4] H. Chung, M. Iorge, J. Voas, S. Lee, "Alex, Can I Trust You," *IEEE Computer Society*, vol.50, no. 9, pp.100-104, Sep 2017, doi:10.1109/MC.2017.3517053. (IEEE Transactions)

[5] Wikipedia, "Virtual Assistant (Artificial Intelligence) ," https://en.wikipedia.org/wiki/Virtual_assistant_artificial_intelligence. 2018.

[6] Google Cloud, "Cloud Speech-to-Text Documentation," Google Cloud Platform,https://cloud.google.com/speech-to-text. 2018.

[7] Christian Science Monitor, "What do Alexa and Siri mean for privacy?"https://www.csmonitor.com/Technology/2017/0114/Devices-sprout-ears-What-do-Alexa-and-Siri-mean-for-privacy. 2017.

[8] Mycroft AI Inc., "Mycroft AI PrivacyPolicy" https://home.mycroft.ai/#/privacy-policy?_k=wxjbkj.

KEY TERMS

Threat - A potential occurrence, malicious of otherwise, that might damage or compromise the assets.

Vulnerability - A weakness in some aspect or feature of a system that makes a threat possible. Vulnerabilities might exist at the network, host or application levels.

Attack (or exploit) - An action taken by someone or something that harms an asset. This could be someone following through a threat or exploiting a vulnerability.

Countermeasure - A safeguard that addresses a threat and mitigates risk.