Data Privacy Concerns with Third-Party Skills in Alexa Amazon Ecosystem

Caelan MacArthur

Central Washington University

9/19/2021

Abstract

Data misuse within smart speakers and the Amazon Alexa ecosystem has been a focus of study within the last couple of years. This study investigates user data misusage by third-party skills in the Amazon Alexa ecosystem using honeytokens.

Honeytokens, like honeypots, hold fake resources (false data) that are generated to track cybercriminals, advisors, or any individual who has the intention of misusing virtual resources that do not belong to them. In this study's case, honeytokens are email addresses, user profiles, names, health data, or device IDs associated with an Amazon account.

When the honeytoken is released to a third-party skill, we give them access to the requested permissions from the third-party skill. Emails are commonly requested by the skills, hence they are the most effective way of tracking data misuse. After the email has been released, we simply wait for targeted advertisements associated with that email address. After receiving we then contact the company behind the skill and request the data be deleted. If they do not delete the data, it is a misuse of data and a violation of their privacy policy.

Additionally, if received emails contain targeted advertising, it is possible to track personally identifying data such as health, name, and location based on the contextual search from the emails received. Sensitive health data (medical data) is of particular concern since if released, it can threat federally protected classes.

Although the built a framework this summer to allow for automated testing of user data violations on third-party Alexa skills, the testing phase was not conducted due to time constraints. This paper will focus on the construction of said framework and how the honeytokens can be used within it.

Keywords: Honeytoken, Smart speakers, Data misuse, Utterances, Intents, Slots, Handlers, Alexa's SDK, Information leakage, OAuth, Amazon Cognito, Amazon Cognito Pools, S3, user interfaces (UI).

Introduction

Smart homes like Alexa have changed the landscape of consumerism towards advancement in day to day uses of smart technology. Consumers now have the ability to automate tasks like scheduling, ordering products, and accessing information without looking at a physical device, leading to everything being much quicker and more convenient. With a consumer base of 90 million adults and growing, these smart homes generate a large amount of data and have created a convoluted ecosystem. It is easy for bad actors to take advantage in order to harvest user data to sell, or misuse it for a more malicious purpose (Edu, 2021).

In previous studies, cyber security professionals and security researchers have used honeytokens to track cybercriminals' access to restricted system resources, as well as generally track malware and its spread. In the HoneyCirculator study by Mistsuaki Akiyama, the HoneyCirculator released false user credentials malware that attacks web-based applications and monitors their attacks (Akiyama al, 2018).

Additionally, in the study CanaryTrap by Zubair Shafiq and Shehroze Farooqi, honeytokens in emails and Facebook profiles were used to measure user data misuse in the Facebook third-party marketplace. Results showed that 48% did not comply with a request to delete user data (Farooqi al, 2020).

In this study, it is proposed to distribute honeytokens to third party Alexa skills using data in the form of user emails, Amazon user profiles, and usernames that all act as user credentials. These user credentials are created, managed, and maintained within an automation framework.

This automation framework utilizes a voice synthesizer, voice recorder, Selenium web driver, and Gmail accounts to track honeytokens and record audio interactions with Alexa.

Alexa Overview

Since Alexa's features are complex, there is a need for a brief overview of the factors that play into automating the testing of honeytokens. Both the Alexa Alexa Software Development Kit (SDK) and Alexa security play a huge role in how the testing process is developed.

Alexa Skill Kit

"Intents", "utterances", "slots", and "handlers" are terms that make up the basic functionality of the Alexa Software Development Kit (SDK) (Alexa Skills Kit, 2021). Intents are the process of Alexa attempting to fulfill the user's vocal requests. Utterances are preprogrammed phrases that Alexa Skills have to be taught to respond to the user's intents. Handlers take the user's intents and hand them off the Alexa SDK. The Alexa SDK will fulfill the user's intent by navigating it through all endpoints, backend solutions, and Amazon Web Services (Amazon *Create Intents, Utterances, and Slots*, 2021). Alexa SDK is responsible for the communication between the Alexa Voice Service SDK, Amazon Web Services, and third-party skills' custom backend solutions (Alexa Skills Kit, 2021).

There are several different skill types in Alexa, and each type has a unique API. This ranges from Smart Home skills that control your home devices, to automotive skills that you can use to control the functionality of your car (Alexa Skills Kit SDKs, 2021).

Alexa Security

Alexa security has made it difficult to fully automate honeytoken testing. Third-party skills can implement account-linking for OAuth authentication for users. Account-linking is the process of connecting Alexa accounts with either an Amazon account, third-party APIs (like

Google's APIs), or a custom solution that third-party applications have developed (Amazon Account Linking, 2021).

Additionally, Amazon Cognito provides a backend solution where users can sign in to other accounts like Facebook, Apple ID account, Google Gmail, etc., where it creates a profile for that user with their data. It serves as a way of streamlining the process of storing data and creating profiles so third-party developers can identify users and make their data accessible on devices (Amazon Account Linking, 2021). Amazon Cognito Pools powers Amazon Cognito by having a user's Alexa app private API token connect to Amazon Web Services S3 (cloud storage), route to the third-party application (Facebook account, Apple ID, etc.), uses that information to create a user profile, and then returns that to Amazon S3 to create an account. When a user signs into another device and uses that skill, their Alexa app pulls up the profile stored in S3, thus making it accessible across multiple devices (Amazon Cognito Pools, 2021).

Alexa Ecosystem

The Alexa ecosystem consists of a massive amount of the skills available to Alexa-enabled devices. Alexa-supported devices range all the way from doorbells to sunglasses (Amazon.com: Echo Smart Speaker, 2021).

Excluding Alexa smart speakers like the Echo and Echo Dot, many of these devices require supplemental devices in order to create a single skill. For example, a developer had an Amazon Firestick app that developer published to the Firestick store and an Alexa skill app that went along with it, that developer would need the video skill API, smart home skill, network and WiFi skills API, Amazon Cognito, an Amazon Firestick, monitor, and TV. Amazon Cognito would act as the backend, smart home skill API would connect all devices together, video skill API would control the TV's functionality, network and wi-fi skills handle network connections between

devices and the router (switch, bus, etc), and the Firestick would act as the smart TV. If that developer as the third-party developer wanted to create a skill with an account linking to Google's OAuth API, that developer would then have to go through the additional setup website and connect the Alexa to the Google API that links to the user's Gmail.

Honeytoken Testing and Automation Data

The study *SkillVet* was the main source of data pulled to base the automation software off of. *SkillVet* scraped nearly the entire Alexa store for key data such as the name and account ID of the skill, permissions requested, and whether account linking was enabled. When doing my own analysis of their dataset, I focused on two main questions: whether a skill was requesting more permissions than it needed, and if skills were listing all required permissions. We noticed early on that many skills that had account linking did not have permissions listed at all, which I'll go into more later. Permissions, name, and skill ID were the main categories analyzed.

Requesting permissions for private data doesn't necessarily mean a skill is doing anything nefarious. Most skills have a need for a specific piece of information, for example, a weather skill may need to know my location first if I ask "Alexa, what's the weather outside today?". At the same time, a weather skill shouldn't need payment information or my email in order to tell me what my location's weather forecast is. I found that a lot of skills that added on data permissions they did not need. The "Sanjeev Kapoor Recipes" Alexa skill, for example, is a cooking app. One of its sample utterances is "Alexa, ask Sanjeev Kapoor Recipes how to prepare cake?" (Alexa Skill Store, 2021). This skill requests lists read and lists write access, Alexa notifications, full name, device country, postal code, mobile number, reminders, and email address. Needless to say, a cooking skill shouldn't require access to my full name, device

country and postal code, mobile number, or even email address just to give me recipe information.

Let's take a deeper look at its privacy policy on its website. It states that “At the same time, we are quite particular that your privacy should be protected” in the first paragraph. Later on, it goes on to describe its data usage for their website, but nothing for their Alexa app. The privacy policy additionally states that “By using the website www.sanjeevkapoor.com, you consent to the collection, storage, and use of your browsing history on this website. We share this information with trusted 3rd parties that use it to show you marketing communications of interest only.” (*Sanjeev Kapoor*, 2021). This is one of the many examples of permissions and privacy policies that do not equate. Nowhere on the website is there a privacy policy for the Alexa skill.

This is a common trend for permissions and privacy policies, so where does this leave us for what type of data to use for honeytokens? When comparing data across four countries (the United States, Australia, Germany, and France), it was found that the most common permission requested was for email. Germany and France in particular are under the protection of the General Data Protection Regulation (GDPR), which can make it particularly important to compare and contrast what permissions are requested there versus non-GDPR countries like Australia and the U.S. Regardless, we decided to build the honeytoken off of email addresses - specifically, Gmail addresses.

Honeytokens

Gmail email has a few key advantages for creating honeytokens. In addition to being free, Gmail has a functionality where it ignores periods inside the email: for example, “your.name@gmail.com” will be sent to the same inbox as “yourname@gmail.com”. Amazon

allows up to 54 characters in an email address, so it is possible to have 54 unique email addresses for each Gmail account used in the study. Each email address is used only once for a single third-party skill, making it easily identifiable where the bad actor is if the Gmail account's inbox receives a targeted advertisement addressed to that email. We can also test misuse of postal codes using targeted advertisements in the same way. For example, if someone live in Madison, Wisconsin and got an ad for Farm and Fleet, a local retailer, we will know that our postal code was also sold.

Account Linking, Automated Testing, and the Alexa Ecosystem

As mentioned previously, when doing data analysis of the *SkillVet* data, I noticed a large number of skills that had account linking, but no permissions listed on the Amazon Alexa store (Edu, al 2021). The skills only list their permissions during the process of signing up for account linking, when users typically click not without thinking about what they're agreeing to. This can potentially give these skills access to information that they do not need, but more problematically for the study, it makes automated testing a nightmare.

From a testing and automation standpoint, this leads to a difficult situation given how complex the Alexa ecosystem is. Automating interactions with Alexa, honeytoken distribution for skills without account linking, and Amazon Cognito is feasible. However, skills with account linking and Amazon Cognito become an issue due to the sheer number of possible ways a skill can have account linking or Amazon Cognito. With so many providers, full automation becomes nearly impossible.

However, there are a couple of possible avenues. The easiest is to have a machine training model that can predict the location and desired test of possible fields that skills will display text boxes for, so that it can navigate account linking user interfaces (UI).

Automated Testing for Skills without Account Linking

Even with this restriction, the *SkillVet* dataset still consists of tens of thousands of Alexa skills that need to be tested. Given the need to scale honeytoken distribution on such a large scale, it was necessary to build a framework that would automate several functions: data record-keeping, data scraping for Alexa store for Utterances, changing Gmail addresses in the Amazon portal, and retrieving one-time-passwords for Amazon authentication. In addition, a voice synthesizer would need to be made to talk to Alexa using the correct intents, voice recording software to log everything said, and automating file management.

In order advective this Selenium WebDriver, Selenium Python bindings, Google Text-to-Speech (gTTS), SpeechRecognition, google-api-python-client, beautifulsoup, Playsound, and Pyaudio. Selenium was the core of the automation framework. It did everything from data scraping to navigating every website (sign in, enter text fields, etc). The voice synthesizer and voice recorder were built on Playsound, gTTS - Google text to speech, SpeechRecognition. Playsound played the audio files that gTTS - Google text to speech provided from the Utterances that Selenium got from the Alexa skills store. SpeechRecognition recorded all of the voice interactions with Alexa. JSON files stored all of the testing data, while Python created directories, files, and moved them for file management. While changing emails, Google Gmail API to get the most recent email, and hand that off to beautifulsoup, where it parsed through the HTML to get the one-time-password. Once the one-time-password was fetched, it was handed back to Selenium to navigate the Amazon account portal to change the email and enter the one-time-password.

The next step would be to have the program parse over each of the Alexa Skills from the SkillVet data, get each skills names, enter them into the search bar of the Alexa skill store, search

for an exact match within the HTML, once that element is found, click to get the skill page, receive the utterances from the third party Alexa, and give them to the voice synthesizer. Once this is achieved, testing would be fully automated provided that reCaptcha and account linking do not cause interference.

Road Blocks

reCaptcha is currently stopping the automation framework during testing. It is rather hard to get around, but the easiest method of making sure that it does occur is to simply imitate human user behavior. If you make the program wait, reCaptcha is less likely to be prompted. However, there is a function that servers use called user agent that requests in the header "application, operating system, vendor, and/or version of the requesting" when a request has been made to connect (Mozilla Developer Docs, (2021). Usually, when testing web applications, automating data scraping, etc this information is not provided and thus, triggers reCatpcha. Although, if you use fake-useragent it provides that information to the server when headers are being exchanged. Fake-useagent is database of all headers that User-Agent would request and puts when them into the header a get request is being made, thus, bypassing this problem (*fake-useragent*, 2018).

Conclusion

Although testing did not happen, the automating framework is put in place to do so and even though it was not able to be fully automated, the ability to scale Honeytokens is provided. Although the roadblock of reCaptcha is present, there are ways to get around it. The only unsolved, or potentially unsolvable roadblock is account linking, which unfortunately seems to have the most potential for bad actors to utilize.

References

Alexa Voice Service SDK (2021a). *GitHub - alexa/avs-device-sdk: An SDK for commercial device makers to integrate Alexa directly into connected products.* GitHub https://github.com/alexa/avs-device-sdk

Alexa Skills Kit SDKs | Alexa Skills Kit. (2021). Amazon (Alexa). https://developer.amazon.com/en-US/docs/alexa/sdk/alexa-skills-kit-sdks.html

Amazon.com: Echo Smart Speaker (2021) *Amazon.com: Echo Smart Speakers & Displays: Amazon Devices & Accessories: Smart Speakers, Smart Displays & More*. Amazon Echo & Alexa Devices. https://www.amazon.com/smart-home-devices/b?ie=UTF8&node=9818047011

Alexa Skill Store (2021) *Amazon.com: Sanjeev Kapoor Smart Recipes : Alexa Skills*. (2021). Alexa Skill Store. https://www.amazon.com/Sanjeev-Kapoors-FoodFood-Kapoor-Recipes/dp/B07C7KZCMR/ref=sr_1_1?dchild=1&keywords=Sanjeev+Kapoor+Recipes&qid=1632392362&s=digital-skills&sr=1-1

Amazon Account Linking (2021). *Add Account Linking to Your Alexa Skill | Alexa Skills Kit*. Amazon (Alexa). https://developer.amazon.com/en-US/docs/alexa/account-linking/add-account-linking.html

Amazon Cognito Pools. (2021). *Amazon Cognito user pools - Amazon Cognito*. Amazon Cognito Pools. https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-identity-pools.html

Amazon *Create Intents, Utterances, and Slots* (2021). *Create Intents, Utterances, and Slots | Alexa Skills Kit*. Amazon (Alexa).https://developer.amazon.com/en-US/docs/alexa/custom-skills/create-intents-utterances-and-slots.html

Akiyama, M., Yagi, T., Hariu, T., & Kadobayashi, Y. (2018). HoneyCirculator: distributing credential honeytoken for introspection of web-based attack cycle. International Journal of Information Security, 17(2), 135–151. https://doi.org/10.1007/s10207-017-0361-5

Chung, H., Park, J., & Lee, S. (2017). Digital forensic approaches for Amazon Alexa ecosystem. Digital Investigation, 22, S15–S25. https://doi.org/10.1016/j.diin.2017.06.010

Edu, J., Ferrer-Aran, X., Such, J., & Suarez-Tangil, G. (2021, March). *SkillVet: Automated Traceability Analysis of Amazon Alexa Skills*. Cornell University. https://arxiv.org/pdf/2103.02637.pdf

Edu, J., Ferrer-Aran, X., Such, J., & Suarez-Tangil, G. (2021, March). *SkillVet/NewFullDataset at main · jideedu/SkillVet*. GitHub. https://github.com/jideedu/SkillVet/tree/main/NewFullDataset

Farooqi, S., Musa, M., Shafiq, Z., & Zaffar, F. (2020). CanaryTrap: Detecting Data Misuse by Third-Party Apps on Online Social Networks. Proceedings on Privacy Enhancing Technologies, 2020(4), 336–354. https://doi.org/10.2478/popets-2020-0076

General Data Protection Regulation (GDPR) – Official Legal Text. (2019, September 2). General Data Protection Regulation (GDPR). https://gdpr-info.eu/

Lau, J., Zimmerman, B., & Schaub, F. (2018). Alexa, Are You Listening? Proceedings of the ACM on Human-Computer Interaction, 2(CSCW), 1–31. https://doi.org/10.1145/3274371

Lentzsch, C., Shah, S., Andow, B., Degeling, M., Das, A., & Enck, W. (2021). Hey Alexa, is this Skill Safe?: Taking a Closer Look at the Alexa Skill Ecosystem. https://anupamdas.org/paper/NDSS2021.pdf

Sanjeev Kapoor, (2021). Sanjeev Kapoor. https://www.sanjeevkapoor.com/Privacy-Policy.aspx

fake-useragent. (2018). PyPI. https://pypi.org/project/fake-useragent/

Mozilla Developer Docs, (2021) *User-Agent - HTTP | MDN*. *User-Agent - HTTP | MDN*. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent