Computers, Materials & Continua CMC, vol.61, no.3, pp.1075-1090, 2019

Security Analysis of Smart Speaker: Security Attacks and Mitigation

Youngseok Park[1], Hyunsang Choi[1], Sanghyun Cho[1] and Young-Gab Kim[2, *]

Abstract: The speech recognition technology has been increasingly common in our lives. Recently, a number of commercial smart speakers containing the personal assistant system using speech recognition came out. While the smart speaker vendors have been concerned about the intelligence and the convenience of their assistants, but there have been little mentions of the smart speakers in security aspects. As the smart speakers are becoming the hub for home automation, its security vulnerabilities can cause critical problems. In this paper, we categorize attack vectors and classify them into hardware-based, network-based, and software-based. With the attack vectors, we describe the detail attack scenarios and show the result of tests on several commercial smart speakers. In addition, we suggest guidelines to mitigate various attacks against smart speaker ecosystem.

Keywords: Smart speaker, personal assistant system, IoT security.

1 Introduction

Nowadays, billions of Internet of Things (IoT) devices which extend internet connectivity beyond traditional devices are increasingly deployed to the market. In such an environment, smartphones play an important role as a ubiquitous computing interface between IoT devices and users. Particularly, Voice User Interface (VUI) is growing as a key interface for IoT devices since it becomes more practical to provide a great user experience to humans due to the impressive recent advances in speech recognition technologies.

The speech recognition is currently used by smart speakers which are also known as an artificial intelligence speaker such as Amazon Echo [Amazon Echo (2019)] and Google Home [Google Home (2019)]. The voice-controlled smart speakers are rapidly becoming the next big thing (i.e., according to Gartner's report, the smart speaker market will reach at $3.52 billion by 2021 [Gartner (2017)]), capable of answering questions, setting timers, playing music and so on. Furthermore, smart speakers can also function as a home assistant, e.g., controlling robot vacuums, smart lights, and door locks.

Smart speaker vendors usually concentrated their efforts on increasing their virtual assistants' communicative abilities but there have been little mentions of security and privacy. Since smart speakers are dealing with personal information and expanding their functionality to paying bills and managing bank accounts [Lifewird (2019);

[1] Naver Corp., 6 Buljeong-ro, Jeongja-dong, Bundang-gu, Gyeonggi-do, Korea.

[2] Sejong University, 209, Neungdong-ro, Gwangjin-gu, Seoul, 05006, Korea.

[*] Corresponding Author: Young-Gab Kim. Email: alwaysgabi@sejong.ac.kr.

StrategyCorps (2017)], securing the smart speaker is imperative. Several studies Robles et al. [Robles, Kim, Cook et al. (2010); Babar, Mahalle, Prasad et al. (2010)] on the security of smart home and IoT devices have been proposed. However, to the best of our knowledge, there have been no previous studies done on the smart speaker security analysis which explores not only general security attributes as an IoT device but also the distinct security features as a speech recognition system.

In this paper, we describe a common structure of a smart speaker ecosystem and enumerate attack surfaces. We classify the attack surfaces into hardware-based, network-based and software-based surfaces based on the structure of the ecosystem. We also illustrate existing smart speaker attacks and assess five commercial smart speakers to launched network-based attacks on test environments. During the analysis, we found several vulnerabilities which enable attackers to steal authentication data and personal information of users. Moreover, attackers can even inject arbitrary commands to the speaker. We suggest guidelines to mitigate the corresponding attacks.

The remainder of this paper is organized as follows. Section 2 describes the background of smart speaker ecosystem. Section 3 clarifies the taxonomy of attack surfaces and possible attack methods. In Section 4, we also propose mitigations concerning for each smart speaker attack. Discussions of this study are presented in Section 5. We summarize related works in Section 6 and offer the conclusion in Section 7.

2 Background

2.1 Smart speaker

A smart speaker is a voice command wireless speaker which offers interactive actions to human with an integrated Artificial Intelligence (AI). Smart speaker ecosystem generally consists of three key components: a device, a cloud-based voice assistant service, and a skill set. The device is hardware typically packed with microphones and speakers. The cloud-based voice assistant service such as Amazon Alexa [Amazon Alexa (2019)] provides speech interpretation, user intent understanding, and spoken results. The skillset enables a user to interact with a smart speaker in a more intuitive way using voice functions such as playing music, setting alarms and providing weather information. Every speech recognition task today is driven by machine learning and statistical language models. Speech recognition has been around for decades but it hits the mainstream recently since deep learning makes the speech recognition accurate enough. In the smart speaker ecosystem, the cloud-based voice service plays a role as the actual brain behind millions of smart speaker devices and voice applications as shown in Fig. 1.

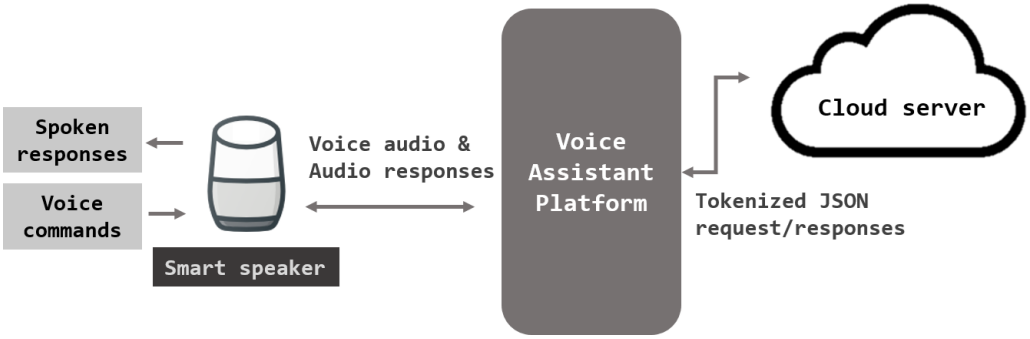


Figure 1: Cloud-based service platform of smart speaker ecosystem

2.2 ASR, NLU and TTS

To make machines understand human speech, the audio data has to be transcribed into text. The process is typically referred to as Automatic Speech Recognition (ASR) [Yu and Deng (2016)]. With a help of Natural Language Understanding (NLU) [Allen (1995)], machines can deduce what human speech actually means by using deep learning algorithms [Young, Hazarika, Poria et al. (2018)]. The NLU also generates a semantic representation of responding text. Finally, Text-To-Speech (TTS) [Dutoit (1997)] converts text into speech. For example, as depicted in Fig. 2, when a user requests to a smart speaker ("*What is status of my online order?*"), the speaker sends the voice data to the ASR server and it transcribes into text. The NLU converts the text to semantic representation as *INTENT ("STATUS", "ORDER")*. It also makes semantic representation for a response as *STATUS ("SHIPPED", "09-19-2018")* and generates natural language interpretation such as "*It is shipped on Sep 19 in 2018*". The TTS synthesizes audio data with the natural language text and the smart speaker plays the synthesized audio data.

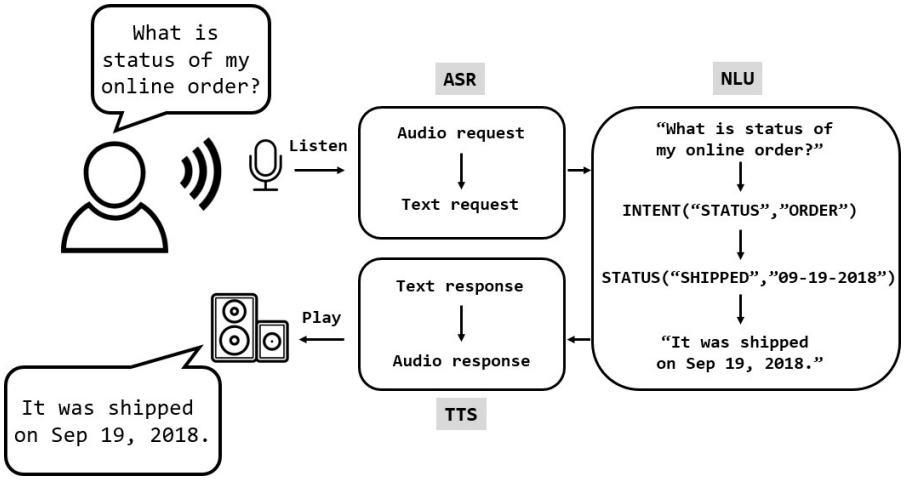


Figure 2: ASR and TTS structure

3 Smart speaker attacks

3.1 Attack Surfaces

In this section, we describe attack vectors of smart speakers and classify them into hardware-based, network-based, and software-based attack vectors as shown in Tab. 1.

Table 1: Smart speaker attack vector taxonomy

Type	Category	Attack vector	Attack example
Hardware	Port	External ports	Access a root shell through USB, JTAG or UART
		Internal ports	
	Chipset	Flash memory	Flash memory dump to obtain firmwares
	Others	Microphone	Dolphin attack
Network	Wi-Fi	Smartphone-speaker	MITM attack on network traffic during initial setup, ARS, TTS, etc.
		Speaker-server	
		Smartphone-server	
	RF	Personal Area Network	Blueborne attack
Software	Client OS	Smart speaker OS	Exploit 0-day, 1-day vulnerabilities
	Client app	Smart speaker app	
		Smartphone app	
	Server	Speech recognition system	Cocaine noodles, Hidden voice commands

The smart speaker has a number of hardware components. Among them, we explore several physical ports and chipsets which are likely to be exploitable. The microphone is a unique hardware attack vector of smart speakers.

Network-based attacks are generally performed by a Man-In-The-Middle attack (MITM) to eavesdrop network traffic and inject commands. For example, unencrypted network traffic during smart speaker setup or communication with ASR/TTS servers, there can be vulnerabilities which enable an attacker to steal user information and inject arbitrary commands to the smart speaker. Personal area network communications such as Bluetooth are also candidates of network-based attack vector.

Smart speaker operating system such as Android can be exploited when there are 0-day or 1-day vulnerabilities of the operating system. Smart speaker applications installed in the device or smartphone applications can be also exploited by adversaries if they have unpatched vulnerabilities. An adversarial machine learning attack on speech recognition system is a unique attack vector of smart speaker ecosystem. The detailed attack scenarios are introduced in subsections in Section 3 and mitigations will be described in Section 4.

As mentioned previously, there are several attack vectors of smart speaker ecosystem, and some of them are derived from unique characteristics of smart speakers. In this

section, we describe the detail attack scenarios and show the result of tests on several commercial smart speakers that we launched.

3.2 Test environment

We tested five commercial smart speakers (Nugu [SKT Nugu (2019)], Gigagenie [KT Gigagenie (2019)], Wave [Naver Wave (2019)], Echo [Amazon Echo (2019)] and Google Home [Google Home (2019)]) and specifications of the speakers are presented in Tab. 2.

All five smart speakers have Application Processor (AP) for their OS (i.e., Android or Linux) and communication modules such as Wi-Fi and Bluetooth. Particularly, Gigagenie has a wired LAN port and an HDMI port since it works as a TV set-top box and Wave has infrared (IR) transmitter to mount remote controller for home appliances. We set up our access point as a proxy (i.e., MITM) to capture network traffic for the network analysis as shown in Fig. 3.

3.3 Hardware-based attacks

Hardware architecture of commercial smart speakers is similar. They consist of a motherboard, speaker modules, and buttons to control the devices. In terms of hardware attack surfaces, physical ports include internal and external ports can be potential targets. Almost all smart speakers have external or internal ports and some of them are able to be used for debugging. If attackers break into the system through the debug ports, they can get a root shell through the ports and firmware of the smart speaker. Also, a microphone in the smart speaker can be another target using specific sounds.

Table 2: Specification of tested commercial smart speakers

	Nugu	Gigagenie	Wave	Echo	GoogleHome
Release	Aug. 2016	Jan. 2017	May 2017	Nov. 2014	Nov. 2016
Vendor	SKT	KT	Naver	Amazon	Google
OS	Android	Android	Linux	Fire OS (Android)	Android
AP	TCC8935	Hi3798CV200	APQ8009	DM3725	Marvell 88DE300
Comm.	802.11n, BT 4.0	802.11a/b/g/ac, BT 4.1	802.11b/g/n, BT 4.0	802.11a/b/g/n, BT 4.0	802.11ac BT4.2
Etc.		Wired LAN HDMI out, USB, SD card	IR transmitter		

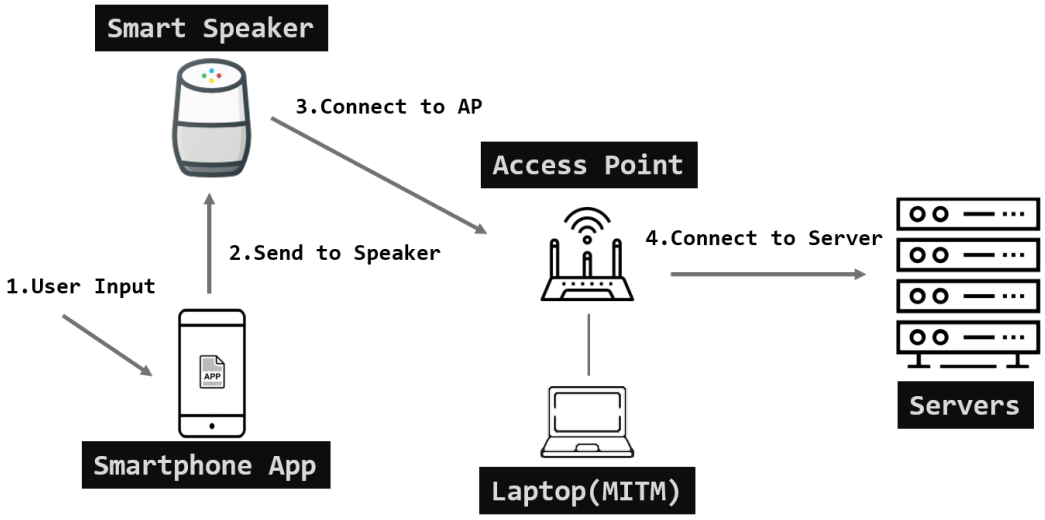


Figure 3: Initial setup process of the smart speaker and the laptop connected to the access point to perform MITM attack

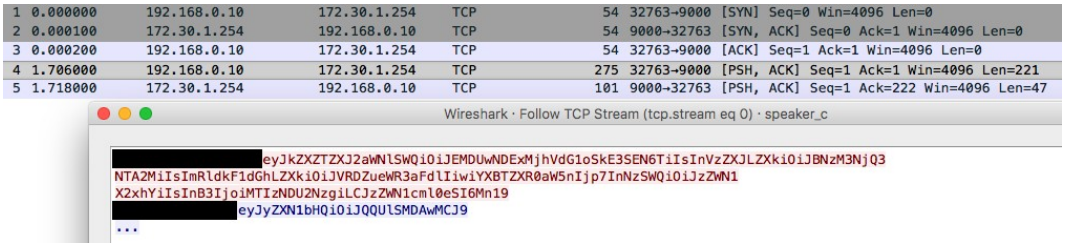
Debug ports: Researchers of MWR InfoSecurity were able to boot into a generic Linux environment from an external SD card attached to exposed UART debug pads of Amazon Echo [Mark (2017)]. By booting into the actual firmware on the Echo, they installed a persistent implant and they succeeded to gain remote root shell access to Amazon Echo.

Chipset: An Attacker can acquire firmware data of a smart speaker from the flash memory. However, recent smart speakers encrypt flash memory data so it is hard to analyze the firmware even the attacker acquire the flash dump.

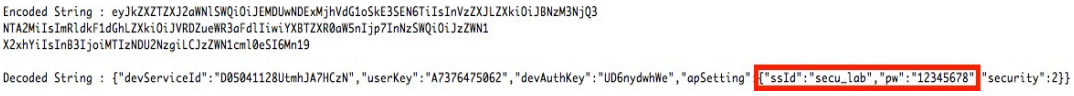
Dolphin attack: As an example of attacking on the microphone of a smart speaker, there is Dolphin attack proposed by Zhang et al. [Zhang, Yan, Ji et al. (2017)]. They set up a speaker to broadcast voice commands that had been shifted into ultrasonic frequencies which are out of range from human hearing but the smart speaker still can receive it as a voice command. It is possible to activate a smart speaker from several feet away using the dolphin attack. Therefore, an attacker can send an arbitrary voice command to a smart speaker without user's perception.

3.4 Network-based attacks

Initial Setup: Similar to other IoT devices, smart speakers need initial configuration. The configuration is for connecting to the voice assistant platform and authenticating an owner. Fig. 3 describes a typical process of the initial setup. Smart speakers are generally connected to the home Wi-Fi access point for an Internet connection. Therefore, SSID and password of the access point have to be provided through a smartphone application. During the initial setup, an attacker can capture packets (i.e., containing SSID and password) sent from the smartphone application to the smart speaker. If the packet is not encrypted, the password of the access point can be stolen.



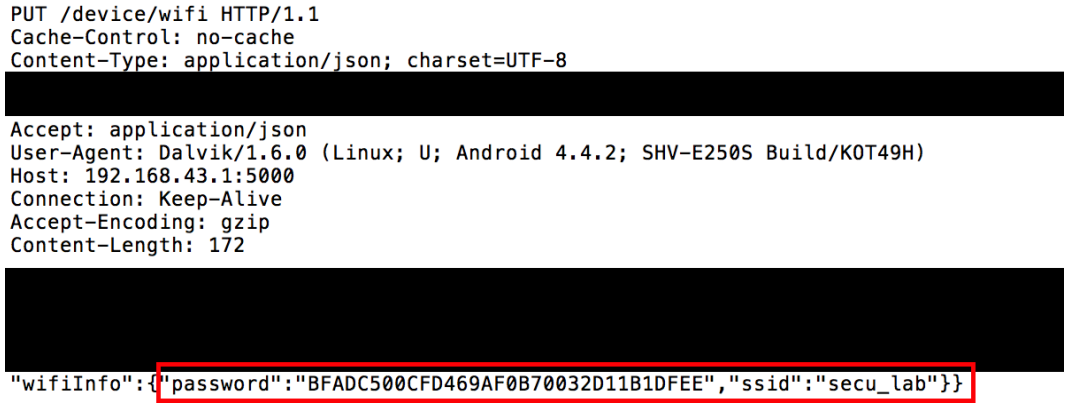
(a)



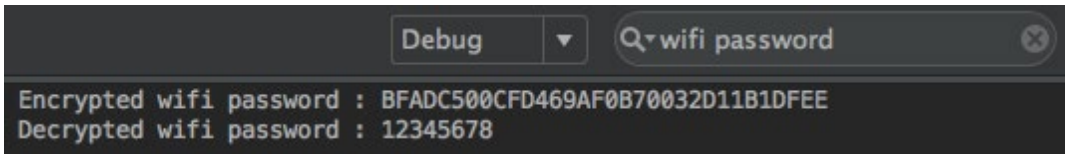
(b)

Figure 4: (a) Initial setup packet of *Speaker B*. (b) The data is encoded with Base64 and plain text of the password is revealed

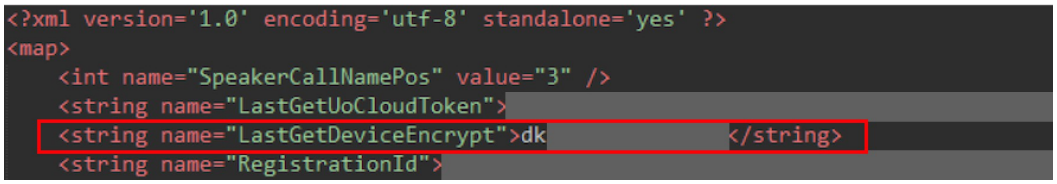
Echo and Google Home encrypted the password of an access point in packets with an asymmetric key from the server. However, Gigagenie used BASE64 encoding to deliver SSID and password of the access point as shown in Fig. 4. The SSID and password were able to be decoded (e.g., Wi-Fi SSID: "*secu_lab*", password: "*12345678*"). In the case of Nugu, encrypted access point password was sent from a smartphone app. We reverse-engineered the smartphone app and found out that it uses AES encryption [Daemen and Rijmen (2013)] and the key was hard-coded in an XML file inside the smartphone app (see Fig. 5(c)). We were able to decrypt the access point password by using the key ("*BFADC500CFD469AF0B70032D11B1DFEE*" to "*12345678*") as shown in Fig. 5(a) and Fig. 5(b). Since the same key was found in the firmware of the speaker, it was capable of decrypting the access point password for all devices with the key.



(a)



(b)



(c)

Figure 5: (a) SSID and password are sent to the speaker through a Wi-Fi access point. HTTP requests contains encrypted password data (b) The password can be decrypted (c) with the hard-coded key in smartphone app

ASR and TTS: While a smart speaker communicates with ASR and TTS server, the packets are likely to have the owner's voice and private information such as schedule and address. Wave, Echo and Google Home used TLS but Nugu and Gigagenie did not encrypt their communication channel. As shown in Fig. 6, ASR packets of Nugu contain plain voice data encoded as Speex format [Valin (2016)]. By capturing these ASR packets, attackers can extract the user's voice data. Afterward, the attacker can synthesize the voice data [Candyvoice (2019)] to send forged commands to smart speakers.

```
20 20 20 20 20 20 20 20  20 20 20 20 20 43 4d 4d                  CMM
44 3d 22 41 44 44 5f 53  50 45 45 43 48 5f 44 41   D="ADD_S PEECH_DA
54 41 22 09 46 4c 41 47  3d 22 30 22 09 4c 45 4e   TA".FLAG ="0".LEN
47 3d 22 38 39 34 22 0d  0a 4f 67 67 53 00 02 00   G="894". .OggS...
00 00 00 00 00 00 00 94  b2 29 13 00 00 00 00 4b   ........ .).....K
a4 b6 1d 01 50 53 70 65  65 78 20 20 20 73 70 65   ....PSpe ex   spe
65 78 2d 31 2e 32 62 65  74 61 33 00 00 00 00 00   ex-1.2be ta3.....
00 01 00 00 00 50 00 00  00 80 3e 00 00 01 00 00   .....P.. ..>.....
00 04 00 00 00 01 00 00  00 ff ff ff ff 40 01 00   ........ .....@..
00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00   ........ ........
00 00 00 00 00 4f 67 67  53 00 00 00 00 00 00 00   .....Ogg S.......
00 00 00 94 b2 29 13 01  00 00 00 52 de a1 3f 01   .....).. ...R..?.
3e 21 00 00 00 45 6e 63  6f 64 65 64 20 77 69 74   >!...Enc oded wit
68 20 53 70 65 65 78 20  73 70 65 65 78 2d 31 2e   h Speex  speex-1.
32 62 65 74 61 33 01 00  00 00 11 00 00 00 61 75   2beta3.. ......au
```

Figure 6: ASR packets of Nugu

Keep-alive Connection: Smart speakers have to maintain a connection with their servers in order to provide connection-oriented service to users and they typically use keep-alive packets to maintain a persistent connection. We found that Nugu used unencrypted keep-alive packets which have authentication information (i.e., token) as shown in Fig. 7. The

token was leveraged for keeping the session information associated with a user. Nugu sent voice data with the token to ASR server and received JSON intent data from Keep-alive server. With the intent data, Nugu sent TTS request and received a TTS response. However, if an attacker sends voice data with a token hijacked from keep-alive packets, the attacker can get the JSON intent data containing the user's information (see Fig. 8).

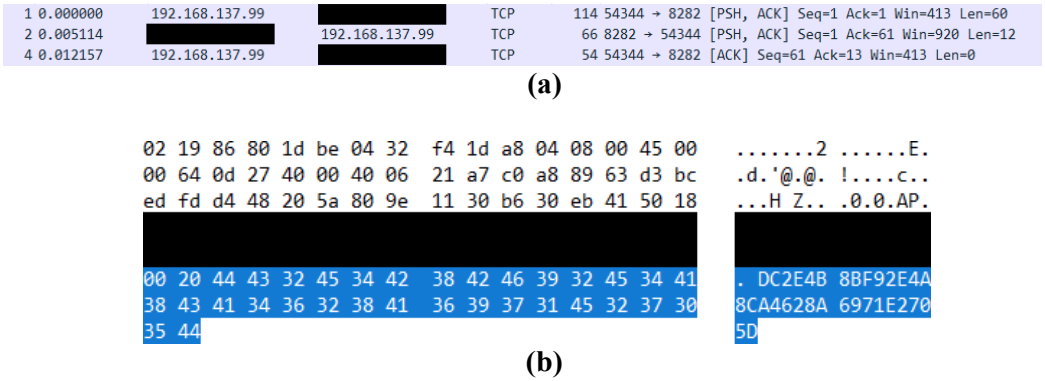


Figure 7: (a) Keep-alive packets (b) Exposed token value

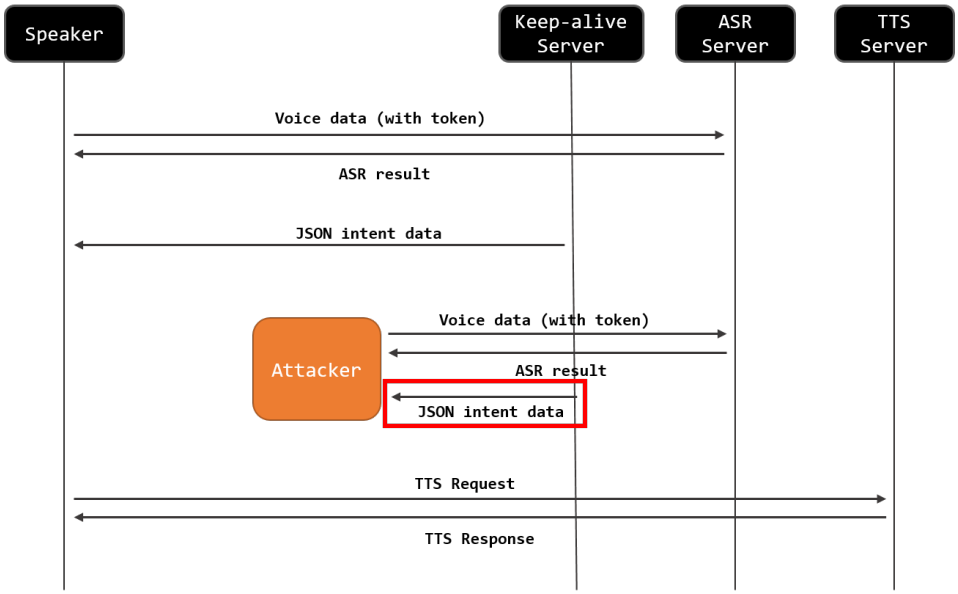


Figure 8: Command injection to Nugu with a hijacked token

Firmware OTA: Recent IoT devices update their firmware or application by downloading files via the Internet. If these packets are not encrypted, an attacker can obtain firmware data and use the data for finding vulnerabilities. Fig. 9 shows the firmware Over-The-Air (OTA) packets of Nugu. We were able to acquire release information and APK file from the OTA packets.

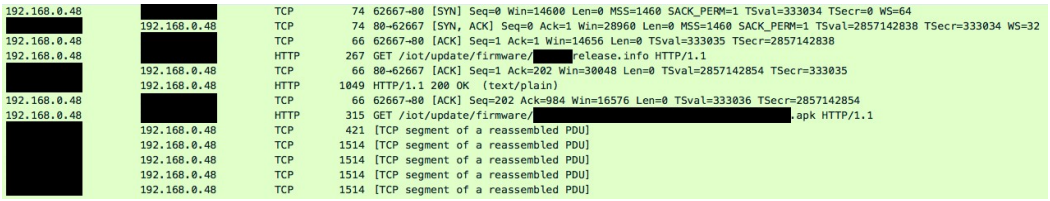


Figure 9: Firmware OTA packets of Nugu

Blueborne: Radio frequency communications such as Bluetooth, NFC, and Zigbee are another network attack vectors of smart speakers. In 2017, Bluetooth vulnerability Blueborne was discovered [Ben and Gregory (2017)]. Security researchers of Armis Lab obtained a remote shell of Amazon Echo using the Blueborne vulnerability. However, the vulnerabilities were already patched for all the tested smart speakers.

3.5 Software-based attacks

Client Operating System: Most of the smart speakers have an Android-based operating system. Therefore, attacking the client operating system of smart speakers is equivalent to exploiting Android operating system using its known or unknown vulnerabilities. Because smart speakers are often built upon an old version of Android which has unpatched vulnerabilities, they would be exposed to recent 1-day attacks. We performed a port scanning on the five smart speakers and the results are shown in Tab. 3. The open ports during the initial setup are different from the open ports for operation. As open ports are identified, each can be tested using a number of automated tools (e.g., fuzz testing [Godefroid, Levin and Molnar (2012)]) to find vulnerabilities.

Table 3: Smart speaker open TCP ports

		Nugu	Gigagenie	Wave	Echo	Google Home
Open ports (TCP)	Initial setup	5000 (http)	7547 (tcpwrapped)	N/A (Blue tooth)	443 (https)	8008 (http)
		5050 (http)	7557 (tcpwrapped)		8080 (http)	8009 (ajp13)
			8058 (senomix07)			9000 (tcpwrapped)
			38520 (unknown)			10001 (tcpwrapped)
	Operation	N/A	7547 (tcpwrapped)	N/A	4070 (tripe)	8008 (http)
			7557 (tcpwrapped)		4071 (aibkup)	8009 (aho13)
			8058 (senomix07)		55442 (nagios-nsca)	9000 (cslistener)
			38520 (unknown)		55443 (unknown)	10001 (scp-config)

Client Application: Attacking client applications is similar to attacking smartphone applications. Adversaries can find security vulnerabilities after they obtain the source code of application via a reverse engineering. The detail of reverse engineering and exploiting smartphone applications will not be covered in this paper.

Server Application: A server-side application such as NLU has been targeted by attackers.

Cocaine Noodles [Vaidya, Zhang, Sherr et al. (2015)], an adversarial machine learning approach to speech recognition system, proves that an adversary can produce sound interpreted as a voice command to speech recognition system but not easily understandable by humans. The same researchers proposed advanced attack, hidden voice commands [Carlini, Mishra, Vaidya et al. (2016)] which are unintelligible to human listeners but which are interpreted as commands by devices by making noise-like sounds.

4 Mitigations

We propose mitigations against the aforementioned smart speaker attacks as shown in Tab. 4. Removing (or disabling) unnecessary debug ports and applying access control for debugging such as secure ADB for Android can help prevent attacks.

Table 4: Mitigation methods for identified attack vectors

Type	Attack Vectors	Mitigations
Hardware	Ports	Remove or disable unnecessary debug port
		Use secure ADB for Android
	Flash memory	'Write only' permission in firmware area
	Microphone	Microphone enhancement (suppress ultrasound range)
		Inaudible voice command cancellation
Network	Wi-Fi communication	Do not use hard coded key
		Apply TLS encryption
		Apply HTTP Public Key Pinning (HPKP) if possible
	RF communication	Maintain up-to-date library and OS
Software	OS and applications	Encrypt firmware
		Firmware code signing
	Speech recognition	Generate audible feedback
		Speaker recognition

The Read-out Protection (RDP) [ST (2016)] is a global flash memory read protection allowing the firmware to be protected against dumping or other means of intrusive attacks. Therefore, it is better off applying RDP to prevent firmware disclosure.

Since the dolphin attack uses ultrasound waves leveraging the nonlinearity of the A/D converter and the original wave already demodulated after passing A/D converter phases. Therefore, the high-frequency waves are needed to be deleted before the waves are converted to digital information.

Adopting network traffic encryption is the key to mitigating network-based attacks against smart speakers. HTTP public key pinning (HPKP) [Evans, Palmer and Sleevi (2015)] can reduce the risk of a MITM attack on encrypted traffic such as SSL strip attacks [Marlinspike (2009)]. Authentication data such as Wi-Fi password have to be encrypted with an asymmetric key, not hard-coded symmetric key. To secure RF communication, maintaining up-to-date OS and libraries with security patches is appropriate.

Code signing for firmware is a proper way of keeping the integrity and thwarting attempts of firmware modification.

To enhance speech recognition robust against adversarial machine learning approaches, generating audible feedbacks for critical commands (e.g., payment commands) can be helpful. In addition, if a smart speaker can distinguish each user (i.e., speaker recognition), crafted voice commands are hardly accepted as valid commands.

5 Discussion

We enumerate a number of approaches to attack smart speakers but some attacks have limitations. First, flash memory dumps are becoming extremely difficult because the latest smart speakers have already adopted mitigation such as code protection as mentioned in Section 4. However, the hardware-based attacks are still possible by leveraging Scanning Electron Microscopy (SEM) or glitching attack [Courbon, Skorobogatov and Woods (2016); Giller (2015)].

Second, the dolphin attack Zhang et al. [Zhang, Yan, Ji et al. (2017)] can be launched from several feet away (e.g., distances vary from 2 cm to a maximum value of 175 cm across devices) but portable attack with a smartphone, an ultrasonic transducer and a low-cost amplifier as described in their paper allows the adversary to hide the attack device inside a pocket (or a bag) and to access to a target close enough.

Third, an attacker has to sniff network traffic via a MITM attack prior to network-based attacks. However, the MITM attack can be carried out because there are a lot of vulnerable access points which have 1-day vulnerabilities or use default admin password as demonstrated in Mirai botnet case [Antonakakis, April, Bailey et al. (2017)].

Notably, some vulnerabilities such as Blueborne are patched or removed. However, vulnerabilities will always exist. Therefore, we have to consider that there will be hidden vulnerabilities and try to find them before they are used by hackers.

6 Related works

A smart speaker is a new type of IoT devices currently in the spotlight. However, the security of the smart speaker has not been introduced before, we refer several attacks related to the smart speaker ecosystem.

6.1 Smart home security

Smart speakers have the role of a hub for a smart home system because of convenience in controlling IoT devices with a voice command. Therefore, a smart speaker can be a new target for an attacker to infiltrate into the smart home system. Heartfield et al. [Heartfield, Loukas, Budimir et al. (2018)] investigated and showed security threat taxonomy in a smart home. They enumerate possible attack vectors in the smart home system from a physical layer such as an infrared sensor and a voice to the application layer. They also referred a method to manipulate personal assistant services with a voice command from television or somewhere not spoken by the legitimated user.

6.2 Voice replay attack

The simplest way to manipulate smart speakers is to record and play a voice to them. There are several studies on distinguishing a person's voice from recording voice. Mankad et al. [Mankad, Shah and Grag (2018)] presented a method for detecting voice replay attacks using spectrum analysis (i.e., MFCC, IMFCC) and classifiers (i.e., ANN, SVM). Nguyen et al. [Nguyen and Vo (2018)] showed a simple study to identify different speakers to prevent a voice command recorded by an attacker. Wu et al. [Wu, Evans, Kinnunen et al. (2015)] surveyed spoofing attacks with a replay, speech synthesis, voice conversion, and countermeasures.

6.3 Attack against speech recognition

There have been proposed various attacks which target the speech recognition systems. Jang et al. [Jang, Song, Chung et al. (2014)] presented the exploit that bypasses the security modules in the various OS such as Windows, Ubuntu, iOS and Android using the accessibility system using voice input. Diao et al. [Diao, Liu, Zhou et al. (2014)] introduced the study bypassing permission in Android with Google voice assistant. Above studies attack non-hidden channel of the speech recognition system so the attack can be discovered by the user. Vaidya et al. [Vaidya, Zhang, Sherr et al. (2015)] introduced a proof-of-concept attack using the difference in mechanisms of the speech recognition between human and machine. Carlini et al. [Carlini, Mishra, Vaidya et al. (2016)] showed the realistic attack against speech recognition system of Android smartphone ("*OK Google*") by making noise-like sounds for humans but the machine can understand. Furthermore, the same authors introduced an adversarial machine learning against DeepSpeech [Hannun, Case, Casper et al. (2014)] that makes any audio waveform by only adding a slight distortion [Carlini and Wagner (2018)]. Zhang et al. [Zhang, Yan, Ji et al. (2017)] presented Dolphin attack using ultrasonic waves instead of using noise-like sounds. They set up a speaker to broadcast voice commands that had been shifted into ultrasonic frequencies which are out of range of human hearing (over 20 kHz) but the smart speaker still can receive it as a voice command. Skill squatting attack [Kumar, Paccagnella, Murley et al. (2018)] is another attack against speech recognition by leveraging systematic errors in the voice recognition system.

7 Conclusion

This paper seeks to present security analysis on artificial intelligence smart speakers by identifying overall system structure and attack vectors of off-the-shelf smart speaker products. We classify the attack vectors into hardware, network, and software vectors. We also perform network-based analysis to the smart speaker products. The analysis is carried out by taking a closer look at smartphone applications and network traffic of smart speakers and we find out several vulnerabilities. By exploiting the vulnerabilities, we could steal an access point password, eavesdrop the user requests and responses. We could also send arbitrary commands to smart speakers by stealing and reusing authentication tokens. Additionally, we propose guidelines to mitigate the corresponding attacks. Since smart speakers will play an important role in home automation, it is necessary to strengthen the security of smart speakers.

Acknowledgement: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00231, Development of artificial intelligence based video security technology and systems for public infrastructure safety)

References

Allen, J. (1995): *Natural Language Understanding*. Pearson.

Amazon Alexa (2019): https://developer.amazon.com/alexa.

Amazon Echo (2019): https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E.

Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E. et al. (2017): Understanding the Mirai botnet. *Proceedings of 26th USENIX Security Symposium*, pp. 1093-1110.

Babar, S.; Mahalle, P.; Stango, A.; Prasad, N.; Prasad, R. (2010): Proposed security model and threat taxonomy for the Internet of Things (IoT). *Proceedings of International Conference on Network Security and Applications*, pp. 420-429.

Ben, S.; Gregory, V. (2017): BlueBorne.

http://go.armis.com/hubfs/BlueBorne%20Technical%20White%20Paper1.pdf?t=1536599737375.

Candyvoice (2019): Voice synthesis service.

https://candyvoice.com/how-it-works?lang=en.

Carlini, N.; Mishra, P.; Vaidya, T.; Zhang, Y.; Sherr, M. et al. (2016): Hidden voice commands. *Proceedings of USENIX Security Symposium*, pp. 513-530.

Carlini, N.; Wagner, D. (2018): Audio adversarial examples: targeted attacks on speech-to-text. *Proceedings of 2018 IEEE Security and Privacy Workshops*, pp. 1-7.

Courbon, F.; Skorobogatov, S.; Woods, C. (2016): Reverse engineering flash eeprom memories using scanning electron microscopy. *Proceedings of International Conference on Smart Card Research and Advanced Applications*, pp. 57-72.

Daemen, J.; Rijmen, V. (2013): *The Design of Rijndael: AES-the Advanced Encryption Standard*. Springer.

Diao, W.; Liu, X.; Zhou, Z.; Zhang, K. (2014): Your voice assistant is mine: how to abuse speakers to steal information and control your phone. *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, pp. 63-74.

Dutoit, T. (1993): *An Introduction to Text-to-Speech Synthesis*. Springer.

Evans, C.; Palmer, C.; Sleevi, R. (2015): Public key pinning extension for HTTP. RFC 7469.

Gartner (2017): Gartner says worldwide spending on VPA-enabled wireless speakers will top $3.5 billion by 2021. http://www.gartner.com/newsroom/id/3790964.

Giller, B. (2015): Implementing practical electrical glitching attacks. *Black Hat Europe*.

Godefroid, P.; Levin, M. Y.; Molnar, D. A. (2012): SAGE: whitebox fuzzing for security testing. *ACM Queue*, vol. 10, no. 1, pp. 1-8.

Google Home (2019): https://madeby.google.com/intl/en_us/home/.

Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G. et al. (2014): Deep speech: scaling up end-to-end speech recognition. arXiv:1412.5567.

Heartfield, R.; Loukas, G.; Budimir, S.; Bezemskij, A.; Fontaine, J. R. et al. (2018): Taxonomy of cyber-physical threats and impact in the smart home. *Computers & Security*, vol. 78, pp. 398-428.

Jang, Y.; Song, C.; Chung, S. P.; Wang, T.; Lee, W. (2014): A11y attacks: exploiting accessibility in operating systems. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 103-115.

KT Gigagenie (2019): https://gigagenie.kt.com.

Kumar, D.; Paccagnella, R.; Murley, P.; Hennenfent, E.; Mason, J. et al. (2018); Skill squatting attacks on Amazon Alexa. *Proceedings of 27th USENIX Security Symposium*, pp. 33-47.

Lifewird (2019): How to shop with Amazon Alexa. https://www.lifewire.com/shop-with-amazon-alexa-4158255.

Mankad, S. H.; Shah, V.; Garg, S. (2018): Towards development of smart and reliable voice based personal assistants. *Proceedings of TENCON 2018*, pp. 2473-2478.

Mark, B. (2017): Alexa, are you listening? https://labs.mwrinfosecurity.com/blog/alexa-are-you-listening.

Marlinspike, M. (2009): More tricks for defeating SSL in practice. *Black Hat USA.*

Naver Wave (2019): https://clova.ai/ko/ko-product-wave.html.

Nguyen, M. S.; Vo, T. L. (2018): Resident identification in smart home by voice biometrics. *Proceedings of International Conference on Future Data and Security Engineering*, pp. 433-448.

Robles, R. J.; Kim, T. H.; Cook, D.; Das, S. (2010): A review on security in smart home development. *International Journal of Advanced Science and Technology*, vol. 15, pp. 13-22.

SKT Nugu (2019): http://www.nugu.co.kr/.

ST (2016): Application note: proprietary code read-out protection on microcontrollers of STM.

https://www.st.com/content/ccc/resource/technical/document/application_note/89/12/c5/e2/0d/0e/45/7f/DM00186528.pdf/files/DM00186528.pdf/jcr:content/translations/en.DM00186528.pdf.

StrategyCorps (2017): Alexa, what's my checking account balance?

https://www.strategycorps.com/new-blog/2017/11/30/alexa-whats-my-checking-account-balance.

Vaidya, T.; Zhang, Y.; Sherr, M.; Shields, C. (2015): Cocaine noodles: exploiting the gap between human and machine speech recognition. *Proceedings of Workshop on Offensive Technologies*, pp. 1-14.

Valin, J. M. (2016): Speex: a free codec for free speech. arXiv:1602.08668.

Wu, Z.; Evans, N.; Kinnunen, T.; Yamagishi, J.; Alegre, F. et al. (2015): Spoofing and countermeasures for speaker verification: a survey. *Speech Communication*, vol. 66, pp. 130-153.

Young, T.; Hazarika, D.; Poria, S.; Cambria, E. (2018): Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55-75.

Yu, D.; Deng, L. (2016): *Automatic Speech Recognition*. Springer.

Zhang, G.; Yan, C.; Ji, X.; Zhang, T.; Zhang, T. et al. (2017): DolphinAttack: inaudible voice commands. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 103-117.