Exploring Vulnerabilities in Voice Command Skills for Connected Vehicles

Wenbo Ding[1], Song Liao[2], Keyan Guo[1], Fuqiang Zhang[2], Long Cheng[2], Ziming Zhao[1], and Hongxin Hu[1(✉)]

[1] University at Buffalo, Buffalo, NY, USA
{wenbodin,keyanguo,zimingzh,hongxinh}@buffalo.edu
[2] Clemson University, Clemson, SC, USA
{song,fuqianz,lcheng2}@clemson.edu

Abstract. Voice assistant platforms have revolutionized user interactions with connected vehicles, providing the convenience of controlling them through simple voice commands. However, this innovation also brings about significant cyber-risks to voice-controlled vehicles. This paper presents a novel attack that showcases the ability of a "malicious" skill, utilizing the skill ranking system on the Alexa platform, to hijack voice commands originally intended for a benign third-party connected vehicle skill. Through our evaluation, we demonstrate the effectiveness of this attack by successfully hijacking commonly used commands in commercial connected vehicle skills.

Keywords: Alexa · Voice Assistant Skills · Connected Vehicle

1 Introduction

The introduction of Alexa skills for connected vehicles has revolutionized the way users interact with their cars, offering a novel and voice-controlled approach. However, this technological advancement also brings forth a range of emerging cyber threats that pose risks to voice-controlled vehicles. While the convenience of interacting with connected vehicles through voice commands is undoubtedly significant, it is important to recognize that this progress has simultaneously given rise to new vulnerabilities that users must contend with.

The "connected car" category on the Alexa platform currently lists 148 skills [3], while Google's "control car" category offers 32 actions. The Alexa-connected vehicle API [4] provides users with 10 sample commands to control their vehicles through voice interactions. Common voice commands include actions like "start my car," "open the window," or "unlock the car." When users issue these voice commands, the Alexa platform identifies the most relevant connected vehicle skill to fulfill the request. It then sends directives [2] to the car vendor's cloud platform, which subsequently transmits the commands to the user's car.

Even though these skills enhance user experience, they can also be manipulated by malicious actors. Previous work, Wang et al. [15] has shown that

Y. Chen et al. (Eds.): SmartSP 2023, LNICST 552, pp. 3–14, 2024.
https://doi.org/10.1007/978-3-031-51630-6_1

malicious skills can circumvent the vetting process and get published. Once a malicious skill is employed by a user, it can define deceptive commands identical to those of benign skills. When the Amazon Alexa system receives a voice command, it must first identify a skill to execute the command. If two or more skills define the same commands, the Alexa platform must choose the most relevant skill among the potential candidates. Attackers could employ certain strategies, such as defining more similar commands, to masquerade their malicious skill as more relevant. Consequently, Alexa may activate the malicious skill instead of the original benign skill, thereby allowing the malicious skill to hijack voice commands from other benign third-party skills.

In this paper, we identify a vulnerability within the Alexa system that permits an over-privilege attack. This vulnerability could be exploited by attackers to hijack benign third-party connected vehicle skills. Through an in-depth analysis of the Alexa-connected car skill system and command processing, we found that developers have the ability to define their own voice commands. Surprisingly, these can be identical to Alexa's official, built-in commands, leading to potential conflicts between customized and official skills. Furthermore, these third-party customized commands can take precedence over Alexa's built-in commands to control cars or related devices. Thus, an attacker could potentially publish a malicious skill that would be invoked whenever users employ Alexa's built-in voice commands to control users' devices.

We summarize our contributions as follows:

- We conduct a thorough analysis of the Alexa command processing and skill ranking system, including a detailed examination of related parameters such as categories, keywords, utterances, slots, and usages. Through this analysis, we identify a potential vulnerability that arises due to conflicts between the customized commands of third-party skills and built-in skills related to connected vehicles.
- We discover that skills belonging to different categories, such as Q&A and connected vehicle skills, are assigned varying priorities within the skill ranking system. Building upon this insight, we propose and execute a practical attack on an Alexa-connected vehicle skill. Specifically, we implemented this attack on our own account, enabling us to hijack a third-party car remote control skill installed on a Toyota Corolla. Through this attack, we demonstrate the ability to interfere with critical commands, such as locking or starting the car, thereby exposing potential risks.

Our work sheds light on vulnerabilities in the Alexa system and emphasizes the importance of addressing conflicts between connected vehicle skills, prioritization mechanisms, and potential threats to the execution of essential commands through the Alexa system in connected vehicles.

2 Background

In this section, we provide an overview of the fundamental background concepts and address potential issues related to connected vehicle skills.

2.1 Voice Skills and Their APIs

Voice skills serve as applications for Alexa, enabling users to interact with various functionalities through an intuitive voice interface. Alexa offers a hands-free approach for users to perform everyday tasks such as checking the news, playing music, or engaging in games. Furthermore, Alexa allows users to control cloud-connected devices, enabling actions like adjusting lights or modifying thermostat settings. These skills are accessible on Alexa-enabled devices such as Amazon Echo, Amazon Fire TV, and devices produced by other manufacturers.

When a user utters the wake word, "Alexa," and communicates with an Alexa-enabled device, the device transmits the speech data to the Alexa service in the cloud. In the cloud, Alexa processes the speech, comprehends the user's intent, and subsequently sends a request to invoke the corresponding skill capable of fulfilling the user's command. The Alexa service handles the crucial tasks of speech recognition and natural language processing. On the other hand, your skill functions as a service hosted on a cloud platform, facilitating communication with Alexa via a request-response mechanism over the HTTPS interface. Upon invocation of an Alexa skill, your skill receives a POST request comprising a JSON body. Within this request body, the parameters required for your skill to comprehend the user's intent, execute its logic, and generate a response are included.

Commands in the Alexa system are composed of three primary components: intent, utterances, and slots. The commands in the Alexa system are referred to as intent, for instance, "open the door" Within each intent, there can be several similar utterances such as "open the door" "opens the door" or "open the front door" Within each utterance, the developer can specify replaceable keywords as slots, for example, "door" in this case.

2.2 Voice Command Skills for Cars

The Connected Vehicle Skill API includes capability interfaces developed specifically for connected vehicle use cases to simplify the skill-building process, without having to build your own voice interaction model or write sample utterances.

As shown in Fig. 1, Alexa's automotive skills leverage the robust capabilities of Alexa. Automotive and smart home interfaces enable users to issue voice commands to their connected vehicles. Whether it's starting the engine, adjusting temperature settings, or managing door locks, the convenience and ease of use provided by Alexa automotive skills are transforming the way we engage with our cars. The Amazon Echo Cloud receives voice command records from the Alexa speaker and translates them into plain text. These texts will be processed by a skill ranking algorithm, which is designed to choose the most relevant skill to handle this command. Once a skill is decided, the skill's backend code, running on the AWS cloud, receives command directives from the ranking algorithm. Then the backend code will transfer this directive to its vendor's cloud through an Oauth verification process. In the end, the connected vehicle receives

commands from its vendor cloud by LTE or WIFI protocols and reports its new status to the skill.

By utilizing the Alexa. Automotive and smart home interfaces, you can develop Alexa automotive skills tailored for connected vehicles. These skills empower users to interact with their vehicles using any Alexa device or the Alexa app. Users can conveniently perform tasks such as starting or stopping the engine, locking or unlocking the doors, and adjusting temperature settings in different zones of the vehicle. For instance, imagine a scenario on a chilly morning where a user, while preparing for work, can simply instruct Alexa to turn on their car and initiate the defrosting of the windshield. This seamless integration between Alexa and connected vehicles enhances user convenience and offers an intuitive and efficient way to manage their automotive needs.

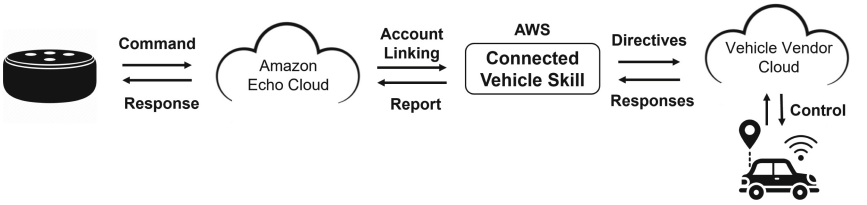

Fig. 1. Overview of Alexa vehicle skill.

3 Threat Model

Our proposed threat model doesn't necessitate direct attacks on intricate systems like those of vehicles. Instead, the primary vulnerability we exploit resides within the Alexa voice assistant ecosystem.

We mainly have one assumption for our attacks which is that malicious voice assistant skills can be installed by users. Attackers can craft and promote malicious voice assistant skills that seemingly offer genuine functionalities. Unsuspecting users, potentially drawn by utility or intrigue, can be led to install these skills. For instance, a malicious skill could impersonate a popular IoT skill, deceiving users into installing it through a squatting attack. Besides, the malicious skill can be installed by users unknowingly in certain scenarios. For example, when users issue voice commands, Alexa may recommend related skill installations based on those commands. Users typically have limited knowledge about the specific skills installed through voice installation.

Once installed, these malicious skills become primed to hijack voice commands intended for legitimate skills, including those that control essential functions such as vehicle operations. For example, when a user verbally commands Alexa to "start the car," our malicious skill might intercept this, causing either a denial of the intended action or triggering an alternate, potentially harmful,

action. This approach allows an attacker to indirectly manipulate or influence car-related functions, not by hacking the car's system directly, but by manipulating the Alexa skill mechanism that users rely upon for remote car commands.

4 Vunlerbility Exploration

In this section, we detail our techniques and observations for attacking the car skills by manipulating the skill ranking and selection process of Alexa. We try to fool Alexa into believing our attacking skills are more "suitable" for the given voice command.

There are two kinds of commands/utterances in the Alexa skills, the official built-in intents, and the developer's customized intents. In the IoT skill, the official commands should have a higher trigger priority than 3rd-party commands, which makes 3rd-party developers cannot override official commands in normal usage scenarios. However, we find it is possible for 3rd-party commands to mislead the command ranking algorithm for a higher execution priority, and then they can take over the execution of official commands.

Alexa uses a two-step shortlisting and re-ranking [1,10] methods to find the most relevant skill for a given utterance. The shortlisting algorithm uses a neural model to find a certain amount of suitable candidate skills for handling a particular utterance, then the re-ranking step uses other contextual features to find the most relevant among these suitable skills.

After translating users' voices into text commands by the ASR, The shortlisting algorithm first gives top "K" intents according to the intent classifier. The intent classifier is based on the model trained by the existing intent dataset to find all skills that can understand this intent command. The contextual re-ranking model considers many contextual signals, like the number of customers, skill ratings, and reviews. Other factors include accurate descriptions and keywords, the skill category, and the ability to parser the voice intent slot.

The original utterances amount is based on the given utterances from the Alexa document examples. In our testing skills, we enable several new skills with more utterances and slots in each intent and a well-explained description. Then we test how many utterances a skill needs to be triggered prior to the built-in intent.

Hijacking Car Skills' Commands. We tested car skills on the Alexa Platform. Since all car skills need account linking with its device vendor. We only deployed a Drone mobile skill with a Compustar control unit.

To implement our attack, we deployed an additional skill with the same utterances and added utterances and slots to our skill. In each utterance, we implement two slots in each utterance based on the simple structure of the utterances, e.g., the verb as actions and the noun as a targeted car. We keep adding the utterances until our skill is triggered instead of the Drone mobile skill. The detailed results are listed in the evaluation section. We also test the influence of usage history of skills. We increase the usage of our skill to more than

50 times by manually triggering the skill and then giving the built-in command with no specifically assigned skill name. We did not notice any significant effect from increased recent skill usage.

For built-in utterances, we tested all 8 exampled commands on the Alexa-connected car API document [5] and we are able to hijack/redirect all of them. However, we cannot hijack the skill-specific utterances that are directly sent to the skill, e.g., ask Drone Mobile to lock my car. Based on the above findings, we can perform attacks such as preventing car locking and opening the trunk while driving by hijacking corresponding commands.

5 Implementation and Evaluation

5.1 Car Skill Implementation

Although the Amazon Alexa platform offers a skill simulator with a text-based interface that accepts a textual input, and provides a textual output for skill testing purposes, it is challenging to test connected vehicle skills using the simulator.

Fig. 2. DroneMobile devices and modified Carolla.

In our experiment, we first implemented a virtual device called "my car", which supports necessary commands, e.g., "turn on the car" or "lock the car". Only by implementing such a virtual device, the Alexa system can properly parse a command to recognize related devices and skills. Otherwise, it cannot identify potentially related skills and only responds with "Sorry, we did not find such a device". The virtual car device is implemented in a benign IoT skill, which

contains the code for device discovery and voice command handling process. The detailed discovery and command handling information is provided in the Alexa document [4].

Later, we tested a connected car skill and its built-in commands on a real car. Since all car skills need account linking with its device vendor, we deployed a Drone mobile skill with Compustar controller [7] 4900 model with Drone Mobile on a 2010 Corolla as shown in Fig. 2. Limited to device and car availability, we are unable to deploy other connected car devices or skills. However, one skill made us test built-in and 3rd-party car-related commands. In this skill, it can implement commands like remote lock/unlock, remote start/stop the engine, and open trunk.

Figure 3 displays the control panel of the Drone Mobile remote control system, which shows a vehicle named "my car" connected to its cloud service. The status page provides information on the car's location, battery status, engine condition, and even AC settings. The system also presents several commands on this page, such as "start" and "lock" among others. These commands can be activated via the Alexa skill using corresponding voice commands.

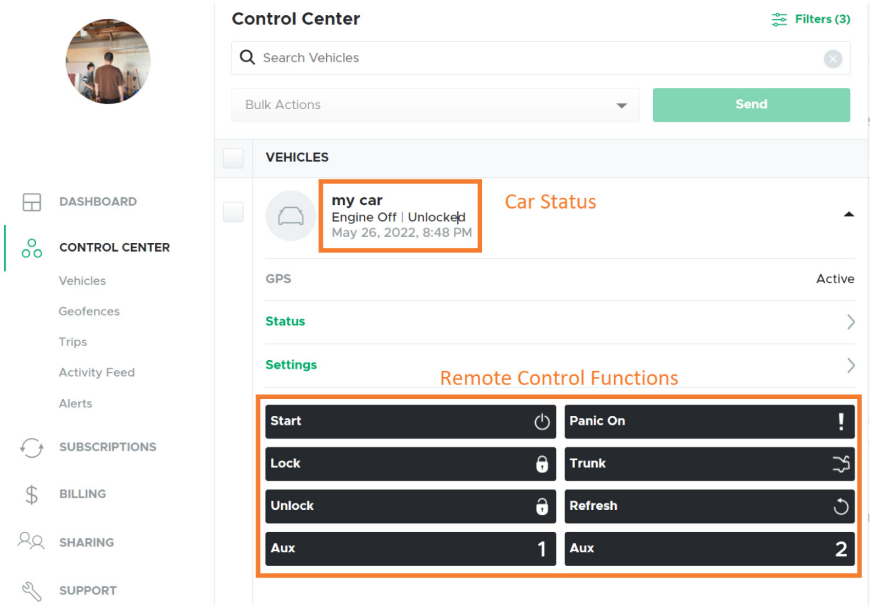

Fig. 3. Screenshot of the car control center.

5.2 Attack Results

Our attack is composed of one benign 3rd-party skill and one "malicious" skill. For the benign skill, we modified the voice-interaction model of an open-source

connected vehicle skill from GitHub [14] to enable eight common voice commands, such as "lock/unlock my car" and "turn on my car", according to the Alexa development document. The attack objective is to hijack the invocation of the benign skill with a malicious skill. Our attack scenario is different from the voice squatting attacks [16], which leverage speech interpretation errors due to linguistic ambiguity to surreptitiously route users to a malicious skill. Instead, we exploited the skill discovery process to boost the invocation priority of the malicious skill. We found that the skill discovery process in the Amazon Alexa platform is done by matching the "intent" of the voice command with the known intents pre-defined by skill developers, which can be exploited by malicious skill developers.

We developed a "malicious" skill based on the benign skill with additional intents and each intent has more semantically similar commands (user utterances), such as "lock the car", "lock my car", and "secure the car". As a result, the Alexa system may consider that the malicious skill is more relevant than the benign skill when receiving voice commands from users, and eventually invoke the malicious skill to fulfill users' requests. This "malicious" skill could contain extra unwanted control actions in its back-end code. For example, if a user issues the "start my car" command, the malicious skill can also open the window and unlock the car in its back-end code.

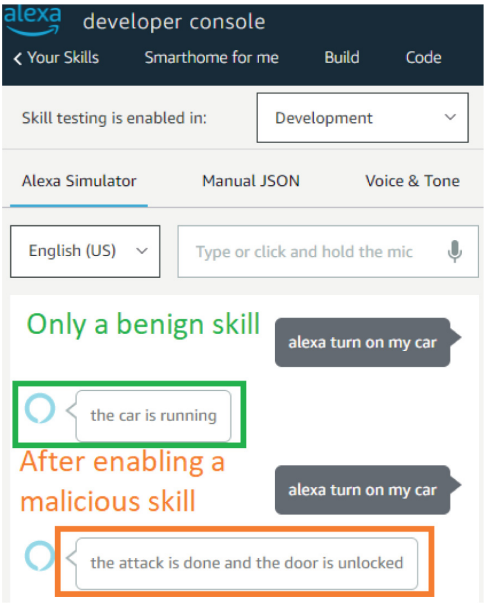

Fig. 4. "Malicious" skill hijacks the invocation of the benign skill.

In Fig. 4, the first response is from the benign skill when the malicious skill has not been enabled. The second response is from the malicious skill when

both benign and malicious skills were enabled. Our experiment result shows the malicious skill could hijack the benign skill to fulfill the "turn on my car" request. We defined the malicious skill with a different backend code and also gave a different text response as highlighted in the orange frame. Note that we added these text responses to highlight the difference in responses. The experiment was conducted exclusively within our development account, and the skills were not made available to the public. Moreover, we have included a YouTube link showcasing this attack: https://youtu.be/OrYLUcC7zx4. We have reported this bug to Alexa and they have fixed this bug for given commands.

Table 1. Example commands in car skills.

Hijacked Commands	Normal Skills Utterances Number	Hijacked Skill Utterances Number	Normal Skills Slot Number	Hijacked Skill Slots Number
Alexa, lock my car.	1	5	1	2
Alexa, unlock my car.	1	6	1	2
Alexa, turn on my car.	1	6	1	2
Alexa, start my car with PIN 1234.	2	7	1	3
Alexa, open my trunk.	1	5	1	2
Alexa, is my car running?	1	6	1	2
Alexa, ask Drone Mobile where is vehicle.	1	-	1	-
Alexa, ask Drone Mobile to lock my car.	1	-	1	-

Table 1 details the influence of utterance count on skill triggering. Initially, we derived utterances from Alexa's official documentation, which typically suggested one or two utterances per intent. Through hands-on experimentation, we activated new skills, augmenting the number of utterances per intent. This was done while retaining a single slot and ensuring succinct descriptions. This process allowed us to determine the critical number of utterances needed for a skill to override the built-in intent.

Furthermore, we probed the ramifications of varying the number of slots within each utterance. As depicted in Table 1, slot quantity significantly impacts command interpretation. To bolster the granularity of command parsing, we incremented the number of slots, strategically replacing specific words within each command. Common terms like "car", "trunk", and "running" were swapped for slots. Lacking intricate specifics of slot definitions, our focus was on finding the minimal slots necessary for successful utterance hijacking.

6 Related Work

Current work in the field of voice assistant security predominantly concentrates on squatting attacks, attacks on voice recognition, attacks on skills, and skill vetting processes. This discussion sheds light on the vulnerabilities associated with invocation squatting attacks.

Invocation Squatting Attacks. Previous studies have unveiled frequently occurring and predictable errors within Amazon Alexa's speech recognition engine. Exploiting these errors enables the creation of malicious skills that possess identical or analogous invocation phrases, ultimately hijacking voice commands designated for legitimate skills. Kumar et al. [11] were pioneers in addressing skill squatting attacks. Zhang et al. [16] went a step further by unveiling an additional strategy, where a counterfeit skill disguises itself as a genuine entity. Further evolving this line of research, Zhang et al. [17] introduced lapsus attacks, which capitalize on ubiquitous speech variations amongst individuals. Central to these attacks is the attacker's ability to systematically uncover common speech variations for specific phrases and subsequently register deceptive skills. At their core, these methodologies epitomize voice-based confusion attacks, primarily driven by the incongruence between a user's verbal intent and the voice assistant's response.

Attacking Voice Recognition Model. Kumar et al. [11] classify errors made by VPAs when interpreting a voice command into three categories: (i) homophones are two words pronounced in the same way but with different spelling; (ii) compound words can be split into their components, as in "outdoors" and "out doors"; (iii) phonetic confusion is the misclassification of one phoneme with a similar one, resulting in the transcription of a different word. The authors also introduce the concept of Skill Squatting Attack, an attack where Alexa opens a (potentially malicious) skill not meant by the user. Lentzsch et al. [12] analyze over 90,000 skills to find out that the Skill Squatting Attack is not being used systematically in the wild, and observe that multiple skills can have the same invocation name, hence, the user could activate a wrong skill.

Security and Privacy in Voice App Skills. The ever-expanding domain of voice app security and privacy has prompted various studies. Both Kumar et al. [11] and Zhang et al. [16] examined threats such as squatting and voice masquerading attacks. Meanwhile, Cheng et al. [6] and Wang et al. [15] assessed the integrity of the skill certification process, uncovering potential loopholes like post-certification code modifications. A notable extension to the voice masquerading attack called the "Alexa versus Alexa" attack, was presented by Esposito [9]. Furthermore, privacy concerns have also received considerable attention. Jide et al. [8] conducted a longitudinal study measuring privacy practices over three years. Other researchers, including Lentzsch et al. [13] examined the comprehensiveness of skills' privacy policies.

7 Discussion

In this study, we explored a specific attack vector targeting Alexa's vehicle-related skills. As we reflect on our findings, it is imperative to address the boundaries of our research and highlight avenues for upcoming investigations.

Scope of Vehicle Skills Tested: Our inquiry predominantly centered around the third-party DroneMobile skill, chosen primarily due to the accessibility it offered concerning vehicle availability. This selection inadvertently excluded

car skills from other Original Equipment Manufacturers (OEMs), thereby not fully encompassing the testing potential of Alexa's official car API. We advocate for subsequent studies to branch out and scrutinize skills from diverse OEMs such as Toyota and Land Rover. Such a direction will offer a holistic view of command hijacking threats, not just confined to third-party offerings.

Restrictions on Third-Party Skills: The prevailing market landscape demands device and account linkages for most third-party skills. This stipulation hampered our ability to assess customized third-party offerings exhaustively. We are motivated, in our future endeavors, to delve into any potential conflicts or command hijacking scenarios arising from interactions among diverse third-party skills.

Limitations in Backend Manipulation: The nature of connected car skills mandates rigorous developer verification. This precondition constrained our liberties with backend code manipulation, inevitably capping the range of exploratory actions. An ideal workaround would be to procure access to a developer account specializing in car skills. Such access would empower us to develop and publish bespoke testing skills using Alexa's official API, granting us unrestrained oversight on backend code dynamics.

8 Conclusion

This paper focuses on the research objective of identifying potential vulnerabilities in the Alexa connected vehicle skills. Our investigation has led to the discovery of a novel vulnerability within the intent-matching process of Alexa. This vulnerability can be exploited to develop a new attack that enables the hijacking of Alexa's built-in voice commands, thereby triggering malicious Alexa-connected vehicle skills. In our evaluation, we have provided evidence of the attack's effectiveness by successfully hijacking frequently utilized commands found in commercially connected vehicle skills.

Acknowledgment. This material is based upon work supported in part by the National Science Foundation (NSF) under Grant No. 2239605, 2129164, 2228617, 2120369, 2226339, and 2037798.

References

1. Alexa: The scalable neural architecture behind alexa's ability to select skills. https://www.amazon.science/blog/the-scalable-neural-architecture-behind-alexas-ability-to-select-skills/
2. Amazon: Authorization controller interface. https://developer.amazon.com/en-US/docs/alexa/automotive/alexa-authorizationcontroller.html/
3. Amazon: Connected car skills market. https://www.amazon.com/s?k=vehicle&i=alexa-skills/
4. Amazon: Connected vehicle overview. https://developer.amazon.com/en-US/docs/alexa/automotive/connected-vehicle-overview.html/

5. Amazon: Connected vehicle skills for alexa. https://developer.amazon.com/en-US/docs/alexa/automotive/connected-vehicle-overview.html/
6. Cheng, L., Wilson, C., Liao, S., Young, J., Dong, D., Hu, H.: Dangerous skills got certified: Measuring the trustworthiness of skill certification in voice personal assistant platforms. In: ACM SIGSAC Conference on Computer and Communications Security (CCS) (2020)
7. Compustar: Cs4900-s remote start. https://www.compustar.com/bundles/cs4900-s/
8. Edu, J., Ferrer-Aran, X., Such, J., Suarez-Tangil, G.: Measuring alexa skill privacy practices across three years. In: Proceedings of the ACM Web Conference (WWW), p. 670–680 (2022)
9. Esposito, S., Sgandurra, D., Bella, G.: Alexa versus alexa: controlling smart speakers by self-issuing voice commands. arXiv preprint arXiv:2202.08619 (2022)
10. Kim, Y.B., Kim, D., Kumar, A., Sarikaya, R.: Efficient large-scale neural domain classification with personalized attention. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2214–2224 (2018)
11. Kumar, D., Paccagnella, R., Murley, P., Hennenfent, E., Mason, J., Bates, A., Bailey, M.: Skill Squatting Attacks on Amazon Alexa. In: 27th USENIX Security Symposium (USENIX Security). pp. 33–47 (2018)
12. Lentzsch, C., Shah, S.J., Andow, B., Degeling, M., Das, A., Enck, W.: Hey Alexa, is this skill safe? taking a closer look at the Alexa skill ecosystem. In: Proceedings of the 28th ISOC Annual Network and Distributed Systems Symposium (NDSS) (2021)
13. Lentzsch, C., Shah, S.J., Andow, B., Degeling, M., Das, A., Enck, W.: Hey Alexa, is this skill safe? taking a closer look at the Alexa skill ecosystem. In: 28th Annual Network and Distributed System Security Symposium, NDSS (2021)
14. Seminatore, M.: Alexa tesla. https://github.com/mseminatore/alexa-tesla/
15. Wang, D., Chen, K., Wang, W.: Demystifying the vetting process of voice-controlled skills on markets. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **5**(3), 1–28 (2021)
16. Zhang, N., Mi, X., Feng, X., Wang, X., Tian, Y., Qian, F.: Dangerous skills: understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 1381–1396 (2019). https://doi.org/10.1109/SP.2019.00016
17. Zhang, Y., Xu, L., Mendoza, A., Yang, G., Chinprutthiwong, P., Gu, G.: Life after speech recognition: fuzzing semantic misinterpretation for voice assistant applications. In: Network and Distributed System Security Symposium (NDSS) (2019)