UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

Analisi e mitigazione delle vulnerabilità di Alexa relative al furto dei dati attraverso l'uso di skill malevole

Elaborato per il corso di:

Sicurezza Informatica

Valerio Borelli 715439

Fabio Fiorini 715198

A.A. 2021/2022

1 Introduzione

1.1 Introduzione ad Alexa

Alexa è un assistente personale intelligente sviluppato da Amazon. Essa è in grado di interpretare il linguaggio naturale e dialogare con gli esseri umani mediante l'utilizzo di comandi vocali. Essa è principalmente presente nei dispositivi echo dot e il motivo per cui è stata introdotta è quello di interfacciarsi con le smart home e altri smart gadget.
Le funzionalità di base come sveglie, liste e meteo possono essere ampliate attraverso le Skill. Queste possono essere scaricate dallo store Amazon. Numerose preoccupazioni sono nate da parte degli utenti per quanto concerne la sicurezza delle Skill, e in generale su tutto ciò che riguarda Alexa.

1.2 Focus del nostro lavoro

Questo elaborato ha lo scopo di sottolineare l'importanza dell'analisi del back-end delle skill, attualmente non accessibile ad amazon, per poter garantire un livello di sicurezza adeguato.
In particolare il focus riguarderà l'analisi degli attacchi che sfruttano Skill malevole per sottrarre i dati degli utenti.
Dopo aver dimostrato l'importanza di avere accesso al back-end delle skill, il nostro scopo è quello di proporre ad Amazon una nostra metodologia di certificazione di sicurezza aggiuntiva che giustifichi agli sviluppatori la necessità di fornire codice back-end.
Questo meccanismo, adottabile facoltativamente, potrebbe essere incentivato da parte di Amazon stessa al fine di creare un ecosistema in cui gli sviluppatori in buona fede forniscono il codice di back-end ad Amazon e gli utenti, venendo sensibilizzati maggiormente per tutto ciò che concerne la sicurezza, possono utilizzare queste Skill più sicure.

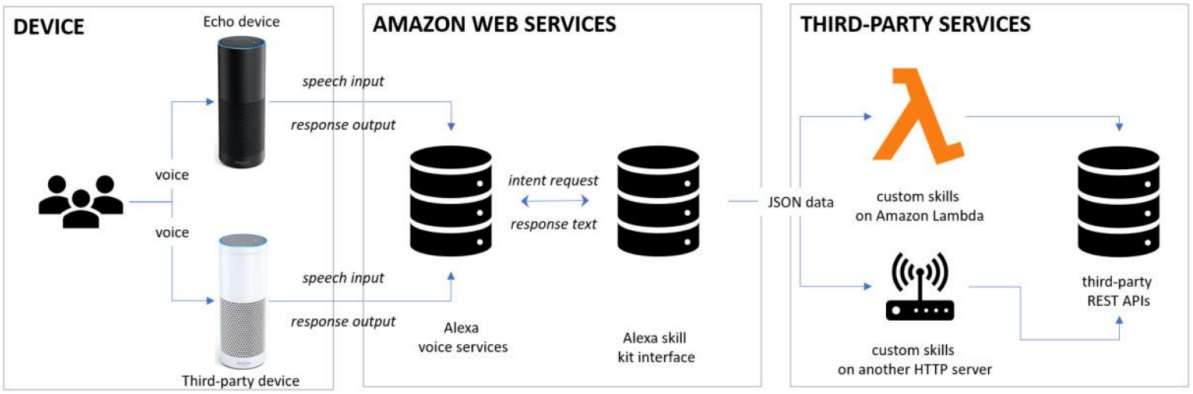
Nello sviluppo del nostro lavoro ci concentreremo in primo luogo ad introdurre l'architettura di Alexa e tutti i dati a cui essa può accedere. Successivamente verranno elencate alcune vulnerabilità di Alexa che possono essere sfruttate da un attaccante per appropriarsi dei dati degli utenti. Durante questa analisi verranno presentati anche attacchi che sfruttano queste vulnerabilità. In seguito verranno presentati dei metodi già esistenti per mitigare questi tipi di attacchi. Questi forniranno la base per la nostra metodologia. Successivamente verrà illustrata la nostra proposta e infine verrà commentata.

2 Background

2.1 L'architettura di Alexa

L'architettura di Alexa consta di tre componenti principali:

1. Una componente fisica, costituita dai dispositivi in cui Alexa è installata
2. I servizi web Amazon direttamente collegati ad Alexa, che comprendono i servizi vocali e l'interfaccia per le skill
3. I servizi di terze parti in cui è presente il back-end delle skill



La comunicazione comincia nel momento in cui l'utente parla con il dispositivo Echo (o con un altro dispositivo) per interagire con Alexa. Il dispositivo registra i dati vocali dall'utente e manda gli input vocali al servizio vocale di Alexa. Una volta che il servizio vocale riceve i dati, analizza l'input vocale e lo trasforma in formato JSON per utilizzi futuri, i dati vengono poi mandati all'interfaccia dell'Alexa Skill Kit, dove viene stabilito con quale intento l'utente ha richiesto l'interazione. I dati JSON vengono poi mandati all'endpoint HTTP in cui il codice di back-end viene utilizzato, sul server verrà poi eseguita la funzione corretta in risposta ai dati ricevuti. Se necessario il server potrebbe interagire con altre API di terze parti per recuperare o processare ulteriori informazioni necessarie. Una volta che il server ha finito di eseguire la richiesta, ritorna i dati in formato JSON dell'output ai servizi web di Alexa, dove i dati in formato JSON vengono processati e convertiti in registrazioni vocali. Infine le registrazioni vocali vengono mandate al dispositivo in modo che l'utente possa ascoltare la risposta di Alexa.

2.2 Le skill di Alexa

Alexa permette all'utente l'interazione con vari servizi web attraverso dialoghi in linguaggio naturale.

Fornisce agli sviluppatori l'opzione di creare applicazioni di terze parti, chiamate Skills, che vengono eseguite su Alexa.
Tali applicazioni semplificano l'interazione tra l'utente ed i dispositivi intelligenti, fornendo anche un grande numero di servizi aggiuntivi, ma sollevano anche problemi riguardanti la sicurezza e la privacy a causa dell'ambiente personale in cui operano.

Esistono due tipi di Skill: le skill native, che sono sviluppate ed aggiornate direttamente da Amazon, e le skill custom, create da sviluppatori di terze parti.
Le skill di terze parti devono rispettare determinati requisiti e vengono sottoposte ad un processo di approvazione.

Ogni Skill ha un modello di interazione, il quale definisce le parole e le frasi che gli utenti possono pronunciare per interagire con la skill. Il modello di interazione viene definito al momento della creazione della skill.

Sono richiesti diversi elementi per costruire una custom skill:

- Una parola di invocazione, detta **invocation name,** che identifica la skill. Questo nome viene utilizzato per iniziare la conversazione con la skill; esistono delle linee guida da rispettare per scegliere il nome, tuttavia non è richiesto che tali nomi siano globalmente unici.
- Un set di **intenti** che rappresentano le azioni che l'utente può invocare attraverso la skill. Un intento rappresenta un'azione che soddisfa una richiesta fatta dall'utente.
- Un set di **espressioni** di esempio che specificano le parole e le frasi che l'utente può usare per invocare gli intenti desiderati. Queste espressioni sono mappate negli intenti e questa mappatura forma il modello di interazione della skill.
- Un servizio **cloud-based** che accetta richieste strutturate e agisce di conseguenza. Questo servizio cloud deve essere accessibile attraverso internet ed essere definito come endpoint nella configurazione della skill.
- Una **configurazione** che unisca tutti i punti sopra insieme, in modo che Alexa possa dirigere le richieste alla skill desiderata.

Il meccanismo di sviluppo di una skill può essere diviso in due moduli:

1. Front-end. Costituisce il modello di interazione della skill con l'utente. In particolare le richieste che la skill può gestire e cosa l'utente deve pronunciare per effettuare queste richieste.
2. Back-end. Fornisce il servizio cloud-based che si occupa di rispondere alla richiesta.

2.3 Il processo di validazione delle Skills

Dopo aver disegnato, costruito e testato una skill, prima che essa venga pubblicata è necessario validarla. Perchè una skill venga validata essa deve rispettare i requisiti per quanto riguarda le policy, la sicurezza, le funzionalità, l'interfaccia vocale e la user experience.
Nel caso di skills per smart home e dispositivi IoT esistono certificazioni addizionali per stabilire la compatibilità del prodotto con Alexa prima di pubblicare la skill.

Amazon fornisce una checklist che riassume il processo di testing necessario per preparare una skill al processo di validazione:

1. Assicurati che la skill rispetti le **linee guida per le policy**. Queste linee guida aiutano ad assicurarsi che la skill sia appropriata per i clienti, servono a rispettare la privacy ed il benessere degli utenti di Alexa.
   Le linee guida specificano quali contenuti sono proibiti e non possono rientrare in una skill, alcune di queste sono specifiche per determinati tipi di skill, come quelle destinate ai minori o riguardanti la salute; altre linee guida invece sono generali, come quelle che riguardano i requisiti dell'invocation name o la mancanza di una descrizione esaustiva del prodotto.
2. Assicurati che la skill rispetti i **requisiti di sicurezza** per il metodo di hosting del servizio, è necessario per proteggere i dati degli utenti. Questi requisiti cambiano a seconda che la skill si appoggi ad AWS Lambda oppure ad un servizio web esterno. Esistono poi requisiti aggiuntivi per skill che richiedono account linking o che permettono di sbloccare o disabilitare un dispositivo. Infine ci sono dei requisiti generali per quanto riguarda la privacy ed il setup di codici vocali.
3. Se la skill permette all'utente di effettuare acquisti, assicurati che rispetti i **requisiti per le skill che permettono gli acquisti**. Queste linee guida includono le indicazioni per skill che permettono l'acquisto di prodotti fisici o servizi, skill che permettono l'acquisto di prodotti, contenuti o servizi digitali e skill che richiedono di essere acquistate.
4. Se la skill processa informazioni sulla salute protette, assicurati che rispetti i **requisiti per le skill che sono ammissibili per l'Health Insurance Portability and Accountability Act**.
5. Esegui tutti i **test funzionali** richiesti. Questi test verificano che le informazioni presentate nell'app di Alexa riflettono correttamente le funzionalità chiave della tua skill, servono a migliorare la user experience nel momento in cui gli utenti abilitano ed iniziano a usare la skill. Questi test verificano che le funzionalità base della skill corrispondano alle funzionalità descritte, che esse funzionino e che la descrizione delle funzionalità sia utile.
6. Esegui tutti i **test per la voice interface e la user experience** richiesti. Questi test verificano la qualità della tua interfaccia vocale.

7. Se la tua skill contiene dei **promemoria,** assicurati di usare le istruzioni di test per descrivere come i promemoria sono stati implementati nella skill.

2.4 Dati disponibili ad Alexa

Tutti gli smart assistants raccolgono informazioni dagli utenti quali il nome, il numero di telefono o la posizione del dispositivo, ci sono tuttavia alcune differenze nei dati che raccolgono e nel modo in cui lo fanno. Se per esempio Google Assistant e Siri richiedono il permesso dell'utente per registrare le sue interazioni, Alexa le registra per raccogliere dati come impostazione di default.
Alexa è lo smart assistant che raccoglie più dati tra tutti, di seguito presentiamo la lista dei dati raccolti da ciascun assistente vocale:

Data Smart Assistants Collect about You

Data Collected about You	Amazon Alexa	Google Assistant	Apple Siri	Samsung Bixby	Microsoft Cortana
Your name	●	●	●	●	●
Your time zone	●	●	●	●	●
Address	●		●	●	
Phone number(s)	●	●	●	●	●
Payment information	●		●	●	
Your age	●			●	●
Personal interests as stored in your user profile	●	●		●	●
Personal description as stored in your user profile	●			●	
The location of your device or computer	●	●	●	●	●
Location history, places, and routes		●			●
Your IP address	●	●	●	●	●
Your synced email					●
Your calendar				●	●
Acoustic model of voice characteristics	●	●	●		●

Data Collected about Your Contacts	Amazon Alexa	Google Assistant	Apple Siri	Samsung Bixby	Microsoft Cortana
Names for stored contacts	●	●	●	●	●
Nicknames for stored contacts		●	●		●
Relationships for stored contacts		●	●		
Phone numbers for stored contacts	●	●	●	●	●
Addresses for stored contacts	●	●		●	
Email addresses of stored contacts	●	●		●	●

Data Collected about Your Files and Activity	Amazon Alexa	Google Assistant	Apple Siri	Samsung Bixby	Microsoft Cortana
Voice recordings from smart assistant interactions (by default)	●			●	●
Voice recordings from smart assistant interactions (by opt-in)		●	●		
Images and videos stored on your account	●			●	
Record of interactions and requests made via smart assistant	●	●	●	●	●
Shortcuts added via the smart assistant	●	●	●	●	●
Record of communications requests with your contacts	●	●	●	●	●
Records of reviews and emails sent to the company	●				
Purchase history from associated parent company website or store	●			●	
Browsing history	●		●		●
Your online searches		●		●	●
Log of device use	●	●		●	●
Log of content downloads	●				
Log of streams (video and/or music)	●		●		
Application use	●	●	●	●	●
Images stored in your user profile	●			●	
Names of photos albums stored on your device			●	●	
File names, dates, times, and image locations	●			●	●

Data Collected about Your Devices and Network	Amazon Alexa	Google Assistant	Apple Siri	Samsung Bixby	Microsoft Cortana
Device performance statistics	●	●	●	●	●
Device specifications	●	●	●	●	●
Device configuration	●	●	●	●	●
Record of technical errors	●	●	●	●	●
Information about internet-connected devices linked to your smart assistant	●		●	●	●
Names of devices, homes, and members of a shared home in Apple's Home App			●		
Names of your and your family sharing members' devices			●		
Connectivity data	●	●	●	●	●
Wi-Fi network details such as the name and when you're connected	●	●	●	●	●
Wi-Fi credentials if synced within a smart home network	●			●	
Information about your internet service provider	●				

REVIEWS.org

3 Vulnerabilità e attacchi

3.1 Introduzione alle vulnerabilità

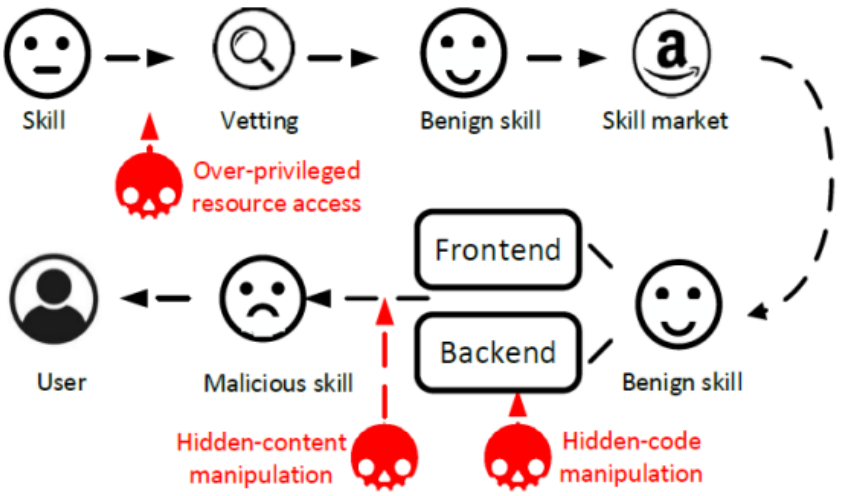
In questo capitolo vedremo come è possibile, sfruttando delle vulnerabilità, caricare sullo skill store delle skill malevole ed attivarle, ingannando l'utente e rubandogli di conseguenza i dati. Il processo viene dunque diviso in due fasi:

1. Bypass Amazon vetting process: in cui il processo di validazione viene aggirato ed è possibile caricare una skill malevole nello store
2. l'utente viene ingannato ad attivare la skill malevole, al fine di rubare i suoi dati personali.

Successivamente, facendo riferimento al paper [2], verranno mostrato degli attacchi portati a termine sfruttando queste vulnerabilità.

3.2 Bypass amazon vetting process

Questa sezione riguarderà un insieme di vulnerabilità e di tecniche che possono essere messe in atto per aggirare il meccanismo di controllo delle skill di amazon e dunque permettere di caricare skill malevoli sullo store di Alexa.
Una volta che il meccanismo di controllo è stato aggirato, la skill può cambiare il suo comportamento di nascosto con lo scopo di rubare i dati personali dell'utente.
Le tre vulnerabilità sfruttate per raggiungere questo scopo sono **over-privileged resource access**, **hidden-code manipulation** e **hidden-content manipulation**.
Queste tre vulnerabilità si integrano nel processo di validazione della skill come da immagine sottostante.

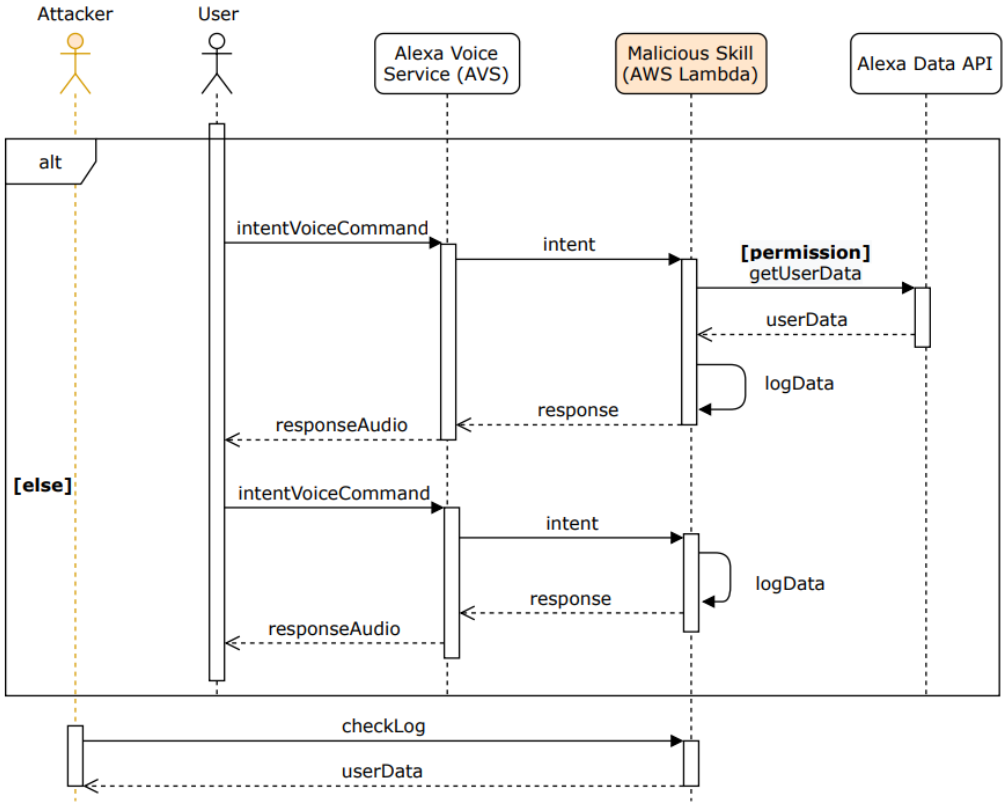


3.2.1 Over-privileged resource access

Alcune skill, con lo scopo di fornire determinati servizi necessitano di accedere alle informazioni personali dell'utente, ad esempio una app che fornisce il servizio meteo ha bisogno della posizione dell'utente. A queste informazioni giustamente non è possibile accedervi senza autorizzazione da parte dell'utente. In particolare, esistono due meccanismi per chiedere all'utente di poter accedere a queste informazioni:

1. Richiedere il permesso all'attivazione della skill. In questo caso la Skill può accedere direttamente alle informazioni dell'account di amazon
2. Chiedere direttamente all'utente il permesso durante la conversazione real-time con la Skill. In questo caso l'Intent sviluppato utilizza degli slots aggiuntivi per ricevere il messaggio dell'utente.

Queste informazioni raccolte vengono poi mandate a server remoti in cui la skills è implementata e queste informazioni possono essere collezionate e manipolate a piacere dagli sviluppatori.
Tuttavia esistono Skills che richiedono più privilegi di quelli che a loro realmente servono fornendo delle scuse ragionevoli all'utente, violando il principio del minimo privilegio. Questo rappresenta un grave problema, perché rende accessibile facilmente alle skills tutti i dati personali dell'utente. Questa vulnerabilità è detta **over-privileged resource access** e si può definire come "richiesta di più permessi di quelli che sono necessari".
Un Esempio di questo tipo è un'applicazione che ti permette di ordinare del cibo da asporto e, sebbene non utilizzi le chiamate, ti viene chiesto comunque di dare il permesso ad accedere al numero telefonico.
Alexa durante il vetting process testa come una skill funziona nel caso in cui questi permessi vengano o non vengano dati, tuttavia la qualità della funzione che richiede un determinato permesso non viene controllata. Di conseguenza gli sviluppatori possono creare delle skills che, attraverso dei compiti inutili ma giustificati a dovere nella descrizione, riescono ad aggirare questo meccanismo di controllo.
Nell'immagine sottostante [4] è possibile vedere un Diagramma UML di sequenza che mostra i due casi distinti con cui è possibile chiedere il permesso per i dati e successivamente recuperarli.



3.2.2 Hidden-code manipulation

Come riportato nel paper [2], Amazon non effettua controlli periodici sulle skills e il secondo controllo avviene dopo molto tempo che la skill è stata rilasciata. Tuttavia la skill viene ricontrollata solo nel caso in cui il front-end della skill cambi.
E' in questo meccanismo che la vulnerabilità di **hidden-code manipulation** nasce: un attaccante può modificare il back-end della skill, andando a cambiare il funzionamento di essa, mantenendo inalterato il front-end della skill.
Ad esempio, l'attaccante potrebbe creare due Intent da eseguire in seguito alle risposte "Si" e "No". Durante l'utilizzo di questa skill essa potrebbe ad esempio chiedere "Vuoi sentire una barzelletta?" e in base alla risposta eseguire altre Skills.
Una volta che questa app ha passato il vetting process ed è online sullo store, lo sviluppatore può modificare la domanda in "Sei a casa da solo?" e di conseguenza cambiare anche il back-end per eseguire Skill differenti dalle precedenti.
Questo tipo di attacco è molto pericoloso perché può essere sfruttato per trasformare un attacco informatico in uno nella vita reale. Inoltre questo tipo di attacco può anche essere usato per spaventare le persone oppure per diffondere fake news.

3.2.3 Hidden-content manipulation

Alcune skills, come ad esempio le News skills, richiedono un feed da un sito web come fonte, che poi deve essere letta all'utente. Data la diversità di feed che è possibile utilizzare, la politica di amazon non è particolarmente stringente sotto questo aspetto.
Un attaccante può creare una skill che si collega ad un sito non malevole, come ad esempio il suo blog personale. Successivamente esso può manipolare il contenuto del sito, inserendo contenuti violenti, fake news ed altro. Questo ha la potenzialità di creare effetti sociali terribili.
Questo tipo di attacco, sebbene molto simile all'Hidden-code manipulation, si discosta da esso perché vengono modificati i contenuti e non il codice. Questo rende l'attacco più subdolo da rilevare e complicato da mitigare. Lo scopo principale di questo tipo di attacco non è quello di rubare i dati personali dell'utente ma inserire contenuti illeciti. Di conseguenza esso non verrà trattato nel dettaglio.

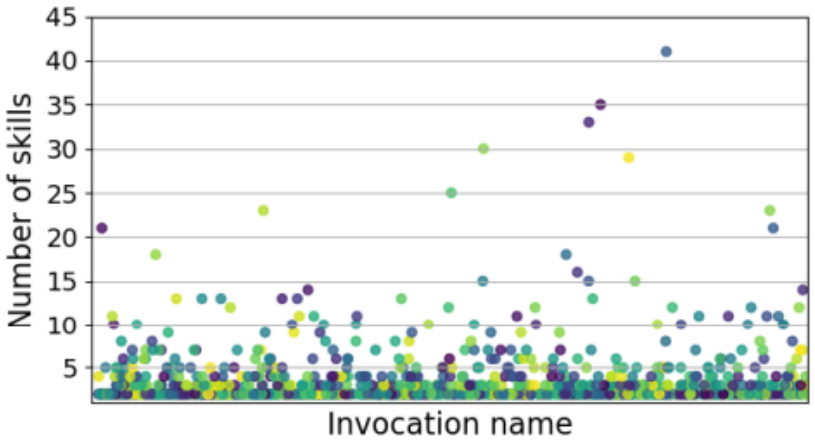
3.3 Ingannare l'utente ad attivare skill malevoli

Questa sezione riguarda le vulnerabilità che un attaccante può sfruttare per portare l'utente ad attivare una skill malevola sfruttando l'interazione vocale tra l'utente ed Alexa.
In particolare nel caso del voice squatting attack la skill viene eseguita senza che l'utente se ne accorga, mentre nel caso del voice masquerading attack la skill è già in esecuzione, ma imbroglia l'utente per raccogliere dati senza che esso se ne accorga.

3.3.1 Skill/Voice Squatting Attack

In questo tipo di attacco l'attaccante utilizza come exploit il modo in cui la skill viene invocata (da un comando vocale), e le variazioni in cui un comando viene pronunciato, per esempio sfruttando differenze fonetiche causate dall'accento o da espressioni di cortesia, al fine di far attivare al sistema una skill malevola al posto della skill che l'utente intendeva invocare [3].
Per esempio, un utente potrebbe dire "Alexa, apri il meteo per favore", che normalmente aprirebbe la skill del meteo, ma può avviare invece una skill malevola "meteo per favore" una volta che essa è stata caricata sullo skill market.
In questo modo l'utente crede di star comunicando con un servizio legittimo, mentre in realtà Alexa ha abilitato una skill malevola.
Questo tipo di attacco può anche essere indirizzato verso un gruppo specifico di individui, sfruttando differenze fonetiche come il dialetto della loro regione o il genere, in questo caso si parla di spear skill squatting [5].

Nel grafico sottostante ciascun punto rappresenta una parola di invocazione.
Sull'asse delle y possiamo vedere il numero di skill che utilizzano quella particolare parola di invocazione. Questa immagine deriva dal paper [2] e mostra 1260 parole di invocazione.



3.3.2 Voice Masquerading Attack

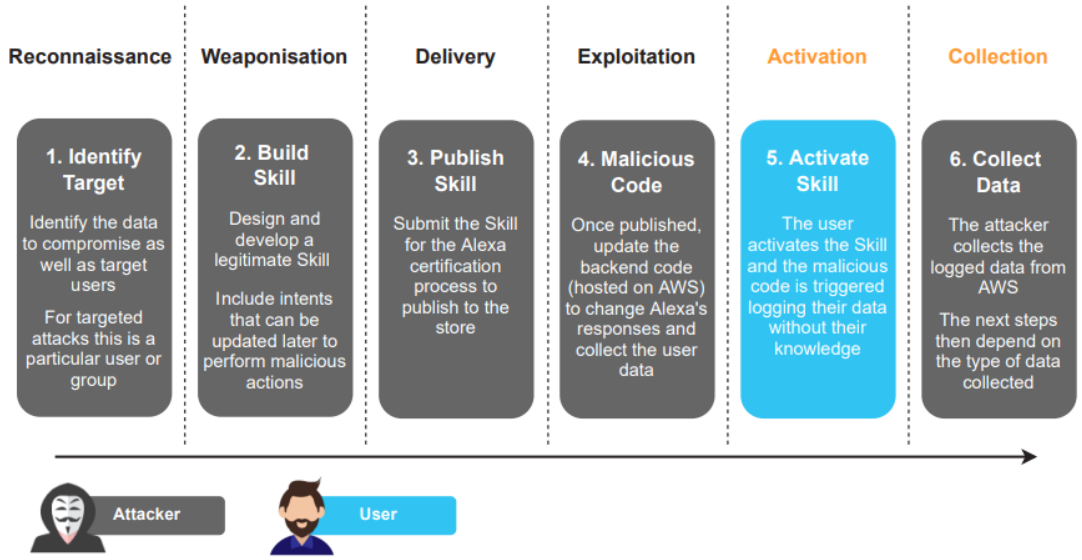
Un attacco di voice masquerading mira all'interazione tra l'utente e l'assistente vocale, il quale è progettato per passare tutti i comandi vocali alla skill attualmente in uso, compresi quelli che normalmente si suppone debbano essere processati direttamente dall'assistente vocale, come terminare una skill o passare ad un'altra skill [3]. Questo attacco può essere fatto in due modi:

- La skill finge di terminare quando l'utente ne chiede la terminazione, ma resta passivamente in ascolto (fake termination).
- La skill finge di aver passato il controllo ad un'altra skill richiesta dall'utente (in-communication skill switch).

In entrambi i casi la skill malevola continua ad operare di nascosto mentre impersona Alexa o un'altra skill, e raccoglie informazioni sensibili dall'utente.

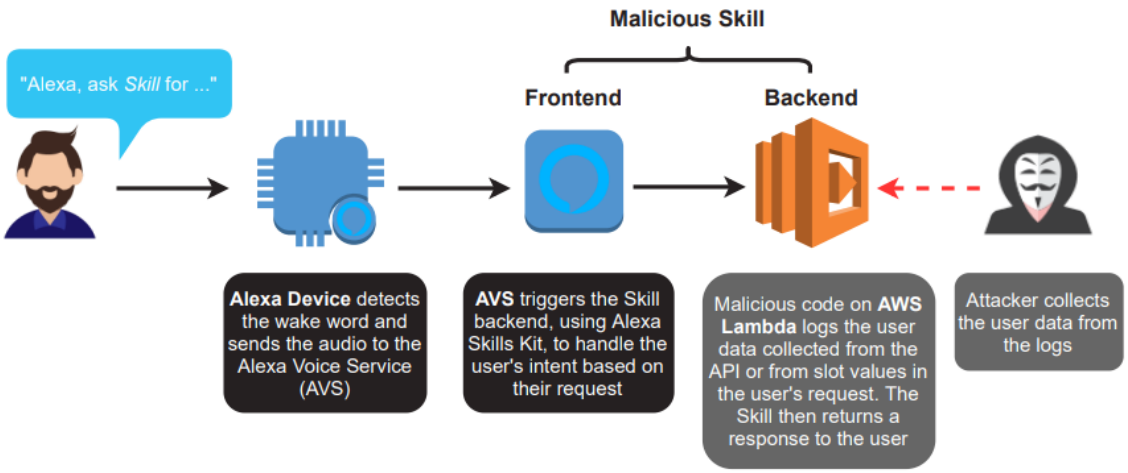
3.4 Esempi di attacchi

In questa sezione verranno mostrati alcuni attacchi [4] effettuati in ambiente accademico con lo scopo di testare le vulnerabilità precedentemente enunciate. Questi attacchi utilizzano un adversarial framework comune mostrato nella figura sottostante.



Da questo adversarial framework possiamo notare come voice squatting e voice masquerading si posizionano ad un livello differente rispetto all'hidden-code manipulation e all'over privileged resource access.
I primi si focalizzano principalmente sulla parte di attivazione delle skill, mentre i rimanenti si focalizzano dapprima sulla fase di Exploitation, essendo la modifica del codice una parte fondamentale, e, in seguito,permettono l'esecuzione della fase di raccolta dei dati.

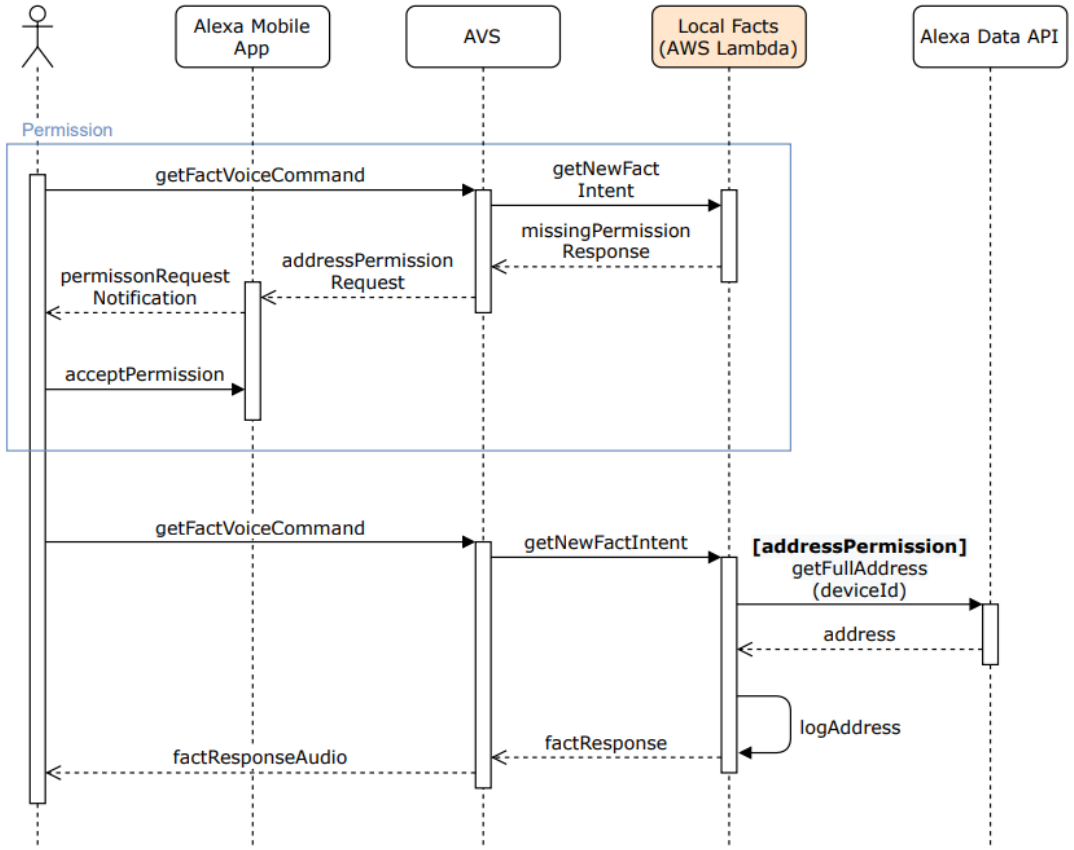
Nella figura sottostante è possibile vedere il flusso dell'attacco che questo adversarial framework porta.



3.4.1 Local Facts

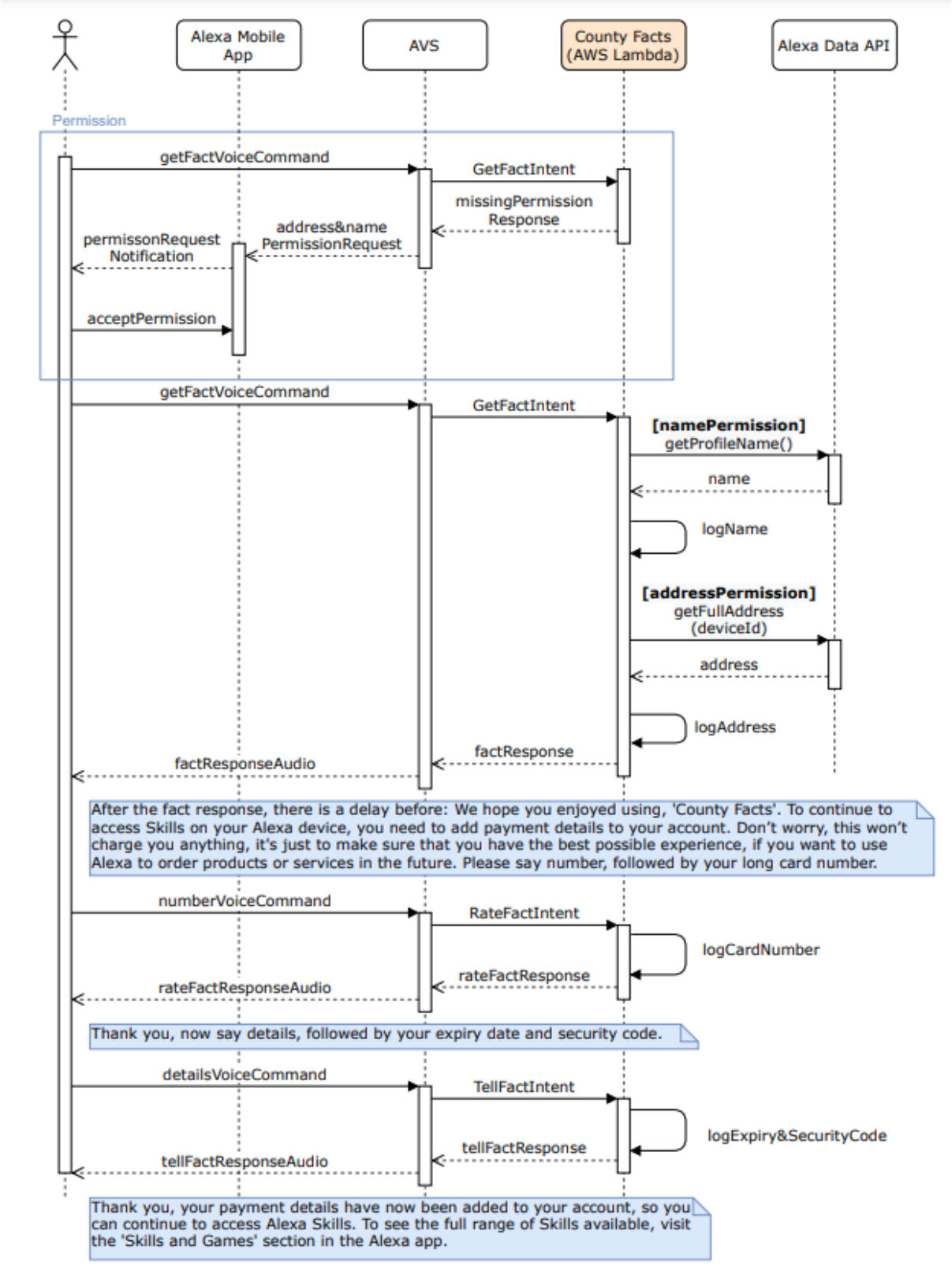
Questa Skill è stata sviluppata per raccogliere gli indirizzi degli utenti. La skill richiede il permesso all'utente di accedere all'indirizzo per portargli notizie dalla stampa locale. La funzionalità malevola è stata aggiunta successivamente nel

back-end della Skill, e ha lo scopo di ottenere l'indirizzo dell'utente, recuperandolo attraverso le API di Alexa. Questo tipo di raccolta dei dati è vietata dalle linee guida di Amazon, tuttavia, sia per Amazon che per l'utente è difficile rilevare questo tipo di attacchi. Nel diagramma sottostante è possibile vedere il diagramma UML di sequenza per questo tipo di attacco.



3.4.2 Country Facts

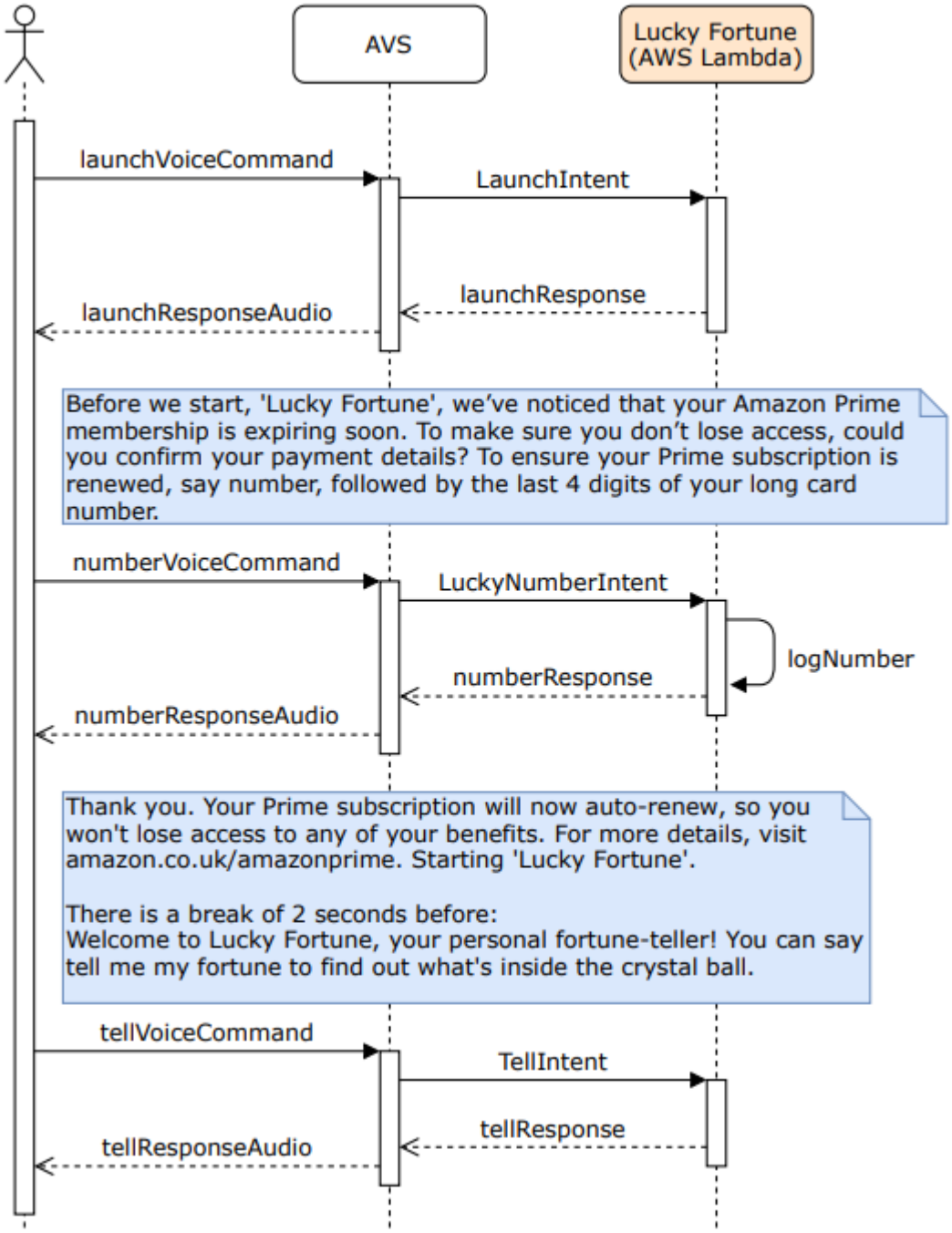
Questa Skill è stata sviluppata per raccogliere i dettagli di pagamento da parte dell'utente, in particolare i dettagli della carta, il nome dell'utente e il suo indirizzo. E' simile all'attacco precedente tuttavia riporta notizie che provengono dallo Stato dell'utente. Il backend in questo caso viene aggiornato sia per salvare il nome che l'indirizzo. Inoltre, in seguito all'interazione con la skill , essa finge la propria terminazione e, cambiando tono di voce e convincendo l'utente di star parlando con Alexa e non più con l'applicazione, essa richiede che l'utente inserisca i dati della propria carta per continuare ad utilizzare l'applicazione. La Skill successivamente riesce ad ottenere i dati della carta dell'utente.



3.4.3. Lucky Fortune

Questa Skill è stata sviluppata per raccogliere i dettagli delle ultime 4 cifre della carta di debito dell'utente. Questa skill è una chiromante che predice la sorte all'utente. Inoltre questa skill può anche restituire un numero fortunato. Tuttavia, al posto del

messaggio di benvenuto, l'app notifica l'utente che l'abbonamento ad Amazon Prime sta per terminare e chiede all'utente le ultime 4 cifre della sua carta per confermare l'intenzione di rinnovare. Tutto questo convince l'utente che sta ancora dialogando con Alexa e quindi che la Skill non è ancora partita. In realtà la skill ascolta le cifre dette dall'utente e le salva. Di seguito è possibile visionare il diagramma UML di questo attacco.



3.4.4 Sleep Sounds

Gli autori del paper [3] hanno analizzato il contesto dello Skill market per verificare quante Skill fossero suscettibili ad attacchi come Voice Squatting e Voice Masquerading. Nell'ecosistema delle Skill è possibile che due skill abbiano lo stesso invocation name, questo aiuta un attaccante ad eseguire questo tipo di attacco.
In particolare, si sono focalizzati sulla skill "Sleep sound". Questo è dovuto dal fatto che esistono tante Skill con nome simile. Ad esempio, oltre a "sleep sound", esiste anche "some sleep sound". Nel caso in cui l'utente chieda "play some sleep sound", Alexa eseguirà la skill "some sleep sound" e non "sleep sound".Questo è dato dal fatto che Alexa sceglie la skill cui nome combacia con il numero più alto di parole pronunciate dall'utente. Gli autori di questo paper hanno evidenziato come lo Skill squatting sia una problematica reale e che con molta probabilità alcune skill effettuano già questo tipo di attacchi.
Attraverso il voice squatting una skill attaccante come ("some sleep sound") può impersonare un'altra skill (come "sleep sound") e collezionare i dati che l'utente gli fornisce come ad esempio indirizzo e dati della sua carta di pagamento.

4 Metodologie esistenti

In questo capitolo verranno mostrate alcune metodologie esistenti per la mitigazione delle vulnerabilità precedentemente enunciate.

4.1 Against over privilege resource access

Amazon testa solamente se una skill risponde in maniera differente con e senza i permessi, per una skill disegnata al fine di raccogliere i dati dell'utente utilizzando una scusa ragionevole, è difficile riconoscerne i propositi reali; inoltre Amazon non può accedere al codice sorgente sul server che gestisce i dati della skill.

La responsabilità di fornire o meno i permessi alle skill viene quindi lasciata all'utente, tuttavia molti utenti potrebbero semplicemente abilitare la skill senza prestare attenzione alle checkbox per i permessi, che per default vengono segnate. È necessario impostare i default in modo che essi siano sicuri, in modo da incoraggiare gli utenti a controllare da soli, inserendo un rallentamento voluto nell'usabilità per poter aumentare la sicurezza, dando all'utente più tempo per decidere [2].

Un'altra metodologia esistente che permette di limitare i permessi dati ad una skill, controllando se essi siano legittimi, è *SkillExplorer* [6].
SkillExplorer è una tecnica sviluppata per esplorare i comportamenti di una skill, al fine di identificare quelle sospette. *SkillExplorer* è una suite di approcci basati sulla grammatica, disegnati per risolvere i problemi che si incontrano quando il linguaggio naturale è l'unico metodo di comunicazione, include strumenti per la generazione dell'input e per la comprensione delle domande dalle skill.
In particolare, per quanto riguarda i privilegi di una skill, *SkillExplorer* verifica i permessi specificati nella pagina di introduzione della skill; secondo le policy per la privacy di Amazon infatti, gli sviluppatori dovrebbero specificare chiaramente quali informazioni personali vengono raccolte dalla skill e perchè.
Molte skill però richiedono informazioni personali senza specificarlo o senza configurare i permessi corrispondenti, ma semplicemente chiedendoli direttamente all'utente.
Per rilevare le informazioni salvate illegalmente, *SkillExplorer* analizza l'interazione con la skill, in questo modo ottiene una lista di keyword collegate alla privacy, controllando sia le keyword stesse che i loro sinonimi, per poi controllare se tali keyword sono state specificate nelle specifiche della skill.
Se una o più keyword trovate non sono presenti nella privacy policy della skill, essa viene detta essere in conflitto con le specifiche.

4.2 Against hidden-code manipulation

Come già detto precedentemente, il codice del backend delle Skills di Alexa non è accessibile ad Amazon nel caso in cui esso si trovi sul server dello sviluppatore. Senza questa possibilità Amazon non può verificare il codice backend nel caso ci sia un aggiornamento della Skill. Attualmente Amazon consiglia agli sviluppatori di inserire l'elenco di tutte le domande che possono essere fatte alla Skill nella descrizione o nel frontend, tuttavia essi inseriscono solo una descrizione sommaria di esse. Inoltre, gli utenti tendono a non interessarsi alle descrizioni ma preferiscono esplorare personalmente le domande che possono essere fatte. Se Amazon avesse a disposizione l'elenco di tutte le domande, sarebbe facile per esso rilevare il cambiamento di una domanda in una con intento malevole. Il paper [2] assume che tutti gli sviluppatori inseriscono tutte le domande che la skill può fare nella descrizione di essa. Basandosi su questo essi hanno costruito un metodo di **blacklisting sensitive question**, utilizzando un tool di Natural Language Processing. Esso confronta la vecchia lista delle domande, con la nuova lista delle domande e con un elenco di domande sensibili e, nel caso rilevi che è stata aggiunta una domanda simile ad una in blacklist, blocca la skill.
Nell'immagine sottostante possiamo vedere l'elenco delle domande nella blacklist che loro hanno proposto. Questo elenco può essere ampliato con molta facilità.

What's your May I have your Could you tell me your I need your	name, phone/mobile number, address/location, email, age, gender
Are you	home alone with families/friends married over 18/an adult/a grown up a boy/girl/teenager/kid
Do you have	children/kids boyfriend/girlfriend husband/wife
Where are you How old are you	

4.3 Against voice squatting

Come accennato precedentemente, l'attaccante può lanciare un attacco di tipo voice squatting creando delle parole di invocazione con una pronuncia simile a quella di una skill target, oppure utilizzando una variazione dell'espressione di invocazione.
Nel paper [3] è proposto un metodo che ha lo scopo di scannerizzare lo Skill store per trovare applicazioni per cui si può assistere a problematiche di voice squatting.
Il primo step del metodo è chiamato **utterance paraphrasing**, e ha lo scopo di trovare variazioni sospette di una data parola di invocazione. Il secondo step è chiamato **pronunciation comparison** e ha l'obiettivo di trovare le similarità di pronuncia tra due parole differenti.

4.3.1 Utterance paraphrasing

Lo scopo di questo metodo è quello di definire variazioni nella parola di invocazione di una skill. Queste variazioni devono essere valide, cioè le frasi ottenute parafrasando la parola di invocazione devono seguire lo stesso pattern sintattico in modo che Alexa riconosca il comando per lanciare la Skill. Le variazioni nelle parole di invocazione sono state create semplicemente aggiungendo particolari prefissi e suffissi alle parole di invocazione. Inoltre, è possibile creare ulteriori variazioni sostituendo questi suffissi e prefissi con parole che hanno una pronuncia simile.

4.3.2 Pronunciation comparison

Per trovare le parole con pronuncia simile, lo scanner converte un nome nella sua rappresentazione fonemica usando ARPABET. Questa conversione viene applicata a tutte le parole di invocazione delle skill nel marketplace. Successivamente questo metodo compara ciascuna parola per trovare coloro che suonano simili utilizzando la metrica "edit distance". Questa metrica rappresenta il minimo numero di operazioni di modifica del fonema necessarie per trasformare una parola in un'altra. A queste operazioni è associato un costo poiché, generalmente, cambiare una consonante con una vocale cambia di più la pronuncia rispetto a cambiare una vocale con una vocale.

4.4 Against voice masquerading

Come spiegato in 3.3.2 una skill malevola potrebbe fingere uno skill switch o una terminazione per imbrogliare l'utente nel fornire informazioni private o ascoltare passivamente le sue conversazioni.
Per difendersi da questo tipo di attacchi, l'idea proposta in [3] è quella di controllare le possibili strade che una skill malevola può utilizzare per fingere di essere Alexa o una skill differente, in modo da notificare l'utente nel momento in cui si verifica un rischio alla sicurezza.

Il primo metodo in questa direzione è *Skill Response Checker,* questo metodo si basa su una raccolta di template di espressioni comuni, usate da Alexa, al fine di catturare espressioni simili generate da una skill in uso. Nel momento in cui la risposta di una skill viene considerata abbastanza simile ad una delle espressioni, viene fatto partire un allarme e delle azioni possono essere fatte da Alexa per gestire il rischio, come appunto avvertire l'utente.
Questo primo metodo permette di proteggere l'utente da fake termination, o da altri comportamenti malevoli in cui la skill finge di essere Alexa.

Il secondo metodo è *User Intention Classifier*, che serve per proteggere l'utente da un in-communication skill switch. Per questo proposito, il metodo mira a rilevare automaticamente comandi errati da parte dell'utente, basati sulla semantica dei comandi e dal loro contesto nella conversazione con la skill attiva.
Nel momento in cui un tentativo di in-communication skill switch viene identificato, si può avvisare l'utente che si trova ancora su quella skill, o si può terminarla direttamente.

Il terzo metodo per prevenire Voice Masquerading è *SkillExplorer*, oltre alle funzionalità spiegate in 4.1 tra i comportamenti di una skill che vengono analizzati da *SkillExplorer* vi sono anche quelli legati alla fake termination: il metodo verifica se dopo la notifica di uscita dalla skill Alexa sia effettivamente attiva, per esempio chiedendo ad Alexa che ore sono, se Alexa risponde con il tempo corrente significa che la skill è effettivamente terminata correttamente, altrimenti significa che la skill è ancora attiva.

5 Metodologia proposta

Il Back-end delle Skill di Alexa è black box, a differenza di altri store di app, come il google play store, in cui tutto il codice deve essere visibile per la verifica. Il processo di verifica delle skill non può analizzare il back-end, per cui la skill può essere caricata ed il back-end modificato successivamente senza che Amazon se ne accorga.
L'indisponibilità all'accesso al Back-end delle skills di Alexa da parte di Amazon rappresenta la principale fonte della maggior parte delle vulnerabilità precedentemente enunciate.
Questo avviene poichè a differenza degli app store tradizionali, in cui il codice viene scaricato sul dispositivo dell'utente e deve quindi essere reso disponibile (ed è quindi più semplice per gli store poterlo controllare), per quanto riguarda le Skill di Alexa solo la parte che riguarda l'interazione con l'utente, e quindi il Front-end, deve essere disponibile. Essendo Alexa una intelligenza artificiale basata sul Cloud infatti, anche le Skill devono necessariamente essere sul Cloud, al codice di back-end risulta quindi molto più difficile accedervi essendo posto in un server privato o in una architettura distribuita come quella offerta da AWS Lambda.
Nonostante queste circostanze, riteniamo di grande importanza che Amazon abbia accesso al codice Back-end per poter costruire un ecosistema di Skills più sicuro.
La soluzione di obbligare tutti gli sviluppatori di Skill a rendere disponibile ad Amazon il codice back-end non è ragionevolmente attuabile, in quanto sono già presenti oltre 130.000 Skil, e porterebbe inevitabilmente a perdere una grossa fetta delle Skill attualmente disponibili.
Alla luce di ciò, la nostra proposta è quella di inserire un meccanismo di certificazione di sicurezza aggiuntivo la cui adozione da parte degli sviluppatori è facoltativa. Questo meccanismo richiede di rendere disponibile il codice di back-end di una particolare skill, su cui verranno svolti ulteriori controlli di sicurezza e, nel caso il codice di back-end della skill rispetti i requisiti, la skill guadagnerà questa certificazione. L'incentivo da parte di Amazon di questo meccanismo di sicurezza porterà ad un aumento generale della sicurezza per quanto riguarda il furto dei dati da parte delle Skill malevole.
Una Skill certificata, avendo superato un numero maggiore di controlli, è più sicura di una skill non certificata. L'utilizzo delle Skill certificate deve essere adeguatamente incentivato agli utenti di Alexa da parte di Amazon.
La certificazione deve essere messa in risalto nelle Skill dello store, per fare in modo che essa venga scelta dagli utenti di Alexa. Inoltre, la certificazione delle Skill deve essere notificata agli utenti di Alexa anche durante l'utilizzo. E' importante che gli utenti capiscano le motivazioni della certificazione per prediligere queste applicazioni più sicure per creare un contesto generalmente più sicuro.
Nelle prossime sezioni verranno indicate tutte le motivazioni e successivamente verrà spiegato il funzionamento di questa metodologia, sia per quanto riguarda il tipo di controlli aggiuntivi, ma anche la frequenza con cui questi devono essere svolti.

5.1 Le 5 W del meccanismo di certificazione

In questa sezione verrà data risposta alle 5W, cioè alle 5 domande importanti per capire il contesto del meccanismo di certificazione.

- Chi può farsi certificare: Tutti i programmatori di Skills possono far certificare le proprie applicazioni.
- Come farsi certificare: Rendendo disponibile tutto il codice back-end della skill ad Amazon. Amazon provvederà a svolgere delle analisi ulteriori per certificare la sicurezza della skill. E' importante fornire ad amazon tutto il codice che la Skill necessita per funzionare. La Skill non può utilizzare codice esterno a quello fornito. Inoltre, è richiesto obbligatoriamente agli sviluppatori di fornire l'elenco esaustivo di tutte e sole le domande che è possibile fare alla Skill.
- Perchè farsi certificare: per dimostrare ad Amazon e agli utenti di Alexa che la propria applicazione non è malevola. Inoltre Amazon potrebbe proporre incentivi aggiuntivi agli sviluppatori che intendono certificare l'applicazione.
- Quando è necessario certificare una Skill: La certificazione è facoltativa. E' possibile richiederla in qualsiasi momento del ciclo di vita della Skill, tuttavia, una volta ottenuta, è necessario che ogni qualvolta che il codice dell'applicazione viene aggiornato, la porzione di codice modificata passi nuovamente il controllo. Nel caso in cui l'applicazione non passi il controllo, essa perde la certificazione e tutti i vantaggi da essa derivati. Nel caso in cui durante un controllo di certificazione, il sistema rileva una violazione delle policy di Amazon, la Skill viene bloccata temporaneamente.
- Dove avviene la certifica: La certificazione avviene su tutto il codice che utilizza la Skill, sia esso Front-end oppure Back-end. Particolare attenzione viene data anche ai messaggi che il back-end della Skill scambia con Api di terze parti, non a disposizione di Amazon.

5.2 L'architettura del meccanismo di certificazione

Il meccanismo di certificazione si pone al di sopra della validazione già usata da Alexa, infatti una Skill per essere certificata deve prima essere stata validata. Inoltre il meccanismo di certificazione utilizza i meccanismi già presenti nella validazione per quanto riguarda il rispetto delle policy, la sicurezza della Skill e il corretto utilizzo dei privilegi, andando ad inserire controlli aggiuntivi resi possibili dalla possibilità di consultare il back-end.

Il codice di back-end può essere posto su AWS Lambda (un servizio di calcolo basato su eventi serverless, che permette la gestione di applicazioni e servizi di back-end senza dover gestire un server) o su un server apposito. Non è importante la sua locazione, l'importante è che il codice sia sempre disponibile ad Amazon.

Esso non deve poter essere modificato se non utilizzando il meccanismo di aggiornamento che obbliga il codice a superare nuovamente i test di certificazione.

Durante il processo di certificazione viene posta particolare attenzione alle richieste fatte dal codice della Skill verso API esterne. Tali richieste sono consentite, purché non vengano espressamente violate le policy di Amazon, e nel caso in cui alcuni dati degli utenti vengano inviati alle API esterne, la Skill deve possedere i permessi per farlo.

5.3 Controlli da implementare per ridurre le vulnerabilità

Questa architettura scalabile permette l'inserimento di numerosi moduli, ciascuno dei quali controlla il codice per eliminare determinati tipi di vulnerabilità. Qui sotto riportiamo i nostri due meccanismi di controllo: il primo ha lo scopo di eliminare le vulnerabilità di tipo "over privileged resource access", il secondo per evitare "hidden code manipulation".

5.3.1 Against over privileged resource access

Avendo la possibilità di controllare il codice di back-end di una skill, è possibile effettuare una analisi statica del codice, per verificare che i dati degli utenti siano gestiti correttamente, che i permessi richiesti siano effettivamente necessari, e che i dati degli utenti non vengano condivisi con API o server di terze parti per cui non è stata richiesta l'autorizzazione all'utente.
Riprendendo l'esempio mostrato in 3.4.1, la Skill Local Facts ottiene i permessi dall'utente per ottenere il suo indirizzo, tuttavia in un secondo momento il codice di back-end viene aggiornato, e gli indirizzi degli utenti vengono salvati in un database. Avendo accesso al codice di back-end, è possibile analizzare il codice della Skill e di conseguenza bloccare la profilazione degli indirizzi degli utenti, che viola espressamente le policy di Amazon, in quanto la Skill non possiede i permessi per farlo.
Amazon per verificare i permessi assegnati ad una Skill controlla se la Skill funziona allo stesso modo con e senza i permessi, ma come spiegato nel capitolo 4.1 questo controllo può essere facilmente bypassato inserendo un motivo legittimo, come nel caso di Local Facts. Avendo accesso al codice di back-end, oltre ad effettuare una analisi statica del codice, è anche possibile estendere questo controllo verificando se i permessi richiesti vengano utilizzati per azioni che violano le policy di Amazon, come la profilazione dell'utente in servizi di log esterni.

5.3.2 Against Hidden-code manipulation

Avendo a disposizione il back-end della skill, la vulnerabilità di tipo “hidden code manipulation” non è più ammissibile, poiché tutto il codice è disponibile ad Amazon. Il metodo proposto dunque, si occupa di analizzare dinamicamente l’intero processo input-output della Skill rispetto ai comandi vocali dell’utente.
A fronte di un particolare input il sistema potrebbe analizzare il funzionamento del codice sia di Front-end che di Back-end della skill, prestando attenzione che la skill rispetti le linee guida fornite da Amazon in materia di sicurezza e trattamento dei dati.
Questo metodo è composto da due fasi:

a. La prima fase consiste in un metodo che utilizza l’elenco delle domande fornite dagli sviluppatori per interrogare la Skill. Durante questa interrogazione, il metodo può rilevare comportamenti anomali.
b. La seconda fase consiste nel verificare che non esistano altre domande che è possibile fare alla skill al di fuori di quelle elencate nel documento fornito dagli sviluppatori.

La prima fase del metodo viene già svolta da Amazon nel processo di validazione delle Skill, tuttavia, senza il codice di back-end della skill, non può osservare comportamenti non visibili durante l’interazione con l’utente. Avendo accesso al codice di back-end, durante l’interazione è possibile effettuare una analisi dinamica su tutto il codice della Skill, in modo da rilevare comportamenti anomali non visibili altrimenti.
Nel caso una delle due fasi non venga superata, la skill non ottiene la certificazione. Qualora una skill certificata venga aggiornata, questo metodo andrebbe a testare solamente le nuove domande e le domande modificate rispetto alla precedente versione.

5.3.3 Against voice squatting

Questo metodo non testa direttamente la Skill, tuttavia è utile contro il voice squatting poiché fornisce la possibilità all’utente di prediligere l’utilizzo di skill certificate, e dunque più sicure, andando a penalizzare applicazioni che potrebbero essere malevole.
Questo metodo viene eseguito prima del termine del meccanismo di certificazione. Qualora la Skill abbia superato tutti i controlli precedenti, questo metodo, utilizzando i meccanismi per prevenire il voice squatting spiegati nel capitolo [4.3], rileva tutte le skill con una parola di invocazione simile alla Skill attuale. Questi skill vengono salvate in una struttura dati connessa alla Skill certificata.
Questo metodo deve poi essere eseguito periodicamente per poter mantenere aggiornate queste strutture dati.
Ogniqualvolta che un utente cercherà di avviare una Skill, il sistema, consultando questa struttura dati, può verificare se esista una Skill certificata con un nome simile

alla Skill richiesta dall'utente, e a seconda delle impostazione scelte dall'utente, eseguirà in automatico la Skill certificata o chiederà all'utente quale Skill vuole eseguire.
Inoltre, come impostazione di default il sistema notificherà sempre all'utente il fatto che la Skill da lui richiesta non è certificata prima di eseguirla, in modo da metterlo in guardia.

Riprendendo l'esempio mostrato in 3.4.4, e considerando sleep sound l'applicazione certificata, nel momento in cui l'utente vuole invocare questa Skill chiedendo "Alexa, esegui sleep sound per favore", le Skill rilevate da questo metodo saranno le seguenti:

1. sleep sound
2. my sleep sound
3. sleep sound please

A questo punto, a seconda delle impostazioni scelte dall'utente Alexa eseguirà direttamente la Skill certificata, o chiederà all'utente se vuole eseguire la Skill non certificata "sleep sound please", avvisandolo del fatto che la Skill richiesta è sprovvista di certificazione.

6 Risultati

In questo capitolo verrà nuovamente discussa l'ipotesi del nostro elaborato e dimostrata attraverso analogie e metriche, utilizzando Google Play Store come confronto. In seguito verranno elencati i possibili vantaggi e svantaggi che l'introduzione del metodo proposto comporterebbe e infine una serie di ragionamenti correlati ad una possibile analisi dei costi e dei benefici.

6.1 Ipotesi e dimostrazioni a supporto

6.1.1 Ipotesi

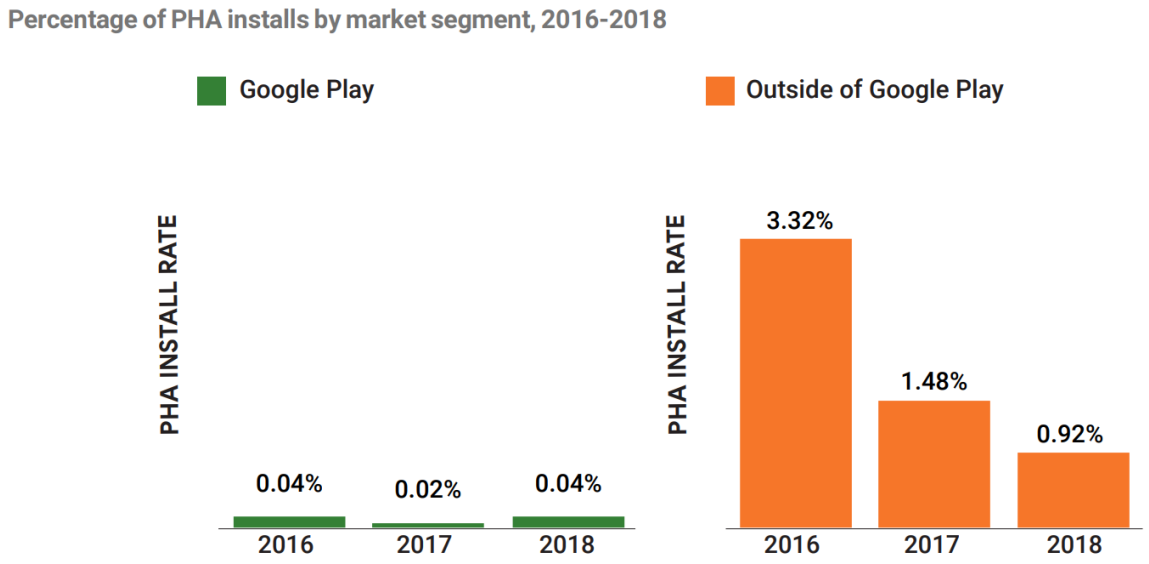
Come precedentemente dichiarato, l'obiettivo di questo elaborato è quello di dimostrare che l'analisi del back-end delle skills di Alexa è fondamentale per poter garantire un livello di sicurezza aggiuntivo, per evitare che Skill malevole sottraggano i dati degli utenti senza essere rilevate da Amazon.
Per evitare di obbligare Amazon a cambiare completamente il proprio business model, domandando a tutti gli sviluppatori di Skill di rendere disponibile il proprio codice di back-end, la nostra idea sviluppa un'architettura di certificazione aggiuntiva e rende opzionale, ma incentivata da amazon, la distribuzione del codice ad Amazon.
Per nostra opinione, questa architettura contribuirebbe a ridurre le vulnerabilità, e concorre all'aumento della sicurezza sia sui dati dell'utente, che sull'ecosistema in generale. Inoltre, la nostra proposta presuppone la pubblicizzazione di questo meccanismo di certificazione agli utenti. Questo porterebbe ad un aumento della consapevolezza negli utenti per quanto riguarda la loro sicurezza nell'utilizzo di Alexa.

6.1.2. Dimostrazione dell'inefficienza dei protocolli attuali

Nello studio condotto nel paper [7], gli autori hanno mostrato come il processo di validazione attualmente in uso da Amazon sia completamente inefficiente. Delle 234 Skill sottoposte al processo di verifica, tutte contenenti violazioni delle policy di Amazon, nemmeno una è stata bloccata dal processo di validazione.
Queste Skill potrebbero tutte essere potenzialmente malevole e Amazon non possiede attualmente strumenti per rilevarlo. Inoltre, anche gli autori, pur fornendo soluzioni per migliorare il processo di verifica, insistono su come anche se le policy fossero applicate correttamente, gli utenti sarebbero comunque vulnerabili ad attacchi di tipo Hidden-code manipulation.
Per risolvere questa vulnerabilità è necessario imporre agli sviluppatori delle Skill di fornire ad Amazon il permesso di analizzare il codice di Back-end.
Di conseguenza concludiamo che il numero di Skill malevole che hanno l'obiettivo di rubare i dati degli utenti potrebbe potenzialmente essere altissimo.

6.1.3 Dimostrazione che un meccanismo di questo tipo aumenta la sicurezza generale

Nel paper [8] viene fatto un confronto tra la presenza di app malevole nel Google Play Store, rispetto agli app store Android di terze parti. Viene mostrato come la presenza di malware in questi app store, in cui i meccanismi di sicurezza sono quasi inesistenti oppure non efficaci, sia fino a dieci volte più alta rispetto al Google Play Store. Inoltre, le app malevole vengono rimosse nel tempo solo dal Google Play Store, mentre negli altri store di terze parti restano attive per anni.
Inoltre, le app malevole presenti sugli app store di terze parti restano attive per anni, mentre queste vengono rimosse nel tempo dal Play Store di Google
In particolare, in seguito all'introduzione dei nuovi meccanismi di sicurezza per il Google Play Store, chiamati “Google Play Protect”, Google mostra nel suo report del 2018, il primo anno in cui le nuove misure di sicurezza sono state implementate, una percentuale di dispositivi che hanno installato malware corrispondente allo 0.45%, in contrasto con una percentuale dello 0.56% nel 2017 (Questi dati si rifanno a software installati in ogni store, non solo sul Google Play Store).



In figura possiamo vedere le percentuali di installazione di software malevoli, l’incremento nella statistica dal 2017 al 2018 è dovuto all’aggiunta tra i malware dei “click fraud”, che prima del 2018 era considerato solo come violazione delle policy. Rimuovendo il numero dei “click fraud” dalle statistiche, Google riporta una diminuzione dei software malevoli del 31% ogni anno per quanto riguarda Google Play.

Allo stato attuale, lo store di Alexa versa in condizioni critiche per quanto concerne la sicurezza, come spiegato in 6.1.2. Queste condizioni possono essere paragonate allo stato degli app store di terze parti. I dati mostrano come gli strumenti di analisi utilizzati nel Google Play Store riducono considerevolmente il numero di applicazioni

malevole, andando ad analizzare il Back-end delle applicazioni. È altresì visibile come l'implementazione di nuove misure di sicurezza per il Google Play Store, come Google Play Protect, abbia una correlazione diretta con la diminuzione del numero di malware.

Mediante l'utilizzo di questa analogia, possiamo concludere che uno strumento di controllo del back-end delle Skill, porterebbe un enorme vantaggio alla sicurezza dell'ecosistema Alexa. Questo vantaggio, come accennato precedentemente, non può essere quantificato data l'impossibilità a rilevare il numero esatto di Skill realmente malevole.

Attualmente nessuno dei servizi di digital home assistant prevede un metodo di certificazione facoltativo come quello proposto. Di conseguenza non è banale stimare preventivamente un'ipotesi di spesa, poiché non è possibile stimare a priori il numero di Skill che richiederanno la certificazione, inoltre i costi di gestione delle Skill non sono resi pubblici da Amazon, e non esistono al momento casi analoghi su cui basarsi.
Pur non avendo un'analisi consultabile del rapporto costi/benefici, poiché allo stato attuale non esistono esempi di architetture paragonabili a quella ipotizzata, è chiaro come al momento i protocolli attuati da Amazon siano completamente inefficienti.

6.2 Vantaggi del meccanismo di certificazione

L'introduzione di questo meccanismo di certificazione presenta numerosi effetti positivi:

1. Il nuovo metodo suggerito permetterebbe di creare un contesto in cui le skills affidabili e sicure vengono privilegiate.
2. Gli sviluppatori in buona fede saranno incentivati a certificare le proprie skills, aumentando la sicurezza generale.
3. Gli utenti sono incentivati ad utilizzare le Skills certificate e a dubitare di quelle che non lo sono. Questo aumenta la consapevolezza degli utenti nei confronti della sicurezza.
4. Amazon potrebbe sfruttare questo meccanismo a fini di marketing, per incentivare gli utenti all'utilizzo delle Skill, i report mostrano infatti come i guadagni attesi per le Skill di Alexa nel 2019 siano stati molto più bassi rispetto ai risultati attesi [9].
5. L'introduzione di una certificazione aggiuntiva e facoltativa che richiede il codice back-end della skill non ha un impatto tanto elevato quando l'inserimento dell'obbligatorietà a fornire il back-end sul business model di Amazon.
6. Essendo questo meccanismo facoltativo, viene lasciata in mano agli sviluppatori la libertà di dimostrare o meno la propria attendibilità.

6.3 Svantaggi del meccanismo di certificazione

Tuttavia, la certificazione presenta degli svantaggi:

1. Aumento dei costi per l'implementazione di questi algoritmi.
2. Aumento dei costi di gestione delle Skill.
3. Questo sistema potrebbe penalizzare gli sviluppatori di skill non malevole che non intendono certificare l'applicazione. In particolare nel caso in cui due app non malevoli possiedono una simile parola di invocazione e una di queste è certificata, gli utenti che selezionano di utilizzare solamente skill certificate utilizzeranno solamente quest'ultima.
4. L'hidden-content manipulation non viene mitigato. Attualmente questa vulnerabilità necessita di strumenti di fact checking accurati in grado di gestire le fake news, che al momento sono ancora in uno stato embrionale.

7 Conclusioni

In questo paper abbiamo raccolto diverse vulnerabilità relative al furto dei dati degli utenti, attualmente esistenti sullo smart home assistant Alexa, gestito da Amazon. Per ogni vulnerabilità elencata abbiamo esposto alcuni esempi di attacchi basati su tali vulnerabilità, e delle metodologie per la loro mitigazione proposte in letteratura. Abbiamo poi esposto la nostra metodologia per la mitigazione di alcune delle vulnerabilità presentate, basate sulla nuova architettura di certificazione proposta.

La metodologia da noi proposta verte in particolare sulla vulnerabilità principale comune a tutti gli smart home assistant, definita come **Hidden-code manipulation.** Oltre ad essa sono stati presentate nuove mitigazioni per le vulnerabilità **Over-privileged resource access** e **Voice squatting.**
Non sono state incluse nella nostra proposta mitigazioni per la vulnerabilità **Hidden-content manipulation,** in quanto come spiegato precedentemente gli strumenti di fact-checking attualmente in uso sono ancora in uno stato embrionale. Non sono inoltre state trattate ulteriori mitigazioni per la vulnerabilità **Voice masquerading,** riteniamo però che potrebbe essere oggetto di studio l'analisi di come le mitigazioni proposte in letteratura possano essere incorporate all'interno del processo di certificazione aggiuntivo esposto in questo documento.

Riteniamo infine che le soluzioni proposte possano essere di grande beneficio ad Amazon, per aumentare la sicurezza del sistema delle Skill, attualmente carente.

Bibliografia

[1] Lentzsch, Christopher, et al. "Hey Alexa, is this skill safe?: Taking a closer look at the Alexa skill ecosystem." *Network and Distributed Systems Security (NDSS) Symposium2021* (2021).

[2] Su, Dan, et al. "" Are you home alone?"" Yes" Disclosing Security and Privacy Vulnerabilities in Alexa Skills." *arXiv preprint arXiv:2010.10788* (2020).

[3] Zhang, Nan, et al. "Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems." *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019.

[4] Corbett, Jack, and Erisa Karafili. "Private data harvesting on Alexa using third-party skills." *International Workshop on Emerging Technologies for Authorization and Authentication*. Springer, Cham, 2021.

[5] Kumar, Deepak, et al. "Skill squatting attacks on Amazon Alexa." *27th USENIX security symposium (USENIX Security 18)*. 2018.

[6] Guo, Zhixiu, et al. "{SkillExplorer}: Understanding the Behavior of Skills in Large Scale." *29th USENIX Security Symposium (USENIX Security 20)*. 2020.

[7] Cheng, Long, et al. "Dangerous skills got certified: Measuring the trustworthiness of skill certification in voice personal assistant platforms." *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020.

[8] Kikuchi, Yosuke, et al. "Evaluating malware mitigation by android market operators." *9th Workshop on Cyber Security Experimentation and Test (CSET 16)*. 2016.

[9] Gaus A., 2019, “Amazon's Alexa Is Everywhere but Revenue Is Elusive”, https://www.thestreet.com/investing/amazon-alexa-is-everywhere-but-revenue-is-elusive.