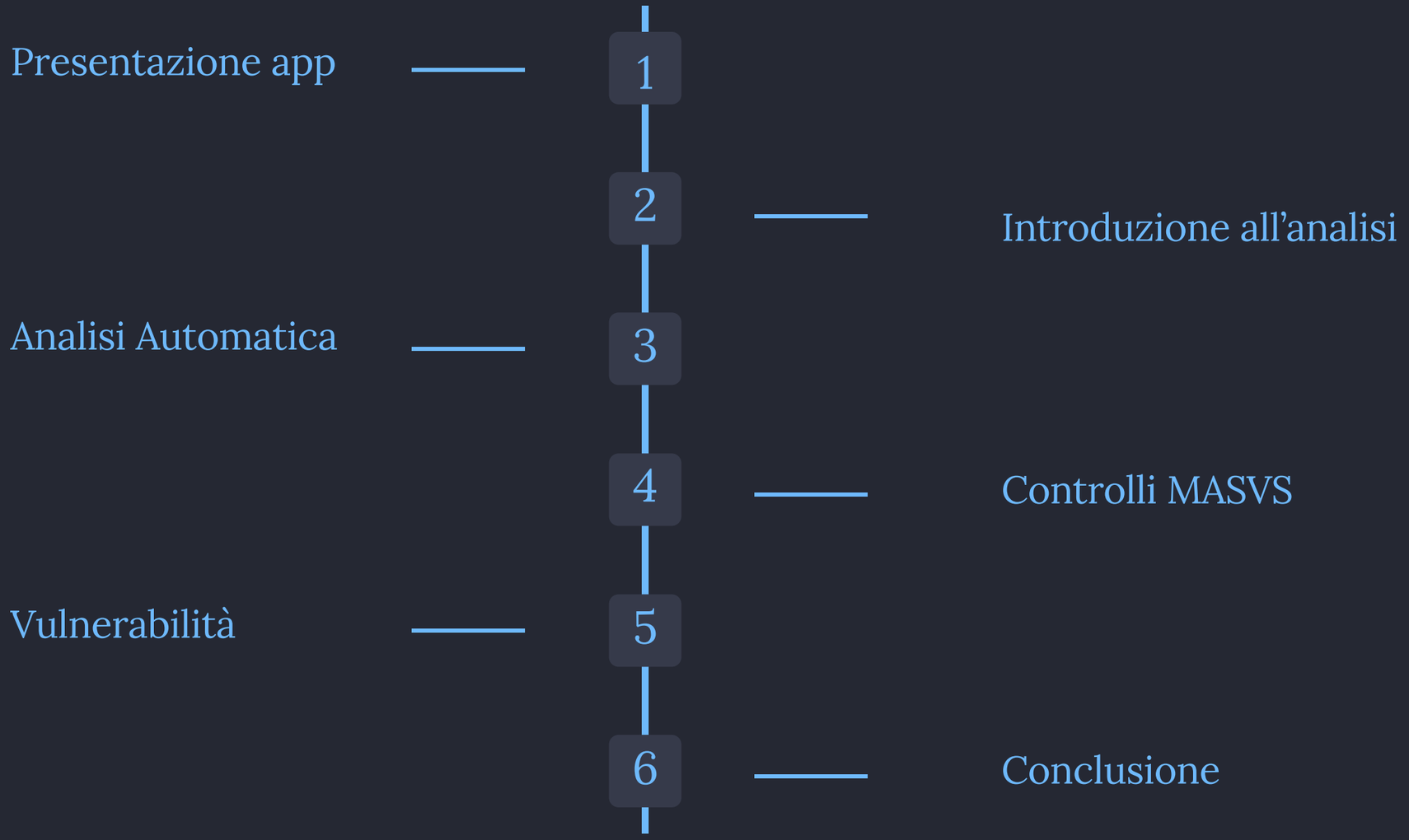
Sicurezza in Ambienti mobili: AigoSmart

Studente: Ludovico Nigro

Sommario della Presentazione



AigoSmart

Applicazione Android

L’applicazione presa in esame per il nostro caso di studio è stata l’app: AigoSmart presente sul Play Store e scaricabile gratuitamente.
La versione presente sullo store è la 2.3.6 uscita il 12 aprile 2024.
La versione presa in esame è la versione 1.9.0 del 18 aprile 2022, scaricata dal sito APKPure.

Funzionalità

Permette anche ai tuoi dispositivi di percepire e interagire tra loro, offrendo un'esperienza senza interruzioni e conveniente.
L'app ti consente di impostare preimpostazioni per i tuoi elettrodomestici in modo che con un solo tocco, tutti i dispositivi collegati passino allo stato desiderato, garantendo il massimo comfort e sicurezza.

Diffusione e Lingue

AigoSmart registra oltre 100.000+ download su PlayStore ed è disponibile in 13 lingue, tra cui Italiano, Cinese, Francese, Inglese e Tedesco.

Funzionalità di AigoSmart

1

Gestione Remota

AigoSmart permette di gestire e controllare i tuoi elettrodomestici smart da remoto, offrendo un'esperienza senza interruzioni e conveniente.

2

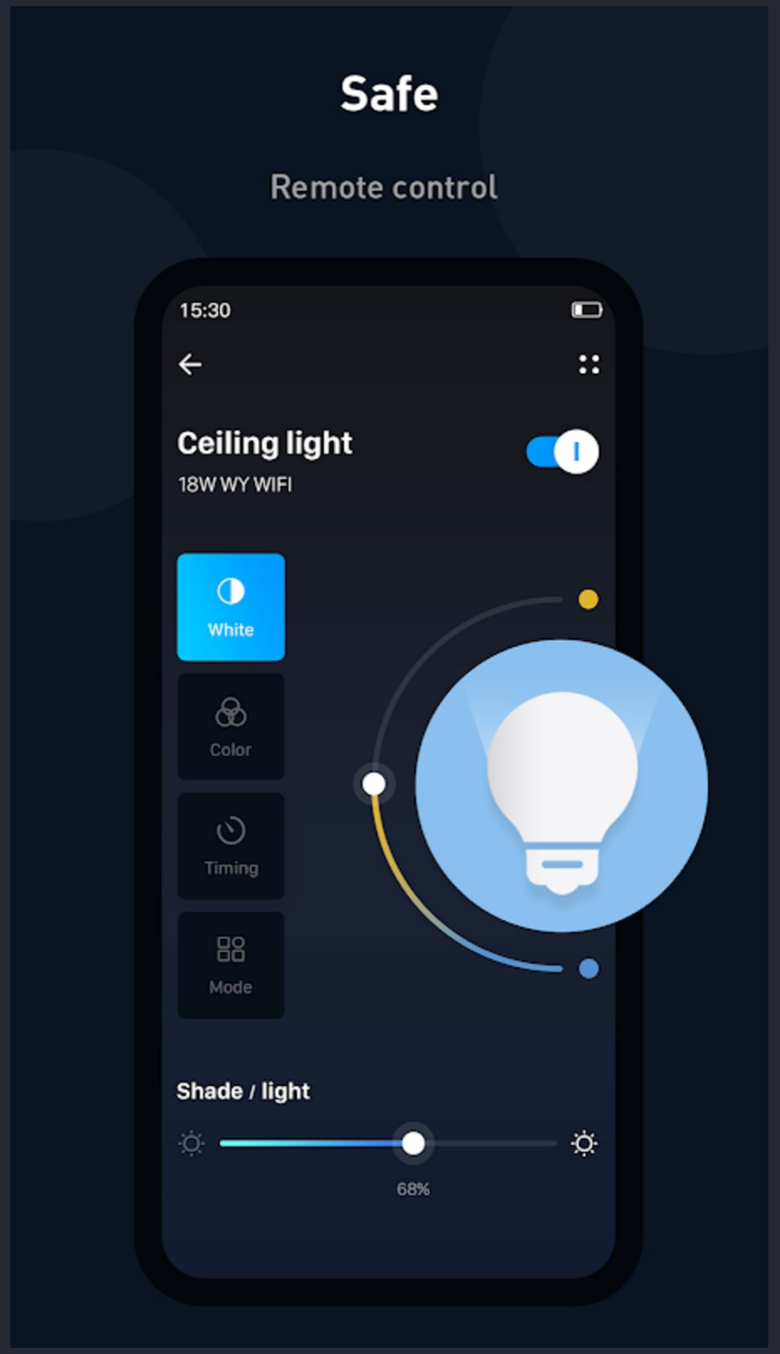
Preimpostazioni

L'app consente di impostare preimpostazioni per i tuoi elettrodomestici, in modo che con un solo tocco tutti i dispositivi collegati passino allo stato desiderato.

3

Interfaccia Intuitiva

L'interfaccia di AigoSmart è facile da navigare. Puoi aggiungere i tuoi dispositivi smart all'app e controllarli con un semplice tocco.



Introduzione all’analisi: Strumenti Utilizzati

VirtualBox

Software di virtualizzazione utilizzato per creare macchine virtuali.

Kali Linux

Distribuzione Linux per test di penetrazione e sicurezza informatica.

Android Tamer

Distribuzione Linux per la sicurezza delle applicazioni Android.

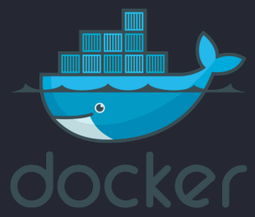
Genymotion

Software di emulazione di dispositivi Android con permessi di root.

Introduzione all’analisi: Strumenti Utilizzati

Android Debug Bridge

Strumento a riga di comando per comunicare con dispositivi Android.



Docker Desktop

Applicazione per la gestione di container Docker, utilizzata per eseguire MobSF.

MobSF

Framework di analisi automatica per la valutazione della sicurezza delle app mobili.

Java Decompiler

Software di decompilazione Java utilizzato per l'analisi del codice sorgente.

Introduzione all’analisi: Strumenti Utilizzati

MARA

Framework per l'ingegneria inversa e l'analisi di applicazioni mobili.

ImmuniWeb

Tool di analisi automatica di applicazioni mobile basato su interfaccia web.

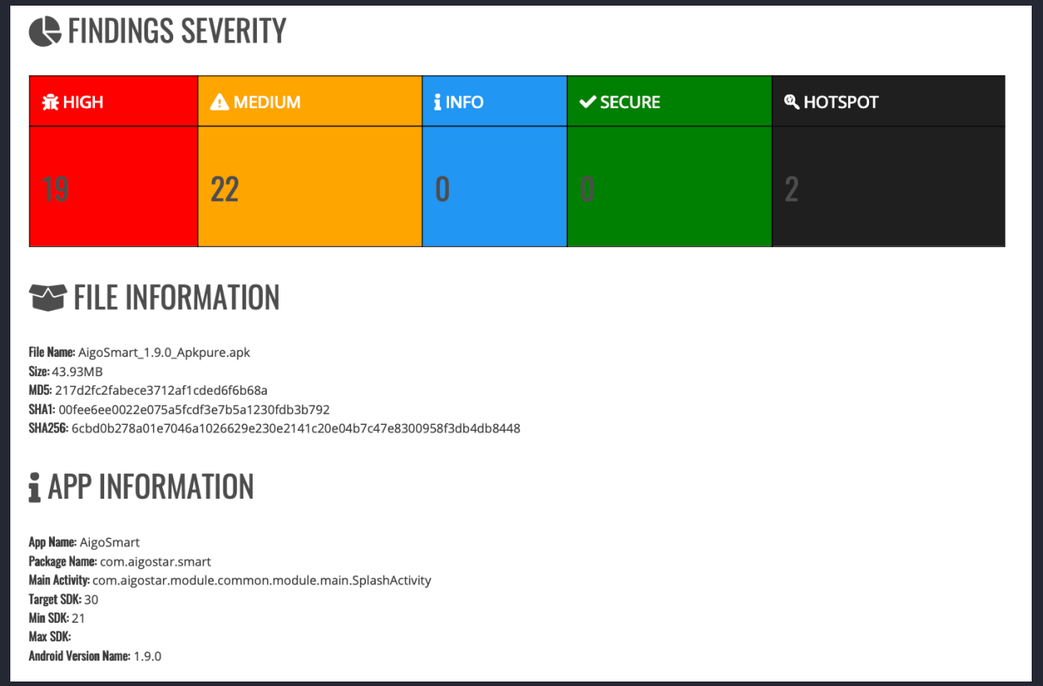
Visual Studio Code

Potente editor di codice sorgente utilizzato per l'analisi del codice.

Analisi Automatica: MobSF

1 Caricamento e Analisi

Il file APK dell'applicazione è stato sottoposto ad analisi automatica utilizzando lo strumento MobSF.



FINDINGS SEVERITY

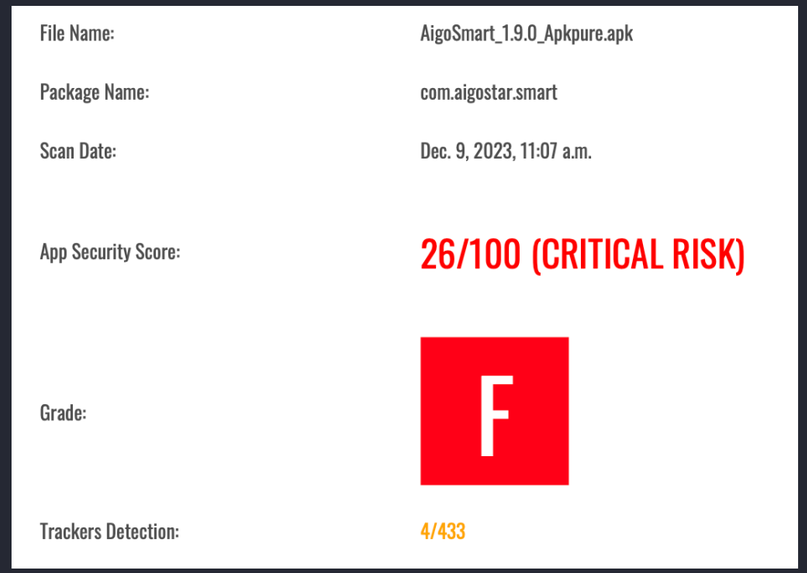
HIGH	MEDIUM	INFO	SECURE	HOTSPOT
19	22	0	0	2

FILE INFORMATION

File Name: AigoSmart_1.9.0_Apkpure.apk
Size: 43.93MB
MD5: 217d2fc2fabece3712af1cded6f6b68a
SHA1: 00fee6ee0022e075a5fcdf3e7b5a1230fdb3b792
SHA256: 6cbd0b278a01e7046a1026629e230e2141c20e04b7c47e8300958f3db4db8448

APP INFORMATION

App Name: AigoSmart
Package Name: com.aigostar.smart
Main Activity: com.aigostar.module.common.module.main.SplashActivity
Target SDK: 30
Min SDK: 21
Max SDK:
Android Version Name: 1.9.0



File Name:	AigoSmart_1.9.0_Apkpure.apk
Package Name:	com.aigostar.smart
Scan Date:	Dec. 9, 2023, 11:07 a.m.
App Security Score:	26/100 (CRITICAL RISK)
Grade:	F
Trackers Detection:	4/433

2 Valutazione della Sicurezza

MobSF genera un punteggio di sicurezza per l'applicazione, basato sulle vulnerabilità rilevate, fornendo un'indicazione della solidità della sicurezza dell'app.

3 Identificazione delle Vulnerabilità

L'analisi di MobSF ha permesso di individuare potenziali problemi di sicurezza nell'applicazione, che verranno approfonditi nelle sezioni successive.

Analisi Automatica: Immuniweb

1 Identificazione delle Vulnerabilità

ImmuniWeb è un servizio Web che offre un tool per l'analisi automatizzata delle applicazioni mobili.
Esso genera un report dettagliato delle vulnerabilità individuate, classificate in quattro livelli di gravità e organizzate secondo le 10 categorie di vulnerabilità di OWASP. Inoltre, ImmuniWeb analizza e categorizza i permessi utilizzati dall'applicazione.

2 Valutazione della Sicurezza

Nel report è incluso anche il rilevamento di falle e debolezze che potrebbero avere un impatto sull'app e che il test automatico ha riscontrato.

Analisi MASVS

Sono stati effettuati i controlli eseguiti secondo gli standard dell'OWASP Mobile Application Verification Standard (MASVS), il punto di riferimento per la sicurezza delle app mobili nel settore.
Ogni test mira a verificare specifiche proprietà definite nei codici MASVS e MSTG.

V1: Architecture, Design and Threat Modeling Requirements
V2: Data Storage and Privacy Requirements
V3: Cryptography Requirements
V4: Authentication and Session Management Requirements
V5: Network Communication Requirements
V6: Environmental Interaction Requirements
V7: Code Quality and Build Setting Requirements
V8: Resiliency Against Reverse Engineering Requirements

Vulnerabilità Identificate nell'App AigoSmart

Attraverso un'analisi approfondita dell'applicazione mobile AigoSmart, sono state identificate diverse vulnerabilità di sicurezza che potrebbero rappresentare potenziali rischi per gli utenti. È fondamentale affrontare queste problematiche per garantire la sicurezza e la privacy degli utenti che utilizzano l'app.

Vulnerabilità Identificate

1. Verifica dei criteri di sicurezza per l'accesso ai dispositivi
2. Test dell'archiviazione locale per i dati sensibili
3. Test della crittografia dei dati sulla rete
4. Uso del protocollo Http
5. Verifica dell'implementazione vulnerabile di PendingIntent
6. Test dei permessi delle app
7. Verifica dell'esecuzione di JavaScript nelle WebView
8. Verifica dell'aggiornamento forzato
9. Test della memorizzazione locale per la convalida degli input
10. Verifica del rilevamento dell'emulatore
11. Verifica dell'offuscamento

Accesso ai Dispositivi Senza Criteri di Sicurezza

1

Accesso Senza Autenticazione

È stato verificato che l'accesso all'app non è moderato da alcun criterio di sicurezza minimo. Una volta autenticato, l'utente resta connesso all'applicazione e chiunque abbia accesso al dispositivo può accedere all'app senza dover autenticarsi.

2

Compatibilità con Versioni Obsolete di Android

L'app supporta una versione minima di Android 5.0 (Lollipop), il che significa che può essere installata e eseguita su dispositivi con sistemi operativi obsoleti e meno sicuri.

3

Necessità di Miglioramenti

Per affrontare questi problemi, sono necessarie soluzioni come l'implementazione di criteri di sicurezza per l'accesso, l'obbligo di aggiornamenti di sicurezza e l'aumento del requisito minimo di Android.

Archiviazione Locale di Dati Sensibili

Dati Sensibili Accessibili

L'analisi dinamica dell'app ha rivelato che i dati sensibili, come i file relativi ai database e le preferenze condivise, sono facilmente accessibili e privi di algoritmi crittografici adeguati.

Permessi Pericolosi

Il Manifest dell'app contiene il permesso "WRITE_EXTERNAL_STORAGE", che consente di archiviare dati in una memoria esterna, esponendo potenzialmente informazioni sensibili.

Soluzioni Consigliate

È fondamentale utilizzare le API e le funzionalità di sicurezza fornite dal sistema operativo, come il KeyChain su iOS o il KeyStore su Android, per gestire in modo sicuro le informazioni sensibili. Inoltre, è essenziale memorizzare gli hash delle password e delle chiavi crittografiche anziché i loro valori in chiaro.

Vulnerabilità nella Crittografia dei Dati in Rete

Attacco Man-in-the-Middle

L'app risulta vulnerabile a un attacco Man-in-the-Middle (MITM), in cui un malintenzionato può intercettare e modificare il traffico tra il dispositivo dell'utente e il server web, rubando informazioni sensibili.

1

Verifica di Nomi Host Insicura

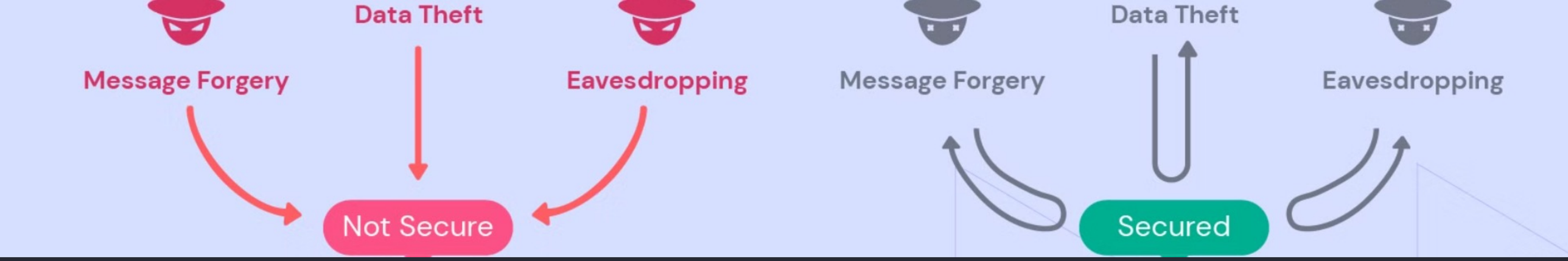
Il codice dell'app utilizza un verificatore di nomi host "ALLOW_ALL_HOSTNAME_VERIFIER", che accetta qualsiasi nome host, indipendentemente dal certificato SSL/TLS del server. Ciò consente a un attaccante di creare un falso server web e intercettare il traffico dell'utente.

2

Soluzione: Verifica di Nomi Host Sicura

Per risolvere questo problema, è necessario utilizzare un verificatore di nomi host più sicuro che controlli che il nome host del server corrisponda effettivamente al certificato SSL/TLS. Ciò aiuta a garantire che l'utente si stia connettendo al server corretto e che il suo traffico non sia intercettato.

3



Comunicazione Insicura Tramite HTTP

Rischio di Intercettazione

L'applicazione mobile utilizza il protocollo HTTP per inviare o ricevere dati, il che significa che il traffico non è crittografato e può essere facilmente intercettato da un aggressore sulla stessa rete o con accesso al canale dati della vittima.

Dati Sensibili Esposti

Se un malintenzionato intercetta i dati trasmessi tramite HTTP non crittografato, potrebbe rubare informazioni sensibili come password, informazioni finanziarie, dati personali e comunicazioni private.

Soluzione: Utilizzo di HTTPS

Per risolvere questa vulnerabilità, è essenziale che l'applicazione utilizzi il protocollo HTTPS, una versione crittografata di HTTP che protegge i dati da intercettazioni non autorizzate.

Problemi con le Autorizzazioni dell'App

Autorizzazioni "Pericolose"

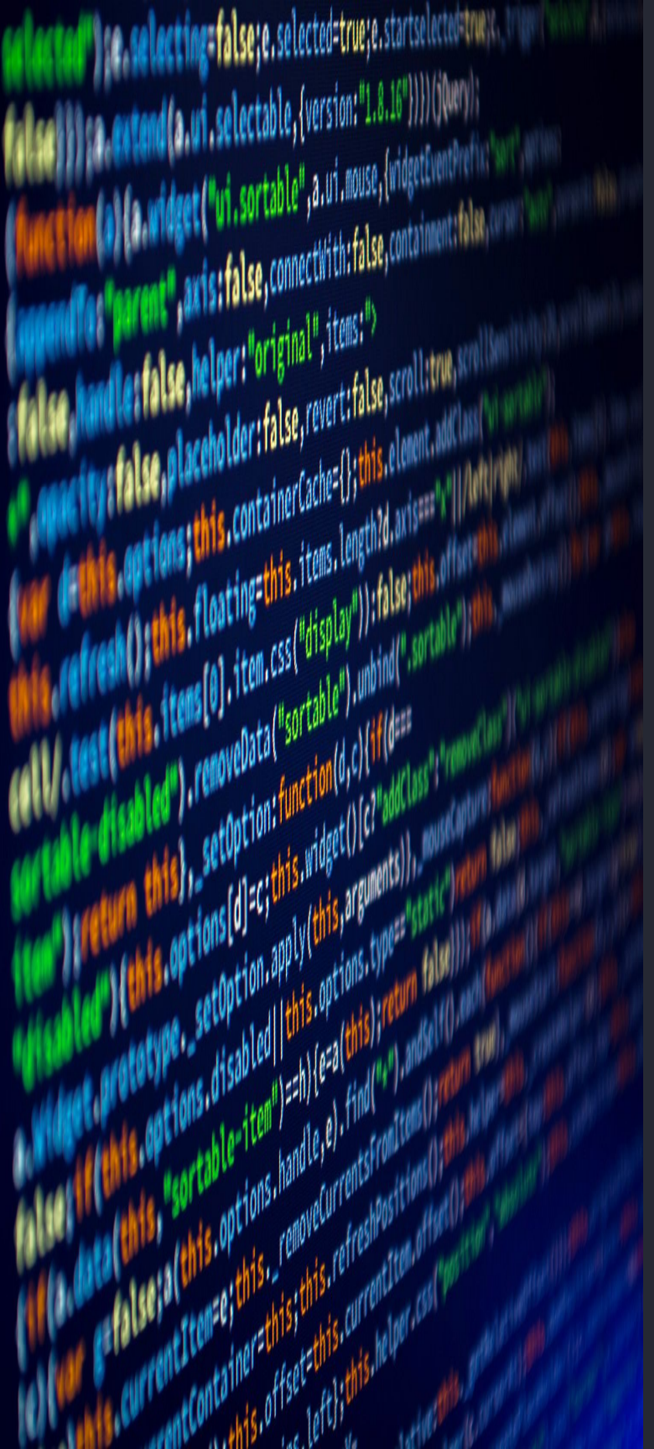
L'analisi ha rilevato che l'app richiede alcune autorizzazioni considerate "pericolose" o "sconosciute" dal punto di vista della sicurezza, il che potrebbe comportare rischi per la sicurezza o la privacy degli utenti.

Necessità di Trasparenza

È essenziale che l'app gestisca attentamente l'utilizzo di tali autorizzazioni "pericolose" e informi chiaramente gli utenti, consentendo loro di concedere o revocare il consenso in modo esplicito.

Mitigazione dei Rischi

Adottando queste pratiche, si può garantire che gli utenti siano pienamente consapevoli e in grado di controllare l'accesso dell'app ai loro dati sensibili e alle funzionalità del dispositivo.



Esecuzione di JavaScript nelle WebView

1

Abilitazione del JavaScript

L'app utilizza WebView per visualizzare pagine web al suo interno e abilita il JavaScript in modo esplicito, senza garantire un canale di comunicazione sicuro tramite HTTPS su TLS.

2

Rischio di Codice Malevolo

L'utilizzo di JavaScript potrebbe essere sfruttato da malintenzionati per eseguire codice dannoso, rappresentando una vulnerabilità.

3

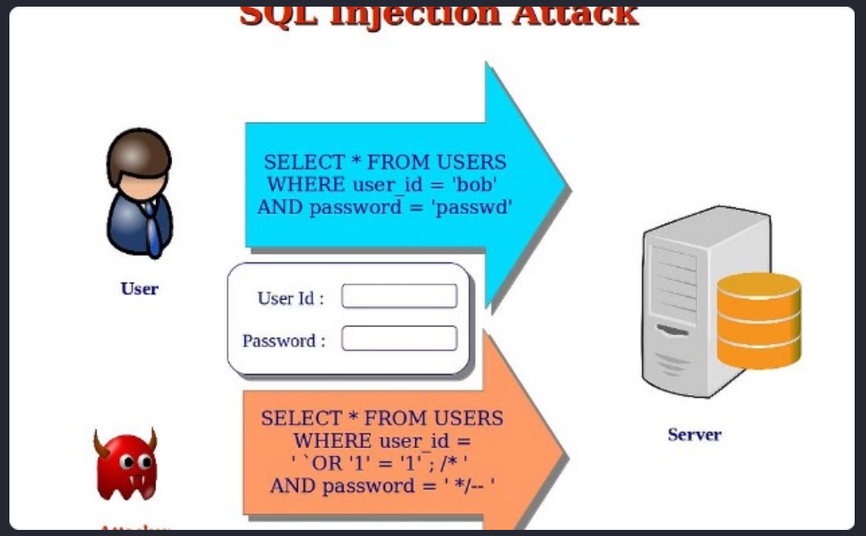
Soluzione: Connessione Sicura

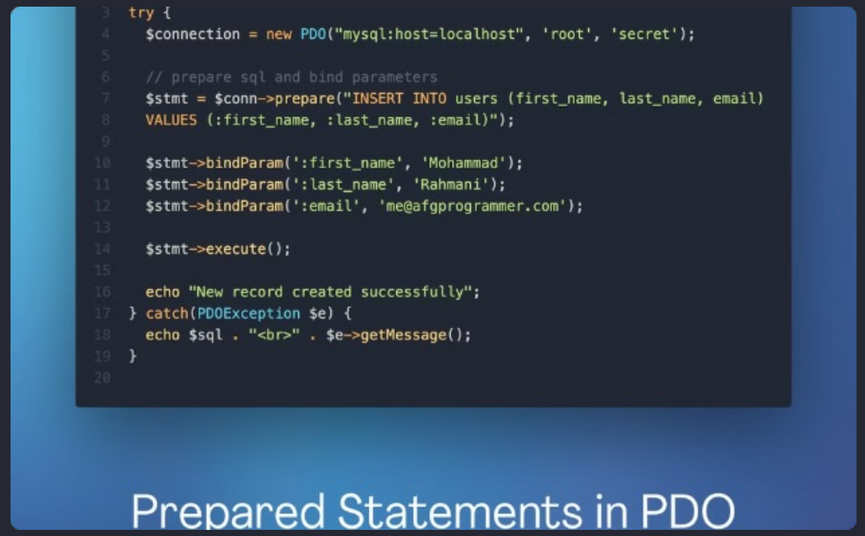
Se è necessario abilitare l'esecuzione di JavaScript nelle WebView, è essenziale farlo attraverso misure di sicurezza rigorose, come l'instaurazione di una connessione sicura tramite i protocolli SSL/TLS.

Mancanza di Aggiornamento Forzato

Avviso di Aggiornamento	L'app avvisa l'utente tramite riquadri che la versione dell'applicazione risulta obsoleta, ma permette all'utente di chiudere l'avvertimento e continuare a utilizzare l'app.
Rischio di Vulnerabilità	Ciò significa che gli utenti potrebbero continuare a utilizzare una versione vulnerabile dell'app, esponendosi a potenziali minacce.
Soluzioni Consigliate	Per imporre gli aggiornamenti, si potrebbero considerare strategie come impedire l'accesso all'app finché non viene effettuato l'aggiornamento, forzare l'aggiornamento all'avvio o utilizzare messaggi persistenti e non saltabili.

Vulnerabilità alla SQL Injection





Rischio di SQL Injection

L'utilizzo di input direttamente nelle query SQL espone l'applicazione mobile al rischio di SQL injection, una vulnerabilità che potrebbe permettere a un attaccante di manipolare le query SQL per accedere o modificare dati sensibili nei database sottostanti.

Soluzione: Istruzioni SQL Preparate

Per mitigare questo rischio, è fondamentale adottare un approccio sicuro utilizzando istruzioni SQL preparate, che separano i dati dall'instradamento delle query e impediscono agli utenti malintenzionati di inserire comandi SQL dannosi.

Mancanza di Rilevamento dell'Emulatore

Rischi degli Emulatori

L'applicazione non riesce a impedire il suo avvio su un dispositivo emulato, il che è preoccupante in quanto gli emulatori offrono un ambiente in cui è più semplice per un potenziale attaccante accedere ai dati del sandbox dell'applicazione, consentendo il furto di informazioni sensibili.

Tecniche di Mitigazione

Per mitigare questi rischi, è necessario implementare controlli aggiuntivi che impediscano l'utilizzo dell'app su un emulatore, come verifiche sulle proprietà del sistema. Tuttavia, è importante notare che esistono diverse tecniche per aggirare questi controlli di sicurezza, quindi è fondamentale proteggere i dati sensibili utilizzando tecniche crittografiche appropriate.

Conclusioni

Analisi Approfondita

L'analisi condotta ha permesso di identificare diverse vulnerabilità nell'applicazione mobile AigoSmart, che potrebbero compromettere la sicurezza e la privacy degli utenti.

Raccomandazioni di Mitigazione

Sono state fornite raccomandazioni specifiche per affrontare le vulnerabilità identificate, come l'implementazione di tecniche di protezione, la gestione corretta dei dati sensibili e l'utilizzo di connessioni crittografate.

Importanza della Sicurezza

Questo studio sottolinea l'importanza di una valutazione rigorosa della sicurezza delle app mobili fin dalle fasi iniziali dello sviluppo, al fine di mitigare i rischi e garantire una migliore protezione per gli utenti.