techno bluetooth

ESP 32

Truc qu’on a pas vu, developper une application sur android bluetooth

Tagger la présence

et continuer a detecter la presence de l’étudiant

afficher un dashboard, liste des présent et absent (diff entre inscrit et présent)

MQTT

Refresh les detecteurs toutes les 5 min / Confirmation de présence

BDD a construire

Node Red

Black box : je suis un etudiant hors de la salle sans rien savoir de l'intérieur

L’étudiant voit 2 possibilité d'attaque à travers un scan :

* bluetooth
* Wifi

Étape 1 : Cartographie et reconnaissance

Avant de chercher à exploiter des failles, vous devez comprendre le fonctionnement du système. Voici les techniques à utiliser :

A. Exploration Bluetooth

Scanner les appareils BLE :

Utilisez un outil comme hcitool, bluetoothctl ou Bettercap pour détecter les appareils Bluetooth proches.

Recherchez des UUID ou des noms d'appareils spécifiques, qui pourraient indiquer un usage lié au système.

Exemple :

sudo hcitool lescan

Objectif : Identifier les appareils (ESP32 ou autres) et vérifier les configurations de sécurité.

Analyser les protocoles BLE :

Si des UUID spécifiques sont détectés, tentez d’interagir avec les services BLE pour en extraire des informations.

Utilisez des outils comme gatttool ou des bibliothèques Python BLE.

Failles potentielles :

Absence d’authentification ou de chiffrement sur les services BLE.

UUID non randomisés permettant de suivre les appareils.

B. Exploration Wi-Fi

Scanner les réseaux Wi-Fi :

Avec un outil comme airodump-ng (ou équivalent sur ESP32), détectez les réseaux Wi-Fi actifs.

Capturez les requêtes probe pour identifier les appareils connectés.

Exemple :

sudo airodump-ng wlan0

Sniffer les paquets Wi-Fi :

Activez le mode promiscuous pour analyser le trafic Wi-Fi échangé.

Cherchez des paquets non chiffrés ou des informations sur le protocole.

Failles potentielles :

Utilisation d’un SSID mal sécurisé ou visible.

Requêtes probe des appareils exposant des informations sensibles.

Étape 2 : Exploitation des vulnérabilités

Une fois les composants identifiés, voici les attaques possibles :

A. Bluetooth

Attaque par force brute sur l’appairage :

Si des appareils nécessitent un appairage Bluetooth (ex. : ESP32 en mode pairable), essayez une attaque brute force pour casser le PIN.

Outils : btcrack, Bettercap.

Bluesnarfing :

Tentez d’accéder aux données sensibles sur les appareils Bluetooth détectés.

Condition : Si le chiffrement n’est pas activé.

Service spoofing :

Usurpez l’identité d’un appareil Bluetooth légitime pour interagir avec les services BLE.

Exemple : Envoyez de fausses informations de présence.

B. Wi-Fi

Attaque Evil Twin :

Configurez un faux point d’accès avec le même SSID que le réseau utilisé.

Capturez les données échangées entre les appareils ESP32/Raspberry Pi et le serveur.

Outils : Airbase-ng, Bettercap.

Injection de paquets :

Exploitez des failles dans la gestion des paquets pour injecter de fausses informations.

Exemple : Envoyez de faux identifiants pour perturber la détection des élèves.

Déni de service (DoS) :

Envoyez des paquets de désauthentification pour déconnecter les ESP32 ou les élèves.

Outils : Aireplay-ng.

Étape 3 : Analyse des résultats

Après avoir identifié les failles exploitables, évaluez leur gravité et leur impact potentiel :

Quels types de données peuvent être exfiltrés ?

Le système peut-il être perturbé (DoS, spoofing) ?

Les élèves peuvent-ils être indûment détectés ou absents ?

Étape 4 : Contre-mesures et recommandations

Proposez des solutions pour chaque faille identifiée.

Bluetooth

Chiffrement et authentification :

Activez le chiffrement BLE (Secure Connections) et exigez un appairage sécurisé avec un PIN robuste.

UUID dynamiques :

Configurez les appareils pour utiliser des UUID randomisés afin d’éviter le suivi.

Filtrage RSSI :

Ajustez les seuils RSSI pour ignorer les appareils trop éloignés.

Wi-Fi

Chiffrement WPA3 :

Utilisez WPA3 pour sécuriser les communications entre les appareils.

Détection d'Evil Twin :

Implémentez une surveillance active pour détecter des points d’accès malveillants.

Authentification des appareils :

Filtrez les appareils autorisés via une liste blanche des adresses MAC.

Hardening des protocoles :

Désactivez les fonctionnalités inutilisées (ex. : WPS, modes legacy).

1. Failles liées à l'appairage Bluetooth

a. Appairage non sécurisé

Problème : Si l'ESP32 utilise un mode d'appairage faible ou ouvert (ex. : mode "Just Works" ou sans authentification).

Exploit :

Force brute sur le code PIN si un appairage PIN est utilisé.

Appairage non autorisé en absence de demande utilisateur.

Impact :

Vous pourriez vous connecter à l'ESP32 et accéder à ses services.

b. Réutilisation des clés d'appairage

Problème : Si l'ESP32 réutilise les mêmes clés d'appairage pour différents appareils ou sessions.

Exploit : Capture des clés lors d'une session et réutilisation pour s'authentifier ultérieurement.

Impact : Accès aux données ou commandes sans appairage.

c. Faiblesse du mode Legacy Pairing

Problème : Si l'ESP32 utilise une ancienne méthode d'appairage (Bluetooth 4.0 ou inférieur).

Exploit : Vulnérabilité à une attaque Eavesdropping (écoute clandestine) pour capturer les clés d'appairage.

Impact : Compromission des sessions futures.

2. Failles dans les services BLE

a. Services ouverts ou non protégés

Problème : Certains services Bluetooth Low Energy peuvent être accessibles sans authentification ni chiffrement.

Exploit :

Exploration avec des outils comme gatttool ou BLEAH pour lire ou écrire dans les caractéristiques exposées.

Modification de paramètres critiques ou récupération de données sensibles.

Impact : Accès non autorisé à des fonctions ou données importantes (ex. : identifiants des élèves, commandes, etc.).

b. Manque de chiffrement des communications

Problème : Les données échangées entre l'ESP32 et les appareils connectés ne sont pas chiffrées.

Exploit : Utilisation d'un sniffer BLE pour intercepter et lire les données en clair.

Impact : Vol d'informations sensibles ou exploitation pour des attaques supplémentaires.

c. Injection de commandes (Command Spoofing)

Problème : Absence de validation des données reçues par l'ESP32.

Exploit : Envoi de fausses données ou commandes via les caractéristiques BLE accessibles.

Impact :

Manipulation du fonctionnement de l'ESP32 (par exemple, en faussant les données collectées).

Déni de service ou modification de la logique de détection.

3. Failles dans la configuration du Bluetooth

a. Public Device Name

Problème : Si l'ESP32 annonce publiquement un nom facilement identifiable ou contenant des informations sensibles.

Exploit : Utilisation de ce nom pour cibler spécifiquement le dispositif dans une attaque.

Impact : Facilitation d'une attaque ciblée (appairage, spoofing).

b. Absence de randomisation d'adresse MAC

Problème : Si l'ESP32 utilise une adresse MAC fixe pour ses communications Bluetooth.

Exploit :

Suivi de l'ESP32 dans différents environnements.

Association d'un ESP32 spécifique avec des actions ou événements.

Impact : Atteinte à la confidentialité du système.

c. Large plage de découverte

Problème : Si l'ESP32 est en mode découverte permanente.

Exploit : L'attaquant peut scanner en permanence pour détecter et interagir avec l'appareil.

Impact : Augmentation des opportunités d'attaques comme l'appairage non autorisé ou le spoofing.

4. Attaques avancées possibles

a. Replay Attack

Problème : Si l'ESP32 ne valide pas correctement l'origine des messages ou utilise des clés non renouvelées.

Exploit :

Capture d'une communication légitime et rediffusion pour réactiver ou déclencher des actions.

Impact : L'attaquant peut manipuler le système sans avoir à le compromettre complètement.

b. BLE Spoofing

Problème : Les ESP32 acceptent les connexions ou interactions d'appareils qui usurpent l'identité d'autres appareils légitimes.

Exploit :

Usurpation d'un appareil Bluetooth légitime pour accéder aux services de l'ESP32.

Impact : L'attaquant peut se faire passer pour un élève ou un capteur valide.

c. Denial of Service (DoS)

Problème : L'ESP32 ne gère pas bien un grand nombre de connexions simultanées ou des paquets non valides.

Exploit :

Envoi de requêtes répétées ou malformées pour surcharger l'ESP32.

Impact : Arrêt ou instabilité du système.

5. Contre-mesures

Pour sécuriser les ESP32 contre ces failles, voici les solutions à implémenter :

Configuration Bluetooth

Activez un mode d'appairage sécurisé (Secure Connections Only) avec PIN fort.

Randomisez l’adresse MAC utilisée par les ESP32 pour empêcher le suivi.

Limitez la durée de la fenêtre de découverte pour réduire la surface d'attaque.

Protection des services BLE

Exigez une authentification et chiffrement pour accéder aux services BLE critiques.

Mettez en place des validations des commandes reçues pour éviter les injections malveillantes.

Utilisez un chiffrement AES-CCM pour toutes les communications BLE.

Gestion des connexions

Implémentez des limitations sur le nombre de connexions simultanées.

Ajoutez un système de détection des requêtes anormales (anti-DoS).

Surveillance et audits

Surveillez les connexions Bluetooth à proximité pour détecter des anomalies.

Effectuez régulièrement des audits de sécurité pour identifier et corriger les vulnérabilités.