Bluetooth Schnittstellen - Sicherheit

Teilsicherheitskonzept – CONBOTICS MalerRoboter

Inhalt

1.) Methodik und Durchführung von Bluetooth-Sicherheitstests

1. Einleitung

Bluetooth hat sich von einer Technologie zur Verbindung von Headsets zu einem fundamentalen Protokoll für das Internet der Dinge (IoT), die Medizintechnik, die Automobilindustrie und die Unterhaltungselektronik entwickelt. Diese Allgegenwart, kombiniert mit einer komplexen und sich ständig weiterentwickelnden Spezifikation, schafft eine erhebliche Angriffsfläche. Oft werden Geräte mit unsicheren Standardkonfigurationen ausgeliefert oder es werden bekannte Protokoll-Schwachstellen nicht durch aktuelle Firmware-Updates behoben.

Dieser Bericht beschreibt eine strukturierte Methodik zur Durchführung von Sicherheitstests für Bluetooth-fähige Geräte. Ziel ist es, Schwachstellen in den Bereichen der Erkennung, des Pairings, der Authentifizierung und der Datenübertragung systematisch zu identifizieren und zu bewerten.

2. Bedrohungslandschaft und Angriffsvektoren

Die Angriffe auf Bluetooth lassen sich in mehrere Kategorien einteilen:

- **Passives Abhören (Eavesdropping / Sniffing): Ein Angreifer fängt die Kommunikation zwischen zwei Geräten ab. Dies ist besonders bei älteren Geräten, die keine starke Verschlüsselung verwenden, oder bei Verbindungen, die gänzlich unverschlüsselt sind (oft bei BLE), erfolgreich.**
- **Aktives Spoofing (Impersonation): Ein Angreifer tarnt sein Gerät als ein vertrauenswürdiges Gerät (z.B. ein bekanntes Headset oder ein Autoradio), um das Opfer zur Verbindung zu verleiten und Daten zu stehlen oder Befehle auszuführen.**
- **Man-in-the-Middle (MitM)-Angriffe: Der Angreifer platziert sich zwischen zwei kommunizierenden Geräten, fängt deren Verkehr ab, kann ihn potenziell verändern und leitet ihn dann weiter. Dies ist besonders während des Kopplungsvorgangs (Pairing) kritisch, um den Schlüsselaustausch zu kompromittieren.**
- **Denial-of-Service (DoS): Ein Angreifer überflutet ein Gerät mit Verbindungsanfragen oder stört die Funkfrequenzen (Jamming), um legitime Verbindungen zu verhindern und das Gerät unbrauchbar zu machen.**

- **Ausnutzung von Protokoll-Schwachstellen: Im Laufe der Jahre wurden zahlreiche spezifische Schwachstellen entdeckt, darunter:**
    - **BlueBorne: Eine Familie von Schwachstellen, die eine Remotecodeausführung ohne jegliche Benutzerinteraktion ermöglicht.**
    - **KNOB-Attack (Key Negotiation of Bluetooth): Ein Angriff, der es ermöglicht, die Verschlüsselungsstärke während des Verbindungsaufbaus auf ein extrem niedriges Niveau zu zwingen.**
    - **SweynTooth / BrakTooth: Familien von Schwachstellen in den SDKs vieler Chiphersteller, die zu Abstürzen oder Deadlocks führen können.**

3. Testmethodik und Phasen

Ein strukturierter Sicherheitstest folgt einem mehrphasigen Ansatz.

Phase 1: Informationsbeschaffung (Reconnaissance)

- **Ziel: Alle erreichbaren Bluetooth-Geräte entdecken, ihre MAC-Adressen, Namen und unterstützten Dienste identifizieren.**
- **Beschreibung: In dieser Phase wird passiv und aktiv nach Geräten gescannt, um eine "Karte" der Bluetooth-Umgebung zu erstellen. Es wird ermittelt, welche Informationen ein Gerät preisgibt, ohne dass eine Verbindung oder Kopplung erforderlich ist.**

Phase 2: Schwachstellenanalyse

- **Ziel: Die gesammelten Informationen auf bekannte Schwachstellen und unsichere Konfigurationen analysieren.**
- **Beschreibung: Analyse der Protokollversion, der Authentifizierungsanforderungen und (bei BLE) der Zugriffsrechte auf die angebotenen Dienste und Charakteristiken. Ist das Gerät anfällig für bekannte Angriffe? Sind sensible Daten ohne Authentifizierung lesbar?**

Phase 3: Aktive Ausnutzungsversuche (Exploitation)

- **Ziel: Die in Phase 2 identifizierten potenziellen Schwachstellen praktisch testen.**
- **Beschreibung: Durchführung gezielter Angriffe in einer kontrollierten Umgebung, um die tatsächliche Auswirkung einer Schwachstelle zu demonstrieren.**

4. Praktische Test-Szenarien

Szenario BT-TEST-01: Discovery und Fingerprinting

- **Ziel: Öffentlich gesendete Informationen eines Geräts sammeln.**
- **Tools: hcitool, bluetoothctl, btscanner (Linux), bettercap.**
- **Durchführung:**
    1. **Einen Scan nach erreichbaren Geräten durchführen: sudo hcitool scan.**
    2. **Mit bluetoothctl detailliertere Informationen über ein gefundenes Gerät abrufen.**
    3. **Bei BLE-Geräten mit sudo bettercap -iface <interface> und dem Befehl ble.recon on nach Advertisements suchen.**
- **Erfolgs-/Fehlerkriterium: Das Gerät sendet unnötig viele identifizierende Informationen (z.B. detaillierte Servicedaten) in seinen Advertisements.**

Szenario BT-TEST-02: Analyse von GATT-Services (nur BLE)

- **Ziel: Die von einem BLE-Gerät angebotenen Dienste und Charakteristiken auf unsichere Zugriffsrechte prüfen.**
- **Tools: gatttool (veraltet, aber noch nützlich), bettercap.**
- **Durchführung:**
    1. **Mit bettercap oder gatttool eine Verbindung zum Gerät herstellen, ohne es zu koppeln.**
    2. **Alle Services und Characteristics enumerieren (ble.enum <MAC-Adresse>).**
    3. **Versuchen, den Wert jeder Characteristic zu lesen (ble.read <handle>).**
    4. **Versuchen, in jede beschreibbare Characteristic zufällige Daten zu schreiben (ble.write <handle> <daten>).**
- **Erfolgs-/Fehlerkriterium: Sensible Daten können ohne Authentifizierung gelesen werden oder kritische Funktionen können durch das Schreiben auf Characteristics ohne Authentifizierung ausgelöst werden. Dies ist eine der häufigsten und kritischsten Schwachstellen bei IoT-Geräten.**

Szenario BT-TEST-03: Test der Pairing-Sicherheit

- **Ziel: Die Sicherheit des Kopplungsvorgangs bewerten.**
- **Tools: bluetoothctl, bettercap.**
- **Durchführung:**
    1. **Versuchen, das Gerät zu koppeln (pair <MAC-Adresse> in bluetoothctl).**
    2. **Beobachten: Ist ein PIN erforderlich? Wird ein fester, unsicherer PIN ("0000", "1234") akzeptiert?**
    3. **Wird "Just Works"-Pairing verwendet, bei dem keine Benutzerinteraktion erforderlich ist? Dies ist anfällig für MitM-Angriffe.**
- **Erfolgs-/Fehlerkriterium: Das Gerät verwendet einen schwachen oder keinen PIN, oder es lässt sich ohne Bestätigung koppeln.**

Szenario BT-TEST-04: Fuzzing von Bluetooth-Diensten

- **Ziel: Das Gerät durch Senden von fehlerhaften oder unerwarteten Daten zum Absturz bringen.**
- **Tools: bettercap, Scapy (für eigene Pakete).**
- **Durchführung:**
    1. **Eine beschreibbare BLE-Characteristic oder einen bekannten Bluetooth-Classic-Dienst identifizieren.**
    2. **Eine große Menge an Daten, Sonderzeichen oder fehlerhaft formatierte Pakete an diese Schnittstelle senden.**
- **Erfolgs-/Fehlerkriterium: Das Gerät stürzt ab, startet neu oder reagiert nicht mehr (Denial-of-Service). Dies deutet auf eine fehlerhafte Eingabeverarbeitung und potenzielle Buffer-Overflow-Schwachstellen hin.**

5. Werkzeuge und Ausrüstung

Software:

- **BlueZ-Stack (Linux): Enthält hcitool, hciconfig, bluetoothctl, gatttool.**
- **Bettercap: Ein mächtiges, modulares Framework für Reconnaissance und MitM-Angriffe auf WLAN und Bluetooth.**
- **Wireshark: Mit entsprechenden Plugins zum Analysieren von aufgezeichnetem Bluetooth-Verkehr.**

Hardware:

- **Standard-Bluetooth-Dongle: Für die meisten aktiven Angriffe und Scans ausreichend.**
- **Spezialisierte Hardware:**
  - **Ubertooth One: Zum passiven Abhören von Bluetooth Classic und zum Aufspüren von nicht-sichtbaren Geräten.**
  - **Btlejack / nRF5x DKs: Zum hochpräzisen Sniffing und Jamming von BLE-Verbindungen.**

6. Fazit und Härtungsstrategien

Bluetooth-Sicherheit ist ein komplexes Feld, das eine sorgfältige Implementierung erfordert. Ein Sicherheitstest deckt häufig Schwachstellen auf, die auf unsichere Standardeinstellungen oder veraltete Software zurückzuführen sind.

Wichtige Strategien zur Absicherung sind:

- **Verwendung aktueller Standards: Implementieren Sie Bluetooth 4.2 oder höher, um von den "LE Secure Connections" zu profitieren.**
- **Starke Authentifizierung: Erzwingen Sie eine sichere Kopplung (idealerweise "Passkey Entry" oder "Numeric Comparison") und vermeiden Sie "Just Works"-Pairing, wo immer möglich.**
- **Autorisierung durchsetzen: Bei BLE müssen alle Charakteristiken, die sensible Daten enthalten oder kritische Aktionen auslösen, eine Authentifizierung und Autorisierung erfordern.**
- **Minimale Informationspreisgabe: Ein Gerät sollte nur die notwendigsten Informationen in seinen Advertisements senden und nicht permanent sichtbar sein.**

- **Regelmäßige Firmware-Updates: Halten Sie die Firmware der Geräte auf dem neuesten Stand, um bekannte Protokoll- und SDK-Schwachstellen zu beheben.**