Progetto Universitario: Analisi delle Vulnerabilità e Hardening di Dispositivi IoT

Relazione Tecnica di Sicurezza IoT

09-12-2025

Introduzione e Obiettivi del Progetto

Il presente elaborato analizza le criticità intrinseche ai sistemi IoT commerciali, focalizzandosi sullo studio di un **Target** specifico: la presa intelligente **Shelly Plug S (Gen1)**.

Il progetto si propone di esplorare l'intero ciclo di vita della sicurezza del dispositivo, articolandosi in tre fasi operative distinte:

1. **Analisi delle vulnerabilità fisiche:** Estrazione del firmware originale (Dump).
2. **Sviluppo di funzionalità offensive:** Creazione e iniezione di un firmware malevolo customizzato.
3. **Implementazione di contromisure difensive:** Sviluppo di un firmware sicuro tramite tecniche di Hardening.

La scelta di analizzare un dispositivo diffuso sul mercato consumer, quale lo Shelly Plug S, mira a contestualizzare lo studio in scenari di minaccia reali, superando le simulazioni puramente teoriche e dimostrando l'impatto sulla sicurezza delle infrastrutture domestiche.

1 Fase 1: Estrazione del Firmware tramite Accesso Fisico

Questa fase illustra la metodologia attraverso la quale un attaccante, disponendo di accesso fisico al dispositivo, può comprometterne l'integrità logica.

1.1 Analisi Hardware ed Estrazione

Le operazioni previste sono le seguenti:

- **Interfacciamento Hardware:** Identificazione dei pinout **UART** sul Circuito Stampato (PCB) e connessione diretta al SoC tramite adattatore USB-UART.
- **Dump della Memoria:** Utilizzo del tool `esptool.py` per eseguire l'estrazione completa (dump) del contenuto della memoria Flash (`firmware_originale.bin`).
- **Analisi Statica e Reverse Engineering:** Il binario estratto viene sottoposto ad analisi tramite **Binwalk** per l'estrazione del filesystem e disassemblato mediante **Ghidra**, al fine di comprendere la logica applicativa e individuare eventuali informazioni sensibili, segreti o credenziali hardcoded.

2 Fase 2: Sviluppo e Iniezione del Firmware Offensivo

2.1 Analisi dei Protocolli Insicuri

Questa sezione evidenzia come l'assenza di crittografia e di meccanismi di autenticazione robusti esponga il dispositivo ad attacchi remoti.

- **Analisi del Traffico:** Sfruttando la configurazione di default, si osserva la comunicazione con il broker MQTT sulla porta **1883 (non cifrata)**. L'analisi pacchetti tramite **Wireshark** conferma che i messaggi di stato (ON/OFF) e i payload sensibili vengono trasmessi in chiaro (cleartext), rendendoli vulnerabili a intercettazione e manipolazione.

2.2 Implementazione delle Funzionalità Malevole

Utilizzando il framework `ESPHome`, viene sviluppato un firmware modificato ("implant") che, pur mantenendo le funzionalità originali per non destare sospetti, introduce vettori d'attacco avanzati:

- **Side-Channel Attack (Energy Spyware)**

  Sfruttamento della presa come sensore per dedurre le abitudini dell'utente attraverso l'analisi dei consumi elettrici.

  - **Data Collection:** Il firmware monitora in tempo reale tensione e corrente, registrando le variazioni associate a eventi semantici rilevanti (es. accensione TV: $\Delta P \approx +80W$, Power Factor: 0.9).
  - **Data Exfiltration:** I dati grezzi vengono esfiltrati via UDP verso un server C2 (Command and Control) esterno.
  - **Pattern Recognition:** I dati raccolti alimentano modelli di Machine Learning lato server per profilare la routine giornaliera dell'utente (User Behavior Analytics).

- **Network Pivoting e Ricognizione**

  Trasformazione del dispositivo IoT in un punto di accesso persistente per attacchi laterali verso la rete interna.

  - **Passive Reconnaissance (ARP Sniffing):** Il dispositivo ascolta il traffico broadcast (richieste ARP) per mappare gli host attivi nella rete locale in modalità totalmente passiva, garantendo l'invisibilità.
  - **Active Scanning (Low-and-Slow):** Una volta identificati i target, viene eseguito un port scanning mirato e a bassa frequenza sulla sottorete (es. 192.168.x.x). Lo script ricerca servizi critici esposti (Porte 22 SSH, 80 HTTP, 445 SMB), trasmettendo i risultati all'attaccante esterno.

2.3 Attacco MITM e Iniezione OTA

L'obiettivo è l'installazione remota del firmware malevolo sfruttando il meccanismo di aggiornamento Over-The-Air (OTA) non sicuro del dispositivo (HTTP).

La catena d'attacco si svolge come segue:

- **ARP Poisoning:** L'attaccante inquina la cache ARP della rete locale, impersonando il Gateway/Router.
- **Traffic Redirection:** Tutto il traffico generato dalla presa viene reindirizzato attraverso la macchina dell'attaccante.
- **HTTP Hijacking:** Quando il dispositivo richiede un aggiornamento (es. `http://.../firmware.bin`), l'attaccante intercetta la richiesta GET e inietta il proprio binario (`malware.bin`) nella risposta.

Il dispositivo, non verificando l'integrità della fonte, installa ed esegue il firmware compromesso, garantendo all'attaccante il controllo persistente.

3 Fase 3: Hardening e Creazione del Firmware Sicuro

La fase difensiva mira alla mitigazione delle vulnerabilità esposte precedentemente, implementando i principi di confidenzialità, integrità e autenticità su hardware vincolato.

3.1 Secure Boot e Verifica dell'Integrità

Implementazione di contromisure contro l'iniezione di codice non autorizzato.

- **Firmware Signing:** Il firmware legittimo viene firmato digitalmente dallo sviluppatore utilizzando una chiave privata.

- **Secure Bootloader:** Il bootloader viene configurato per contenere la **Chiave Pubblica** corrispondente. Ad ogni ciclo di avvio o tentativo di aggiornamento OTA, il sistema verifica la firma digitale σ del binario.
- **Mitigazione MITM:** Questo meccanismo rende inefficace l'attacco descritto nella Fase 2. Un attaccante, non possedendo la chiave privata, non potrà generare una firma valida per il firmware malevolo; di conseguenza, il bootloader rifiuterà l'installazione del pacchetto compromesso.

3.2 Sicurezza del Canale di Comunicazione (MQTTs)

- **Crittografia del Trasporto:** La comunicazione verso il Broker MQTT viene migrata sulla porta **8883** implementando il protocollo **TLS/SSL**.
- **Mutual Authentication (mTLS):** Viene configurata l'autenticazione mutua tramite certificati X.509. Sia il client (presa) che il server (broker) verificano reciprocamente la validità dei certificati, garantendo che solo i dispositivi autorizzati possano interagire con l'infrastruttura.