Laborator 9: Reducerea Suprafeței de Atac în Aplicații Embedded (ESP32)

Echipa 3

May 22, 2025

1. Identificarea corectă a punctelor de expunere ale aplicației (25%)

Suprafața de atac identificată

Componentă	Descriere	Risc potențial
WiFi SSID/Password	Hardcodate în cod sursă ESP32	Expunere la interceptare sau brute-force
MQTT (port 1883)	Folosit în unele fișiere .ino fără criptare TLS	Trafic nesecurizat, posibilitate de MITM
Topice MQTT globale	Comenzi primite fără autentificare/ACL-uri	Abuz prin comenzi neautorizate (ex: activare cameră)
Certificate TLS	Includere directă în cod sursă	Posibilitate de scurgere a cheilor dacă sursa e compromisă
Acces cameră live	Activabil prin topic MQTT	Expunere a imaginilor video fără control de acces
Broker MQTT	IP fix, fără autentificare în unele cazuri	Expunere în rețea locală fără protecție suplimentară

Tool-uri utilizate pentru analiză

- OWASP ZAP – pentru testare web (dacă se adaugă interfață HTTP)
- Nikto – pentru scanare servere web în viitor
- Trivy / Grype – folosite pentru containere Mosquitto (acolo unde se aplică)
- Wireshark – pentru analiză pachete MQTT (securizare traficului)

2. Implementarea măsurilor de reducere a suprafeței de atac (30%)

Măsuri aplicate în codul sursă și infrastructură

- **Eliminare componente neesențiale**: dezactivarea funcționalităților inactive (mod live, flash neutilizat).
- **Autentificare MQTT**: activarea autentificării cu username/parolă pentru fiecare dispozitiv ESP32.
- **Limitare acces topicuri**: configurare ACL pe broker pentru fiecare topic (ex: dev123/commands).
- **Rate limiting**: introducere limitări de frecvență pentru trimitere de imagini (ex: max 1/sec).
- **Validare comenzi**: parsarea JSON cu validare completă (valori acceptabile, formate).
- **Izolare topice MQTT**: separare clară între topicuri de comandă și status.
- **Stocare certificatelor**: mutarea certificatelor TLS în SPIFFS sau criptarea acestora.

3. Aplicarea securizării containerelor și infrastructurii (30%)

Mediu containerizat auxiliar (ex: broker Mosquitto în Docker)

- **Imagine minimală**: utilizarea Alpine Linux pentru containerele Docker, reducând dependențele inutile.
- **Utilizator non-root**: rularea containerelor cu un UID/GID non-privilegiat în loc de root.
- **Politici de izolare**: activarea profilurilor AppArmor și Seccomp în Docker pentru limitarea accesului la sistemul de operare.
- **Permisiuni restrânse**: montarea volumelor ca `read-only` acolo unde este posibil.
- **Limitarea porturilor**: expunerea doar a porturilor strict necesare (ex: doar 1883 pentru MQTT).

Măsuri suplimentare pentru ESP32

- **Separare fizică a rețelelor**: rularea ESP32 într-o rețea VLAN dedicată pentru IoT.
- **Limitarea firmware-ului**: eliminarea librăriilor și funcțiilor neutilizate pentru a reduce vectorii de atac.

- **Semnarea firmware-ului**: opțional, adăugarea verificării semnăturii binare la boot.

4. Integrarea măsurilor în pipeline-ul CI/CD (15%)

- **GitHub Actions**: configurare workflow care verifică codul C++/Arduino cu `arduino-lint` sau `cppcheck`.
- **Dependabot**: activare pentru actualizări automate ale bibliotecilor utilizate în proiect (ex: Adafruit, PubSubClient).
- **Scanare SAST**: integrare cu tool-uri de static analysis pentru identificarea vulnerabilităților în codul sursă.
- **Policy de patching**: revizie lunară a dependențelor, însoțită de un changelog pentru fiecare versiune de firmware.