Gestione sicurezza per dati sensoristici in applicazioni Cloud-Based

Preambolo

L'obiettivo generale del progetto è di garantire la protezione delle informazioni da rivelazioni, manomissioni o cancellazioni non autorizzate di un sistema di interconnessione fra più AMR ed utenti registrati su un comune Cloud-Storage service. Nel dettaglio il progetto nella fase iniziale e nei futuri sviluppi prevede i seguenti punti:

- Le macchine sono assimilabili a computer industriali multicore con due sistemi operativi a bordo: un sistema operativo real-time proprietario, e Ubuntu con middleware ROS;
- I dispositivi sono dotati di connettività WiFi e la rete di telecomunicazioni di interesse è non cablata;
- Il delay per la comunicazione fra i due sistemi operativi in parallelo è inferiore al secondo, mentre per la comunicazione con il server su cloud potrebbe raggiungere i 10 minuti;
- Inizialmente verrà considerata una comunicazione unidirezionale dalle macchine al server, in futuri sviluppi potrebbe essere prevista l'istanziazione di un serversocket su AMR;
- In futuro sarà necessario tenere conto di potenziali manomissioni a bordo macchina, quali reinstallazione forzata della distro Ubuntu, copia e analisi offline del sistema per forgiare attacchi Ad-Hoc e sfruttare l'AGV manomesso;
- Il caricamento dei segreti crittografici sulle macchine viene inizialmente ipotizzato manuale, in loco o con l'ausilio di una sessione SSH, in seguito questa assunzione verrà rivisitata;
- Sarà previsto un database su server sul quale verranno memorizzati i record delle macchine in forma cifrata, garantendo la confidenzialità e l'autenticità dei dati.

Fase iniziale

Passando a descrivere nel dettaglio la fase iniziale, partendo dalla lista dei punti sopra elencati è necessario imporre alcuni assunzioni:

- Il sistema a bordo macchina viene difeso da attacchi passivi ma ritenuto non manomissibile: non è necessario tener conto di eventuali manomissioni dirette alle apparecchiature, limitandosi per ora alla cifratura dei dati sensibili sulle macchine, delle comunicazioni su rete non cablata e su cloud;
- La comunicazione fra macchine e server è monodirezionale: solo la macchina può contattare il server instaurando un socket attivo, ad oggi non è previsto il contrario, questo impone che le macchine e gli utenti non possano comunicare direttamente fra di loro, solo tramite la mediazione del server, inoltre gli AGV non possono ad oggi ricevere comunicazioni non previste dal client a bordo;
- I segreti crittografici possono essere caricati solo manualmente sulle macchine, senza prevedere un sistema automatico, con questo vincolo si assume che l'unico modo per intercettare i segreti sia la diretta manomissione della macchina, già negata dalla precedente assunzione, o la violazione del protocollo SSH o simili, fuori dal perimetro progettuale;
- I dati memorizzati su cloud restano cifrati, mantenendo non cifrato solo un riferimento temporale e alla configurazione usata per registrare il log, questo impedisce query complesse, limitando i parametri di ricerca disponibili, permette di scaricare e decifrare con l'ausilio di un estensione i record.

Con tali assunzioni a mente, inizialmente il progetto si concentra sulla cifratura e il formato dei dati inviati su rete non sicura, sulla struttura dei log salvati su cloud e le query per prelevare i record di questi ultimi.

Vincoli e Criticità

Il progetto rappresenta un importante passo iniziale, tuttavia molte problematiche sono ancora aperte e dovranno essere affrontate nelle fasi successivi:

- Non è prevista alcuna protezione dalla diretta manomissione del server o del sistema a bordo macchina: in caso di manomissione la sicurezza verrebbe compromessa non essendo possibile rivelare le modifiche apportate al client, si fa quindi affidamento alla sicurezza del sistema operativo che ospita il codice, fuori dal perimetro progettuale;
- La comunicazione monodirezionale impedisce al server di inviare puntualmente comunicazioni critiche o impreviste alle macchine e la comunicazione diretta fra AGV e utenti;
- L'iniezione manuale dei segreti crittografici sulle macchine, in caso di errore al sistema di un AMR, volontario o involontario, richiederebbe una lavorazione del guasto da parte di un operatore fidato, senza che questa possa essere automaticamente sanata;
- I dati memorizzati sul database come sommariamente descritto impediscono l'esecuzione di query complesse e riducono il server ad un sistema di backup.

Dei quattro punti elencati l'unico che a prima analisi prevede una criticità futura è il database che andrebbe rivisitato per ammettere query più complesse, per quanto riguarda gli altri punti:

- La rivelazione di manomissioni dirette al sistema, qualunque sarà la tecnica adottata, dovrebbe risultare un problema trasversale a quelli inizialmente in esame;
- La comunicazione bidirezionale richiederà l'istanza di un socket passivo sulle macchine con un'espansione del sistema di comunicazione che non dovrebbe entrare in conflitto con il sistema sviluppato finora;
- L'iniezione automatica dei segreti crittografici sulle macchine sarebbe, in ogni caso, una comunicazione sicura parallela a quelle già definite.
- La trasmissione delle direttive per il controllo di flotta e lo scambio di dati sensoristici fra AGV potrà seguire il modello proposto dalle funzionalità ad oggi implementate, associando come già previsto delle chiavi private e pubbliche ai dispositivi.

Cenni sull’implementazione del sistema

Il sistema prevede il controllo e l’analisi dei dati sensoristici di più dispositivi distribuiti fra più imprese. Garantisce un controllo degli accessi alle risorse del RESTful web service basato su ruoli ed aree di appartenenza di utenti e dispositivi registrati e crittografia end-to-end per i record degli AGV. Il progetto si concentra su confidenzialità ed integrità dei dati:

- Per garantire la crittografia end-to-end fra dispositivi e utenti si fa uso di cifratura a chiave simmetrica e asimmetrica (AES-256, RSAES-OAEP): un estensione cross-browser conserva la chiave privata dell’utente, protetta da una master-password, mentre il database ospita la chiave pubblica; lato macchina ad ogni nuovo record una sua descrizione e il contenuto vengono cifrati con una nuova chiave simmetrica scelta casualmente, con le chiavi pubbliche di ogni destinatario viene cifrata una copia della chiave segreta, la descrizione e il dizionario delle chiavi vengono caricati sul server assieme al contenuto del record. Non è prevista per ora la cifratura delle configurazioni dei record, facilmente implementabile seguendo lo stesso schema sopra riportato;
- Per garantire il principio del privilegio minimo, oltre a limitare le risorse disponibili sulla base dei ruoli di utenti e AGV, l’accesso viene gestito tramite ruolo ed area di appartenenza, organizzati gerarchicamente; allo stesso modo le query al database ammesse sono distinte in base al ruolo minore definito dalla risorsa e dall’utente;
- Per evitare che il server condivida alcun segreto con gli utenti e i dispositivi registrati, per autenticare l’accesso alle risorse si prevede un semplice Basic authorization header, mantenendo aperta una sessione con dei cookie d’autenticazione che contengono lo username ed una one time password, i dati d’accesso degli utenti sono opportunamente custoditi con hash e salted-hash, per custodire le credenziali d’accesso è possibile adottare l’uso di una KDF, gestita dal software a bordo macchine e dall’estensione cross-browser per gli utenti. Questo impone l’uso del protocollo TLS su HTTP già menzionato;
- Il server è predisposto per il registro delle attività senza prevedere ad oggi alcun sistema di periodica cancellazione o di cifratura degli activity log.

Note tecnologiche

Il progetto richiede una comunicazione client-server-db tramite il protocollo HTTP/TLS, affidando a questo confidenzialità ed autenticità dei dati trasmessi e la predisposizione di un Cloud Storage Service. I linguaggi di programmazione adottati finora sono i seguenti:

- Java con API JAX-RS e framework Jersey per il back-end del server con database relazionale MySQL;
- Javascript per l'estensione cross-browser che custodisce i segreti degli utenti ed esegue funzioni crittografiche sui dati, con l'ausilio del framework Angular.js e HTML per lo sviluppo del front-end e la GUI su browser;
- Python per il client su AGV che interfaccia il middleware ROS con il server ed esegue le funzioni crittografiche richieste.

Possibili sviluppi futuri

Tra i vincoli accennati nei paragrafi precedenti è possibile esplorare gli sviluppi futuri del progetto, espandendone il perimetro.

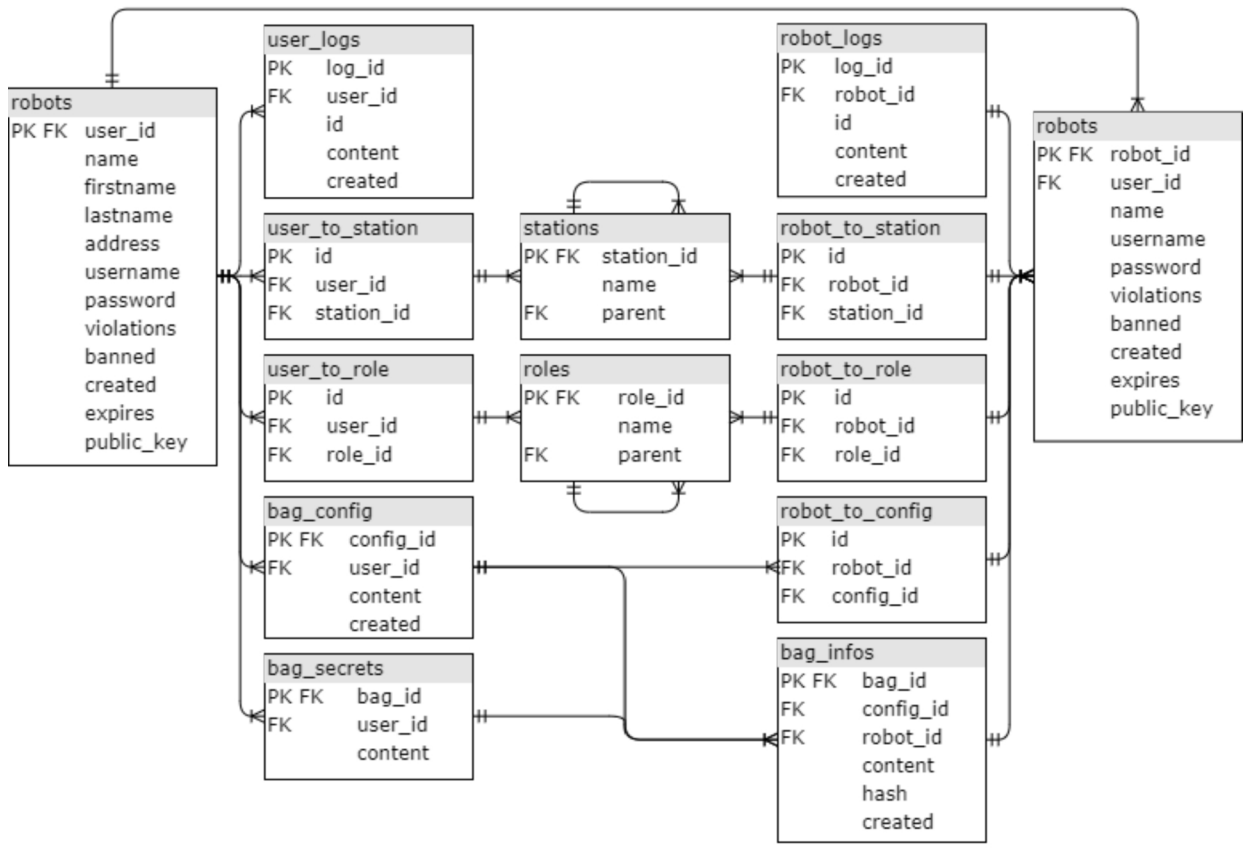

Diagramma EER del database