SICUREZZA IN AMBITO AUTOMOTIVE: VULNERABILITÀ, MINACCE, ATTACCHI E POSSIBILI CONTROMISURE

Autori: Marco Deano (Matricola: VR503057)
Mattia Bernardi (Matricola: VR502842)

INTRODUZIONE

Negli ultimi anni, il mondo dei trasporti è stato testimone di una profonda rivoluzione, dovuta all'avanzamento tecnologico delle auto a guida autonoma, dette anche CAV (Connected and Autonomous Vehicles).
Questa rivoluzione non si limita semplicemente all'introduzione di nuovi veicoli, ma abbraccia un complesso ecosistema che coinvolge tecnologie innovative, strategie di sicurezza, e nuovi paradigmi di gestione delle infrastrutture stradali. I progressi tecnologici nell'industria automobilistica, quindi, hanno portato i veicoli moderni a diventare dei concentrati di dati. Infatti, essi:

- Generano,
- Consumano,
- Trasmettono costantemente grandi quantità di dati.

L'analisi e l'utilizzo di questi dati offrono molte opportunità, come il miglioramento delle funzioni del veicolo e la generazione di nuovi flussi di reddito (quindi nuove forme di guadagno). Tuttavia, questo passaggio a un'industria basata sui dati, comporta diverse sfide e responsabilità per il settore automobilistico. In questo senso, durante il documento andremo a trattare una grande problematica del settore automobilistico, ovverosia: essendo i dati monetizzabili, vi sono molteplici minacce alla sicurezza informatica e conseguentemente, possono essere attirati criminali informatici, i quali potrebbero tentare di rubare i dati delle auto e sfruttarli per scopi illegali ed economici [1].

Importanza delle auto a guida autonoma

Le auto a guida autonoma promettono di rivoluzionare il concetto stesso di mobilità, offrendo un potenziale impatto significativo sull'efficienza dei trasporti, sulla sicurezza stradale e sull'esperienza di guida. Questa trasformazione è alimentata dall'obiettivo di ridurre la congestione del traffico, prevenire incidenti stradali e migliorare l'accessibilità al trasporto per diverse fasce di popolazione. Molte persone vedono questa rivoluzione nel settore dei trasporti come il risultato dei progressi tecnologici degli ultimi anni. L'introduzione delle auto a guida autonoma rappresenterà un miglioramento significativo delle prestazioni del settore dei trasporti, offrendo una maggiore efficienza e sicurezza dei mezzi di trasporto. I CAV saranno automatizzati e useranno la tecnologia dei sensori per migliorare l'efficienza dei trasporti e la gestione delle reti stradali. Questi veicoli saranno in grado di automatizzare le operazioni umane, sostituendo i conducenti in diverse situazioni di guida. Ciò permetterà alle auto di guidare in modo autonomo, migliorando la sicurezza stradale per pedoni e automobilisti.

La crescita dei veicoli a guida autonoma è stata motivata dalla necessità di creare un'infrastruttura di trasporto globale, che garantisca mezzi di trasporto sicuri, veloci e affidabili. L'evoluzione dei CAV richiederà certamente una proliferazione di beni ad alta tecnologia, in quanto essi sono dotati di

tecnologie avanzate come: telecamere, radar e altre antenne (che sostituiscono gli specchietti retrovisori).

Tutto questo, però, avrà come vantaggio che tali veicoli saranno in grado di compiere manovre autonomamente, riducendo lo stress e la fatica dei conducenti. Di conseguenza, le auto a guida autonoma avranno la necessità di integrare sistemi avanzati di assistenza alla guida (ADAS avanzati), tecnologie di parcheggio automatico e applicazioni di controllo del traffico e di sicurezza stradale, al fine di migliorare l'efficienza e la sicurezza del trasporto. Queste applicazioni richiedono una struttura condivisa per consentire la loro efficace implementazione. Ad esempio, l'integrazione tra le applicazioni di controllo del traffico e quelle di controllo del parcheggio può aiutare a ridurre la congestione stradale facilitando la ricerca di un posto auto disponibile.

I sensori, invece, sono un componente chiave di queste applicazioni, in quanto rilevano la presenza di pedoni e veicoli sulla strada, contribuendo così a evitare incidenti. I sensori RSU (Unità di segnalazione stradale) sono in grado di rilevare ogni movimento sia sulla strada che sui marciapiedi, fornendo un'indicazione tempestiva di eventuali rischi di incidente. Quindi, l'utilizzo di sensori di alta tecnologia e una migliore comunicazione tra i veicoli, contribuiranno a rendere il trasporto più sicuro ed efficiente, con la conseguenza diretta, che avranno un impatto positivo sul settore dei trasporti a livello mondiale.

Tuttavia, la tecnologia delle auto a guida autonoma comporta anche rischi e minacce, come gli attacchi informatici da parte di virus, bug e hacker. Per garantire l'affidabilità e la protezione dei dati, le auto a guida autonoma sono dotate di avanzate crittografie e sistemi di sicurezza dei dati. Nonostante ciò, i CAV rappresentano ancora un rischio elevato di essere esposti a minacce, in quanto gli attacchi possono verificarsi su tutti i dispositivi tecnologici in essi installati.

È dunque importante affrontare e mitigare i rischi associati alla tecnologia delle auto a guida autonoma, per garantire un'implementazione sicura e affidabile di queste nuove tecnologie nel settore dei trasporti [2].

Ecosistema delle auto a guida autonoma

L'industria automobilistica sta abbracciando sempre più la connettività, trasformando le auto in vere e proprie "smart car". Le applicazioni di terze parti, collegate al cloud, svolgono un ruolo fondamentale nell'esperienza di guida e di comfort dei passeggeri. Questa tendenza è iniziata a partire dalle case automobilistiche di lusso, che hanno eliminato i pulsanti fisici per passare a cruscotti digitali. Oggi, infatti, anche i veicoli di fascia media offrono cruscotti smart. Questi non solo gestiscono le funzioni tradizionali dell'auto (come il controllo del clima o il cockpit), ma anche applicazioni come: mappe, internet radio, browser web, social media, messaggistica e assistenti virtuali. Vediamo la Figura 1.1, per capire meglio l'ecosistema del veicolo connesso al cloud:

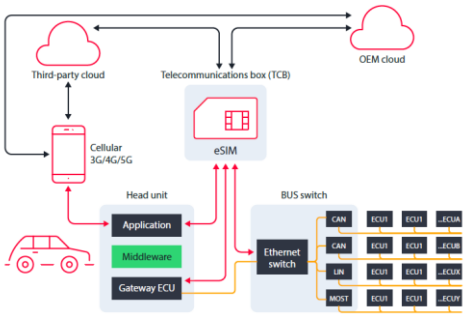


Figura 1.1: Visione astratta ecosistema veicolo

L'unità principale (ovvero l'Head unit) supporta l'esecuzione di diverse applicazioni. C'è poi un livello middleware, il quale astrae i dettagli E/E del veicolo (che comprendono tutti i sistemi elettrici ed elettronici che contribuiscono al funzionamento, al controllo e al monitoraggio di varie funzioni all'interno di un veicolo) e rende più facile agli sviluppatori creare applicazioni basate sull'auto. Il middleware può comunicare con l'ECU gateway (si tratta di un'unità di controllo elettronico che funge da punto di comunicazione centrale tra diversi sistemi elettronici presenti all'interno del veicolo. La sua principale funzione è facilitare lo scambio di dati e informazioni tra le varie unità di controllo elettroniche (ECU) presenti nel veicolo) e dà accesso alle API alle applicazioni che devono inviare messaggi all'ECU. C'è poi il bus switch, il quale instrada i pacchetti verso le corrette centraline di destinazione. Le app possono parlare con il cloud OEM (ovvero la piattaforma cloud, che fornisce servizi specifici del produttore automobilistico) o con il cloud di terze parti (come Netflix e Google) attraverso la connessione del cellulare o attraverso la SIM elettronica integrata (eSIM) nella TCU. A seconda dell'architettura E/E dell'auto, l'ECU gateway può anche comunicare direttamente con i servizi cloud.
Man mano che le auto diventano sempre più connesse e smart, si assiste allo sviluppo di applicazioni specifiche per l'auto, le quali probabilmente non avranno bisogno in futuro del middleware per accedere alla centralina del gateway. Le API middleware creano un ricco ecosistema per le auto smart, ma creano anche delle nuove opportunità per i cybercriminali, che possono sfruttare un facile accesso alle API per attaccare i sistemi delle auto, come l'architettura E/E o le centraline. Questo può portare alla diffusione di malware come: trojan, ransomware e botnet, i quali possono essere installati tramite, per esempio, attacchi di phishing o distribuiti attraverso l'utilizzo di telefoni jailbroken, collegati alle auto [1].

Minacce e requisiti di sicurezza

Lo standard ETSI ITS è stato implementato in diversi paesi europei per migliorare la sicurezza e la qualità del traffico stradale. Tuttavia, essendo basato su comunicazioni wireless, ci sono diverse minacce che possono interferire con il suo funzionamento e causare incidenti. La Tabella 1.2 fornisce un elenco di possibili minacce alla sicurezza, che possono colpire i diversi componenti del sistema ITS (come ad esempio: la manipolazione dei segnali stradali o la falsa reazione dei veicoli):

ITS components	Possible threats	Direct impact	Hazardous situations created
Infrastructure sign	Change/ add/ remove road signs (e.g., speed limit, messages)	False/ No reaction,	Traffic disturbance, collision, and congestions
Radar/Camera	Creating blind spot and presenting false image	False reaction	Driver disturbance
GPS	Spoofing and jamming	Inaccurate location information and wrong maneuver	Traffic disturbance and crash hazard
In-vehicle devices	Malware and head unit attack	Depends on malware capability	Serious traffic congestions and driver/traffic disturbance
Acoustic sensors	Interference and fake sound	False positive/negative obstacle detection and sensor malfunction	Traffic disturbance and low/high speed crash
Lidar	Jamming and smart material (absorbent, reflective)	False detection and degraded Lidar performance	Loss of situation awareness and traffic disturbance
In-vehicle sensors	Eavesdropping and malware	Privacy leak, reverse engineering and false message generation	Traffic disturbance, disabling vehicle automation service and accident
Infrastructure (RSU)	Denial of Service and fake WSA (RSA, SPAT)	Wrong notification to driver, wrong detection and no information for ITS	Traffic disturbance, safety issues and critical incident

Tabella 1.2: Elenco minacce alla sicurezza

Queste minacce possono portare a situazioni pericolose (come collisioni e congestioni) e di conseguenza, per prevenire tali minacce, sono stati identificati diversi requisiti di sicurezza. La distribuzione efficace delle strutture ITS nelle applicazioni pratiche richiede diversi requisiti di sicurezza, al fine di garantire che le comunicazioni sicure producano esperienze di guida sicure. I requisiti di sicurezza sono i seguenti:

- Autenticazione → si tratta delle disposizioni di sicurezza ITS fondamentali, che si suddividono in:
    - Verifica dell'utente;
    - Verifica della fonte, al fine di assicurarsi che i messaggi siano stati prodotti da unità ITS autentiche.
    - Verifica della posizione, al fine di protegge l'affidabilità e la significatività dei dati attuali.
- Integrità dei dati → tutte le unità dell'ITS devono essere in grado di verificare e autenticare l'affidabilità delle comunicazioni, per limitare qualsiasi operazione illecita.
- Riservatezza dei dati → i messaggi scambiati devono essere ben codificati e protetti, per evitare la perdita di dati sensibili a parti non autorizzate.
- Privacy e anonimato → l'unicità dei proprietari di auto e delle auto non dovrebbe essere visibile dai canali di comunicazione automatizzati.
- Disponibilità → i dati scambiati devono essere gestiti e preparati istantaneamente.7
- Tracciabilità e revoca → gli istituti ITS dovrebbero essere in grado di rintracciare le unità ITS, che abusano delle strutture ITS e di revocarle rapidamente.
- Autorizzazione → è essenziale delineare un regolamento di accesso stabilito sui privilegi di autorizzazione per le diverse unità ITS. È necessario applicare procedure specifiche per l'accesso o negare l'accesso a gruppi ITS specifici, a compiti individuali e all'uso delle informazioni.
- Non ripudio → tutte le unità ITS devono essere collegate esclusivamente ai propri dati e alle proprie attività per ottenere validità e inizializzazione dei dati.
- Robustezza contro gli attacchi esterni → le unità ITS devono essere contro i diversi attacchi periferici e il software dell'ITS deve essere privo di vulnerabilità [2] [3].

Attacchi recenti nel mondo automotive

Fin dalle prime fasi dei test di guida autonoma dei veicoli, sono stati riscontrati vari tipi di attacchi alle diverse unità del veicolo, come l'unità di misura interna, il Lidar, il GPS, la telecamera, l'unità di monitoraggio dei propulsori, l'AU e i messaggi di avviso. Sono stati identificati diversi tipi di attacchi potenziali che potrebbero causare gravi conseguenze, come incidenti e compromissione della sicurezza stradale. È importante prendere in considerazione questi problemi di sicurezza e sviluppare soluzioni per proteggere le auto a guida autonoma e le comunicazioni tra veicoli. Riportiamo, quindi, di seguito alcuni attacchi recenti avvenuti nel mondo dell'automotive:

- Attacchi malware → la prima intrusione remota su un veicolo mediante controlli cibernetici è stata eseguita nel 2011 su una Chevrolet Chevy Malibu. L'attacco è stato effettuato manipolando la radio del veicolo attraverso una vulnerabilità nello stack Bluetooth e utilizzando codici malware sincronizzati con i telefoni cellulari. Dopo che la radio è stata compromessa, un sistema gateway ha scollegato l'intruso dalla rete CAN ad alta velocità. Tuttavia, potevano riutilizzare questo gateway dalla loro rete CAN a bassa velocità. Capiamo, allora, che la rete CAN può operare a diverse velocità e spesso si fa riferimento a due

principali categorie: alta velocità (High-Speed CAN) e bassa velocità (Low-Speed CAN). Queste due reti sono progettate per soddisfare diverse esigenze di comunicazione all'interno di un veicolo. In particolare:

- La rete CAN ad alta velocità è progettata per gestire flussi di dati ad alta velocità e viene utilizzata per comunicazioni critiche e veloci tra i sistemi, che richiedono un trasferimento rapido dei dati (come il controllo del motore o della trasmissione);
- La rete CAN a bassa velocità è progettata per gestire flussi di dati a velocità più basse e viene utilizzata per comunicazioni meno critiche, che non richiedono una trasmissione veloce dei dati (come i sistemi di infotainment e multimediali).

In seguito, il codice inserito ha permesso di inviare messaggi alla centralina del veicolo, che potevano bloccare i freni. Inoltre, la diagnostica di bordo (OBD) è una delle parti più vulnerabili delle auto a guida autonoma agli attacchi malware. Gli autori di uno studio del febbraio 2014, hanno dimostrato che un attaccante potrebbe utilizzare uno strumento diagnostico infettato da malware per inserire malware nelle centraline attraverso l'OBD, compromettendo la sicurezza delle auto a guida autonoma.

- Attacchi man-in-the-middle → le auto a guida autonoma utilizzano la comunicazione wireless per interagire con altri veicoli e infrastrutture stradali, ma sono vulnerabili agli attacchi informatici. In particolare, gli attacchi di tipo man-in-the-middle permettono a un aggressore di manipolare i messaggi tra le entità coinvolte. Questo tipo di attacco è stato esemplificato nel 2015 quando degli hacker sono riusciti ad assumere il controllo di una Jeep Cherokee, sfruttando una vulnerabilità del sistema di comunicazione. Gli hacker sono riusciti a modificare i messaggi trasmessi tra le centraline e il sistema frenante, in modo da influenzare lo sterzo, i freni e l'accelerazione del veicolo. L'unità di elaborazione centrale del veicolo non è stata in grado di rilevare la manipolazione dei messaggi. Questo dimostra la necessità di rafforzare la sicurezza delle auto autonome per evitare possibili attacchi informatici che potrebbero mettere a rischio la sicurezza degli occupanti e degli altri utenti della strada.
- Attacchi Sybil → gli attacchi di Sybil sono problemi che possono causare malfunzionamenti nei sistemi di rete dei veicoli autonomi, impedendo loro di trasmettere dati e di rilevare gli attacchi in corso. Nel 2018, l'auto di Google è stata vittima di un attacco Sybil in cui i falsi nodi hanno inviato informazioni fuorvianti sulla posizione e sulle condizioni del traffico, causando il malfunzionamento del veicolo.

Di seguito nel documento verranno dunque presentate due tipologie di attacchi e possibili contromisure da adottare, inerenti al mondo dell'automotive:

- Attacchi al sistema di infotainment
- Attacchi ai sistemi di "road safety"

Chiaramente sono due tiplogie di attacchi che non ricoprono tutti quelli possibili nel mondo dell'automotive, però sono importanti in quanto fanno comprendere molto bene il contesto in cui le auto di oggi e del futuro lavorano [2].

ATTACCHI AL SISTEMA DI INFOTAINMENT

Panoramica del sistema di Infotainment

Il sistema di Infotainment High-Performance Computing (HPC) svolge un ruolo importante nelle automobili moderne, in quanto offre funzioni avanzate come: musica, navigazione, comunicazione ed intrattenimento. Utilizzando diverse tecnologie come la rete Wi-Fi, la connettività cellulare, NFC e Bluetooth, il sistema di Infotainment garantisce una costante connessione a Internet per l'accesso alle informazioni. Da queste prime frasi, possiamo immediatamente capire che HPC è:

1. Dotato di informazioni integrate → il sistema HPC è in grado di presentare o incorporare dati, provenienti da diverse fonti, in modo coerente e accessibile attraverso il suo sistema di intrattenimento e informazione;
2. Progettato per migliorare la sicurezza e la comodità dei conducenti e dei passeggeri dei veicoli automobilistici → in questo senso, infatti, gli utenti possono comodamente utilizzare tale sistema con una sola mano, continuando a porre l'attenzione sulla strada.

Come abbiamo detto sopra, l'integrazione di informazioni e tecnologia all'interno di questo sistema, avviene attraverso una moltitudine di fonti, come ad esempio:

- Dispositivi mobile dei passeggeri;
- Veicoli circostanti;
- Server remoti;
- Infrastrutture del traffico (semafori, telecamere di sorveglianza del traffico, segnaletica stradale).

Chiaramente, da un punto di vista puramente umano, l'integrazione di un sistema di Infotainment HPC all'interno di un autoveicolo, può comportare diversi vantaggi, come ad esempio:

- Accesso a varie informazioni (in quanto il veicolo è sempre connesso a Internet);
- Navigazione GPS;
- Assistenza alla guida;
- Integrazione con lo smartphone;
- Aggiornamenti software (in quanto alcuni sistemi di Infotainment possono ricevere aggiornamenti software, permettendo ai produttori di correggere bug, migliorare le prestazioni di guida e aggiungere nuove funzionalità nel tempo).

Tuttavia, vedendo questi aspetti da un punto di vista di software security, possiamo immediatamente cogliere il concetto, che l'interconnessione dei servizi con le automobili aumenta le vulnerabilità della sicurezza e infatti, sempre più frequentemente si verificano incidenti di hacking delle auto. Questo ha portato ad un'attenzione crescente verso la sicurezza nei veicoli automobilistici, dato che il sistema di Infotainment del veicolo si collega ad una rete complessa, la quale a sua volta è composta da:

- Internet → consente aggiornamenti in tempo reale della navigazione, servizi di streaming e aggiornamenti software over-the-air;
- Reti interne del veicolo (VAN) → collegano le unità di controllo elettronico (ECU) e quindi, assicurano uno scambio di dati efficiente tra i diversi componenti del veicolo;

- Reti di telecomunicazione → utilizzano la rete cellulare per la diagnostica remota e la localizzazione del veicolo, collegandosi al Cloud e utilizzando il GPS per accedere alla posizione del veicolo;
- Sensori dell'auto;
- Tecnologia wireless → il sistema di Infotainment In-Vehicle (IVI) sfrutta i servizi di rete all'interno del veicolo, come la connettività Wi-Fi, per creare una connessione tra il veicolo e l'ambiente esterno. In altre parole, l'IVI utilizza la tecnologia Wi-Fi (o altri servizi di rete all'interno dell'auto), per consentire al veicolo di comunicare con i dispositivi esterno o per accedere a risorse online. Questo tipo di connettività consente al sistema IVI di offrire una serie di funzionalità, come ad esempio: accesso ad Internet oppure servizi di streaming.

La composizione della rete, a cui il sistema di Infotainment si collega, può essere osservata meglio graficamente nella Figura 1:

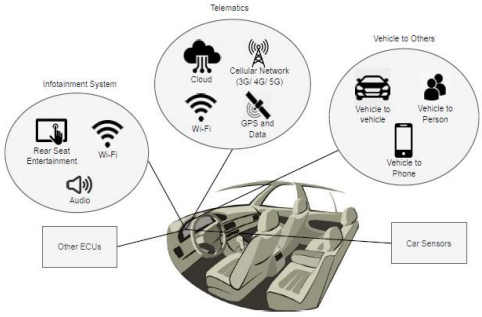


Figura 1: Composizione rete a cui si collega il sistema di Infotainment

Gli attaccanti, quindi, potrebbero tentare di accedere al sistema e manipolare la configurazione o accedere (da remoto) ai dati sensibili dell'utente. Questo rappresenta un grave problema di sicurezza (che analizzeremo più approfonditamente nei seguenti sottocapitoli) e per dare un'idea della portata, riportiamo due vulnerabilità identificate nei servizi del sistema IVI:

1. Un attaccante ha tentato di ottenere i privilegi di root e di stabilire un accesso remoto attraverso l'interfaccia Wi-Fi. Tale accesso può portare alla manipolazione della configurazione del sistema e l'attaccante potrebbe accedere ai dati sensibili dell'utente. Inoltre, poiché gli utenti possono accedere alle informazioni personali attraverso il Bluetooth durante la guida, quest'ultimo può anche essere una superficie di attacco per l'attaccante;
2. Le applicazioni integrate nei veicoli devono affrontare notevoli sfide in termini di sicurezza, soprattutto per quanto riguarda la comunicazione tra i componenti. Esiste la possibilità che applicazioni dannose possano manipolare o eludere il sistema, mettendo a repentaglio i dati sensibili degli utenti attraverso potenziali accessi non autorizzati. Una potenziale fonte di vulnerabilità risiede nel Controller Area Network (CAN), dove la trasmissione broadcast dei messaggi è a rischio a causa della topologia del bus della rete. In tale contesto, i messaggi vengono scambiati tra le centraline senza l'implementazione di autenticazione o crittografia, rendendo il sistema vulnerabile ad attacchi. Vi sono, quindi, tre aspetti fondamentali:
    a. CAN → si riferisce al protocollo di comunicazione, che gestisce il flusso di dati tra i vari componenti elettronici di un veicolo (come per esempio: centraline e sensori) e rappresenta una possibile vulnerabilità del sistema;
    b. Trasmissione broadcast dei messaggi → i messaggi inviati attraverso il CAN sono trasmessi a tutti i componenti della rete senza una destinazione specifica;

c. Rischio a causa della topologia del bus della rete → la struttura o la configurazione della rete CAN (la topologia del bus) può essere sfruttata per mettere a rischio la sicurezza del sistema. Questo potrebbe avvenire, ad esempio, perché i messaggi vengono scambiati senza adeguata autenticazione o crittografia, rendendo più facile per eventuali attaccanti manipolare o intercettare i dati trasmessi.

Le due vulnerabilità appena sopra citate, mettono in evidenza il fatto, che le applicazioni di Infotainment di bordo possono contenere bug (come qualsiasi altro software). Mentre l'architettura delle automobili (in particolar modo il CAN) inizialmente era stata sviluppata per un ambiente chiuso con limitati rischi di sicurezza, è ora esposta ad attacchi esterni (talvolta remoti). Capiamo, allora, che la sicurezza è un problema significativo, poiché i produttori di automobili permettono l'integrazione di dispositivi di intrattenimento e di utilità generale nei veicoli [4].

Infotainment e OBP apps

Per rimanere al passo con lo sviluppo dell'elettronica di consumo, le case automobilistiche hanno introdotto applicazioni di Infotainment, utilizzando maggiormente lo standard aperto Android Auto, ovverosia una piattaforma di Infotainment sviluppata da Google, progettata per consentire l'integrazione di dispositivi Android con il sistema di Infotainment dell'auto. Utilizzare uno standard aperto come Android Auto, ha come conseguenza diretta, il fatto che le case automobilistiche adottino un approccio, che consente una maggiore flessibilità e interazione con dispositivi Android di terze parti. Nonostante Google proponga un elenco completo di misure di sicurezza e di standard per le applicazioni, questi non sempre vengono rispettati e conseguentemente, gli attaccanti possono sfruttare tali mancanze/vulnerabilità. In particolar modo, Android Auto introduce le seguenti minacce:

Tecnica d'attacco	Descrizione
Virus/Malware	Utilizzo non autorizzato delle funzioni di infotainment attraverso l'impersonificazione o attacchi ai bug.
Autenticazione	Le informazioni inerenti al veicolo vengono rubate o mascherate per un utilizzo illegale.
Impostazione illegale	I dati del veicolo vengono compromessi attraverso impersonificazione o attacchi ai bug.
Informazioni false	Le app dannose inviano messaggi falsi al sistema di infotainment, al fine di ingannare il conducente o eseguire delle azioni illegali.
Jamming	Le app dannose ottengono il controllo del percorso di comunicazione, dirottano le comunicazioni regolari e si mescolano con quelle illegali.
Tracciamento	Gli attaccanti ottengono illegalmente informazioni sul veicolo e tracciano le informazioni sullo stato del veicolo come velocità, posizione e destinazione.
Distrazioni del conducente	Le app dannose distraggono il conducente visualizzando immagini o riproducendo audio o video.

La sicurezza delle reti di bordo è un problema che ha implicazioni per la sicurezza dei veicoli moderni. I veicoli comunicano con vari sensori e attuatori (gli attuatori sono componenti del sistema che convertono un segnale di controllo in un'azione fisica. Nel contesto dei veicoli, gli attuatori possono essere dispositivi che eseguono azioni meccaniche, elettriche o idrauliche in risposta ai segnali ricevuti dal sistema di controllo) attraverso un bus elettronico, al quale vengono collegati dispositivi esterni tramite la porta OBD-II e comunicano con l'auto tramite comandi AT. Questi comandi AT sono spesso utilizzati nei protocolli di comunicazione seriale, come ad esempio l'Interfaccia Standardizzata a Bordo (OBD-II), che è un sistema utilizzato per diagnosticare elettronicamente i problemi dei veicoli. I comandi AT possono includere istruzioni specifiche, inviate attraverso la porta OBD-II, per richiedere informazioni o per eseguire operazioni specifiche all'interno del sistema elettronico del veicolo. Purtroppo, si è dimostrato che il CAN, sul quale si basa il sistema, non offre protezione dalla manipolazione e questo ha come conseguenze dirette che:

- La manipolazione del CAN può consentire a terzi non autorizzati di influenzare il normale funzionamento del veicolo. Ad esempio, un attaccante potrebbe compromettere la sicurezza del veicolo manipolando i comandi inviati attraverso la porta OBD-II;
- La manipolazione del sistema potrebbe consentire a terzi di ottenere informazioni sensibili dal veicolo o di monitorare le attività dell'utente senza il suo consenso;
- La manipolazione del CAN potrebbe portare all'inserimento di dati falsi o alla modifica delle informazioni trasmesse attraverso il bus elettronico. Questo potrebbe influenzare negativamente la corretta diagnosi dei problemi del veicolo durante la manutenzione e la riparazione;
- Se un attaccante fosse in grado di manipolare il sistema di controllo del veicolo, potrebbe compromettere la sicurezza stradale influenzando il comportamento del veicolo.

In passato, la porta OBD-II è stata utilizzata per scaricare dati diagnostici e per eseguire test, ma attualmente ci sono pressioni per consentire ai conducenti di accedere ai dati attraverso il proprio smartphone e persino tramite Internet, in modo tale da consentire un'interazione diretta con il sistema di controllo del veicolo e questo, ovviamente, crea rischi di sicurezza. Elenchiamo brevemente i rischi associati all'utilizzo delle app OBD-II [5] [8]:

Tecnica d'attacco	Descrizione
CAN Injection	Gli attaccanti inviano messaggi CAN modificati per eseguire operazioni pericolose.
Compromissione del dispositivo OBD-II	Gli attaccanti utilizzano un dispositivo OBD non sicuro per assumere il controllo di componenti cruciali dell'auto.
Bluebugging	Gli attaccanti prendono il controllo del dispositivo OBD quando non è collegato al dispositivo del proprietario e accedono a componenti cruciali dell'auto.
Violazione della privacy	Gli attaccanti utilizzano i dispositivi OBD per attaccare i dispositivi mobile dell'auto e rubare i dati personali.
Tracciamento	Gli attaccanti ottengono illegalmente le informazioni del veicolo e ne tracciano lo stato.

Vulnerabilità

L'ecosistema Android è una complessa rete aperta di aziende, che collaborano tra di loro (basti pensare al fatto, che utilizza più di 170 progetti open-source). Inoltre, i produttori di hardware e i fornitori di rete, in base alle proprie esigenze, personalizzano Android e tali personalizzazioni, da un lato possono essere necessarie per adeguare Android ai dispositivi specifici e migliorare l'esperienza utente, ma dall'altro lato possono creare un ambiente, in cui il sistema è maggiormente suscettibile a vulnerabilità. Visto che i diversi attori apportano modifiche indipendenti al sistema, possiamo immediatamente capire che questo rende più difficile garantire un controllo uniforme sulla sicurezza complessiva del sistema operativo, in quanto ad esempio vi sono:

- Differenze di competenze → gli attori potrebbero avere livelli diversi di competenza nella gestione della sicurezza. Alcuni potrebbero essere più consapevoli delle best practice di sicurezza, mentre altri potrebbero commettere errori che mettono a rischio la sicurezza complessiva;
- Divergenze di obiettivi → gli attori potrebbero avere obiettivi diversi, e le modifiche che apportano potrebbero essere orientate verso obiettivi specifici o funzionalità che potrebbero non essere allineati con le priorità della sicurezza complessiva del sistema;
- Complessità delle modifiche → le modifiche indipendenti possono aumentare la complessità del sistema operativo. La gestione della sicurezza diventa più difficile quando ci sono numerose modifiche da monitorare e gestire, specialmente se non c'è una visione unificata delle conseguenze sulla sicurezza di tali modifiche.

Riassumendo questi concetti, quindi, possiamo affermare che la varietà e le diversità introdotte dalla personalizzazione, possono creare opportunità per potenziali vulnerabilità, poiché il sistema non è completamente omogeneo e standardizzato in termini di configurazione e implementazione di sicurezza [5].

Vulnerabilità Android Auto apps

Google definisce una serie di requisiti di qualità per le applicazioni Android Auto, la cui violazione comporta molteplici vulnerabilità e in questa sezione vorremmo elencarne alcune:

1. External File Access Detection → i file creati nella memoria esterna, come le schede SD, sono globalmente leggibili e scrivibili. Pertanto, i dati dell'app non devono contenere informazioni sensibili utilizzando l'archiviazione esterna, che possono essere rimosse dall'utente e modificate da qualsiasi app dannosa. Inoltre, le app che utilizzano l'archiviazione esterna devono eseguire la convalida dell'input quando gestiscono i dati da una memoria esterna, in quanto potrebbero contenere eseguibili o dati provenienti da fonti non attendibili, che causano danni all'auto. Capiamo, allora, che l'External File Access Detection comporta diverse problematiche:
    - Accesso non autorizzato → la possibilità di lettura e scrittura globale sui file nella memoria esterna potrebbe consentire ad app dannose o non autorizzate di accedere a informazioni sensibili contenute in quei file, tra cui: dati personali dell'utente, configurazioni sensibili dell'app o altre informazioni riservate;
    - Modifica non autorizzata → la capacità di scrivere su file nella memoria esterna apre la porta a modifiche non autorizzate. App dannose potrebbero alterare i dati dell'applicazione, inserire informazioni malevole o danneggiare l'integrità complessiva delle informazioni memorizzate;

- Rischio per la sicurezza dell'auto → nel contesto di Android Auto, dove le applicazioni possono interagire direttamente con il sistema IVI dell'auto, un accesso non autorizzato o modifiche non autorizzate ai file potrebbero influire sulla sicurezza del veicolo. Ad esempio, l'esecuzione di eseguibili provenienti da fonti non attendibili potrebbe causare danni all'auto o comportare rischi per la sicurezza durante la guida;
- Validazione dell'input → la mancanza di convalida dell'input quando si gestiscono dati provenienti dalla memoria esterna aumenta il rischio di inserimento di dati dannosi o eseguibili malevoli. La convalida dell'input, quindi, è essenziale per prevenire l'esecuzione di codice dannoso o l'inserimento di dati malevoli nel sistema.

Vediamo, nella Figura 2, un estratto di codice per osservare un utilizzo non attendibile di una directory esterna, andando a chiamare il metodo '*Context.getExternalCacheDir()*':

```
public static File getDiskCacheDir(Context c) {
   File dir = c.getExternalCacheDir();
   if (dir == null)
      dir = c.getCacheDir();
   return dir;
}
```

Figura 2: Uso non attendibile di una directory esterna

Questo codice definisce un metodo statico chiamato '*getDiskCacheDir'*, che accetta un oggetto di tipo *Context* come parametro e restituisce un oggetto di tipo *File*. Il metodo cerca di ottenere il percorso della directory esterna di cache del dispositivo utilizzando il metodo '*getExternalCacheDir*()'. La cache esterna è una posizione di memorizzazione temporanea generalmente associata all'archiviazione esterna del dispositivo (come una scheda SD). Se il risultato di "getExternalCacheDir()" è nullo (ad esempio, se lo spazio esterno non è montato), allora viene chiamato il metodo '*getCacheDir*()', che restituisce il percorso della directory di cache interno del dispositivo.

Analizzando questa prima vulnerabilità, ci siamo immediatamente chiesti quali potrebbero essere delle misure di protezione, per mitigare tale vulnerabilità. Alcune possibili soluzioni, che abbiamo pensato, sono:

- Limitare l'uso dell'archiviazione esterna → ridurre al minimo l'utilizzo dell'archiviazione esterna per i dati sensibili o critici;
- Imporre restrizioni d'accesso → applicare rigorose politiche di controllo degli accessi, al fine di garantire che solamente le applicazioni autorizzate possano accedere e modificare i file nella memoria esterna. Inoltre, utilizzare un meccanismo di autorizzazione, in modo tale da controllare l'accesso ai file;
- Crittografare i dati sensibili → nel caso sia inevitabile memorizzare alcuni dati sensibili e/o critici all'interno di memorie esterne, essi potrebbero essere crittografati, in modo tale da proteggerli da accessi non autorizzati, dato che tali dati devono solamente essere decodificati e non eseguiti. Di conseguenza, non abbiamo il problema di rimetterli in chiaro;
- Monitorare l'accesso ai file → implementare meccanismi di monitoraggio, al fine di tracciare l'accesso e le modifiche ai file nella memoria esterna, in modo tale da identificare comportamenti sospetti o tentativi di accesso non autorizzato;
- Educare gli sviluppatori → informare gli sviluppatori sull'importanza di gestire correttamente l'archiviazione esterna, in quanto degli utenti non consapevoli possono rappresentare una superficie di attacco.

2. Utilizzo di WORLD_WRITEABLE → di default, Android fa sì che solo l'applicazione che ha creato un file sulla memoria interna possa accedervi. Tuttavia, alcune app utilizzano MODE_WORLD_WRITEABLE o MODE_WORLD_READABLE per i file, aggirando così questa restrizione e sfruttano anche la possibilità di modificare e controllare il formato dei dati. In questo modo, le app dannose possono manomettere e/o rubare informazioni private dal sistema di Infotainment dell'auto o dallo smartphone. Nella Figura 3, analizziamo una porzione di codice per capire meglio tale vulnerabilità:

```
File f = new File(getFilesDir(), "filename.ext");
f.delete();
FileOutputStream fos = openFileOutput("filename.ext", Context.MODE_WORLD_WRITEABLE);
fos.close();
File f = new File(getFilesDir(), "filename.ext");
```

Figura 3: Uso vulnerabile di WORLD_WRITEABLE

Possiamo vedere, innanzitutto, che viene creato un oggetto *File*, che rappresenta il percorso del file '*filename.ext*' nella directory privata dell'applicazione, ottenuta tramite il metodo *getFilesDir()*.Nel caso in cui il file esista già, esso viene eliminato, altrimenti crea un *FileOutputStream* con l'opzione '*Context.MODE_WORLD_WRITEABLE'* e successivamente viene chiuso il *FileOutputStream*. Abbiamo, quindi:

- MODE_PRIVATE → è la modalità predefinita e significa che il file creato sarà accessibile solo all'applicazione chiamante;
- MODE_WORLD_READABLE → le altre applicazioni possono leggere il file creato, ma non possono modificarlo;
- MODE_WORLD_WRITEABLE → le altre applicazioni hanno i permessi di scrittura per il file creato.

Anche per questa vulnerabilità, ci siamo chiesti come potessimo mitigarla e le soluzioni che vorremmo proporre sono le seguenti:

- Evitare l'utilizzo di *MODE_WORLD_WRITEABLE* e *MODE_WORLD_READABLE* → evitare completamente l'uso di queste modalità e conseguentemente preferire modalità più restrittive, come *MODE_PRIVATE*, che consente l'accesso solamente all'applicazione stessa;
- Limitare i permessi d'accesso → nel caso in cui fosse necessario condividere dati con altre applicazioni, considerare l'utilizzo di meccanismi che consentano di definire in modo granulare i permessi di accesso;
- Controlli di sicurezza durante la lettura e scrittura dei dati.

3. Background downloads → per scaricare un file, se la posizione di memorizzazione non è stata impostata esplicitamente, il programmatore utilizza il metodo *DownloadManager.openDownloadedFile()*, con il valore dell'ID memorizzato nelle preferenze, per ottenere un *ParcelFileDescriptor* che può essere trasformato in un input stream. Inoltre, senza una destinazione specifica, i file scaricati rimangono nella cache dei download condivisi. In questo caso, il sistema si riserva il diritto di eliminarli in qualsiasi momento per recuperare spazio. Ciò lascia l'applicazione in uno stato di vulnerabilità, poiché i dati condivisi possono essere liberamente accessibili. Per capire meglio ciò, guardiamo il frammento di codice nella Figura 4:

```
Request.setDestinationInExternalFilesDir(): Set the destination to a hidden directory on external storage
Request.setDestinationInExternalPublicDir(): Set the destination to a public directory on external storage
Request.setDestinationUri(): Set the destination to a file Uri located on external storage
```

Figura 4: Download in background non affidabile

Osserviamo il comportamento dei tre metodi:

- Request.setDestinationInExternalFilesDir → imposta la destinazione del download in una directory nascosta su uno storage esterno. La directory nascosta è specifica dell'applicazione e non è facilmente accessibile da altre app o dall'utente. Questa scelta, quindi, potrebbe essere fatta per mantenere i file scaricati privati e limitarne l'accesso;
- Request.setDestinationInExternalPublicDir() → imposta la destinazione del download in una directory pubblica su uno storage esterno. Una directory pubblica è accessibile da altre app e dall'utente e di conseguenza, l'utilizzo di una directory pubblica potrebbe essere appropriata se si desidera consentire l'accesso a questi file da parte di altre app o per rendere i file visibili all'utente;
- Request.setDestinationUri() → imposta la destinazione del download utilizzando un URI di file situato su uno storage esterno. Anche in questo caso, il file potrebbe essere accessibile da altre app o dall'utente, a seconda del percorso specificato nel file URI.

Anche in questo caso, abbiamo pensato a delle possibili contromisure, che sono:

- Esplicitare la posizione di memorizzazione → i programmatori dovrebbero essere incoraggiati a impostare esplicitamente la posizione di memorizzazione per i file scaricati. Ciò può essere fatto utilizzando metodi come '*Request.setDestinationInExternalFilesDir*', '*Request.setDestinationInExternalPublicDir()*' o '*Request.setDestinationUri()'*. In questo modo, si evita la vulnerabilità associata alla mancata impostazione esplicita della destinazione;
- Evitare directory pubbliche per i dati sensibili → le directory pubbliche sono accessibili da altre app e dall'utente;
- Monitorare e gestire la cache dei download → implementare un meccanismo per monitorare e gestire la cache dei download. Se i file rimanessero nella cache senza una destinazione specifica, l'applicazione dovrebbe essere proattiva nel gestire questi file o spostarli in una posizione sicura [5].

Vulnerabilità OBD-II apps

In questa sezione analizziamo le vulnerabilità di sicurezza, di alcune delle più comuni delle applicazioni OBD-II:

1. Privacy Breach → le app OBD-II possono raccogliere informazioni sensibili da diversi sensori e attuatori, come ad esempio: posizione del veicolo, la velocità e lo stato del parabrezza. Le app dannose possono inviare tali informazioni ad un Server remoto, utilizzando servizi web, dove i dati raccolti vengono trasmessi o memorizzati in modo non sicuro. Chiunque, quindi, abbia accesso ai registri, può ricostruire gli spostamenti del veicolo e questo, chiaramente, rappresenta un grave problema di privacy. Come abbiamo fatto per le vulnerabilità presenti nella applicazioni Android, anche in questo caso abbiamo pensato a delle possibili contromisure. In particolare, per tale vulnerabilità di Privacy Breach abbiamo optato per le seguenti contromisure:

- Crittografia dei dati → implementare una robusta crittografia per i dati sensibili raccolti dalle app OBD-II. Questo assicura che, nel caso in cui i dati vengano intercettati, non saranno facilmente decifrabili senza la chiave appropriata;
- Accesso autorizzato → limitare l'accesso ai dati sensibili solo a utenti autorizzati. Utilizzare un'autenticazione forte e autorizzazioni granulari per garantire, che solo le persone o i sistemi autorizzati possano accedere alle informazioni sensibili;
- Politiche di conservazione dei dati → stabilire politiche chiare sulla conservazione dei dati e garantire che i dati vengano cancellati quando non sono più necessari per le finalità previste;
- Firewall → implementare misure di sicurezza, come firewall e filtri di rete, per controllare e monitorare il traffico in entrata e in uscita. Questo può contribuire a prevenire l'invio non autorizzato di dati sensibili a server remoti.

2. Iniezioni di dati nel CAN → le applicazioni possono fornire accesso diretto al CAN di un veicolo con un'autenticazione minima o nulla. Inoltre, è possibile utilizzare questo canale insicuro per iniettare messaggi CAN modificati per ottenere vari livelli di controllo fisico sul veicolo (per esempio, diverse centraline potrebbero reagire a tali messaggi, sbloccando l'auto). Le possibili contromisure, che abbiamo pensato, sono:
   - Filtraggio dei messaggi CAN → utilizzare filtri per controllare i messaggi CAN in ingresso ed escludere quelli che potrebbero essere potenzialmente dannosi o non autorizzati;
   - Crittografia dei messaggi → crittografare i messaggi CAN per garantire che siano sicuri durante la trasmissione e che non possano essere facilmente manipolati o intercettati da terze parti non autorizzate;
   - Test di sicurezza → sottoporre le applicazioni e i sistemi al testing di sicurezza, incluso il testing di penetrazione, per identificare e correggere eventuali vulnerabilità;
   - Isolare le applicazioni dal sistema CAN.
3. Dati CAN memorizzati in un Database → le applicazioni possono memorizzare diversi dati dei sensori delle centraline nei Database per vari scopi di analisi. Questo può portare ad attacchi di SQL-injection e compromettere l'affidabilità delle applicazioni, se non adeguatamente sanificate. Pertanto, è necessario assicurarsi che i dati che passano dal CAN al Database non siano contaminati. Le possibili contromisure, che abbiamo pensato, sono:
   - Validazione dei dati di input → implementare controlli di validazione rigorosi per assicurarsi che i dati provenienti dal CAN siano conformi a un formato specifico e non contengano caratteri dannosi;
   - Evitare la costruzione di query SQL concatenando direttamente i dati dell'utente con le query;
   - Privilegi minimi → assegnare privilegi al Database in modo appropriato, garantendo che le applicazioni abbiano solo le autorizzazioni necessarie per eseguire operazioni specifiche. Evitare di concedere privilegi eccessivi, in modo da limitare la portata di eventuali attacchi;
   - Escape dei dati → quando è necessario incorporare dati dinamici nelle query SQL, assicurarsi di utilizzare correttamente le funzioni di escape fornite dalla piattaforma del database per neutralizzare eventuali caratteri speciali che potrebbero essere utilizzati in un attacco di SQL injection [5].

Esempio di attacco (MirrorLink)

Le automobili moderne sono sempre più controllate da sistemi computerizzati, che sono collegati direttamente o indirettamente a Internet. Questo significa, a sua volta, che le automobili sono potenzialmente vulnerabili ad attacchi da parte di attaccanti, i quali possono ottenere il controllo di uno di questi dispositivi esterni. Come abbiamo citato nelle sezioni precedenti, le automobili sono vulnerabili ad un attaccante, che ha accesso interno al Controller Area Network del veicolo o che può controllare da remoto uno di questi dispositivi. Nonostante questi rischi, il numero di dispositivi esterni che vengono connessi sia ad un sistema automobilistico, sia a Internet continua ad aumentare. In particolare, si registra una tendenza all'integrazione di smartphone e sistemi di In-Vehicle-Infotainment (IVI) collegati al CAN e tale integrazione (tra IVI e smartphone) è tipicamente facilitata da una coppia di applicazioni:

1. Una che viene eseguita sullo smartphone;
2. L'altra eseguita sull'IVI.

Attualmente queste piattaforme di app sono un ecosistema chiuso di sviluppatori di terze parti fidati, ma molte case automobilistiche hanno in programma di aprire le loro piattaforme di applicazioni IVI a un gruppo più ampio di sviluppatori meno controllati. Recentemente sono emersi diversi protocolli standardizzati per consentire una connettività fluida e senza interruzioni (nella trasmissione dei dati) tra gli smartphone e i sistemi di Infotainment delle auto, come: Android Auto, Apple CarPlay e MirrorLink. Sorge a questo punto spontanea la domanda: “In che misura queste applicazioni, protocolli e implementazioni IVI sottostanti siano vulnerabili ad un attaccante, il quale può ottenere il controllo dello smartphone del conducente?” Per rispondere a questa domanda, dobbiamo diramarci in tre livelli, ovvero:

1. Analisi di sicurezza del protocollo MirrorLink e delle modalità di implementazione;
2. Analisi di sicurezza delle parti critiche dell’applicazione IVI MirrorLink;
3. Dimostrare le vulnerabilità dell’implementazione dell’applicazione IVI MirrorLink [6].

Analisi protocollo MirrorLink

Lo standard MirrorLink1 specifica un protocollo, che facilita l'integrazione di uno smartphone in un sistema di Infotainment per autoveicoli. L'obiettivo principale è quello di offrire all'utente la possibilità di utilizzare le applicazioni del proprio smartphone (ad esempio, le applicazioni di navigazione) sullo schermo più grande del sistema di Infotainment, invece di acquistare un IVI più costoso con connettività Internet e applicazioni integrate. Inoltre, aggiornare un sistema di Infotainment non è così semplice come uno smartphone. Questa relazione tra smartphone e IVI si basa su un modello Client-Server, in cui lo smartphone funge da Server e il sistema di Infotainment è il suo Client. In questo modello, tutte le applicazioni verranno eseguite in modo nativo sullo smartphone e lo schermo dello smartphone viene replicato sullo schermo dell'Infotainment. Quindi, il Server MirrorLink (ovvero lo smartphone) esegue in modo nativo le applicazioni richieste e rispecchia la visualizzazione dell'applicazione sullo schermo dell'Infotainment.

Le specifiche MirrorLink definiscono i livelli di connettività del protocollo, che possiamo osservare nella Figura 5:

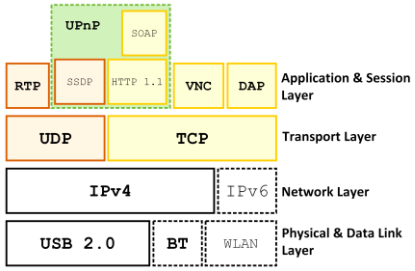


Figura 5: Livelli di connettività del protocollo MirrorLink

Possiamo vedere, quindi, che abbiamo quattro livelli di connettività, ovvero:

1. Livello fisico e di collegamento dati → MirrorLink richiede almeno USB 2.0, in quanto essa soddisfa il requisito minimo di larghezza di banda e allo stesso tempo fornisce una soluzione di ricarica per il dispositivo mobile (invece il supporto WLAN e Bluetooth è opzionale);
2. Livello di rete e di trasporto → IPv4 è specificato per il livello di rete con UDP e TCP come protocolli di trasporto. Sia il Client (IVI) che il Server (smartphone) devono supportare DHCP per l'indirizzamento IPv4;
3. Livello di sessione e applicazione → il livello applicativo di MirrorLink comprende quattro componenti di base del livello di sessione e utilizza socket TCP o UDP per interagire con questi componenti. Tali componenti di base sono:
    - UPnP → è utilizzato per la negoziazione di servizi, la pubblicità dei dispositivi Server abilitati a MirrorLink e profili Client abilitati, nonché il controllo delle applicazioni basate su Server MirrorLink. Utilizza UDP per il broadcasting e la pubblicità e TCP per il controllo delle applicazioni remote;
    - VNC → fornisce la funzione di mirroring per il protocollo MirrorLink. Replica il contenuto del display del Server MirrorLink (smartphone) al Client MirrorLink (IVI). Trasferisce le informazioni di controllo, come eventi dei tasti, del puntatore o del tocco, nonché i comandi vocali dal client al server. Il protocollo VNC utilizza socket TCP;
    - RTP → permette di effettuare lo streaming dell'audio per diversi tipi di payload ed è fornito dal protocollo RTP;
    - DAP → protocollo di attestazione del dispositivo, ovverosia è il responsabile della verifica dell'hardware, su cui opera il software del Server MirrorLink. Si riferisce, quindi, al Client MirrorLink (IVI), il quale verifica che il Server MirrorLink (smartphone) sia di un produttore conforme e che esegue il software approvato [6].

Data extraction

Iniziamo descrivendo i metodi, che sono stati utilizzati per riuscire ad ottenere l'accesso, e conseguentemente riuscire ad estrarre i dati memorizzati sui chip di memoria, attraverso il protocollo MirrorLink. Innanzitutto, è stato identificato un chip flash NOR nel IVI. La memoria flash è un tipo di memoria a stato solido e non volatile, che per le sue prestazioni può anche essere usata come memoria a lettura-scrittura. In particolare, la memoria flash NOR è uno dei due tipi di tecnologie di archiviazione non volatile (mentre l'altra è la NAND, che ha una capacità di memoria superiore). Le memorie NOR sono più veloci da leggere rispetto alle memorie NAND, ma sono anche

più costose e richiedono più tempo per cancellare e scrivere nuovi dati. Il primo passo, quindi, è stato costruire un lettore di memorie flash, al fine di leggere la NOR flash. Il chip NOR flash conteneva il BIOS, un boot-loader e due certificati root:

1. Uno dell'organizzazione MirrorLink;
2. L'altro del produttore automobilistico.

Per installare un'immagine firmware non ufficiale (ricordandoci, che l'immagine del firmware si riferisce a un file che contiene il software di sistema operativo incorporato (firmware) destinato a eseguire su dispositivi hardware specifici, come router, stampanti, telecamere di sorveglianza, dispositivi di rete e altri componenti embedded), questi certificati devono essere sostituiti con quelli utilizzati per firmare l'immagine. Per questo motivo, non sarebbe possibile per un attaccante remoto tentare di ingannare un conducente ad installare un aggiornamento dell'immagine potenzialmente dannoso, a meno che non ottenga l'accesso alle chiavi di firma private del produttore automobilistico e generare un aggiornamento dell'immagine validamente firmato. Quindi, è stata decompressa l'immagine di aggiornamento e si è ottenuto:

4. L'immagine di runtime di Windows embedded CE;
5. Gli eseguibili dell'applicazione utente e del kernel;
6. I file di configurazione;
7. L'immagine completa della memoria flash NOR.

Per studiare il comportamento del protocollo, successivamente è stato monitorato e catturato il traffico USB tra lo smartphone e l'IVI. Per catturare il traffico USB, è stato costruito lo sniffer USB, utilizzando la BeagleBoard-xM (ovverosia una versione avanzata di BeagleBoard, una famiglia di computer a scheda singola utilizzate per la prototipazione e lo sviluppo di software e progetti embedded) e una variante del modulo di cattura pacchetti USB open source prodotto da BeagleBoard, un modulo di cattura dei pacchetti USB open source prodotto da un progetto Google Summer of Code [6].

Osservazioni

Sulla base dell'analisi del traffico USB tra lo smartphone e l'IVI, possiamo elaborare delle osservazioni sul protocollo MirrorLink:

1. Prevede una prima fase di configurazione del Linklayer: consiste nella negoziazione e nell'accettazione della classe USB, nonché la corretta configurazione delle interfacce USB dell'IVI. Lo smartphone funge da server DHCP, assegnando un indirizzo IP al IVI;
2. Dopo la configurazione, lo smartphone avvia la sessione UPnP, pubblicizza i suoi servizi e fornisce un URL dal quale l'IVI scarica file di configurazione XML (inclusi tmclient.xml e tmapplicationserver.xml, che contengono rispettivamente informazioni di configurazione specifiche per il Client e per il Server). Dall'analisi dei pacchetti catturati, non sono emerse indicazioni di scambio di chiavi, crittografia o autenticazione nei messaggi XML. Nessuno di questi XML era firmato, il che potrebbe rendere vulnerabile il sistema IVI a un attacco da parte di un dispositivo con accesso al Linklayer;
3. Successivamente abbiamo la comunicazione tra IVI e Smartphone: l'IVI utilizza il protocollo SOAP su HTTP per comunicare i servizi di controllo e di eventi allo smartphone. Questa comunicazione include la conferma della configurazione hardware e software, il recupero di un elenco di applicazioni disponibili e la gestione dello stato dell'applicazione;

4. Dopo l'avvio dell'applicazione MirrorLink, l'IVI agisce come Client di visualizzazione VNC che si connette al Server VNC dello smartphone. La negoziazione del protocollo mostra un potenziale rischio, poiché il Client seleziona un tipo di sicurezza "none", il che potrebbe consentire un dirottamento della sessione VNC da parte di un utente malintenzionato. Inoltre, non sono stati rilevati messaggi di richiesta o risposta di attestazione del contenuto durante il traffico catturato [6].

Software analisi

Per esaminare la presenza di protezioni di sicurezza standard e per individuare potenziali vulnerabilità sfruttabili dall'attaccante, sono state eseguite delle analisi statiche e dinamiche al protocollo MirrorLink. In particolare, l'analisi statica ha evidenziato le seguenti vulnerabilità del protocollo:

8. Funzioni di invio messaggi CAN → attraverso l'analisi, sono state identificate due funzioni, *Send2Micom <redacted>Msg* e *SendMsg <redacted>Msg*, che consentono l'invio di byte di dati su un CAN tramite il controllore Micom CAN del IVI;
9. Controllore Micom CAN → gestisce un elenco di ID CAN, che limita le comunicazioni dell'IVI ad altri dispositivi con lo stesso CAN. L'analisi suggerisce che un attaccante potrebbe utilizzare dei metodi (noti) per aggiornare il firmware Micom, ottenendo così la possibilità di inviare messaggi arbitrari sul CAN, se riuscisse a ottenere il controllo di un processo sull'IVI;
10. Vulnerabilità nell'aggiornamento del firmware Micom → il firmware del controllore Micom CAN può essere aggiornato attraverso due metodi: uno tramite l'esecuzione di un file eseguibile specifico (MgrUpg.exe) durante la procedura di aggiornamento principale e l'altro attraverso l'interfaccia DevMode con un bypass dell'autenticazione;
11. Mancanza di controlli di autenticazione nel firmware Micom → l'analisi non ha rilevato prove di funzioni di autenticazione o verifica nel firmware Micom. Tuttavia, l'aggiornamento ha causato il blocco dell'IVI con il messaggio "NO VIN," probabilmente legato a un sistema antifurto;
12. Vulnerabilità nel codice sorgente → l'analisi statica ha rivelato che il firmware e le DLL IVI sono scritte in C++ e ci sono diverse chiamate a funzioni libc pericolose, senza un adeguato controllo dei confini. Ad esempio, sono presenti chiamate memcpy, con il rischio di copiare input potenzialmente dannosi senza controllo dei limiti.

L'analisi dinamica, invece, riporta i seguenti risultati: Grazie ai risultati dell'analisi statica, si è scoperto che non esiste un controllo dei limiti o un altro meccanismo di protezione. Per questo motivo, è stata costruita un'applicazione che invia all'IVI degli XML modificati, per capire se questo avrebbe generato errori di eccezione. Per impostare l'analisi dinamica, è stato fatto il Reverse Engineering del Server dell'app MirrorLink (smartphone) del protocollo MirrorLink. Successivamente, è stato re-implementato una parte dell'app, che abilita l'Ethernet su USB. Con questo sistema, si è iniziato a imitare il resto del protocollo MirrorLink, replicando e inviando i messaggi UPnP corretti. È stato creato un description.xml sostitutivo e tmapplicationserver.xml, modificando alcuni dei valori degli elementi (ad esempio, appname, appid, bluetoothaddress) in modo da far traboccare i buffer struct statici, in cui sarebbero stati copiati i valori degli elementi durante l'analisi. Attraverso tali XML appositamente creati, si è potuto assistere a diversi errori di eccezione runtime. In particolare, gli XML (appositamente realizzati) causavano corruzioni della memoria Heap e a loro volta, tali corruzioni della memoria, sovrascrivevano strutture di dati e puntatori a funzioni che risiedevano sull'Heap. Per dare seguito alla scoperta di questi Heap overflow e capire meglio se potessero essere utilizzati per ottenere l'esecuzione di controllo dell'IVI,

è stata sviluppata un'applicazione dannosa per smartphone. Tale applicazione, prevedeva che la lunghezza del valore dell'elemento XML <appName> fosse aumentato, al fine di far traboccare l'Heap. Questa applicazione configura correttamente la modalità USB e stabilisce la connettività di rete con l'IVI, poi imita correttamente il protocollo DHCP e i messaggi iniziali dell'applicazione DriveLink tra lo smartphone e l'IVI. Una volta che l'IVI ha recuperato il file description.xml, l'applicazione invia una versione dannosa, accuratamente creata, che include diversi valori di grandi dimensioni negli elementi dello schema XML. Questi valori di grandi dimensioni causano una serie di Heap overflow in TMScontrolPoint.dll che organizza correttamente l'Heap e sovrascrive un puntatore a una funzione sull'Heap.

Infine, sono stati creati una serie di gadget sull'Heap, che permettono di ottenere il flusso di controllo dell'esecuzione e di riservare spazio sull'Heap, che può essere utilizzato per iniettare codice maligno. Poiché Windows Embedded CE separa l'Heap meta-dati dall'Heap vero e proprio, dobbiamo sovrascrivere con attenzione una serie di puntatori ai dati tra l'inizio dell'Heap e il puntatore alla funzione di destinazione. Se questi puntatori di dati non vengono sovrascritti con indirizzi validi, il processo si arresterà prima di chiamare il puntatore alla funzione sovrascritta sull'Heap [6].

Modellazione delle minacce

La modellazione delle minacce è un metodo per identificare e priorizzare i pericoli legati al sistema, al fine di sviluppare delle contromisure efficaci contro le minacce. In parole povere, mira a rispondere a domande come: "Dove il sistema potrebbe essere vulnerabile alle minacce?", "Quali minacce sono più significative?" e "Dove sono le debolezze del sistema?". Questo modello cerca di individuare le vulnerabilità del sistema e le debolezze e di conseguenza, un modello di minaccia include la dimensione offensiva e difensiva di un'entità logica, un dato, un host, un'applicazione, un sistema o un ambiente. Nonostante l'esistenza di vari modelli di modellazione delle minacce (come ad esempio: Attack Tree o PASTA), il documento (preso in esame) utilizza STRIDE, in quanto è ampiamente accettato nel mondo accademico e industriale. Per migliorare la sicurezza del sistema di Infotainment, quindi, si ha effettuato una modellazione delle minacce utilizzando STRIDE di Microsoft e si ha valutato i rischi utilizzando SAHARA e DREAD. Microsoft STRIDE identifica le minacce alla cybersecurity utilizzando sei categorie:

1. Spoofing;
2. Tampering;
3. Repudiation;
4. Information Disclosure;
5. Denial of Service;
6. Elevation of Privilege.

Ogni componente di un sistema di Infotainment può essere analizzato con il metodo STRIDE e può essere soggetto a una o più minacce di ciascuna categoria e ciò lo possiamo vedere dalla Tabella 6:

STRIDE Category	External Entity	Process	Data Flow	Data Store
Spoofing	✓	✓		
Tampering		✓	✓	✓
Repudiation	✓	✓		
Information Disclosure		✓	✓	✓
Denial of Service		✓	✓	✓
Elevation of Privilege		✓		

Tabella 6: Minacce associate ai componenti del sistema di Infotainment

Lo strumento STRIDE avvia il processo di modellazione delle minacce presentando una DFD, ovverosia un diagramma di flusso dei dati, il quale fornisce una rappresentazione completa di tutti i componenti del sistema e dei relativi flussi di dati. Notiamo, allora, che il primo passo per effettuare un'analisi attraverso STRIDE, è di identificare e delineare i componenti del sistema, in modo tale da potere creare un DFD. Successivamente, viene generato un rapporto sulle minacce, che comprende informazioni sulle categorie di minacce, le descrizioni delle minacce e le strategie di mitigazione proposte. Per esempio, la Figura 7 illustra l'interazione (con STRIDE) tra l'NFC e il computer di bordo (NFC_to_OBC).

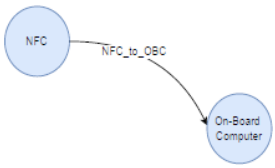


Figura 7: Interazione NFC_to_OBC

Notiamo, che il modello STRIDE identifica tre minacce distinte per tale interazione, che sono:

1. Negazione del servizio;
2. Divulgazione delle informazioni;
3. Manomissione.

Quindi, i dati che passano dall'NFC al computer di bordo, possono diventare il bersaglio di un attaccante in questi tre modi elencati sopra. Capiamo, allora, che la complessa architettura dei veicoli moderni può essere vulnerabile ai cyberattacchi, in quanto l'intero sistema è una combinazione dei rischi associati a ciascun componente interconnesso [4].

SAHARA

La metodologia SAHARA integra l'approccio HARA (Hazard Analysis and Risk Assessment) del settore automobilistico con il modello STRIDE orientato alla sicurezza. Il metodo SAHARA utilizza un elemento fondamentale dell'approccio HARA, in particolare la definizione dei livelli di sicurezza automobilistica (ASIL), per valutare i risultati dell'analisi STRIDE. Le minacce sono valutate considerando le risorse (R) e le competenze (K) necessarie per l'esecuzione della minaccia, insieme alla sua criticità (T). Le minacce alla sicurezza, che hanno il potenziale di compromettere gli obiettivi di sicurezza (T = 3) possono essere trasmesse al processo HARA per un'ulteriore analisi di sicurezza. Questi tre fattori definiscono collettivamente un livello di sicurezza (SecL), come illustrato nelle Tabelle 8 e 9. Questo SecL aiuta a determinare il numero appropriato di contromisure da prendere in considerazione [4].

Level	Knowledge Example	Resources Example	Threat Criticality Example
0	No previous knowledge	No tools required	No impact
1	Basic knowledge of system	Standard tools, screwdriver	Partial service disruption
2	Proficient knowledge of internals with focused interests	Simple tools like sniffer, oscilloscope	Significant damage, manipulation of invoice and privacy
3		Advanced tools like bus communication simulators, flasher	High security impact possible

Tabella 8: Classificazione dei valori K, R e T delle minacce alla sicurezza

R	K	T			
		0	1	2	3
0	0	0	3	4	4
	1	0	2	3	4
	2	0	1	2	3
1	0	0	2	3	4
	1	0	1	2	3
	2	0	0	1	2
2	0	0	1	2	3
	1	0	0	1	2
	2	0	0	0	1
3	0	0	0	1	2
	1	0	0	0	1
	2	0	0	0	1

Tabella 9: Determinazione del SecL

DREAD

DREAD è un metodo di valutazione del rischio, il cui nome corrisponde a cinque criteri di valutazione:

- Danno (D) → indica l'impatto potenziale di un attacco;
13. Riproducibilità (R) → indica la facilità di replicare l'attacco;
14. Sfruttabilità (E) → valutazione dello sforzo necessario per eseguire l'attacco;
15. Utenti copiti (A) → numero di individui che subiranno l'impatto;
16. Scopribilità (D) → misura la facilità di identificazione della minaccia.

Come illustrato nella Tabella 10, lo schema di valutazione del metodo DREAD per ogni minaccia prevede l'assegnazione di punti da 1 a 3, con un cumulo di 15 punti che indica il rischio più grave.

Rating	High	Medium	Low
Damage (D)	Extensive data loss, compromise of full system	Moderate data loss, potential compromise of personal or sensitive data	Limited data loss, minor information
Reproducibility (R)	Highly unlikely to be reproduced, requires extremely specific and uncommon circumstances	Possible to reproduce, but requires specialized knowledge or specific conditions	Easily reproducible with minimal effort
Exploitability (E)	Requires extensive knowledge, sophisticated tools and complex methods	Requires moderate technical skills, advanced tools and some effort	Requires basic technical knowledge and commonly available tools
Affected Users (A)	Many users affected, substantial impact on user privacy or security	Some users affected, potential inconvenience or minimal harm	Few users affected, limited impact on individuals
Discoverability (D)	Highly hidden, requires specialized expertise, extensive analysis, or insider knowledge	Hidden but discoverable with careful examination or targeted testing	Easily detected

Tabella 10: Schema di valutazione del metodo DREAD

Il rischio DREAD può essere calcolato come segue:

$$\text{Rischio} = (D + R + E + A + D)$$

Dopo aver sommato i punteggi, il risultato può variare nell'intervallo 5-15. Successivamente, le minacce possono essere classificate come:

- Quelle con valutazioni totali di 12-15 sono considerate ad alto rischio;
- Quelle di 8-11 indicano un rischio medio;
- Quelle di 5-7 sono considerate a basso rischio [4].

Valutazione delle minacce con DREAD

La modellazione delle minacce viene eseguita per valutare la possibilità di attacchi informatici associati ai principali flussi di dati e processi della DFD. Infatti, vi è da considerare il fatto che non tutti i componenti del DFD vengono analizzati per individuare potenziali minacce. In questo senso, la modellazione delle minacce non viene eseguita su: buffer video, sul controller del touch screen, sullo schermo posteriore, sul touch screen, sul sistema audio dell'auto, sull'altoparlante e sul sistema di intrattenimento, la telecamera, il microfono, la radio digitale, il GPS e il sensore di temperatura, perché non vi è alcuna funzione di trasmissione di dati o file. Inoltre, non viene eseguita neanche sull'interfaccia USB e sul lettore multimediale portatile, perché devono essere fisicamente inseriti nel sistema. Le informazioni e i comandi vengono trasmessi tramite NFC, Wi-Fi, rete cellulare e Bluetooth, mentre il CAN è responsabile della comunicazione con le centraline del veicolo. Pertanto, questi punti possono essere potenziali bersagli di accessi non autorizzati da parte di attaccanti, ed è per questo che vengono considerati i seguenti punti di minaccia: da NFC a computer di bordo (NFC_to_OBC), da computer di bordo a Wi-Fi e rete cellulare (OBC_to_Wi-Fi), Wi-Fi e rete cellulare a computer di bordo (Wi-Fi_to_OBC), computer di bordo a Bluetooth (OBC_to_Bluetooth), Bluetooth a computer di bordo (Bluetooth_to_OBC), da computer di bordo a bus CAN (OBC_a_CB) e da bus CAN a computer di bordo (CB_a_OBC). Vediamo meglio i punti, appena sopra citati, nella Figura 11:

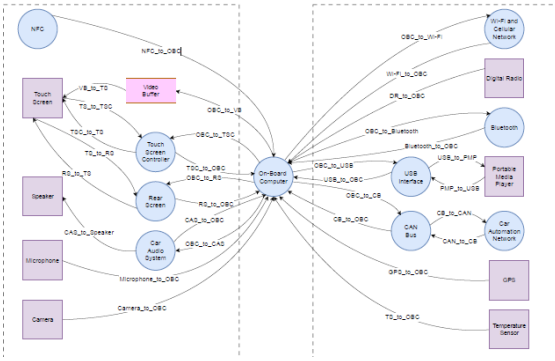


Figura 11: DFD basato sui componenti del sistema IVI

Tali punti, bersagli di accesso non autorizzati, potrebbero consentire di manipolare il sistema di Infotainment, accedere ai dati personali, controllare i componenti del veicolo o interrompere il normale funzionamento del sistema o interrompere le normali operazioni del sistema. Pertanto, è fondamentale riconoscere la possibilità di problemi di sicurezza nel sistema di Infotainment di un veicolo automobilistico. Utilizzando lo strumento di modellazione delle minacce STRIDE, le organizzazioni possono identificare efficacemente le potenziali minacce considerando ogni categoria. Ciò consente alle organizzazioni di valutare la probabilità e l'impatto degli attacchi all'interno di ciascuna categoria e attraverso tali informazioni, le organizzazioni possono sviluppare possibili strategie di mitigazione per salvaguardare i propri sistemi e le proprie reti da un'ampia gamma di minacce potenziali. La Tabella 12 elenca le minacce identificate (con ulteriori dettagli) [4]:

Components or Interactions	Threat No.	Threat Details	Threat Category
On-Board Computer	1	An adversary can replicate the user actions to impersonate the process of on-board computer.	Spoofing
	2	An adversary may modify any given command and instruction resulting in the modification of the system such as NFC to on-board computer.	Tampering
	3	Without proper monitoring and control, the on-board computer can be subject to malicious exploitation.	Repudiation
	4	An adversary may steal or share any personal information with anyone, which may violate the user's privacy.	Information Disclosure
	5	In order to deny users of the on-board computer's services, an adversary may flood it with requests so normal traffic cannot be processed.	Denial of Service
	6	Without the required authorization, an adversary might obtain access to the on-board computer and carry out privileged operations.	Elevation of Privilege
NFC_to_OBC	7	On-Board Computer may crash, halt, stop, or run slowly because of the fake requests sent by the adversary through NFC.	Denial of Service
	8	An adversary may interrupt data flowing across NFC to on-board computer with a snipping device and send a massive volume of data over the communication channel.	Denial of Service
	9	An adversary can intercept NFC data and use it to attack other parts of the system.	Information Disclosure
	10	An adversary may tamper the data flow from NFC to on-board computer in order to gain a particular advantage (not unlocking the door).	Tampering
OBC_to_Wi-Fi	11	Wi-Fi and cellular network may crash or halt due to the overflow of traffic causing not connecting to the network.	Denial of Service
	12	An adversary may interrupt data flowing across on-board computer to Wi-Fi and cellular network with a snipping device, and session hijacking may occur.	Denial of Service
	13	The data passing from on-board computer to Wi-Fi and cellular network may sniffed by the adversary causing the leakage of personal information.	Information Disclosure
	14	An adversary may tamper the data flow from on-board computer to Wi-Fi and cellular network and modify information to take remote control of the device.	Tampering
Wi-Fi_to_OBC	15	On-Board Computer may crash, halt, stop, or run slowly due to the adversary making the resources and services unavailable.	Denial of Service
	16	An adversary can disrupt the on-board computer's performance by overwhelming its communication channels with a high volume of data, interrupting Wi-Fi and cellular network data flow.	Denial of Service
	17	The data passing from Wi-Fi and cellular network to on-board computer may sniffed by the adversary. This may lead to compliance violations.	Information Disclosure
	18	An adversary may tamper the data flow from Wi-Fi and cellular network to on-board computer and alter information.	Tampering
OBC_to_Bluetooth	19	Bluetooth may crash, halt, stop, or run slowly due to the adversary making the resources and services unavailable.	Denial of Service
	20	An external adversary may interrupt data flowing across a trust boundary by sending a large amount of data over communication channel.	Denial of Service
	21	The data passing from on-board computer to Bluetooth may sniffed by the adversary and disclose call logs or messages.	Information Disclosure
	22	An adversary may tamper the data flow from on-board computer to Bluetooth and alter information.	Tampering
Bluetooth_to_OBC	23	On-Board Computer may crash, halt, stop, or run slowly because of the fake requests sent by the adversary.	Denial of Service
	24	An external adversary may interrupt data flow and keep the system busy to respond to fake requests.	Denial of Service
	25	The data passing from on-board computer to Bluetooth may sniffed by the adversary. Based on the type of information disclosure, this may lead to attacks on other parts of the system.	Information Disclosure
	26	An adversary may tamper with the data flow from Bluetooth to on-board computer and make unauthorized manipulation to the system.	Tampering
OBC_to_CB	27	An adversary may tamper the data flow from on-board computer to CAN bus and disclose the system information.	Denial of Service
	28	An adversary may interrupt data flowing across on-board computer to CAN bus in either direction.	Denial of Service
	29	An adversary may tamper the data flow from on-board computer to CAN bus and disclose the system information.	Information Disclosure
	30	An adversary can manipulate Bluetooth data to cause a denial of service or elevation of privilege on the CAN bus.	Tampering
CB_to_OBC	31	On-Board Computer may crash, halt, stop, or run slowly due to the adversary making the resources and services unavailable.	Denial of Service
	32	An adversary may interrupt data flow across CAN bus to on-board computer in either direction.	Denial of Service
	33	An adversary can sniff the data flow, potentially enabling attacks on other system components based on the disclosed information.	Information Disclosure
	34	An adversary may tamper the data flow from CAN bus to on-board computer and alter information.	Tampering

Tabella 12: Elenco minacce sistema di Infotainment

Valutazione delle minacce con SAHARA

Il metodo SAHARA viene utilizzato per analizzare le minacce alla sicurezza nelle prime fasi dello sviluppo automobilistico. Questo metodo si concentra sullo sviluppo di un singolo veicolo e sull'identificazione delle minacce e dei rischi per la sicurezza durante le fasi iniziali di sviluppo. L'analisi SAHARA è condotta attraverso un processo convenzionale, determinando il SecL. Il metodo SAHARA prevede un requisito moderato (valore di K pari a 2) e risorse di calcolo moderate (valore di R pari a 2) per il calcolo dei valori di rischio associati ad una minaccia specifica. Da notare, che utilizzando il metodo DREAD (visto nella sezione precedente), le minacce vengono valutate in base all'impatto degli attacchi. In particolare, le minacce ricevono un punteggio DREAD di 3 per l'impatto elevato, mentre il fattore di distruzione (D) ottiene un punteggio di 2 per l'impatto medio. Le minacce con un punteggio cumulativo superiore sono considerate di alta priorità. I valori di rischio calcolati utilizzando i metodi SAHARA e DREAD sono presentati nella Tabella 13 [4]:

Threat No.	SAHARA					DREAD						
	K	R	T	SecL	Priority	D	R	E	A	D	Sum	Priority
1	2	2	3	1	High	3	3	3	3	2	13	High
2	2	2	2	0	Low	3	2	3	2	2	10	Medium
3	2	3	3	1	High	3	2	3	2	2	12	High
4	2	2	3	1	High	3	2	2	3	2	12	High
5	1	2	2	1	Low	2	2	3	2	2	11	Medium
6	2	3	3	1	High	3	2	2	2	3	12	High
7	1	2	2	1	Low	2	3	2	3	2	12	High
8	2	3	3	1	High	3	3	2	3	1	12	High
9	2	2	3	1	High	3	2	3	2	2	12	High
10	2	1	3	2	High	3	2	3	3	2	13	High
11	1	3	2	0	Low	2	3	1	2	2	10	Medium
12	2	3	3	1	High	2	2	3	3	2	12	High
13	1	2	3	2	High	3	2	3	3	2	13	High
14	2	3	3	1	High	3	2	3	2	2	12	High
15	2	3	3	1	High	3	2	2	3	2	12	High
16	2	3	3	1	High	2	3	2	3	2	12	High
17	2	2	3	1	High	3	2	2	2	3	12	High
18	2	3	3	1	High	3	2	3	2	2	12	High
19	1	2	2	1	Low	3	2	2	3	2	12	High
20	2	3	3	1	High	2	2	3	3	2	12	High
21	2	2	3	1	High	3	2	2	3	2	12	High
22	2	2	3	1	High	3	2	3	2	2	12	High
23	1	2	3	2	High	2	3	2	3	2	12	High
24	2	2	3	1	High	2	3	2	3	2	12	High
25	2	2	3	1	High	3	2	2	2	2	12	High
26	2	2	3	1	High	3	2	2	3	2	12	High
27	1	2	3	2	High	3	2	3	3	2	13	High
28	2	2	3	1	High	2	2	3	3	2	12	High
29	2	2	3	1	High	3	2	3	2	2	12	High
30	2	3	3	1	High	3	2	3	3	2	13	High
31	1	3	3	1	High	3	2	2	3	2	12	High
32	2	2	3	1	High	3	2	3	3	2	13	High
33	2	2	3	1	High	3	2	2	3	2	12	High
34	2	2	3	1	High	3	2	2	3	2	12	High

Tabella 13: Categorizzazione delle minacce utilizzando le metodologie di classificazione delle minacce SAHARA e DREAD

Considerazioni finali

Nel processo di identificazione delle minacce alla sicurezza informatica, lo strumento Microsoft STRIDE viene applicato ai componenti, ai flussi di dati, agli archivi di dati e alle entità esterne selezionati all'interno della DFD. In totale sono state riconosciute 34 minacce, sistematicamente classificate in sei categorie STRIDE. È importante notare, che tutte le minacce identificate, come derivate dallo scenario del caso d'uso, sono potenzialmente soggettive e possono presentare variazioni in scenari diversi. Queste minacce riconosciute devono essere prese in considerazione prima dell'implementazione del sistema di Infotainment in veicoli automobilistici reali, per garantire la sicurezza del sistema. Per la valutazione del rischio delle minacce identificate, sono state utilizzate le metodologie SAHARA e DREAD ed i relativi valori di rischio sono classificati per priorità: alta, media e bassa. Utilizzando la metodologia SAHARA, 29 minacce sono classificate come ad alta priorità, mentre nessuna rientra nel rischio medio e 5 sono classificate come priorità bassa.

Utilizzando la metodologia DREAD, invece, 31 minacce sono identificate come ad alta priorità, 3 come a media priorità e nessuna a bassa priorità. Il numero di minacce ad alta priorità che richiedono attenzione immediata è quasi simile in entrambe le metodologie. Le minacce ad alta priorità, che comportano valori di rischio significativi, richiedono l'implementazione immediata di contromisure.

Per garantire la sicurezza e l'integrità del sistema e proteggerlo da potenziali compromissioni, vengono adottate una serie di meccanismi di difesa. In particolare, quando si tratta di minacce associate allo spoofing, l'implementazione di un'autenticazione a più fattori o di metodi di autenticazione biometrica si dimostrano molto efficaci nel mitigare queste minacce all'interno del sistema. Per affrontare attacchi di manomissione, è fondamentale impiegare tecnologie di crittografia e firma digitale, che possono rafforzare la resistenza del sistema contro le alterazioni non autorizzate e la manipolazione dei dati. A panoramica completa dell'insieme dei meccanismi di difesa è riportata nella Tabella 14. Queste strategie lavorano collettivamente per aumentare la sicurezza del sistema e ridurre al minimo le sue vulnerabilità a vari tipi di minacce informatiche [4].

STRIDE Category	Threat Details	Mitigation
Spoofing	Adversary pretends to be a legitimate user or system	Multi-factor authentication [50–52], Biometric authentication [53,54]
Tampering	Adversary modifies data or software without authorization	Encryption [55], Digital signature [56]
Repudiation	Adversary denies responsibility for actions they have taken	Logging and auditing mechanisms to track and trace user actions [57]
Information Disclosure	Adversary gains access to sensitive information	Access controls and permissions to limit access to sensitive data [58,59]
Denial of Service	Adversary prevents legitimate users from accessing a system or service	Rate limiting and load balancing to distribute traffic across multiple servers [60,61]
Elevation of Privilege	Adversary gains higher levels of access than they are authorized to have	Secure coding practices [62], User activity monitoring and logging to detect potential privilege escalation attempts [63]

Tabella 14: Meccanismi di difesa della cybersecurity contro la categoria STRIDE.

Attacchi alla connettività Bluetooth

Le automobili moderne sono dotate di funzioni di connettività per migliorare il comfort dell'utente. Il Bluetooth è una di queste tecnologie di comunicazione, che viene utilizzata per accoppiare un dispositivo personale con l'unità di Infotainment dell'automobile. Dopo l'accoppiamento, l'utente può accedere alle informazioni personali sul telefono attraverso l'unità principale dell'automobile, con la minima distrazione durante la guida. Pertanto, le case automobilistiche lavorano costantemente per migliorare la connettività, al fine di ottenere una mobilità intelligente. Tuttavia, funzionalità crescenti aprono la strada a molteplici vulnerabilità di sicurezza e potenziali attacchi. Ad esempio, una maggiore connettività wireless aumenta il numero di dispositivi esterni che hanno accesso alla rete veicolare. Di conseguenza, aumenta il numero di possibili punti deboli per sfruttare un sistema, attraverso i quali si possono realizzare azioni di controllo non sicure, che possono portare a situazioni di pericolo. Visto che, le tecniche di difesa standard nella sicurezza del software (come i firewall e la crittografia) falliscono nei complessi sistemi embedded a livello veicolare, a causa della natura eterogenea di molte varianti e configurazioni del veicolo, è pertanto necessario l'esecuzione di un'analisi approfondita del sistema, in modo tale da prevenire l'accesso non autorizzato alle informazioni personali. A questo punto, quindi, ci concentriamo sull'automotive Android e svolgiamo un'analisi a livello di sistema, di sottosistema e di controllo sulle funzionalità Bluetooth del sistema IVI [7].

Panoramica Bluetooth

Il Bluetooth è una tecnologia wireless che opera nella banda di frequenza ISM (Industrial, Scientific, and Medical) senza licenza 2,4 GHz. Il Bluetooth utilizza una frequenza ultraelevata (UHF) con un raggio d'azione effettivo di 10-100 metri (senza estensori esterni come amplificatori e antenne direzionali). Durante la connessione in Bluetooth, un dispositivo è designato come leader e tutti gli altri dispositivi sono follower. Quando la connessione è riuscita, il follower si sincronizza con il clock del leader per ottenere il corretto schema di salto di frequenza. Con 79 canali di frequenza da saltare, la probabilità di interferenze tra altri dispositivi Bluetooth è estremamente bassa. I due tipi principali di dispositivi Bluetooth sono:

1. Il dispositivo Bluetooth classico che opera alla velocità di base (BR) o alla velocità di trasmissione potenziata (EDR);
2. Bluetooth Low Energy (BLE).

Questi dispositivi, con architetture diverse, comunicano tra loro in modalità duale, ovverosia che ognuno di questi dispositivi è in grado di supportare entrambe le modalità di comunicazione Bluetooth contemporaneamente. Il Bluetooth è protetto attraverso autenticazione, crittografia e autorizzazione e tutti i dispositivi Bluetooth hanno un indirizzo univoco di 48 bit assegnato dal produttore. I componenti principali dell'architettura Bluetooth sono:

- Controllore Bluetooth;
- Interfaccia controllore host (HCI);
- Host Bluetooth [7].

Architettura Bluetooth Android

La versione astratta dell'architettura Bluetooth di Android è mostrata nella Figura 15:

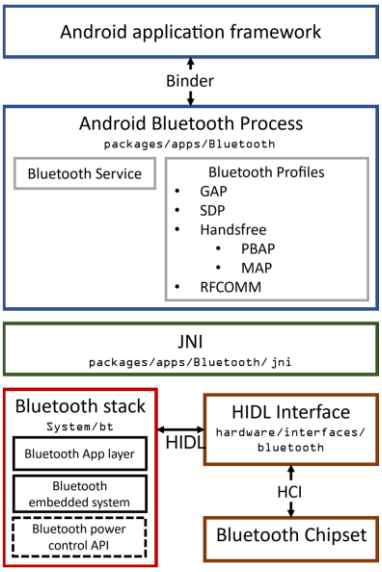


Figura 15: Architettura astratta di Android

Possiamo osservare, che le API del livello applicazione comunicano con i servizi Bluetooth e i profili Bluetooth situati in packages/apps/Bluetooth attraverso Binder. Il processo Bluetooth comunica con lo Stack Bluetooth tramite Java Native Interface (JNI). Le configurazioni richieste nell'HAL sono

implementate attraverso lo Stack Bluetooth. Lo Stack Bluetooth personalizzabile comunica con il chipset Bluetooth incorporato attraverso HIDL (HAL Interface Definition Language). I controlli di livello inferiore del chipset Bluetooth comprendono il controller radio, il controller della banda base, ecc. Comunicano con l'host attraverso HCI. I rispettivi host implementano i protocolli nel livello intermedio (Stack Bluetooth e processo Bluetooth) e nel livello applicazione. Pertanto, l'implementazione dello Stack Bluetooth e i requisiti dei profili Bluetooth dipendono dall'host, e qualsiasi modifica allo Stack o ai profili può introdurre nuove vulnerabilità. Queste vulnerabilità sono i bug di implementazione introdotti nel sistema operativo. Per restringere l'analisi in conformità con STPA, nel documento si decide di selezionare (in modo specifico) solamente quei sottosistemi che portano a vulnerabilità e che violano il nostro vincolo di privacy dichiarato. I profili Bluetooth e lo Stack Bluetooth sono due di questi sottosistemi. Questo perché il processo di memorizzazione e di cancellazione dei dati riservati dell'utente viene eseguito nello Stack Bluetooth e l'autorizzazione per l'accesso ai dati dell'utente è ottenuto attraverso i profili Bluetooth [7].

Profili Bluetooth

I profili Bluetooth definiscono il protocollo standard delle applicazioni del dispositivo Bluetooth. Definisce in modo specifico quali dati vengono trasmessi attraverso la connessione Bluetooth. A seconda del profilo, Bluetooth SIG (Bluetooth Special Interest Group, ovverosia un'organizzazione che sovrintende lo sviluppo degli standard Bluetooth) ha diversi protocolli di trasporto fisico. Con più di 30 profili standardizzati, il documento preso in esame, si concentra sui profili PBAP e MAP, in quanto sono i profili Bluetooth legati alla privacy che memorizzano i dati in chiaro nei registri Bluetooth. A questo punto, vengono analizzati in maniera più approfondita i due profili appena sopra citati:

- Phone Book Access Profile (PBAP) → si basa su un'interazione Client-Server, in cui il Client (Phone book Client Equipment-PCE) riceve l'oggetto della rubrica telefonica dal dispositivo Server (Phone book Server Server-PSE). Nel nostro caso, il telefono dell'utente è il PSE e l'unità IVI è PCE. Per la sua convenienza nell'applicazione a mani libere all'interno del veicolo, il PBAP è uno dei profili Bluetooth più importanti dell'unità IVI. In questo caso, il Bluetooth SIG impone requisiti di sicurezza specifici per il PBAP:
    - Il PCE può richiedere al PSE l'accesso alla rubrica telefonica solo dopo che una connessione è riuscita;
    - L'inizializzazione della connessione deve includere la scoperta del servizio, i messaggi di inizializzazione della sicurezza, le chiavi di collegamento e la crittografia;
    - La procedura di autenticazione come descritto nel Generic Access Profile (GAP) deve essere eseguita;
    - L'utente del PSE deve confermare l'accesso per la condivisione della propria rubrica telefonica.

  L'intera rubrica viene solitamente scaricata e memorizzata nel dispositivo PCE. La trasmissione dei dati dal PSE al PCE utilizza il Profilo generico di scambio di oggetti;

- Messagge Access Profile (MAP) → MAP è simile a PBAP e utilizza un'interazione Client-Server simile per lo scambio di oggetti messaggio. Il MAP nel profilo hands-free dell'unità IVI offre la comodità di utilizzare l'HMI o anche i comandi vocali attraverso l'impianto audio per leggere, inviare, notificare o sfogliare facilmente i messaggi. Le versioni MAP supportate sono SMS, MMS, e-mail e messaggi istantanei (IM). I requisiti di sicurezza per il MAP sono molto simili a

quelli del PBAP: requisiti di accoppiamento e crittografia, autenticazione con GAP, e la conformità dell'utente [7].

Modello di minaccia e descrizione dell'attacco

La sicurezza Bluetooth prevede l'autenticazione, l'autorizzazione e la crittografia basati sulla premessa che l'utente si fida del dispositivo con il quale sta accoppiando il suo dispositivo personale. Tuttavia, questo presupposto non può essere sempre vero con le unità IVI nei veicoli, soprattutto quelli gestiti da più utenti. Un utente malintenzionato potrebbe manipolare l'unità IVI e compromettere furtivamente la privacy dell'utente. Android, essendo un sistema operativo gratuito e open-source, è molto developer friendly e quindi, uno dei modi migliori per testare un software o un'applicazione sviluppata è quello di testarla fisicamente su un dispositivo. Per questo motivo, a scopo di test, Android dispone di opzioni per gli sviluppatori, che consentono loro di accedere ad alcune funzionalità del dispositivo che di solito sono bloccate. Le opzioni per gli sviluppatori nei dispositivi Android sono "nascoste" in una posizione facilmente accessibile. Secondo la comunità Android, è sicuro e protetto avere le opzioni per gli sviluppatori abilitate, e l'abilitazione di tali opzioni non invalida la garanzia del dispositivo. Per questo motivo, è estremamente difficile per un utente tipico sapere se le opzioni per gli sviluppatori sono abilitate o meno, a meno che non lo si va a vedere nelle impostazioni del dispositivo. Una caratteristica essenziale delle opzioni per gli sviluppatori è il registro di snoop (snoop log) di Bluetooth HCI, in quanto questo registro memorizza PBAP e MAP in plain text. Il file di log viene creato al momento della connessione Bluetooth e cattura, monitora e analizza i pacchetti Bluetooth. Questi dati vengono memorizzati nel dispositivo e possono essere recuperati tramite USB o debug wireless di Android. Nell'articolo selezionato, viene sfruttata questa caratteristica per condurre con successo attacchi alla privacy sull'unità IVI con sistema operativo Android. Il successo dell'esecuzione dell'attacco dipende dall'abilitazione delle opzioni dello sviluppatore, che consideriamo un bug di implementazione e una debolezza sfruttabile nello Stack Bluetooth.

Quando l'utente accoppia il proprio telefono con il sistema IVI tramite Bluetooth per applicazioni hands-free, come: chiamate, sms e intrattenimento, le loro informazioni personali vengono sincronizzate con l'unità IVI in base ai profili Bluetooth. L'attaccante può recuperare queste informazioni memorizzate attraverso una connessione cablata con il veicolo (USB nel nostro caso) o attraverso una connessione wireless (la comune rete Wi-Fi nel nostro caso). I dati (contatti personali, messaggi e registri delle chiamate) recuperati dall'attaccante attraverso le opzioni dello sviluppatore non sono crittografati. Inoltre, si evidenzia un'altra importante scoperta, ovverosia che dopo aver analizzato la macchina a stati del sistema IVI per l'alimentazione dell'auto (vedi Figura 16), si è anche scoperto che la funzione di cancellazione della memoria (ovvero la funzione di Sospensione dalla memoria RAM) è attivata dallo spegnimento del veicolo e non in base allo stato della connessione Bluetooth.

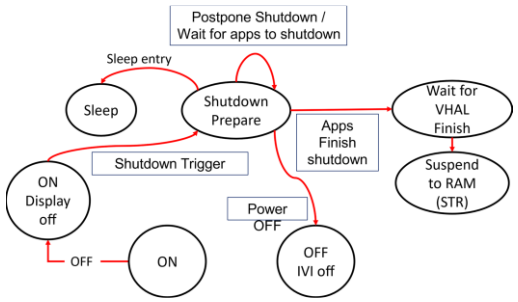


Figura 16: Macchina a stati proposta per mitigare l'attacco

Pertanto, i dati memorizzati nel sistema IVI rimangono nel sistema fino allo spegnimento del veicolo, aumentando la finestra temporale per l'attacco. L'articolo evidenzia, che sono stati testati gli attacchi proposti su un veicolo di produzione reale con sistema operativo Android versione 6.0.1 nel suo sistema di Infotainment. La procedura di attacco è la seguente:

1. Abilitare le opzioni di sviluppo nell'unità Android IVI:

    a. In alcuni sistemi, questa potrebbe essere un'operazione simile al telefono cellulare Android. Nei dispositivi con Android 9 o superiore, toccando 7 volte il numero di build in Impostazioni > Informazioni sul telefono > Numero build;

    b. In alcuni veicoli, questa funzione potrebbe essere nascosta dal costruttore, che può essere sbloccata selezionando una combinazione di pulsanti nell'unità IVI. Tuttavia, la decifrazione della combinazione di pulsanti non è complicata; per le unità più diffuse, sono disponibili sui forum online.

2. Abilitare lo snoop log di Bluetooth HCI nelle opzioni per gli sviluppatori. Impostazioni > Opzioni sviluppatore > Attiva Bluetooth HCI snoop log.

3. Recuperare i dati dall'unità IVI dopo che l'utente ha collegato il proprio dispositivo personale. Il file di log viene solitamente memorizzato in una memoria. Questo può essere trasferito su un dispositivo di memoria attraverso la porta USB;

4. Analizzare il registro catturato in Wireshark. I pacchetti Bluetooth OBEX catturati nel registro rivelano i contatti telefonici, i registri delle chiamate e i messaggi in chiaro. Ciò implica che l'attaccante avrebbe accesso a tutti i contatti e ai messaggi di testo precedenti e attuali, che potrebbero includere messaggi critici per la sicurezza da parte delle banche, messaggi di reimpostazione della password o anche One-Time password (OTP), il che potrebbe portare a violazioni della privacy.

Come possiamo osservare, le fasi di esecuzione dell'attacco non sono tecnicamente complicate e non richiedono costosi strumenti di calcolo. Di conseguenza, l'attaccante potrebbe facilmente utilizzare l'attacco proposto per sfruttare la privacy di una vittima target. L'attacco evidenziato nell'articolo, quindi, identifica due potenziali vulnerabilità e propone un metodo per sfruttare tali vulnerabilità. Tuttavia, è evidente che l'attacco ha una portata limitata nell'esecuzione pratica, in quanto l'attacco si basa sulla possibilità di alterare lo stato originale del sistema di Infotainment, ovverosia l'attacco si basa sul fatto, che l'attaccante deve abilitare le opzioni dello sviluppatore e lo snoop log di Bluetooth HCI. Capiamo, allora, che l'attaccante può accedere a informazioni riservate e private dal dispositivo della vittima solo quando l'unità di Infotainment si trova nella stessa rete dell'attaccante o l'attaccante ha accesso fisico al veicolo. Da notare, infine, che l'attacco non è possibile se il veicolo viene spento, prima che l'attaccante possa effettivamente accedere al veicolo [7].

Possibili contromisure

Questa sezione propone potenziali contromisure basate sui risultati dell'analisi del sistema. Anche se la crittografia del gestore della rubrica è un'opzione valida, non è sufficiente. Alcune interfacce HMI del sistema di Infotainment richiedono dati decifrati per determinati servizi. Per questo motivo, vengono formulate delle contromisure per l'attacco specifico, in modo tale da garantire l'affidabilità del sistema con le informazioni personali. Tali contromisure sono le seguenti:

- Controllare se le opzioni dello sviluppatore sono abilitate nell'unità IVI. Se le opzioni dello sviluppatore sono abilitate, informare l'utente e richiedergli un ulteriore consenso. In questo modo l'utente sarà consapevole della potenziale situazione di attacco o dei rischi per la privacy. Potrebbe essere disposti a spegnere l'accensione prima di consegnare il veicolo a un aggressore o di essere prudenti quando collegare il proprio dispositivo ad auto sconosciute;
- Controllare frequentemente lo stato del Bluetooth e, se lo stato si sta preparando a una disconnessione, chiamare lo stato di Wait for VHAL Finish e passare allo stato Sospensione su RAM (STR). Lo stato STR cancella la memoria e tutti i dati personali salvati sull'unità IVI al momento della disconnessione. Pertanto, lo stato STR cancella automaticamente la memoria quando l'utente si allontana dopo aver lasciato l'auto all'aggressore e si allontana dal raggio d'azione del Bluetooth. Di conseguenza, restringe in larga misura il tempo di attacco [7].

Altre contromisure che abbiamo pensato sono:

- Implementare un meccanismo di autenticazione più robusto, per esempio a due fattori o con chiavi di sicurezza più lunghe, per la connessione Bluetooth tra smartphone e il sistema di Infotainment IVI;
- Implementare delle politiche di accesso basate sulla posizione del veicolo. Per esempio, si potrebbe applicare delle politiche più severe (come l'impossibilità di accedere all'IVI) quando il veicolo è parcheggiato in luoghi pubblici o quando l'utente si trova lontano dal veicolo;
- Monitorare l'integrità del sistema di Infotainment e allertare gli utenti o intraprendere azioni correttive se vengono rilevati cambiamenti non autorizzati o attività sospette.

ATTACCHI AI SISTEMI DI "ROAD SAFETY"

Introduzione

Una delle regole fondamentali delle strade è garantire la sicurezza dei passeggeri e dei conducenti, e proprio per questo motivo, in ambito automotive sono state pensate delle "applicazioni" che permettessero di migliorare alcuni aspetti della sicurezza stradale.
Alcune di queste applicazioni in realtà non sono altro che sistemi informatici già altamente utilizzati al di fuori dell'ambito automotive, mentre altri sono stati creati ad-hoc per il mondo della guida autonoma; tra tutte queste applicazioni, alcune tra le più importanti sono:

- I sistemi *LiDAR* (Light Detection And Ranging): sistemi di rilevamento di oggetti nei pressi della strada, come per esempio altri veicoli o pedoni.
- I sistemi *GPS* (Global Positioning System): conosciutissimi ed essenziali affinché i veicoli possano funzionare correttamente e autonomamente senza l'impiego dell'interazione umana.
- Le comunicazioni *V2X* (Vehicle to everything): tecnologie wirless moderne utilizzate dalle auto a guida autonoma per connettere tra loro i vari elementi delle strade (veicoli, pedoni, infrastrutture e le strade stesse) e consentire la trasmissione e ricezione di messaggi relativi alle velocità delle vetture, presenza di incidenti ecc.
- Le reti *VANET* (Vehicular Ad-hoc NETworks): reti sviluppate per facilitare la sicurezza del traffico e l'ottimizzazione del flusso del traffico.

Queste tecnologie fortemente adottate dalle case automobilistiche per la creazione di auto a guida autonoma, consentono una risposta rapida alle emergenze, contribuiscono a ridurre il problema del sovraffollamento delle strade in determinate situazioni e in generale permettono di ridurre drasticamente il rischio di incidenti; proprio per questo vengono definiti sistemi di "road safety".
Tuttavia, come molte altre tecnologie, queste appena presentate non sono esenti da vulnerabilità che un eventuale attaccante potrebbe sfruttare per i più diversi scopi; basti pensare al semplice *GPS* per esempio, che è uno standard aperto e di pubblico dominio.
È chiaro però che un eventuale attacco a questi sistemi o un danneggiamento/una manomissione degli stessi, metterebbe in grave pericolo l'incolumità dei conducenti, dei passeggeri ma anche di tutte le persone che si trovano nei pressi della strada; essendo quindi dei sistemi critici per quanto riguarda la sicurezza nelle strade, è importante saper individuarne tutte le possibili vulnerabilità, i possibili attacchi e soprattuto le contromisure per prevenirli o eventualmente identificarli e bloccarli.
In questa sezione del progetto dunque, verranno descritte più nel dettaglio le tecnologie sopra citate, verranno riportate le possibili vulnerabilità e minacce associate a ciascuna di esse ed infine verranno proposte e discusse una serie di possibili contromisure (alcune delle quali sono solo "proof-of-concept") da poter adottare.

Attacchi ai sistemi LiDAR

Le auto a guida autonoma sono altamente dipendenti dai sistemi *LiDAR*, i quali vengono utilizzati per generare mappe 3D dell'ambiente che circonda un veicolo, per localizzare ed evitare ostacoli e per la navigazione in generale.
Tra i diversi sensori *LiDAR* disponibili, c'è una tipologia principale che viene comunemente utilizzata nei veicoli a guida autonoma: i *LiDAR* rotanti.
Questi *LiDAR* rotanti utilizzano ottiche di alta qualità e hardware meccanici rotanti per ottenere la visione completa a 360° dell'ambiente circostante il veicolo.
Tipicamente, questi tipi di *LiDAR* sono composti da una "pila" verticale di diodi laser a infrarossi che emettono tutti assieme impulsi laser ed una pila corrispondente di fotodiodi; entrambe le pile ruotano e ciascuna coppia fotodiodo-diodo laser copre un particolare angolo, formando così le linee della mappa 3D risultante.
Se un raggio laser viene parzialmente o totalmente intercettato da ostacoli sulla sua traiettoria, i risultati delle riflessioni sulle superfici dell'ostacolo (chiamati *echi*) vengono retrodiffusi verso il sensore.
Il circuito interno del *LiDAR* misura il tempo tra l'emissione dell'impulso laser e il ritorno ai fotodiodi, che viene poi tradotto nella distanza tra gli ostacoli e il veicolo; il sensore cattura anche la riflettanza (intensità) degli *echi*.
Varie ricerche hanno però dimostrato come le auto a guida autonoma siano vulnerabili agli attacchi ai sensori *LiDAR*[9].

Esempio proof-of-concept di attacco al sistema LiDAR

L'attacco proposto in uno dei paper presi in esame, consiste nell'iniettare *echi* invisibili in prossimità del sensore *LiDAR* per forzare l'eliminazione automatica dei punti di nuvola legittimi nella scena (ovvero i punti di nuvola prodotti da ostacoli veri) e quindi non permettere il riconoscimento degli ostacoli e la loro posizione.
Gli *echi* falsi sono sincronizzati con la sequenza temporale di attivazione del sensore *LiDAR* per controllare la posizione precisa dei punti di nuvola falsificati: l'aggressore può sincronizzare il dispositivo spoofer per modificare la distanza Δr dei punti falsificati dal *LiDAR* e ciò si ottiene modificando il ritardo degli impulsi laser spoofer sparati in base alla sequenza di attivazione del raggio laser *LiDAR* della vittima.

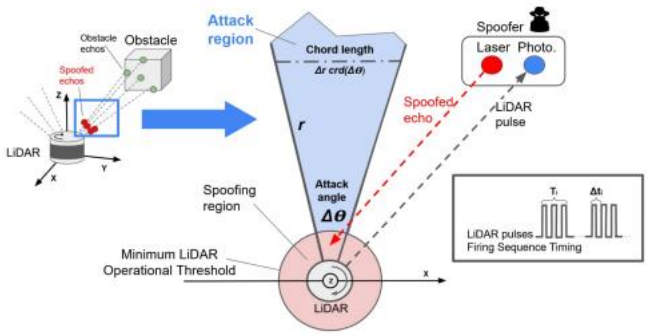


Figura 17. Rappresentazione delle entità coinvolte nell'attacco

L'avversario può quindi sfruttare questo effetto per nascondere oggetti, altri veicoli o pedoni davanti al veicolo per un periodo di tempo sufficiente a provocare un potenziale incidente o a indurre manovre automatiche non sicure dell'ultimo secondo come sterzare e deviare nelle corsie adiacenti, aumentando così il rischio di essere colpiti dai veicoli vicini.

Questo tipo di attacco, purtroppo, funziona a causa di due fattori principali:

1. I sensori *LiDAR*, per loro natura, possono ricevere più di un segnale di *eco* da ostacoli del mondo reale (se per esempio il fascio laser colpisce più oggetti lungo il suo percorso); essendo però l'acquisizione e gestione multipla di echi complessa, i sensori *LiDAR* danno intrinsecamente priorità agli echi con intensità maggiore.
   Ciò si basa sul presupposto che gli echi con intensità maggiore provengono da ostacoli vicini che sono più critici da rilevare.
2. I sensori *LiDAR*, filtrano automaticamente i punti di nuvola più vicini all'involucro del sensore

Nell'attacco proposto quindi, vengono iniettati *echi* ad alta intensità in prossimità del sensore per far sì che questo registri i falsi riflessi come quelli più forti, ignorando altri *echi* generati da ostacoli reali più lontani; ovviamente i punti di nuvola falsificati, vengono si considerati dal *LiDAR* come gli unici da considerare però verranno processati in modo che nella mappa 3D risultante risultino invisibili.
Inoltre per aumentare la percentuale di successo dell'attacco, gli autori del paper hanno mirato a falsificare la più ampia regione di attacco possibile.

Per illustrare le conseguenze dell'attacco in un contesto automotive, viene simulato il tutto in un simulatore; l'obiettivo della simulazione consiste nel valutare i cambiamenti di traiettoria e velocità del veicolo, in diversi scenari in cui l'avversario lancia l'attacco di rimozione con l'intento di nascondere un ostacolo sul percorso della vittima.
Per condurre l'attacco, viene considerato uno scenario in cui il veicolo autonomo avanza, avvicinandosi a un ostacolo che si trova al centro della traiettoria dell'auto, con lo spoofer dell'aggressore situato sul lato della strada.
I risultati dela simulazione dimostrano come un attacco di questo tipo, possa portare a gravi conseguenze e mettere in pericolo la vittima; la Figura 2 esemplifica bene ciò.
Ciò accade perché senza l'attacco si prevede che l'auto:

1. Acceleri per raggiungere una velocità preimpostata (32 km/h).
2. Non appena viene rilevato l'ostacolo, deceleri in modo uniforme e si fermi prima di raggiungere l'ostacolo.

Pertanto, quando inizia l'attacco e l'ostacolo bersaglio viene rimosso, l'auto della vittima accelera per raggiungere la velocità preimpostata, senza decelerare.
Anche nel caso in cui l'ostacolo fosse percepito in un secondo momento e non subito, l'auto può ancora scontrarsi con esso a meno che la distanza *d* tra i due sia tale che $d < \frac{v^2}{2a}$ dove *v* è la velocità dell'auto quando l'ostacolo riappare e *a* è la decelerazione.
I risultati mostrano infatti che questo tipo di attacco può impedire alle auto a guida autonoma di frenare e fermarsi prima di scontrarsi con un ostacolo anche se l'ostacolo si trova all'interno della regione di attacco solo per il 40% dell'intero tempo di percorso; quindi anche con capacità limitate di attacco, la rimozione può portare a gravi conseguenze per le auto a guida autonoma[9].

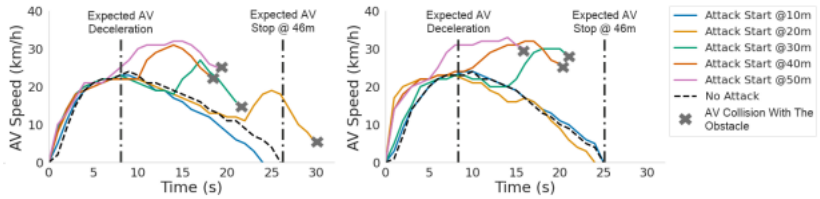


Figura 18. Grafici delle simulazioni effettuate per l'attacco al sistema LiDAR di un veicolo. La prima figura è associata ad un attacco con presenza di pedone, mentre la seconda con presenza di un altro veicolo fermo.

Contromisure proposte per attacchi ai sistemi LiDAR

Nello stesso paper viene proposta una metodologia di mitigazione per questi tipi di attacchi: la metodologia *FSD* (Fake Shadow Detection).
Consiste nell'identificare le regioni d'ombra presenti nel terreno e successivamente confrontarle con le ombre previste degli oggetti rilevati dal *LiDAR*; questo confronto si ottiene proiettando sul terreno i punti di nuvola di un oggetto rilevato e filtrando la corrispondente regione d'ombra in base a questa proiezione.
Ovviamente se il sistema rileva la presenza di una zona d'ombra ma non riesce a trovare dei punti di nuvola che possano essere associati a tale zona, la *FSD* funziona in maniera analoga.
Un'altra soluzione proposta, più semplice e pratica per rilevare il nostro attacco di rimozione, è cercare "disparità" nei dati grezzi dei punti di nuvola; possiamo ottenere ciò esaminando la vista angolare orizzontale del *LiDAR* poiché l'attacco di rimozione crea uno spazio totalmente privo di punti di nuvola lungo l'intero angolo di attacco.
Pertanto l'approccio consiste nel calcolare i valori di tutti i punti di nuvola nella scena, ordinarli e collocarli in una ipotetica mappa 3D e tutti i punti di nuvola mancanti in un certo angolo orizzontale (1°-2°), corrisponderanno a degli ostacoli mancanti[9].

Analisi critica sulle metodologie di mitigazione proposte e possibili altre misure di protezione

La prima metodologia porposta nel paper [lidar], è sicuramente una misura di difesa molto sofisticata e ipoteticamente ben funzionante, ma probabilmente anche molto difficile da mettere in atto, soprattutto quando le condizioni meteo/stradali non sono ottimali; il fatto di fare affidamento a delle ombre sul terreno non sembra essere né tanto semplice da implementare nè affidabile al 100% in qualsiasi momento e in qualsiasi situazione.
Il secondo metodo invece, rispetto all'approccio precedente, è sicuramente una tecnica di mitigazione più semplice da poter implementare e su cui è più facile fare affidamento: il fatto che non ci siano dei punti di nuvola per la totalità di un certo range angolare, risulta essere un buon criterio su cui potersi basare.
Il rischio ovviamente è quello di imbattersi in falsi positivi, ma le conseguenze (anche nella peggiore delle ipotesi) sarebbero tollerabili.
L'esempio di attacco ai sistemi *LiDAR* sopra riportato, è solo uno dei tanti che potrebbero nuocere alla sicurezza delle auto a guida autonoma e dei loro passeggeri; non basta dunque sfruttare le metodologie sopra riportate per proteggere totalmente i sistemi *LiDAR*, ma è importante adottare anche altre misure di protezione e prevenzione di attacchi.
Alcuni esempi potrebbero essere:

- Assicurarsi che il *LiDAR* sia fisicamente protetto da accessi non autorizzati o manipolazioni esterne (come nel caso dell'attacco descritto sopra).

- Posizionare i sensori in posizioni difficili da raggiungere.
- Utilizzare più di un tipo di sensore o tecnologia di rilevamento per diversificare le fonti di informazioni sull'ambiente; in questo modo sarebbe possibile fare un confronto tra i dati rilevati dai diversi sensori.
- Utilizzare protocolli di crittografia robusti per proteggere le comunicazioni tra il *LiDAR* e il sistema di controllo dell'auto (considerando che un attaccante esterno potrebbe semplicemente corrompere o manipolare le comunicazioni all'interno del veicolo stesso).

Attacchi ai sistemi GPS

Il sistema *GPS* (Global Positioning System) è un sistema di navigazione satellitare che consente di determinare la posizione geografica e l'altitudine di un ricevitore in un qualsiasi punto della Terra, in qualsiasi momento, purché il ricevitore sia in linea di vista con almeno quattro satelliti *GPS* operativi; Il ricevitore *GPS* misura la distanza da questi satelliti e poi utilizza algoritmi matematici per calcolare la posizione precisa in termini di longitudine, latitudine e altitudine.
Il *GPS* è uno standard aperto disponibile al pubblico dominio e la sua architettura è trasparente e purtroppo segnali distorti provenienti da attività malevole possono essere facilmente generati per infastidire e bloccare un dispositivo *GPS*; inoltre segnali *GPS* codificati vengono utilizzati solo in un insieme limitato di sistemi, come i sistemi *GPS* per i militari.

Tipologie di attacchi ai sistemi GPS

Il funzionamento affidabile e sicuro del sensore *GPS* è un fattore cruciale per il corretto funzionamento dei veicoli autonomi, nonché per l'implementazione di reti veicolari ad-hoc (*VANET*). Tuttavia, il *GPS* è suscettibile ad attacchi come jamming e spoofing; tali attacchi di disturbo possono bloccare completamente il funzionamento del *GPS* attraverso la trasmissione di segnali disturbanti sulle stesse frequenze dei segnali *GPS*: d'altro canto, un attacco di spoofing *GPS* inganna l'utente attraverso la trasmissione di segnali che hanno le stesse caratteristiche di quelli legittimi dei segnali satellitari *GPS*.
Gli spoofer utilizzati per gli attacchi di spoofing, come prima cosa si sincronizzano con i segnali *GPS* inviati dai satelliti, estraggono la posizione, l'ora e le effemeridi (cordinate spaziali) satellitari ed infine, conoscendo il vettore di puntamento 3D dal satellite verso l'antenna ricevente target, generano il segnale di spoofing.
Tutto ciò è possibile perché i segnali *GPS* non sono autenticati e sui ricevitori standard la coerenza del contenuto dei dati nei segnali ricevuti non viene controllata: con un dispositivo hardware appropriato, chiunque può trasmettere un segnale *GPS* non autentico, in questo modo l'aggressore può anche modificare la posizione dichiarata dei satelliti.
Ovviamente questi tipo di attacco può essere effettuato solo se il dispositivo di spoofing si trova nelle immediate vicinanze della vittima, altrimenti non sarebbe possibile interferire con il segnale originale ed evitare che il ricevitore *GPS* lo riceva.
Il jamming invece è il processo di attacco che utilizza dispositivi di trasmissione per bloccare o interrompere le comunicazioni tra i dispositivi *GPS*: questo tipo di attacco viene solitamente effettuato generando segnali con una potenza maggiore di quella del segnale originale, in questo modo i ricevitori *GPS* non possono riconoscere il segnale originale e non riescono ad associare al segnale ricevuto un set di dati autentici.
Per condurre questi tipi di attacchi, negli ultimi anni sono state sempre più utilizzati dispositivi come le Software Defined Radio (*SDR*), ovvero sistemi di ricetrasmissione radio fortemente

personalizzabili e basati su appositi software in grado di instaurare canali di comunicazione radio su una vasta gamma di frequenze, tra le quali anche le bande utilizzate dai satelliti GPS[10].

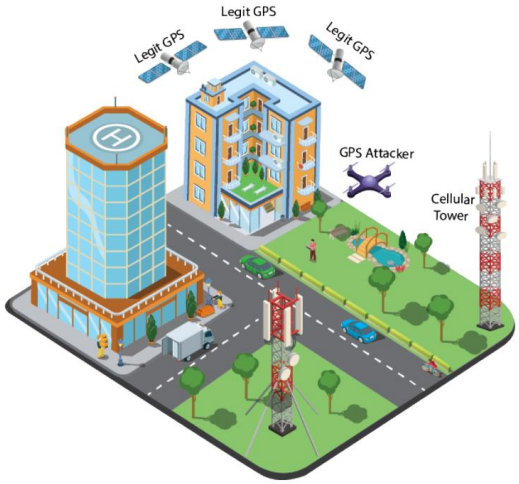


Figura 19. Tipico scenario di un attacco di spoofing ad un sistema GPS

Contromisura proposta per attacchi di jamming e spoofing ai sistemi GPS

Gli attori principali nello scenario di un attacco di spoofing *GPS* sono: l'auto a guida autonoma, l'aggressore, l'infrastruttura satellitare *GPS* e un'infrastruttura di rete wireless.
Il modello di sistema considerato nel paper include un'auto che si muove sulla rete stradale e la sua posizione e velocità reali sono, rispettivamente, $p_k = [x_k, y_k]$ e $u_k = [\dot{x}_k, \dot{y}_k]$, dove x_k e y_k sono le coordinate della posizione nell'istante temporale *k*.
L'auto è dotata di un ricevitore *GPS* che elabora i segnali di posizionamento satellitare e fornisce in uscita la posizione *GPS* $p_k^G = [x_k^G, y_k^G]$ dell'auto stessa; tale posizione viene modellata come una variabile gaussiana.
Un utente malintenzionato può utilizzare apparecchiature *COTS* (Commercial Off-The-Shelf), inclusi hardware, amplificatore e antenna Software Defined Radio (*SDR*) per interferire con i segnali *GPS* legittimi e disturbare i dati *GPS* originali.
L'auto è inoltre dotata di un dispositivo specializzato che monitora i segnali provenienti dall'infrastruttura di rete wireless circostante e dagli altri veicoli, indipendentemente dalle misurazioni *GPS*; questo dispositivo implementa un algoritmo di localizzazione che stima la posizione attuale $p_k^L = [x_k^L, y_k^L]$ dell'auto sulla base di questi segnali e viene anch'essa modellata come variabile casuale gaussiana.
Nella fase di Previsione, le letture dei sensori di bordo raccolte tramite l'*OBU* (On Board Unit-dispositivo elettronico integrato, utilizzato dalle automobili per la connettività wireless verso l'esterno dell'abitacolo) vengono utilizzate per prevedere la posizione $\hat{p}_{k+1} = [\hat{x}_{k+1}, \hat{y}_{k+1}]$ dell'auto al tempo $k + 1$, data la posizione precedentemente memorizzata $p_k = [x_k, y_k]$.
Nella fase di Aggiornamento, le misure di localizzazione senza *GPS* p_{k+1}^L vengono utilizzate per aggiornare la previsione e produrre la stima della posizione raffinata p'_{k+1}.
Infine, nella fase di Attack Detection, la posizione *GPS* p_{k+1}^G,fornita dal ricevitore *GPS* del veicolo viene confrontata con p'_{k+1}; nel caso in cui lo scostamento superi una soglia T_d prestabilita, viene attivato un allarme che segnala il rilevamento di un attacco[10].

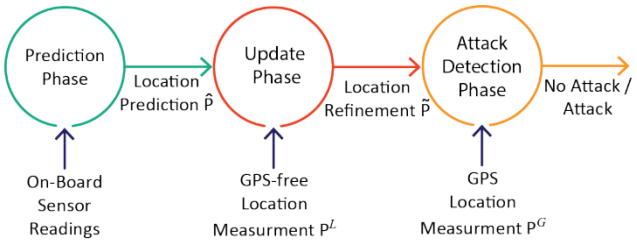


Figura 20. Flusso di dati del framework proposto nel paper, per il rilevamento di attacchi a sistemi GPS

Algorithm 1: GPS Location Spoofing Detection.

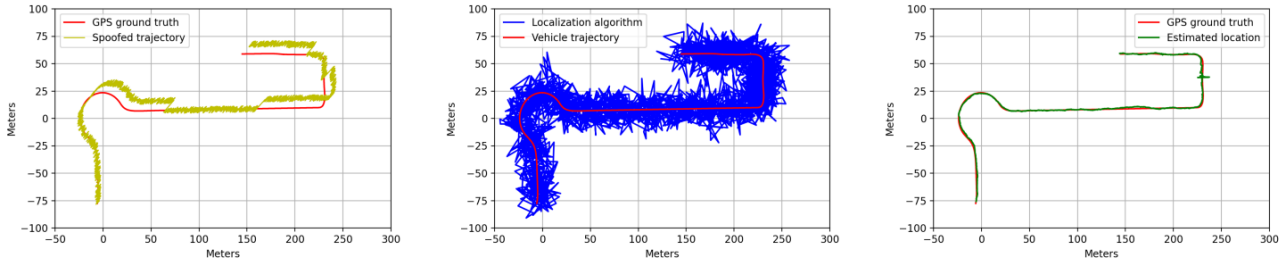
input : Previous location estimate $[\tilde{\mathbf{p}}_k, \mathbf{\Sigma}_k^{\tilde{p}}]$, CAV's sensory data $(\alpha, \dot{\varphi}, v)$, radio signal data, GPS location $[\mathbf{p}_{k+1}^G, \mathbf{\Sigma}_{k+1}^G]$, window size w, threshold T_d

output: GPS spoofing attack detection

$[\hat{\mathbf{p}}_{k+1}, \mathbf{\Sigma}_{k+1}^{\hat{p}}] \leftarrow$ *EKF_predict*$(\tilde{\mathbf{p}}_k, \alpha, \dot{\varphi}, v)$;
$[\mathbf{p}_{k+1}^L, \mathbf{\Sigma}_{k+1}^L] \leftarrow$ *LA(radio signal data)*;
$[\tilde{\mathbf{p}}_{k+1}, \mathbf{\Sigma}_{k+1}^{\tilde{p}}] \leftarrow$ *EKF_update*$([\hat{\mathbf{p}}_{k+1}, \mathbf{\Sigma}_{k+1}^{\hat{p}}], [\mathbf{p}_{k+1}^L, \mathbf{\Sigma}_{k+1}^L])$;
$d_{k+1}^{\{E,B\}} \leftarrow$ *distance*$([\mathbf{p}_{k+1}^G, \mathbf{\Sigma}_{k+1}^G], [\tilde{\mathbf{p}}_{k+1}, \mathbf{\Sigma}_{k+1}^{\tilde{p}}])$;
$\bar{d}_{k+1}^{\{E,B\}} \leftarrow$ *filter*$([d_{k-w+2}^{\{E,B\}}, \ldots, d_{k+1}^{\{E,B\}}])$;
if $\bar{d}_{k+1}^{\{E,B\}} > T_d^{\{E,B\}}$ **then**
  GPS location spoofing attack detected;

Figura 21. Algoritmo di localizzazione

Questo modello di intercettazione di attacchi di spoofing, è stato poi simulato ed i risultati ottenuti sono stati sintetizzati nel paper stesso grazie a i seguenti 3 grafici:



(a) (b) (c)

Figura 22. Risultati della simulazione del framework di mitigazione per gli attacchi di spoofing GPS

- (a) Rappresentazione del percorso reale del veicolo (in rosso) e del percorso falsificato (in giallo)
- (b) Rappresentazione del percorso reale del veicolo (in rosso) e delle traiettorie calcolate con l'algoritmo (in blu)
- (c) Rappresentazione del percorso reale del veicolo (in rosso) e del percorso stimato grazie alla combinazione dei dati raccolti dall'OBU e l'algoritmo

Come si può evincere dai grafici, nonostante il percorso reale del veicolo sia stato falsificato per quasi il 50% del totale, l'algoritmo di localizzazione ha permesso di sitimare il percorso corretto effettuato dal veicolo con precisione elevata.

Analisi critica sulla metodologia di mitigazione proposta e possibili altre misure di protezione

Il metodo di "attack detection" proposto nel paper per i sistemi *GPS*, è sicuramente uno dei tanti approcci possibili per renderne sicuro il suo utilizzo: utilizzare sia i segnali *GPS* che le misure provenienti dall'infrastruttura di rete wireless e da altri veicoli, rappresenta una strategia robusta e l'algoritmo di localizzazione basato su questi segnali può rilevare facilmente eventuali discrepanze e contribuire a migliorarne la sicurezza.
D'altrocanto, l'implementazione di un sistema di monitoraggio continuo e l'uso di algoritmi di localizzazione possono richiedere una notevole potenza di calcolo, specialmente considerando le risorse disponibili su veicoli a guida autonoma.
Inoltre l'efficacia dell'algoritmo di localizzazione dipende molto dalla disponibilità e dalla qualità dei segnali wireless che riceve dalle infrastrutture esterne al veicolo, quindi in ambienti con copertura wireless limitata o con interferenze, l'accuratezza della localizzazione potrebbe diminuire; considerando, per di più, che il dispositivo specializzato che monitora questi segnali potrebbe essere soggetto a sua volta da attacchi mirati, rende il tutto ancora più difficile da attuare.
Un altro paper invece, ritiene che sia ampiamente chiaro che la contromisura più forte e forse l'unica davvero efficace contro lo spoofing, è l'uso della crittografia a livello militare; c'è da considerare però che spesso impiega algoritmi complessi e robusti che sono progettati, in generale, per resistere a varie forme di attacchi crittografici.
Altre possibili contromisure, possono essere:

- Implementazione di firmware sicuri.
- Utilizzo di "filtri anti-spoofing": grazie al loro utilizzo, potrebbe essere essere possibile individuare le sorgenti dei vari segnali satellitari.
- Sfruttare la presenza di altri veicoli nelle vicinanze per poter comparare la posizione rilevata dal proprio ricevitore *GPS* con quella degli altri veicoli.

Attacchi alle comunicazioni V2X

Le tecnologie di comunicazione nelle automobili collegano tra loro vari elementi come veicoli, pedoni, infrastrutture, strade, piattaforme di servizi di cloud computing, ecc. e ciò ha dato origine al concetto di comunicazione *V2X* (vehicle-to-everything).
Lo scopo di questo tipo di comunicazioni, è quello di migliorare la sicurezza e l'efficienza del traffico veicolare attraverso lo scambio di informazioni in tempo reale; questo scambio di informazioni può essere utilizzato per prevenire incidenti, ottimizzare il flusso del traffico e migliorare la gestione delle infrastrutture stradali.
In un documento che tratta proprio gli attacchi ai sistemi di comunicazione V2X, viene descritta in generale l'idea ed il funzionamento alla base delle comunicazioni V2X: l'architettura interna di un veicolo è interconnessa con le *ECU* (unità di controllo elettroniche – piattaforme embedded che monitorano/controllano i vari sistemi automobilistici) le quali sono "accoppiate" ciascuna con sensori e attuatori.
La comunicazione tra il veicolo e il mondo esterno, come altri veicoli o le unità stradali (*RSU*-Road Side Units), avviene invece tramite interfacce esterne; queste sono collegate all'unità di bordo (*OBU*-On Board Unit), ovvero una *ECU* che fornisce connettività wireless.
Dunque ogni unità di controllo del veicolo si coordina con l'*OBU* per raccogliere e diffondere i dati del veicolo; il tutto è ben rappresentato nella seguente figura:

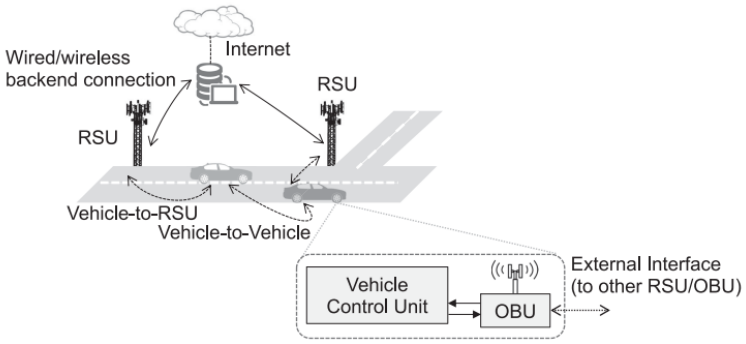


Figura 23. Un'illustrazione della comunicazione V2X

Le tecnologie applicate per le comunicazioni *V2X*, sono varie: Wi-Fi, reti cellulari 4G e 5G, ma anche tecnologie di sensoristica e radar; è implicito dunque che tutte queste categorie possano essere bersagli di attacchi malevoli da parte di terzi.
Gli attacchi ai sistemi di comunicazione veicolare possono causare per esempio perdita di dati, guasti di componenti dei veicoli e anche danni all'ambiente o alle infrastrutture; è essenziale dunque attuare delle misure di mitigazione che possano prevenire degli attacchi di questo tipo[11].

MAC Header	Basic Network Header	Security Header (certificate info, generation time/location, *etc.*)	Secured Network Header (id, timestamp, longitude, latitude, speed, heading, *etc.*)	Secure Transport Header	Secure Transport Header	Payload (BSM, CAM, DENM) • Message id • Generation time • Station id • Position (longitude, latitude, elevation, heading, *etc.*) • ….	Security Trailer (signature)	MAC check sequence

Figura 24. Schema di alto livello del formato dei pacchetti utilizzati nelle comunicazioni V2X

Esempi di attacchi alle comunicazioni V2X

Nello stesso paper, gli autori categorizzano i vari attacchi ai sistemi *V2X* e li sudduvidono in *attivi* e *passivi*: nel caso di attacchi attivi, l'avversario interagisce attivamente con il sistema mentre gli aggressori passivi potrebbero intercettare dati critici (come chiavi private, certificati, informazioni sui sensori, ecc.) senza interagire direttamente con il sistema e/o interrompere il suo normale comportamento.
Esempi di attacchi *attivi* possono essere:

- Attacchi di Denial of Services: gli attacchi *DoS* possono verificarsi in diversi livelli della rete, e consistono in un avversario malevolo che invia più richieste di quante il sistema possa gestire; questo può causare la perdita o l'impossibilità di inviare pacchetti.
  Questo scenario è chiaramente catastrofico per le applicazioni legate alla sicurezza: ad esempio, un veicolo coinvolto in un incidente stradale dovrebbe propagare messaggi di avviso, ma ad altri veicoli potrebbe essere impedito di ricevere questi messaggi di avviso da un utente malintenzionato che perde intenzionalmente i pacchetti.
- Attacchi Sybil: si tratta di un noto attacco dannoso nelle reti veicolari wireless in cui un veicolo finge di avere più di un'identità contemporaneamente.
  Ad esempio, se un veicolo malevolo cambia più volte identità, può utilizzare più pseudonimi per apparire come un veicolo diverso in movimento o far sembrare che la strada sia

congestionata (anche se non lo è) e inviare informazioni errate sulle condizioni stradali ai veicoli/*RSU* vicini/e.

- Iniezione di dati falsi: un veicolo non autorizzato potrebbe generare falsi messaggi di traffico/sicurezza o informazioni errate sulla stima del traffico (che differiscono dalle informazioni del mondo reale) e trasmetterli alla rete con l'intenzione di interrompere il traffico stradale o innescare una collisione.
  Oppure attraverso lo spoofing *GPS*, un utente malintenzionato potrebbe iniettare false informazioni sulla posizione utilizzando simulatori *GPS*[11].

Contromisure proposte per attacchi alle comunicazioni V2X

Per proteggere le comunicazioni *V2X* (ad esempio, per garantire l'integrità e l'autenticità dei messaggi), gli autori propongono un approccio che consiste nell'utilizzare la crittografia asimmetrica che sfrutta un'infrastruttura a chiave pubblica (*PKI*) per la gestione delle credenziali di sicurezza.
La *PKI* consente lo scambio sicuro di messaggi sulla rete: ad ogni veicolo viene fornita una coppia di chiavi asimmetriche ed un certificato che contiene la chiave pubblica con attributi specifici *V2X* (come l'ID) ed è firmato dall'autorità emittente della chiave.
La *PKI* include i seguenti elementi chiave: (a) una parte fidata, ad esempio un'autorità di certificazione (*RCA*- Root Ceritificate Authority), che fornisce servizi per autenticare l'identità delle entità, (b) un'autorità di registrazione certificata da una *RCA* che emette certificati per usi specifici consentiti dalla *RCA*, (c) un database che archivia le richieste di certificato e che emette/revoca certificati e (d) un archivio di certificati a bordo del veicolo – per salvare i certificati emessi e le chiavi private.
Il tutto può essere sintetizzato e ben rappresentato nella seguente figura:

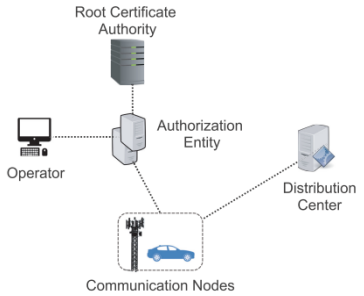


Figura 25. Rappresentazione del funzionamento della crittografia asimmetrica nell'ambito automotive

Per i casi specifici di attacchi descritti precedentemente, il paper propone anche altri metodi di mitigazione, ognuno dei quali si focalizza sulla tipologia di attacco considerato.
Dato che il routing nelle comunicazioni *V2X* è prevedibile e standardizzato, gli autori affermano che gli attacchi *DoS* a livello di rete possono essere rilevati da meccanismi di watchdog in cui ciascun veicolo utilizza il concetto di "fiducia del vicino" (determinato dal rapporto tra i pacchetti inviati al vicino e i pacchetti che vengono inoltrati dal vicino): se un veicolo rilascia ripetutamente pacchetti (fino al superamento di una soglia di tolleranza), il veicolo viene considerato dannoso.
Per potersi difendere, invece, da attacchi di Syblin, è stato pensato il seguente approccio: prima di inviare qualsiasi messaggio, un veicolo ottiene innanzitutto un timestamp per il messaggio da una *RSU* vicina, poi se un veicolo riceve serie di messaggi con timestamp uguali/simili dallo stesso nodo (veicolo o *RSU*) per un certo periodo di tempo, allora il veicolo mittente viene considerato come nodo Sybil.

Un’approccio più generale al problema della sicurezza delle comunicazioni *V2X*, propone invece un meccanismo di filtraggio dei messaggi che combina i parametri dei messaggi stessi in un'unica entità chiamata "certezza dell'evento" (*CoE*-Certainty Of Event); questa rappresenta il livello di confidenza di un messaggio ricevuto e viene calcolato combinando i dati provenienti da varie fonti come sensori locali, *RSU* e utilizzando meccanismi di consenso (ad esempio, messaggi da altri veicoli).
Un meccanismo simile discusso nel paper, si basa sul “consenso”: ciascun veicolo raccoglie segnalazioni sullo stesso evento dai veicoli vicini fino a quando non viene superata una certa soglia di segnalazioni di supporto (dopo di che il messaggio è considerato affidabile).
Il metodo proposto consente al sistema di prendere una decisione entro un tempo di attesa limitato e quindi adatto per applicazioni critiche in termini di tempo/sicurezza (ad esempio, la decisione se fidarsi di un avviso di incidente stradale, il quale deve essere emesso in anticipo in modo che il veicolo possa rallentare o cambiare corsia di conseguenza).
Infine gli autori hanno anche proposto tecniche statistiche per prevedere le tendenze nel flusso di traffico e determinare se un mittente è dannoso o meno: ciascun veicolo V_i stima il proprio parametro di flusso F_i (che dovrebbe essere simile per i veicoli situati vicino a V_i) utilizzando un modello che utilizza la densità dei veicoli e la velocità media degli altri veicoli nelle sue vicinanze.
I veicoli si scambiano i propri parametri di flusso, valori di densità, velocità e informazioni sulla posizione e per ogni messaggio ricevuto, i veicoli confrontano la media dei parametri ricevuti con quelli da loro calcolati: se la differenza è inferiore ad una soglia prestabilita allora il messaggio viene accettato; in caso contrario, viene monitorato il comportamento del mittente ed eventualmente verrà segnalato ad altri veicoli e isolato dalla rete[11].

Analisi critica sulle metodologie di mitigazione proposte e possibili altre misure di protezione

Le comunicazioni *V2X* sono molto eterogenee, hanno a che fare con molti tipi di dati (anche molto diversi tra di loro per natura) e di entità e le vulnerabilità che possono essere sfruttate dagli attaccanti sono purtroppo varie; è dunque improbabile pensare di rendere questo tipo di comunicazione sicuro al 100%, ma sicuramente certi approcci e certe tecniche possono aiutare molto a mitigare la maggior parte degli attacchi.
Le metodologie proposte nel paper sono varie e molto valide: alcune sono state pensate con lo scopo di prevenire certi tipi di attacchi specifici, mentre altre cercano di rendere le comunicazioni *V2X* più sicure e monitorate a livello generale; ovviamente quelli presentati nel paper sono solo alcuni attacchi/contromisure che riguardano da vicino questo tipo di comunicazioni, ed è ovvio che la sicurezza in questo contesto ha continua necessità di essere migliorata.
L'utilizzo della crittografia asimmetrica basata su un'infrastruttura a chiave pubblica (*PKI*), è una pratica comune e generalmente robusta per garantire l'integrità e l'autenticità dei messaggi nelle reti veicolo-veicolo e veicolo-infrastruttura: ogni veicolo riceve una coppia di chiavi asimmetriche e un certificato che consentono la verifica dell'identità del mittente dei messaggi, viene garantita l'integrità dei messaggi scambiati tra i veicoli e con l'infrastruttura ed è possibile proteggere il contenuto dei messaggi da osservatori indesiderati.
D’altro canto però la gestione di una *PKI* su larga scala può essere complessa, richiede risorse significative e la scalabilità può diventare un problema quando il numero di veicoli e partecipanti alla rete *V2X* aumenta considerevolmente; in più la sicurezza del sistema dipende dalla protezione delle chiavi private dei veicoli ed una cattiva gestione o la compromissione delle chiavi private potrebbe compromettere la sicurezza dell'intero sistema.

Per quanto riguarda invece l'approccio basato sulla "fiducia del vicino", sembra essere abbastanza semplice da implementare, può adattarsi a variazioni nella topologia di rete e alle condizioni operative senza richiedere complessi algoritmi di rilevamento, però c'è la possibilità che si generino di falsi positivi causati da (per esempio) pacchetti persi.
L'utilizzo dei timestamp all'interno dei messaggi invece, ha un grosso limite: due veicoli provenienti da direzioni opposte potrebbero essere erroneamente contrassegnati come nodi Sybil poiché verranno associati a timestamp simili per un breve periodo di tempo; nonostante ciò, questo approccio è economico poiché non utilizza costose infrastrutture computazionali a chiave pubblica (PKI) o RSU accessibili da Internet.
Un approccio alternativo potrebbe basarsi sul presupposto che tutte le entità fisiche hanno risorse limitate e si possono sfruttare quindi "programmi" computazionali per testare le risorse a disposizione di ciascun veicolo: quindi se un utente malintenzionato impersona entità diverse contemporaneamente, avrà un overhead di computazioni da dover svolgere talmente alto da rendere impossibile risolverlo e verrà rilevato l'attacco.
Il metodo CoE, il meccanismo basato sul "consenso" e quello che sfrutta le tecniche statistiche, risultano essere molto flessibili in un ambiente dinamico come quello stradale in cui le condizioni variano in maniera rapida e non sempre predicibile; inoltre possono contribuire a rendere il sistema più resistente agli attacchi mirati, in quanto un singolo veicolo dannoso avrebbe una limitata influenza sulla decisione complessiva.

Attacchi alle reti VANET

Le reti ad-hoc veicolari (*VANET*) sono una tecnica di comunicazione emergente per facilitare la sicurezza del traffico e la sua ottimizzazione: i veicoli condividono le informazioni dalle unità di bordo (*OBU*) installate in ogni veicolo e ne ricevono dalle unità a bordo strada (*RSU*) installate lungo la strada.
Sulla base di tutte le informazioni fornite dalla rete di comunicazione, i veicoli prendono decisioni sul miglior percorso da prendere nella rete stradale.
Dall'altro lato, l'intera rete di comunicazione *VANET* è esposta ad un ambiente ad accesso aperto, che la rende molto vulnerabile; difatto recentemente sono emerse molte forme di attacchi contro *VANET* che hanno allarmato la situazione sulla sicurezza di queste reti.
Essendo un'implementazione della Mobile Ad hoc NETworks (*MANET*), le *VANET* ereditano tutte le vulnerabilità di sicurezza e privacy scoperte e non scoperte relative ai *MANET*; inoltre, le *VANET* hanno una serie di proprietà distintive che potrebbero rappresentare altre vulnerabilità sfruttabili.
Le connessioni in una *VANET* in particolare (e in qualsiasi rete Wireless Ad hoc in generale) si basano su comunicazioni da nodo a nodo: ogni nodo è in grado di agire sia come host che richiede dati sia come router che inoltra dati; esistono due tipi di nodi: le RoadSide Unit (*RSU*) e le OnBoard Unit (*OBU*).
Inoltre la rete stradale è di sua natura molto dinamica, dunque ci si aspetta che i nodi siano continuamente "in contatto" tra loro per mantenere la sopravvivenza della rete; questo aspetto delle *VANET* sembra essere molto vulnerabile[12].

Esempi di attacchi alle reti VANET

In un documento che tratta le vulnerabilità nelle reti VANET, viene proposto uno scenario di attacco ad una rete *VANET* con annesse conseguenze alle prestazioni della rete stradale.
Quest'ultima viene rappresentata nel documento come un grafo diretto *G(V,E)* costituito dall'insieme delle intersezioni *V* e dall'insieme dei collegamenti *E* che collegano direttamente le intersezioni; poiché i veicoli hanno bisogno solo dello stato della strada (cioè, bloccato o sbloccato) per scegliere il percorso, viene definito lo stato della strada s_e del collegamento *e* in una forma discreta dove $s_e = 1$ denota che il collegamento *e* è congestionato e $s_e = 0$ denota che il collegamento *e* è sbloccato.
Sebbene la congestione sia parte di un continuum di volume di traffico, lo stato di un collegamento può essere semplificato in una forma discreta in cui se la densità stradale supera una certa soglia, lo stato del collegamento è considerato bloccato; la densità stradale è definita come il rapporto tra il numero di veicoli sulla strada e la lunghezza della strada e la capacità di ciascun collegamento sono limitate.
Vengono poi introdotti due tipi diversi di attacchi;

- Fake congestion pollution: lo stato stradale reale del collegamento *e* è sbloccato ma un attaccante lo fa percepire come bloccato; intuitivamente, la falsa congestione influisce sulle prestazioni del sistema sotto due aspetti: uno è il degrado diretto delle prestazioni su tali strade con congestione falsa, e l'altro è la congestione stradale che si è verificata nei percorsi alternativi delle strade attaccate.
- Unreported congestion pollution: accade quando lo stato di un collegamento bloccato viene percepito, a casusa di un attacco, sbloccato; la congestione non segnalata influisce sulle prestazioni del sistema sotto due aspetti: uno è il degrado diretto delle prestazioni su tali strade congestionate, e l'altro è la cascata di congestione nelle strade vicine a tali strade non segnalate.

Un esempio per ogni tipo di attacco, può essere rappresentato nella seguente figura:

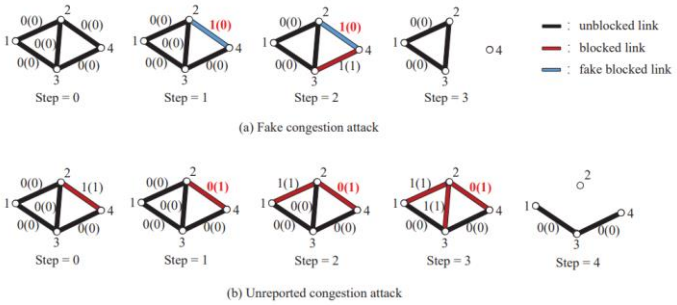


Figura 26. Esempi di degrado delle prestazioni causato dai due tipi di attacchi di congestion pollution

Viene inoltre definito un indicatore di prestazione per valutare la vulnerabilità del sistema in caso di attacchi di inquinamento delle informazioni. L'efficienza *P* del sistema è calcolata come:

$$P = \frac{\eta}{\eta_0}$$

$$\eta = \sum_{o \neq d \in V} \psi_{od}$$

dove ψ_{od} indica la "domanda" da un nodo di origine *o* a un nodo di destinazione *d*, l'efficienza η è definita come la somma dei ψ_{od}, η_0 denota l'efficienza iniziale della rete, quando non si sono verificati attacchi e *P* indica la capacità del sistema di mantenere le proprie prestazioni di trasporto pur subendo attacchi: una *P* alta riflette l'invulnerabilità della rete.
Tutte e due le tipologie di attacchi vengono dunque simulate, utilizzando come reti stradali di riferimento una rete a griglia ed una eterogenea come quelle di seguito rappresentate:

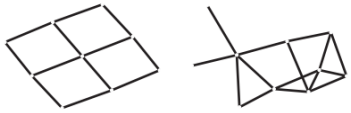


Figura 27. Rete a griglia e rete eterogenea

I risultati ottenuti sono i seguenti:

- Gli attacchi di *unreported congestion pollution* sono più pericolosi nella rete a griglia: le prestazioni del sistema diminuiscono della metà quando viene attaccato il 30% dei collegamenti in un attacco di unreported congestion pollution, ma un degrado di prestazionale equivalente deve attaccare il 50% dei collegamenti in un fake congestion pollution, il che indica che la rete a griglia è più robusta sotto attacchi di fake congestion pollution.
- Gli attacchi di *fake congestion pollution*, sono più minacciosi in una rete eterogenea: le prestazioni del sistema diminuiscono della metà quando viene attaccato il 15% dei bordi in un attacco di fake congestion pollution, ma un uguale degrado delle prestazioni deve attaccare il 25% dei bordi in un attacco di congestione non segnalato, il che indica sia la maggiore vulnerabilità della rete eterogenea sotto attacchi di fake congestion pollution sia la maggior pericolosità (in generale) degli attacchi a reti di tipo eterogeneo.
  Difatti la centralità di alcuni collegamenti e l'eventuale attacco ad uno di questi, degradano in maniera pesante la capacità di trasporto del sistema[12].

Chiaramente nel paper appena sintetizzato, vengono solo discusse le conseguenze sulle prestazioni della rete stradale in caso di attacchi di congestion pollution; di seguito invece verranno trattati più nello specifico le tecniche sfruttate dagli attaccanti per compiere queste tipologie di attacchi
Essendo le reti *VANET* e le comunicazioni *V2X* molto legate tra loro, è chiaro che molti attacchi sono i medesimi, altri invece risultano essere specifici delle reti *VANET* in quanto coinvolgono dinamiche di interazione veicolare e caratteristiche specifiche della comunicazione per la gestione del traffico.
Alcune delle tipologie di attacchi citate in un altro paper e che possono intaccare la sicurezza nelle reti *VANET,* sono le seguenti:

- Attacchi di impersonification and masquerade: nelle *VANET*, un host è identificato in modo univoco dall'indirizzo *IP* e *MAC*, ma non sono sufficienti per autenticare i veicoli; difatto un aggressore può utilizzare lo spoofing *MAC* e *IP* per ottenere l'identità di altri nodi nella rete.
  Quindi se non esiste un processo di autenticazione per rendere la rete sicura da nodi dannosi, un veicolo dannoso può inviare messaggi per conto di altri veicoli per creare caos, ingorghi o incidenti.
- Attacchi di timing: in generale le reti *VANET* sono diventate molto importanti per la sicurezza delle strade, tuttavia utilizzano applicazioni critiche in termini di tempo e richiedono la trasmissione di dati da un veicolo a un altro al momento giusto; dunque negli attacchi temporali, quando un veicolo malevolo riceve un messaggio, non lo inoltra normalmente al resto della rete ma con un delay aggiuntivo per creare ritardo.

Pertanto, i veicoli vicini agli aggressori ricevono il messaggio dopo che ne hanno effettivamente bisogno.

- Illusion attacks: negli attacchi di illusione, l'avversario "inganna" intenzionalmente i sensori della sua auto per produrre letture errate dei sensori e quindi informazioni errate sul traffico: di conseguenza, vengono trasmessi ai vicini messaggi di avviso errati sul traffico.
  In generale, il comportamento degli automobilisti dipenderà dai messaggi di avviso sul traffico che hanno ricevuto e dunque a causa di questi attacchi di illusioni, molto probabilmente cambieranno di conseguenza il loro comportamento di guida.
  Per di più, l'aggressore può causare incidenti, ingorghi e ridurre le prestazioni manipolando in modo invisibile la topologia della rete.
- Black hole attacks: un buco nero è un'area in cui viene reindirizzato il traffico di rete e dove in realtà o non c'è nessun nodo in quell'area oppure i nodi che risiedono in quell'area non partecipano in maniera collaborativa con il resto della rete.
  In un attacco black hole, un nodo dannoso si presenta al resto della rete come quello più vicino da cui recuperare varie informazioni e quindi inganna il protocollo di routing; di conseguenza, il nodo attaccante ottiene il diritto di rispondere alle richiesta di instradamento e quindi è in grado di intercettare i pacchetti di dati e decidere cosa farne; una volta stabilito con successo il percorso contraffatto, decide il nodo dannoso se e dove rilasciare o inoltrare i pacchetti[13].

Contromisure proposte per attacchi alle reti VANET

Nello stesso paper vengono quindi proposte delle metodologie che possano mitigare i tipi di attacchi più comuni alle reti *VANET*; come nel caso delle comunicazioni *V2X*, le vulnerabilità sono varie e non è semplice trovare delle tecniche che possano rendere queste reti completamente sicure e protette da tutte le tipologie di attacchi.
Un primo approccio proposto per mitigare gli attacchi di impersonification e masquerade, è quello che utilizza, ancora una volta, l'infrastruttura a chiave pubblica (*PKI*) di cui è già stato fornito in precedenza il funzionamento di base: in questo modo i messaggi proveniente dalle varie *OBU* vengono sottoposti a verifica di integrità e autenticità prima di poter essere considerati attendibili.
Per evitare gli attacchi temporali invece, viene proposto il *TPM* (Trusted Platform Module): è un componente hardware di che fornisce funzionalità di sicurezza avanzate per la protezione delle informazioni e la gestione delle chiavi crittografiche su un sistema informatico.
Il *TPM* può misurare l'integrità di un sistema eseguendo una serie di misurazioni (hash) sul firmware, sull'avvio del sistema operativo e su altri componenti critici; queste misurazioni possono essere utilizzate per verificare se il sistema è stato compromesso o è stato soggetto a modifiche non autorizzate, come nel caso degli attacchi temporali sopra descritti.
L'autenticazione tradizionale dei messaggi e la verifica dell'integrità dei messaggi non possono difendersi però totalmente dagli attacchi di illusioni, perché l'avversario manipola e confonde direttamente i sensori di un veicolo per riportare informazioni false.
Plausibility Validation Network (*PVN*) è un modello di sicurezza per proteggere le *VANET* dagli attacchi di questo tipo: il *PVN* raccoglie i dati grezzi dei sensori e verifica se i dati raccolti sono plausibili o meno.
Vengono presi in considerazione due tipi di input:

- Dati in ingresso dalle *RSU*
- Dati raccolti dai sensori

ed il *PVN* sfrutta una serie di regole e un modulo di controllo dei dati, che aiuta a verificare la validità dei dati di input e ad intraprendere le azioni necessarie di conseguenza; un messaggio è considerato affidabile se supera tutte le verifiche e in caso contrario, verrà dichiarato messaggio non valido e scartato automaticamente.
Le soluzioni esistenti agli attacchi black hole, considerano l'utilizzo di protocolli in cui vi è più di un percorso per i pacchetti dalla sorgente alla destinazione; un'altra soluzione consiste nell'utilizzare i numeri di sequenza dei pacchetti nell'intestazione degli stessi in modo che, se un pacchetto viene perso, la destinazione possa semplicemente identificarlo dal numero di sequenza del pacchetto mancante.
Infine nel paper, viene anche fatta la seguente considerazione: la privacy, a causa di alcuni degli attacchi presenti nelle reti *VANET,* può venire intaccata quando abbiamo a che fare con questo tipo di attacchi. Sarebbe quindi doveroso impedire la divulgazione dei propri percorsi di guida o della posizione in tempo reale, i quali possono essere facilmente individuati tenendo traccia dei messaggi inviati dalla propria OBU.
Pertanto è necessario un protocollo di comunicazione anonimo (simile al conosciuto sistema *TOR*), che allo stesso tempo consenta di poter identificare con precisione l'identità di un veicolo quando necessario: ad esempio, un conducente malevolo non deve essere in grado di "fuggire" utilizzando un'identità anonima dopo aver inviato messaggi falsi e causato un incidente[13].

Analisi critica sulle metodologie di mitigazione proposte e possibili altre misure di protezione

Tutte le metodologie proposte nel paper sembrano essere molto valide ed efficaci per rendere le reti *VANET* più sicure; l'utilizzo delle infrastrutture a chiave pubblica è una pratica comune e sicuramente efficiente contro molte tipologie di attacchi che riguardano le reti *VANET* (come d'altronde era già stato effermato in precedenza), le altre invece sembrano essere definite ad hoc per determinati tipi di attacchi.
Purtroppo è impensabile che un auto implementi contemporaneamente un numero così elevato di meccanismi/protocolli, ma nonostante ciò le idee proposte possono essere un punto di partenza da cui poter costruire col tempo delle metodologie di mitigazione sempre più efficaci e con un'efficienza computazionale minima.
Entrando nel particolare, il *TPM* sembra offrire funzionalità di sicurezza avanzate: la sua presenza a livello hardware è un vantaggio, poiché fornisce un metodo affidabile per rilevare eventuali modifiche non autorizzate e rende più difficile per gli attaccanti compromettere il sistema; un eventuale attaccante infatti dovrebbe mettere mano ai dispositivi elettronici di un auto per attuare un attacc di timing, il che risulta essere enormemente più complicato se non impossibile.
ll *PVN,* invece, rappresenta un approccio innovativo alla sicurezza nelle *VANET*, affrontando gli attacchi diretti ai sensori che possono generare informazioni illusorie; l'uso di regole e controlli per valutare la plausibilità dei dati, è una strategia proattiva e forse l'unica in grado di intercettare degli attacchi di questo tipo.
Ovviamente, come tutte le altre metodologie che richiedono l'utilizzo di infrastrutture esterne al veicolo, l'implementazione di un *PVN* può comportare una certa complessità di implementazione, specialmente quando si considerano regole complesse e gestione di dati provenienti da diverse fonti.
Considerando inoltre che non tutte le reti stradali possono essere dotate di *RSU* e che l'efficacia del modello potrebbe dipendere dalle risorse computazionali disponibili in un certo istante, sarebbe opportuno rivalutare parte della metodologia.

Più o meno le stesse considerazioni possono essere fatte anche per gli approcci proposti contro gli attacchi di black hole: sicuramente considerare più di un percorso possibile per i pacchetti da instradare e utilizzare un metodo di enumerazione degli stessi, può aiutare ad isolare i nodi considerati “buchi neri”, ma allo stesso tempo sono dei protocolli che possono richiedere un overhead eccessivo .
Infatti i nodi che fungono da router per l’instradamento, dovrebbero utilizzare algoritmi decisionali complessi, mentre i nodi di destinazione che rilevano la mancanza di alcuni pacchetti dovrebbero chiedere il ri-invio al nodo mittente: purtroppo però il contesto molto dinamico delle reti stradali e la non continua disponibilità delle risorse necessarie, non rendono possibile ciò.

CONCLUSIONI

Le industrie automobilistiche, i giganti della tecnologia e i governi di tutto il mondo stanno adottando iniziative per costruire veicoli autonomi sicuri e convenienti per commercializzarli il prima possibile. Sebbene questa sia la più grande svolta tecnologica nel campo dei trasporti e dell’esperienza di guida, ciò può essere realizzato solo rendendo i veicoli a guida autonoma sicuri e resistenti contro qualsiasi tipo di attacco informatico.
In questo documento abbiamo analizzato la natura e le caratteristiche degli attacchi ad alcuni componenti essenziali di un CAV, quali: sistema di Infotainment, GPS, Lidar, e reti VANET; abbiamo delineato le possibili contromisure per tali attacchi (di cui sono stati discussi i punti di forza e i limiti) ed abbiamo proposto delle possibili alternative.
Abbiamo cercato di evidenziare come gli attacchi alla sicurezza delle automobili siano di diversa natura e cambino in continuazione poiché le nuove applicazioni e tecnologie non sono esenti da vulnerabilità; anche le tecniche di mitigazione si evolvono di pari passo ovviamente, però risulta sempre più difficile riuscire ad integrarne sempre di più in un'unica automobile.
Dai capitoli precedenti del documento è facile evincere che i tempi di attesa per avere sulle nostre strade delle auto intelligenti, a guida autonoma e soprattutto sicure, sono ancora lunghi: d’altronde le strade sono un luogo molto pericoloso e non è ammissibile che siano possibili anche i più semplici attacchi alle auto.
Anche la più piccola vulnerabilità che può essere sfruttata da un entità malevola, ha un potenziale più o meno elevato di mettere in pericolo la privacy degli utenti ma soprattutto la vita degli stessi; proprio per questo c’è bisogno di ancora tanto tempo e lavoro prima che certe tecnologie e sistemi vengano integrate in maniera permanente nella quasi totalità dei veicoli.
Un altro aspetto da considerare, è il costo necessario per la realizzazione ed il mantenimento di tutte le infrastruttre “Side road” necessarie al corretto funzionamento di alcuni sistemi integrati nell’automotive e alla sicurezza stradale in generale: basti pensare all’elevato numero di *RSU* e *PKI* (solo per citarne alcune) che sono necessarie per rendere disponibili tutta una serie di funzionalità e sistemi di sicurezza di un auto a guida autonoma.
Per non pensare al fatto che, se per esempio per un tratto più o meno lungo queste infrastrutture non sono presenti, queste funzionalità non sono proprio disponibili e molti tipi di vulnerabilità posso essere messe allo scoperto per un determinato lasso di tempo.
Per concludere, nonostante i notevoli progressi tecnologici, è innegabile che i sistemi di sicurezza attuali siano ancora in evoluzione e possano beneficiare di ulteriori ricerche e sviluppi; inoltre la diffusione su vasta scala dei veicoli intelligenti richiede una ponderata considerazione delle sfide e delle limitazioni attuali nei sistemi di sicurezza.
In aggiunta, è essenziale promuovere la consapevolezza pubblica sull'uso responsabile dei veicoli intelligenti: l'educazione del pubblico riguardo alle funzionalità e alle limitazioni di tali veicoli

contribuirà a garantire una transizione sicura e responsabile verso una mobilità intelligente; solo attraverso un approccio consapevole possiamo garantire che la rivoluzione nell'ambito automobilistico contribuisca effettivamente a una strada più sicura per tutti gli utenti.

BIBLIOGRAFIA

[1] "AUTOMOTIVE DATA Opportunities, Monetization, and Cybersecurity Threats in the Connected Vehicle Landscape", Numaan Huq, Vladimir Kropotov, Philippe Lin, Rainer Vosseler";

[2] "Attacks on Self-Driving Cars and Their Countermeasures: A Survey", November 16 2020, ABDULLAHI CHOWDHURY , (Member, IEEE), GOUR KARMAKAR , (Member, IEEE), JOARDER KAMRUZZAMAN , (Senior Member, IEEE), ALIREZA JOLFAEI , (Senior Member, IEEE), AND RAJKUMAR DAS.";

[3] "Practical Vulnerability-Information-Sharing Architecture for Automotive Security-Risk Analysis, November 27 2020, YOUSIK LEE, SAMUEL WOO, YUNKEUN SONG, JUNGHO LEE, AND DONG HOON LEE4, (Member, IEEE)";

[4] "STRIDE-Based Cybersecurity Threat Modeling, Risk Assessment and Treatment of an Infotainment High Performance Computing (HPC) System, 3 January 2024, Popy Das, Md. Rashid Al Asif, Sohely Jahan, Rahamatullah Khondoker, Kawsar Ahmed, Francis M. Bui";

[5] "Static Analysis of Android Auto Infotainment and ODB-II Apps, 20 May 2019, Amit Kr Mandal, Federica Panarotto, Agostino Cortesi, Pietro Ferrara, Fausto Spoto";

[6] "A Security Analysis of an In-Vehicle Infotainment and App Platform, August 8-9 2016, Sahar Mazloom, Mohammad Rezaeirad, and Aaron Hunter, *George Mason University;* Damon McCoy, *New York University*";

[7] "Valet attack on privacy: a cybersecurity threat in automotive Bluetooth infotainment systems, 2022, Vishnu Renganathan, Ekim Yurtsever, Qadeer Ahmed and Aylin Yener";

[8] "Common Attacks Against Car Infotainment Systems, July 2019, Lin, Tong, Chen, Luhai";

[9] "You Can't See Me: Physical Removal Attacks on LiDAR-based Autonomous Vehicles Driving Frameworks", August 9–11, 2023, Yulong Cao, University of Michigan; S. Hrushikesh Bhupathiraju and Pirouz Naghavi, University of Florida; Takeshi Sugawara, The University of Electro-Communications; Z. Morley Mao, University of Michigan; Sara Rampazzi, University of Florida.

[10] "GPS Location Spoofing Attack Detection for Enhancing the Security of Autonomous Vehicles", Mohsin Kamal, Arnab Barua, Christian Vitale, Christos Laoudias and Georgios Ellinas.

[11] "Securing Vehicle-to-Everything (V2X) Communication Platforms" Monowar Hasan , Sibin Mohan, Takayuki Shimizu , and Hongsheng Lu.

[12] "Vulnerability Analysis of Road Network under Information Pollution Attacks in VANET" Jingjing Yang, Yuchun Guo, Yishuai Chen, Yongxiang Zhao, Naipeng Li.

[13] "SECURITY ATTACKS AND SOLUTIONS IN VEHICULAR AD HOC NETWORKS: A SURVEY", Vinh Hoa LA, Ana CAVALLI.