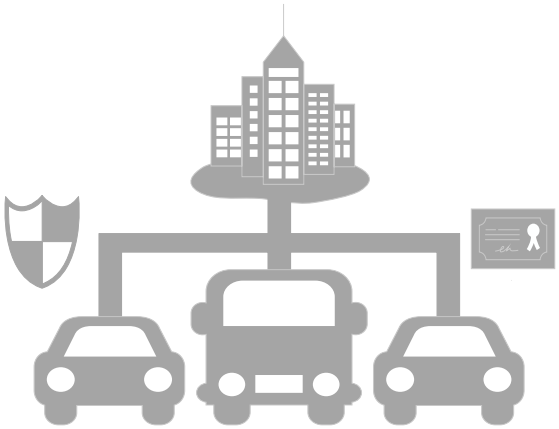
Solución de ciberseguridad aplicada al vehículo conectado



Jordi Nogués López
Trabajo de Final de Grado en Ingeniería de Telecomunicaciones
Área de Administración de redes y sistemas operativos

Consultor: Mario Prieto Vega
Profesor: David Bañeres Besora

Fecha de entrega: 12 de junio de 2023

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Solución de ciberseguridad aplicada al vehículo conectado*
Nombre del autor:	*Jordi Nogués López*
Nombre del consultor/a:	*Mario Prieto Vega*
Nombre del PRA:	*David Bañeres Besora*
Fecha de entrega (mm/aaaa):	06/2023
Titulación:	*Grado en Ingeniería de Telecomunicaciones*
Área del Trabajo Final:	*Seguridad en redes*
Idioma del trabajo:	*Castellano*
Palabras clave	*Ciberseguridad, Vehículo conectado*

Resumen del Trabajo:

Entre las fuentes estudiadas sobre tecnologías de comunicación vehicular, resuena una anécdota de un fabricante alemán, quien sostenía que los coches modernos se están convirtiendo en “smartphones con ruedas”. Es un símil acertado, pues gracias al Internet de las Cosas y a las tecnologías de comunicación, los vehículos modernos ofrecen comodidades e incluso elementos de entretenimiento que difícilmente hubiésemos imaginado hace unos años.

La problemática que motivó este proyecto es la ciberseguridad del vehículo conectado, un factor clave para que el coche autónomo llegue a ser una realidad.

Inicialmente, se analizaron los mecanismos de intercambio de información V2X (Vehicle to Everything) existentes en la actualidad para entender los riesgos asociados a las comunicaciones, puesto que éstas representan el principal vector de ataque.

También se examinó el marco regulatorio de la industria automotriz para identificar los aspectos que requieren mayor atención. De este análisis surgió el Centro de Operaciones de Seguridad Vehicular como principal foco de investigación de este proyecto.

En base a esto, se estudiaron los requisitos de arquitectura de un VSOC y, siguiendo las recomendaciones de los analistas, se seleccionó la tecnología apropiada. Después se diseñó una solución y, por último, se desarrolló un experimento para responder a un determinado tipo de ataque.

Como conclusión, este trabajo pone de relieve los desafíos de ciberseguridad que afronta la industria automotriz, y demuestra que la solución puede apoyarse en los mismos principios aplicables a las redes de ordenadores, aunque con estrategias específicas de seguridad IoT para el proceso de captura de alertas.

Abstract:

One of the funniest thoughts to highlight from the study of vehicular communication technologies is from a German manufacturer, who asserted that modern cars are becoming "smartphones on wheels". This analogy makes a lot of sense as, thanks to the Internet of Things and communication technologies, modern vehicles provide conveniences and even entertainment features that would have been hard to imagine just a few years ago.

The problem that motivated this project is the cybersecurity of the connected vehicle, a pivotal factor for level 5 autonomous cars to become a reality.

Firstly, the existing V2X (Vehicle to Everything) information exchange mechanisms were analyzed to understand the risks associated with communications, as they represent the main attack vector.

The regulatory framework of the automotive industry was also examined to identify the aspects that require the most attention. This investigation led to the Vehicle Security Operations Center being identified as the main research focus for this project.

Subsequently, the architectural requirements of a VSOC were studied and, according to the analysts' recommendations, the appropriate technology was selected. Afterwards, a solution was designed, and finally, an experiment was developed to address a specific type of attack.

In conclusion, this work underscores the cybersecurity challenges the automotive industry faces and illustrates that solutions can draw upon the principles applicable to computer networks, albeit with specific IoT security strategies tailored for the process of alert capture.

Índice

1. INTRODUCCIÓN 1

1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO 1
1.2 OBJETIVOS DEL TRABAJO 4
1.3 ENFOQUE Y MÉTODO SEGUIDO 4
1.4 PLANIFICACIÓN DEL TRABAJO 6
1.5 BREVE SUMARIO DE PRODUCTOS OBTENIDOS 7
1.6 BREVE DESCRIPCIÓN DE LOS CAPÍTULOS DE LA MEMORIA 7

2. ESTADO DEL ARTE 9

2.1. INTRODUCCIÓN 9
2.2. PROTOCOLOS DE COMUNICACIÓN V2X 10
2.2.1. IEEE DSRC y ETSI ITS-G5 10
2.2.2. 5G-V2X 13
2.3. NORMATIVAS DE SEGURIDAD 15
2.3.1. Introducción 15
2.3.2. Normativa UN ECE WP.29 y estándar ISO/SAE 21434 16
2.4. CONCLUSIONES 17

3. MODELADO DE AMENAZAS 18

3.1. ANATOMÍA DEL VEHÍCULO CONECTADO 18
3.1.1. El bus CAN 19
3.1.2. El conector OBD-II 20
3.1.3. La OBU 22
3.2. TIPOLOGÍA DE AMENAZAS 24
3.3. LA CADENA DE CIBERATAQUE 26
3.4. TÁCTICAS, TÉCNICAS Y PROCEDIMIENTOS 28
3.5. CASOS DE ESTUDIO 30
3.5.1. Reempaquetado de aplicaciones Android 30
3.5.2. Distribución de Ransomware 35

4. DISEÑO DE LA SOLUCIÓN 37

4.1. ENFOQUE 37
4.2. CAPACIDADES DEL VSOC 38
4.2.1. Monitorización y detección 38
4.2.2. Respuesta a incidentes y Threat Hunting 39
4.2.3. Ingeniería para la detección y la automatización 39
4.2.4. Threat Intelligence 39
4.3. ARQUITECTURA 41
4.3.1. Requisitos de arquitectura 41
4.3.2. Selección de la tecnología 42
4.3.3. Modelo de arquitectura 43
4.4. DEFINICIÓN DEL CASO DE USO 47
4.5. EXPERIMENTO 49
4.5.1. Instrumental requerido 49
4.5.2. Escaneo del bus CAN 50
4.5.3. Envío de logs al VSOC 53
4.5.1. Automatización de la respuesta al incidente 59

5. CONCLUSIONES 65

6. GLOSARIO DE ACRÓNIMOS Y TÉRMINOS 67

7. BIBLIOGRAFÍA 70

8. ANEXOS 72

8.1. ESPECIFICACIONES TÉCNICAS DE LA OBU FABRICADA POR Q-FREE 73
8.2. ESPECIFICACIONES TÉCNICAS DE LA RSU FABRICADA POR Q-FREE 74
8.3. ESPECIFICACIONES TÉCNICAS DE LA UNIDAD DE CONTROL DE Q-FREE 75
8.4. COMANDOS AT DEL PROTOCOLO ELM327 76
8.5. SCRIPT POWERSHELL PARA ENVÍO DE MENSAJES MEDIANTE DATA COLLECTOR API 78

Lista de ilustraciones

Ilustración 1. Why the future for cars is connected. Fuente: World Economic Forum 1
Ilustración 2. Escenario de aplicación V2V/V2I [2] 2
Ilustración 3. Representación simbólica de la analogía de la red V2X con las redes de ordenadores 3
Ilustración 4. Arquitectura de referencia de ciberseguridad de Microsoft 5
Ilustración 5. Diagrama de Gantt 6
Ilustración 6. Canales reservados para los servicios V2X en Europa y US. [5] 10
Ilustración 7. Pila de protocolos del sistema IEEE-DSRC [6] 11
Ilustración 8. Pila de protocolos del sistema ITS-G5 [6] 11
Ilustración 9. Comparativa de las arquitecturas IEEE DSRC y ETSI ITS-G5 [7] 11
Ilustración 10. Hoja de ruta del 3GPP con el estándar 5G. [9] 13
Ilustración 11. Desafíos en las redes vehiculares 5G (extracto) [10] 14
Ilustración 12. Estándares de seguridad en el automóvil [11] 15
Ilustración 13. Implementación del requisito UN R155. Fuente: Gartner [13] 17
Ilustración 14. Anatomía del vehículo conectado 18
Ilustración 15. Formato de trama CAN 2.0B 19
Ilustración 16. Terminales del conector OBD-II [17] 20
Ilustración 17. Cuadro de mando de una app de diagnósticos 21
Ilustración 18. Plataforma genérica para comunicaciones V2X 22
Ilustración 19. Arquitectura de la OBU [18] 22
Ilustración 20. Tipología de ataques, y funciones amenazadas [19] 25
Ilustración 21. La cadena de ciberataque (Cyber Kill Chain) [20] 26
Ilustración 22. Etapas donde se expone el atacante. Fuente: Gartner 27
Ilustración 23. La matriz MITRE ATT&CK 28
Ilustración 24. Juegos de matrices MITRE ATT&CK 29
Ilustración 25. Despliegue de la aplicación de diagnósticos 30
Ilustración 26. Modelo de comunicación ELM327 entre el smartphone y el vehículo 31
Ilustración 27. Comunicación remota con el adaptador ELM327 32
Ilustración 28. Detección de las tramas CAN en el código original de la app 32
Ilustración 29. Inserción del código malicioso en la app 33
Ilustración 30. Modelo de ataque con app maliciosa 34
Ilustración 31. Distribución de Ransomware 35
Ilustración 32. Tercer requisito UN ECE WP.29 R155: VSOC 37
Ilustración 33. Matriz de capacidades de un SOC. Fuente: Gartner 38
Ilustración 34. Threat detection best practice 40
Ilustración 35. Requisitos de arquitectura VSOC 41
Ilustración 36. Magic Quadrant for Security Information and Event Management 42
Ilustración 37. Arquitectura de ciberseguridad para el vehículo conectado 46
Ilustración 38. Instrumental del laboratorio 49
Ilustración 39. Adaptador ELM327 enchufado al conector OBD-II 50
Ilustración 40. Configuración y exportación de registros de Torque 51
Ilustración 41. Exportación del fichero de diagnósticos del vehículo 51
Ilustración 42. Configuración de Wireshark en modo promiscuo 52
Ilustración 43. Captura de mensajes CAN en Wireshark [3] 52

1. Introducción

1.1 Contexto y justificación del Trabajo

Durante los últimos años se han conseguido importantes progresos en el sector del vehículo inteligente orientados a la sostenibilidad, el confort y la seguridad en la conducción, y también en la carrera hacia el coche autónomo. Todo esto ha sido en gran medida gracias a la evolución de las comunicaciones móviles (Bluetooth, Wifi, telefonía celular…), de la inteligencia artificial, y del Internet de las Cosas o más concretamente del nuevo paradigma IoV (Internet de los Vehículos), surgido para habilitar la comunicación V2X ("Vehicle to Everything") que caracteriza al vehículo inteligente por su capacidad para conectarse con otros vehículos, con las infraestructuras, con personas, y con los dispositivos IoT de las ciudades inteligentes.

Ilustración 1. Why the future for cars is connected. Fuente: World Economic Forum

Gartner [1] predice un aumento en el número de vehículos conectados, de 357 millones en 2022 a 898 millones en 2030. Paralelamente, la cantidad de funciones controladas por software también está aumentando para habilitar nuevos modelos de negocio, como por ejemplo los servicios de diagnóstico remoto del vehículo, las actualizaciones del software vía OTA, la gestión de flotas, o las pólizas de seguro basadas en el comportamiento del conductor, entre muchos otros [2].

Estos dos factores combinados aumentan la superficie de ataque y el número y la gravedad de los riesgos de ciberseguridad asociados: violación de la privacidad del usuario, fugas de información sensible,

intrusiones con suplantación de identidad, ataques de denegación del servicio, y un largo etcétera [3].

En la implementación de los nuevos escenarios de negocio se pueden combinar diferentes estrategias de cooperación vehicular, haciéndolas coexistir para adaptarse a los requisitos específicos de cada escenario y a la diversidad del equipamiento tecnológico de los vehículos y de las infraestructuras viarias.

A modo de ejemplo, en un sistema como el que muestra la Ilustración 2, que tiene el doble objetivo de proteger la seguridad de las personas implicadas en un accidente, y de mitigar por otra parte la consiguiente congestión del tráfico desviándolo por rutas alternativas, se trata en primer lugar de comunicar el evento de forma inmediata a los vehículos que se están aproximando y que podrían no tener la suficiente visibilidad para eludir el peligro, y en segundo lugar y con menor prioridad, comunicar la información a los centros de control del tráfico para que los sistemas de navegación avisen a los conductores recomendándoles una ruta alternativa.

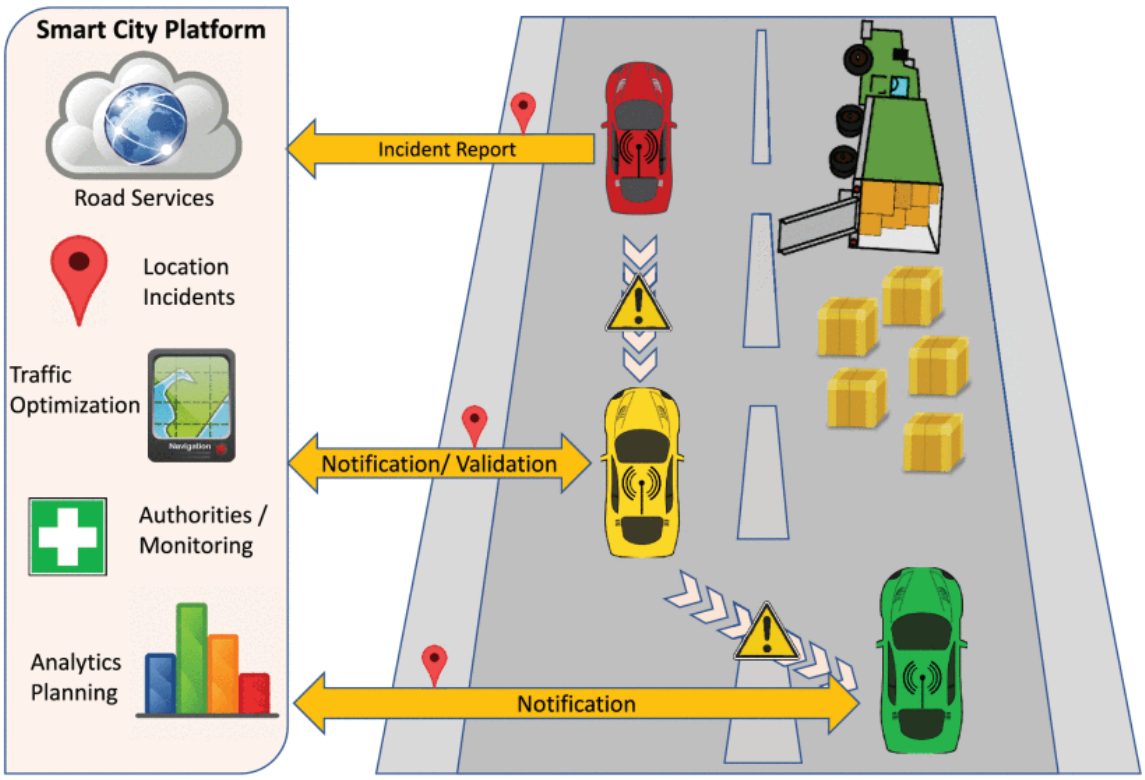


Ilustración 2. Escenario de aplicación V2V/V2I [2]

Para dar solución a este escenario se pueden contemplar y combinar tres estrategias distintas:

1- Detección directa en el vehículo: los sistemas sensoriales con los que va equipado un coche moderno podrían reconocer la obstrucción en el carril por el que circula y reaccionar en consecuencia, pero esta es una capacidad de la que muchos vehículos no disponen todavía.

2- V2V: Los protocolos de comunicación V2V están optimizados para asegurar la mínima latencia y la máxima velocidad de transmisión de los mensajes (cortos) de seguridad del vehículo. Esta comunicación permite propagar el evento de vehículo a vehículo de forma rápida y efectiva para que puedan reaccionar a tiempo, pero es de corto alcance, así que solamente funcionará mientras haya una mínima densidad de vehículos.

3- V2I: Los mismos protocolos de comunicación también permiten el intercambio de información entre el vehículo y las RSUs desplegadas en la infraestructura viaria. Si el vehículo siniestrado dispone del sistema eCall, su propia llamada al centro de emergencias a través de la red celular advertirá también al centro de control de tráfico, pero, en cualquier caso, la comunicación V2I desde ese vehículo o desde los que circulen por ese segmento de la ruta, juega un papel imprescindible no sólo para notificar el incidente sino también posteriormente para reportar la vuelta a la normalidad una vez resuelta la obstrucción del carril.

Este ejemplo es uno de los múltiples escenarios que habilitan las comunicaciones V2X, y permite intuir los posibles efectos de un ciberataque y su impacto en la confiabilidad del sistema, tanto desde el punto de vista de los usuarios como de los gobiernos.

Con el propósito de contribuir a la mejora del sistema y fomentar la confianza en el vehículo autónomo, este proyecto se enfoca en profundizar en la materia y demostrar cómo se pueden proteger los sistemas vehiculares cooperativos utilizando metodologías y herramientas de ciberdefensa análogas a las utilizadas tradicionalmente en las redes de ordenadores.

Ilustración 3. Representación simbólica de la analogía de la red V2X con las redes de ordenadores

1.2 Objetivos del Trabajo

El objetivo estratégico de este proyecto es facilitar la comprensión de los riesgos de ciberseguridad que amenazan el entorno del vehículo conectado, y el tipo de medidas/soluciones que debe adoptar la industria automotriz para combatirlos.

Para alcanzar esa aspiración se han establecido los siguientes objetivos más concretos:

- Identificar y analizar los riesgos específicos de ciberseguridad que afectan el entorno del vehículo conectado.
- Conocer el panorama regulatorio al que se enfrentan los fabricantes de vehículos y su cadena de suministro.
- Comprender las medidas que deben adoptar para afrontar las exigencias regulatorias, y los plazos para implementarlas.
- Identificar los elementos clave de una arquitectura de ciberseguridad de propósito general, y a continuación distinguir los aspectos particulares de la ciberseguridad en el vehículo conectado.
- Entender el modelo de solución a través de un caso de uso como ejemplo ilustrativo.

1.3 Enfoque y método seguido

Dado el enfoque eminentemente teórico de este trabajo, se ha realizado una investigación y análisis del estado del arte de los protocolos y **estándares de comunicación** orientados al vehículo conectado (IEEE 802.11p, 4G-LTE y 5G-NR), así como del **marco regulatorio** existente sobre los requisitos de ciberseguridad para los fabricantes de automóviles y sus cadenas de suministro (UN ECE WP.29 e ISO/SAE 21434). Este estudio es esencial para identificar la **ventana de oportunidad**.

En cualquier proceso de modelado de amenazas siempre es fundamental conocer el activo que se desea proteger. En este caso, se ha realizado un estudio de los principales componentes del vehículo conectado y sus mecanismos de interacción tanto internos como externos, es decir, su **arquitectura de comunicación** intra-vehicular e inter-vehicular.

Además, se ha llevado a cabo un amplio análisis de la literatura existente sobre los nuevos vectores de ataque y las técnicas y tácticas de los adversarios, con el fin de identificar y comprender las vulnerabilidades que pueden requerir mayor atención en el modelo de amenazas.

Con este modelo definido, se ha utilizado la **arquitectura de referencia de ciberseguridad** de Microsoft (https://learn.microsoft.com/es-es/security/cybersecurity-reference-architecture/mcra) para diseñar una solución integrable en una plataforma de mercado, y se ha implementado

un experimento utilizando datos sintéticos para demostrar el funcionamiento del sistema.

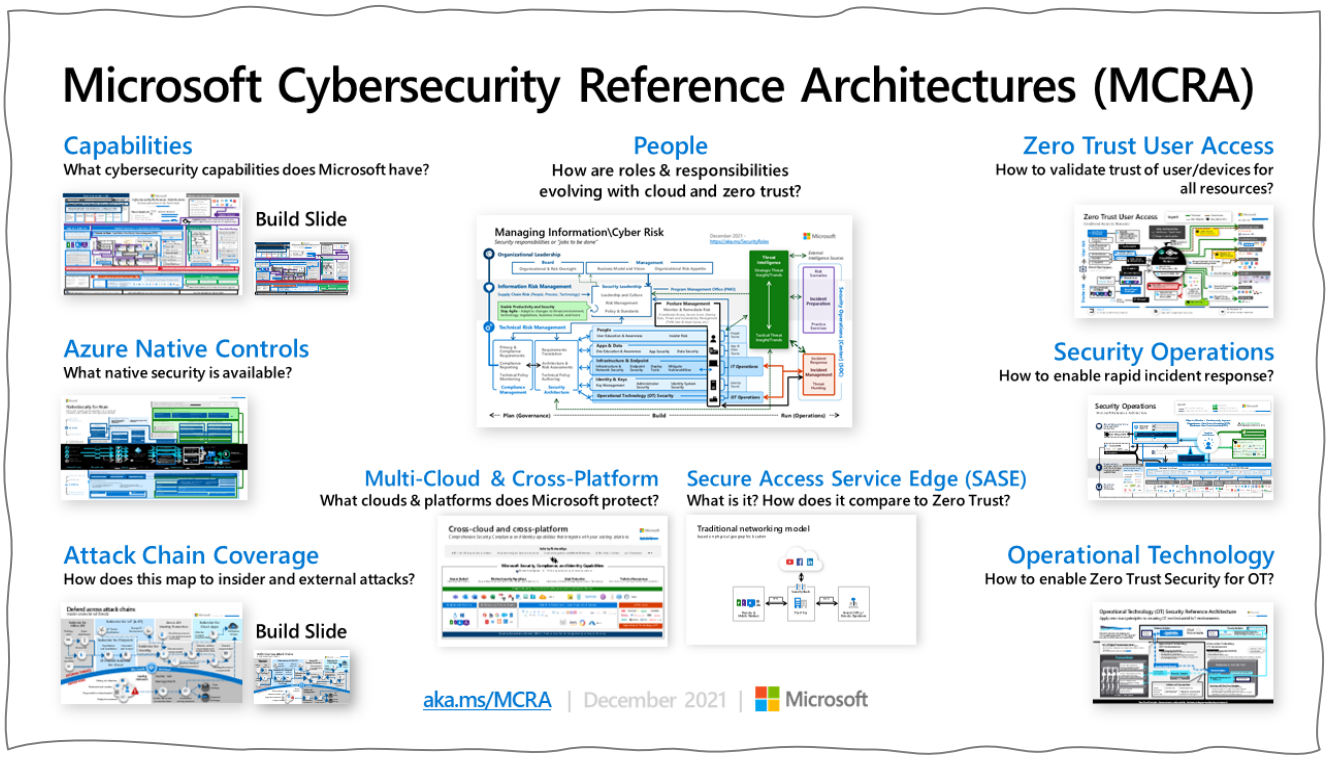


Ilustración 4. Arquitectura de referencia de ciberseguridad de Microsoft

1.4 Planificación del Trabajo

El siguiente cronograma muestra las actividades llevadas a cabo en el proyecto, completadas el 12 de junio de 2023 sin cambios significativos respecto a la planificación inicial:

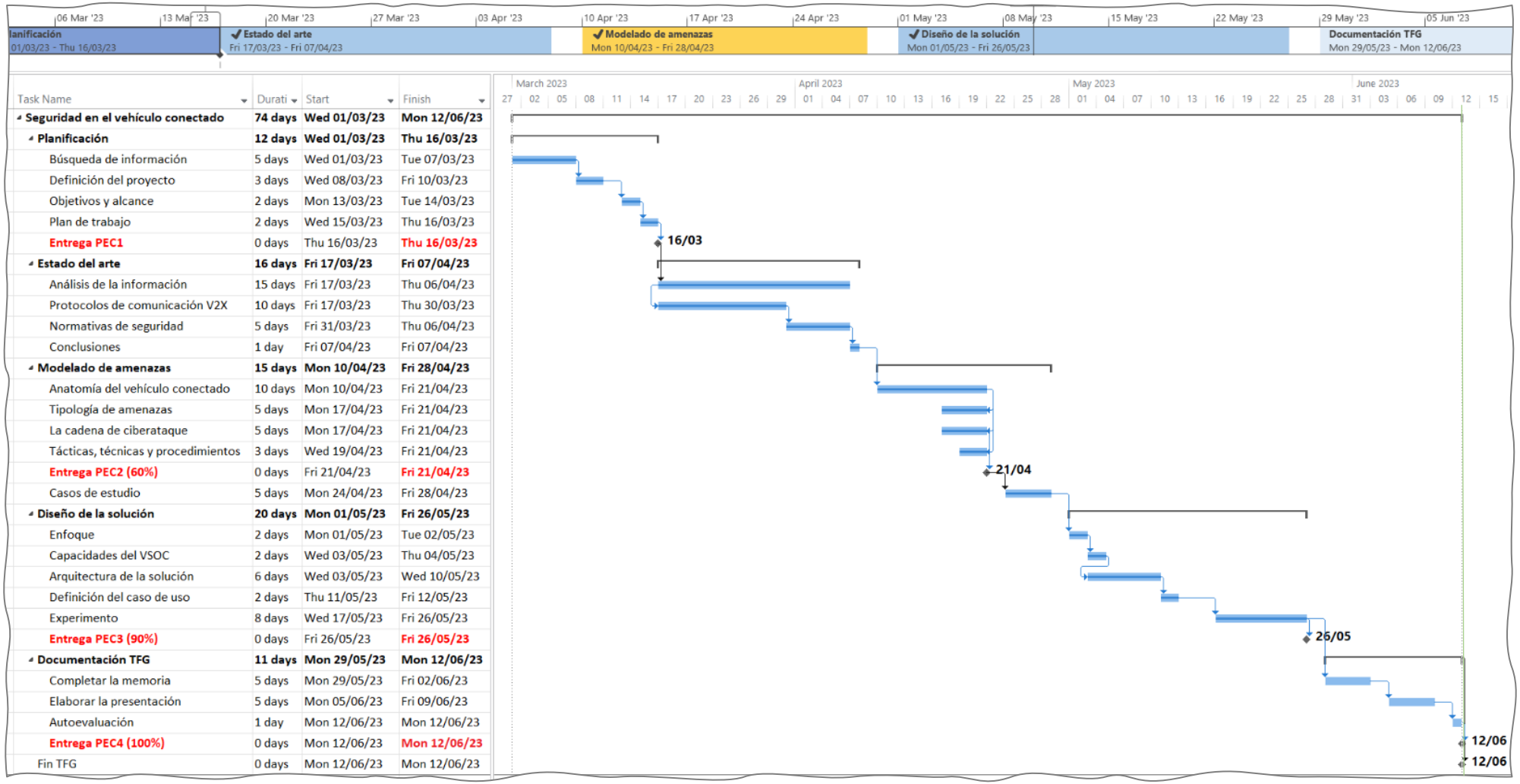


Ilustración 5. Diagrama de Gantt

1.5 Breve sumario de productos obtenidos

A continuación, se describen los entregables generados a través de las diferentes actividades del proyecto:

Actividad	Entregable
• Análisis de los protocolos y estándares de comunicación para el vehículo conectado (IEEE 802.11p, 4G-LTE y 5G-NR)	• Síntesis sobre las actuales tecnologías de comunicación V2X
• Análisis de las principales directivas de seguridad aplicables en la industria del automóvil	• Estado del arte del marco regulatorio y conclusiones (ventana de oportunidad)
• Estudio de la arquitectura de comunicación intra-vehicular e inter-vehicular • Analizar la literatura existente sobre los nuevos vectores de ataque y las técnicas y tácticas de los adversarios (amenazas y vulnerabilidades)	• Modelo de amenazas
• Diseñar una solución a alto nivel basada en la arquitectura de referencia de Microsoft	• Diseño de alto nivel
• Definición e implementación de un caso de uso con datos sintéticos, a modo ilustrativo	• Guía de implementación

1.6 Breve descripción de los capítulos de la memoria

El contenido de esta memoria está estructurado de la siguiente manera:

- Capítulo 1. Introducción

  Este capítulo describe la motivación del proyecto justificándola dentro del contexto actual del vehículo conectado, y define los objetivos, el enfoque y la planificación del trabajo.

- Capítulo 2. Estado del arte.

  En este capítulo se hace un doble estudio.

  La primera parte aborda el estado actual de las tecnologías de comunicación empleadas en el vehículo conectado, incluyendo tanto las basadas en Wifi (DSRC e ITS-G5) como las que utilizan la red celular 5G.

  El propósito no es compararlas, ya que todas tienen su función y ámbito de aplicación, sino conocerlas con el nivel de detalle necesario para comprender la naturaleza de los riesgos vinculados a posibles intrusiones que puedan producirse a través de las comunicaciones. Este es el principal vector de ataque en el vehículo conectado y, por lo tanto, debe ser la máxima prioridad en la estrategia de ciberseguridad.

  La segunda parte se ocupa de los requisitos regulatorios en materia de ciberseguridad establecidos en la normativa vigente para el sector del automóvil (UN ECE WP.29), con el propósito de extraer conclusiones acerca de las áreas en las que los fabricantes puedan requerir más ayuda.

- Capítulo 3. Modelado de amenazas

  El modelado de amenazas es una práctica clave en la metodología de gestión de riesgos de ciberseguridad, que permite evaluar las posibles vulnerabilidades y amenazas, y desarrollar las correspondientes estrategias de mitigación y respuesta.

  Este capítulo presenta un proceso de modelado de amenazas fundamentado en el análisis de los elementos que conforman la infraestructura de comunicación y control del vehículo, y lo ilustra con dos casos de estudio documentados.

- Capítulo 4. Diseño de la solución

  Este capítulo se divide en tres partes principales.

  La primera sección es introductoria, detalla las capacidades con las que debe contar un VSOC (Centro de Operaciones de Seguridad Vehicular), dado que este aspecto es el requisito clave de la normativa UN ECE WP.29 en el cual se enfoca este trabajo, en base a las conclusiones derivadas del análisis del estado del arte.

  En la segunda parte, se establecen los requisitos de arquitectura que la solución debe cumplir, se elige la tecnología considerando las recomendaciones de Gartner en su “Magic Quadrant” más reciente, y se diseña la solución conforme al patrón de arquitecturas de referencia de ciberseguridad de Microsoft (MCRA).

  Por último, se define el caso de uso sobre el que se realizará el experimento, y se desarrolla la solución en un entorno de laboratorio, documentándola paso a paso.

2. Estado del arte

2.1. Introducción

Este capítulo se divide en dos partes. En la primera se analiza el estado actual de los protocolos y estándares de comunicación utilizados en el vehículo conectado. En la segunda se estudia el marco normativo que regula actualmente los requisitos de ciberseguridad para los fabricantes de automóviles y sus cadenas de suministro.

En un automóvil moderno existen del orden de 150 ECUs ("Electronic Control Units") recibiendo señales de los sensores integrados en las redes internas del vehículo, como por ejemplo el bus CAN ("Controller Area Network"), para suministrarles continuamente información de estado sobre aceleración, velocidad, presión de los neumáticos, etcétera [4].

Al mismo tiempo, estos ECUs están conectados al mundo exterior mediante redes de comunicación inalámbrica a través de las cuales obtienen datos sobre la situación del tráfico, el estado de las carreteras, riesgos de colisión y de atropello de peatones dentro y fuera de nuestra línea de visión, entre otros.

Así pues, en primer lugar, se analizará el estado del arte de las tecnologías utilizadas para la conexión del vehículo con el mundo exterior; y seguidamente se estudiará la situación de los estándares de ciberseguridad que se están imponiendo en el ecosistema de la industria del automóvil.

Es importante comprender los protocolos de comunicación inalámbrica utilizados en los vehículos conectados y los estándares que los definen, ya que las comunicaciones del vehículo con el mundo exterior son la principal vía de intrusión.

También es fundamental conocer las normativas de cumplimiento regulatorio sobre ciberseguridad para el sector del automóvil y los estándares que nos ayudan a interpretarlas. De esta manera, es posible desarrollar una solución compatible con las normativas vigentes y, al mismo tiempo, identificar el tipo de solución que puede aportar más valor para el sector.

2.2. Protocolos de comunicación V2X

2.2.1. IEEE DSRC y ETSI ITS-G5

La base de los sistemas vehiculares cooperativos es el intercambio de información entre los vehículos y con el entorno circundante. Para este propósito, se han desarrollado tecnologías como DSRC en Estados Unidos e ITS-G5 en Europa.

En Estados Unidos, DSRC utiliza la banda de frecuencia de 5,9 GHz y se basa en el estándar IEEE 802.11p, que se publicó en 2010 y regula sus especificaciones técnicas. Esta tecnología permite la comunicación inalámbrica de corto alcance entre vehículos y con la infraestructura de la carretera.

Por otro lado, unos años más tarde en Europa se implementó ITS-G5 como una variante del estándar IEEE 802.11p, que también utiliza la banda de frecuencia de 5,9 GHz. ITS-G5 incorpora ciertas especificaciones técnicas adicionales para adaptarse a las necesidades específicas de las aplicaciones de transporte inteligente en Europa.

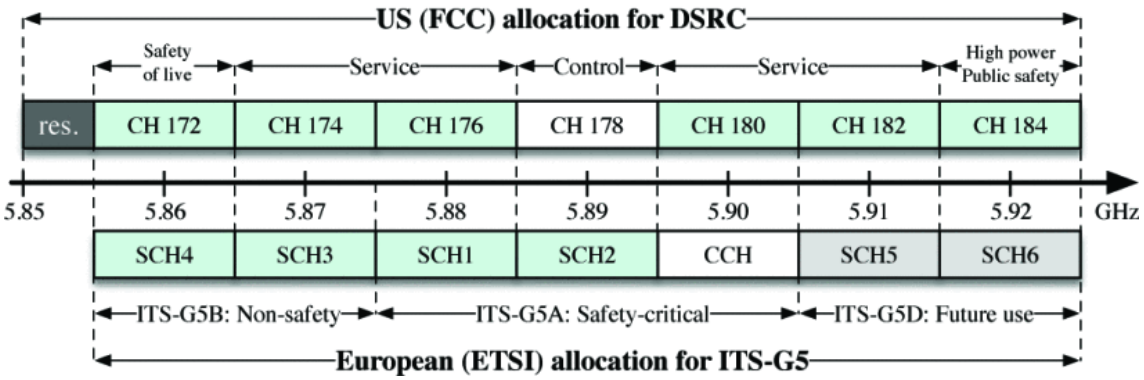


Ilustración 6. Canales reservados para los servicios V2X en Europa y US. [5]

Ambos sistemas se basan por lo tanto en el estándar IEEE 802.11p, que es una adaptación del Wifi para su uso en el entorno vehicular.

Este estándar se caracteriza por su capacidad para crear redes de nodos en movimiento de forma dinámica, lo que permite una comunicación efectiva en situaciones de alta velocidad y alta densidad de tráfico.

Además, ofrece buenas prestaciones en términos de latencia y velocidad de transmisión, lo cual lo hace adecuado para aplicaciones de seguridad vial y tráfico inteligente [6].

A continuación, se muestra la pila de protocolos y los estándares asociados a ambas tecnologías:

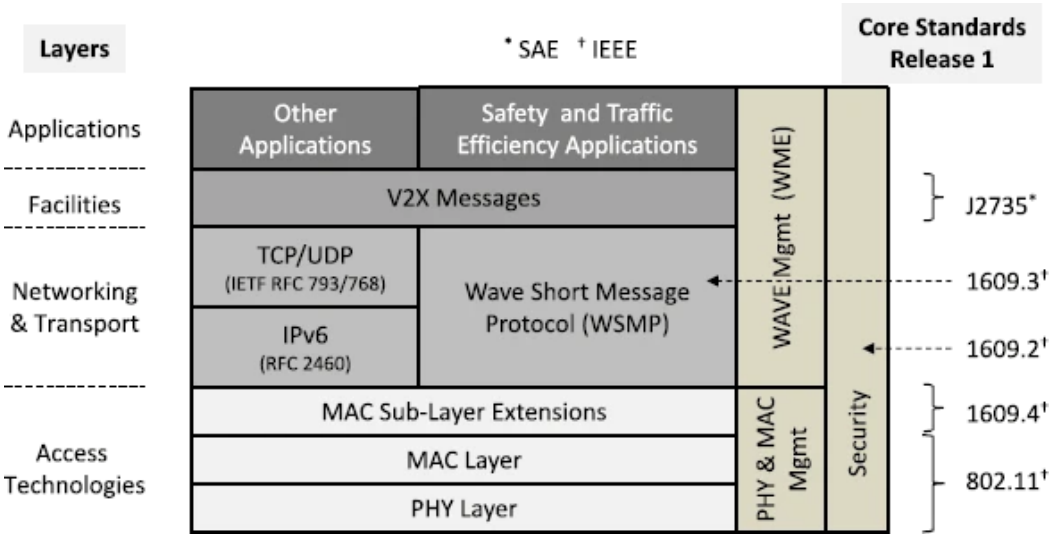


Ilustración 7. Pila de protocolos del sistema IEEE-DSRC [6]

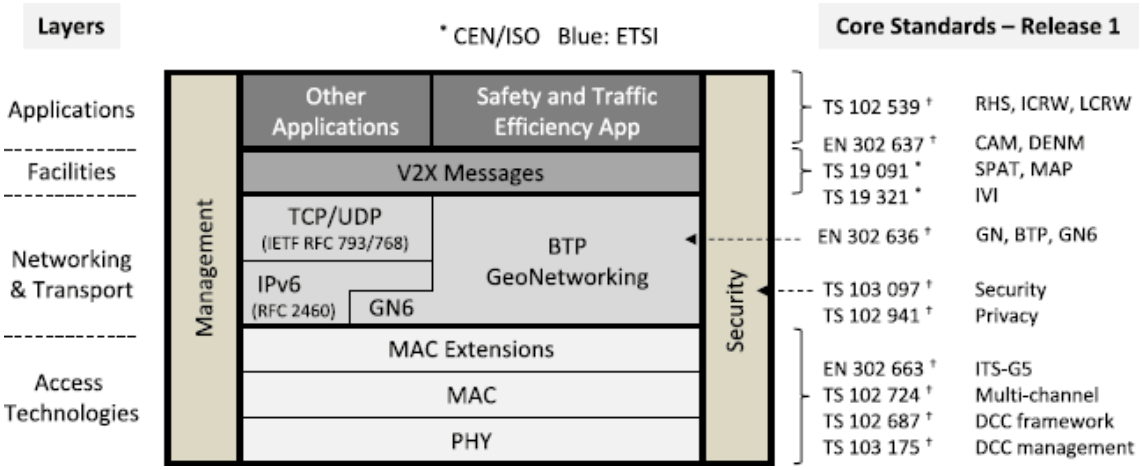


Ilustración 8. Pila de protocolos del sistema ITS-G5 [6]

La siguiente ilustración ofrece una comparativa de las dos arquitecturas:

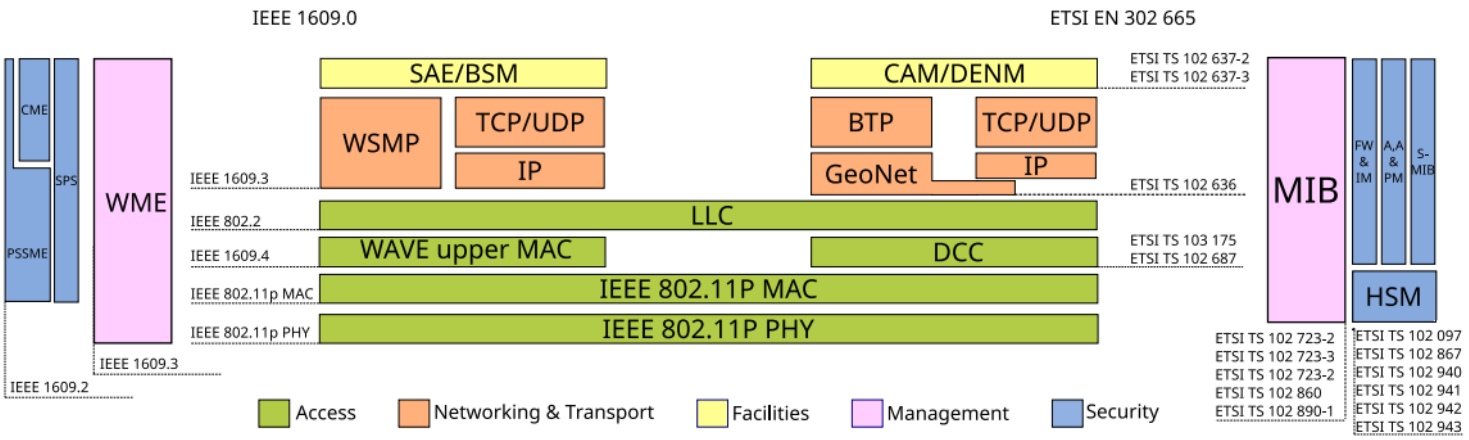


Ilustración 9. Comparativa de las arquitecturas IEEE DSRC y ETSI ITS-G5 [7]

Los dos sistemas utilizan la pila de protocolos del estándar IEEE 802.11, que se basa en una arquitectura en capas similar a la del modelo OSI.

La capa física (nivel OSI 1) define las especificaciones de modulación y codificación de la señal, y la capa MAC (nivel OSI 2) define el acceso al medio inalámbrico y el control de la comunicación -transmisión de los paquetes en el canal, enrutamiento, y priorización para favorecer a las aplicaciones de seguridad frente a cualquier otra-.

En las capas de red y transporte (niveles OSI 3 y 4) se utilizan los protocolos estándar IPv6 y TCP/UDP, y los protocolos específicos WSMP (Wave Short Message Protocol) en DSRC y BTP (Basic Transport Protocol) GeoNetworking en ITS-G5. Este último permite el enrutamiento multi-hop (multi-salto), y se basa en las coordenadas geográficas de los nodos para el direccionamiento y envío de los paquetes. Con este método es posible hacer el broadcasting de un paquete a múltiples nodos, pero además permite la entrega de un paquete a un nodo determinado mediante rutas alternativas en caso necesario, apoyándose en otros nodos que de forma cooperativa reenvían el paquete.

En la capa de “Facilities” (niveles OSI 5 y 6), se define un catálogo normalizado de mensajes V2X a intercambiar entre los nodos de la red.

El tipo de mensaje más relevante en DSRC es el BSM (Basic Safety Message), conocido como “beacon” (baliza) porque se intercambia constantemente entre los vehículos que forman la red transmitiendo la posición de cada uno junto con algunas características básicas del vehículo (tamaño, estado, etc.). De esta manera se mantiene la sesión activa para que cuando lo necesiten puedan intercambiarse cualquier tipo de dato de forma inmediata.

De forma análoga, en ITS-G5 se define el mensaje CAM (Cooperative Awareness Message), que periódicamente -entre 1 y 10 veces por segundo dependiendo de la velocidad del vehículo- transmite la información crítica del estado del vehículo hacia todos los nodos colindantes como soporte para la aplicación de eficiencia y de seguridad en la conducción (prevención de atascos, etc.). Adicionalmente se definen los mensajes de tipo DENM (Decentralized Environmental Notification Message), que se transmiten en respuesta a algún evento cuando por ejemplo una aplicación ITS detecta una situación de peligro a partir de la información facilitada por sensores embarcados en otro vehículo.

Estos mensajes contienen datos sobre la situación en cuestión, y también sobre la velocidad del vehículo, y su rumbo y posición. Se transmiten con una frecuencia de entre 1 y 20 Hz, y solamente mientras dura la situación de peligro.

En cuanto a Seguridad, en DSRC el estándar IEEE 1609.2 define las especificaciones para una comunicación segura entre las aplicaciones y los procesos que se ejecutan en las distintas capas de la pila de protocolos. Proporciona los mecanismos de autenticación y también para el cifrado de los mensajes, cuando se requiera, mediante firma y certificado digital, pero protegiendo la privacidad del conductor.

La arquitectura ITS-G5 también se basa en el estándar 1609.2 para definir las especificaciones de seguridad, aunque introduce ligeras variantes, principalmente en la infraestructura PKI para la gestión de los certificados.

2.2.2. 5G-V2X

Aunque el estándar IEEE 802.11p ofrece un buen rendimiento en escenarios de corto alcance, las tecnologías LTE 4G y NR 5G habilitan nuevos casos de uso en los que se requiere un mayor rango de cobertura y más ancho de banda para aplicaciones con alto tráfico de datos.

El 3GPP (3rd Generation Partnership Project), que nació dentro del ETSI para mantener actualizadas las versiones de UMTS (3G), tiene la responsabilidad de evolucionar las especificaciones 5G en un modelo compatible con los requisitos V2X, especialmente en términos de rendimiento en movilidad, tamaño de los mensajes y, sobre todo, de latencia.

La Ilustración 10 muestra la hoja de ruta del 3GPP con relación al 5G, en la que cabe señalar los servicios de proximidad (ProSe) como una de las capacidades más relevantes para la comunicación V2X.

Estos servicios ProSe son un conjunto de normas desarrolladas para permitir la comunicación directa entre dispositivos cercanos, utilizando bandas de frecuencia no licenciadas para las redes dinámicas basadas en Wifi, sin depender de la infraestructura de red celular, pero aprovechando las características de latencia, alcance y velocidad de transmisión de la tecnología 5G, superiores a las de DSRC e ITS-G5 [8].

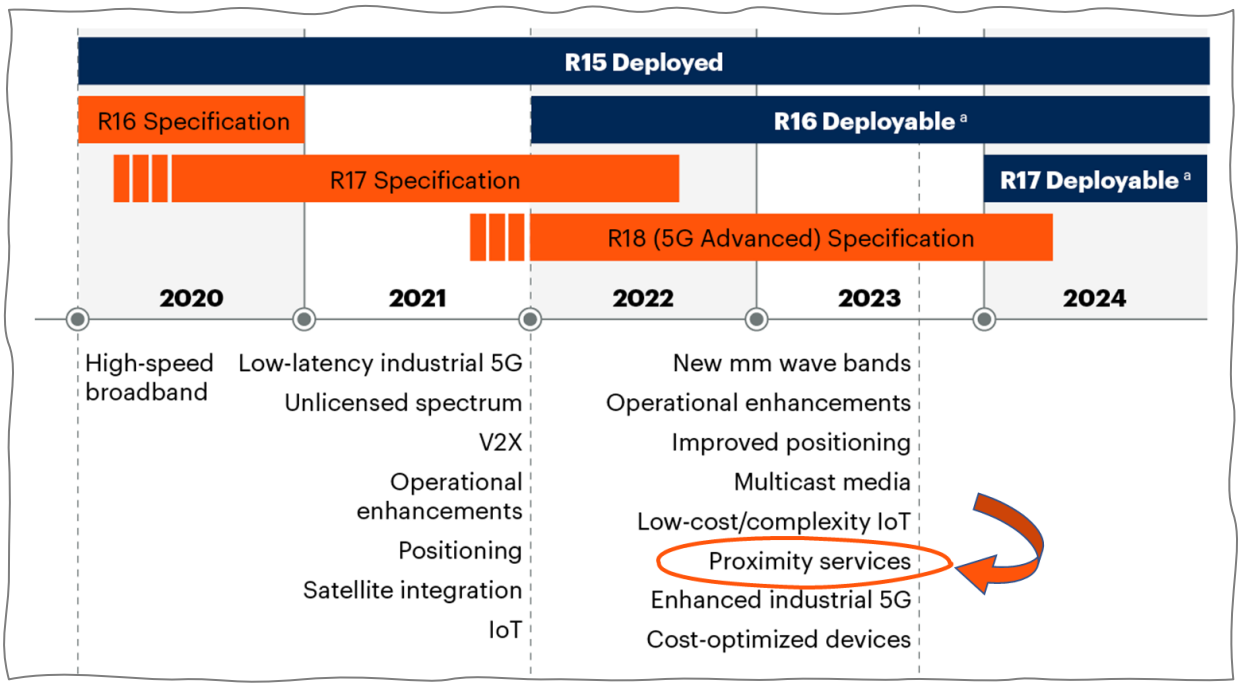


Ilustración 10. Hoja de ruta del 3GPP con el estándar 5G. [9]

En términos de Seguridad, el 3GPP ha diseñado un conjunto de especificaciones de seguridad para proteger la comunicación V2X en tecnología 5G [10]:

- **Confidencialidad**: Utiliza IPSec para establecer canales seguros que protejan los mensajes de señalización entre los puntos de conexión, y mecanismos de cifrado TLS para la comunicación iniciada desde los OBUs hacia los servicios ProSe.
- **Integridad**: Utiliza la firma digital para proteger los datos contra la manipulación durante la transmisión.
- **Autenticidad**: Ha desarrollado un esquema de autenticación para asegurar que solo los vehículos autorizados puedan acceder a la red de comunicación V2X. Este esquema utiliza certificados digitales para garantizar la identidad del vehículo.
- **Resistencia a ataques**: El 3GPP ha desarrollado técnicas para detectar y prevenir ataques en la comunicación V2X. Esto incluye la monitorización constante de la red para detectar anomalías y la implementación de medidas de seguridad para prevenir ataques.
- **Privacidad**: El 3GPP ha diseñado un esquema de privacidad para proteger la información personal de los usuarios de V2X. Este esquema utiliza técnicas de anonimización para proteger la privacidad de los usuarios.

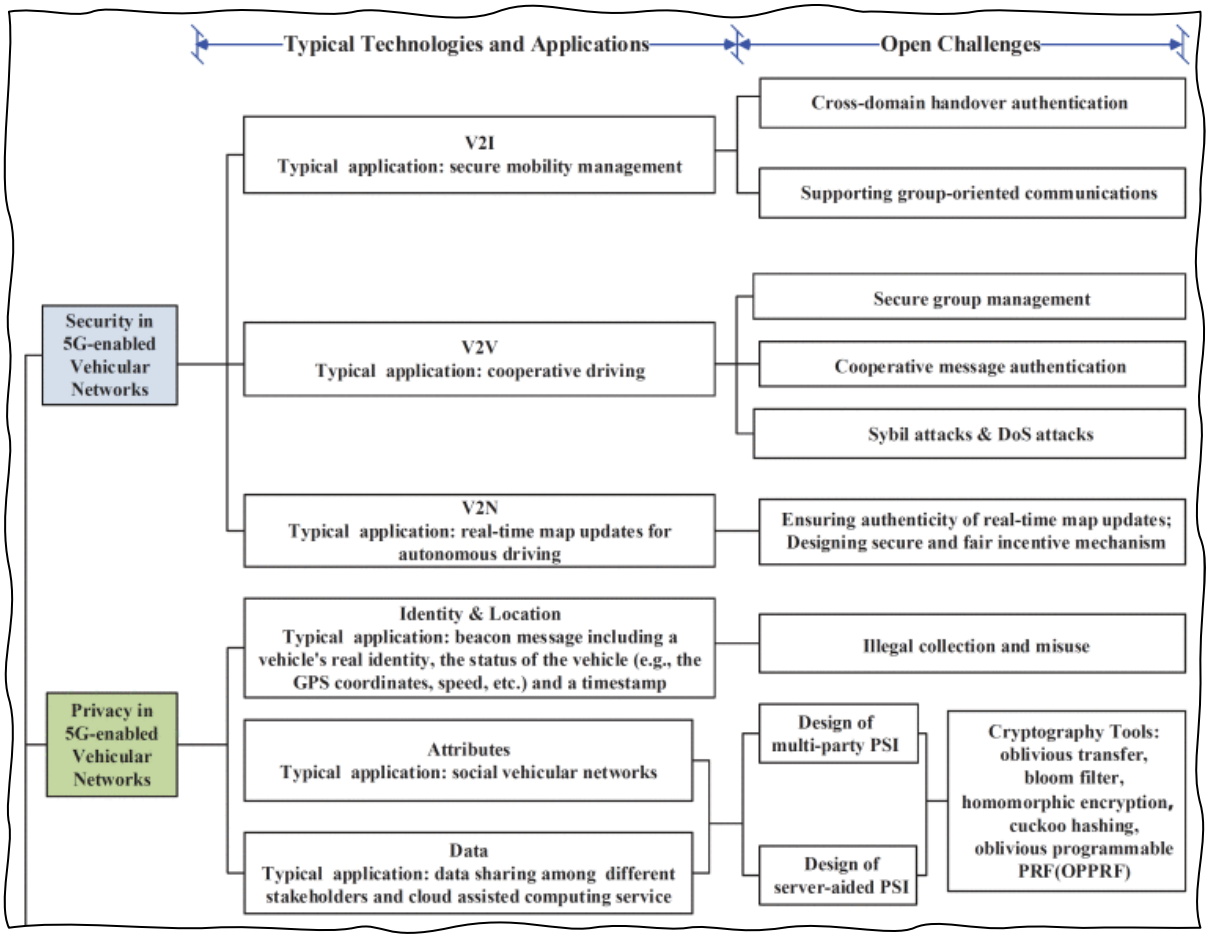


Ilustración 11. Desafíos en las redes vehiculares 5G (extracto) [10]

2.3. Normativas de seguridad

2.3.1. Introducción

A lo largo del tiempo, según han ido apareciendo nuevos elementos en el ecosistema de los Sistemas Inteligentes de Transporte, diferentes organismos (ISO, SAE, ETSI, ITU…) han ido desarrollando distintos grupos de estándares [11].

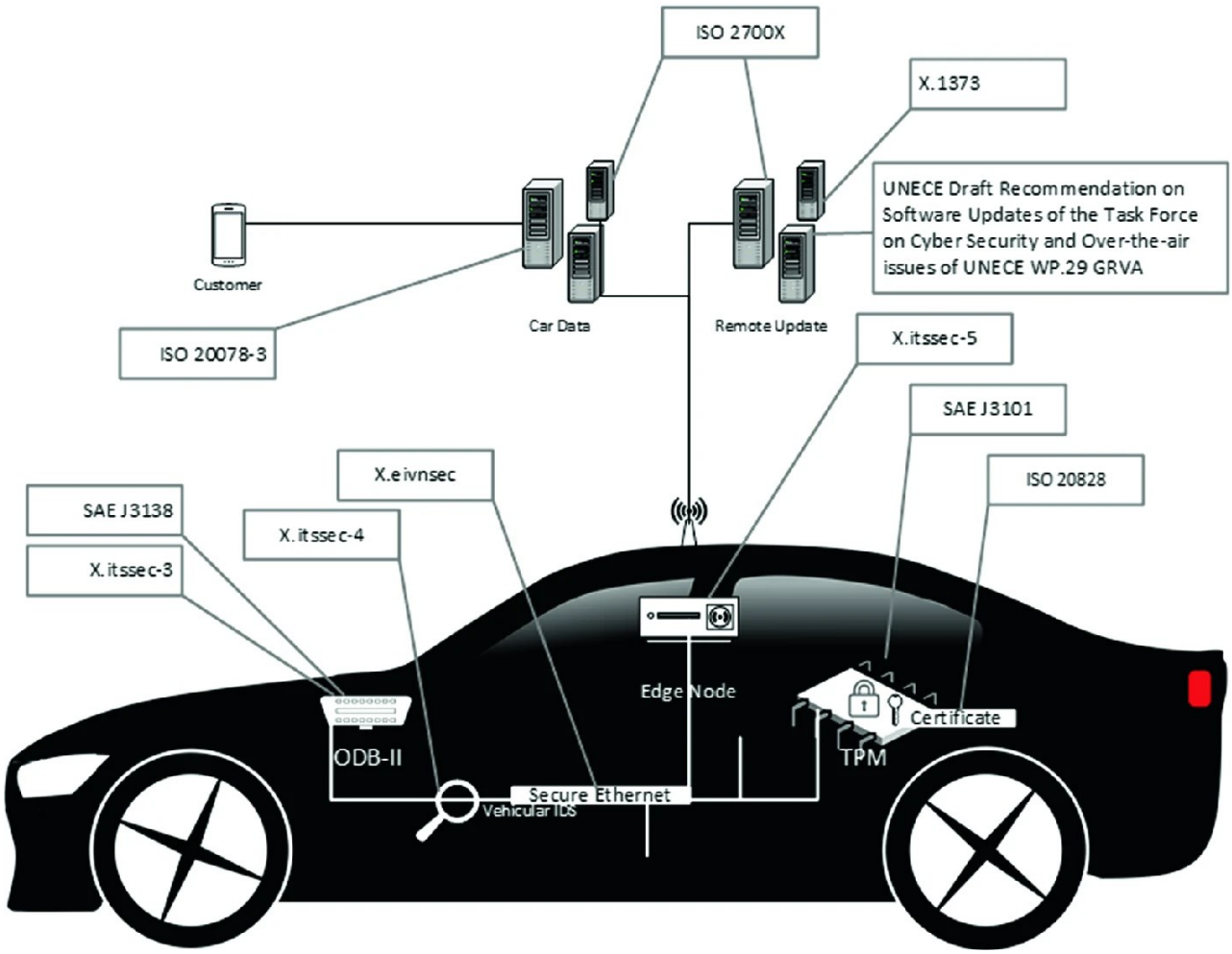


Ilustración 12. Estándares de seguridad en el automóvil [11]

La falta de una regulación global ha provocado solapes que se evidencian por ejemplo en las especificaciones de seguridad del conector OBD-II, para las que existen guías del grupo de seguridad basada en hardware, y también en el grupo de certificados digitales, publicadas en 2006 y que no tienen en cuenta los progresos conseguidos posteriormente con la seguridad basada en hardware.

Por ese motivo, la Comisión Económica de las Naciones Unidas para Europa ha desarrollado nuevos aspectos en su acuerdo UN ECE WP.29 para la

armonización de la reglamentación sobre Vehículos, centrados principalmente en la implantación de un Sistema de Gestión de la Ciberseguridad (CSMS, Cyber Security Management System) [12].

2.3.2. Normativa UN ECE WP.29 y estándar ISO/SAE 21434

La normativa UN ECE (*United Nations Economic Commission for Europe*) WP.29 define los siguientes requisitos, de cumplimiento obligatorio para todos los nuevos tipos de vehículos que se fabrican desde julio de 2022 y para todos los nuevos vehículos que se produzcan a partir de julio de 2024:

- R155 (CSMS): Gestión de la ciberseguridad
- R156 (SUMS): Gestión de las actualizaciones del software

Gartner [13] destaca la enorme responsabilidad que a partir de ahora recae sobre los CIOs y los CISOs de las empresas de automoción para construir un CSMS compatible con el requisito regulatorio UN R155, y opina que este es un cambio definitivo y sin precedentes para la ciberseguridad del automóvil, porque es la primera regulación de aplicación internacional y ha sido adoptada por más de 60 países.

Gartner aconseja a los CIOs que aborden sus procesos de homologación con acuerdo a las siguientes prioridades:

1. Utilizar el nuevo estándar **ISO/SAE 21434** [4] [14] para definir el marco de trabajo del **TARA** (Threat Analysis and Risk Assessment), ya que está ampliamente aceptado y puede resultar de gran ayuda en aspectos donde el UN R155 resulta ambiguo -principalmente cuando se refiere a aplicar buenas prácticas, sin concretar cuales son éstas-.
2. Crear el **CSMS** contando con expertos en ciberseguridad para resolver las cuestiones interpretables de la norma desde el conocimiento de los riesgos y de las estrategias de defensa asociadas.
3. Desarrollar los procesos del **VSOC** (Vehicle Security Operations Center) para la detección de intrusiones y la respuesta ante amenazas, definiendo los activos de la arquitectura del vehículo que pueden estar afectados por cada vector de ataque y la forma de proceder en cada caso.
4. Controlar la **cadena de suministro**, exigiendo en cascada el cumplimiento de la misma normativa a los proveedores, y certificando que las posibles vulnerabilidades introducidas por éstos tienen su plan de mitigación.
5. Implantar un proceso de **reevaluación** continua creando un repositorio con los incidentes registrados y las mejores prácticas a aplicar frente a cada tipo de vulnerabilidad.

2.4. Conclusiones

Tras analizar el estado de la cuestión, se concluye que es crucial llevar a cabo una tarea rigurosa en relación con los sistemas y los procedimientos de seguridad que protegen el entorno del vehículo conectado. Esta tarea es aún más urgente debido a la adopción del acuerdo UN ECE WP.29 en más de 60 países, el cual establece que el fabricante será responsable de cualquier incidente de ciberseguridad que afecte al vehículo, a menos que esté cumpliendo con el requisito regulatorio **UN R155**.

Dicho requisito tiene como objetivo principal conseguir que los fabricantes de automóviles y sus cadenas de suministro refuercen el control y la visibilidad de sus procesos de ciberseguridad.

Según el informe de Gartner [13], *"**la ciberseguridad es una lucha sin fin, en la que sólo es cuestión de tiempo que una vulnerabilidad llegue a ser explotada**"*.

Por ese motivo, este proyecto pondrá el acento en estudiar la arquitectura y las características de un VSOC (Vehicle Security Operations Center), puesto que es uno de los 5 aspectos exigidos por la regulación, y es la base sobre la cual los equipos de ciberdefensa pueden implementar y mantener actualizados sus procesos de detección temprana de amenazas, e investigar los incidentes y reaccionar a ellos con eficacia.

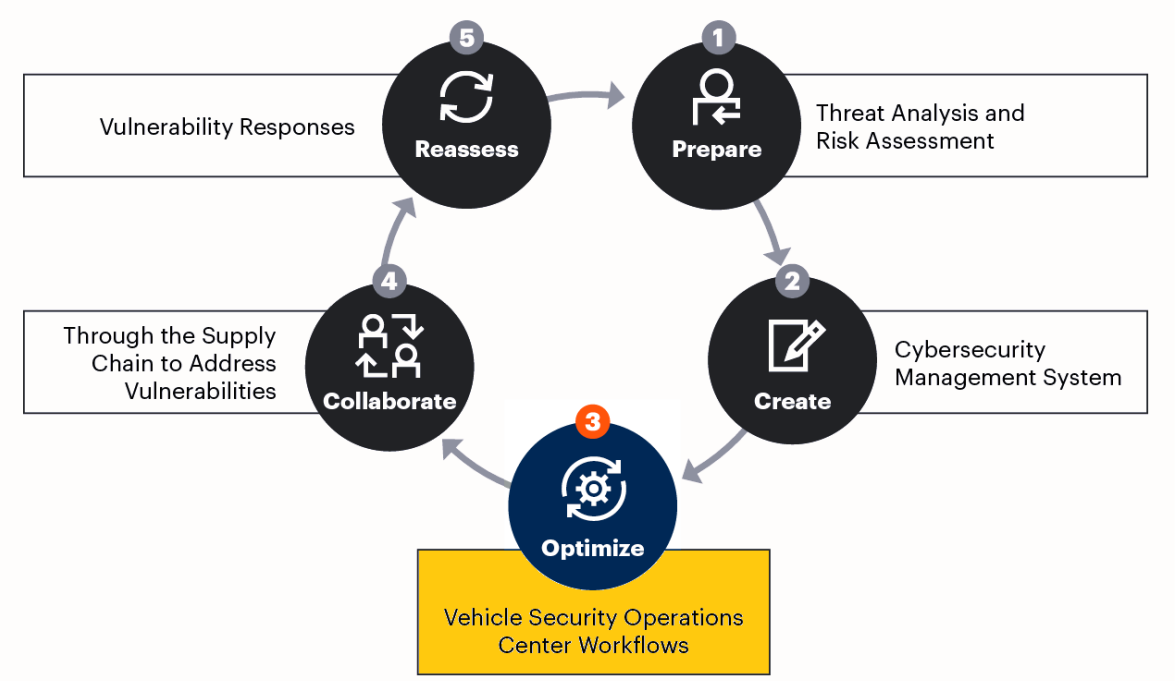


Ilustración 13. Implementación del requisito UN R155. Fuente: Gartner [13]

3. Modelado de amenazas

3.1. Anatomía del vehículo conectado

Un principio básico de cualquier proceso de modelado de amenazas consiste en conocer el objeto que se desea proteger, es decir, entender cuáles son sus principales componentes y cuáles son las interacciones entre ellos y con el mundo exterior. Esto permite identificar las superficies de ataque.

Se debe contemplar el vehículo conectado como una red de ordenadores sobre ruedas, y no como un solo ordenador, puesto que el vehículo contiene una red interna con múltiples unidades de control electrónico (ECUs), que son procesadores ejecutando aplicaciones sobre sistemas operativos como Linux o Android [15].

De hecho, coexisten varias redes conectadas entre sí a través de un Gateway, tal como muestra la siguiente ilustración:

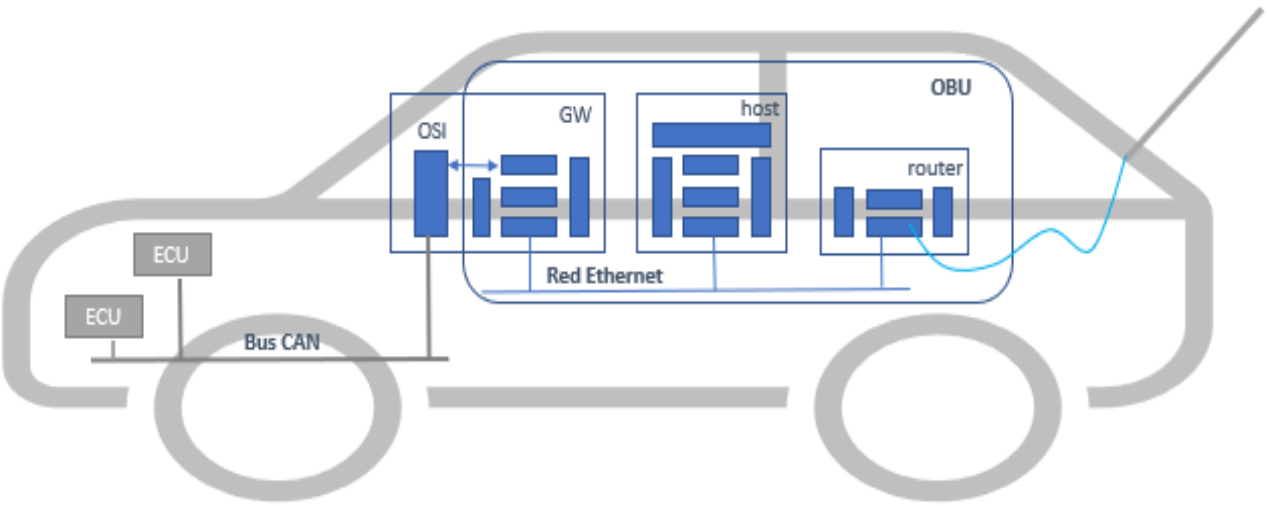


Ilustración 14. Anatomía del vehículo conectado

1. La primera red es el bus CAN (Controller Area Network), que es un protocolo de bajo nivel diseñado específicamente para el control en tiempo real de los mecanismos básicos de gestión del vehículo: motor, transmisión, frenos, diagnósticos de funcionamiento, etcétera, y está optimizado para garantizar una alta fiabilidad con un consumo muy bajo de energía.
2. La segunda es una red Ethernet, que por sus mejores características de ancho de banda habilita la comunicación TCP/IP de aplicaciones con alto consumo de datos, como pueden ser las que gestionan la consola de Infotainment, las de los sistemas ADAS (Advanced Driver Assistance System), o las actualizaciones del software vía OTA (Over-The-Air).
3. Adicionalmente, de un tiempo a esta parte se está adoptando un uso cada vez mayor de las tecnologías de red inalámbrica como complemento o

alternativa a la red Ethernet, principalmente para evitar el exceso de cableado en el interior del vehículo.

3.1.1. El bus CAN

El bus CAN es la arteria más crítica del sistema puesto que conecta las órdenes desde y hacia los elementos que controlan las acciones del vehículo, y por lo tanto requiere una especial atención.

Es un bus de transmisión compartido, de dos hilos, en el que las ECUs se comunican entre sí en modo punto-multipunto mediante la identificación del emisor de los datos, lo que significa que cada ECU que envía datos incluye su propia identificación en el mensaje para que el resto de ECUs reconozcan su procedencia para discriminar si deben procesar o no la información.

La siguiente ilustración muestra el formato de la trama de datos del protocolo CAN 2.0B [16]:

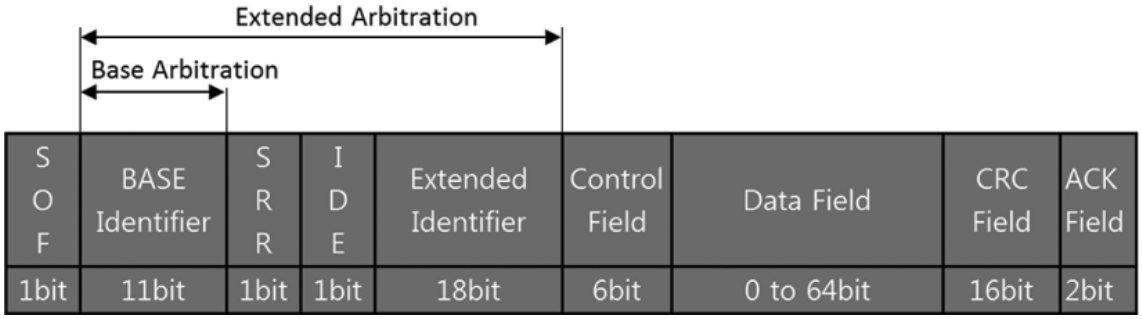


Ilustración 15. Formato de trama CAN 2.0B

En esa estructura se puede observar que existe un campo de 11 bits (“BASE Identifier”) destinado a la identificación del emisor -que puede extenderse con 18 bits adicionales-, una serie de campos de señalización y control de errores, y un campo de 0 a 8 bytes (“Data Field”) para la carga de datos útiles.

A pesar de que los mensajes incorporan el identificador del emisor, el protocolo CAN fue diseñado únicamente para entornos cerrados de red y por lo tanto no proporciona de forma intrínseca ningún mecanismo que asegure la confidencialidad ni la autenticidad de los datos.

En la sección Casos de estudio se tratará sobre los riesgos y las consecuencias que esto puede representar, y un aspecto importante para tener en cuenta será el conector OBD-II (On-Board Diagnostics 2), a través del cual las aplicaciones de diagnósticos (o de cualquier otro tipo) pueden acceder al BUS CAN utilizando el protocolo ELM327 con una conexión Bluetooth o USB para obtener información y/o manipular las diversas funciones del vehículo.

3.1.2. El conector OBD-II

A continuación, se presenta un listado ilustrativo de las funciones controlables a través del conector OBD-II [3]. No es un listado exhaustivo puesto que la implementación depende de cada fabricante:

- Airbags y sistemas de seguridad
- Asientos eléctricos
- Control de crucero
- Controles de climatización
- Estado de la batería
- Frenado antibloqueo (ABS)
- Iluminación exterior (faros, luces traseras, luces de giro)
- Indicadores de dirección
- Indicadores de fallos en el motor
- Indicadores de nivel de aceite
- Luces interiores (luz de cabina, luz del tablero)
- Nivel de combustible
- Nivel de emisiones
- Presión de los neumáticos
- Presión del aceite
- RPM del motor
- Sistema de audio (estéreo, altavoces)
- Temperatura ambiente
- Temperatura de los neumáticos
- Temperatura del aceite
- Temperatura del motor y del líquido de transmisión
- Velocidad del motor
- Velocidad del vehículo

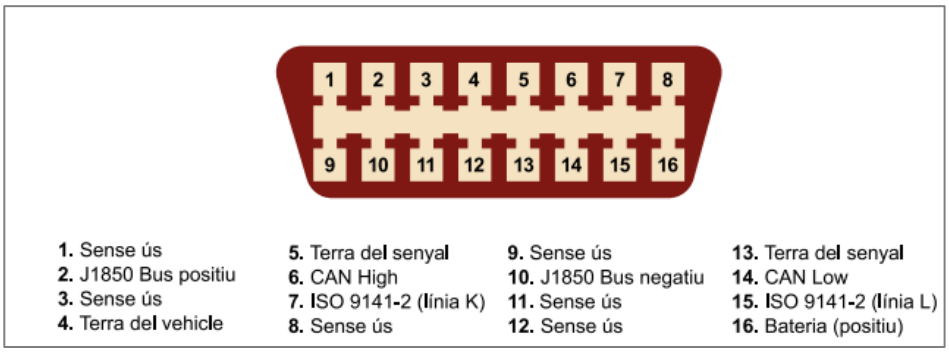


Ilustración 16. Terminales del conector OBD-II [17]

La información de diagnósticos que proporciona el conector OBD-II no es de uso exclusivo de los instrumentos utilizados en los talleres de reparación de coches, sino que también existen múltiples aplicaciones de smartphone para que cualquier conductor pueda monitorizar el estado de su vehículo.

Torque es una de las aplicaciones más populares en los mercados (*stores*) de Apple y Google, y ofrece un cuadro de mandos con un interfaz gráfico que se puede personalizar para mostrar en tiempo real la información que más le interese al usuario, por ejemplo:

- RPM del motor.
- Velocidad.
- Aceleración.
- Potencia del motor y par motor instantáneos.
- Códigos de error del motor con información detallada.
- Estado del sistema eléctrico y fusibles.
- Seguimiento del mantenimiento del vehículo.
- Lectura de las emisiones del vehículo.
- Temperatura de transmisión.

Ilustración 17. Cuadro de mando de una app de diagnósticos

La aplicación se conecta con el vehículo emparejándose por Bluetooth a un adaptador ELM327 que se enchufa al conector OBD-II (localizado normalmente debajo del salpicadero). Ese adaptador puede encontrarse por ejemplo en Amazon, por menos de 20€.

3.1.3. La OBU

Adicionalmente al bus CAN y la red Ethernet, la ilustración 14 muestra también una pieza clave en el sistema del vehículo conectado. Se trata de la **OBU** (On-Board Unit), y es el componente que integra ambas redes y otras de tipo inalámbrico (Wifi, Bluetooth) que pudiera haber, las conecta con el entorno exterior vía radio, y da soporte a la capa de aplicación, que se encarga de procesar toda la información entrante y saliente del vehículo a través de sus múltiples inferfaces [17]:

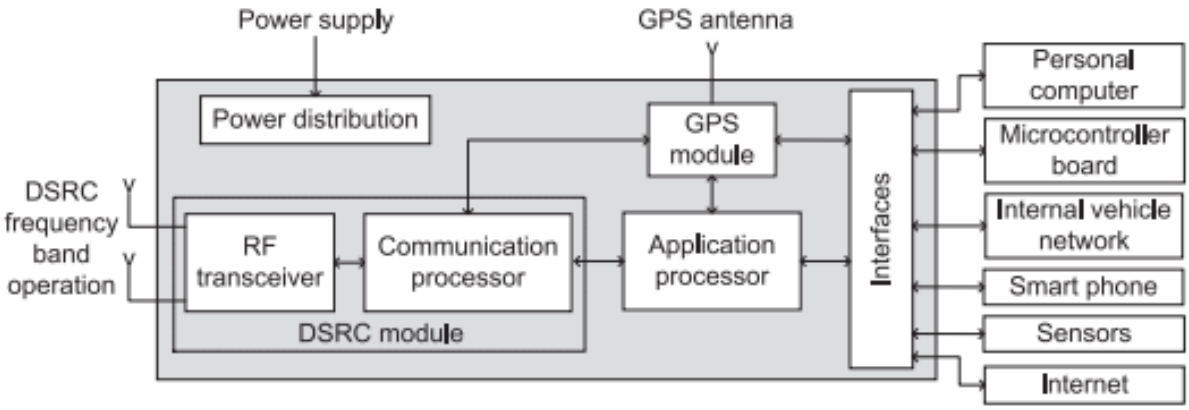


Ilustración 18. Plataforma genérica para comunicaciones V2X

La siguiente ilustración presenta la arquitectura de capas de la OBU:

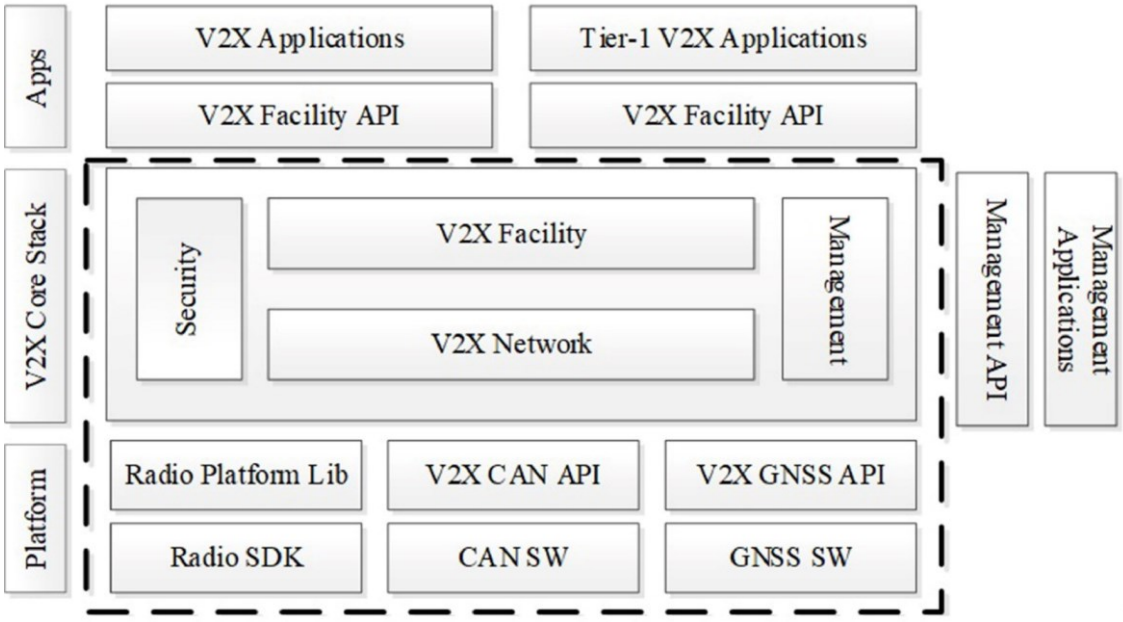


Ilustración 19. Arquitectura de la OBU [18]

La OBU está conectada al medio aire para recibir de forma constante mensajes procedentes de los vehículos a su alrededor (localización, velocidad, aceleración, estado de salud…), y mensajes procedentes del equipamiento de la vía (RSUs) con información sobre el estado y las señales de tráfico (luces de semáforos y tiempo restante, límites de velocidad, límites de altura…). Toda esa información es procesada por la pila de protocolos y es la base principal del sistema de seguridad activa del vehículo.

Toda la información recogida se está evaluando continuamente por un microprocesador que en base a unas reglas se encarga de decidir si existe una situación de peligro, y cómo gestionarla.

En el capítulo de anexos se incluyen las especificaciones técnicas de un modelo de OBU y de RSU de un determinado fabricante, como material de soporte para profundizar en sus funcionalidades y características. Se incluye también las especificaciones de un módulo que proporciona el mismo fabricante para la gestión centralizada de los datos recogidos a lo largo de la red de RSUs.

3.2. Tipología de amenazas

Con la apertura al mundo exterior a través de las comunicaciones móviles, queda expuesta una nueva superficie de ataque frente a un conjunto de amenazas de ciberseguridad, que podríamos agrupar de la siguiente manera [7]:

- **Amenazas a la disponibilidad**: Este tipo de amenazas intentan interrumpir el comportamiento continuo de un sistema ITS. La subcategoría más común es el ataque de denegación de servicio (DoS), que es muy difícil de proteger. En estos ataques, se genera de manera malintencionada y artificial un alto volumen de mensajes falsos. Esto tiene un impacto significativo en los sistemas ITS porque las aplicaciones de seguridad requieren respuestas en tiempo real, así como garantías de tiempo real. Estos ataques normalmente hacen que una estación ITS no funcione correctamente (recibir, responder, transmitir, producir y enviar mensajes).
- **Amenazas a la integridad**: La integridad de la información se ve comprometida cuando se accede sin autorización, se pierde, se manipula o se corrompe. La información restringida incluye toda la información que está asociada específicamente con un usuario o una estación ITS. La mayoría de los ataques se basan en la suplantación de identidad a través de ataques enmascarados. Normalmente, en este tipo de ataques, la información no llega al destino final y, por lo tanto, se pierde. Además, usando una interfaz ITS-G5, un mensaje puede cambiarse antes de ser enviado para interferir con el protocolo y corromper la información.
- **Amenazas a la autenticidad**: Asegurar la autenticidad de la información es de suma importancia, ya que todas las estaciones ITS tienen la capacidad de enviar, recibir y responder a todos los mensajes básicos. Por ejemplo, en ITS, tener vehículos que se hagan pasar por vehículos de alta prioridad (como ambulancias o camiones de bomberos) puede afectar fácilmente el tráfico e incluso crear caos.
- **Amenazas a la confidencialidad**: Uno de los problemas de usar interfaces abiertas es que los mensajes transmitidos pueden ser interceptados y la información extraída. Debe recordarse que los mensajes de seguridad contienen información sobre el estado del vehículo y, en algunos casos, sobre el usuario. Evitar problemas de privacidad es uno de los principales motivadores para las comunicaciones vehiculares seguras.
- **No repudio**: En escenarios legales, la capacidad de demostrar la ocurrencia de un cierto evento para evitar que los usuarios nieguen su acción es crucial. Garantiza que el comportamiento erróneo pueda (si es necesario) ser perseguido legalmente. Por ejemplo, aunque la confidencialidad del usuario se asegura a través del uso de seudónimos, una parte autorizada puede solicitar la información del usuario al emisor.

La siguiente ilustración muestra ejemplos concretos de amenazas y las funciones y entidades del sistema afectadas en función de la red de comunicaciones utilizada: OBUs o unidades embarcadas en el vehículo (V), RSUs o unidades instaladas en la vía (R), servidores de aplicaciones (AS), y funciones de seguridad (SF), entre algunas otras:

Attacks and Threats	Types	Network	Involved and Affected Entities/ Functions
Authentication and Authorization Attacks	Brute Force, Weak Validation, Access violation, Session control, Broken Authentication, ACL Modification	DSRC	V, R
		V2X, NS-5G-V2X	V, eNB, MME, AS, HSS, S-GW
		5G-V2X	V, gNB, SMF, AMF, UPF, AF
Malicious Node Attacks	Black-Hole. Grey-Hole, Sink Hole Attacks	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, MME, AS
		5G-V2X	V, gNB, SMF, AMF, UPF, AF
Certificate Forgery	Replication, Duplication, Modification, Alteration	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, MME, AS, HSS, S-GW
		5G-V2X	V, gNB, SMF, AMF, UPF, AF
Channel Interference	Noise, Jamming, Signal Storming, covert and overt channels	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, MME, AS
		5G-V2X	V, gNB, SMF, AMF, UPF, AF
Cipher Text /Plain Text Attacks	Known and Chosen	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, MME, AS, HSS, S-GW
		5G-V2X	V, gNB, SMF, AMF, UPF, AF
Data Deletion, Data Disclosing, Data Forgery and Distributions	Replication, Duplication, Modification, Alteration	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, MME, AS, HSS, S-GW
		5G-V2X	V, gNB, SMF, AMF, UPF, AF, SF
De-Synchronization Attacks	TCP De-Synchronization, DNS poisoning, Port identification, ICMP attacks	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, MME
		5G-V2X	V, gNB, SMF, AMF
DoS and DDoS Attacks	UDP Flood, SYN Flood, Ping of Death	DSRC	V, R, AS
		LTE-V2X, NS-5G-V2X	V, eNB, MME, AS, BS
		5G-V2X	V, gNB, SMF, AMF
Access Attacks	Eavesdropping, Impersonation, Man-in-the-Middle, Masquerade Attack	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, MME
		5G-V2X	V, gNB, SMF, AMF
Fabrication Attacks	Falsified Information Injection, Falsified Sensor readings and Misinterpretations	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB
		5G-V2X	V, gNB, SMF, AMF
Terminals Attacks	Hidden Terminals and Exposed Terminals	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB
		5G-V2X	V, gNB, SMF, AMF
Key Exploitation	-	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, MME, AS, HSS, S-GW
		5G-V2X	V, gNB, UPF, AF, SMF, AMF, SF
Message Modification and Tampering	Content Modification and Header Modification, SQL Injections, Code obfuscation	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, AS, MME
		5G-V2X	V, gNB, UPF, AF, SMF, AMF,
Network Stalking and Penetration Attacks	Sniffing, Forensics, Spoofing, Spamming	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, AS, MME
		5G-V2X	V, UPF, gNB, SMF
Reprogramming Attacks	Cloning attacks, Code obfuscation, XSS-scripting	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB
		5G-V2X	V, gNB, SMF, AMF
Resource Depletion Attacks	-	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, AS, MME, BS
		5G-V2X	V, gNB
Routing Attacks	Topology-based, Resources-based, Traffic-based	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, MME, AS, BS
		5G-V2X	V, UPF, gNB, SMF, AMF
Service based network Prevention and Session Hijacking	-	DSRC	-
		LTE-V2X, NS-5G-V2X	V, AS, eNB
		5G-V2X	V, UPF, AF, gNB, SMF, AMF
Side Channel Attacks	Cache attack, Timing attack, Power-monitoring attack and Electromagnetic attack, Acoustic attack	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB
		5G-V2X	V, gNB
Zero-day	Exterior and Interior	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, MME, AS, HSS, S-GW
		5G-V2X	V, UPF, gNB, SMF, AMF, SF
Timing Attacks	Message Connect, Service-Access based, Range-based, Replay Attacks	DSRC	V, R
		LTE-V2X, NS-5G-V2X	V, eNB, MME
		5G-V2X	V, gNB, SMF, AMF
Tunneling Attacks	ICMP, DNS, Port, HTTP	-	-
		LTE-V2X, NS-5G-V2X	V, eNB, MME, AS, HSS, S-GW
		5G-V2X	V, UPF, gNB, SMF, AMF, SF

Ilustración 20. Tipología de ataques, y funciones amenazadas [19]

3.3. La cadena de ciberataque

Los ataques cibernéticos siguen una cadena de procesos que suelen llevar tiempo y que se desarrollan en varias etapas.

Si un equipo de ciberdefensa es capaz de entender cómo se lleva a cabo un ataque, podrá anticiparse a las diferentes etapas de la cadena y tomar medidas para frustrarlo.

Por ejemplo, si un atacante está tratando de penetrar en una red (vehicular o corporativa), primero llevará a cabo una fase de reconocimiento para identificar las vulnerabilidades de la red.

Si el equipo de ciberdefensa es capaz de identificar los patrones de actividad durante esta fase, podrá tomar medidas para remediar las vulnerabilidades conocidas y evitar que el atacante tenga éxito en la siguiente fase del ataque.

Las etapas más comunes en una cadena de ataque son las siguientes:

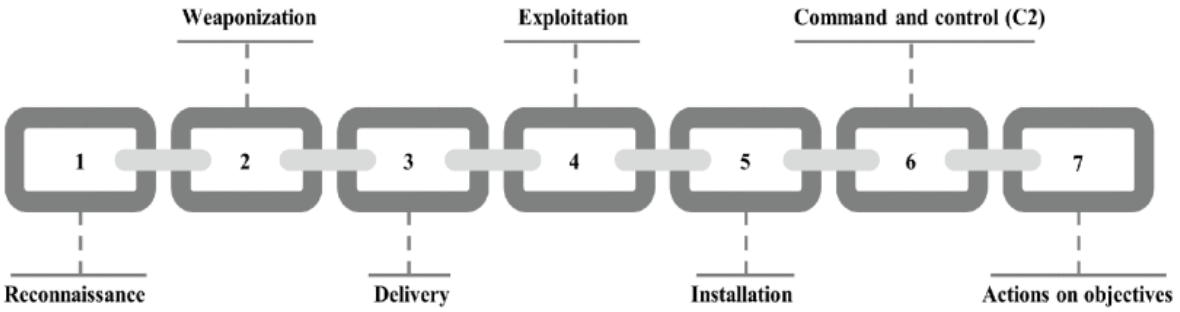


Ilustración 21. La cadena de ciberataque (Cyber Kill Chain) [20]

1. *Reconnaissance* (reconocimiento): el atacante recopila información sobre el objetivo del ataque, como el tipo de sistema, la infraestructura y los empleados.
2. *Weaponization* (creación del armamento): el atacante utiliza la información recopilada durante la fase de reconocimiento para construir malware u otras herramientas o armas de ataque.
3. *Delivery* (entrega): una vez que el atacante ha obtenido acceso al sistema, trata de introducir el objeto atacante y posteriormente poder expandir su acceso y control a otros sistemas dentro de la red.
4. *Exploitation* (explotación): el atacante utiliza el arma introducida para explotar la vulnerabilidad con la intención de obtener privilegios elevados para poder acceder más adelante a información crítica o sistemas de mayor importancia.

5. *Installation* (instalación): el atacante busca extraer información confidencial o dañar los sistemas desde el interior de la red.
6. *Command* and Control (mantenimiento de acceso): una vez que el atacante ha logrado penetrar en la red, busca mantener su acceso para futuros ataques o para seguir exfiltrando información.
7. *Actions on objectives* (limpieza de huellas): el atacante trata de borrar cualquier evidencia de su presencia en la red para evitar ser detectado.

La imagen a continuación señala las etapas donde el adversario se expone a ser detectado [21]:

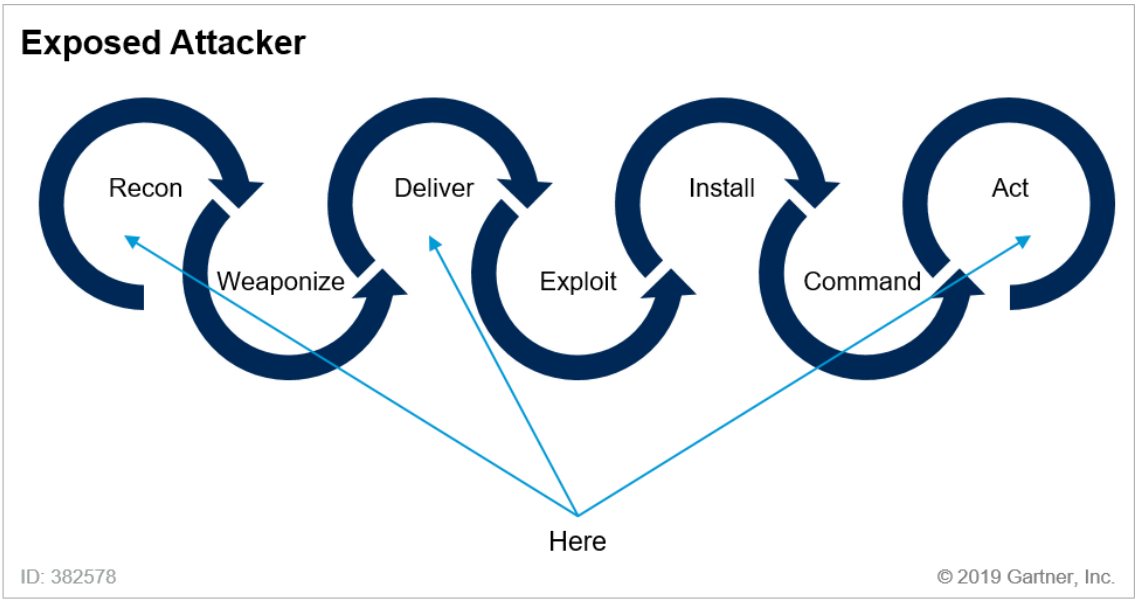
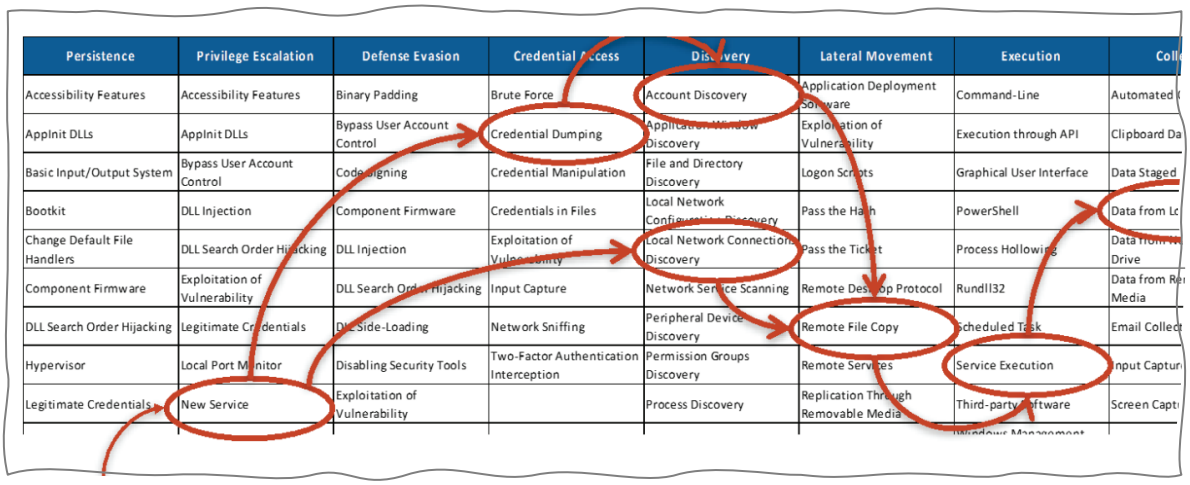


Ilustración 22. Etapas donde se expone el atacante. Fuente: Gartner

3.4. Tácticas, técnicas y procedimientos

Existen diferentes organizaciones dedicadas al análisis de los patrones de ciberdelincuencia y a la identificación de amenazas de seguridad cibernética. Estas organizaciones recopilan y analizan información sobre las tácticas, técnicas y procedimientos utilizados por los ciberdelincuentes para crear marcos de inteligencia de amenazas y herramientas para ayudar a las organizaciones a defenderse contra los ataques cibernéticos.

MITRE (https://attack.mitre.org/) es la organización de referencia en este ámbito por su trabajo en la matriz de ataque, una herramienta estandarizada de evaluación de amenazas que proporciona una estructura detallada de las tácticas, técnicas y procedimientos utilizados por los atacantes.



Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Coll
Accessibility Features	Accessibility Features	Binary Padding	Brute Force	Account Discovery	Application Deployment Software	Command-Line	Automated
AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Application Window Discovery	Exploitation of Vulnerability	Execution through API	Clipboard Da
Basic Input/Output System	Bypass User Account Control	Code Signing	Credential Manipulation	File and Directory Discovery	Logon Scripts	Graphical User Interface	Data Staged
Bootkit	DLL Injection	Component Firmware	Credentials in Files	Local Network Configuration Discovery	Pass the Hash	PowerShell	Data from Lo
Change Default File Handlers	DLL Search Order Hijacking	DLL Injection	Exploitation of Vulnerability	Local Network Connections Discovery	Pass the Ticket	Process Hollowing	Data from N Drive
Component Firmware	Exploitation of Vulnerability	DLL Search Order Hijacking	Input Capture	Network Service Scanning	Remote Desktop Protocol	Rundll32	Data from Re Media
DLL Search Order Hijacking	Legitimate Credentials	DLL Side-Loading	Network Sniffing	Peripheral Device Discovery	Remote File Copy	Scheduled Task	Email Collect
Hypervisor	Local Port Monitor	Disabling Security Tools	Two-Factor Authentication Interception	Permission Groups Discovery	Remote Services	Service Execution	Input Captur
Legitimate Credentials	New Service	Exploitation of Vulnerability		Process Discovery	Replication Through Removable Media	Third-party Software	Screen Capt
						Windows Management	

Ilustración 23. La matriz MITRE ATT&CK

La matriz **MITRE ATT&CK** es una herramienta que se utiliza para describir y clasificar las tácticas y técnicas que los atacantes utilizan para comprometer la seguridad de los sistemas informáticos. Básicamente, es una lista de los diferentes pasos y acciones que los atacantes pueden tomar para acceder a sistemas y redes, robar información, dañar los sistemas, etc.

Esta matriz es útil para los equipos de ciberseguridad porque les ayuda a entender cómo piensan los atacantes y les orienta sobre lo que deben buscar para detectar y prevenir los ataques.

La matriz MITRE ATT&CK se divide en dos categorías principales: **tácticas y técnicas**. Las tácticas son los objetivos generales que los atacantes buscan lograr durante un ataque, mientras que las técnicas son las acciones específicas que los atacantes utilizan para lograr esas tácticas. Por ejemplo, una táctica común utilizada por los atacantes es el "movimiento lateral", que se refiere al proceso de moverse de un sistema comprometido a otro dentro de la red. Una técnica común utilizada para lograr esta táctica es el "robo de credenciales", que implica la obtención de contraseñas y nombres de usuario para obtener acceso no autorizado a otros sistemas.

Además de las tácticas y técnicas, la matriz MITRE ATT&CK también proporciona detalles sobre los "**procedimientos**" utilizados por los atacantes para llevar a cabo un ataque. Los procedimientos son una serie de técnicas que los atacantes pueden utilizar en un orden específico para lograr sus objetivos. Por ejemplo, un procedimiento común utilizado por los atacantes para eludir la detección es "ocultar archivos y directorios", que implica ocultar los archivos y directorios comprometidos para evitar ser detectados por los sistemas de seguridad.

El equipo de ciberdefensa puede utilizar la matriz MITRE ATT&CK de varias maneras, en primer lugar, para entender mejor cómo los atacantes llevan a cabo los ataques y cómo se mueven a través de la red, y así puede detectar patrones de actividad maliciosa y prevenir futuros ataques; y también para desarrollar planes de seguridad efectivos, que incluyen la implementación de controles de seguridad adicionales para prevenir las tácticas y técnicas identificadas.

MITRE proporciona juegos de matrices diferenciados por plataforma:

- Enterprise: Windows, macOS, Linux, PRE, Azure AD, Office 365, Google Workspace, SaaS, IaaS, Network y Containers
- Mobile: Android e iOS;
- ICS: Industrial Control Systems

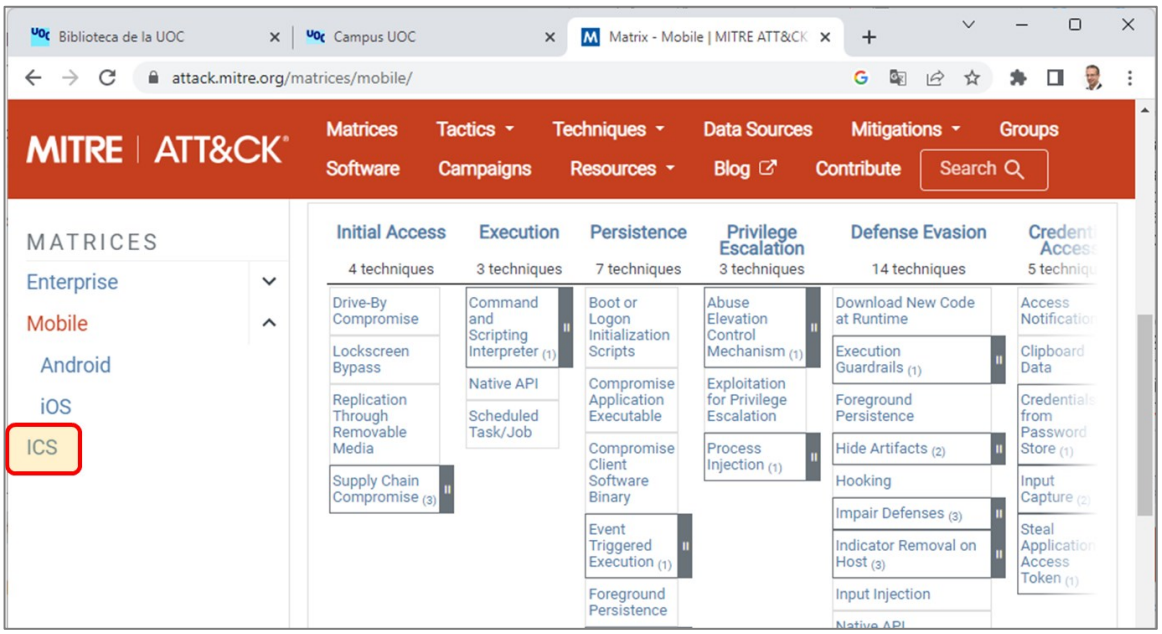


Ilustración 24. Juegos de matrices MITRE ATT&CK

En la categoría ICS dispone de una matriz específica para la plataforma del vehículo conectado (“ICS for Vehicular Networks”) a la que se accede con autorización a través de su portal de colaboradores.

3.5. Casos de estudio

Esta sección expone dos casos de estudio categorizando las técnicas utilizadas según la matriz de ataque de MITRE.

3.5.1. Reempaquetado de aplicaciones Android

En este caso de estudio publicado por *Hindawi* [23] el atacante utiliza la técnica de reempaquetado de un archivo de instalación APK (Android Package Kit) para alterar la estructura de la aplicación Android original introduciendo código malicioso o modificando su comportamiento.

La imagen siguiente ilustra el proceso una vez la falsa aplicación ha sido publicada en el Play Store. A partir de ese momento queda disponible para que cualquier usuario la descargue en su dispositivo móvil pensando que se trata de la legítima aplicación de diagnósticos y la ponga en marcha utilizando un simple adaptador ELM327 que va enchufado al conector OBD-II.

Una vez conectada la aplicación (vía Wifi o Bluetooth) a este adaptador ya habrá conseguido acceso al bus CAN y por consiguiente a todas las ECUs del vehículo:

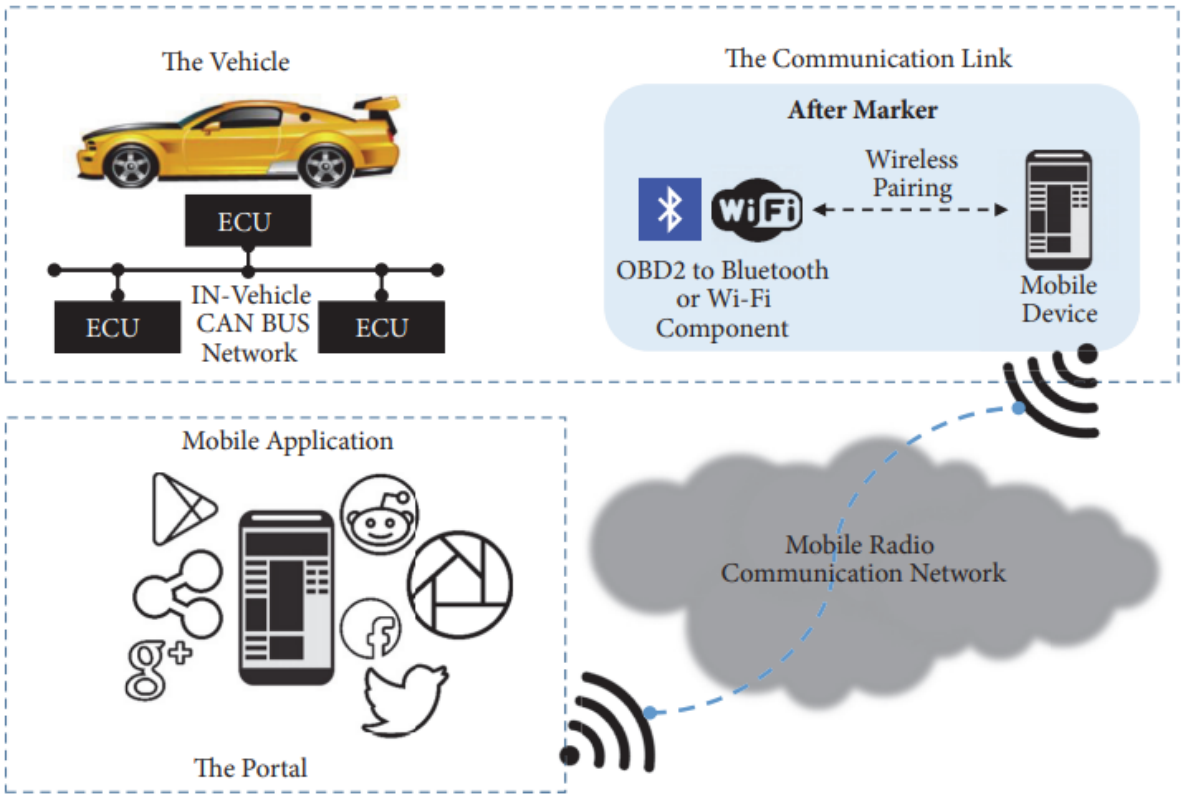


Ilustración 25. Despliegue de la aplicación de diagnósticos

En este caso de estudio, se asume que la víctima se ha descargado la falsa aplicación de diagnósticos y la ha conectado al bus CAN emparejándola por Bluetooth con el adaptador ELM327.

A continuación, se analiza cada etapa según el modelo de la cadena de ciberataque.

Etapa 1: Reconnaissance

En la etapa de reconocimiento, el atacante analiza el protocolo ELM327, observando que el proceso de intercambio de información entre el smartphone y el vehículo se inicia con un comando AT (Attention) enviado desde el smartphone al chip ELM327 y se desarrolla en un ciclo de peticiones de parámetros (OBD PIDs) y sus correspondientes respuestas por parte de las ECUs:

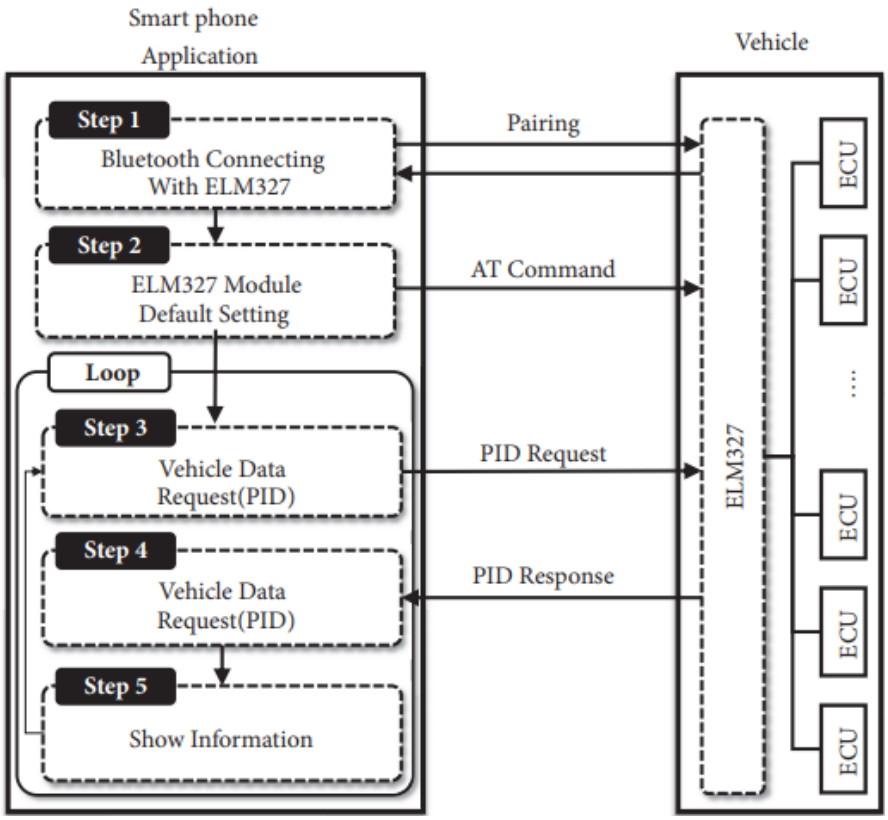


Ilustración 26. Modelo de comunicación ELM327 entre el smartphone y el vehículo

En el capítulo de anexos se encuentra la lista de comandos completa, extraída del manual publicado por el fabricante del chip.

Los comandos AT (*Attention*) del protocolo ELM327 permiten crear un entorno de vehículo conectado, y controlarlo.

El adaptador utiliza un CAN ID (identificador del ECU emisor) fijo para comunicarse con el OBD-II, pero tal como muestra la imagen siguiente, este ID se puede modificar mediante el comando AT SH:

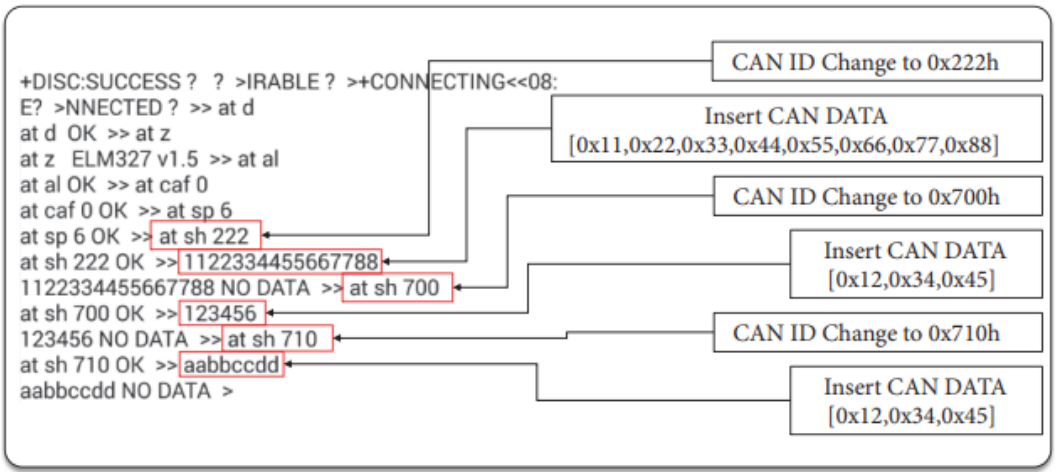


Ilustración 27. Comunicación remota con el adaptador ELM327

El paso siguiente en la etapa de reconocimiento consiste en seleccionar una determinada aplicación de diagnósticos y analizar sus vulnerabilidades.

Las conclusiones de ese análisis indican que el código de la aplicación no está ofuscado y se puede desensamblar fácilmente, y que las cadenas de caracteres utilizadas para el envío de los comandos AT y los parámetros OBD PID quedan expuestas en texto plano.

Etapa 2: Weaponization

Una vez desensamblado el código de la aplicación, el adversario localiza los comandos AT y los OBDs PID utilizados por la misma, para poder analizar la lógica de funcionamiento:

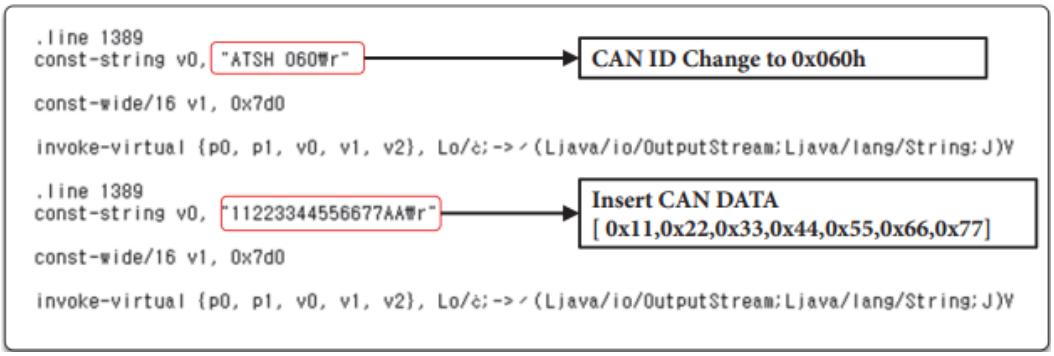


Ilustración 28. Detección de las tramas CAN en el código original de la app

La herramienta de ataque se puede programar o bien para que actúe en el instante en que se inicia la aplicación de diagnósticos, o bien cuando se detecte una determinada condición mediante la escucha continua del valor de un parámetro, por ejemplo, para detectar el momento en el que el vehículo alcanza una determinada velocidad:

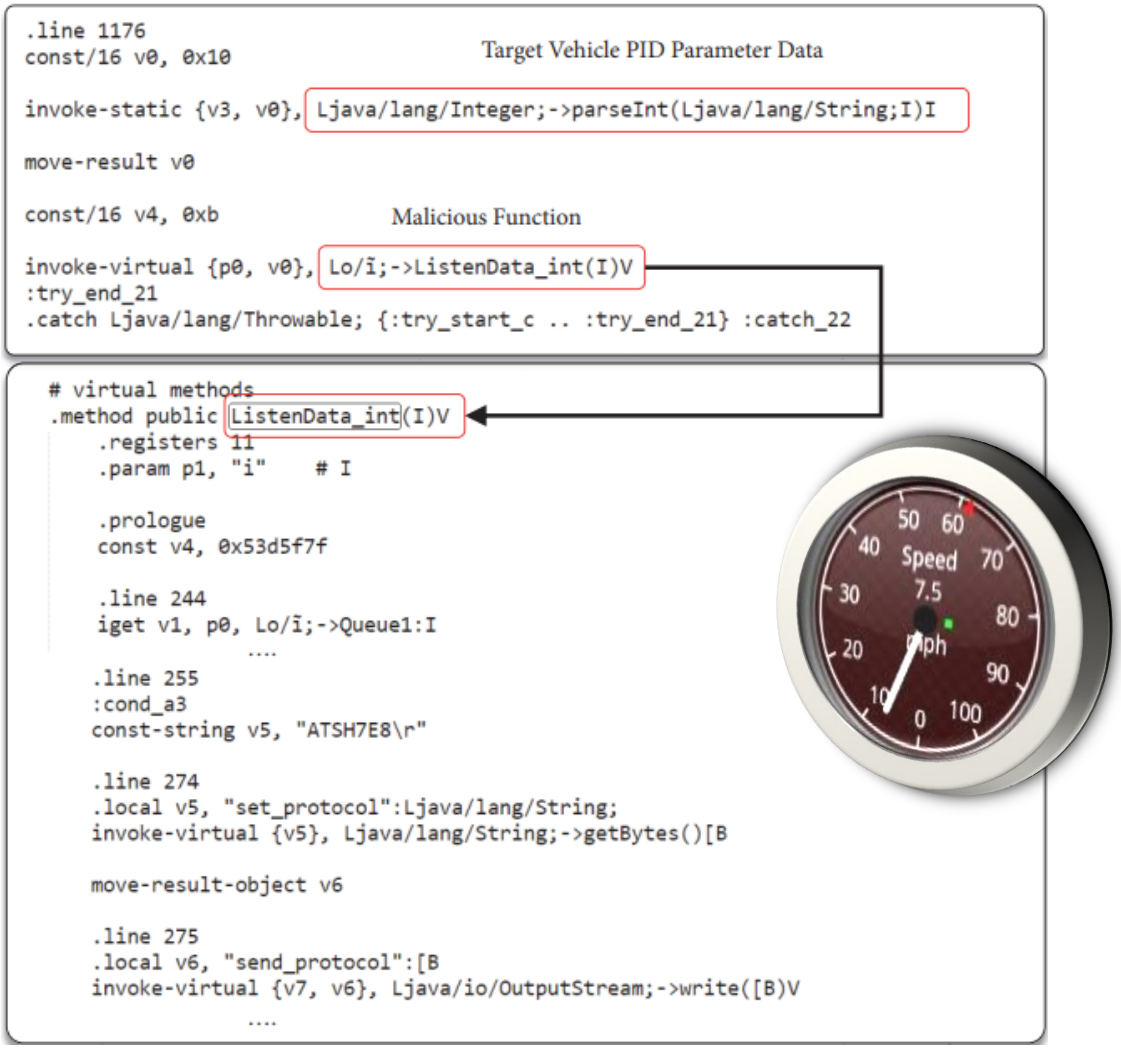


Ilustración 29. Inserción del código malicioso en la app

Etapa 3: Delivery

Una vez introducido el código malicioso y recompilada la aplicación, la etapa de Delivery consiste en distribuirla a través del *Marketplace*.

Etapa 4: Exploitation

En esta etapa el usuario se descarga en su smartphone la aplicación del *Marketplace*, sin darse cuenta de que es falsa.

Etapa 5: Installation

Esta es la etapa que corresponde a la instalación de la app en el smartphone del usuario, y al montaje del adaptador ELM327 en su coche.

Etapa 6: Command and Control

En esta etapa tiene lugar la toma de control del objetivo por parte del adversario, y en este caso se distinguen 3 posibles escenarios de ataque:

- Desbloqueo de puertas. Cuando la víctima abandona su vehículo, el adversario puede desbloquear las puertas utilizando su aplicación de smartphone y acceder al interior con cualquier fin malicioso.
- Parada del motor al inicio. El código introducido en la aplicación fuerza la parada del motor en el instante en que se pone en marcha la aplicación de diagnósticos.
- Parada del motor durante la marcha. Se fuerza la parada del motor cuando el vehículo alcanza una determinada velocidad.

Etapa 7: Actions on objectives

En esta última etapa es donde se materializa la finalidad del ataque. Según los escenarios descritos en el punto anterior, el atacante podrá perpetrar el robo o bien forzar el bloqueo del vehículo para provocar un accidente de tráfico.

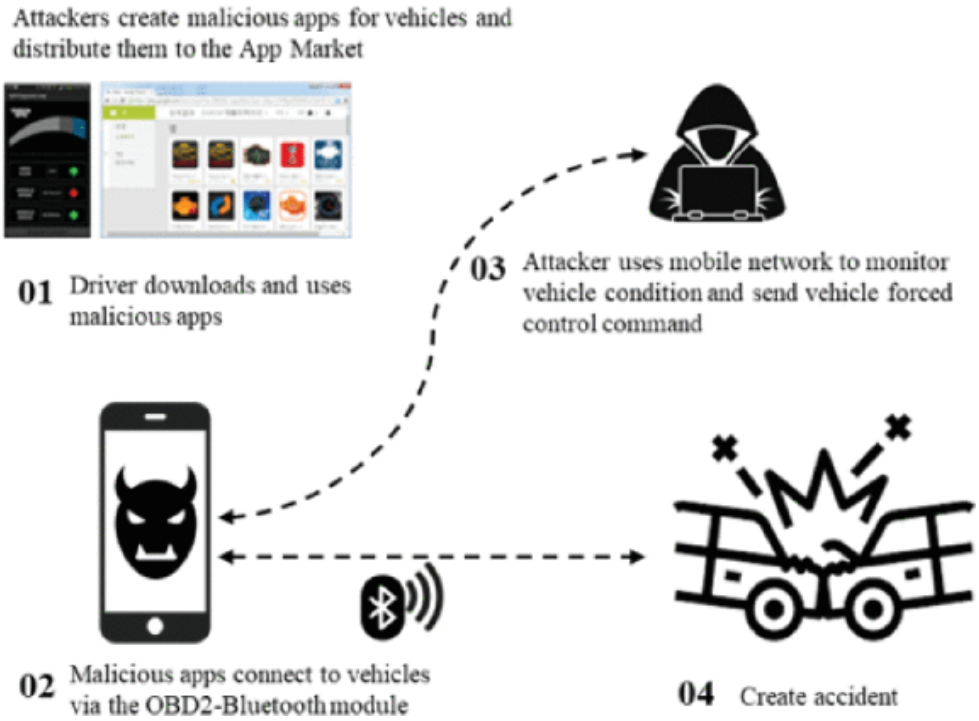


Ilustración 30. Modelo de ataque con app maliciosa

3.5.2. Distribución de Ransomware

En este otro caso, referenciado por la revista IEEE Access *[21]*, se describe un ejemplo de Ransomware que puede penetrar en el vehículo infectando un servicio web utilizado por los sistemas de *Infotainment*.

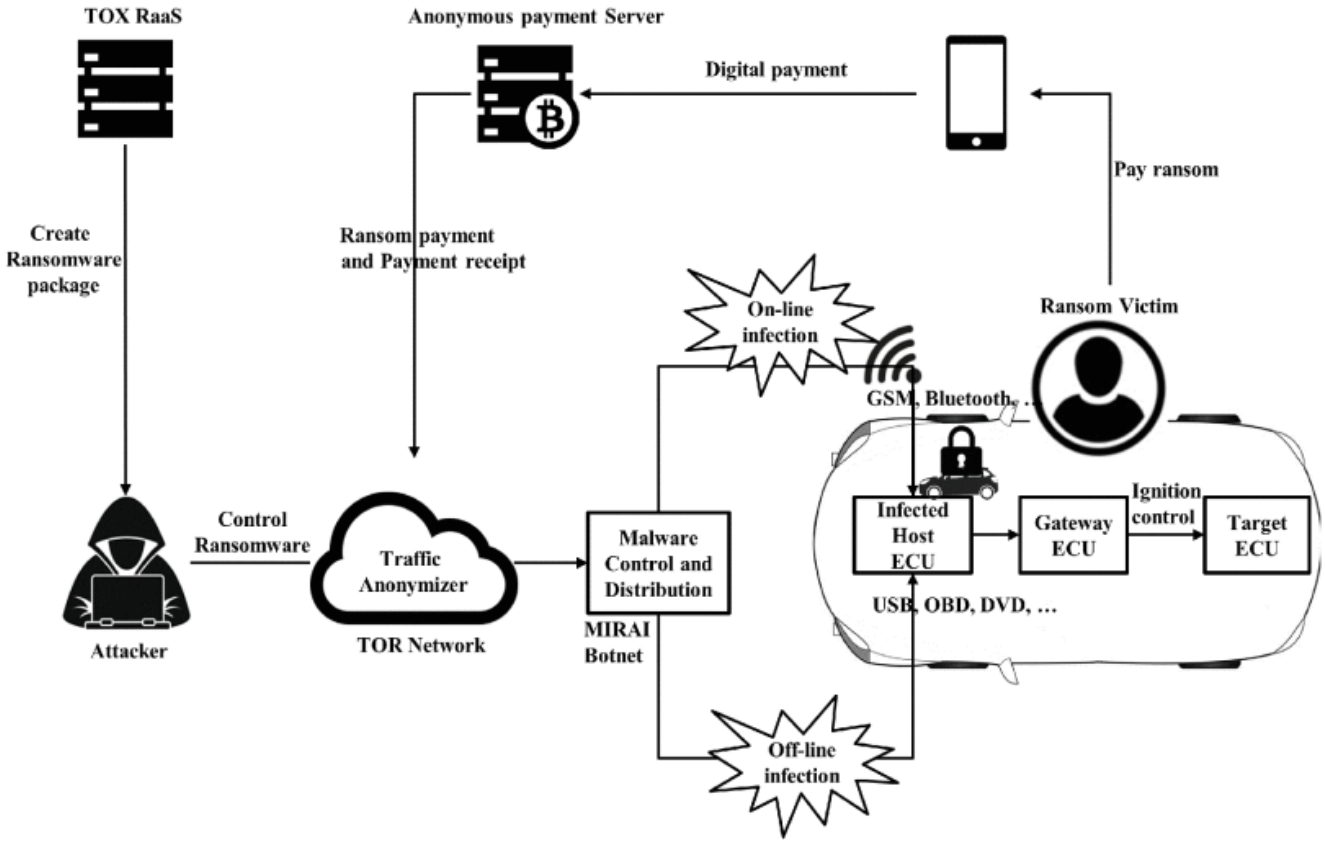


Ilustración 31. Distribución de Ransomware

De forma esquemática, los pasos seguidos en la cadena de ataque son los siguientes

1- **Reconnaissance**: Se analizan las vulnerabilidades de los dispositivos IoT desplegados en los vehículos.

2- **Weaponization**: Se utiliza un RaaS (*Ransomware as a Service*) como TOX o STAMP para crear el *malware* de manera sencilla, puesto que estos servicios incluyen utilidades para la encriptación de archivos, la gestión de las claves, la gestión del pago con Bitcoins, y también funciones de *botmaster* para el control de la distribución del *ransomware*.

3- **Delivery**: Distribución del *ransomware* a través de sistemas de botnets basados en TOR, como por ejemplo MIRAI. Estos botnets no infectarán directamente el vehículo, sino los servicios web que consume su sistema de *Infotainment*.

4- **Exploitation**: Despliegue del *ransomware* a través del sistema de Infotainment.

5- **Installation**: Ejecución automática del ransomware al encender los sistemas del vehículo.

6- **Command and Control**: Bloqueo de componentes críticos para el funcionamiento del vehículo.

7- **Actions on Objectives**: Pago del rescate por parte de la víctima.

4. Diseño de la solución

4.1. Enfoque

Con acuerdo al razonamiento expuesto en las conclusiones del estudio del estado del arte, este capítulo se enfoca al análisis de la arquitectura de un Centro de Operaciones de Seguridad para el Vehículo (VSOC), puesto que este es uno de los 5 aspectos exigidos por la regulación UN ECE WP.29, y es la base sobre la cual los equipos de ciberdefensa pueden implementar y mantener actualizados sus procesos de detección temprana de amenazas, e investigar los incidentes y tomar las medidas necesarias con la máxima eficacia.

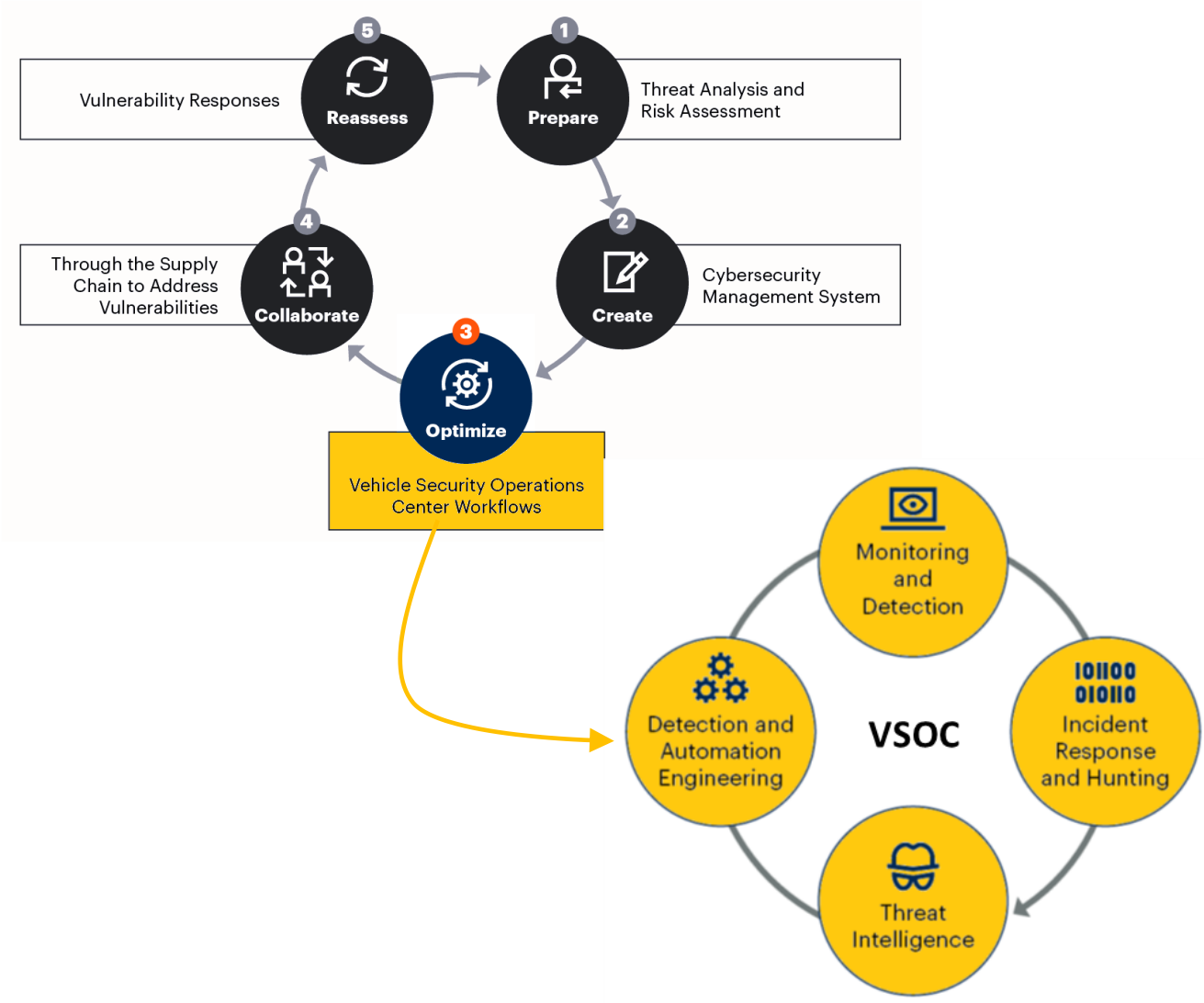


Ilustración 32. Tercer requisito UN ECE WP.29 R155: VSOC

4.2. Capacidades del VSOC

Gartner [24] define las capacidades de un SOC en base a la siguiente matriz, estructurada por grupos de funciones y niveles de relevancia:

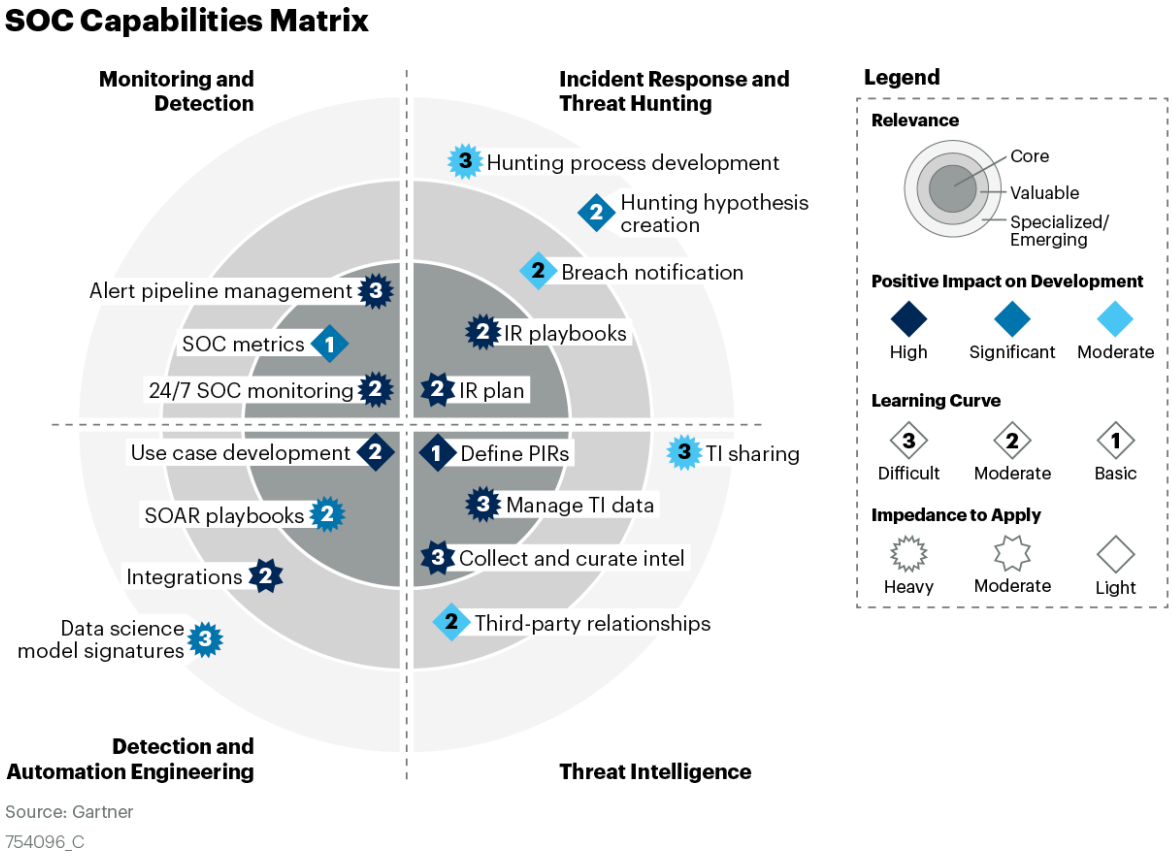


Ilustración 33. Matriz de capacidades de un SOC. Fuente: Gartner

Las funciones que debe desempeñar un SOC Vehicular (VSOC) se estructuran bajo ese mismo modelo:

4.2.1. Monitorización y detección

Esta es la función esencial de cualquier SOC, y en el contexto del VSOC implica la vigilancia continua de los sistemas de los sistemas vehiculares y de transporte para identificar posibles amenazas y vulnerabilidades. Las actividades en esta área incluyen:

a. Revisión de registros y eventos de seguridad a través del SIEM (*Security Information and Event Management*).

b. Análisis de tráfico de red y detección de anomalías en sistemas de comunicación vehicular.

c. Monitoreo de endpoints y sistemas de control de vehículos (ECUs, sistemas de navegación, etcétera).

d. Evaluación de vulnerabilidades y gestión de parches.

4.2.2. Respuesta a incidentes y Threat Hunting

Esta función se centra en responder a incidentes de seguridad en vehículos y sistemas de transporte, y en buscar amenazas de forma proactiva. Las actividades en esta área incluyen:

a. Investigación y análisis de incidentes de seguridad.

b. Contención y erradicación de amenazas.

c. Recuperación y remediación post-incidente.

d. Threat hunting proactivo para identificar y neutralizar amenazas antes de que causen daños.

4.2.3. Ingeniería para la detección y la automatización

Esta función se centra en el desarrollo y mantenimiento de herramientas, tecnologías y procesos para mejorar la eficiencia y efectividad de las operaciones del VSOC. Las actividades en esta área incluyen:

a. Diseño y despliegue de herramientas y sistemas de seguridad.

b. Integración y automatización de procesos de seguridad y respuesta a incidentes.

c. Desarrollo de soluciones personalizadas para abordar necesidades específicas del entorno vehicular que se está protegiendo en concreto.

d. Mantenimiento y mejora continua de las herramientas y procesos existentes.

4.2.4. Threat Intelligence

Esta función implica la recopilación, análisis y compartición de información sobre amenazas en el ámbito vehicular y de los sistemas de transporte, para anticipar y defenderse contra ataques de ciberseguridad. Las actividades a desempeñar en esta área incluyen:

a. Recopilación de información sobre amenazas de fuentes abiertas y privadas.

b. Análisis de la información recopilada para identificar patrones y tendencias.

c. Creación y mantenimiento de un repositorio de inteligencia sobre amenazas.

d. Compartir información de inteligencia sobre amenazas con otras organizaciones y entidades de seguridad del sector automotriz y de transporte.

Las fuentes de *Threat Intelligence* deben proporcionar una amplia cantidad de información, que incluya: actores maliciosos y sus patrones de comportamiento, tácticas, técnicas y procedimientos (TTP), campañas de ataque, vulnerabilidades y exposiciones conocidas (CVE), e indicadores de compromiso (IoC) de los ataques registrados y formalmente documentados.

La mayor base de datos de CVEs a nivel mundial es la que gestiona MITRE (https://cve.mitre.org/), con prácticamente 200.000 registros en el momento de escribir esta memoria, pero su taxonomía no permite distinguir los que corresponden específicamente a la seguridad del entorno vehicular y del transporte.

Lo mismo ocurre con la NVD (*National Vulnerability Database*), gestionada por el NIST (*National Institute of Standards and Technology*), cuya función consiste en utilizar la base de datos de MITRE para aportar un detallado análisis de impacto sobre cada vulnerabilidad, incluyendo la puntuación CVSS (*Common Vulnerability Scoring System*) sobre su nivel de gravedad, y proporcionando enlaces a las soluciones o parches de seguridad las pueden remediar.

Debido a esa circunstancia se hace necesaria la existencia de organizaciones como Auto-ISAC (Automotive Information Sharing and Analysis Center), creada en 2015 por los fabricantes de automóviles para compartir y analizar la inteligencia sobre los riesgos de ciberseguridad en la industria del automóvil. Auto-ISAC proporciona además una serie de principios basados en buenas prácticas sobre la gestión de la ciberseguridad [25]:

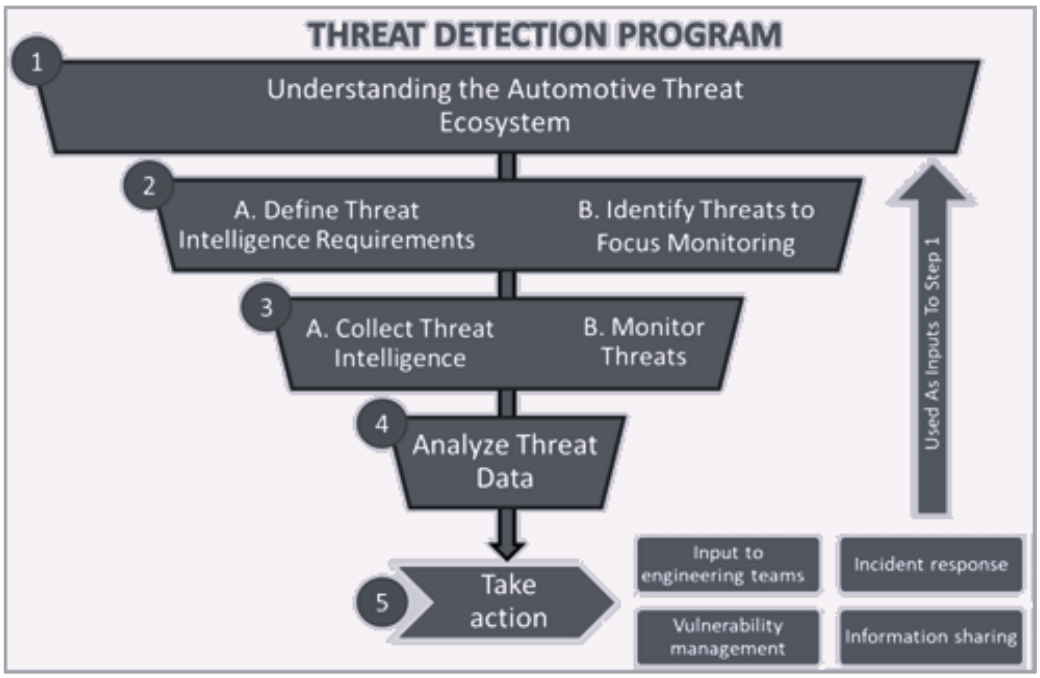


Ilustración 34. Threat detection best practice

4.3. Arquitectura

4.3.1. Requisitos de arquitectura

Para cubrir las funciones del VSOC con acuerdo a las buenas prácticas mencionadas en el capítulo anterior es necesario desplegar una arquitectura que cubra los siguientes requisitos:

- Plataforma SIEM para recopilar, analizar y correlacionar eventos de seguridad en tiempo real, identificar actividades sospechosas y generar alertas.
- Inteligencia de amenazas: capacidad de integración de *feeds* de *Threat Intelligence* en la plataforma SIEM para obtener información actualizada sobre ciberamenazas en el entorno vehicular; y participación en comunidades y grupos de intercambio de información sobre amenazas.
- Automatización y orquestación: Implantación de soluciones de SOAR para automatizar procesos de seguridad, mejorar la eficiencia del VSOC y facilitar la respuesta a incidentes.
- Respuesta a incidentes y *threat hunting*: Uso de herramientas que permitan buscar activamente amenazas en los sistemas vehiculares y las redes de comunicación y ejecutar acciones de respuesta.

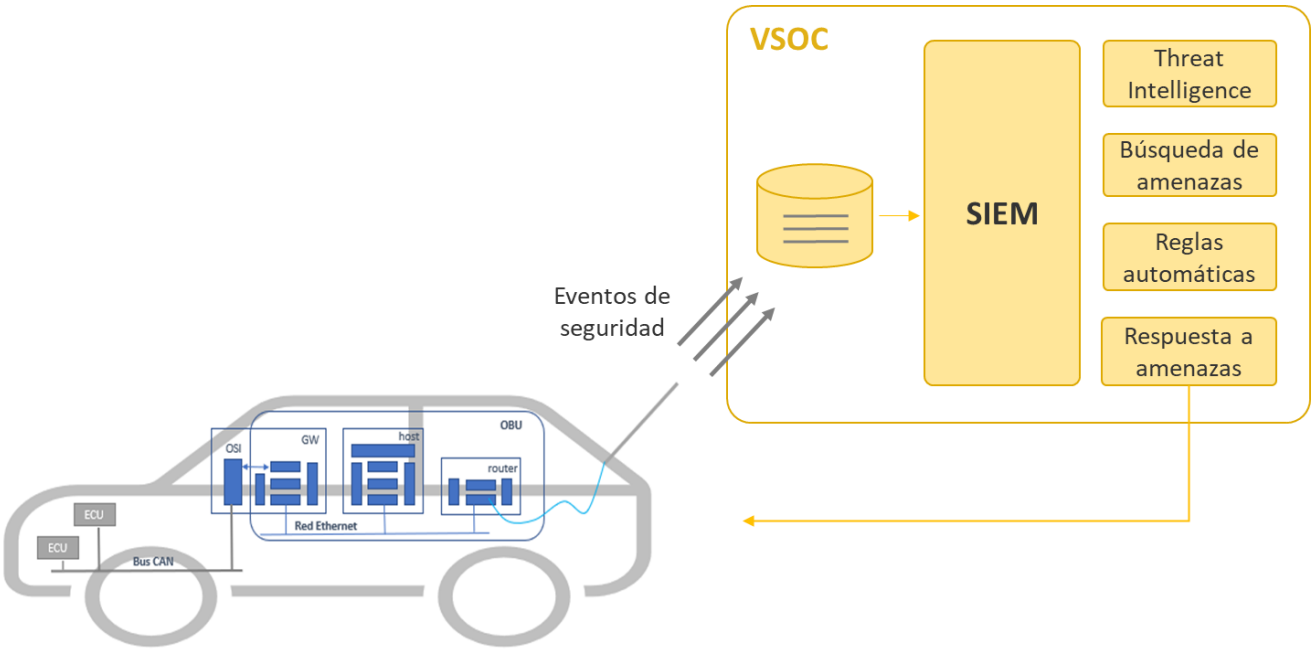


Ilustración 35. Requisitos de arquitectura VSOC

Adicionalmente, la flota de vehículos debe estar equipada con:

- Sistemas de detección de intrusiones (IDS) para monitorizar el tráfico de la red vehicular y detectar actividades anómalas.
- Comunicaciones seguras, para asegurar las comunicaciones entre el vehículo y la infraestructura VSOC.

- Sistema de registro y auditoría, para almacenar los eventos de seguridad relevantes y transmitir esta información al VSOC para su análisis y correlación.

4.3.2. Selección de la tecnología

La pieza fundamental en la operativa de un SOC es el denominado SIEM (Security Information and Event Management System), cuya función primordial consiste en agregar los datos de eventos generados por los elementos de seguridad integrados en los entornos de redes, servidores, dispositivos y aplicaciones.

Las capacidades del SIEM deben incluir la detección de amenazas a través de correlación de datos y análisis de comportamiento de usuarios y entidades (UEBA), e integraciones de respuesta gestionadas con mecanismos de orquestación, automatización y respuesta de seguridad (SOAR), y con fuentes de inteligencia de amenazas (*Threat Intelligence*).

Según el último informe de Gartner sobre plataformas SIEM, publicado el pasado mes de octubre [24], **Microsoft Sentinel** se ha convertido en la herramienta líder en cuanto a capacidad de ejecución:

Ilustración 36. Magic Quadrant for Security Information and Event Management

4.3.3. Modelo de arquitectura

Dentro del conjunto de características que diferencian a Microsoft Sentinel del resto de soluciones de mercado, dos de ellas son especialmente relevantes en el ámbito de este proyecto:

1- Su mayor grado de cobertura sobre los casos de uso contemplados en la matriz MITRE ATT&CK.

2- Un amplio ecosistema de productos de seguridad integrados, que incluye soluciones de **Seguridad IoT** y de **Edge Computing**. Se trata de dos cuestiones relacionadas e inherentes a la problemática del vehículo conectado, y que están contempladas en el modelo de arquitecturas de referencia de ciberseguridad de Microsoft (MCRA) [27] a través de las herramientas que se exponen a continuación.

Microsoft Defender for IoT

En un entorno tradicional de TI se pueden desplegar herramientas de ciberseguridad EDR (*Endpoint Detection and Response*) sobre los activos que se desea proteger, puesto que normalmente son sistemas que están diseñados para soportar la instalación de este tipo de mecanismos, con capacidad para la detección de intrusiones y la respuesta automática que puede activarse o bien de forma directa en base a unas reglas almacenadas en el propio dispositivo, o bien tras recibir la correspondiente orden desde el SOC.

Sin embargo, la realidad del vehículo conectado es algo distinta. Obliga a aplicar soluciones de seguridad IoT puesto que las redes vehiculares no conectan servidores y estaciones de trabajo, sino ECUs y sensores de madurez y capacidad tecnológica diversa.

Algunos de estos dispositivos soportarán la instalación de ciertos agentes de seguridad con funcionalidad EDR, pero en muchos otros casos no será posible y se requerirá otro tipo de estrategia.

La solución *Defender for IoT* de Microsoft contempla esa casuística y combina diferentes enfoques para solventarla:

Para los dispositivos que admiten la instalación de agentes (*greenfield devices*), monitoriza en tiempo real la actividad del dispositivo y la detección de amenazas de seguridad, así como la aplicación de políticas de seguridad y la gestión de parches y actualizaciones de seguridad.

Para los dispositivos que no admiten la instalación de agentes (*brownfield o legacy devices)* hace una monitorización no-intrusiva del tráfico de red sobre un puerto SPAN (*Switched Port ANalyzer*) y aplica técnicas de análisis de comportamiento basadas en *Machine Learning*, que le permiten detectar, por ejemplo:

- Comportamientos de red sospechosos, como la transferencia de un alto volumen de datos a destinos desconocidos o la concurrencia de un número atípico de peticiones a un determinado dispositivo.
- Comportamientos de dispositivo anómalos, como la aparición de dispositivos desconocidos en la red, la instalación de software no autorizado o la alteración del firmware del dispositivo.

Defender for IoT también aporta un servicio en la nube para analizar las comunicaciones entre los vehículos y los dispositivos IoT de las vías de transporte y de las ciudades inteligentes, destinado por ejemplo a aplicaciones de gestión de flotas de vehículos eléctricos.

En ese escenario, los vehículos enviarán regularmente datos de telemetría, como la ubicación, el estado de la batería o el rendimiento del motor a un servicio en la nube que procesará esos datos para proporcionar información de negocio a los operadores de las flotas (estado de mantenimiento, predicción de averías, optimización de rutas, etcétera), y *Defender for IoT* podría cubrir las siguientes funciones de seguridad:

- Identificación de posibles amenazas, como intentos de intrusión, filtración de datos o ataques de denegación de servicio.
- Autenticación de los vehículos y los dispositivos IoT al conectarse al servicio.
- Monitorización de los patrones de comunicación y los eventos de seguridad para identificar cualquier actividad sospechosa que pueda indicar una amenaza de seguridad.

Microsoft Azure IoT Hub

Azure IoT Hub es un servicio de comunicación bidireccional a través de Internet entre los dispositivos IoT y la nube, con las siguientes características:

- Puede ser utilizado para enviar telemetría desde vehículos a la nube, y para enviar comandos de control desde la nube a los vehículos.
- Azure IoT Hub también proporciona características de seguridad como la autenticación de dispositivos y la autorización de acceso.

Microsoft Azure Sphere

Azure Sphere protege los dispositivos embarcados y los módulos de comunicación en los vehículos conectados mediante un microcontrolador certificado y un sistema operativo seguro basado en Linux, con dos funciones primordiales:

- Garantizar la integridad y la confidencialidad de los datos y el *firmware* del vehículo utilizando características de seguridad avanzadas, como

el arranque seguro y como el almacenamiento seguro de claves criptográficas.
- Facilitar la actualización segura de firmware y software en los dispositivos IoT.

Microsoft Azure IoT Edge

Azure IoT Edge es un servicio que permite a los dispositivos IoT ejecutar análisis de datos, inteligencia artificial y otros procesos directamente en el dispositivo, en lugar de depender únicamente de la nube. Esto es necesario para habilitar el siguiente tipo de escenarios:

- Procesamiento y análisis de datos en tiempo real en el perímetro, para sistemas vehiculares que requieren decisiones rápidas y autónomas, como vehículos autónomos y sistemas de asistencia al conductor (ADAS).
- Ejecución de modelos de inteligencia artificial y aprendizaje automático en el vehículo para funciones como reconocimiento de objetos, detección de obstáculos y análisis de comportamiento del conductor.
- Reducción de la latencia y el ancho de banda de las comunicaciones al enviar solo datos críticos o agregados a la nube, en lugar de transmitir todos los datos en bruto que generan los sensores y en general los sistemas embarcados en el vehículo.

Requisitos de hardware y software

Hay que tener en cuenta que Azure IoT Edge y Azure Sphere no se pueden instalar en cualquier sensor. Ambas soluciones tienen requisitos específicos de hardware y software.

Azure IoT Edge está diseñado para funcionar en dispositivos IoT más potentes y gateways que ejecutan sistemas operativos compatibles, como Linux o Windows. No se instala directamente en sensores simples, ya que estos dispositivos generalmente carecen de capacidades de computación y de la memoria necesarias para ejecutar el entorno de Azure IoT Edge.

Azure Sphere, por otra parte, es una solución que incluye hardware, software y servicios en la nube. Los dispositivos Azure Sphere deben contar con un microcontrolador certificado de Azure Sphere (por ejemplo, MediaTek MT3620) que incluye características de seguridad avanzadas.

Al igual que ocurre con Azure IoT Edge, Azure Sphere tampoco se instala en sensores simples, ya que requiere un hardware específico y un conjunto de características de seguridad.

Para conectar sensores simples o dispositivos IoT de menor capacidad a la nube de Azure, existen las siguientes opciones:

1. Utilizar un gateway IoT: Los sensores y dispositivos de menor capacidad pueden conectarse a un gateway IoT compatible con Azure IoT Edge. El gateway IoT puede recopilar y procesar datos de los sensores y luego transmitirlos a la nube a través de Azure IoT Hub. Este enfoque permite que los sensores con capacidades limitadas se beneficien de las características de Azure IoT Edge y Azure IoT Hub sin tener que ejecutar el software directamente.
2. Conectar directamente a Azure IoT Hub: Si los sensores tienen alguna capacidad de procesamiento y conectividad a Internet, posiblemente se pueden conectar a Azure IoT Hub mediante el protocolo MQTT, AMQP o HTTP. Los sensores pueden enviar datos de telemetría a Azure IoT Hub y recibir comandos desde la nube. Sin embargo, esta opción no proporciona las capacidades de computación en el perímetro que ofrece Azure IoT Edge ni las características de seguridad avanzadas de Azure Sphere.

La imagen siguiente muestra las relaciones entre los principales elementos de la arquitectura:

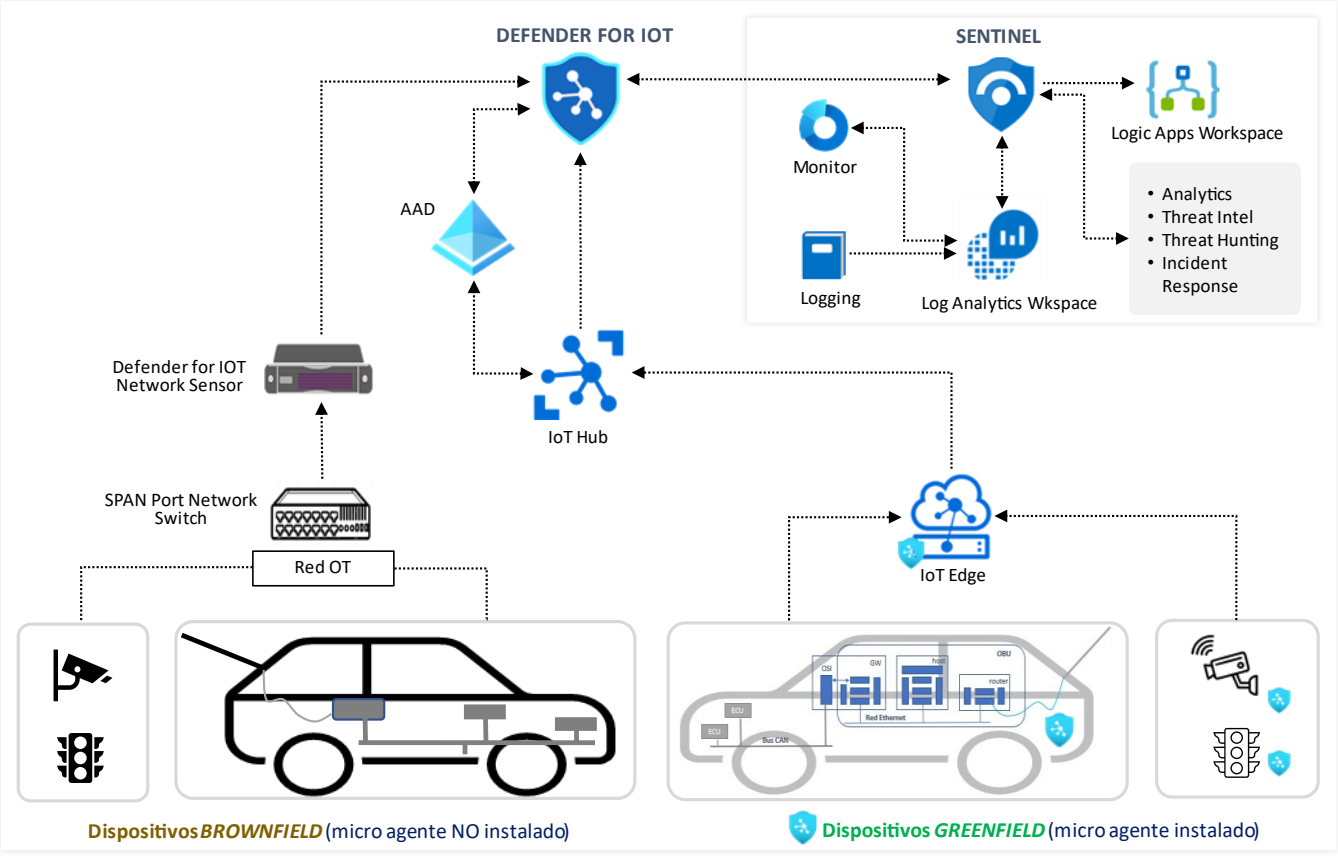


Ilustración 37. Arquitectura de ciberseguridad para el vehículo conectado

4.4. Definición del caso de uso

En el capítulo de casos de estudio se ha expuesto un ejemplo de ataque realizado a través de una aplicación smartphone que había sido manipulada para permitir que el adversario filtrara mensajes malintencionados en la red interna del vehículo a través del conector OBD-II, y posteriormente tomara el control provocando la parada del motor.

Sin duda existen fórmulas mediante las cuales se podría haber evitado de antemano aquel ataque, empezando por el fabricante de la app, que podría haber utilizado técnicas de ofuscación del código para impedir la localización de los comandos AT, o por el fabricante del vehículo, que podría haber implantado un cortafuegos en el puerto del OBD-II para bloquear todos los mensajes CAN con un identificador no autorizado.

Sin embargo, sólo sería cuestión de tiempo que los adversarios descubriesen y consiguiesen explotar cualquier otra vulnerabilidad del sistema.

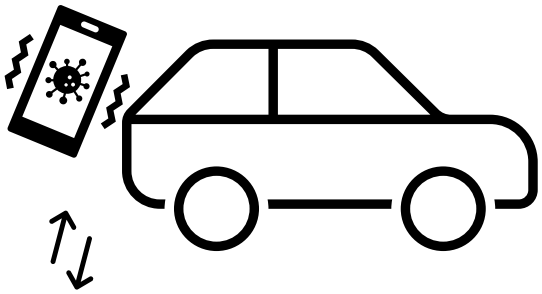
Por consiguiente, este ejercicio no trata sobre cómo bastionar el perímetro del vehículo sino sobre cómo se puede identificar una amenaza, antes y después de que se convierta en incidente.

Caso de uso

El caso de uso sobre el que se basa el ejercicio consiste en la detección automática de un mensaje circulando por el bus CAN con un identificador “sospechoso”.

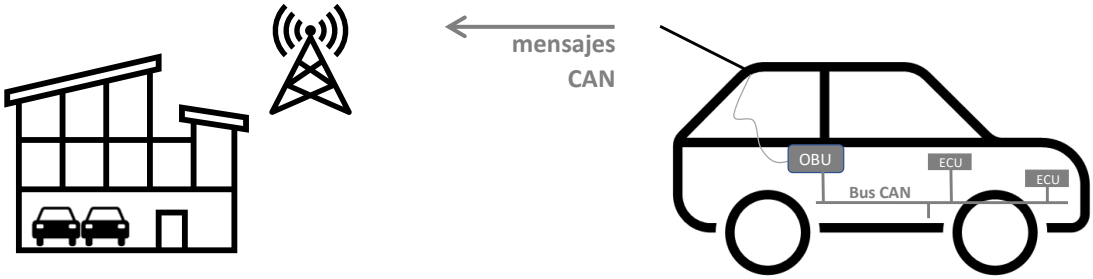
A continuación, se explica el caso mediante una secuencia de 5 fases:

1- La víctima ha instalado en su smartphone una aplicación de diagnósticos sin darse cuenta de que es falsa. El smartphone infectado se empareja por Bluetooth o Wifi al conector OBD-II de su vehículo a través de un adaptador ELM327.

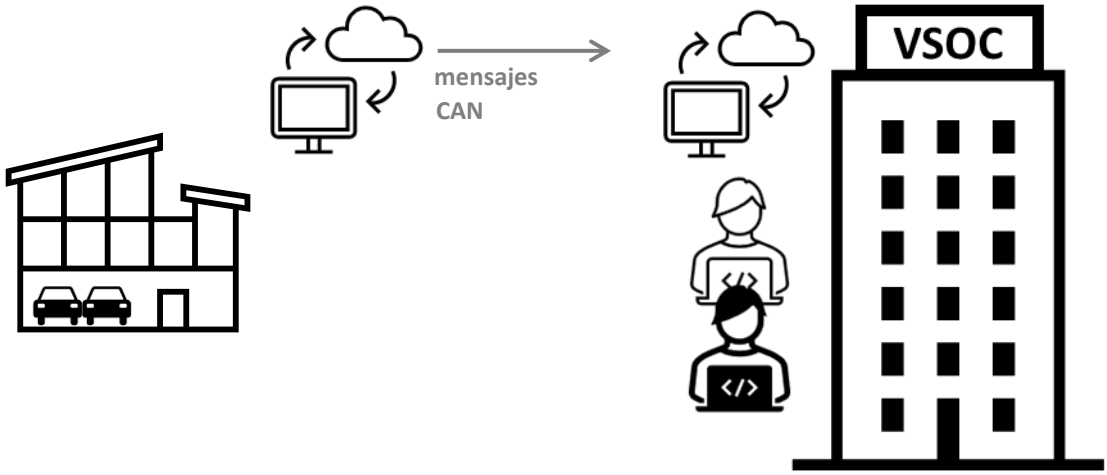


2- La aplicación manipulada se conecta a un servicio web malicioso desde el que el atacante puede obtener información del vehículo y también controlar su comportamiento mediante el envío de comandos AT destinados al adaptador ELM327.

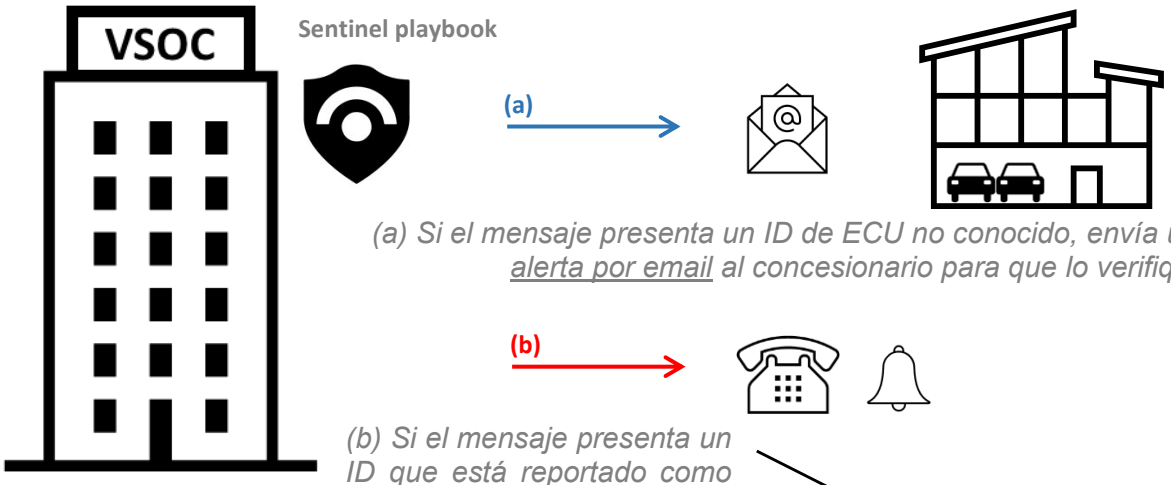
3- El automóvil está conectado a la red celular a través de una tarjeta eSIM, y con una cierta frecuencia le envía al concesionario una muestra de los mensajes de diagnóstico que circulan por el bus CAN:



4- El concesionario transforma y reenvía en tiempo real al VSOC los mensajes procedentes de la flota, a través de Internet:

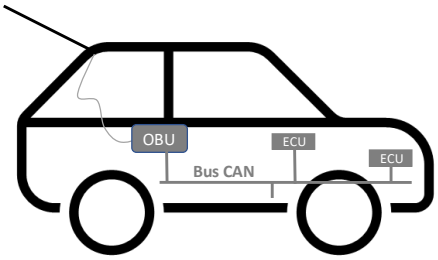


5- El SIEM del VSOC dispondrá de un automatismo para este caso de uso, y actuará en función del nivel de severidad del incidente:



(a) Si el mensaje presenta un ID de ECU no conocido, envía una alerta por email al concesionario para que lo verifique.

(b) Si el mensaje presenta un ID que está reportado como malicioso por los concesionarios o por las fuentes de Threat Intelligence, en ese caso se enviará una llamada de emergencia directamente al propio vehículo para advertirle que se ha producido un incidente de seguridad y que debe dejar de circular a la mayor brevedad posible.



4.5. Experimento

4.5.1. Instrumental requerido

En el experimento se han utilizado los siguientes materiales:

- 1 ordenador portátil.
- 1 tenant de Azure, que incluye:
  - 1 instancia de Microsoft Sentinel
  - 1 licencia de Logic Apps
- 1 adaptador ELM327 Vgate iCar 2 Wifi WLAN EOBD OBDII.
- 1 smartphone con la App de diagnósticos Torque Pro instalada.
- 1 analizador de protocolo de red Wireshark.
- 1 servicio cloud de VoIP (Twilio) para el envío automático de mensajes de voz.

Ilustración 38. Instrumental del laboratorio

4.5.2. Escaneo del bus CAN

Se puede interactuar con la red interna del vehículo, es decir, inyectar y monitorizar los mensajes que circulan por el bus CAN, gracias al conector OBD-II.

Para ello, en primer lugar, es necesario localizar el conector, que suele estar debajo del salpicadero, a la izquierda de la posición del volante, y a continuación hay que enchufarle el adaptador ELM327:

Ilustración 39. Adaptador ELM327 enchufado al conector OBD-II

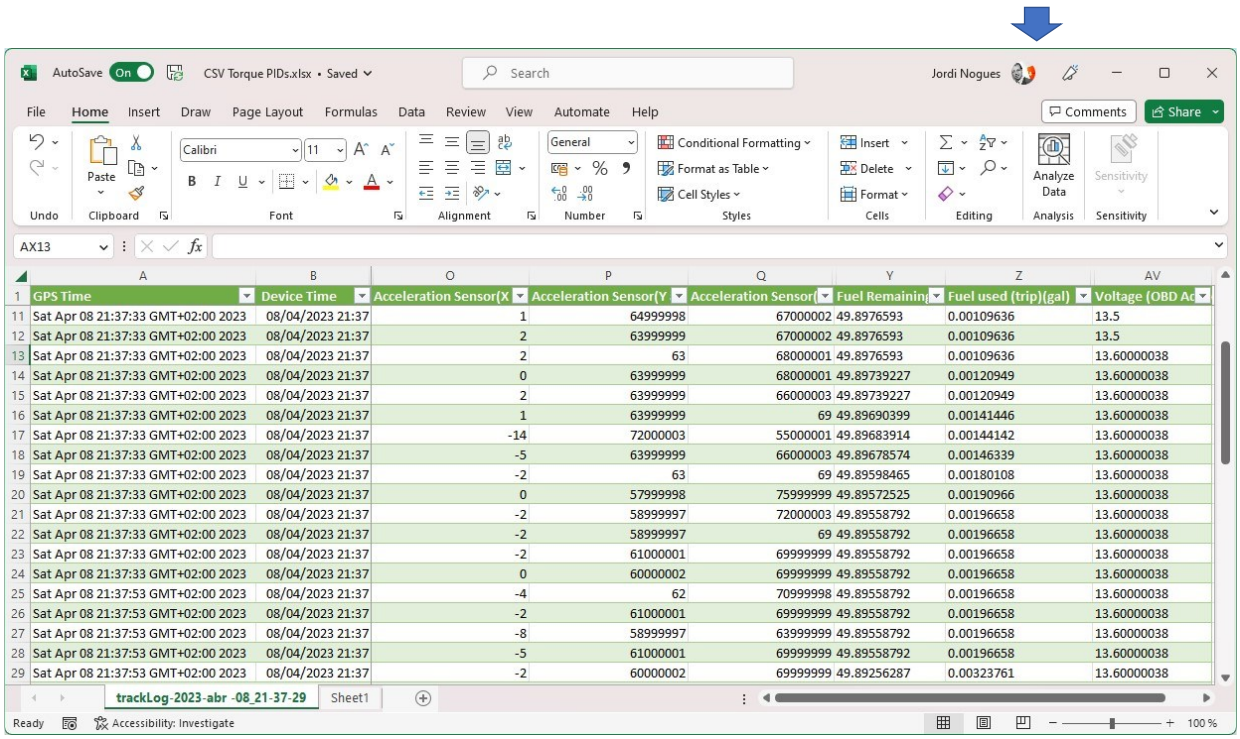
A continuación, a través de una aplicación smartphone de gestión de diagnósticos, Torque Pro en este caso, se debe establecer la comunicación con el adaptador.

La mayoría de los adaptadores ofrecen emparejamiento Bluetooth, aunque en este caso es posible hacerlo a través de Wifi, con la ventaja que esto representa de cara a poder escanear la comunicación con Wireshark.

El adaptador proporciona un punto de acceso Wifi en el puerto 35000 de la dirección IP 192.168.0.10, y esa es la configuración que por defecto utiliza Torque, puesto que es la habitual en la mayoría de los adaptadores del mercado.

Las imágenes siguientes muestran el proceso de configuración de los parámetros (PIDs) que deseamos consultar al conector OBD-II, y cómo pueden exportarse posteriormente por correo electrónico (en formato CSV) al finalizar las mediciones.

Ilustración 40. Configuración y exportación de registros de Torque



	A	B	O	P	Q	Y	Z	AV
1	GPS Time	Device Time	Acceleration Sensor(X	Acceleration Sensor(Y	Acceleration Sensor(	Fuel Remainin	Fuel used (trip)(gal)	Voltage (OBD Ac
11	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	1	64999998	67000002	49.8976593	0.00109636	13.5
12	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	2	63999999	67000002	49.8976593	0.00109636	13.5
13	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	2	63	68000001	49.8976593	0.00109636	13.60000038
14	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	0	63999999	68000001	49.89739227	0.00120949	13.60000038
15	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	2	63999999	66000003	49.89739227	0.00120949	13.60000038
16	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	1	63999999	69	49.89690399	0.00141446	13.60000038
17	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	-14	72000003	55000001	49.89683914	0.00144142	13.60000038
18	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	-5	63999999	66000003	49.89678574	0.00146339	13.60000038
19	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	-2	63	69	49.89598465	0.00180108	13.60000038
20	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	0	57999998	75999999	49.89572525	0.00190966	13.60000038
21	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	-2	58999997	72000003	49.89558792	0.00196658	13.60000038
22	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	-2	58999997	69	49.89558792	0.00196658	13.60000038
23	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	-2	61000001	69999999	49.89558792	0.00196658	13.60000038
24	Sat Apr 08 21:37:33 GMT+02:00 2023	08/04/2023 21:37	0	60000002	69999999	49.89558792	0.00196658	13.60000038
25	Sat Apr 08 21:37:53 GMT+02:00 2023	08/04/2023 21:37	-4	62	70999998	49.89558792	0.00196658	13.60000038
26	Sat Apr 08 21:37:53 GMT+02:00 2023	08/04/2023 21:37	-2	61000001	69999999	49.89558792	0.00196658	13.60000038
27	Sat Apr 08 21:37:53 GMT+02:00 2023	08/04/2023 21:37	-8	58999997	63999999	49.89558792	0.00196658	13.60000038
28	Sat Apr 08 21:37:53 GMT+02:00 2023	08/04/2023 21:37	-5	61000001	69999999	49.89558792	0.00196658	13.60000038
29	Sat Apr 08 21:37:53 GMT+02:00 2023	08/04/2023 21:37	-2	60000002	69999999	49.89256287	0.00323761	13.60000038

Ilustración 41. Exportación del fichero de diagnósticos del vehículo

En el fichero de diagnósticos del vehículo que proporciona Torque, los registros exportados solamente incluyen los valores solicitados. Para obtener información más detallada es necesario escanear el tráfico del bus CAN, por

ejemplo, con Wireshark, aunque existen herramientas más específicas para ello, que facilitan el formateo de los mensajes.

Puesto que se desea interceptar el tráfico transmitido entre el smartphone y el conector OBD2, es necesario activar el modo promiscuo en Wireshark:

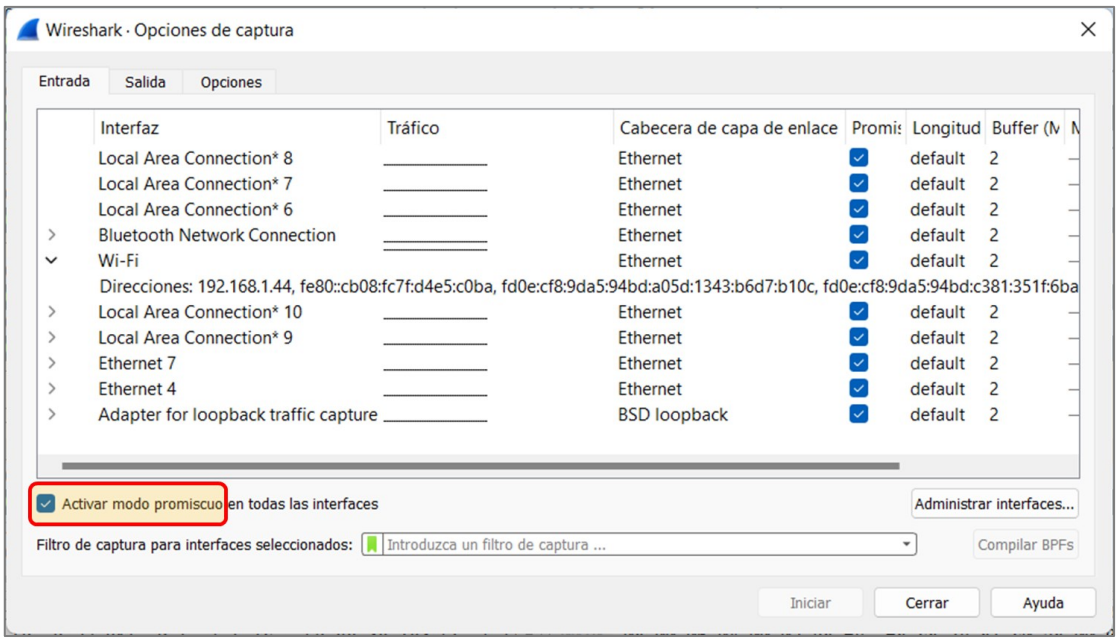


Ilustración 42. Configuración de Wireshark en modo promiscuo

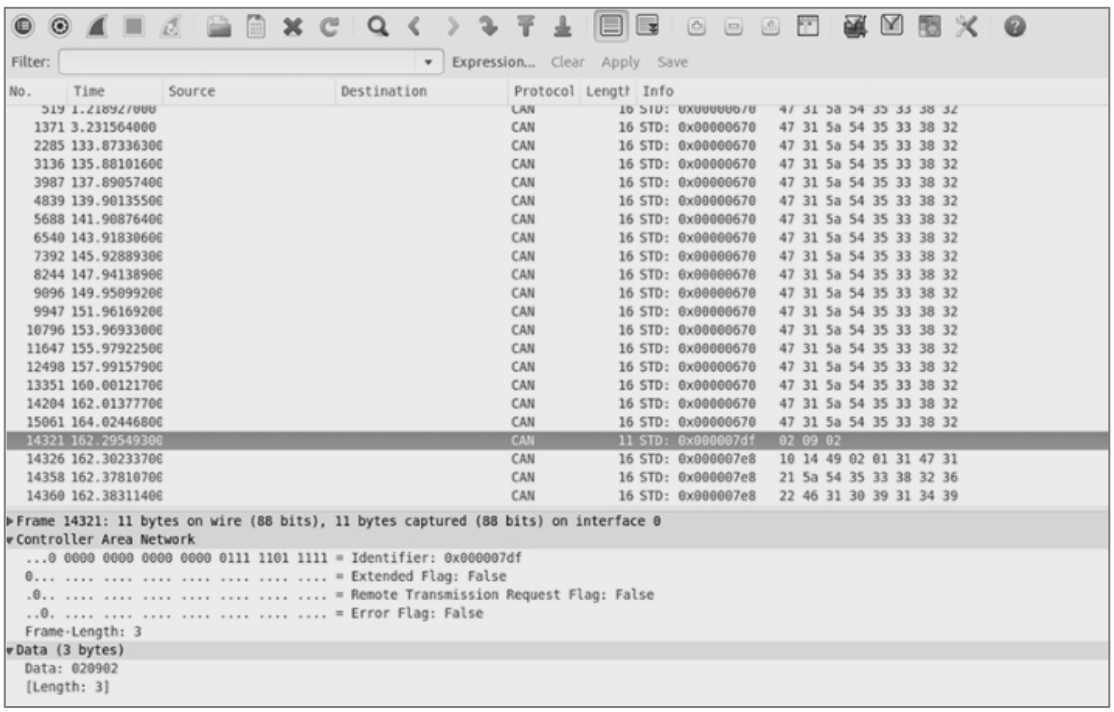


Ilustración 43. Captura de mensajes CAN en Wireshark [3]

4.5.3. Envío de logs al VSOC

Existen varias alternativas para enviar a Sentinel los logs de eventos generados en el concesionario.

Según los expertos [28], la arquitectura de Sentinel incluye un agente para la ingesta de logs (*Log Analytics Agent*) que admite indistintamente los dos estándares de formato más extendidos en el mercado: Syslog y CEF (*Common Event Format*), aunque recomienda este último porque requiere menor esfuerzo de formateo para su posterior interpretación en el motor de reglas analíticas.

Azure Monitor HTTP Data Collector API

Para escenarios especiales como el de este experimento, donde la estructura de los datos se aleja de esos formatos estándar y además la instalación del agente plantea cierta dificultad por sus requerimientos de infraestructura, se recomienda una alternativa basada en el *Azure Monitor HTTP Data Collector API*, que puede utilizarse de forma sencilla desde un cliente REST. El único requisito de este método es que los mensajes deben enviarse en formato JSON.

Por consiguiente, cuando el servidor del concesionario reciba los mensajes CAN procedentes de la flota de vehículos, los transformará a un formato JSON con la siguiente estructura:

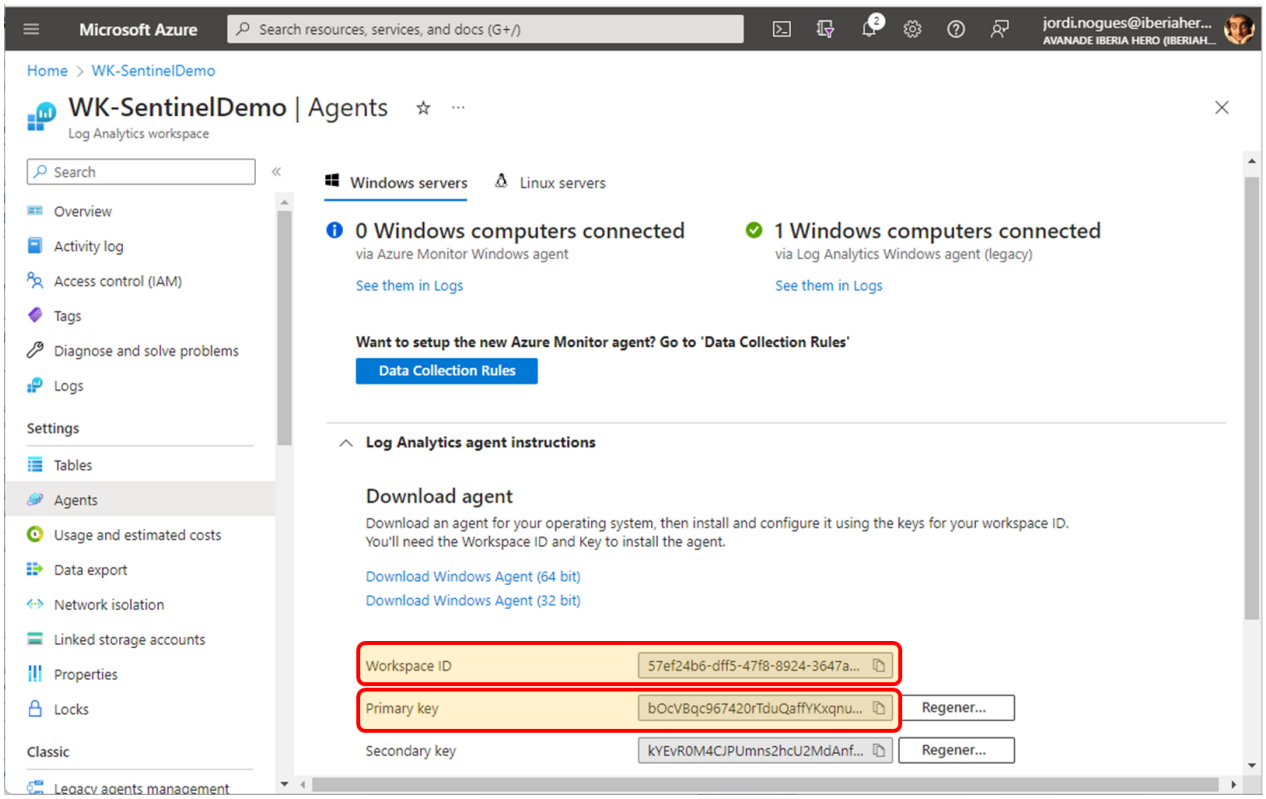
```
[
  {
    "timestamp0": "2023-04-13T22:33:00.000Z",
    "ID_Vehiculo": "Vehiculo001",
    "ID_ECU": "0x123",
    "longitud": 8,
    "datos": ["0x12", "0x34", "0x56", "0x78", …]
  },
  {
    "timestamp0": "2023-04-13T22:33:01.000Z",
    "ID_Vehiculo": "Vehiculo001",
    "ID_ECU": "0x128",
    "longitud": 8,
    "datos": ["0x12", "0x34", "0x56", "0x78", …]
  },
…
]
```

Una vez construido el fichero JSON, se enviará al Data Collector mediante un script de PowerShell, siguiendo las pautas que se detallan a continuación.

En el capítulo de anexos se puede consultar el código de este script, que ha sido construido con acuerdo al patrón propuesto en el siguiente artículo de Microsoft: https://learn.microsoft.com/en-us/azure/azure-monitor/logs/data-collector-api?tabs=powershell.

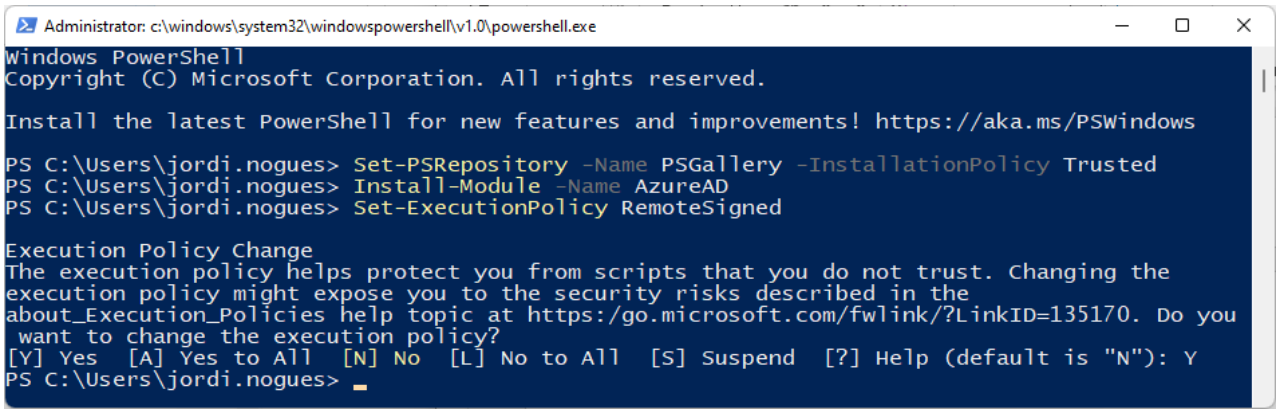
En el código del script puede observarse que el API debe conocer el identificador del workspace y también su clave compartida (shared key) para calcular la firma de la autorización.

La siguiente imagen muestra cómo se pueden obtener estos dos datos navegando a través de la interfaz de Azure:

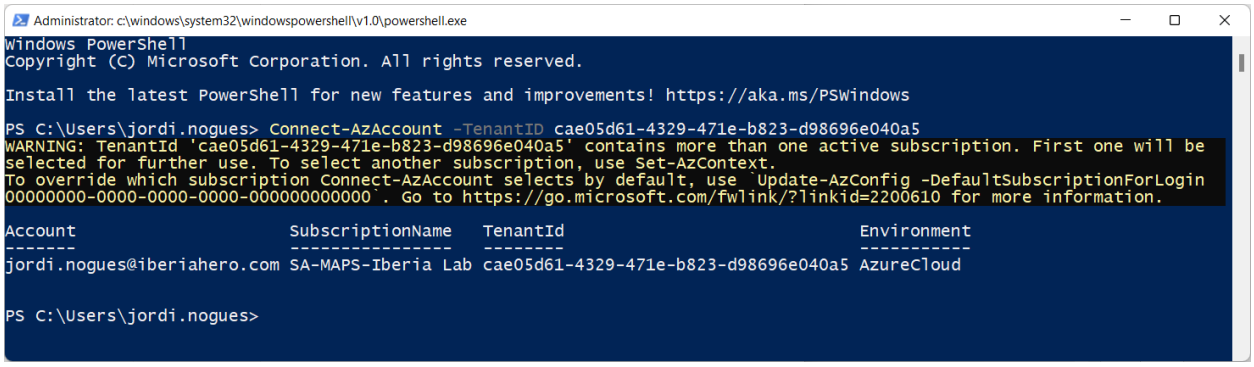


Una vez resueltos los preliminares, se trata de crear el script de PowerShell, para lo cual, en primer lugar, debe instalarse el módulo de Azure AD indicándole previamente al sistema que confíe en el repositorio PSGallery.

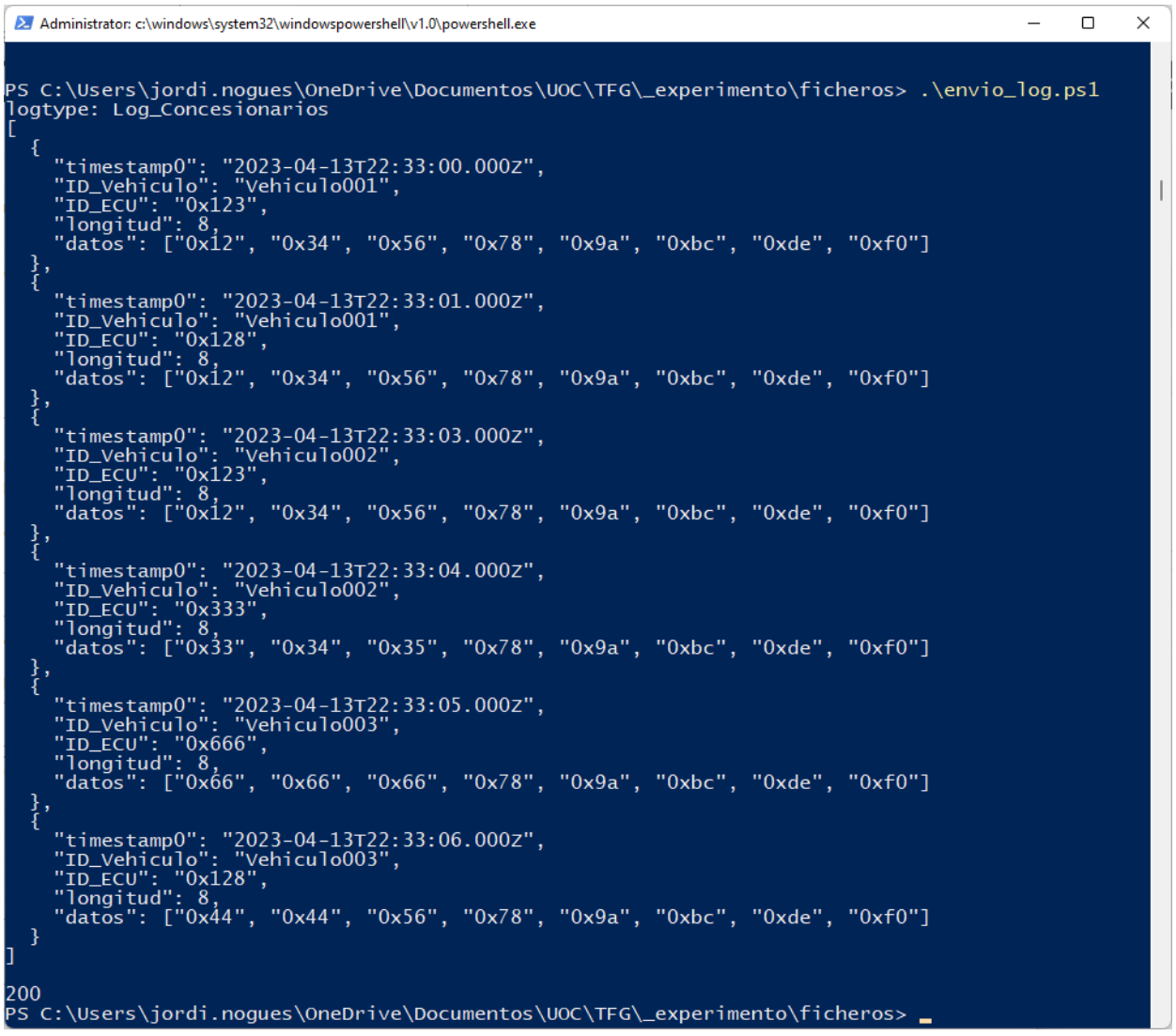
También es necesario forzar la política de ejecución de scripts, y situarnos en el directorio donde se almacena el script y el fichero JSON con los eventos a enviar:



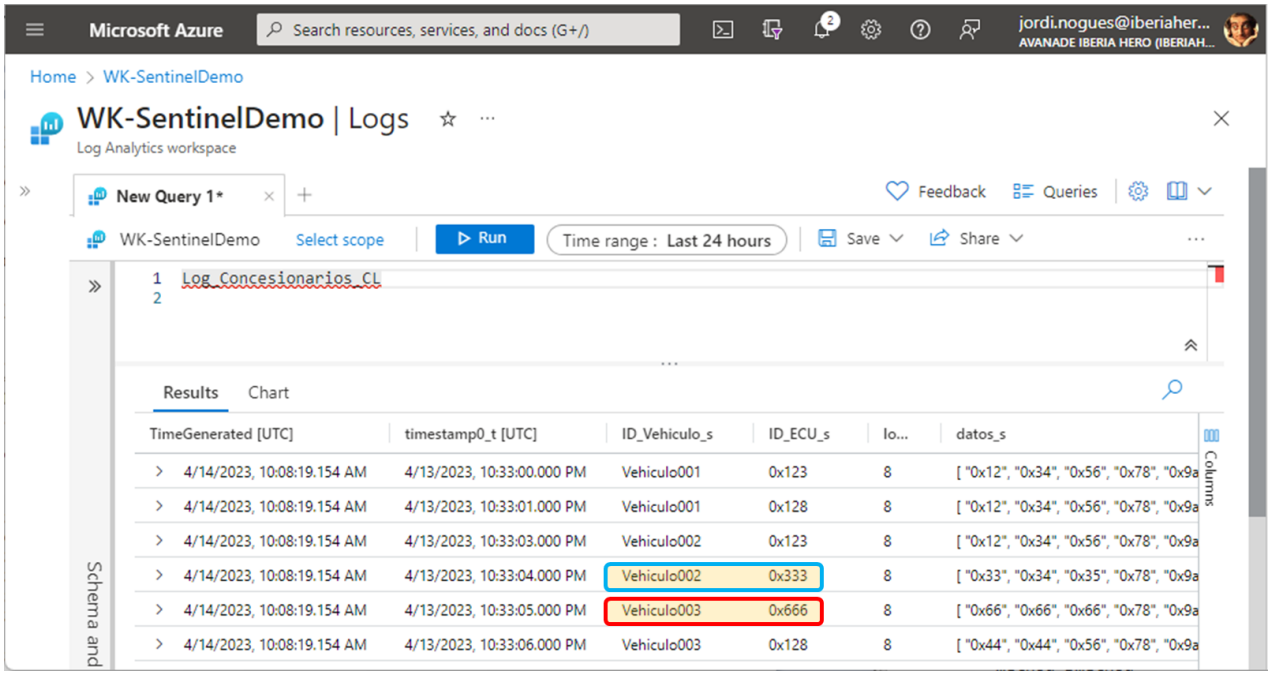
El último paso antes de lanzar el script es abrir la conexión con el tenant de Azure mediante el comando `Connect-AzAccount -TenantId <Tenant ID>`:



Finalmente se lanza el script, que deberá retornar un código HTTP 200 indicando que el resultado ha sido correcto. El script incluye un *cmdlet* "Write-Host" para mostrar en consola el JSON cargado desde el fichero:

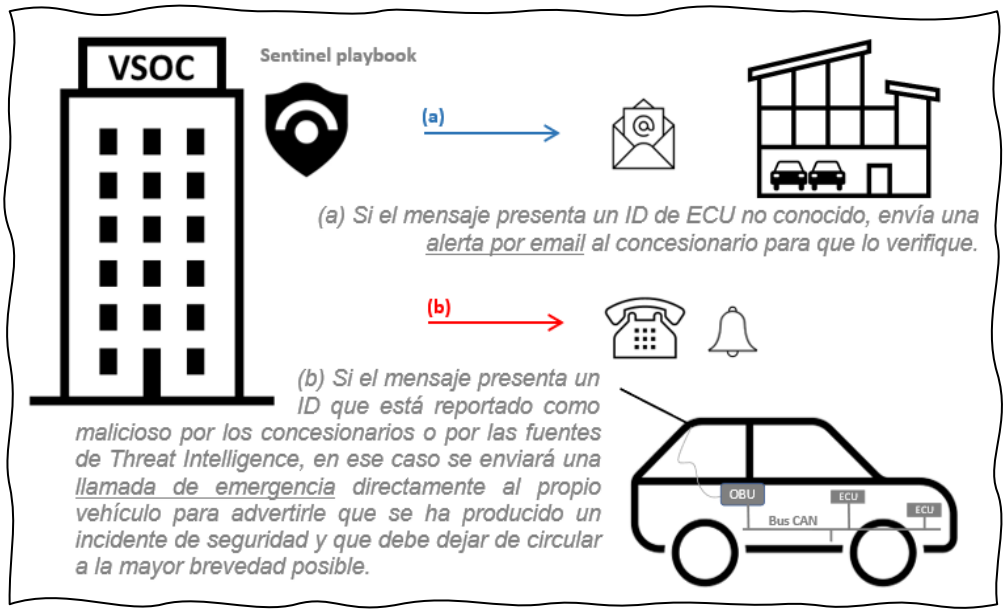


Para comprobar que los registros se han cargado en el Log Analytics Workspace, se ejecuta la consulta KQL:



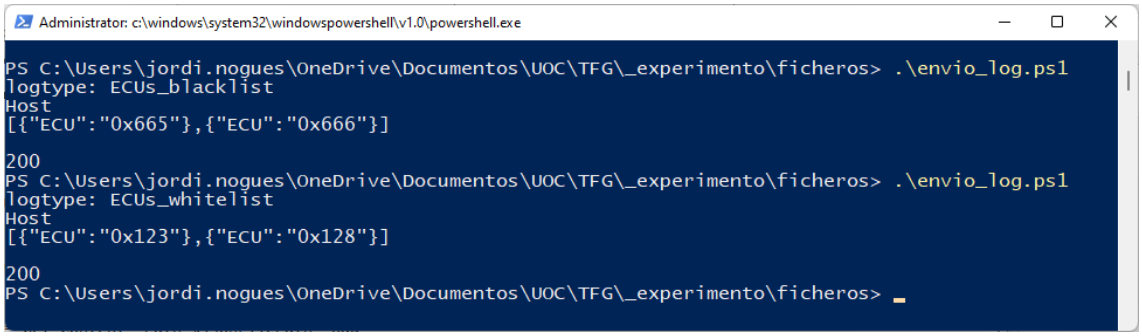
En la imagen de arriba aparecen remarcados los códigos que desencadenarán una acción en la monitorización del VSOC:

- En azul se muestra un mensaje procedente del vehículo 2 con un ID ECU (0x333) no confiable ➔ condición (a) del caso de uso.
- En azul se muestra un mensaje procedente del vehículo 3 con un ID ECU (0x666) denunciado como maligno ➔ condición (b).



Una vez resuelta la carga de los eventos procedentes del concesionario en el *custom log* del *workspace*, es necesario crear la *blacklist* y la *whitelist* para poder hacer las validaciones desde las reglas de Sentinel.

Estas listas podrían crearse como *custom logs* siguiendo el procedimiento anterior, simplemente modificando en el script el nombre del fichero de origen y el de la tabla en destino:



En ese caso, la regla identificaría los códigos maliciosos en el log de eventos del concesionario mediante el siguiente *join* en KQL:

```
ECUs_blacklist_CL
| join (Log_Concesionarios_CL) on $left.ECU_s == $right.ID_ECU_s
```

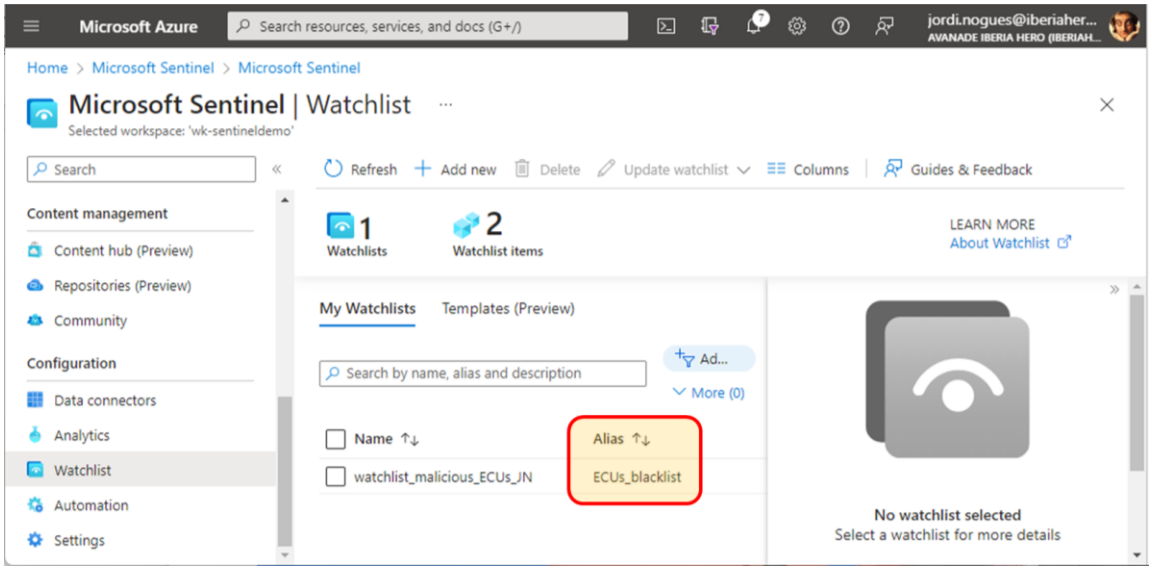
Sentinel Watchlist

No obstante, Sentinel integra una función específica para este cometido, denominada *Watchlist*, que en relación con el *join* anterior está optimizada para un máximo rendimiento mediante el uso de su atributo *SearchKey:*

La sintaxis en las reglas de detección será la siguiente:

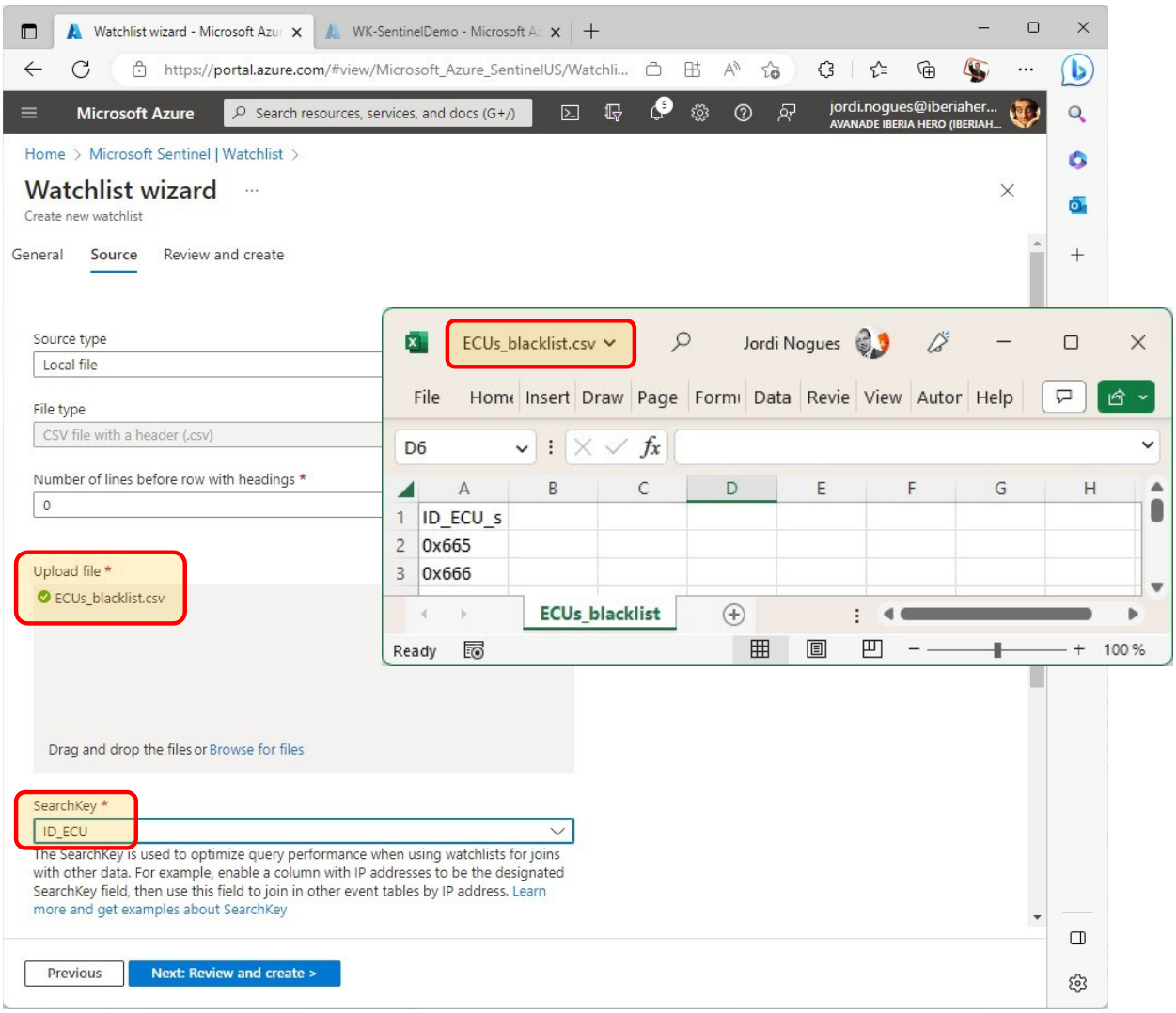
```
Log_Concesionarios_CL
| where ID_ECU_s in ((_GetWatchlist('ECUs_blacklist')
| project SearchKey))
```

Donde 'ECUs_blacklist' es el alias asignado a la *watchlist*:

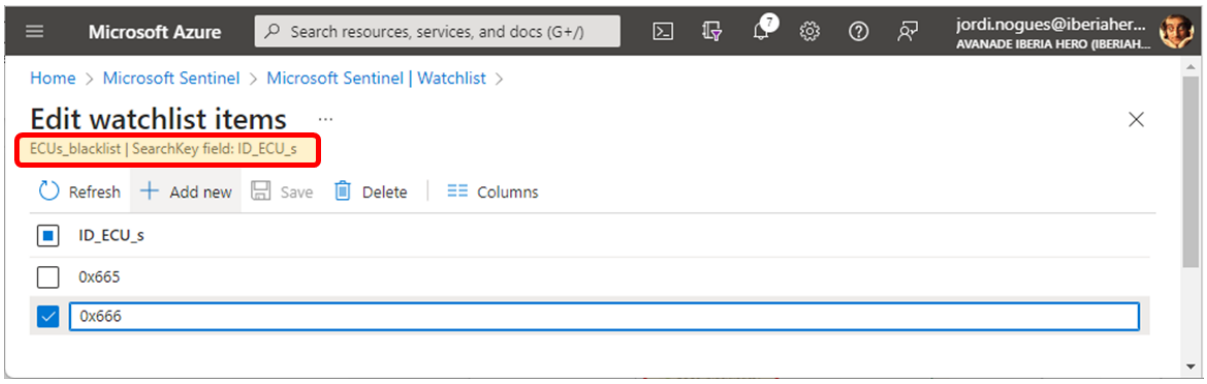


La función *Watchlist* no solamente está optimizada en rendimiento, sino que, además está orientada a facilitar el mantenimiento de las listas de códigos.

Este mantenimiento puede hacerse mediante una sencilla carga de un fichero CSV, como muestra la imagen de abajo correspondiente al asistente de configuración de la *watchlist*…



…e incluso también mediante la interfaz que proporciona Sentinel para editar manualmente los elementos de la *watchlist*:

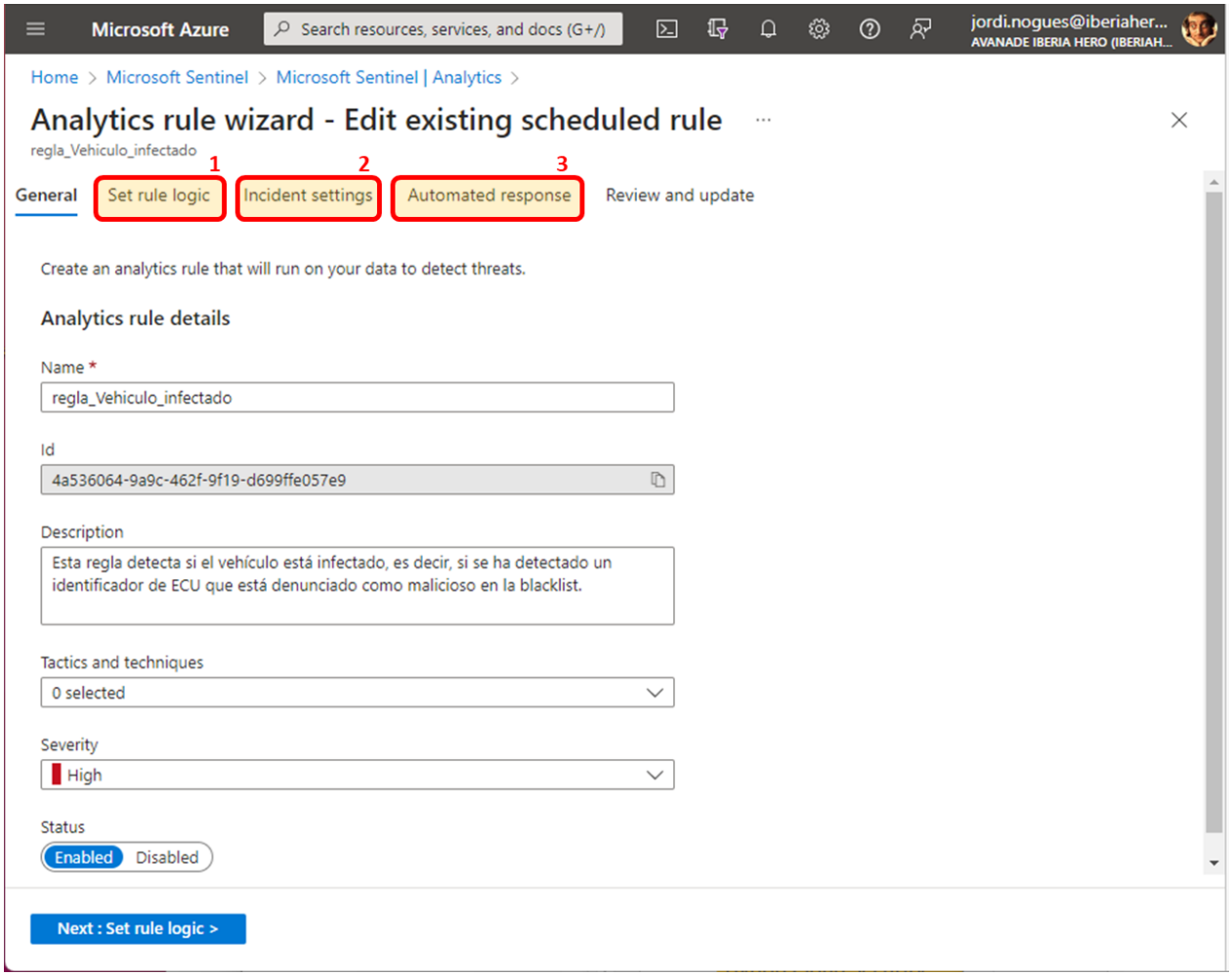


4.5.1. Automatización de la respuesta al incidente

Esta es la parte final del experimento. En ella se utilizará la capacidad SOAR de Sentinel para orquestar y automatizar la respuesta al incidente que se ha producido en el vehículo 3, en el cual se ha capturado un código de ECU identificado como malicioso (0x666) por las fuentes de *Threat Intelligence* (en este caso, por la *blacklist* creada mediante la función *Watchlist* de Sentinel).

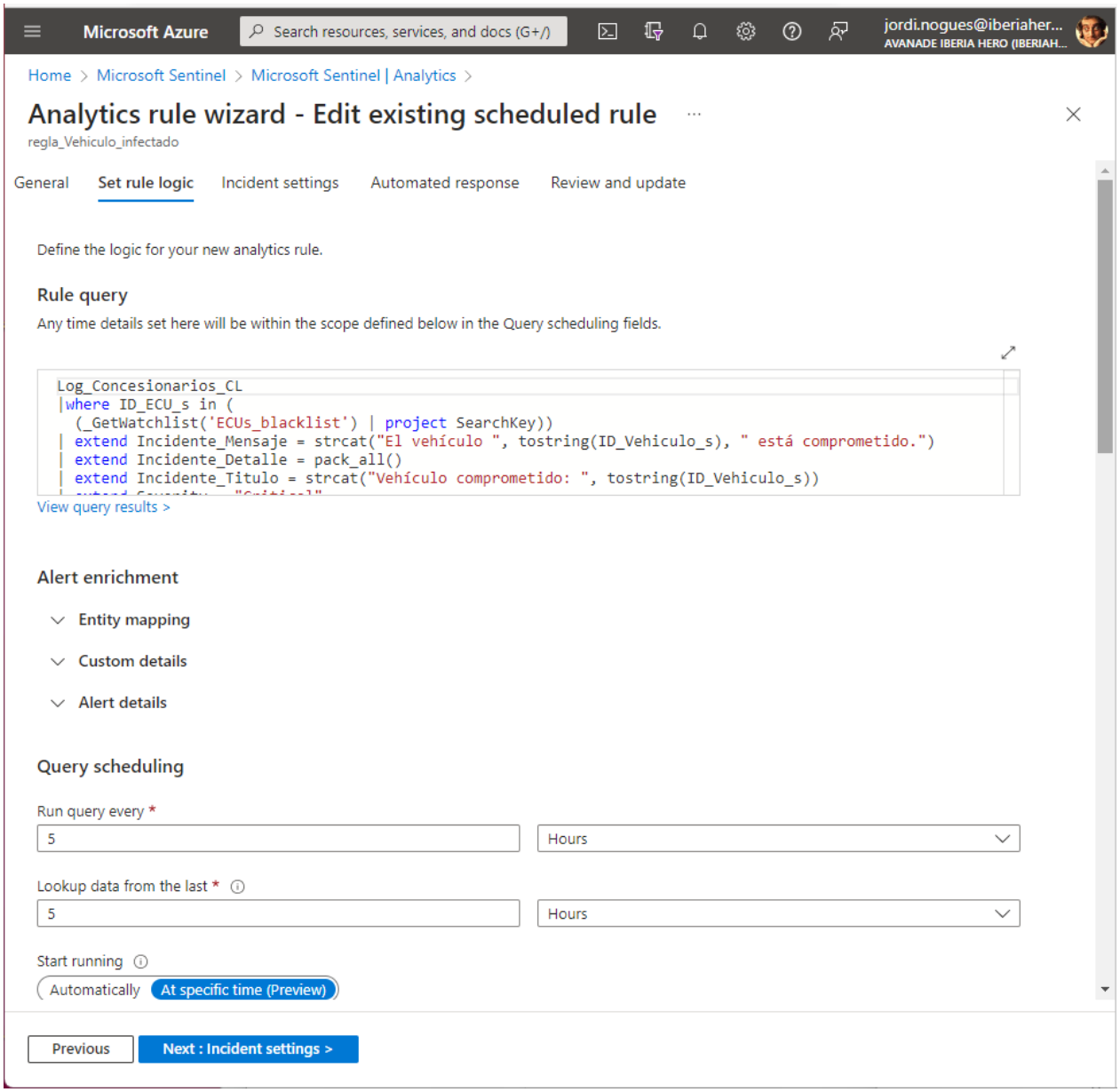
Para implementar la solución, debe crearse una regla analítica con las siguientes configuraciones:

1. Lógica de la regla analítica
2. Configuración del incidente
3. Respuesta automática
    - Llamada a un playbook creado con Azure Logic Apps



Lógica de la regla analítica

En este paso se construye la consulta KQL que debe detectar la amenaza cruzando los identificadores que contiene el log enviado por el concesionario (Log_Concesionarios_CL), contra los identificadores de la blacklist. Adicionalmente se configuran ciertos parámetros como por ejemplo la frecuencia de escaneo:

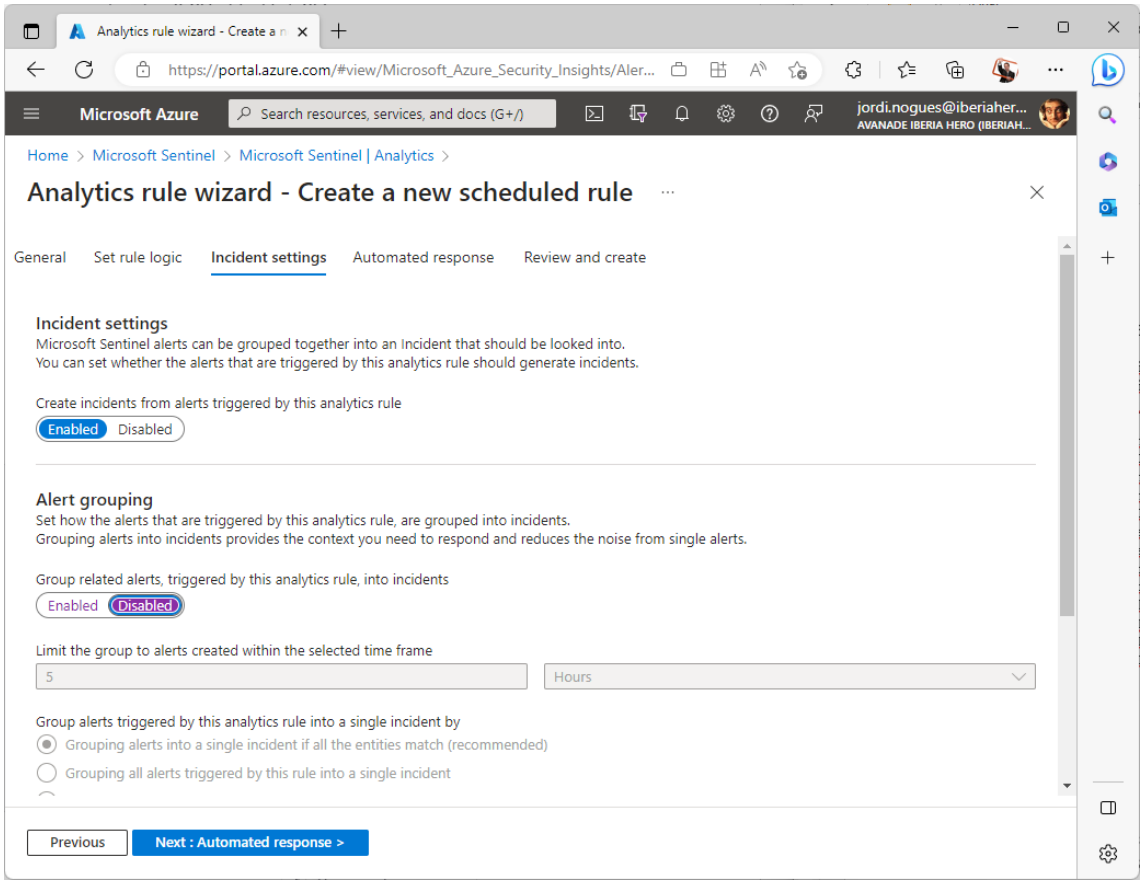


A continuación, se muestra el código de la *query*, en el que puede observarse que además de buscar los códigos maliciosos dentro del log, también incorpora al resultado una serie de columnas calculadas, con información útil para la posterior gestión del incidente:

```
Log_Concesionarios_CL
|where ID_ECU_s in (
  (_GetWatchlist('ECUs_blacklist') | project SearchKey))
| extend Incidente_Mensaje = strcat("El vehículo ",
tostring(ID_Vehiculo_s), " está comprometido.")
| extend Incidente_Detalle = pack_all()
| extend Incidente_Titulo = strcat("Vehículo comprometido: ",
tostring(ID_Vehiculo_s))
| extend Severity = "Critical"
| extend Recommendation = "Investigar el vehículo de inmediato."
| extend Confidence = 100
| extend ConfidenceLevel = "High"
| extend Classification = "Malware"
| project
    Incidente_Mensaje,
    Incidente_Detalle,
    Incidente_Titulo,
    Severity,
    Recommendation,
    Confidence,
    ConfidenceLevel,
    Classification
```

Configuración del incidente

En este paso se indica si debe generarse un incidente por cada registro "anómalo", o bien si deben agruparse en base a determinados criterios. En este experimento, se genera un incidente por cada registro anómalo:

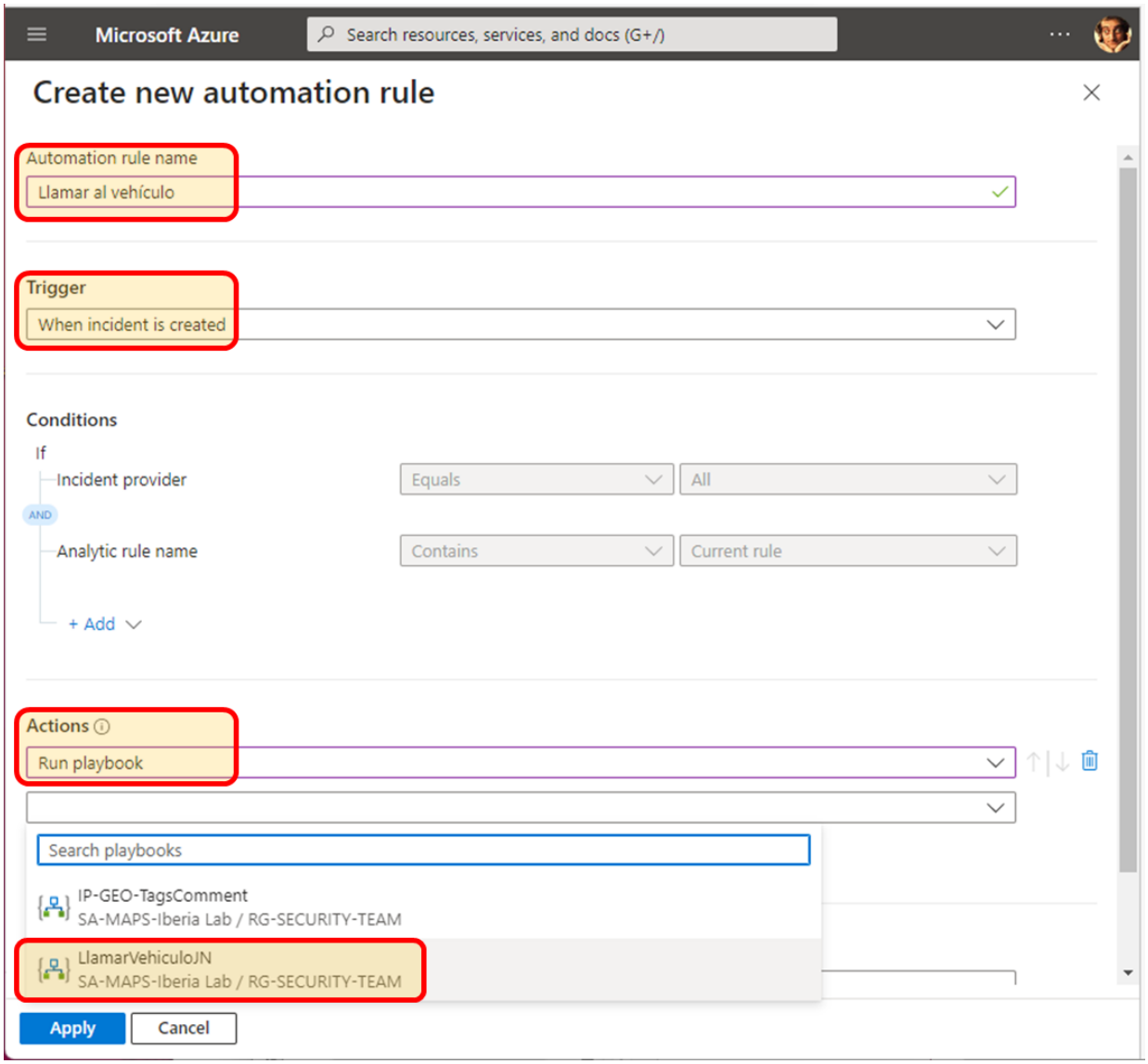


Configuración de la respuesta automática

En este paso se pueden especificar diversas acciones para que se ejecuten en secuencia, por ejemplo:

1- Asignar un responsable para que investigue el incidente
2- Ejecutar un playbook de Logic Apps (o más de uno)

La imagen siguiente muestra cómo, en este caso, asignamos la acción de ejecutar el playbook "LlamarVehiculoJN" (que ha debido ser creado previamente):

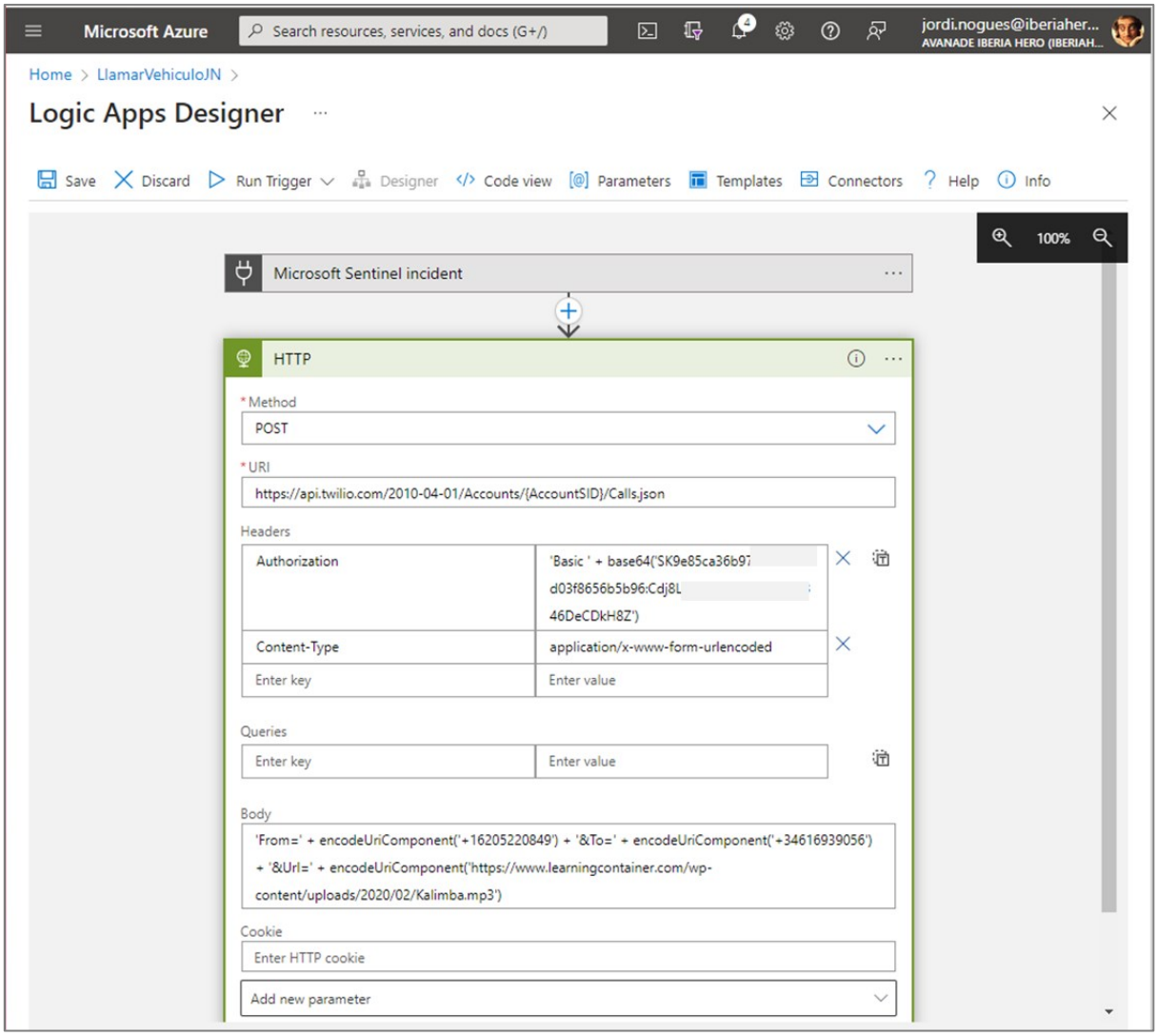


Creación del playbook en Azure Logic Apps

Al generarse el incidente se lanzará un automatismo para efectuar una llamada automática al vehículo. Este automatismo debe crearse con la herramienta estándar de Azure para la construcción de flujos de trabajo: **Logic Apps**.

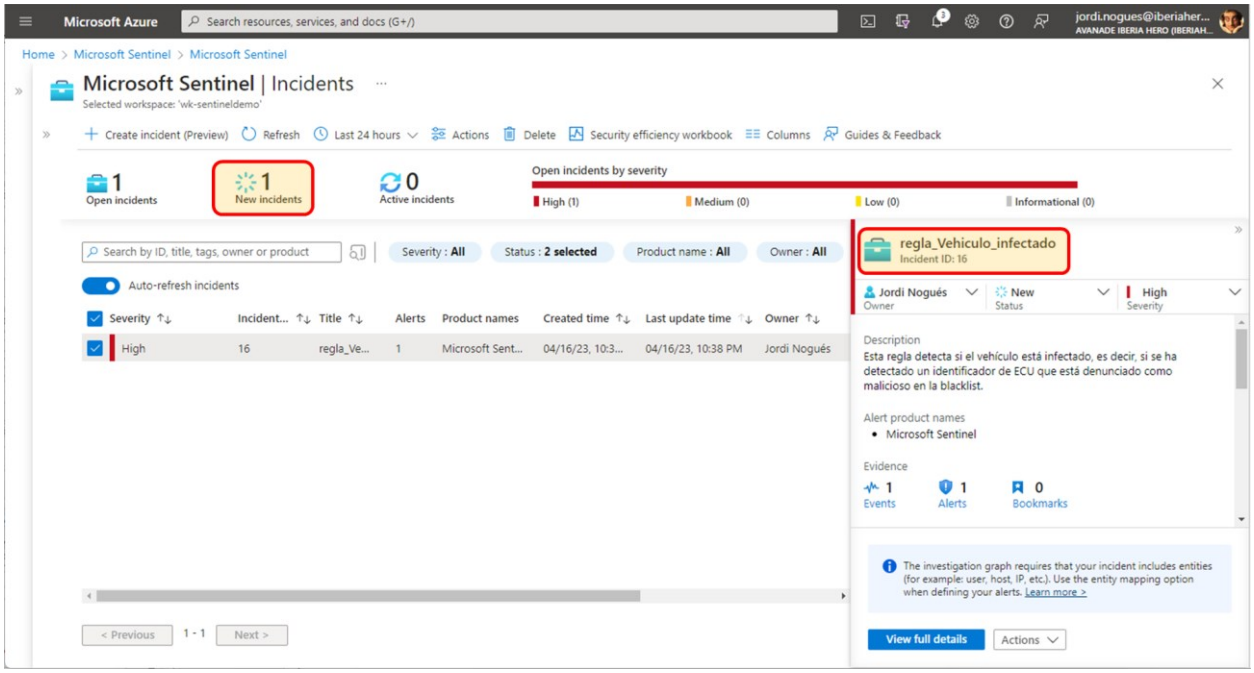
Existen diversas opciones para lanzar la llamada telefónica, como por ejemplo la creación de un bot de llamadas para Microsoft Teams utilizando el Microsoft Bot Framework, y después conectándolo vía HTTP con una Logic App; o también se podría utilizar JAIN-SIP, que es una plataforma de VoIP basada en Java, desde una solución desarrollada con el Java Media Framework.

En este experimento se utilizará una opción de implementación rápida que consiste en utilizar el servicio REST de Twilio invocándolo desde Logic Apps, según se muestra a continuación:

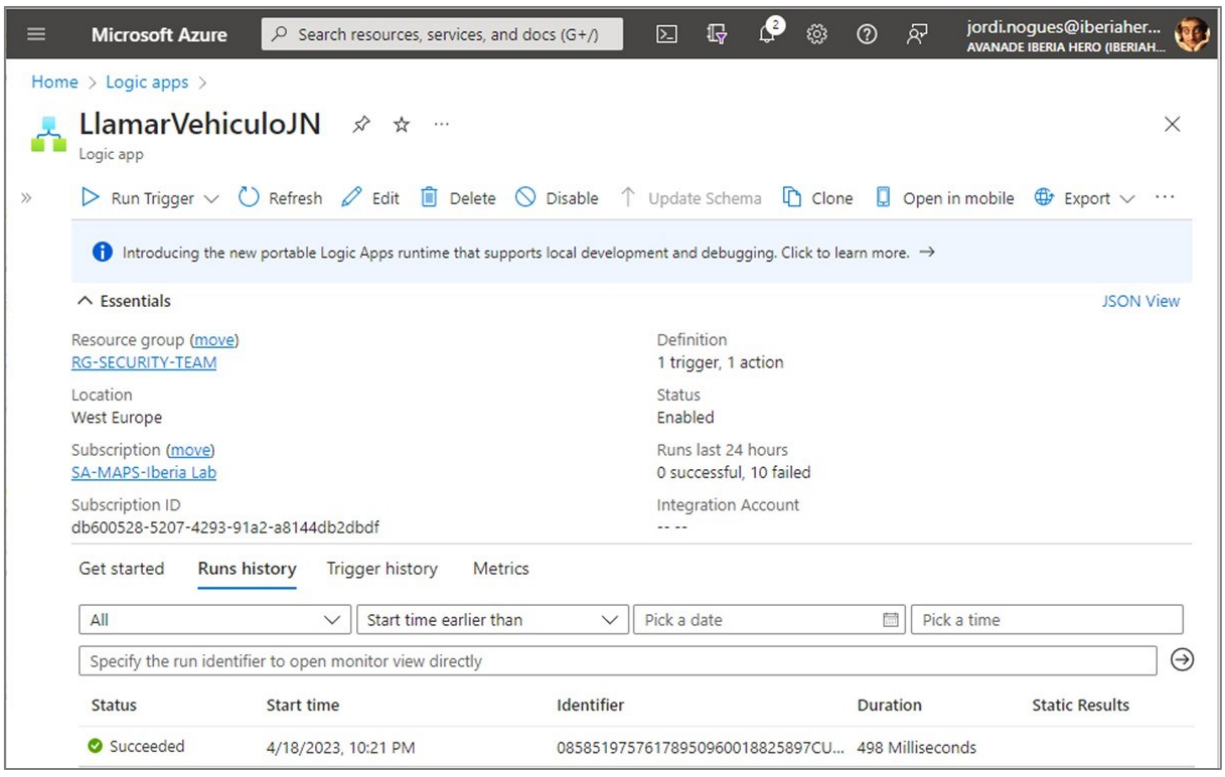


Ejecución del experimento

Una vez creado el playbook en Azure Logic Apps y configurada la regla analítica en Sentinel, y tras ejecutar nuevamente el proceso de envío de logs desde el concesionario al VSOC, desde la consola de Sentinel se observa el incidente que se ha creado automáticamente:



Y de la misma forma, en la consola de Azure Logic Apps se observa que se ha producido con éxito la ejecución automática del playbook que envía el aviso al vehículo:



5. Conclusiones

A lo largo del proceso de investigación y desarrollo de este trabajo, se generan reflexiones de diversas índoles.

En relación con el análisis del estado del arte de las tecnologías y los estándares de comunicación vehicular, la principal conclusión es que la continua evolución del 3GPP en su hoja de ruta habilita de forma sistemática nuevos escenarios de uso y de negocio, lo que a su vez origina nuevas superficies de ataque.

Aunque se puede contribuir a la mejora de la seguridad desde múltiples ángulos, el Foro Mundial para la Armonización de la Reglamentación sobre Vehículos (UN ECE WP.29) ha establecido unos criterios concretos que serán de obligado cumplimiento para la industria automotriz a partir de julio de 2024.

Siguiendo esos criterios y considerando la afirmación de Gartner que indica que “la ciberseguridad es una lucha sin fin, en la que sólo es cuestión de tiempo que una vulnerabilidad llegue a ser explotada” (Gartner, 2021), este trabajo se ha centrado en el diseño de un Centro de Seguridad de Operaciones Vehicular (VSOC), ya que esta herramienta constituye la base sobre la cual los equipos de ciberdefensa pueden trabajar para detectar amenazas y abordar incidentes en una etapa lo más temprana posible.

El diseño de la solución se ha enfocado partiendo de un modelo de SOC IT, es decir, orientado a la monitorización de redes de ordenadores, pero incluyendo los elementos necesarios para que esa monitorización sea capaz de llegar hasta los dispositivos IoT que controlan el comportamiento de los vehículos y las infraestructuras de transporte.

Desde el inicio, la meta de este proyecto era presentar de manera clara y comprensible los riesgos de ciberseguridad que afectan al entorno del vehículo conectado, así como las medidas necesarias para abordarlos. Esta meta puede considerarse alcanzada, ya que se han cumplido de manera fehaciente todos los objetivos intermedios previamente establecidos:

- **Identificación y análisis de los riesgos** *(capítulo 3: modelado de amenazas)*.
- **Descubrimiento del panorama regulatorio** al que se enfrentan los fabricantes de vehículos y su cadena de suministro, incluyendo las medidas y los plazos de implantación exigidos *(capítulo 2, sección 3: normativas de seguridad)*.
- **Definición de los elementos clave** en una arquitectura de ciberseguridad para el vehículo conectado *(capítulo 4, secciones 2 y 3: capacidades y arquitectura del VSOC)*.
- **Presentación del modelo de solución** a través de un caso de uso *(capítulo 4, secciones 4 y 5: definición del caso de uso, y experimento en el entorno de laboratorio)*.

Además, el proyecto ha cumplido sus objetivos respetando la planificación inicial. El principal riesgo de desviación provenía de la gran cantidad de literatura de calidad a considerar en la etapa de análisis del estado del arte, incluyendo protocolos de comunicación, normativas regulatorias del sector de la automoción, principios de ciberseguridad, soluciones de ciberdefensa, y tecnología IoT. Este riesgo fue mitigado aplicando un enfoque iterativo en las actividades de análisis y documentación, es decir, avanzando en el entregable conforme se progresaba en el análisis, balanceando ambas actividades.

Sin embargo, aunque se han cumplido sus objetivos, la contribución de este proyecto como guía para la implementación de un Centro de Operaciones de Seguridad en el ámbito del vehículo conectado representa únicamente un primer paso dentro de un proceso mucho más amplio y complejo.

Se ha desarrollado un caso de uso relativamente sencillo, basado en la identificación de códigos de dispositivo (ECUs) reportados como maliciosos en una “blacklist”, para gestionarlos de forma análoga a los IoCs (Indicadores de Compromiso) que típicamente proporcionan las fuentes de Inteligencia de Amenazas con direcciones IP o *hashes* de archivos identificados como maliciosos, pero existen muchos más casos de uso a explorar, como aquellos basados en mecanismos de inteligencia artificial, por ejemplo:

- Recepción de códigos no habituales
- Contenido atípico en los mensajes
- Frecuencia de envíos atípica
- Etcétera

En resumen, se ha aplicado un enfoque de “producto mínimo viable” para ilustrar adecuadamente los beneficios potenciales de la solución y su lógica de funcionamiento, pero limitado por los plazos que exige un Trabajo de Final de Grado. Por lo tanto, este proyecto está abierto a una posible continuidad, quizás en un futuro Trabajo de Fin de Máster.

6. Glosario de acrónimos y términos

3GPP:	3rd Generation Partnership Project.
4G-LTE:	4th Generation – Long Term Evolution.
5G-NR:	5th Generation – New Radio.
ADAS:	Advanced Driver Assistance System.
Amenaza:	Toda circunstancia o evento que potencialmente pueda afectar de forma adversa al ecosistema del vehículo a través de un acceso no autorizado al sistema que bloquee el servicio o que provoque la destrucción, alteración o divulgación indebida de la información.
Atacante:	Individuo, grupo, organización o gobierno que realiza o tiene la intención de realizar un ataque.
Botmaster:	Persona o entidad que controla una red de dispositivos infectados con *malware*, también conocidos como *botnets*.
Botnet:	Red de dispositivos infectados y controlados remotamente por un atacante, utilizada para llevar a cabo ataques cibernéticos.
CAN (BUS):	Controller Area Network.
CEF:	Common Event Format
CIO:	Chief Information Officer.
CISO:	Chief Information Security Officer.
CSMS:	Cyber Security Management System.
CVE:	Common Vulnerabilities and Exposures.
CVSS:	Common Vulnerability Scoring System.
DENM:	Decentralized Environmental Notification Message.
DSRC:	Dedicated Short Range Communications.
eCall:	Es un sistema de llamada de emergencia automatizado para vehículos, diseñado para proporcionar asistencia rápida en caso de un accidente automovilístico grave.
ECU:	Electronic Control Unit.
EDR:	Endpoint Detection and Response.
ETSI:	European Communications Standards Institute.
IDS:	Intrusion Detection System.
IEEE:	Institute of Electrical and Electronics Engineers
Infotainment:	Plataforma *hardware* y *software* que proporciona funciones de entretenimiento multimedia y navegación, así como información de diagnóstico y monitorización del vehículo, a través de una consola -que puede ser táctil- instalada en el salpicadero.
IoC:	Indicadores de Compromiso.
IoT:	Internet of Things.

IP:	Internet Protocol.
ISO:	International Standards Organization.
ITS:	Intelligent Transportation System.
ITU:	International Telecommunication Union.
JAIN-SIP:	Java APIs for Integrated Networks - Session Initiation Protocol.
JMF:	Java Media Framework.
JSON:	JavaScript Object Notation.
KQL:	Kusto Query Language (lenguaje de consultas utilizado por Microsoft Sentinel).
Malware:	Software malicioso diseñado para dañar, alterar o tomar control de un sistema informático sin el conocimiento o consentimiento del usuario.
Machine Learning:	Rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan.
MCRA:	Microsoft Cybersecurity Reference Architectures.
MIRAI:	Tipo de *malware* utilizado para crear *botnets*.
NIST:	National Institute of Standards and Technology.
NVD:	National Vulnerability Database.
OBD:	On-Board Diagnostics.
OBU:	On-board Unit (unidad embarcada en el vehículo).
OSI:	Open Systemss Interconnection.
OTA:	Over The Air (actualización del SW por vía inalámbrica).
RaaS:	Ransomware as a Service.
REST:	REpresentational State Transfer.
RSU:	Road-Side Unit (unidad instalada en la vía de transporte).
SAE:	Society of Automobile Engineers.
SIEM:	Security Information and Event Management.
SOAR:	Security Orchestration, Automation and Response.
SOC:	Security Operations Center.
SPAN:	Switched Port ANalyzer.
SUMS:	Software Update Management System.
TARA:	Threat Analysis and Risk Assessment.
TCP:	Transmission Control Protocol.
TI:	Tecnologías de la Información.
TOR.	The Onion Router (red de comunicaciones anónimas utilizada, entre otros, por los delincuentes cibernéticos).
TTP:	Tácticas, Técnicas y Procedimientos.
UDP:	User Datagram Protocol.
UEBA:	User and Entity Behavior Analytics.
UNECE:	United Nations Economic Commission for Europe.

USB:	Universal Serial Bus.
V2I:	Vehicle to Infrastructure communications.
V2V:	Vehicle to Vehicle communications.
V2X:	Vehicle to Everything communications.
Vector de ataque:	Camino o medio a través del cual el atacante puede conseguir el acceso a su objetivo con fines maliciosos.
VoIP:	Voice over IP.
VSOC:	Vehicle Security Operations Center.
Vulnerabilidad:	Debilidad de un activo que puede ser explotada por una o más amenazas.

7. Bibliografía

[1] J. D. Pedro Pacheco, «3 Steps to Implementing a Vehicle Security Operations Center,» Gartner, 2022.

[2] K. F. Karnouskos S, «Privacy and Integrity Considerations in Hyperconnected Autonomous Vehicles,» *Proceedings of the IEEE,* vol. 106, nº 1, pp. 160-170, 2018.

[3] C. Smith, The Car Hacker's Handbook (A Guide for the Penetration Tester), San Francisco: No Starch Press, 2016.

[4] A. B. J. S. G. Khan, "Framework for Calculating Residual Cybersecurity Risk of Threats to Road Vehicles in Alignment with ISO/SAE 21434," *Applied Cryptography and Network Security Workshops. ACNS 2022. Lecture Notes in Computer Science,* vol. 13285, pp. 235-247, 2022.

[5] N. S. F. B. a. G. F. R. Jacob, "Congestion-aware Packet Repetitions for IEEE 802.11bd-based Safety-critical V2V Communications," in *ICC 2022 - IEEE International Conference on Communications*, Seoul, Korea, 2022.

[6] A. Festag, «Standards for vehicular communication—from IEEE 802.11p to 5G.,» *Elektrotech. Inftech.,* vol. 132, nº 7, pp. 409-416, 2015.

[7] B. R. J. A. M. &. F. J. Fernandes, «Implementation and Analysis of IEEE and ETSI Security Standards for Vehicular Communications,» *Mobile Networks and Applications,* vol. 23, nº 3, pp. 469-478, 2018.

[8] L. S. G. L. R. R. Ali Z, «3GPP NR V2X Mode 2: Overview, Models and System-Level Evaluation,» *IEEE Access,* vol. 9, p. 89554–89579, 2021.

[9] A. L. J. F. Karen Brown, «The Top 5 Trends in Enterprise Networking and Why They Matter: A Gartner Trend Insight Report,» Gartner, 2022.

[10] R. L. D. Z. a. X. S. C. Lai, «Security and Privacy Challenges in 5G-Enabled Vehicular Networks.,» *IEEE Network,* vol. 34, nº 2, pp. 37-45, 2020.

[11] C. M. G. Schmittner, «Automotive Cybersecurity Standards - Relation and Overview,» *Lecture Notes in Computer Science,* vol. 11699, pp. 153-165, 2019.

[12] B. S. K. D. E. K. &. N. F. Scott McLachlan, «Tempting the Fate of the furious: cyber security and autonomous cars,» *International Review of Law, Computers & Technology,* vol. 36, nº 2, pp. 181-201, 2022.

[13] J. D. Pedro Pacheco, «How Automotive CIOs Can Lead a Successful Cybersecurity Implementation and Comply With WP.29 UN R155,» Gartner, 2021.

[14] D. L. J. K. S. Püllen, «ISO/SAE 21434-Based Risk Assessment of Security Incidents in Automated Road Vehicles,» *Computer Safety, Reliability, and Security. SAFECOMP 2021. Lecture Notes in Computer Science,* vol. 12852, pp. 82-97, 2021.

[15] A. Knight, Hacking Connected Cars. Tactics, Techniques, and Procedures, John Wiley & Sons, Inc., 2020.

[16] S. Woo, H. J. Jo y D. H. Lee, «A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN,» *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, nº 2, pp. 993-1006, 2015.

[17] U. H. Jayo, Sistemes telemàtics aplicats als sistemes de transport intel.ligent, Barcelona: UOC, 2019.

[18] K. Abboud, H. A. Omar y W. Zhuang, «Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey,» *IEEE transactions on vehicular technology,* vol. 65, nº 12, pp. 9457-9470, 2016.

[19] W. N. S. Z. H. Y. H. C. S. W. Yang Wang, «Architecture and key terminal technologies of 5G-based internet of vehicles,» *Computers and Electrical Engineering,* vol. 95, nº 107514, 2021.

[20] I. Y. a. N. G. V. Sharma, «Security of 5G-V2X: Technologies, Standardization, and Research Directions,» *IEEE Network,* vol. 34, nº 5, pp. 306-314, 2020.

[21] Y. Lee, S. Woo, Y. Song, J. Lee y D. H. Lee, «Practical Vulnerability-Information-Sharing Architecture for Automotive Security-Risk Analysis.,» *IEEE Access,* vol. 8, p. 120009–120018, 2020.

[22] R. K. Gorka Sadowski, «Improve Your Threat Detection Function With Deception Technologies,» Gartner Group, 2019.

[23] Y. W. S. L. J. S. Y. M. H. &. L. D. H. Lee, «Enhanced Android App-Repackaging Attack on In-Vehicle Network,» *Wireless Communications & Mobile Computing,* vol. 2019, 2019.

[24] M. S. P. S. John Collins, «SOC Model Guide,» Gartner Group, 2023.

[25] AUTO-ISAC, «Threat Detection, Monitoring and Analysis - Best Practice Guide,» 2019. [En línea]. Available: https://automotiveisac.com/best-practices-threat-detection. [Último acceso: 07 04 2023].

[26] A. D. M. S. Pete Shoard, «Magic Quadrant for Security Information and Event Management,» Gartner Group, 2022.

[27] Microsoft, «Arquitecturas de referencia de ciberseguridad de Microsoft,» 23 01 2023. [En línea]. Available: https://learn.microsoft.com/es-es/security/cybersecurity-reference-architecture/mcra. [Último acceso: 07 04 2023].

[28] J. M. S. Y. Yuri Diogenes, «Mitigate threats using Azure Sentinel,» de *Microsoft Security Operations Analyst*, Microsoft Press, 2022, pp. 185-302.

[29] E. Electronics, «OBD to RS232 interpreter,» 2017. [En línea]. Available: https://www.elmelectronics.com/wp-content/uploads/2016/07/ELM327DS.pdf. [Último acceso: 05 04 2023].

8. Anexos

8.1. Especificaciones técnicas de la OBU fabricada por Q-Free

C-ITS ON-BOARD UNIT PRODUCT SHEET

SUPPORTED STANDARDS

Architecture	ISO 21217/ETSI 320 665
ETSI transport and networking	EN 302 636 series
ETSI security standards	TS 103 097/102 940 series
EU PKI	C-ITS Point of Contact (CPOC) Protocol
ETSI media access	G5 series
IEEE	1609 series

ISO ITS STANDARDS

- ISO 21210 lower layer series for IPv6 networking
- ISO 24102 Management series
- ISO 29281 ITS Station series

ITS MESSAGES

- ETSI EN 102 637-2 Cooperative Awareness Basic Service (CAM)
- ETSI EN 302 637-3 Decentralized Environmental Notification Basic Service (DENM)
- ETSI TS 103 301 Facilities layer protocols and communication requirements for infrastructure services
- SAE J2735 DSRC Message Set Dictionary (BSM, TIM, PSM, SPAT, MAP, SRM, SSM)
- CEN ISO TS 19321 In-Vehicle Information (IVI)
- CEN ISO TS 19091 Signal Phase and Timing (SPAT), Maps (MAP), Signal Request Message (SRM), Signal Status Message (SSM)

APPLICATION ENVIRONMENTS

- Linux native applications in C/C++ for real-time, I/O, and computing-intensive tasks
- Data download, analysis, real-time view, and diagnostics
- Java-based OSGi environment for the most flexible and portable ITS applications

TECHNICAL SPECIFICATIONS

PRODUCT NUMBER

V2X OBU controller:	ITS803
V2X OBU antenna:	ITS804

HARDWARE SPECIFICATIONS

Wireless connectivity:	GSM/3G/LTE voice and internet, eSIM included, option for removable SIM
	IEEE 802.11p — two channels
	Wi-Fi AP or STA
	Bluetooth 5.1
Wired connectivity:	Gigabit Ethernet, RJ45
	CAN bus (3x)
	General purpose digital I/O
	USB-C
Power:	12-24 Volt DC, 20 Watts
	Rechargeable battery
Sensors:	GPS, Galileo, and GLONASS
	Accelerometer, gyro, and magnetometer
Audio:	Speakers built-in
Security:	Hardware security module
Main processor:	ARM® i.MX8, 4GB memory, 16GB flash drive
Enclosure:	Two piece design: Controller: 180 x 150 x 45 mm, IP41 Antenna: 100 x 30 x 20 mm, IP41

SOFTWARE SPECIFICATIONS

Operating system:	Linux, remote web management
Applications environment:	OSGi and Java, remote management
Communications and networking:	ETSI G5 and IEEE 802.11p
	ETSI GeoNetworking, IEEE 1609 WAVE
	IPv4/IPv6 and ITS Messaging

Controller unit, network connectors

Controller unit, power connector

Antenna unit

8.2. Especificaciones técnicas de la RSU fabricada por Q-Free

C-ITS ROADSIDE UNIT PRODUCT SHEET

SUPPORTED STANDARDS

Architecture	ISO 21217/ETSI 320 665
ETSI transport and networking	EN 302 636 series
ETSI security standards	TS 103 097/102 940 series
EU PKI	C-ITS Point of Contact (CPOC) Protocol
ETSI media access	ITS G5 series
IEEE	1609 series

ISO ITS STANDARDS

- ISO 21210 lower layer series for IPv6 networking
- ISO 24102 management series
- ISO 29281 ITS station series

ITS MESSAGES

- ETSI EN 102 637-2 Cooperative Awareness Basic Service (CAM)
- ETSI EN 302 637-3 Decentralized Environmental Notification Basic Service (DENM)
- ETSI TS 103 301 Facilities layer protocols and communication requirements for infrastructure services
- SAE J2735 DSRC Message Set Dictionary (BSM, TIM, PSM, SPAT, MAP, SRM, SSM)
- CEN ISO TS 19321 In-Vehicle Information (IVI)
- CEN ISO TS 19091 Signal Phase and Timing (SPAT), Maps (MAP), Single Request Message (SRM), Signal Status Message (SSM)

APPLICATION ENVIRONMENTS

- Linux native applications in C/C++ for real-time, I/O, and computing-intensive tasks
- Java-based OSGi environment for the most flexible and portable ITS applications

TECHNICAL SPECIFICATIONS

PRODUCT NUMBER

V2X RSU controller:	ITS801
V2X RSU antenna:	ITS802

HARDWARE SPECIFICATIONS

Wireless connectivity:	GSM/3G/LTE voice and internet, eSIM included, option for removable SIM
	IEEE 802.11p — two channels
	Wi-Fi AP or STA
	Bluetooth 5.1
Wired connectivity:	Gigabit Ethernet, RJ45
	USB-C
Power:	12-24 Volt DC, 20 Watts
Sensors:	Timing with GPS and GLONASS
Security:	Hardware security module
Main processor:	ARM® i.MX8, 4GB memory, 16GB flash drive
Enclosure:	Two piece design: Controller: 180 x 150 x 45 mm, IP41 Antenna: 110 x 75 x 30 mm, IP68

SOFTWARE SPECIFICATIONS

Operating system:	Linux, remote web management
Applications environment:	OSGi and Java, remote management
Communications and networking:	ETSI G5 and IEEE 802.11p
	ETSI GeoNetworking, IEEE 1609 WAVE
	IPv4/IPv6 and ITS Messaging

Controller unit, network connectors

Controller unit, power connector

Antenna unit

8.3. Especificaciones técnicas de la unidad de control de Q-Free

Adicionalmente a la OBU y la RSU, Q-Free fabrica también un módulo llamado "C-ITS Manager", que conectado a la red de RSUs proporciona un servicio centralizado de back-end con información relevante para los vehículos (seguridad, tráfico, medioambiente…). Esta información puede integrarse automáticamente en este módulo desde los centros de información del tráfico, o bien puede introducirse manualmente.

C-ITS MANAGER PRODUCT SHEET

EVENT AND SCENARIO MANAGEMENT

As the primary communication hub for external devices, the C-ITS Manager allows messaging for specific events or scenarios to be created manually or automatically activated based on established criteria.

MANUAL

Manually creating events, such as road work or obstacles on the roadway, can be achieved quickly and easily in the C-ITS by enter event parameters and it will appear:

- Type of event
- Location of the event
- Required speed limits
- Closed lanes

AUTOMATIC

Events can be automatically triggered based on established criteria, such as severe weather and ghost drivers. Data acquisition is carried out via the communication server and messaging is automatically generated from external data sources, such as a traffic management center.

SYSTEM INTEGRATION

External systems acquire data either automatically via traffic sensors or manually through the traffic management center SCADA system. The V2I data concentrator connects these systems via the communication server, processes them properly, and transfers them to the location of the appropriate road side units.

WEB APPLICATION

A user-friendly web interface provides:

- Geographical representation of the concerned highway section with all the V2I road side units and its properties (status of the unit, outgoing messages, etc.)
- Manual activation of DENM messages for specific RSU
- Manual activation of IVI messages
- System reports

V2I DATA CONCENTRATOR

Part of the C-ITS Manager, the V2I data concentrator acquires data from various data sources, processes it, and sends it to the appropriate RSU via the communication system.

Data is processed to determine the appropriate:

- Message type – DENM or IVI
- Message content according to the location of the RSU (distance to the event, appropriate speed limit, etc.)
- Message priority in terms of the co-existence of various different events

C-ITS MANAGER

Acquire, process, and communicate data with road side units

Store data and operate as the main hub for external sources

ROAD SIDE UNITS

Collect and distribute messages to and from vehicles/on-board units

Communicate data with C-ITS Manager

ON-BOARD UNITS

Receive, analyze, and display messages in vehicles

Display road side unit locations

Send messages to road side units

8.4. Comandos AT del protocolo ELM327

A continuación, se muestra la lista de comandos. La información detallada puede encontrarse en el manual del fabricante del chip ELM327, ELM Electronics [24].

AT Commands

Several parameters within the ELM327 can be adjusted in order to modify its behaviour. These do not normally have to be changed before attempting to talk to the vehicle, but occasionally the user may wish to customize these settings – for example by turning the character echo off, adjusting a timeout value, or changing the header bytes. In order to do this, internal 'AT' commands must be used.

Those familiar with PC modems will immediately recognize AT commands as a standard way in which modems are internally configured. The ELM327 uses essentially the same method, always watching the data sent by the PC, looking for messages that begin with the character 'A' followed by the character 'T'. If found, the next characters will be interpreted as an internal configuration or 'AT' command, and will be executed upon receipt of a terminating carriage return character. If the command is just a setting change, the ELM327 will reply with the characters 'OK', to say that it was successfully completed.

Some of the following commands allow passing numbers as arguments in order to set the internal values. These will always be hexadecimal numbers which must generally be provided in pairs. The hexadecimal conversion chart in the OBD Commands section (page 30) may be helpful if you wish to interpret the values. Also, you should be aware that for the on/off types of commands, the second character is the number 1 or the number 0, the universal terms for on and off.

The remainder of this page, and the two pages following provide a summary of all of the commands that the current version of the ELM327 recognizes. A more complete description of each command begins on page 12. Note that the settings which are shown with an asterisk (*) are the default values.

AT Command Summary

General Commands

<CR>	repeat the last command
BRD hh	try Baud Rate Divisor hh
BRT hh	set Baud Rate Timeout
D	set all to Defaults
E0, E1	Echo off, or on*
FE	Forget Events
I	print the version ID
L0, L1	Linefeeds off, or on
LP	go to Low Power mode
M0, M1	Memory off, or on
RD	Read the stored Data
SD hh	Save Data byte hh
WS	Warm Start (quick software reset)
Z	reset all
@1	display the device description
@2	display the device identifier
@3 ccccccccccc	store the @2 identifier

Programmable Parameter Commands

PP xx OFF	disable Prog Parameter xx
PP FF OFF	all Prog Parameters disabled
PP xx ON	enable Prog Parameter xx
PP FF ON	all Prog Parameters enabled
PP xx SV yy	for PP xx, Set the Value to yy
PPS	print a PP Summary

Voltage Reading Commands

CV dddd	Calibrate the Voltage to dd.dd volts
CV 0000	restore CV value to factory setting
RV	Read the input Voltage

Other

IGN	read the IgnMon input level

AT Command Summary (continued)

OBD Commands

AL	Allow Long (>7 byte) messages
AMC	display Activity Monitor Count
AMT hh	set the Activity Mon Timeout to hh
AR	Automatically Receive
AT0, 1, 2	Adaptive Timing off, auto1*, auto2
BD	perform a Buffer Dump
BI	Bypass the Initialization sequence
DP	Describe the current Protocol
DPN	Describe the Protocol by Number
H0, H1	Headers off*, or on
MA	Monitor All
MR hh	Monitor for Receiver = hh
MT hh	Monitor for Transmitter = hh
NL	Normal Length messages*
PC	Protocol Close
R0, R1	Responses off, or on*
RA hh	set the Receive Address to hh
S0, S1	printing of Spaces off, or on*
SH xyz	Set Header to xyz
SH xxyyzz	Set Header to xxyyzz
SH wwxxyyzz	Set Header to wwxxyyzz
SP h	Set Protocol to h and save it
SP Ah	Set Protocol to Auto, h and save it
SP 00	Erase stored protocol
SR hh	Set the Receive address to hh
SS	use Standard Search order (J1978)
ST hh	Set Timeout to hh x 4 msec
TA hh	set Tester Address to hh
TP h	Try Protocol h
TP Ah	Try Protocol h with Auto search

J1850 Specific Commands (protocols 1 and 2)

IFR0, 1, 2	IFRs off, auto*, or on
IFR H, S	IFR value from Header* or Source

ISO Specific Commands (protocols 3 to 5)

FI	perform a Fast Initiation
IB 10	set the ISO Baud rate to 10400*
IB 48	set the ISO Baud rate to 4800
IB 96	set the ISO Baud rate to 9600
IIA hh	set ISO (slow) Init Address to hh
KW	display the Key Words
KW0, KW1	Key Word checking off, or on*
SI	perform a Slow (5 baud) Initiation
SW hh	Set Wakeup interval to hh x 20 msec
SW 00	Stop sending Wakeup messages
WM [1 - 6 bytes]	set the Wakeup Message

CAN Specific Commands (protocols 6 to C)

CEA	turn off CAN Extended Addressing
CEA hh	use CAN Extended Address hh
CAF0, CAF1	Automatic Formatting off, or on*
CF hhh	set the ID Filter to hhh
CF hhhhhhhh	set the ID Filter to hhhhhhhh
CFC0, CFC1	Flow Controls off, or on*
CM hhh	set the ID Mask to hhh
CM hhhhhhhh	set the ID Mask to hhhhhhhh
CP hh	set CAN Priority to hh (29 bit)
CRA	reset the Receive Address filters
CRA hhh	set CAN Receive Address to hhh
CRA hhhhhhhh	set the Rx Address to hhhhhhhh
CS	show the CAN Status counts
CSM0, CSM1	Silent Monitoring off, or on*
CTM1	set Timer Multiplier to 1*
CTM5	set Timer Multiplier to 5

8.5. Script Powershell para envío de mensajes mediante Data Collector API

```
# Variables
$CustomerId = <Identificador del Workspace>
$jsonFilePath = "eventos CAN.json"
$SharedKey = <Clave primaria del Workspace>

# Leer el contenido del archivo JSON
$json = Get-Content -Path $jsonFilePath -Raw

# Nombre de la tabla en destino (custom log Type)
$logType = "Log_Concesionarios"

# Timestamp de la carga en Sentinel
$TimeStampField = ""

# Función para crear la firma utilizando la Shared Key
Function Build-Signature ($customerId, $sharedKey, $date, $contentLength, $method,
$contentType, $resource)
{
    $xHeaders = "x-ms-date:" + $date
    $stringToHash = $method + "`n" + $contentLength + "`n" + $contentType + "`n" +
$xHeaders + "`n" + $resource
    $bytesToHash = [Text.Encoding]::UTF8.GetBytes($stringToHash)
    $keyBytes = [Convert]::FromBase64String($sharedKey)

    $sha256 = New-Object System.Security.Cryptography.HMACSHA256
    $sha256.Key = $keyBytes
    $calculatedHash = $sha256.ComputeHash($bytesToHash)
    $encodedHash = [Convert]::ToBase64String($calculatedHash)
    $authorization = 'SharedKey {0}:{1}' -f $customerId,$encodedHash
    return $authorization
}

# Construcción de parámetros y cabeceras e invocación al API REst
Function Post-LogAnalyticsData($customerId, $sharedKey, $body, $logType)
{
    $method = "POST"
    $contentType = "application/json"
    $resource = "/api/logs"
    $rfc1123date = [DateTime]::UtcNow.ToString("r")
    $contentLength = $body.Length
    $signature = Build-Signature `
        -customerId $customerId `
        -sharedKey $sharedKey `
        -date $rfc1123date `
        -contentLength $contentLength `
        -method $method `
        -contentType $contentType `
        -resource $resource
    $uri = "https://" + $customerId + ".ods.opinsights.azure.com" + $resource + "?api-
version=2016-04-01"
    $headers = @{
        "Authorization" = $signature;
        "Log-Type" = $logType;
        "x-ms-date" = $rfc1123date;
        "time-generated-field" = $TimeStampField;
    }
    $response = Invoke-WebRequest -Uri $uri -Method $method -ContentType $contentType
-Headers $headers -Body $body -UseBasicParsing
    return $response.StatusCode
}

# Llamada a la función para envío de la solicitud al Data Collector API
Post-LogAnalyticsData -customerId $customerId -sharedKey $sharedKey -body
([System.Text.Encoding]::UTF8.GetBytes($json)) -logType $logType
```