Plataforma de demostración para ataques extremo a extremo de dispositivos con interfaz OBD-II

Ignacio Gutiérrez*, Gregorio López†, Roberto Gesteira-Miñarro‡ y Rafael Palacios§
Instituto de Investigación Tecnológica
Universidad Pontificia Comillas
*ign.gutierrez@alu.comillas.edu, †gllopez@comillas.edu, ‡rgesteira@comillas.edu, §rafael.palacios@iit.comillas.edu

***Resumen*—La ciberseguridad en vehículos ha cobrado un interés significativo en los últimos años, dada la superficie de ataque que sus sistemas de información exponen y el creciente marco normativo en materia de ciberseguridad. Un foco de interés en particular se sitúa en dispositivos que se conectan a los puertos de diagnóstico a bordo de los vehículos, cuyo acceso a los buses de comunicación internos del vehículo podría ser explotado por adversarios mediante las interfaces inalámbricas disponibles por estos. En este artículo se propone una arquitectura para una plataforma de pruebas y demostración de ataques dirigidos a este tipo de dispositivos que permita a los investigadores de ciberseguridad agilizar los procesos de análisis de vulnerabilidades y auditoría en estos dispositivos. El diseño se contextualiza sobre un caso de uso en el que se identifica el valor añadido que pueden aportar al investigador.**

***Index Terms*—Ciberseguridad, Vehículos, Diagnósticos a bordo, OBD, Dispositivos empotrados, Android**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

En los últimos años, los medios de transporte de uso común en nuestra sociedad han incorporado cada vez más funcionalidades dependientes de los sistemas de información. En los vehículos modernos, consolas centrales de entretenimiento, cámaras exteriores e incluso los sistemas básicos de control forman parte de una telaraña de ECUs (*Electronic Control Units*), sistemas empotrados heterogéneos de diversas capacidades que trabajan en conjunto para orquestar y supervisar los procesos físicos que ocurren en un vehículo como respuesta a las entradas del usuario y del entorno que le rodea. Nuevas funcionalidades y paradigmas, como la interoperabilidad con los dispositivos móviles, la electrificación de los vehículos e incluso la conducción asistida o automatizada, resultan en un aumento exponencial en la complejidad de estos sistemas. Como resultado de estos factores, la necesidad de considerar la ciberseguridad en los vehículos, ahora transformados en pequeños sistemas industriales con una considerable superficie de ataque, cada vez es más alta.

Bajo la perspectiva de un adversario, una de las superficies de ataque más atractivas en un vehículo es el acceso directo a los buses CAN (*Controller Area Network*) del vehículo, utilizado por los diversos ECUs como medio de comunicación mutua. El sistema de diagnóstico a bordo (OBD, del inglés *On-Board Diagnostics*), cuya inclusión y fácil acceso a los talleres mecánicos está regulada en la Unión Europea bajo el Reglamento (UE) 2018/858 [1] (previamente sobre la Directiva 98/69/EC [2]), constituye una forma de acceder a dichos buses. Aunque la implementación de un puerto OBD es obligatoria bajo esta normativa, sorprende la ausencia de aplicación de principios básicos de ciberseguridad como la segmentación de red en la arquitectura y diseño de los sistemas informáticos de un vehículo [3]. Como resultado, los puertos OBD pueden utilizarse como medio para explotar vulnerabilidades en sistemas internos del vehículo en una variedad de escenarios de ataque y con un diverso repertorio de impactos potenciales. El acceso al bus CAN del vehículo que el puerto OBD ofrece permite, por ejemplo, interactuar con las ventanillas de un coche [4], desbloquear el vehículo haciéndose pasar por la ECU que realiza la autenticación de la llave electrónica de ignición [5], e incluso desactivar los frenos durante la marcha [6], [7].

Fuera de su uso en los talleres de servicio de vehículos, estos puertos de diagnóstico cuentan con otros usos populares, que en su mayoría se materializan en los denominados *dongles* OBD (Fig. 1): dispositivos de pequeño tamaño que el usuario conecta al puerto OBD del vehículo para obtener información y telemetría sobre este con diversos fines legítimos. Por lo general, dichos dispositivos cuentan con conectividad externa, bien por Bluetooth o WiFi para el acceso "auto-servido" por parte del conductor mediante una aplicación móvil, o bien mediante conexión a la red de datos móviles (M2M) para casos de uso en los que se requiera una gestión centralizada. Por otro lado, el acceso directo al puerto OBD por parte de estos dispositivos es capaz de extender la superficie de ataque a los alrededores del vehículo obviando la necesidad de interacción física [4], o incluso posibilitando la explotación a través de Internet si el dispositivo utiliza la red de datos móviles como medio de comunicación.

La creciente complejidad de las plataformas computacionales empotradas en los vehículos, en combinación con el elevado impacto potencial en la envolvente de *safety* del vehículo como resultado de un ataque con éxito, ha despertado en los últimos años un creciente interés por parte de investigadores y expertos en ciberseguridad en auditar plataformas vehiculares y los diversos componentes que conforman su ecosistema con el fin de identificar vulnerabilidades concretas, modelar potenciales amenazas y cadenas de explotación, desarrollar mitigaciones y protecciones que dificulten los potenciales ciberataques resultantes e informar futuras regulaciones y estándares industriales.

Uno de los mayores obstáculos a la investigación en este campo tiende a ser el elevado coste de adquisición de un vehículo que pueda servir como demostrador, suponiendo una considerable barrera económica. Ante esta problemática, los investigadores en materia de ciberseguridad que buscan experimentar en este campo de estudio desde contextos académicos y/o industriales requieren de plataformas de pruebas mediante las cuales puedan auditarse este tipo de dispositivos en materia

Figura 1. Ejemplos de *dongles* OBD comerciales. De arriba a abajo, izquierda a derecha: (a) *BlueDriver* (modelo LSB2) de Lemur Vehicle Monitors, (b) *Bluetooth OBDII Reader* de Bafx Products, (c) lector genérico compatible con el protocolo ELM327.

de ciberseguridad a una fracción del coste. Este artículo propone el diseño de una plataforma de dicha índole orientada a la auditoría de *dongles* OBD mediante la integración de dispositivos y componentes *off-the-shelf*, proporcionando a los investigadores un conjunto común de herramientas que permitan la automatización de ataques sencillos y asistan en la elaboración de ataques más complejos.

El resto del artículo se organiza de la siguiente manera: la sección II profundizará sobre los requerimientos identificados sobre los cuales se guía la propuesta de diseño. La sección III describirá el diseño propuesto atendiendo a dichos requerimientos. La sección IV contextualizará el diseño propuesto mediante un caso de uso centrado en la vulnerabilidad CVE-2016-2354 [8] previamente identificada en dispositivos *BlueDriver* de la marca *Lemur Vehicle Monitors*, describiendo cómo podría utilizarse la plataforma propuesta y las capacidades que ofrece para identificar vulnerabilidades similares. Finalmente, la sección V proporcionará conclusiones y describirá el trabajo futuro alrededor de este proyecto.

II. Análisis de requerimientos

Una auditoría efectiva de un sistema de información requiere considerar su comportamiento en contexto de cómo recoge, procesa e interactúa con su entorno. En el caso de un *dongle* OBD, este se comunica principalmente por medio del puerto OBD con el bus CAN del vehículo, e interactúa con otros sistemas o dispositivos mediante interfaces tipo Bluetooth, WiFi o redes móviles M2M (*machine-to-machine*) según la lógica definida en su *firmware*. Bajo el alcance de este estudio, el objetivo principal que se considerará dentro del alcance de las pruebas que se buscan realizar es auditar dispositivos que se comuniquen con una aplicación móvil vía Bluetooth.

Con el fin de interactuar con el dispositivo, este ofrece un protocolo de alto nivel a través del cual el usuario puede recoger la información del vehículo y realizar las acciones deseadas. Muchos dispositivos implementan el protocolo propietario ofrecido por el producto ELM327, originalmente desarrollado por ELM Electronics; como resultado, hoy día se ha convertido en estándar *de facto*, con un diverso ecosistema de dispositivos y aplicaciones móviles que soportan dicho protocolo. Otros dispositivos implementan protocolos propietarios, en conjunto con aplicaciones móviles específicas para acceder al dispositivo en cuestión.

Por lo general, un número significativo de estos dispositivos se comunican con los diversos ECUs vía el protocolo de bus CAN[1] con el propósito de recoger telemetría del vehículo y/o mensajes diagnósticos, replicando la funcionalidad ofrecida en las herramientas utilizadas por los talleres mecánicos para diagnosticar fallos en los diversos subsistemas del vehículo.

Derivado de este contexto, se identifican varias capacidades deseadas de la plataforma a diseñar, centrados por un lado en la **superficie de ataque** a estudiar, y por otro en los **tipos de ataques** que se busca realizar. Por un lado, entre las potenciales superficies de ataque que se buscan auditar dentro del alcance de esta plataforma, se encuentran:

- La **aplicación móvil** de interacción con el dispositivo,
- La propia **capa de comunicación** (Bluetooth, WiFi, etc.) mediante la cual la aplicación se comunica con el dispositivo.

Sobre estas superficies de ataque, se busca realizar, entre otros, los siguientes tipos de ataques:

- Ingeniería inversa.
- Control a bajo nivel.
- Ataques de hombre en el medio (del inglés *man-in-the-middle*, MitM).
- *Fuzzing*.
- Explotación de vulnerabilidades a nivel de *firmware*.

Adicionalmente, se identifican una serie de **requerimientos no funcionales** a considerar en el diseño de la plataforma:

1. **Accesibilidad**, contando con capacidades y herramientas listas para usar, ofreciendo así una base mediante la cual realizar ataques y recoger resultados en menor tiempo.
2. **Eficacia**, asistiendo a los investigadores en las labores de auditoría mediante la automatización de pruebas sencillas, liberando tiempo para pruebas más complejas y especializadas.
3. **Extensibilidad**, permitiendo su evolución continua a medida que nuevas innovaciones tecnológicas llegan al mercado.
4. **Coste**, permitiendo a una audiencia diversa beneficiarse de las capacidades ofrecidas.

III. Descripción de la solución

Para responder a las necesidades identificadas, se ha determinado un diseño que emplea los siguientes componentes (véase Fig. 2):

1. Un **dispositivo Android** en el cual se pueda instalar la aplicación proporcionada por el proveedor (o aplicaciones desarrolladas por terceros, en el caso de *dongles* OBD empleando estándares *de facto* como el protocolo ELM327 previamente mencionado) con la cual comunicarse con el dispositivo a probar. Para el demostrador de esta plataforma, se utiliza un dispositivo Xiaomi Redmi 9.
2. Un **simulador de ECU**, aparato electrónico con un puerto OBD-II al cual se conecta el dispositivo a probar,

[1] O bien, según la antigüedad del vehículo, utilizando otros protocolos como ISO 9141-2 [9].

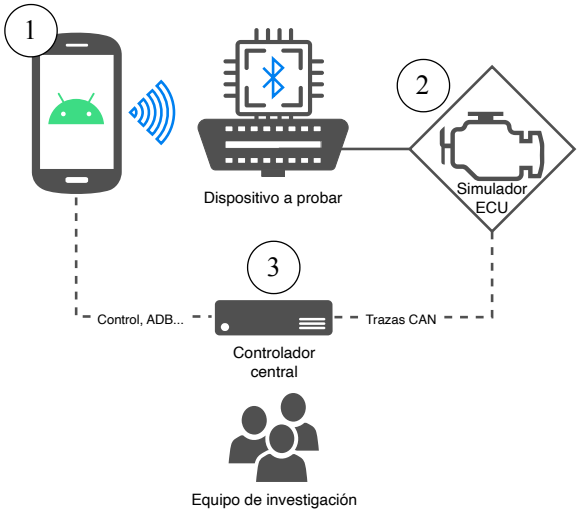

Figura 2. Diagrama del demostrador con los elementos principales indicados. *El robot de Android se reproduce o modifica a partir del trabajo generado y compartido por Google, y se usa conforme a lo descrito en la Licencia de Atribución de Creative Commons 3.0.*

presentándose a este último como una de las ECUs de un vehículo y proporcionando una emulación parcial de los mensajes que un vehículo podría intercambiar vía el bus CAN. Para el demostrador, se utiliza un simulador fabricado y distribuido por Lianghung Co [10]. Este simulador en particular permite la simulación de múltiples protocolos además del estándar CAN, con la posibilidad adicional de capturar paquetes entrantes en tiempo real.

3. Un **controlador central**, cuyo propósito es centralizar las labores y herramientas utilizadas en el análisis y experimentación con el resto de la plataforma. Para el demostrador, se utiliza una Raspberry Pi 4B, con la opción de utilizar un dispositivo de similares características.

Con respecto a (1), las capacidades de depuración y análisis ofrecidas por el sistema operativo Android hacen posible estudiar las interacciones entre ambos, interceptar comunicaciones y actualizaciones de *firmware* e incluso modificar el comportamiento de la aplicación; esto permite experimentar con una diversa gama de escenarios de ataque, desde ataques MitM tanto en la comunicación con el dispositivo como en las conexiones a Internet hasta la explotación de vulnerabilidades en el *dongle* OBD o incluso en la propia aplicación móvil.

Por otro lado, al evitar requerir el uso de un vehículo real, el uso de un simulador de ECU (2) simplifica el diseño de la plataforma, reduce su coste total y permite realizar pruebas más arriesgadas que podrían comprometer la disponibilidad de un vehículo real.

Los diversos componentes de la plataforma se conectan al controlador central (3), y el investigador interactúa con estos dispositivos mediante las herramientas disponibles en el controlador central. Por ejemplo, el dispositivo Android se conecta al controlador central mediante USB, y se interactúa con este mediante el protocolo ADB (*Android Debug Bridge*) con el fin de instalar aplicaciones que comuniquen con los dispositivos, monitorizar su comportamiento y realizar labores de depuración, y gestionar el propio dispositivo Android. Del mismo modo, el simulador de ECU podría conectarse mediante USB con el fin de recoger las comunicaciones realizadas en el bus CAN.

Además de estas capacidades, se disponen de herramientas con las cuales interactuar con los dispositivos conectados, y analizar y procesar los datos recibidos. Se propone, entre otros, utilizar el programa de código abierto **MobSF** (*Mobile Security Framework*) [11] en el controlador central, permitiendo realizar análisis estático y dinámico sobre la aplicación Android bajo estudio. Mediante automatismos para gestionar los componentes del sistema y los datos que se recogen de estos, en combinación con integraciones entre las diversas herramientas según corresponda, se busca mejorar la eficiencia y estandarizar un marco de trabajo común que ayude al investigador a conseguir resultados más rápido.

Disponer de un controlador central abierto y compacto con diversas interfaces (WiFi, Bluetooth, USB, SPI, I^2C, UART, etc.) permite integrar *hardware* adicional para expandir y modificar arbitrariamente las capacidades de la plataforma según requiera la situación. Por ejemplo, podrían realizarse ataques a la capa de comunicación Bluetooth bien con el *hardware* del propio controlador central o con un adaptador Bluetooth externo, o podría interactuarse con el dispositivo en pruebas mediante interfaces de depuración físicas como JTAG o UART, dentro de lo que sea posible.

IV. Caso de estudio y aplicación

A continuación, se propone un escenario de investigación utilizando una vulnerabilidad real descubierta en un dispositivo comercial. A partir de este escenario, se identifican en el proceso de investigación de dicha vulnerabilidad aplicaciones en las que la plataforma pueda aportar un valor añadido al investigador.

En Abril de 2016, una investigación realizada por Dan Klinedinst, por aquel entonces uno de los investigadores en vulnerabilidades de sistemas TI del CERT/CC (*Universidad Carnegie Mellon*), identificó que los dispositivos *BlueDriver* comercializados por Lemur Vehicle Monitors no proporcionaban autenticación sobre la interfaz Bluetooth. A través de esta se podía leer el estado del vehículo mediante la aplicación móvil del fabricante [8]. Un adversario en proximidad del vehículo podía, conectándose vía Bluetooth al dispositivo, enviar comandos arbitrarios al puerto OBD-II mediante el cual el dispositivo interactúa con el bus CAN del vehículo, sirviendo como potencial punto de entrada para realizar explotación adicional. Debe destacarse que no sería necesario acceder físicamente al interior del vehículo, siendo la comunicación Bluetooth con el dispositivo el punto de entrada utilizado para explotar la vulnerabilidad.

El descubrimiento causó cierto revuelo en los medios de comunicación [12], [13], y Lemur Vehicle Monitors rápidamente liberó actualizaciones tanto para su aplicación móvil como para el *firmware* del dispositivo en cuestión. La remediación principal consistió en provocar que el dispositivo dejase de aceptar nuevas conexiones Bluetooth tras un periodo de 60 segundos, forzando a los usuarios a físicamente desconectar y reconectar el dispositivo del vehículo para poder conectarse al dispositivo mediante la aplicación móvil. Adicionalmente, se implementó cifrado en la conexión entre la aplicación y

el dispositivo, y el protocolo de comunicación entre ambos cambió para restringir los comandos CAN que la aplicación podía realizar mediante el dispositivo [14].

En el proceso de investigación de esta vulnerabilidad, la plataforma podría aportar valor añadido en varias actividades:

1. El investigador puede instalar la aplicación móvil propietaria del proveedor en el dispositivo móvil de la plataforma. Mediante las herramientas de análisis estático y dinámico que se proporcionan, el investigador puede determinar la falta de autenticación en la capa de comunicaciones Bluetooth.
2. Del mismo modo, mediante el análisis de la aplicación y la intercepción de las comunicaciones Bluetooth entre la aplicación y el dispositivo, el investigador puede determinar que existe la capacidad de enviar tramas CAN arbitrarias a través del dispositivo, mediante las cuales podría llegarse a comprometerse la seguridad del vehículo.
3. Realizando ingeniería inversa del protocolo, modificando la aplicación o incluso alterando la lógica del programa y/o la memoria volátil sobre la cual el programa opera, sería posible enviar paquetes arbitrarios al dispositivo; entre otros, esto permitiría realizar ataques de *fuzzing* "hechos a medida" sobre el dispositivo siendo auditado, para lo cual existen técnicas específicas de aplicación en dispositivos IoT [15], [16]. El éxito de los ataques podría determinarse examinando las comunicaciones ocurridas en el bus CAN, que se registran por el simulador de ECU y se envían al controlador central.

Adicionalmente, podrían validarse las remediaciones realizadas posteriormente por el fabricante, verificando que arreglan o mitigan las vulnerabilidades originalmente identificadas, y que estas no introducen vulnerabilidades adicionales. Mediante las capacidades que ofrece, la plataforma puede así asistir en diversas fases del proceso de investigación.

V. Conclusiones y trabajo futuro

En este trabajo se ha propuesto, en base a un conjunto de requerimientos tanto funcionales como no funcionales, un diseño de una plataforma de pruebas y demostración para ataques a *dongles* OBD que busca permitir a los investigadores de ciberseguridad agilizar los procesos de análisis de vulnerabilidades y auditoría en estos dispositivos, y se ha ilustrado su utilidad a través un caso de estudio de relevancia en la industria.

En trabajo futuro se busca utilizar la plataforma para identificar nuevas vulnerabilidades en varios dispositivos de este tipo, aplicando metodologías de auditoría de seguridad relevantes como IoTSF [17] y BSAM [18]. En una fase inicial, se centrará la investigación en los dispositivos ilustrados en la Fig. 1. Adicionalmente, se busca experimentar con las capacidades de expansión de la plataforma, añadiendo funcionalidades como la intercepción de comunicaciones Bluetooth mediante la integración de *chipsets* como el NRF52. Finalmente, sería posible generalizar y expandir el enfoque el diseño propuesto en este trabajo a efectos de permitir el análisis de otros tipos de dispositivos IoT (*Internet-of-Things*).

Referencias

[1] "Regulation (EU) 2018/858 of the European Parliament and of the Council of 30 May 2018 on the approval and market surveillance of motor vehicles and their trailers, and of systems, components and separate technical units intended for such vehicles, amending Regulations (EC) No 715/2007 and (EC) No 595/2009 and repealing Directive 2007/46/EC (Text with EEA relevance.)," May 2018, legislative Body: EP, CONSIL. [Online]. Available: http://data.europa.eu/eli/reg/2018/858/oj/eng

[2] "Directive 98/69/EC of the European Parliament and of the Council of 13 October 1998 relating to measures to be taken against air pollution by emissions from motor vehicles and amending Council Directive 70/220/EEC," Oct. 1998. [Online]. Available: http://data.europa.eu/eli/dir/1998/69/oj/eng

[3] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental Security Analysis of a Modern Automobile," in *2010 IEEE Symposium on Security and Privacy*. Oakland, CA, USA: IEEE, 2010, pp. 447–462. [Online]. Available: http://ieeexplore.ieee.org/document/5504804/

[4] A. Luo and S. Hsieh, "Remotely Hacking a car through an OBD-II Bluetooth Dongle," Automotive Security Research Group, Jul. 2023. [Online]. Available: https://www.youtube.com/watch?v=f19_BNgVrWQ

[5] K. Tindell, "CAN Injection: keyless car theft," Apr. 2023. [Online]. Available: https://kentindell.github.io/2023/04/03/can-injection/

[6] A. Greenberg, "Hackers Remotely Kill a Jeep on the Highway—With Me in It," *Wired*, Jul. 2015, section: tags. [Online]. Available: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[7] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Proceedings of the 20th USENIX Conference on Security*, ser. SEC'11. San Francisco, CA, USA: USENIX Association, 2011, p. 6. [Online]. Available: https://www.usenix.org/legacy/events/sec11/tech/full_papers/Checkoway.pdf

[8] D. Klinedinst, "CERT/CC Vulnerability Note VU#615456," Apr. 2016. [Online]. Available: https://www.kb.cert.org/vuls/id/615456

[9] ISO Central Secretary, "Road vehicles – Diagnostic systems – Part 2: CARB requirements for interchange of digital information," International Organization for Standardization, Geneva, CH, Standard ISO 9141-2:1994, 1994. [Online]. Available: https://www.iso.org/standard/16738.html

[10] Lianghung Co, "OBD-II ECU Simulator CAN BUS." [Online]. Available: https://vehelec.com/products/obd-ii-ecu-simulator-iso15765-iso9141-2-iso14230-can-bus

[11] A. Abraham and MobSF Collaborators, "MobSF/Mobile-Security-Framework-MobSF," Mar. 2024, original-date: 2015-01-31T04:36:01Z. [Online]. Available: https://github.com/MobSF/Mobile-Security-Framework-MobSF

[12] M. Honorof, "This Bluetooth Car Dongle Might Just Kill You," Apr. 2016. [Online]. Available: https://www.tomsguide.com/us/lemur-bluedriver-security-flaw,news-22521.html

[13] P. Roberts, "CERT: Aftermarket Add-On Opens Cars To Life Threatening Hacks," Apr. 2016. [Online]. Available: https://securityledger.com/2016/04/cert-warns-on-hacking-risk-to-bluedriver-plug-in/

[14] BlueDriverSupport, "Reply to "Friendly PSA: BlueDriver OBD2 Scanner Company (Lemur Vehicle Monitors) Seemingly Pulled from Market"," Nov. 2018. [Online]. Available: www.reddit.com/r/MechanicAdvice/comments/9t6e7m/friendly_psa_bluedriver_obd2_scanner_company/e8ujm1h/

[15] J. Chen, W. Diao, Q. Zhao, C. Zuo, Z. Lin, X. Wang, W. C. Lau, M. Sun, R. Yang, and K. Zhang, "IoTFuzzer: Discovering Memory Corruptions in IoT Through App-based Fuzzing," in *Proceedings 2018 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2018. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_01A-1_Chen_paper.pdf

[16] X. Feng, R. Sun, X. Zhu, M. Xue, S. Wen, D. Liu, S. Nepal, and Y. Xiang, "Snipuzz: Black-box fuzzing of iot firmware via message snippet inference," 2021.

[17] "IoTSF IoT Security Assurance Framework Release 3.0," Nov. 2021. [Online]. Available: https://iotsecurityfoundation.org/wp-content/uploads/2021/11/IoTSF-IoT-Security-Assurance-Framework-Release-3.0-Nov-2021-1.pdf

[18] Tarlogic, "BSAM: Bluetooth Security Assessment Methodology." [Online]. Available: https://www.tarlogic.com/bsam/