Hacking Automóvel

Modelo de exploração de vulnerabilidades

Sérgio António Monteiro e Silva

Dissertação para obtenção do Grau de Mestre em

Segurança de Informação e Direito no Ciberespaço

Orientador(es): Prof. Dr. Miguel Nuno Dias Alves Pupo Correia

Júri

Presidente: Prof. Dr. Carlos Manuel Costa Lourenço Caleiro
Orientador: Prof. Dr. Miguel Nuno Dias Alves Pupo Correia
Vogal: Prof. Dr. António Casimiro Ferreira da Costa

Dezembro 2020

Dedicado a todos os que pensam fora da caixa e tornam os sistemas mais seguros, encontrando e corrigindo vulnerabilidades.

Agradecimentos

Quero agradecer em primeiro lugar ao orientador Professor Doutor Miguel Nuno Dias Alves Pupo Correia, do Instituto Superior Técnico de Lisboa, por toda a sua disponibilidades, compreensão e acompanhamento na elaboração desta dissertação.

Gostava de expressar a minha gratidão por todos os profissionais de cibersegurança, que testam os sistemas e os tornam mais seguros.

A todos os familiares e amigos que me apoiaram nesta caminhada, contribuindo assim para que eu tivesse conseguido chegar ao fim desta etapa.

Muito obrigado

Resumo

O mundo é cada vez mais digital. O conceito de ciberespaço começa a ultrapassar a realidade analógica a uma velocidade alucinante. Desta forma, os sistemas informáticos são o alvo de uma geração emergente de criminosos que se aproveitam das vulnerabilidades de cada sistema para ter acesso a informação segura ou para provocar danos nos sistemas de modo a que não possam funcionar.

Neste contexto a indústria automóvel é forçada a dotar os seus veículos de funcionalidades, componentes e sistemas interligados, que permitem não só uma maior eficiência e segurança das viaturas, mas também a disponibilização ao condutor e seus ocupantes de uma série de comodidades que lhe permitem obter informação em tempo real, aceder à *Internet* e controlar remotamente o funcionamento do veículo.

Um veículo moderno é, então, composto, para além dos componentes mecânicos habituais, por redes de comunicação, protocolos, processadores e *firmware*. É nesta componente tecnológica, que surgem as vulnerabilidades que podem ser exploradas pelos *hackers*.

Este trabalho consiste na identificação das principais superfícies de ataque de uma viatura e mapeamento dos vectores de ataque associados, com o objectivo de construir um modelo de *hacking* automóvel. Partindo de 3 vectores de ataque: Wi-Fi, Bluetooth e OBD-II, o modelo foi aplicado a 3 viaturas e classificaram-se as vulnerabilidades, segundo o sistema de classificação de CVSS.

Mesmo em viaturas recentes, com menos de um ano, foram encontradas várias vulnerabilidades associadas à possibilidade do *tracking* do automóvel por Wi-Fi e Bluetooth, bem como na exploração do sistema de comunicações interno com injeção de pacotes que levam ao controlo remoto de componentes como os piscas ou a buzina. O sistema Wi-Fi é, sem dúvida, o mais permeável, sendo consequência da adopção de protocolos de cifra com vulnerabilidades identificadas, bem como com a falta de isolamento dos equipamentos que se ligam a essa rede.

Como conclusão, são evidentes as falhas de segurança na construção dos automóveis potenciando o *hacking* automóvel, de frisar que não é necessário um nível de conhecimento muito elevado para executar estes ataques dado que, hoje em dia, as ferramentas que existem conseguem executar algumas destas explorações, de forma automática. É notória a necessidade de testes de exploração exaustivos, bem como a adopção de normas e procedimentos de cibersegurança por parte dos construtores, sendo a maior dificuldade a retrocompatibilidade com alguns protocolos, como é o caso do CANbus.

Palavras-chave: Cibersegurança; Automóvel; CANbus; Hacking

Abstract

The world is more and more digital. The concept of cyberspace begins to surpass the analogical reality at breakneck speed. In this way, computer systems are the target of an emerging generation of criminals who take advantage of vulnerabilities of each system to have access to secure information or to damage systems so that they cannot function.

In this context, the automotive industry is forced to provide its vehicles with interconnected features, components and systems, which allow not only greater efficiency and safety of the vehicles but also the provision to the driver and its occupants of a series of amenities that allow them to obtain information in real time, access the Internet and remotely control the operation of the vehicle.

A modern vehicle is then composed, in addition to the usual mechanical components, by communications networks, protocols, processors and source code. And it is in this technological component that vulnerabilities arise that can be exploited by hackers.

This work consists of identifying the main attack surfaces of a vehicle and mapping the associated attack vectors, with the aim of building a model of car hacking. Starting from 3 attack vectors: Wi-Fi, Bluetooth and OBD-II, the model was applied to 3 vehicles and the vulnerabilities were classified according to the CVSS classification system.

Even in recent vehicles less than a year old, several vulnerabilities were found associated with the possibility of tracking the car via Wi-Fi and Bluetooth, as well as in the exploitation of the internal communications system with packet injection that leads to the remote control of components such as the turn signals or the horn. The Wi-Fi system is undoubtedly the most permeable, a consequence of the adoption of encryption protocols with identified vulnerabilities as well as the lack of isolation of the equipment that connect to this network.

As a conclusion, the security flaws in the construction of automobiles are evident, enhancing car hacking, to emphasize that it is not necessary a very high level of knowledge to execute these attacks, given that nowadays the tools that exist manage to execute some of these explorations automatically. There is a clear need for exhaustive exploration tests as well as the adoption of cybersecurity standards and procedures by builders, with the greatest difficulty being backward compatibility with some protocols such as CANbus.

Keywords: Cybersecurity, Vehicle, CANbus, Hacking

Conteúdo

Lista de Tabelas

Lista de Figuras

Lista de Símbolos

Subscritos

AP Access Point

BSSID Basic Service Set Identifier

CAN Controller Area Network

CNCS Centro Nacional de Cibersegurança

CVE Common Vulnerabilities and Exposures

DLC Data Link Connector

DNS Domain Name Server

DOS Denial of Service

GSM Global System for Mobile Communications

IDE Identifier extension

kbps Kilobits Per Second

OBD-II On Board Diagnostics

OEM Original Equipment Manufacturer

OTA Over-the-Air update

SMS Short Message Service

SQL Structured Query Language

SSID Service Set Identifier

SUV Sport utility vehicle

TPMS Tire-Pressure Monitoring System

USB Universal Serial Bus

Capítulo 1

Introdução

Foi na década de 90, pouco tempo após ter tirado a carta de condução, que consegui, com muito custo, comprar um *Citroën* Zx 1.6 em quarta ou quinta mão. Esta aquisição transformou completamente a minha vida. De um momento para o outro, deixei de estar dependente de transportes públicos e podia chegar mais rápido a qualquer lugar, desde que tivesse dinheiro para a gasolina.

Analogamente ao impacto da aquisição de uma viatura na minha vida, é exactamente igual ao impacto que a indústria automóvel tem tido na vida de milhões de pessoas, durante séculos. Em 1679, um belga de nome *Ferdinand Verbiest*[1] [1], apresentou ao imperador da China um protótipo do primeiro carro a vapor. No entanto, pela sua reduzida dimensão, não era capaz de transportar passageiros, sendo quase uma prova de conceito.

No decorrer dos seguintes séculos, existiu uma evolução constante do automóvel. Em 1770, *Nicolas-Joseph Cugnot*[2] desenvolveu um veículo movido a vapor que tinha como objectivo o transporte de canhões e, embora esta viatura tivesse vários problemas de funcionamento, foi a ideia base para outro projectos.

O nome de *Karl Benz*[3] [2] é o mais consensual quando se fala da invenção do automóvel. *Benz*, inspirado nas correntes das bicicletas, desenvolveu o *Benz Patent Motorwagen*[4]. Foi ainda o responsável por inúmeras patentes ligadas ao automóvel e, que continuamos a usar nos dias de hoje como, por exemplo: as velas do motor, a embraiagem, o carburador, mudanças de velocidades e outras.

Foi, no entanto, outro nome que fez com que os automóveis chegassem ao público em geral. Em 1902, *Ransom Eli Olds*[5] começou a fabricar automóveis em grande escala e, em 1917, *Henry Ford* pegou neste modelo de produção e montou as linhas de produção, do famoso *Model T*[6] que viria

[1] Astrónomo, matemático, cientista, jesuíta e missionário católico flamengo.
[2] Engenheiro militar francês.
[3] *Karl Friedrich Benz*, foi um engenheiro de automóveis e inventor de motores de combustão.
[4] A patente foi concedida em 1886, ao primeiro carro movido a gasolina.
[5] Fundador da empresa *Oldsmobile*.
[6] Modelo barato, robusto e que qualquer um conseguia consertar, esteve em produção durante 19 anos.

a mudar a vida de milhões de pessoas, encurtando distâncias e possibilitando que o transporte de pessoas e cargas fosse feito de forma mais fácil e eficiente [3].

Em mais de 100 anos, o automóvel evoluiu de uma forma incrível. É importante notar que, durante toda a evolução do automóvel, também foi evoluindo a área da electrónica. A evolução dos inúmeros circuitos electrónicos veio a substituir alguns processos, puramente mecânicos, que existiam. O famoso *Volkswagen* carocha[7] era quase sempre reparável com um arame e alicate. Actualmente, as novas viaturas são, simplesmente, computadores com rodas, onde dezenas de controladores comunicam em tempo real entre eles e tomam decisões que influenciam a condução, com o objectivo de tornarem os automóveis mais seguros e, em alguns casos, autónomos.

Como não existem sistemas totalmente seguros um automóvel também tem vulnerabilidades e, pode sofrer um ciberataque, podendo causar a perda de vidas humanas.

1.1 Motivação

A segurança de um automóvel, hoje em dia, depende da cibersegurança. No entanto, os construtores e fabricantes de peças OEM[8] tendem a não implementar componentes seguros. *Hardware*, *software* e canais de comunicação são repletos de vulnerabilidades que podem ser exploradas por actores mal intencionados.

Aquando do início da utilização de computadores nos automóveis, nunca foi pensado que estes poderiam ser atacados e as suas comunicações adulteradas fazendo com que a viatura fosse levada a executar uma acção estranha como, por exemplo: guinar para a esquerda quando circula a 120 km/h.

Existe uma enorme comunidade envolvida no *hacking* automóvel[9], e têm sido descobertas muitas vulnerabilidade e respectivos *exploits*. Este cenário é muito preocupante, tendo em conta que muitos modelos não têm forma de ser actualizados, ou seja, temos actualmente carros em circulação, com 8 a 10 anos, com vulnerabilidades identificadas que podem ser facilmente exploradas. Estas viaturas permanecerão vulneráveis, dado que o fabricante não tem como proceder à actualização dos sistemas, fora da sua rede de oficinas.

Desta forma, este documento tem como principal motivação a identificação de vectores de ataque e a consciencialização para que a indústria automóvel desenhe sistemas mais resilientes do ponto de vista da cibersegurança. A consciencialização passa sempre pela demonstração prática das vulnerabilidades, pelo que será desenvolvido um modelo de ataque contra vários modelos de automóvel e reportados os resultados desses ataques.

[7] Este foi o primeiro modelo fabricado pela companhia alemã, *Volkswagen*, tendo sido o carro mais vendido no mundo.
[8] *Original Equipment Manufacturer*
[9] Um exemplo destas comunidades é a *Car Hacking Village*, disponível em `https://www.carhackingvillage.com`, que fornece uma série de recursos gratuitos para todos os que queiram aprender a explorar os sistemas de um automóvel.

1.2 Objetivos

O *hacking* automóvel é uma área nova. Cada construtor utiliza protocolos e tabelas de funções próprias que, muitas vezes, variam de modelo para modelo, logo o mapeamento é um trabalho moroso e economicamente dispendioso, dado que têm de ter acesso a cada modelo de automóvel para validar as suas vulnerabilidades.

Desta forma, o propósito desta dissertação é definir uma metodologia de ataque a um automóvel, ou de *hacking*, que possa, de uma maneira célere, encontrar vulnerabilidades numa determinada viatura, usando como superfície de ataque o Wi-Fi, Bluetooth ou ODB-II, de forma a injectar ou alterar os dados dentro da rede CANbus.

Conseguindo manipular a rede CANbus, o próximo passo é executar movimentos laterais de modo a que sejam comprometidas as zonas mais críticas da viatura como, por exemplo: o motor, a direcção e os travões.

1.3 Estrutura da dissertação

Capítulo 2: Contextualização, descrição do funcionamento de um automóvel moderno e dos seus principais sistemas, considerados no âmbito desta dissertação. Definição de cibersegurança automóvel e os seus componentes, definição de vectores de ataque, bem como apresentação de vários casos de *hacking* automóvel.

Capítulo 3: Apresentação do modelo de *hacking* automóvel, metodologia e exploração dos vetores de ataque do modelo.

Capítulo 4: Aplicação do modelo de *hacking* automóvel a vários modelos de viaturas e apresentação dos resultados

Capítulo 5: Conclusão dos resultados obtidos, nomeadamente vulnerabilidades encontradas e impacto para a utilização dos veículos. Sugestão do trabalho futuro do *hacking* automóvel.

Capítulo 2

Contextualização

2.1 Funcionamento de um automóvel moderno

Tal como foi referido anteriormente, num automóvel moderno em que existem centenas de sistemas que trocam informação entre si, em média cada viatura actual tem mais de 80 processadores. Desta forma, a indústria automóvel sentiu a necessidade de desenvolver protocolos de comunicação, entre esses vários componentes, assegurando que acções críticas, como a travagem e a mudança de direcção, estavam protegidas por protocolos rápidos, seguros e resilientes.

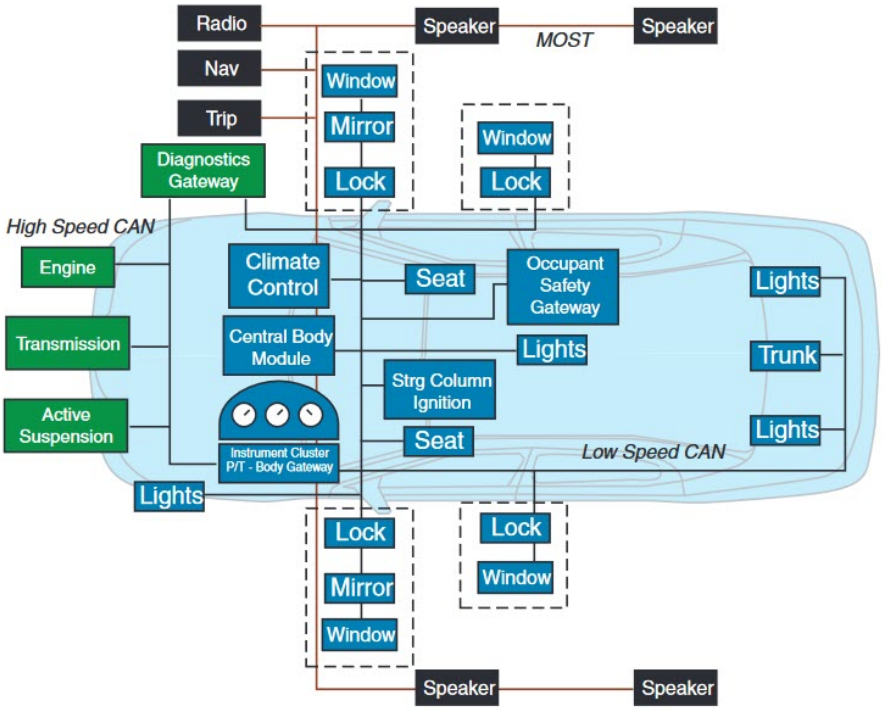

Figura 2.1: Representação da rede de comunicações de um automóvel. Figura retirada de Cook [4]

Como podemos ver na figura 2.1, todos os componentes de uma viatura estão interligados e existe uma segmentação, onde são isolados componentes críticos como o motor e os travões, de sistemas menos críticos onde se incluem o rádio, os vidros eléctricos e o ar condicionado. É interessante observar que, embora exista esta separação, existe sempre um ponto de contacto, e é nesse ponto que poderá ser a ponte para um ataque de movimento lateral[1] onde se explora um componente menos importante, como o limpa para-brisa para ganhar acesso ao motor.

Embora cada fabricante decida quais os componentes e protocolos de cada viatura e, muitas vezes, com variações no mesmo modelo, e seja muito relutante em publicar a maneira como é executada a comunicação entre cada componente, existe um *standard* de comunicação que é o CANbus. Este protocolo, pode ser analisado através de um conector, existente em todas as viaturas, denominado de ODB-II, que será descrito neste capítulo.

2.1.1 CANbus

O protocolo CANbus foi desenvolvido em 1986 pela *Robert Bosch* GmbH[2], mas só quatro anos mais tarde é que foi comercializada a primeira viatura com esta tecnologia, tratou-se do *Mercedes* Classe S[3] modelo w140 [5].

Este é o protocolo standard de comunicação presente em todos os veículos atuais, embora possa existir em veículos mais antigos que usem o CANbus. Em 2001, surgiu, na Europa, a EOBD[4], obrigatória em todos os veículos, no seguimento da directiva 98/69/CE do Parlamento Europeu [6] e do Conselho de 13 de Outubro de 1998. Esta directiva, surgiu como base às medidas a tomar contra a poluição do ar pelas emissões provenientes dos veículos a motor, ou seja, a partir desta data é possível encontrar facilmente o CANbus e identificar algumas mensagens do sistema através da ligação ODB-II.

Para transmitir a informação, o CANbus utiliza uma camada física de dois fios. Num dos fios é transmitido o estado CAN high (CANH) e em outro o estado CAN low (CANL). Habitualmente, cada fio é percorrido com uma voltagem de 2.5v. Quando existe informação a circular, o CANH passa para 3.75v e o CANL para 1.25v, gerando, assim, um diferencial de 2.5v, como podemos visualizar na figura 2.2.

[1] Técnica com o objectivo de obter acesso a um componente com privilégios mais reduzidos e escalar os privilégios para administração.

[2] Robert Bosch GmbH é uma empresa multinacional alemã de engenharia e electrónica, com sede em *Gerlingen*, perto de Estugarda, na Alemanha.

[3] Topo de gama da *Mercedes* e o modelo mais evoluído tecnologicamente

[4] European On-Board Diagnostics

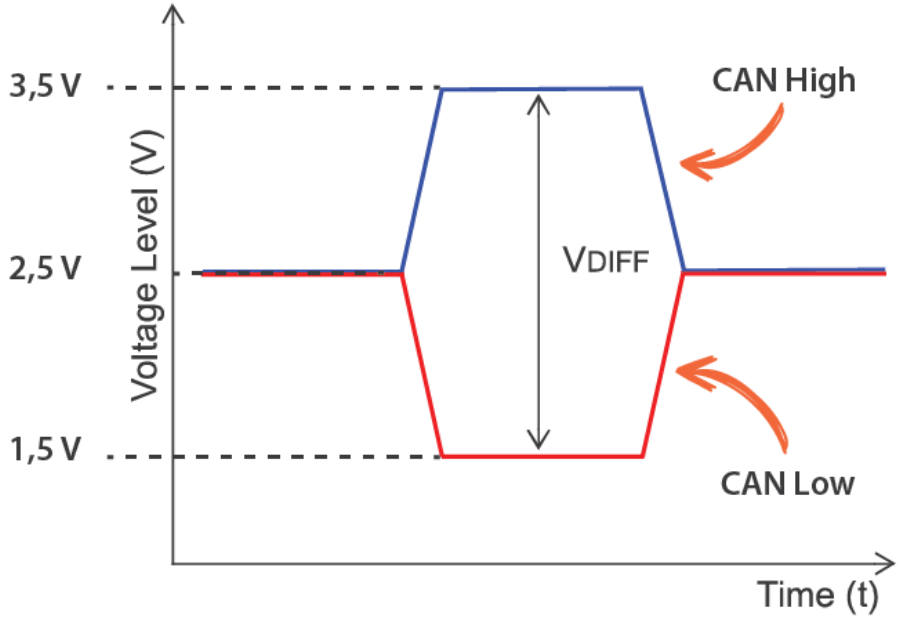

Figura 2.2: Representação esquemática da comunicação CANbus. Figura retirada de Solutions [7]

O CANbus é um protocolo extremamente resiliente a picos indutivos, campos eléctricos ou outros ruídos, reduzindo a mínimos a possibilidade de erro de transmissão, assegurando, assim, a qualidade da comunicação, factor muito importante num automóvel, dado que existem centenas de peças, que com o seu funcionamento, produzem interferências eléctricas.

Uma das particularidades deste protocolo é que qualquer dispositivo ligado à rede pode ver todas as comunicações. Não existe qualquer tipo de cifra ou de validação dos remetentes das mensagens, o que torna extremamente vulnerável este sistema [8].

Para transmissão dos pacotes de CANbus, são suportadas as seguintes velocidades de transmissão:

- 125 kbps
- 250 kbps
- 500 kbps
- 1000 kbps

As velocidades acima descritas, podem variar com o comprimento dos fios que compõem o CANbus. Podem ser usados até 250 metros de cabo, obtendo 250 kbps. O tamanho máximo que o CANbus pode ter é de cerca de 1km, com uma velocidade de 10 kbps. Para obter 1000 kbps, os cabos não podem exceder 40 metros [9].

Existem quatro tipos de *frames* no CANbus:

- *Data frame*, contém os dados do nó para a transmissão
- *Remote frame*, pede a transmissão de um identificador específico
- *Error frame*, deteta um erro transmitido por qualquer dos nós
- *Overload frame*, injecta uma pausa entre a *data frame* e o *remote frame*.

No âmbito deste trabalho, iremos-nos debruçar sobre a *data frame*, dado que é aqui que conseguimos ler e alterar os dados enviados aos vários controladores alimentados pelo CANbus.

Existem duas normas de transmissão de pacotes no CANbus, o *Standard* e o *Extended* [10], denominadas de CAN 2.0A e CAN 2.0B. São muito semelhantes, diferindo só na capacidade para armazenar uma variável denominada de ID, que será descrito a seguir.

CAN 2.0A - Pacotes *Standard*

A representação de um pacote *standard* do CANbus é a seguinte:

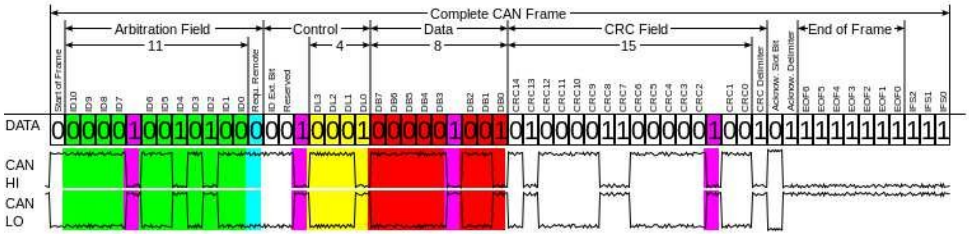

Figura 2.3: Representação esquemática da *data frame* do CANbus. Figura retirada de Fassak et al. [8]

Podemos destacar quatro elementos como sendo os mais importantes:

- *Arbitration ID*, é o identificador do ID do dispositivo que está a tentar comunicar. Se, por ventura, existir uma comunicação ao mesmo tempo que tiver o ID mais baixo, ganha prioridade no CANbus [11].
- *Identifier extension*, para mensagens CANbus *standard* está sempre com o valor 0.
- *Data length code*, define o tamanho dos dados e varia entre 0 e 8 bytes.
- *Data*, são os dados enviados pelo dispositivo, o tamanho máximo são 8 bytes.

Neste formato, temos identificadores com 11 bits e uma limitação de 2048 mensagens.

CAN 2.0B - Pacotes Extended

A grande diferença do CAN 2.0A para o CAN 2.0B é o tamanho do identificador que passa para 29 bits [12], podendo ter a circular 537 milhões de mensagens devido ao aumento do tamanho do identificador. As mensagens podem levar mais tempo a circular, sendo esta uma desvantagem do CAN 2.0B.

2.1.2 ECU

A ECU[5], ou centralina, em português, é sistema embutido que controla o funcionamento do veículo. Pode dizer-se que são o coração do automóvel, existindo vários tipos de ECU numa viatura, nomeadamente:

- ECM, módulo de controlo do motor.
- PCM. módulo que controla a articulação entre a transmissão e o motor.
- TMC, módulo de controlo da transmissão.
- BCM, módulo de controlo dos vidros e ar condicionado, entre outros.
- SCM, módulo de controlo da suspensão.
- EBCM, módulo de controlo dos travões.

O ECM recebe os valores de vários sensores e determina o que é necessário para que o motor responda a várias solicitações como, por exemplo: o pedido de aceleração ou um aumento de temperatura. Sendo a evolução natural dos carburadores, o ECM desempenha o seu trabalho em tempo real. Esta centralina é, normalmente, modificada por quem procura mais potência para as suas viaturas ou consumos mais reduzidos, dado que os parâmetros de injecção de combustível, entrada de ar, e pressões de turbo, entre outros, podem ser afinados através do *upload* de uma nova programação para a centralina.

Os elementos principais que compõem um ECM são: Um microcontrolador, memória[6], entradas de sinal, ligações com o CAN, e *software* embutido.

2.1.3 OBD-II

Um dos grandes problemas no início da década de 90 era que os diagnósticos dos automóveis eram um processo caro e que só podia ser feito nos concessionários da marca. Desta forma, surgiu

[5] Engine Control Unit
[6] Pode ser do tipo SRAM, EEPROM ou *Flash*

a necessidade de um *standard* para a detecção de erros e possíveis problemas que fosse acessível e barato. Assim, surgiu o OBD-II, uma evolução do OBD-I[7], que hoje em dia está disponível em todas as viaturas modernas.

O sistema OBD-II é acedido através de um DLC - *Data Link conector*, que é um conector fêmea de 16 pinos J1962, que permite acesso aos dados do veículo. A localização deste conector, varia entre fabricantes e, muitas vezes, entre modelos da mesma marca.

No entanto, o lugar mais habitual para o encontrar é por baixo do painel de instrumentos do lado do condutor. De acordo com as especificações, o conector OBD-II tem de estar até 60cm de distância do volante.

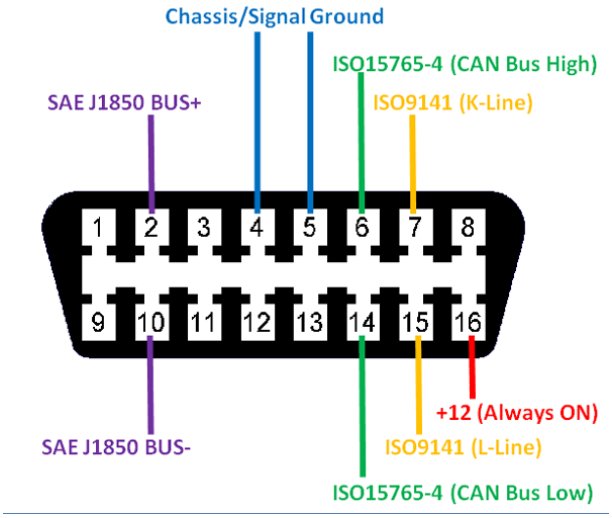

Figura 2.4: Representação esquemática de um conetor OBD-II. Figura retirada de Components101 [13]

Como podemos ver, os pinos 6 e 14 são, respectivamente, CANH e CANL, ou seja, através do OBD-II conseguimos ler o tráfego da rede CANbus e, consequentemente, comprometer a segurança da viatura. Este é um dado muito importante, dado a única segurança que existe neste conector, sendo mesmo a limitação física de acesso.

A viatura deverá estar aberta, para que seja ligado um dispositivo ao OBD-II.

A distribuição de funções por cada um dos 16 pinos é representada na tabela 2.1 [14]:

[7] Inicialmente, desenhado para controlar a emissão de gases, só suporta viaturas anteriores a 1996.

Tabela 2.1: Atribuição dos pinos do conetor OBD-II

Pino	Atribuição
1	Definido pelo fabricante automóvel
2	Condutor de dados (+) – SAE J1850
3	Definido pelo fabricante automóvel
4	Massa do chassis
5	Massa do sinal
6	Condutor de dados CAN, linha “alta” – ISO 15765-4
7	Linha K – ISO 9141-2/ISO 14230-4
8	Definido pelo fabricante automóvel
9	Definido pelo fabricante automóvel
10	Condutor de dados (-) – SAE J1850
11	Definido pelo fabricante automóvel
12	Definido pelo fabricante automóvel
13	Definido pelo fabricante automóvel
14	Condutor de dados CAN, linha “baixa” – ISO 15765-4
15	Linha L – ISO 9141-2/ISO 14230-4
16	Tensão da bateria

Em princípio, existem dois tipos de código no OBD-II [15], a saber:

- *Diagnostic Trouble Code* (DTC), código relativo a algum problema no sistema. Aqui cada código pode ser único ou definido pelo fabricante.
- *Parameter ID* (PID), código usado para extrair dados da ECU, tais como temperatura do motor ou velocidade de rotação.

2.2 Cibersegurança Automóvel

Existe uma verdade que é o pilar da cibersegurança que é: não existem sistemas totalmente seguros, existem é sistemas mais resilientes que outros aos ciberataques, e essa resiliência está associada ao risco de um determinado sistema ser ou não comprometido. Do lado do atacante, tudo se resume ao tempo e recursos que dispõe para o ataque, sendo o nosso objecto de estudo o automóvel e os possíveis ciberataques que possa sofrer. Deduzimos, então, que não existem carros totalmente seguros e que todos podem sofrer um ciberataque.

Hoje em dia, assistimos a um aumento exponencial de novos sistemas ligados a uma automóvel, sejam internos ou externos e, associado a este crescimento, está um aumento de vulnerabilidades que podem ser usadas para fins maliciosos contra um automóvel e os seus passageiros. Estas vulnerabilidades podem ir desde a quebra de privacidade dos dados pessoais do condutor, accionamento do sistema de travagem da viatura em andamento, ou até mesmo a sua anulação quando solicitada.

A complexidade dos sistemas de um automóvel é enorme, existem mais linhas de código numa viatura atual do que um caça f35 de 2013, ou mesmo do que a última versão do *Google Chrome*. Perante esta imensidão de código fonte é fácil deduzir que existirão muita vulnerabilidades por descobrir e explorar.

Esta complexidade pode ter, também, os seus efeitos no lançamento de novos modelos, como aconteceu com o novo ID3[8] onde várias falhas no software [16] obrigaram quase 20.000 viaturas a uma actualização manual com todos os problemas de logística e imagem de marca que isso acarreta.

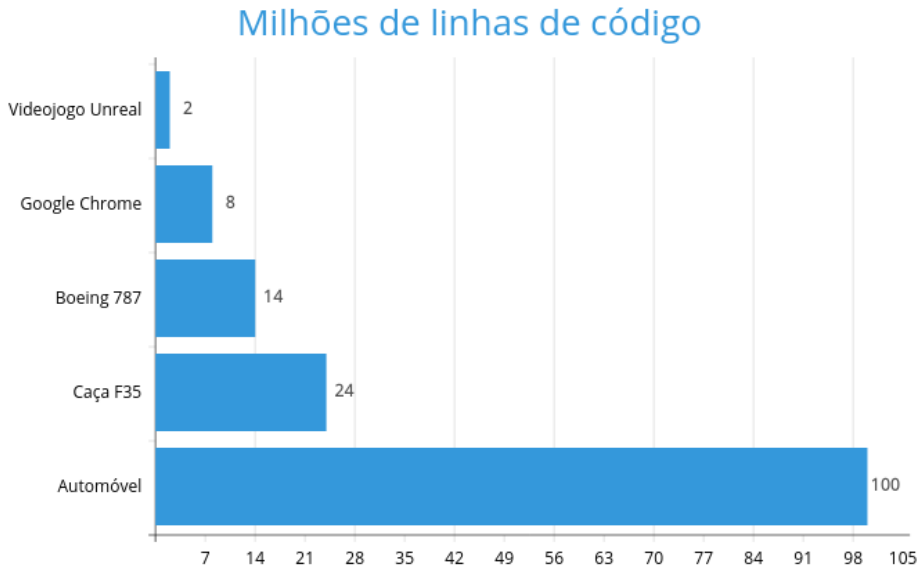

Figura 2.5: Comparação do total de linhas de código entre vários sistemas. Figura adaptada de Lévy-Bencheton [17]

Segundo a *Kaspersky* [18], cibersegurança é a prática que protege computadores, servidores, dispositivos móveis, sistemas electrónicos, redes e dados contra ataques maliciosos.

A cibersegurança automóvel incide sobre todos os sistemas que servem de suporte ao automóvel moderno, desde a simples ligação USB às mais complexas actualizações OTA[9].

Vários ataques com sucesso ao longo dos últimos anos, têm demonstrado que é urgente a implementação de medidas de segurança mais restritivas, bem como a abolição da utilização de protocolos usados no passado e, cujo desenho os incapacitam de serem dotados de medidas de segurança tão simples como a cifra de comunicação entre os vários componentes.

[8] Novo modelo totalmente eléctrico do grupo VW.
[9] Método de distribuição de actualizações de aplicações ou *firmware* por redes sem fio.

2.2.1 Vulnerabilidades

Uma vulnerabilidade é uma falha que permite que um atacante consiga comprometer um sistema de forma a fazer com que ele tenha um comportamento não previsto por quem o desenvolveu. Tendo em conta o crescimento exponencial dos sistemas, bem como a sua complexidade, é expectável que exista, também, um aumento das vulnerabilidades.

A resiliência de um sistema a um ciberataque é inversamente proporcional ao número e qualidade das vulnerabilidades dos vários componentes desse sistema.

***Common Vulnerabilities and Exposures* (CVE)**

Até ao final da década de 90, não existia um *standard* de identificação de vulnerabilidades, cada fabricante tinha as suas listas e era impossível saber se existiam vulnerabilidades duplicadas ou não. Em 1999, *David E. Mann* e *Steven M. Christey* da *Mitre*[10], publicaram um artigo intitulado "*Towards a Common Enumeration of Vulnerabilities*"[11]. Este artigo propunha a unificação e normalização do processo de publicação de uma vulnerabilidade, foi aqui que surgiu a CVE[12] que toda a indústria adotou como norma. Desta forma, em Setembro de 1999 surgiu a primeira lista de CVE com 321 vulnerabilidades.

O CVE, tem como objectivo primário a identificação de vulnerabilidades [19], ou seja, a cada vulnerabilidade é atribuído um identificador único no formato **CVE-YYYY-NNNN**, cujas atribuições são as seguintes:

- CVE, termo fixo e igual para todas as vulnerabilidades.
- YYYY, ano de atribuição do CVE.
- Numero de série da CVE que normalmente tem 4 dígitos, mas que pode ir até 5 ou mais em caso de necessidade.

Como exemplo temos a CVE-2018-18071, que afecta o fabricante Mercedes. Desconstruindo o formato de CVE, sabemos que foi atribuída em 2018 e que o número de série é 18071.

Quando um investigador encontra uma falha de segurança num determinado sistema, o mesmo submete essa possível vulnerabilidade no formulário disponível em https://cveform.mitre.org/, com todos os detalhes da vulnerabilidade, de modo a que possa ser replicada e entendida de uma forma célere. Se a vulnerabilidade for validada é, então, atribuído um CVE.

Sempre que um investigador, ou mesmo um agente malicioso, quer atacar/testar um determinado sistema, um dos primeiros passos é pesquisar se existe alguma CVE associada a esse produto ou

[10] *The Mitre Corporation*, é uma organização americana sem fins lucrativos.
[11] Artigo disponível em `https://cve.mitre.org/docs/docs-2000/cerias.html`
[12] Common Vulnerabilities and Exposures

fabricante, até porque podem existir vulnerabilidades antigas que ainda existem nos sistemas. Exemplo disso é a CVE-2017-0144, que, embora tenha sido disponibilizada uma correcção do fabricante em 2017, no ano de 2020 ainda existem milhares de máquinas sem o patch instalado, ou seja, a persistência das vulnerabilidades ao longo do tempo é potenciada pela falta de políticas de actualização dos sistemas por parte dos administradores de rede.

Common Vulnerability Scoring System (CVSS)

Foi o National Infrastructure Advisory Council (NIAC)[13] que em 2005 lançou o CVSS, sendo este um método de classificação do grau de risco de uma vulnerabilidade através das suas características. À data, o CVSS já vai na versão 3.0 [20]. Nesta classificação, o risco é numérico e pode variar entre 0 e 10. O 10 é o valor mais crítico.

O cálculo do CVSS é feito com base em critérios de avaliação divididos em três grupos, a saber [21]:

- Base
- Temporal
- Ambiental

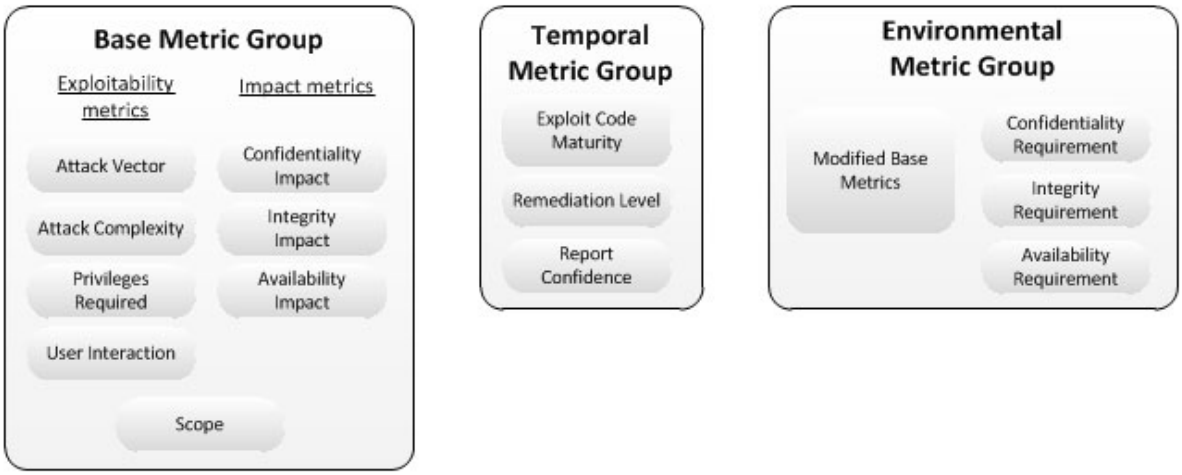

Figura 2.6: Grupos de métricas da CVSS 3.0. Figura retirada de CFIRST [20]

O grupo base abarca as características que não variam com a variável tempo. Este grupo é a métrica mais importante das três, sendo a que tem maior impacto para a pontuação final da CVSS. Por sua vez, esta métrica divide-se em outras duas categorias, sendo elas a exploração e o impacto.

[13]Organismo dos Estados Unidos que tem como principal área de actuação a segurança. O seu site pode ser visitado em https://www.cisa.gov/niac

A exploração avalia a facilidade com que se pode explorar a vulnerabilidade, representando quatro métricas:

- Superfície de ataque, por onde é executado o ataque.
- Complexidade do ataque, tempo e esforço necessário para explorar a vulnerabilidade.
- Privilégios necessários para executar o ataque.
- Interacção com o utilizador e nível de interacção com o utilizador, necessário para a exploração.

Por sua vez, o grupo impacto mede os dados causados no sistema atacado através da permeabilidade da integridade, disponibilidade e acessibilidade do sistema.

No grupo temporal, temos a avaliação em como o risco da vulnerabilidade pode variar com o tempo. Tal como a base, esta métrica também é dividida nas seguintes métricas:

- Maturidade do código de exploração, mede qual o estado do código fonte da exploração da vulnerabilidade.
- Nível de remediação, mede a solução para a vulnerabilidade, nomeadamente se é temporária, definitiva e oficial entre outras.
- Grau de confiança, mede a credibilidade do emissor da vulnerabilidade bem como nas suas especificações técnicas.

Por ultimo temos o grupo ambiental, é relativo à importância do sistema, considerando os requisitos de confidencialidade, integridade e disponibilidade bem como dos mecanismos de segurança que existem no ecossistema onde existe a vulnerabilidade, as métricas que contribuem para este grupo são as seguintes [22]:

- Requisitos de segurança, ajuda à caracterização da CVE como base no cenário em que está inserida.
- Modificação da métrica base, como o ambiente em que o atacante se movimenta pode ser influenciando pelas medidas de segurança da organização,ajuda ao ajuste da métrica de base.

O cálculo do CVSS, é complexo, desta forma poder ser usada uma calculadora *online* disponibilizada pelo *National Vulnerability Database*[14] e disponível em https://nvd.nist.gov/, aqui pode ser inseridos todos as métricas que constituem os grupos da CVSS e desta forma obter o valor de risco de determinada vulnerabilidade para a nossa organização.

[14]Maior base de dado do mundo de vulnerabilidades conhecidas, é gerida pelo governo dos estados unidos

Vulnerabilidades Conhecidas

As vulnerabilidades conhecidas já passaram por todo o processo de descoberta e resolução por parte do fabricante, ou seja estão bem identificadas e existem correcções disponíveis , a estas vulnerabilidades normalmente é atribuído um CVE no entanto pode não acontecer em todos os casos. Existem bases de dados públicas e privadas onde é possível pesquisar vulnerabilidades conhecidas.

Como exemplos de bases de dados públicas de vulnerabilidades conhecidas, temos a CVE *Details* [15] e a *National Vulnerability Database* [16]

Vulnerabilidades Desconhecidas

Uma vulnerabilidade é desconhecida quando o vendedor ou fabricante tem desconhecimento dessa falha, logo não existe nenhuma correcção disponibilizada, o termo desconhecida não quer dizer necessariamente que ninguém a conheça e que não a explore, exemplo disso foi a CVE-2017-0144, conhecida como EternalBlue[17], que foi explorada durante anos antes que o fabricante lançasse uma correcção em 2017, e quando tornada publica originou os ataques *wannacry*[18]

Vulnerabilidades *Zero day*

Uma vulnerabilidade *zero day* é aquela que é desconhecida por todos sejam fabricantes, investigadores ou o publico em geral, como é uma vulnerabilidade desconhecida não existe a capacidade de defesa dos sistemas [23], sendo das mais perigosas é também das mais procuradas e consequentemente das mais bem pagas. Por exemplo uma vulnerabilidade *0 day* que permita a execução remota de código num *iphone* pode chegar ao valor de um milhão de dólares [24].

Vulnerabilidades na indústria automóvel

Na indústria automóvel ainda não existem muitas as vulnerabilidades públicas com CVE atribuído, no então é possível encontrar algumas, por exemplo se pesquisarmos o fabricante BMW na plataforma CVE *Details*, temos sete vulnerabilidades identificadas todas no ano de 2018 e com um *score* entre 5,7 e 10.

Podemos ainda encontrar uma vulnerabilidade do fabricante Mercedes, a CVE-2018-18071, que afecta a aplicação *Daimler Mercedes* Me App 2.11.0-846 no sistema operativo IOS, que obteve a classificação de 5.0 segundo o CVSS.

[15] Disponível em `https://www.cvedetails.com/`
[16] Disponível em `https://nvd.nist.gov/`
[17] *Exploit* que permite a execução de código remoto em sistemas operativos *windows*
[18] *Ransomware* que ataca os sistema operativo *windows*

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2018-9322	693		Bypass	2018-05-31	2018-06-29	7.2	None	Local	Low	Not required	Complete	Complete	Complete
The Head Unit HU_NBT (aka Infotainment) component on BMW i Series, BMW X Series, BMW 3 Series, BMW 5 Series, and BMW 7 Series vehicles produced in 2012 through 2018 allows local attacks involving the USB or OBD-II interface. An attacker can bypass the code-signing protection mechanism for firmware updates, and consequently obtain a root shell.														
2	CVE-2018-9320	693			2018-05-31	2018-06-29	7.2	None	Local	Low	Not required	Complete	Complete	Complete
The Head Unit HU_NBT (aka Infotainment) component on BMW i Series, BMW X Series, BMW 3 Series, BMW 5 Series, and BMW 7 Series vehicles produced in 2012 through 2018 allows a local attack when a USB device is plugged in.														
3	CVE-2018-9318	693			2018-05-31	2018-06-29	10.0	None	Remote	Low	Not required	Complete	Complete	Complete
The Telematics Control Unit (aka Telematic Communication Box or TCB), when present on BMW vehicles produced in 2012 through 2018, allows a remote attack via a cellular network.														
4	CVE-2018-9314	693			2018-05-31	2018-06-29	7.2	None	Local	Low	Not required	Complete	Complete	Complete
The Head Unit HU_NBT (aka Infotainment) component on BMW i Series, BMW X Series, BMW 3 Series, BMW 5 Series, and BMW 7 Series vehicles produced in 2012 through 2018 allows an attack by an attacker who has direct physical access.														
5	CVE-2018-9313	693			2018-05-31	2018-06-29	5.7	None	Local Network	Medium	Not required	None	None	Complete
The Head Unit HU_NBT (aka Infotainment) component on BMW i Series, BMW X Series, BMW 3 Series, BMW 5 Series, and BMW 7 Series vehicles produced in 2012 through 2018 allows a remote attack via Bluetooth when in pairing mode, leading to a Head Unit reboot.														
6	CVE-2018-9312	693			2018-05-31	2018-06-29	7.2	None	Local	Low	Not required	Complete	Complete	Complete
The Head Unit HU_NBT (aka Infotainment) component on BMW i Series, BMW X Series, BMW 3 Series, BMW 5 Series, and BMW 7 Series vehicles produced in 2012 through 2018 allows a local attack when a USB device is plugged in.														
7	CVE-2018-9311	693			2018-05-31	2018-06-29	10.0	None	Remote	Low	Not required	Complete	Complete	Complete
The Telematics Control Unit (aka Telematic Communication Box or TCB), when present on BMW vehicles produced in 2012 through 2018, allows a remote attack via a cellular network.														

Figura 2.7: Vulnerabilidades públicas da BMW. Figura retirada de DETAILS [25]

No entanto, tal como já foi referido ao longo deste documento, um veículo é composto pela integração de vários sistemas, por vezes de origens distintas, e cada um desses sistemas pode contribuir para que existam vulnerabilidades num automóvel, por exemplo a simples disponibilização de uma porta USB pode trazer várias vulnerabilidades para cima da mesa, um exemplo disso é a CVE-2018-18203 que afecta os sistema da Subaru através da exploração de uma vulnerabilidade conhecida no USB.

Existem plataformas que permitem aos *hackers* submeter vulnerabilidades encontradas nos sistemas e receber uma compensação monetária em troca, essas plataformas funcionam como intermediários, e garantem a protecção de quem descobriu a vulnerabilidade ou exploração do sistemas, as marcas podem promover estes programas de recompensa de uma forma privada, endereçando o convite de participação a determinados investigadores seleccionados. Alguns dos fabricantes já aderiram a estes programas de recompensa, entre eles temos a *General Motors* que está na plataforma *hackerOne*[19] em `https://hackerone.com/gm?type=team`, a *Fiat Chrysler* na plataforma *bugcrowd*[20], https://bugcrowd.com/fca, que paga entre 150 e 7500 dólares por vulnerabilidade encontrada e a Tesla que também na plataforma *bug crowd*, `https://bugcrowd.com/tesla`, que recompensa os investigadores com prémios entre 100 e 15000 dólares por descoberta.

2.2.2 Ciclo do *Hacking*

Por definição um *hacker* é um indivíduo que se dedica, com intensidade incomum, a conhecer e modificar os aspectos mais internos de dispositivos, programas e redes de computadores. Graças a esses conhecimentos, um *hacker* frequentemente consegue obter soluções e efeitos extraordinários, que extrapolam os limites do funcionamento ”normal”dos sistemas como previstos pelos seus criadores; incluindo, por exemplo, contornar as barreiras que supostamente deveriam impedir o controle de certos sistemas e acesso a certos dados.

[19] Criada em 2012 por investigadores da *Facebook* e *Google*, é a maior plataforma de *bug bounty*

[20] Fundada em 2011, foi das primeiras organizações a implementar o *bug bounty*, em 2018 foi avaliada em 26 milhões de dólares

Quando ouvimos nos media noticias sobre *hackers*, existem uma conotação negativa, ou seja, parte-se do princípio que todos os *hackers* são criminosos, ora isto é errado. A palavra *hacker* não significa criminoso, ter as competências e o conhecimento não significa por indução que esse individuo o vai usar para o "mal". Um *hacker* é acima de tudo um conhecedor, um estudioso habituado a usar a sua inteligência e raciocínio para resolver problemas complicados, contribuindo assim para o melhoramento da segurança dos sistemas.

Actualmente o Hacking tem um papel crucial na determinação de níveis de segurança de sistemas e organizações, contribuindo inclusivamente para o salvamento de vidas humanas,o ciclo do processo de hacking está bem definido e é composto por cinco fases, sendo que a cada uma delas estão associadas técnicas e ferramentas especificas. O sucesso de cada fase dependente da qualidade da fase precedente.

Este ciclo pode ser representado da seguinte forma:

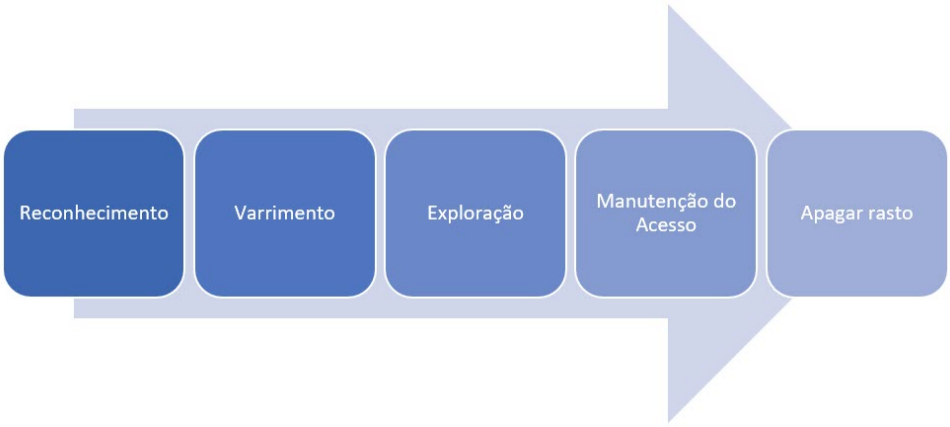

Figura 2.8: Fases de um processo de Hacking

A fase inicial é denominada de reconhecimento e é aqui que é angariada toda a informação sobre o sistemas que vamos testar, podem ser usados vários processos locais ou remotos, a maior parte das vezes apoiados em OSINT[21] , tão distintos como o mapeamento das redes externas e internas com ferramentas tais como o theharvester[22], Shodan[23] ou mesmo google dorks[24] para recolher informação sobre o sistemas, a nível local, uma das ferramentas passivas que pode ser usada é o Netdiscover[25].

A fase do reconhecimento deve ser maioritariamente passiva, de forma a que não seja despoletado nenhum tipo de alarme do lado do sistema que está a ser testado, uma analogia que pode ser feita é o olhar para uma porta e pesquisar sobre o fabricante e aquele modelo especifico de forma a que quando se for tentar abrir a porta já termos o máximo de informação possível sobre ela, aumentado assim a possibilidade de sucesso e reduzindo o numero de tentativas falhadas.

[21] Processo de recolha de informação em fontes abertas
[22] Software que recolhe informação sobre determinada organização com base em fontes abertas
[23] Motor de pesquisa, que permite localizar computadores e serviços ligados à internet, a plataforma está disponivel em www.shodan.io
[24] Operadores que permitem afinar as pesquisas em www.google.pt
[25] Software usado para a descoberta de endereços IP activos que pode ser descarregado em https://github.com/netdiscover-scanner/netdiscover

O varrimento tem com objectivo identificar serviços e possíveis portas de entrada no sistema, aqui podemos recorrer a ferramentas como o nmap[26]. Nesta fase do hacking, são usadas técnicas mais agressivas que podem despoletar alarmes do lado do sistema que está a ser testado, aqui vamos descobrir máquinas ou serviços que estão a funcionar na rede, bem como possíveis vulnerabilidades e sistemas operativos.

A exploração é a etapa onde com base no reconhecimento anterior e vulnerabilidades detetadas, é obtido o acesso ao sistema, depois do acesso poderá existir a necessidade do atacante elevar os seus privilégios para administrador desse sistema, de forma a conseguir ler, modificar ou apagar informação

A manutenção do acesso assegura a persistência de acesso ao sistema por parte do atacante, podendo assim continuar a interferir com o funcionamento do mesmo, para isso deverá ser encontrada uma forma de manter o acesso mesmo que o sistema seja reinicializado, por exemplo associando uma backdoor[27] a um processo legitimo.

Finalmente quando o objetivo do ataque está concretizado, ou o atacante sente que pode ser descoberto dentro da rede, temos a fase de apagar rasto, aqui são eliminados logs de servidores, ficheiros temporários, histórico de linhas de comando, emails ou qualquer outro tipo de informação que possa ligar o atacante à exploração do sistema.

2.2.3 Superfície de Ataque

Por definição a superfície de ataque é onde o *hacker* consegue explorar vectores de ataque, como analogia podemos imaginar uma casa, com portas, janelas, chaminés, etc, cada um deste pontos pode servir para o atacante entrar dentro da casa usando uma técnica distinta. Por exemplo pode partir um vidro de uma janela ou simplesmente tentar forçar a fechadura da porta de entrada.

Podemos definir superfície de ataque como sendo o conjunto de maneiras que um atacante pode comprometer determinado sistema, quanto maior for a superfície de ataque maior é o risco de o sistema sofrer um ataque com sucesso

Não existem sistemas sem superfícies de ataque, nos automóveis cada nova funcionalidade ou tecnologia que é adicionada, pode fazer com que aumentem as superfícies de ataque.

Nos automóveis existem dezenas de ECU[28] que controlam varias funções criticas tais como a gestão do motor, a travagem e mudança de direção,ou menos criticas como o ajuste de temperatura do ar condicionado e a regulação do rádio. As várias ECU de uma viatura correm milhares de milhões de linhas código que podem conter vulnerabilidades, depois todas essas ECU comunicam entre si com

[26] Software que executa o varrimento a vários portos e endereços IP numa determinada rede.
[27] Método que o atacante usa para aceder remotamente ao sistema comprometido
[28] Componente electrónico que controla os diversos sistemas de um automóvel

recurso a protocolos como o CANbus[29] que pode também ter uma serie de vulnerabilidades, ou seja os sistemas que fazem com que o automóvel desempenhe as suas várias funções são os mesmos que contribuem para o aumento da superfície de ataque e consequentemente para um possível sucesso do *hacker*.

Num automóvel existem diversas superfícies de ataque que podem ser usadas por alguém mal intencionado para comprometer o funcionamento da viatura ou exfiltrar dados dos seus ocupantes, entre as várias superfícies de ataque existentes, as mais importantes são as seguintes [26]:

Portas USB

Utilizadas para fornecer energia a outros dispositivos, como por exemplo telemóveis, para carregar ficheiros para o sistema de multimédia tais como músicas em formato MP3 e para actualizações do sistema.

Ligações Bluetooth

São usadas essencialmente para o emparelhamento com telemóveis para permitirem que o condutor faça por exemplo chamadas em alta voz, leia e responda a mensagens sem ser necessário tirar as mãos do volante ou para que o sistema de multimédia funcione como uma extensão do telemóvel podendo aceder no visor do sistema de multimédia a varias aplicações, isto acontece no Android Auto[30] e Apple Car[31].

Ligações Wi-Fi

Podem servir para disponibilizar um HOTSPOT[32] aos ocupantes do veículo, mas também o inverso ou seja o veículo usa o Wi-Fi para se ligar a um hotspot, muitas vezes o Wi-Fi está diretamente ligado ao sistema de multimédia.

Ligações GSM

Existem varias utilizações da tecnologia GSM[33] num automóvel. Nas viaturas mais recentes existe um cartão GSM, denominado de cartão SIM, embutido no hardware que permite as comunicações entre a viatura e o fabricante podendo desta formar enviar e receber vários tipos de informação e permitindo inclusivamente actualizações remotas.

[29] Rede interna de comunicação num automóvel
[30] Aplicação da Google que permite espelhar as funcionalidades de um telemóvel na viatura
[31] Sistemas semelhante ao Android Auto destinado a utilizadores da Apple
[32] Equipamento para partilha de internet por Wi-Fi
[33] Tecnologia utilizada para a comunicação de equipamentos móveis, tais como o telemóvel

Sensor do monitor de pressão dos pneus

Este sistema, denominado de TPMS[34], permite enviar informação sobre o estado dos pneus, de forma a que se existir uma perda de pressão num dos pneumáticos o condutor seja imediatamente informado. Os TPMS estão instalados dentro dos pneus e asseguram a comunicação com a ECU através de sinais rádio.

Porta de diagnostico

A porta de diagnóstico, ou ODB-II permite a leitura externa do tráfego da CANbus, bem como a injeção de frames CAN, o funcionamento desta porta foi descrito na secção funcionamento automóvel desta dissertação.

Aplicações

As aplicações de um automóvel podem estar instaladas directamente no sistema de infoentretenimento[35] ou no smartphone do condutor que por sua vez se ligam remotamente ao sistema de infoentretenimento , muitas delas ligam-se externamente a uma cloud[36] para armazenamento e troca de informação. A maioria dos carros actuais fornece aos seus clientes uma aplicação onde podem consultar todo o tipo de informações tais como consumos, distâncias percorridas ou mesmo se as portas estão ou não fechadas, por sua vez o condutor pode enviar comandos para a viatura através destas aplicações para executar tarefas como buzinar, abrir as portas, accionar o ar condicionado ou mesmo ligar o motor.

2.2.4 Vectores de ataque

Tendo em conta as superfícies de ataque acima enunciadas, podemos então identificar os vectores de ataque que serão a porta de entrada para comprometer e explorar a viatura, estes assentam em vulnerabilidades de desenho ou de codificação dos componentes de cada sistema, ou seja irá existir uma decomposição de cada vector de ataque nos seus componentes mais pequenos de forma a encontrar uma possível vulnerabilidade.

[34] Tire-Pressure Monitoring System
[35] Sistema que controla várias funções do automóvel, auxilia a condução e fornece várias opções de entretenimento
[36] Rede de servidores interligados num ecossistema único, existem clouds privadas e públicas

Exploração Wi-Fi

O Wi-Fi[37] é sem duvida o protocolo de comunicações sem fio mais conhecido, e provavelmente o mais usado hoje em dia. Quase todos os equipamentos pessoais, tais como computadores portáteis, telemóveis e mesmo relógios, usam o Wi-Fi para se ligarem à Internet, a indústria automóvel ciente desta realidade, começou a oferecer aos seus clientes a possibilidade de usarem o Wi-Fi para interagirem como as suas viaturas, para acederem à Internet ou até mesmo para que a própria viatura aceda à Internet e faça actualizações de software.

Relativamente ao Wi-Fi, as viaturas podem funcionar em dois modos: Como um ponto de acesso ou ligarem-se a um ponto de acesso.

Ao funcionar como um ponto de acesso,a viatura disponibiliza uma rede Wi-Fi a que vários dispositivos podem ser ligados, como por exemplo telemóveis. Para conseguir esta funcionalidade existem vários sistemas no veiculo dedicados a esta função tais como um router, um servidor de DNS, etc.

Por outro lado quando a viatura se liga a um ponto de acesso externo, será um dispositivo numa rede não controlada pelo veiculo e onde todas as configurações são definidas externamente, tais com o endereço IP ou o servidor de DNS.

Existem várias vulnerabilidades conhecidas nas redes Wi-Fi que permitem os ataques a estas redes, uma delas é o facto de não existir validação do SSID[38] a qual o dispositivo se está a ligar. Desta forma é possível clonar esse SSID e colocar no ar uma rede exactamente com o mesmo identificador, este ataque é denominado por evil twin, e é assente não só na premissa que os dispositivos só procuram o SSID para se ligarem, mas também que preferem a rede com mais potencia caso existam dois SSID com o mesmo identificador.

Ora partindo dos dois modos de funcionamento das viaturas nas redes Wi-Fi,é fácil entender que em ambos será possível o ataque *evil twin*, seja para levar o automóvel a ligar-se a uma rede Wi-Fi do atacante com o mesmo nome que a rede original, seja para levar os dispositivos externos a pensarem que estão ligados à rede da viatura e me vez disso estão ligados à rede do atacante.

Seguidamente temos o ataque de desautenticação, onde utilizamos uma frame de desautenticação para forçar que os dispositivos de desliguem da AP [28], também pode ser considerado um ataque de DOS dado que se for continuo a rede Wi-Fi pura e simplesmente deixa de funcionar, esta é uma vulnerabilidade de desenho do protocolo, que permite que um utilizador não autenticado emita estas frames de desautenticação.

Existe um outro ataque em redes WI-FI, denominado de probe, que permite duas coisas, a primeira é analisar por onde andou a viatura e também levar a que esta se ligue à rede do atacante de forma pas-

[37] Nomenclatura que deriva das palavras Wireless Fidelity, o Wi-Fi foi desenvolvido pelo Institute of Electrical and Eletronics Enginners

[38] Identificador da rede WI-FI

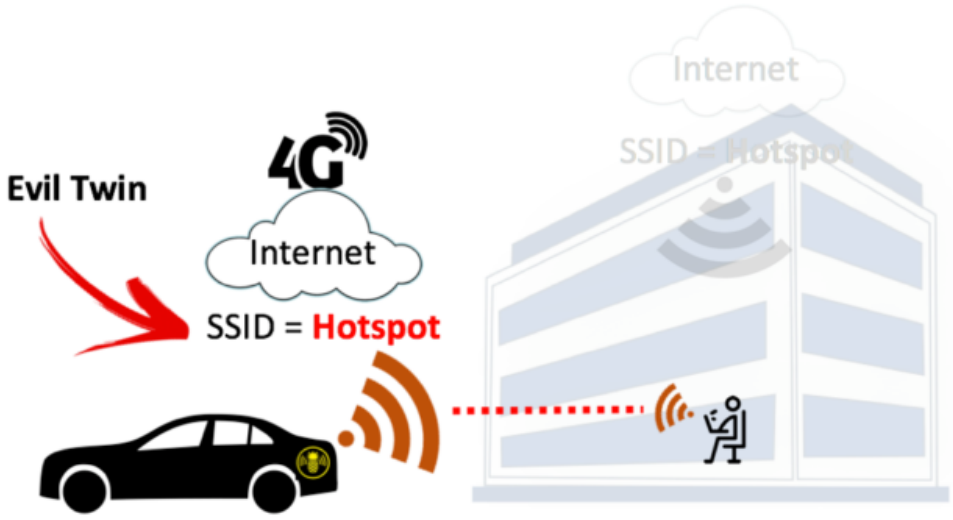

Figura 2.9: Representação do ataque Evil Twin, Figura retirada de Orsi [27]

siva, e isto deve-se ao facto de que os dispositivos que já estiveram ligados a uma rede Wi-Fi, mantêm uma lista dessas redes e emitem regularmente um probe request, ou seja perguntam se existem ao alcance deles o SSID a que já estiveram ligados. O atacante por sua vez pode analisar estes probe requests [29] e ver quais foram os SSID a que a veículo já esteve ligado, esta informação pode ser usada para duas coisas: Análise dos movimentos da viatura e sua geolocalização ou a disponibilização de uma rede Wi-Fi com o mesmo SSID que a viatura procura, estabelecendo assim uma ligação entre o veiculo e uma rede com intenções maliciosas. [30]

```
19:39:37 -70dBm 6c:ad:f8:37:59:7b "khazad-dum"
19:39:44 -90dBm fc:3f:db:de:90:3c "HOME-7F4F"
19:39:50 -90dBm 00:11:d9:47:38:b2 "george123"
19:39:50 -91dBm 00:11:d9:47:38:b2 "george123"
19:39:50 -92dBm 00:11:d9:47:38:b2 "george123"
19:39:52 -89dBm b8:3e:59:ab:d4:7f "J-lou's.homedigs"
19:39:54 -89dBm fc:3f:db:de:90:3c "HOME-7F4F"
19:40:00 -86dBm b8:3e:59:ab:d4:7f "J-lou's.homedigs"
19:40:00 -88dBm b8:3e:59:ab:d4:7f "J-lou's.homedigs"
19:40:06 -92dBm 6c:ad:f8:5f:07:44 "The Internationale"
19:40:06 -95dBm 6c:ad:f8:5f:07:44 "The Internationale"
19:40:08 -86dBm b8:3e:59:ab:d4:7f "J-lou's.homedigs"
19:40:16 -86dBm b8:3e:59:ab:d4:7f "J-lou's.homedigs"
19:40:16 -87dBm b8:3e:59:ab:d4:7f "J-lou's.homedigs"
19:40:25 -45dBm d0:e7:82:f0:d4:db "khazad-dum"
19:40:25 -48dBm d0:e7:82:f0:d4:db "khazad-dum"
19:40:43 -93dBm f4:f5:d8:28:bc:26 "NETGEAR85-5G"
19:40:47 -93dBm fc:3f:db:de:90:3c "HOME-7F4F"
19:40:48 -87dBm b8:3e:59:ab:d4:7f "J-lou's.homedigs"
19:40:49 -85dBm b8:3e:59:ab:d4:7f "J-lou's.homedigs"
19:40:49 -89dBm b8:3e:59:ab:d4:7f "J-lou's.homedigs"
```

Figura 2.10: Resultados obtidos ao analisar os probe request

Por ultimo temos o cracking da password da rede Wi-Fi por bruteforce, neste ataque é interceptado

uma determinada mensagem entre o AP e cliente e de seguida com base num dicionário de palavras, é testada cada uma dessas palavras como sendo a possível palavra passe da rede WI-FI. Este método depende da qualidade do dicionário.

Esta vulnerabilidade baseia-se na intercepção do handshake[39] do protocolo WPA/WPA2[40], após esta intercepção e recorrendo a uma vulnerabilidade na cifra WPA [31] é possível saber se determinada palavra passe é ou não a chave dessa rede.

Exploração Bluetooth

O Bluetooth é uma norma de comunicações sem fio e de baixo consumo, utiliza a frequência 2.4GHz onte tem definidos 79 canais de radio[41] espaçados de 1 MHz, esta tecnologia está divida em três classes consoante o alcance da comunicação, a saber [32]:

- Classe 1: potência máxima de 100 mW, alcance de até 100 metros;
- Classe 2: potência máxima de 2,5 mW, alcance de até 10 metros;
- Classe 3: potência máxima de 1 mW, alcance de até 1 metro.

A classe mais comum é a classe 2 ou seja um alcance até 10 metros, neste tipo de redes[42] um dos dispositivos tem o papel de Master e todos os outros têm o papel de Slave, o Master tem a função de decidir qual dos Slaves tem acesso ao canal, ou seja o Slave só pode entregar um pacote de comunicação ao Master se o Master tiver dado a sua autorização [33]. Numa piconet cada dispositivo tem um identificador único de 48bits

A pilha do protocolo Bluetooth é composta por um misto de componentes específicos do Bluetooth e outros, entre os específicos temos o Link Manager Protocol - LMP e o Logical Link Control and Adaptation Protocol - L2CAP.

Não querendo apresentar de uma foram exaustiva toda a pilha do protocolo Bluetooh, é no entanto importante referir alguns dos principais componentes que terão influencia na exploração desta superfície de ataque.

LMP, é responsável pela ligação nomeadamente aspectos como a cifra e a autenticação, controla ainda os estados da ligação na piconet.

L2CAP, valida os requisitos e qualidade da ligação, controla as ligações lógicas.

[39] Primeiras quatro mensagens entre o cliente e o AP a que esse cliente se quer ligar
[40] Modelos de cifra das redes Wi-Fi
[41] Designados de RF channels
[42] Estas redes Bluetooth são denominadas de piconet

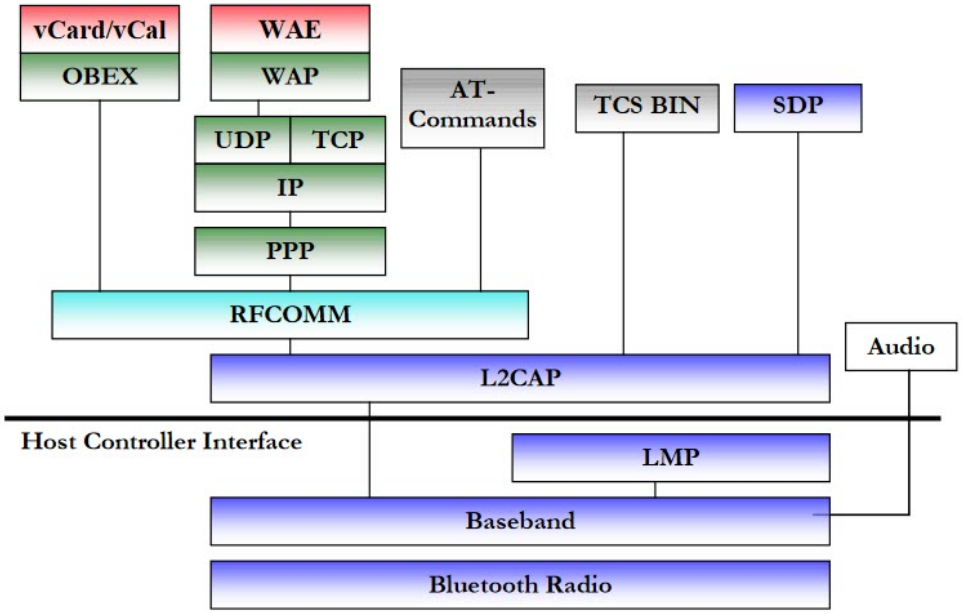

Figura 2.11: Pilha do protocolo Bluetooth. Figura retirada de Group [34]

SDP[43], é responsável por detetar serviços disponibilizados por outros dispositivos Bluetooth alcance, mantém ainda uma lista desse dispositivos ao alcance. Recorrendo à arquitectura cliente/servidor, o dispositivo que fornece o serviço corre o servidor SDP e o que vai usar o serviço o cliente SDP.

RFCOMM[44], consegue uma emulação de uma porta de serie[45], desta forma os dispositivos que já existem podem integrar este ecossistema facilmente.

OBEX, permite a troca de vcard[46] entre dispositivos, o interessante é que não é necessário emparelhamento entre dispositivos Bluetooth para a troca de um vcard.

Existem várias vulnerabilidades do Bluetooth, embora algumas possam ter sido corrigidas em novas versões, a ligação entre dois dispositivos por Bluetooth é tão forte como o que for mais vulnerável, ou seja dispositivos mais antigos podem ser excelente vectores de ataque [35], e isto é um dado muito importante tendo em conta que existem muitas viaturas com versões antigas de Bluetooth que nunca serão actualizadas.

Os ataques mais comuns a dispositivos Bluetooh são os seguintes:

- MAC Spoofing
- PIN Cracking

[43] Service Discovery Protocol
[44] Baseado na norma TS 07.10 da European Telecommunications Standards Institute
[45] RS-232, Recommend Standard – 232, porta padrão de comunicações
[46] Também conhecidos por VCF (Virtual Contact File)

- Man-in-the-Middle
- BlueJacking
- BlueSnarfing

MAC Spoofing, antes de ser estabelecido o canal seguro de comunicação, o atacante faz-se passar por um dispositivo legitimo, fazendo spoof[47] do MAC address[48], desta forma consegue o emparelhamento.

PIN Cracking, este ataque é despoletado quando está a ocorrer o emparelhamento, o atacante consegue interceptar atráves de um processo de intercepção da comunicação uma parte do emparelhamento que lhe vai permitir pela técnica de força bruta[49] conseguir chegar ao PIN.

Man-in-the-Middle, identificada pela CVE-2018-5383, esta vulnerabilidade permite que o atacante intercepte as comunicações entre os dispositivos legítimos.

BlueJacking, envio de mensagens anónimas não não solicitadas através com recurso à funcionalidade vcard[50] sobre Bluetooth, que podem levar a vitima a revelar informação confidencial, por exemplo uma password, ou a executar determinada acção [36].

BlueSnarfing, através da exploração do OBEX File Transfer Protocol[51], o atacante consegue acesso a toda a informação no dispositivo da vitima, nomeadamente imagens, lista de contactos, mensagens, etc.

Exploração Portas USB

Existem dois tipo de exploração das portas USB: Os destrutivos e que se enquadram no âmbito da sabotagem e os de tentativa de exfiltração do sistema.

Os destrutivos são baseados num dispositivo denominado de USBKILL[52], esta peça de hardware funciona de uma forma muito simples, ao ser ligada à porta USB armazena energia que depois devolve à maquina a que está ligada numa descarga de alta voltagem. O resultado é a destruição dos componentes electrónicos do sistema e a sua inevitável inutilização[53], o USBKILL pode fazer com que um automóvel deixe de responder completamente, ou em casos específicos que as portas USB sejam queimadas [37].

Do ponto de vista da exploração do sistema a porta USB pode ser usada para uma actualização de

[47] O atacante usa a identidade de um dispositivo autorizado
[48] Identificador único do dispositivo Bluetooth
[49] Consiste em tentar todas as combinações possíveis da chave até ter acesso ao sistema, pode ser um processo muito moroso
[50] Formato padrão para o envio de catões de visita eletrónicos.
[51] Protocolo de transferência de objetos em redes sem fios
[52] Pode ser adquirido em https://usbkill.com/
[53] Pode ser vista uma demonstração do USBKILL em https://usbkill.com/blogs/news/usb-kill-vs-car-are-you-at-risk

um firmware comprometido, podendo desta forma ser a porta de entrada para um acesso de privilégios de administração ao atacante, isto acontece porque muitas vezes não existe validação da assinatura do firmware que é carregado nestas portas, através da porta USB pode ser também possível um ataque com ransomware[54] onde os sistemas do veículo são cifrados pelo atacante, só após um pedido de resgate é que os sistemas são decifrados. [39]

Alguns sistemas permitem ainda a ligação de dispositivos usb-to-ethernet[55], desta forma é possível explorar o sistema através do endereço IP .

Exploração da porta de diagnostico

A porta de diagnóstico ou OBD-II, é considerada um dos melhores pontos de ataque para explorar um automóvel, dado que está ligada directamente ao CANbus e permite não só a leitura mas também a escrita no canal de comunicação [40]. Para executar o ataque e tendo em conta que a OBD-II é uma porta local, podem ser usadas duas técnicas: Ligação remota com recurso a um adaptador do tipo ELM327[56], em que o atacante pode estar a uma distancia até 10 metros. Ou com recurso a uma adaptador USB-CAN, que permitem a ligação directa a um computador.

Figura 2.12: Exploração remota da porta OBD-II. Figura retirada de Zhang et al. [40]

Em ambos os cenários de ataque, é necessário um acesso físico à OBD-II, aqui pode aplicar-se técnicas de engenharia social para conseguir a ligação à porta de diagnostico, uma vez com acesso físico à viatura pode ser instalado um dispositivo[57] que permite o controlo da OBD-II atraves de Wi-Fi, ou seja a partir desse momento o atacante não necessita de ter mais acesso físico à viatura.

Umas das técnicas mais usadas neste vector de ataque é a intercepção de todo o tráfego no CANbus e a deteção dos valores que variam quando é executada determinada acção, por exemplo

[54] Ransom malware, ou ransomware, é um tipo de malware que impede os utilizadores de aceder ao seu sistema ou ficheiros pessoais e exige-lhes o pagamento de um resgate para devolver o acesso [38].
[55] Dipositivo plug-and-play que ligados à porta USB disponibilizam uma ligação ethernet
[56] Ficha OBD para diagnóstico simples e leitura por ligação Bluetooth
[57] Um exemplo deste dispositivo é o OBD ELM327 Wi-Fi, cujas especificações podem ser vistas em https://www.totalcardiagnostics.com/elm327-Wi-Fi/

quando se liga o pisca qual a alteração na frame do CANbus, fazendo este mapeamento é possível replicar estas frames e injetar directamente no CANbus obtendo uma acção por parte do veículo, não nos podemos esquecer que não existe validação da origem das frames neste protocolo e que todos os componentes conseguem ver todo o trafego.

Exploração TPMS

O TPMS envia a cada 60 a 90 segundos informação sobre a pressão dos pneus, rotação da roda e temperatura bem como aviso sobre o estado da bateria dos próprios sensores, os dados são depois transmitidos à ECU e apresentados ao condutor no painel de instrumentos [11], cada TPMS tem um identificador único de forma a não existir interferência de veículos próximos com o mesmo sistema [41], dependendo do veículo a informação existem sistemas de TPMS que só enviam os dados quando se atinge os 50 km/h e existem outros que mesmo com o veículo parado emitem a informação, potenciado assim a possibilidade de um ataque de tracking[58].

Um dado muito importante do desenhos das comunicações dos sistemas TPMS é o facto de não existir qualquer cifra na transmissão de dados, deixando assim a porta aberta para que qualquer um explore estes sistemas, por outro lado a desactivação do TPMS é algo que exige algum conhecimento e ferramentas especificas para desmontar o pneu, ou seja não está ao alcance de qualquer um.

Tendo em conta que a tecnologia de comunicação que o TPMS utiliza funciona em distâncias até 10 metros, o atacante deverá estar muito próximo da viatura, ou em alternativa terá de recorrer a um amplificador de sinal, usando esta técnica, investigadores da University of South Carolina conseguiram explorar o TPMS a uma distância de 40 metros [42].

Os principais ataques ao TPMS são:

- Tracking
- Ativação de Sinal
- Envio de pacotes forjados
- Despoletar eventos

Tracking, neste cenário o atacante recorre aos identificadores únicos do TPMS para saber por onde anda a viatura, e consequentemente os seus ocupantes, sendo classificado como quebra de privacidade é impossível de combater devido à arquitectura do sistema.Por exemplo numa determinada estrada podemos saber quando é que o veículo entrou e saiu e consequentemente a que velocidade circulava.

[58] Ataque passivo que permite o rastreamento de determinado sistema com base em identificadores únicos

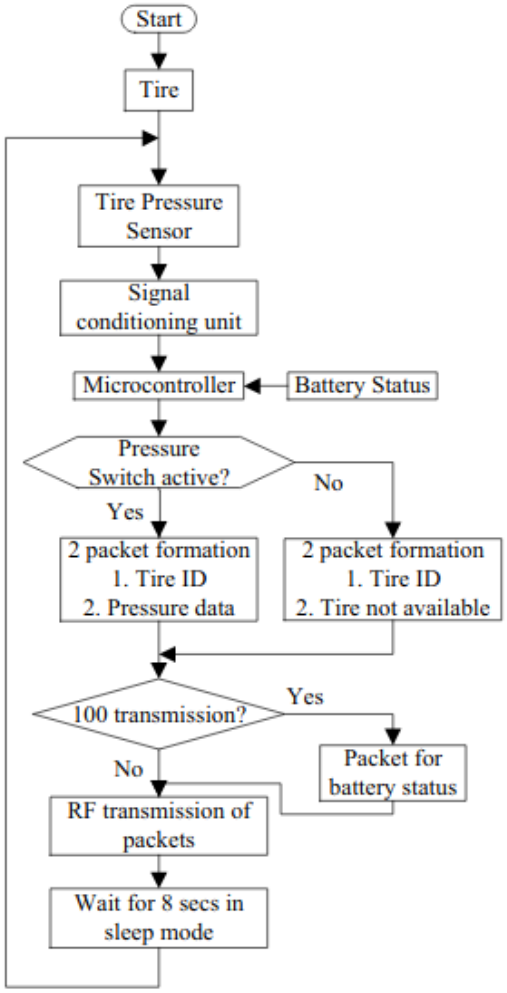

Figura 2.13: Fluxograma do funcionamento do TPMS. Figura retirada de Hasan et al. [41]

Activação de Sinal, embora os sensores só emitam um sinal entre cada 60 a 90 segundos é possível ao atacante forçar esse sinal antes deste período de tempo, dado que não existe nenhuma validação nos sensores.

Envio de pacotes forjados, aqui podemos enviar informação alterada para o receptor de informação do TPMS, por exemplo mesmo que o pneu tenha a pressão correcta pode ser enviada a informação que está vazio obrigando o condutor a parar em determinado local, esta técnica pode ser usada como um ataque de DOS provocando uma grave instabilidade na ECU [42].

Despoletar eventos, com a leitura dos identificadores únicos do TPMs, podemos usar essa informação para despoletar alguma acção, que pode ser algo tão inócuo com o o abrir de uma garagem ou tão crítico como a activação de uma bomba [11].

Exploração GSM

A maioria dos automóveis actuais têm uma ligação GSM de forma a poderem executar uma serie de funções que vão desde as actualizações remotas até à disponibilização de Internet aos ocupantes do veículo.

Para explorar o GSM a técnica mais usada é recorrer a uma antena falsa denominada de rogue base station. [43] O conceito deste ataque é levar a que a viatura equipada como tecnologia GSM se ligue a esta antena em vez de ligar a uma antena fidedigna do operador de telecomunicações. Com isto o atacante consegue interceptar e analisar o tráfego GSM e em alguns casos pode mesmo alterar o conteúdos dos dados transmitidos.

Tendo em conta que muitos fabricantes recorrem a actualizações OTA, é fácil entender que este tipo de ataque pode provocar uma exploração permanente da viatura quando o atacante consegue reescrever o seu firmware.

Este tipo de vector de ataque é facilitado pelo desenho do protocolo de comunicações em redes GSM, dado que o processo de autenticação é executado só no sentido do dispositivo para a rede em que se quer ligar, por outro lado as rogue base station permitem que seja feito um downgrade na tecnologia da comunicação, ou seja pode forçar o equipamento para que não use cifra nas comunicações, de forma a interceptar todo o tráfego , nomeadamente chamadas de voz, sms e comunicações pela internet.

Contrariamente a outras tecnologias de comunicações sem fio, em Portugal está expressamente proibida pela ANACOM[59] a utilização de equipamentos que simulem antenas GSM, pelo que não será possível nesta dissertação a exploração de vulnerabilidades GSM em veículos automóveis.

Exploração Aplicações

A exploração de aplicações pode ser feita apoiada em várias técnicas. Essas técnicas estão categorizas das no OWASP[60] de forma a enumerar as formas de explorar uma aplicação web. Um dos principais projectos da OWASP é o OWASP TOP 10[61], que é uma listagem das 10 falhas mais comuns na aplicações web. Este é um excelente ponto de partida para explorar as aplicações que são usadas num determinado veículo, estas podem ter várias funções tais como abrir as portas, localizar a viatura ou simplesmente controlar o rádio.

Cada uma das categorias do OWASP TOP TEN tem a seguinte descrição[44]:

- A1:2017, falhas de injecção, tais como injecções de SQL, OS e LDAP ocorrem quando dados não-

[59] Autoridade Nacional de Comunicações, autoridade reguladora em Portugal das comunicações postais e das comunicações electrónicas

[60] Organização sem fins lucrativos cujo objectivo é principal disponibilizar informação fidedigna e independente na área da cibersegurança

[61] Projecto disponível em https://owasp.org/www-project-top-ten/

confiáveis são enviados para um interpretador como parte de um comando ou consulta legítima. Os dados hostis do atacante podem enganar o interpretador levando-o a executar comandos não pretendidos ou a aceder a dados sem a devida autorização.

- A2:2017, as funções da aplicação que estão relacionadas com a autenticação e gestão de sessões são muitas vezes implementadas incorrectamente, permitindo que um atacante possa comprometer passwords, chaves, tokens de sessão, ou abusar doutras falhas da implementação que lhe permitam assumir a identidade de outros utilizadores (temporária ou permanentemente).

- A3:2017, muitas aplicações web e APIs não protegem de forma adequada dados sensíveis, tais como dados financeiros, de saúde ou dados de identificação pessoal (PII). Os atacantes podem roubar ou modificar estes dados mal protegidos para realizar fraudes com cartões de crédito, roubo de identidade, ou outros crimes. Os dados sensíveis necessitam de protecções de segurança extra como encriptação quando armazenados ou em trânsito, tal como precauções especiais quando trocadas com o navegador web

- A4:2017, muitos processadores de XML mais antigos ou mal configurados avaliam referências a entidades externas dentro dos documentos XML. Estas entidades externas podem ser usadas para revelar ficheiros internos usando o processador de URI de ficheiros, partilhas internas de ficheiros, pesquisa de portas de comunicação internas, execução de código remoto e ataques de negação de serviço, tal como o ataque Billion Laughs

- A5:2017, as restrições sobre o que os utilizadores autenticados estão autorizados a fazer nem sempre são correctamente verificadas. Os atacantes podem abusar destas falhas para aceder a funcionalidades ou dados para os quais não têm autorização, tais como dados de outras contas de utilizador, visualizar ficheiros sensíveis, modificar os dados de outros utilizadores, alterar as permissões de acesso, entre outros.

- A6:2017, as más configurações de segurança são o aspecto mais observado nos dados recolhidos. Normalmente isto é consequência de configurações padrão inseguras, incompletas ou ad hoc, armazenamento na nuvem sem qualquer restrição de acesso, cabeçalhos HTTP mal configurados ou mensagens de erro com informações sensíveis. Não só todos os sistemas operativos, frameworks, bibliotecas de código e aplicações devem ser configurados de forma segura, como também devem ser actualizados e alvo de correções de segurança atempadamente.

- A7:2017, as falhas de XSS ocorrem sempre que uma aplicação inclui dados não-confiáveis numa nova página web sem a validação ou filtragem apropriadas, ou quando actualiza uma página web existente com dados enviados por um utilizador através de uma API do browser que possa criar JavaScript. O XSS permite que atacantes possam executar scripts no browser da vítima, os quais podem raptar sessões do utilizador, descaraterizar sites web ou redirecionar o utilizador para sites maliciosos.

- A8:2017, desserialização insegura normalmente leva à execução remota de código. Mesmo que

isto não aconteça, pode ser usada para realizar ataques, incluindo ataques por repetição, injecção e elevação de privilégios.

- A9:2017, componentes tais como, bibliotecas, frameworks e outros módulos de software, são executados com os mesmos privilégios que a aplicação. O abuso dum componente vulnerável pode conduzir a uma perda séria de dados ou controlo completo de um servidor. Aplicações e APIs que usem componentes com vulnerabilidades conhecidas podem enfraquecer as defesas da aplicação possibilitando ataques e impactos diversos.

- A10:2017, o registo e monitorização insuficientes, em conjunto com uma resposta a incidentes inexistente ou insuficiente permite que os atacantes possam abusar do sistema de forma persistente, que o possam usar como entrada para atacar outros sistemas, e que possam alterar, extrair ou destruir dados. Alguns dos estudos demonstram que o tempo necessário para detectar uma violação de dados é de mais de 200 dias e é tipicamente detectada por entidades externas ao invés de processos internos ou monitorização.

Tabela 2.2: OWASP TOP TEN

Id	Categoria
A1:2017	Injeção
A2:2017	Quebra de Autenticação
A3:2017	Exposição de Dados Sensíveis
A4:2017	Entidades Externas de XML (XXE)
A5:2017	Quebra de Controlo de Acessos
A6:2017	Configurações de Segurança Incorrectas
A7:2017	Cross-Site Scripting (XSS)
A8:2017	Desserialização Insegura
A9:2017	Utilização de Componentes Vulneráveis
A10:2017	Registo e Monitorização Insuficiente

Como podemos observar são várias as explorações que podem ser associadas a cada categoria, no topo e consequentemente a falha mais comum em aplicações web é a injeção, que pode ser por exemplo uma SQL[62] injection, onde através de argumentos enviados para o servidor e não tratados pelo código fonte, é possível navegar em toda a base de dados ou mesmo executar comandos no sistema operativo que serve de suporte à base de dados.

Ao usar aplicações nos seus automóveis, os fabricantes aumentaram de forma exponencial a sua superfície de ataque, tendo em conta que existem várias formas de explorar essa aplicações e obter o controlo total ou parcial da viatura.

[62] A linguagem SQL é uma norma do sistemas de gestão de bases de dados relacionais

2.2.5 Exemplos de hacking automóvel

Nos últimos anos têm vindo a publico vários ataques com sucesso a automóveis, estes têm sido muito importantes para despoletar a consciencialização para a necessidade de um aumento nos níveis de segurança dos sistemas de informação de um veículo, só com o trabalho dos vários investigadores e *hackers* a nível mundial é possível testar ao máximo os veículos e desta forma contribuir para a sua segurança.

Um dos casos de hacking automóvel mais mediático, foi a exploração executada pelos investigadores *Charlie Miller* e *Chris Valasek*[63], que em 2015 conseguiram o controlo total de um *Jeep Cherokee*[64]. Através de uma vulnerabilidade no sistema *Uconnect*[65], foi possível aos investigadores controlar remotamente o ar condicionado, os travões e o acelerador de uma viatura em andamento. A vulnerabilidade expunha o endereço IP da viatura, a partir dai foi possível comprometer a unidade multimédia e injectar um novo *firmware* que deu acesso ao CANbus [45].

Samy Kamkar, um *hacker* americano, conseguiu executar um ataque contra a *OnStar RemoteLink*[66], com recuso a um *Raspberry Pi*[67]. Este investigador desenvolveu um dispositivo denominado de *OwnStar*. Com recurso a esse dispositivo conseguiu localizar, destrancar e ligar qualquer veículo que utilizasse o *OnStar RemoteLink* [46].

Vários modelos das marcas *Audi* e *Volkswagen* tiveram as suas vulnerabilidades exploradas para obter o controlo do sistema de navegação e microfone, os investigadores *Daan Keuper* e *Thijs Alkemade* conseguiram através do WiFi comprometer o sistema multimédia, numa fase posterior constatou-se que também era possível executar o ataque através de redes GSM [47].

A BMW também teve os seus sistemas comprometidos por uma vulnerabilidade das mais perigosas que podem existir, as 0 day, esta falha permitia a manipulação das contas no seu portal e consequente manipulação dos dados nas viaturas [48]. Esta descoberta foi feita por investigadores da Vulnerability Labs. A marca levou quase dois meses a resolver este problema que afectava todos os clientes BMW que usassem o portal.

Em 2019, uma equipa composta pelos investigadores *Amat Cama* e *Richard Zhu*, ganhou com prémio um Tesla num concurso de *hacking* denominado de *Pwn2Own*[68], ao explorar uma vulnerabilidade no *browser* de um *Tesla* que permitiu a execução remota de código e consequente exploração do sistema multimédia [49].

Já em 2020, *Lennert Wouters* um investigador da *Belgian university KU Leuven*, conseguiu através

[63] Investigadores de segurança da empresa *Cruise Automation*
[64] veículo produzido pela *Jeep* e classificado como SUV
[65] Sistema multimédia usado pela *Alfa Romeo*, *Fiat*,*Chrysler*, *Dodge*, *Fiat*, *Jeep*, *Maserati* e *Ram*
[66] Aplicação da *General Motors* que permite controlar remotamente várias funções das suas viaturas, tais como a buzina, abrir e fechar as portar e ligar o motor
[67] Computador de baixo custo
[68] Organizada pela *Trend Micro's* Zero-Day Initiative, é considerada a melhor prova a nível mundial para investigadores de segurança

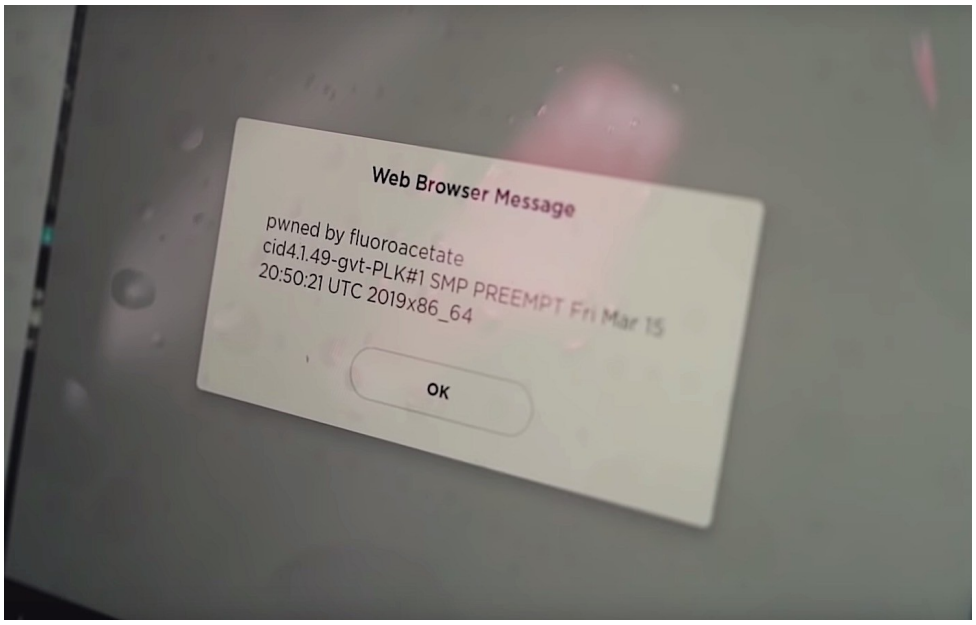

Figura 2.14: Evidência do tesla comprometido na Pwn2Own, Figura retirada de Cimpanu [49]

do *Bluetooth* reescrever o *firmware* do comando remoto do *Tesla Model X*. Desta forma em cerca de 90 segundos *Lennert Wouters* abriu o veículo. As vulnerabilidades encontradas e mais criticas são a falta de validação do dos *updates* de *firmware* por parte do comando remoto bem como o facto de não existir nenhum verificação quando um novo comando é emparelhado com a viatura [50].

Capítulo 3

Metodologia

Tendo em conta a variedade de superfícies de ataque a um automóvel, no modelo de exploração apresentado nesta dissertação irá incidir sobre três vectores de ataque, um local e dois remotos, o vector de ataque remoto irá incidir sobre a ficha OBD-II, por sua vez os ataques remotos irão explorar as vulnerabilidades de comunicações com recurso a WiFi e Bluetooth .

O método utilizado passa pela recolha de informação da viatura, nomeadamente superfícies de ataque disponíveis, tais como possíveis CVE atribuídos ao veiculo ou a componentes que façam parte dos sistemas do automóvel.

Os primeiros dados a serem recolhidos serão:

- Ano de fabrico
- Marca
- Modelo
- Versão

Tendo esta informação, o primeiro passo será pesquisar possíveis CVE, de forma a comprometer a viatura em análise, irá ser usada a Common Vulnerabilities and Exposures[1]. A importância desta etapa advém do facto de poderem existir vulnerabilidades não corrigidas na viatura. O fluxograma de pesquisa de vulnerabilidades está representado na figura 3.1

De seguida irão ser mapeadas as superfícies de ataque definidas neste modelo, a cada uma será atribuída um valor S ou N em que S significa que essa superfícies de ataque está presente e um N significa que a viatura não tem essa superfície de ataque.

[1] https://cve.mitre.org

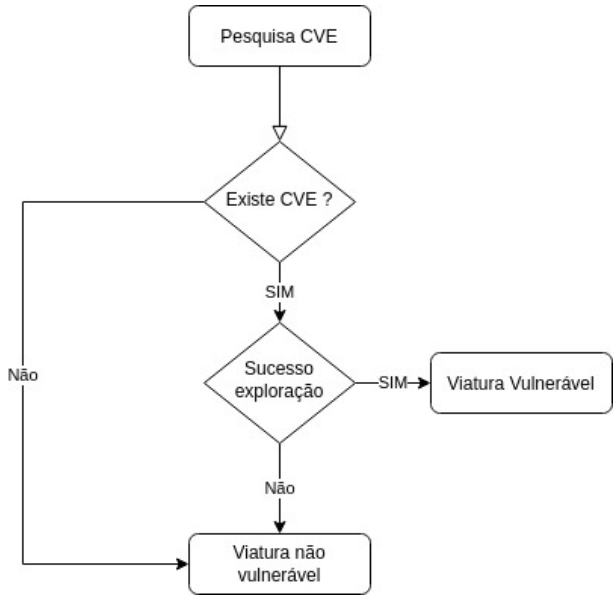

Figura 3.1: Diagrama de pesquisa de CVE

A cada superfície de ataque, e tendo em conta os vectores de ataque definidos no modelo de hacking, é atribuído uma ferramenta de exploração de vulnerabilidades, caso existam as vulnerabilidades são então classificadas com a calculadora de CVSS disponível em `https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator`.

Tabela 3.1: Superfícies de ataque da viatura

Superfície de Ataque	Valor(S/N)
Existem CVE disponíveis?	
A viatura tem WiFi?	
A viatura tem Bluetooth ?	
A viatura tem porta ODB-II	

Se o automóvel estiver capacitado a tecnologia de comunicações sem fios WiFi, teremos então três tipos de exploração WiFi neste modelo, a saber:

- WPA crack e WiFiphisher para ter acesso à rede interna do automóvel
- Denial of service, em que o objectivo é a interrupção das comunicações WiFi entre o automóvel e outros dispositivos
- Tracking com vista a identificar locais por onde a viatura passou bem como a sua presença ou não em determinado local.

A sequência da exploração será:

- Identificação se a viatura disponibiliza um Hotspot.

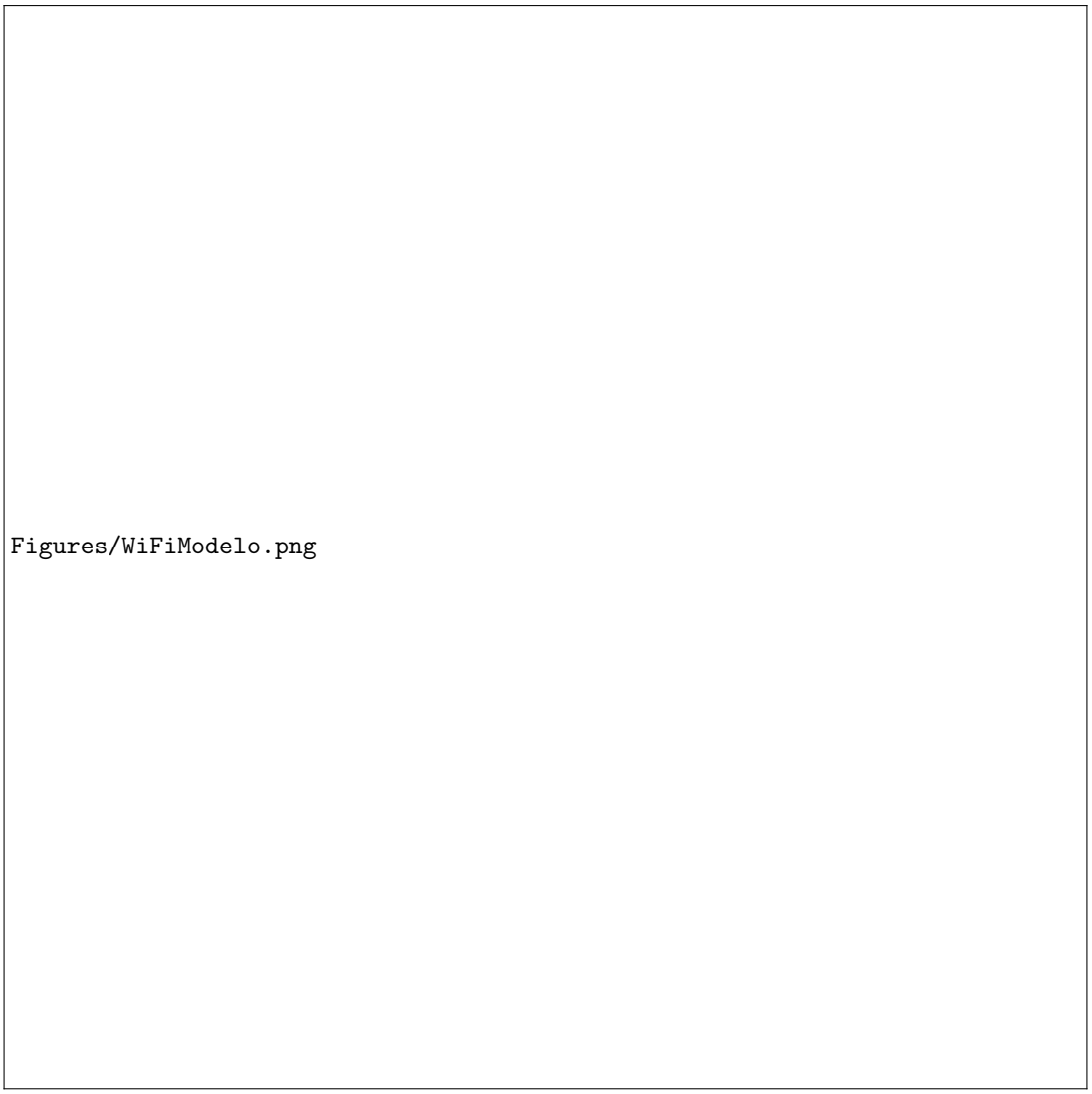

Figura 3.2: Módulo de exploração WiFi

- Identificação se a viatura se liga a um Hotspot.
- Ataque WPA CRACK
- Ataque DEAUTH
- Ataque Evil Twin
- Ataque DOS
- Ataque Probe
- Exploração da rede interna
- Classificação das vulnerabilidades

Relativamente à exploração Bluetooth , será testada a visibilidade do automóvel em determinado local com o seu identificador único da placa de Bluetooth , os serviços que oferece a através do Bluetooth e a sua resiliência a um denial of service, os procedimentos serão os seguintes:

- Identificação do dispositivo
- Descoberta dos serviços Bluetooth
- Execução de l2ping ao dispositivo
- Validação de vulnerabilidade a DOS
- Classificação das vulnerabilidades

A exploração da porta OBD-II dentro deste modelo da hacking automóvel, terá como objectivo a intercepção de tráfego CANbus, identificação dos código de activação dos dois piscas em simultâneo e posterior injecção de pacotes CAN de forma a activar os piscas sem intervenção do condutor, serão seguidos os seguintes passos:

- Localização da porta OBD-II
- Ligação do USB2CAN
- Execução do cansniffer
- Activação manual dos piscas
- Intercepção das mensagens CAN que exigem alteração de valores
- Identificação da mensagem de ligar os piscas
- Injecção da mensagem no CANbus
- Observação do sucesso do procedimento
- Classificação das vulnerabilidade

3.1 Exploração ODB-II

Como vimos anteriormente através do ODB-II temos acesso directo ao CANbus, no entanto para que seja possível a analise dos pacotes é necessário recorrer a algum hardware e software. Do lado do hardware e no âmbito desta dissertação, foi usado o Korlan USB2CAN[2] cujas especificações técnicas podem ser vistas em https://www.8devices.com/media/products/usb2can_korlan/, este dispositivo permite a ligação a sistemas Windows e Linux, na página de suporte do produto existe inclusivamente um plugin para o wireshark de modo a que seja possível de uma maneira mais fácil a captura de pacotes.

[2]Adaptador desenvolvido pela empresa 8devices, que permite a monitorização e comunicação com uma rede CAN, permite a interligação ente uma porta USB e OBD-II

Com a implementação da framework SocketCAN[3], o Linux consegue comunicar com o CANbus nativamente, para que isso aconteça é necessário iniciar o interface de rede can0 com o seguinte comando:

```
$ sudo ip link set can0 up type can bitrate 125000 sample-point 0.875
```

O parâmetro bitrate 125000 especifica uma velocidade de ligação de 125Kbps no interface CAN, este valor poderá ser ajustado consoante o modelo a ser testado, uma das velocidade mais comuns é 500Kbps ou seja bitrate 500000.

Relativamente ao software para exploração da OBD-II é usado o can-utils na sua versão v2020.11.0 [4], a instalação é feita com seguinte comando:

```
$ sudo apt install can-utils
```

O pacote can-utils disponibiliza as seguintes ferramentas:

- candump: mostra, filtra e faz log dos pacotes.
- canplayer: executa um replay dos pacotes.
- cansend: envia uma única frame
- cangen: gera tráfego aleatorio no CAN
- cansequence: envia e valida sequências de frames CAN com um payload incremental.
- cansniffer: mostra a diferença entre mensagens CAN.

Um dos processos mais importantes usado no modelo de exploração automóvel através da OBD-II será a comparação de mensagens CAN, isto porque terá de existir um processo de mapeamento das mensagens utilizando uma técnica de divisão em metade das massagens recebidas até que seja encontrada a mensagem que corresponde à acção desencadeada. Por exemplo se quisermos saber qual o pacote CAN que corresponde a ligar o pisca direito, devemos seguir a seguinte sequência[11]:

- Inicio da gravação.
- Ligar o pisca direito.
- Parar a gravação.
- Injectar a gravação
- Validar se o pisca direito foi accionado

[3]Contribuição da Volkswagen Research para o kernel do Linux
[4]O código fonte está disponível em https://github.com/linux-can/can-utils

Figura 3.3: Korlan USB2CAN utilizado nos testes

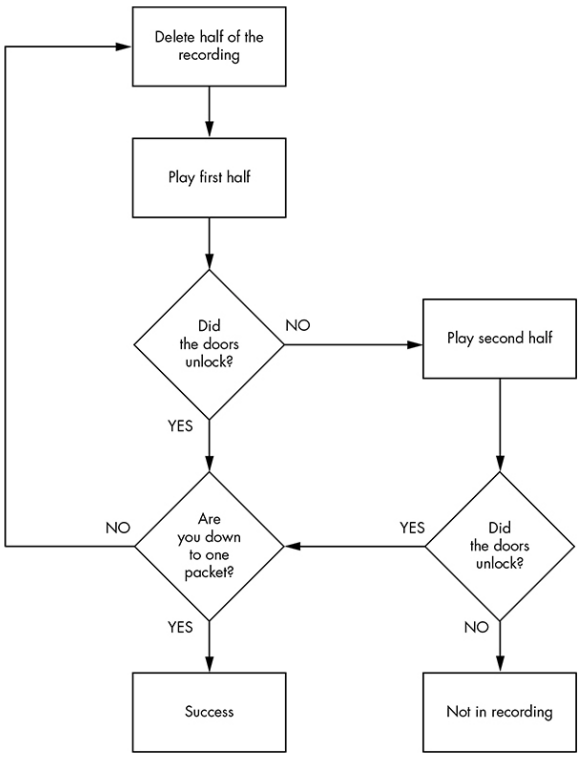

Figura 3.4: Isolamento do pacote que abre as portas. Figura retirada de Smith [11]

Tendo em conta a quantidade enorme de pacotes que podem circular, isolar o pacote CAN que faz accionar o pisca pode ser um trabalho moroso, podemos no entanto usar uma técnica que consiste em ir dividindo os dados capturados em metades, até isolar o pacote que executa a acção.

Para gravar um log dos pacotes que circulam no CANbus deverá ser executado o comando:

```
$ candump -l can0
```

O resultado do comando será um ficheiro de texto onde estarão todos os pacotes que circulavam no CANbus o nome do ficheiro tem o seguinte formato candump-2020-12-27_123930.log, de seguida a injecção desta captura é assegurada como seguinte comando:

```
$ canreplay -I candump-2020-12-27_123930.log can0
```

Outra forma de identificar determinado pacote será usar a ferramenta cansniffer, que nos permite visualizar em tempo real quais as mensagens que sofrem alterações quando é executada determinada acção, a sintaxe de utilização do cansniffer será:

```
$ cansniffer -c can0
```

A recolha de dados das vulnerabilidades OBD-II será feita da tabela 3.2

Tabela 3.2: Vulnerabilidades OBD-II

Exploração OBD-II	Valor (S / N)
CANbus vulnerável?	
Intercepção de pacotes?	
Injecção de pacotes?	

3.2 Exploração WiFi

Dentro da exploração da superfície de ataque WiFi, usaremos as seguintes técnicas:

- DOS
- DEAUTH
- EVIL TWIN
- WPA CRACK
- PROBE

A exploração do WiFi exige que seja usada uma antena com um chipset especifico, de forma a que seja possível colocar este dispositivo em modo monitor[5], só assim conseguimos interceptar o tráfego

[5]Modo monitor é um dos quatro modos de uma placa de rede WiFi e que permite a intercepção de pacotes sem estar associado a uma rede WiFi.

WiFi e injectar pacotes na rede. Nos testes executados nesta dissertação a escolha recaiu sobre o modelo ALFA Network AWUS036NH WLAN 150 Mbit/s dotado de um chipset Atheros AR9271[6].

A antena deverá ser colocada em modo monitor através do comando airmon-ng[7] com a seguinte sintaxe:

```
$ sudo airmon-ng start wlan0
```

A recolha de dados das vulnerabilidades WiFi será feita da tabela 3.3

Tabela 3.3: Vulnerabilidades WiFi

Exploração WiFi	Value (S / N)
Vulnerável a WPA crack?	
Vulnerável a Evil Twin?	
Vulnerável a DOS?	
Vulnerável a PROBE	
Vulnerável a DEAUTH	

3.2.1 WiFi DOS

Para executar os ataques de DOS, a software usado será o mdk3 versão 3.0 v6, cujo código fonte pode ser encontrado em https://github.com/charlesxsh/mdk3-master, esta aplicação permite executar vários dois de ataque DOS a saber:

- Beacon Flood Mode
- Authentication DoS mode

Beacon Flood Mode consiste em encher o espaço de comunicações WiFi com dezenas ou centenas de SSID com nomes aleatórios ou então copia de um SSID existente, já o Authentication DoS mode consiste em fazer vários pedidos de associação ao AP fazendo com que deixe de responder ao pedidos legítimo.

As opções de utilização do mdk3 São as seguintes:

- b - Beacon Flood Mode Sends beacon frames to show fake APs at clients. This can sometimes crash network scanners and even drivers!
- a - Authentication DoS mode Sends authentication frames to all APs found in range. Too much clients freeze or reset some APs.

[6] Especificações podem ser vistas em https://www.alfa.com.tw/products/awus036nha?variant=36473966166088
[7] Disponível no pacote aircrack-ng, pode ser instalado com *sudo apt install aircrack-ng*

- p - Basic probing and ESSID Bruteforce mode Probes AP and check for answer, useful for checking if SSID has been correctly decloaked or if AP is in your adaptors sending range SSID Bruteforcing is also possible with this test mode.
- d - Deauthentication / Disassociation Amok Mode Kicks everybody found from AP
- m - Michael shutdown exploitation (TKIP) Cancels all traffic continuously
- x - 802.1X tests
- w - WIDS/WIPS Confusion Confuse/Abuse Intrusion Detection and Prevention Systems
- f - MAC filter bruteforce mode This test uses a list of known client MAC Adresses and tries to authenticate them to the given AP while dynamically changing its response timeout for best performance. It currently works only on APs who deny an open authentication request properly
- g - WPA Downgrade test deauthenticates Stations and APs sending WPA encrypted packets. With this test you can check if the sysadmin will try setting his network to WEP or disable encryption.

Logo para um ataque de DOS temos o seguinte comando:

```
$ mdk3 wlan0mon b
```

3.2.2 DEAUTH

Com recurso ao software mdk3, utilizando a opção 'd' é possível executar um ataque de desautenticação em massa, ou seja todas as estações de trabalho ligadas ao AP vão perder a sua ligação. Com isto é possível afectar permanentemente as comunicações via WiFi se o ataque tiver tempo indefinido. A sintaxe de utilização do comando será:

```
$ sudo mdk3 wlan0mon d
```

3.2.3 WPA CRACK

Nesta exploração o objectivo é interceptar um handshake, para que através da técnica de brute force com base num dicionário, seja obtida a password da rede WPA

O software usado será o WiFiTE na sua versão 2.2.3[8]. Esta aplicação permite que todo o processo seja feito de forma automatizada, bastando para isso identificar o alvo a atacar e esperar que seja capturado o handshake e iniciado o processo de cracking da password da rede WiFi. A instalação da ferramenta é feita da seguinte forma:

[8]Código fonte está disponível em `https://github.com/derv82/WiFite2`

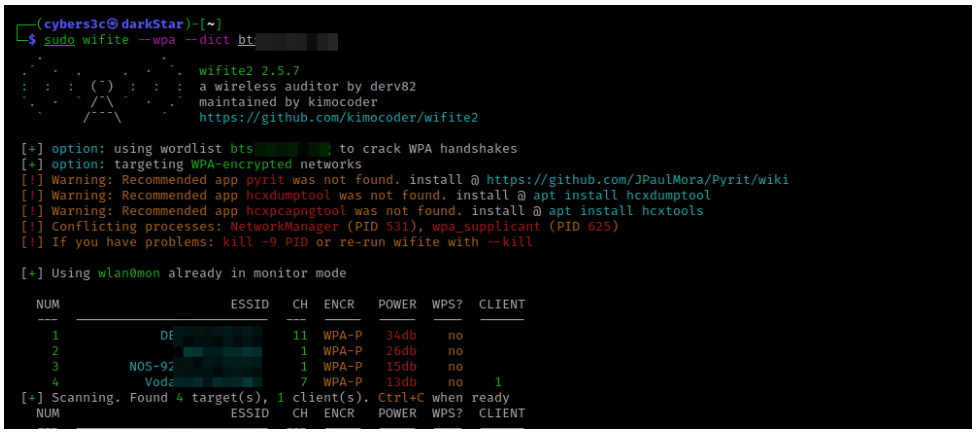

Figura 3.5: WiFite em utilização

```
$ git clone https://github.com/derv82/WiFite2.git
$ cd WiFite2
$ ./WiFite.py
```

O ataque é feito da seguinte forma:

```
$ WiFite -dict dicionario.txt
```

3.2.4 EVIL TWIN

Para o ataque evil twin iremos utilizar o sotware WiFiphisher versão 1.4[9] . O WiFiphisher permite executar ataques de evil twin a redes WiFi de forma automatizada.

Para iniciar o ataque, o utilizador deve indicar qual o é o cenário de ataque que deseja, ou seja o que acontece depois da vitima ser desligada da rede WiFi fidedigna e passar a estar ligada à rede comprometida com o mesmo SSID, estes cenários podem ser:

- firmware-upgrade, na vitima irá aparecer uma página igual à do router onde é pedida a password da rede WiFi.
- plugin_update, simula a necessidade de executar um update a um plugin e desta forma instala software malicioso na estação de trabalho da vitima.
- oauth-login, a vitima vê uma pagina de login do facebook, permitindo assim ao atacante interceptar estas credenciais.

Com exemplo de utilização do WiFiphisher temos:

```
$ WiFiphisher --essid "BMW CONNECT" -p oauth-login -kB
```

[9]Está disponível em https://github.com/WiFiphisher/WiFiphisher

Aqui é criada uma rede WiFi com o SSID ”BMW CONNECT”e que irá solicitar à vitima as suas credenciais de Facebook[10], dado que foi usado o parâmetro -p oauth-login

3.2.5 PROBE SNIFFING

De forma a executar um probe snnifing das viaturas, e saber a que redes já estiveram ligadas e onde, o software usado será o probeSniffer versão 3.0, cujo código fonte pode ser encontrado em https://github.com/xdavidhu/probeSniffer. Esta ferramenta permite analisar os probe request e armazena os mesmo numa base de dados SQLite[11], os argumentos que o probeSniffer pode receber são os seguintes:

- -h / display the help message
- -d / do not show duplicate requests
- -b / do not show ’broadcast’ requests (without ssid)
- -a / save duplicate requests to SQL
- –filter / only show requests from the specified mac address
- –norssi / do not include RSSI in output
- –nosql / disable SQL logging completely
- –addnicks / add nicknames to mac addresses
- –flushnicks / flush nickname database
- –noresolve / skip resolving mac address
- –debug / turn debug mode on

A sitaxe de utilização será então:

```
$ python3 probeSniffer.py wlan0
```

Após a captura e tratamento dos probe requests, podemos recorrer à geolocalização para saber os locais onde a viatura esteve, e isso é conseguido através do portal WIGLE, disponível em https://wigle.net/ e que agrega redes WiFi, permite a pesquisa e apresenta a localização exacta de cada SSID num mapa.

Um outro dado interessante que podemos retirar do probeSniifer é a potencia do sinal emitido pelo dispositivo, ora isto pode ser usado para determinar a distância do dispositivo à antena e assim calcular a sua posição ou velocidade de deslocação entre duas antenas que estejam a fazer probeSniffer.

[10] Rede Social

[11] SQLite é uma biblioteca escrita na linguagem C, está embutida na aplicação e não funciona no modelo cliente servidor

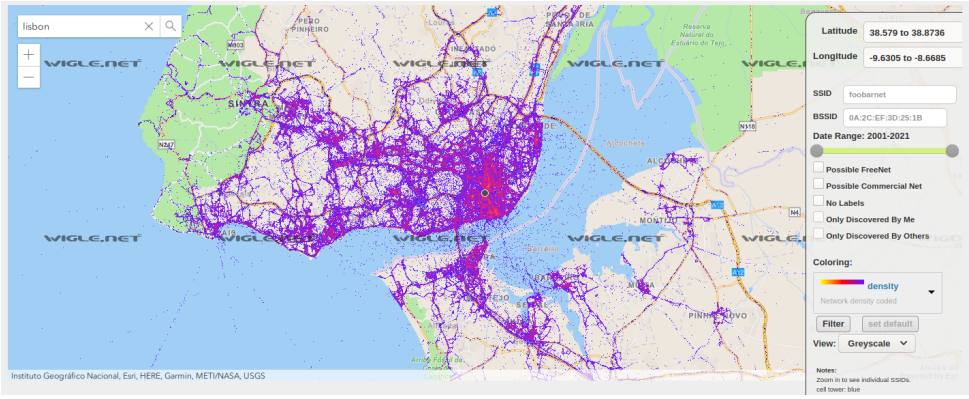

Figura 3.6: Localização das redes WiFi em Lisboa na plataforma WIGLE

3.2.6 Exploração da rede interna

Após o sucesso em comprometer a rede WiFi e consequente associação a essa rede, o próximo passado é a exploração da rede interna através de um varrimento para encontrar máquinas e serviços disponíveis.

Para isso será usado o software *nmap* na sua versão 7.80.[12] Esta ferramenta permite vários tipos de descoberta maquinas, serviços e vulnerabilidades, no âmbito desta dissertação serão usados os seguintes argumentos com o *nmap*:

Descoberta de serviços

```
$ nmap -sX IP
```

Descoberta de vulnerabilidades na rede

```
$ nmap -v --script vuln (network)
```

3.3 Exploração Bluetooth

Para os ataques Bluetooth de uma viatura, e no âmbito do modelo de exploração desta dissertação, irá um dos software a ser usado é o bluez[13], que contem várias ferramentas dedicadas ao Bluetooth , a instalação faz-se com o seguinte comando:

```
$ sudo apt install bluez
```

Após a instalação temos acesso a estas ferramentas:

- hcitool para o scan de dispositivos próximos,descoberta do tipo de dispositivo.

[12] Disponível em https://github.com/nmap/nmap.git
[13] Stack oficial do protocolo Bluetooth em Linux e que está disponível me http://www.bluez.org/

- sdptool para descoberta de serviços a correr nos dispositivos próximos.
- l2ping que serve para fazer um ping[14] a um dispositivo Bluetooth

Desta forma é utilizado o hcitool para descobrir os dispositivos bluetooh do veiculo que estão na proximidade através do comando

```
$ hcitool scan
```

Para obter mais informação sobre os dispositivos, nomeadamente a categoria a sintaxe será:

```
$ hcitool inq
```

A enumeração de serviços a correr no dispositivo bluettoth pode ser obtida com:

```
$ sdptool browse (MAC ADDRESS)
```

os resultados do sdptool pode ser filtrado com um grep com a seguinte sintaxe:

```
$ sdptool browse (MAC ADDRESS) | grep 'Service Name:'
```

O l2ping é usado para saber se um automóvel está próximo, a sintaxe do comando será:

```
$ l2ping (MAC ADDRESS)
```

A recolha de dados das vulnerabilidades Bluetooth será feita na tabela 3.4

Tabela 3.4: Vulnerabilidades Bluetooth

Exploração Bluetooth	Valor (S / N)
Visível no scan?	
Descoberta de serviços?	
Responde a l2ping?	

[14] Envia um pedido de echo e recebe a resposta

Capítulo 4

Resultados

4.1 Aplicação do modelo de exploração

Após a criação do modelo, foram testadas três viaturas, de marcas distintas de forma a explorar as várias superfícies de ataque. No processo de escolha dos automóveis a ser testados um dos critérios foi que pelo menos um deles tivesse mais de dez anos, de forma a avaliar a evolução ao longo do tempo, da segurança nos sistemas comuns às viaturas.

Tendo em conta que este trabalho poderá revelar vulnerabilidades e métodos de ataque de modelos específicos, foi tomada a decisão de ocultar a marca, modelo e versão das viaturas testadas, tendo sido as vulnerabilidades comunicadas aos fabricantes.

4.2 Resultados da amostra

4.2.1 Viatura A

Identificação da viatura

- Ano de fabrico: 2005
- Marca: Oculta
- Modelo: Oculto
- Versão: Oculta

Superfícies de ataque

Tabela 4.1: Superfícies de ataque da Viatura A

Superfície de ataque	valor(S/N)
Existem CVE disponíveis?	N
A viatura tem Wi-Fi?	N
A viatura tem Bluetooth ?	N
A viatura tem porta ODB-II	S

Exploração da porta OBD-II

Tabela 4.2: Vulnerabilidades OBD-II da viatura A

Exploração OBD-II	Valor (S / N)
CANbus vulnerável?	S
Intercepção de pacotes?	S
Injecção de pacotes?	S

Através da análise do tráfego do CANbus foi possível isolar o pacote que liga os piscas que tem o valor de 612#6230007F00000000, podemos então ligar os piscas remotamente através da injecção no CANbus usando o seguinte comando:

```
$ cansend can0 612#6230007F00000000
```

Foi explorado com sucesso a captura e injecção de pacotes no CANbus nesta viatura, de acordo com a calculadora CVSS temos a classificação de: 7.5

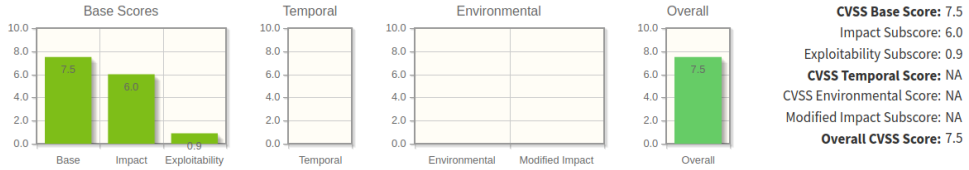

Figura 4.1: Classificação CVSS da exploração OBD-II da viatura A

4.2.2 Viatura B

Identificação da viatura

- Ano de fabrico: 2019
- Marca: Oculta
- Modelo: Oculto
- Versão: Oculta

Superfícies de ataque

Tabela 4.3: Superfícies de ataque da Viatura B

Superfície de ataque	valor(S/N)
Existem CVE disponíveis?	N
A viatura tem Wi-Fi?	S
A viatura tem Bluetooth ?	S
A viatura tem porta ODB-II	S

Exploração do Wi-Fi

A viatura B disponibiliza um *hotspot* aos seus ocupantes, embora se possa mudar a palavra passe desta rede WIFI, é impossível mudar o nome da rede nas configurações do automóvel. Os resultados da exploração dos ataques WIFI à viatura B estão descritos na tabela 4.4

Tabela 4.4: Vulnerabilidades WIFI viatura B

Exploração WIFI	Value (S / N)
Vulnerável a WPA crack?	S
Vulnerável a Evil Twin?	S
Vulnerável a DOS?	S
Vulnerável a PROBE	N
Vulnerável a DEAUTH	S

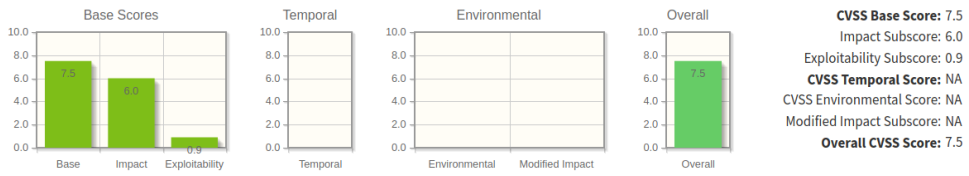

Figura 4.2: Classificação CVSS da exploração WIFI da viatura B

Exploração da rede interna

Máquinas e serviços descobertos

Nmap scan report for 172.16.222.97

Host is up (0.0016s latency).

Not shown: 992 filtered ports

PORT STATE SERVICE

3000/tcp closed ppp

4004/tcp open pxc-roid

4005/tcp open pxc-pin

4006/tcp open pxc-spvr

5000/tcp closed upnp

5001/tcp closed commplex-link

7000/tcp open afs3-fileserver

9877/tcp closed x510

Exploração do Bluetooth

Tabela 4.5: Vulnerabilidades Bluetooth viatura B

Exploração Bluetooth	Valor (S / N)
Visível no scan?	S
Descoberta de serviços?	S
Responde a l2ping?	S

Serviços de Bluetooth descobertos com a ferramenta sdptool da viatura B.

Serviço: Phonebook Access PCE Service Name: Phonebook Access PCE Service RecHandle: 0x10000 Service Class ID List: ”Phonebook Access - PCE”(0x112e) Language Base Attr List: code_ISO639: 0x656e encoding: 0x6a base_offset: 0x100 Profile Descriptor List: ”Phonebook Access”(0x1130) Version: 0x0102

Serviço: Advanced Audio Service Name: Advanced Audio Service Provider: OpenSynergy Service RecHandle: 0x10002 Service Class ID List: ”Audio Sink”(0x110b) Protocol Descriptor List: ”L2CAP”(0x0100) PSM: 25 ”AVDTP”(0x0019) uint16: 0x0103 Language Base Attr List: code_ISO639: 0x656e encoding: 0x6a base_offset: 0x100 Profile Descriptor List: ”Advanced Audio”(0x110d) Version: 0x0103

Serviço: AVRCP Remote Control Service Name: AVRCP Remote Control Service Provider: OpenSynergy GmbH Service RecHandle: 0x10003 Service Class ID List: ”AV Remote”(0x110e) ”AV Remote Controller”(0x110f) Protocol Descriptor List: ”L2CAP”(0x0100) PSM: 23 ”AVCTP”(0x0017) uint16: 0x0104 Profile Descriptor List: ”AV Remote”(0x110e) Version: 0x0106

Serviço: AAP Service Name: AAP Service RecHandle: 0x10005 Service Class ID List: ”Serial Port”(0x1101) Protocol Descriptor List: ”L2CAP”(0x0100) ”RFCOMM”(0x0003) Channel: 2 Language Base Attr List: code_ISO639: 0x656e encoding: 0x6a base_offset: 0x100 Profile Descriptor List: ”Serial Port”(0x1101) Version: 0x0102

Serviço: RecHandle Service RecHandle: 0x10006 Service Class ID List: ”PnP Information”(0x1200)

Serviço: CarPlay Service Name: CarPlay Service RecHandle: 0x10008 Service Class ID List: UUID 128: d31fbf50-5d57-2797-a240-41cd484388ec Language Base Attr List: code_ISO639: 0x656e encoding: 0x6a base_offset: 0x100

Serviço: IAP2 Service Name: IAP2 Service RecHandle: 0x10009 Service Class ID List: UUID 128: 00000000-deca-fade-deca-deafdecacaff Language Base Attr List: code_ISO639: 0x656e encoding: 0x6a base_offset: 0x100

Serviço: AAP Service Name: AAP Service RecHandle: 0x1000a Service Class ID List: UUID 128: 4de17a00-52cb-11e6-bdf4-0800200c9a66 Protocol Descriptor List: ”L2CAP”(0x0100) ”RFCOMM”(0x0003) Channel: 4 Language Base Attr List: code_ISO639: 0x656e encoding: 0x6a base_offset: 0x100

Serviço: RecHandle Service RecHandle: 0x1000b Service Class ID List: ”Message Access - MNS”(0x1133) Protocol Descriptor List: ”L2CAP”(0x0100) ”RFCOMM”(0x0003) Channel: 5 ”OBEX”(0x0008) Profile Descriptor List: ”Message Access”(0x1134) Version: 0x0104

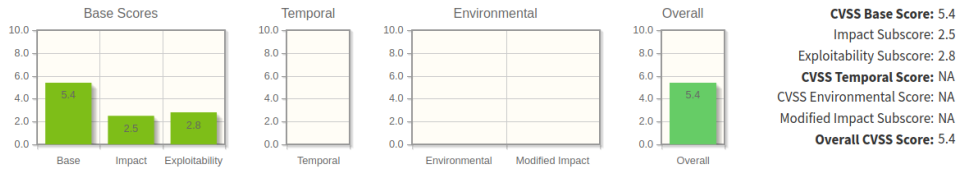

Figura 4.3: Classificação CVSS da exploração Bluetooth da viatura B

Exploração da porta OBD-II

Tabela 4.6: Vulnerabilidades OBD-II da viatura B

Exploração OBD-II	Valor (S / N)
CANbus vulnerável?	S
Intercepção de pacotes?	S
Injecção de pacotes?	S

Através da análise do tráfego do CANbus foi possível isolar o pacote que liga os piscas que tem o valor de 7D3#59170309000A024F, podemos então ligar os piscas remotamente através da injecção do pacote no CANbus usando o seguinte comando:

```
$ cansend can0 7D3#59170309000A024F
```

Foi explorado com sucesso a captura e injecção de pacotes no CANbus nesta viatura, de acordo com a calculadora CVSS temos a classificação de: 7.5

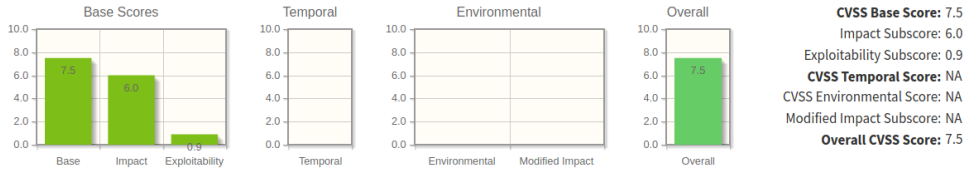

Figura 4.4: Classificação CVSS da exploração OBD-II da viatura B

4.2.3 Viatura C

Identificação da viatura

- Ano de fabrico: 2020
- Marca: Oculta
- Modelo: Oculto
- Versão: Oculta

Superfícies de ataque

Tabela 4.7: Superfícies de ataque da Viatura C

Superfície de ataque	valor(S/N)
Existem CVE disponíveis ?	N
A viatura tem WIFI ?	N
A viatura tem Bluetooth ?	S
A viatura tem porta ODB-II	S

Exploração do Bluetooth

Tabela 4.8: Vulnerabilidades Bluetooth viatura C

Exploração Bluetooth	Valor (S / N)
Visível no scan?	S
Descoberta de serviços?	S
Responde a l2ping?	S

Serviços de Bluetooth descobertos com a ferramenta sdptool da viatura C.

Serviço: AVRC Controller Service

Service Name: AVRC Controller Service RecHandle: 0x10000 Service Class ID List: ”AV Remote”(0x110e) ”AV Remote Controller”(0x110f) Protocol Descriptor List: ”L2CAP”(0x0100) PSM: 23 ”AVCTP”(0x0017) uint16: 0x0104 Profile Descriptor List: ”AV Remote”(0x110e) Version: 0x0105

Serviço: SDA01MOA2DP012

Service Name: SDA01MOA2DP012 Service RecHandle: 0x10001 Service Class ID List: ”Audio Sink”(0x110b) Protocol Descriptor List: ”L2CAP”(0x0100) PSM: 25 ”AVDTP”(0x0019) uint16: 0x0102 Profile Descriptor List: ”Advanced Audio”(0x110d) Version: 0x0102

Serviço: SDA01MOHFP016

Service Name: SDA01MOHFP016 Service RecHandle: 0x10002 Service Class ID List: ”Headset”(0x1108) ”Generic Audio”(0x1203) (0x1131) Protocol Descriptor List: ”L2CAP”(0x0100) ”RFCOMM”(0x0003) Channel: 2 Profile Descriptor List: ”Headset”(0x1108) Version: 0x0102

Serviço: SDA01MOHFP016

Service Name: SDA01MOHFP016 Service RecHandle: 0x10003 Service Class ID List: ”Handsfree”(0x111e) ”Generic Audio”(0x1203) Protocol Descriptor List: ”L2CAP”(0x0100) ”RFCOMM”(0x0003) Channel: 3 Profile Descriptor List: ”Handsfree”(0x111e) Version: 0x0106

Serviço: OBEX Phonebook Access Server

Service Name: OBEX Phonebook Access Server Service RecHandle: 0x10005 Service Class ID List: ”Phonebook Access - PSE”(0x112f) Protocol Descriptor List: ”L2CAP”(0x0100) ”RFCOMM”(0x0003) Channel: 19 ”OBEX”(0x0008) Profile Descriptor List: ”Phonebook Access”(0x1130) Version: 0x0101

A viatura C exibiu uma permeabilidade aos ataques Bluetooth revelando a sua identidade, os serviços que oferece bem como a sua localização em determinado espaço, recorrendo à calculadora de CVSS, temos a classificação de 4.3 para estas vulnerabilidades da viatura C.

Exploração da porta OBD-II

Tabela 4.9: Vulnerabilidades OBD-II da viatura C

Exploração OBD-II	Valor (S / N)
CANbus vulnerável?	S
Intercepção de pacotes?	S
Injecção de pacotes?	S

Foi feita a localização da porta OBD-II e captura de pacotes CAN recorrendo ao cansniffer.

Através da análise do tráfego do CANbus foi possível isolar o pacote que liga os piscas que tem o valor de 348#001A2249C1210000, podemos então ligar os piscas remotamente através da injecção no CANbus usando o seguinte comando:

```
$ cansend can0348#001A2249C1210000
```

Foi explorado com sucesso a captura e injecção de pacotes no CANbus nesta viatura, de acordo com a calculadora CVSS temos a classificação de: 7.5

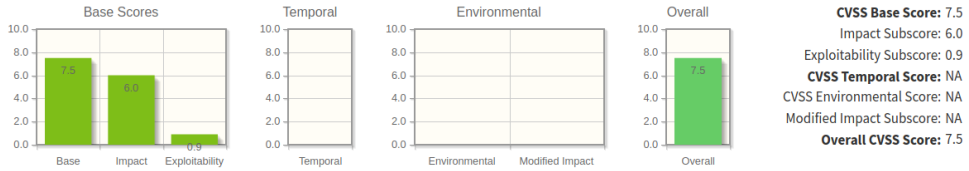

Figura 4.5: Classificação CVSS da exploração OBD-II da viatura C

Capítulo 5

Conclusões

A indústria automóvel vive dias difíceis no que diz respeito à cibersegurança, a complexidade dos sistemas aliado à necessidade da compatibilidade com sistemas mais antigos torna este ecossistema difícil de gerir os protocolos mais antigos e que se tornou norma o CANbus, está presente em todas as viaturas modernas o que apresenta um risco elevado, dado que é um protocolo de comunicação não seguro e que facilmente pode ser manipulado não existindo nenhum mecanismo de segurança.

Relativamente às outras superfícies de ataque exploradas nesta dissertação, e dentro das amostras recolhidas, são notórias as vulnerabilidades herdadas de tecnologias inseguras como, por exemplo, a adopção de cifras WPA nas redes WiFi disponibilizadas pelas viaturas torna possível a revelação da palavra passe, devido a uma vulnerabilidade bem conhecida. Por outro lado, a impossibilidade de alterar o MAC ADDRESS da placa WiFi e Bluetooth, permite que os passageiros da viatura sejam seguidos e localizados.

Mesmo em viaturas recentes, do ano 2020, foi possível manipular o CANbus e extrair informação do Bluetooth. Seria expectável que os fabricantes tivessem outro tipo de implementações e dotassem as suas viaturas de medidas efectivas que aumentassem o nível de segurança. A manipulação do CANbus foi explorada com sucesso numa viatura de 2005, ou seja, em 15 anos existiram poucos progressos nesta matéria.

É importante frisar que todas as viaturas apresentaram vulnerabilidades nas superfícies de ataque definidas no modelo de exploração. Em todos os veículos foi possível explorar o CANbus. Na viatura B e C os ataques ao Bluetooth tiveram sucesso, na viatura B foi, ainda, possível explorar a tecnologia WiFi.

Existem, também, falta de mecanismos eficientes para actualizar os sistemas de um automóvel, bem como a notificação dos seus proprietários de que é necessário fazer a actualização.

Concluo que é necessário incluir nas equipas de desenvolvimento automóvel, especialistas de

cibersegurança, bem como definir programas de *bug bounty* por parte dos fabricantes para que as viaturas sejam, constantemente, testadas e que acima de tudo o ciclo de produção e disponibilização de correcções sejam céleres, para minimizar a janela de oportunidade dos atacantes.

5.1 Trabalho Futuro

Foi evidenciado nesta dissertação a fragilidade do sistema CANbus, pelo que um trabalho futuro a desenvolver nesta área, seria a implementação de um sistema de cifra de comunicações no CANbus, bem como a detecção e correcção de possíveis anomalias. Este sistema deixaria de permitir a intercepção de mensagens e asseguraria a validação do emissor e destinatário da mensagem. O grande desafio será, sem duvida, a necessidade de retrocompatibilidade com uma série de componentes e a manutenção da velocidade de transmissão dentro no CANbus das mensagens cifradas.

Mais trabalhos futuros na área da exploração automóvel poderão, também, passar pela concepção de sistemas automáticos de detecção de ataques. Esses sistemas assentes em inteligência artificial e numa rede de partilha de dados entre viaturas, iriam permitir modelos de defesa automóvel em tempo real.

Bibliografia

[1] M. Tutor. Ferdinand verbiest. URL https://mathshistory.st-andrews.ac.uk/Biographies/Verbiest/. Acedido em 2020-03-01.

[2] Who invented the first car? 6 different inventors might have., 2015. URL https://www.buddsautocredit.com/2015/12/22/who-invented-the-first-car-6-different-inventors-might-have/.

[3] S. Parissien. *The life of the automobile : the complete history of the motor car*. Thomas Dunne Books, St. Martin's Press, New York, 2014. ISBN 978-1250040633.

[4] J. Cook, J.A. & Freudenberg. Controller area network (can)., 2008. URL https://www.eecs.umich.edu/courses/eecs461/doc/CAN_notes.pdf. Acedido em 2020-10-27.

[5] CIA. Mercedes w140: First car with can. URL https://can-newsletter.org/engineering/applications/160322_25th-anniversary-mercedes-w140-first-car-with-can/. Acedido em 2020-08-01.

[6] P. Europeu. Directiva 98/69/ce do parlamento europeu. URL https://eur-lex.europa.eu/legal-content/PT/ALL/?uri=CELEX%3A31998L0069. Acedido em 2020-08-01.

[7] K. D. Solutions. Practical tips: Can-bus. URL https://www.kmpdrivetrain.com/paddleshift/practical-tips-can-bus/. Acedido em 2020-08-01.

[8] S. Fassak, Y. Idrissi, N. Zahid, and M. Jedra. A secure protocol for session keys establishment between ecus in the can bus. pages 1–6, 11 2017. doi: 10.1109/WINCOM.2017.8238149.

[9] Axiomatic. Q&a – what is can? URL https://www.axiomatic.com/whatiscan.pdf. Acedido em 2020-08-01.

[10] A. Alshammari, M. Zohdy, D. Debnath, and G. Corser. Classification approach for intrusion detection in vehicle systems. *Wireless Engineering and Technology*, 09:79–94, 01 2018. doi: 10.4236/wet.2018.94007.

[11] C. Smith. *The car hacker's handbook : a guide for the penetration tester*. No Starch Press, San Francisco, CA, 2016. ISBN 978-1-59327-703-1.

[12] A. Guedes. Can bus barramento controller area network “conceituaÇÃo”. URL http://www.alexag.com.br/CAN_Bus_Parte_2.html. Acedido em 2020-12-12.

[13] Components101. Odb-ii connector. URL https://components101.com/connectors/obd2. Acedido em 2020-12-01.

[14] Infortronica. Guia geral – códigos de avaria. URL https://infortronica.pt/guia-geral-codigos-de-avaria/. Acedido em 2020-08-01.

[15] D. Rimpas, A. Papadakis, and M. Samarakou. Obd-ii sensor diagnostics for monitoring vehicle operation and consumption. *Energy Reports*, 6:55 – 63, 2020. ISSN 2352-4847. doi: https://doi.org/10.1016/j.egyr.2019.10.018. URL http://www.sciencedirect.com/science/article/pii/S2352484719308649. Technologies and Materials for Renewable Energy, Environment and Sustainability.

[16] M. Freitag. Volkswagen kämpft mit massiven softwareproblemen beim id.3. URL https://www.manager-magazin.de/unternehmen/autoindustrie/volkswagen-ag-elektroauto-id-3-mit-massiven-softwareproblemen-a-1301896.html. Acedido em 2020-07-01.

[17] D. C. Lévy-Bencheton. The road to secure smart cars: Enisa approach. URL https://www.troopers.de/media/filer_public/89/05/890580f6-f723-4bc1-9ea1-54ca3b8639ec/enisa_-_troopers16_-_17-03-2016.pdf. Acedido em 2020-08-01.

[18] Kaspersky. O que é cibersegurança? URL https://www.kaspersky.com.br/resource-center/definitions/what-is-cyber-security. Acedido em 2020-12-01.

[19] A. S. Daniel Adinolfi. Cve ids and how to get them. URL https://cve.mitre.org/CVEIDsAndHowToGetThem.pdf. Acedido em 2020-12-01.

[20] CFIRST. Common vulnerability scoring system v3.0: Specification document. URL https://www.first.org/cvss/v3.0/specification-document. Acedido em 2020-02-01.

[21] H. Attila, P. M. Erdősi, and F. Kiss. *The Common Vulnerability Scoring System (CVSS) generations – usefulness and deficiencies*, pages 137–153. 01 2016. ISBN 978-615-80061-5-6.

[22] B. Security. Cvss explained. URL https://beyondsecurity.com/vulnerability-assessment-requirements-cvss-explained.html?cn-reloaded=1. Acedido em 2020-08-01.

[23] L. Bilge and T. Dumitras. *Before we knew it: An empirical study of zero-day attacks in the real world*. 10 2012. doi: 10.1145/2382196.2382284.

[24] T. Brewster. Apple confirms 1 million reward for anyone who can hack an iphone, 2019. URL https://www.forbes.com/sites/thomasbrewster/2019/08/08/apple-confirms-1-million-reward-for-hackers-who-find-serious-iphone-vulnerabilities. Acedido em 2020-08-01.

[25] C. DETAILS. Bmw : Security vulnerabilities published in 2018. URL https://www.cvedetails.com/vulnerability-list/vendor_id-18059/year-2018/BMW.html. Acedido em 2020-08-01.

[26] C. Miller and C. Valasek. A survey of remote automotive attack surfaces, 2014. URL https://ioactive.com/a-survey-of-remote-automotive-attack-surfaces/. Acedido em 2020-10-27.

[27] R. Orsi. Understanding evil twin ap attacks and how to prevent them, 2018. URL https://www.darkreading.com/attacks-breaches/understanding-evil-twin-ap-attacks-and-how-to-prevent-them-/a/d-id/1333240. Acedido em 2020-08-01.

[28] C. Kamani, D. Bhojani, R. Bhagyoday, V. Parmar, and D. Dave. De-authentication attack on wireless network. *International Journal of Engineering and Advanced Technolog*, 2019. ISSN 2249 –8958. URL https://www.ijeat.org/wp-content/uploads/papers/v8i3S/C11860283S19.pdf.

[29] W. Pattanusorn, I. Nilkhamhang, S. Kittipiyakul, K. Ekkachai, and A. Takahashi. Passenger estimation system using wi-fi probe request. pages 67–72, 03 2016. doi: 10.1109/ICTEmSys.2016.7467124.

[30] L. Oliveira, D. Schneider, J. Souza, and W. Shen. Mobile device detection through wifi probe request analysis. *IEEE Access*, PP:1–1, 06 2019. doi: 10.1109/ACCESS.2019.2925406.

[31] P. Ambavkar, P. Patil, P. Meshram, K. Pamu, and Swamy. *WPA Exploitation In The World Of Wireless Network*, volume Volume 2. 06 2012.

[32] INFOWESTER. Tecnologia bluetooth: o que é e como funciona? URL https://www.infowester.com/bluetooth.php. Acedido em 2020-08-01.

[33] R. Bruno, M. Conti, and E. Gregori. *Bluetooth: Architecture, Protocols and Scheduling Algorithms*, volume 5. 04 2002. doi: 10.1023/A:1013989524865.

[34] B. S. I. Group. Bluetooth protocol architecture, 1999. URL https://www.cs.colorado.edu/~rhan/CSCI_7143_002_Fall_2001/Paper/Bluetooth_Protocol_Architecture.pdf. Acedido em 2020-08-01.

[35] A. Lonzetta, P. Cope, J. Campbell, B. Mohd, and T. Hayajneh. Security vulnerabilities in bluetooth technology as used in iot. *Journal of Sensor and Actuator Networks*, 7:28, 07 2018. doi: 10.3390/jsan7030028.

[36] R. Bali. Bluejacking technology: Overview, key challenges and initial research. *International Journal of Engineering Trends and Technology*, 2013. ISSN 2231-5381. URL http://ijettjournal.org/volume-4/issue-7/IJETT-V4I7P148.pdf. International Journal of Engineering Trends and Technology.

[37] O. Angelopoulou, S. Pourmoafi, A. Jones, and G. Sharma. Killing your device via your usb port. 07 2019.

[38] Marlwarebytes. Tudo sobre ransomware. URL https://pt.malwarebytes.com/ransomware/. Acedido em 2020-08-01.

[39] M. Wolf, R. Lambert, A.-D. Schmidt, and T. Enderle. Wannadrive? feasible attack pathsand effective protection against ransomware in modern vehicles. *International Journal of Engineering and Advanced Technolog*, 2017. URL https://www.escrypt.com/sites/default/files/documents/Ransomware-against-cars.pdf.

[40] Y. Zhang, B. Ge, X. Li, B. Shi, and B. Li. *Controlling a Car Through OBD Injection*. 2016. doi: 10.1109/CSCloud.2016.42.

[41] N. N. Hasan, A. Arif, and U. Pervez. Tire pressure monitoring system with wireless communication. In *2011 24th Canadian Conference on Electrical and Computer Engineering(CCECE)*, pages 000099–000101, 2011. doi: 10.1109/CCECE.2011.6030417.

[42] I. Rouf, R. Miller, H. A. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar. Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. In *USENIX Security Symposium*, 2010.

[43] A. Dubey, D. Vohra, K. Vachhani, and A. Rao. Demonstration of vulnerabilities in gsm security with usrp b200 and open-source penetration tools. 08 2016. doi: 10.13140/RG.2.2.10901.12002.

[44] OWASP. Owasp top 10 - 2017, the ten most critical web application security risks. URL https://wiki.owasp.org/images/0/06/OWASP_Top_10-2017-pt_pt.pdf. Acedido em 2020-11-01.

[45] A. Greenberg. Hackers remotely kill a jeep on the highway—with me in it, 2015. URL https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway. Acedido em 2020-08-01.

[46] S. Kamkar. Drive it like you hacked it, 2015. URL https://samy.pl/defcon2015/2015-defcon.pdf. Acedido em 2020-11-21.

[47] L. Tung. Vw-audi security: Multiple infotainment flaws could give attackers remote access, 2018. URL https://www.zdnet.com/article/vw-audi-security-multiple-infotainment-flaws-could-give-attackers-remote-access/. Acedido em 2020-10-27.

[48] C. Osborne. Zero-day flaw lets hackers tamper with your car through bmw portal, 2016. URL https://www.zdnet.com/article/hackers-can-tamper-with-car-registration-through-bmw-connected-car-portal/. Acedido em 2020-10-27.

[49] C. Cimpanu. Tesla car hacked at pwn2own contest, 2019. URL https://www.zdnet.com/article/tesla-car-hacked-at-pwn2own-contest/. Acedido em 2020-10-27.

[50] A. Greenberg. A bluetooth attack can steal a tesla model x in 90 seconds. URL https://www.wired.co.uk/article/tesla-model-x-hack-bluetooth. Acedido em 2020-12-14.

Apêndice A

Anexo - Evidências da exploração das viaturas

A.1 Viatura A

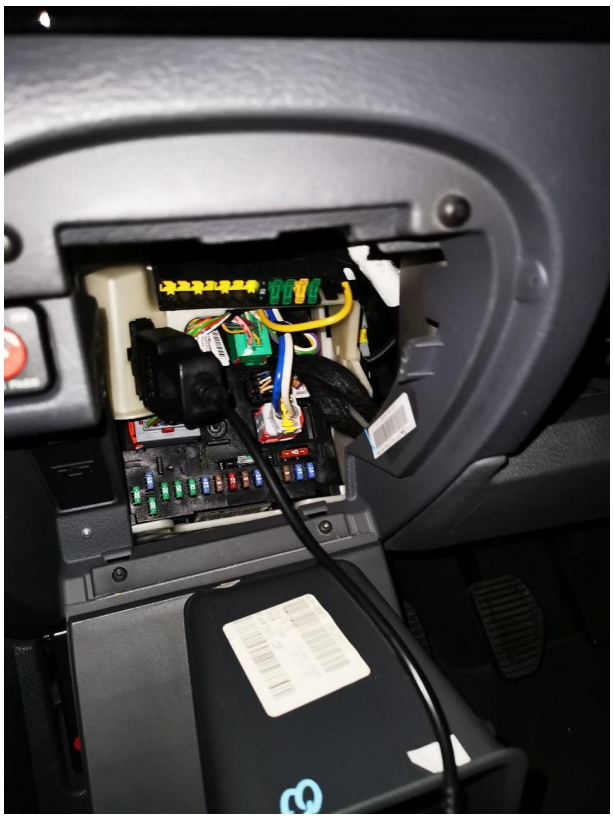

Figura A.1: Localização da porta OBD-II na Viatura A

A.2 Viatura B

Figura A.2: Localização da porta OBD-II na Viatura B

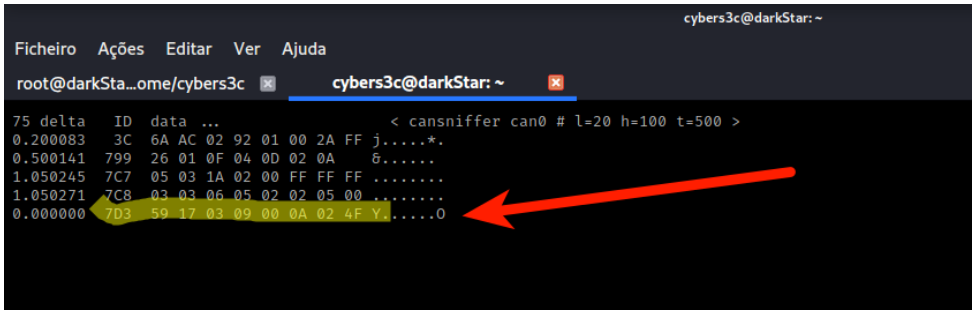

Figura A.3: Identificação da mensagem de ligar os piscas da Viatura B

Figura A.4: WPA crack da Viatura B

A.3 Viatura C

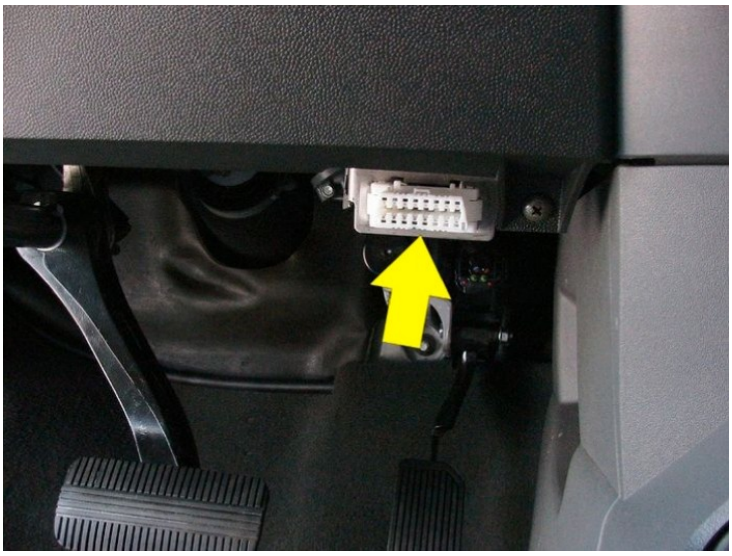

Figura A.5: Localização da porta OBD-II na Viatura C

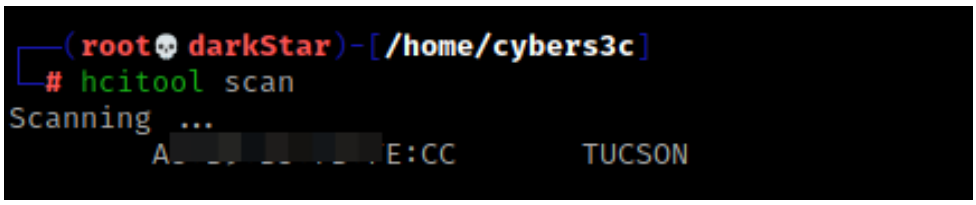

Figura A.6: Identificação do Bluetooth da Viatura C

Figura A.7: Identificação da mensagem de ligar os piscas da Viatura C

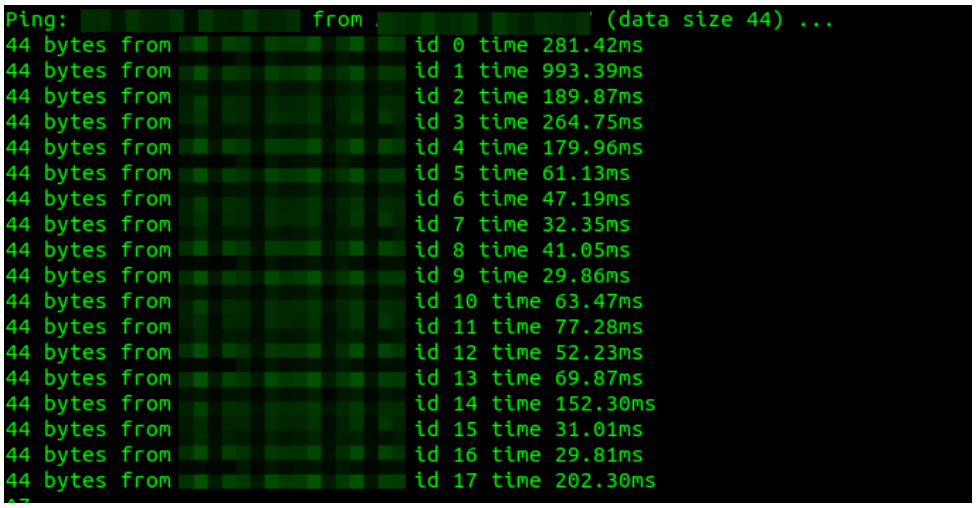

Figura A.8: Localização através de Bluetooth da Viatura C