Broadcasting your attack: Security testing DAB radio in cars

Andy Davis, Research Director

Image: computerworld.com.au

Agenda

- Who am I and why am I interested in security testing DAB?
- Overview of DAB
- How do we broadcast DAB?
- DAB attack surface
- How did we create a DAB security testing tool?
- Demo
- Example vulnerabilities
- Implications of exploitable DAB protocol bugs

Who am I?

- Research Director at NCC Group
- NCC Group is a global cyber security assurance specialist
- Personal interests include wired and wireless interface security, SDR and developing security testing tools – previous examples:
  - `Umap, Frisbee` – USB
  - `CECSTeR, EDIDfuzzer` – HDMI/VGA
  - RFTM - RF Testing Methodology

Why am I interested in DAB?

- Majority of new vehicles are factory fitted with DAB radios
- Often head unit (that contains the DAB radio) has some form of connectivity to the CAN bus, which is in turn connected to cyber-physical systems such as braking
- Doesn’t appear to have received much attention from security research community
- Software Defined Radios getting cheaper

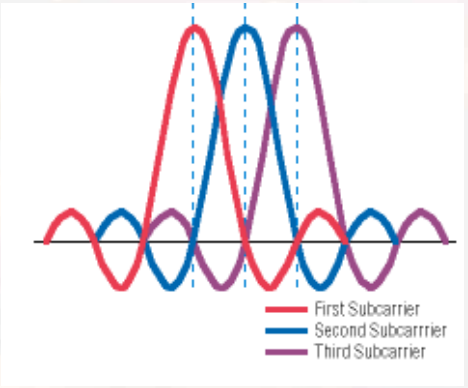
Overview of Digital Audio Broadcasting (DAB)

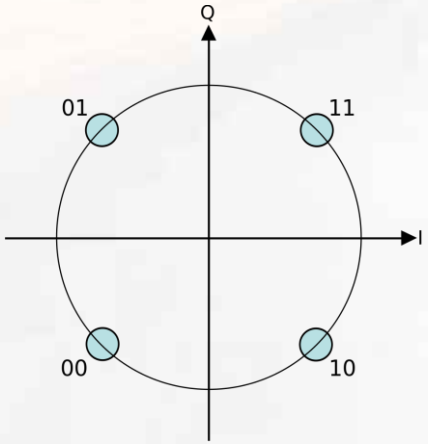
- Digital radio technology for broadcasting radio stations
- Originated as the European Eureka 147 project
- Norwegian Broadcasting Corporation (NRK) launched first DAB channel in June 1995
- Upgraded version called DAB+ released in February 2007
- Benefits over FM are:
    - Better signal reception quality
    - Many more data services can be transmitted
    - Electronic Programme Guide

Image: wikimedia.org

Modulation & Transmission

- Why was DAB developed?
  - Multipath interference
- What is one of the solutions?
  - OFDM
- The maximum number of modulated carriers in the DAB signal is 1536
- Actually COFDM “Coded” OFDM, as Forward Error Correction used
- Modulation scheme is QPSK





Images: wikimedia.org, tenettech.com

Modulation & Transmission

- Audio signals are digitised & multiplexed together with other data to produce a “bit stream”
- Forward error protection then applied by adding redundant bits to the bit stream
- During each consecutive symbol, bits are divided into 1536 pairs
- Each pair is differentially encoded with respect to its counterpart for the previous symbol
- Each of the 1536 differentially encoded bit-pairs are then used to define the phase of a QPSK carrier
- Which together form the spectrum of a 1536-carrier signal
- This is the OFDM generation process, and it is repeated symbol-by-symbol

Image: ak.picdn.net

Multiplexing

- Main Service Channel (MSC) – bulk of the DAB signal
  - Frames of 55296 bits - known as “Common Interleaved Frames” (CIFs)
  - Each CIF divided into time-slots in which logical frames of data for individual services are transmitted
  - Repetitive bursts for each service provide “sub-channels”
  - Data for each CIF transmitted in 18 consecutive symbol-blocks
  - First symbol-block in each transmission frame is used for synchronisation
  - Remaining 3 symbol-blocks at the beginning of the transmission frame are used to carry the Multiplex Configuration Information (MCI), which includes the Fast Information Channel (FIC)
- Ancillary channels – for synchronisation & housekeeping

Image: media.licdn.com

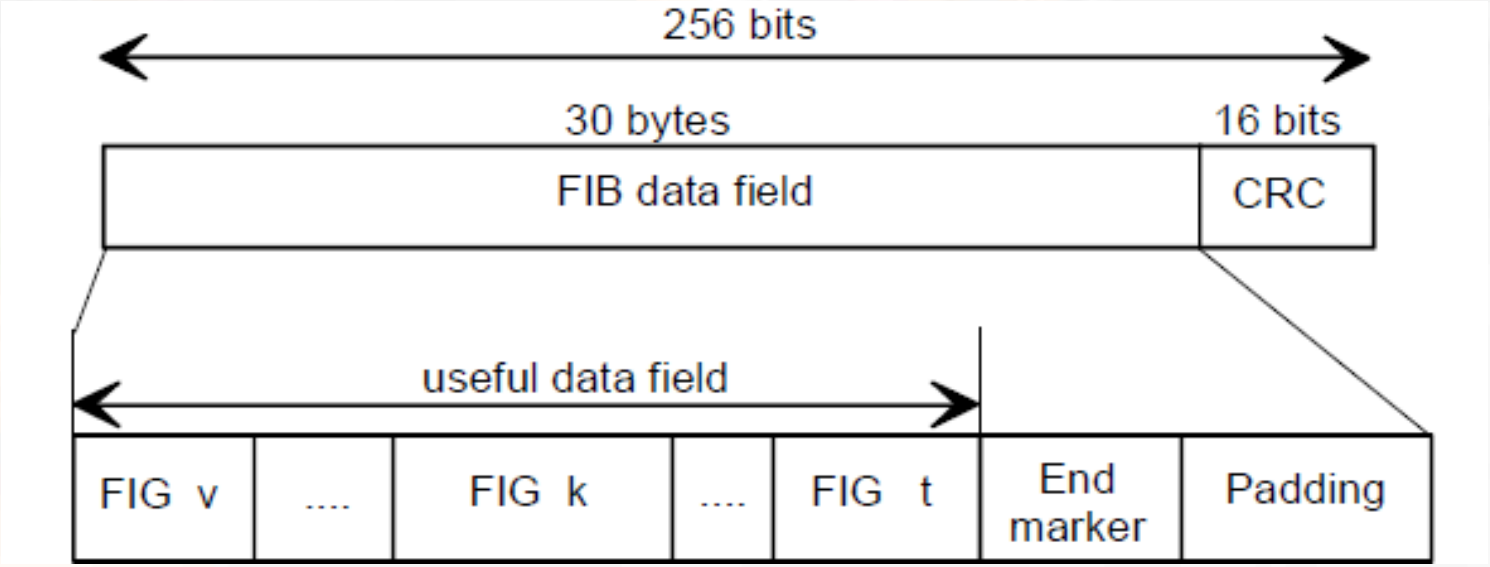
The (ETI) Ensemble Transport Interface

- Standardised output stream from a DAB multiplexer
- 2Mbps synchronous data stream
- Network adaptation is defined for G.703 lines (E1)
- ETI is an ETSI standard: EN 300 799
- ETIsnoop tool available to decode some of the data:
    - http://wiki.opendigitalradio.org/Etisnoop

Image: excellgroup.com

Fast Information Channel (FIC)

- FIC required to make receiver respond rapidly to the user when it is first switched on
- FIC is divided up into Fast Information Blocks (FIBs)
- Each FIB contains a number of Fast Information Groups (FIGs)



Fast Information Groups (FIGs)

- Each FIG is used for a specific signalling purpose:

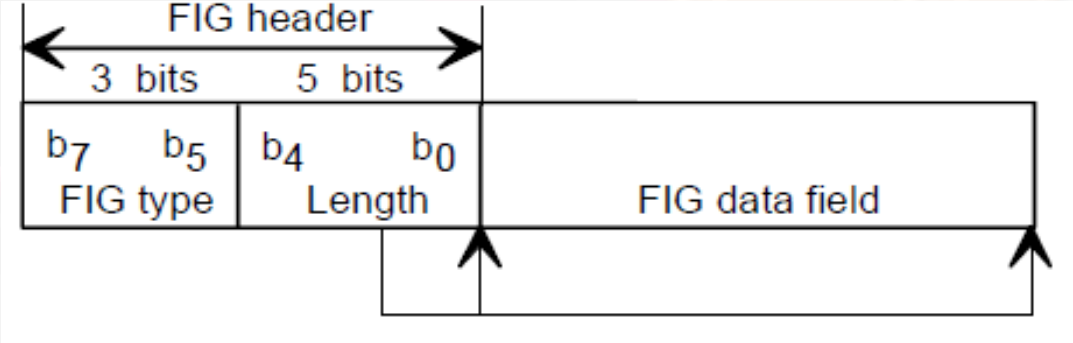
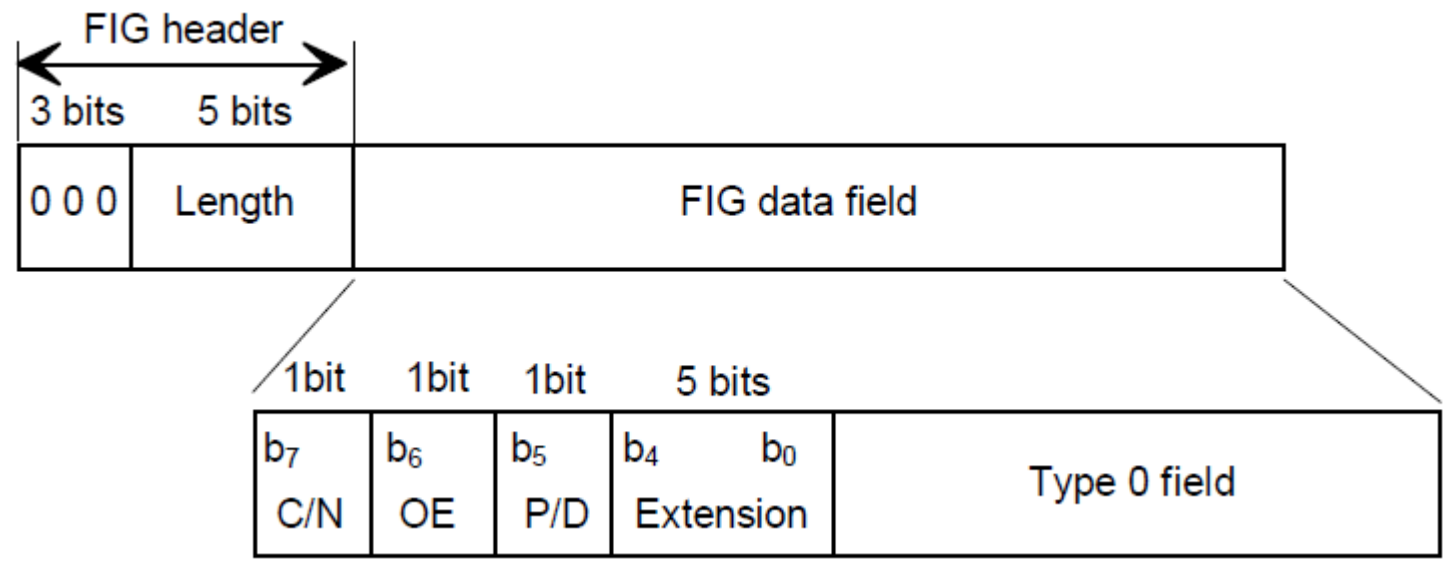


FIG type number	FIG type	FIG application
0	000	MCI and part of the SI
1	001	Labels, etc. (part of the SI)
2	010	Labels, etc. (part of the SI)
3	011	Reserved
4	100	Reserved
5	101	FIC Data Channel (FIDC)
6	110	Conditional Access (CA)
7	111	Reserved (except for Length 31)

FIG data field

- The FIG data field for each FIG type has the following structure:



- Each FIG type has a number of extensions, which provide specific Service Information (SI) configuration functionality

Service Information features - example FIGs

Service Information (SI) features are signalled using extensions of FIG types 0 & 1:

- FIG 0/6 - Service linking information
- FIG 0/13 - User application information
- FIG 0/18 - Announcement support
- FIG 0/21 - Frequency Information
- FIG 0/22 - Transmitter Identification Information (TII) database
- FIG 1/0 – Ensemble label
- FIG 1/5 - Data service label

FIG 0/13 - User application information

- FIG 0/13 signals the type of data sent over DAB – interesting…

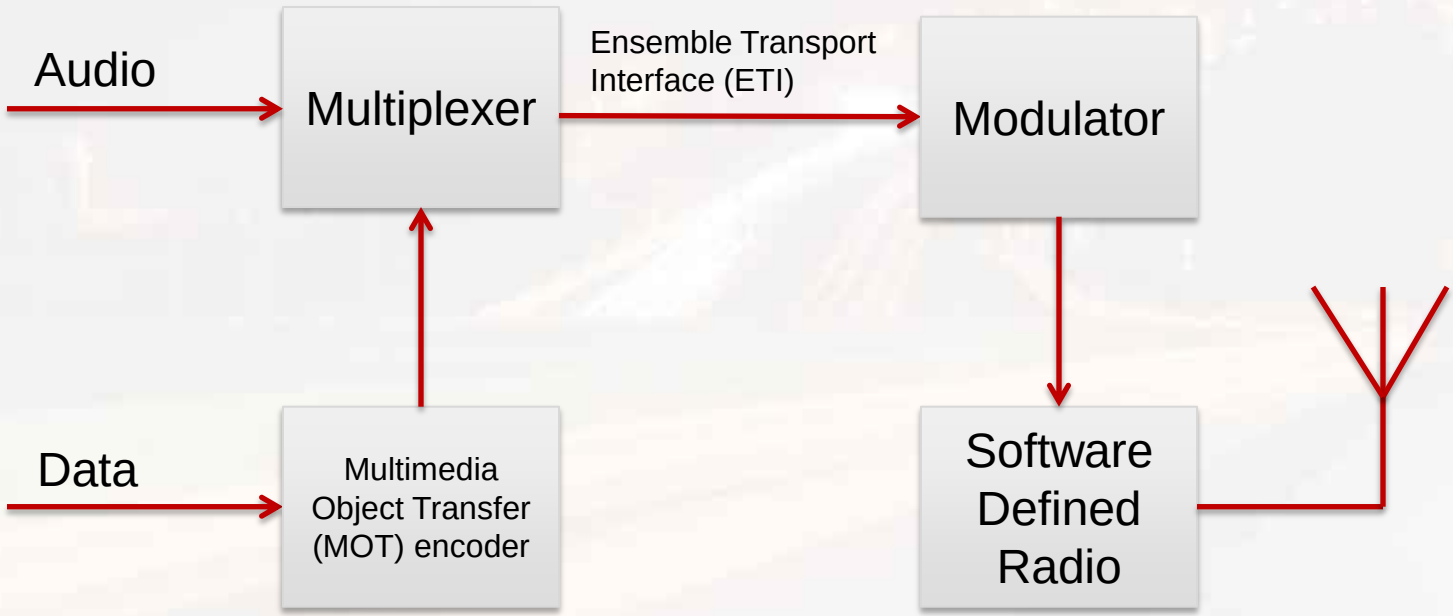
User Application Type (hexadecimal)	User Application	Reference
0x000	Reserved for future definition	
0x001	Not used	
0x002	MOT Slideshow	TS 101 499 [23]
0x003	MOT Broadacst Web Site	TS 101 498 [22]
0x004	TPEG	
0x005	DGPS	
0x006	TMC	TS 102 368 [24]
0x007	EPG	TS 102 818 [25]
0x008	DAB Java	TS 101 993 [26]
0x009 to 0x3ff	Reserved for future definition	
0x400 to 0x449	Reserved for proprietary applications	
0x44a	Journaline®	Fraunhofer IIS
0x44b to 0x7ff	Reserved for proprietary applications	

Programme Associated Data (PAD)

- Each DAB audio frame contains bytes which may carry Programme Associated Data
- PAD is information which is synchronous to the audio
- An example of PAD data is DLS (Dynamic Label Segment) which is often used to display the name of the song playing

Ok, enough of the DAB theory…

Simple DAB transmitter



How do we broadcast DAB?

Here's why we don't need to understand the radio part of the protocol…

- Open source DAB transmitter from http://www.opendigitalradio.org/
  - `odr-dabmux` – allows DAB ensembles to be created
  - `odr-dabmod` – uses DAB modulation schemes for use with an SDR
  - `fdk-aac-dabplus` - includes support for DAB MOT Slideshow & DLS
- USRP B200 SDR
- Legal considerations

Images: www.ettus.com, opendigitalradio.org

DAB attack surface

- The underlying DAB transport protocols & interfaces e.g:
  - FIG data within the ETI (Ensemble Transport Interface)
  - MOT (Multimedia Object Transfer)
- The HMI (Head unit rendering of DLS and DAB labels)
- The media formats that are processed by the receiver e.g:
  - Audio
  - Images
  - Video
- Apps processing Java/IP/raw data

Image: pngimg.com

How did we create a DAB security testing tool?

- The tool `mot-encoder` is bundled with `fdk-aac-dabplus`
- `mot-encoder` enables DLS & slideshow protocols to be added to DAB Program Associated Data (PAD) within an Ensemble
- DLS (text) & slideshow (JPEG/PNG) can then be fuzzed via a FIFO being consumed by `mot-encoder`
- The `mot-encoder` tool was modified to enable an external process (via a TCP socket) to man-in-the-middle the MOT protocol header & data
- The multiplexer `ODR-DabMux` was modified to enable the FIG data to be manipulated (again via a TCP socket)

The DABble fuzzer

- Current DABble capabilities:
  - Fuzz DLS via a FIFO
  - Fuzz JPEG & PNG via a FIFO
  - Fuzz MOT protocol via modified version of mot-encoder
  - Fuzz the Ensemble data via modified version of `ODR-DabMux`
- Planned capabilities:
  - Fuzz the other protocols being sent over DAB (Video/IP/Java etc.)
  - Implement some of the other FIGs that are currently not supported by `ODR-DabMux`

The DABble fuzzer

The DABble fuzzer

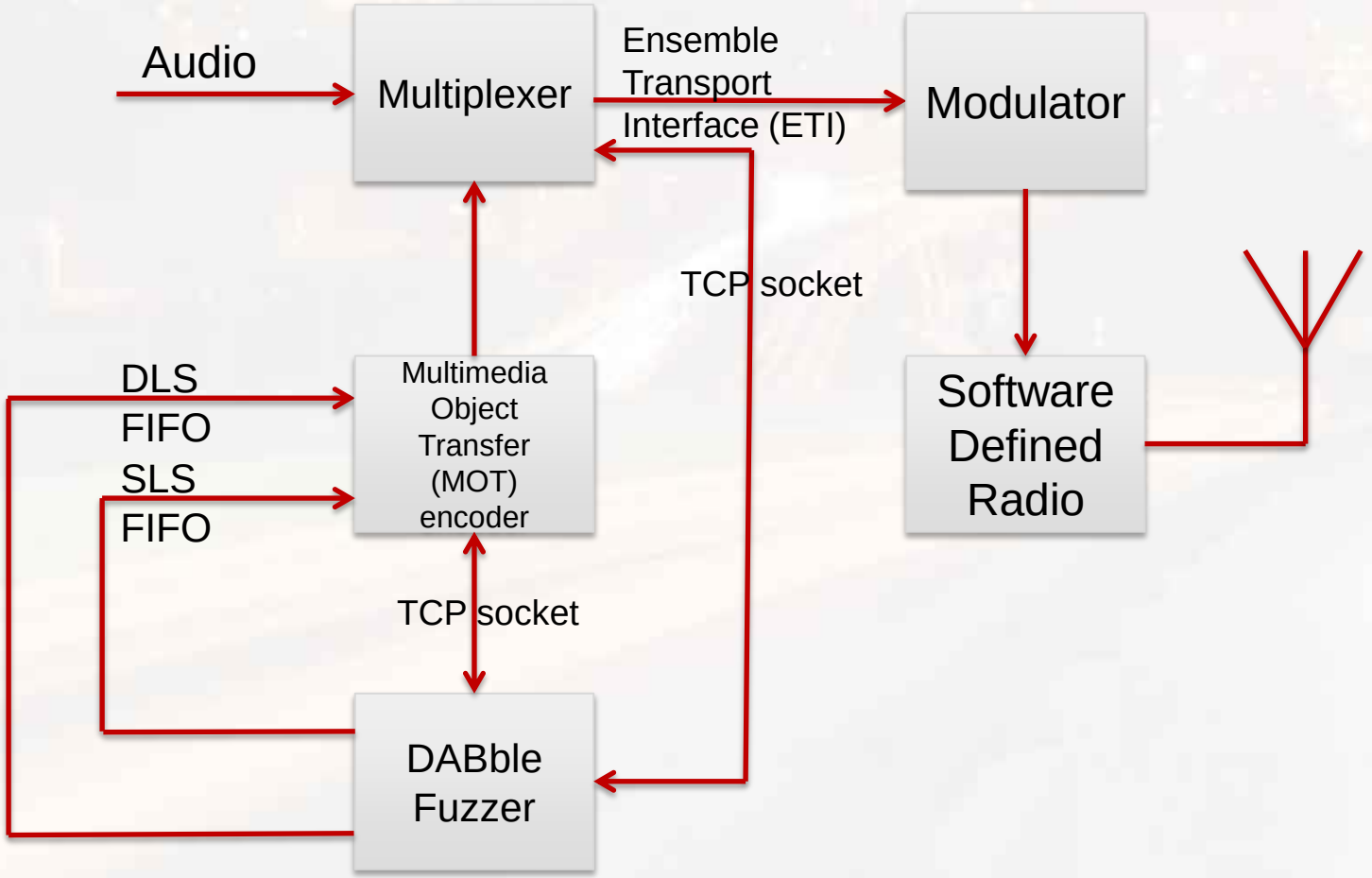


Image: thegapmedia.com

Some example DAB vulnerabilities

FIG 0/13 – MOT Slideshow (SLS)

- JPEGs & PNGs are rendered by the receiver in the vehicle head unit
- Vulnerability in the image parsing library results in code execution

FIG 1/0 – Ensemble label and PAD data

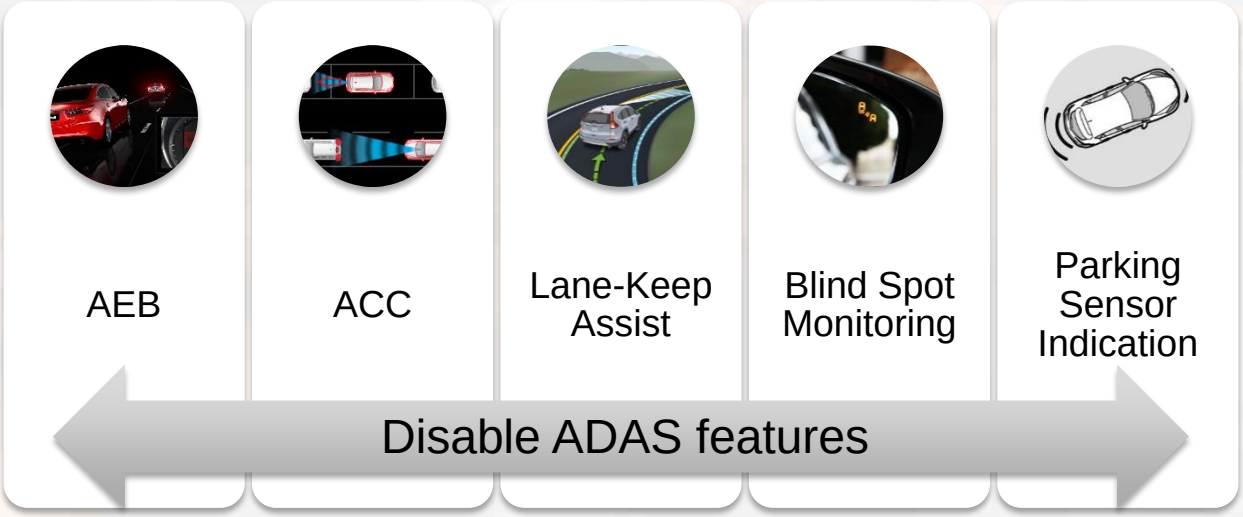
- Ensemble name & DLS information is rendered by the HMI on the head unit & any arbitrary text can be sent.
  - Buffer overflows unlikely, as there is a fixed maximum size
  - Format string bugs possible
  - Ensemble information sometime stored in a local database – SQL injection
  - Head units increasingly connected to the Internet - XSS

Databases of information

- FIG 0/6 - Service linking information
  - Where DAB broadcasts have local services
- FIG 0/22 - Transmitter Identification Information (TII) database
  - The TII database provides a cross-reference between transmitter identifiers & geographic location of the transmitters
- Potential for buffer overflows where fixed size buffers are allocated to store these databases that are downloaded over DAB by the receiver

Implications for other vehicle systems

- System architecture is often insecure:
  - Direct access to CAN bus, or via D-Bus
  - D-Bus bound to all network interfaces
  - D-Bus messages used to directly disable ADAS features



Implications of DAB as a broadcast medium

Multiple vehicles can be attacked simultaneously

Scenario #1

- Attacker uses a high power transmitter to replicate a public DAB ensemble and overpowers the public transmission
    - Major disadvantage: Not stealthy – would likely be spotted quickly

Scenario #2

- Attacker uses a low power transmitter and creates a new DAB ensemble on an unused local frequency
    - Most DAB receivers constantly re-tune
    - Attacker chooses station name to entice target audience

Conclusions

- DAB is an obvious remote attack route into a vehicle
- A single attack could be broadcast to many targets
- There are many protocols that can be transmitted over DAB, which could be attacked
- The core DAB protocols e.g. ETI & MOT can also be attacked
- How many DAB radio developers have assumed that the broadcast data is trusted?

Further reading

- DAB specification:
  http://www.etsi.org/deliver/etsi_en/300400_300499/300401/01.04.01_40/en_300401v010401o.pdf
- MOT specification:
  http://www.etsi.org/deliver/etsi_en/301200_301299/301234/02.01.01_40/en_301234v020101o.pdf
- ETI specification:
  http://www.etsi.org/deliver/etsi_i_ets/300700_300799/300799/01_30_9733/ets_300799e01v.pdf

Questions?

Andy Davis
andy.davis@nccgroup.trust