CSL Dualcom CS2300-R security analysis

Executive Summary

Motivation

I am a reverse engineer and ethical hacker specialising in embedded systems and web/mobile applications. After work on wired and wireless alarm systems, several installers and ARC operators wanted me to investigate signalling devices and systems. These systems seem to have largely escaped the attention of the wider IT security community before now.

Local attacks on wired and wireless alarm systems are interesting, but attacking signalling systems remotely is a far more credible and likely threat.

Most outside interest was in the CSL Dualcom product range. CSL Dualcom signalling devices are extremely common place but are also one of the cheapest signalling products on the market.

I embarked on reverse engineering the system to find vulnerabilities, exploits and bugs. I expected to find some small issues, but instead found a worryingly large number of fairly serious issues.

Goal

The goal of the work is to find issues and vulnerabilities that would allow a malicious attacker to enter a premises without the alarm receiving centre (ARC) taking appropriate action such as calling a key holder or alerting security/police.

This includes, but is not limited to:

- Preventing genuine alarms from ever being sent.
- Spoofing status reports so that the ARC believes the alarm is OK.
- Spoofing alarm reports for an individual alarm so that the signalling path is manually disabled (the “boy who cried wolf” situation).
- Spoofing alarm reports for so many alarms that it is impossible to discern genuine alarms from spoofed (a denial-of-service attack).
- Exploiting vulnerabilities in the polling servers such that they stop responding entirely.

Scope

The devices (secure premises transmitters or SPTs) are all CSL Dualcom GPRS CS2300-R boards, with firmware versions ranging from v1.25 to v3.53. The v3.53 boards were upgraded from earlier versions using downloaded firmware from the CSL website.

13 SPTs were provided after being removed from existing installations. They arrived at my lab in late 2013 and my work commenced in early 2014. They were out of use for at least 6 months, possibly as long as 2-3years. There were no records of install date, installer name or otherwise available. These may or may not be representative of the current CSL Dualcom boards in use.

The SPTs have been tested in a “black-box” fashion i.e. no co-operation, source code, or internal documentation has been provided by CSL. This is in contrast to “white-box” testing where CSL cooperation would be provided, including access to source code for the SPT and servers, documentation, and allowing connection to test servers.

This has somewhat limited the scope of the work, for a number of reasons:

- Only passive monitoring of communications between the SPT and CSL polling servers has been carried out. Active spoofing, fuzzing and other aggressive/offensive techniques could be used with CSL's cooperation.
- Source code analysis is far quicker than disassembled code analysis and gives important hints to code quality, development practices and other important factors.
- Analysis of the server side code would open up a massive additional part of the attack surface.
- No active testing of proposed vulnerabilities has been carried out.

Live boards were observed communicating on both the GPRS and IP path. GPRS communications were sniffed on the serial side of the GPRS modem so that no GPRS reception/sniffing was required. This is because it is difficult to sniff GPRS communications without unintentionally observing third-party communication. IP communications were sniffed on the serial side, but also on the IP side using a mirror port on a switch. No PSTN communications were sniffed, but operation has been inferred from disassembled code.

Disclosure Timeline

- March 2014 - begin live blogging reverse engineering of CS2300-R
- May 2014 - Rob Evans of CSL Dualcom contacts me to request removal of blog posts and social media posts. I comply.
- 19 June 2014 - issues of weak encryption and SMS remote control raised with CSL in person at IFSEC 2015.
- 30 June 2014 - a Python implementation of the encryption is provided to Santosh Chandorkar as a proof of concept.
- Various communications with CSL but they are unwilling to discuss without an NDA.
- 19 April 2015 - full PDF report is sent to CSL
- 02 October 2015 - report is sent to CERT for disclosure
- 23 November 2015 - report is disclosed via CERT

Method

Board description

All SPTs are marked CSL Dualcom GPRS with a part number of CS2300-R.

All SPTs still contained SIM cards. These are all Vodafone M2M SIMs.

Each board comprises of several major components:

- An NEC/Renesas 78K0R 16bit microcontroller (specific part number µPD78F1154) with 128KB of flash ROM and 8KB of RAM.
- A socketed GPRS modem board (either a Wavecom GR64 or Q64). The firmware supports Cinterion, Sierra and Telit GPRS modems, but none of these were found on boards.
- A socketed SIM card is on the main board and is connected to the GPRS modem.
- An RS485 transceiver for connection to alarm panels.
- An optional LAN card connected by two socket headers on the board.
- A PSTN modem operating at 2400bps (v22bis)
- Two serial EEPROMS of 16K in size. One is socketed (the “NVRAM”) and stores settings. The other is soldered to the board.
- A switch mode power supply.
- Two 7-segment LED displays
- Four status LEDs.
- Two buttons.
- A reset button.
- Terminals for I/O.

There are no tamper resistant features evident, the security of the SPT must be ensured by placing the device in a secure area.

Reverse engineering

A number of techniques and processes have been used:

- Downloading the firmware from the CSL website, then disassembling in Renesas Cubesuite and IAR Studio. This allows the function of the software to be inferred from assembly code, but doesn’t provide any high-level code such as C or C++.
- Running the firmware in the simulator in Renesas Cubesuite and IAR Studio. This allows code to be stepped through. Without connection to external I/O (e.g. GPRS modem and EEPROM) this can only be of limited utility.
- Logic tracing live boards, including serial communication on the GPRS, IP and PSTN path. This involves connecting a logic analyser to the boards and sampling signals periodically. These signals can then by analysed as asynchronous serial, I2C, or SPI, and then reassembled at higher levels.
- Reading out entire EEPROM content. This allows the examination of settings and other data in the EEPROM. By mapping known variables in the EEPROM (e.g. the ICCID is stored at address 0x130), the functionality of areas of assembly can be inferred.
- Logic tracing EEPROM communications. This allows us to see where and when the board accesses data on the SPT. This can provide key information e.g. why is the ICCID read from the EEPROM immediately before any encrypted message?
- Reading SIM data. This allows us to see any old SMS messages received by the SPT.
- Inspecting and reverse engineering Windows utilities.
- Sniffing TCP/IP connections on the LAN path between SPT and server. This was achieved using a mirror port on a switch, and allows any communication on the LAN to be sniffed.
- Google site searches of CSL websites. This often reveals documentation and other parts of a site that are hidden by obscurity.
- Speaking with installers and ARC operators.

- Emulating CSL servers using Python scripts.

Whilst these techniques are specialist and not commonplace, there are many individuals and organisations capable of this level of work. Specifically, no advanced knowledge of cryptography or cryptanalysis was required.

The equipment used included:

- A laptop PC
- A USB 8 channel logic analyser (used to trace electronic signals on the boards) ~£100
- A Bus Pirate USB multi-tool (used to read out EEPROM) ~£30
- An Arduino Mega (used to emulate the GPRS modem) ~£30
- A SIM card reader ~£15
- An ethernet switch with a monitor port (used to sniff network traffic) ~£30

This equipment is all part of my normal lab equipment. The only additional piece of equipment required was a Renasas MiniCube2 in-circuit programmer/debugger at ~£110. No software has been purchased - either open source or free versions have sufficed.

No assistance has been given from CSL Dualcom outside of their support email answering a few minor questions.

The biggest hurdle has been learning to understand and interpret 78K0R assembly.

Issues

These issues are presented in no particular order.

To work within the stated scope of the testing, there has been no practical exploitation made of any of these vulnerabilities.

No identity verification of polling server to the SPT

Summary

There appears to be no identity verification between polling server and SPT.

Detail

Communications observed on the GPRS and IP paths use a very simple protocol that does not verify the identity of the polling server (PS) to the SPT.

```
SPT:        DC4 (“Dualcom 4” protocol version)
PS:         HS58 (the “starting variable”)
SPT:        0x01|ZSh8L2DaRXVWmn6GOqP3j8JYCOgFSekXmUY15hgGQLdH (encrypted
            message)
PS:         0x16 (encrypted message valid)
PS:         0x06
SPT:        0x03
```

The SPT sends *DC4* as ASCII and the polling server responds *HSxx* (where xx is a number in the range of 50-98). Any server can respond with *HSxx*, so the SPT cannot be sure of the identity of the server.

The SPT verifies it’s own identity to the polling server by encrypting a message including the ICCID (SIM card identification number, machine printed on the SIM and the CS2300-R) and “chip number” (a number programmed into the NVRAM of the CS2300-R and often hand-written on the CS2300-R).

From this point onwards, this protocol has been termed the “DC4 protocol”.

This would be termed unilateral authentication, as opposed mutual authentication where both parties confirm their identity with each other.

Potential vulnerability

A protocol which does not validate the identity of one or both of the parties can easily be targeted by several attacks.

It would be trivial to impersonate the polling server, which would lead to the SPT believing that it is communicating with the genuine server.

A lack of mutual authentication greatly facilitates man-in-the-middle (MITM) attacks. This is where a malicious party sits between the SPT and the polling server and can read, intercept, block and modify communications. For example, normal polling messages could be passed and alarm messages blocked.

There is potential for this to occur on all three paths - GPRS, IP and PSTN.

On the IP path, it would be almost trivial to execute a MITM attack. A number of mechanisms could be used to redirect all IP traffic through an malicious host:

- ARP spoofing - a machine on the local network segment can intercept all traffic by claiming to “own” all IP addresses.
- Rogue DHCP server - the SPT is supplied false details for the local gateway, allowing a local machine to intercept all traffic.
- Router compromise - many routers for both business and home use have multiple vulnerabilities allowing administrative access, allowing traffic to be re-routed and sniffed.
- Reconfiguring the SPT to use a different gateway or server using SMS remote commands (detailed later).

Whilst several of these require a compromised machine on the local network segment, this can be achieved by a number of means, such as social engineering, direct attack, or placing a remote KVM (keyboard, video, mouse) onsite.

A Python DC4 protocol proxy has been written which allows messages on the IP path to be decrypted and either blocked or modified depending on their content.

On the PSTN path, it is possible to splice into the telephone line and insert a device comprising of a line simulator and two modems, one answering calls from the SPT, and the other dialling the polling server. Data can then be forwarded from one modem to the other.

The GPRS path is more challenging. Since 2001 it has been known that the GSM specification does not authenticate the base station (i.e. the cell site and network) to the mobile station (i.e. the SPT). This allows for fake base stations (often called IMSI catchers) to intercept traffic between a mobile station and the network. Since 2011, the same attacks have been possible with GPRS as well.

10 years ago, attacks such as this would have been classified as government level. Software Defined Radio (SDR) base stations such as the USRP and nano-cells have reduced the costs and complexity greatly, and brought GPRS MITM attacks down to the level of a skilled attacker.

It is currently considered unwise to rely solely on the security of the GSM/GPRS network.

Remediation

It is essential that secure protocols perform mutual authentication to mitigate against many attacks. Mutual authentication doesn’t entirely remove the risk but is a big improvement compared to unilateral authentication.

A secure protocol should not rely on the security of the underlying communication channel such as GSM/GPRS and should be able to withstand attack independently.

IP and PSTN paths can be sniffed trivially

Summary

The IP and PSTN paths can be sniffed using easily available equipment. Combined with the weak encryption, this results in poor confidentiality.

Detail

The IP path can be trivially sniffed by a number of means. The simplest of these is by connecting the SPT to a network switch with a mirror port, which forwards all the traffic onto a machine running software such as Wireshark. This is an extremely common method used in diagnosing network issues.

There are a number of other methods that can be used to redirect traffic through a malicious machine on the local network segment. These do not require special equipment or physical reconfiguration of the network.

The PSTN path uses v22bis (2400bps) modem signalling. Sniffing this traffic can be achieved with a €500 capture device that uses the audio input on a soundcard. If the traffic were any faster than v22bis, passive sniffing becomes challenging and there appear to be no readily available commercial solutions.

As described above, the PSTN path can be spliced into and a line simulator and two modems used to forward the traffic.

Potential vulnerability

Relying on the inherent security of either the IP or PSTN path is unwise as the communications can easily be sniffed. Combined with the weak encryption, this results in poor confidentiality and security of the messages.

Remediation

It would be wise to advise installers to use a virtual LAN (VLAN) to segregate the SPT from local machines. This would mean that several of the attacks that allow traffic to be redirected would not work.

The encryption used needs to be stronger if the communication channel can be sniffed. This would require ground-up redesign of the encryption.

Key used in encryption is constant across all devices

Summary

The key used to encrypt messages is constant across all devices and cannot be changed.

Detail

The simple encryption algorithm used requires a mapping table ~48bytes in length.

This mapping table is stored in flash memory and is part of the firmware. It appears to be constant across all devices studied from v1.25 (the earliest SPT observed) to v3.53 (the latest firmware that could be downloaded from CSL's website).

Initially the mapping table was derived by disassembly of the v3.53 downloaded firmware image. This was then repeated with v3.10. Both images have the same mapping table.

It has not been possible to recover the firmware from SPTs, so direct observation of the mapping table from flash has not been possible for earlier versions of firmware. Regardless, the basic nature of the encryption means that it has been possible to confirm that the same mapping table is in use just by observing communications.

This was checked by decrypting messages to yield the already known ICCID and chip number.

Preventing the firmware from being downloaded would make direct recovery of the mapping table far harder.

However, as described below, unless the encryption algorithm changes, it is possible to determine the mapping table from observed communications.

This aspect of the encryption is essentially constant and known so cannot reasonably be viewed as increasing the security of the protocol.

There appears to be no mechanism to change the mapping table without flashing the firmware, and there is no means to flash the firmware remotely.

Even if newer devices used a different mapping table, capturing one device of the same version (by purchasing, stealing or borrowing one) would yield the new mapping table to an attacker.

Potential vulnerability

As this part of the encryption is constant, it is trivially easy to encrypt and decrypt data.

Remediation

Key material must vary from device to device and there must be a means of changing or rotating keys.

Messages can be decrypted without knowing starting variable

Summary

The polling server starts each communication with a starting variable ranging from HS50 to HS98. This does not need to be known to decrypt messages.

Detail

An example of an observed communication is shown below:

```
SPT:        DC4 (“Dualcom 4” protocol version)
PS:         HS58 (the “starting variable”)
SPT:        0x01|ZSh8L2DaRXVWmn6GOqP3j8JYCOgFSekXmUY15hgGQLdH (encrypted
            message)
PS:         0x16 (encrypted message valid)
PS:         0x06
SPT:        0x03
```

It can be seen by observing further communications and disassembled code that the polling server sends a value in the range of *HS52* to *HS98* to the SPT. This is used as the index to start at in the mapping table for the polyaphabetic substitution cipher. In many encryption algorithms this would be called a starting variable or initialisation vector.

Generally the purpose of a starting variable is to ensure that if the same data is sent twice, it will still end up being encrypted to a different ciphertext. A starting variable needs to be long enough that it is highly unlikely that two messages will encrypt to the same cipher text.

Unfortunately, the use of a simple substitution cipher, combined with a partially known plaintext and known mapping table, means that we don’t need to know the starting variable to decrypt a message.

The message sent always begins with the ICCID of the SIM card. For Vodafone M2M SIM cards, these always start with *894410* (*89* = telecoms, *44* = UK, *10* = Vodafone). This means that we have a partially known plaintext.

As described elsewhere, the mapping table is already known and cannot be changed.

Finally, a simple, single character substitution cipher means that we can quickly test the encrypted string character by character until we known the starting variable. This is compared to a good block cipher where a small change in the input plaintext should cause a large change in the output ciphertext.

A single character substitution cipher behaves as follows.

```
Plaintext:      Hello World!
Ciphertext:     AHbs*hsbd1fF

Plaintext:      Hello world!
Ciphertext:     AHbs*hbbd1fF
```

A good block cipher behaves as follows:

```
Plaintext:      Hello World!
Ciphertext:     ji8ywbuwhg76

Plaintext:      Hello world!
Ciphertext:     yhdu*u8f678g
```

To illustrate the difference, it is analogous to handling a 4-digit pin in two different ways:

1. The user enters the entire pin and is told if it is right or wrong. There are 10,000 choices and only one is right - brute forcing this is hard and time consuming.
2. The user enters a single digit and is told, digit by digit, if it is right or wrong. In this situation there are a mere 40 choices and brute forcing is trivial.

A simple example of the issue with actual observed communications:

```
Encrypted:   iX1ZRjh7ILpU1hcIZEPmOOahWmVUX1dcCMH8B7j02SQ
Known chars: 894410              A                    7
Derived SV:  HS88
Decrypted:   894410123456789012 37A12345611111111111117
```

The only starting variable that encrypts *8* to *i* is *HS88.* We can now decrypt the message using the starting variable, giving us the ICCID and chip number. This can then be leveraged in other attacks.

There are some mapping tables that would encrypt more than a single character to i. We would then have to check additional known characters to derive the starting variable. It is possible, although highly unlikely, that we would exhaust the known characters.

However, the mapping table found in the firmware never requires more than a check for the first *8* in the string.

In many situations, observed communications would show both sides of the conversation, which means that the starting variable would be known anyway. However, some IP MITM attacks only allow observation of outgoing traffic from the SPT, which is where being able to derive the starting variable would be useful.

Potential vulnerability

The starting variable sent from the polling server to the board is not required to decrypt messages. The starting variable provides little additional security.

Remediation

If a starting variable is used by an encryption algorithm, it must be of sufficient length so that it cannot be derived from an encrypted message.

If a substitution cipher is used, known plaintext attacks must be considered as a serious risk. The direct mapping on one character to another greatly simplifies brute forcing.

Within the context of the encryption algorithm used, there are no simple fixes to this issue. It would require ground-up redesign.

Effective key length is very short

Summary

Given that the mapping table is constant and known, and that the starting variable is extremely short, the effective key length cannot be considered to have any length.

Detail

EN50136 2008 states:

> When symmetric encryption algorithms are used, key length shall be no less than 128 bits. When other algorithms are deployed, they shall provide similar level of crypto graphical strength. Any hash functions used shall give a minimum of 256 bits output. Regular automatic key changes shall be used with machine generated randomized keys.

Given that the mapping table is constant and known, and that the starting variable is extremely short, the effective key length is zero.

If we consider the mapping table as a key, then it has a length of ~48bytes or 384bits. This is technically longer than the requirement above, but it is extremely tenuous to call a constant and known value a key. We also need to consider that this is a substitution cipher and the key would need to be far longer than 384bits to approach the security of a peer-reviewed block cipher such as AES-128.

If we consider the starting variable as a key, then it has a length of 6bits, far short of the requirement above. Given the issues with the algorithm though, it would be tenuous to call this a key.

Even if the mapping table is not known, it is possible to derive the mapping table from a limited number of observed communications between the SPT and polling server.

There appears to be no mechanism to allow for key change at all, as required by EN50136 2008.

Potential vulnerability

The effective key length is so short that it is possible to both encrypt and decrypt messages without any further knowledge.

Remediation

To be secure, a protocol needs a key that is:

- Adequately long to protect against brute forcing.
- Adequately protected to prevent it being known.
- Changed frequently enough so that even if it is uncovered, it only compromises the system for a limited period of time.

A common mechanism to satisfy these requirements is to setup a temporary session key. This can be established in a number of ways:

- Use of a time synchronised pseudo-random number generator.

- Use of asymmetric (public key cryptography) to establish a random session key. This has the advantage of providing mutual authentication as well.
- Use of a key exchange mechanism like Diffie-Hellman which allows non-authenticated key exchange over an insecure channel (this is still subject to man-in-the-middle attacks if no authentication is performed).

Realistically, the entire encryption algorithm and protocol would need to be redesigned for this to not be an issue.

No sequence number in messages

Summary

There appears to be no sequence number in the messages, facilitating replay attacks and making it hard to detect missing messages.

Detail

Observed communications between the SPT and polling server have no sequence number.

Potential vulnerability

Sequence numbers enable the detection of missing messages or packets in a communication channel. Without sequence numbers, you need to rely on timing to detect missing messages. This is adequate for synchronous communications such as polling messages, but inadequate for asynchronous ones such as alarm messages.

It would be possible for an attacker to block alarm messages whilst allowing polling messages through based solely on timing. The polling server would not be able to detect the missing messages.

Remediation

Best practice dictates that a sequence number should be included in each message.

No checksum or hash in messages

Summary

There appears to be no checksum or hash in the messages

Detail

Observed communications between the SPT and polling server have no checksum or hash either in the unencrypted or encrypted message.

The ICCID (SIM card identification number) has a check digit at the end of it, using the Luhn algorithm. This would allow for some errors in the ICCID part of the message to be detected.

Potential vulnerability

Without a checksum or hash, it is difficult or impossible to detect that a message that has been altered either accidentally or maliciously.

Because a substitution cipher has been used, there is the potential for an individual character to be altered and the message remain valid.

For example, we could encrypt a valid ICCID and chip number with a status message as follows:

```
Key:        81
```

```
Status:    894410123456789012 37A12345656555555555567
Encrypted: ZSh8L2EcR10WqoeFImR1j4K1GRlJTejXmUY15hgGQMe
```

A single bit being flipped can change this into:

```
Key:       81
Encrypted: ZSh8L2EcR10WqoeFImR1j4K1GRlJTejXmTY15hgGQMe
Status:    894410123456789012 37A12345656655 4555555567
```

This is still a valid message but where a closing event (4) has been sent on one of the alarm channels instead of no event (5).

A checksum or hash is largely for the detection of accidental alteration. An attacker who wishes to alter the message could also regenerate the checksum.

Remediation

Best practice dictates that a checksum or hash should be included in each message.

No message authentication code in messages

Summary

There appears to be no message authentication code (MAC) in any of the messages.

Detail

Observed communications between the SPT and polling server have no message authentication code.

Potential vulnerability

A message authentication code would use a secret key to generate a hash of the message. This allows someone else possessing the secret key to be sure that the message was the same as when it was sent (integrity) and that the message was sent from someone else in possession of the key (authenticity).

A message authentication code can protect against both accidental and malicious alteration of messages.

In the DC4 protocol, no message authentication code is used. This means that it is trivial to generate messages which the polling server believes are from the SPT and vice versa.

Remediation

Best practice dictates that a message authentication code should be included in each message.

No key exchange or rotation

Summary

There is no facility to change the key material without programming the flash memory in the SPT.

Detail

The key material (mapping table) is stored in flash memory, and there is no means to update this except by flashing the SPT. This can only be done by removing the case from the SPT and connecting a programmer and computer to it. There is no facility for remote firmware updates.

The lack of remote firmware update functionality has been observed by firmware disassembly and confirmation from CSL support.

Potential vulnerability

A encryption algorithm and protocol where the key material is known, constant, and difficult to change affords little secrecy.

Remediation

Each device could have a custom mapping table. This would improve security greatly, but the encryption algorithm is so weak that it would be possible to derive the mapping table using a very limited number of observed communications.

The key should be stored in a way that allows it to be updated easily, frequently, securely and remotely.

It is common practice to establish a temporary session key using public-key cryptography or to use a key exchange mechanism like Diffie-Hellman. Both methods mean that the session key used is only used for a short period of time. If this period of time is less than the period of time taken to derive or brute-force the key, the system will remain secure.

Failing this, keys can be exchanged or rotated periodically by any other means.

Identification information is poorly protected

Summary

The identification information encrypted in each message is poorly protected, facilitating several attacks.

Detail

The messages sent from the SPT to the polling server are of the form:
`<ICCID>|A|<Chip number>|<Status>`

The ICCID is always machine printed on the case of the CS2300-R, is printed on the SIM, and stored in the NVRAM (so can be read out after removal of the NVRAM chip, or by logic tracing in-situ).

The chip number is often hand-written on the case of the CS2300-R.

The Gemini Insight website shows the ICCID (without check digit) and chip number in many summary screens, which can easily be shoulder surfed (this has been observed in promotional videos from CSL).

Further to this, weaknesses in the encryption algorithm means that it is possible to derive the ICCID and chip number from a single observed communication.

None of the identification information is protected well enough to be considered secure by any reasonable standard.

The CS2300-R appears to reach the S2 ATS criteria (substitution security) in EN50131. In EN 50131-1:2006+A1:2009, we have the following requirement around S2:

Substitution security: Protection against unauthorised substitution of the alarm system transceiver with similar equipment along the Alarm transmission system transmission path shall be provided in one of the following ways:

S0 No measures.

S1 Measures to detect substitution of the supervised premises transceiver by addition of an identity or address in all messages transmitted on the alarm transmission path.

S2 Measures to detect substitution of the supervised premises transceiver by

a) encryption of an identity or address in all messages transmitted on the alarm transmission path,

b) authentication of the supervised premises transceiver by the addition of a different and un-revealed code for each connected transceiver, or

c) another measure as specified by the manufacturer.

Authentication always requires a sufficient number of keys to provide each connected transceiver with a unique code. The identity range in S2 must not be less than 250 unique addresses.

This states that a “un-revealed code” can be used to detect substitution of the SPT with similar equipment. It is not stated who should be prevented from knowing this code.

If the intent is for no-one (installer, ARC, attacker) to know the code, then it is clear that the installer and ARC can easily view the code.

If the intent is for just the attacker to be prevented from knowing the code, then he can either view the unit or sniff a single communication.

It may be the case that CSL are using “another measure as specified by the manufacturer” as specified in part c) of the standard. If so, it would be reasonable to assume that “another measure” should provide similar security to parts a) and b). There is no evidence in the firmware of any further substitution detection taking place.

It may be the case that CSL are transmitting an identity out-of-band on the GPRS path, and using this to detect substitution. For example, the SIM IMSI number is stored on the SIM and transmitted as part of the GPRS protocol. The CSL polling server could be made aware of the IMSI of the SIM. By checking that the ICCID/chip number in the message matches the IMSI of the SIM, they could detect substitution.

Neither the patent nor observed operation of the system indicate that this is happening.

Importantly, out-of-band signalling of the IMSI could not be used on the PSTN or IP paths, so they would still remain vulnerable to substitution.

Potential vulnerability

Viewing the CS2300-R unit or observing a single encrypted message reveals all the data required to send fake messages claiming to be from the SPT on the IP and PSTN path. It may reveal all of the data required to send fake messages claiming to be from the SPT on the GPRS path, but further research would be required to confirm this.

This could allow someone to substitute the SPT for another, or to generate spoofed messages from entirely different equipment such as a desktop PC.

Remediation

Within the context of the encryption algorithm used, there are no simple fixes to this issue. It would require ground-up redesign to protect any information sent.

If strong encryption was used instead, it would be possible to store a secret unique ID internally on the SPT. This could be communicated to the polling server without being displayed on the unit or in the web management interface.

There would still potential for the secret ID to be read out from NVRAM and then used to spoof communications. Ideally this secret ID would be stored in memory internal to the microcontroller, greatly increasing the difficultly in reading it out.

Encryption algorithm is weak

Summary

The encryption algorithm used is very basic and suffers from a number of serious flaws.

Detail

The encryption algorithm is a simple polyalphabetic substitution cipher.

A substitution cipher is one where each character is substituted for another, for example the Caesar Cipher. If the key is 5, the substitution table is:

```
Original:  ABCDEFGHIJKLMNOPQRSTUVWXYZ
Encrypted: FGHIJKLMNOPQRSTUVWXYZABCDE
```

To encrypt the word DUALCOM, we follow the substitutions:

```
Original:  DUALCOM
Encrypted: IZFQHTR
```

A polyalphabetic substitution cipher uses a different shift for each character, which can be called a mapping table. As each character is encrypted, we move to the next shift in the mapping table.

The CSL DC4 encryption algorithm uses a 48byte long mapping table. The starting position in the table is sent from the polling server to the board as a starting variable. This increases security slightly as two identical messages sent with different starting variables will result in different ciphertext.

The algorithm has some further complexity around constraining the input and output characters to be within the printable ASCII range.

This algorithm, combined with the messages being sent, suffers from a number of serious flaws. None of these are subtle and are widely known issues.

Some observations and assumptions about the input data need to be noted. Each message comprises of:

```
<ICCID>|A|<Chip number>|<Status>
```

The ICCID always begins with *894410*, with the remaining 14 characters being numeric.

The *A* appears to be constant.

The chip number is a 6 digit numeric string such as *512387.*

The status appears to be loosely based on the Dualcom96 protocol and is often of the form *5665555555555567* where *5* is “no event, input is in quiescent state”, *6* is “no event, input in active state”, and the final *7* is constant.

This means we have a partially known plaintext with tight restriction on many of the other characters.

Deriving mapping table from observed communications
If we observe a single communication between the polling server and board, of the form:

```
PS:              HS52
SPT:             0x01|ZSh8L2DaRXVWmn6GOqP3j8JYCOgFSekXmUY15hgGQLdH
Known chars:          894410              A                      7
```

We can determine 8 bytes of the mapping table at the positions where we know the plaintext.

As we observe more and more communications, we can build up a complete picture of the mapping table. The starting variable (*HSxx)* sent by the polling server appears to be random. This means you would need to observe a fair number of messages before being able to derive the entire mapping table, but it would still be perfectly reasonable task.

If we already know the ICCID and chip number (by virtue of them being printed on the SPT), each observed communication will reveal 28 consecutive bytes of the mapping table (20 digits of ICCID, 1 digit of A, 6 digits of chip number, and 1 digit of 7 at the end of the status).

If we connect the SPT to a fake polling server where we have control of the starting variable, we only need to observe two communications before the entire mapping table is known (as each one reveals 28 bytes of the 48 byte mapping table. Because there is no identity verification, this is easy to achieve.

Trivial to alter message content
Given that we know the content of the message, there is no hash, checksum or message authentication code, and that a substitution cipher is in use, we can easily change one valid message into another.

Take for example a message showing that all inputs are in “no event, input is in quiescent state” condition. This is encoded as a 5 in the status.

```
Status:      894410123456789012 37A123456555555555555557
Encrypted:   2i7MZCNhHRdkZpYTX2fiLMIaEbo46WOi9P7IfT20Von
Decrypted:   894410123456789012 37A123456555555555555557
```

If we wish to change the inputs to be “new alarm event” which is encoded as a 1, we simply need to shift the encrypted characters down by 4 i.e. 6 becomes 2 and T becomes P:

```
Status:      894410123456789012 37A123456555555555555557
Encrypted:   2i7MZCNhHRdkZpYTX2fiLMIaEbo46WOi9P7IfT20Von
Altered:     2i7MZCNhHRdkZpYTX2fiLMIaEbo02SKe5L3EbPYWRkn
Decrypted:   894410123456789012 37A123456111111111111117
```

Without knowing the mapping table or starting variable, we have significantly altered the meaning of the message.

Potential vulnerability

The algorithm, combined with other issues, fails to meet any of the goals of encryption:

- It provides little confidentiality - it is trivial to decrypt messages.

- It provides no integrity - it is possible to alter one part of a message and the other party is unaware.
- It provides no authentication - neither party knows it is speaking to a genuine counterparty.

Remediation

The encryption algorithm needs redesigning from the ground-up to meet the goals of encryption.

NVRAM is unencrypted and easy to copy

Summary

The NVRAM is a socketed SPI EEPROM which can be read and copied using readily available tools. The data on it is unencrypted and un-hashed/checksummed, making it easy to modify data.

Detail

The NVRAM contains a number of settings, including:
- ICCID and chip number used in messages
- APN connection details
- Telephone numbers used on the PSTN path
- SIM remote command PIN

The NVRAM is a socketed SPI EEPROM. These can be read and copied using readily available tools, such as the Bus Pirate or any other EEPROM reader. Alternatively, a logic analyser can sniff communications in-situ.

The data is not encrypted in any form. Encryption would make reading and modifying the data by an attacker harder.

There is no hash or checksum on the data. A hash or checksum would have the primary purpose of protecting against accidental errors, but would also make deliberate modification of data slightly harder.

This means that the confidentially, authenticity and integrity of the data in the NVRAM cannot be relied upon.

Potential vulnerability

Data that must remain secret to maintain the security of the CSL system can be easily read from NVRAM.

Device capture is a real threat to embedded devices. Whilst it could be argued that capturing the actual system under attack would be difficult to impossible, commonality between multiple devices means that capturing a low-risk SPT (e.g. from a rural newsagent) could yield key information about a higher-risk SPT (e.g. a bank or jewellers).

For example, the APN connection details seem to be common across all boards (discussed below), and the SMS PIN seems to be common among many boards.

Remediation

Data that must remain confidential should be protected adequately.

This could mean encrypting data on the external EEPROM. EEPROMs have a finite lifetime in terms of write cycles, which means that writes should be minimised both in terms of number of writes and the amount of data written. EEPROM writes are also slow.

This can make encryption challenging, as in a good encryption algorithm, a small change to plaintext should result in a large change in ciphertext. This can cause a small settings change to impact a large part of the EEPROM, reducing it's lifetime. This may not be a problem as the NVRAM is rarely updated.

Irrespective of this, a checksum or hash on EEPROM is good practice to protect against accidental modification. In a critical security system, this would be consider best practice.

An alternative protection strategy would be to use EEPROM internal to the microcontroller. The 78K0R doesn't have internal EEPROM, but can emulate EEPROM using the flash memory. Many other microcontrollers offer internal EEPROM.

Firmware has no encryption, checksum or hash

Summary

The firmware has no encryption, checksum or hash, facilitating deliberate modification and allowing accidental errors.

Detail

There is no evidence in the firmware of any encryption, checksum or hash on the flash memory.

Firmware downloaded from the CSL website is not encrypted.

Potential vulnerability

It is easy to obtain and disassemble the firmware by downloading it from CSL. This allows an attacker to examine the operation of the device in great detail and obtain secret material such as the mapping table. Most of the issues in this document were found from examination of downloaded firmware.

It is also possible for an attacker to modify the firmware. This facilitates a number of attacks. For example:

- A modified firmware that ignores alarm signals between 1800 on Friday and 0700 on Monday, allowing an attacker unencumbered access to the premises.
- A modified firmware that skips the comparison between ICCID in the NVRAM and ICCID in the SIM, allowing spoofed GPRS messages to be sent.

Remediation

Encrypted firmware (either encrypted for transmission, or for transmission and storage) is a useful technique to prevent attackers from examining the firmware or modifying the firmware. This would be part of a defence-in-depth strategy.

Most microcontrollers have a facility to store a protected bootloader and secret key to decrypt and run firmware. These are widely documented and often available as application notes from the manufacturer.

Encryption in isolation would only provide confidentiality, but to prove authenticity and integrity, a message authentication code should also be used.

A checksum or hash protects against accidental modification and would be regarded as good practice, especially in a security system.

It is important to note that it is possible to develop a secure system where the firmware and even source code is public. If just the secret material (e.g. keys, mapping tables, identification data) is

protected, the system should remain secure.However, for many embedded systems, protecting the firmware is wise as part of a layered security model.

Poor backend controls for active SIMs/SPTs

Summary

A number of used/out-of-use SIM cards and SPTs could still successfully connect to the Vodafone network, CSL private APN and CSL polling servers.

Detail

4 out of the 13 SPTs could connect to the Vodafone network, CSL private APN and CSL polling servers, and would report that alarm messages could be sent successfully.

These boards had been out-of-use for at least 6 months (they had been in my possession for 6 months), some probably as long as 2-3 years.

3 of the remaining 13 boards failed to get to the stage of trying to connect to the Vodafone network due to various startup issues (e.g. unable to connect to a panel on the RS485 port), so it is unknown if they were active or not.

These 4 working SPTs were shut-down at the network level after images showing their ICCID were posted on a blog and CSL contacted about potential issues. It is not known if they were shut-down because of this, or for another reason (e.g. they were found by backend audit).

Regardless, this suggests that CSL have poor backend controls for active SIMs/SPTs.

Potential vulnerability

Access to the CSL private APN appears to be restricted by IMSI number i.e. only SIM card explicitly provisioned by CSL to be able to connect to their APN.

By failing to keep track of active SIMs and SPTs, CSL are allowing access to their APN and polling servers.

An attacker could purchase a used SPT or steal an active one and use the SIM, GPRS modem, or entire SPT to facilitate an attack.

Remediation

Backend controls should be used to prevent out-of-use SIMs/SPTs accessing the Vodafone network and CSL APN.

Marketing materials imply that AES-256 and an IPSEC VPN is used

Summary

It is implied that AES-256 and an IPSEC VPN are used to secure data. This does not appear to be the case from the SPT to the polling server.

Detail

From CSL marketing material:

```
Data is encrypted with Advanced Encryption Standard (AES)-256 via CSL
DualCom's IPSec VPN.
```

This could be true for newer products, or from the CSL polling server to ARC, but there is no evidence that AES is used in the CS2300-R SPT, or that an IPSec VPN is used.

It is highly unlikely that the 78K0R would be able to do AES-256 and IPSec whilst also performing other activities.

Potential vulnerability

The marketing implies that strong, proven encryption is used from the SPT to the ARC when this is not the case.

Users, installers, ARCs and insurers might reasonably expect this encryption to be used along the entire path.

It could be argued that this gives CSL an unfair advantage in the market.

It could also mean that users, installer and ARCs do not implement further layers of security (e.g. VLANs, firewalling etc.) as they are relying on the functionality and security that CSL provide.

It also raises questions around differences between the published and realised specifications. If part of the specification is overstated or incorrect, there could be more significant differences.

Remediation

CSL should be open about the encryption used in their system, including over which portion of the path the encryption covers. It is possible for a secure protocol to be public still remain secure.

Common SMS remote command PIN across many devices

Summary

The SPT uses a 6-digit PIN to restrict access to SMS remote commands. A significant number of devices observed had the same PIN of 001984.

Detail

Observation of the NVRAM chips across 13 SPTs showed that 9 of them had the same SMS remote command PIN of 001984.

This PIN is used to restrict access to the SMS remote commands on the SPT.

The PIN is stored in NVRAM and can only be changed by reprogramming the NVRAM. Very few installers have the facilities to reprogram the NVRAM, so it is unlikely that the PIN would ever be changed.

The history of the SPTs tested is not known. It could be that 9 of the SPTs were installed by the same installer/ARC and it was requested that the PINs were the same across all devices.

Any telephone number with the correct PIN appears to be able to control the SPT. There is no concept of whitelisting a limited number of telephone numbers.

There appears to be no rate-limiting or lock-out when attempting to brute-force the PIN. Often with PINs this is a serious vulnerability. However, the use of SMS limits the rate at which PINs can be attempted. Vodafone currently offer unlimited SMS on their own network for £10/month, making this a low-cost option.

Potential vulnerability

Knowing the PIN and the telephone number of the SPT would allow an attacker to run SMS remote commands.

This requires that the attacker knows the telephone number of this device. This can be obtained by a number of means:

- On some SPTs the telephone number is printed on the device and can be viewed.
- GPRS communications can be passively sniffed and the encryption broken to yield the MSISDN (telephone number).
- The network operator's systems can be used to map ICCID or IMSI to MSIDSN. ICCID can be obtained by observation of a single IP or PSTN path communication.

There are undocumented SMS remote commands that allow the configuration of the device to be changed, including changing the IP address and port that the device connects to over the IP path.

Remediation

If a PIN is to be used to restrict access to SMS remote commands, then it should be a random 6 digit PIN, not a predefined one. It should also be easy for installers to change it at the time of install, and for it to be changed remotely later on.

Brute-forcing is not a likely attack due to the time involved. Regardless, the penalty for switching a 6-digit PIN to a longer password is minimal and would reduce this risk even further.

Undocumented SMS remote commands

Summary

There are several undocumented SMS remote commands that allow the device to be reconfigured, facilitating several other attacks.

Detail

The SPT can receive SMS remote commands. CSL document several of these, including VER to get the software version, TEST to test SMS.

There are also other commands that allow important configuration to be altered such as *<PIN> 4 2 xxx.xxx.xxx.xxx* which results in a response of *GPRS DualCom IP addr changed>xxx.xxx.xxx.xxx*

Another command, *CALL*, triggers a "download" from CSL servers. This appears to contain further settings and configuration.

Potential vulnerability

In combination with the common SMS remote command PIN, it is possible for an attacker to alter the configuration of an SPT.

This could include altering the IP address of the polling server that the SPT connects to, facilitating a MITM attack.

Alternatively, communications settings could be changed to point to a server under an attacker's control, and then a "download" triggered. This could allow the SPT to be significantly reconfigured.

EN 50136:2008 also seems to indicate that encryption should apply to all data and management functions of the ATS. This is a management function and it is not encrypted.

Remediation

Undocumented functionality is rarely viewed favourably in security products. This kind of functionality should either be documented and adequately secured, or removed from the firmware.

As the firmware has not been exhaustively examined, there could be further undocumented functionality.

Digi Connect ME based IP boards have an open configuration protocol

Summary

Some LAN boards have an open configuration protocol, allowing reconfiguration of the device.

Detail

Some LAN boards used on the SPTs use an integrated LAN module called a Digi Connect ME. This is a small module similar in appearance to a PCB-mounted RJ45 socket, but contains a microcontroller and flash memory with a full OS and networking stack. It is used as a serial to Ethernet bridge in the CSL products.

Digi Connect ME modules use a protocol call ADDP (Advanced Device Discovery Protocol) to discover devices on the local network segment and allow them to be reconfigured. This includes changing the IP address and default gateway. This protocol can be disabled as a configuration option.

Potential vulnerability

Allowing the IP address and default gateway to be changed could facilitate a man-in-the-middle attack. Alternatively, it could be used as a denial-of-service attack if set to invalid values.

Whilst it was initially intended for ADDP to only be used on the local network segment, it appears to be able to be used outside of this scope i.e. from anywhere on the Internet. This can be used to discover devices by port scanning, although there appears to be no easy way to differentiate between a CSL SPT and another device.

Remediation

It is essential that configuration protocols are either disabled or secured to prevent them being used in other attacks.

Digi Connect ME based IP boards have an open web interface

Summary

Some LAN boards have an open web interface, allowing reconfiguration of the device and upload of new firmware.

Detail

As above, some LAN boards on the SPTs use an integrated LAN module called a Digi Connect ME.

These have an open web interface, which allows the device to be reconfigured (IP address, default gateway) and for new firmware to be uploaded. Access to the web interface is not filtered in any way.

Potential vulnerability

An attacker can reconfigure the device to facilitate a man-in-the-middle attack.

A custom firmware could be uploaded that appears to operate normally but fails to send alarm messages.

Remediation

The web interface should be secured by a password, ideally different for each device.

If possible, access should be prevented unless from the local network segment. This is a common defensive technique on embedded systems.

IP path polling servers have multiple open ports and open directory listings

Summary

Several of the CSL polling servers that are exposed to the wider Internet have a number of open ports and open directory listings.

Detail

Several of the CSL polling servers have multiple ports open, including FTP, SMB and MSRPC. They are connected to the open Internet so can be accessed by an attacker.

There are also open directories which show web pages with content unrelated to the task of acting as a polling server.

The IP addresses of these servers were found by examination of NVRAM contents.

Potential vulnerability

Leaving ports open on critical servers increases the attack surface greatly. By exploiting an attack on one ports, it could be possible to impact the polling server.

For example, FTP sends credentials in the clear. This makes it vulnerable to sniffing. FTP credentials could allow an attacker to modify files on the polling server.

The open ports on the server make it clear which OS the server is running, what software it is running (Windows Server 2008, IIS 6.0) and how up to date it is. Finding targeted exploits is far easier when the specific OS is known.

Open directories allow an attacker to gain further information about a system. This could be something as simple as seeing the source of a web script, through to directory traversal leaking a password file.

Both of these indicate that the servers are used for multiple purposes and have not been adequately secured.

Remediation

Servers used for critical services should be locked down as far as is possible. This should include:

- Stopping services that are not used.
- Firewalling services to restrict access from the wider Internet (e.g. only allow whitelisted IPs or connections from a VPN)
- Not using protocols that send credentials in the clear when there are secure alternatives (e.g. swap FTP for SFTP).

Open directories are a common issue on servers. There is rarely a need to have this functionality turned on.

Techniques such as virtualisation allow critical servers such as the polling server to be securely isolated from less critical servers such as web hosting and product ordering.

No means to update firmware remotely

Summary

The SPT has no means of updating firmware remotely.

Detail

The only means to update firmware on the SPT is for them to be flashed locally. This involves visiting the site, powering down the system, removing the cover and using a Renesas MiniCube2 to flash the firmware.

Most installers do not have the equipment to flash the SPT with new firmware.

This means that fixing issues by firmware updates is expensive and time consuming.

Potential vulnerability

Once a SPT is installed, it is highly unlikely that the firmware will be updated. This means that vulnerabilities and bug fixes, even if in newer firmware, are unlikely to ever be deployed.

This, in turn, means it is unlikely that vulnerabilities and bug fixes will be developed as the likelihood of them being deployed is so low.

An attacker is likely to be able to exploit known issues in existing systems as a result.

Remediation

A security device which is connected to GPRS, PSTN and IP networks must have functionality to allow the firmware to be updated remotely.

This would not be a trivial task within the constraints of the existing hardware.

Microcontroller has limited headroom for additional functionality and fixes

Summary

The microcontroller is extremely limited in capacity and functionality compared to most modern microcontrollers used in similar products.

Detail

The most recent firmware for download appears to use 81KB out of 128KB of flash memory available, leaving limited room for expansion.

There is no real-time clock available. This can make implementing many cryptographic algorithms difficult. It is hard to create an event log with accurate times as well.

There is no hardware random number generator (RNG) or other source of entropy on the SPT. This is frequently an issue on small embedded systems with no user interaction.

There is no unique, read-only hardware ID that can be used to identify the boards.

There is no hardware acceleration for cryptographic functions.

Potential vulnerability

The limited capacity and functionality available mean that it would be hard to implement a secure protocol that used a common, peer-reviewed encryption algorithm whilst maintaining existing functionality.

The same microcontroller has been used over a range of products introduced over a number of years. This indicates a reluctance to move to newer hardware, which in turn means an attacker can rely on newer systems also being limited in capacity.

Remediation

Microcontroller capacity and functionality is improving all of the time. The design of an SPT should not be locked into a given architecture, and designs should evolve to allow for a secure system to be developed and maintained.

Poor access controls to “Installer shop”

Summary

There appears to be no control over who can access the “Installer shop”.

Detail

You are required to register for an account to access the “Installer shop”. Inside the “Installer shop” there is documentation, high resolution photographs, and firmware available for download.

It appears that all accounts are approved regardless of who has created them, even if they have clearly bogus details.

If it is important that access to the “Installer shop” is restricted to installers only, then the controls around this are inadequate.

Potential vulnerability

Several of the findings in this report were found using data from the “Installer shop”

- Documentation uncovered the open discovery protocol and web interface on LAN boards.
- High-resolution photos of the flash programming interface uncovered the means by which to program firmware
- Firmware download allowed disassembly and then reverse engineering of the encryption algorithm.

Whilst inadequate as the only layer of security, hiding these from all but vetted installers would slow down attacks greatly.

As it stands, it is incredibly easy to gain access. It appears that this layer of security is notionally present, but the implementation of it is token.

Remediation

Controlling access to firmware and documentation is a useful layer of security, but it needs to be carried out properly.

ARCs with CSL routers run IKEv1

Summary

Several ARCs have been observed connecting to CSL using the IKEv1 protocol which suffers from a number of vulnerabilities including trivial password brute-force and denial-of-service.

Detail

During routine penetration testing of several ARCs, Cisco ASA 5505 devices have been found. These are used to connect the ARC into the CSL Dualcom network.

These Cisco ASA 5505 devices are running IKEv1 using a pre-shared key (this is essentially a password) in aggressive mode. This allows the pre-shared key to be brute forced.

It is also possible to perform a denial of service attack by quickly requesting multiple key slots on the device.

The CSL end of the VPN has not been inspected.

Remediation

There is little reason to use IKEv1 when IKEv2 is available and suffers from none of these issues.

Conclusions

In isolation, no single issue listed here would result in a catastrophic failure of the CSL system.

I have had to halt investigation into a number of issues as I clearly cannot take actions which step on the wrong side of the law, and I also do not wish to impact day-to-day operations of either CSL, the ARCs that use them as a signalling provider, or their customers.

I suspect further investigation would result in more issues being found, some of which may be of a serious nature.

Irrespective of this, the sheer number and basic nature of some of the mistakes being made indicate that the CSL system has poor security. The DC4 protocol is so badly designed and so flawed that it adds no security above sending messages in the plain.

However, of largest concern is that the product doesn't seem to meet the standards it is claimed to. I am no expert in reading standards, so it would require someone with more experience and knowledge to verify this.