sensepost

Security Assessment Report

Client	**Sigma Designs**
Project Name	**Security 2 Command Class Protocol Review**
Project Code	**SP02508**
Date	**2017-08-18**

TABLE OF CONTENTS

1 Executive Summary

1.1 Assessment Overview

The assessment of Sigma Designs' *Security 2 Command Class* commenced on the 19th of June 2017 and concluded on the 21st of June 2017. This assessment was the culmination of several previous projects – both on a documentation review level as well as a technical assessment of the protocol implemented on hardware provided by Sigma Designs. This final project was requested by Sigma Systems in order to identify any final concerns prior to the standard being finalised and published.

Sigma Designs engaged the services of SensePost in order to:

- Evaluate whether the risk of replay attacks identified during previous projects had been effectively mitigated.
- Evaluate whether the security controls introduced in the *Security 2 Command Class* were effective when physically implemented.
- Gauge whether the risk identified within the protocol was at a level acceptable and that such risk would not have a significant impact on the delivery of the service, expose clients to harm or loss or other such consequences.

The results provided are the output of the security assessment performed and should be used as input into a larger risk management process.

These results are a point in time assessment of the system and environment as they were presented for testing. Any changes could yield a different set of results.

1.2 Motivation for conducting security review

Sigma Designs wanted to verify the security of their newly developed S2 security framework. Third party independent verification from a qualified security analyst chosen in addition to in-house testing. From experience, this greatly improves the analysis coverage and number of flaws found. Analysts from SensePost had previously demonstrated knowledge and skills needed to uncover flaws in older Z-Wave devices, and hiring them was a natural choice.

1.3 About SensePost

As trusted advisors we deliver insight, information and systems to enable our customers to make informed decisions about Information Security that support their business performance. SensePost is an independent and objective organisation specialising in Information Security Consulting, Training, Security Assessment Services, Security Vulnerability Management and Research. SensePost is about security but specifically, Information Security. Even more specifically - measuring Information Security. We've made it our mission to develop a set of competencies and services that deliver to our customers, insight into the security posture of their information and information systems. Our roots run deep, we were established in 2000 and since 2014 form part of the SecureData Group of companies based in the United Kingdom.

1.4 Risk Summary

The overall information security risk rating was calculated as: **Informational**.

It should be mentioned, however, that this rating has been attributed as a result of the highest criticality finding discovered during the course of the assessment, and that this specific finding may be by design. More details regarding this will be presented later in this report.

This is based on the following statistics:

0 Critical	0 High	0 Medium
0 **Low**	**2** **Info**	**2** **Total**

Table 1 – Risk Summary

1.5 Conclusion & Recommendations

Overall, the implementation of the protocol was found to be relatively robust, with only one finding associated with the protocol. Consultation with the client after the project also indicated that, not only was this behaviour by design, but that an attacker would require very privileged access to an existing network to conduct such attacks. Furthermore, several procedures were in place, from a certification perspective, ensuring that persons installing a node on an existing network would be notified should protocol downgrade attacks occur.

Other than that, it was found that serial communications between the computer and the USB devices were not obfuscated. Neither were the application binaries. This made it trivial to reverse-engineer the communications protocol used to control the devices. Once again, consultation with Sigma Designs indicated that the PC Controller software is not an end-user tool, and is provided purely for people to develop and test Z-Wave networks. In real-world deployments, controller software would be run from secure devices, and end-users would not have ready access to binaries or firmware. This would make reverse engineering controller applications significantly more complex.

2 Z-WAVE SECURITY

2.1 Z-Wave Network Security

The *Security 2 Command Class* allows various nodes on a Z-Wave network to communicate securely with each other. Backwards compatibility with nodes implementing prior versions of the *Security Command Class* is supported by means of Security Scheme 0, although newer devices are envisioned to support the use of Security Scheme 2, which offers numerous advantages over prior schemes.

The *Security 2 Command Class* provides support for secure key exchange as well as secure single-case and multi-case communication. Replay attacks are prevented by means of Pre-Agreed Nonces. For singlecast communication, the "Singlecast Pre-Agreed Nonce" (SPAN) is utilised. Likewise, for multicast communication, the "Multicast Pre-Agreed Nonce" (MPAN) is used.

Earlier reviews were based on the revised protocol standard document provided by Sigma Designs titled: *Security 2 Command Class, version 1 ALPHA (S0, S2, Security Command Class)*. The document version was SDS11274.

This assessment was based on the finalised standard document provided by Sigma Designs, titled: *Z-Wave Transport-Encapsulation Command Class Specification*. The document number was SDS13783, and the review focussed on sections 3.6 – *Security 2 (S2) Command Class, version 1* – and 3.7 – *Supervision Command Class, version 1*.

3 TESTING METHODOLOGY

3.1 Summary

The following section details the research approach and methodology used during the course of the assessment. The section describes the components which were tested, as well as the manner in which tests were performed.

3.2 Z-Wave Controller Components

Various components of the Z-Wave PC Controller application were decompiled, and the USB communications between the application and the USB controller were intercepted. It was found that the USB Static controller provided a virtual communications port. This COM port was used by the PC Controller application to communicate with the USB controller. Unlike the Zniffer USB controller, the Static Controller communicated using 115200 baud.

Whilst the analyst originally implemented his own application for communicating with the controller by means of the COM port, the software provided by Sigma Designs on the 22nd of February proved to be better implemented. The analyst relied on modified versions of the Sigma-developed python scripts in order to continue testing. Modified firmware was also provided by Sigma Designs, which would allow the analyst to inject frames with spoofed source ID's.

The analyst made use of the *sscon.py* python script and modified firmware in order to attempt replay attacks, and cause undesirable behaviour on various devices. None of the attempts were successful.

In addition to this, various hard-coded and reserved parameters within frames were modified in the *sscon.py* and associated dependencies during the course of the assessment, in order to determine whether there were other vulnerabilities present within the protocol stack. No such vulnerabilities were determined to be present.

Finally, the analyst also attempted to generate exceptionally long frames during the course of the assessment. This was not successful.

3.3 Decryption and Replay Attacks

The analyst conducted brute force attacks against captured frames in an attempt to determine whether it was possible to brute force network keys. This was performed in conjunction with the *Security 2 Command Class* specifications outlined in the document *Security 2 Command Class, version 1 ALPHA (S0, S2, Security Command Class)* (Document number: SDS11274).

It was not possible to recover the network key during the course of the assessment. Furthermore, implementation of the *Pre-Agreed Nonce* (PAN) between different nodes made this process more difficult. The PAN's were also found to effectively protect devices against replay attacks.

4 TECHNICAL MANAGEMENT SUMMARY

4.1 Results Summary

In total, two security issues were identified during the assessment, of which one was originally classified as High risk. Further discussion with the client and subsequent investigation into this finding indicated that the specific behaviour observed was by design.

4.2 Z-Wave Device Assessment

Only one finding related to the *Security 2 Command Class* protocol was identified, and this was a potential downgrade attack. As earlier mentioned, subsequent investigation by the client determined that this behaviour was by design, and there were several further mitigating factors.

Besides the potential protocol downgrade attack, it was found that Sigma Designs distributed their applications without any form of obfuscation. Further to this, serial communications with USB devices were also found to be unobfuscated. This made it trivial for the analyst to reverse-engineer the protocol for controlling the USB devices and implement his own toolset.

Table 2 provides an overview of the risk identified per application assessment category, along with recommendations for resolving the issues identified. The business impact rating method used can be viewed in Appendix B.

Category	Risk	Summary	Recommendations
Data Encryption (DE)	**Info**	A potential protocol downgrade attack was identified, where it was possible to force a device supporting the *Security 2 Command Class* to be included into a Z-Wave network via the *Security 1 Command Class*. As mentioned earlier, discussions with the client indicated that this specific behaviour was by design.	It is highly recommended that Sigma Designs investigate this issue. Details on reproducing the behaviour are outlined in **Error! Reference source not found.**. Should it be found that this behaviour is unintended, the matter would need to be addressed on a firmware level. Investigation by Sigma Designs into the relevant finding indicated that this behaviour was by design. Furthermore, only highly privileged persons would be in a position to initiate this downgrade. Procedures are in place in terms of certification to notify and alert installers in cases where secure devices connect to networks using downgraded communications.
	Info	Application .NET assemblies were distributed without any form of obfuscation. Furthermore, serial communications with USB devices were also unobfuscated. This made it trivial to reverse engineer applications and protocols and obtain a deeper understanding of the devices.	It is highly recommended that applications be distributed with robust obfuscation. Furthermore, it should be considered that serial communication with USB devices be obfuscated to some extent. The .NET assemblies provided with the Z-Wave development kit are experimental. In real-world deployments, controllers would be deployed on secure systems, and

Category	Risk	Summary	Recommendations
			end users would not have easy access to application binaries.

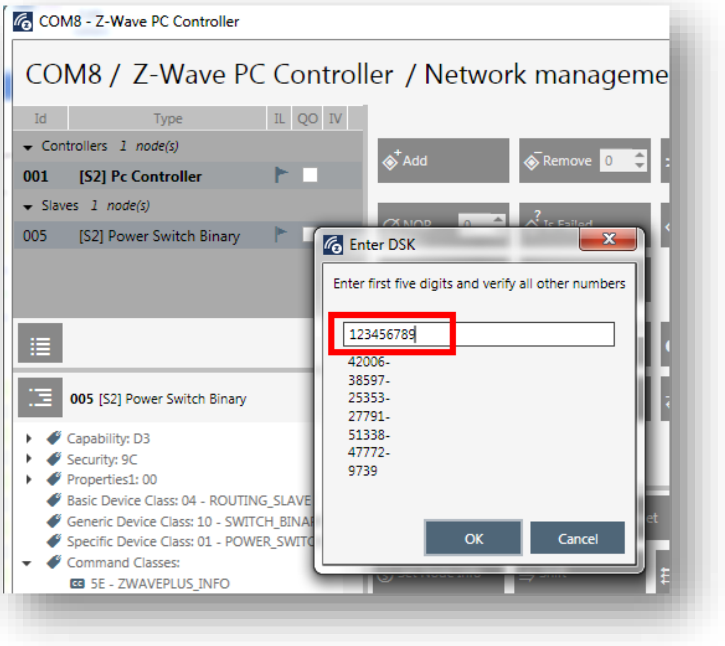
Table 2 – Z-Wave Device Assessment Results Summary

5 Assessment Results

5.1 Z Wave Device Assessment

5.1.1 Data Encryption

Title	**Potential Protocol Downgrade Vulnerability**				
Reference	**R01**	**Risk Rating**	**Informational**	CVSS:3.0/AV:P/AC:H/PR:H/UI:R/S:U/C:N/I:N/A:N	**0.0**
Technical Overview	During the assessment, it was discovered that it was possible to downgrade the *Security 2 Command Class* communication between the controller and the door locks to *Security 0 Command Class* by leveraging a known bug in the inclusion process in the version of the software used. Details on reproducing this behaviour are outlined in Appendix A.1. Further discussion with the client indicated that this behaviour was by design, due to the fact that devices need to maintain backwards compatibility in order to communicate with older devices.				
Attack Conditions	An attacker would need to follow the exact steps outlined in Appendix A.1 in order to reproduce this behaviour.				
Business Impact	A successful attack would result in an attacker being able to downgrade the protocol used by devices implementing the *Security 2 Command Class* to the *Security 0 Command Class* during inclusion. This would render newer devices to the security shortcomings present in the *Security 0 Command Class*.				
Recommendations	It is highly recommended that Sigma Designs investigate this issue. Details on reproducing the behaviour are outlined in Appendix A. Should it be found that this behaviour is unintended, the matter would need to be addressed on a firmware level. Investigation by Sigma Designs into the relevant finding indicated that this behaviour was by design. Furthermore, only highly privileged persons would be in a position to initiate this downgrade. Furthermore, procedures are in place in terms of certification to notify and alert installers in cases where secure devices connect to networks using downgraded communications.				
Attack Example	The following series of images illustrate a Secure Keypad Door Lock being included with an incorrect *Device Specific Key* (DSK) during inclusion, and later communicating via the *Security 0 Command Class*.				



COM8 - Z-Wave PC Controller

COM8 / Z-Wave PC Controller / Network manageme

Id	Type	IL	QO	IV
Controllers 1 node(s)				
001	**[S2] Pc Controller**			□
Slaves 1 node(s)				
005	[S2] Power Switch Binary			

005 [S2] Power Switch Binary

- Capability: D3
- Security: 9C
- Properties1: 00
- Basic Device Class: 04 - ROUTING_SLAVE
- Generic Device Class: 10 - SWITCH_BINAI
- Specific Device Class: 01 - POWER_SWITC
- Command Classes:
  - 5E - ZWAVEPLUS_INFO

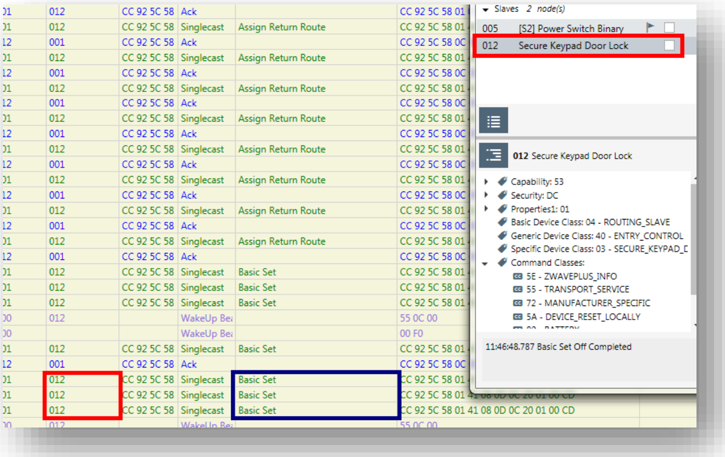
Enter DSK

Enter first five digits and verify all other numbers

123456789

42006-
38597-
25353-
27791-
51338-
47772-
9739

Figure 1 – Incorrect DSK Specified During Inclusion



```
01  012  CC 92 5C 58  Ack                              CC 92 5C 58 01
01  012  CC 92 5C 58  Singlecast   Assign Return Route CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast   Assign Return Route CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast   Assign Return Route CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast   Assign Return Route CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast   Assign Return Route CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast   Assign Return Route CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast   Assign Return Route CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast   Assign Return Route CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast   Basic Set           CC 92 5C 58 01
01  012  CC 92 5C 58  Singlecast   Basic Set           CC 92 5C 58 01
01  012  CC 92 5C 58  Singlecast   Basic Set           CC 92 5C 58 01
00  012               WakeUp Bea                       55 0C 00
00                    WakeUp Bea                       00 F0
01  012  CC 92 5C 58  Singlecast   Basic Set           CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast   Basic Set           CC 92 5C 58 01
01  012  CC 92 5C 58  Singlecast   Basic Set           CC 92 5C 58 01 41 08 0D 0C 20 01 00 CD
01  012  CC 92 5C 58  Singlecast   Basic Set           CC 92 5C 58 01 41 08 0D 0C 20 01 00 CD
00  012               WakeUp Bea                       55 0C 00
```

Slaves 2 node(s)

005	[S2] Power Switch Binary	□
012	Secure Keypad Door Lock	□

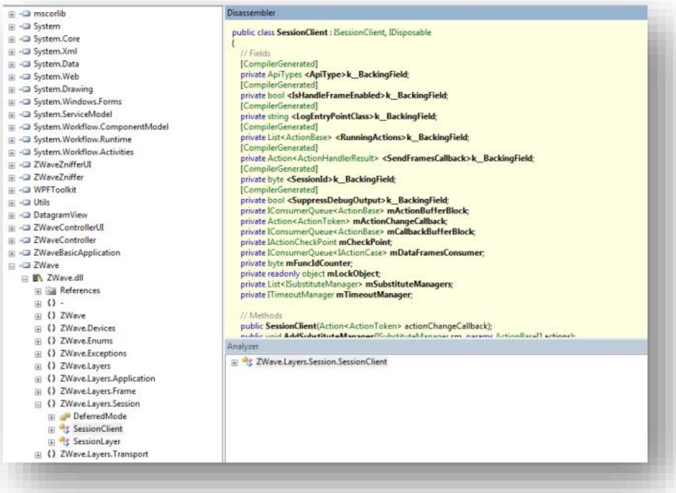
012 Secure Keypad Door Lock

- Capability: 53
- Security: DC
- Properties1: 01
- Basic Device Class: 04 - ROUTING_SLAVE
- Generic Device Class: 40 - ENTRY_CONTROL
- Specific Device Class: 03 - SECURE_KEYPAD_D
- Command Classes:
  - 5E - ZWAVEPLUS_INFO
  - 55 - TRANSPORT_SERVICE
  - 72 - MANUFACTURER_SPECIFIC
  - 5A - DEVICE_RESET_LOCALLY

11:46:48.787 Basic Set Off Completed

Figure 2 – Secure Keypad Door Lock (node 12 - highlighted in red) included using *Security 0 Command Class* (highlighted in dark blue)

Title	**Inadequate Obfuscation of Binaries and Serial Communication**				
Reference	**R02**	**Risk Rating**	**Informational**	https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:P/AC:H/PR:H/UI:R/S:U/C:L/I:N/A:N	**1.6**
Technical Overview	During the assessment, it was found that the application binaries for the PC Controller software as well as the PC Zniffer applications were .NET applications, and neither was compiled with any form of obfuscation. Likewise, serial communications with the USB devices were also found to be very straight forward. Between the unobfuscated binaries and the unobfuscated serial communication, it was trivial to reverse-engineer the protocols used to control the PC Controller and Zniffer USB devices. This allowed the analyst to reverse-engineer the serial communications, using the reversed binary applications as a reference, and develop his own tool-set for communicating with the devices.				
Attack Conditions	An attacker would need access to the application binaries and would need to be familiar with reverse engineering applications and communications.				
Business Impact	This issue would make it easier to reverse-engineer the application or the protocol. Information gained in this manner may be used to bypass various checks or even repurpose Sigma Designs' applications. In addition to this, it may also result in the loss of Sigma Designs' intellectual property.				
Recommendations	Whilst code obfuscation does not necessarily render applications immune to reverse-engineering, it does make it considerably more difficult. As such, it is highly recommended that applications be obfuscated prior to distribution. Dotfuscator is provided with Visual Studio, and would provide an adequate level of protection. The .NET assemblies provided with the Z-Wave development kit are experimental. In real-world deployments, controllers would be deployed on secure systems, and end users would not have easy access to application binaries.				
Attack Example	The following series of images displays unobfuscated application binaries, decoding of serial communications and a toolset implemented by the analyst.				



```
mscorlib
System
System.Core
System.Xml
System.Data
System.Web
System.Drawing
System.Windows.Forms
System.ServiceModel
System.Workflow.ComponentModel
System.Workflow.Runtime
System.Workflow.Activities
ZWaveZnifferUI
ZWaveZniffer
WPFToolkit
Utils
DatagramView
ZWaveControllerUI
ZWaveController
ZWaveBasicApplication
ZWave
  ZWave.dll
    References
    {} -
    {} ZWave
    {} ZWave.Devices
    {} ZWave.Enums
    {} ZWave.Exceptions
    {} ZWave.Layers
    {} ZWave.Layers.Application
    {} ZWave.Layers.Frame
    {} ZWave.Layers.Session
      DeferredMode
      SessionClient
      SessionLayer
    {} ZWave.Layers.Transport
```

Disassembler

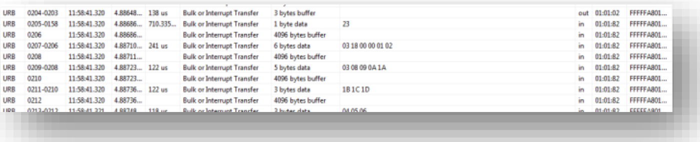
```
public class SessionClient : ISessionClient, IDisposable
{
    // Fields
    [CompilerGenerated]
    private ApiTypes <ApiType>k__BackingField;
    [CompilerGenerated]
    private bool <IsHandleFrameEnabled>k__BackingField;
    [CompilerGenerated]
    private string <LogEntryPointClass>k__BackingField;
    [CompilerGenerated]
    private List<ActionBase> <RunningActions>k__BackingField;
    [CompilerGenerated]
    private Action<ActionHandlerResult> <SendFramesCallback>k__BackingField;
    [CompilerGenerated]
    private byte <SessionId>k__BackingField;
    [CompilerGenerated]
    private bool <SuppressDebugOutput>k__BackingField;
    private IConsumerQueue<ActionBase> mActionBufferBlock;
    private Action<ActionToken> mActionChangeCallback;
    private IConsumerQueue<ActionBase> mCallbackBufferBlock;
    private IActionCheckPoint mCheckPoint;
    private IConsumerQueue<IActionCase> mDataFramesConsumer;
    private byte mFuncIdCounter;
    private readonly object mLockObject;
    private List<ISubstituteManager> mSubstituteManagers;
    private ITimeoutManager mTimeoutManager;

    // Methods
    public SessionClient(Action<ActionToken> actionChangeCallback);
    public void AddSubstituteManager(ISubstituteManager sm, params ActionBase[] actions);
```

Analyzer

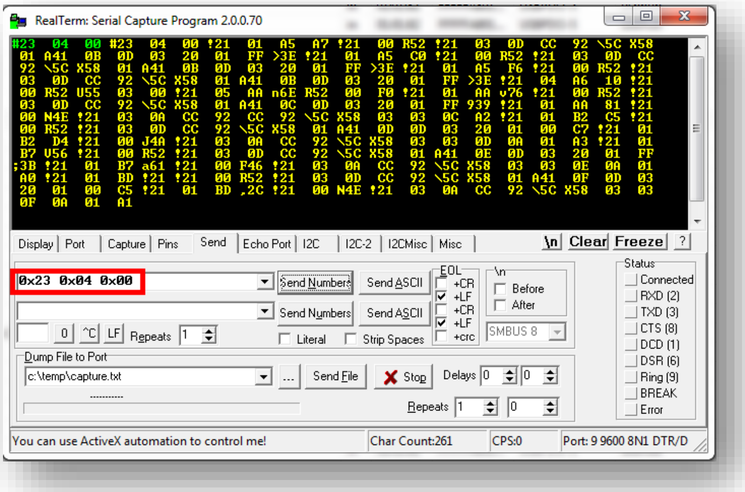
ZWave.Layers.Session.SessionClient

Figure 3 – Unobfuscated Application Binaries



URB	0204-0203	11:58:41.320	4.88648...	138 us	Bulk or Interrupt Transfer	3 bytes buffer		out	01:01:02	FFFFFA801...
URB	0205-0158	11:58:41.320	4.88686...	710.335...	Bulk or Interrupt Transfer	1 byte data	23	in	01:01:82	FFFFFA801...
URB	0206	11:58:41.320	4.88686...		Bulk or Interrupt Transfer	4096 bytes buffer		in	01:01:82	FFFFFA801...
URB	0207-0206	11:58:41.320	4.88710...	241 us	Bulk or Interrupt Transfer	6 bytes data	03 18 00 00 01 02	in	01:01:82	FFFFFA801...
URB	0208	11:58:41.320	4.88711...		Bulk or Interrupt Transfer	4096 bytes buffer		in	01:01:82	FFFFFA801...
URB	0209-0208	11:58:41.320	4.88723...	122 us	Bulk or Interrupt Transfer	5 bytes data	03 08 09 0A 1A	in	01:01:82	FFFFFA801...
URB	0210	11:58:41.320	4.88723...		Bulk or Interrupt Transfer	4096 bytes buffer		in	01:01:82	FFFFFA801...
URB	0211-0210	11:58:41.320	4.88736...	122 us	Bulk or Interrupt Transfer	3 bytes data	1B 1C 1D	in	01:01:82	FFFFFA801...
URB	0212	11:58:41.320	4.88736...		Bulk or Interrupt Transfer	4096 bytes buffer		in	01:01:82	FFFFFA801...
URB	0213-0212	11:58:41.321	4.88748...	118 us	Bulk or Interrupt Transfer	3 bytes data	04 05 06	in	01:01:82	FFFFFA801...

Figure 4 – Sniffing USB Communications to determine protocol



RealTerm: Serial Capture Program 2.0.0.70

```
#23  04  00 #23  04  00 !21  01  A5  A7 !21  00 R52 !21  03  0D  CC  92 \5C X58
 01 A41  0B  0D  03  20  01  FF >3E !21  01  A5  C0 !21  00 R52 !21  03  0D  CC
 92 \5C X58  01 A41  0B  0D  03  20  01  FF >3E !21  01  A5  F6 !21  00 R52 !21
 03  0D  CC  92 \5C X58  01 A41  0B  0D  03  20  01  FF >3E !21  04  A6  10 !21
 00 R52 U55  03  00 !21  05  AA n6E R52  00  F0 !21  01  AA v76 !21  00 R52 !21
 03  0D  CC  92 \5C X58  01 A41  0C  0D  03  20  01  FF 939 !21  01  AA  81 !21
 00 N4E !21  03  0A  CC  92  CC  92 \5C X58  03  03  0C  A2 !21  01  B2  C5 !21
 00 R52 !21  03  0D  CC  92 \5C X58  01 A41  0D  0D  03  20  01  00  C7 !21  01
 B2  D4 !21  00 J4A !21  03  0A  CC  92 \5C X58  03  03  0D  0A  01  A3 !21  01
 B7 V56 !21  00 R52 !21  03  0D  CC  92 \5C X58  01 A41  0E  0D  03  20  01  FF
;3B !21  01  B7 a61 !21  00 F46 !21  03  0A  CC  92 \5C X58  03  03  0E  0A  01
 A0 !21  01  BD !21 !21  00 R52 !21  03  0D  CC  92 \5C X58  01 A41  0F  0D  03
 20  01  00  C5 !21  01  BD ,2C !21  00 N4E !21  03  0A  CC  92 \5C X58  03  03
 0F  0A  01  A1
```

0x23 0x04 0x00

Repeats 1

EOL: ☐ +CR ☑ +LF ☐ +CR ☑ +LF ☐ +crc

\n: ☐ Before ☐ After

☐ Literal ☐ Strip Spaces

SMBUS 8

Dump File to Port: c:\temp\capture.txt

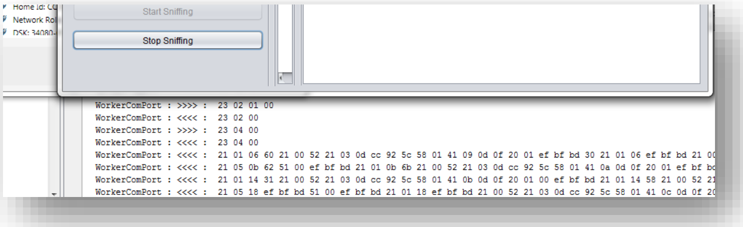
Delays 0 0

Repeats 1 0

Status: ☐ Connected ☐ RXD (2) ☐ TXD (3) ☐ CTS (8) ☐ DCD (1) ☐ DSR (6) ☐ Ring (9) ☐ BREAK ☐ Error

You can use ActiveX automation to control me! | Char Count:261 | CPS:0 | Port: 9 9600 8N1 DTR/D

Figure 5 – Initiating capture (command highlighted) with RealTerm



```
WorkerComPort : >>>> :  23 02 01 00
WorkerComPort : <<<< :  23 02 00
WorkerComPort : >>>> :  23 04 00
WorkerComPort : <<<< :  23 04 00
WorkerComPort : <<<< :  21 01 06 60 21 00 52 21 03 0d cc 92 5c 58 01 41 09 0d 0f 20 01 ef bf bd 30 21 01 06 ef bf bd 21 00
WorkerComPort : <<<< :  21 05 0b 62 51 00 ef bf bd 21 01 0b 6b 21 00 52 21 03 0d cc 92 5c 58 01 41 0a 0d 0f 20 01 ef bf bd
WorkerComPort : <<<< :  21 01 14 31 21 00 52 21 03 0d cc 92 5c 58 01 41 0b 0d 0f 20 01 00 ef bf bd 21 01 14 58 21 00 52 21
WorkerComPort : <<<< :  21 05 18 ef bf bd 51 00 ef bf bd 21 01 18 ef bf bd 21 00 52 21 03 0d cc 92 5c 58 01 41 0c 0d 0f 20
```

Figure 6 – Sniffing traffic via custom tool-set

6 CONCLUSION

Sigma Designs' *Security 2 Command Class* specification appears to have a robust security posture. Concerns identified during previous assessments have all been adequately addressed. The manner in which PANs are used in both multicast and singlecast communication would be effective in preventing replay attacks. Furthermore, the constraints implemented in the activation and verification of devices, as well as the grouping of different device types dependant on the security of the group they belong to would make it difficult for an attacker (even with physical access to a connected device) to attack other nodes on the Z-Wave network.

Furthermore, the technical phases of the various projects also highlighted few issues, with the potential protocol downgrade attack being mitigated by various factors.

Appendix A. Further Details and Examples

Appendix A.1. Potential Protocol Downgrade Vulnerability

In the document provide to SensePost by Sigma Designs – *SensePost Security Assessment Devkit Guide* – page five lists a known bug present within the devices / software when including a *Security 2 Command Class* device. The relevant section states:

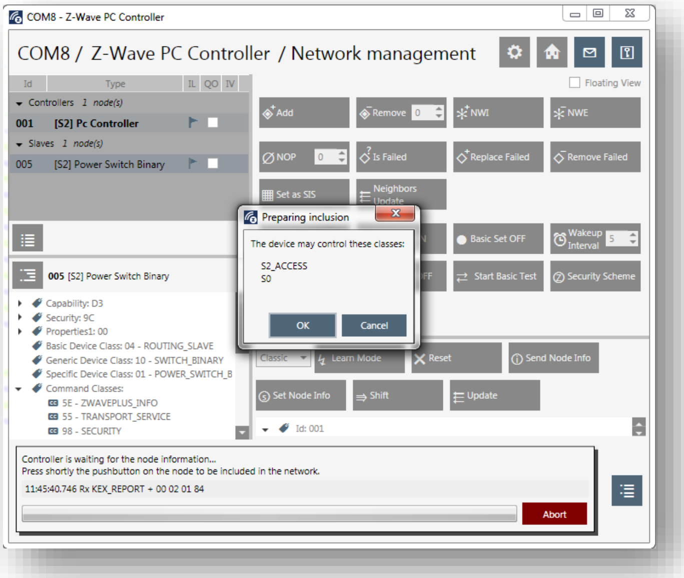
“Note: Due to a bug, this step must be completed in less than 10 seconds, otherwise inclusion will fail”.

In order to ensure that a device capable of supporting the *Security 2 Command Class* is included to the network as a *Security 0 Command Class* device, the following steps can be taken.

1: In the PC Controller Application, click on the Add button.

2: Press the inclusion button on the Secure Door lock device three times in order to initiate the inclusion of the device.

3: At this point, the following image will be displayed.



COM8 - Z-Wave PC Controller

COM8 / Z-Wave PC Controller / Network management

Id	Type	IL	QO	IV
Controllers 1 node(s)				
001	[S2] Pc Controller			
Slaves 1 node(s)				
005	[S2] Power Switch Binary			

005 [S2] Power Switch Binary

- Capability: D3
- Security: 9C
- Properties1: 00
- Basic Device Class: 04 - ROUTING_SLAVE
- Generic Device Class: 10 - SWITCH_BINARY
- Specific Device Class: 01 - POWER_SWITCH_B
- Command Classes:
  - 5E - ZWAVEPLUS_INFO
  - 55 - TRANSPORT_SERVICE
  - 98 - SECURITY

Preparing inclusion

The device may control these classes:

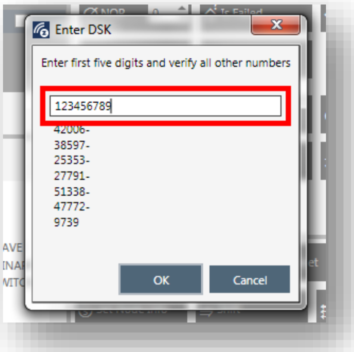
S2_ACCESS
S0

Controller is waiting for the node information...
Press shortly the pushbutton on the node to be included in the network.

```
11:45:40.746 Rx KEX_REPORT + 00 02 01 84
```

Figure 7 – Including *Security 2 Command* Class Device

4: The next step would be to input the DSK. As per the documentation, this has to be entered within ten seconds as a result of the known bug. In this case, we wait for longer than ten seconds in order to see whether we can leverage off the known bug. After a period greater than ten seconds, we enter the DSK. This can be entered correctly or incorrectly. Either option would result in the device being included as a *Security 0 Command Class* device.



Enter DSK

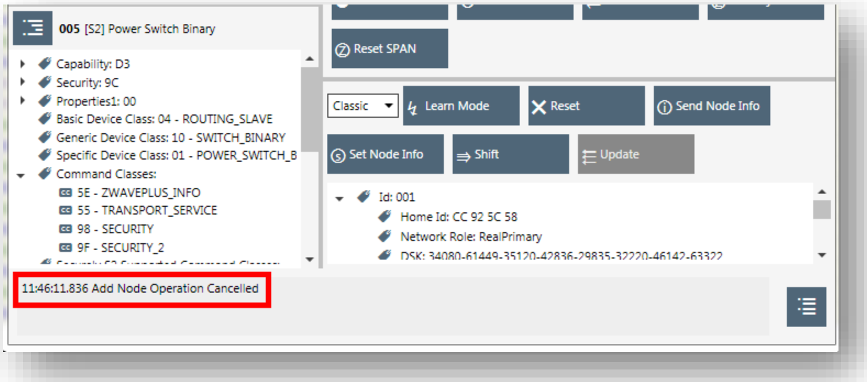
Enter first five digits and verify all other numbers

123456789

42006-
38597-
25353-
27791-
51338-
47772-
9739

Figure 8 - Incorrect DSK Entered

5: Since the PC Controller will not receive a response and successfully include the device, we abort the operation.



005 [S2] Power Switch Binary

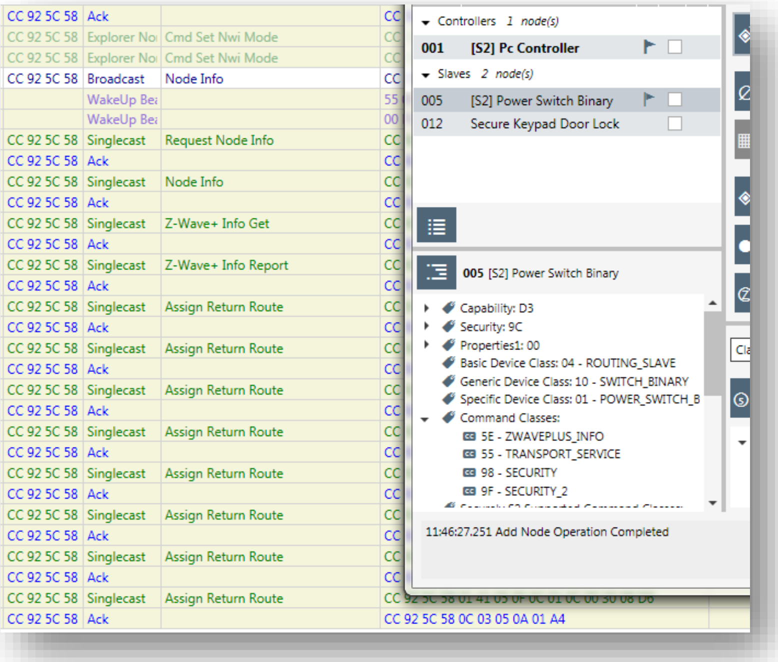
- Capability: D3
- Security: 9C
- Properties1: 00
- Basic Device Class: 04 - ROUTING_SLAVE
- Generic Device Class: 10 - SWITCH_BINARY
- Specific Device Class: 01 - POWER_SWITCH_B
- Command Classes:
  - 5E - ZWAVEPLUS_INFO
  - 55 - TRANSPORT_SERVICE
  - 98 - SECURITY
  - 9F - SECURITY_2

- Id: 001
  - Home Id: CC 92 5C 58
  - Network Role: RealPrimary
  - DSK: 34080-61449-35120-42836-29835-32220-46142-63322

```
11:46:11.836 Add Node Operation Cancelled
```

Figure 9 – Inclusion Aborted

6: We once again click on the Add button in the PC Controller application and press the inclusion button on the door lock three times. The device is now included in the network.



```
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Explorer No  Cmd Set Nwi Mode       CC
CC 92 5C 58  Explorer No  Cmd Set Nwi Mode       CC
CC 92 5C 58  Broadcast    Node Info              CC
             WakeUp Be:                          55
             WakeUp Be:                          00
CC 92 5C 58  Singlecast   Request Node Info      CC
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Singlecast   Node Info              CC
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Singlecast   Z-Wave+ Info Get       CC
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Singlecast   Z-Wave+ Info Report    CC
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Singlecast   Assign Return Route    CC
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Singlecast   Assign Return Route    CC
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Singlecast   Assign Return Route    CC
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Singlecast   Assign Return Route    CC
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Singlecast   Assign Return Route    CC
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Singlecast   Assign Return Route    CC
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Singlecast   Assign Return Route    CC
CC 92 5C 58  Ack                                 CC
CC 92 5C 58  Singlecast   Assign Return Route    CC 92 5C 58 01 41 05 0F 0C 01 0C 00 30 08 D6
CC 92 5C 58  Ack                                 CC 92 5C 58 0C 03 05 0A 01 A4
```

Controllers 1 node(s)

001 [S2] Pc Controller □

Slaves 2 node(s)

005 [S2] Power Switch Binary □

012 Secure Keypad Door Lock □

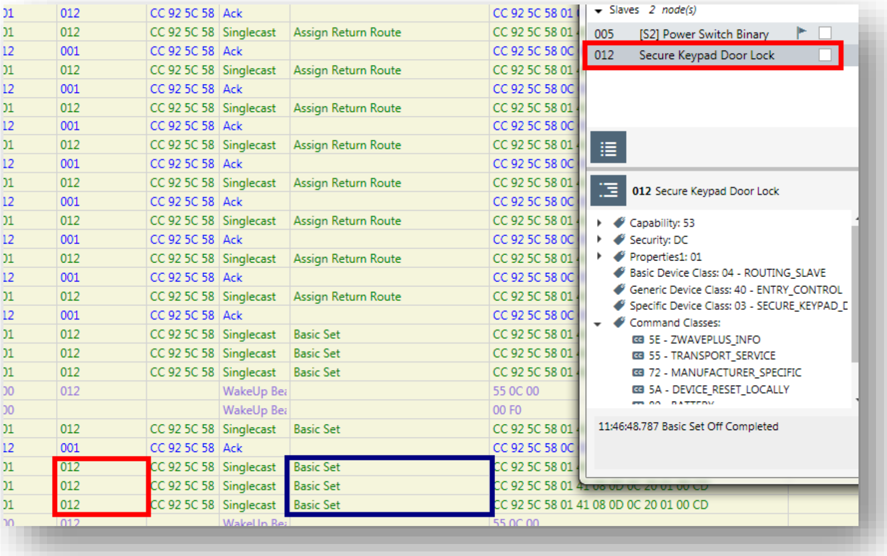
005 [S2] Power Switch Binary

- Capability: D3
- Security: 9C
- Properties1: 00
- Basic Device Class: 04 - ROUTING_SLAVE
- Generic Device Class: 10 - SWITCH_BINARY
- Specific Device Class: 01 - POWER_SWITCH_B
- Command Classes:
  - 5E - ZWAVEPLUS_INFO
  - 55 - TRANSPORT_SERVICE
  - 98 - SECURITY
  - 9F - SECURITY_2

11:46:27.251 Add Node Operation Completed

Figure 10 – Including the node for the second time

7: Analysis of the communication indicate that the device will now be communicating using *Security 0 Command Class*, as no *Security 2 Command Class* frames are present when communicating with the device.



```
01  012  CC 92 5C 58  Ack                              CC 92 5C 58 01
01  012  CC 92 5C 58  Singlecast  Assign Return Route  CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast  Assign Return Route  CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast  Assign Return Route  CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast  Assign Return Route  CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast  Assign Return Route  CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast  Assign Return Route  CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast  Assign Return Route  CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast  Assign Return Route  CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast  Basic Set            CC 92 5C 58 01
01  012  CC 92 5C 58  Singlecast  Basic Set            CC 92 5C 58 01
01  012  CC 92 5C 58  Singlecast  Basic Set            CC 92 5C 58 01
00  012               WakeUp Be:                       55 0C 00
00                    WakeUp Be:                       00 F0
01  012  CC 92 5C 58  Singlecast  Basic Set            CC 92 5C 58 01
12  001  CC 92 5C 58  Ack                              CC 92 5C 58 0C
01  012  CC 92 5C 58  Singlecast  Basic Set            CC 92 5C 58 01
01  012  CC 92 5C 58  Singlecast  Basic Set            CC 92 5C 58 01 41 08 0D 0C 20 01 00 CD
01  012  CC 92 5C 58  Singlecast  Basic Set            CC 92 5C 58 01 41 08 0D 0C 20 01 00 CD
00  012               WakeUp Be:                       55 0C 00
```

Slaves 2 node(s)

005 [S2] Power Switch Binary □

012 Secure Keypad Door Lock □

012 Secure Keypad Door Lock

- Capability: 53
- Security: DC
- Properties1: 01
- Basic Device Class: 04 - ROUTING_SLAVE
- Generic Device Class: 40 - ENTRY_CONTROL
- Specific Device Class: 03 - SECURE_KEYPAD_D
- Command Classes:
  - 5E - ZWAVEPLUS_INFO
  - 55 - TRANSPORT_SERVICE
  - 72 - MANUFACTURER_SPECIFIC
  - 5A - DEVICE_RESET_LOCALLY

11:46:48.787 Basic Set Off Completed

Figure 11 – Communication with Node 12 (highlighted red). No Security 2 Command Class packets seen (highlighted dark blue)

Appendix B. Risk Rating System

Appendix B.1. CVSS3: An Open Standard for Vulnerability Scoring

The Common Vulnerability Scoring System (version 3) is an established method for scoring technical vulnerabilities identified in systems.

The CVSS3 is based on three metric groups:

- Base Metric Group: “represents the intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments.” It covers metrics relating to the complexity (proximity of attacker, authentication requirements) of the attack and its impact on the security qualities of the system (confidentiality, integrity and availability).
- Temporal Metric Group: “represents the characteristics of a vulnerability that change over time but not among user environments.” It covers metrics relating to the current state of the vulnerability (exploitability and remediation options) and to the confidence of the issue at hand.
- Environmental Metric Group: “represents the characteristics of a vulnerability that are relevant and unique to a particular user's environment”. These metrics allow the client to ensure that the controls in place are factored in to the assessment of the vulnerabilities actual relationship with the environment, leading to a more accurate representation of the technical risk.

During an assessment only the base metric group is calculated for each vulnerability. By request, and provided with additional information, the temporal and environmental metric groups can be calculated.

For further information on the CVSS3 system, see the following reference site:

- http://www.first.org/cvss/user-guide

Appendix B.2. Qualitative Severity Rating Scale (QSR)

The Qualitative Severity Rating (QSR) used by SensePost follows the CVSS3 guidelines and allows for a textual representation of the CVSS3 scores and also provides an intuitive means of communicating an understanding of the risk to non-technical stakeholders. The model is a simple ranking of issues from Low to Critical, in descending order of severity.

The following table provides an explanation of each level.

BIR	Description
Critical	Successful attacks within this category could result in an attacker gaining access to view, modify or destroy highly confidential information; conduct or falsify large numbers of unauthorised financially sensitive operations (e.g. falsification of financial transactions, deletion of data records), or lead to a complete compromise of the target. Such attacks could have a catastrophic impact on the confidentiality, integrity and availability of the systems and the business. This could result in a significant financial loss, significant reputational damage, serious legal and compliance related fines and other effects on the business. An immediate remediation plan should be developed to address issues rated at this level.
High	Successful attacks within this category could result in an attacker gaining access to view, modify or destroy confidential information; conduct or falsify unauthorised financially sensitive operations (e.g. falsification of financial transactions, deletion of data records), or lead to significant compromise of the target. Such attacks could have a significant impact on the confidentiality, integrity and availability of the systems and the business. This could result in a significant financial loss, significant reputational damage, serious legal and compliance related fines and other effects on the business. An immediate remediation plan should be developed to address issues rated at this level.
Medium	A Medium BIR could lead to a noticeable impact on the business. Successful attacks within this category could allow an attacker to gain access to sensitive information or to private (personal) records, or could cause the system to perform unauthorised, but non-business critical, operations, or could lead to a significant outage of services. Such attacks could have a noticeable impact on the confidentiality, integrity and availability of the systems and the business, which could result in a noticeable financial loss, considerable reputational damage, legal and compliance related fines and other effects on the business. A timely remediation plan should be developed to address issues rated at this level. However, business requirements may dictate other actions are appropriate.
Low	A Low BIR is unlikely to have a noticeable impact on the business. However, such issues do not exist in isolation and may be used by an attacker as part of more complicated, blended attack, and should not be dismissed. Issues should be considered both individually and collectively. Issues identified at this level should be addressed as part of normal improvement exercises. However, business requirements may dictate other actions are appropriate.

Table 3 – Business Impact Rating

Appendix B.3. Mapping CVSS3 to the Qualitative Rating Scale

The following table provides a mapping of CVSS3 metric scores to each QSV, and follows the CVSS3 guidelines:

QSR	CVSS3 Range
Critical	9.0-10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9

Appendix B.4. Your Risk Methodology

The QSRs and CVSS3 ratings provided in this report do not constitute a complete business risk assessment. SensePost analysts rarely have sufficient information to conduct a company-specific risk assessment. This would require more information than is typically available for such projects, such as knowledge of Sigma Designs' risk appetite.

SensePost recommends that the information communicated via QSVs and CVSS3 metrics be used as input into the business risk methodology. However, where possible, SensePost analysts can assist in the assessment of the identified risks and how they should be interpreted by the business should this be required.

Appendix C. Methodologies

SensePost follows a number of methodologies when conducting security assessments. These methodologies are based on our extensive assessment experience and include a large amount of information.

In order to keep the length of this report to a manageable level, all of the current methodologies used by SensePost analysts can be viewed at http://sensepost.com/assessments/methodologies.pdf

Appendix D. Document Management

Appendix D.1. Document Information

Project Title	Security 2 Command Class Protocol Review for Sigma Designs
Project Code	SP02508
Project Type	Security 2 Command Class Protocol Review
Report Date	2017-08-18
Document Name	SP02508-Sigma-Designs-Security2-Command-Class
Author	Ian de Villiers

Appendix D.2. Change Management

Ver.	Date	Summary
0.1	2017-06-27	First Draft
0.2	2017-06-27	Technical Peer Review
0.3	2017-06-27	Second Draft
0.3	2017-06-28	Quality Assurance
1.0	2017-06-28	Final
1.0	2017-06-28	Final version released to Sigma Designs.
1.1	2017-07-14	First Draft – Technical Testing Information Added
1.2	2017-07-17	Technical Peer Review
1.3	2017-07-17	Second Draft – Technical Testing Information Added
1.4	2017-07-17	Quality Assurance
2.0	2017-07-17	Final Draft
2.0	2017-07-17	Final version released to Sigma Designs.