hard.

***CVE-2015-1790 Impact analysis on WR SSL and test approach***

**Prepared For RICOH**

**CONFIDENTIAL**

Version: 1.00

Date: Aug 28, 2015

Version: 1.00

Wind River Systems, Inc.

9855 Scranton Road, Building 5

San Diego, CA 92121

858-824-3100 phone

858-824-3198 fax

[www.windriver.com](http://www.windriver.com)

Copyright

Unless otherwise agreed in writing, all copyright and intellectual property rights embodied in this document are and shall remain the property of Wind River Systems, Inc. This document is provided solely for the purposes of evaluating the work proposed and no other rights whatsoever to use the information herein are granted. The contents of this document may not be disclosed to any third party without the prior written consent of Wind River Systems, Inc.

Trademarks

Wind River, the Wind River logo, Tornado, and VxWorks are registered trademarks of Wind River Systems, Inc. Any third party trademarks referenced are the property of their respective owners. For further information regarding Wind River trademarks, please see <http://www.windriver.com/company/terms/trademark.html>.

| Revision History | | | |
| --- | --- | --- | --- |
| Date | Version | By | Description of Change |
| Aug 28, 2015 | 0.01 |  | First Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Overview 5](#_Toc428683621)

[1.1 Purpose and Scope 5](#_Toc428683622)

[1.2 Applicable Documents 5](#_Toc428683623)

[1.3 Glossary 5](#_Toc428683624)

[2 Impact analysis and Test Approach 6](#_Toc428683625)

[2.1 Vulnerability Summary 6](#_Toc428683626)

[2.2 Details of Vulnerability & Method of Exploit 6](#_Toc428683627)

[2.2.1 Background 6](#_Toc428683628)

[2.2.2 Method of Exploit 7](#_Toc428683629)

[2.3 Implementation Behavior – OpenSSL & Wind River SSL 7](#_Toc428683630)

[2.3.1 OpenSSL 0.9.8zf 7](#_Toc428683631)

[2.3.2 OpenSSL 0.9.8zg 7](#_Toc428683632)

[2.3.3 Wind River – SSL 8](#_Toc428683633)

[2.3.4 Required Actions on WR-SSL 8](#_Toc428683634)

[2.3.5 Test Approach 8](#_Toc428683635)

[2.4 Summary/Conclusion 9](#_Toc428683636)

[2.5 Attachments 9](#_Toc428683637)

# Overview

## Purpose and Scope

The Purpose of this Document is to explain details of impact analysis carried out on Wind River SSL stack used by RICOH for security vulnerabilities as described in CVE-2015-1790.

Document also includes details of fix released by OpenSSL community, applicability and methods of test and verification of same under Wind River SSL.

## Applicable Documents

The following documents are referenced within:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Document | Version | Scope |
| 1 | CVE-2015-1790\_Test\_Report1 | 0.01 | Report1: test results for WR-SSL as server WITHOUT Fix, OpenSSL 1.0.1 as client, CRAFTED client certificate |
| 2 | CVE-2015-1790\_Test\_Report2 | 0.01 | Report2: test results for WR-SSL as server WITH Fix, OpenSSL 1.0.1 as client, CRAFTED client certificate |
| 3 | CVE-2015-1790\_Test\_Report3 | 0.01 | Report 3: test results for WR-SSL as server WITH Fix, OpenSSL 1.0.1 as client, NOT CRAFTED (good) client certificate |
|  |  |  |  |

## Glossary

|  |  |
| --- | --- |
| Term | Definition |
| WR | Wind River |
| SSL | Secure Sockets Layer |
| TLS | Transport Layer Security |
|  |  |
|  |  |

# Impact analysis and Test Approach

## Vulnerability Summary

CVE-2015-1790 is security vulnerability discovered in the OpenSSL implementation in June 2015 timeframe.

CVE-2015-1790 uncovers the security threat that PKCS7\_dataDecode function in crypto/pkcs7/pk7\_doit.c in OpenSSL before 0.9.8zg, 1.0.0 branch before 1.0.0s, 1.0.1 branch before 1.0.1n, and 1.0.2 branch before 1.0.2b allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) via a PKCS#7 blob that uses ASN.1 encoding and lacks inner EncryptedContent data.

Details of the vulnerability are described in following section.

## Details of Vulnerability & Method of Exploit

### Background

PKCS7 is a public-key cryptography standard by RSA Securities to sign and/or encrypt messages under PKI. The standard describes a general syntax for data that may have cryptography applied to it, such as digital signatures and digital envelopes. OpenSSL provides PKCS7 functions for use by applications. Two such functions are PKCS7\_encrypt and PKCS7\_decrypt which performs encryption and decryption of a given content using a public key in the certificate. These functions internally use pkcs7 parsing functions. PKCS7\_dataDecode is one of the parsing functions used by PKCS7\_decrypt.

General structure of PKCS#7 digital envelope is as given below:

EnvelopedData ::= SEQUENCE {

version Version,

recipientInfos RecipientInfos,

encryptedContentInfo EncryptedContentInfo }

RecipientInfos ::= SET OF RecipientInfo

EncryptedContentInfo ::= SEQUENCE {

contentType ContentType,

contentEncryptionAlgorithm

ContentEncryptionAlgorithmIdentifier,

encryptedContent

[0] IMPLICIT EncryptedContent OPTIONAL }

EncryptedContent ::= OCTET STRING

### Method of Exploit

PKCS7\_dataDecode() function takes a PKCS7 object to be decoded, key and certificate as inputs and returns decoded output of PKCS7 object.

PKCS7 object follows the syntax as outlined in previous section. Since in some cases content is allowed to be streamed, EncryptedContent part of the object is optional. An attacker can make use of this property and exploit weakness in parsing code in function PKCS7\_dataDecode().

When exploited successfully, application calling the PKCS7\_decrypt may crash because of a null pointer dereference.

Details of weakness in parsing code are explained in the following section.

## Implementation Behavior – OpenSSL & Wind River SSL

### OpenSSL 0.9.8zf

PKCS7\_dataDecode function checks the input content type and sets the variable data\_body to the content to be decrypted, however it is expected that if the data\_body happens to be NULL the input data should be available through in\_bio argument to the function. However the code is not strict in checking if in\_bio too is NULL.

Following code fragments explain the vulnerability.

|  |  |
| --- | --- |
| Code blocks (not continuous) | Description/Analysis |
| case NID\_pkcs7\_enveloped:  data\_body = p7->d.enveloped->enc\_data->enc\_data; | data\_body can be NULL here but not checked |
| case NID\_pkcs7\_signed:  data\_body = PKCS7\_get\_octet\_string(p7->d.sign->contents); | data\_body can be NULL here but not checked |
| case NID\_pkcs7\_signedAndEnveloped:  data\_body = p7->d.signed\_and\_enveloped->enc\_data->enc\_data; | data\_body can be NULL here but not checked |

### OpenSSL 0.9.8zg

Revision to function PKCS7\_dataDecode() to fix CVE-2015-1790 vulnerability introduces a NULL check for data\_body before choosing to use in\_bio.

Details of code change are as below:

|  |  |
| --- | --- |
| Code | Description/Analysis |
| i = OBJ\_obj2nid(p7->type);  switch (i) {  case NID\_pkcs7\_signed:  ..  data\_body = PKCS7\_get\_octet\_string(p7->d.sign->contents);  break;  case NID\_pkcs7\_signedAndEnveloped:  ..  data\_body = p7->d.signed\_and\_enveloped->enc\_data->enc\_data;  ..  }  break;  case NID\_pkcs7\_enveloped:  ..  data\_body = p7->d.enveloped->enc\_data->enc\_data;  ..  goto err;  }  **+ /\* Detached content must be supplied via in\_bio instead. \*/**  **+ if (data\_body == NULL && in\_bio == NULL) {**  **+ PKCS7err(PKCS7\_F\_PKCS7\_DATADECODE, PKCS7\_R\_NO\_CONTENT);**  **+ goto err;**  **+ }** | Code in blue bold is newly added (starting with ‘+’)  Newly added code checks if data\_body and in\_bio are both null and returns ERRORs if so. This saves the application from null pointer dereference and potential crash. |

### Wind River – SSL

For this CVE, relevant part of WR-SSL implementation is same OpenSSL 0.9.8zg.

Hence all changes made by OpenSSL team to fix the CVE are applicable to WR-SSL as well.

### Required Actions on WR-SSL

Required code changes for WR-SSL are same as those changed between OpenSSL0.9.8zf nad OpenSSL0.9.8zg. See section 2.3.2 for an outline of major changes.

### Test Approach

**Test Scenario 1 (Report 1):**

* Perform connection Test: WR-SSL WITHOUT Fix, run test application cve\_1790app with cve1790\_Not\_crafted\_input file as input.
* Expected Result – Application should decrypt the input file successfully and print results on console.
* Evidence: console output

**Test Scenario 2 (Report 2):**

* Perform connection Test: WR-SSL WITHOUT Fix, run test application cve\_1790app with cve1790\_crafted\_input file as input.
* Expected Result – Application should decrypt the input file successfully but get into null pointer problem.
* Evidence: console output

**Test Scenario 3 (Report 3):**

* Perform connection Test: WR-SSL WITH Fix, run test application cve\_1790app with cve1790\_crafted\_input file as input.
* Expected Result – decrypt should fail and return with error message. No null pointer problem.
* Evidence: console output

**Test Scenario 4 (Report 4):**

* Perform connection Test: WR-SSL WITH Fix, run test application cve\_1790app with cve1790\_Not\_crafted\_input file as input.
* Expected Result – Application should decrypt the input file successfully and print results on console.
* Evidence: console output

## Summary/Conclusion

Detailed examination of the WR-SSL implementation reveals that WR-SSL is vulnerable to the security issue described in CVE-2015-1790.

Required code changes were made and tests performed.

## Attachments

1. Report 1: WR-SSL WITHOUT Fix, run test application cve\_1790app with cve1790\_Not\_crafted\_input file as input
2. Report 2: WR-SSL WITHOUT Fix, run test application cve\_1790app with cve1790\_ crafted\_input file as input
3. Report 3: WR-SSL WITH Fix, run test application cve\_1790app with cve1790\_\_crafted\_input file as input
4. Report 4: WR-SSL WITH Fix, run test application cve\_1790app with cve1790\_Not\_crafted\_input file as input

/EOD