hard.

***CVE-2015-0204 – Impact analysis on WR SSL and test approach***

**Prepared For RICOH**

**CONFIDENTIAL**

Version:

Date :

Version:

Date:

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 |
| May 8, 2015 | 1.00 | Wind River | Official Release. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Overview 5](#_Toc418917517)

[1.1 Purpose and Scope 5](#_Toc418917518)

[1.2 Applicable Documents 5](#_Toc418917519)

[1.3 Glossary 5](#_Toc418917520)

[2 Impact analysis and Test Approach 6](#_Toc418917521)

[2.1 Vulnerability Summary 6](#_Toc418917522)

[2.2 Details of Vulnerability & Method of Exploit 6](#_Toc418917523)

[2.2.1 Background 6](#_Toc418917524)

[2.2.2 Method of Exploit 7](#_Toc418917525)

[2.3 Implementation Behavior – Openssl & Wind River SSL 7](#_Toc418917526)

[2.3.1 OpenSSL 0.9.8zc 7](#_Toc418917527)

[2.3.2 Wind River – SSL 8](#_Toc418917528)

[2.4 Code changes & Impact of OpenSSL Fix 9](#_Toc418917529)

[2.5 Required Actions on WR-SSL 10](#_Toc418917530)

[2.5.1 Code Changes 10](#_Toc418917531)

[2.5.2 Test/verification 10](#_Toc418917532)

[2.6 Summary/Conclusion 10](#_Toc418917533)

[2.7 Attachments 10](#_Toc418917534)

# Overview

## Purpose and Scope

The Purpose of this Document is to explain details of impact analysis carried out on Wind River SSL stack used RICOH for security vulnerabilities as described in CVE-2015-0204.

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. 1 | CVE-2015-0204\_Test\_Report1 | 0.01 | Test plan and different Scenarios  OpenSSL 0.9.8zc handshake success with non-export rsa. |
| 1. 2 | CVE-2015-0204\_Test\_Report2 | 0.01 | Test plan and different Scenarios  OpenSSL 0.9.8zd handshake fail with non-export rsa. |
| 1. 3 | CVE-2015-0204\_Test\_Report3 | 0.01 | Test plan and different Scenarios  WR SSL without fix - handshake success with non-export rsa. |
| 1. 4 | CVE-2015-0204\_Test\_Report4 | 0.01 | Test plan and different Scenarios  WR SSL with fix - handshake fail with non-export rsa. |

## Glossary

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

# Impact analysis and Test Approach

## Vulnerability Summary

An OpenSSL client will accept the use of an RSA temporary key in a non-export RSA key exchange ciphersuite. A server could present a weak temporary key and downgrade the security of the session. Applicable to OpenSSL versions before 0.9.8zd, 1.0.0p and 1.0.1k.

The ssl3\_get\_key\_exchange function in s3\_clnt.c allows remote SSL servers to conduct RSA-to-EXPORT\_RSA downgrade attacks and facilitate brute-force decryption by offering a weak ephemeral RSA key in a noncompliant role. NOTE: the scope of this CVE is only client code based on OpenSSL, not EXPORT\_RSA issues associated with servers.

Details of the vulnerability are described in following section.

## Details of Vulnerability & Method of Exploit

### Background

Brief descriptions about EXPORT\_RSA and RSA are given below.

**EXPORT\_RSA:** Refers to a cipher suite where the RSA key in ServerKeyExchangeMessage, is less than or equal to 512 bits. This is historical as US restricted export of crypto to maximum of 512 bit RSA key length way back in 90s.

**RSA:** Any cipher suite where the ServerKeyExchangeMessage RSA key is more than 512 bits.

if OpenSSL server using a non-export RSA cipher, the RSA key can be used for encryption.

if OpenSSL server using export RSA cipher,

a. RSA public key of length greater than 512 bits can be used for signing.

b. RSA key of 512 bits or shorter (EXPORT) can be used for either encryption or signing.

As per RFC 2246 section 7.4.3,

Note: According to current US export law, RSA moduli larger than 512 bits may not be used for key exchange in software exported from the US. With this message, the larger RSA keys encoded in certificates may be used to sign temporary shorter RSA keys for the RSA\_EXPORT key exchange method. Hence it can be noted that in case of customer, 512 bit RSA key should be used in server key exchange message.

If SSL server is using a temporary rsa key, i.e., ephermal rsa, it must send ServerKeyExchange message. However, In this case, server certificate may have RSA key of strength 512 or 1024 or 2048 bit. ServerKeyExchange messages, as explained above shall have a 512-bit strong RSA key or a weaker RSA key (i.e. lesser than 512 bits).

In such a scenario, a vulnerable openSSL client implementation, will accept weaker RSA key from the ServerKeyExchange instead of using the stronger RSA key in the certificate message. This behavior is referred to as RSA-to-EXPORTRSA downgrade. CVE-2015-0204 is a finding that OpenSSL client implementation is vulnerable to this kind of downgrade.

### Method of Exploit

A openSSL client starts SSL handshake and proceeds as explained in previous section until ServerKeyExchange is received.. At this point, server’s ServerKeyExchange message may have temporary rsa or ephemeral rsa which is maximum 512 bit length whereas Certificate can contain stronger (larger than 512-bit) RSA. Client will accept and use key from the ServeKeyExchange even if it is weaker key compared to RSA key in certificate.

OpenSSL fix to this vulnerability is that client should not accept ServerKeyExchange message in case of non export rsa ciphers(ex:- SSL3\_TXT\_RSA\_NULL\_SHA) thus preventing RSA-to-RSA\_EXPORT downgrade.

## Implementation Behavior – Openssl & Wind River SSL

### OpenSSL 0.9.8zc

OpenSSL uses state machine to implement the handshake part. This logic works following way.

*Note: state machine transitions and actions are simplified for easier understanding.*

Server initiates state machine and performs handshake message exchange up to Server Certificate. After receiving these messages, client will change state toSSL3\_ST\_CR\_KEY\_EXCH\_A. In SSL3\_ST\_CR\_KEY\_EXCH\_A state, client invokes ssl3\_get\_key\_exchange() to receive and process ClientKeyExchange packet.

.

Relevant part of ssl3\_get\_key\_exchange() in ssl/s3\_clnt.c,is explained below.

|  |  |  |
| --- | --- | --- |
| Line | Source code | Explanation |
| 324 | ret=ssl3\_get\_key\_exchange(s); | After changing the state ssl3\_get\_key\_exchnage function will be called. |
|  | …. |  |
| 1165 | alg=s->s3->tmp.new\_cipher->algorithms | Getting algorithm type(cipher may be rsa or rsa\_export or something else) |
|  | …. |  |
| 1170 | if (alg & SSL\_kRSA)  {  ....  ....  } | Check if algorithm is rsa or not, if it is rsa it will continue to processing the key incase of both export non export rsa ciphers. |
| 1229 | pkey=X509\_get\_pubkey(s->session->sess\_cert->peer\_pkeys[SSL\_PKEY\_RSA\_ENC].x509); | Get the rsa key sent in the server key exchange message. |
|  |  |  |
| 1488 | if (pkey->type == EVP\_PKEY\_RSA) | **Control is reached here and handshake will be continued with rsa key got from Serverkeyexchange message by downgrading rsa-to-rsa\_export(using smaller size key instead of more secure key in certificate.)** |
|  |  |  |

Above vulnerability is fixed in 0.9.8zd by aborting handshake in case of non-export rsa cipher.

### Wind River – SSL

WR SSL is developed from OpenSSL 0.9.7e base. Implementation related to serverkeyexchange message processing of SSL client is same as OpenSSL 0.9.8zc explained above.

## Code changes & Impact of OpenSSL Fix

Code change to fix the vulnerability is in ssl3\_get\_key\_exchange () in ssl/s3\_clnt.c file. Following table explains the fix.

|  |  |  |
| --- | --- | --- |
| Line | Source code | Explanation |
| 333 | ret=ssl3\_get\_key\_exchange(s); | After changing the state ssl3\_get\_key\_exchnage function will be called. |
|  | …. |  |
| 1019 | alg=s->s3->tmp.new\_cipher->algorithms | Getting algorithm type(cipher may be rsa or rsa\_export or something else) |
|  | …. |  |
| 1170 | if (alg & SSL\_kRSA)  {  if(!SSL\_C\_IS\_EXPORT(s->s3->tmp.new\_cipher))  {  al=SSL\_AD\_UNEXPECTED\_MESSAGE;  SSLerr(SSL\_F\_SSL3\_GET\_KEY\_EXCHANGE,SSL\_R\_UNEXPECTED\_MESSAGE);  goto err;  }  .....  .....  } | If algorithm is rsa, reject connection if cipher is non-export RSA.  Execution continues if cipher is export-RSA. |
| 1076 | pkey=X509\_get\_pubkey(s->session->sess\_cert->peer\_pkeys[SSL\_PKEY\_RSA\_ENC].x509); | Get the rsa key sent in the server key exchange message.(control reaches here only in case of export\_rsa cipher) |
| 1193 | if (pkey->type == EVP\_PKEY\_RSA) | **Handshake will be continued with rsa key received in Serverkeyexchange message.(control reaches here only in case of export\_rsa cipher).** |
|  |  |  |

## Required Actions on WR-SSL

### Code Changes

Code changes are applicable to Wind River SSL version.

### Test/verification

Since vulnerability is applicable to WR SSL, following is the test/verification approach.

1. Create test method and files, verify that test files are correct by testing them with 0.9.8zc and 0.9.8zd

Report 1 – 0.9.8zc handshake success with non-export RSA.

Report 2 – 0.9.8zd handshake fail with non-export RSA.

1. Apply the same test bed, certificates and files to WR SSL. Verify that server accepted by client and confirm that WR SSL is vulnerable.

Report 3 – WR SSL without fix

1. Apply code changes to WR-SSL. To ensure that code changes applicable to the WR SSL. server rejected by the client and WR SSL is not vulnerable to this CVE.

Report 4 – WR SSL with fix

## Summary/Conclusion

It has been found that current version of WR-SSL used by customer is vulnerable to the threat described in CVE-2015-0204, and code changes are applicable. Required code changes are made and tests are performed to confirm that vulnerability is not applicable after applying code changes.

## Attachments

1. Report 1 – OpenSSL 0.9.8zc handshake Success
2. Report 2 – OpenSSL 0.9.8zd handshake Fail.
3. Report 3 – WR SSL without fix - handshake Success
4. Report 4 – WR SSL with fix - handshake Fail.