hard.

***CVE-2014-3572 – Impact analysis on WR SSL and test approach***

**Prepared For RICOH**

**CONFIDENTIAL**

Version: 1.00

Date : May 8, 2015

Version: 1.00

Date: May 8, 2015

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](#_Toc418917648)

[1.1 Purpose and Scope 5](#_Toc418917649)

[1.2 Applicable Documents 5](#_Toc418917650)

[1.3 Glossary 5](#_Toc418917651)

[2 Impact analysis and Test Approach 6](#_Toc418917652)

[2.1 Vulnerability Summary 6](#_Toc418917653)

[2.2 Details of Vulnerability & Method of Exploit 6](#_Toc418917654)

[2.2.1 Background 6](#_Toc418917655)

[2.2.2 Method of Exploit 7](#_Toc418917656)

[2.3 Implementation Behavior – OpenSSL & Wind River SSL 8](#_Toc418917657)

[2.3.1 OpenSSL 0.9.8zc 8](#_Toc418917658)

[2.3.2 Wind River – SSL 9](#_Toc418917659)

[2.4 Code changes & Impact of OpenSSL Fix 9](#_Toc418917660)

[2.5 Required Actions on WR-SSL 10](#_Toc418917661)

[2.5.1 Code Changes 10](#_Toc418917662)

[2.5.2 Test/verification 11](#_Toc418917663)

[2.6 Summary/Conclusion 12](#_Toc418917664)

[2.7 Attachments 12](#_Toc418917665)

# 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-2014-3572.

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 |
|  | CVE-2014-3572\_Test\_Report1 | 0.01 | Test results for 0.9.8zc client – handshake with ECDHE key exchange |
|  | CVE-2014-3572\_Test\_Report2 | 0.01 | Test results for 0.9.8zd client – handshake with ECDHE key exchange |
|  | CVE-2014-3572\_Test\_Report3 | 0.01 | Test results for WR SSL without fix –DHE-RSA cipher |
|  | CVE-2014-3572\_Test\_Report4 | 0.01 | Test results for WR SSL with fix – DHE-RSA cipher |
|  | CVE-2014-3572\_Test\_Report5 | 0.01 | Test results for WR SSL with fix –DH-RSA cipher |

## Glossary

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

# Impact analysis and Test Approach

## Vulnerability Summary

CVE-2014-3572 is security vulnerability discovered in the OpenSSL implementation in Jan 2015 timeframe. Security threat uncovered is that an SSL client will accept a handshake using an ephemeral ciphersuite if the server key exchange message is omitted. This effectively removes forward secrecy from the ciphersuite.

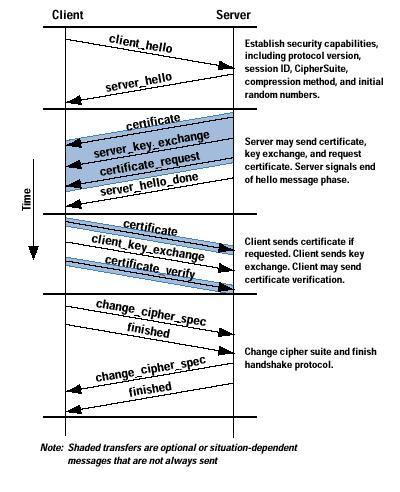
Details of the vulnerability are described in following section.

## Details of Vulnerability & Method of Exploit

### Background

ServerKeyExchange message is optional. Exchanges between a client and server for a SSL handshake that includes ServerKeyExchange are as shown below.

[From <http://shenzi.cs.uct.ac.za/~honsproj/cgi-bin/view/2003/kritzinger_truter.tar.gz/www/duncan/report/node24.html> ]



Purpose of ServerKeyExchange is for SSL server to create a temporary key and send it to client. This key can be used by the client to encrypt the ClientKeyExchange message later in the process. The step is only required when the public key algorithm does not provide the key material necessary to encrypt the ClientKeyExchange message, such as when the server’s certificate does not contain a public key. Common case is a DHE public value using which client can complete a key exchange.

As per RFC 5246 (TLS),

The ServerKeyExchange message is sent by the server only when the

server Certificate message (if sent) does not contain enough data

to allow the client to exchange a premaster secret. This is true

for the following key exchange methods:

DHE\_DSS

DHE\_RSA

DH\_anon

It is not legal to send the ServerKeyExchange message for the

following key exchange methods:

RSA

DH\_DSS

DH\_RSA

Other key exchange algorithms, such as those defined in [TLSECC],

MUST specify whether the ServerKeyExchange message is sent or not;

and if the message is sent, its contents.

Same way, for key exchange involving elliptic curve cryptography, following apply. From RFC 4492,

In ECDH\_ECDSA, the server's certificate MUST contain an ECDH-capable

public key and be signed with ECDSA. A ServerKeyExchange MUST NOT be sent.

In ECDHE\_ECDSA, the server's certificate MUST contain an ECDSA- capable public key and be signed with ECDSA.

### Method of Exploit

Assume that a client starts handshake with SSL server on the ciphersuite ECDHE or DHE. As explained in the above section, server in this case must send SeverKeyExchange immediately after Certificate message (or immediately after ServerHello if the negotiation is anonymous).

However, if the server does not send ServerKeyExchange message, client will downgrade to the non-ephemeral cipher i.e. ECDH or DH. In such a session it won’t be possible to use ephemeral keys and this result in loss of perfect forward secrecy (PFS).

OpenSSL versions 0.9.8zd, 1.0.0p and 1.0.1k include code changes to fix this problem. All versions prior to that with support for ECDHE or DHE are vulnerable.

Following section explain the behavior of various OpenSSL versions and WR-SSL version used by customer under the above scenario of attack.

## Implementation Behavior – OpenSSL & Wind River SSL

### OpenSSL 0.9.8zc

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

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

Client, after sending ClientHello, state is changed to SSL3\_ST\_CR\_SRVR\_HELLO\_A.

In state SSL3\_ST\_CR\_SRVR\_HELLO\_A, ssl3\_get\_server\_hello() function is invoked and state is changed to SSL3\_ST\_CR\_CERT\_A. (Client receive Certificate)

In SSL3\_ST\_CR\_CERT\_A state, if key exchange protocol is DH\_anon, state is changed to SSL3\_ST\_CR\_KEY\_EXCH\_A immediately else function ssl3\_get\_server\_certificate() is called and after that state is changed to SSL3\_ST\_CR\_KEY\_EXCH\_A. (Client receive Key exchange state).

In CR\_KEY\_EXCH state, function ssl3\_get\_key\_exchange() is called to receive and process ServerKeyExchange message.

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

|  |  |  |
| --- | --- | --- |
| Line | Source code | Explanation |
| 1117 | n=s->method->ssl\_get\_message(s,  SSL3\_ST\_CR\_KEY\_EXCH\_A,  SSL3\_ST\_CR\_KEY\_EXCH\_B,  -1,  s->max\_cert\_list,  &ok); | Read Handshake message from network. Any message type will be accepted. First 4 bytes of message will be read in KEY\_EXCH\_A state and remaining message will be read in state EXCH\_B. |
| 1124 | if (!ok) return((int)n); | If failed to read handshake message return. |
| 1126 | if (s->s3->tmp.message\_type != SSL3\_MT\_SERVER\_KEY\_EXCHANGE)  { | Check if received message is ServerKeyExchange or not. |
| 1128 | s->s3->tmp.reuse\_message=1;  return(1);  } /\* if msg type not ServerkeyExchange \*/ | Here, even though msg received is not ServerKeyExchange, handshake is not aborted. |
|  | ….. | Function continues.. |

### Wind River – SSL

WR SSL is developed from OpenSSL 0.9.7e base. Implementation related to ServerKeyExchange processing of SSL client is same as OpenSSL 0.9.8zc.However, ECDHE is not supported in the case of WR-SSL.

## Code changes & Impact of OpenSSL Fix

OpenSSL Fix is in 0.9.8zd. Code change to fix the vulnerability is in ssl3\_get\_key\_exchange() function in ssl/s3\_clnt.c file. Following table explains the fix.

|  |  |  |
| --- | --- | --- |
| Line | Source code | Explanation |
| 1117 | n=s->method->ssl\_get\_message(s,  SSL3\_ST\_CR\_KEY\_EXCH\_A,  SSL3\_ST\_CR\_KEY\_EXCH\_B,  -1,  s->max\_cert\_list,  &ok); | Read Handshake message from network. Any message type will be accepted. First 4 bytes of message will be read in KEY\_EXCH\_A state and remaining message will be read in state EXCH\_B. |
| 1124 | if (!ok) return((int)n); | If failed to read handshake message return. |
| **1126** | **alg=s->s3->tmp.new\_cipher->algorithms;** | **Get type flags of key exchange algorithm.** |
| 1129 | if (s->s3->tmp.message\_type != SSL3\_MT\_SERVER\_KEY\_EXCHANGE)  { | Check if received message is ServerKeyExchange or not. |
| **1131** | **/\***  **\* Can't skip server key exchange if this is**  **\* an ephemeral ciphersuite.**  **\*/**  **if (alg & (SSL\_kEDH|SSL\_kECDHE))**  **{**  **SSLerr(SSL\_F\_SSL3\_GET\_KEY\_EXCHANGE, SSL\_R\_UNEXPECTED\_MESSAGE);**  **al = SSL\_AD\_UNEXPECTED\_MESSAGE;**  **goto f\_err;**  **}** | **If condition above(line 1308) is true indicates that ServerKeyExchange was NOT received. Check if the key exchange algorithm is ephemeral type, and if so, abort handshake here.** |
| 1141 | s->s3->tmp.reuse\_message=1;  return(1);  } /\* if msg type not ServerkeyExchange \*/ | Here, even though msg received is not ServerKeyExchange, handshake is not aborted. |
|  | ….. | Function continues.. |

Following is the impact of code change under different scenarios

|  |  |  |
| --- | --- | --- |
| No | Scenario | Applicability |
| 1 | Server sends ServerKeyExchange | Fix not applicable |
| 2 | Handshake using DHE or ECDHE or DH\_anon.  Server does not send ServerKeyExchange. | **Applicable. Handshake fails. (Success before fix)** |
| 3 | Handshake using DH or ECDH.  Server does not send ServerKeyExchange. | **Applicable. Handshake success.** |

*Note: meaning of applicable or not applicable is that modified part of source code is executed or not in that particular scenario.*

## Required Actions on WR-SSL

### Code Changes

WR SSL is developed from OpenSSL 0.9.7e base. However OpenSSL ECDHE support is starting only from 0.9.8 (0.9.7h). Hence ECDHE is not supported in the case of WR-SSL. As a result, vulnerability as described in CVE-2014-3572 is applicable to DHE functionality only.

OpenSSL fix shall be applied to WR-SSL as described below.

|  |  |  |
| --- | --- | --- |
| Line | Source code | Explanation |
| 1117 | n=s->method->ssl\_get\_message(s,  SSL3\_ST\_CR\_KEY\_EXCH\_A,  SSL3\_ST\_CR\_KEY\_EXCH\_B,  -1,  s->max\_cert\_list,  &ok); | Read Handshake message from network. Any message type will be accepted. First 4 bytes of message will be read in KEY\_EXCH\_A state and remaining message will be read in state EXCH\_B. |
| 1124 | if (!ok) return((int)n); | If failed to read handshake message return. |
| **1126** | **alg=s->s3->tmp.new\_cipher->algorithms;** | **Get type flags of key exchange algorithm.** |
| 1129 | if (s->s3->tmp.message\_type != SSL3\_MT\_SERVER\_KEY\_EXCHANGE)  { | Check if received message is ServerKeyExchange or not. |
| **1131** | **/\***  **\* Can't skip server key exchange if this is**  **\* an ephemeral ciphersuite.**  **\*/**  **if (alg & SSL\_kEDH)**  **{**  **SSLerr(SSL\_F\_SSL3\_GET\_KEY\_EXCHANGE, SSL\_R\_UNEXPECTED\_MESSAGE);**  **al = SSL\_AD\_UNEXPECTED\_MESSAGE;**  **goto f\_err;**  **}** | **If condition above(line 1308) is true indicates that ServerKeyExchange was NOT received. Check if the key exchange algorithm is ephemeral type, and if so, abort handshake here. [Note the absence of SSL\_kECDHE]** |
| 1141 | s->s3->tmp.reuse\_message=1;  return(1);  } /\* if msg type not ServerkeyExchange \*/ | Here, even though msg received is not ServerKeyExchange, handshake is not aborted. |
|  | ….. | Function continues.. |

### Test/verification

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 client – handshake with ECDHE key exchange –handshake success

Report 2 – 0.9.8zd client – handshake with ECDHE key exchange – handshake fail

1. Apply the same test bed environment to WR SSL client with and without fix under following combinations.

|  |  |  |  |
| --- | --- | --- | --- |
| Report | WR-SSL | Cipher | Expected result |
| Report 3 | WR SSL client without fix | DHE-RSA cipher | Client will accept server without server keyexchange message. (But handshake fails because of WRSSL support limitations, client is moving forward without server keyexchange message so it is vulnerable.) |
| Report 4 | WR SSL client with fix | DHE-RSA cipher | Handshake should fail because of code changes.(code changes applicable) |
| Report 5 | WR SSL client with fix | DH-RSA cipher | Handshake should succeed. (To make sure code changes are not affecting normal WRSSL behavior.) |

## Summary/Conclusion

It has been found that current version of WR-SSL used by customer is vulnerable to the threat described in CVE-2015-0205 for DHE ciphersuites. Code changes are made and tests are performed to confirm that vulnerability is fixed and non-ephemeral ciphers are not affected by this code change.

## Attachments

1. Report1 Test results for 0.9.8zc client – handshake with ECDHE key exchange
2. Report2 Test results for 0.9.8zd client – handshake with ECDHE key exchange
3. Report3 Test results for WR SSL without fix –DHE-RSA cipher
4. Report4 Test results for WR SSL with fix – DHE-RSA cipher
5. Report5 Test results for WR SSL with fix –DH-RSA cipher

/EOD