hard.

***CVE-2015-4000 Impact analysis on WR SSL and test approach***

**Prepared For RICOH**

**CONFIDENTIAL**

Version: 1.00

Date: Sep 03, 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 |
| Sep 03, 2015 | 0.01 |  | First Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Overview 5](#_Toc429155251)

[1.1 Purpose and Scope 5](#_Toc429155252)

[1.2 Applicable Documents 5](#_Toc429155253)

[1.3 Glossary 5](#_Toc429155254)

[2 Impact analysis and Test Approach 6](#_Toc429155255)

[2.1 Vulnerability Summary 6](#_Toc429155256)

[2.2 Details of Vulnerability & Method of Exploit 6](#_Toc429155257)

[2.2.1 Background 6](#_Toc429155258)

[2.2.2 Method of Exploit 6](#_Toc429155259)

[2.3 CVE-2015-4000 fix (Protection from Logjam) 7](#_Toc429155260)

[2.4 Implementation Behavior – OpenSSL & Wind River SSL 8](#_Toc429155261)

[2.4.1 OpenSSL 1.0.2b/1.0.1n 8](#_Toc429155262)

[2.4.2 Wind River – SSL - Changes 9](#_Toc429155263)

[2.5 Test Approach 11](#_Toc429155264)

[2.6 Summary/Conclusion 12](#_Toc429155265)

[2.7 Attachments 12](#_Toc429155266)

# 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-2015-4000.

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-2015-4000\_Test\_Report01 to  CVE-2015-4000\_Test\_Report12 | 0.01 | As described in table in section 2.5. |

## Glossary

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

# Impact analysis and Test Approach

## Vulnerability Summary

CVE-2015-4000 is security vulnerability discovered in the SSL/TLS protocol in May 2015 timeframe.

CVE-2015-4000 uncovers the security threat that when a DHE\_EXPORT ciphersuite is enabled on a server but not on a client, does not properly convey a DHE\_EXPORT choice, which allows man-in-the-middle attackers to conduct cipher-downgrade attacks by rewriting a ClientHello with DHE replaced by DHE\_EXPORT and then rewriting a ServerHello with DHE\_EXPORT replaced by DHE, aka the "Logjam" issue.

Details of the vulnerability are described in following section.

## Details of Vulnerability & Method of Exploit

### Background

EXPORT Ciphers are those cipher suites that were designed to be sufficiently weak they could legally be exported from the US back in the 1990s when there were much stricter legal limits on exporting encryption from the US.

DHE cipher suites are those that use a Diffie-Hellman key exchange method with ephemeral keys. DHE\_EXPORT cipher suites, by design, requires that the prime number used, (p) be not longer than 512 bits.

A client, SSL/TLS, informs a list of cipher suites of its preference in the ClientHello packet to server and server responds with a chosen cipher in its response i.e. serverHello. If an DHE\_EXPORT or DHE (non-export) cipher was agreed by these exchanges, server is expected to send DH parameters in a message ServerKeyExchange that follows. However the ServerKeyExchange message does not include any information as to the cipher chosen by earlier message exchanges were EXPORT or not. This lack of information combined with weakness of 512-bit prime is exploited by attackers in logjam attack.

### Method of Exploit

The attack, which is popularly known as ***Logjam***, is depicted in Figure 1 and relies on a flaw in the way TLS composes DHE and DHE\_EXPORT. When a server selects DHE\_EXPORT for a handshake, it proceeds by issuing a signed ServerKeyExchange message containing a 512-bit p512, but the structure of this message is identical to the message sent during standard DHE cipher suites. Critically, the signed portion of the server’s message fails to include any indication of the specific ciphersuite that the server has chosen. Provided that a client offers DHE, a man-in-the-middle attacker can rewrite the client’s ClientHello to offer a corresponding DHE\_EXPORT ciphersuite accepted by the server and remove other cipher suites that could be chosen instead. The attacker rewrites the ServerHello response to replace the chosen DHE\_EXPORT ciphersuite with a matching non-export ciphersuite and forwards the ServerKeyExchange message to the client as is. The client will interpret the export-grade tuple (p512, g, gb) as valid DHE parameters chosen by the server and proceed with the handshake. The client and server have different handshake transcripts at this stage, but an attacker who can compute b in close to real time can then derive the master secret and connection keys to complete the handshake with the client, and then freely read and write application data pretending to be the server.

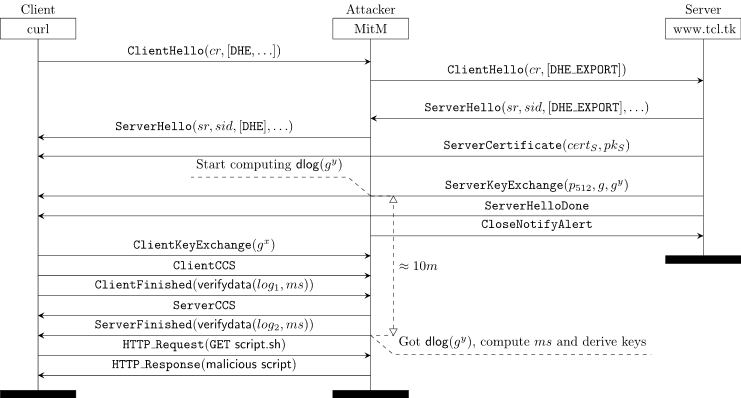


Fig. 1 – Logjam Attack

[Excerpted from <https://weakdh.org/imperfect-forward-secrecy-ccs15.pdf> and <https://weakdh.org/logjam.html> ]

## CVE-2015-4000 fix (Protection from Logjam)

Protection from Logjam attack requires protection for client and server as described below.

**SSL Client**

As seen in Fig 1, an attacker performing Logjam attack alters ClientHello and ServerHello in such a way that though client has requested DHE ciphers, server thinks that DHE\_EXPORT was requested. Similarly, client receives a response for DHE cipher whereas client’s response was for DHE\_EXPORT. Since the following ServerKeyExchange message has no means of communicating that server actually received and agreed to a DHE\_EXPORT cipher request, Client should reject the handshake if DH parameters in the ServerkeyExchange message are not longer than 512 bits.

While this fix also makes client reject genuine DHE ciphers with 512-bit size DH prime, it is only protecting the client and server not to use a weak cipher. However this may break interoperating with servers that cannot support DH parameters more than 512bits which are very rare. But in a specific case where that is required, customer should avoid this logjam patch for client.

**SSL Server**

Protection for SSL server requires two parts.

1. SSL server not accepting any EXPORT ciphers in response to the client requests.

This would mean no support to clients that cannot support any cipher suite other than EXPORT, which are very rare.

1. SSL server rejecting any handshake with non-EXPORT ciphers with DH parameter size not more than 512.

This would mean server rejecting any client requesting non-EXPORT ciphers with DH parameter 512. Like the case of client, this is only protecting server from talking to clients over a weak cipher. However it may make server not able to service to those clients that cannot support DH parameters more than 512bits which are again rare. In the specific case where such a client requires to be supported, customer can disable only this part of the fix by using a Makefile flag.

## Implementation Behavior – OpenSSL & Wind River SSL

### OpenSSL 1.0.2b/1.0.1n

OpenSSL 1.0.2b and 1.0.1n versions of OpenSSL fix the CVE by making SSL/TLS client to reject handshakes with Diffie-Hellman parameters shorter than 768 bits. The limit will be increased to 1024 bits in a future release of OpenSSL.

Details of the fix are described below.

File: ssl/s3\_clnt.c, function: ssl3\_check\_cert\_and\_algorithm()

The function ssl3\_check\_cert\_and\_algorithm() is invoked by a SSL client after receiving ServerkeyExchange message from the server it is trying to connect to. SSL structure passed as argument to this function will include details of the certificate and parameters and other values from server required to derive shared secret.

The function checks various parameters based on the key exchange method. To fix CVE-2015-4000, following additional checks are introduced if dealing with DH cipher suite.

|  |  |
| --- | --- |
| if (algs & (SSL\_kEDH | SSL\_kDHr | SSL\_kDHd)) {  int dh\_size;  if (algs & SSL\_kEDH) {  dh\_size = BN\_num\_bits(dh->p);  } | If ephemeral DH, get length of prime p in bits and store in dh\_size. |
| else {  DH \*dh\_srvr = get\_server\_static\_dh\_key(sc);  if (dh\_srvr == NULL)  goto f\_err;  dh\_size = BN\_num\_bits(dh\_srvr->p);  DH\_free(dh\_srvr);  } | If not ephemeral, get the parameters from the server certificate, get length of prime p and store in dh\_size. |
| if ((!SSL\_C\_IS\_EXPORT(s->s3->tmp.new\_cipher) && dh\_size < 768)  || (SSL\_C\_IS\_EXPORT(s->s3->tmp.new\_cipher) && dh\_size < 512)) {  SSLerr(SSL\_F\_SSL3\_CHECK\_CERT\_AND\_ALGORITHM, SSL\_R\_DH\_KEY\_TOO\_SMALL);  goto f\_err;  }  } | See Note1 below |

Note1:

If the ciphersuite is not EXPORT and dh\_size calculated above is less than 768 , then client is getting into a non\_EXPORT connection that isn’t safe from logjam attack. Reject it.

If the ciphersuite is EXPORT but the dh\_size calculated is less than 512, client is getting into EXPORT cipher based connection but does not use the maximum possible strength for EXPORT ciphers (512bits). Reject it. (Note that EXPORT ciphers with DH parameter size 512bit may still be allowed).

OpenSSL leaves the server part of preventing logjam to user/user applications to take care not to set 512bit DH parameters or try to use EXPORT ciphers.

### Wind River – SSL - Changes

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

Hence changes made by OpenSSL team as described in previous section to fix the CVE are applicable to WR-SSL also.

Details for server part of the fix for WR-SSL are as given below:

1. SSL server not accepting any EXPORT ciphers in response to the client requests.

Disable EXPORT cipher suites for applications.

File: ssl/s3\_lib.c, global array SSL\_CIPHER ssl3\_ciphers[]

Disabled following ciphers

1. EXP-ADH-RC4-MD5
2. EXP-ADH-DES-CBC-SHA
3. EXP-RC4-MD5
4. EXP-RC2-CBC-MD5
5. IDEA-CBC-SHA
6. EXP-DES-CBC-SHA
7. EXP-DH-DSS-DES-CBC-SHA
8. EXP-DH-RSA-DES-CBC-SHA
9. EXP-EDH-DSS-DES-CBC-SHA
10. EXP-EDH-RSA-DES-CBC-SHA
11. EXP1024-DHE-DSS-DES-CBC-SHA
12. EXP1024-DHE-DSS-RC4-SHA
13. SSL server rejecting any handshake with non-EXPORT ciphers with DH parameter size not more than 512.

File: ~/target/ssl/ssl/s3\_srvr.c, function: ssl3\_send\_server\_key\_exchange()

The function, ssl3\_send\_server\_key\_exchange() is invoked when server is required to send a ServerKeyExchange message to client. ServerKeyExchange messages are sent in cases where the keys in ServerCertificate cannot be used for key exchange. E.g. ephemeral DH.

ssl3\_send\_server\_key\_exchange() function initializes a local DH structure with values from tmp\_dh which is supposed to be filled up with prime p and other parameters by the application code or via input file given as command line argument to application. (example is OpenSSL s\_server application).

Changes are made as below to reject a handshake if the cipher is non-export and DH parameter size is not more than 512.

|  |  |
| --- | --- |
| static int ssl3\_send\_server\_key\_exchange(SSL \*s) | Function start |
| { |  |
| … |  |
| if (type & SSL\_kEDH) {  .. | If cipher is ephemeral DH |
| dhp=cert->dh\_tmp; | get access to the parameters set by application |
| … | Validate dhp and contents |
| if ((dh=DHparams\_dup(dhp)) == NULL) { … | Duplicate dhp to dh structure |
| if ((dhp->pub\_key == NULL ||  dhp->priv\_key == NULL ||  (s->options & SSL\_OP\_SINGLE\_DH\_USE)))  {  if(!DH\_generate\_key(dh))  {  SSLerr(SSL\_F\_SSL3\_SEND\_SERVER\_KEY\_EXCHANGE,  ERR\_R\_DH\_LIB);  goto err;  }  }  else  {  dh->pub\_key=BN\_dup(dhp->pub\_key);  dh->priv\_key=BN\_dup(dhp->priv\_key);  if ((dh->pub\_key == NULL) ||  (dh->priv\_key == NULL))  {  SSLerr(SSL\_F\_SSL3\_SEND\_SERVER\_KEY\_EXCHANGE,ERR\_R\_DH\_LIB);  goto err;  }  } | Generate public and private keys for dhp if not existing and duplicate them to dh. |
| #ifdef RIC\_190022 //cve 2015-4000  #ifndef WR\_SSL\_SVR\_NO\_LOGJAM\_PATCH  if(BN\_num\_bits(dh->p) < 768)  {  SSLerr(SSL\_F\_SSL3\_SEND\_SERVER\_KEY\_EXCHANGE,SSL\_R\_DH\_KEY\_TOO\_SMALL);  goto err;  }  #endif  #endi | New code added for CVE-2015-4000 Fix.  Check if prime p is less than 768 bits in size and reject handshake if so. |
| r[0]=dh->p;  r[1]=dh->g;  r[2]=dh->pub\_key; | Copy DH parameters |
| } //end of if (type & SSL\_kEDH) |  |

*Note: Makefile flag WR\_SSL\_SVR\_NO\_LOGJAM\_PATCH can be used to disable the fix in server. Add -DWR\_SSL\_SVR\_NO\_LOGJAM\_PATCH to Makefile in ~/target/ssl/ssl/ directory.*

## Test Approach

**Test scenarios**

Connection tests to be performed as described in following table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Scenario** | | | **Expected Result** | **Evidence** |
| **SSL Server** | **SSL Client** | **Cipher suite** |
| SCN01  (Report1) | WR-SSL WITHOUT Fix | OpenSSL | EXPORT | Handshake SUCCESS | Console o/p for client & server, wireshark capture |
| SCN02  (Report2) | WR-SSL WITHOUT Fix | OpenSSL | NON-EXPORT, DH prime = 512 bit | Handshake SUCCESS | Console o/p for client & server, wireshark capture |
| SCN03  (Report3) | WR-SSL WITHOUT Fix | OpenSSL | NON-EXPORT, DH prime > 512 bit | Handshake SUCCESS | Console o/p for client & server, wireshark capture |
| SCN04  (Report4) | OpenSSL | WR-SSL WITHOUT Fix | EXPORT | Handshake SUCCESS | Console o/p for client & server, wireshark capture |
| SCN05  (Report5) | OpenSSL | WR-SSL WITHOUT Fix | NON-EXPORT, DH prime = 512 bit | Handshake SUCCESS | Console o/p for client & server, wireshark capture |
| SCN06  (Report6) | OpenSSL | WR-SSL WITHOUT Fix | NON-EXPORT, DH prime > 512 bit | Handshake SUCCESS | Console o/p for client & server, wireshark capture |
| SCN07  (Report7) | WR-SSL WITH Fix | OpenSSL | EXPORT | Handshake FAIL | Console o/p for client & server, wireshark capture |
| SCN08  (Report8) | WR-SSL WITH Fix | OpenSSL | NON-EXPORT, DH prime = 512 bit | Handshake FAIL | Console o/p for client & server, wireshark capture |
| SCN09  (Report9) | WR-SSL WITH Fix | OpenSSL | NON-EXPORT, DH prime > 512 bit | Handshake SUCCESS | Console o/p for client & server, wireshark capture |
| SCN10  (Report10) | OpenSSL | WR-SSL WITH Fix | EXPORT | Handshake FAIL | Console o/p for client & server, wireshark capture |
| SCN11  (Report11) | OpenSSL | WR-SSL WITH Fix | NON-EXPORT, DH prime = 512 bit | Handshake FAIL | Console o/p for client & server, wireshark capture |
| SCN12  (Report12) | OpenSSL | WR-SSL WITH Fix | NON-EXPORT, DH prime > 512 bit | Handshake SUCCESS | Console o/p for client & server, wireshark capture |

*Note: OpenSSL in the above table always means OpenSSL without fix for CVE-2015-4000.*

## Summary/Conclusion

Detailed examination of the WR-SSL implementation reveals that WR-SSL is vulnerable to the security issue described in CVE-2015-4000 in both client and server roles.

Required code changes were made and tests performed.

## Attachments

Reports 1 to 12 as described in table in section 2.5.

/EOD