



ASK PENTEST

Malayan Banking Berhad ('MBB')

Source Code Review for eCustody

December 2021


Confidentiality Notice

The information transmitted in this document is intended only for the addressee and may contain confidential and/or privileges material. Any interception, review, retransmission, dissemination, or the other use of, or taking of any action upon this information by persons or entities other than intended recipients is prohibited by law and may subject them to criminal and/ or civil liability.

DOCUMENT AUTHORIZATION

The enclosed document has been reviewed and accepted by the following people:

Ask Pentest Sdn Bhd

NAME	POSITION	SIGNATURE	DATE
Rahezar Rahmat	Senior Security Consultant		3/12/2021

The enclosed document has been verified by the following people:

Malayan Banking Berhad

NAME	POSITION	SIGNATURE	DATE

DOCUMENT AMENDMENT REGISTER

The enclosed document has been amended according to the following description:

#	DATE	REASON	CHAPTER	VERSION	AUTHOR
1.	1 December 2021	Draft	All	0.1	Irwan Ishak
2.	2 December 2021	Review & Comment	All	0.2	Amirul
3.	2 December 2021	Amend & Final	All	1.0	Irwan Ishak

Table of Contents

DOCUMENT AUTHORIZATION	i
DOCUMENT AMENDMENT REGISTER	ii
DEFINITION	iv
1 EXECUTIVE SUMMARY	1
1.1 Introduction	1
1.2 Background Information	1
1.3 Objective and Scope	2
1.4 Summary of Findings.....	3
2 SUMMARY OF TECHNICAL FINDINGS	4
2.1 Introduction	4
2.2 Risk Level Summary	4
2.3 Risk Level Distribution.....	4
2.4 Findings Matrix.....	5
3 DETAILED TECHNICAL FINDINGS	6
3.1 Findings for eCustody	6
3.1.1 Unsupported Algorithms (DES nor DESede)	6

DEFINITION

Risk level:

Risk Level	Description
Critical	<i>Severe and may lead to complete loss of confidentiality, controls, services or client/partner confidence. The risk categorized to critical often to be trivial to exploit using publicly available tools or technique.</i>
High	<i>Severe or serious loss of confidentiality, controls, services or client/partner confidence.</i>
Medium	<i>Significant or medium loss of confidentiality, controls, services or client/partner confidence.</i>
Low	<i>Minor loss of confidentiality, controls, services or client/partner confidence.</i>

CVSS Scoring System:

Common Vulnerability Scoring System (CVSS) is a free and open industry standard for assessing the severity of computer system security vulnerabilities under the custodianship of the Forum of Incident Response and Security Teams (FIRST). It attempts to establish a measure of how much concern a vulnerability warrants, compared to other vulnerabilities, so efforts can be prioritized. CVSS scores are based on a series of measurements (called metrics) based on expert assessment ranged from 0 to 10. Vulnerabilities with a base score of 10 are **Critical**, range 7.0-9.9 are **High**, those in the range 4.0-6.9 as **Medium**, and 0-3.9 as **Low**.

CVSS Base Scoring

CVSS Base Scoring attempt to measure qualities intrinsic of vulnerability base on several metric as describe on table below:

Value	Description
Exploitability Metric	
<i>Access Vector (AV) - shows how a vulnerability may be exploited</i>	
Local (L)	<i>The attacker must either have physical access to the vulnerable system (e.g. firewire attacks) or a local account (e.g. a privilege escalation attack).</i>
Adjacent Network (A)	<i>The attacker must have access to the broadcast or collision domain of the vulnerable system (e.g. ARP spoofing, bluetooth attacks).</i>
Network (N)	<i>The vulnerable interface is working at layer 3 or above of the OSI Network stack. These types of vulnerabilities are often described as remotely exploitable (e.g. a remote buffer overflow in a network service)</i>
<i>Access Complexity (AC) - describes how easy or difficult it is to exploit the discovered vulnerability.</i>	
High (H)	<i>Specialized conditions exist, such as a race condition with a narrow window, or a requirement for social engineering methods that would be readily noticed by knowledgeable people.</i>

Medium (M)	<i>There are some additional requirements for access, such as a limit on the origin of the attacks, or a requirement for the vulnerable system to be running with an uncommon, non-default configuration.</i>
Low (L)	<i>There are no special conditions for access to the vulnerability, such as when the system is available to large numbers of users, or the vulnerable configuration is ubiquitous.</i>
<i>Authentication (Au) - describes the number of times that an attacker must authenticate to a target to exploit it</i>	
Multiple (M)	<i>Exploitation of the vulnerability requires that the attacker authenticate two or more times, even if the same credentials are used each time.</i>
Single (S)	<i>The attacker must authenticate once in order to exploit the vulnerability.</i>
None (N)	<i>There is no requirement for the attacker to authenticate.</i>
Impact Metric	
<i>Confidentiality (C) - describes the impact on the confidentiality of processed by the system.</i>	
None (N)	<i>There is no impact on the confidentiality of the system.</i>
Partial (P)	<i>There is considerable disclosure of information, but the scope of the loss is constrained such that not all of the data is available.</i>
Complete (C)	<i>There is total information disclosure, providing access to any / all data on the system.</i>
<i>Integrity (I) - describes the impact on the integrity of the exploited system</i>	
None (N)	<i>There is no impact on the integrity of the system.</i>
Partial (P)	<i>Modification of some data or system files is possible, but the scope of the modification is limited.</i>
Complete (C)	<i>There is total loss of integrity; the attacker can modify any files or information on the target system.</i>
<i>Availability (A) - describes the impact on the availability of the target system.</i>	
None (N)	<i>There is no impact on the availability of the system.</i>
Partial (P)	<i>There is reduced performance or loss of some functionality.</i>
Complete (C)	<i>There is total loss of availability of the attacked resource.</i>

CVSS Temporal Scoring

CVSS Temporal Scoring attempt to measure characteristics that evolve over the lifetime of vulnerability as described in table below:

Value	Description
Exploitability Metric	
<i>Exploitability (E) - the current state of exploitation techniques or automated exploitation code</i>	
Unproven (U)	<i>No exploit code is available, or the exploit is theoretical.</i>
Proof-of-concept (P)	<i>Proof-of-concept exploit code or demonstration attacks are available, but not practical for widespread use. Not functional against all instances of the vulnerability.</i>
Functional (F)	<i>Functional exploit code is available, and works in most situations where the vulnerability is present.</i>
High (H)	<i>The vulnerability can be exploited by automated code, including mobile code (such as a worm or virus).</i>

Not Defined (ND)	<i>This is a signal to ignore this score.</i>
<i>Remediation level (RL) - current state of mitigation level of the vulnerability</i>	
Official Fix (O)	<i>A complete vendor solution is available - either a patch or an upgrade.</i>
Temporary Fix (T)	<i>There is an official but temporary fix / mitigation available from the vendor.</i>
Workaround (W)	<i>There is an unofficial, non-vendor solution or mitigation available - perhaps developed or suggested by users of the affected product or another third party.</i>
Unavailable (U)	<i>There is no solution available, or it is impossible to apply a suggested solution. This is the usual initial state of the remediation level when a vulnerability is identified.</i>
Not Defined (ND)	<i>This is a signal to ignore this score.</i>
<i>Report confidence (RC) - measures the level of confidence in the existence of the vulnerability and also the credibility of the technical details of the vulnerability.</i>	
Unconfirmed (UC)	<i>A single unconfirmed source, or multiple conflicting sources. Rumored vulnerability.</i>
Uncorroborated (UR)	<i>Multiple sources that broadly agree - there may be a level of remaining uncertainty about the vulnerability</i>
Confirmed (C)	<i>Acknowledged and confirmed by the vendor or manufacturer of the affected product.</i>
Not Defined (ND)	<i>This is a signal to ignore this score.</i>

Vulnerability References:

Reference	Description
BID	<p><i>BugTraq is a full disclosure moderated mailing list for the *detailed* discussion and announcement of computer security vulnerabilities: what they are, how to exploit them, and how to fix them. Each vulnerability is assigned a unique "BugTraq ID" which can be access from URL:</i></p> <ul style="list-style-type: none"> - <a href="http://www.securityfocus.com/bid/<BID>">http://www.securityfocus.com/bid/<BID>
CVE	<p><i>CVE is a dictionary of publicly known information security vulnerabilities and exposures. CVE's common identifiers enable data exchange between security products and provide a baseline index point for evaluating coverage of tools and services. Each CVE assigned vulnerability can be access from URL:</i></p> <ul style="list-style-type: none"> - <a href="https://cve.mitre.org/cgi-bin/cvename.cgi?name=<CVE>">https://cve.mitre.org/cgi-bin/cvename.cgi?name=<CVE>
CWE	<p><i>CWE provides a unified, measurable set of software weaknesses that is enabling more effective discussion, description, selection, and use of software security tools and services that can find these weaknesses in source code and operational systems as well as better understanding and management of software weaknesses related to architecture and design. Each CWE assigned weakness can be access from URL:</i></p> <ul style="list-style-type: none"> - <a href="https://cwe.mitre.org/data/definitions/<CWE>.html">https://cwe.mitre.org/data/definitions/<CWE>.html

CERT	<p>The CERT Knowledgebase is a collection of internet security information related to incidents and vulnerabilities. Each security information will be assigned a unique id which can be access from URL:</p> <ul style="list-style-type: none">- <a href="https://www.kb.cert.org/vuls/id/<CERT>">https://www.kb.cert.org/vuls/id/<CERT>
OSVDB	<p>Open Sourced Vulnerability Database (OSVDB) is an independent and open-sourced database. The goal of the project is to provide accurate, detailed, current, and unbiased technical information on security vulnerabilities. The project promotes greater, open collaboration between companies and individuals. Each vulnerability assigned by OSVDB can be access from URL:</p> <ul style="list-style-type: none">- <a href="http://osvdb.org/show/osvdb/<OSVDB>">http://osvdb.org/show/osvdb/<OSVDB>

Compliance Result:

Result	Description
Passed	Configuration value comply with compliance checklist
Warning	Configuration not applicable on the system
Failed	Configuration value not comply with compliance checklist

1 EXECUTIVE SUMMARY

1.1 Introduction

Ask Pentest Sdn Bhd ('ASK') performed Source Code Review for eCustody on behalf of Malayan Banking Berhad ('MBB').

ASK performed the fieldwork from 28 November 2021 to 1 December 2021.

This report contains the details of the exercise including test scope, identified security weaknesses, risks analysis and recommendations to mitigate each weakness found during the exercise.

1.2 Background Information

Source code review is the process of auditing the source code for an application to verify that the proper security controls are present, that they work as intended, and that they have been invoked in all the right places. The auditing approach involves looking inside the applications internal, with full access to design documentation, source code and other materials.

The goal is to find common and uncommon vulnerabilities inside the source code following guidelines from OWASP Top 10:

- i. A1-Injection
- ii. A2-Broken Authentication and Session Management
- iii. A3-Cross-Site Scripting (XSS)
- iv. A4-Insecure Direct Object References
- v. A5-Security Misconfiguration
- vi. A6-Sensitive Data Exposure
- vii. A7-Missing Function Level Access Control
- viii. A8-Cross-Site Request Forgery (CSRF)
- ix. A9-Using Components with Known Vulnerabilities
- x. A10-Unvalidated Redirects and Forwards

The tests were conducted using the following tools:

#	Name	Functionality
1	Notepad++	Free Source Code Editor
2	Sublime Text	Source Code Editor
3	Visual Code Grepper (VCG)	Static Code Analysis Tool

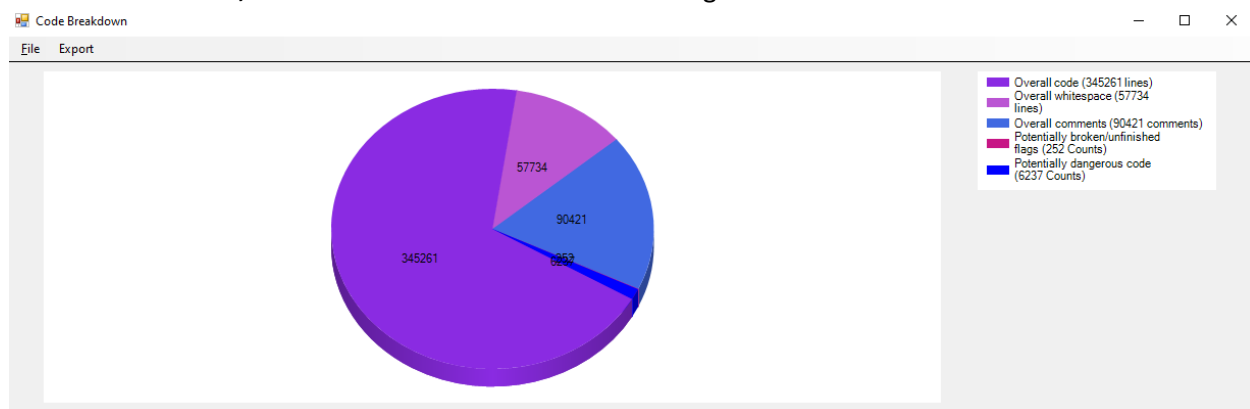
1.3 Objective and Scope

The objective of the exercise was to provide Maybank with a point in time assessment of the security controls within the tested system. The assessment result will be a list of known security weaknesses found during the time of the test, together with recommendations for remedial action to increase security level of the tested system.

The security assessment was performed against the following system:

#	Module	Information
1	eCustody	Source Code Review

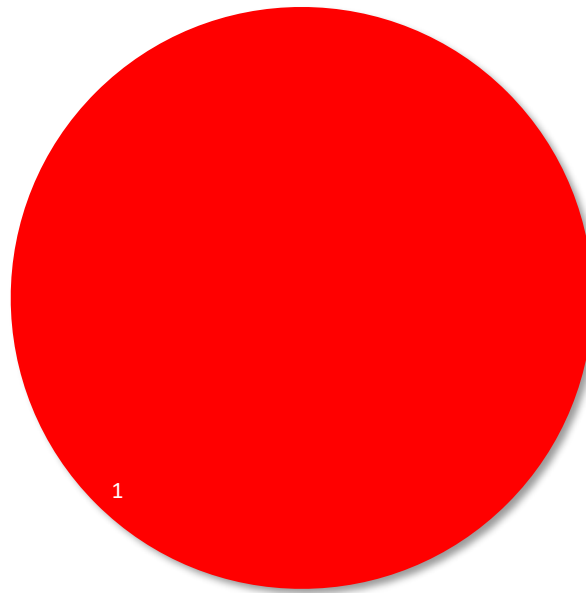
Overall total of **345,261 line** of codes were reviewed during the assessment:



1.4 Summary of Findings

The figure below demonstrates the distribution of all security weaknesses found based on their risk level and list of technical findings reference which can be found in **SECTION 3** of the report.

Vulnerability Distribution



■ Critical ■ High ■ Medium ■ Low

	Critical	High	Medium	Low
Total	0	1	0	0

2 SUMMARY OF TECHNICAL FINDINGS

2.1 Introduction

This section provides summary information about the security weaknesses identified, including a summary of findings, their business implications, an overview of the testing process, and a matrix containing summary information about weakness found.

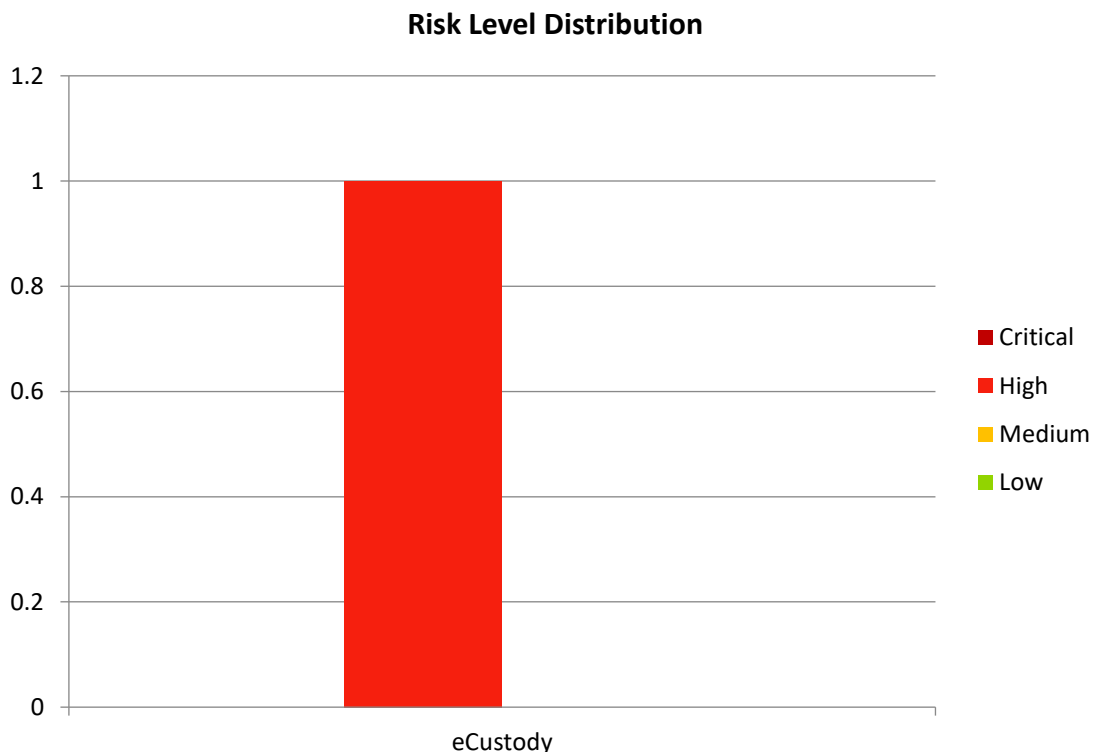
2.2 Risk Level Summary

The table below contains summary of findings found during the exercise:

#	Module	Critical	High	Medium	Low	Total
1	eCustody	0	1	0	0	1
Total		0	1	0	0	1

2.3 Risk Level Distribution

The graph below demonstrates the distribution on findings found during the exercise:



2.4 Findings Matrix

The table below contains a list of all the findings identified during this assessment:

#	Name	Total	Module	Risk
1	Unsupported Algorithms (DES nor DESede)	1	eCustody	HIGH

3 DETAILED TECHNICAL FINDINGS

This section provides details information about the security weaknesses identified during the exercise, including a description of findings, observation process’s output, and recommendation to mitigate the issue.

3.1 Findings for eCustody

3.1.1 Unsupported Algorithms (DES nor DESede)

Service	Risk	CVSS	CVSS Vector
	HIGH		

Description

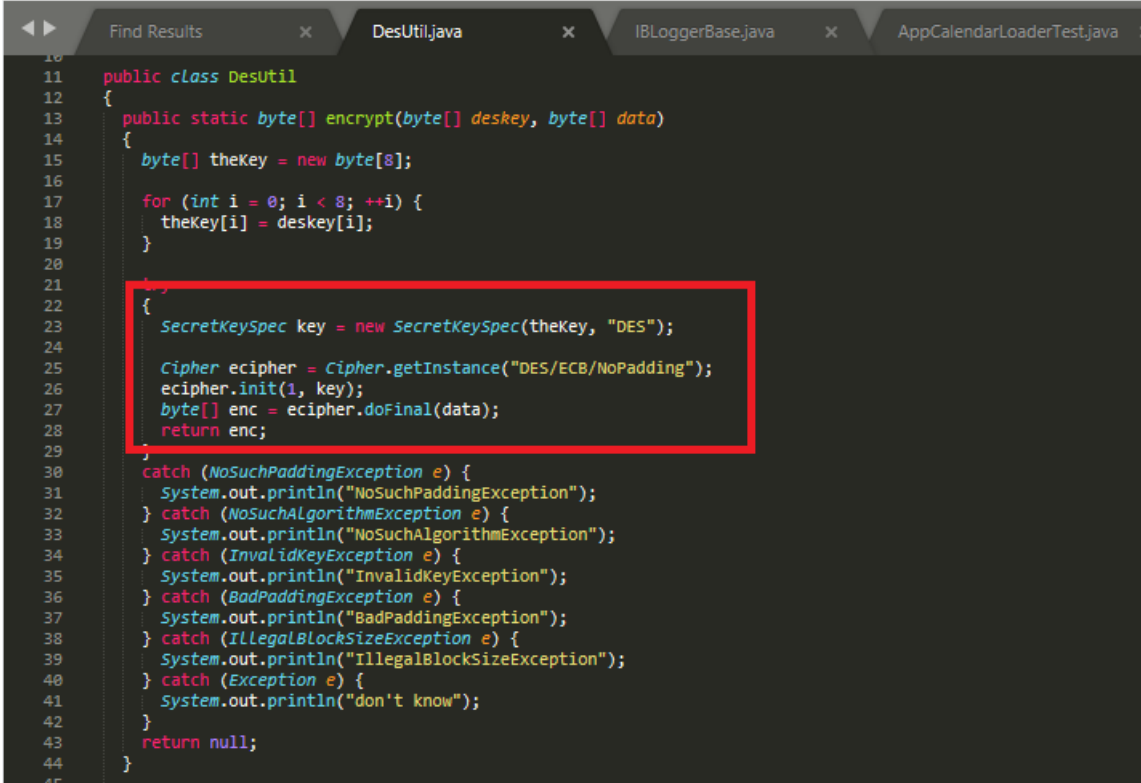
According to the US National Institute of Standards and Technology (NIST), the Data Encryption Standard (DES) is no longer considered secure:

- Adopted in 1977 for federal agencies to use in protecting sensitive, unclassified information, the DES is being withdrawn because it no longer provides the security that is needed to protect federal government information. Federal agencies are encouraged to use the Advanced Encryption Standard, a faster and stronger algorithm approved as FIPS 197 in 2001. For similar reasons, RC2 should also be avoided.

#	Affected Path	Line No
1	\\eCustody\ecustody_Code_Comp\mbb-custody-common\src\java\com\mbb\custody\utils\MACUtils.java	193-214
2	\\eCustody\ecustody_Code_Comp\mbb-custody-common\src\java\com\mbb\custody\utils\DesUtil.java	13-45

Proof of Concept

de Review\Part 3\ecustody\ecustody_Code_Comp\mbb-custody-common\src\java\com\mbb\custody\utils\DesUtil.java (eCustody) -
ls Project Preferences Help



```
10  
11 public class DesUtil  
12 {  
13     public static byte[] encrypt(byte[] deskey, byte[] data)  
14     {  
15         byte[] thekey = new byte[8];  
16  
17         for (int i = 0; i < 8; ++i) {  
18             thekey[i] = deskey[i];  
19         }  
20  
21  
22         {  
23             SecretKeySpec key = new SecretKeySpec(thekey, "DES");  
24  
25             Cipher ecipher = Cipher.getInstance("DES/ECB/NoPadding");  
26             ecipher.init(1, key);  
27             byte[] enc = ecipher.doFinal(data);  
28             return enc;  
29         }  
30  
31         catch (NoSuchPaddingException e) {  
32             System.out.println("NoSuchPaddingException");  
33         } catch (NoSuchAlgorithmException e) {  
34             System.out.println("NoSuchAlgorithmException");  
35         } catch (InvalidKeyException e) {  
36             System.out.println("InvalidKeyException");  
37         } catch (BadPaddingException e) {  
38             System.out.println("BadPaddingException");  
39         } catch (IllegalBlockSizeException e) {  
40             System.out.println("IllegalBlockSizeException");  
41         } catch (Exception e) {  
42             System.out.println("don't know");  
43         }  
44         return null;  
45     }  
46 }
```

Figure 1: The screenshot shows the encryption using unsupported algorithms ("DES/ECB/NoPadding") in DesUtil.java on line 25

Recommendation

Recommended for block ciphers (AES/GCM/NoPadding)

See also

- <https://rules.sonarsource.com/java/type/Vulnerability/RSPEC-5547>
- <https://cwe.mitre.org/data/definitions/327.html>
- <https://wiki.sei.cmu.edu/confluence/x/hDdGBQ>

Remark

--