Malayan Banking Berhad ('Maybank')

Source Code Review for eCustody

October 2017

DOCUMENT AUTHORIZATION

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

Ask Pentest Sdn Bhd

NAME	POSITION	COMPANY	SIGNATURE	DATE

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.	25 October	Technical Report	All	1.0	Nizam
2.					
3.					
4.					
5.					

Table of Contents

DEFINITION

Risk level:

Risk Level	Description
Critical	*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.*
High	*Severe or serious loss of confidentiality, controls, services or client/partner confidence.*
Medium	*Significant or medium loss of confidentiality, controls, services or client/partner confidence.*
Low	*Minor loss of confidentiality, controls, services or client/partner confidence.*

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	
Access Vector (AV) - shows how a vulnerability may be exploited	
Local (L)	*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).*
Adjacent Network (A)	*The attacker must have access to the broadcast or collision domain of the vulnerable system (e.g. ARP spoofing, bluetooth attacks).*
Network (N)	*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)*
Access Complexity (AC) - describes how easy or difficult it is to exploit the discovered vulnerability.	
High (H)	*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.*

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

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

Not Defined (ND)	*This is a signal to ignore this score.*
Remediation level (RL) - current state of mitigation level of the vulnerability	
Official Fix (O)	*A complete vendor solution is available - either a patch or an upgrade.*
Temporary Fix (T)	*There is an official but temporary fix / mitigation available from the vendor.*
Workaround (W)	*There is an unofficial, non-vendor solution or mitigation available - perhaps developed or suggested by users of the affected product or another third party.*
Unavailable (U)	*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.*
Not Defined (ND)	*This is a signal to ignore this score.*
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.	
Unconfirmed (UC)	*A single unconfirmed source, or multiple conflicting sources. Rumored vulnerability.*
Uncorroborated (UR)	*Multiple sources that broadly agree - there may be a level of remaining uncertainty about the vulnerability*
Confirmed (C)	*Acknowledged and confirmed by the vendor or manufacturer of the affected product.*
Not Defined (ND)	*This is a signal to ignore this score.*

Vulnerability References:

Reference	Description
BID	*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:* - *http://www.securityfocus.com/bid/<BID>*
CVE	*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:* - *https://cve.mitre.org/cgi-bin/cvename.cgi?name=<CVE>*
CWE	*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:* - *https://cwe.mitre.org/data/definitions/<CWE>.html*

CERT	*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:* - *https://www.kb.cert.org/vuls/id/<CERT>*
OSVDB	*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:* - *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 ('MAYBANK').

ASK performed the fieldwork from 20 September 2017 to 13 October 2017.

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 such as:

- Buffer Overruns and Overflows
- OS Injection
- SQL Injection
- Cross Site Scripting
- Cross-Site Forgery
- Logging Issues
- Session Integrity
- Race Conditions.

The tests were conducted using the following tools:

Table 1 : List of tools

#	Name	Functionality
1.	Notepad++	Free Source Code Editor
2.	Visual Code Grepper (VCG)	Automated Source Code Analysis Tool
3.	Yasca	Automated Source Code Analysis Tool
4.	RIPS	Automated Source Code Analysis Tool

1.3 Objective and Scope

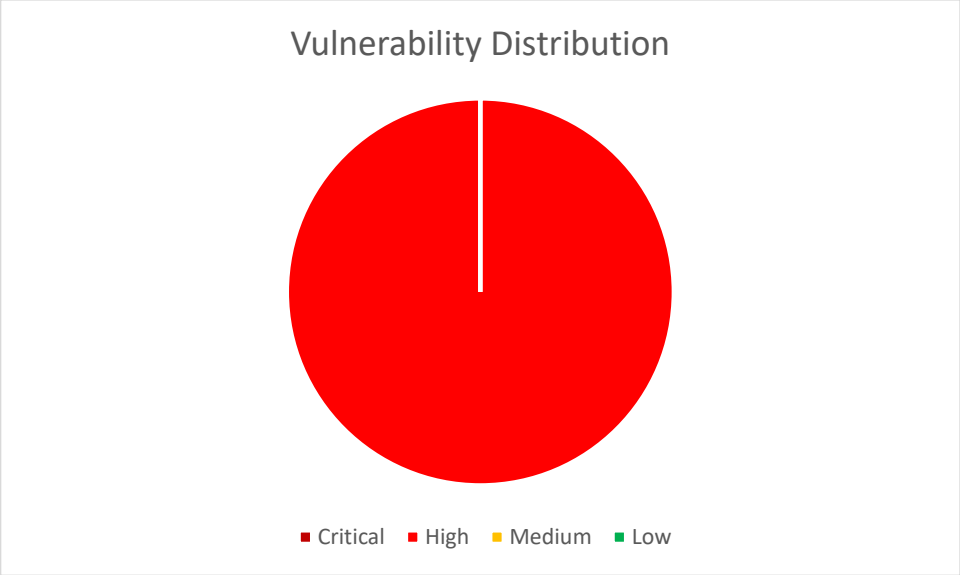
The objective of the exercise was to provide MAYBANK with a point in time assessment of the security controls within the tested system. The assessment result will be a list of known security weaknesses found during the time of the test, together with recommendations for remedial action to increase security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

The security assessment was performed against the following system:

#	Name
1	eCustody

1.4 Summary of Findings

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



	Critical	High	Medium	Low
Findings Reference		3.1.1 3.1.2		
Total	0	2	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.

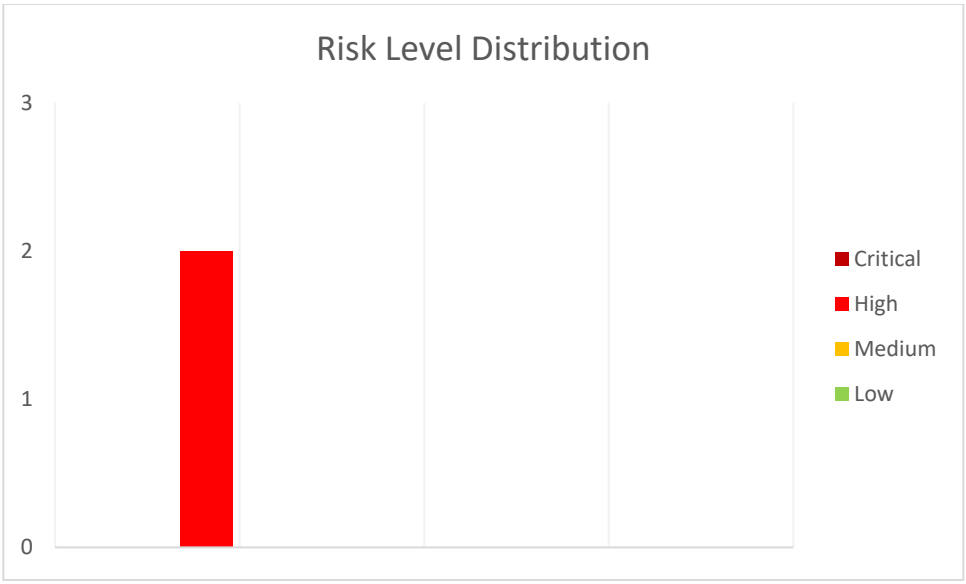
2.2 Risk Level Summary

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

#	System	Critical	High	Medium	Low	Total
1	eCustody	0	2	0	0	2
Total		0	2	0	0	2

2.3 Risk Level Distribution

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



3 DETAILED TECHNICAL FINDINGS

3.1.1 SQL Injection (SQLi)

Service	Risk	CVSS	CVSS Vector
	HIGH		

Vulnerability Status

CVSS Temporal Score	CVSS Temporal Vector	Exploit Available

References

BID	CERT	OSVDB	CWE	CVE

Description

SQL injection is a vulnerability that allows an attacker to alter backend SQL statements by manipulating the user input. An SQL injection occurs when web applications accept user input that is directly placed into a SQL statement and doesn't properly filter out dangerous characters.

This is one of the most common application layer attacks currently being used on the Internet. Despite the fact that it is relatively easy to protect against, there is a large number of web applications vulnerable.

An unauthenticated attacker may execute arbitrary SQL statements such as insert, update and delete data on the vulnerable system. This may compromise the integrity of your database and/or expose sensitive information.

In the worst case, depend on your database server type, configuration and version; an attacker may use this vulnerability to exploit the existing vulnerability on your database server and take control of the system.

Screen capture

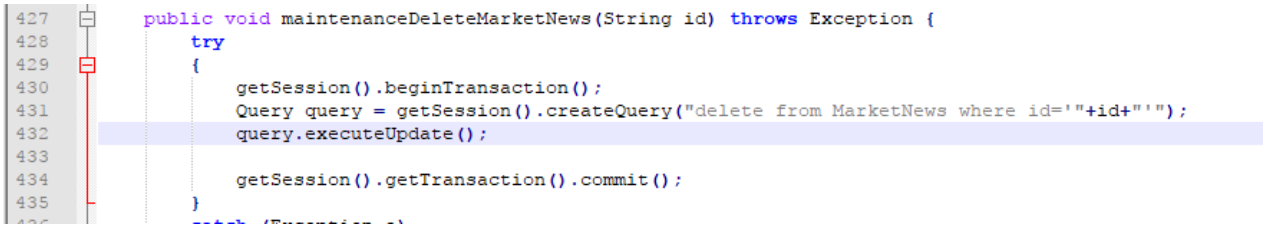


Figure 1: String concatenation used in sql query

List of affected Item

Module	Request Filename	Request Parameter	SQL Filename/Function
mbb-custody-admin	cua/src/com/ibm/cua/app/web/AlternateUserListEvent.java	userPK sysId sysName altUserId	AlternateUserDO.java getTotalRecords findBy

Recommendation

Filter user supplied value in parameters and variables before use in any SQL query.

Example:

- Integer - make sure only integer are supplied by user.
- String - Filter again dangerous character such as ' and ".
- Use prepare statement.

Management Response

3.1.2 Cross Site Scripting (XSS)

Service	Risk	CVSS	CVSS Vector
	HIGH		

Vulnerability Status

CVSS Temporal Score	CVSS Temporal Vector	Exploit Available

References

BID	CERT	OSVDB	CWE	CVE

Description

Cross site scripting (also referred to as XSS) is a vulnerability that allows an attacker to send malicious code (usually in the form of JavaScript) to another user. Because a browser cannot know if the script should be trusted or not, it will execute the script in the user context allowing the attacker to access any cookies or session tokens retained by the browser.

This vulnerability usually performs in conjunction with phishing attack to make trusted website serve attacker's malicious code such as fake login page or code to steal user's information.
Attacker may inject JavaScript, VBScript, ActiveX, HTML or Flash into a vulnerable application to fool a user in order to gather data from them. An attacker can steal the session cookie and take over the account, impersonating the user and modify the content of the page presented to the user.

There is also possibility for attacker to manipulate HTML code to exploit existing web browser vulnerability such as Windows Metafile vulnerability and infecting the user's system with computer Virus/ Trojan horse.

Screen capture

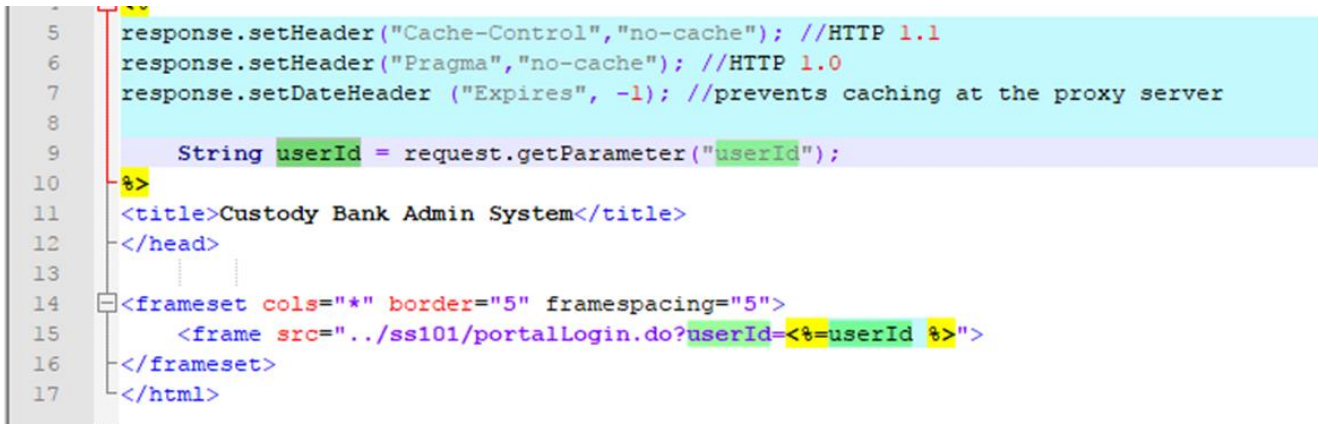


Figure 2: userid supplied parameter is without any filtering

Affected Item

Module	Filename	Request Parameter
mbb-custody-admin	mbb-custody-admin\WebContent\template\ibssFrame.jsp	userId
mbb-custody-admin	mbb-custody-admin\WebContent\jsp\ss105_enquiry\ibssReports.jsp	nId
mbb-custody-main	mbb-custody-main\WebContent\jsp\alerts\ecsCorpActionAlerts.jsp	nId
mbb-custody-main	mbb-custody-main\WebContent\jsp\alerts\ecsSubstanHold.jsp	nId
mbb-custody-main	mbb-custody-main\WebContent\jsp\alerts\ ecsTradeSummary.jsp	nId
mbb-custody-main	mbb-custody-main\WebContent\jsp\alerts\ ecsTrnxCreditDebit.jsp	nId
mbb-custody-main	mbb-custody-main\WebContent\jsp\ib102_account_info\ecsAccountEnquiryDetails.jsp	cacheLevel
mbb-custody-main	mbb-custody-main\WebContent\jsp\ib106_report\ecsReports.jsp	nId
mbb-custody-main	mbb-custody-main\WebContent\jsp\ib106_report\ecsReportsMaintenance.jsp	nId
mbb-custody-main	mbb-custody-main\WebContent\layout\ibsFrame.jsp	action
mbb-custody-main	mbb-custody-main\WebContent\layout\ibsWelcomeMessage.jsp	nId

Recommendation

Replace HTML special character with HTML special code before display back to user which is:

- < = <
- > = >
- (= &40;
-) = &41
- " = "
- ' = &39;

In some cases where the input being used to generate URL, this character must be convert to URL encoding format.

See also

- http://www.owasp.org/index.php/Cross_Site_Scripting
- http://en.wikipedia.org/wiki/Cross-site_scripting

Management Response