SEC Consult
an atos company

Report 9001055414

# Security Assessment – MBB Web Application Pen Test

eCustody - Web Application

for

Maybank Shared Services Sdn Bhd


conducted by

## SEC Consult

**Version:** 1.0
**Responsible**: K. Hakimin
**Confidentiality class**: Strictly confidential

**Date:** 2023-01-25
**Author**: M. Azri

# Table of Contents

**1 Management Summary ..... 3**
1.1 Scope ..... 3
1.2 Goal ..... 3
1.3 Results ..... 3
1.3.1 Worst-Case Scenarios ..... 3
1.3.2 Technical Risk Assessment ..... 4
1.4 Suggested Measures ..... 4
1.4.1 Measures with Immediate Need for Action ..... 4
1.4.2 Further Measures ..... 5
**2 Approach ..... 6**
2.1 Testing Method ..... 6
2.2 Scope and Timetable ..... 6
2.3 Test Classes Performed ..... 7
2.3.1 Server Configuration ..... 7
2.3.2 Patch Level ..... 7
2.3.3 Standard Software and Proprietary Applications ..... 7
2.4 Actions to be Taken After the Conducted Test ..... 9
2.5 Disclaimer ..... 9
**3 Vulnerability Summary ..... 10**
3.1 Total Risk Per System ..... 10
3.2 Risk of Each Vulnerability ..... 10
**4 Detailed Analysis ..... 11**
4.1 eCustody – Web Application ..... 11
4.1.1 General Information ..... 11
4.1.2 Whois Information ..... 11
4.1.3 Port Scan Results ..... 11
4.1.4 Multiple TLS(/SSL) Vulnerabilities ..... 12
4.1.5 Outdated Software ..... 16
4.1.6 Improper HTTP Security Header (CSP) ..... 19
4.1.7 Insecure Cookie Configuration ..... 21
4.1.8 General Information Disclosure ..... 23
**5 Version History ..... 25**
**Appendix A Risk Calculation ..... 26**
A.1 Definition of the Term Likelihood ..... 26
A.2 Definition of the Term Severity ..... 27
A.3 Total Risk ..... 27

# 1 Management Summary

The following chapter summarizes the scope of the security assessment, the results of the assessment and outlines the measures recommended by SEC Consult.

## 1.1 Scope

During the internal security assessment for the company Maybank Shared Services Sdn Bhd (Maybank), SEC Consult assessed the eCustody Web Application, using the graybox approach.

The security consultants were provided with various valid user accounts in different roles. Therefore, the consultants' knowledge about the application was similar to that of an attacker who had gained access to valid user accounts.

Please refer to the disclaimer in chapter 2.5.

## 1.2 Goal

It was the goal of the assessment to find all kinds of vulnerabilities and reveal common configuration issues in the eCustody Web Application.

## 1.3 Results

SEC Consult did not find any critical vulnerabilities in the eCustody Web Application in the **given timeframe** of the assessment.

However, SEC Consult discovered **5 low risk problems**. The root causes of these vulnerabilities are:

- Insufficient patch management
- Insecure configuration of services

### 1.3.1 Worst-Case Scenarios

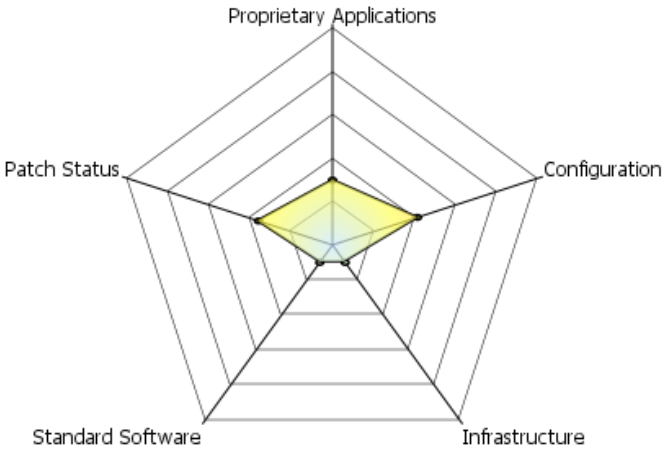
If an attacker exploits the identified vulnerabilities, the following attack scenarios are possible:

- **An unauthenticated attacker (not logged in) is able to:**
  - Gather valuable information about the underlying system.

    An attacker can effectively collect information about the application and the deployed server software to aid further attacks.

### 1.3.2 Technical Risk Assessment

The following risk profile results from a five-dimensional risk assessment:




Legend: The risk is visualized in amplitude and color (Light blue: low risk, Red: high risk).

- **Proprietary Applications:** In the assessed proprietary web application low risk issue were identified.
- **Patch Status:** During the assessment, low vulnerabilities resulting from outdated software were found.
- **Standard Software:** In the deployed standard software no new security flaws were discovered.
- **Configuration:** The configuration of network components is acceptable. However, low risk vulnerabilities were found in the configuration of the web server.
- **Infrastructure:** Out of scope

The risk score of **13.45 (Medium) out of 125.00** indicates that the company Maybank Shared Services Sdn Bhd needs to take actions to raise the level of security.

## 1.4 Suggested Measures

Based on the results of the security assessment, SEC Consult recommends the following measures:

### 1.4.1 Measures with Immediate Need for Action

The identified vulnerabilities are based on flaws in processes and procedures in the **Application Security Management**. This implies:

1. **Correction of the discovered vulnerabilities.** Multiple vulnerabilities have been found during the security assessment. Those vulnerabilities should be corrected as soon as possible. Recommended solutions can be found in the corresponding chapters.
2. **Recheck of the assessed applications.** A recheck can ensure that the countermeasures are applied correctly, and all found vulnerabilities have been eliminated.
3. **Root-cause analysis for all vulnerabilities identified.** To prevent similar issues in the future, a root-cause analysis must be performed that identifies the reasons, why the vulnerabilities were introduced in the first place.

   Only a root-cause analysis leads to proper measures that will prevent similar vulnerabilities in the first place in the future. Measures might be required on the technical level, the process level or the awareness level.

### 1.4.2 Further Measures

In the mid- and long-term SEC Consult recommends the following measures to mitigate / solve the identified problems.

1. **Security acceptance tests.** A security assessment should be done for every system before its use in production in an extent that reflects the system's criticality. By performing such a test before rollout to production, risks can be drastically reduced, and potential downtimes avoided.
2. **Implementation of an Application Security Management process.** A process around managing application security ensures that all aspects of information security in the whole application lifecycle are considered and that no applications left out. It enables efficiency by properly prioritizing activities based on potential risks and available resources.
3. **Periodic internal security assessment.** An internal security check will reveal vulnerabilities in the IT infrastructure and intranet applications. It is recommended to raise the security level of all intranet components – such as file servers, DNS servers, database servers or Active Directory – by performing an internal security assessment.

# 2 Approach

The following chapter outlines the security assessment approach of SEC Consult.

## 2.1 Testing Method

SEC Consult conducts security assessments to check the security of a complete system or single system components. The tools, methods and techniques used by SEC Consult fall into three categories:

1. Well known throughout both the computer security and "hacker" communities.
2. In-house tools developed to extend the boundaries beyond the usual hacker's toolkit.
3. Expert knowledge. Security consultants look for vulnerabilities that may not be discovered by using automated tools.

## 2.2 Scope and Timetable

The security assessment took place from 2023-01-03 to 2023-01-06. Objective of this test was to check the eCustody Web Application against all kinds of vulnerabilities and common configuration issues.

The application was assessed in the version available during the timeframe of the security assessment and was accessible via the following URL:

- 172.29.75.65/custody-main-test/common/Login.do

The following application accounts were provided:

- User 1:
  - Username: sonymaker
  - Organization: sony
  - Role: Normal

Existing WAF, firewall and IDS/IPS solutions were not deactivated/reconfigured for the assessment.

No checks were conducted where the availability of services would be deliberately put at risk.

## 2.3 Test Classes Performed

Systems in scope have been tested against the following test classes.

[1]Tested: This attack vector was tested against in this security assessment by SEC Consult.

[2]Exploitable: This attack vector was successfully exploited in course of this security assessment.

### 2.3.1 Server Configuration

| | **Server Configuration** | |
|---|---|---|
| | This class covers exploitable configuration errors for all kinds of server software. | |
| Attack pattern | Tested[1] | Exploitable[2] |
| Enumerating server contents | YES | NO |
| Exploiting default accounts | YES | NO |
| Enumerating user accounts | YES | NO |
| Exploiting dangerous protocol methods | YES | NO |
| Exploiting inappropriate access permissions | YES | NO |
| Exploiting unprotected functionality | YES | NO |
| Gathering internal information | YES | NO |
| Guessing passwords | YES | NO |
| Reading unencrypted sensitive data | YES | NO |

### 2.3.2 Patch Level

| | **Server Patch Level** | |
|---|---|---|
| | It is possible to exploit known software bugs, although a patch is already available | |
| Attack pattern | Tested[1] | Exploitable[2] |
| Exploiting known application vulnerabilities | YES | NO |

### 2.3.3 Standard Software and Proprietary Applications

| | **Authentication** | |
|---|---|---|
| | The web application provides insufficient means of authentication to protect its resources. | |
| Attack pattern | Tested[1] | Exploitable[2] |
| Bypassing authentication | YES | NO |

| | **Authorization** | |
|---|---|---|
| | An unauthenticated or unprivileged user can gain access to resources that are or should be protected. | |
| Attack pattern | Tested[1] | Exploitable[2] |
| Accessing protected functions | YES | NO |
| Accessing protected resources | YES | NO |

| | **Business Logic Issues*** | |
|---|---|---|
| | An attacker can violate business rules of the application | |
| Attack pattern | Tested[1] | Exploitable[2] |
| Bypassing business rules | YES | NO |

| | **Disclosure of Information** | |
|---|---|---|
| | An attacker can collect information about application internals or the server environment | |
| Attack pattern | Tested[1] | Exploitable[2] |
| Exploiting file extension handling | YES | NO |
| Gathering information from code comments | YES | NO |
| Gathering information from system- and error messages | YES | NO |
| Reading old, backup and unreferenced files | YES | NO |

| | **Client-Side Attacks (Web Browser)** | |
|---|---|---|
| | This vulnerability class is web-related. It covers attacks that target the web browser. | |
| Attack pattern | Tested[1] | Exploitable[2] |
| Cross-Site Request Forgery (XSRF/CSRF) | YES | NO |
| HTML Injection / Cross-Site Scripting (XSS) | YES | NO |
| HTTP Response Splitting / header injection | YES | NO |
| Frame Spoofing | YES | NO |
| Session fixation | YES | NO |

| | **Interpreter Injection and Input Validation Problems** | |
|---|---|---|
| | The application passes input parameters to the database, operating system APIs, or other interpreters without proper validation. | |
| Attack pattern | Tested[1] | Exploitable[2] |
| Accessing the file system | YES | NO |
| Code injection | YES | NO |
| Command injection | YES | NO |
| Format string injection | YES | NO |
| IMAP/SMTP injection | NO | NO |
| LDAP injection | NO | NO |
| ORM injection | NO | NO |
| Overflowing character buffers | YES | NO |
| Path traversal | YES | NO |
| SQL injection | YES | NO |
| SSI injection | YES | NO |
| Server-Side Request Forgery (SSRF) | YES | NO |
| XML injection | YES | NO |
| XPath injection | YES | NO |

| | **State and Session Management** | |
|---|---|---|
| | State- or session-variables are initialized and used incorrectly. | |
| Attack pattern | Tested[1] | Exploitable[2] |
| Enumerating session identifiers | YES | NO |
| Exploiting session state issues | YES | NO |

| | **Management of Trusted Data** | |
|---|---|---|
| | Trusted or application-internal data can be manipulated by an attacker | |
| Attack pattern | Tested[1] | Exploitable[2] |
| Manipulation of application-internal data on the client | YES | NO |
| Reading application-internal/confidential data on the client | YES | NO |

| | **Unneeded / Unsafe Functionality** | | |
|---|---|---|---|
| | The application provides inherently unsafe functionality | | |
| Attack pattern | | Tested[1] | Exploitable[2] |
| Exploiting sample applications | | YES | NO |
| Upload of arbitrary files | | YES | NO |

| | **Unsafe Algorithms** | | |
|---|---|---|---|
| | Use of unsafe algorithms allows compromise of sensitive data | | |
| Attack pattern | | Tested[1] | Exploitable[2] |
| Breaking encryption | | YES | NO |
| Exploiting weak RNG | | NO | NO |

| | **Denial-of-Service** | | |
|---|---|---|---|
| | The service can be rendered unusable by an attacker | | |
| Attack pattern | | Tested[1] | Exploitable[2] |
| Exploiting unlimited resource allocation | | NO | NO |
| Locking customer accounts | | NO | NO |

## 2.4 Actions to be Taken After the Conducted Test

**Remove publicly visible test content from the security assessment.** During the assessment various vulnerability classes are tested by submitting form data or other input fields. Some of the submitted content is publicly visible within the application afterwards.

## 2.5 Disclaimer

This report is strictly confidential and intended for internal, confidential use by the customer. The recipient is obligated to ensure that the highly confidential contents are kept secret on behalf of the organization. The recipient assumes responsibility for further distribution of this document.

In this particular project, a timebox approach was used to define the consulting effort. This means that SEC Consult allotted a prearranged amount of time to identify and document vulnerabilities. Because of this, there is no guarantee that the project has discovered all possible vulnerabilities and risks.

Furthermore, the security check is only an immediate evaluation of the situation at the time the check was performed. An evaluation of future security levels or possible future risks or vulnerabilities may not be derived from it.

During the security assessment, local files (e.g. temporary files, log files, or uploaded programs provided by the contractor to exploit possible vulnerabilities) may have been created on the systems subject to investigation. This occurred, if required, either manually or using an automated vulnerability scanner. After the security assessment, as many of these files as possible were removed by the contractor. A complete removal is nevertheless not always possible due to the approach taken in the security assessment (e.g. due to lack of access to the system or insufficient authorization). Therefore, some subset of these local files may still be present after completion of the assignment, which must be removed by the client as required.

The vulnerabilities identified in this report may lead to a breach of the obligations of the contracting authority as defined in Art. 5 and Art. 32 GDPR. In addition, these could possibly be used by attackers to carry out attacks that may have an implication pursuant to Art. 33 GDPR. In this context, SEC Consult recommends that internal or external legal expertise be obtained if necessary.

# 3 Vulnerability Summary

This chapter contains all identified vulnerabilities in the assessed systems of the company Maybank Shared Services Sdn Bhd.

| Risk assessment | No. of vulnerabilities |
|---|---|
| Critical | 0 |
| High | 0 |
| Medium | 0 |
| Low | 5 |
| **Total** | **5** |

## 3.1 Total Risk Per System

The following table contains a risk assessment for each system which contained security flaws.

| System | Risk |
|---|---|
| eCustody – Web Application | Medium (13.45) |
| **Total** | **Medium (13.45)** |

## 3.2 Risk of Each Vulnerability

The following table contains a risk assessment for the discovered vulnerabilities.

| Vulnerability | System | Risk | Page |
|---|---|---|---|
| Multiple TLS(/SSL) Vulnerabilities | eCustody – Web Application | Low (8.00) | 12 |
| Outdated Software | eCustody – Web Application | Low (8.00) | 16 |
| Improper HTTP Security Header (CSP) | eCustody – Web Application | Low (5.00) | 19 |
| Insecure Cookie Configuration | eCustody – Web Application | Low (5.00) | 21 |
| General Information Disclosure | eCustody – Web Application | Low (5.00) | 23 |
| **Total** | **-** | **Medium (13.45)** | **-** |

# 4 Detailed Analysis

This chapter outlines the attacks and found vulnerabilities in detail.

## 4.1 eCustody – Web Application

### 4.1.1 General Information

The eCustody web application is running on the host with the IP address 172.29.75.65. This system is vulnerable to a low risk of typical application-based vulnerabilities.

### 4.1.2 Whois Information

Whois information was gathered for all in-scope IP addresses to verify whether the given information is correct and that the IP addresses belong to the company being assessed. The following table represents the public whois information from a database which holds the owner of the domains/IP addresses.

```
etRange:        172.16.0.0 - 172.31.255.255
CIDR:           172.16.0.0/12
NetName:        PRIVATE-ADDRESS-BBLK-RFC1918-IANA-RESERVED
NetHandle:      NET-172-16-0-0-1
Parent:         NET172 (NET-172-0-0-0-0)
NetType:        IANA Special Use
OriginAS:
Organization:   Internet Assigned Numbers Authority (IANA)
RegDate:        1994-03-15
Updated:        2013-08-30
```

### 4.1.3 Port Scan Results

| Port number | Protocol | Service | Version |
|---|---|---|---|
| 21 | TCP | FTP | Solaris ftpd |
| 22 | TCP | SSH | SunSSH 1.1.9 (protocol 2.0) |
| 111 | TCP | RPCBIND | 2-4 (RPC #100000) |
| 443 | TCP | HTTPS | - |
| 1720 | TCP | H323Q931? | - |
| 4045 | TCP | NLOCKMGR | 1-4 (RPC #100021) |
| 4118 | TCP | SSL | - |
| 6481 | TCP | SERVICETAGS | - |
| 32771 | TCP | STATUS | 1 (RPC #100024) |

Please be aware that the identification of services and versions might not be accurate in all cases.

### 4.1.4 Multiple TLS(/SSL) Vulnerabilities

TLS/SSL is a protocol that ensures that the communication partners can be certain that the information transmitted has not been tampered with and that it can only be read by the intended recipient. The weaknesses found in the TLS/SSL services partially break those security promises.

During the assessment, the discovered ciphers are vulnerable to the following categories:

**Weak CBC Cipher Algorithm**

In 2013, Timing Attack against several TLS protocol using CBC has been demonstrated by cryptographers. CBC mode is also a known as vulnerable to Plaintext Attack on TLS 1.0, SSL 3.0 and lower.

**Weak SHA1 Hash Algorithm**

In 2005, cryptographers discovered a theoretical vulnerability in the SHA1 hash algorithm that leads to real world attack scenarios. Moreover, in 2017, a successful collision was found. Since then, the collision attacks have improved drastically. While most modern browsers warn their users when connecting to sites that use SHA1 signed certificates, some browsers often does not display such warnings.

**Cipher does not support Perfect Forward Secrecy (PFS)**

Forward secrecy (FS), also known as perfect forward secrecy (PFS), is a feature of specific key agreement protocols that gives assurances that session keys will not be compromised even if long-term secrets used in the session key exchange are compromised.

**Deprecated TLS Version**

The remote server currently supports encryption connection which use the deprecated TLS 1.0 and 1.1 protocol which is affected by several cryptographic flaws. This vulnerability could allow an attacker to decrypt sensitive messages transferred between server and the clients or to perform Man-in-The-Middle (MiTM) attacks.

**Weak 3DES Cipher Algorithm**

Due to the security analysis and practical attacks that found on the Triple Data Encryption Standard (3DES) algorithm such as SWEET32, it is noted that the application should migrate to stronger algorithm such as Advanced Encryption Standard (AES) according to draft of NIST SP 800-67 Revision 2 guideline.

**RC4 Cipher Suites Supported**

The remote host supports the use of RC4 in one or more cipher suites. The RC4 cipher is flawed in its generation of a pseudo-random stream of bytes. Research has shown that RC4 encrypted data can be decrypted within 75 hours if the attacker is able to trick the client into sending a high number of requests (9x2^27) containing static secret data (i.e. cookies) to the victim server. In practice this attack has been proven to work within 52 hours.

#### 4.1.4.1 Proof of Concept

To check the supported protocols and ciphers, the following Nmap script can be used:

```
map --script=ssl-enum-ciphers -Pn -p 443 172.29.75.65
```

The output shows all the ciphers supported by the server. The weak ciphers and deprecated TLS version were highlighted in red as shown below.

```
Nmap scan report for 172.29.75.65
Host is up (0.0041s latency).

PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   SSLv3:
|     ciphers:
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
|       TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
|       TLS_RSA_WITH_RC4_128_SHA (rsa 2048) - C
|       TLS_RSA_WITH_RC4_128_MD5 (rsa 2048) - C
|     compressors:
|       NULL
|     cipher preference: server
|     warnings:
|       64-bit block cipher 3DES vulnerable to SWEET32 attack
|       Broken cipher RC4 is deprecated by RFC 7465
|       CBC-mode cipher in SSLv3 (CVE-2014-3566)
|       Ciphersuite uses MD5 for message integrity
|       Forward Secrecy not supported by any cipher
|   TLSv1.0:
|     ciphers:
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
|       TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
|       TLS_RSA_WITH_RC4_128_SHA (rsa 2048) - C
|       TLS_RSA_WITH_RC4_128_MD5 (rsa 2048) - C
|     compressors:
|       NULL
|     cipher preference: server
|     warnings:
|       64-bit block cipher 3DES vulnerable to SWEET32 attack
|       Broken cipher RC4 is deprecated by RFC 7465
|       Ciphersuite uses MD5 for message integrity
|       Forward Secrecy not supported by any cipher
|_  least strength: C
```

#### 4.1.4.2 Recommended Solution

The TLS configuration of the affected services should be adapted. In general, modern applications should only use TLS 1.2 and TLS 1.3 with ciphers that are rated as secure. For TLS 1.2, the cipher suites providing the following properties should be preferred ('server cipher preference'):

- Perfect Forward Secrecy (PFS) key exchange algorithms.
- Authenticated Encryption with Associated Data (AEAD) ciphers.
- Use of elliptic curve ciphers for the key exchange.

TLS 1.3 prevents usage of many weak ciphers by design. The use of new features introduced by TLS 1.3 (e.g., 0-RTT support) should be evaluated on a case-by-case basis. Algorithms with known security issues should be explicitly blacklisted.

**Several resources for secure TLS configuration can be found here:**

https://ciphersuite.info/cs/?security=secure

https://www.sogis.eu/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.2.pdf

https://tools.ietf.org/html/rfc7525

https://wiki.mozilla.org/Security/Server_Side_TLS

#### 4.1.4.3 Risk Matrix

| Severity / Likelihood | **1** | **4** | **9** | **16** | **25** |
|---|---|---|---|---|---|
| **1** | 1 | 4 | 9 | 16 | 25 |
| **2** | 2 | **8** | 18 | 32 | 50 |
| **3** | 3 | 12 | 27 | 48 | 75 |
| **4** | 4 | 16 | 36 | 64 | 100 |
| **5** | 5 | 20 | 45 | 80 | 125 |

Severity: Identifies the severity / impact of the flaw (1...low - 25...very severe).
Likelihood: Identifies the probability that the flaw can be exploited by an attacker in the defined scope and under the defined attack-specific prerequisites (1...unlikely - 5...very likely).

#### 4.1.4.4 Risk Classification

| | |
|---|---|
| **Attack-specific Prerequisites** | The following attack-specific prerequisites need to be fulfilled for a successful exploitation:<br>• An attacker has internal authenticated access over the internet on port 443. |
| **Likelihood** | It is unlikely that the identified flaw can be exploited by an attacker considering the defined attack-specific prerequisites. |
| **Severity** | An attacker can read and decrypt the data traffic between a user and the server. |
| **Risk** | Low (8) |
| **CVSS v3.1 Vector String** | CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:L |

### 4.1.5 Outdated Software

According to the gathered version information, outdated software that is affected by publicly known vulnerabilities was identified on the system. Once the information about existing vulnerabilities is made public, it has to be assumed that an attacker is able to use publicly available exploit code or analyze the update to develop a working exploit themselves.

The severity depends on the actual vulnerabilities and can range from Denial of Service (DoS) attacks to remote code execution (RCE)

The following known vulnerabilities were identified:

| Vulnerability | ID | Service | Vulnerability Type |
|---|---|---|---|
| Security vulnerabilities in jQuery version 1.3.2.js<br><br>Based on fingerprint at:<br>https://172.29.75.65/main_js/jquery-1.3.2.js | CVE-2011-4969 | TCP/443 | Cross Site Scripting (XSS) |
| Security vulnerabilities in jQuery-ui version 1.7.2<br><br>Based on fingerprint at:<br>https://172.29.75.65/main_js/ui.core.js | CVE-2021-41184 | TCP/443 | Cross Site Scripting (XSS) |

The outdated components were identified based on disclosed version information or by automated tools. Due to the limited timeframe, not all potential vulnerabilities have been verified manually. Hence, there might exist some false positives, especially if the installed software contains backported security patches.

#### 4.1.5.1 Proof of Concept

**Example 1: jQuery v1.3.2.js**

The outdated jQuery can be found here:

```
https://172.29.75.65/main_js/jquery-1.3.2.js
```


*Figure 1: The application used an outdated jQuery version 1.3.2*

**Example 2: jQuery-ui v1.7.2**

The outdated jQuery-ui can be found here:

```
https://172.29.75.65/main_js/ui.core.js
```


*Figure 2: The application used an outdated jQuery-ui v1.7.2.*

### 4.1.5.2 Recommended Solution

It is recommended to update outdated software to be the latest stable version. In addition, a patch management process should be implemented, or an existing process optimized to cover the identified systems. The latest stable version can be found in the following URL:

**Example 2: jQuery**

Update jQuery to the latest stable version available.

```
https://jquery.com/
```

**Example 3: jQuery-ui**

Update jQuery to the latest stable version available.

```
https://jqueryui.com/download/
```

### 4.1.5.3 Risk Matrix

| Severity / Likelihood | **1** | **4** | **9** | **16** | **25** |
|---|---|---|---|---|---|
| **1** | 1 | 4 | 9 | 16 | 25 |
| **2** | 2 | **8** | 18 | 32 | 50 |
| **3** | 3 | 12 | 27 | 48 | 75 |
| **4** | 4 | 16 | 36 | 64 | 100 |
| **5** | 5 | 20 | 45 | 80 | 125 |

Severity: Identifies the severity / impact of the flaw (1...low - 25...very severe).
Likelihood: Identifies the probability that the flaw can be exploited by an attacker in the defined scope and under the defined attack-specific prerequisites (1...unlikely - 5...very likely).

#### 4.1.5.4 Risk Classification

| | |
|---|---|
| **Attack-specific Prerequisites** | The following attack-specific prerequisites need to be fulfilled for a successful exploitation:<br>• An attacker has internal authenticated access over the internet on port 443. |
| **Likelihood** | It is unlikely that the identified flaw can be exploited by an attacker considering the defined attack-specific prerequisites. |
| **Severity** | The attacker can perform Cross-Site Scripting (XSS) attacks on the outdated vulnerable JavaScript component. |
| **Risk** | Low (8) |
| **CVSS v3.1 Vector String** | CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:L |

### 4.1.6 Improper HTTP Security Header (CSP)

Usage of Content-Security-Policy (CSP) headers is an additional layer of protection that can stop attackers from exploiting various security vulnerabilities such as Cross-Site Scripting (XSS) and clickjacking. During the assessment, it was found that the application improperly configured or not set some of the security features that are available in modern web browsers. The CSP header on the web application contains `*`, `unsafe-eval`, and `unsafe-inline` which is dangerous in the `script-src` directive.

#### 4.1.6.1 Proof of Concept

To verify this vulnerability, it is sufficient to open the following URL (no special manipulation of the request is needed) and analyze the HTTP response from the web server:

```
https://172.29.75.65/custody-main-test/common/Login.do
```

Response to the request shows that the mentioned headers are not properly set as according to security best practice:

```
HTTP/1.1 200 OK
Date: Thu, 05 Jan 2023 06:05:09 GMT
Cache-control: no-cache
Pragma: no-cache
Content-length: 298
Content-type: text/html; charset=ISO-8859-1
Expires: Wed, 31 Dec 1969 23:59:59 GMT
Set-cookie: JSESSIONID=h5JCj2jJwXrcVvjBvM9Lkq76wnvMCd6BL27tTTpQCyyq7mjCYH1T!-1781724053;
path=/; secure
X-xss-protection: 1; mode=block
Strict-transport-security: max-age=31536000; includeSubDomains; preload
Content-security-policy: default-src https:; script-src https: 'unsafe-inline' 'unsafe-eval';
style-src https: 'unsafe-inline'
X-content-type-options: nosniff
X-frame-options: SAMEORIGIN
Connection: close
[...]
```

#### 4.1.6.2 Recommended Solution

The Content-Security-Policy header has to be set. However, the value of the header has to be closely monitored to avoid access from unintended sources. The values for `script-src` which are `*`, `unsafe-eval` and `unsafe-inline` lower the security promises drastically and should be avoided as much as possible.

It is recommended to use the new CSP v3 attribute `strict-dynamic` as there is no need for maintaining a whitelist. The suggested values for the Content-Security-Policy headers are as below:

```
Content-Security-Policy: default-src 'none'; script-src 'self'; img-src 'self'; media-src
'self'; connect-src 'self'; style-src 'self'; object-src 'none'; frame-ancestors 'self'
```

**More information about CSP can be found at:**

https://www.w3.org/TR/CSP3/#strict-dynamic-usage

https://csp.withgoogle.com/docs/index.html

https://research.google.com/pubs/pub45542.html

#### 4.1.6.3 Risk Matrix

| Severity / Likelihood | **1** | **4** | **9** | **16** | **25** |
|---|---|---|---|---|---|
| **1** | 1 | 4 | 9 | 16 | 25 |
| **2** | 2 | 8 | 18 | 32 | 50 |
| **3** | 3 | 12 | 27 | 48 | 75 |
| **4** | 4 | 16 | 36 | 64 | 100 |
| **5** | **5** | 20 | 45 | 80 | 125 |

Severity: Identifies the severity / impact of the flaw (1...low - 25...very severe).
Likelihood: Identifies the probability that the flaw can be exploited by an attacker in the defined scope and under the defined attack-specific prerequisites (1...unlikely - 5...very likely).

#### 4.1.6.4 Risk Classification

| | |
|---|---|
| **Attack-specific Prerequisites** | The following attack-specific prerequisites need to be fulfilled for a successful exploitation:<br>• An attacker has internal authenticated access over the internet on port 443. |
| **Likelihood** | It is very likely that the identified flaw can be exploited by an attacker considering the defined attack-specific prerequisites. |
| **Severity** | By its nature, the described issue is not a vulnerability. However, it is an improper configuration of a security feature. |
| **Risk** | Low (5) |
| **CVSS v3.1 Vector String** | CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:L |

### 4.1.7 Insecure Cookie Configuration

The web application uses cookies in an insecure way. If session cookies or other security-relevant cookies can be stolen by an attacker, this could allow the attacker to take over user sessions and user accounts.

The application's cookies are transmitted without the `HttpOnly` flag. The `HttpOnly` flag prevents JavaScript from accessing the cookie data, thus preventing an attacker from directly extracting session tokens or other cookie values in case of cross-site scripting vulnerabilities in the application.

#### 4.1.7.1 Proof of Concept

To verify this vulnerability, it is sufficient to open the following URL (no special manipulation of the request is needed) and analyse the HTTP response from the web server:

```
https://172.29.75.65/custody-main-test/common/Login.do
```

The response to this request shows that the cookie headers does not consist of the mentioned flags.

```
HTTP/1.1 200 OK
Date: Thu, 05 Jan 2023 06:05:09 GMT
Cache-control: no-cache
Pragma: no-cache
Content-length: 298
Content-type: text/html; charset=ISO-8859-1
Expires: Wed, 31 Dec 1969 23:59:59 GMT
Set-cookie: JSESSIONID=h5JCj2jJwXrcVvjBvM9Lkq76wnvMCd6BL27tTTpQCyyq7mjCYH1T!-1781724053;
path=/; secure
X-xss-protection: 1; mode=block
Strict-transport-security: max-age=31536000; includeSubDomains; preload
Content-security-policy: default-src https:; script-src https: 'unsafe-inline' 'unsafe-eval';
style-src https: 'unsafe-inline'
X-content-type-options: nosniff
X-frame-options: SAMEORIGIN
Connection: close
[...]
```

#### 4.1.7.2 Recommended Solution

The server should set the `HttpOnly` and `Secure` flag for all critical cookies.

**More information about CSP can be found at:**

https://owasp.org/www-community/HttpOnly

https://owasp.org/www-community/controls/SecureCookieAttribute

#### 4.1.7.3 Risk Matrix

| Severity / Likelihood | **1** | **4** | **9** | **16** | **25** |
|---|---|---|---|---|---|
| **1** | 1 | 4 | 9 | 16 | 25 |
| **2** | 2 | 8 | 18 | 32 | 50 |
| **3** | 3 | 12 | 27 | 48 | 75 |
| **4** | 4 | 16 | 36 | 64 | 100 |
| **5** | **5** | 20 | 45 | 80 | 125 |

Severity: Identifies the severity / impact of the flaw (1...low - 25...very severe).
Likelihood: Identifies the probability that the flaw can be exploited by an attacker in the defined scope and under the defined attack-specific prerequisites (1...unlikely - 5...very likely).

#### 4.1.7.4 Risk Classification

| | |
|---|---|
| **Attack-specific Prerequisites** | The following attack-specific prerequisites need to be fulfilled for a successful exploitation:<br>• An attacker has internal authenticated access over the internet on port 443. |
| **Likelihood** | It is very likely that the identified flaw can be exploited by an attacker considering the defined attack-specific prerequisites. |
| **Severity** | An attacker potentially get access to the cookie session of the user and utilize the web-application with the same permissions as the user. |
| **Risk** | Low (5) |
| **CVSS v3.1 Vector String** | CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:L |

### 4.1.8 General Information Disclosure

Error messages, comments, and other information in static or dynamically generated web pages or default and customized components of heterogeneous systems often contain critical information an end user should not have access to. In many cases, existing critical vulnerabilities on the system cannot be reliably exploited without further knowledge about the system. Information disclosures facilitate the exploitation of those vulnerabilities. Typical information disclosures are directory listing, accessible configuration files, version number or paths (e.g., path to the web root).

Disclosed information during the assessment as below:

- Oracle default web page

#### 4.1.8.1 Proof of Concept

To verify this vulnerability, it is sufficient to open the following URL (no special manipulation of the request is needed) and analyse the HTTP response from the web server:

```
https://172.29.75.65/
```

The following screenshots show that the Oracle default web page is accessible by end user.

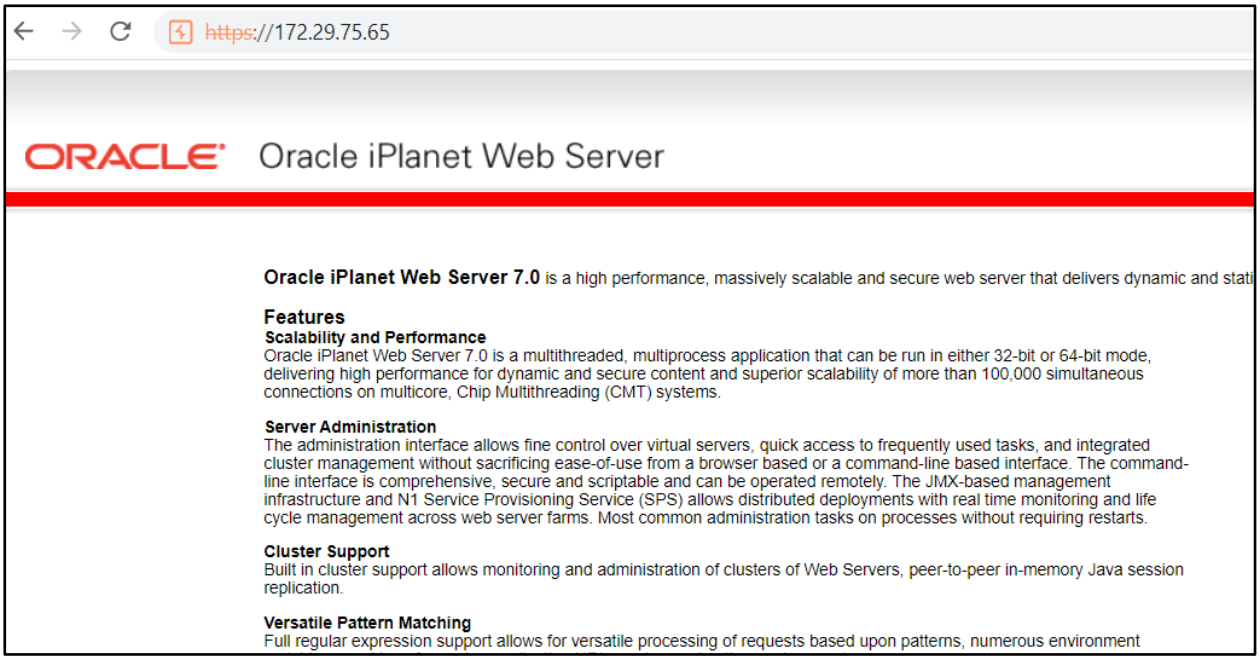



*Figure 3: Default Oracle web page*

#### 4.1.8.2 Recommended Solution

The web server/application should be configured to not disclose error messages or detailed server information to the user.

#### 4.1.8.3 Risk Matrix

| Severity / Likelihood | **1** | **4** | **9** | **16** | **25** |
|---|---|---|---|---|---|
| **1** | 1 | 4 | 9 | 16 | 25 |
| **2** | 2 | 8 | 18 | 32 | 50 |
| **3** | 3 | 12 | 27 | 48 | 75 |
| **4** | 4 | 16 | 36 | 64 | 100 |
| **5** | **5** | 20 | 45 | 80 | 125 |

Severity: Identifies the severity / impact of the flaw (1...low - 25...very severe).
Likelihood: Identifies the probability that the flaw can be exploited by an attacker in the defined scope and under the defined attack-specific prerequisites (1...unlikely - 5...very likely).

#### 4.1.8.4 Risk Classification

| | |
|---|---|
| **Attack-specific Prerequisites** | The following attack-specific prerequisites need to be fulfilled for a successful exploitation:<br>• An attacker has internal authenticated access over the internet on port 443. |
| **Likelihood** | It is very likely that the identified flaw can be exploited by an attacker considering the defined attack-specific prerequisites. |
| **Severity** | The vulnerability is an information disclosure which does not lead to a compromise of the system. However, an attacker can use the gained information during further attacks. |
| **Risk** | Low (5) |
| **CVSS v3.1 Vector String** | CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N |

# 5 Version History

| Version | Date | Status/Changes | Created by | Responsible |
|---|---|---|---|---|
| 1.0 | 2023-01-25 | Final Report | M. Azri | K. Hakimin |

SEC Consult Report Template v3.9.1

# Appendix A Risk Calculation

All security risks discovered were evaluated with a risk score. The risk score is calculated from a risk matrix, which consists of likelihood and severity. The likelihood describes the probability that an attacker discovers the vulnerability and can exploit it. The severity refers to the severity of the vulnerability as well as its impact. As the severity influences the risk stronger than the likelihood, it is included squared in the equation. By multiplying likelihood and severity, the risk score is determined, which allows an assessment of the risks posed by a vulnerability.

| Likelihood \ Severity | 1 | 4 | 9 | 16 | 25 |
|---|---|---|---|---|---|
| 1 | 1 | 4 | 9 | 16 | 25 |
| 2 | 2 | 8 | 18 | 32 | 50 |
| 3 | 3 | 12 | 27 | 48 | 75 |
| 4 | 4 | 16 | 36 | 64 | 100 |
| 5 | 5 | 20 | 45 | 80 | 125 |

To allow for a simple textual description of the risk, the scores were classified into four main categories:

| Risk Score | Risk assessment |
|---|---|
| 1 – 10 | low |
| 11 – 24 | medium |
| 25 – 60 | high |
| 61 – 125 | critical |

## A.1 Definition of the Term Likelihood

The "likelihood" identifies the probability that the flaw can be exploited by an attacker. It is influenced by a combination of the following factors:

- **User Privileges Required / Network access required:**

  In general, the lower the privileges required by an adversary, the higher the likelihood of an exploit. However, this factor heavily depends on the defined attack scope and the assessment goal, e.g. are we assuming that an attacker is already administrator or are we assuming that an attacker starts as an unauthenticated user.

- **User Interaction:**

  The fewer user interactions required (in UI) by the victim(s), the higher the likelihood of an exploitation by an adversary.

- **Attack Complexity / Time Required:**

  The lower the "attack complexity", the higher the likelihood of an exploit. This factor only decreases the likelihood notably if large resources (time/computing power) and / or very large samples of data (e.g. network traffic) are required for a successful exploit.

- **Existence of Public Exploits:**

  If exploits are available to the public (for free or via readily available commercial tools), the likelihood increases significantly.

- **Knowledge about System Internals:**

  The less knowledge required about the systems internals (e.g. access to configurations), the higher the likelihood of an exploit. This factor only decreases the likelihood notably if the security consultant has significantly more knowledge than the assumed attacker.

- **Chaining of Vulnerabilities:**

  In some cases, a vulnerability can only be fully leveraged when chained with other vulnerabilities. Based on the specific attack assumptions and other relevant (non-)existing vulnerabilities, the factor "Chaining of vulnerabilities" can increase or decrease the likelihood significantly in certain cases.

Depending on the specific flaw identified and the defined security assessment scope, certain factors may be weighted more than others.

The **skill level of attacker** is not factored in. We always assume that an attacker is at least as capable as a SEC Consult security consultant.

## A.2 Definition of the Term Severity

The term "severity" defines the impact of the identified flaw. The higher the severity, the higher the costs associated with a successful exploitation of the identified flaw by an adversary.

## A.3 Total Risk

To determine a total risk for a system, a network or an entire corporation, the single risks need to be summed up. However, a simple addition is not applicable as this does not comply with the real behavior of individual vulnerabilities with respect to each other. Two vulnerabilities with the same risk do not result in an overall risk, which is twice as high.

Therefore, the energetic sum formula is used to calculate the total risk:

$$10 \lg (10^{R_1/10} + 10^{R_2/10} + \ldots + 10^{R_n/10}) = R_{total}$$

**R ... Single Risk**
**$R_{total}$ ... Total Risk**

The highest possible risk value is 125. If the total risk exceeds this value, it is reduced to 125.