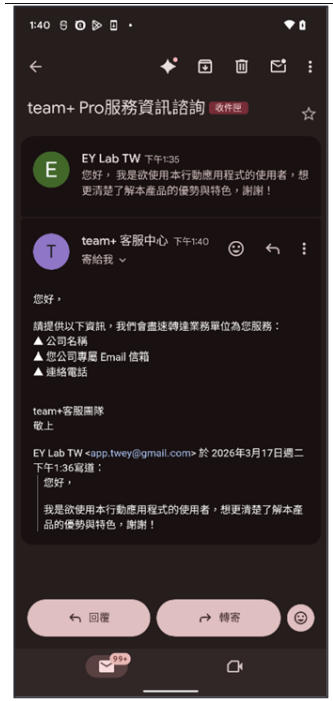
審案實務分享：提升送審案件品質與技術一致性探討

行動應用資安聯盟 技術專家

We are Team MAS

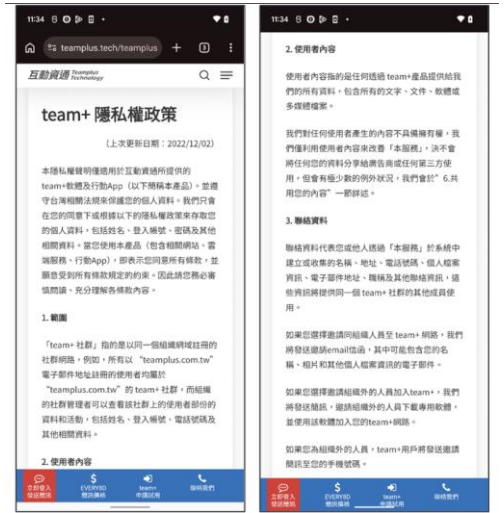
4.1.1.3.1檢測結果說明-建議修正

- 若行動應用程式已發布，檢查行動應用程式是否於應用程式商店 或行動應用程式內，提供聯絡網頁、留言板、電子郵件、電話或 其他類型聯絡方式，並經測試可實際聯絡成功。 若行動應用程式尚未發布，檢查調查表內是否有說明預計提供回 報安全性問題之管道與聯絡方式，並經測試可實際聯絡成功。 如為「是」則符合本項檢測基準；「否」則不符合本項檢測基 準。
- **建議修正原因：佐證資料時間點一樣**



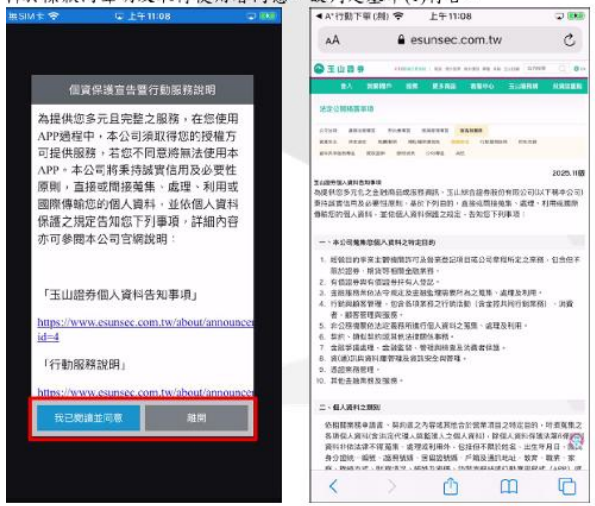
4.1.2.3.2檢測結果說明-不通過

- 行動應用程式提供使用者拒絕儲存敏感性資料之功能，且在 使用者拒絕敏感性資料儲存的情況下，行動應用程式未檢出 儲存敏感性資料。
- **不通過原因：所提供之佐證資料僅可符合4.1.2.3.1(若行動應用程式已發布，檢查行動應用程式所有儲存之敏感性資 料，是否於應用程式商店內聲明，且於行動應用程式內聲明及取得 使用者同意。)**

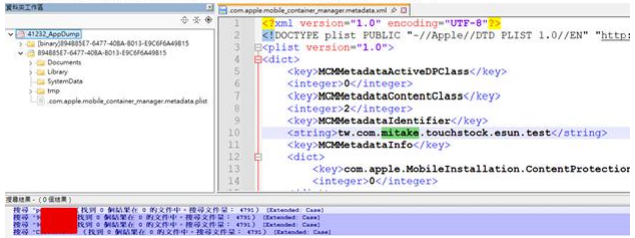


4.1.2.3.2檢測結果說明-可參考

檢測工具：人工檢視、notepad++
經檢查，行動應用程式確實於應用程式內皆有提供個人資料使用條款，所有儲存之敏感性資料於條款內聲明及取得使用者同意，故判定基準(1)符合。



經檢查，應用程式於使用者拒絕儲存敏感性資料情境下，未將個人資料、帳號、密碼、生物辨識資訊等敏感性資料寫入本地儲存區，故判定基準(2)符合。



4.1.2.3.5檢測結果說明-不通過

4.1.2.3.5 行動應用程式應避免將敏感性資料儲存於冗餘檔案或日誌檔案中

- 檢測分類：L3
- 檢測基準：

  (1)檢查行動應用程式是否未檢出將敏感性資料儲存於冗餘檔案。如為「是」則符合本項檢測基準；「否」則不符合本項檢測基準

  (2)檢查行動應用程式是否未檢出將敏感性資料儲存於日誌檔案。如為「是」則符合本項檢測基準；「否」則不符合本項檢測基準

  (3) 檢查行動應用程式是否將敏感性資料儲存於冗餘檔案或日誌檔案且已用安全加密函式保護。如為「是」則符合本項檢測基準； 「否」則不符合本項檢測基準

- 檢測結果：

  ☑符合要求：符合所有檢測基準，或行動應用程式未檢出儲存敏感性資料

  ☐不符合要求：任一檢測基準不符合

行動應用App基本資安檢測基準-4.1.2.3.5

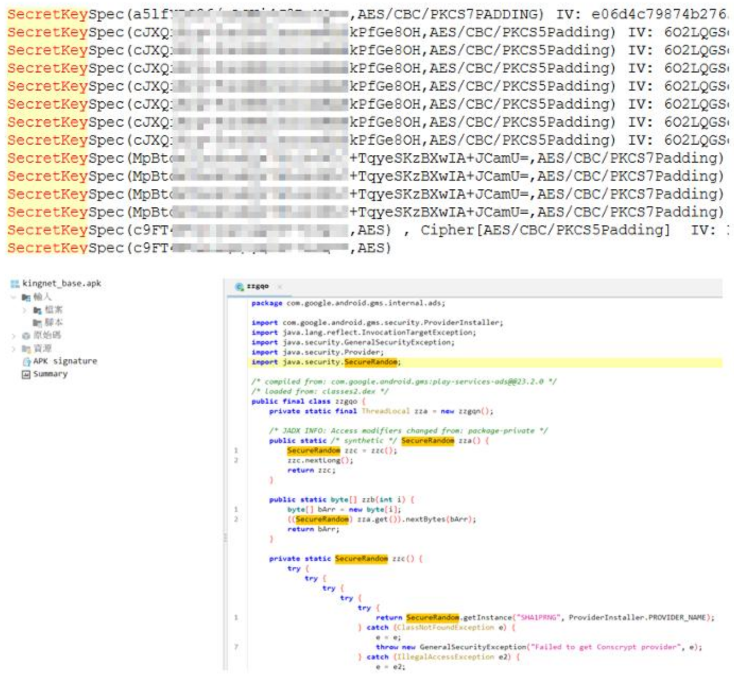
4.1.2.3.5. 行動應用程式應避免將敏感性資料儲存於冗餘檔案或日誌檔案中

檢測編號	4.1.2.3.5
檢測項目	行動應用程式冗餘檔案或日誌檔案敏感性資料儲存限制
檢測分類	L3
檢測依據	「行動應用 App 基本資安規範」4.1.2.3. 敏感性資料儲存、MASVS-STORAGE-2
技術要求	行動應用程式應避免將敏感性資料儲存於冗餘檔案或日誌檔案中
檢測基準	(1) 檢查行動應用程式是否未檢出將敏感性資料儲存於冗餘檔案。如為「是」則符合本項檢測基準；「否」則不符合本項檢測基準。
	(2) 檢查行動應用程式是否未檢出將敏感性資料儲存於日誌檔案。如為「是」則符合本項檢測基準；「否」則不符合本項檢測基準。
	(3) 檢查行動應用程式是否將敏感性資料儲存於冗餘檔案或日誌檔案且已用安全加密函式保護。如為「是」則符合本項檢測基準； 「否」則不符合本項檢測基準。
檢測結果	符合要求：檢測基準(1)、(2)皆符合或符合檢測基準(3)。 不符合要求：檢測基準(1)或(2)不符合且檢測基準(3)不符合。
備註	根據美國國家標準技術研究院建議：於 2023 年 12 月 31 日以後不允許對 Triple DES (3DES、DES3) 加密之支援。故將 3DES 加密從基準中剔除。

4.1.2.3.6檢測結果說明-不通過(1/2)

- 檢查行動應用程式之非冗餘檔案及非日誌檔案內之敏感性資料是否僅採用 金鑰有效長度為 128 位元 (含)以上之先進加密標準 (AES)，或使用 ChaCha20。如為「是」則符合本項檢測基準；「否」則不符合本項檢測基 準。
- 不通過原因：無法證明金鑰長度

經檢測，行動應用程式有對安全敏感性資料進行加密儲存，採用 AES 加密演算法且金鑰長度符合 128bit 以上要求，且有採用 Google 所提供的安全亂數產生函式，故判定符合。



4.1.2.3.6檢測結果說明-不通過(2/2)

- 檢查行動應用程式所使用之加密函式之金鑰是否僅採用符合ANSI X9.17、 FIPS 140-3、NIST SP 800-22 及 SP 800-90A (CAVP Testing: Random Number Generators)至少其中一項之安全的亂數產生函式。如為「是」則 符合本項檢測基準；「否」則不符合本項檢測基準。
- **不通過原因：無法證明符合標準**

測試人員檢視動態分析結果，觀測到行動應用程式多次呼叫 java.security.SecureRandom 產生金鑰所需之隨機位元組，在 Android 系統中，SecureRandom 會調用作業系統提供的安全亂數來源，屬於符合國際安全標準要求的安全亂數生成方式，因此符合此項檢測基準。

```
java.security.SecureRandom called
Generated bytes: -112,68,-13,-122,-64,-39,13,-13,-119,-46,-68,119,-93,89,41,97
java.security.SecureRandom called
Generated bytes: -31,-36,51,32,-99,126,31,-37,-75,11,-102,50,-116,67,-64,-9
java.security.SecureRandom called
Generated bytes: -53,92,-101,16,-76,-7,-66,103,-82,-31,-114,-35,75,-109,0,-90
java.security.SecureRandom called
Generated bytes: -106,-13,-51,24,-17,-81,113,68,-124,-79,-117,-105,-116,11,107,72
java.security.SecureRandom called
Generated bytes: 120,1,17,98,-65,-69,79,95,1,14,52,-121,3,33,-36,81
java.security.SecureRandom called
Generated bytes: 112,21,-102,-107,-100,-23,127,-100,15,96,35,60,-45,-86,-81,75
```

經檢測，行動應用程式有對安全敏感性資料進行加密儲存，採用 AES 加密演算法且金鑰長度符合 128bit 以上要求。金鑰產生過程採用 SecureRandom 作為亂數產生函式，該函式於 Android 平台未指定 Provider 時，預設由 Conscrypt Provider 實作，其底層採用 Google BoringSSL/BoringCrypto 函式庫，已通過 NIST CMVP FIPS 140-2 驗證（Certificate #3753）。BoringCrypto 採用符合 NIST SP 800-90A 規範之 CTR-DRBG 作為系統所有亂數輸出之主要介面（RAND_bytes），包含 SecureRandom 在內之所有亂數需求均透過此機制產生，故判定符合。

```
if (provider != null) {
    try {
        secureRandom = SecureRandom.getInstance("SHA1PRNG", provider);
    } catch (GeneralSecurityException unused) {
    }
} else {
    try {
        provider2 = (Provider) Class.forName("org.conscrypt.Conscrypt").getMethod("newProvider", null).invoke(null, null);
    } catch (Throwable unused2) {
    }
    if (provider2 != null) {
        try {
            secureRandom = SecureRandom.getInstance("SHA1PRNG", provider2);
        } catch (GeneralSecurityException unused3) {
        }
    } else {
        secureRandom = new SecureRandom();
    }
}
secureRandom.nextLong();
return secureRandom;
}
```

4.1.2.3.6檢測結果說明-可參考(1/3)

- 檢查行動應用程式之非冗餘檔案及非日誌檔案內之敏感性資料是否僅採用 金鑰有效長度為 128 位元 (含)以上之先進加密標準 (AES)，或使用 ChaCha20。如為「是」則符合本項檢測基準；「否」則不符合本項檢測基 準。

MitakeJni.c 程式碼 使用 AES_CBC_ENCRYPE 進行文字加密

```
/**
 * AES cbc模式 加密用
 * c1 有用
 * c2 沒用
 */
JNIEXPORT jbyteArray JNICALL Java_com_mitake_jni_MitakeJni_c(
        JNIEnv *env, jclass jcl, jbyteArray c1, jbyteArray c2)
{
    //LOGI("%s", __FUNCTION__);
    jbyteArray OutData;

    if (c1)
    {
        int InDataLen = (int) (*env)->GetArrayLength(env, c1);

        if (InDataLen > 0)
        {
            /* 計算加密後的資料長度 */
            int ResDataLen = AES_BLOCK_SIZE;
            if (InDataLen >= AES_BLOCK_SIZE)
                ResDataLen = ((int) ((double) InDataLen / (double) AES_BLOCK_SIZE) + 1) *

            unsigned char *InDataPadding = malloc(ResDataLen + 1);
            InDataPadding[ResDataLen] = '\0';
            unsigned char *ResData = malloc(ResDataLen + 1);
            ResData[ResDataLen] = '\0';

            /* 拷貝 輸入資料，java_mem to c_mem */
            (*env)->GetByteArrayRegion(env, c1, 0, InDataLen, (jbyte *) InDataPadding);

            /* 填充，固定填充0x10 */
            memset(InDataPadding + InDataLen, 0x10, ResDataLen - InDataLen);

            /* 加密 */
            AES_CBC_ENCRYPT(InDataPadding, ResData, ResDataLen, NULL, NULL, 256);
```

4.1.2.3.6檢測結果說明-可參考(2/3)

- 檢查行動應用程式所使用之加密函式之金鑰是否僅採用符合ANSI X9.17、 FIPS 140-3、NIST SP 800-22 及 SP 800-90A (CAVP Testing: Random Number Generators)至少其中一項之安全的亂數產生函式。如為「是」則 符合本項檢測基準；「否」則不符合本項檢測基準。

使用 NIST 所提供的亂數檢測工具(sts-2.1.2)對 SecureRandom 進行檢測，經檢測符合 NIST SP 800-22。

使用SecureRandom生成亂數

securerandom.java

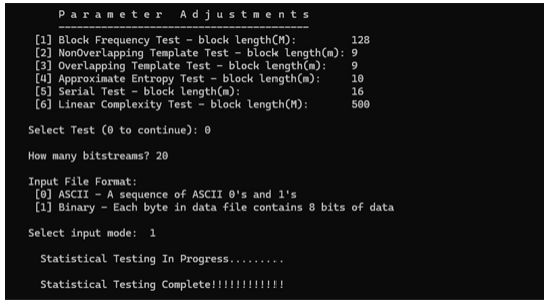
```java
import java.io.FileOutputStream;
import java.security.SecureRandom;

public class SecureRandomTestData {
    public static void main(String[] args) throws Exception {
        SecureRandom sr = new SecureRandom();
        byte[] randomBytes = new byte[20 * 1024 * 1024]; // 20MB
        sr.nextBytes(randomBytes);

        try (FileOutputStream fos = new FileOutputStream("randomData20.bin")) {
            fos.write(randomBytes);
        }
        System.out.println("Generated 20MB of random data for testing.");
    }
}
```

行動應用程式使用SecRandomCopyBytes亂數函式。

使用NIST所提供的亂數檢測工具(sts-2.1.2)對 SecRandomCopyBytes 進行檢測，經檢測符合 NIST SP 800-22。



檢測結果SecRandomCopyBytes符合NIST SP 800-22

4.1.2.3.6檢測結果說明-可供參考(3/3)

- 檢查行動應用程式所使用之加密函式之金鑰是否僅採用符合ANSI X9.17、 FIPS 140-3、NIST SP 800-22 及 SP 800-90A (CAVP Testing: Random Number Generators)至少其中一項之安全的亂數產生函式。如為「是」則 符合本項檢測基準；「否」則不符合本項檢測基準。

經檢視開發商提供的程式碼片段，確認行動應用程式使用 SecureRandom 產生金鑰所需之亂數。

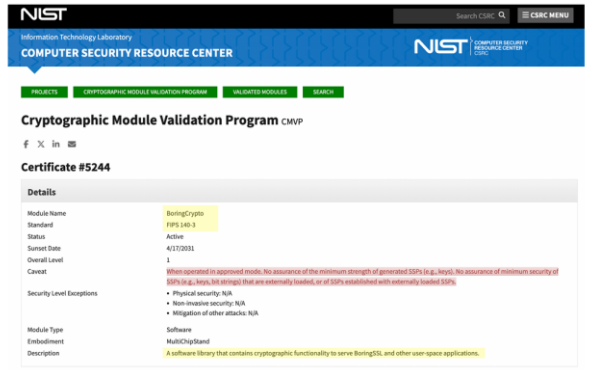
附錄 B：AES 金鑰與 IV 產生

```
private void genAESKey() throws Exception {
    // 1. 使用 CSPRNG 產生 128-bit AES 金鑰
    byte[] aesKey = new byte[16];
    SecureRandom secureRandom = new SecureRandom();
    secureRandom.nextBytes(aesKey);

    // 2. 使用 CSPRNG 產生 12-byte GCM IV，Base64 編碼後存入 SharedPreferences
    byte[] generated = secureRandom.generateSeed(12);
    String iv = Base64.encodeToString(generated, Base64.DEFAULT);
    prefsHelper.setIV(iv);  // → 寫入 PREF_KEY_IV

    // 3. AES 金鑰以 KeyStore 中的 RSA 公鑰加密後，存入 SharedPreferences
    String encryptAESKey = encryptRSA(aesKey);
    prefsHelper.setAESKey(encryptAESKey);  // → 寫入 PREF_KEY_AES
}
```

依據 NIST Cryptographic Module Validation Program (CMVP)Certificate #5244，BoringCrypto 已通過 FIPS 140-3 驗證。另 Android 預設 Security Provider (Conscrypt)之 cryptographic backend 採用 BoringSSL 實作，因此可確認測試標的所使用之亂數生成機制係建立於具備現代密碼學安全設計之 cryptographic implementation 上，因此符合此項檢測基準。



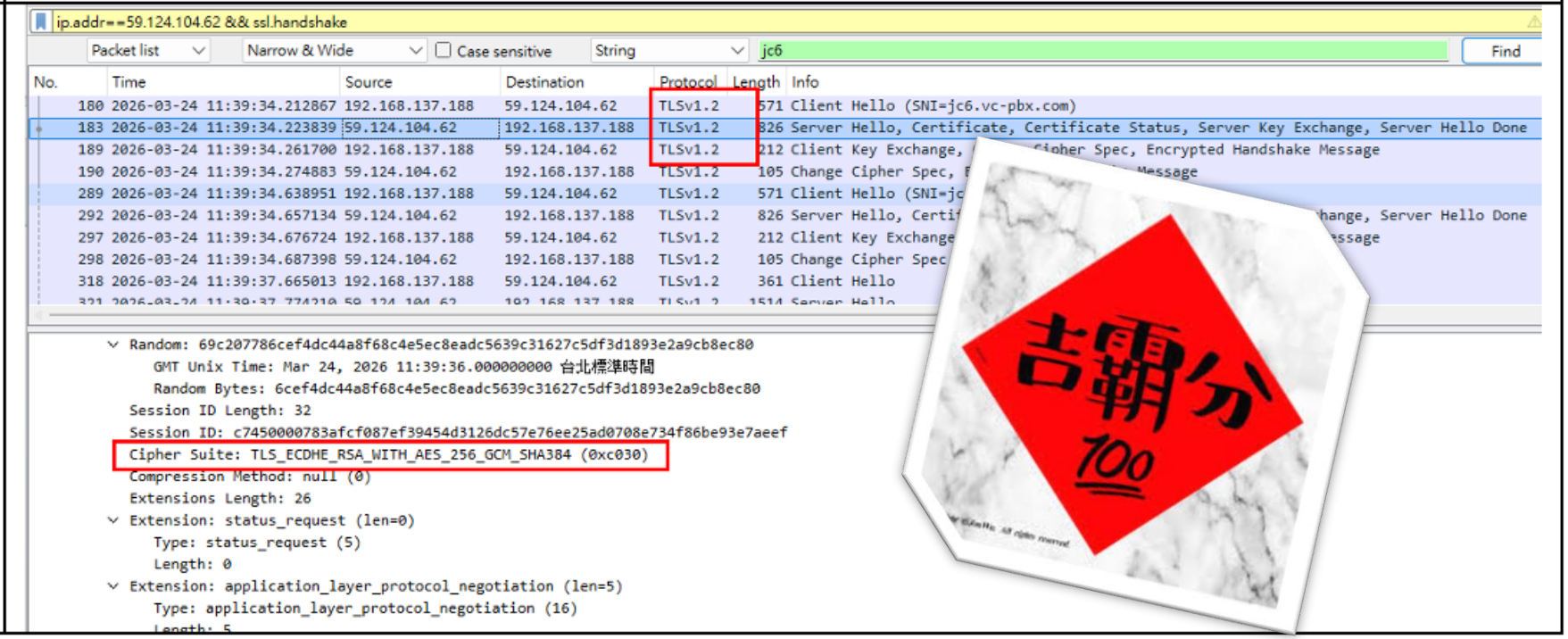
4.1.2.3.11檢測結果說明-不通過

- 檢查行動應用程式是否在輸入敏感性資料欄位中將鍵盤快取機制關閉： 檢查行動應用程式於使用者輸入敏感性資料時，是否未自動修正且未帶入可能字串。如為「是」則符合檢測基準;「否」則不符合本項檢測基準。 檢測結果 符合要求：符合檢測基準，或行動應用程式未檢出蒐集敏感性資料
- **不通過原因：所檢附之佐證資料皆有遮蔽，無法顯示是否未自動修正且未帶入可能字串。**

4.1.2.3.11檢測結果說明-可參考

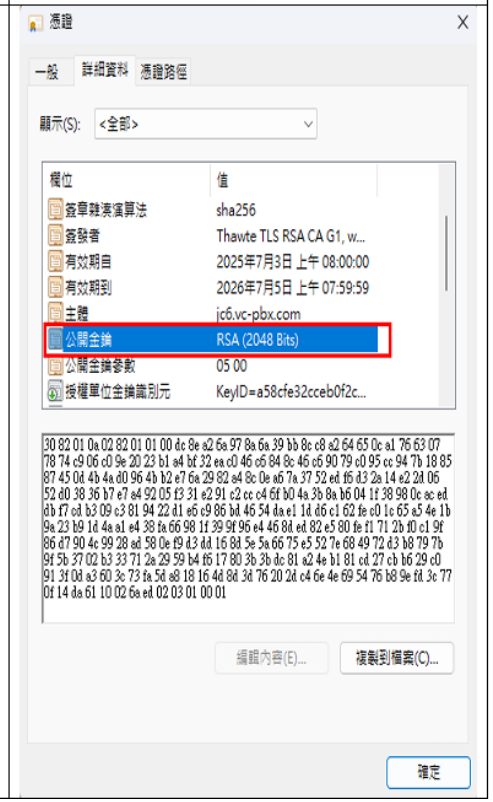
4.1.2.4.1. 檢測結果說明-可參考(1/3)

1. 行動應用程式僅採用 TLS 1.2（含）以上版本 加密協定傳輸資料，如本項檢測結果截圖說明之 項次1所示。符合基準

項次	1
圖示說明	

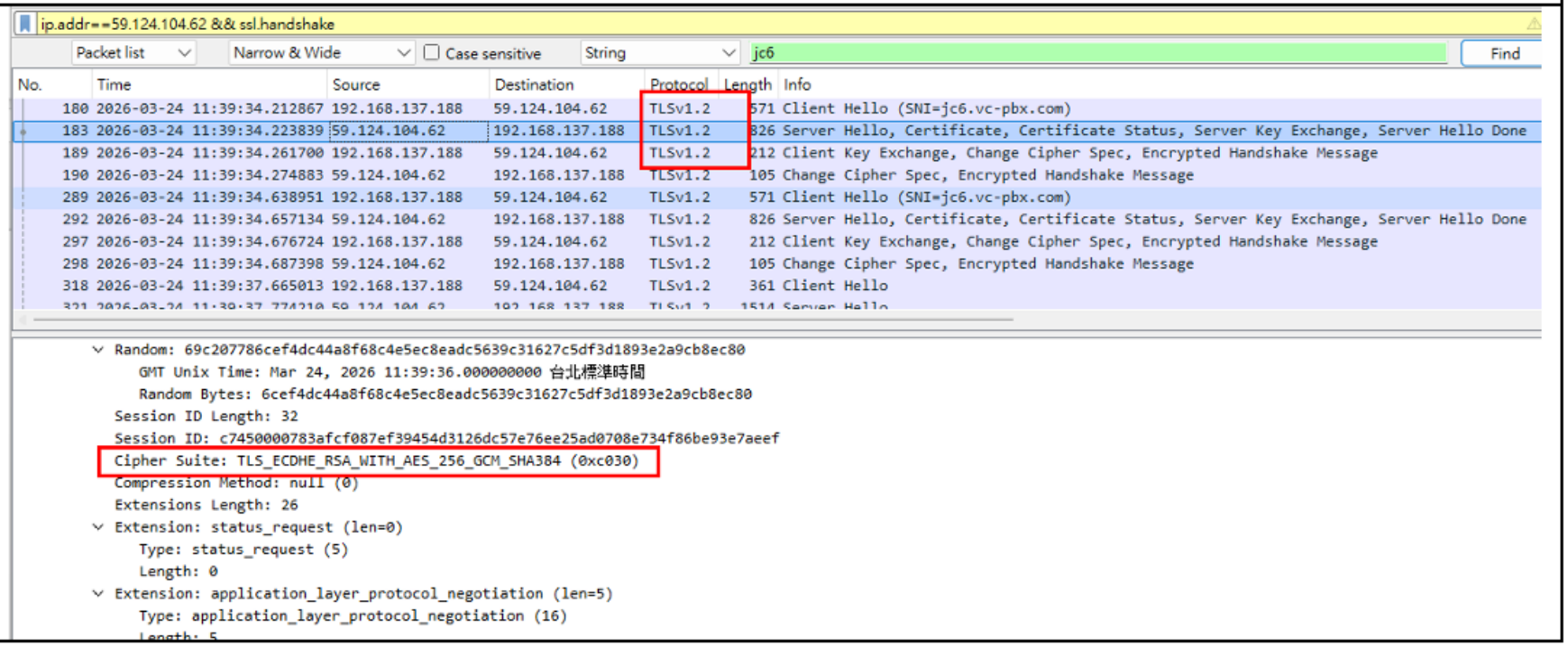
4.1.2.4.1. 檢測結果說明-可參考(2/3)

2. 行動應用程式僅採用金鑰有效長度為 2048 位 元（含）以上之RSA加密演算法，如本項檢測結 果截圖說明之項次2所示。符合基準

項次	2
圖示說明	

4.1.2.4.1. 檢測結果說明-可參考(3/3)

3. 行動應用程式僅採用金鑰有效長度為 128 位 元（含）以上之進階加密標準（AES），如本項檢 測結果截圖說明之項次1所示。

項次	1
圖示說明	

4.1.2.5.2.檢測結果說明-不通過

- (2) 檢查在使用者拒絕敏感性資料分享的情況下，行動應用程式是否未分享敏感性資料。如為「是」則符合本項檢測基準；「否」則不符合本項檢測基準。
- **不通過原因：所提供之佐證資料僅可符合4.1.2.5.1(若行動應用程式已發布，檢查行動應用程式所有儲存之敏感性資料，是否於應用 程式商店內聲明，且於行動應用程式內聲明及取得使用者同意。)**

4.1.2.5.2.檢測結果說明-可參考

檢測工具：人工檢視、iTunes
經檢查，行動應用程式「玉山證券『A+行動下單』」未發現分享敏感性資料，故判定此項符合。

4.1.3.2.2檢測結果說明-不通過

- 檢查行動應用程式於交易後，是否提供查詢交易記錄之管道，且交易記錄至少包含交易商品名 稱、交易時間及交易金額之記錄。如為「是」則符合檢測基準；「否」則不符合檢測基準。
- 不通過原因：所提供之佐證資料沒有包括交易時間

4.1.3.2.2檢測結果說明-可參考

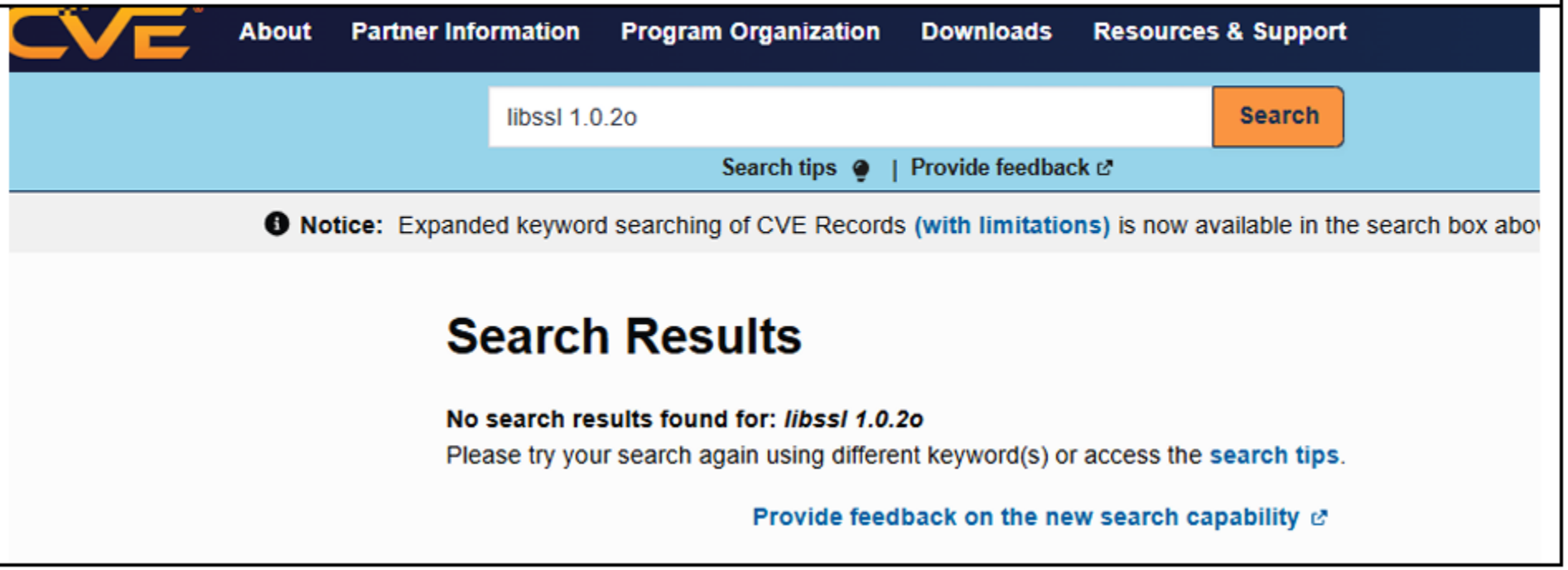
4.1.5.3.1. 檢測結果說明(1/3)

- 行動應用程式引用之第三方函式庫不存在已知 安全性漏洞，如本項檢測結果截圖說明之項次 1 至項次7 所示

項次	1		
圖示說明	15	引用函式庫名稱、版本、來源（包含作業系統內建及第三方函式庫）	<libssl / 1.0.2o / GPL> <libcrypto / 1.0.2o / GPL> <x264 / 1.4.2 / GPL> <libavcodec / 56.35.101 / LGPL> CoreTelephony AVFoundation MobileCoreService

4.1.5.3.1. 檢測結果說明(2/3)

- 行動應用程式引用之第三方函式庫不存在已知 安全性漏洞，如本項檢測結果截圖說明之項次 1 至項次7 所示

項次	3
圖示說明	

4.1.5.3.1. 檢測結果說明(3/3)

- 行動應用程式引用之第三方函式庫不存在已知 安全性漏洞，如本項檢測結果截圖說明之項次 1 至項次7 所示

項次	6
圖示說明	

4.1.5.4.1檢測結果說明-不通過

- 檢查行動應用程式是否針對預期使用者輸入之字串驗證型別， 如欄位本身需要接受特殊字元，亦屬於可預期的輸入字串型別。 如為「是」則符合本項檢測基準；「否」則不符合本項檢測基準。
- 不通過原因：無法證明是否有輸入特殊字元

4.1.5.4.1檢測結果說明-可供參考(1/2)

- 可證明對特殊字元之輸入有所限制或檢驗

4.1.5.4.1檢測結果說明-可供參考(1/2)

(1). 檢查行動應用程式是否針對預期使用者輸入之字串驗證型別，如欄位本身需要接受特殊字元，亦屬於可預期的輸入字串型別。如為「是」則符合本項 檢測基準；「否」則不符合本項檢測基準。

頁面	欄位	是否檢查/限制輸入
註冊帳號	身分證字號	是 (格式檢查)
	姓名	是 (無法輸入)
	電話號碼	是 (數字鍵盤)
	電子信箱	是 (格式檢查)
	密碼	為可預期的輸入字串型別
我的親友	姓名	是 (無法輸入)
	身分證字號	是 (格式檢查)
	電話號碼	是 (數字鍵盤)

4.1.5.4.1檢測結果說明-可供參考(2/2)

(2). 檢查行動應用程式是否針對使用者輸入字串驗證長度。如為「是」則符合本 項檢測基準；「否」則不符合本項檢測基準。

頁面	欄位	是否檢查/字串長度
註冊帳號	身分證字號	是 (10)
	姓名	是 (2-50)
	電話號碼	是 (10)
	電子信箱	是 (255)
	密碼	是 (6-20)
我的親友	姓名	是 (2-50)
	身分證字號	是 (10)
	電話號碼	是 (10)

4.1.5.4.2檢測結果說明-可供參考

- (1). 檢查行動應用程式是否防護使用者輸入 SQL Injection 字串之設計。如為「是」則符合本項檢測基準；「否」則不符合本項檢測基準。

頁面	欄位	是否注入成功
註冊帳號	身分證字號	否
	姓名	否
	電話號碼	否
	電子信箱	否
	密碼	否
我的親友	姓名	否
	身分證字號	否
	電話號碼	否

感謝聆聽

也期待您的回饋，讓我們共創更有公信力的制度