CPL-01:行動應用 App 應設計並實作適當身分認證機制，並依使用者身分授權，以防止敏感資料被非授權人員存取。

安全程式碼範例(iOS：Swift)-參考 CPF-13 安全範例：使用 OAuth2。

```
通常會使用第三方帳號認證，茲提供 OAuth2 範例，以取得第三方的 Token，當作認證 ID
OAuth 2 應用範例

1.註冊並宣告服務
let googleConfig = GoogleConfig(
clientId: "YOUR GOOGLE CLIENT ID",                                  // [1] Define a
Google configuration
scopes:["https://www.googleapis.com/auth/drive"])                   // [2] Specify
scope

let gdModule = AccountManager.addGoogleAccount(googleConfig)        // [3] Add it to
AccountManager
self.http.authzModule = gdModule                                    // [4] Inject
the AuthzModule
                                                                    // into the HTTP
layer object

let multipartData = MultiPartData(data: self.snapshot(),            // [5] Define
multi-part
          name: "image",
          filename: "incognito photo",
          mimeType: "image/jpg")
let multipartArray =  ["file": multipartData]

self.http.POST("https://www.googleapis.com/upload/drive/v2/files",    // [6]
Upload image
               parameters: multipartArray,
       completionHandler: {(response, error) in
  if (error != nil) {
    self.presentAlert("Error", message: error!.localizedDescription)
  } else {
    self.presentAlert("Success", message: "Successfully uploaded!")
  }
})
```

```
2.註冊 App 應用連結
<key>CFBundleURLTypes</key>
<array>
    <dict>
        <key>CFBundleURLSchemes</key>
        <array>
            <string>com.raywenderlich.Incognito</string>
        </array>
    </dict>
</array>
```

```
3.接收 App 應用連結
func application(application: UIApplication,
  openURL url: NSURL,
  sourceApplication: String?,
  annotation: AnyObject?) -> Bool {
    let notification = NSNotification(name: AGAppLaunchedWithURLNotification,
      object:nil,
      userInfo:[UIApplicationLaunchOptionsURLKey:url])
    NSNotificationCenter.defaultCenter().postNotification(notification)
    return true
}
```

CPM-03:行動應用 App 連線使用交談識別碼，應實作具備逾時失效(Session time-out)機制。

安全程式碼範例(iOS：Objective-C)

- *iOS: local-session-timeout code 實作*
- *1.啟動時間計算*

```
    [myApp setLoginDate:[NSDate date]];
- (void) setLoginDate: (NSDate *)indate
{
    loginDate = indate;
}
```

- *2.檢查是否在安全時間內*

```
[NSString stringWithFormat:@"isValidDate=%d",[myApp
isLoginDateValid],
- (BOOL) isLoginDateValid
{
    if (!loginDate)
        return FALSE;
    //valid for 60 seconds
    NSDate *max date = [loginDate dateByAddingTimeInterval:60];
    return
        (([loginDate compare:[NSDate date]] == NSOrderedAscending)
&&
        ([max date compare:[NSDate date]] ==
NSOrderedDescending));
}
```

CPM-05:行動應用程式應使用憑證綁定(Certificate Pinning)方式驗證並確保連線之伺服器為行動應用程式開發人員所指定。

安全程式碼範例(iOS：Objective-C)

完整的 SSL Pinning 範例(以 https://www.owasp.org 與 https://gca.nat.gov.tw)
準備：取出 SSL 網站的公鑰

1.由網站獲取公鑰，以利 https ssl ceriticate 認證時比對

```
  ex +'/BEGIN CERTIFICATE/,/END CERTIFICATE/p' <(echo | openssl s client -
showcerts -connect www.owasp.org:443) -scq > file.crt
  ex +'/BEGIN CERTIFICATE/,/END CERTIFICATE/p' <(echo | openssl s client -
showcerts -connect gca.nat.gov.tw:443) -scq > gca.crt
```

2.然後轉成 der

```
  openssl x509 -outform der -in owasp.crt -out owasp2.der
  openssl x509 -outform der -in gca.crt -out gca2.der
```

3.附加為專案內容，以用於比對

範例一：元件 NSURLConnectionDelegate

1.實作 NSURLConnectionDelegate

2.建立 NSURLConnection

```
   urlconnection = [[NSURLConnection alloc] initWithRequest:theRequest
delegate:self];
```

3.監聽、驗證 SSL Pinning

```
- (BOOL)connection:(NSURLConnection *)connection
canAuthenticateAgainstProtectionSpace:(NSURLProtectionSpace *)protectionSpace {
return [protectionSpace.authenticationMethod
isEqualToString:NSURLAuthenticationMethodServerTrust];
}
//可在 didReceiveAuthenticationChallenge 進行憑證驗證
//或者實作 willSendRequestForAuthenticationChallenge 自行更早決定成功/失敗/驗證等
//於 willSendRequestForAuthenticationChallenge 可回傳以下幾種方式
// 1.useCredential:forAuthenticationChallenge:
// 2.continueWithoutCredentialForAuthenticationChallenge:
// 3.cancelAuthenticationChallenge:
// 4.performDefaultHandlingForAuthenticationChallenge:
// 5.rejectProtectionSpaceAndContinueWithChallenge:
- (void)connection:(NSURLConnection *)connection
didReceiveAuthenticationChallenge:(NSURLAuthenticationChallenge *)challenge {
   //Enable Pinning
   if ([[[challenge protectionSpace] authenticationMethod] isEqualToString:
NSURLAuthenticationMethodServerTrust]){
       do
       {
           SecTrustRef serverTrust = [[challenge protectionSpace] serverTrust];
           if(nil == serverTrust)
           {
               [self AppendLog:@"serverTrust is nil"];
               break; /* failed */
           }
           OSStatus status = SecTrustEvaluate(serverTrust, NULL);
           if(!(errSecSuccess == status))
           {
               [self AppendLog:@"SecTrustEvaluate status is errSecSuccess"];
               break; /* failed */
           }
           SecCertificateRef serverCertificate =
SecTrustGetCertificateAtIndex(serverTrust, 0);
```

```
          if(nil == serverCertificate)
            {
                [self AppendLog:@"serverCertificate is nil"];
                break; /* failed */
            }
            CFDataRef serverCertificateData =
SecCertificateCopyData(serverCertificate);
            if(nil == serverCertificateData)
            {
                [self AppendLog:@"serverCertificateData is nil"];
                break; /* failed */
            }
            const UInt8* const data = CFDataGetBytePtr(serverCertificateData);
            const CFIndex size = CFDataGetLength(serverCertificateData);
            NSData* cert1 = [NSData dataWithBytes:data length:(NSUInteger)size];
            if(nil == cert1)
            {
                [self AppendLog:@"server cert is nil"];
                break; /* failed */
            }
            NSString *file = [[NSBundle mainBundle] pathForResource:@"domain"
ofType:@"der"];
            NSData* cert2 = [NSData dataWithContentsOfFile:file];
            if(nil == cert1 || nil == cert2)
            {
                [self AppendLog:@"local pinning cert is nil"];
                break; /* failed */
            }
            const BOOL equal = [cert1 isEqualToData:cert2];
            if(!equal){
                [self AppendLog:@"certs not equal"];
                break; /* failed */
            }
            [self AppendLog:@"certs are GOOD"];
            // The only good exit point
            return [[challenge sender] useCredential: [NSURLCredential
credentialForTrust: serverTrust]
                                          forAuthenticationChallenge: challenge];

        } while(0);
    }
    // Bad dog
    return [[challenge sender] cancelAuthenticationChallenge: challenge];
}
```

範例二：元件 NSURLSession

1.建立 NSURLSession，並啟動之 by NSURLSessionDataTask

```
•    NSURLSessionConfiguration *sessionConfig = [NSURLSessionConfiguration
  defaultSessionConfiguration];
•    urlsession = [NSURLSession sessionWithConfiguration:sessionConfig
  delegate:self delegateQueue:nil];
•    [[urlsession dataTaskWithURL:url completionHandler:^(NSData *  Nullable
  data, NSURLResponse *  Nullable response, NSError *  Nullable error) {
•        // response management code
•        if (error)
•        {
•            [self AppendLog:[NSString stringWithFormat:@">> dataTask Error:%@",
  error.description]];
•        }
•        else
•        {
•            [self AppendLog:[NSString stringWithFormat:@">> dataTask got data
  %lu", data.length]];
•        }
•    }] resume];
```

2.驗證 SSL Pinning

```
• -(void)URLSession:(NSURLSession *)session
  didReceiveChallenge:(NSURLAuthenticationChallenge *)challenge
```

```
completionHandler:(void (^)(NSURLSessionAuthChallengeDisposition,
NSURLCredential *  Nullable))completionHandler {
    // Get remote certificate
    SecTrustRef serverTrust = challenge.protectionSpace.serverTrust;
    SecCertificateRef certificate = SecTrustGetCertificateAtIndex(serverTrust,
0);
    // Set SSL policies for domain name check
    NSMutableArray *policies = [NSMutableArray array];
    [policies addObject:(  bridge transfer id)SecPolicyCreateSSL(true,
(  bridge CFStringRef)challenge.protectionSpace.host)];
    SecTrustSetPolicies(serverTrust, (  bridge CFArrayRef)policies);
    // Evaluate server certificate
    SecTrustResultType result;
    SecTrustEvaluate(serverTrust, &result);
    BOOL certificateIsValid = (result == kSecTrustResultUnspecified || result
== kSecTrustResultProceed);
    // Get local and remote cert data
    NSData *remoteCertificateData =
CFBridgingRelease(SecCertificateCopyData(certificate));
    NSString *pathToCert = [[NSBundle mainBundle]pathForResource:@"domain"
ofType:@"der"];
    NSData *localCertificate = [NSData dataWithContentsOfFile:pathToCert];
    // The pinnning check
    if ([remoteCertificateData isEqualToData:localCertificate] &&
certificateIsValid) {
        NSURLCredential *credential = [NSURLCredential
credentialForTrust:serverTrust];
        [self AppendLog:@"URLSession pinning completionHandler
credential(GOOD)"];
        completionHandler(NSURLSessionAuthChallengeUseCredential, credential);
    } else {
        [self AppendLog:@"URLSession pinning completionHandler BAD(NULL)"];
        completionHandler(NSURLSessionAuthChallengeCancelAuthenticationChalleng
e, NULL);
    }
}
```

CPQ-05:行動應用 App 應實作過濾使用者輸入及伺服器端傳入資料中易導致 SQL injection 之字串。

不安全程式碼範例(iOS：Objective-C)

sqlite 錯誤方式

```
- (IBAction) insertUnsafe : (id) sender
{
    BOOL rst = [self executeSQL:[NSString stringWithFormat:@"INSERT INTO
Setting(SKey,SValue) VALUES('%@','%@')", self.edit key.text,
self.edit value.text]];
}
```

安全程式碼範例(iOS：Objective-C)

資料庫：sqlite 正確方式，即 CPQ-03：Query parameterization

```
- (IBAction) insertSafe : (id) sender
{
    sqlite3 stmt *statement;
    BOOL DONE = TRUE;
    char *sql = "INSERT INTO Setting(SKey,SValue) VALUES(?,?)";

    if (sqlite3 prepare v2(database,sql,-1,&statement,NULL) == SQLITE OK) {
        sqlite3 bind text(statement,1,[self.edit key.text UTF8String],-1,
SQLITE TRANSIENT);
        sqlite3 bind text(statement,2,[self.edit value.text UTF8String],-1,
SQLITE TRANSIENT);
        if (sqlite3 step(statement) != SQLITE DONE) {
            DONE = FALSE;
            NSLog(@"Error");
        }
    }
    else
        DONE = FALSE;
    sqlite3 finalize(statement);
}
```

iOS-01:謹慎使用 Keychain 儲存密碼(Use the Keychain carefully)。

不安全程式碼範例

使用其他屬性的參數，造成可能的漏洞，例如備份檔被其他機器還原。

安全程式碼範例(Objective-C)

使用嚴格參數，以免資料洩漏

```
• // Protect the keychain entry so it's only valid when the device is
  unlocked.
• [dictionary setObject:( bridge
  id)kSecAttrAccessibleWhenUnlockedThisDeviceOnly forKey:( bridge
  id)kSecAttrAccessible];
```

iOS-02：為保護敏感資料不被以截圖方式儲存於檔案系統，行動應用 App 使用 API 設定或編寫程式阻擋敏感資料區快照功能(Snapshots)或以覆蓋方式清除。

安全程式碼範例(Objective-C)

- 提供蓋掉螢幕的範例碼、提供隱藏欄位的建議方法
- 先製作滿版圖片 secure-image.png (320x568 for iPhone)

```
- (void)applicationDidEnterBackground:(UIApplication *)application {
    // Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.
    // If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.
        if (!self.backgroundImage){
            UIImageView *myBanner = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"secure-image.png"]];
            self.backgroundImage = myBanner;
        }
        [self.window addSubview:self.backgroundImage];
    // you can hide security field here
    // 隱藏欄位
    // [viewControl.secure field setHidden:TRUE];
}
- (void)applicationWillEnterForeground:(UIApplication *)application {
    // Called as part of the transition from the background to the inactive state; here you can undo many of the changes made on entering the background.
    if (self.backgroundImage)
        [self.backgroundImage removeFromSuperview];
    // you should visi security field here
    // 回復已隱藏欄位
    // [viewControl.secure field setHidden:FALSE];
}
```

iOS-04:啟用 App Transport Security(ATS)設定。

不安全程式碼範例

範例一：全面取消 ATS

```
NSAppTransportSecurity

    NSAllowsArbitraryLoads = YES
```

範例二：某網域降級 TLS

```
NSAppTransportSecurity

    NSExceptionDomains

        "less-secure.example.com"

            NSExceptionRequiresForwardSecrecy = NO

            NSExceptionMinimumTLSVersion = "TLSv1.0"
```

範例三：某網域取消 ATS

```
NSAppTransportSecurity

    NSAllowsArbitraryLoads = NO  // Shown for clarity; this is the
default

    NSExceptionDomains

        "secure-server-i-control.example.com"

        NSExceptionAllowsInsecureHTTPLoads = YES

        NSExceptionRequiresForwardSecrecy = NO

        NSExceptionMinimumTLSVersion = "TLSv1.0"
```

Server-01 與行動應用 App 連接之所有後端服務伺服器(包含網頁、資料庫及中介等)作業系統應有效強化及進行安全設定配置，並持續進行安全性程式修補。

不安全程式碼範例(網站主機：Server)

例如過時 OpenSSL 版本 OpenSSL/1.0.1e 與太多資訊洩漏。

```
telnet ---.---.---.tw 80
Trying 0.0.0.0……..….
Connected to ---.---.---.tw.
Escape character is '^]'.
HEAD / HTTP/1.0
```

```
HTTP/1.1 200 OK
Date: Sat, 06 Aug 2016 01:57:04 GMT
Server: Apache/2.2.29 (Unix) mod_ssl/2.2.29 OpenSSL/1.0.1e-fips DAV/2 PHP/5.3.29
Last-Modified: Thu, 30 Jun 2016 06:22:49 GMT
ETag: "bc16f1-7f8-53678e6296040"
Accept-Ranges: bytes
Content-Length: 2040
Connection: close
Content-Type: text/html


遺失與主機的連線。
```

安全程式碼範例(網站主機：Server)

較少資訊洩漏

```
telnet www.------.com.tw 80
HTTP/1.0 200 OK
Date: Sat, 30 Jul 2016 05:52:56 GMT
P3P: policyref="http://info.------.com/w3c/p3p.xml", ... Cache-Control: max-age=3600,
public
Vary: Accept-Encoding
Content-Type: text/html; charset=UTF-8
Server: ATS
```

安全程式碼範例(網站主機：Server .NET)

啟用 HSTS 機制

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <rewrite>
            <rules>
                <rule name="HTTP to HTTPS redirect" stopProcessing="true">
                    <match url="(.*)" />
                    <conditions>
                        <add input="{HTTPS}" pattern="off" ignoreCase="true" />
                    </conditions>
                    <action type="Redirect" url="https://{HTTP HOST}/{R:1}"
                        redirectType="Permanent" />
                </rule>
            </rules>
            <outboundRules>
                <rule name="Add Strict-Transport-Security when HTTPS" enabled="true">
                    <match serverVariable="RESPONSE Strict Transport Security"
                        pattern=".*" />
                    <conditions>
                        <add input="{HTTPS}" pattern="on" ignoreCase="true" />
                    </conditions>
                    <action type="Rewrite" value="max-age=31536000" />
                </rule>
            </outboundRules>
        </rewrite>
    </system.webServer>
</configuration>
```