情報処理安全確保支援士への道(9)：令和７年 春 午後問題 問3を解いてみる(設問1(1)編)

ルチルMike
2025年8月10日 23:00

問題冊子・配点割合・解答例・採点講評（2025年度、令和7年度） | 試験情報 | IPA 独立行政法人 情...

情報処理推進機構（IPA）の「問題冊子・配点割合・解答例・採点講評（2025年度、令和7年度）」に関する情報です。

www.ipa.go.jp

目次

設問１（１）

ステップ1：脆弱性の核心を理解する

ステップ2：可能な攻撃手法を特定する

ステップ3：具体的な攻撃手順を組み立てる

すべて表示

要旨

問3を解くには、モバイルアプリとWebサービスが連携する現代的なシステムにおける、複合的なセキュリティ知識が求められます

前提となる知識分野

Web通信のセキュリティ (HTTPS/TLS) 通信の盗聴・改ざん

- **TLS/SSLの基本**: なぜHTTPS通信が安全なのか、その中核である「暗号化」「サーバ認証」「改ざん検知」の仕組みを理解している必要があります 。
- **サーバ証明書の検証**: アプリがサーバに接続する際、そのサーバが本物であることを「サーバ証明書」で検証するプロセスが重要です。この検証を怠ると、中間者攻撃（Man-in-the-Middle Attack）に対して脆弱になります 。
- **中間者攻撃**（Man-in-the-Middle Attack）: 攻撃者が通信の途中に割り込み、通信内容を盗聴・改ざんする攻撃です。プロキシツールを使った通信解析が、この攻撃と非常によく似た状況を作り出します 。

API・Webサービスの認証・認可

- **APIキー/アクセストークンの扱い**: アプリがサービスを利用する際、認証情報（APIキーやトークン）をどのようにサーバに送信し、サーバがそれをどう検証するかという基本的な流れの知識が必要です 。
- **署名付きURL** (Signed URL): 脆弱性2の対策として登場する重要な概念です 。永続的で強力な権限を持つ「アクセスキー」をアプリに直接埋め込むのではなく、「特定の操作（例: ファイル1つのアップロード）を、限られた時間だけ許可する」ための一時的なURLを発行する方式です。これにより、万が一URLが漏洩しても被害を限定できます。車のキーに例えるなら、エンジン始動とドア開閉だけができる「バレーキー」のようなものです。

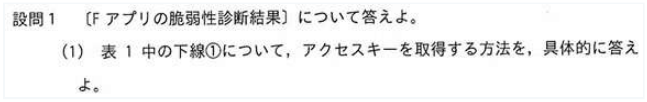
WebViewとクライアントサイドのセキュリティ

- **WebViewの基本とリスク**: WebViewはアプリ内にWebページを表示する便利な機能ですが、アプリ本体とWebコンテンツが連携する部分（JavaScriptブリッジ）に脆弱性が生まれやすいことを知っている必要があります 。
- **カスタムURLスキーム**: f-app:// のような独自のURLスキームを使ってアプリを起動したり、情報を渡したりする機能です 。ここで受け取るURLなどのパラメータを適切に検証しないと、意図しないWebサイトに接続させられる（オープンリダイレクト）などの攻撃につながります 。

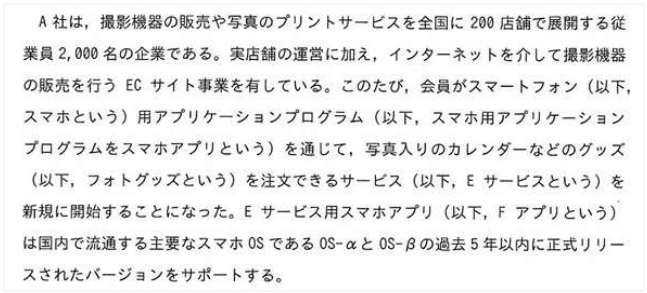
モバイルアプリの基本的なセキュリティ

- **リバースエンジニアリング**: アプリのプログラムファイル（apkやipa）を解析して、ソースコードやロジック、埋め込まれた秘密情報などを盗み出す行為です。「アクセスキーをアプリ内に保存する」 という設計がなぜ危険なのかを理解するために、この知識が背景として必要です。

設問１（１）


設問１　〔F アプリの脆弱性診断結果〕について答えよ。
(1)　表 1 中の下線①について，アクセスキーを取得する方法を，具体的に答えよ。

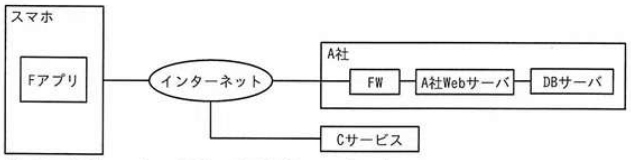
脆弱性1「サーバ証明書の検証不備」がある状況で、攻撃者がどのようにして「アクセスキー」を取得するのか、その具体的な方法を答える問題です 。


A 社は，撮影機器の販売や写真のプリントサービスを全国に 200 店舗で展開する従業員 2,000 名の企業である。実店舗の運営に加え，インターネットを介して撮影機器の販売を行う EC サイト事業を有している。このたび，会員がスマートフォン（以下，スマホという）用アプリケーションプログラム（以下，スマホ用アプリケーションプログラムをスマホアプリという）を通じて，写真入りのカレンダーなどのグッズ（以下，フォトグッズという）を注文できるサービス（以下，E サービスという）を新規に開始することになった。E サービス用スマホアプリ（以下，F アプリという）は国内で流通する主要なスマホ OS である OS-α と OS-β の過去 5 年以内に正式リリースされたバージョンをサポートする。

導入部

〔E サービスの説明〕

E サービスは，F アプリとサーバサイドのシステム群で構成される。F アプリは，インターネットを介して E サービス用 Web サーバ（以下，A 社 Web サーバといい，FQDN は www.a-sha.co.jp とする）及び大手クラウドサービスプロバイダ C 社のクラウドストレージサービス（以下，C サービスという）との間で HTTPS を使用して通信する。フォトグッズの作成に使う写真は，F アプリから C サービスにアップロードする。E サービスのネットワーク構成を図 1 に，機能概要を図 2 に，F アプリの画面構成を図 3 に，フォトグッズの注文処理の流れを図 4 に，C サービスの仕様を図 5 に示す。



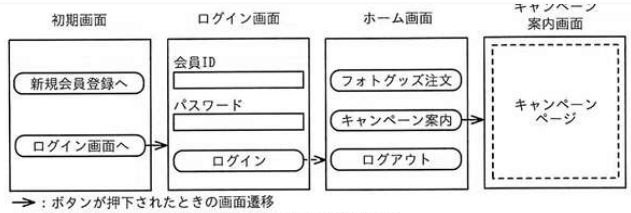
FW：ファイアウォール　　DBサーバ：データベースサーバ

図 1　E サービスのネットワーク構成（概要）

1. 新規会員登録機能
   E サービスを利用するための新規会員登録を行う。
2. ログイン機能
   会員 ID とパスワードでログインする。ログインした会員には，認証トークン [1] が払い出され，ログアウトするまでの間，F アプリに保存される。認証トークンは，A 社 Web サーバ上で会員のセッションを識別するために使用する推測困難な値である。
3. フォトグッズ注文機能
   F アプリ上でフォトグッズを注文する。ログイン済み会員だけが利用できる。
   なお，フォトグッズは，指定した A 社の実店舗で受け取ることができる。
4. キャンペーン案内機能
   キャンペーンの Web ページ（以下，キャンペーンページという）を表示する。ログイン済み会員だけが利用できる。
   なお，キャンペーンに応募することによって，フォトグッズの割引などに利用可能なクーポンを入手できる。会員には，電子メール（以下，メールという）などを通じて，期間限定のキャンペーンを案内する。キャンペーンの内容は，2 週間ごとに更新される。

注 [1]　認証トークンは，ログイン後に F アプリが A 社 Web サーバに HTTP リクエストを送信する際，Authorization ヘッダーに指定される。

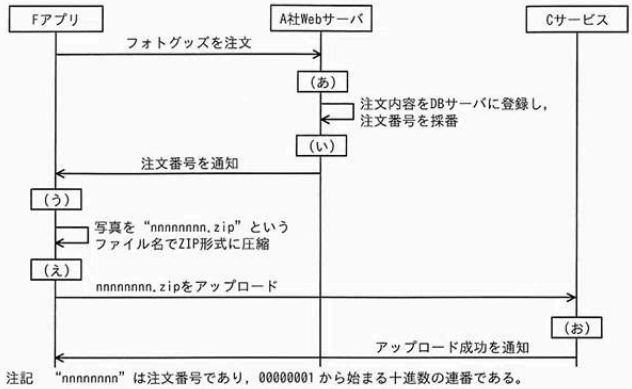
図 2　E サービスの機能概要（抜粋）



→：ボタンが押下されたときの画面遷移
-->：適切な値を入力してボタンが押下されたときの画面遷移

注記　キャンペーンページは HTML 形式で作成し，A 社 Web サーバにアップロードしておく。

図 3　F アプリの画面構成（抜粋）



注記　“nnnnnnnn”は注文番号であり，00000001 から始まる十進数の連番である。

図 4　フォトグッズの注文処理の流れ

1. サービスの概要
(1) Cサービスはマルチテナントのストレージサービスである。テナントの管理権限をもつ利用者（以下，管理者という）が作成したストレージに対し，テナントの作成したシステム（以下，利用システムという）からファイルのアップロードやダウンロード（以下，アップロード，ダウンロードを併せてファイル操作という）を行う。
(2) 管理者は，ストレージの作成時に任意のストレージ名を設定する。ストレージを作成すると，Cサービスからアクセスキーが発行される。アクセスキーはストレージごとに異なる40字の英数字である。
(3) 利用システムは，ストレージ上のファイルを URL のパスで指定する。例えば，ストレージ“●●●”上のファイル“▲▲▲”をファイル操作する際は，“/●●●/▲▲▲”を指定する。ファイルのアップロードには HTTP の PUT メソッドを，ファイルのダウンロードには GET メソッドを用いる。
(4) 利用システムは，次の方式 a 又は方式 b でファイル操作を行う。
2. 方式 a
(1) 説明
アクセスキーを HTTP リクエストの Authorization ヘッダーに指定する方式である。利用システムは，Cサービスから発行されたアクセスキーを用いることによって，アクセスキーに対応するストレージに格納された全てのファイルに対するファイル操作が可能となる。
Authorization ヘッダーに正しいアクセスキーが指定されていない場合，ファイル操作は拒否される。

図5　Cサービスの仕様（抜粋）

(2) 使用例
アクセスキー“○○○”を指定して，ストレージ“abc”上のファイル“xyz”をダウンロードする際に送信する HTTP リクエストの例を次に示す。
リクエストライン　：GET /abc/xyz HTTP/1.1
ヘッダーフィールド：Host: storage.c-sha.jp
Authorization: Bearer ○○○
3. 方式 b
(1) 説明
有効期限（Expires）と署名値（Signature）をクエリパラメータとして付加した URL（以下，署名付き URL という）を用いて，特定のファイルに対するファイル操作を一時的に可能とする方式である。ここで，Expires パラメータに指定する有効期限は UNIX タイムスタンプ形式である。署名値の生成は次のように行う。
（i）GET 又は PUT から始まり，パス中の“/●●●/▲▲▲?Expires=【有効期限】”で終わる文字列を署名対象文字列とする。
（ii）アクセスキーを秘密鍵とする。
（iii）署名対象文字列と秘密鍵から HMAC-SHA256 値を求める。
（iv）（iii）で求めた値を base64url エンコードする。
利用システムは，署名付き URL を生成し，ファイル操作を許可する利用者に伝える。伝えられた利用者は，署名付き URL を指定して HTTP リクエストを送ることによって，ファイル操作ができる。有効期限が切れた場合やサーバ側で署名値の検証が失敗した場合は，ファイル操作が拒否される。
(2) 使用例
ストレージ“abc”上のファイル“xyz”をダウンロードする場合で，かつ，署名値が“△△△”の場合の HTTP リクエストの例を次に示す。xyz は，日本時間の 2025 年 5 月 30 日 0 時 0 分 0 秒，つまり UNIX タイムスタンプで 1748530800 までの間，ダウンロードが許可されている。
リクエストライン　：GET /abc/xyz?Expires=1748530800&Signature=△△△ HTTP/1.1
ヘッダーフィールド：Host: storage.c-sha.jp

図5　Cサービスの仕様（抜粋）（続き）

Eサービスで使うCサービスのストレージ名は，e-service である。Fアプリでは，方式 a を利用する。アクセスキーは，鍵長 256 ビットの共通鍵と AES-CBC アルゴリズムで暗号化し，F アプリ内にリソースとして保存する。C サービスのストレージ名並びに AES-CBC の共通鍵及び初期ベクトルは，F アプリのコード中に定数として定義する。

〔キャンペーン案内機能の実装方法〕

F アプリでのキャンペーンページの表示には，WebView という仕組みを用いる。WebView は，スマホ OS の提供する仕組みであり，スマホアプリの画面の一部に Web ページを表示させることができる。キャンペーンページの HTML は，WebView を用いて，F アプリの画面上に表示させる。

会員に送るキャンペーン案内のメール本文中には，F アプリでキャンペーンページを表示するための URL（以下，F-URL という）を含める。会員がメールアプリから F-URL を開くと，F アプリが起動し，WebView 上にキャンペーンページが表示される。キャンペーンページからキャンペーンに応募する際の会員のセッションの識別には，F アプリに保存されている認証トークンを用いる。OS-αでは，キャンペーンページが表示された後に，WebView の機能によってキャンペーンページ上の ECMAScript コードが F アプリの getToken 関数を呼び出す。これによって，認証トークンが F アプリからキャンペーンページに引き渡される。OS-βでは別の方式で同様の機能を実現する。キャンペーンページ上の ECMAScript コードを図 6 に示す。

```
const token = f_app.getToken();
```

図6　キャンペーンページ上の ECMAScript コード

キャンペーンページ以外から getToken 関数を悪用されないように，getToken 関数の内部では，図 7 のように呼出し元の Web ページの URL を確認する。

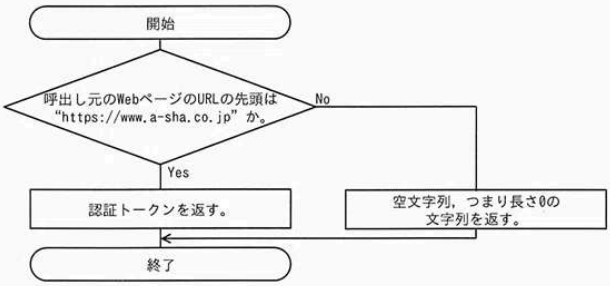


図 7　getToken 関数の処理の流れ

F-URL の例を図 8 に示す。

```
f-app://campaign?url=https://www.a-sha.co.jp/campaign/□□□
```

注記 1　“f-app”はカスタム URL スキームである。
注記 2　“□□□”はキャンペーンページの URL のパスである。

図 8　F-URL の例

〔F アプリの脆弱性診断結果〕

A 社は，セキュリティ専門会社の D 社に依頼して F アプリの脆弱性診断を実施した。その結果，表 1 に示す脆弱性が検出された。

表 1　脆弱性診断結果（抜粋）

脆弱性	脆弱性の概要	解説
1	サーバ証明書の検証不備がある。	F アプリは，HTTPS でサーバと接続する際，サーバ証明書の検証エラーがあっても無視し，通信を続行する。そのため，HTTPS 通信の内容が盗聴されたり，改ざんされたりするおそれがある。盗聴されると，①盗聴した内容からアクセスキーとストレージ名を攻撃者が取得するおそれがある。 （省略）
2	C サービスのアクセスキーの保護に不備がある。	②攻撃者が F アプリから平文のアクセスキーとストレージ名を取得できる。そのアクセスキーを用いて，③攻撃者が E サービスの全利用者の写真を不正にダウンロードするおそれがある。 （省略）
3	F-URL の処理にアクセス制御の不備がある。	F-URL の url クエリパラメータに，④細工した URL が指定されることによって，攻撃者の Web サイトにアクセスしてしまうおそれがある。また，攻撃者が会員の認証トークンを取得するおそれがある。 （省略）

ステップ1：脆弱性の核心を理解する

まず、脆弱性1の内容を正確に把握します。

脆弱性:「Fアプリは、HTTPSでサーバと接続する際、サーバ証明書の検証エラーがあっても無視し、通信を続行する。」

意味: アプリが通信相手をきちんと確認しない、ということです。
本来、HTTPS通信ではアプリはサーバから提示された「サーバ証明書」を見て、「自分が通信しようとしている本物のサーバか？」を検証します。このアプリはその検証をサボってしまいます。

ステップ2：可能な攻撃手法を特定する

この「通信相手を確認しない」という脆弱性を悪用できる、典型的な攻撃手法を考えます。

攻撃手法: 中間者攻撃（Man-in-the-Middle Attack）

- **攻撃の概要:** 攻撃者が、利用者のスマホと正規のサーバとの間に割り込み、通信を中継することで、通信内容を盗聴・改ざんする攻撃です。

ステップ3：具体的な攻撃手順を組み立てる

中間者攻撃を成功させ、アクセスキーを盗むまでの具体的な手順を組み立てます。

1. **準備（割り込み）:** 攻撃者は、カフェや空港などの**公衆無線LAN**になりすました悪意のあるWi-Fiアクセスポイントを設置し、利用者のスマートフォンを接続させます。これにより、アプリの通信経路上に割り込むことができます。
2. **偽装（なりすまし）:** アプリがCサービス（クラウドストレージ）に接続しようとすると、その通信を攻撃者が横取りします。攻撃者は、Cサービスの代わりに**自身の偽のサーバ証明書**をアプリに提示します。
3. **脆弱性の悪用:** アプリは脆弱性を持っているため、提示された証明書が偽物であることに気づかず、**攻撃者を本物のサーバだと信じ込んでHTTPS通信を確立**してしまいます。
4. **盗聴（キーの取得）:** これで、アプリと攻撃者の間の通信は、攻撃者によって完全に復号（解読）できる状態になります。利用者がフォトグッズを注文し、写真をアップロードする際、アプリはCサービスへのリクエストの**Authorizationヘッダーにアクセスキーを含めて送信**します 。攻撃者はこの通信を復号して、ヘッダーからアクセスキーを平文で読み取ります。

ステップ4：解答をまとめる

以上の手順を、解答として具体的にまとめます。

解答例: 公衆無線LANなどで中間者攻撃を仕掛け、偽のサーバ証明書をアプリに提示してHTTPS通信を復号する。利用者が写真をアップロードする際の通信を盗聴し、Authorizationヘッダーからアクセスキーを取得する。

IPA解答例

(1)	C サービスに送られた HTTP リクエストの Authorization ヘッダーから取り出す。

IPAの解答は、最終的にどこからアクセスキーを抜き取るかという点に絞ったもので、非常に簡潔に書かれており核心を突いた解答ですね。