情報処理安全確保支援士への道(11)：令和７年 春 午後問題 問3を解いてみる(設問2(1)~(3)編)

ルチルMike
2025年8月11日 10:24

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

ステップ2：アプリの通信先を確認する

ステップ3：Subject Alternative Name (SAN) の役割を知る

ステップ4：ツールが設定すべき値を導き出す

解答

すべて表示

前提となる知識

HTTPS通信の裏側で何が起きているかについての、具体的で実践的な知識が求められます。特に、開発現場で通信内容をデバッグする際に行われる、プロキシツールを使った通信解析の仕組みが深く関わってきます。

TLS通信とサーバ証明書の深い理解

設問の中心にあるのはTLS（HTTPSの根幹をなすプロトコル）です。

- **サーバ証明書の役割:** なぜ証明書があると安全なのか、その証明書が「誰によって発行されたか（認証局：CA）」、「誰のための証明書か（コモンネーム/SANs）」といった内容を理解している必要があります。特に、現代のブラウザやアプリがサーバの身元確認に使う**SANs (Subject Alternative Name)** という項目を知っていることが、設問2(1)を解く鍵となります。
- **信頼の連鎖 (Trust Chain):** スマートフォンなどのデバイスが、なぜ特定のサーバ証明書を「信頼」できるのかを知っている必要があります。デバイス内には「信頼できる認証局（CA）のリスト（トラストストア）」が保存されており、このCAが発行した証明書だから信頼する、という仕組みです。

TLSハンドシェイクの具体的な流れ

設問2(2)では、TLS通信が確立されるまでの一連の手順（ハンドシェイク）に関する知識が直接問われます。

- **メッセージの順序:** 通信開始時にクライアントとサーバが交換するメッセージの種類と順序を知っている必要があります。
  - Client Hello (クライアントからの挨拶)
  - Server Hello (サーバからの返事)

- Certificate (サーバからの証明書提示)
- Certificate Verify (サーバが証明書の持ち主であることの証明)
- Finished (ハンドシェイク内容の確認)

中間者攻撃とプロキシツールの動作原理

　この設問のシナリオは、プロキシツールを使った通信解析ですが、これはセキュリティの世界でいう「**中間者攻撃 (Man-in-the-Middle Attack)**」と全く同じ原理で動作します。

- **通信の割り込みと復号:** プロキシツールは、アプリとサーバの間に割り込み、暗号化されたHTTPS通信を一度解読（復号）してから、再度暗号化してサーバに送ります。
- **偽の証明書の発行:** この割り込みを成功させるため、プロキシツールはその場で**偽のサーバ証明書を動的に生成**してアプリに提示します。
- **ルートCA証明書のインストール:** 通常、アプリはこの偽の証明書を信頼せずにエラーを出しますが、開発時にこれを回避するためには、**プロキシツール自身の認証局（ルートCA）証明書を開発用スマホにインストール**します。これにより、スマホはプロキシが発行する偽の証明書を「信頼できる」と判断するようになります。この知識が設問2(3)の答えに直結します。

これらの知識は、単なる暗記ではなく、アプリケーション開発やセキュリティ診断の実務で頻繁に利用される、非常に実践的な内容です。

設問２（１）

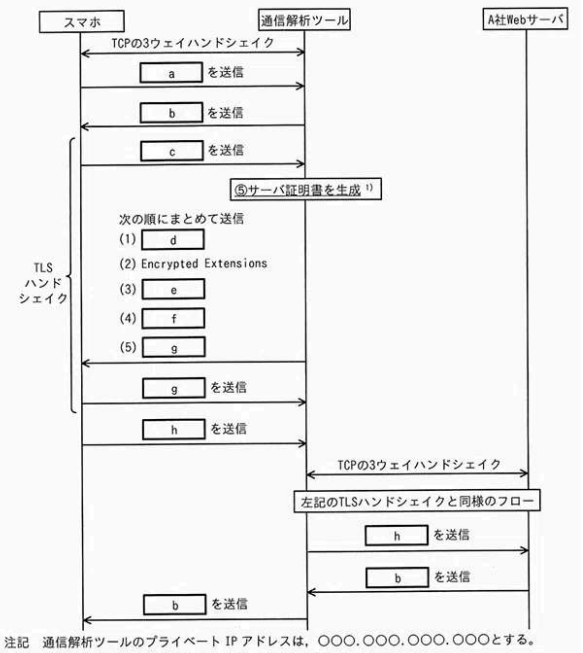
〔脆弱性１〕

Ｆアプリの開発チームに所属するＵさんは，Ｄ社のＳさんが開催する診断結果報告会に参加した。

Ｕさんは，脆弱性１が作り込まれた経緯を説明した。Ｕさんによると，ＦアプリとＡ社 Web サーバとの間の通信内容に異常がないかどうかを調査するために，開発用 PC で通信解析ツールを利用した。この通信解析ツールはプロキシサーバとして動作する。このツールを利用すると，Ｆ アプリでは，サーバ証明書の検証エラーが発生し，ＦアプリとＡ社 Web サーバとの間の通信が中断されてしまった。そこで，インターネット上のある記事でエラーが発生しても通信を続行する方法が紹介されていたのを

参考にして，F アプリのコードを変更したということであった。

この通信解析ツールを利用し，“https://www.a-sha.co.jp/campaign/□□□”にアクセスした際のレイヤー4～7 の通信フローの例を図 9 に示す。



注記　通信解析ツールのプライベート IP アドレスは，○○○.○○○.○○○.○○○とする。
注 1)　通信解析ツールは，自身のプライベート認証局機能を用いる。

図 9　通信解析ツールを利用した際の通信フローの例（抜粋）

(1)　図 9 中の下線⑤について，サーバ証明書の Subject Alternative Name の値を，具体的に答えよ。

この設問は、HTTPS通信を解析するツール（中間者）が、どのようにしてサーバになりすますのかを理解しているかを問う問題です。

ステップ1：通信解析ツールの目的を理解する

　まず、図9に登場する「通信解析ツール」の役割を把握します。このツールは、スマートフォンアプリとA社Webサーバの間に割り込み、暗号化されたHTTPS通信を解読（復号）するために使われます。これを行うには、ツールがA社Webサーバになりすまして、アプリと直接TLS通信を確立する必要があります。

ステップ2：アプリの通信先を確認する

　次に、アプリが本来通信しようとしている相手を確認します。問題文には、アプリがアクセスするURLが「https://www.a-sha.co.jp/campaign/□□□」であると記載されています 。つまり、アプリは www.a-sha.co.jp というホスト名のサーバに接続しようとしています。

ステップ3：Subject Alternative Name (SAN) の役割を知る

　サーバ証明書に含まれる　Subject Alternative Name (SAN)　は、その証明書がどのホスト名（ドメイン名）に対して有効であるかを示す、最も重要な項目です。アプリやブラウザは、サーバから受け取った証明書のSANに、自分がアクセスしようとしているホスト名（今回はwww.a-sha.co.jp）が含まれているかを確認することで、通信相手が本物か（なりすましでないか）を検証します。

ステップ4：ツールが設定すべき値を導き出す

　ステップ1～3を総合すると、通信解析ツールがA社Webサーバになりすますためには、**アプリがアクセスしようとしているホスト名と全く同じ値をSANに設定した、偽のサーバ証明書をその場で生成してアプリに提示する**必要があります。

　そうでなければ、アプリは「アクセスしようとしているホスト名と証明書の内容が違う」と判断し、通信を即座に中断してしまいます。

　したがって、サーバ証明書のSANに設定されるべき値は、アプリの接続先ホスト名そのものです。

解答

www.a-sha.co.jp

IPA解答例:設問２（１）

(1)	www.a-sha.co.jp

設問２（２）

(2) 図 9 中の [a] ～ [h] に入れる適切な字句を，解答群の中から選び，記号で答えよ。

解答群

ア Certificate

イ Certificate Verify

ウ Client Hello

エ CONNECT ○○○.○○○.○○○.○○○:443 HTTP/1.1

オ CONNECT www.a-sha.co.jp:443 HTTP/1.1

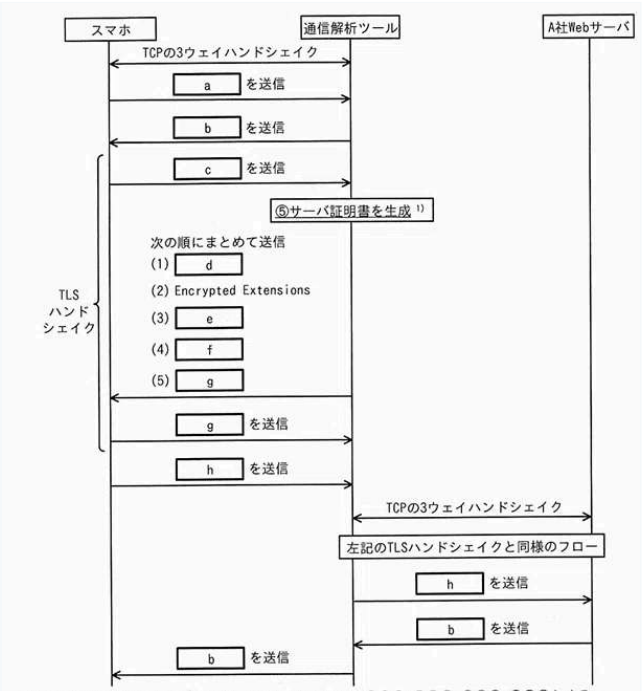
カ Finished

キ GET /campaign/□□□ HTTP/1.1

ク HTTP ステータスコード 101 (Switching Protocols)

ケ HTTP ステータスコード 200 (OK)

コ Server Hello



注記 通信解析ツールのプライベート IP アドレスは，○○○.○○○.○○○.○○○とする。

注 1) 通信解析ツールは，自身のプライベート認証局機能を用いる。

図 9 通信解析ツールを利用した際の通信フローの例（抜粋）

この設問は、図9に示された一連の通信フローの各段階（a～h）が、どのプロトコルメッセージに該当するかを特定する問題です。

このフローは、スマートフォンが「通信解析ツール（プロキシ）」を経由してHTTPSサイトにアクセスする際の典型的な流れを示しています。これを解くには、**HTTPSプロキシの動作（CONNECTメソッド）と、その後のTLSハンドシェイク**という2段階のプロセスを理解している必要があります。

ステップ1：プロキシへの接続確立フェーズ（a, b）

まず、アプリは目的のサーバと通信する前に、経由するプロキシとの間で「トンネル」を確立する必要があります。

- **a (スマホ → ツール):** アプリがプロキシに対し、「www.a-sha.co.jpの443番ポートに接続したいので、トンネルを用意してください」と要求します。この要求に使うのがHTTPの**CONNECTメソッド**です。
  - したがって、**a** は **オ (CONNECT www.a-sha.co.jp:443 HTTP/1.1)** となります。（選択肢工は接続先がIPアドレスなので不適切です）
- **b (ツール → スマホ):** プロキシは、トンネルの準備ができたことをアプリに伝えます。この応答は「成功」を意味する **HTTPステータスコード 200 (OK)** です。
  - したがって、**b** は **ケ (HTTP ステータスコード 200 (OK))** となります。（図の下部でも同じ応答がラベルbで使われていますが、ここではプロキシの応答と解釈します）

ステップ2：TLSハンドシェイクフェーズ（c, d, e, f, g）

トンネルが確立されると、いよいよそのトンネル内で暗号化通信のためのTLSハンドシェイクが始まります。

- **c (スマホ → ツール):** TLSハンドシェイクの最初のメッセージは、クライアント（スマホ）から送られる「こんにちは、暗号化通信を始めましょう」という挨拶、**Client Hello**です。
  - したがって、**c** は **ウ (Client Hello)** となります。（※図の矢印の向きは不正確ですが、文脈上これが最も論理的です）
- **d, e, f, g (ツール → スマホのブロック):** Client Helloに対し、サーバ側（ツール）が応答します。このブロックはTLSのサーバ側メッセージ群です。
  - **d:** 最初の応答は **Server Hello** です。→ **d** は **コ**
  - **e:** 次にサーバ証明書を送ります。→ **e** は **ア (Certificate)**
  - **f:** 証明書の持ち主であることを証明します。→ **f** は **イ (Certificate Verify)**
  - **g:** サーバ側のハンドシェイク完了通知です。→ **g** は **カ (Finished)**

ステップ3：アプリケーションデータ通信フェーズ（h）

ハンドシェイクが完了し、暗号化された通信路が確立されると、その上で実際のHTTPリクエストが送られます。

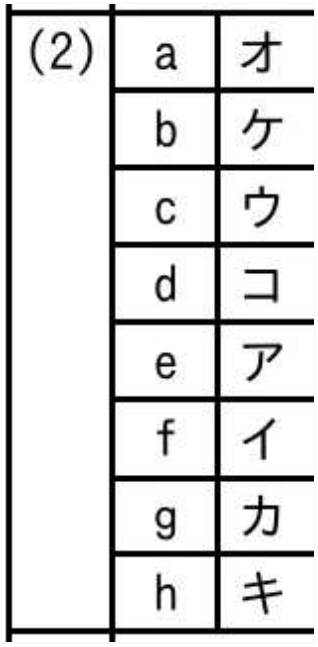
- **h (スマホ → ツール):** アプリが、目的のキャンペーンページを取得するための **GETリクエスト**を送信します。
  - したがって、**h** は **キ (GET /campaign/ HTTP/1.1)** となります。

解答のまとめ

以上のステップをまとめると、各記号に対応する解答は以下の通りです。

- **a: オ**
- **b: ケ**
- **c: ウ**
- **d: コ**
- **e: ア**
- **f: イ**
- **g: カ**
- **h: キ**

IPA解答例:設問２（２）



(2)	a	オ
	b	ケ
	c	ウ
	d	コ
	e	ア
	f	イ
	g	カ
	h	キ

設問２（３）

この設問は、開発時にHTTPS通信を安全にデバッグするための、標準的な設定方法を理解しているかを問う問題です

U さんは，通信解析ツールを利用してテストを行う際も通信を正常に続行させる方

法をチーム内で話し合った。その結果，今後，開発用のスマホに⑥必要な設定を行うことにした。加えて，OS-αではテストを行う際だけその設定を有効化するように，F アプリの中にも設定を追加した。

(3)　本文中の下線⑥について，設定の内容を，具体的に答えよ。

ステップ1：解決すべき課題を理解する

まず、開発チームが直面している課題を整理します。

- **やりたいこと:** 通信解析ツール（プロキシ）を使って、アプリのHTTPS通信の内容を確認したい。
- **問題点:** プロキシを使うと、アプリ側で「サーバ証明書のエラー」が発生して通信ができない。
- **誤った対策:** エラーを回避するため、アプリに証明書検証を無視させるコードを実装してしまった（これが脆弱性1の原因）。
- **目指すゴール:** アプリのコードは正規のまま（証明書検証を行う）で、かつ開発時にはプロキシを使った通信解析ができるようにしたい。

ステップ2：証明書エラーが起きる根本原因を考える

なぜプロキシを使うと証明書エラーが起きるのか、その仕組みを理解します。

1. プロキシは、A社Webサーバの代わりになりすまして、独自のサーバ証明書をその場で発行します。
2. この証明書は、プロキシ自身が作った「俺が認証局だ」という　自己署名の認証局（ルートCA）によって署名されています。
3. スマートフォンは、このプロキシの「自称・認証局」を全く知らないため、「この証明書は信頼できない、怪しい発行元だ」と判断し、エラーを出します。これがエラーの正体です。

ステップ3：正しい解決策を導き出す

課題は「スマホがプロキシの発行元を信頼していないこと」です。したがって、解決策は「開発用のスマホにだけ、特別にプロキシの発行元を信頼させる」ことになります。

これを実現する具体的な設定が、**通信解析ツールが使うルートCA証明書を、開発用スマートフォンのトラストストア（信頼された証明書の一覧）にインストールする**ことです。

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: 通信解析ツールが使用するルートCA証明書を、開発用のスマートフォンの信頼された証明書としてインストールする設定。

これにより、開発用スマホはプロキシが発行した証明書を正当なものとして扱うため、アプリは証明書検証を有効にしたまま、問題なく通信内容の解析ができるようになります。

IPA解答例:設問２（３）

(3)	通信解析ツールのプライベート認証局のルート証明書をインストールし，信頼設定を行う。