令和 4 年度 春期 情報処理安全確保支援士

＜午後Ⅱ解答・解説＞

＜問 1＞ Web サイトのセキュリティ

●設問 1

［試験センターによる解答例］

(1) ア

(2)・ダウンロードするライブラリに既知の脆弱性がないかを確認する。(30 字)
・特定の Web サイトからの入手をルール化し、明文化する。(27 字)

＜解説＞

(1) リクエストラインが「GET /confirm?」で始まり、クエリ文字列がダブルクオート["]で囲ってあるアの記述が正しい。

(2) B 社では安全性が確認できているライブラリを公開している Web サイトからファイルサーバにダウンロードして利用しているとあるが、今回既知の XSS 脆弱性の対策をしていないライブラリ M を使っていたということは、それ以外の Web サイトからダウンロードされた可能性がある。再発防止策としては、ダウンロードするライブラリに既知の脆弱性がないかを確認すること、もしくは、安全であることが確認できている特定の Web サイトからの入手をルールとして明文化することが考えられる。

●設問 2

［試験センターによる解答例］

a：利用者 ID
b：セッションオブジェクト
※a と b は順不同

＜解説＞

利用者 A の利用者 ID でログインした時の csrftoken の値が、利用者 B の利用者 ID でログインした時にも有効で、利用者 B として処理されたということは、csrftoken と利用者 ID、もしくはセッション ID などのセッションオブジェクトとのひも付けが行われていないと考えられる。

●設問３

［試験センターによる解答例］

(1) c：イ
d：ア
e：ア
f：ア
g：イ
h：イ

(2) i：エ
j：イ

＜解説＞

(1) クリックジャッキングは、利用者を騙すために用意した画面βの手前に、利用者にクリックさせたい画面αを透明な状態で重ねて表示することで、利用者には画面βをクリックしていると思わせながら実際には画面αでクリックイベントを発生させる攻撃手法である。したがって、c にはイの「β」、d にはアの「奥」、e にはアの「可視の」、f にはアの「α」、g にはイの「手前」、h にはイの「透明な」が入る。

(2) クリックジャッキングへの対策として、レスポンスヘッダに"X-Frame-Options"を含める方法と、"Content-Security-Policy"を含める方法があり、後者は W3C（World Wide Web Consortium）で標準化が行われている。
"X-Frame-Options"は frame 又は iframe でページを表示することの可否を指定する仕組みであり、「DENY」を設定すると frame 又は iframe でページを表示することを一切禁止する。一方「SAMEORIGIN」を指定すると、アドレスバーに表示されたドメインと同じドメインの場合のみ許可する。
"Content-Security-Policy"は、クリックジャッキング対策だけでなく、クロスサイトスクリプティング対策等にも用いられているもので、"X-Frame-Options"と同様に、frame 又は iframe でページを表示することを許可するドメインを限定することが可能である。

●設問４

［試験センターによる解答例］
topic の値を https://db-y.b-sha.co.jp/に変更した。(39 字)

＜解説＞
図 4 にあるように、通常のリクエストでは、topic の値に A 社のニューストピックの URL を指定することで、コンテンツを取得し、表示する仕組みになっている。この仕組みを使ってサイト Y の DB サーバのメンテナンス用の Web インタフェースにアクセスできたとすれば、topic の値を、表 2 の注記にあるサイト Y の DB サーバの URL である https://db-y.b-sha.co.jp/ に変更したと考えられる。

●設問５

［試験センターによる解答例］
(1) k：V 氏が用意したサイト (10 字)

(2) returnURL の値を固定値にする。(19 字)

＜解説＞
(1) 図 5 の注記にあるように、サイト Z の P 社宿泊との連携機能では、番号(1)で Host ヘッダにセットされた値が(2)の returnURL 中のホスト名となる。表 4 の SSRF 脆弱性を検出した手順では、順序 1 で Host ヘッダの値を V 氏が用意したサイトの FQDN に変更し、サイト Z にリクエストを送っている。このとき、登録されていない駅名のリクエストであれば、順序 3 では Location ヘッダに V 氏が用意したサイトの URL を含めたレスポンスがサイト Z に返され、順序 4 では V 氏が用意したサイトにリクエストが送られる。したがって、［ k ］には「V 氏が用意したサイト」が入る。

(2) D 社から提案された対策も有効であるが、より確実な対策としては、returnURL の値を指定させる仕様を見直し、固定値とすることである。

●設問６

［試験センターによる解答例］

(1)・一部のセッション管理の脆弱性（14 字）
　・認可・アクセス制御の脆弱性（13 字）

(2) 改良フェーズにおける 1 か月の休止期間（18 字）

(3)・専門技術者による脆弱性診断が必要なときは、改良リリースを次回に持ち越す。(36 字)
　・半年に一度、改良リリースの期間を長くする。(21 字)
　・定期的に、期間の長い改良リリースを設ける。(21 字)

(4) CSRF 対策用トークンの発行、HTML への埋め込み、必要なひも付け、及びこれを検証する処理（45 字）

＜解説＞

(1) 表 1 の Web セキュリティ管理基準（抜粋）で、項番 2、3、4 では対象外となるが、項番 5 では対象となる脆弱性として挙げられているのは、「セッション管理の脆弱性」と「認可・アクセス制御の脆弱性」である。ただし、セッション管理の脆弱性の一部は項番 2、3、4 でも対象となっているため、項番 5 で対象となるのはそれ以外のセッション管理の脆弱性という意味として、「一部のセッション管理の脆弱性」となる。

(2) 表 1 より、専門技術者による脆弱性診断の実施期間は 10 日間くらいが目安であることがわかる。図 7 の注記にあるように、改良フェーズには、半年に 1 回、1 か月の休止期間が設けられているので、この期間中に実施すれば、専門技術者による脆弱性診断が長期間行われないことを避けることができる。

(3) 開発プロセスで見直すべき点としては、2 週間周期で行われている改良リリースを、脆弱性診断の実施を考慮して柔軟に調整することである。具体的には、専門技術者による脆弱性診断が必要な場合には、改良リリースを見送り、次回に持ち越すことが考えられる。または、現状でも半年に 1 回休止期間を設けているように、半年に 1 回改良リリースの期間を長めにするなど、定期的に期間の長い改良リリースを設けること等が考えられる。

(4) 問題文の〔サイト X の CSRF 脆弱性〕を参考にして必要な機能を挙げるとよい。まず、CSRF 対策用トークン（以下「トークン」）を発行し、それを HTML に埋