平成 27 年度 春期 情報セキュリティスペシャリスト

＜午後 I　解答･解説＞

＜問1＞ Web サイトの脆弱性と対策

■設問 1

〔試験センターによる解答例〕

(1) 画面 B

(2) 暗号化されない HTTP 通信において，セッション ID が送信されるから（33 字）

secure 属性を設定すると，HTTPS（SSL/TLS）で通信している場合のみ Cookie を送出する。逆に secure 属性を設定していないと，暗号化されていない HTTP 通信においても Cookie が送信されるため，第三者にセッション ID を盗聴されるリスクがある。図 1 を見ると，試験用サイトの画面において HTTP で通信しているのは**画面 B** である。そのため，画面 B に遷移するときにセッション ID を盗聴されるリスクがある。

■設問 2

〔試験センターによる解答例〕

(1) a：%0d%0a%0d%0a

(2) b：ウ

(3) c：・出力文字列に改行コードがあるとエラー画面を出力（23 字）

・出力文字列の改行コード以降の文字列を削除（20 字）

(1) HTTP のメッセージヘッダとメッセージボディの境界は空行（二つの連続した改行コード）で識別する仕様となっている。HTTP ヘッダインジェクションとは，ユーザの入力データをもとに HTTP メッセージのレスポンスを生成する Web アプリケーションに対し，任意の場所に空行を挿入して不正なスクリプトを実行させたり，任意のヘッダフィールドを追加したりする攻撃手法である。

図 8 の ASCII 文字一覧より，図 7 では，

"<html><body><script>alert("1")</script></body><html>"

というスクリプトであることが分かる。HTTP ヘッダインジェクションの脆弱性を突いてこれを実行させるには，上記文字列の前に空行を挿入すればよい。空行は二つの連続した改行コード（CRLFCRLF）であるため，これを図 8 に従って URL エンコードすると**"%0d%0a%0d%0a"**となる。

(2) 攻撃者が指定した任意のスクリプトをクライアント側で実行する攻撃は，**クロスサイトスクリプティング**である。

(3) HTTP ヘッダインジェクションへの対策としては，問題文に示されているもののほか，HTTP レスポンスヘッダを生成する**出力文字列に改行コードが含まれていた場合にエラーメッセージを出力**したり，**改行コード以降の文字列を削除**したりするなどの処理を実装する方法がある。

■設問３

〔試験センターによる解答例〕

(1) d：09 又は 17

e：27

※d，e は順不同

(2) f：01234

(3) 攻撃者 J が取得したセッション ID で利用者 K にログインさせているから（33 字）

(4) g：新しいセッション ID によるセッションを開始する（23 字）

(1) セッションフィクセーションとは，Web アプリケーションにおけるセッションハイジャックの手法の一つであり，「セッション ID の固定化攻撃」とも呼ばれる。セッションフィクセーションは，既に確立されているセッションをハイジャックするわけではなく，ターゲットユーザに対して攻撃者が生成したセッション ID を含む不正な URL を送りつけることで意図的にセッションを確立させ，そのセッションをハイジャックするというものである。
会員制のサイトなどで，ログイン画面を表示した時点でセッション ID が発行され，ログイン後も同じセッション ID を使用する仕様になっているような場合には，攻撃者がセッション ID を容易に入手可能であるため，この攻撃が成立する可能性がある。
図 6 の HTTP ヘッダを見ると，行番号 09, 17, 27 の 3 か所にセッション ID があるが，いずれも同じ値になっている。注記にあるように，これらのセッション ID は下記で使用されていることが分かる。

①リクエスト X とレスポンス X は最初に画面を表示した際の HTTP ヘッダ
②リクエスト Y とレスポンス Y は画面 A から画面 B に遷移した際の HTTP ヘッダ
③リクエスト Z とレスポンス Z は画面 C から画面 D に遷移した際の HTTP ヘッダ

これらのうち，①と②が同じセッション ID であることは問題ないが，③はログイン後の商品取引画面であるため，このままではセッションフィクセーションが成立する可能性がある。したがって，[d]，[e] には，"**09** 又は **17**", "**27**"（順不同）が入る。

(2) Cookie の属性の一つに"domain"があり，その Cookie が有効となるドメイン名を「.」から始まる形式（例：.shoeisha.co.jp）で指定する。指定があった場合は，そのドメイン名が含まれていることが Cookie を送出する条件となり，サブドメイン名やホスト名が異なる場合であっても Cookie の共有が可能となるが，セキュリティ確保のため，「.co.jp」「.com」「.net」などの指定は無効となる。
Cookie Monster Bug とは，一部のブラウザでこの"domain"の指定が正しく機能しないというバグであり，これにより，セッションフィクセーションやクロスサイトリクエストフォージェリ（CSRF）といった攻撃を成立させやすくしてしまう可能性がある。
前述のように，セッションフィクセーションでは，攻撃者は自分が取得したセッション ID を含む不正な URL を利用者に送りつけることで意図的にセッションを確立させ，そのセッションをハイジャックするというものである。したがって，[f] には"**01234**"が入る。

(3) 攻撃者 J は，**自分が取得したセッション ID"01234"で利用者 K にログインさせている**ため，後から同じセッション ID を用いることで，利用者 K になりすまし，本来はアクセス権限がない画面にアクセスできるようになる。

(4) ログインの前後で同じセッション ID を使い続けていると，セッションフィクセーションが成立する可能性が高まる。そのため，ユーザがログインに成功した後に**新たなセッション ID を発行し，それを用いてセッションを開始する**ようにすることが有効な対策となる。