問1　Webサイトのセキュリティに関する次の記述を読んで，設問1～6に答えよ。

A社は，従業員1,500名の中堅システム開発会社である。A社では，セキュリティ品質の高いWebサイトを開発するために，表1に示すWebセキュリティ管理基準を定めて運用している。

表1　Webセキュリティ管理基準（抜粋）

項番	管理策	概要
1	セキュリティ要件レビュー	・概要設計，基本設計，詳細設計それぞれの設計レビューにおいて，Webサイトに関するセキュリティ要件をレビューする。
2	ツールによるソースコードレビュー	・Webサイトのリリースまでに実施する。 ・期間は，3日間くらいが目安である。 ・開発環境の特性などが原因で実施できない場合，項番3を行う。 ・ツールが検出した指摘事項について，開発担当者は，脆弱性かどうか，対策が必要かどうかを判断する。 ・セッション管理の脆弱性は，一部だけが対象である。 ・認可・アクセス制御の脆弱性は，対象外である。
3	プロジェクトメンバによるソースコードレビュー	・項番2が実施できない場合，Webサイトのリリースまでに実施する。 ・期間は，10日間くらいが目安である。 ・A社の指定した既知の脆弱なコードパターンを見つける。 ・レビューでの指摘事項について，開発担当者は，脆弱性かどうか，対策が必要かどうかを判断する。 ・セッション管理の脆弱性は，一部だけが対象である。 ・認可・アクセス制御の脆弱性は，対象外である。
4	ツールによる脆弱性診断	・Webサイトのリリースまでに実施する。 ・期間は，3日間くらいが目安である。 ・Webサイトをテスト環境で稼働させ，ツールでWebサイトに様々なHTTPリクエストを送り，その応答を評価する。 ・ツールが検出した指摘事項について，開発担当者は，脆弱性かどうか，対策が必要かどうかを判断する。 ・セッション管理の脆弱性は，一部だけが対象である。 ・認可・アクセス制御の脆弱性は，対象外である。
5	専門技術者による脆弱性診断	・Webサイトのリリースまでに実施する。 ・期間は，10日間くらいが目安である。 ・専門会社[1)]に委託する。 ・Webサイトをテスト環境で稼働させ，WebサイトにHTTPリクエストを送り，その応答を評価する。 ・診断による指摘事項について，専門技術者と開発担当者は，対策が必要かどうかを協議して判断する。 ・セッション管理の脆弱性は，対象である。 ・認可・アクセス制御の脆弱性は，対象である。

注[1)]　脆弱性診断サービスを提供しているD社に委託している。

〔A 社による B 社の子会社化〕

B 社は，従業員 200 名の新興の IT サービス会社であり，各種 SaaS を提供している。アジャイル開発の能力が高く，機能の追加や性能の改善を頻繁に行っている。B 社と A 社とは協業関係にあったが，両社の経営陣は，双方の強みを生かせると判断し，A 社による B 社の子会社化について合意した。

B 社のクラウド環境には，コーポレートサイト（以下，サイト B という），及び B 社が提供中の三つの SaaS それぞれの Web サイト（以下，サイト X，サイト Y，サイト Z という）がある。それらの概要を表 2 に示す。

表 2　B 社の Web サイトの概要

サイト名及び URL	概要	システム構成
サイト B https://www.b-sha.co.jp/	・B 社に関する情報を発信している。 ・コンテンツマネジメントシステム（CMS）を導入し，運用している。	IaaS 上の Web サーバで構成されている。
サイト X https://x.b-sha.co.jp/	・会社又は組織向けのコミュニケーションサービスであり，利用する会社間又は組織間で，情報共有やチャットが行える。 ・利用する会社又は組織は，B 社の提携企業の新商品のモニターになると，“キャンペーン応募”をサイト X で行える。	IaaS 上の Web サーバ及びデータベースサーバ [1]（以下，DB サーバという）で構成されている。
サイト Y https://y.b-sha.co.jp/	・個人向けのブログサイトであり，利用者が情報を発信できる。 ・“A 社のニューストピック”を表示できる。	
サイト Z https://z.b-sha.co.jp/	・ソフトウェア開発企業向けの Web サービスである。利用者はグループを作ることができ，そのグループ内で，スケジュール，タスク，ソースコードなどのプロジェクト情報を共有できる。 ・外部の Web サイトと連携して，経費精算，出張申請などの業務手続を行う機能を提供予定である。	

注 [1]　DB サーバには，B 社のシステム担当者と，それぞれのサイトの Web サーバとがアクセスできる。各 DB サーバには，レコードの更新や削除が簡単にできるメンテナンス用の Web インタフェースがある。その URL を次に示す。

・サイト X の DB サーバ：https://db-x.b-sha.co.jp/
・サイト Y の DB サーバ：https://db-y.b-sha.co.jp/
・サイト Z の DB サーバ：https://db-z.b-sha.co.jp/

子会社化に当たって，B 社の Web サイトのセキュリティ水準を確認することになり，A 社の品質管理部でセキュリティ技術を担当している R さんが対応することになった。

〔B社のWebサイトのセキュリティ水準の確認〕

Rさんは，サイトB，サイトX，サイトY及びサイトZに対する脆弱性診断をD社に依頼した。診断の結果，検出された脆弱性を表3に示す。

表3　検出された脆弱性（抜粋）

サイト名	脆弱性名
サイトB	・クロスサイトスクリプティング（以下，XSSという）脆弱性
サイトX	・XSS脆弱性 ・クロスサイトリクエストフォージェリ（以下，CSRFという）脆弱性 ・クリックジャッキング脆弱性
サイトY	・XSS脆弱性 ・サーバサイドリクエストフォージェリ（以下，SSRFという）脆弱性
サイトZ	・XSS脆弱性 ・SSRF脆弱性

〔サイトBのXSS脆弱性〕

D社は，サイトBに①診断用リクエストを送ることで，XSS脆弱性があることを確認した。このリクエストは，ライブラリMを使ってプログラムCが処理する。ライブラリMのコードを図1に示す。

```
   (省略)
1:  out.println("<meta property=\"og:url\" content=\"https://"+serverName+"/"
   +scriptName+"?"+queryString+"\">");
   (省略)
```

注記　serverNameには，リクエストのURLのホスト名が格納されている。scriptNameには，URLのパス名が格納されている。queryStringには，URLのクエリ文字列以降の値がURLデコードされて格納されている。

図1　ライブラリMのコード

ライブラリMは，SEOのためのライブラリである。

B社では，開発部のメンバそれぞれが，開発時に利用可能なライブラリを収集している。使用するライブラリは，マルウェアが含まれていない，既知の脆弱性が修正された，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，利用している。ファイルサーバは，開発部のメンバであればアクセス可能である。

今回使われていたライブラリ M は，既知の XSS 脆弱性の対策をしていないバージョンであった。その結果，ライブラリ M を使っているサイト B，サイト X，サイト Y 及びサイト Z において，同じ XSS 脆弱性が検出された。

これを受けて，B 社における②再発防止策について検討した。

〔サイト X の CSRF 脆弱性〕

サイト X は，セッション ID を JSESSIONID という cookie に格納している。D 社は，サイト X のキャンペーン応募ページで CSRF 脆弱性を検出した。

CSRF 脆弱性を確認した手順は，次のとおりであった。

(1) 診断用利用者（以下，利用者 A という）の利用者 ID でサイト X にログインし，キャンペーン応募ページで送信されるリクエストの内容をツールを使って確認した。リクエストの内容を図 2 に示す。

```
POST /campaign HTTP/1.1
Host: x.b-sha.co.jp
Cookie: JSESSIONID=KCRQ88ERH2G8MGT319E5OSMOAJFDIVEM

csrftoken=3f4aee446f680df6e0842d7179fcefd00fe5b232
```

注記 1 リクエストヘッダ部分は，設問に必要なものだけを記載している。
注記 2 JSESSIONID について，SameSite 属性は指定されていない。
注記 3 csrftoken の値は，サーバが発行する推測困難な値であり，ほかの利用者の利用時には別の値が発行される。
注記 4 リクエストを送るとトークンが破棄される可能性があるので，リクエストの内容は，ツールで確認しただけであり，実際にはサイト X に届いていない。

図 2　リクエストの内容

リクエストの内容を確認後，csrftoken を CSRF 対策用のパラメタと考え，リクエスト中の csrftoken の値を削除して送信した場合と 1 文字変更して送信した場合を試したところ，どちらもエラーになった。

(2) 利用者 A とは別の診断用利用者（以下，利用者 B という）の利用者 ID でサイト X にログインし，キャンペーン応募ページで送信されるリクエスト中の csrftoken の値に，図 2 の csrftoken の値を設定して送信したところ，利用者 B として処理された。

この結果から，csrftoken と [a] 又は [b] とをひも付けるという対策ができていないことが分かった。

〔サイトXのクリックジャッキング脆弱性〕

サイトXでは，クリックジャッキングによって，利用者が気付かずに利用者情報の公開範囲を変更させられてしまう脆弱性が検出された。攻撃者が図3の画面を用いてクリックジャッキングを行う場合を仮定してみる。このとき，クリックイベントは，利用者から見て手前にある画面上で発生するものとする。

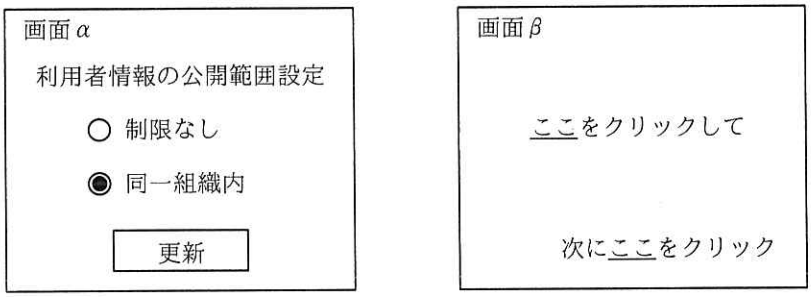


図3　攻撃者が用いる画面

攻撃者は，画面 [c] を利用者から見て [d] に [e] 状態で罠(わな)サイトに公開し，サイト X の画面 [f] を利用者から見て [g] に [h] 状態で重ねて表示する。この状態のサイトにアクセスした利用者は，意図せず利用者情報の公開範囲を変更させられてしまう可能性がある。

クリックジャッキング脆弱性の対策には，レスポンスヘッダに [i] を含める方法と [j] を含める方法がある。後者は標準化されている。

〔サイトYのSSRF脆弱性〕

サイトYでは，例えば，図4のリクエストを受け取ると，A社のニューストピックを取得し，表示するようになっている。

```
GET /news?topic=https://www.a-sha.co.jp/news/20220417.html HTTP/1.1
（省略）
```

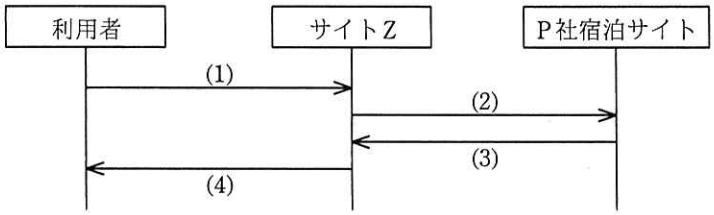
注記　topicパラメタにA社のニューストピックのURLを指定している。

図4　A社のニューストピックを取得するリクエスト

この処理にSSRF脆弱性があった。D社は，③図4のリクエスト中の値を変更してサイトYに送り，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ることを確認した。

〔サイトZのSSRF脆弱性〕

サイトZでは，最近，新機能が開発された。新機能の一つである，旅行会社P社の宿泊サイト（以下，P社宿泊サイトという）との連携機能でSSRF脆弱性が検出された。その機能は，駅名を入力すると，その駅近辺のホテルの割引クーポンなどの“お得情報”を表示できるというものである。利用者が，P社宿泊サイトに登録されている駅名の一つ，“東京”を入力したときの流れを図5に，登録されていない架空の駅名である“abc”を入力したときの流れを図6に，D社の専門技術者V氏がSSRF脆弱性を検出した手順を表4に示す。



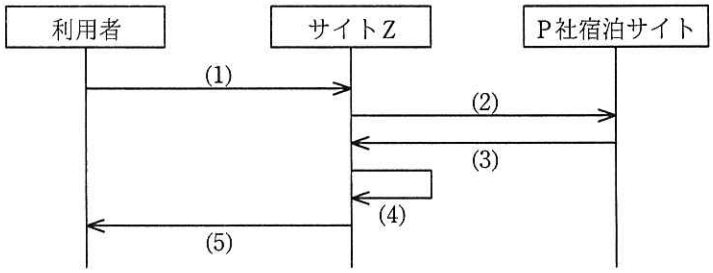
番号	送信されるデータ
(1)	“東京”駅近辺の“お得情報”を取得するリクエスト[1)] GET /station/%E6%9D%B1%E4%BA%AC HTTP/1.1 Host: z.b-sha.co.jp
(2)	“東京”駅近辺の“お得情報”を取得するリクエスト[1) 2)] GET /station/%E6%9D%B1%E4%BA%AC?returnURL=https://z.b-sha.co.jp/error HTTP/1.1 Authorization: Basic （省略）
(3)	“東京”駅近辺の“お得情報”
(4)	“東京”駅近辺の“お得情報”を含むページ

注記　送信されるデータのリクエストヘッダ部分は，一部省略している。

注[1)]　リクエストURIには，“/station/▲▲▲▲▲”の形式で，▲▲▲▲▲に駅名をURLエンコードした値を指定する。

注[2)]　returnURLには，登録されていない駅名が入力されたときに利用されるURLを指定する。サイトZは，(1)のHostヘッダの値を，returnURL中のホスト名として指定する。

図5　登録されている駅名である“東京”を入力したときの流れ



番号	送信されるデータ
(1)	“abc”駅近辺の“お得情報”を取得するリクエスト GET /station/abc HTTP/1.1 Host: z.b-sha.co.jp
(2)	“abc”駅近辺の“お得情報”を取得するリクエスト GET /station/abc?returnURL=https://z.b-sha.co.jp/error HTTP/1.1 Authorization: Basic （省略）
(3)	“abc”駅が登録されていなかったことを知らせるレスポンス HTTP/1.1 301 MOVED PERMANENTLY
(4)	サーバ上にあるエラーページのテンプレート情報 GET /error HTTP/1.1
(5)	登録されていない駅名が入力されたことを示すエラーページ

注記　送信されるデータのリクエストヘッダ部分は，一部省略している。

図6　登録されていない駅名である“abc”を入力したときの流れ

表4　SSRF 脆弱性を検出した手順

順序	手順
1	P社宿泊サイトに登録されていない駅名，例えば，“abc”を入力し，Host ヘッダの値を，V氏が用意したサイトの FQDN に変更して，サイト Z にリクエストを送る。
2	サイト Z は，P 社宿泊サイトに，“/station/abc”をリクエスト URI に指定したリクエストを送る。
3	P 社宿泊サイトは，Location ヘッダに［　k　］の URL を含めたレスポンスをサイト Z に返す。
4	サイト Z は，受け取ったレスポンスを基に，［　k　］にリクエストを送る。

表4の手順によって，V氏は，Authorization ヘッダの値を受け取ることができた。P社の協力の下，この値を用いることで，本来許可なしではアクセスできないP社宿泊サイトや同じ Authorization ヘッダの値を利用する P 社保有のサーバへのアクセスが可能になることを確認した。

D 社からは，P 社宿泊サイトからのレスポンスに含まれる URL が想定されたもの

かを調べて想定外の値の場合はその URL にはアクセスしないようにするという，SSRF 脆弱性への対策が提案された。加えて，④別の対策も実施することが望ましいとのことであった。

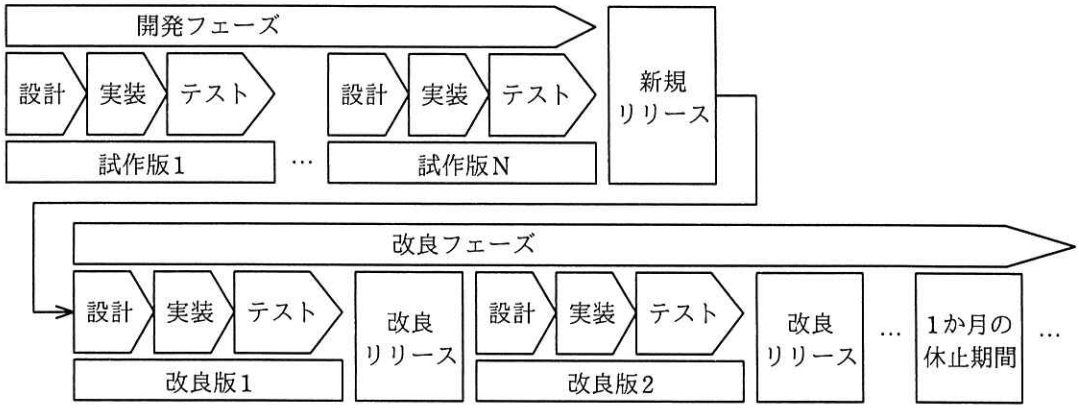
R さんは，D 社の脆弱性診断で検出された脆弱性について，B 社の開発部の E 課長に報告した。その後，B 社の開発部によって対策が実施され，D 社による再度の脆弱性診断で問題が修正されたことが確認された。

〔開発プロセスの見直し〕

B 社の Web サイトのセキュリティ水準について，R さんは，開発プロセスの観点からも調査を進めた。

B 社では，顧客に機能の追加要望や性能の改善要望をヒアリングしながら，開発部内で目標を設定し，アジャイル開発を行っている。また，社外の研修などでセキュアプログラミングの知識を習得し，その知識を生かして Web サイトを開発している。

B 社の開発プロセスの概要を図 7 に示す。



注記 1 ライブラリの活用などで 2 週間周期での改良リリースを実現しているが，およそ 20 回に 1 回は大規模な改修があり，改良リリース間隔を 1 か月とすることがある。

注記 2 改良フェーズにおいて，半年に 1 回，1 か月の休止期間を設けている。その間，開発部のメンバは，長期休暇の取得，長期研修の受講，Web サイトの点検などを実施している。

図 7　B 社の開発プロセスの概要

Rさんは，B社のWebサイトのセキュリティ水準を十分なものにするためには，A社のようなWebセキュリティ管理基準をB社に導入する必要があると考えた。次は，B社の開発プロセスについてのRさんとE課長の会話である。

Rさん：B社でも表1のとおりに実施できますか。

E課長：開発フェーズにおいてはできると思います。しかし，改良リリースの周期は2週間程度です。専門技術者による脆弱性診断には，その周期の大半を費やしてしまうので，省略できないでしょうか。

Rさん：⑤ソースコードレビューやツールによる脆弱性診断では発見できないが，専門技術者による脆弱性診断では発見できる脆弱性が多くあります。専門技術者による脆弱性診断を改良リリースにおいて毎回実施できない場合でも，当該診断が長期間行われないことを避けるために，⑥時期を決めて実施することや，⑦開発プロセスを見直すことを検討してみてください。

E課長：分かりました。そのほかに，アジャイル開発に合った脆弱性対策はないでしょうか。

Rさん：Webサイトの実装に必要となる一般的な機能や定型コードを，ライブラリとしてあらかじめ用意したフレームワークには，⑧脆弱性対策が組み込まれていて，それがデフォルトで有効になっているものもあるので，利用を検討してみてください。

その後，B社は，セキュリティを考慮したアジャイル開発を行うことになった。

設問１　〔サイトＢのXSS脆弱性〕について，(1)，(2)に答えよ。

(1) 本文中の下線①における診断用リクエストの構成要素を，解答群の中から選び，記号で答えよ。

解答群

ア　リクエストライン：GET /confirm?"><img src=1 onerror=alert(1)><"

イ　リクエストライン：GET /confirm?><img src=1 onerror=alert(1)><"

ウ　リクエストライン：POST /confirm

　　リクエストボディ："><img src=1 onerror=alert(1)><"

エ　リクエストライン：POST /confirm

　　リクエストボディ：><img src=1 onerror=alert(1)><"

(2) 本文中の下線②について，考えられる再発防止策を，35字以内で述べよ。

設問２　本文中の［　a　］，［　b　］に入れる適切な字句を答えよ。

設問３　〔サイトＸのクリックジャッキング脆弱性〕について，(1)，(2)に答えよ。

(1) 本文中の［　c　］～［　h　］に入れる適切な字句を，それぞれの解答群の中から選び，記号で答えよ。

c，fに関する解答群

ア　α　　イ　β

d，gに関する解答群

ア　奥　　イ　手前

e，hに関する解答群

ア　可視の　　イ　透明な

(2) 本文中の［　i　］，［　j　］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　Content-Disposition　　イ　Content-Security-Policy

ウ　X-Content-Type-Options　　エ　X-Frame-Options

設問４　本文中の下線③について，図4のリクエスト中のどの値をどのような値に変更したか。45字以内で具体的に述べよ。

設問5　〔サイトZのSSRF脆弱性〕について，(1)，(2)に答えよ。

(1) 表4中の［　k　］に入れる適切な字句を，15字以内で答えよ。

(2) 本文中の下線④について，別の対策とは何か。B社で実施することが望ましい対策を，25字以内で述べよ。

設問6　〔開発プロセスの見直し〕について，(1)～(4)に答えよ。

(1) 本文中の下線⑤について，該当する脆弱性を二つ挙げ，それぞれ15字以内で答えよ。

(2) 本文中の下線⑥について，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ためには，どのような時期に実施すればよいか。改良リリースの実施に影響を与えないことを前提に，20字以内で答えよ。

(3) 本文中の下線⑦について，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ためには，開発プロセスをどのように見直せばよいか。アジャイル開発の継続を前提に，40字以内で述べよ。

(4) 本文中の下線⑧について，CSRF脆弱性の場合では，どのような処理を行う機能が考えられるか。その処理を，55字以内で具体的に述べよ。