

AppGoatを利用した集合教育補助資料 -セッション管理の不備編-

独立行政法人情報処理推進機構 (IPA)
セキュリティセンター

※事前準備

「セッション管理の不備」の演習を行うには、以下の準備が必要となります

- 2種類のブラウザを用意する。
 - ・AppGoat推奨ブラウザ
Internet Explorer、FireFox、Google Chrome、Edge
- ネットワークキャプチャツール「Fiddler」(Telerik社からフリーソフトウェアとして公開)をインストールする。
<https://www.telerik.com/download/fiddler>

Fiddlerのインストール方法および設定方法は、
補助資料『Fiddlerの使い方』を参照してください。

- 脆弱性の原理解説・基礎知識
- 脆弱性の発見方法
- 演習1:セッションIDの推測
- 演習解説



セッション管理の不備とは？

- 通信の際の個人認証情報の管理に不備がある脆弱性
- セッション管理の不備を悪用し、第三者になりすまして通信を行う攻撃を、「セッション・ハイジャック」と呼ぶ。
- 攻撃者がなりすましに成功した場合、利用者本人にしか許可されていない操作を不正に行うことができる。
- ログイン機能を持つウェブサイトは注意する必要がある。特に、ログイン後に決済処理等の重要な処理を行うサイトは攻撃された際の被害が大きくなる。

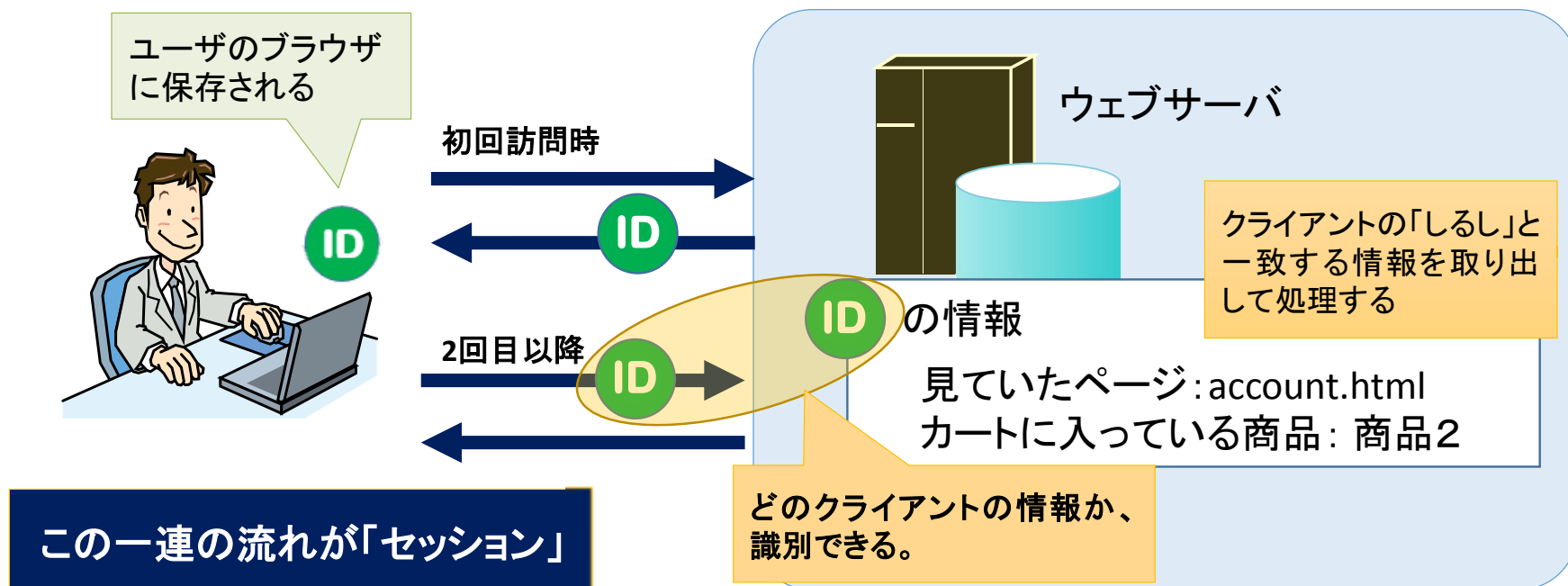
セッション管理の基礎知識

● セッションとは？

ウェブサイトへの要求から応答までの一連の流れ

● セッションIDとは？

クライアントを識別するためのしるし



● セッションIDの送信手段

セッションIDの送信手段には主に以下の3つがある。

- ①URLリライティング
URLパスにセッションIDを書き込む
(セキュリティ上、推奨されない)
- ②Cookieを利用して送信
- ③POSTデータのhidden属性で送信

HTTP リクエスト ①

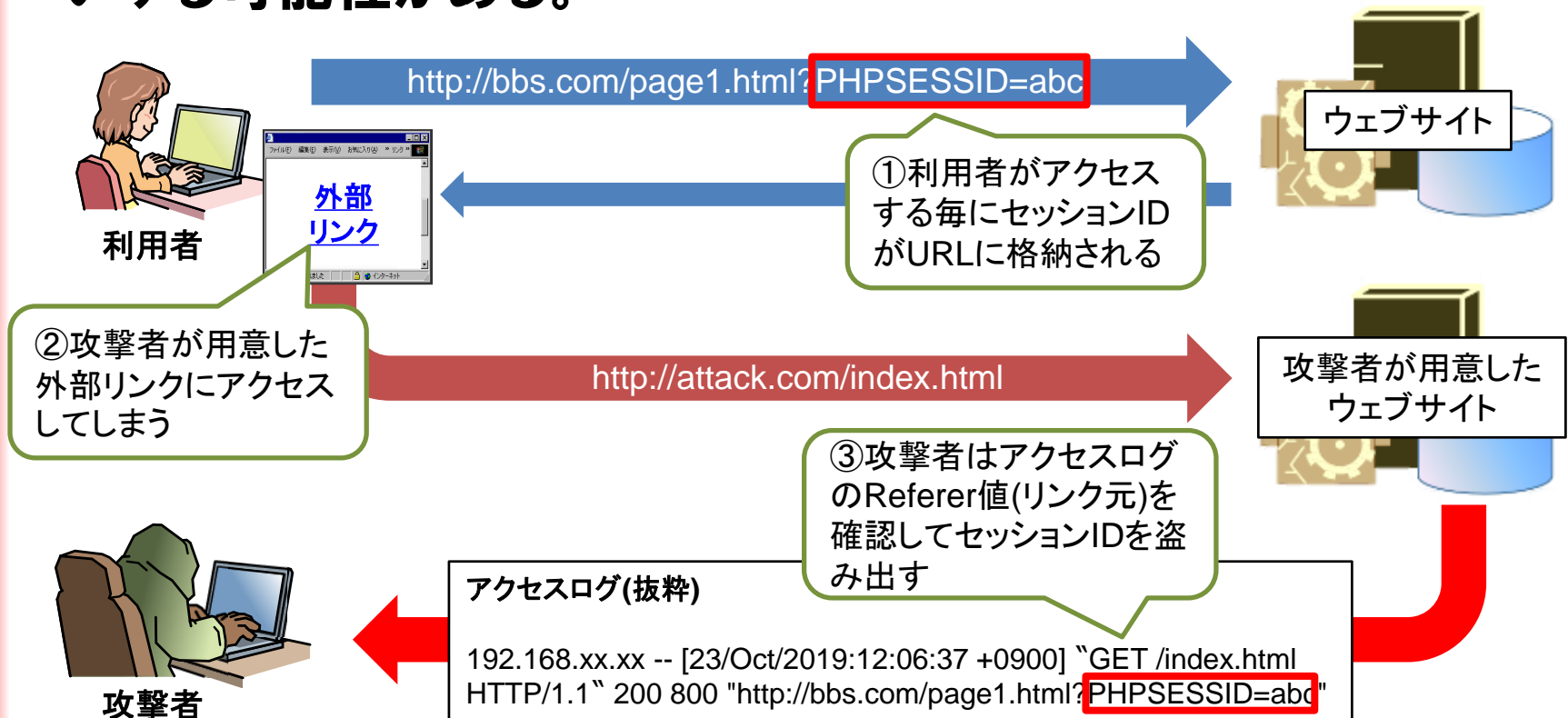
```
POST /ipa/index.html sessionid=3M90L2 HTTP/1.1
Host: www.ipa.go.jp
Referer: http://www.ipa.go.jp/top.html
Cookie: sessionid=3M90L2 ②
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 71
```

```
loginpass=&sessionid=3M90L2 ③
```

セッション管理の不備のタイプ

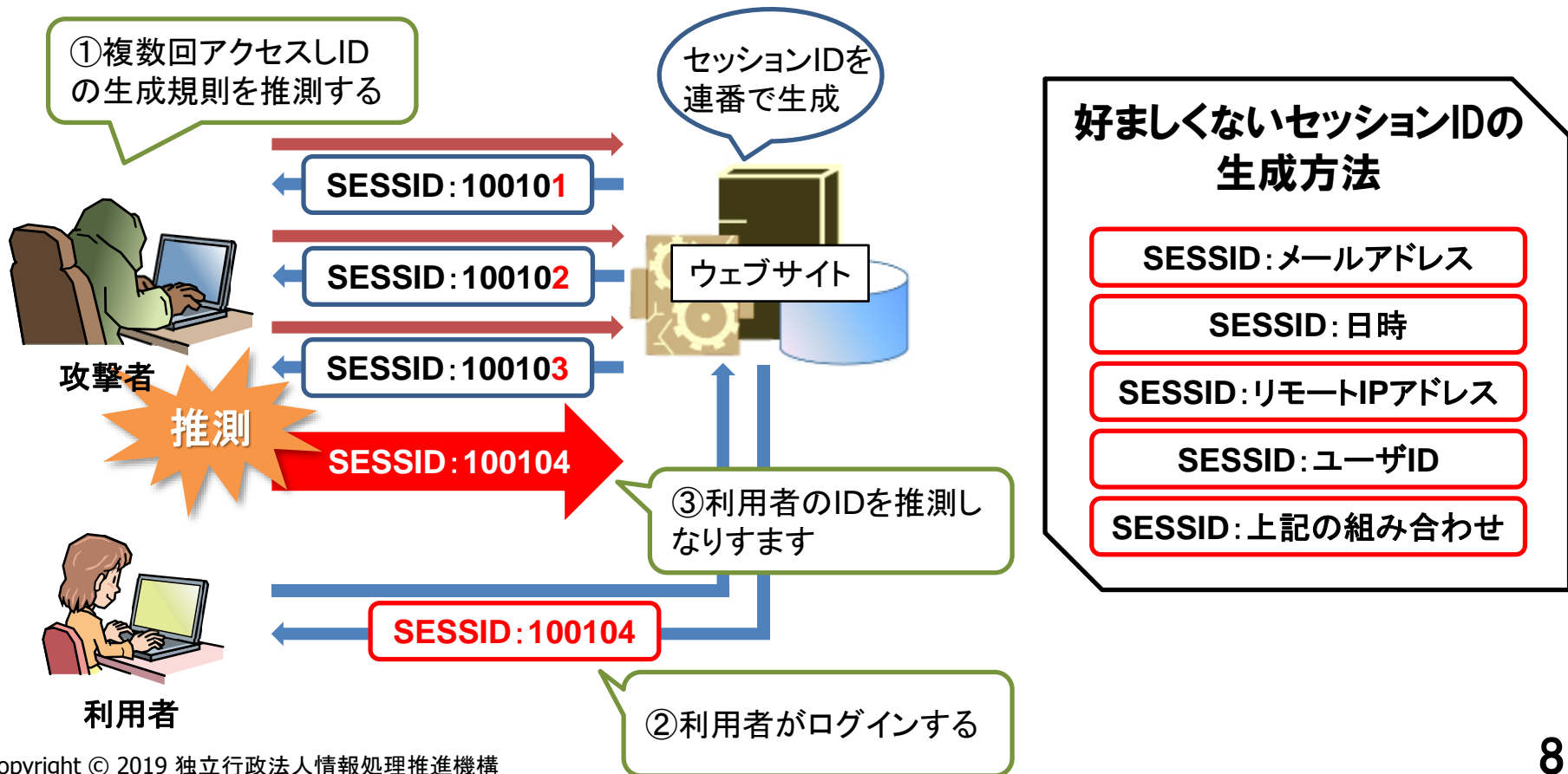
● セッションIDの漏えい

セッションIDをURLに埋め込んでいたり、セキュリティ設定に不備がある場合、通信を盗聴されるなどしてセッションIDが漏えいする可能性がある。



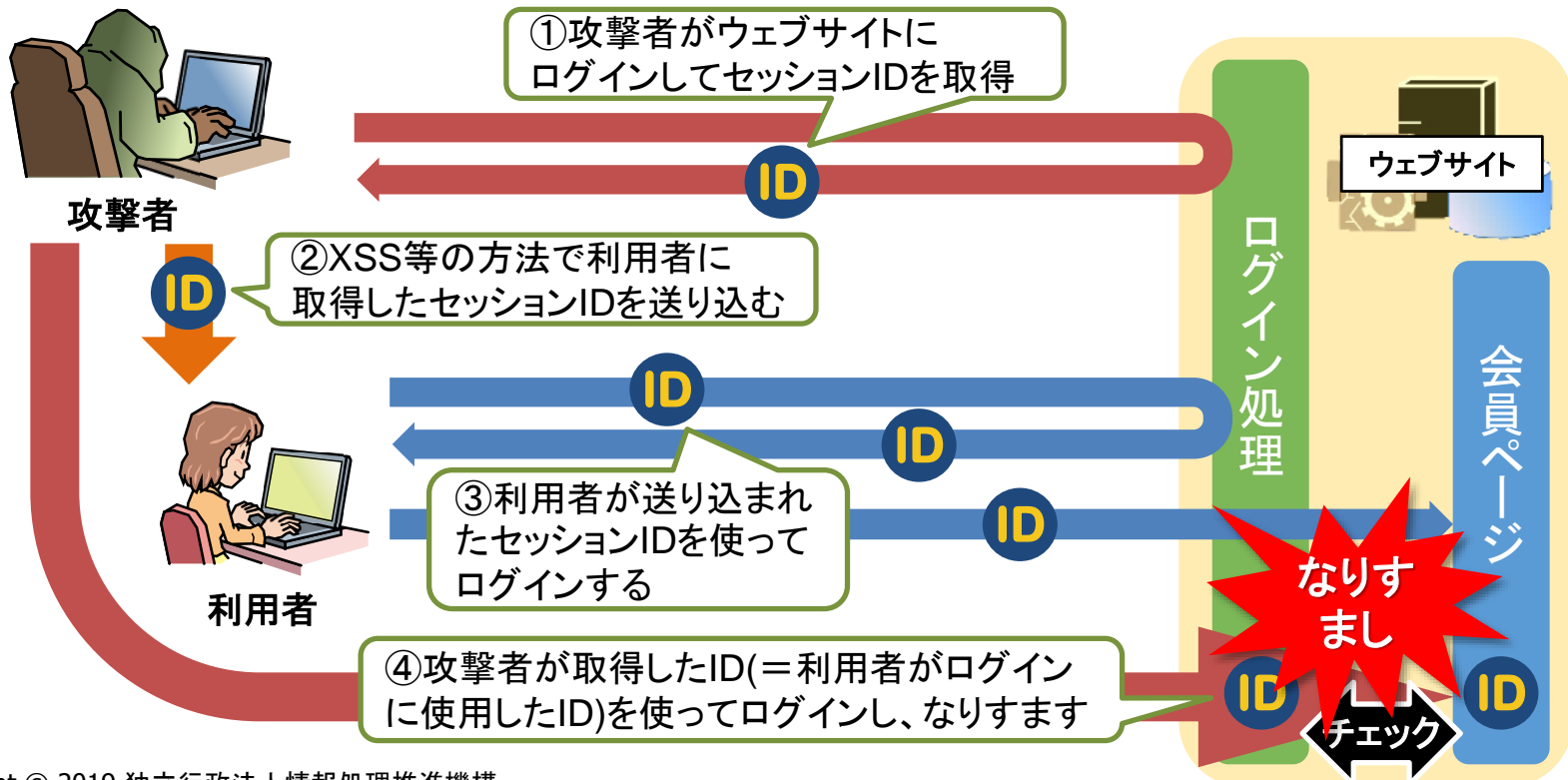
● セッションIDの推測

セッションIDを生成する規則が単純な場合、攻撃者にセッションIDを予測され、利用者になりすまされる可能性がある。



●セッションIDの固定化

セッションIDを更新せずに使い続けているウェブアプリケーションでは、攻撃者が取得したセッションIDを他の利用者のIDとして固定されてしまうことで、なりすまされる可能性がある。



セッション管理の不備を発見するために

ポイント: ネットワークキャプチャツールを使って セッションIDを確認する

- セッションIDの**生成規則が単純**
→ セッションIDの推測
- ログインの前後でセッションIDが**更新されない**
→ セッションIDの固定化



ネットワークキャプチャツール (Fiddler) の使い方は、
補助資料『Fiddlerの使い方』を参照してください。

[演習] AppGoatの準備

① AppGoatを起動します

② 以下の遷移で演習画面に移動します

1. 「学習環境へ」クリック

2. 「はい」クリック

3. 「セッションIDの推測」クリック

4. 「演習(発見)」クリック

注意事項

本ツールには、故意に脆弱性を内在させたウェブアプリケーションを使った演習が含まれています。これらの演習は、脆弱性の原因や対策方法を理解するためのものです。

本ツールで学習した知識を基に、下記に該当する行為を行わないでください。

- 第三者のコンピュータシステムの正常な動作に支障をきたす行為
- 第三者のウェブサイトへの検査や攻撃行為
- AppGoatにおいて、自分が演習する範囲以外への検査や攻撃行為

なお、脆弱性の対策方法は、複数ある方法のうちの一例であり、必ずしもこの方法で対策される際は、当該ウェブアプリケーションに合わせた対策方法を使用して下さい。

上記に同意しますか？

はい いいえ

セッション管理の不備

- セッション管理の不備
- Level1
 - 脆弱性の概要および発見演習
- Level2
 - セッションIDの漏えい
- Level3
 - セッションIDの固定化
- 習熟度テスト
 - テスト問題 全5問

基本情報技術者試験の合格者、または同等のスキル

PHPを使ったウェブアプリケーション開発経験6ヶ月以上の方または同等のスキル

学習を行う脆弱性によっては以下のスキルがあることが前提になります。

- ・ 正規表現を使ったプログラムが作成できる
- ・ SQLを使ったプログラマーが作成できる

学習に必要なソフトウェアおよび設定

テマー一覧の操作方法

左のメニューから、学習をしたい脆弱性(最初はイントロダクション配下を選択)を選んで「基礎」のように記載されている「+」と「-」はメニューの開閉状態を表しています。「+」のすると、配下のメニューが表示され、「-」のときにクリックすると配下のメニューが非表示

演習(発見)

チャレンジ
複数ログインを行い、セッションIDの規則性を発見し、セッションIDを推測しましょう。

[演習] AppGoatを用いた疑似攻撃体験

IPA

AppGoat
～突いてみますか？脆弱性！～

- 演習テーマ：

「セッションIDの推測」



- ミッション：

セッションIDの規則性を見つけましょう！



[演習] 演習の進め方

■ Step1:セッションIDを収集する

- ・ウェブサイトに複数回ログインして、セッションIDを確認する。



オンラインバンキング

ログインID

パスワード

ログインID:sato

パスワード:sato123

■ Step2:セッションIDの規則性を考える

- ・収集した複数のセッションIDから、セッションIDの生成規則を推測する。

ヒント3を見て答え合わせをしましょう

演習はじめてください。

※演習が終わったら次のページで解説を行います。

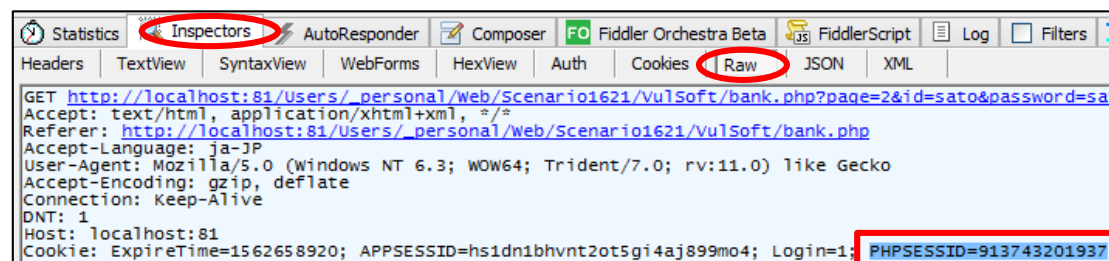


[手順1] セッションIDを複数回取得する

演習の手順

繰り返しログインをして、リクエストヘッダからセッションIDを確認しましょう

- ログイン名「sato」、パスワード「sato123」を入力し、ログインを行います。
- Fiddlerでリクエストヘッダを表示し、Inspectorsタブ内RawのCookieでセッションIDを確認します。



```
GET http://localhost:81/Users/_personal/Web/Scenario1621/VulSoft/bank.php?page=2&id=sato&password=sato
Accept: text/html,application/xhtml+xml,*/*
Referer: http://localhost:81/Users/_personal/Web/Scenario1621/VulSoft/bank.php
Accept-Language: ja-JP
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
DNT: 1
Host: localhost:81
Cookie: ExpireTime=1562658920; APPSESSID=hs1dn1bhvnt2ot5gi4aj899mo4; Login=1; PHPSESSID=913743201937
```

PHPSESSID=913743201937



この作業を複数回 (3~4回) 繰り返しましょう。

[手順2]



セッションIDの生成規則を推測する

IPA

AppGoat

～突いてみますか？脆弱性！～

演習の手順

取得したセッションIDから、生成規則を考えましょう。

- 取得した複数のセッションIDを見比べ、変化している部分に注目して生成規則を推測します。

PHPSESSID= 91374**3**2019**37**

PHPSESSID= 91374**4**2019**06**

PHPSESSID= 91374**4**2019**41**

考えられる生成規則の例

SESSID:メールアドレス

SESSID:日時

SESSID:リモートIPアドレス

SESSID:ユーザID

SESSID:上記の組み合わせ

日+時+月+分+年+秒でセッションIDが生成されていた。(2019/7/9 13:4**X:XX**)

- 演習2:セッションIDの固定化
- 演習解説
- 対策方法



[演習] AppGoatの準備

①以下の遷移で演習画面に移動します

- セッション管理の不備	<ul style="list-style-type: none">・ 基本情報技術者試験の合格者、または同等のスキル・ PHPを使ったウェブアプリケーション開発経験6ヶ月以上 <p>学習を行う脆弱性によっては以下のスキルがあることが</p> <ul style="list-style-type: none">・ 正規表現を使ったプログラムが作成できる・ SQLを使ったプログラムが作成できる・ JavaScriptを使ったWebページを作成できる
- イントロダクション	
セッション管理の不備とは	
- Level1	
脆弱性の概要および発見演習	
- Level2	
セッションIDの準備	
セッションIDの漏えい	
- Level3	
セッションIDの固定化	
- 習熟度テスト	

1. 「セッションIDの固定化」クリック

テーマ一覧の操作方法

左のメニューから「基礎」のようになると、配下の

IPA 脆弱性体験学習ツール AppGoat

総合メニュー 学習を進める前に 学習環境へ 学習状況の初期化 学習状況表示 FAQ 利用者マニュアル AppGoatの終了方法

テーマ一覧

表示中のページ

- 基礎
 - セッション管理の不備
 - Level3
 - セッションIDの固定化
- 基礎
 - + クロスサイト・スクリプティング
 - + SQLインジェクション
 - + CSRF(クロスサイト・リクエストフォージェリ)
 - + ディレクトリ・トラバーサル
 - + OSコマンド・インジェクション
- セッション管理の不備
- イントロダクション
 - セッション管理の不備とは

セッションIDの固定化

2. 「演習(発見)」クリック

戻る 次へ

演習(発見) (1/2)

チャレンジ

セッションIDの固定化を行い、他人に成りすましてウェブアプリケーションに不正にログインしてみましょう。

疑似的な攻撃

[演習] AppGoatを用いた疑似攻撃体験



- 演習テーマ:

「セッションIDの固定化」

- ミッション:

他の利用者になりすましてみましょう。



ネット証券に内在する脆弱性を発見しました。

[演習] 演習の進め方

■ Step1: 脆弱性を発見する



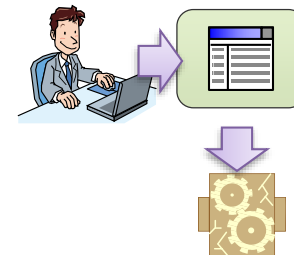
- ・ログインしてセッションIDを取得し、脆弱性があることを確認する。

■ Step2: 罣リンクを設置する



- ・クロスサイトスクリプティングの脆弱性を利用し、外部サイトに罣リンクを設置する。

■ Step3: 罣リンク経由でウェブサイトへアクセスする



- ・被害者の立場になって罣リンクへアクセスし、セッションIDが書き換えられていることを確認する。

■ Step4: 再度ログインし、利用者になります



- ・最初取得したセッションIDでログインすることで、ユーザ名や口座番号が被害者のものになっていることを確認する。

演習はじめてください。

※演習が終わったら次のページで解説を行います。



[Step 1]



セッションIDを取得する (攻撃者)

IPA

AppGoat

～突いてみますか？脆弱性！～

演習の手順

セッションIDを取得し、セッション管理の不備がないか確認しましょう。

- ログイン画面にアクセスしてFiddlerでリクエストヘッダを表示し、Inspectorsタブ内RawのCookieでセッションIDを確認します。

```
Statistics Inspectors AutoResponder Composer FO Fiddler Orchestra Beta FiddlerScript Log Filters Timeline
Headers TextView SyntaxView WebForms HexView Auth Cookies Raw JSON XML
GET http://localhost/Users/_personal/Web/Scenario1631/VulSoft/bond.php HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Referer: http://localhost/main.php?scenario=Scenario1631&stage=stage4&page=page1
Accept-Language: ja-JP
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
DNT: 1
Host: localhost
Cookie: ExpireTime=1563426703; APPSESSID=q6u86nt6d76u1dsum45acutqe7; Login=1; PHPSESSID=q6u86nt6d76u1dsum45acutqe7
```

PHPSESSID=q6u86nt6d76u1dsum45acutqe7

- ログイン名「sato」、パスワード「sato123」を入力し、ログインを行います。

[Step 1]



セッションIDを取得する (攻撃者)

IPA

AppGoat

～突いてみますか？脆弱性！～

演習の手順

セッションIDを取得し、セッション管理の不備がないか確認しましょう。

- 再びFiddlerでレスポンスのセッションIDを確認します。

```
Response body is encoded. Click to de...
Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching Cookies Raw JSON XML
HTTP/1.1 200 OK
Date: Thu, 18 Jul 2019 03:15:33 GMT
Server: Apache
Set-Cookie: ExpireTime=1563426933; expires=Thu, 18-Jul-2019 05:15:33 GMT; Max-Age=7200; path=/; httponly
Set-Cookie: APPSESSID=q6u86nt6d76u1dsum45acutqe7; expires=Thu, 18-Jul-2019 05:15:33 GMT; Max-Age=7200; path=/; httponly
Set-Cookie: Login=1; path=/; httponly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Set-Cookie: PHPSESSID=q6u86nt6d76u1dsum45acutqe7; expires=Thu, 18-Jul-2019 05:15:33 GMT; Max-Age=7200; path=/
X-Frame-Options: SAMEORIGIN
```

PHPSESSID=q6u86nt6d76u1dsum45acutqe7

※このセッションIDは次のStepで使用するため、メモ等に保存してください。

ログイン前と後のセッションIDが同じ
→セッションの固定化ができてしまう

[演習] 演習の進め方

■ Step1: 脆弱性を発見する



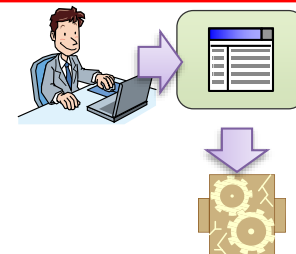
- ・ログインしてセッションIDを取得し、脆弱性があることを確認する。

■ Step2: 罣リンクを設置する



- ・クロスサイトスクリプティングの脆弱性を利用し、外部サイトに罣リンクを設置する。

■ Step3: 罣リンク経由でウェブサイトへアクセスする



- ・被害者の立場になって罣リンクへアクセスし、セッションIDが書き換えられていることを確認する。

■ Step4: 再度ログインし、利用者になります



- ・最初取得したセッションIDでログインすることで、ユーザ名や口座番号が被害者のものになっていることを確認する。

[Step2]



罣リンクを設置する (攻撃者)

IPA



演習の手順

XSSの脆弱性を利用し、外部サイトにIDを書き換える罣リンクを設置しましょう。

- ネット証券アプリケーションの「お問い合わせ」ページの「ご連絡先メールアドレス」欄にクロスサイトスクリプティングの脆弱性が存在します。攻撃方法は「クロスサイトスクリプティング編」を参照してください。

1. XSS攻撃に使うスクリプトを含んだURLを、「ヒント2」を参考に作成します。

ヒント2

- ・【掲示板】のURL欄に以下のURLを投稿します。attackidの箇所に④でメモしたセッションIDを埋め込みます。
「`http://ホスト名:ポート番号/Users/アカウント名/Web/Scenario1631/VulSoft/bond.php?page=9&mail=<script>document.cookie="PHPSESSID=attackid;"</script>`」



`http://ホスト名:ポート番号/Users/アカウント名/Web/Scenario1631/VulSoft/bond.php?page=9&mail=<script>document.cookie="PHPSESSID=q6u86nt6d76u1dsum45acutqe7;"</script>`

[Step2]



罣リンクを設置する (攻撃者)

IPA

AppGoat

～突いてみますか？脆弱性！～

演習の手順

XSSの脆弱性を利用し、外部サイトにIDを書き換える罣リンクを設置しましょう。

2. 掲示板サイトにアクセスし、罣のリンクを作成します。

URLの欄には、先ほど作成したスクリプトを実行させるURLを入力します。

掲示板

管理者の発言:このページでは、決して誹謗中傷などは行わないでください。
*のついている項目は入力必須です。

*名前:

*タイトル:

*本文:

URL:

3. 投稿するとリンクを設置することができます。

11 [キャンペーン中](#) : NetBond-ネット証券 2019年07月18日 12:19
今なら手数料1万円キャッシュバック！

4. ネット証券アプリケーションに戻り、一旦ログアウトします。

[演習] 演習の進め方

■ Step1: 脆弱性を発見する



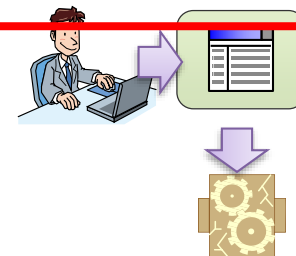
- ・ログインしてセッションIDを取得し、脆弱性があることを確認する。

■ Step2: 罣リンクを設置する



- ・クロスサイトスクリプティングの脆弱性を利用し、外部サイトに罣リンクを設置する。

■ Step3: 罣リンク経由でウェブサイトへアクセスする



- ・被害者の立場になって罣リンクへアクセスし、セッションIDが書き換えられていることを確認する。

■ Step4: 再度ログインし、利用者になります



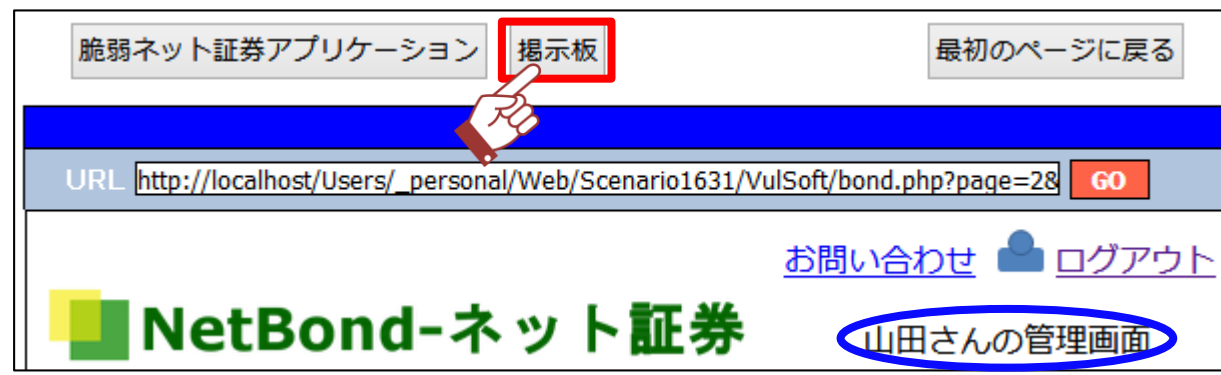
- ・最初取得したセッションIDでログインすることで、ユーザ名や口座番号が被害者のものになっていることを確認する。

[Step3] 罣リンクにアクセスする (被害者)

演習の手順

被害者の立場になって罣リンクにアクセスし、セッションIDを確認しましょう。

- 被害者側のブラウザで、ログイン名「yamada」、パスワード「yamada123」でウェブサイトにログインした後、掲示板サイトで罣リンクにアクセスします。



利用者の管理画面

11 キャンペーン中 : NetBond-ネット証券 2019年07月18日 12:19
今から手数料1万円キャッシュバック!

[Step3] 罠リンクにアクセスする (被害者)



演習の手順

被害者の立場になって罠リンクにアクセスし、セッションIDを確認しましょう。

- リンクをクリックすると、ネット証券のお問い合わせフォームにアクセスします。(XSSの脆弱性が存在するページ)
- この時点でスクリプトが実行され、セッションIDが書き換えられているため、他のページにアクセスしようとするすると再度ログインを求められます。

NetBond-ネット証券 山田さんの管理画面

お問い合わせ

口座残高表示 *のついている項目は入力必須です。

株式購入 お名前を入力してください。

持ち株確認 ご連絡先メールアドレスが不正です。

*お名前

*ご連絡先メールアドレス



NetBond-ネット証券

ログイン

ログインが必要です。

ログインIDとパスワードを入力し「ログイン」ボタンを押してください。

ログインID

パスワード

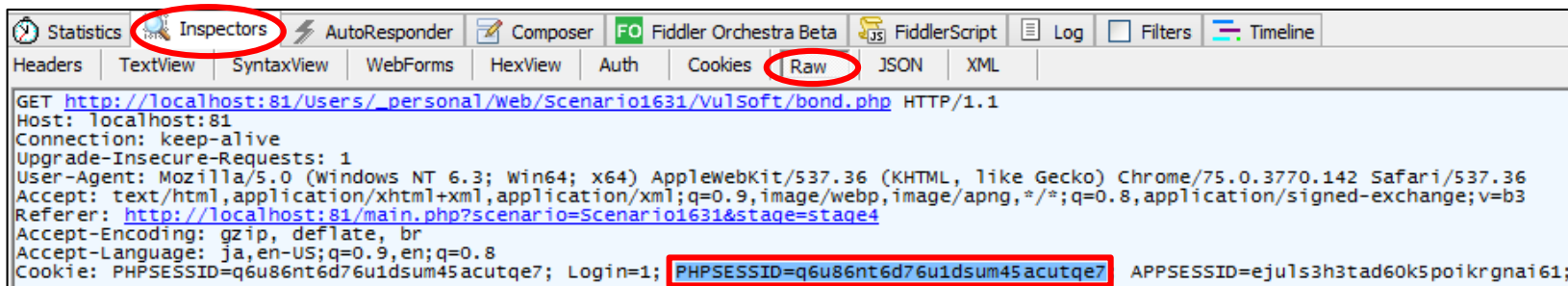
ログイン クリア

[Step3] 罣リンクにアクセスする (被害者)

演習の手順

被害者の立場になって罣リンクにアクセスし、セッションIDを確認しましょう。

- 再度ログインした後、Fiddlerでリクエストヘッダを表示し、Inspectorsタブ内RawのCookieでセッションIDを確認します。



```
Statistics Inspectors AutoResponder Composer FO Fiddler Orchestra Beta FiddlerScript Log Filters Timeline
Headers TextView SyntaxView WebForms HexView Auth Cookies Raw JSON XML
GET http://localhost:81/Users/_personal/Web/Scenario1631/VulSoft/bond.php HTTP/1.1
Host: localhost:81
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.142 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Referer: http://localhost:81/main.php?scenario=Scenario1631&stage=stage4
Accept-Encoding: gzip, deflate, br
Accept-Language: ja,en-US;q=0.9,en;q=0.8
Cookie: PHPSESSID=q6u86nt6d76u1dsum45acutqe7; Login=1; PHPSESSID=q6u86nt6d76u1dsum45acutqe7 APPSESSID=eju1s3h3tad60k5poikrgnai61;
```

PHPSESSID=q6u86nt6d76u1dsum45acutqe7

セッションIDの更新がされない仕様の
ため、攻撃者が取得したIDと同じ
→セッションIDの固定化

[演習] 演習の進め方

■ Step1:脆弱性を発見する



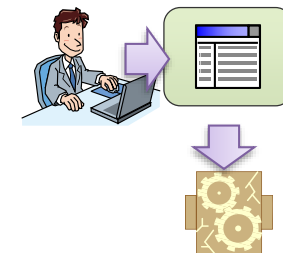
- ・ログインしてセッションIDを取得し、脆弱性があることを確認する。

■ Step2:罨リンクを設置する



- ・クロスサイトスクリプティングの脆弱性を利用し、外部サイトに罨リンクを設置する。

■ Step3:罨リンク経由でウェブサイトアクセスする



- ・被害者の立場になって罨リンクにアクセスし、セッションIDが書き換えられていることを確認する。

■ Step4:ログイン後のページにアクセスし、利用者になります

- ・ログイン後のページにアクセスすることで、ユーザ名や口座番号が被害者のものになっていることを確認する。



[Step4]



利用者になります (攻撃者)

IPA

AppGoat

～突いてみますか？脆弱性！～

演習の手順

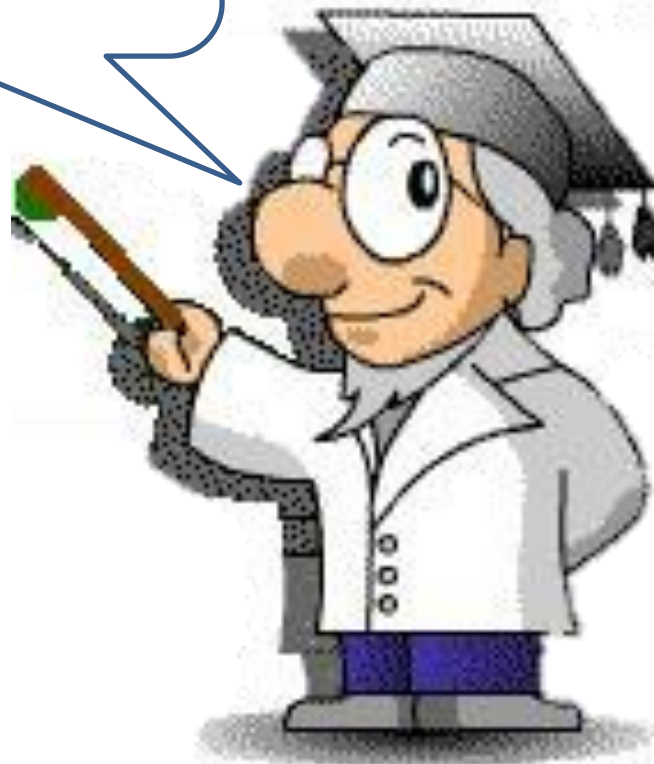
ログイン後のページにアクセスし、なりすましできることを確認しましょう。

- 攻撃者側のブラウザに戻り、ログイン後のURLにアクセスします。
(ここでは、【脆弱ネット証券アプリケーション】のボタンをクリックすることで可能です。)

The screenshot shows a web browser window with the URL `http://localhost:81/Users/_personal/Web/Scenario1631/VulSoft/bond.php`. The page displays the NetBond logo and a message: "ネット証券に内在する脆弱性を発見しました。" (A vulnerability inherent in the NetBond securities was discovered). Below the message, there are links for "お問い合わせ" (Contact Us) and "ログアウト" (Logout). A blue oval highlights the link "山田さんの管理画面" (Yamada's Management Screen), which the attacker is impersonating. A hand icon points to the "脆弱ネット証券アプリケーション" button in the top navigation bar.

攻撃者が、利用者になりすまして
管理画面を操作できてしまう。

対策の解説



● セッションIDの漏えいの対策

■ セッションIDの送信手段としてURLを使わない

URLではなくCookie等を使うことで、攻撃者にセッションIDを窃取されるリスクを減らすことができる。

■ HTTPS通信で利用するCookieにはセキュア属性を付与する

Cookieの設定項目にはセキュア属性があり、これを設定することで、HTTPS通信に限定することができる。

これによって、セッションIDの送信にCookieを使っている場合、HTTP通信の盗聴による漏えいを防ぐことができる。

● セッションIDの推測の対策

■ 推測が困難なセッションIDを使用する

主要なウェブアプリケーション開発ツールには、推測困難なセッションIDを生成してくれるセッション管理機構が実装されている。

どうしてもセッション管理機構を自作する必要がある場合は暗号論的疑似乱数生成器を利用して、十分な桁数のセッションIDを生成する。

例えば、PHPでは、`mcrypt ()` が良く使用される。

● セッションIDの固定化の対策

■ ログイン後にセッションIDを更新する

ログインが成功した時点から新しいセッションを開始し、既存のセッションIDを無効にすることでなりすましを防ぐことができる。

■ セッションIDに変わる「しるし」を用いて対策する

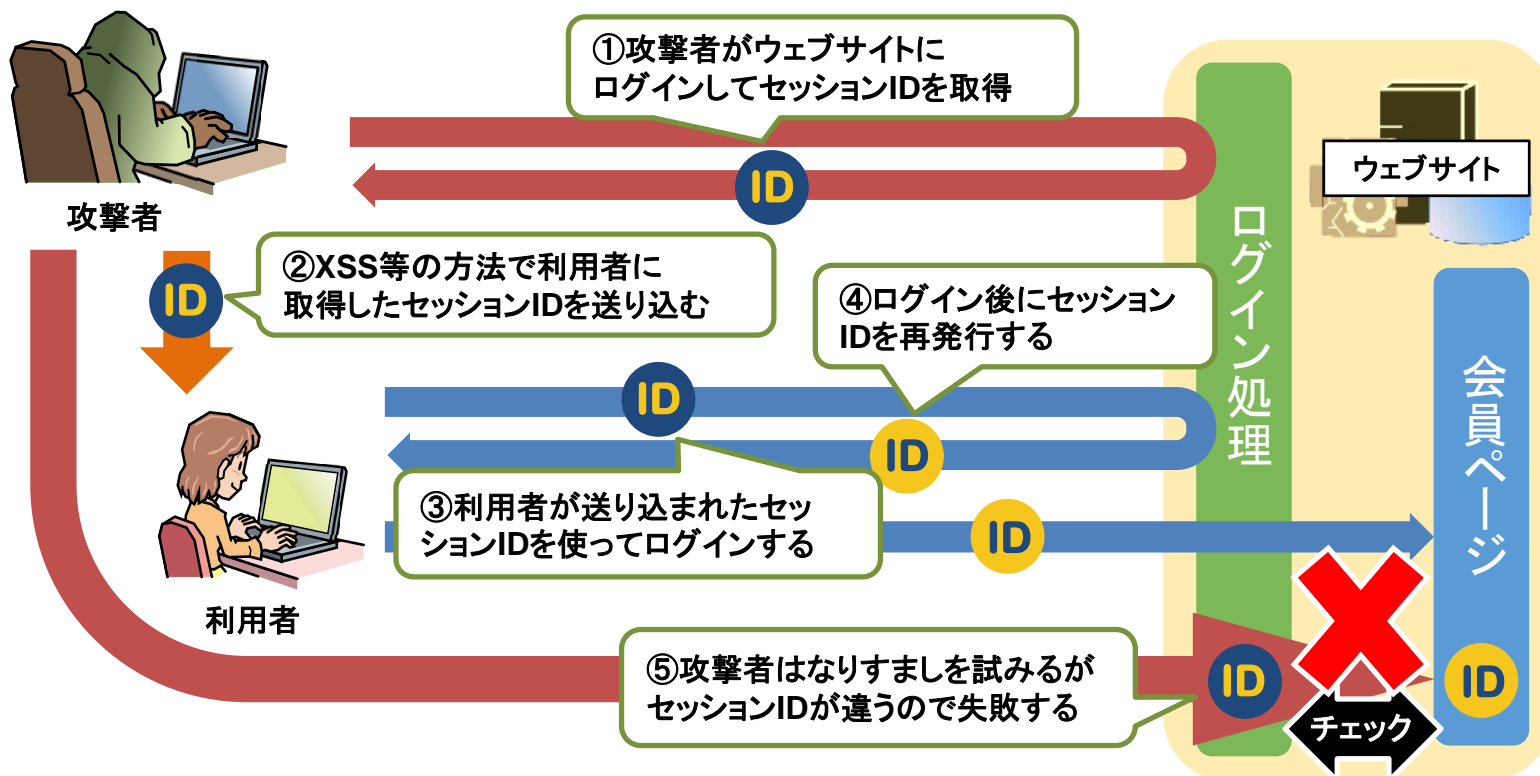
セッションIDの更新ができない場合は、セッションIDとは別に、ログイン成功時に推測困難な秘密情報(しるし)を作成してCookieに格納する。

すべてのページで秘密情報とCookieの値が一致するかをチェックすることで、なりすましを防ぐことができる。

セッション管理の不備の対策

● セッションIDの固定化の対策

■ ログイン後にセッションIDを変更する対策のイメージ



以上で、
セッション管理の不備の解説は
終了です。

