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。

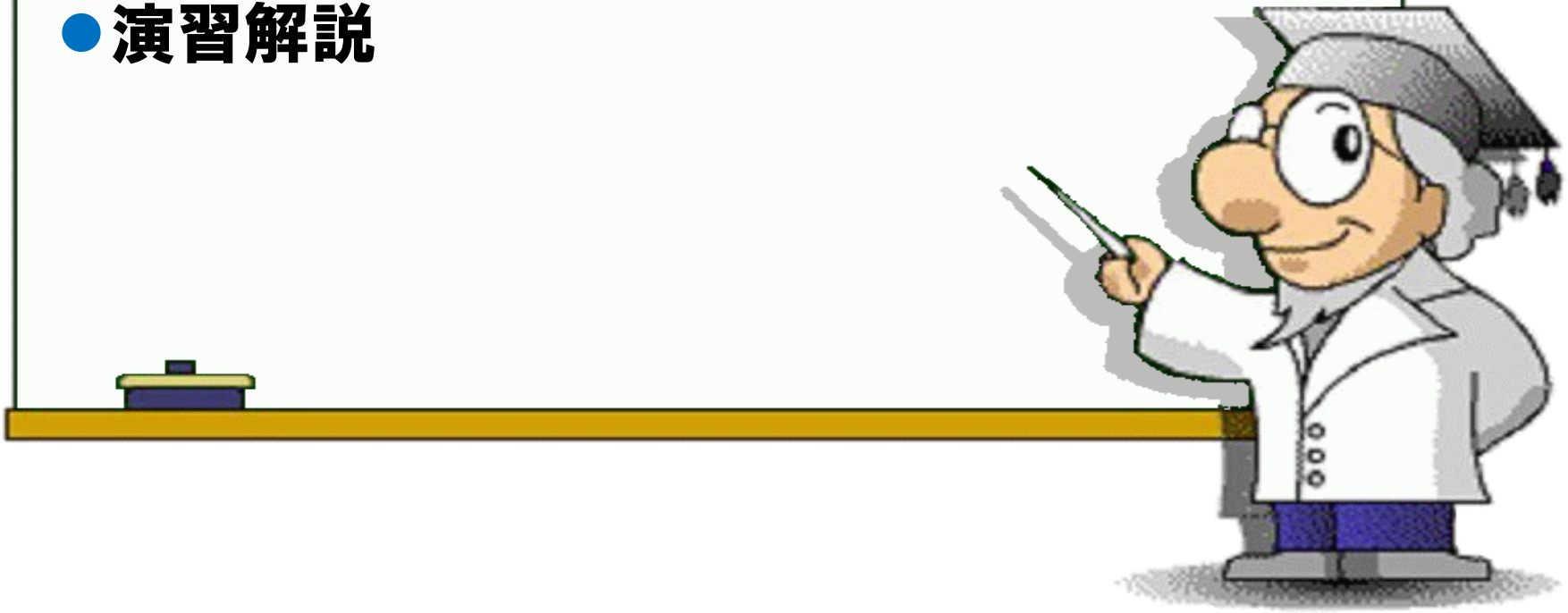
  ・AppGoat推奨ブラウザ
  Internet Explorer、FireFox、Google Chrome、Edge

- 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。

内容（演習１）

- 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

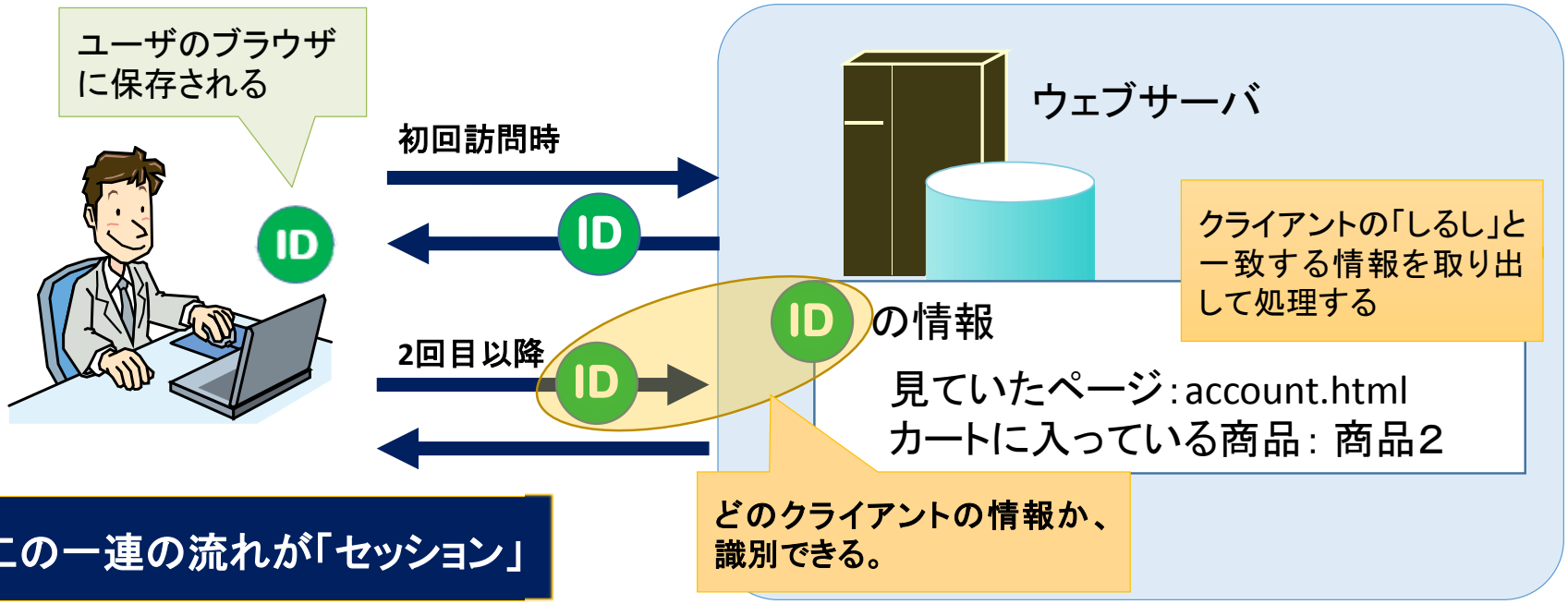
IPA

- **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



セッション管理の基礎知識

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 ❶

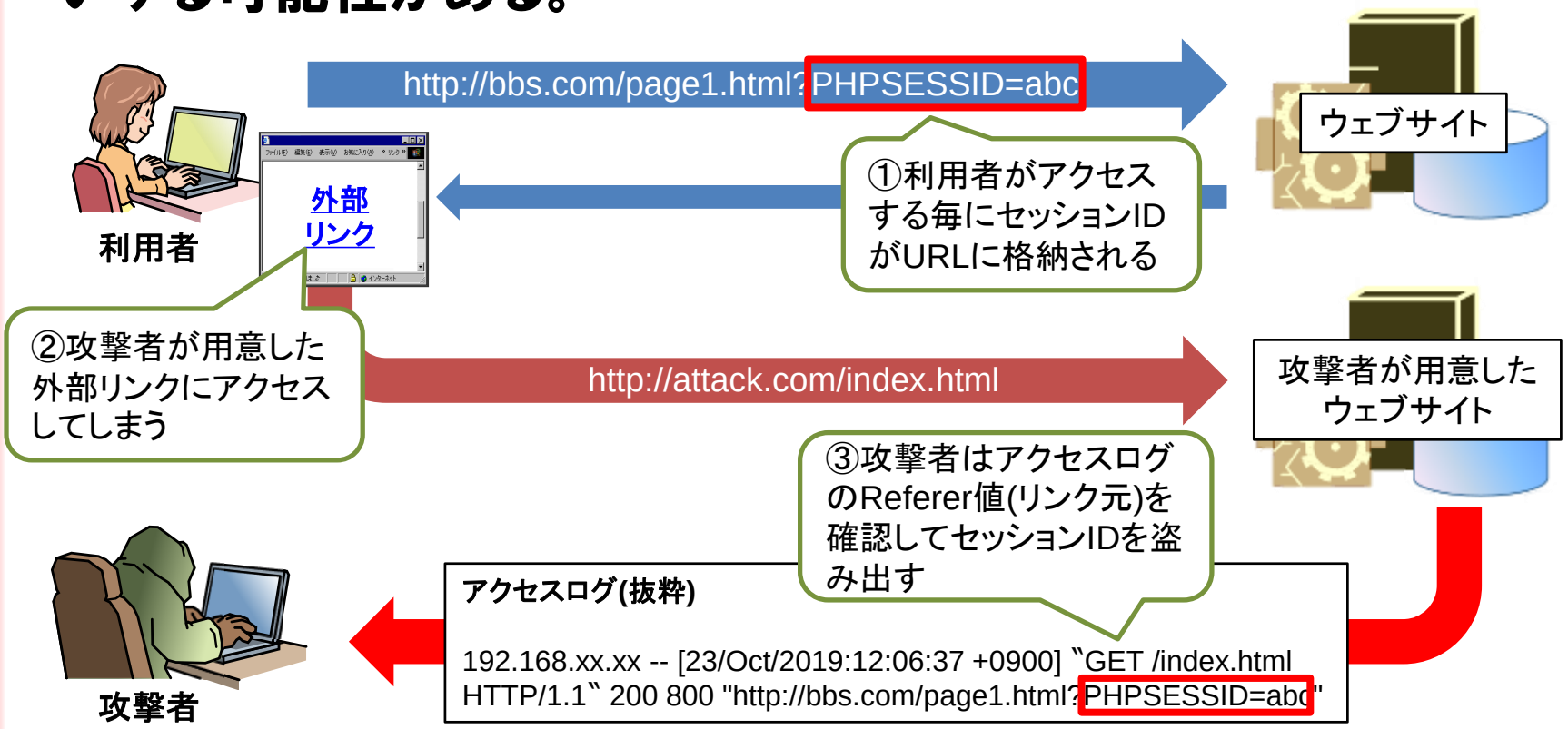
```
POST /ipa/index.html;sessionid=3M90L2 HTTP/1.1
Host: www.ipa.go.jp
Referer: http://www.ipa.go.jp/top.html
Cookie: sessionid=3M90L2 ❷
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 71

loginpass=&sessionid=3M90L2 ❸
```

セッション管理の不備のタイプ

IPA

セッションIDの漏えい

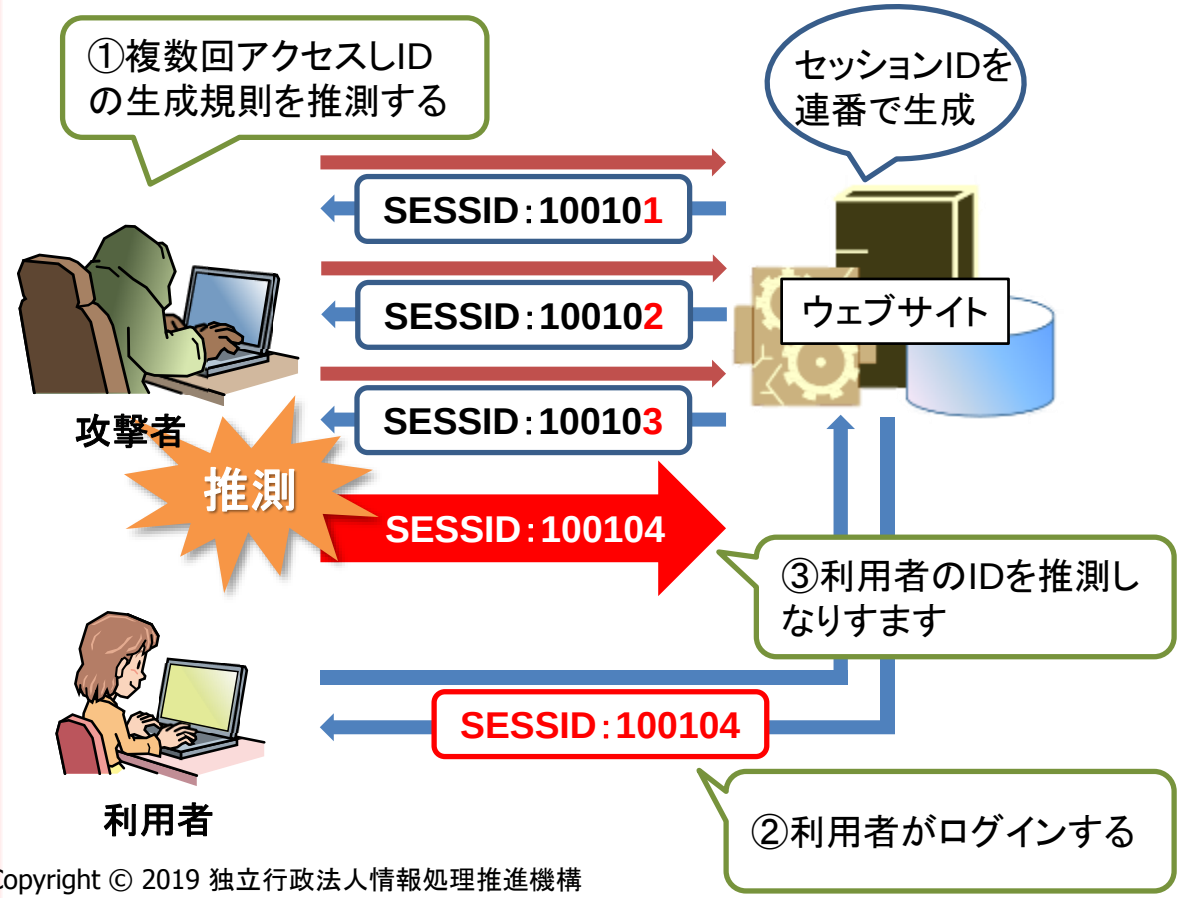
セッションIDをURLに埋め込んでいたり、セキュリティ設定に不備がある場合、通信を盗聴されるなどしてセッションIDが漏えいする可能性がある。



セッション管理の不備のタイプ

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。



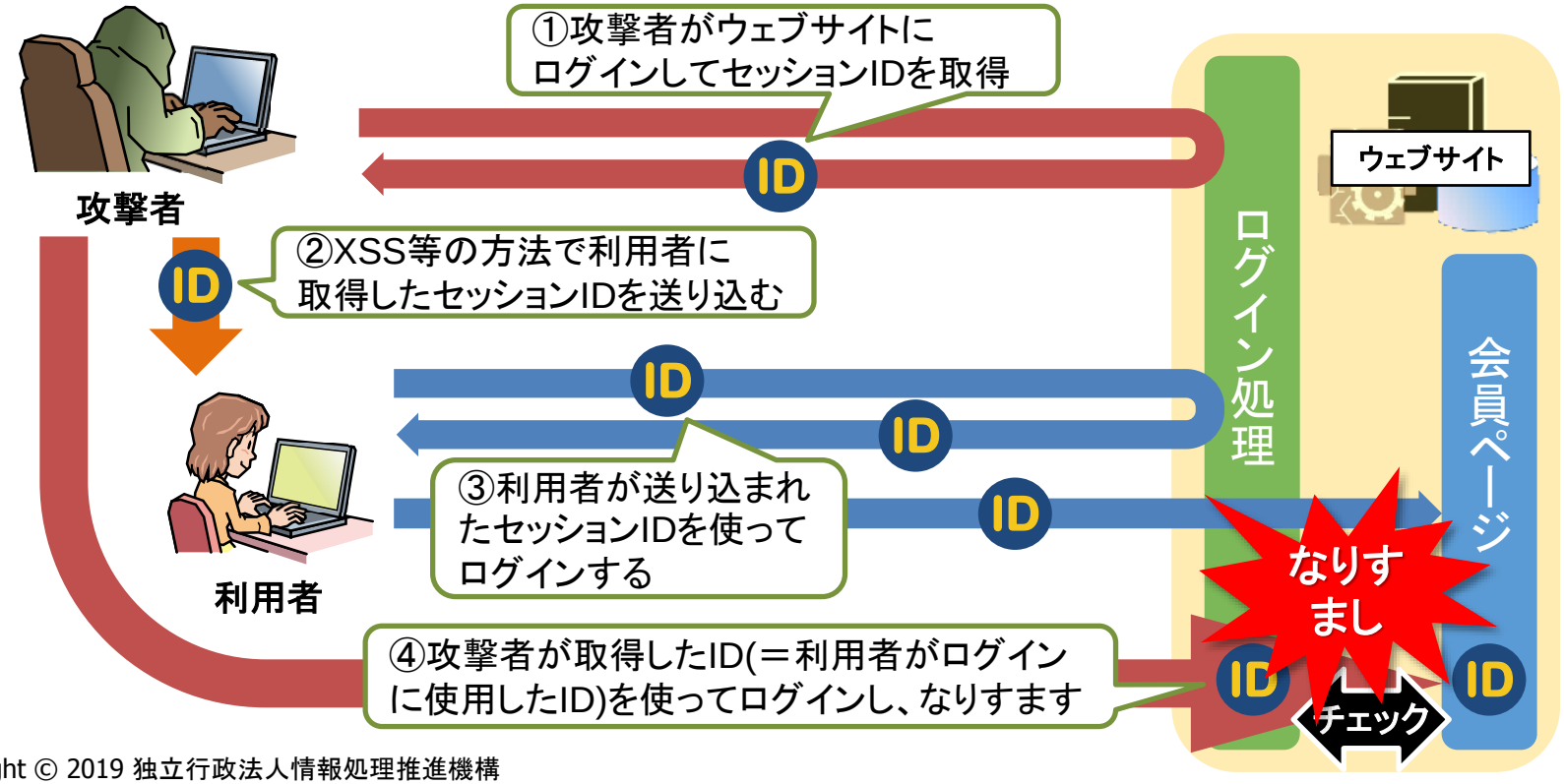
好ましくないセッションIDの生成方法

- 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

セッション管理の不備のタイプ

● 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

IPA

AppGoat
～突いてみますか？脆弱性！～

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

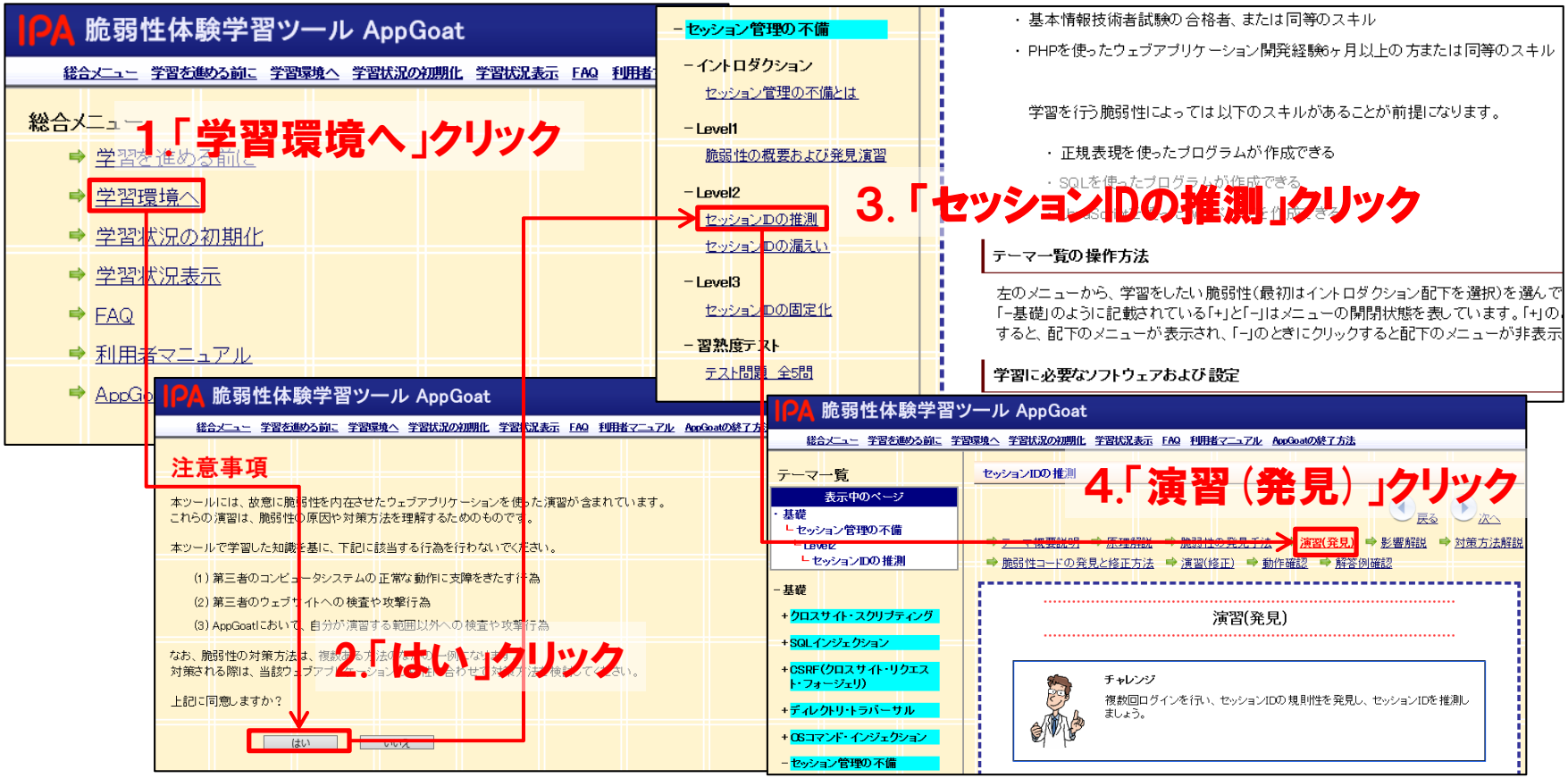
- セッションIDの**生成規則が単純**
  → セッションIDの推測
- ログインの前後でセッションIDが**更新されない**
  →セッションIDの固定化

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

①AppGoatを起動します

②以下の遷移で演習画面に移動します



[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。

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

IPA

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。

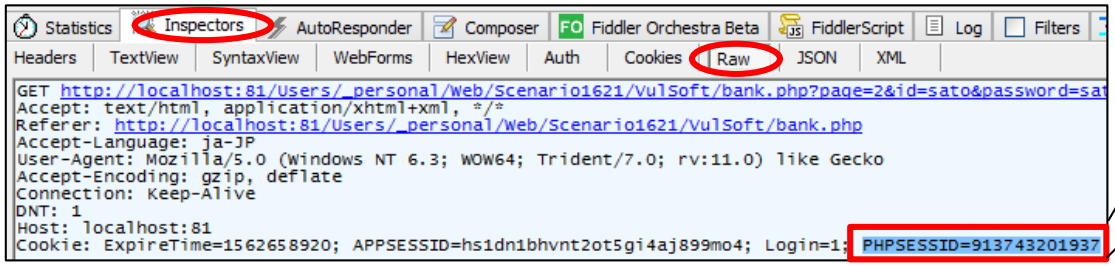
[手順1]  セッションIDを複数回取得する

IPA AppGoat ～突いてみますか？脆弱性！～

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。



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= 913743201937

PHPSESSID= 913744201906

PHPSESSID= 913744201941

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

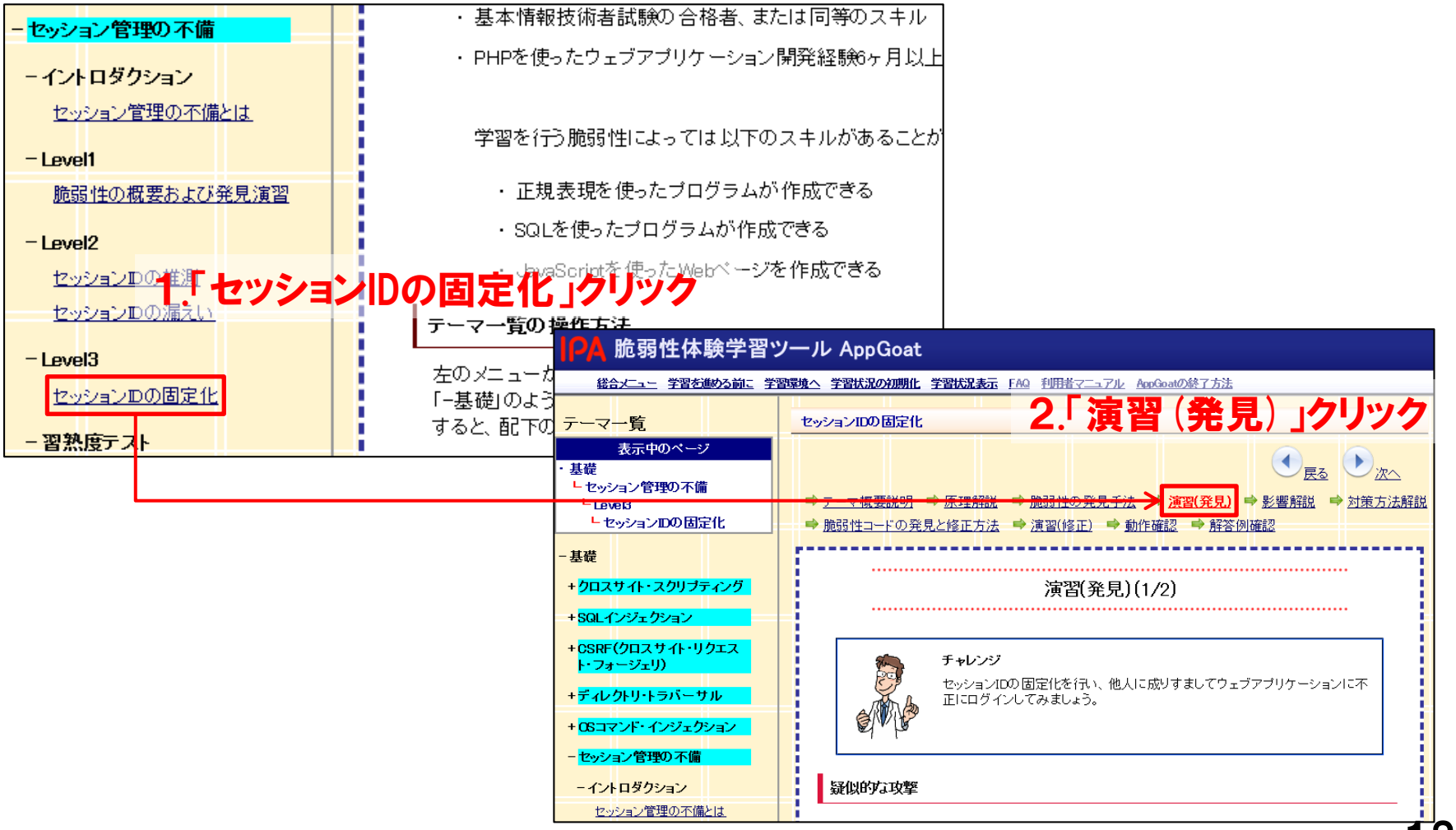
日＋時＋月＋分＋年＋秒でセッションIDが生成されていた。（2019/7/9 13:4X:XX）

内容（演習2）

- 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



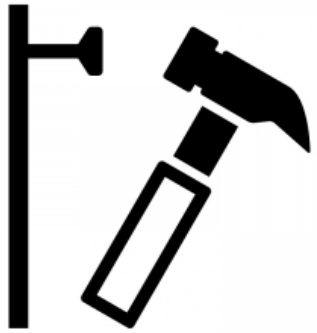
[演習] AppGoatを用いた疑似攻撃体験

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。

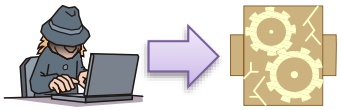
ネット証券に内在する脆弱性を発見しました。

[演習] 演習の進め方

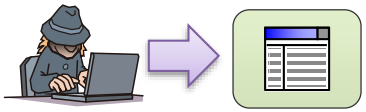
IPA

AppGoat
～突いてみますか？脆弱性！～

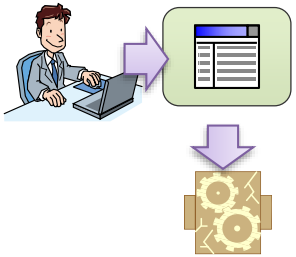
- Step1：脆弱性を発見する



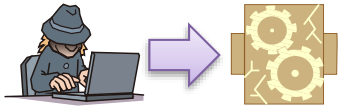
  - ログインしてセッションIDを取得し、脆弱性があることを確認する。
- Step2：罠リンクを設置する



  - クロスサイトスクリプティングの脆弱性を利用し、外部サイトに罠リンクを設置する。
- Step3：罠リンク経由でウェブサイトにアクセスする



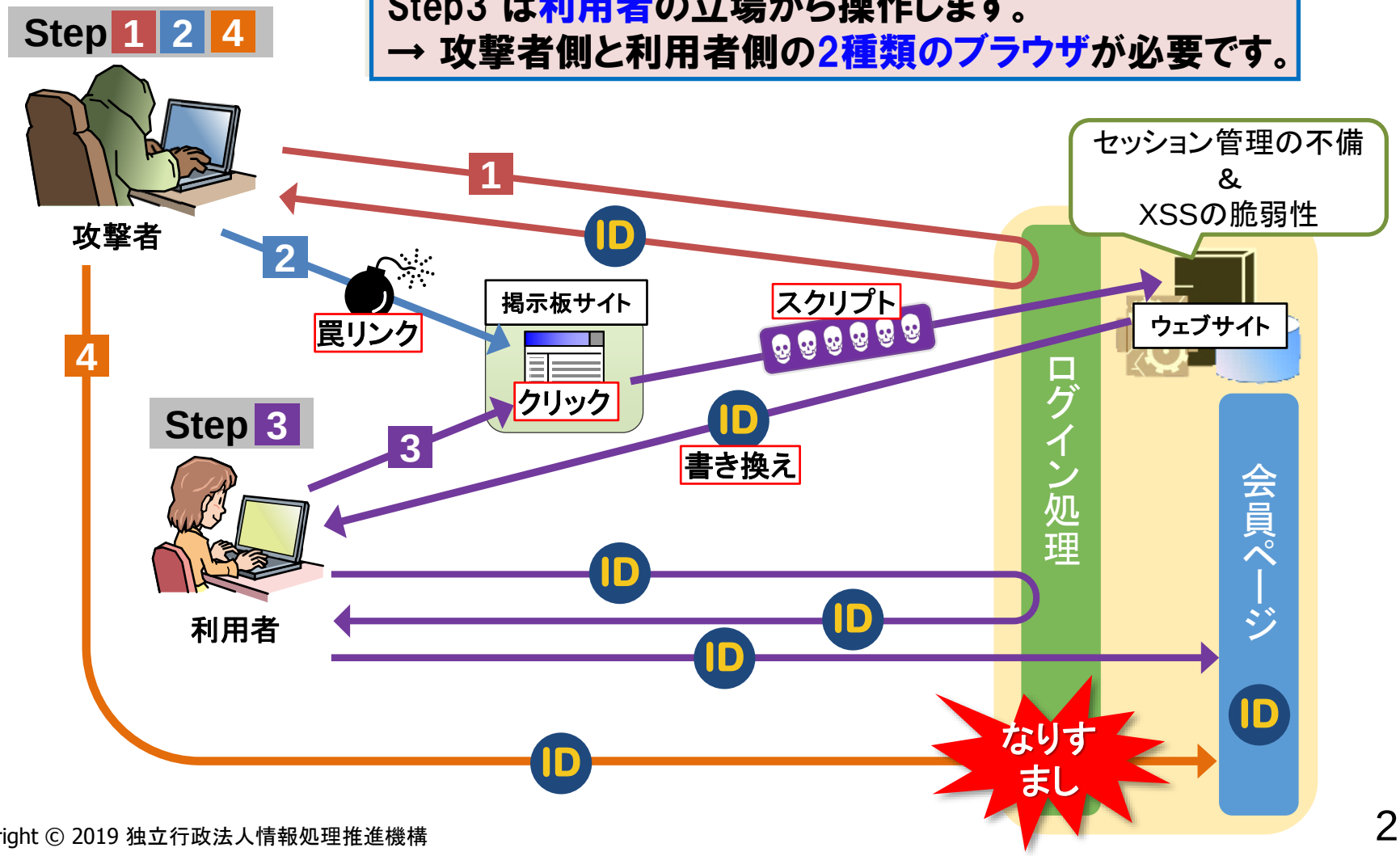
  - 被害者の立場になって罠リンクにアクセスし、セッションIDが書き換えられていることを確認する。
- Step4：再度ログインし、利用者になりすます



  - 最初に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変わっていることを確認する。

[演習] 演習のイメージ

Step1,2,4 は攻撃者の立場から、
Step3 は利用者の立場から操作します。
→ 攻撃者側と利用者側の2種類のブラウザが必要です。



演習

IPA

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。

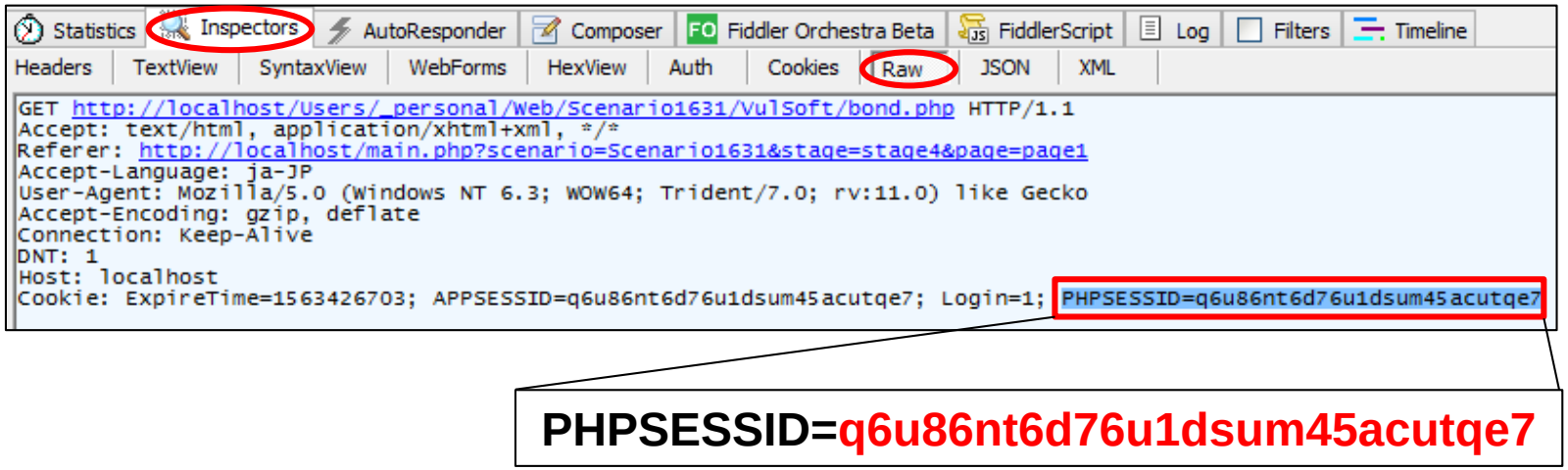
[Step1] セッションIDを取得する（攻撃者）

IPA AppGoat ～突いてみますか？脆弱性！～

演習の手順

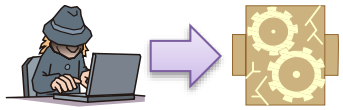
セッションIDを取得し、セッション管理の不備がないか確認しましょう。

- 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。



- 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。

[Step1] 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。

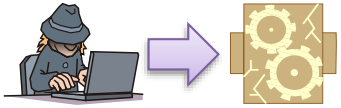
ログイン前と後のセッションIDが同じ
→セッションの固定化ができてしまう

[演習] 演習の進め方

IPA

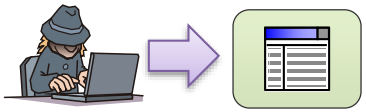
AppGoat ～突いてみますか？脆弱性！～

- Step1：脆弱性を発見する



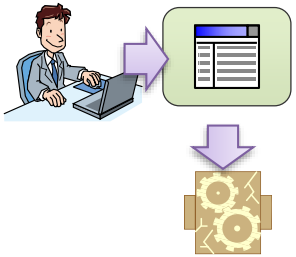
  - ログインしてセッションIDを取得し、脆弱性があることを確認する。

- Step2：罠リンクを設置する



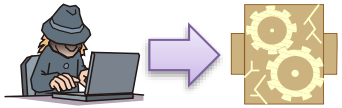
  - クロスサイトスクリプティングの脆弱性を利用し、外部サイトに罠リンクを設置する。

- Step3：罠リンク経由でウェブサイトにアクセスする

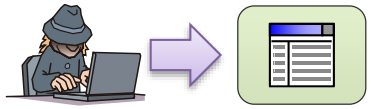


  - 被害者の立場になって罠リンクにアクセスし、セッションIDが書き換えられていることを確認する。

- Step4：再度ログインし、利用者になりすます



  - 最初に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変わっていることを確認する。

[Step2] 
罠リンクを設置する（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を書き換える罠リンクを設置しましょう。

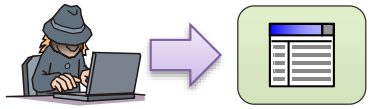
- ネット証券アプリケーションの「お問い合わせ」ページの「ご連絡先メールアドレス」欄にクロスサイトスクリプティングの脆弱性が存在します。攻撃方法は「クロスサイトスクリプティング編」を参照してください。

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>

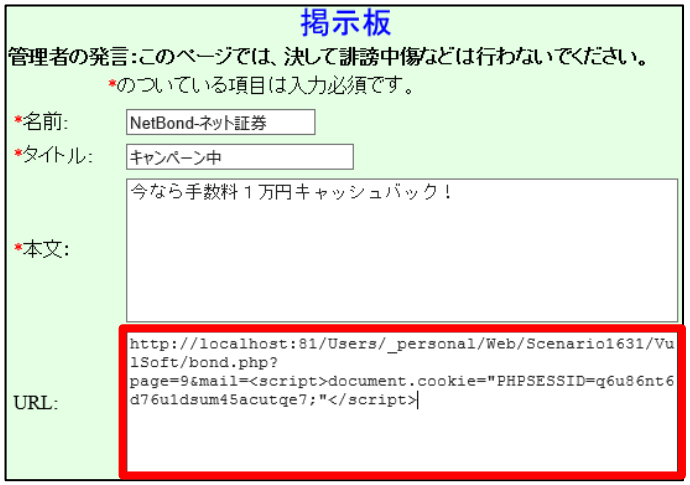
[Step2]  罠リンクを設置する（攻撃者）

IPA AppGoat ～突いてみますか？脆弱性！～

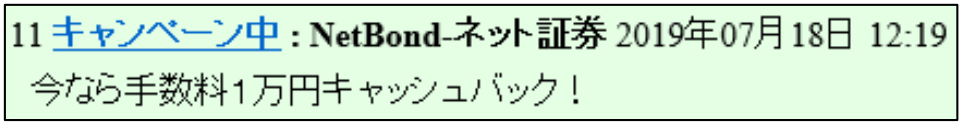
演習の手順

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

2. 掲示板サイトにアクセスし、罠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。



3. 投稿するとリンクを設置することができます。



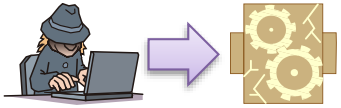
4. ネット証券アプリケーションに戻り、一旦ログアウトします。

[演習] 演習の進め方

IPA

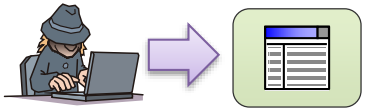
AppGoat
～突いてみますか？脆弱性！～

- Step1：脆弱性を発見する



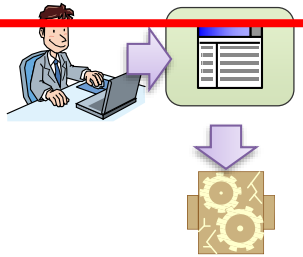
  - ログインしてセッションIDを取得し、脆弱性があることを確認する。

- Step2：罠リンクを設置する



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

- Step3：罠リンク経由でウェブサイトにアクセスする



  - 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、利用者になりすます

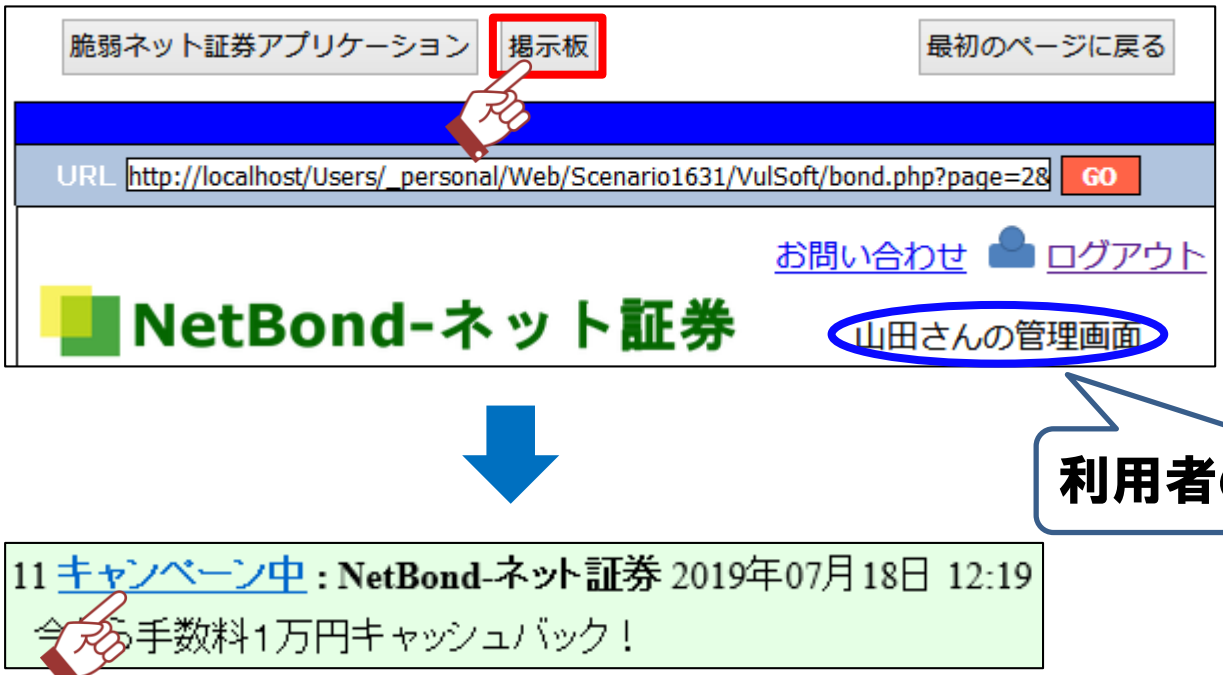
  - 最初に取得したセッションIDでログインすることで、ユーザ名や口座番号が被害者のものに変わっていることを確認する。

[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。

- 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、掲示板サイトで罠リンクにアクセスします。



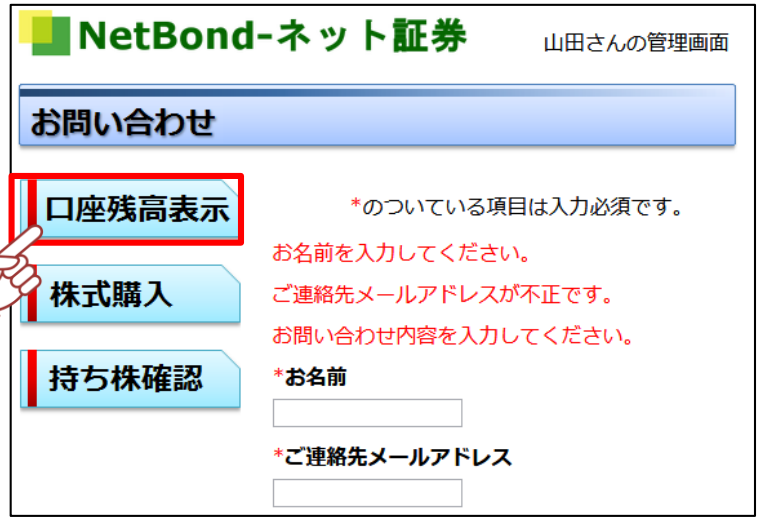
[Step3] 罠リンクにアクセスする（被害者）

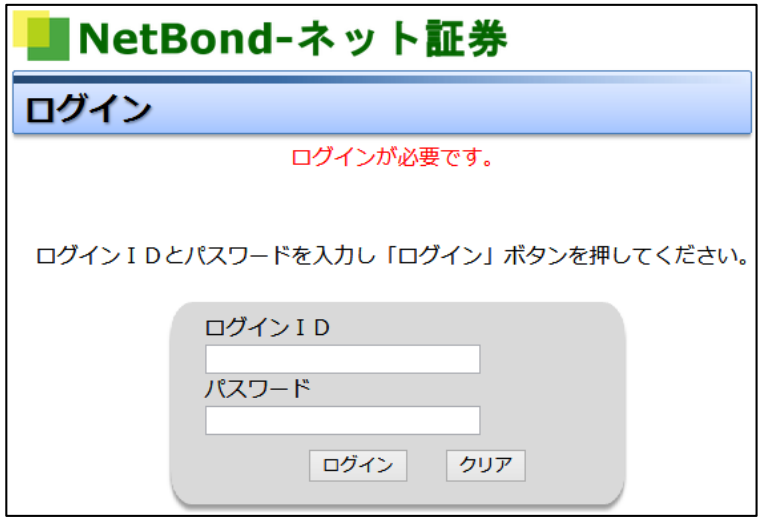
IPA AppGoat ～突いてみますか？脆弱性！～

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と再度ログインを求められます。





[Step3] 罠リンクにアクセスする（被害者）

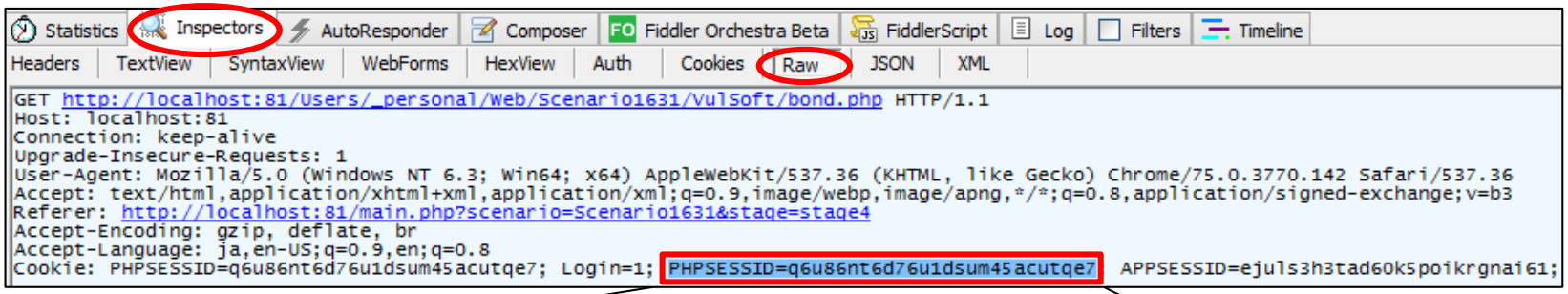
IPA

AppGoat ～突いてみますか？脆弱性！～

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。

- 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。



PHPSESSID=q6u86nt6d76u1dsum45acutqe7

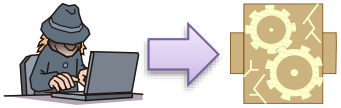
セッションIDの更新がされない仕様のため、攻撃者が取得したIDと同じ
→セッションIDの固定化

[演習] 演習の進め方

IPA

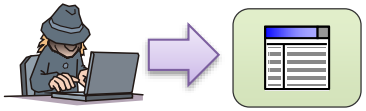
AppGoat ～突いてみますか？脆弱性！～

- Step1：脆弱性を発見する



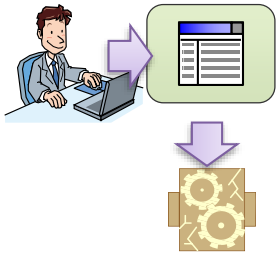
  - ログインしてセッションIDを取得し、脆弱性があることを確認する。

- Step2：罠リンクを設置する



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

- Step3：罠リンク経由でウェブサイトにアクセスする



  - 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、利用者になりすます
  - 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変わっていることを確認する。

[Step4] 利用者になりすます（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。）

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。

対策方法の解説

セッション管理の不備の対策

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。

セッション管理の不備の対策

IPA

- 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

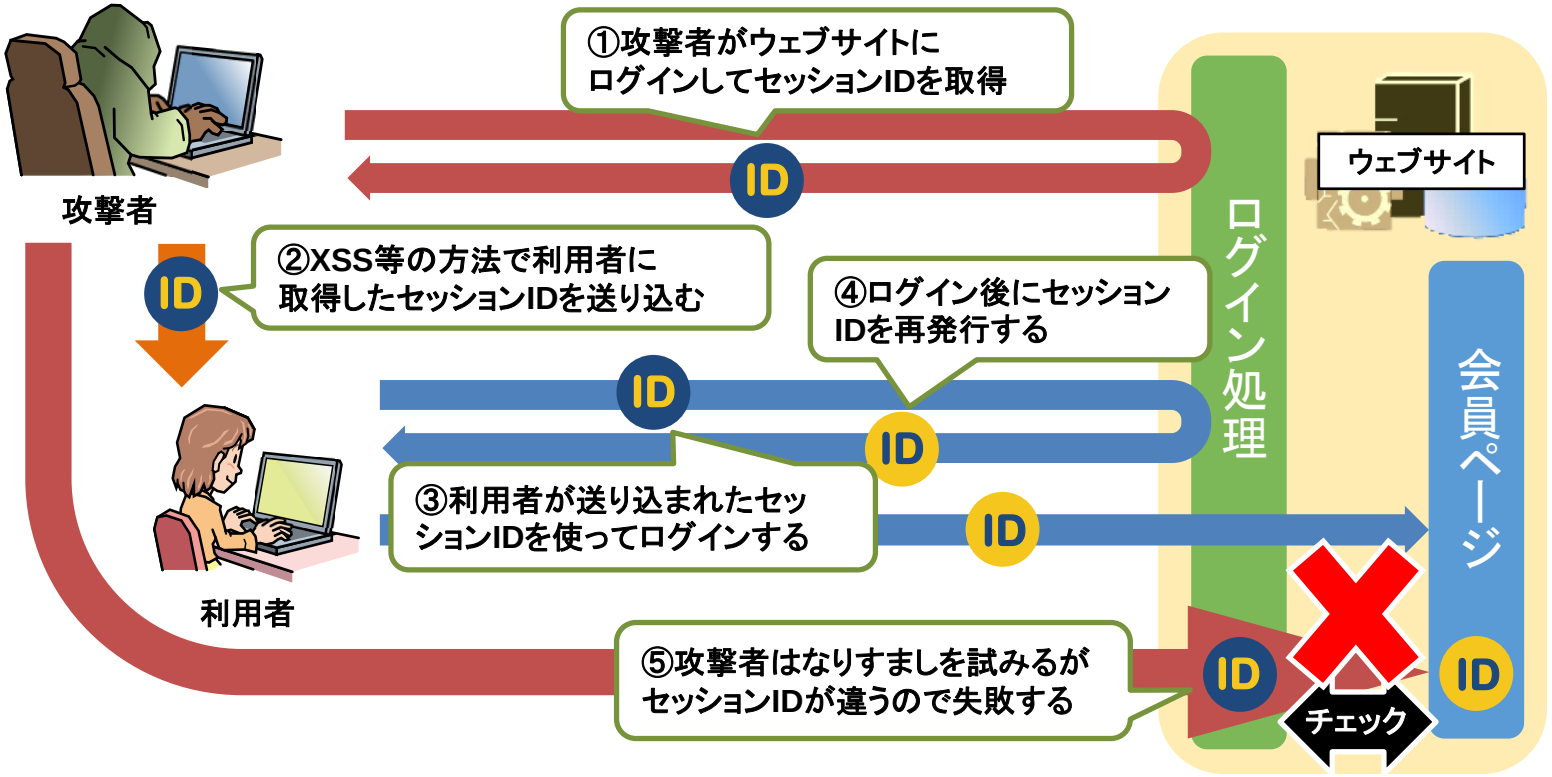
セッションIDの更新ができない場合は、セッションIDとは別に、
ログイン成功時に推測困難な秘密情報(しるし)を作成して
Cookieに格納する。

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。