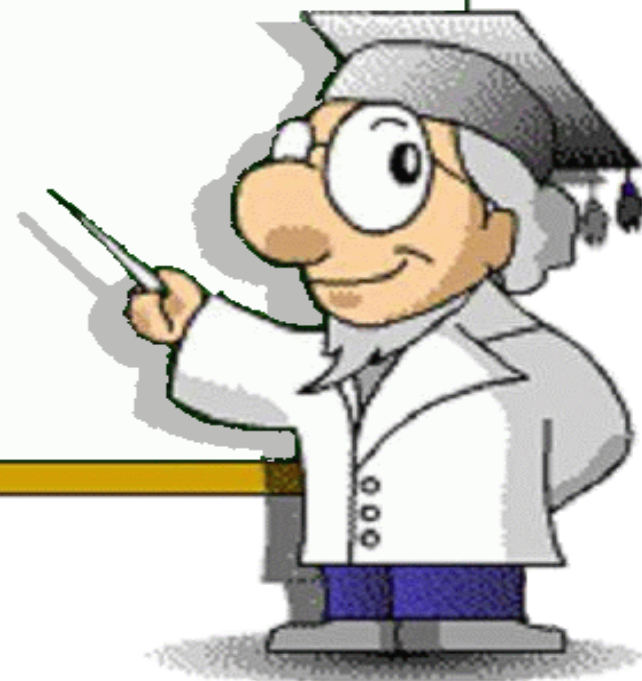


AppGoatを利用した集合教育補助資料 -クロスサイトリクエストフォージェリ編-

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

- 脆弱性の原理解説・基礎知識
- 脆弱性の発見方法
- 演習1:意図しない命令の実行
- 演習解説



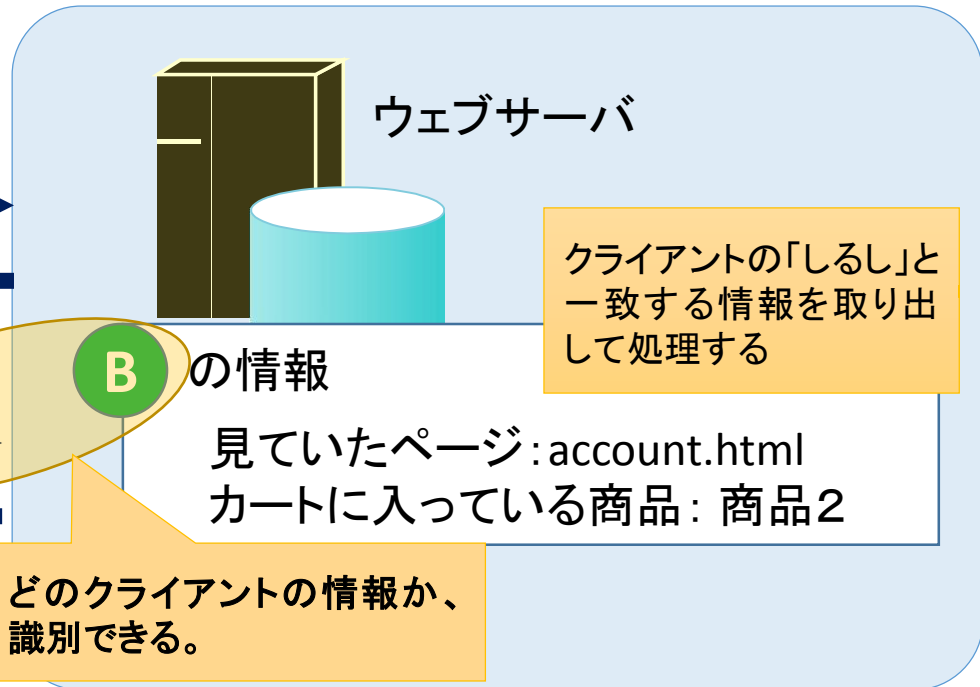
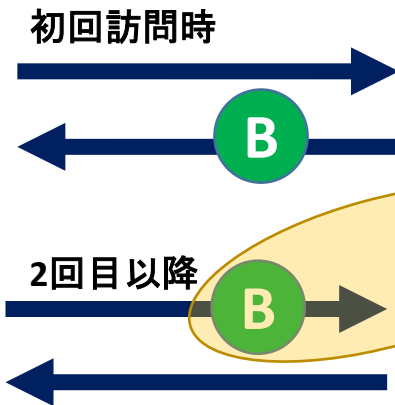
クロスサイト・リクエスト・フォージェリ (CSRF) とは？

- **CSRF (Cross Site Request Forgeries) = サイトを横断してリクエストを偽造**
- **セッションIDを悪用し、ウェブサイトにログインしている利用者に意図しない操作をさせる攻撃**
- **わかりやすく言うと、通常ログイン者しか実行できない操作を、攻撃者が不正に操作する攻撃**
- **過去にはCSRFを悪用された可能性がある事件も**
 - ✓ **横浜市のサイトに犯行声明が投稿され、大学生が誤認逮捕されてしまった事件 (2012年6月)**

セッションとは？

- セッション：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：クライアントを識別するためのしるし

クライアントの「しるし」が発行されている場合



この一連の流れが「セッション」です

その操作が本人かどうかの確認に利用される

セッション ID をウェブサイトへ運ぶ方法は？

● セッション ID を運ぶ方法は以下の3つ

1. Cookie

2. POST メソッド

3. GET メソッド

HTTP リクエスト

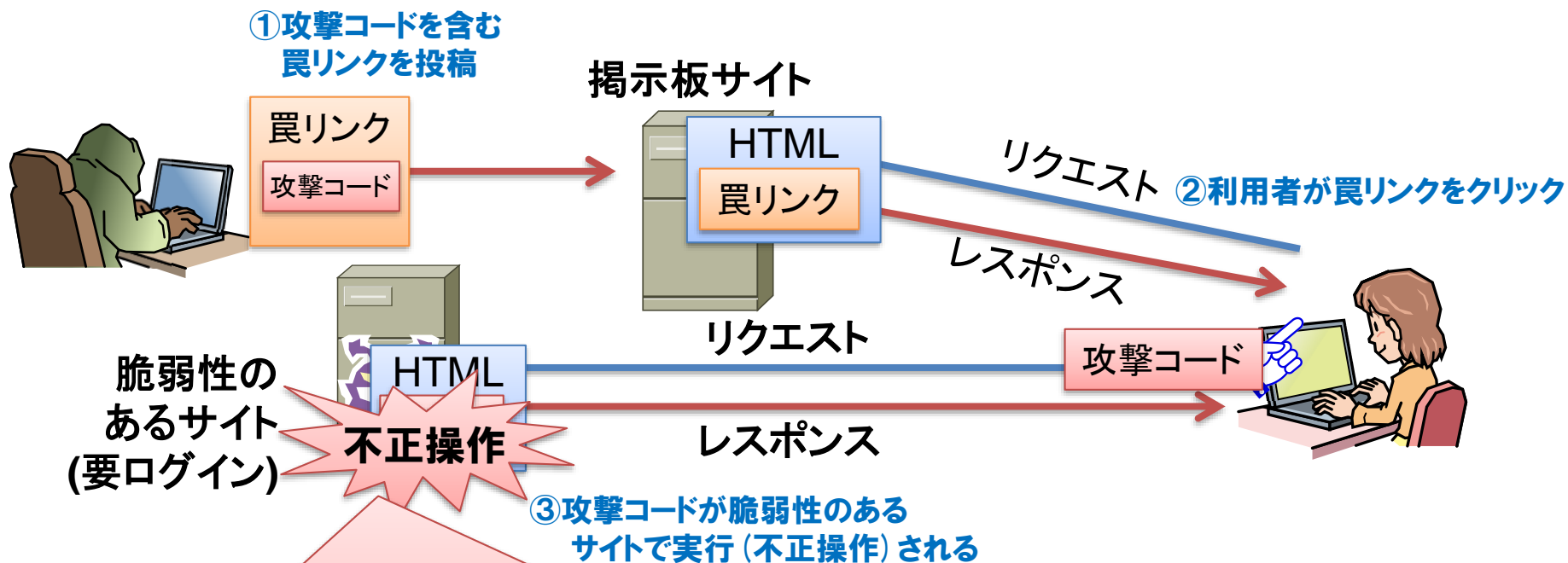
```
POST /ipa/index.html
HTTP/1.1
Host: www.ipa.go.jp
Referer: http://www.ipa.go.jp/top.html
Cookie:
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 71

loginpass=8
```

The diagram illustrates an HTTP request with the following fields: POST /ipa/index.html, HTTP/1.1, Host: www.ipa.go.jp, Referer: http://www.ipa.go.jp/top.html, Cookie: , Connection: keep-alive, Content-Type: application/x-www-form-urlencoded, Content-Length: 71, and loginpass=8. Three red boxes highlight the session ID 'sessionid=3M90L2' in the URL, the Cookie field, and the query string. Red lines connect these boxes to the numbered list items on the left: '1. Cookie' points to the Cookie field, '2. POST メソッド' points to the URL, and '3. GET メソッド' points to the query string.

GETメソッドは暗号化できないURL部分にセッションIDを保持するため、盗聴された場合対策できない⇒非推奨の方法

● ログインが必要なサイトで不正に操作



- ・セッションIDが発行済み (ログイン済み) の場合不正操作
- ・セッションIDが未発行 (未ログイン) の場合処理されない

ウェブアプリケーション側でリクエストを識別する仕組みがないと [セッション切断時]

通常時

① 攻撃者が掲示板に罠リンクを設置する

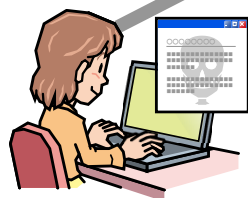


攻撃者

② サーバBにログインしていない利用者が、攻撃者の設置した罠リンクをクリックする

リンク先=ウェブサーバB/PW変更画面
パラメータ=aaaaa

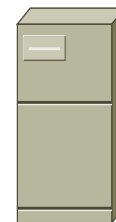
③ 利用者Aが罠のリンクをクリックすると、サーバBに勝手にパスワード変更のリクエストを送ってしまう



ログインしていない利用者A

パラメータ=aaaaa

未ログイン



ウェブサーバB

④

未ログインで、直接パスワード変更画面にアクセスした為、アクセス拒否

セッション切断している為、パスワード変更できず

ウェブアプリケーション側でリクエストを識別する仕組みがないと

攻撃時

① 攻撃者が掲示板に
罾リンクを設置する

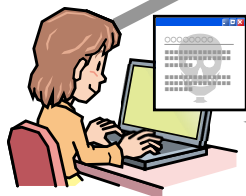


攻撃者

② サーバBにログイン済み
の利用者が、攻撃者の
設置した罾リンクをク
リックする

リンク先=ウェブサーバB/PW変更画面
パラメータ=aaaaa

③ 利用者Aが罾のリンクをクリックすると、サーバBに
勝手にパスワード変更のリクエストを送ってしまう



ログイン済みの
利用者A

パラメータ=aaaaa

⑤ サーバが、パスワード変更のリクエストを受け付け
た旨の応答を返す

④

利用者Aからパスワード変更依
頼があったと判断し、パスワード
を **aaaaa** に変更する

ログイン中

ウェブサーバB

利用者の意図に反して、設定を変更されてしまう

クロスサイトリクエストフォージェリを 発見するために

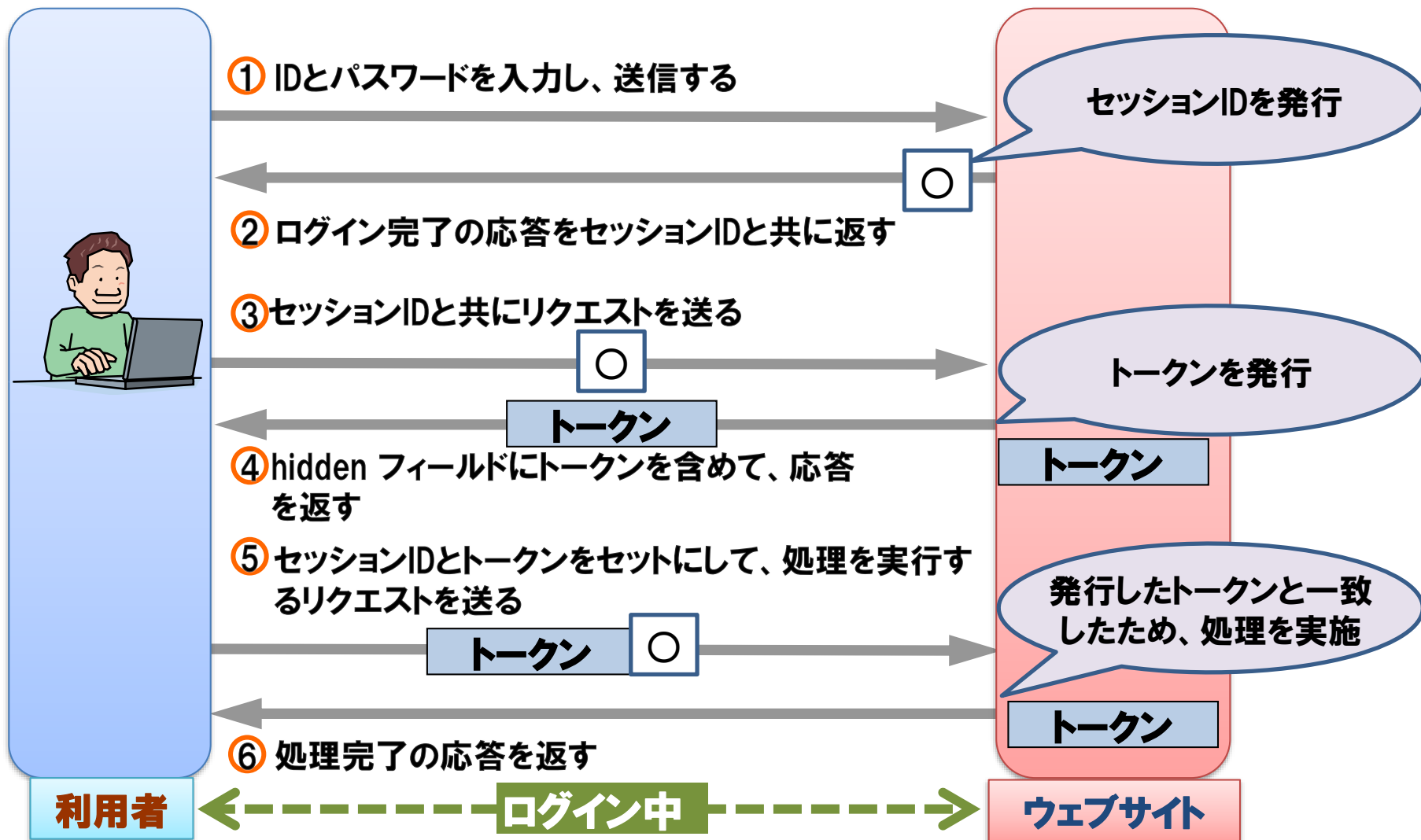
- トークンの有無を確認する
- トークンって何？

- ✓ 第三者が知りえない秘密情報
- ✓ 登録ページや注文ページなどで発行し、利用者が意図したリクエストかを判別する



- ✓ トークンはhiddenフィールドに埋め込んで送信する
 - hiddenフィールド: 秘密情報などを格納するために使われる

トークンの基礎知識



[演習] AppGoatの準備

IPA

AppGoat

～突いてみますか？脆弱性！～

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. 「実習環境へ」クリック

2. 「はい」クリック

3. 「意図しない命令の実行」クリック

4. 「演習(発見)」クリック

**5. ID:sato
パスワード:sato123**

ログインID: sato
パスワード:

ログイン クリア

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。
(1) 第三者のコンピュータシステムの正常な動作に支障を及ぼす行為
(2) 第三者のウェブサイトへの検査や攻撃行為
(3) 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、複数ある方法のなかの一例になります。対策される際は、当該ウェブアプリケーションの特性に合わせて対策方法を検討してください。
上記に同意しますか？

意図しない命令の実行
このテーマでは、【脆弱SNSアプリケーション】を使った演習を通して、CSRF(クロスサイトリクエストフォージェリ)の脆弱性(せいじゅくせい)を学習しましょう。
この脆弱性は、ログインしたユーザーからのリクエストについて、そのユーザーが意図したリクエストであるかどうかを判断する仕組みを持たないウェブアプリケーションに存在する脆弱性です。クロスサイトリクエストフォージェリの脆弱性が含まれていると、意図しないサービスの利用(たとえば、ショッピングサイトで商品を購入)をさせられてしまうなどの問題を引き起こします。

[演習] AppGoatを用いた疑似攻撃体験

IPA



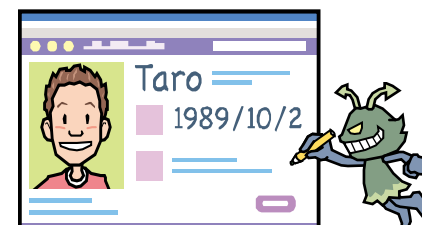
- 演習テーマ:
「意図しない命令の実行」

- ミッション:

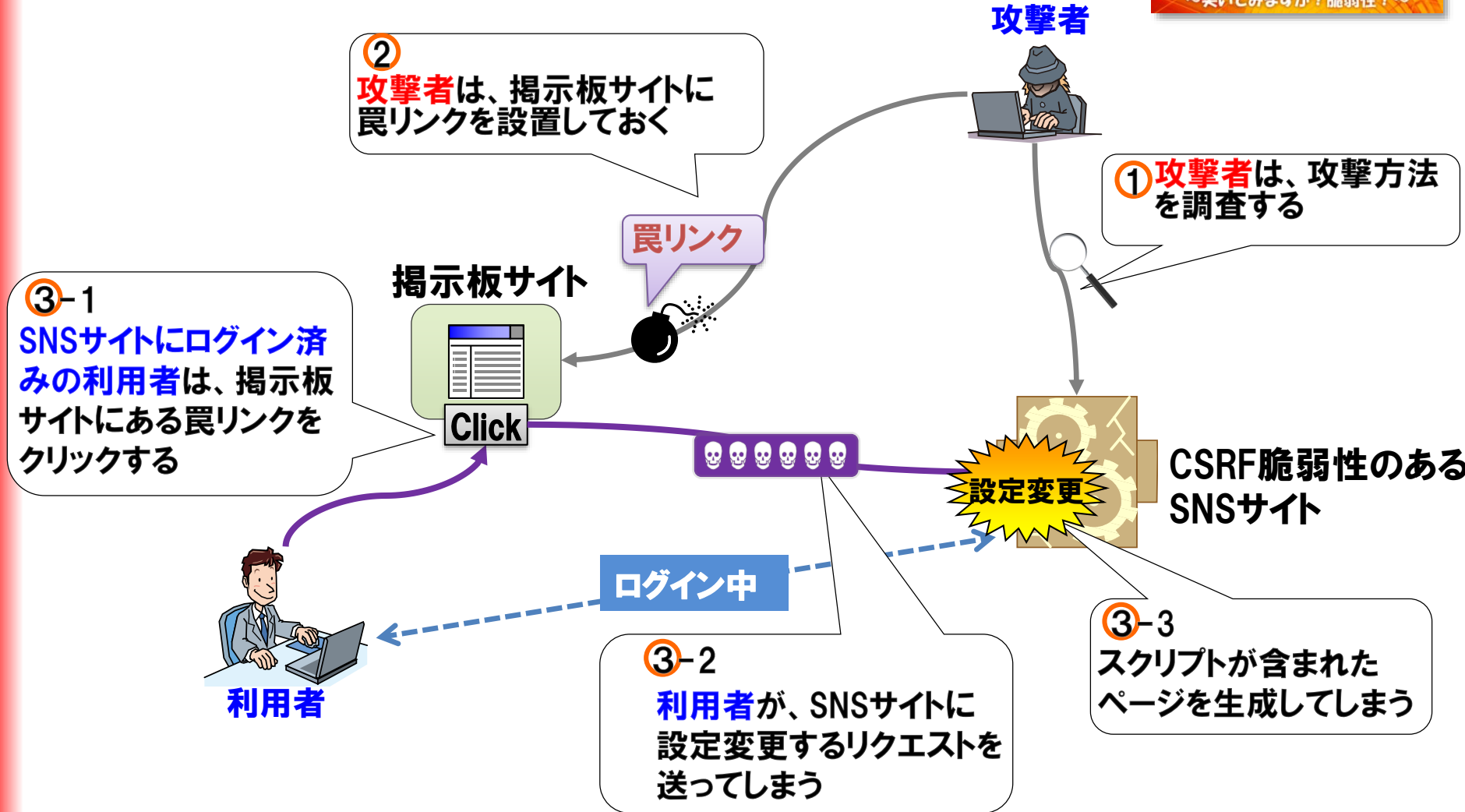
他人のSNSの設定を変更してみましょう



Congratulations!!演習の目標を達成しました。



[演習] 疑似攻撃のイメージ



[演習] 演習の進め方

■ Step1: 攻撃準備を行う

- ・SNS (satoさん) にログインし、設定変更(公開設定)を行う際のリクエストを確認する (※特殊な演習のため、トークンの確認は不要です)



SNSサイト

ホーム 設定変更 ログアウト

yamadaさんの個人情報設定

名前

生年月日 1990年 1月 1日

メールアドレス

個人情報公開
 公開する
 非公開にする

URL `me=yamada&year=1990&month=1&day=1&mail=yamada%40example.com&public=1`

SNSサイト

個人情報設定を変更しました。

[個人情報公開設定へ戻る](#)

リクエスト

■ Step2: 掲示板にアクセスし、設定変更を行うリンクを設置する

- ・設定変更(公開設定)を行うリンクを掲示板に投稿する
※分からない場合は、ヒントを参照してください。



管理者の発言: このページでは、決して建設中欄などは行わないでください。
*のついてる項目は入力必須です。

掲示板

名前:

タイトル:

本文:

URL:

■ Step3: 攻撃を確認してみる



- ・SNS (yamadaさん) にログインした状態で、掲示板のリンクをクリックしてみる

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]



どのようなリクエストを送っているか確認する

IPA

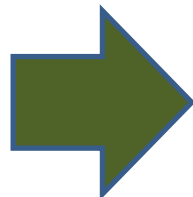


- 設定変更する際にどのようなリクエストを送信しているのか確認しましょう。

SNSサイト

Click

[ホーム](#) [設定変更](#) [ログアウト](#)



satoさんの個人情報設定

名前

生年月日 年 月 日

メールアドレス

公開する

非公開にする

公開設定に変更した場合に、
どのようなリクエストが送信されるのかを確認する。

個人情報公開

送信されるリクエスト

```
http://IPアドレス/Users/ログインID/Web/Scenario1321/VulSoft/sns.php?
page=4&secret_token=b5c3181f0f833ba46f27c7fa0e1faee089344ba797e
e580b5bff33ff68327519&name=yamada&year=1990&month=1&day=1&mail
=yamada@example.com&public=1
```

「個人情報公開」の設定を変更するためのパラメータ

補足: secret_tokenパラメータがあるため、トークンが存在するのようになりますが、演習のために無効化しています。
トークンが存在しない(脆弱性が存在する)前提での演習となります。

[演習] 演習の進め方



■ Step1: 攻撃準備を行う

- ・SNS (satoさん) にログインし、設定変更(公開設定)を行う際のリクエストを確認する

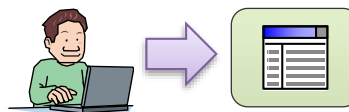


■ Step2: 掲示板にアクセスし、設定変更を行うリンクを設置する

- ・設定変更(公開設定)を行うリンクを掲示板に投稿する
- ※分からない場合は、ヒントを参照してください。



■ Step3: 攻撃を確認してみる



- ・SNS (yamadaさん) にログインした状態で、掲示板のリンクをクリックしてみる

[Step2]



罣リンクのURLを考える

IPA



- SNSサイトのCSRF脆弱性を突いて、「個人情報公開」の設定を「公開する」に変更するURLを考えてみましょう。

1. 前スライドで確認したリクエストから、「個人情報公開」の設定を変更するパラメータを確認します。赤字が該当のパラメータです。

確認したリクエスト

```
http://IPアドレス/Users/ログインID/Web/Scenario1321/VulSoft/sns.php?  
page=4&secret_token=b5c3181f0f833ba46f27c7fa0e1faee089344ba797  
ee580b5bff33ff68327519&name=yamada&year=1990&month=1&day=1&m  
ail=yamada@example.com&public=1
```

2. 必要最低限のパラメータのみ残すと下記のようなリクエストになります。下記のリクエストが送信されるように罣リンクを設置します。

罣リンクのURL

```
http://IPアドレス/Users/ログインID/Web/Scenario1321/VulSoft/sns.php?  
page=4&public=1
```

[Step2]



罣リンクを掲示板サイトに投稿する

IPA

AppGoat

～突いてみますか？脆弱性！～

1. **攻撃者の立場になり**、掲示板に罣のリンクを作成します。罣のリンクには、先ほど作成したURLを入力します。

*名前:

*タイトル:

*本文:

炎上していますよ

URL:

`http://localhost/Users/user01/Web/Scenario1321/VulSoft/sns.php?page=4&public=1`

2. 「投稿」ボタンを押下します。これで罣リンクの設置が完了しました。

11 [あの有名なSNSサイトですが...](#) : IPA次郎 2017年08月30日 15:23

削除

炎上していますよ

[Step3]



設定変更されてしまうことを確認する

IPA



1. (SNSサイトにログインした)利用者の立場になり、罠のリンクをクリックし、SNSサイトにアクセスします。

11 [あの有名なSNSサイトですが...](#) : IPA次郎 2017年08月30日 15:23
炎上していますよ

削除

2. その結果、利用者のSNS設定情報が変更されます。

yamadaさんの個人情報設定

名前	<input type="text" value="yamada"/>
生年月日	1990年 1月 1日
メールアドレス	<input type="text" value="yamada@example.com"/>
個人情報公開	<input checked="" type="radio"/> 公開する <input type="radio"/> 非公開にする

意図せず、個人情報を公開する設定に変更された。

- 演習2:演習3 (グループウェア)
- 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

<p>- 応用</p> <ul style="list-style-type: none">+ 認証制御や認可制御の欠落+ HTTPヘッダ・インジェクション+ バッファオーバーフロー+ クリックジャッキング+ メールヘッダ・インジェクション+ その他の脆弱性(システム情報漏えい等) <p>- 総合</p> <ul style="list-style-type: none">+ 総合問題 <p>- 脆弱性検査</p> <ul style="list-style-type: none">演習(ネット証券)演習(ネットショッピング)演習(グループウェア) <p>+ 補足</p>	<p>学習に必要なソフトウェアおよび設定</p> <ul style="list-style-type: none">・ Adobe Reader(利用者マニュアル用)・ ブラウザのJavaScriptの実行許可設定 <p>パスの表記法</p> <p>一部のテーマの演習ステージでは、ファイルのパスを示すときに、本ツールがC:\IPATool\Y... されたものと仮定して表記します。他のフォルダに展開した場合は、パスを適宜読み替えるよう ください。</p> <p>パスの表記例(環境設定アプリの場合): C:\IPATool\YAppGoatSettings.exe</p> <p>URL、手順、ヒントに記載するURLの表記法</p> <p>一部のテーマの演習・動作確認ステージの脆弱性検査モードではURL、手順、ヒントを示す 以下のように表記します。個人学習者、集合学習者の例を参考に適宜読み替えるようにし てください。</p> <ul style="list-style-type: none">■ 表記パターン http://ホスト名:ポート番号/Users/アカウント名/****■ 読み替え例 個人学習者(IPアドレス192.168.0.1、ポート番号4567の場合) http://192.168.0.1:4567/Users/_personal/****	
--	--	--

1. 「演習3(グループウェア)」をクリック

3. 「休暇申請」をクリック

2.IDに「yamada」、パスワードに「yamada123」と入力しログイン

[演習] AppGoatを用いた疑似攻撃体験

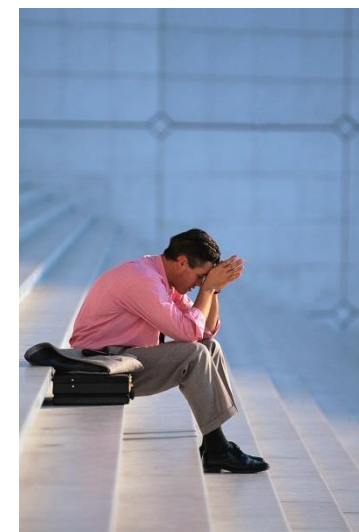
IPA

AppGoat
～突いてみますか？脆弱性！～

- 演習テーマ：
「意図しない命令」

- ミッション：
意図しない休暇申請の操作

Congratulations!!演習の目標を達成しました。



[演習] 演習の進め方

■ Step1:脆弱となる箇所を特定する

- ・リクエスト (URL) にトークンが含まれずに処理を実施できてしまう操作を発見してみましよう



■ Step2:掲示板にアクセスし、登録変更を行うリンクを設置する



■ Step3:攻撃を確認してみる

- ・SNSにログインした状態で、掲示板に設置したリンクにアクセスしてみましよう。

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。



[Step 1]



トークンを送らずに処理ができてしまう操作を確認する

IPA

AppGoat

～突いてみますか？脆弱性！～

演習の手順

処理を実行する際のURLにトークンが含まれているか確認しましょう

- 休暇申請書を作成し、URLを確認してみましょう。

```
http://IPアドレス/Users/ログインID/Web/Scenario3401/VulSoft/  
groupware.php?page=8&token=0197f33f7b071d855608a0e3a92381b  
b144b3cad6a1298e3e6e69e506b8749a1&start_time=2017%2F09%2F  
01&end_time=2017%2F09%2F08&date=6&type=%E5%85%A8%E6%97%A  
5%E4%BC%91%E6%9A%87&reason=%E7%A7%81%E7%94%A8&comment=  
&save=%E7%94%B3%E8%AB%8B%E6%9B%B8%E3%81%AE%E4%BD%9C%E6  
%88%90
```

- 演習用に数個休暇申請書を作成してみましょう。
- 休暇申請を削除するURLを確認してみましょう。

```
http://IPアドレス/Users/ログインID/Web/Scenario3401/VulSoft/  
groupware.php?page=8&holiday_id=1&delete=%E5%89%8A%E9%99%A4
```

トークンが含まれていない

[Step2]



罾リンクを掲示板サイトに投稿する

IPA

AppGoat

～突いてみますか？脆弱性！～

1. **攻撃者の立場になり**、掲示板に罾のリンクを作成します。罾のリンクには、先ほど休暇申請を削除したURLを入力します。

*名前:

*タイトル:

*本文:

URL:

2. 「投稿」ボタンを押下します。これで罾リンクの設置が完了しました。

11 [グループウェアの裏技](#) : IPA次郎 2017年08月30日 16:34

リンクをクリックしてグループウェアの裏技を知ろう

[Step3]



休暇申請が削除されることを確認する

IPA



1. (グループウェアにログインした)利用者の立場になり、罫のリンクをクリックし、グループウェアにアクセスします。

11 [グループウェアの裏技](#) : IPA次郎 2017年08月30日 16:34

削除

リンクをクリックしてグループウェアの裏技を知ろう

2. その結果、**利用者の休暇申請が削除**されてしまいます。

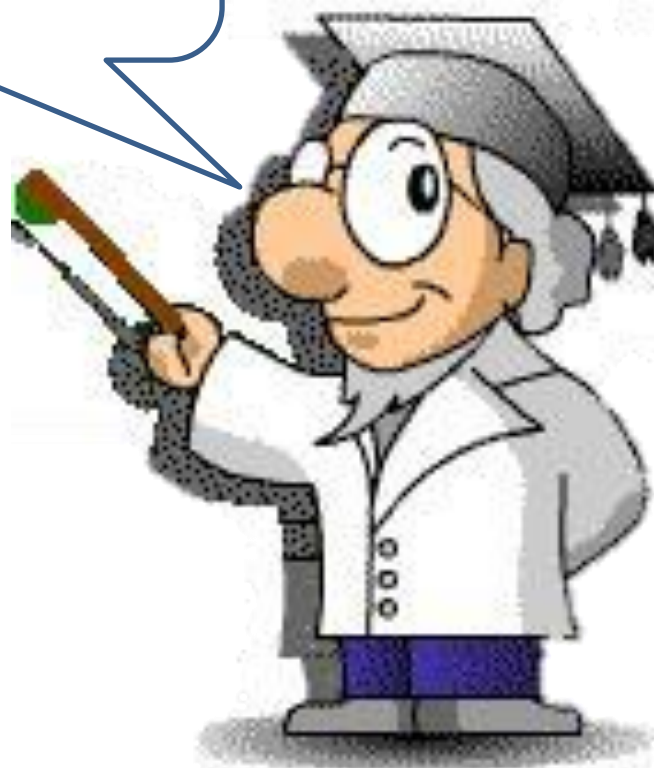
休暇申請書の削除が完了しました。

削除	期間	区分	事由	備考
<input checked="" type="radio"/>	2015/2/23-2015/2/24 1日間	全日休暇	私用	

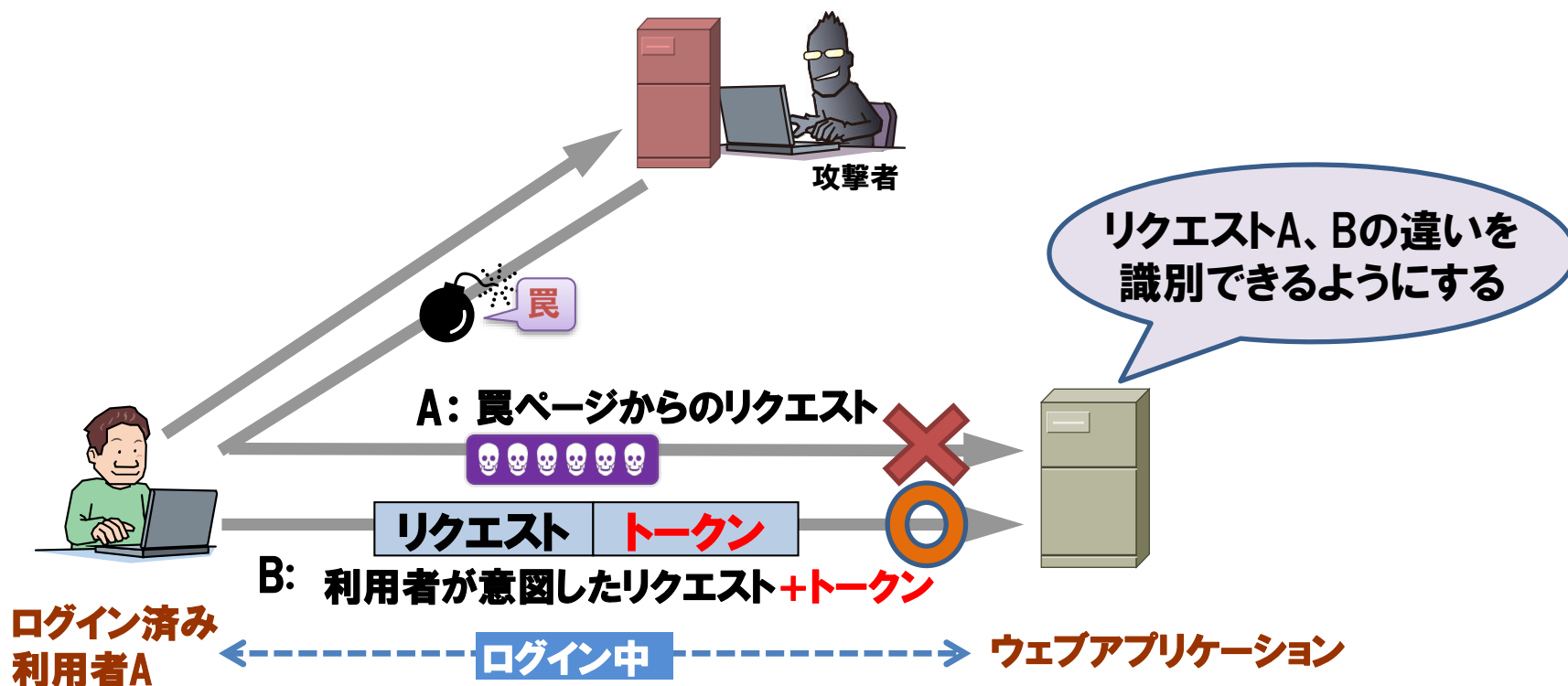
削除

意図せず、休暇申請が削除されてしまった

対策の解説



- 罨ページを経由したリクエストと正規のリクエストを識別できるようにリクエストにトークン (秘密情報) を付加する



● リクエストに推測困難な秘密情報を埋め込む

- 処理を実行する際に、秘密情報を要求し、その値が正しい場合のみに処理を実行する

- ✓ POSTメソッドで送る

⇒ GETメソッドで秘密情報を送付すると、外部サーバのアクセスログに秘密情報がRefererとして漏れる可能性がある

**秘密情報が上記の条件を満たさないと、
対策漏れが生じる可能性がある**

以上で、
クロスサイトリクエストフォージェリの
解説は終了です。

