

Vexトレーニングテキスト



DAST スキャンニング 基礎編

目次

- はじめに …2
- 1. DASTツール使用時のアプローチ …3
- 2. 画面遷移の分析 …5
 - 認証機能（ログイン、ログアウト）
 - データ登録系機能
 - データ削除系機能
 - データ編集系機能
 - キャプチャ認証系機能
- 3. データフローの分析 …13
 - 各画面間で受け渡すデータの分析
 - データ発行箇所のパターン

Version 1.1

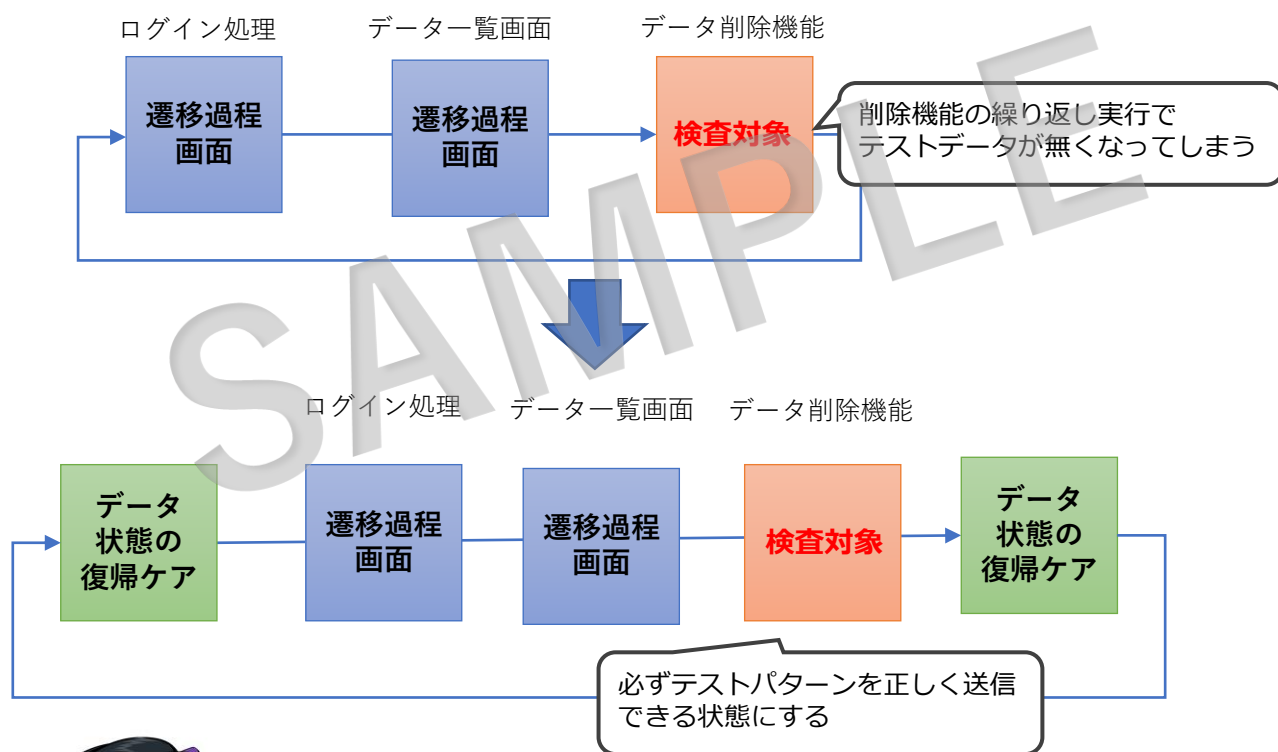
1. DASTツール使用時のアプローチ

VexはDAST(**D**ynamic **A**pplication **S**ecurity **T**esting)に分類されるセキュリティテストを実施します。DASTとは実際に動作しているアプリケーションにテストパターンを送信し、アプリケーションの動作状態を観測するセキュリティテストの手法です。Vexに限らずDASTツール使用時はツール設定の考え方にいくつかのポイントがあります。

1-1. DASTツール使用時の原則

DASTツールは、検査対象画面に大量の検査パターンを送信します。検査パターンを送信するごとに、検査対象画面のデータステータスを同一の状態(テストパターンを送信できる状態)となるよう、テストシナリオ設定をする必要があります。

例：削除機能をテストする場合



繰り返しテストパターンを送信できるように、テストシナリオ
(画面の遷移順)を構成しましょう。

2.画面遷移の分析

DASTのテストシナリオはいくつかのパターンに分類されます。本章では代表的なテストシナリオパターンを解説します。

機能分類		基本パターン
1	認証機能 (ログイン、ログアウト)	・アカウントロック ・重複ログインチェック
2	データ登録系機能	・登録数制限-アカウント単位の制限がある場合 ・登録数制限-対象サイト全体で制限がある場合
3	データ削除系の機能	
4	データ編集系の機能	・繰り返し編集処理が可能な場合 ・繰り返し編集処理が不可能な場合
5	キャプチャ認証系機能	・原則不可 ・特例で検証用環境や実装不備がある場合



基本パターンに沿って、どのような画面遷移を構成すれば
よいか検討してみましょう。

3.データフローの分析

データフローの分析は以下の観点があります。

分析観点	基本パターン
1 各画面に対して引き継ぐデータはどれか	・セキュリティトークン ・データ識別のID ・日付・時刻情報
2 データの発行箇所はどこか	・データが必要となる画面の直前に発行される場合 ・必要なデータが複数画面で発行される場合 ・画面からの引継ぎでなく新たにデータを発行する場合



引き継ぐデータはアプリケーションにより異なり、様々なデータがあります。
アプリケーションを分析して、各データがどのように使われるかを理解しましょう。

Vexトレーニングテキスト



VCA 模擬試験 シナリオ解説編

Version 1.2

目次

1	はじめに	p.4
2	模擬試験サイトについて	p.5
3	セキュリティテスト対象の概要把握	p.6
3-1.	Webアプリケーションの利用目的	p.7
3-2.	Webアプリケーションの全体像	p.8
3-3.	セキュリティテスト対象機能の把握	p.15
3-4.	セキュリティテスト対象関連機能の把握	p.22
4	認証機能のテストシナリオ	p.28
4-1.	アカウントロックの確認方法	p.30
4-2.	アカウントロックの回避方法	p.33
4-3.	ログインのテストシナリオ作成	p.36
4-4.	ログアウトの設定	p.46
5	データ閲覧系機能のテストシナリオ	p.50
5-1.	閲覧機能に対する効率的なテストシナリオ	p.52
5-2.	データ状態が変化した場合の影響	p.61
5-3.	カテゴリ詳細のテストシナリオ作成	p.63
6	データ登録・削除機能のテストシナリオ	p.64
6-1.	アカウント削除機能のテストシナリオ分析	p.67
6-2.	ログインのテストシナリオ作成	p.69
6-3.	アカウント登録のテストシナリオ作成	p.71
6-4.	アカウント削除のテストシナリオ作成	p.83

6-5.	カテゴリ登録のテストシナリオ分析	p.111
6-6.	カテゴリ登録のテストシナリオ作成	p.118
7	データ編集系機能のテストシナリオ	p.126
7-1.	編集対象データの分析	p.128
7-2.	データ干渉を考慮したテストシナリオ作成	p.133
7-3.	パスワード変更機能の分析	p.146
7-4.	パスワード変更機能のテストシナリオ戦略	p.148
7-5.	パスワード変更機能のテストシナリオ作成	p.152
7-6.	ユーザ情報変更の回数制限確認	p.167
7-7.	ユーザ情報変更のテストシナリオ分析	p.168
7-8.	ユーザ情報変更のテストシナリオ作成	p.174
7-9.	セッション変数の考察	p.182
7-10.	セッション変数を利用した機能への対処	p.184
7-11.	ユーザ情報変更のテストシナリオ作成（拡張処理）	p.185
8	CAPTCHA認証系機能のテストシナリオ	p.188
8-1.	ソースコードの分析(ユーザ基本情報変更)	p.190
8-2.	スキャン結果の分析(ユーザ基本情報変更)	p.192
8-3.	ソースコードの分析(パスワードリセット)	p.195
8-4.	スキャン結果の分析(パスワードリセット)	p.201
8-5.	パスワードリセット完了のテストシナリオ作成	p.203

2. 模擬試験サイトについて

模擬試験の検査対象Webアプリケーションは、UBsecurePict社の商品販売管理サイトです。管理サイトには、ログイン機能、ユーザ管理、商品管理など一般的な機能が実装されています。

模擬試験で出題されている検査対象機能は以下の機能となります。出題対象の機能は模擬試験環境上のヒアリングシートより確認可能です。

機能カテゴリ	出題対象となる検査対象機能
1 認証機能	・ ログイン
2 データ閲覧系機能	・ カタログ一覧
3 データ登録系機能	・ 管理者アカウント登録 ・ カテゴリ登録
4 データ削除系の機能	・ 管理者アカウント削除
5 データ編集系の機能	・ 管理者アカウント変更 ・ 管理者パスワード変更 ・ ユーザ情報変更
6 CAPTCHA認証系機能	・ ユーザパスワードリセット ・ ユーザ情報変更



セキュリティ検査を始める前に、検査対象サイトの機能概要を把握しましょう

3. セキュリティテスト対象の概要把握

セキュリティテストを円滑にすすめるためには、セキュリティテスト対象となるWebアプリケーションの機能概要を把握し、Webアプリケーションの仕様に沿ったテストシナリオを作成する必要があります。

本章では、Webアプリケーションの仕様に沿ったテストシナリオを作成するための、Webアプリケーションの分析方法を解説します。

■ 解説概要

効果的・効率的なテストシナリオを作成するためには、Webアプリケーションの利用用途や全体像の把握と、テスト対象機能の動作確認が重要となります。これには、2つの目的があります。

目的1. Webアプリケーションのビジネスリスクを把握する

セキュリティテストの目的は、Webアプリケーションに存在するリスク（脆弱性）を洗い出すことにあります。Webアプリケーションの利用目的や想定利用シーン、取扱情報を含めたアプリケーション全体像を把握することで、Webアプリケーションにおけるビジネスリスクを明確にすることができます。

また、ビジネスリスクが明確になれば、リスクに応じたテストスコープ（テスト対象範囲）の判断が可能となり、効率的にセキュリティテストを実施できるようになります。

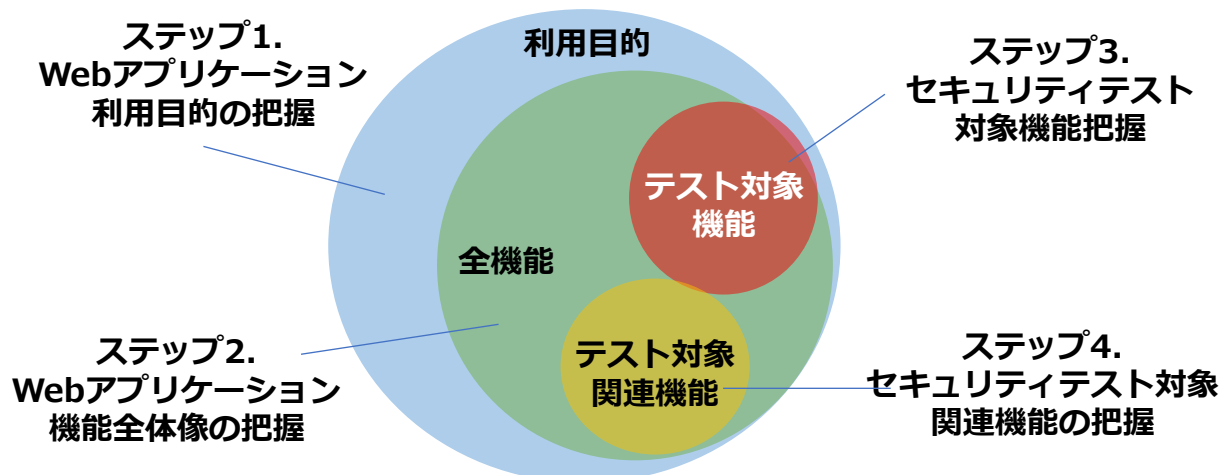
目的2. テスト対象機能の仕様とテストシナリオに利用可能な関連機能を把握する

セキュリティテストでは、連続で安定的にアクセス可能なテストシナリオを作成する必要があります。テストシナリオは、テスト対象機能の仕様に合わせて画面遷移順とデータフロー(データの受け渡し)を設定する必要があるため、以下のような内容を確認する必要があります。

- ・対象機能のアクセスに必要なアクセス権限/アクセス時のデータ要否など
- ・データ量や実行回数の制限、データ作成方法など

なお、テスト対象機能の仕様によっては、テスト対象以外の機能へのアクセスを必要とするケースもあるため、利用可能な関連機能(パーツ)についても把握しておく必要があります。

Webアプリケーションの分析ステップは以下のイメージとなります。



4. 認証機能のテストシナリオ

本章では認証機能のテストシナリオ構築について解説します。模擬試験の出題範囲において認証機能に属する画面は以下の通りです。

Function ID	機能名
MNGAUT0001	ログイン入力
MNGAUT0002	ログイン処理(リダイレクト)

ログイン入力～ログイン処理(リダイレクト)

■ 解説対象機能

Function ID	機能名
MNGAUT0001	ログイン入力
MNGAUT0002	ログイン処理(リダイレクト)

Function IDMNGAUT0001Exam Information

Function Nameログイン入力

UBsecure PICT

検証用データ復帰ページへ



UBsecure PICT

ログインID

パスワード

管理者ログイン

Copyright 2020 UBsecure Inc. All Rights Reserved.

■ 解説概要

ログイン処理に対するテストシナリオを検討します。一般的にログイン処理のテストシナリオにおいて、最も注意すべき点は、認証機能に対するセキュリティ対策機構の実装有無です。セキュリティ対策機構を考慮せずにテストシナリオを構築すると、認証機能がロックされます。（一定時間ログイン試行ができないなどの状態となります。）

認証機能に対する一般的なセキュリティ対策機構の例を以下に示します。

認証機能に対するセキュリティ対策機構	概要
1 アカウントロック（連続認証失敗時）	連続してログインを失敗すると、 ・一定時間ログインができなくなる ・ロックの解除操作が必要となる など
2 アカウントロック（単位時間辺りのログイン試行）	単位時間あたり（例：1分間、10分間など）に 大量のログイン試行を検知すると、 ・一定時間ログインができなくなる ・ロックの解除操作が必要となる など
3 重複ログイン検知	同一アカウントで、既存のログイン中セッションがあると、ログインができない。 実装方法はさまざまあるが、重複ログインを検知した場合は、関連セッションを強制的にログアウトさせる画面が表示されるケースなどがある

模擬試験サイトではアカウントロック（連続認証失敗時）が実装されています。そのため、本章ではアカウントロックを回避するためのテストシナリオの作成方法を解説します。

本章で学ぶ内容

- ✓ アカウントロックを確認する方法
- ✓ アカウントロックを回避するためのテストシナリオ
- ✓ CSRFトークンの設定
- ✓ ログアウト処理の設定

Vexトレーニングテキスト



VCA 模擬試験 トリアージ解説編

Version 1.0

目次

1	はじめに	p.5
2	模擬試験で出題される脆弱性	p.6
3	脆弱性判定の基本アプローチ	p.7
	3-1. 脆弱性内容の把握	p.8
	3-2. 検出箇所の把握	p.9
	3-3. 再現性の確認	p.12
	3-4. 検出根拠の把握	p.13
4	過度な情報漏えい	p.20
	4-1. 脆弱性概要	p.20
	4-2. 確認方法	p.20
	4-3. 検知リスト	p.20
	4-4. 脆弱性判定 (パスワードリセット)	p.21
5	不適切なエラー処理	p.23
	5-1. 脆弱性概要	p.23
	5-2. 確認方法	p.23
	5-3. 検知リスト	p.24
	5-4. 脆弱性判定 (管理者アカウント登録確認/adminemail)	p.25
	5-5. 脆弱性判定 (カタログ一覧/category)	p.27
6	クロスサイトスクリプティング	p.29
	6-1. 脆弱性概要	p.29
	6-2. 確認方法	p.30

6-3.	検知リスト	p.38
6-4.	脆弱性判定 (管理者アカウント登録確認/lastname_kanji)	p.39
6-5.	脆弱性判定 (管理者アカウント登録確認/firstname_kanji)	p.42
6-6.	脆弱性判定 (管理者アカウント削除完了/return_url)	p.51
6-7.	脆弱性判定 (ユーザ住所情報変更/lastname/firstname)	p.55
7	クロスサイトリクエストフォージェリ	p.56
7-1.	脆弱性概要	p.56
7-2.	確認方法	p.57
7-3.	検知リスト	p.58
7-4.	脆弱性判定 (管理者アカウント登録完了)	p.59
7-5.	脆弱性判定 (管理者アカウント削除完了)	p.65
7-6.	脆弱性判定 (管理者アカウント変更完了)	p.68
7-7.	脆弱性判定 (管理者アカウントパスワード変更完了)	p.69
7-8.	脆弱性判定 (カタログ一覧)	p.71
8	オープンリダイレクト	p.73
8-1.	脆弱性概要	p.73
8-2.	確認方法	p.74
8-3.	検知リスト	p.75
8-4.	脆弱性判定 (管理者アカウント削除完了)	p.76
9	SQLインジェクション	p.78
9-1.	脆弱性概要	p.78
9-2.	確認方法	p.79
9-3.	検知リスト	p.85

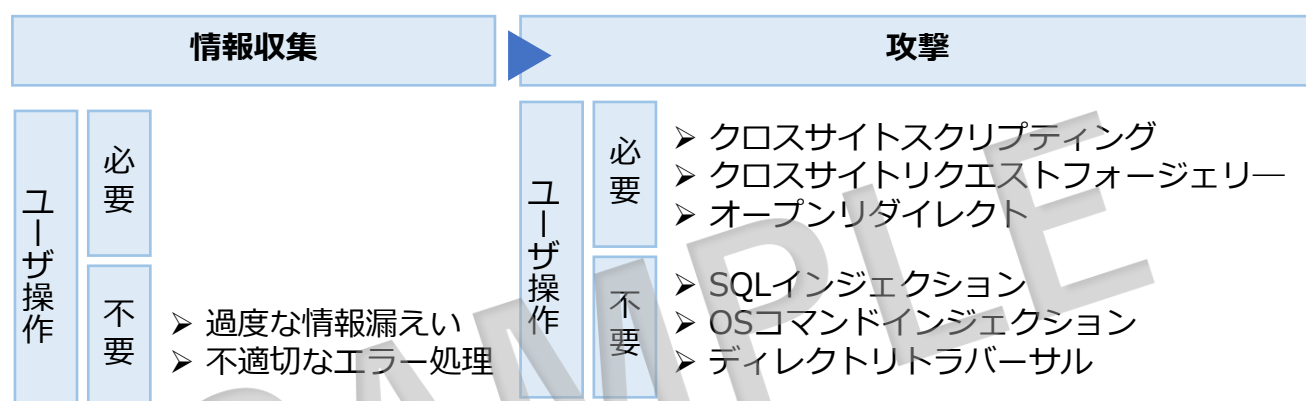
9-4.	脆弱性判定 (ログイン処理 (リダイレクト) /login_id)	p.86
9-5.	脆弱性判定 (管理者アカウント基本情報変更確認/lastname_kanji)	p.93
10	OSコマンドインジェクション	p.100
10-1.	脆弱性概要	p.100
10-2.	確認方法	p.101
10-3.	検知リスト	p.102
10-4	脆弱性判定 (カタログ一覧/category)	p.103
11	ディレクトリトラバーサル	p.107
11-1.	脆弱性概要	p.107
11-2.	確認方法	p.108
12	脆弱性の総合分析	p.110
12-1.	SQLインジェクション攻撃	p.112
12-2.	スキーマ名の取得	p.114
12-3.	テーブル名の取得	p.119
12-4.	カラム名の取得	p.120
12-5.	データの取得	p.121
12-6.	ブラインドSQLインジェクション	p.122
12-7.	OSコマンドインジェクション攻撃	p.126
12-8.	ディレクトリトラバーサル攻撃	p.128

2. 模擬試験で出題される脆弱性

VCAで出題対象となる脆弱性一覧は以下の通りです。

- | | |
|-----------------------|-------------------|
| 1. 過度な情報漏えい | 5. オープンリダイレクト |
| 2. 不適切なエラー処理 | 6. SQLインジェクション |
| 3. クロスサイトスクリプティング | 7. OSコマンドインジェクション |
| 4. クロスサイトリクエストフォージェリー | 8. ディレクトリトラバーサル |

サイバーセキュリティの攻撃工程は様々な工程に分解することが可能です。Webアプリケーションに関する脆弱性を「情報収集」、「攻撃」の工程に分類すると以下のようになります。



情報収集に関する脆弱性は、システム内部の情報収集が可能です。利用プロダクトのバージョンやサーバ内部のパス情報などを取得することが可能です。これらの情報は、別の攻撃を実行する際に活用される可能性があります。

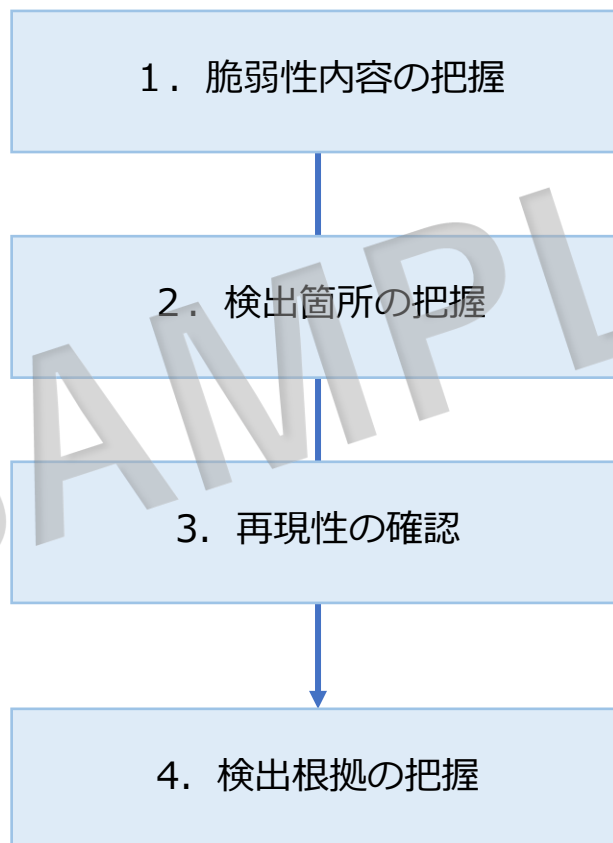
また、攻撃手法の分類としては、「ユーザ操作を必要する攻撃」と「ユーザ操作を必要としない攻撃」に分類されます。「ユーザ操作を必要する攻撃」は、「受動的攻撃」とも呼ばれます。受動的攻撃とは、攻撃者は脆弱性が発動するような罠を作成し、利用者を罠に誘導する攻撃です。受動的攻撃に属する攻撃としては、クロスサイトスクリプティング、クロスサイトリクエストフォージェリーなどがあります。

「ユーザ操作を必要としない攻撃」は、「能動的攻撃」とも呼ばれます。攻撃者はWebアプリケーションに直接攻撃を実行します。SQLインジェクションやOSコマンドインジェクションなど、〇〇インジェクションと呼ばれる脆弱性は大半が能動的脆弱性に分類されます。

3. 脆弱性判定の基本アプローチ

DASTツールで検出された脆弱性の精査（検知 or 過剰検知）をするため基本的なステップを解説します。ここで解説する内容は、Vexに限らず多くのツールに活用できる内容となっています。

ツールで検出された脆弱性の精査フローは以下のようになります。1～2の工程で、どのような脆弱性が、どこの画面で検出しているかを把握します。次に、再現性があることを確認します。最後に、ツールが脆弱性を検出した根拠を確認します。



次ページから各ステップについて詳細に解説していきます。

3-1. 脆弱性内容の把握

まずは、検出した脆弱性の内容（概要）について理解することが大切です。Vexでは検査結果の詳細画面から、脆弱性の概要解説を確認することが可能です。

検査情報

リクエスト

レスポンス



キャプチャ再取得

HTML

ソース

オリジナルログと比較

検査ログを保存

リクエスト編集

URL:

http://shop.ubsecurepict.com:80/MngAdmin/account_regist_confirm

機能名:

130.管理者アカウント登録確認-MNGA-10000

詳細	リクエストヘッダ	リクエストパラメータ	レスポンスパラメータ	レスポンス内コメント
適用シグネチャ:	クロスサイトスクリプティング (015530_MultiCrossSiteScripting)			
カテゴリ:	クロスサイトスクリプティング			
判定 危険度:	中			
タイプ:				
パラメータ名:				
元値:				
変更値:				
検出トリガ:				
レポートの解説変更:				
親検査結果ID:				

脆弱性カテゴリをクリックすると解説が表示される

シグネチャ情報

015530_MultiCrossSiteScripting

カテゴリ	クロスサイトスクリプティング
危険度	中
概要	クロスサイトスクリプティング
解説	パラメータに対してJavaScriptを挿入したところ、レスポンスでJavaScriptが動作しました。この挙動から、攻撃者により悪意あるJavaScriptが挿入された場合、Cookieの漏えいや画面の改ざんが発生する可能性があります。さらに、セッション管理に使用しているCookieが漏えいした場合、なりすましにつながる可能性があります。 ユーザ入力値の出力箇所に応じた対策が必要です。共通の対策を行った後、出力箇所に応じた対策をしてください。 [共通の対策]Content-Typeヘッダおよび、X-Content-Type-Options:nosniffヘッダを出力してください。また、ユーザ入力値の出力箇所を属性値や変数に代入する場合、出力箇所の前後を " または " で括ってください。 [出力箇所A]HTML本文およびHTMLタグ属性値 対策:HTML特殊文字 (&、<、>、"、') をエスケープしてください。 [出力箇所B]URLへアクセスするSRC属性HREF属性 対策:[出力箇所A]の対策とあわせて、スキーム (http、httpsなど) の指定およびURL形式のチェックをしてください。
推奨する対策	

閉じる