Webアプリケーション脆弱性診断ガイドライン Ver1.2.1

No	大分類	中分類	小分類	診断を実施すべき機能	ペイロード・検出パターン	診断方法	脆弱性があると疑われる挙動	備考
1	Webアプリケーションの脆弱性	インジェクション	SQLインジェクション	すべて	'(シングルクォート) 1/0	パラメータの値に検出パターンを挿入し、リクエストを送信	DB関連のエラーやInternal Server Errorが表示される	DB関連のエラー（SQL Syntax、SQLException、pg_exec、ORA-5桁数字、ODBC Driver Managerなど）は画面に表示されることもあれば、HTMLソースに表示されることもある SQLインジェクションがあるが、エラーが画面にでない場合には正常時と挙動が異なることもある ただし、この診断手法の脆弱性の有無については確定ではなく、あくまで可能性を示唆するものである
2				すべて	1/0	パラメータの値に検出パターンを挿入し、リクエストを送信	演算が実行される（ゼロ除算のエラーになる）	
3				すべて	(1)「(元の値)」 (2)「(元の値)' and 'a'='a」 (3)「(元の値)' and 'a'='b」	パラメータの値に検出パターンを挿入し、リクエストを送信	(1)を送信して正常系の動作を確認し、(1)と(2)を比較して同一のレスポンスとなり、(2)と(3)で異なるレスポンスが返ってくる	
4				すべて	(1)「(元の値:数値)」 (2)「(元の値) and 1=1」 (3)「(元の値) and 1=0」	パラメータの値に検出パターンを挿入し、リクエストを送信	(1)を送信して正常系の動作を確認し、(1)と(2)を比較して同一のレスポンスとなり、(2)と(3)で異なるレスポンスが返ってくる	
5				すべて	(1)「(元の値:数値)」 (2)「(元の値)-0」 (3)「(元の値)-1」	パラメータの値に検出パターンを挿入し、リクエストを送信	(1)を送信して正常系の動作を確認し、(1)と(2)を比較して同一のレスポンスとなり、(2)と(3)で異なるレスポンスが返ってくる	
6				すべて	MySQLの場合 (1) (元の値)-(0*SLEEP(10)) (2) ' and SLEEP(5) = SLEEP(5) and 'a' = 'a PostgreSQLの場合 (1) (元の値) - (0*pg_sleep(10)) (2) ' and pg_sleep(5) = pg_sleep(5)	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが10秒遅くなる	データベースの種類が判明している場合、そのデータベースに合わせた検出パターンのみを送信すれば良い。 記載のないデータベースである場合は、各データベースにおけるsleep関数またはそれに相当する関数を検出パラメータとして送信する。
7			コマンドインジェクション	すべて	|/bin/sleep 20|	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる	
8				すべて	;/bin/sleep 20;	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる	
9				すべて	../../../../../../bin/sleep 20|	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる	
10				すべて	;ping -nc 20 127.0.0.1;	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが通常より遅くなる	
11				すべて	&ping -nc 20 127.0.0.1&	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが通常より遅くなる	
12				すべて	$(../../../../../../bin/sleep 20)	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる	
13				すべて	``/bin/sleep 20``	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる	
14				すべて	`/bin/sleep 20`	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる	
15			CRLFインジェクション	ヘッダに値を出力している箇所	%0d%0aSet-Cookie:kensa%3dkensa%3b	パラメータの値に検出パターンを挿入し、リクエストを送信	パラメータに改行が挿入され、新たなSet-Cookieヘッダフィールドが挿入される	主な診断対象はSet-CookieやLocationヘッダフィールド
16				ヘッダに値を出力している箇所	%0d%0a%0d%0akensa	パラメータの値に検出パターンを挿入し、リクエストを送信	パラメータに改行コードが2つ挿入され、「kensa」文字列がボディ部分に表示される	主にHTTPボディやメール本文だが、それらに限らない
17				メールメッセージのヘッダに値を出力している箇所	%0d%0aTo:(任意のメールアドレス)	パラメータの値に検出パターンを挿入し、リクエストを送信	挿入したメールアドレス宛にメールが配送される	受信可能なメールアドレスを用意する必要がある
18			クロスサイトスクリプティング(XSS)	すべて	">'><s>XSS	パラメータの値に検出パターンを挿入し、リクエストを送信	適切にエスケープされずに出力され、sタグが動作する、JavaScriptの文法エラーがスローされる等の事象が発生する	JavaScriptの文法エラーは開発者ツールで確認できる。
19				すべて	<script>alert(1)</script>	パラメータの値に検出パターンを挿入し、リクエストを送信	適切にエスケープされずに出力され、alert関数が動作する、JavaScriptの文法エラーがスローされる等の事象が発生する	JavaScriptの文法エラーは開発者ツールで確認できる。
20				すべて	'><svg/onload=confirm(1)>	パラメータの値に検出パターンを挿入し、リクエストを送信	適切にエスケープされずに出力され、confirm関数が動作する	
21				すべて	javascript:alert(1)	パラメータの値に検出パターンを挿入し、リクエストを送信	URIとして解釈される箇所に挿入される	
22				すべて	`+alert(1)+`	パラメータの値に検出パターンを挿入し、リクエストを送信	適切にエスケープされずに出力され、alert関数が動作する、JavaScriptの文法エラーがスローされる等の事象が発生する	
23				すべて	"onmouseover="alert(1)	パラメータの値に検出パターンを挿入し、リクエストを送信	適切にエスケープされずに出力され、alert関数が動作する、JavaScriptの文法エラーがスローされる等の事象が発生する	出力される箇所によっては、onload,onclick等ほかのイベントハンドラでしかalert関数が動作しない場合がある。詳細な属性値についてはhttps://portswigger.net/web-security/cross-site-scripting/cheat-sheet等を参照。また、"(ダブルクォート)を'(シングルクォート)にしなければ動作しない場合もある。
25			CSSインジェクション	すべて	XSSのペイロード・検出パターンと同様	パラメータの値に検出パターンを挿入し、リクエストを送信	styleタグやスタイルを指定してるプロパティ内など、CSSとして解釈される箇所に出力される	

No	大分類	中分類	小分類	診断を実施すべき機能	ペイロード・検出パターン	診断方法	脆弱性があると疑われる挙動	備考
26			Relative Path Overwrite	すべて	/test/test/ //	パラメータの値に検出パターンを挿入し、リクエストを送信	以下の条件を満たす場合に脆弱 1)出力されたページ内に相対パスで指定されたCSS/JavaScriptがあり、パラメータを操作した際にそのパスを起点としてCSS/JavaScriptの相対パスが決定される 2)指定先のファイルが操作できる（アップロードしたファイル等）	詳細は以下の論文を参照 https://www.mbsd.jp/Whitepaper/rpo.pdf . JavaScriptの場合はXSS、CSSの場合はCSSインジェクションの脆弱性と同様となる。
27			サーバサイドテンプレートインジェクション（SSTI）	すべて	${{<%[%'"]}%\	パラメータの値に検出パターンを挿入し、リクエストを送信	内部的に例外が発生するため、レスポンスでエラーが示される。	
28				すべて	対象の環境で利用しているテンプレートエンジンで、プレースホルダとみなされる文字列 テンプレートエンジンごとの例) - Twig (PHP) => ${7*7} - Jinja2 (Python) => ${7*7} - ERB (Ruby) => <%=7*7%> - Slim (Ruby) => #{7*7}	パラメータの値に検出パターンを挿入し、リクエストを送信	送信したプレースホルダ部分が、当該プレースホルダ内の式や表現がテンプレートエンジンに評価された後の値に置換され、レスポンスに出現する。または、内部的に例外が発生し、レスポンスでエラーが示される。	各テンプレートエンジンの記法については、テンプレートエンジンの公式ドキュメントを適宜参照。また、https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Template%20Injection なども有用。
29		パストラバーサル		ファイル名を扱っている画面や機能	../../../../../../../../../etc/hosts	パラメータの値に検出パターンを挿入し、リクエストを送信	/etc/hostsの内容が表示される	
30				ファイル名を扱っている画面や機能	../../../../../../../../etc/hosts%00	パラメータの値に検出パターンを挿入し、リクエストを送信	/etc/hostsの内容が表示される	
31				ファイル名を扱っている画面や機能	/etc/hosts	パラメータの値に検出パターンを挿入し、リクエストを送信	/etc/hostsの内容が表示される	
32				ファイル名を扱っている画面や機能	(1) ./(元の値) (2) ../(元の値)	パラメータの値に検出パターンを挿入し、リクエストを送信	(1)で正常系と同様の内容が表示され、(2)で(1)と異なる画面が表示される	
33				ファイル名を扱っている画面や機能	../../../../../../../../windows/win.ini	パラメータの値に検出パターンを挿入し、リクエストを送信	win.iniの内容が表示される	
34				ファイル名を扱っている画面や機能	../../../../../../../../windows/win.ini%00	パラメータの値に検出パターンを挿入し、リクエストを送信	win.iniの内容が表示される	
35				ファイル名を扱っている画面や機能	C:/windows/win.ini	パラメータの値に検出パターンを挿入し、リクエストを送信	win.iniの内容が表示される	
36		ファイルアップロードに関する不備	クロスサイトスクリプティング(XSS)	ファイルアップロード機能	htmlとして認識されうるファイルやSVGファイル等のアップロード ・HTMLファイル <script>alert(1)</script> ・SVGファイル <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd" ><svg version="1.1" baseProfile="full" xmlns="http://www.w3.org/2000/svg"> <script> alert(1); </script> </svg>	ファイルアップロード機能を使い検出パターンを含むファイルをアップロードした後、当該ファイルが設置されたと考えられるパスにアクセス	ペイロードに含めたJavaScriptプログラムが実行されアラートボックスが表示される	ファイルアップロードに関するペイロードについては、以下も参照。 https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Upload%20Insecure%20Files . .
37				ファイルアップロード機能およびファイル名を取り扱っている画面や機能	ファイル名やContent-typeにXSSの文字列を指定 "><svg onload=alert(1)>.jpg 中分類「インジェクション」中の小分類「クロスサイトスクリプティング(XSS)」を参照	パラメータの値に検出パターンを挿入し、リクエストを送信 ファイル名が出力されると考えられる画面にアクセス	ペイロードに含めたJavaScriptプログラムが実行されアラートボックスが表示される	

38				ファイルアップロード機能およびファイルのメタデータの出力箇所	メタデータに攻撃文字列を指定
					
					以下のコマンドでコメントにhtmlとして認識されうる文字列を含むJPGファイルを作成し、出来たファイルをペイロードとして用いる
					$ exiftool -comment='"><svg onload=alert(1)>' payload.jpg
					$ exiftool payload.jpg
					...
					Comment : "><svg onload=alert(1)>
					...
						パラメータの値に検出パターンを挿入し、リクエストを送信
						
						ファイルのメタデータ情報が出力されると考えられる画面にアクセス
							ペイロードに含めたJavaScriptプログラムが実行されアラートボックスが表示される	ExifToolコマンドについては以下を参照。
								https://exiftool.org/

39			任意コード実行	ファイルアップロード機能	各環境に合わせたスクリプトやコンパイル済みのプログラム アップロードできるファイル種別に指定がある場合は、その種別および識別方法に合わせてスクリプトを変更する。 ・phpが動く環境の場合 <?php echo 1+1; ・phpが動く環境で、PNGファイルしか正しく処理されない場合 [0x89][0x50][0x4E][0x47][0x0D][0x0A][0x1A][0x0A]<?php echo 1+1;	パラメータの値に検出パターンを挿入し、リクエストを送信 ファイルアップロード機能を使い任意のファイルをアップロードしたあと、当該ファイルが設置されたと考えられるパスにアクセス	1+1の演算結果として、2が表示される。	

No	大分類	中分類	小分類	診断を実施すべき機能	ペイロード・検出パターン	診断方法	脆弱性があると疑われる挙動	備考
40			LFI	ファイル名を扱っている画面や機能	中分類「パストラバーサル」を参照	ファイルアップロード機能を使い小分類「任意コード実行」のペイロードファイルをアップロードしたあと、検出パターンにあるパス・ファイル名をアップロードしたファイル名に合わせて変更し、リクエストを送信	アップロードしたファイルに記載した処理が実行される	
41			DoS攻撃	ファイルアップロード機能	各環境に合わせた大きなサイズのファイル 例) 1GBのバイナリファイルを作成する $ dd if=/dev/zero of=1G.bin bs=1M count=$((1024 * 1))	受け入れられる形式の大きなサイズのファイルを作成し、そのファイルをアップロード	レスポンスが異常に遅延したり、サーバが停止する。 または、アップロードされたファイルがファイルを格納しているストレージを完全に使用し尽くした場合には、ファイルアップロード機能のみが利用できなくなる。	一回のアップロードではDoSにならない場合でも、連続的にアップロードした場合にDoS攻撃が成立する可能性もある。
42			DoS攻撃	画像ファイルのアップロード機能	ヘッダ領域に細工し縦横長を最大サイズに設定した画像ファイルや、GIF形式でヘッダを改変しフレーム数を最大にした画像ファイル	細工した画像ファイルをアップロード	レスポンスが異常に遅延したり、サーバが停止する。 または、アップロードされたファイルがファイルを格納しているストレージを完全に使用し尽くした場合には、ファイルアップロード機能のみが利用できなくなる。	
43			圧縮ファイルの取り扱い不備	アップロードした圧縮ファイルを展開する機能(zipやdocxなど)	展開すると数GBになる圧縮ファイル（ZIP BOMB)	ファイルをアップロード	レスポンスが異常に遅延したり、サーバが停止する。 または、アップロードされたファイルがファイルを格納しているストレージを完全に使用し尽くした場合には、ファイルアップロード機能のみが利用できなくなる。	
44				アップロードした圧縮ファイルを展開する機能(zipやdocxなど)	ファイル名にパスを含むファイルを圧縮したファイル	ファイルをアップロードし、展開されたと推測されるファイルにアクセス	ファイル名に含まれるパスに応じてファイルが作成または上書きされる。	
45				アップロードした圧縮ファイルを展開する機能(zipやdocxなど)	シンボリックリンクを含む圧縮ファイル	ファイルをアップロード	シンボリックリンクが展開され、リンク先ファイルに応じた処理結果になる。	
46			XML外部エンティティ参照 (XXE)	アップロードされたDOCXやPPTXなどのXMLが含まれるファイルを処理する機能	中分類「XML外部エンティティ参照 (XXE)」を参照	xmlファイルやdocxファイルなどアップロードするファイルの内容に検出パターンを挿入し、リクエストを送信	挿入した外部エンティティが参照され、展開される	
47		XML外部エンティティ参照 (XXE)		リクエストにXMLが含まれている箇所	元の値: <?xml version="1.0" encoding="ISO-8859-1"?> <foo>test</foo> 試行例: <?xml version="1.0" encoding="ISO-8859-1"?> <!DOCTYPE foo [<!ELEMENT foo ANY > <!ENTITY xxe SYSTEM "file:///etc/hosts" >]><foo>&xxe; </foo>	XMLに検出パターンを挿入し、リクエストを送信	/etc/hostsの内容が表示される	指定する検出パターンのfoo の箇所は実装に合わせて変更する 「OWASP Top10 2017」の改定を受けて追加（ A4:XML外部エンティティ参照:XXE）
48				リクエストにXMLが含まれている箇所	元の値: <?xml version="1.0" encoding="ISO-8859-1"?> <foo>test</foo> 試行例: <?xml version="1.0" encoding="ISO-8859-1"?> <!DOCTYPE foo [<!ELEMENT foo ANY > <!ENTITY xxe SYSTEM "file:///c:/windows/win.ini" >] <foo>&xxe;</foo>	XMLに検出パターンを挿入し、リクエストを送信	win.iniの内容が表示される	指定する検出パターンのfoo の箇所は実装に合わせて変更する 「OWASP Top10 2017」の改定を受けて追加（ A4:XML外部エンティティ参照:XXE）
49				リクエストにXMLが含まれている箇所	元の値: <?xml version="1.0" encoding="ISO-8859-1"?> <foo>test</foo> 試行例: <?xml version="1.0" encoding="ISO-8859-1"?> <!DOCTYPE foo [<!ELEMENT foo ANY > <!ENTITY xxe SYSTEM "http://192.0.2.0/" >]><foo>&xxe; </foo>	XMLに検出パターンを挿入し、リクエストを送信	レスポンスの表示が遅延する	指定する検出パターンのfoo の箇所は実装に合わせて変更する 「OWASP Top10 2017」の改定を受けて追加（ A4:XML外部エンティティ参照:XXE）
50				リクエストにXMLが含まれている箇所	"元の値: <?xml version=""1.0"" encoding=""ISO-8859-1""?> <foo>test</foo> 試行例: <?xml version=""1.0"" encoding=""ISO-8859-1""?> <!DOCTYPE foo [<!ENTITY xxe ""test"">]> <foo>&xxe;</foo>"	XMLに検出パターンを挿入し、リクエストを送信	fooの値が表示されていた箇所にtestと表示される	指定する検出パターンのfoo の箇所は実装に合わせて変更する 「OWASP Top10 2017」の改定を受けて追加（ A4:XML外部エンティティ参照:XXE）

Webアプリケーション脆弱性診断ガイドライン Ver1.2.1

No	大分類	中分類	小分類	診断を実施すべき機能	ペイロード・検出パターン	診断方法	脆弱性があると疑われる挙動	備考
51				リクエストにXMLが含まれている箇所	元の値: <?xml version="1.0" encoding="ISO-8859-1"?> <foo>test</foo> 試行例: <?xml version="1.0" encoding="ISO-8859-1"?> <!DOCTYPE foo [<!ELEMENT foo ANY > <!ENTITY xxe SYSTEM "http://example.com/" >] <foo>&xxe;</foo>	XMLに検出パターンを挿入し、リクエストを送信	http://example.com/ にアクセスが来る	外部Webサーバを用意し、アクセスがログなどで確認できる必要がある 指定する検出パターンのfooの箇所は実装に合わせて、, http://example.com は用意した外部Webサーバに合わせて、それぞれ変更する 「OWASP Top10 2017」の改定を受けて追加（A4:XML外部エンティティ参照:XXE）
52				すべて		Content-typeをXMLを示す値に変更し、上記検出パターンを含めてリクエストを送信	XMLとして解釈され、上記挙動が起こる	
53		オープンリダイレクト		リダイレクトが実行される画面や機能	https://www.example.com/	パラメータの値に検出パターンを挿入し、リクエストを送信	https://www.example.comにリダイレクトされる	指定する検出パターンのURLの形式は必要に応じて変更する 主な診断対象は、Locationヘッダフィールド、METAタグのRefresh、JavaScriptコード（location.href、location.assign、location.replace）
54				リダイレクトが実行される画面や機能	///www.example.com/	パラメータの値に検出パターンを挿入し、リクエストを送信	https://www.example.comにリダイレクトされる	指定する検出パターンのURLの形式は必要に応じて変更する 主な診断対象は、Locationヘッダフィールド、METAタグのRefresh、JavaScriptコード（location.href、location.assign、location.replace）
55				リダイレクトが実行される画面や機能	https://(正常系で遷移しているドメイン)@www.example.com/	パラメータの値に検出パターンを挿入し、リクエストを送信	https://www.example.comにリダイレクトされる	指定する検出パターンのURLの形式は必要に応じて変更する 主な診断対象は、Locationヘッダフィールド、METAタグのRefresh、JavaScriptコード（location.href、location.assign、location.replace）
56		シリアライズされたオブジェクト		すべて	正常系に以下のようなシリアライズされた値が含まれていること（言語によってシリアライズ形式は変わります） ■PHPの場合 「a:4:{i:0;i:132;i:1;s:7:"Mallory";i:2;s:4:"user";i:3;s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";}」のような値 ■Javaの場合 rO0（小文字アール、大文字オー、数字0）から始まるBase64文字列 H4sIA（大文字エイチ、数字4、小文字エス、大文字アイ、大文字エー）から始まるBase64文字列	検出パターンがリクエストに含まれていないか確認	検出パターンが含まれている	ただし、この診断手法の脆弱性の有無については確定ではなく、あくまで可能性を示唆するものである。 詳細については、以下のドキュメントを参照。 https://cheatsheetseries.owasp.org/cheatsheets/Deserialization_Cheat_Sheet.html
57		サーバサイドリクエストフォージェリ（SSRF）		ファイル名やURLを扱っている画面や機能	外部サーバのURL	パラメータの値に検出パターンを挿入し、リクエストを送信	外部サーバのアクセスログに診断対象サーバからのアクセスログが残る。	アクセスログが閲覧できる外部Webサーバを用意する必要がある。
58				ファイル名やURLを扱っている画面や機能	存在しないホストを示すURL	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが遅延する。	
59				ファイル名やURLを扱っている画面や機能	1)空いている可能性の高いポートを指定したURL https://localhost:443/ 2) 空いていない可能性の高いポートを指定したURL https://localhost:23456/	パラメータの値に検出パターンを挿入し、リクエストを送信	(1)と(2)でレスポンスの時間や内容に差がある。	
60				ファイル名やURLを扱っている画面や機能	パスやドメインなどが正常系として処理される条件を満たし、かつ対象システムの内部ネットワークに存在すると推測されるホストにリダイレクトするURL URL例） ・https://169.254.169.254/ （AWSにおけるメタデータサーバ） ・https://127.0.0.1:8080/ （ループバックアドレス）	パラメータの値に検出パターンを挿入し、リクエストを送信	リダイレクト先ホストから出力されたと考えられる情報が返される	
61		WebSocket	クロスサイトウェブソケットハイジャッキング(CSWSH)	WebSocketで通信している機能	WebSocketで通信を行っている、もしくはWebSocket通信に対応していること	対象と異なるオリジン上に設置した罠ページからWebSocket通信接続を行い、アプリケーションの機能を操作するメッセージを送信	WebSocket通信を経由してアプリケーションを操作できる	
62		クリックジャッキング		確定処理の直前画面	レスポンスヘッダにX-Frame-Optionsヘッダが存在すること	検出パターンがリクエストに含まれていないか確認	X-Frame-Optionsヘッダがない、または、値が「DENY」「SAMEORIGIN」ではない。	Content-Security-Policyヘッダ相当の要素があるときは脆弱性が無い場合がある。

Webアプリケーション脆弱性診断ガイドライン Ver1.2.1

No	大分類	中分類	小分類	診断を実施すべき機能	ペイロード・検出パターン	診断方法	脆弱性があると疑われる挙動	備考
63				確定処理の直前画面	レスポンスヘッダまたはレスポンスボディにContent-Security-Policyヘッダ相当の要素が存在すること	検出パターンがリクエストに含まれていないか確認	Content-Security-Policyヘッダ相当の要素に「frame-ancestors」「frame-src」が設定されていない、または、*など攻撃者がフレームを設置可能な要素が指定されている。	X-Frame-Optionsヘッダがあるときは脆弱性が無い場合がある。 なおIEにおいては指定を無視するおそれがあるため、Content-Security-Policyヘッダ相当の要素が指定されていても、X-Frame-Optionsヘッダが無い場合は脆弱性があると疑われる。
64		認証	認証回避	認証が必要な箇所	認証状態を保持しているパラメータ（例：SESSIONID） SESSIONID=（空の値） （パラメータ名ごと削除）	ペイロードを送信して認証後のページにアクセスする	認証後の情報が出力される	
65				ログイン機能	password=（空の値） （パラメータ名ごと削除） password=passworda（正常文字列に文字列を付与する）	ペイロードを送信してログインを試行する	認証が成功する	
66			ログアウト機能の不備や未実装	ログアウト機能		ログアウト機能が存在するかを確認	ログアウト機能が存在しない	
67				ログアウト機能		認証で使っているセッションIDをメモし、ログアウト機能を実行後、メモしたセッションIDを付与してログイン状態になることを確認	認証状態でしかアクセスできない画面や機能にアクセスできる（ログイン状態になる）	ログアウト機能の実行時にセッションIDが破棄されていない場合に発生する
68			過度な認証試行に対する対策不備や未実装	ログイン機能		異なるセッションにおいて、同じユーザ名でパスワードを連続して20回間違えてから、正しいパスワードでログインを試行	認証が成功する	試行するパスワードの文字種についてはパスワードポリシーに従うこと。
69			脆弱なパスワードポリシー	パスワード登録・変更	(空) 1234567 abcdefg abcd123	パスワード文字列の桁数が8文字未満の文字列を登録・変更できないことを確認	脆弱なパスワードが登録・変更できる	要件定義書の基準も参照 https://github.com/OWASP/www-chapter-japan/blob/master/secreq/OWASP_WebApplicationSecurityRequirements.pdf
70				パスワード登録・変更	Rf@9yY8&wk	パスワード文字列の桁数が8文字以上、かつ文字種が大小英字、数字、記号の3種類が混在している文字列を登録・変更できることを確認	登録・変更できない	
71				パスワード登録・変更		ユーザ名と同じパスワードが登録・変更できないことを確認	脆弱な(推測可能な)パスワードが設定できる	脆弱性ではないが、指摘すべき事項である。
72			復元可能なパスワード保存	パスワード登録・変更		パスワードリマインダ機能でパスワードを問い合わせて確認	登録したパスワードが返ってくる	
73				すべて		設定したパスワードが、いずれかのページで表示や埋め込まれていないことを確認	レスポンスにパスワードが埋め込まれている	
74			パスワードリセットの不備	パスワードリセット		パスワードリセットを実行して、再設定時に本人確認をしていることを確認	本人確認されずにパスワードのリセットが可能	要件定義書の基準も参照 https://github.com/OWASP/www-chapter-japan/blob/master/secreq/OWASP_WebApplicationSecurityRequirements.pdf
75				パスワードリセット		パスワードリセットを実行して、ユーザ自身による新たなパスワード設定が強制されることを確認	システムが生成したパスワードが送付され、そのまま使い続けられる	脆弱性ではないが、指摘すべき事項である。
76		セッション管理の不備	セッションフィクセイション(セッション固定攻撃)	ログイン機能		認証に使用しているセッションIDに関して、ログイン成功後にログイン成功前の値が継続して使用できるか確認	ログイン成功前のセッションIDを使用して、ログイン状態と同等の操作が行える場合	
77				ログイン前に機微情報がセッション変数に格納されていると想定できる箇所		氏名やメールアドレス等の機微情報を管理しているセッションIDに関して、機微情報を入力した後に入力前の値が継続して使用できるか確認	機微情報入力前のセッションIDを使用して、入力した機微情報を用いた操作が行える場合	
78			推測可能なセッションID	セッションID発行時		セッションIDを複数集めて規則性があることを確認し、セッションIDを推測 ・ユーザアカウントごとに差異の比較 ・発行時の日時による差異の比較 ・発行回数による差異の比較	セッションIDに規則性があり推測可能	
79		クロスサイトリクエストフォージェリ(CSRF)		登録、送信などの確定処理をGETで行っている箇所		Cookieなどリクエストヘッダに含まれたSamesite属性がnoneまたはlaxのパラメータによってセッション管理が行われている確定処理において、以下の3点を確認 ①以下のいずれかの情報がパラメータ中に含まれているかを確認 A. 利用者のパスワード B. CSRF対策トークン C. セッションID ②A～Cが含まれている場合に、ユーザαで利用されている値を削除、もしくはパラメータごと削除してリクエストを送信し、処理が行われるか確認 ③Refererを削除、もしくは正規のURLではない値に変更して、リクエストを送信し、処理が行われるか確認	1) A～Cいずれもが含まれていない 2) A～Cが含まれているが、値を削除、もしくはパラメータごと削除した場合に処理が行われる 3) Refererチェックが行われていない	脆弱性があると疑われる挙動を示していても、RefererやOrigin、Sec-Fetch-*等のヘッダによって対策されている場合もあるため、罠ページを作成して動作を確認する必要がある
80				登録、送信などの確定処理をPOSTで行っている箇所		CookieのうちSamesite属性がnoneの値によってセッション管理が行われている確定処理において、以下の3点を確認 ①以下のいずれかの情報がパラメータ中に含まれているかを確認 A. 利用者のパスワード B. CSRF対策トークン C. セッションID ②A～Cが含まれている場合に、ユーザαで利用されている値を削除、もしくはパラメータごと削除してリクエストを送信し、処理が行われるか確認 ③Refererを削除、もしくは正規のURLではない値に変更して、リクエストを送信し、処理が行われるか確認	1) A～Cいずれもが含まれていない 2) A～Cが含まれているが、値を削除、もしくはパラメータごと削除した場合に処理が行われる 3) Refererチェックが行われていない	脆弱性があると疑われる挙動を示していても、RefererやOrigin、Sec-Fetch-*等のヘッダによって対策されている場合もあるため、罠ページを作成して動作を確認する必要がある
81				CSRF対策トークンを使用している箇所		①ユーザαで利用されている値をユーザβで利用されている値に変更してリクエストを送信し、処理が行われるか確認 ②CSRF対策トークンを複数集めて規則性があることを確認し、CSRF対策トークンを推測 ・ユーザアカウントごとに差違の比較 ・同一ユーザでログインするごとに差違の比較	CSRF対策トークンが推測可能	脆弱性があると疑われる挙動を示していても、RefererやOrigin、Sec-Fetch-*等のヘッダによって対策されている場合もあるため、罠ページを作成して動作を確認する必要がある

Webアプリケーション脆弱性診断ガイドライン Ver1.2.1

No	大分類	中分類	小分類	診断を実施すべき機能	ペイロード・検出パターン	診断方法	脆弱性があると疑われる挙動	備考
82		情報漏洩	クエリストリング情報の漏洩	すべて		セッションIDや機微情報がURLに含まれていないか確認	URLにセッションIDや機微情報が含まれている	他サイトにリクエストを送信する際に、Refererヘッダから内容が漏洩する。 Webサーバやプロキシーサーバにログとして残る。
83			キャッシュからの情報漏洩	機微情報が含まれる画面		レスポンス内で適切にキャッシュ制御を行っていることを確認	レスポンスヘッダのCache-Controlヘッダフィールド値に"no-store"が指定されていない	CDNによっては"private"を設定する必要がある。 参考：https://engineering.mercari.com/blog/entry/2017-06-22-204500/
84			パスワードフィールドのマスク不備	パスワード入力画面		パスワード入力に使用するinputタグのtype属性に"password"が指定されていることを確認	inputタグのtype属性が"password"ではない	
85			画面表示上のマスク不備	全般		マスクすべき情報が画面上に表示されていないことを確認	マスクすべき情報が画面上に表示されている	主なマスクすべき情報としてはクレジットカード番号やPINコード、パスワード、マイナンバーなど
86			HTTPS利用時のCookieのSecure属性未設定	Set-Cookieヘッダフィールドがある箇所		HTTPS利用時のSet-Cookieヘッダフィールドに Secure属性があることを確認	レスポンスヘッダの Set-Cookieヘッダフィールド値に"Secure"属性が指定されていない	
87			CookieのHttpOnly属性未設定	Set-Cookieヘッダフィールドがある箇所		Set-Cookieヘッダフィールドに HttpOnly属性があることを確認	レスポンスヘッダの Set-Cookieヘッダフィールド値に"HttpOnly"属性が指定されていない	
88			パスワードの管理不備	パスワードリセット		パスワードリセットを実行	登録したパスワードが表示される/メールで送られてくる	
89				パスワード変更画面、管理者画面		パスワード変更画面、管理者画面にアクセス	登録したパスワードが表示される/inputタグに埋まっている	
90			HTTPSの不備	全般		Webページにアクセス	HTTPで通信している	組織内NWについてもHTTPSの使用が推奨される
91				HTTPS箇所		HTTPSを使用しているコンテンツを確認(HTTPおよびHTTPSの併用)	HSTS(Strict-Transport-Security)ヘッダをつけていない	組織内NWについてもHTTPSの使用が推奨される
92				HTTPS箇所		HTTPSを使用しているコンテンツを確認(HTTPとHTTPSの混在)	HTTPSとHTTPのコンテンツが混在している	
93				HTTPS箇所		動作対象ブラウザで証明書を確認	ブラウザで証明書の警告が出る	中間者攻撃が行われいるか否かの判別が困難になるため、組織内NWであっても警告が出ない実装が推奨される
94			不要な情報の存在	すべて		HTTPレスポンスやメールなどに「攻撃に有用な情報（設計やデータベース構造になどに係わる情報、バージョン情報）」や「公開不要な情報（個人名、メールアドレス、ミドルウェアの情報、過去の公開していたコンテンツのリンク、プライベートIPアドレスなど）」が含まれていることを確認	情報が含まれている	
95	Webアプリケーションの動作環境への診断項目	サーバソフトウェアの設定の不備	ディレクトリリスティング	すべて		Webサーバ上の発見したディレクトリにアクセスして、ディレクトリ内のファイルが一覧表示されないかを確認	ディレクトリ内のファイルが一覧表示される	含まれているファイルによってリスクは異なる
96			不要なHTTPメソッド	すべて	TRACE、TRACK	メソッドを変更してサーバにアクセス	TRACE、TRACKメソッドが機能する	
97			不要なHTTPメソッド	すべて	OPTIONS	メソッドを変更してサーバにアクセス	Allowヘッダに GET、HEAD、POST、OPTIONS以外のメソッドが存在する（PUT、DELETE、TRACEなど）	REST APIなどでは、正常系でPUT/DELETEを用いている場合もある。
98		公開不要な機能・ファイル・ディレクトリの存在		すべて	.bak, .old, .org, .htaccess, web.config, file.html~, /admin/, /test/, /.git/, test.html など	サンプルファイルや、バックアップファイルなど、アプリケーションの動作に不必要なファイルの有無を確認 不特定多数に公開する必要がないファイルの有無を確認	該当するファイルがある	
99		既知脆弱性		すべて		取得したバージョン情報から、フレームワークやライブラリに既知脆弱性がないか確認	既知脆弱性がある （CVE番号が付与されている、パッチノートに記載があるなど）	