CONFIDENTIAL

セキュリティブランケット　診断レポート

作成日時：2012/11/08 14:35:07

株式会社M&K

1.はじめに 3
目的 3
2.概要 4
2.1.診断情報 4
2.2.診断対象URL 4
2.3.総合評価 5
2.4.脆弱性検出件数 5
2.5.脆弱性カテゴリ 6
3.Webサイト毎の脆弱性詳細（demo profile/demo) 7
3.1.ディレクトリトラバーサル 7
3.2.クロスサイトスクリプティング 11
3.3.SQLインジェクション(エラーメッセージ) 15
3.4.HTMLインジェクション(リンク) 19
3.5.特殊文字のエスケープ漏れ 23
3.6.HttpOnly属性の欠如 27
3.7.発行されているCookieの情報 31
3.8.コメントの検出 35
付録 40
診断カテゴリ概要 40
危険度の判定基準 42
評価基準 42
お問合せ先 43

1.はじめに

本報告書は、「2012年11月08日 13:47:42～2012年11月08日 13:50:31」に実施したWebアプリケーション脆弱性検査の検査結果についてご報告するものです。

目的

本検査の目的は、検査対象Webアプリケーションに対してリモートから脆弱性の検査を行い、Webアプリケーションに存在する脆弱性を検出することにあります。また、脆弱性が検出された場合、そのリスク評価、及び、脆弱性への対策を支援する情報の提供も行います。

2.概要

2.1.診断情報

診断ID	プロファイル/サブプロファイル	URL	IPアドレス
	診断日時	登録ユーザ/実行ユーザ	
665	demo profile/demo (終了)	http://security-blanket.jp/mutillidae/	
	開始 2012/11/08 13:47:42 終了 2012/11/08 13:50:31	デモユーザ管理者/システム管理者	

2.2.診断対象URL

demo profile/demo

診断ID:665 - 開始:2012/11/08 13:47:42 終了:2012/11/08 13:50:31

No.	URL	タイトル
1	http://security-blanket.jp/mutillidae/	
2	http://security-blanket.jp/mutillidae/?page=show-log.php	
3	http://security-blanket.jp/mutillidae/framer.html	Framer
4	http://security-blanket.jp/mutillidae/index.php	
5	http://security-blanket.jp/mutillidae/index.php?do=toggle-security&page=login.php	
6	http://security-blanket.jp/mutillidae/index.php?page=login.php	
7	http://security-blanket.jp/mutillidae/index.php?page=login.php	
8	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=anonymous	
9	http://security-blanket.jp/mutillidae/setupreset.php	

2.3.総合評価

今回実施した検査の検査結果に基づき、

D	大きな被害を受けることが懸念される危険性の高い脆弱性が確認されております。早急に対策を行うことを推奨します。

と評価しました。

この評価は、SaaS型脆弱性診断サービス「セキュリティブランケット」による検査結果を根拠に弊社の評価基準に照合したものです。本検査は全ての脆弱性を網羅するものでありませんのでご注意ください。

2.4.脆弱性検出件数

診断ID	プロファイル/サブプロファイル	対象	緊急	重大	高	中	低	情報	小計
665	demo profile/demo	Web	4	16	10	0	12	8	50
		Net	0	0	0	0	0	0	0
合計			4	16	10	0	12	8	50

2.5.脆弱性カテゴリ

診断ID: 665 demo profile/demo

Webアプリケーション　検出カテゴリ

No.	危険度	カテゴリ	件数
1	緊急	ディレクトリトラバーサル	4
2	重大	クロスサイトスクリプティング	10
3	重大	SQLインジェクション(エラーメッセージ)	6
4	高	HTMLインジェクション(リンク)	10
5	低	特殊文字のエスケープ漏れ	10
6	低	HttpOnly属性の欠如	2
7	情報	発行されているCookieの情報	2
8	情報	コメントの検出	6

3.Webサイト毎の脆弱性詳細(demo profile/demo)

診断ID:665 - 開始:2012/11/08 13:47:42 終了:2012/11/08 13:50:31

3.1.ディレクトリトラバーサル Directory Traversal	4件

概要

公開することを意図していないディレクトリのファイルに対して、不正にディレクトリパスをさかのぼりアクセス可能です。
この脆弱性が悪用された場合、
1. 重要情報の漏洩
2. アプリケーションの作りによってはファイルの改ざん
などの影響があります。

参考情報　http://www.ipa.go.jp/security/vuln/vuln_contents/dt.html

対策

ファイル名を外部から指定できないよう「../」や「/」などのパス名として識別される文字列のエスケープを適切に行って下さい。また、クライアントからの取得データは英数字のみ許可することも対策となります。

検出例

URL	Method	パラメータ
http://security-blanket.jp/mutillidae/?page=..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd	GET	page

診断用文字列

```
../../../../../etc/passwd
```

脆弱性検出文字列

code:	
header:	
content:	root:x:0:0:root:/root:/bin/bash

リクエストライン

```
GET http://security-blanket.jp/mutillidae/?page=..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd
```

リクエストヘッダ

```
1 : GET http://security-blanket.jp/mutillidae/?page=..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1
2 : Accept-language: ja,en-us;q=0.7,en;q=0.3
3 : Keep-alive: 115
4 : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 : User-agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
6 : Accept-charset: Shift_JIS,utf-8;q=0.7,*;q=0.7
7 : Host: security-blanket.jp
8 : Cookie: showhints=0; PHPSESSID=vkpo6pd2olrpiifd3ihtjg2b76
9 : Authorization: Basic eWFyYXJlc2FpdG86bWtfcGVudGVzdA==
```

リクエストボディ

```

```

HTTPステータス

```
200
```

レスポンスヘッダ

```
1 : Date: Thu, 08 Nov 2012 04:48:13 GMT
2 : Server: Apache
3 : Logged-In-User:
4 : Vary: Accept-Encoding
5 : Connection: close
6 : Transfer-Encoding: chunked
7 : Content-Type: text/html
```

レスポンスボディ

```
386 : <blockquote>
387 : <!-- Begin Content -->
388 : root:x:0:0:root:/root:/bin/bash
389 : daemon:x:1:1:daemon:/usr/sbin:/bin/sh
390 : bin:x:2:2:bin:/bin:/bin/sh
```

該当箇所一覧

No.	URL	Method	パラメータ
1	http://security-blanket.jp/mutillidae/?page=show-log.php	GET	page
2	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	page
3	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	page
4	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	-

3.2.クロスサイトスクリプティング
Cross Site Scripting

10件

概要

[<,>,",'] 等の特殊文字がエスケープされていない為、スクリプトの実行、ページ改ざんが可能です。
この脆弱性が悪用された場合、
1. Cookieが盗まれることによる個人情報漏洩
2. Webページ改竄にるフィッシング詐欺
などの影響があります。

参考情報　http://www.ipa.go.jp/security/vuln/vuln_contents/xss.html

対策

この脆弱性に対応するには、Web アプリケーションにおいて出力部分に応じた文字列の適切なエスケープ/エンコード処理を行うことが必要です。
また、Web アプリケーション開発時に以下を原則とすることで、クロスサイトスクリプティングを含め、多くの脆弱性による影響を大幅に緩和することが可能です。

入力値の形式や文字種別、桁数を厳密に定義し、正しい入力値のみを受け付けるように処理する。
例：電話番号の値には10-12 桁の半角数字のみを許可等
JavaScript 等を使用したクライアント側での入出力チェックに依存せず、サーバ側で入出力チェックを行う。

なお、クロスサイトスクリプティングの原因は文字列処理が適切にされていないことに起因するため、本指摘事項以外にも、文字列処理を行う全ての部分に注意する必要があります。そのため指摘部分への対策だけではなく、アプリケーション全体の確認と対策を推奨いたします。
・エスケープ対象文字列

【 <> " ' % () & + ; 】

検出例

URL	Method	パラメータ
http://security-blanket.jp/mutillidae/?page=show-log.php%22%27%3E%3CsCrIpT%3Ealert(%22-@18736-61118-665@-%22)%3C%2FScRiPt%3E	GET	page

診断用文字列

```
"'><sCrIpT>alert("-@18736-61118-665@-")</ScRiPt>
```

脆弱性検出文字列

code:	
header:	
content:	<td><sCrIpT>alert("-@18736-61118-665@-")</ScRiPt>">Toggle Security</td>

リクエストライン

```
GET http://security-blanket.jp/mutillidae/?page=show-log.php%22%27%3E%3CsCrIpT%3Ealert(%22-@18736-61118-665@-%22)%3C%2FScRiPt%3E
```

リクエストヘッダ

```
1 : GET http://security-blanket.jp/mutillidae/?page=show-log.php%22%27%3E%3CsCrIpT%3Ealert(%22-@18736-61118-665@-%22)%3C%2FScRiPt%3E HTTP/1.1
2 : Accept-language: ja,en-us;q=0.7,en;q=0.3
3 : Keep-alive: 115
4 : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 : User-agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
6 : Accept-charset: Shift_JIS,utf-8;q=0.7,*;q=0.7
7 : Host: security-blanket.jp
8 : Cookie: showhints=0; PHPSESSID=7i00uqk02f22b7fgoaqusb0811
9 : Authorization: Basic eWFyYXJlc2FpdG86bWtfcGVudGVzdA==
```

リクエストボディ

HTTPステータス

```
200
```

レスポンスヘッダ

```
1 : Date: Thu, 08 Nov 2012 04:48:04 GMT
2 : Server: Apache
3 : Logged-In-User:
4 : Vary: Accept-Encoding
5 : Connection: close
6 : Transfer-Encoding: chunked
7 : Content-Type: text/html
```

レスポンスボディ

```
92 : <a href="./index.php?page=login.php">Login/Register</a>
93 : </td>
94 : <td><a href="./index.php?do=toggle-security&page=show-log.php"'><sCrIpT>alert("-@18736-61118-665@-")
     </ScRiPt>">Toggle Security</a></td>
95 : <td><a href="setupreset.php">Setup/Reset the DB</a></td>
96 : </tr>
```

該当箇所一覧

No.	URL	Method	パラメータ
1	http://security-blanket.jp/mutillidae/?page=show-log.php	GET	page
2	http://security-blanket.jp/mutillidae/index.php	GET	uid
3	http://security-blanket.jp/mutillidae/index.php	GET	-
4	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	uid
5	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	page
6	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	-
7	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	username
8	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	uid
9	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	page
10	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	-

3.3.SQLインジェクション(エラーメッセージ)
Detect SQL Error Message

6件

概要

SQLエラーメッセージを検出しました。SQLインジェクションが発生する可能性があります。
この脆弱性が悪用された場合、
1. 重要情報の漏洩
2. データの改ざん
3.サービスの停止
などの影響があります。

参考情報　http://www.ipa.go.jp/security/vuln/vuln_contents/sql.html

対策

ストアード・プロシージャーを使用し、直接データベースにアクセスしないでください。開発環境にストアド・プロシージャーが用意させていない場合は、SQL文作成に使用される値の特殊文字をエスケープ処理してください。また、エラーメッセージはカスタムエラーページを用意するなどし、画面に表示しないように設定してください。
・エスケープ対象文字列

【' " ￥' ￥") ; 】

検出例

URL	Method	パラメータ
http://security-blanket.jp/mutillidae/index.php	GET	uid

診断用文字列

```
' SELECT _SCAN_;--
```

脆弱性検出文字列

code:	
header:	
content:	<td class="error-label">Message</td><td class="error-detail">Error executing query: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'SELECT _SCAN_'' at line 1</td>

リクエストライン

```
GET http://security-blanket.jp/mutillidae/index.php
```

リクエストヘッダ

```
1 : GET http://security-blanket.jp/mutillidae/index.php HTTP/1.1
2 : Accept-language: ja,en-us;q=0.7,en;q=0.3
3 : Keep-alive: 115
4 : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 : User-agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
6 : Accept-charset: Shift_JIS,utf-8;q=0.7,*;q=0.7
7 : Host: security-blanket.jp
8 : Referer: http://security-blanket.jp/mutillidae/index.php?page=login.php
9 : Cookie: uid=15' SELECT _SCAN_;--; PHPSESSID=pj4m5qcqo7iinjtoaf8ge2t5s6
10 : Authorization: Basic eWFyYXJlc2FpdG86bWtfcGVudGVzdA==
```

リクエストボディ

```

```

HTTPステータス

```
200
```

レスポンスヘッダ

```
1 : Date: Thu, 08 Nov 2012 04:48:50 GMT
2 : Server: Apache
3 : Vary: Accept-Encoding
4 : Connection: close
5 : Transfer-Encoding: chunked
6 : Content-Type: text/html
```

レスポンスボディ

```
14 : </tr>
15 : <tr>
16 : <td class="error-label">Message</td><td class="error-detail">Error executing query: You have an erro
     r in your SQL syntax; check the manual that corresponds to your MySQL server version for the right s
     yntax to use near 'SELECT _SCAN_'' at line 1</td>
17 : </tr>
18 : <tr>
```

該当箇所一覧

No.	URL	Method	パラメータ
1	http://security-blanket.jp/mutillidae/index.php	GET	uid
2	http://security-blanket.jp/mutillidae/index.php	GET	-
3	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	uid
4	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	-
5	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	uid
6	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	-

3.4.HTMLインジェクション(リンク) HTML Injection(Link)	10件

概要

ページ内に任意のリンクを埋め込むことが可能です。
この脆弱性が利用された場合、悪意のあるページへの誘導や意図しない操作の実行を強制される可能性があります。

対策

この脆弱性に対応するには、Web アプリケーションにおいて出力部分に応じた文字列の適切なエスケープ/エンコード処理を行うことが必要です。
また、Web アプリケーション開発時に以下を原則とすることで、リンクインジェクションを含め、多くの脆弱性による影響を大幅に緩和することが可能です。

入力値の形式や文字種別、桁数を厳密に定義し、正しい入力値のみを受け付けるように処理する。
例：電話番号の値には10-12 桁の半角数字のみを許可等
JavaScript 等を使用したクライアント側での入出力チェックに依存せず、サーバ側で入出力チェックを行う。

なお、リンクインジェクションの原因は文字列処理が適切にされていないことに起因するため、本指摘事項以外にも、文字列処理を行う全ての部分に注意する必要があります。そのため指摘部分への対策だけではなく、アプリケーション全体の確認と対策を推奨いたします。

検出例

URL	Method	パラメータ
http://security-blanket.jp/mutillidae/?page=show-log.php%27%22%3E%3CIMG+SRC%3D%22%2F-@18736-61118-665@-%2F_SCAN_.html%22%3E	GET	page

診断用文字列

```
'"><IMG SRC="/-@18736-61118-665@-/_SCAN_.html">
```

脆弱性検出文字列

code:	
header:	
content:	<td><IMG SRC="/-@18736-61118-665@-/_SCAN_.html">">Toggle Security</td>

リクエストライン

```
GET http://security-blanket.jp/mutillidae/?page=show-log.php%27%22%3E%3CIMG+SRC%3D%22%2F-@18736-61118-665@-%2F_SCAN_.html%22%3E
```

リクエストヘッダ

```
1 : GET http://security-blanket.jp/mutillidae/?page=show-log.php%27%22%3E%3CIMG+SRC%3D%22%2F-@18736-61118-665@-%2F_SCAN_.html%22%3E HTTP/1.1
2 : Accept-language: ja,en-us;q=0.7,en;q=0.3
3 : Keep-alive: 115
4 : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 : User-agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
6 : Accept-charset: Shift_JIS,utf-8;q=0.7,*;q=0.7
7 : Host: security-blanket.jp
8 : Cookie: showhints=0; PHPSESSID=f0cr2nm9d2pg2puiog561ien27
9 : Authorization: Basic eWFyYXJlc2FpdG86bWtfcGVudGVzdA==
```

リクエストボディ

HTTPステータス

```
200
```

レスポンスヘッダ

```
1 : Date: Thu, 08 Nov 2012 04:48:30 GMT
2 : Server: Apache
3 : Logged-In-User:
4 : Vary: Accept-Encoding
5 : Connection: close
6 : Transfer-Encoding: chunked
7 : Content-Type: text/html
```

レスポンスボディ

```
92 : <a href="./index.php?page=login.php">Login/Register</a>
93 : </td>
94 : <td><a href="./index.php?do=toggle-security&page=show-log.php'"><IMG SRC="/-@18736-61118-665@-/_SCAN
     _.html">">Toggle Security</a></td>
95 : <td><a href="setupreset.php">Setup/Reset the DB</a></td>
96 : </tr>
```

該当箇所一覧

No.	URL	Method	パラメータ
1	http://security-blanket.jp/mutillidae/?page=show-log.php	GET	page
2	http://security-blanket.jp/mutillidae/index.php	GET	uid
3	http://security-blanket.jp/mutillidae/index.php	GET	-
4	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	uid
5	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	page
6	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	-
7	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	username
8	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	uid
9	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	page
10	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	-

3.5.特殊文字のエスケープ漏れ
Escape leak

10件

概要

[<,>,",',&] 等の特殊文字いずれかがエスケープされずそのまま出力されています。特殊文字をそのまま出力することにより、クロスサイトスクリプティングが発生する可能性があります。

対策

[<,>,",',&] 等の特殊文字は開発環境などで用意されているエスケープ関数を使用して必ずエスケープしてください。プログラム内の表示直前の処理を確認してください。
・エスケープ対象文字列

【 <> " ' % () & + ;】

検出例

URL	Method	パラメータ
http://security-blanket.jp/mutillidae/?page=show-log.php-@18736-61118-665@-_SC%3EAN__SC%3CAN__SC%22AN__SC%27AN___SC&AN_	GET	page

診断用文字列

```
-@18736-61118-665@-_SC>AN__SC<AN__SC"AN__SC'AN___SC&AN_
```

脆弱性検出文字列

code:	
header:	
content:	<td>AN__SC<AN__SC"AN__SC'AN___SC"> Toggle Security</td>

リクエストライン

```
GET http://security-blanket.jp/mutillidae/?page=show-log.php-@18736-61118-665@-_SC%3EAN__SC%3CAN__SC%22AN
     __SC%27AN___SC&AN_
```

リクエストヘッダ

```
1 : GET http://security-blanket.jp/mutillidae/?page=show-log.php-@18736-61118-665@-_SC%3EAN__SC%3CAN__SC%
     22AN__SC%27AN___SC&AN_ HTTP/1.1
2 : Accept-language: ja,en-us;q=0.7,en;q=0.3
3 : Keep-alive: 115
4 : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 : User-agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
6 : Accept-charset: Shift_JIS,utf-8;q=0.7,*;q=0.7
7 : Host: security-blanket.jp
8 : Cookie: showhints=0; PHPSESSID=cp05uru9qvjdo9o1in5knck4l1
9 : Authorization: Basic eWFyYXJlc2FpdG86bWtfcGVudGVzdA==
```

リクエストボディ

HTTPステータス

```
200
```

レスポンスヘッダ

```
1 : Date: Thu, 08 Nov 2012 04:48:21 GMT
2 : Server: Apache
3 : Logged-In-User:
4 : Vary: Accept-Encoding
5 : Connection: close
6 : Transfer-Encoding: chunked
7 : Content-Type: text/html
```

レスポンスボディ

```
92 : <a href="./index.php?page=login.php">Login/Register</a>
93 : </td>
94 : <td><a href="./index.php?do=toggle-security&page=show-log.php-@18736-61118-665@-_SC>AN__SC<AN__SC"AN
     __SC'AN___SC">Toggle Security</a></td>
95 : <td><a href="setupreset.php">Setup/Reset the DB</a></td>
96 : </tr>
```

該当箇所一覧

No.	URL	Method	パラメータ
1	http://security-blanket.jp/mutillidae/?page=show-log.php	GET	page
2	http://security-blanket.jp/mutillidae/index.php	GET	uid
3	http://security-blanket.jp/mutillidae/index.php	GET	-
4	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	uid
5	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	page
6	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	-
7	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	username
8	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	uid
9	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	page
10	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	-

3.6.HttpOnly属性の欠如
Detect No HttoOnly Flg

2件

概要

HttpOnly属性が設定されていません。
CookieにHttpOnly属性が付加されていない場合、JavaScrptなどによってCookieが送信され
Cookieが盗まれる可能性があります。

対策

httpによる通信以外でCookieを送信する必要がない場合は、
CookieにHttpOnly属性を設定することを推奨いたします。

検出例

URL	Method	パラメータ
http://security-blanket.jp/?%3DPHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000	GET	sessionid=bdb57a16cca993990c677e004d3a4e47

診断用文字列

脆弱性検出文字列

code:	
header:	Set-Cookie: sessionid=bdb57a16cca993990c677e004d3a4e47; Path=/; secure
content:	

リクエストライン

```
GET http://security-blanket.jp/?%3DPHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000
```

リクエストヘッダ

```
1 : GET http://security-blanket.jp/?%3DPHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000 HTTP/1.1
2 : Accept-language: ja,en-us;q=0.7,en;q=0.3
3 : Keep-alive: 115
4 : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 : User-agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
6 : Accept-charset: Shift_JIS,utf-8;q=0.7,*;q=0.7
7 : Host: security-blanket.jp
8 : Authorization: Basic eWFyYXJlc2FpdG86bWtfcGVudGVzdA==
```

リクエストボディ

HTTPステータス

```
200
```

レスポンスヘッダ

```
1 : Date: Thu, 08 Nov 2012 04:49:21 GMT
2 : Server: Apache
3 : Vary: Cookie,Accept-Language,Accept-Encoding
4 : Content-Language: ja
5 : Set-Cookie: csrftoken=3705484a7b45eb8b867afd59f1818eda; Max-Age=31449600; Path=/
6 : Set-Cookie: sessionid=bdb57a16cca993990c677e004d3a4e47; Path=/; secure
7 : Connection: close
8 : Transfer-Encoding: chunked
9 : Content-Type: text/html; charset=utf-8
```

レスポンスボディ

```
1 :
2 :
3 :
4 :
5 : <?xml version="1.0" encoding="utf-8"?>
```

該当箇所一覧

No.	URL	Method	パラメータ
1	http://security-blanket.jp/?%3DPHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000	GET	sessionid
2	http://security-blanket.jp/server-info	GET	Max-Age

3.7.発行されているCookieの情報 Get Cookie Infomation	2件

概要

発行されているCookie情報です。

対策

情報の為、対策はありません。

検出例

URL	Method	パラメータ
http://security-blanket.jp/?%3DPHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000	GET	sessionid=bdb57a16cca993990c677e004d3a4e47

診断用文字列

脆弱性検出文字列

code:	
header:	Set-Cookie: sessionid=bdb57a16cca993990c677e004d3a4e47; Path=/; secure
content:	

リクエストライン

```
GET http://security-blanket.jp/?%3DPHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000
```

リクエストヘッダ

```
1 : GET http://security-blanket.jp/?%3DPHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000 HTTP/1.1
2 : Accept-language: ja,en-us;q=0.7,en;q=0.3
3 : Keep-alive: 115
4 : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 : User-agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
6 : Accept-charset: Shift_JIS,utf-8;q=0.7,*;q=0.7
7 : Host: security-blanket.jp
8 : Authorization: Basic eWFyYXJlc2FpdG86bWtfcGVudGVzdA==
```

リクエストボディ

HTTPステータス

```
200
```

レスポンスヘッダ

```
1 : Date: Thu, 08 Nov 2012 04:49:21 GMT
2 : Server: Apache
3 : Vary: Cookie,Accept-Language,Accept-Encoding
4 : Content-Language: ja
5 : Set-Cookie: csrftoken=3705484a7b45eb8b867afd59f1818eda; Max-Age=31449600; Path=/
6 : Set-Cookie: sessionid=bdb57a16cca993990c677e004d3a4e47; Path=/; secure
7 : Connection: close
8 : Transfer-Encoding: chunked
9 : Content-Type: text/html; charset=utf-8
```

レスポンスボディ

```
1 :
2 :
3 :
4 :
5 : <?xml version="1.0" encoding="utf-8"?>
```

該当箇所一覧

No.	URL	Method	パラメータ
1	http://security-blanket.jp/?%3DPHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000	GET	sessionid
2	http://security-blanket.jp/server-info	GET	Max-Age

3.8.コメントの検出 Detect Commnet	6件

概要

検出したコメントの一覧です。

対策

コメント内にログイン情報、個人情報といった重要な情報が記述されていないか確認してください。

検出例

URL	Method	パラメータ
http://security-blanket.jp/mutillidae/?page=show-log.php	GET	

診断用文字列

脆弱性検出文字列

code:	
header:	
content:	["<!-- I think the database password is set to blank or perhaps samurai.\r\n\t\t\tIt depends on whether you installed this web app from irongeeks site or\r\n\t\t\tare using it inside Kevin Johnsons Samurai web testing framework. \r\n\t\t\tIt is ok to put the password in HTML comments because no user will ever see \r\n\t\t\tthis comment. I remember that security instructor saying we should use the \r\n\t\t\tframework comment symbols (ASP.NET, JAVA, PHP, Etc.) \r\n\t\t\trather than HTML comments, but we all know those \r\n\t\t\tsecurity instructors are just making all this up. -->", "<!-- Begin Content -->", "<!-- End Content -->"]

リクエストライン

```
GET http://security-blanket.jp/mutillidae/?page=show-log.php
```

リクエストヘッダ

```
1 : GET http://security-blanket.jp/mutillidae/?page=show-log.php HTTP/1.1
2 : Accept-language: ja,en-us;q=0.7,en;q=0.3
3 : Keep-alive: 115
4 : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 : User-agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
6 : Accept-charset: Shift_JIS,utf-8;q=0.7,*;q=0.7
7 : Host: security-blanket.jp
8 : Authorization: Basic eWFyYXJlc2FpdG86bWtfcGVudGVzdA==
9 : Cookie: PHPSESSID=qq8f4rifgnbo5ick5vtstff8b6
```

リクエストボディ

HTTPステータス

```
200
```

レスポンスヘッダ

```
1 : Date: Thu, 08 Nov 2012 04:50:15 GMT
2 : Server: Apache
3 : Logged-In-User:
4 : Vary: Accept-Encoding
5 : Connection: close
6 : Transfer-Encoding: chunked
7 : Content-Type: text/html
```

レスポンスボディ

```
1 :
2 : <!-- I think the database password is set to blank or perhaps samurai.
3 : It depends on whether you installed this web app from irongeeks site or
4 : are using it inside Kevin Johnsons Samurai web testing framework.
5 : It is ok to put the password in HTML comments because no user will ever see
6 : this comment. I remember that security instructor saying we should use the
7 : framework comment symbols (ASP.NET, JAVA, PHP, Etc.)
8 : rather than HTML comments, but we all know those
9 : security instructors are just making all this up. -->
10 : <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/1999/REC-html40
     1-19991224/loose.dtd">
11 : <html>
---------------------------------------
385 : <td valign="top">
386 : <blockquote>
387 : <!-- Begin Content -->
388 :
389 :
---------------------------------------
426 : </TR>
427 : </TABLE>
428 : <!-- End Content -->
429 : </blockquote>
430 : </td>
```

該当箇所一覧

No.	URL	Method	パラメータ
1	http://security-blanket.jp/mutillidae/?page=show-log.php	GET	-
2	http://security-blanket.jp/mutillidae/index.php	GET	-
3	http://security-blanket.jp/mutillidae/index.php?do=logout	GET	-
4	http://security-blanket.jp/mutillidae/index.php?do=toggle-security&page=home.php	GET	-
5	http://security-blanket.jp/mutillidae/index.php?page=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php	GET	-
6	http://security-blanket.jp/mutillidae/index.php?page=password-generator.php&username=demo	GET	-

付録

診断カテゴリ概要

クロスサイトスクロプティング

クロスサイトスクリプティングとはWebアプリケーションソフトウェアの脆弱性で、「サイトを跨っ てスクリプトを実行する」という意味です。　Webアプリケーションで、入力されたデータの内容を充分チェックせずにHTML内に出力していると、H TML内にJavaScriptなどの任意のコードを埋め込むことができてしまいます。このような状態を「クロ スサイトスクリプティング脆弱性がある」と言います。　例として、任意のタグがそのまま書き込めてしまう掲示板が挙げられます。悪意あるユーザが「」などのHTMLタグを含む内容を投稿すると、投稿内容を閲覧したときにスクリプトが実行されてし まう危険性があります。スクリプトの内容によってはCookieデータの盗聴や改竄などが可能なため、商 取引に使ったCookieを横取りして、本人に成りすまして物品の購入を行ったり、Cookieを認証やセッシ ョン管理に使用しているサイトに侵入するなど、より広範かつ深刻な被害を与える可能性があります。

SQLインジェクション

「インジェクション(injection)」とは「注入」という意味で、SQLデータベースに対し外部から任意 のSQL文を実行可能な状態を示します。　受ける被害として、あるユーザが他のユーザのデータを見たり、パスワード情報を取得されるが考え られます。また、発行可能なSQL文の種類や設定によってはデータベース内容の改竄や削除、さらには サーバ内で任意のコマンドを実行することが可能な場合があります。

ディレクトリトラバーサル

アプリケーションやシステムが想定している公開ディレクトリを越えて、ディレクトリを遡ることが 可能な状態を示します。本来公開されていないパスワードファイル等のシステムファイルや個人情報を 含んだファイル等が外部に漏洩する可能性があります。　典型的なパターンとしては、「../../../../../etc/passwd」のように「../」を多用してディレクト リを遡りパスワードファイルを取得しようとする攻撃があります。

コマンドインジェクション

外部から任意のOSのコマンドが実行することが可能な状態のことです。ユーザの入力がそのままコマ ンドとして実行可能な個所で使用されている場合に発生します。

強制ブラウジング

意図していないコンテンツが公開ディレクトリ上に存在し、第三者がURLを直接指定することでそれ らのコンテンツが漏洩する可能性のある状態を示します。コンテンツの内容によってはシステム情報や 個人情報の漏洩に繋がることがあります。　例えば、アプリケーションのソースコードやアンケート結果のファイル等が公開ディレクトリにその まま置かれている場合や、Webサーバの設定ミスによりディレクトリの一覧が出力されるものが該当し ます。

HTTPレスポンス分割

ウェブサーバから返されるレスポンスヘッダーを改竄できる状態を示します。レスポンスヘッダーを改竄する事で悪意のあるサイトへの誘導や、偽ページを表示させ個人情報を盗み出すなどの攻撃に利用される可能性があります。送信データをレスポンスヘッダーにそのままセットし、かつ改行コードのエスケープに不備がある場合は攻撃に利用される可能性があります。

Cookie管理の不備

検査対象サイトで発行されているCookieの管理状態に何らかの不備がある状態を示します。例えば暗号化通信（https）で発行されるCookieにSecure属性が設定されていない物が該当します。Secure属性が設定されていないCookieは非暗号化通信（http）でも送信されるため盗聴の危険性があります。盗聴した情報は不正アクセスに利用される可能性があります。

エラーメッセージの検出

検査実行中に検査用リクエスト等でエラーが発生した状態を示します。エラーメッセージによっては使用している製品やバージョンが判明する場合があります。これら情報は攻撃の際に利用される可能性があります。

製品情報の検出

使用している製品の情報が何らかの手段で取得できる状態を示します。例えば、サーバへのリクエストのレスポンスにバージョン情報が含まれている物が該当します。製品のバージョンによっては既知の脆弱性があるため、これらの情報は攻撃の際に利用される可能性があります。

内部情報の検出

HTMLソースコード内に内部ネットワークのIPアドレスやpathなどの公開する必要のない情報が取得できる状態を示します。これらの情報は攻撃の際に利用される可能性があります。

危険度の判定基準

本検査の目的は、検査対象Webアプリケーションに対してリモートから脆弱性の検査を行い、Webアプ リケーションに存在する脆弱性を検出することにあります。また、脆弱性が検出された場合、そのリスク評価、及び、脆弱性への対策を支援する情報の提供も行 います。

危険度	判定基準
緊急	パスワード漏えい、管理者権限昇格など、システム全体に影響する問題です。これらの問題が発生する可能性が極めて高く、即日対応する必要があります。
重大	情報漏洩や、なりすましなど、ユーザ被害が発生する可能性が高い問題です。このレベルには、クロスサイトスクリプティングやSQLインジェクションなどの問題があり、インシデント報告やOWASP TOP10などで上位を占めるセキュリティ上の問題です。このことから、早急に対応する必要があります。
高	総当たり攻撃や認証回避など、セキュリティ上の問題が発生する可能性があります。システムの仕様などにより、セキュリティ上必要な対策が実施されていない場合このレベルに分類されます。問題が発生する可能性があるため、対応を必ず行うことを推奨します。
中	システムの設定情報や管理情報の漏洩等、システムに対する攻撃手段を提供する可能性がある問題です。直接被害が発生する可能性は高くないですが、他のセキュリティ上の問題と組み合わさるとレベルが上がる可能性があります。問題になる可能性があるため対策を検討してください。
低	バージョン情報表示や、バナー情報表示など、攻撃者の興味を引く可能性のある問題です。直接悪用されるよりは、このレベルの情報から攻撃手法を絞っていくことがあります。予防するうえで対策を検討してください。
情報	品質やセキュリティのさらなる向上のために弊社が推奨する項目です。

判定基準はあくまでも目安であり、脆弱性の検出された箇所・内容等により判定基準とは異なる危険 度を脆弱性に与えることもございますので、あらかじめご了承ください。

評価基準

本報告書における総合評価は、以下に規定される絶対評価によるものです。絶対評価は、A、B、C、D のいずれかのアルファベット1文字で表記され、検査結果を絶対評価の評価基準に照合し適合するクラ スが評価として与えられます。

評価レベル	評価基準	基準検出件数
A	早急に対策が必要な脆弱性は検出されませんでした。	検出件数0件
B	直接的に被害を受ける可能性は低いと推測されますが、脆弱性が確認されております。検出内容を確認の上、対策の検討を行うことを推奨します。	危険度高以上の脆弱性は1件も検出されていないが、危険度中以下の脆弱性検出件数は1件以上
C	被害を受ける可能性のある脆弱性が確認されております。早急に対策の検討を行うことを推奨します。	危険度緊急以上の脆弱性は1件も検出されず、危険度高の脆弱性が1件以上検出
D	大きな被害を受けることが懸念される危険性の高い脆弱性が確認されております。早急に対策を行うことを推奨します。	危険度緊急の脆弱性が1件以上検出

なお、上記評価基準は本検査において検出された脆弱性の検出件数を基に、検査結果を簡潔に表現す るために作成された独自基準になります。　上記評価基準による評価は、あくまでも検査結果を簡潔に表現するためのものであり、弊社は評価に 対しての保証や責任は負いかねますので、あらかじめご了承ください。

お問合せ先

株式会社M&K
info@m-kcompany.co.jp