

Contents

はじめに	003
Part1 Webアプリケーションのセキュリティとは	009
1-1 攻撃方法を学ぼう	010
1-1-1 Webアプリケーションの脆弱性とは	010
1-1-2 Webアプリケーションの特殊性とは	011
Column 本書を読む上での前提知識	011
Column サーバサイドスクリプト言語とクライアントサイドスクリプト言語	012
1-1-3 Webアプリケーションへの攻撃方法を知ること	013
1-1-4 Webアプリケーション側の防御方法を、体系的・網羅的に学習すること	013
1-1-5 チェックシートで総ざらえする	014
1-2 下位レイヤーでのセキュリティ維持	015
1-2-1 防御対策	015
1-2-2 管理者権限を持っている場合	015
Column サーバサイドセキュリティとクライアントサイドセキュリティ	018
Part2 Webアプリケーションを実際に攻撃してみよう	019
2-1 クラッキングの実体験	020
2-1-1 攻撃方法をなぜ学ぶのか	020
2-1-2 攻撃目標となる脆弱なゲストブックスクリプト	020
2-1-3 サンプルスクリプトのプログラムとしての流れ	023
2-2 攻撃方法と防御方法	026
2-2-1 攻撃してみよう (Script Insertion編)	026
2-2-2 防御してみよう (Script Insertion編)	028
Column HTML出力用サニタイズ	029
2-2-3 攻撃してみよう (XSS編)	030
Column クッキー	033
2-2-4 防御してみよう (XSS編)	034
2-2-5 攻撃してみよう (SQL Injection編)	036
2-2-6 防御してみよう (SQL Injection編)	037
Column SQL用サニタイズ	038
Column データベースエンジンによる差異	040
2-2-7 攻撃してみよう (CSRF編)	041
2-2-8 防御してみよう (CSRF編)	043
2-2-9 脆弱性をすべて対策したゲストブックスクリプト	045
Column クッキー表示のサニタイズ	048

Part3 攻撃方法 14種類 総ざらえ	049
3-1 リンクを踏ませてJavaScriptを実行させる	
XSS (クロス・サイト・スクリプティング)	050
3-2 コンテンツ内にJavaScriptを埋め込む	
Script Insertion (クライアント・サイド・スクリプト埋め込み攻撃)	054
3-3 データベースへの問い合わせを乗っ取る	
SQL Injection	058
SQL Injection対策の失敗例①	061
SQL Injection対策の失敗例②	062
3-4 意図しない操作を強いる	
CSRF (クロス・サイト・リクエスト・フォージェリー)	064
3-5 ファイルなどのチェック機構をすり抜ける	
ヌルバイト攻撃	068
3-6 想定外のディレクトリにアクセスする	
Directory Traversal (ディレクトリ遡り攻撃)	072
Directory Traversal (ディレクトリ遡り攻撃)対策の失敗例①	076
Directory Traversal (ディレクトリ遡り攻撃)対策の失敗例②	077
3-7 セッション変数等を外部から操作する	
変数汚染攻撃	078
変数汚染攻撃対策の失敗例①	082
変数汚染攻撃対策の失敗例②	083
3-8 不正なHTTPレスポンスを読み込ませる	
HTTPレスポンス分割攻撃	084
3-9 任意のファイルを読み込み実行する	
インクルード攻撃	088
3-10 任意のPHPコードを実行する	
eval利用攻撃	092
3-11 任意の外部コマンドを実行する	
外部コマンド実行攻撃	096
3-12 アップロードしたファイルを実行する	
ファイルアップロード攻撃	100
ファイルアップロード攻撃対策の失敗例	103
3-13 第三者のセッションを乗っ取る	
セッションハイジャック	104
3-14 迷惑メールの送信サーバとして利用する	
スパムメール踏み台攻撃	108
Column Internet Explorerのバグと仕様	112
 Part4 Webアプリケーションセキュリティの理論	113
4-1 セキュリティに関する基本トピック	114
4-1-1 フォーム値の誤解	114
4-1-2 ドキュメントルートの内側と外側	116
4-1-3 エラー制御とPath Disclosure	117
Column 新エラーレベル E_STRICT	120
4-1-4 ホワイトリスト法とブラックリスト法	121

4-1-5	攻撃を検知した時の対応	122
4-1-6	パスワードクラック	124
4-1-7	SSLの効果	124
4-1-8	DoSアタック	125
4-1-9	投稿の修飾機能	127
4-1-10	外部画像の表示許可	129
	Column 任意のFlashを表示するという事	131
4-2	PHPセッション	132
4-2-1	PHPセッション関連の実行時設定	132
4-2-2	セッションを利用したXSSとHTTPレスポンス分割	136
4-2-3	SSL利用時のセッション鍵の取り扱い	137
4-2-4	セッション鍵の変更	138
4-3	データステータス	139
4-3-1	データステータスとは	139
4-3-2	「用途」というステータス	140
	Column 本書のサンプルコードについて	143
4-3-3	HTML出力用サニタイズの注意点	144
	Column UTF-7 XSS	145
Part5 HTTPセッションから攻撃の本質を知る		147
5-1	HTTPにtelnetで攻撃する	148
5-1-1	HTTPとは	148
5-1-2	HTTPをtelnetで知る	148
	Column WindowsXPでのtelnet	148
5-1-3	PHPスクリプトとHTTP	151
5-2	HTTPセッションで見るXSS	154
5-2-1	XSSリクエストの送信	154
5-2-2	XSSについての考察	156
5-3	HTTPセッションで見るScript Insertion	157
5-3-1	JavaScriptを埋め込んだHTTPリクエストの送信1	157
5-3-2	より一般的なHTTPリクエストとCOOKIE	158
5-3-3	JavaScriptを埋め込んだHTTPリクエストの送信2	159
5-3-4	HTTPリクエストにおけるPOSTメソッド	160
5-3-5	JavaScriptを埋め込んだHTTPリクエストの送信3	162
5-4	HTTPセッションで見るファイルアップロード攻撃	163
5-4-1	\$_FILESの理解	163
	Column multipart/form-data FORMでのPOSTデータ	165
5-4-2	不正なファイルアップロード	166
	Column \$_FILES[]各パラメータに対する「ヌルバイト攻撃」と「ディレクトリ遡り攻撃」	168
5-5	HTTPセッションで見るHTTPレスポンス分割攻撃	169
5-5-1	header() 関数の動作	169
5-5-2	レスポンスヘッダ Location	170
5-5-3	HTTPレスポンスの分割	171
	Column HTTPレスポンスヘッダの隠蔽	173
5-6	HTTPセッションで見るPHPセッションへの攻撃	174

5-6-1	セッション鍵の漏洩	174
5-6-2	不正なセッション鍵の埋め込み	176
Part6	脆弱性スキャナーを利用する	179
6-1	クライアント型脆弱性スキャナーを利用する	180
6-1-1	脆弱性スキャナーとは	180
6-1-2	Max Patrolのインストール	181
6-1-3	スキャン対象の準備	181
6-1-4	Max Patrolの実行	182
	Column TCP 80番ポートのみを指定する理由	183
6-1-5	スキャン結果の確認	183
	Column MaxPatrolに残るスキャン結果について	186
6-2	プロキシ型脆弱性スキャナーを利用する	187
6-2-1	プロキシ型脆弱性スキャナーとは	187
6-2-2	Parosのインストール	187
6-2-3	Parosでの脆弱性スキャン	189
6-2-4	ParosのTrap機能	192
Part7	チャート式：脆弱性の見つけ方	193
7-1	ソースコードが手元にないときの脆弱性の見つけ方	194
7-1-1	既知のWebアプリケーションかどうかを調べる	194
7-1-2	スキャナーアプリケーションの利用	194
7-1-3	URIを精査しGETでつづく	195
7-1-4	telnetで探る	196
7-1-5	実際に攻撃してみる	197
7-1-5-1	XSS (クロス・サイト・スクリプティング)	197
7-1-5-2	Script Insertion (クライアント・サイド・スクリプト埋め込み攻撃)	197
7-1-5-3	SQL Injection	197
7-1-5-4	CSRF (クロス・サイト・リクエスト・フォージェリー)	198
7-1-5-5	ヌルバイト攻撃	198
7-1-5-6	Directory Traversal (ディレクトリ遡り攻撃)	198
7-1-5-7	変数汚染攻撃	198
7-1-5-8	HTTPレスポンス分割攻撃	198
7-1-5-9	インクルード攻撃	199
7-1-5-10	eval利用攻撃	199
7-1-5-11	外部コマンド実行攻撃	199
7-1-5-12	ファイルアップロード攻撃	199
7-1-5-13	セッションハイジャック	200
7-2	ソースコードが手元にあるときの脆弱性の見つけ方	201
7-2-1	既知の脆弱性を調べる	201
7-2-2	eval()とpreg_replace()のgrep検索	201
	Column 関数の検索方法	201
	Column phpBBという名の反面教師	202
7-2-3	includeとrequireのgrep検索	203
7-2-4	ファイル名指定関数のgrep検索	204

7-2-5	外部コマンド実行関数のgrep検索	204
7-2-6	スーパーグローバル変数展開のgrep検索	204
7-2-7	\$_FILESのgrep検索	205
7-2-8	session_start()関数のgrep検索	205
7-2-9	mail()関数のgrep検索	205
7-2-10	致命的な操作のピックアップ	206
7-2-11	query発行関数からの遡上	206
7-2-12	スーパーグローバル変数の追跡	207
Column	セキュリティ最新情報の取得方法	208
Column	攻撃されてしまったら	208
Appendix		209
A-1	PHPの設定変更方法	210
A-2	各変数の改竄可能性一覧表	217
A-3	先頭プロテクタ	219
A-4	Webアプリケーションセキュリティ「ガセ情報の沼」	221
A-5	クラッキングを体験するテスト環境をつくる	224
Index		234

