

Webアプリケーション監査 ～攻撃者の観点から～

2005/12/22

国分裕(bun@devnull.jp)

Template designed by harupu

1

Agenda

- 1 攻撃する人は何をしてくる？
- 1 守るためにはどうすればよい？

2

検査の手順

3

検査手法

- a ブラックボックステスト
- a ホワイトボックステスト
- a グラスボックステスト

4

検査手順

- ④ 大きく分けて2工程
 - 調査
 - 検査

の繰り返し

5

調査

- ④ 通常のユーザのように巡回
 - どんな機能があるのか？
 - そのサイトで扱っている情報は何か？
 - 開発言語は？
- ④ HTMLコメント

```
お申込は締め切りました。  
<!--  
<a href="entry.asp">お申込はこちら</a>  
-->
```

- ④ Google？

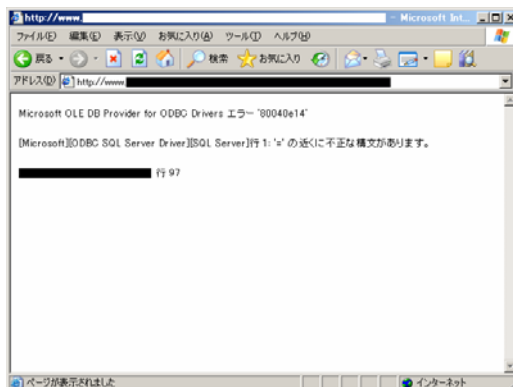
6

エラーメッセージ

- ④ 適当な文字を入れてみる
 - 記号
 - 数字→アルファベット
 - 空
 - パラメータごと削除
- ④ 嫌がりそうな操作
 - 多重登録
 - 画面遷移無視

7

システムが出力するエラー



- Appサーバの種類/バージョン
- DBの種類/バージョン
- 発生した例外の種類
- ソースコードのファイル名
- 使用しているクラス名
- エラーが起きた行番号

8

アプリケーションが出力するメッセージ

- a 「ユーザ名が間違っています」
- a 「パスワードが間違っています」
→ ユーザIDは当たってるんだ (´_ゝ`)フーン

- a 「そのメールアドレスは既に登録されています」
→ 登録者リスト作れるかもな (´_ゝ`)フーン

9

検査

調査結果を基に怪しいところから

- a XSS
- a SQL Injection
- a Directory Traversal
- a Session Hijack
- a CSRF(Cross Site Request Forgeries)

:

10

検査

基本的に try & error

```
<script>alert();</script>
```

```
"><script src=http://example.com/hoge.js>  
a" onerror="alert();
```

想定とは別のエラーが発生する場合も

11

検査

決まったパターンをとりあえず試すツール

- Nikto (<http://www.cirt.net/code/nikto.shtml>)
- AppScan (<http://www.watchfire.com/>)
- WebInspect (<http://www.spidynamics.com/>)
- ScanDo (<http://www.kavado.com/>)
- Hailstorm (<http://www.cenzic.com/>)
- Burp intruder (<http://www.portswigger.net/>)
- その他自家製？

12

検査

- ⌚ お仕事検査はここまでで終了
- ⌚ でも悪意のある攻撃者は...
 - 実際に被害を起こせるまでがんばる
 - 似た作りの別のアプリケーション/サイトへ

13

対策

14

攻撃に対する耐性

- ④ 入力値チェックとサニタイジング
 - 6割くらいはこれで防げる
- ④ 認証やセッション管理の見直し
 - 3割くらいはこれで防げる

でも、漏れる orz

参考:「安全なWebアプリ開発の鉄則 2004」
<http://www.soi.wide.ad.jp/class/20040031/slides/09/>

15

緩和

- ④ 調査を邪魔しましょう
 - 有益なエラーメッセージを出力しない
 - ↳ デバッグ用のメッセージは、利用者見せる意味がない
 - ↳ デフォルトのエラー出力先をファイルへ
- ④ 攻撃を邪魔しましょう
 - 画面遷移の厳密な管理
 - エラーが発生したら強制ログアウト
 - 過度な対策は不便になるので注意
- ④ WAF?

16

Q&A

17