Deloitte.

OWASP & .NET

Arris Huijgen

Security Consultant, Deloitte

AHuijgen@deloitte.nl

www.deloitte.nl

Agenda

- OWASP
- Injection
- Authentication & Session Management
- Cross-site scripting

Agenda

- **OWASP**
- Injection
- Authentication & Session Management
- Cross-site scripting

OWASP

The Open Web Application Security Project

https://www.owasp.org

OWASP

1. Injection
2. Broken Authentication and Session Management
3. Cross Site Scripting (XSS)
4. Insecure Direct Object References
5. Security Misconfiguration
6. Sensitive Data Exposure
7. Missing Function Level Access Control
8. Cross Site Request Forgery (CSRF)
9. Using Known Vulnerable Components
10. Unvalidated Redirects and Forwards

Agenda

- OWASP
- **Injection**
- Authentication & Session Management
- Cross-site scripting

Injection

```
SELECT * FROM orders WHERE OrderID = 12345;
```

```
SELECT * FROM orders WHERE OrderID =
    12345; DROP TABLE orders;
```

Injection

- ```
  var results = ExecuteQuery("
  SELECT * FROM accounts WHERE
  custID='"+Request.QueryString["id"]
  +"'");
  ```

- ```
  http://example.com/AccountView.aspx
  ?id=' or '1'='1
  ```

- ```
  var results = ExecuteQuery("
  SELECT * FROM accounts WHERE
  custID='' or '1'='1'");
  ```
  ```

Injection

- Bad

```
conn = pool.getConnection();

string sql = "SELECT * FROM user
WHERE username='" + username +"'");

stmt = conn.createStatement();

rs = stmt.executeQuery(sql);
```

Injection

- Good

```
conn = pool.getConnection();

string selectStatement = "SELECT * FROM user
WHERE username = ?";

PreparedStatement prepStmt =
conn.prepareStatement(selectStatement);

prepStmt.setString(1, username);

ResultSet rs = prepStmt.executeQuery();
```

Injection

- Command Injection
```
Process.Start("find", args[0]);
```

- LDAP Injection
```
string ldapQuery = "cn=" + getParameter("userName")
+ ", ou=Users";
```

- HTTP Response Header Splitting
```
string referrer = Request.QueryString["page"];
Response.Redirect(referrer);

GET /redirect?page=%0D%0A%0D%0AHTTP/1.1 200
OK%0D%0AContent-Type:
text/html%0D%0A%0D%0%3Chtml%3Ehello%3C/html%3E

HTTP/1.1 302 Found        →     HTTP/1.1 200 OK
Server: Apache                  Content-Type: text/html
Location:                       <html>hello</html>
```

Injection

Demo time!

Agenda

- OWASP
- Injection
- **Authentication & Session management**
- Cross-site scripting

Broken Authentication & Session Management

Broken Authentication & Session Management

- Implementation;
- Authentication credentials not encrypted when stored;
- Credentials or SessionIDs sent over an unencrypted channel.

Broken Authentication & Session Management

- Session ID in Link: http://example.com/sale/saleitems.aspx?sessionid=2P0OC2JSNPSKHCJUN2JV
- No session time-out;
- Session mixing;
- Session fixation.

Broken Authentication & Session Management

- Regenerate session at login/logout:

```
Session.Abandon();
Response.Cookies.Add(
    new HttpCookie("ASP.NET_SessionId", ""));
```

- Session timeout:

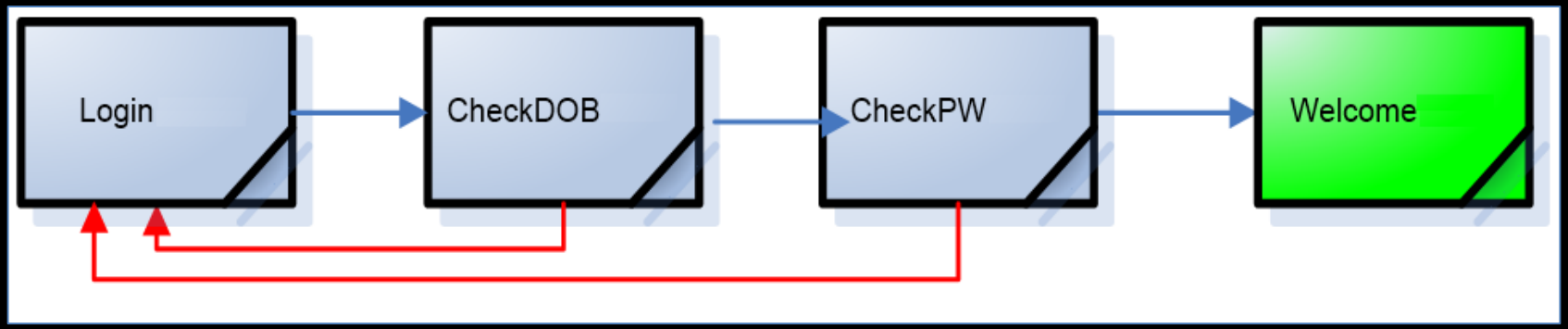
```
<system.web>
    <sessionState timeout="20"></sessionState>
</system.web>
```

- Logout:
  - Invalidate session;
  - Destroy cookie.

Broken Authentication & Session Management

Example

- Customers need to authenticate in two steps:
  - Provide username and Date of Birth;
  - Provide password.



Broken Authentication & Session Management

Login

```
Session.Abandon();
add(new LoginForm("loginForm"));
```

CheckDOB

```
string username = Request.Form["username"];
Session["username"] = username;
If (DoB_Correct(username, Request.Form["DoB"])) {
  add(new PasswordEntryForm("passwordForm"));
} else {
  add(new LoginForm("loginForm"));
}
```

CheckPW

```
If (PW_Correct(Session["username"], Request.Form["pw"])) {
  Session["loggedIn"] = true;
  add(new HomePage("WelcomeForm"));
} else {
  add(new LoginForm("loginForm"));
}
```

Agenda

- OWASP
- Injection
- Authentication & Session Management
- **Cross-site scripting**

Cross-site Scripting (XSS)

Cross-site Scripting (XSS)

- Reflected:

```
http://website.com/Default.aspx?title=
<script>alert('XSS')</script>
```

- Stored:

Cross-site Scripting (XSS)

XSS may lead to:

- Hijacked Sessions;
- Insertion of Content;
- Defacement;
- Redirecting.

Cross-site Scripting (XSS)

Prevention:

- Properly escape untrusted data:
  - HTML Entities (< > ");
- Whitelist Input Validation;
- Auto-Sanitization Libraries;
- Content Security Policy (CSP);
- HTTPOnly flag.

Cross-site Scripting (XSS)

- XSS:
  - Context specific;
  - Browser specific.
- Use Anti-XSS library: Microsoft AntiXSS.
- Additionally; enable Anti-XSS components.

Cross-site Scripting (XSS)

AntiXss

- LDAP encoders
- CssEncode()
- HtmlEncode() (shortcut in MVC <%: Model.Value %>)
- HtmlAttributeEncode()
- UrlEncode()
- HtmlFormUrlEncode()
- XmlEncode()
- XmlAttributeEncode()
- JavaScriptEncode()
- VisualBasicScriptEncode()
- But AntiXss does not offer:
  - JSON encoding.

Cross-site Scripting (XSS)

Notes:

- ‘Split’ attacks.

```
add(new Label("lblHello", "hello " + user.FirstName
+ " " + user.LastName));

user.firstName = "Jan<script type=\"javascript\"";
user.lastName = ">alert(\"XSS\");</script>Klaassen";
```

- HttpOnly cookie flag.

Injection

Demo time!

Thank you!

AHuijgen@deloitte.nl

www.deloitte.nl

Trainings

Want to know more about security/hacking?

Follow one of our HackLab trainings:

HackLab: Demo editie - 26 & 27 maart - Zwolle

HackLab: Hands-on Hacking - 7-11 april - Amsterdam

HackLab: Malware Analysis - 14-16 april – Amsterdam

www.deloitte.nl/academy

AHuijgen@deloitte.nl

dev Net Noord

Deloitte.

Bzz.

You've heard about application security.
Now make it visible.

www.owasp.org

AHuijgen@deloitte.nl

www.deloitte.nl

Deloitte.