# **If you’re not setting content disposition**

IE 8,9,etc will all run HTML content.

Start doing it immediately.

# **Be wary accepting .HTM/HTML files**

Can run scripts in IE 8. Mitigate with X-Download-Options

# **If you’re supporting IE 8, be very wary of file uploads**

Content disposition set.

Accepting .HTML files? You can run scripts in IE 8.

If the server returns a content type it doesn’t know (application/octet-stream) or it is (text/plain), IE 8 will content type sniff.

We can upload .zip/.txt file and IE 8 will treat it as a .htm file.

Check mime type/file extensions

# Other gotchas

# AutoComplete in a browser

# Cross domain javascript sources

Cache-control / pragma

Corss site request forgery

How to work around?

**Error handling in MVC**

Validate input, encode output.

How to ensure consistent validation?

Use of ModelState.IsValid. White list rather than black list.

**Cross site request forgeries**

Start thinking – how can I make this impossible to screw up?

* RedirectUrl

Apply

* It might seem that you could rely on the checking the Referer to prevent this attack, but some proxy servers etc… will strip out the Referer field in order to maintain privacy. Also, there may be ways to spoof the Referer field.
* Another mitigation is to constantly change the URL used for performing sensitive operations like this. AntiForgeryToken, but apply it globally.

**Next step: Hijax**

Don’t rely on ModelState.IsValid everywhere. By convention – use AJAX, validate all input on non-GET actions. Return standard JSON model.

**Consistency and conventions are king – don’t rely on remembering to do it.**