**SQL injection**

solution : cast inputs

**XSS**

injecting scripts into inputs

solution :

[ValidateInput(True)]

HTML.Encode

go to anti cross-site scripting Library

**CSRF**

*scenario 1:*

after the user connects to the server, gets a cookie saved on computer

user brows Evil.com and scripts will be download

since then evil.com use cookie and impersonate as you

*scenario 2: [basically repeating request by tem,,pered data]*

1-temper hidden form data

2-before enpoint of fiddler

3-QS

solution : need to make server sure that the request had been initiated by app not evil

@

{

using(HTML.BegiForm()

{

@HTML.AntiModelInjectionFor(x=>x.usrId)

@HTML.AntiForgeryToken()

}

}

and in function side

[ValidateAntiModelInjectionAttribute("usrId")]

[ValidateAntiForgeryToken()]

public function ....

left:

hqck-proffing > enc> config

email confirmation protocol

use 2 door scenarion for killing connection inn deployment

control session Exit by blowser

put access control to resources and Methods

**application sctucture:**

components can have access to comp and modules but modules to other modules

**Design pattern**

1-Adapter

using interface [multiple implementation and fast port definition]

Client > Adapter > component

use yield to generate filtered on-the-fly repository with different format possibly

use stack for track change

use built in interfaces [system understands and require them] and override , update their behaviour

in test use fake repository

FF cant handle concurrencies in updates > use SP

every RESORCE that needs to be shared needs to handle its own concurrency and DL issues

in database level use numbering for handling deadlocks, basically have a sketck on how resources behave

**.NET**

async : resolves IO latency

parallel : devide work load and make a better use of CPU

toList() makes IEnumerable repository in-memory

pass well typed obkects to MVC views

in using "using" to resolve lazyloading use "include"

in EF Mix of objects and other ones need to be built by hand

Area > actionLink('x','x2','x3',new{area='x'},null)

html.action()

renderPartial()

renderAction()

**TPL:**

already we had :

thrreads , async(delegate invocation) , event based async, Queue user work item

thread is dedicated to task till completes

async ops open other thraed and call them ats block begins and wait for it as block ends

**race condition Solution :**

1-bring shared block into local (best)

2-thread safe entities

3-sync on shared resources by LOCK (worse)

**HTML:**

iframe means having access to other systems domain if allowed and possible to inject inputs and using other domain system send inputs to that domain

**tools for performance:**

profiler,yslow,perfmon,loadrunner,app center test,webapp stress tool,vs load test

**Note OOP**

polimorphysm does not change the real object although do changes in gateways