CSI3670
Web Server (IIS)

Erik Fredericks (fredericks@oakland.edu)

Web Services

W3C -- “A software system designed to provide a standard means of interoperating between different software applications, running on a variety of platforms and frameworks.”

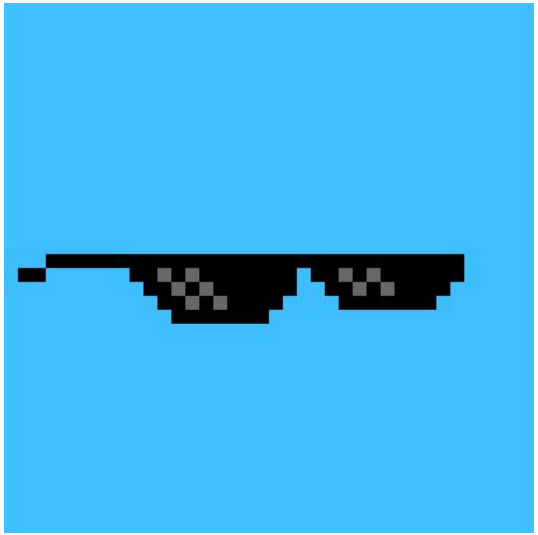
So, a website is effectively an instance of a web service

Let’s Talk about the Internet

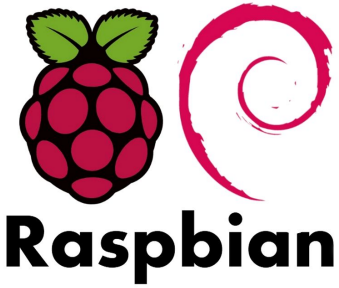
https://www.youtube.com/watch?v=iDbyYGrswtg

Yes, I am subjecting you to another IT Crowd clip

Deal with it



How are web servers different from application servers?



Interoperability

- Client access from:
  - Different devices (Computer, tablet, phone, etc.)
  - Different platforms (Windows, Linux, OSX, Raspbian, etc.)

- Via HTTP, XML, various web languages

Extensibility

- Web services support extensions
  - RSS, ASP, PHP, Perl, etc.

And..

Loose coupling

Individual service components not aware of others

E.g., a web service doesn't necessarily care about how a RDBMS pulls out the data

Only needs it to come out and maybe go back

However, we need a foundation

IIS ➔ Windows, Apache/NGINX ➔ Linux (or Windows)

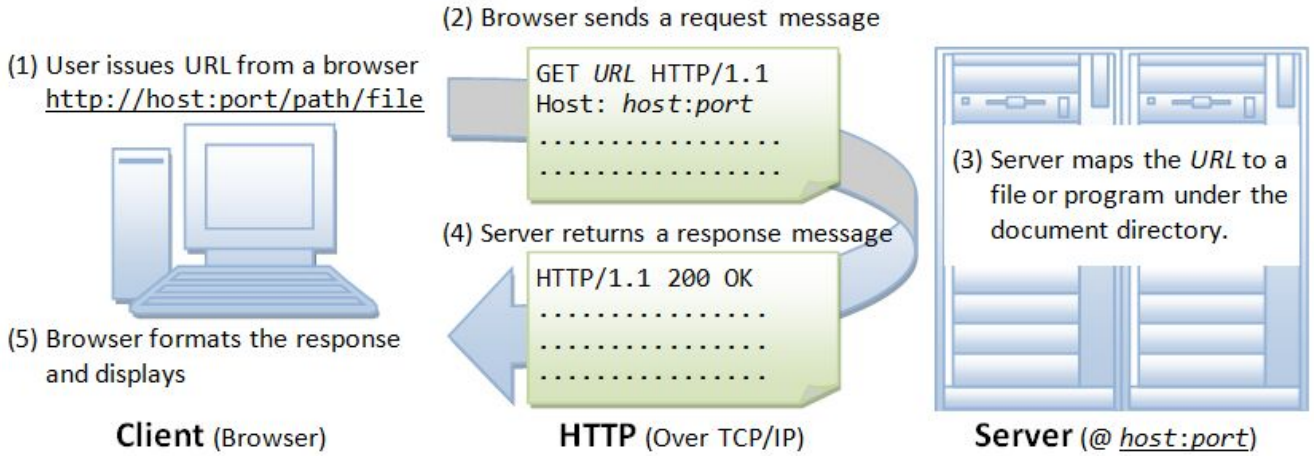
Enter IIS

Windows server application ➜ serves web content

Serve HTML pages, graphics (JPG, GIF, etc), server-side files (ASP, PHP, etc)

HTTP ➜ protocol that defines communications between server and client

HTTPS ➜ HTTP encrypted by SSL



Enter IIS

Web server side of the client/server paradigm

Uses *application pools* (worker processes)

- Each website hosted in IIS gets its own application pool
- (Similar to how Apache kicks off a thread for each client connection)

IIS Components

HTTP stack ➔ HTTP.sys

- Kernel-level driver that listens to HTTP traffic
- Serves HTTP request and directs towards worker processes
- Does not actually process any request

Worker processes (w3wp.exe)

- Handles actual processing at user-level
- Runs application code (ASP.NET, XML web services, etc.)
- Generates static page after processing

IIS Components

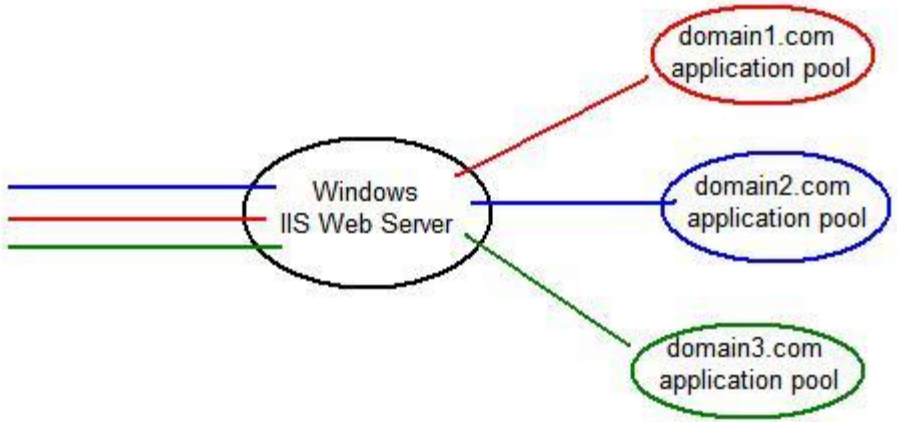
ISAPI (Internet Server Application Programming Interface)
- API to **extend** IIS functionality
- E.g., any *.aspx pages are handled by an ISAPI extension
  - Virtual extension that points to aspnet_isapi.dll

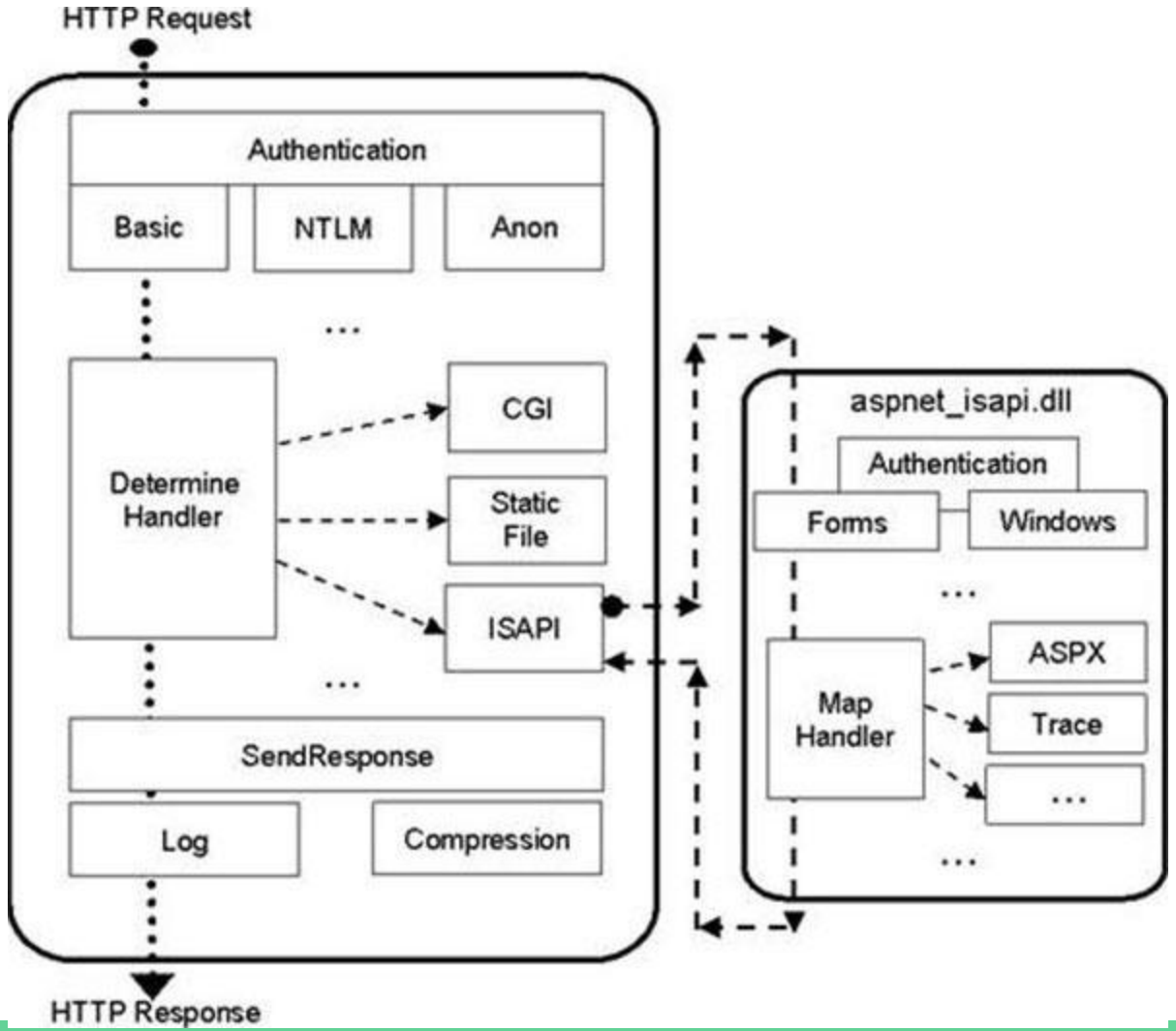
Inetinfo.exe
- User-mode component that describes IIS metadata
- Non-web services such as FTP
- Manages ISAPI applications

IIS Components

Application pools

- Group of web apps / websites
- Each application in a pool shares the worker process (w3wp.exe)
- Process boundary exists between different application pools
  - E.g., web app 1 in application pool 1 can't impact web app 2 in application pool 2




HTTP Request
Authentication
Basic
NTLM
Anon
...
Determine Handler
CGI
Static File
ISAPI
...
SendResponse
Log
Compression
HTTP Response
aspnet_isapi.dll
Authentication
Forms
Windows
...
Map Handler
ASPX
Trace
...
...

Installing IIS

Add Role ➔ Web Server (IIS)
(Include FTP feature if you want)

Configuring IIS

Things we care about (possibly)

Protocol listeners: receive client request

Primary listener is HTTP (HTTP.sys)

Provides listeners for HTTP and HTTPS

Additional can be provided by modules

Services:

WWW publishing service (HTTP listener)

Windows Process Activation service (processing ➔ manages application pools)

Adding an additional listener

Possibilities:

- Update the configuration (add Gopher listener -- packet listener written in Go):
  - Generally `C:\Windows\System32\inetsrv\config\applicationHost.config`

```
<system.applicationHost>
   <listenerAdapters>
      <add name="gopher"
         protocolManagerDll="%SystemRoot%\system32\inetsrv\gophersvc.dll"
         protocolManagerDllInitFunction="GopherInit" />
   </listenerAdapters>
</system.applicationHost>
```

- Other possibilities (code-based, command line-based, etc.): https://www.iis.net/configreference/system.applicationhost/listeneradapters/add

Web Server Security

Two major facets: authentication and encryption

Authentication

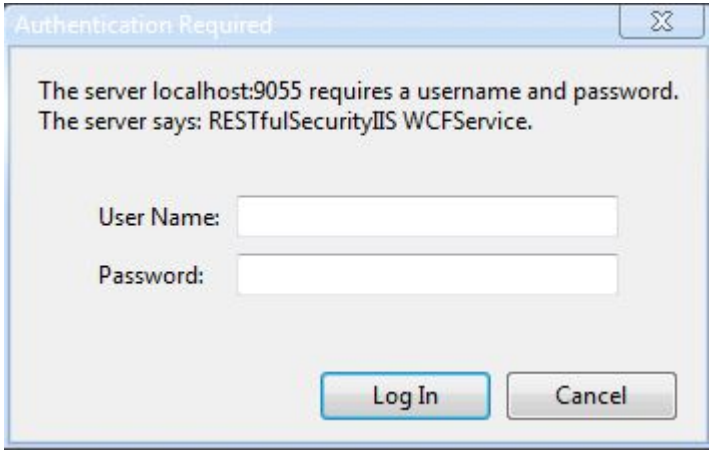
- Proving that you are who you are
- Not authorization ➜ granting access to resource/object
- Common methods:
  - Password/PIN
  - Smart card/key
  - Biometrics

Authentication

Authentication scheme should reflect security needs
- Don’t unnecessarily burden users

Sometimes a single method is not sufficient
- Multifactor authentication uses > 1 credentials
- E.g., password + texted PIN

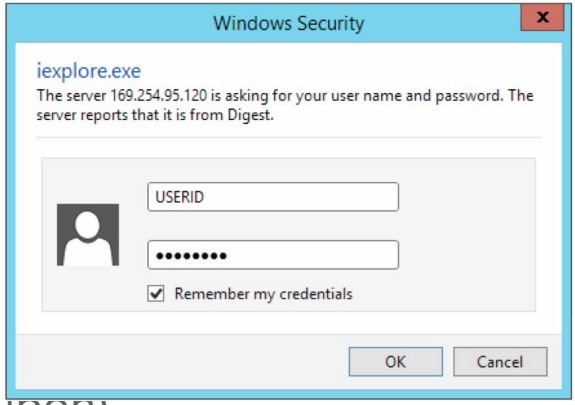
IIS Authentication (5 forms)



Anonymous
- Website is public access
- No credentials

Basic
- Usernames and passwords
- Should enable SSL
  - If not, easily interceptable via something like Wireshark

IIS Authentication



Digest

- Based on secret keys
- Not sent across network (basic is easily eavesdropped-upon)
- Use when full encryption not required

Forms

- Web form used for login
- E.g., Sharepoint, Exchange Web App, etc.

IIS Authentication

Windows

- Uses AD DS
- Server/client will most likely use Kerberos (network authentication protocol)
- NTLM possible but deprecated

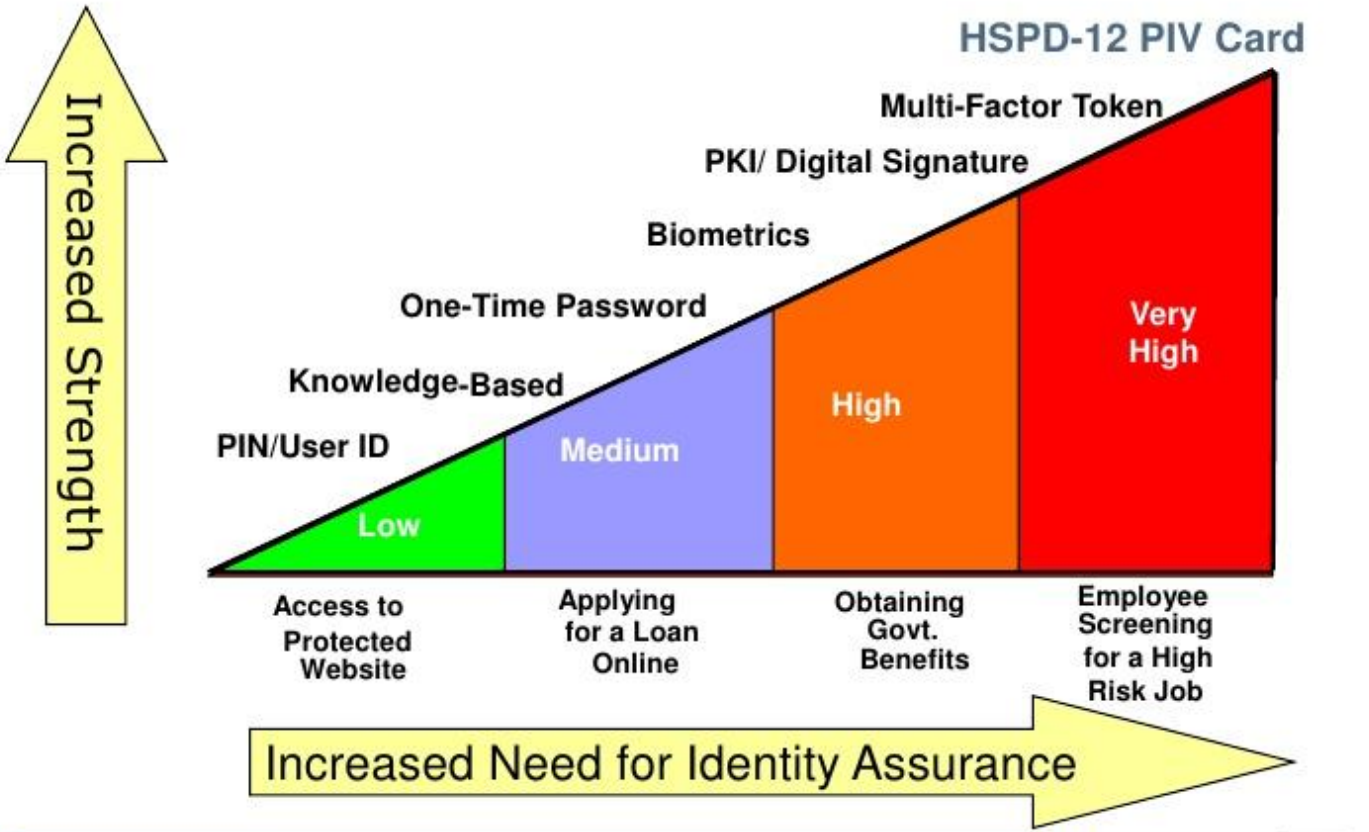
Authentication

Generally a good idea to secure web traffic if important information is flowing
Pick appropriate level of security
    If public-facing, probably not going to go with Windows authentication
        In this case, use SSL encryption

    If intranet-based, can use Windows auth

    If it's not horribly important, can use basic / digest authentication
        Basic tends to have the username/password stored in **cleartext**

E-Authentication Assurance Levels (OMB M-04-04)



Encryption

Converting data to unreadable state

Unreadable ➜ Ciphertext

Readable ➜ Plaintext/Cleartext

Example of encryption using a private key of 3

original_data = 108

Formula: (original_data / private_key) + (3 * private_key)

108 / 3 + (3 * 3) = 45

How to recover data?

- Need algorithm + key
- (45-(3*3))*3 = 108

IIS Encryption

SSL encryption between server and clients using HTTPS protocol

Requires certificates

Certificate → “digitally-signed statement containing information about entity and its public key” (Dictionary of Information Security, 2006)

Generated internally if public key infrastructure (PKI) setup

Usually for **intra**net

Or purchased through third-party

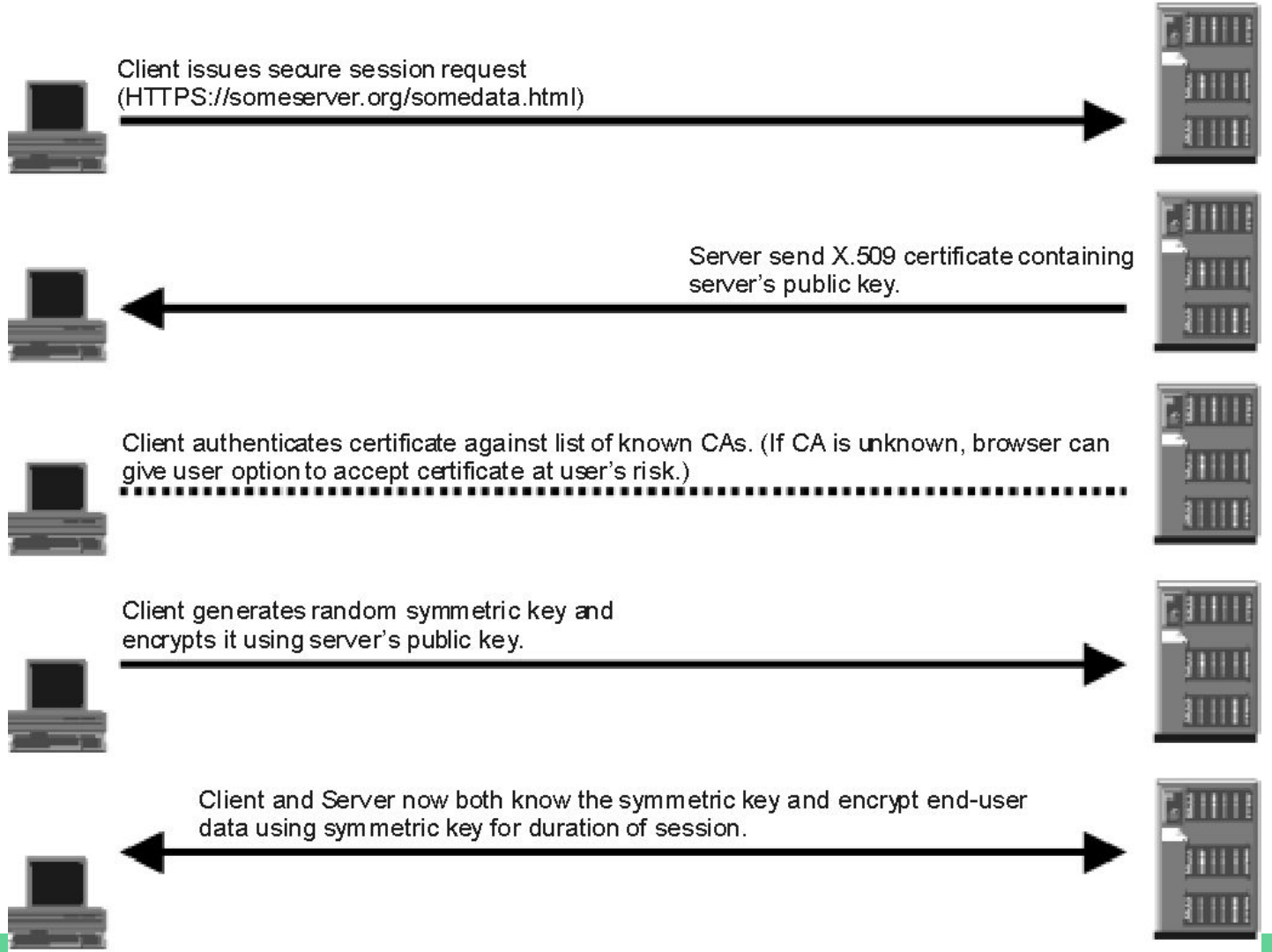
Usually for **inter**net

E.g., VeriSign, Thawte

Self-Signed Certificate

Option for web servers

- Users receive warning first time (like when you go to vcenter)
- After accepting, the site is trusted

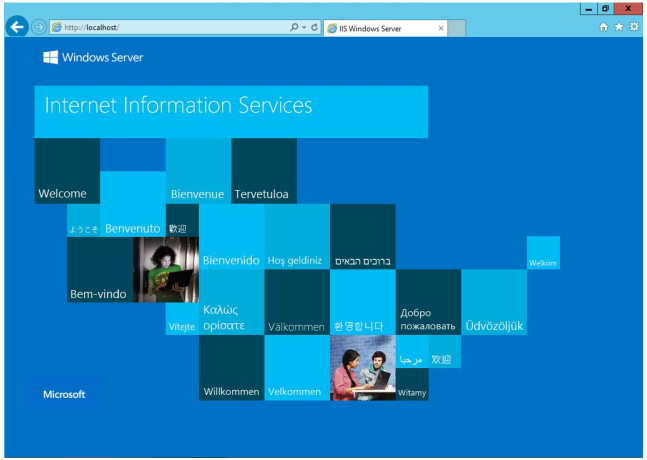

Client issues secure session request
(HTTPS://someserver.org/somedata.html)
Server send X.509 certificate containing
server's public key.
Client authenticates certificate against list of known CAs. (If CA is unknown, browser can
give user option to accept certificate at user's risk.)
Client generates random symmetric key and
encrypts it using server's public key.
Client and Server now both know the symmetric key and encrypt end-user
data using symmetric key for duration of session.

Configuring IIS ➔ Creating a Website

Open up IIS management console (as always, Tools menu)

Default website already running on port 80

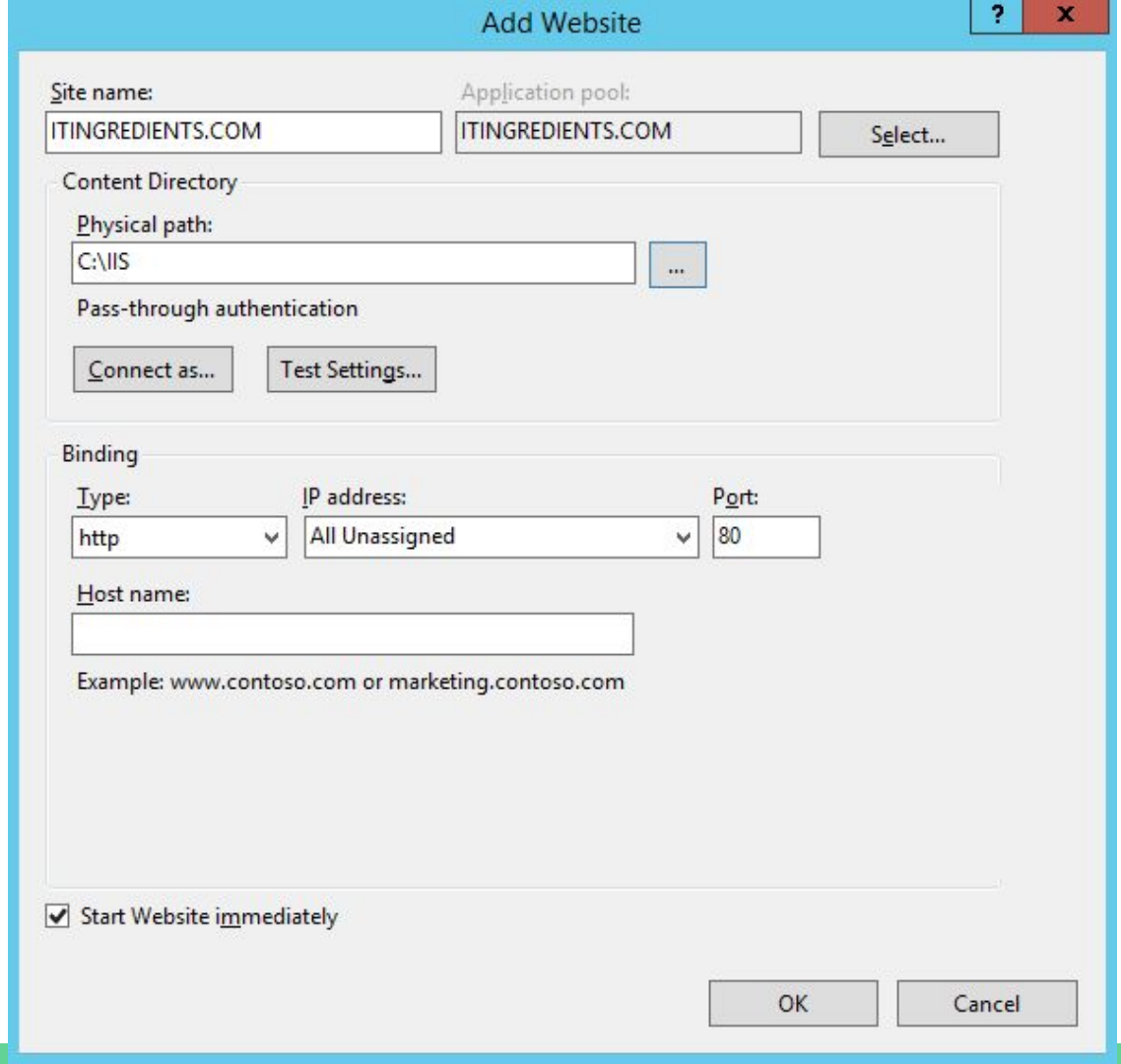
Can either just modify that one or make our own



http://www.itingredients.com/create-website-on-iis-in-windows-server-2012-r2/

Creating a Website

Click ‘Stop’ on default website (right click or use right menu)
Right click on Sites and ‘Add Website’
Configure…


Add Website
Site name:
ITINGREDIENTS.COM
Application pool:
ITINGREDIENTS.COM
Select...
Content Directory
Physical path:
C:\IIS
...
Pass-through authentication
Connect as...
Test Settings...
Binding
Type:
http
IP address:
All Unassigned
Port:
80
Host name:
Example: www.contoso.com or marketing.contoso.com
Start Website immediately
OK
Cancel

Sites, Applications, Virtual Directories

Site

Contains one or more applications

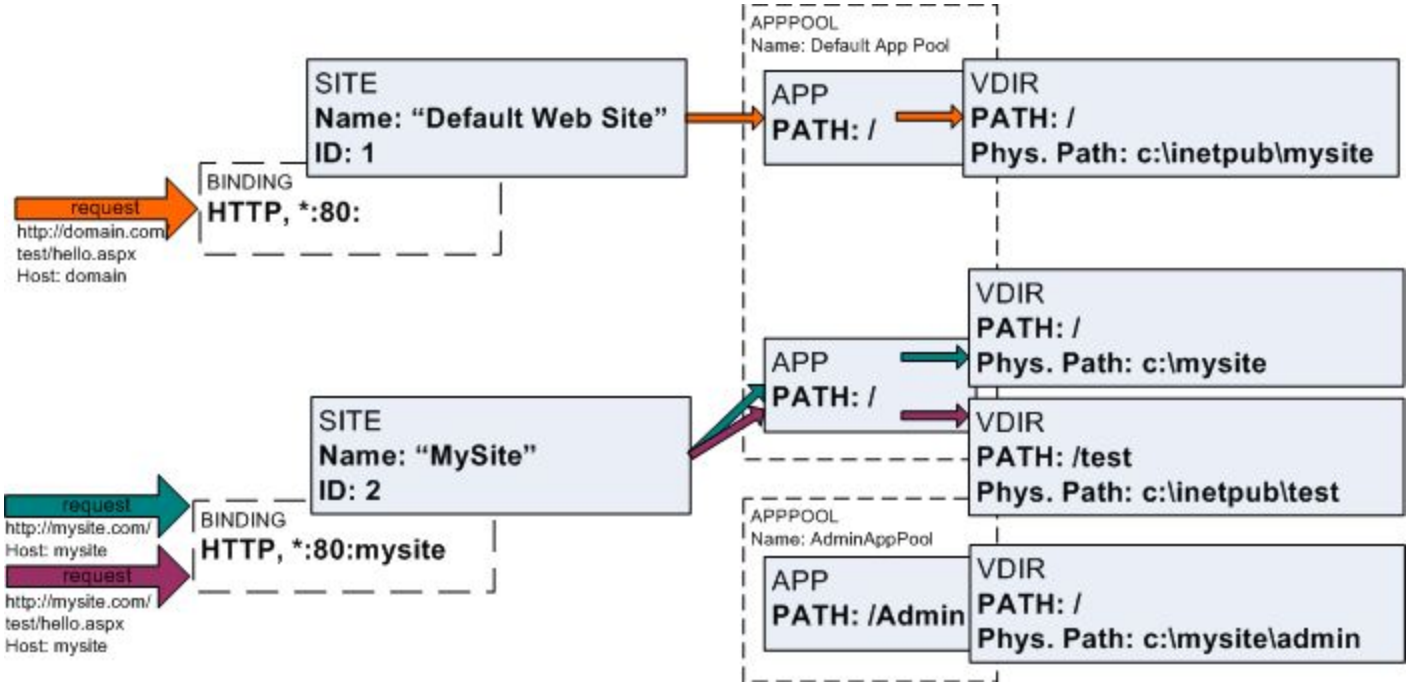
Application

Contains one or more virtual directories

Virtual directory

Maps to physical directory

https://www.iis.net/learn/get-started/planning-your-iis-architecture/understanding-sites-applications-and-virtual-directories-on-iis


APPPOOL
Name: Default App Pool
SITE
Name: "Default Web Site"
ID: 1
APP
PATH: /
VDIR
PATH: /
Phys. Path: c:\inetpub\mysite
BINDING
HTTP, *:80:
request
http://domain.com/
test/hello.aspx
Host: domain
VDIR
PATH: /
Phys. Path: c:\mysite
APP
PATH: /
SITE
Name: "MySite"
ID: 2
VDIR
PATH: /test
Phys. Path: c:\inetpub\test
request
http://mysite.com/
Host: mysite
BINDING
HTTP, *:80:mysite
APPPOOL
Name: AdminAppPool
request
http://mysite.com/
test/hello.aspx
Host: mysite
APP
PATH: /Admin
VDIR
PATH: /
Phys. Path: c:\mysite\admin

Site

Container for applications/virtual directories

Accessed via unique *binding*

Binding: protocol and information

Protocol ➜ HTTP, HTTPS, Named Pipes, Windows Communication Foundation, etc.

Basically, not strictly HTTP traffic…

Information ➜ IP address, host, any HTTP headers

Application

Group of files that:

- Deliver content
- Provide service over protocol

Need at least one per site

- Multiples possible

  E.g., one for login, one for shopping, one for a forum, etc.

Belongs to application pool

- Isolated from other applications on server

Virtual Directories

Effectively just a pointer to a folder (local or remote)

- Every website needs a virtual directory

Need at least one per site

- Could include more
- Why more than one?
  - Perhaps include images folder from another location
    - Without moving it

Configuring IIS ➜ Virtual Directories

Select site and add virtual directory

Permissions error? Add IIS_IUSRS to Security tab

Configuring IIS ➔ Applications

Select site and add application

(Pretty similar to virtual directories)

https://docs.microsoft.com/en-us/iis/manage/powershell/powershell-snap-in-creating-web-sites-web-applications-virtual-directories-and-application-pools

Let’s Powershell it up..

First, new directories

```
New-Item C:\inetpub\wwwroot\ps -type Directory
New-Item C:\inetpub\wwwroot\ps\app -type Directory
New-Item C:\inetpub\wwwroot\ps\vd -type Directory
```

And some demo content

```
Set-Content C:\inetpub\wwwroot\ps\Default.htm "DemoSite Default
Page"

Set-Content C:\inetpub\wwwroot\ps\app\Default.htm
"DemoSite\DemoApp Default Page"

Set-Content C:\inetpub\wwwroot\ps\vd\Default.htm
"DemoSite\DemoVirtualDir1 Default Page"
```

And our app pool

```
Import-Module "WebAdministration"

New-Item IIS:\AppPools\DemoAppPool
```

And the remaining creations

```
New-Item IIS:\Sites\DemoSite -physicalPath C:\inetpub\wwwroot\ps -bindings
@{protocol="http";bindingInformation=":8080:"}

Set-ItemProperty IIS:\Sites\DemoSite -name applicationPool -value DemoAppPool

New-Item IIS:\Sites\DemoSite\DemoApp -physicalPath C:\inetpub\wwwroot\ps\app -type
Application

Set-ItemProperty IIS:\sites\DemoSite\DemoApp -name applicationPool -value DemoAppPool

New-Item IIS:\Sites\DemoSite\DemoVirtualDir1 -physicalPath C:\inetpub\wwwroot\ps\vd
-type VirtualDirectory
```

Let’s check it out!

8080 will be blocked via VPN...

Alternatively…

```
$webclient = New-Object Net.WebClient
$webclient.DownloadString("http://localhost:8080/");
$webclient.DownloadString("http://localhost:8080/DemoApp");
$webclient.DownloadString("http://localhost:8080/DemoVirtualDir1")
;
```

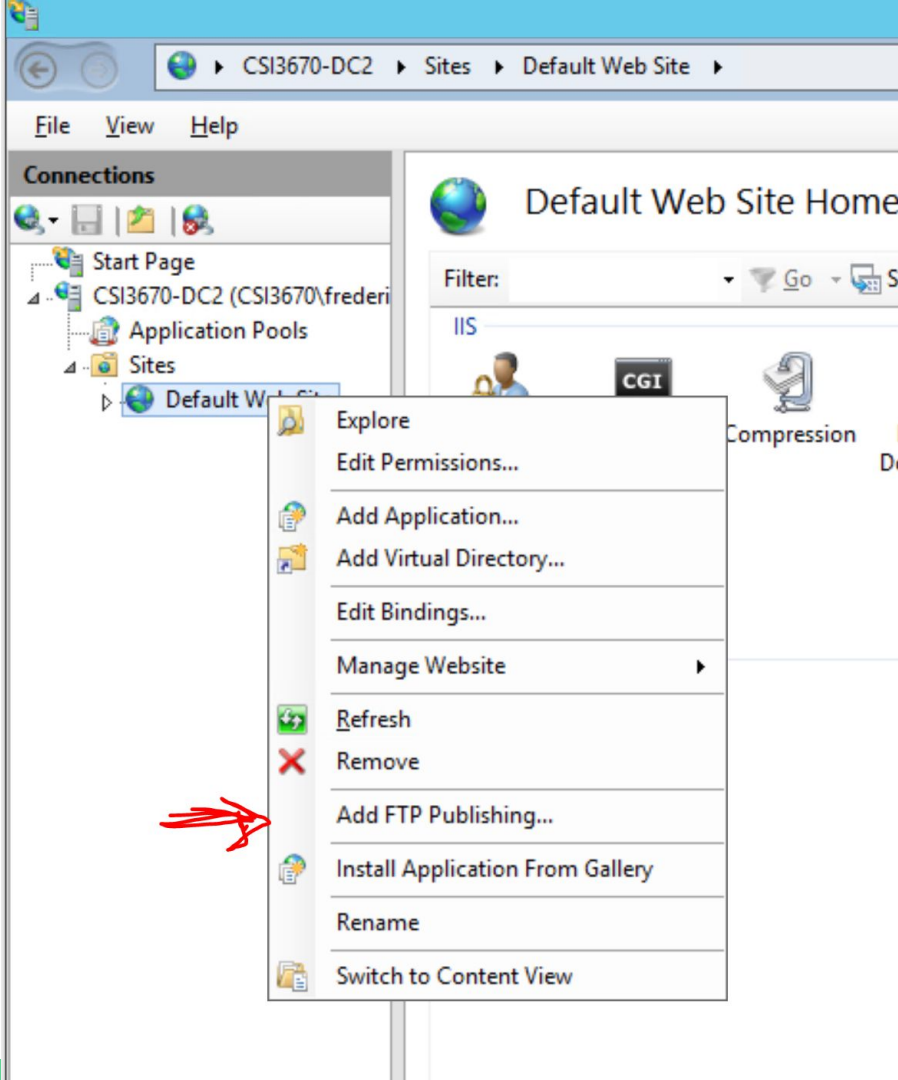
FTP

Roles → Web Server → Enable FTP

Right click on website you want to allow it for:

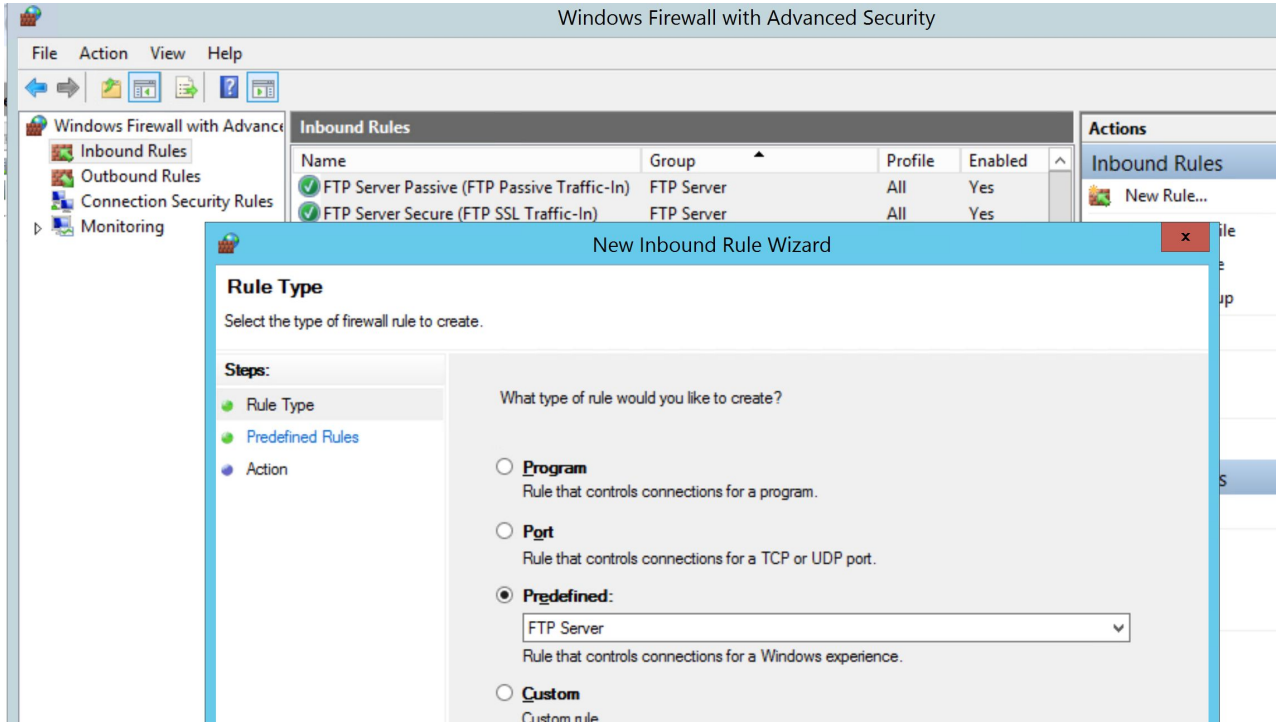
Set it up as you want.

- Anonymous means **completely open**
- Basic maps logins to AD



FTP

Update the firewall



Microsoft Web Platform

Extremely handy tool for IIS components

One (well, several) click solution for installing common components

- Azure plugins
- MySQL
- PHP
- Etc.

Separate download, but IIS will ask you if you want it after you install the role

Microsoft Web Platform

...installation of tools seems to work about half the time

WIMP Server

(PHP / MySQL)
https://blogs.msdn.microsoft.com/africaapps/2013/11/05/creating-a-wimp-stack-windows-server-iis-mysql-php-on-windows-azure/

via Web Installer:

- Install PHP/MySQL (try try again if PHP fails...I had luck with 5.6)
- Index.php: phpinfo() test
- Download phpmyadmin and unzip to C:\inetpub\wwwroot
- Give the IUSR full control in the Security tab
- Config and awayyyy we go!!

Project Work Time

Between the extra time you have this class and next class I would like to see some **forward progress** made on your term projects.

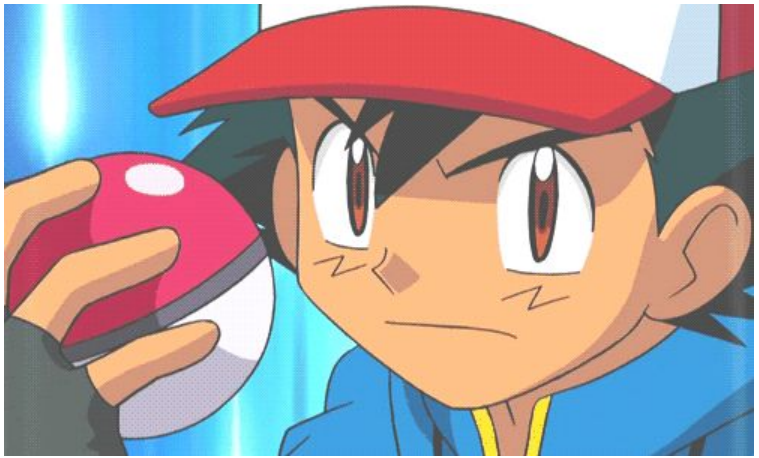
Pick **one** thing that you need to work on to move it forward and implement it.

- This is intentionally vague

Super Project Group Work Time Go

Considerations:

- You'll need a web server



Deliverables by Friday night

You'll have time on Thursday to work on this as well!

Project Work

Time to pick your topic

Write down (2) ideas of what you could do

Keep in mind you have a Windows server, Ubuntu server, and Google Cloud instance(s)

Deliverables **04/17**:

- Functioning project
- Descriptive website
- **Single-sign on**

A bit different requirements for today

So the writeup that you do today is due on **Moodle** by tomorrow night

Don’t turn anything in to me during class, you want electronic copies of this

But

I want to meet with **each** group before you leave and hear about your plans

So each group should come talk to me after they’re done discussing

Tell me your group name

(So I can record)

Your ideas

And I’ll give you feedback

Otherwise the Moodle assignment = 0

Hooking up a Website to MS SQL Server

Part of your in-class lab NEXT WEEK TUESDAY

Project Work Time

Tuesday you'll **all** be setting up IIS and a basic website

But,

You'll need to figure out the web portion of your project

Come together as teams and figure out

1) What your website will be (it has to be database-driven)
    a) Should describe your project
    b) Provide some manner of interaction (wiki, forum, blog, etc) AND hook into AD
2) What those requirements will be (do you need MySQL, will MS SQL work?)
3) Who will be hosting it

Take a picture for yourselves and submit your writeup through Moodle by tomorrow night