QM07

The Top Web Application Attacks: Are you vulnerable?

John Burroughs, CISSP

Sr Security Architect, Watchfire Solutions

jburroughs@uk.ibm.com

IBM Rational Software Development Conference 2008

WHERE TEAMS ARE R-HEROES

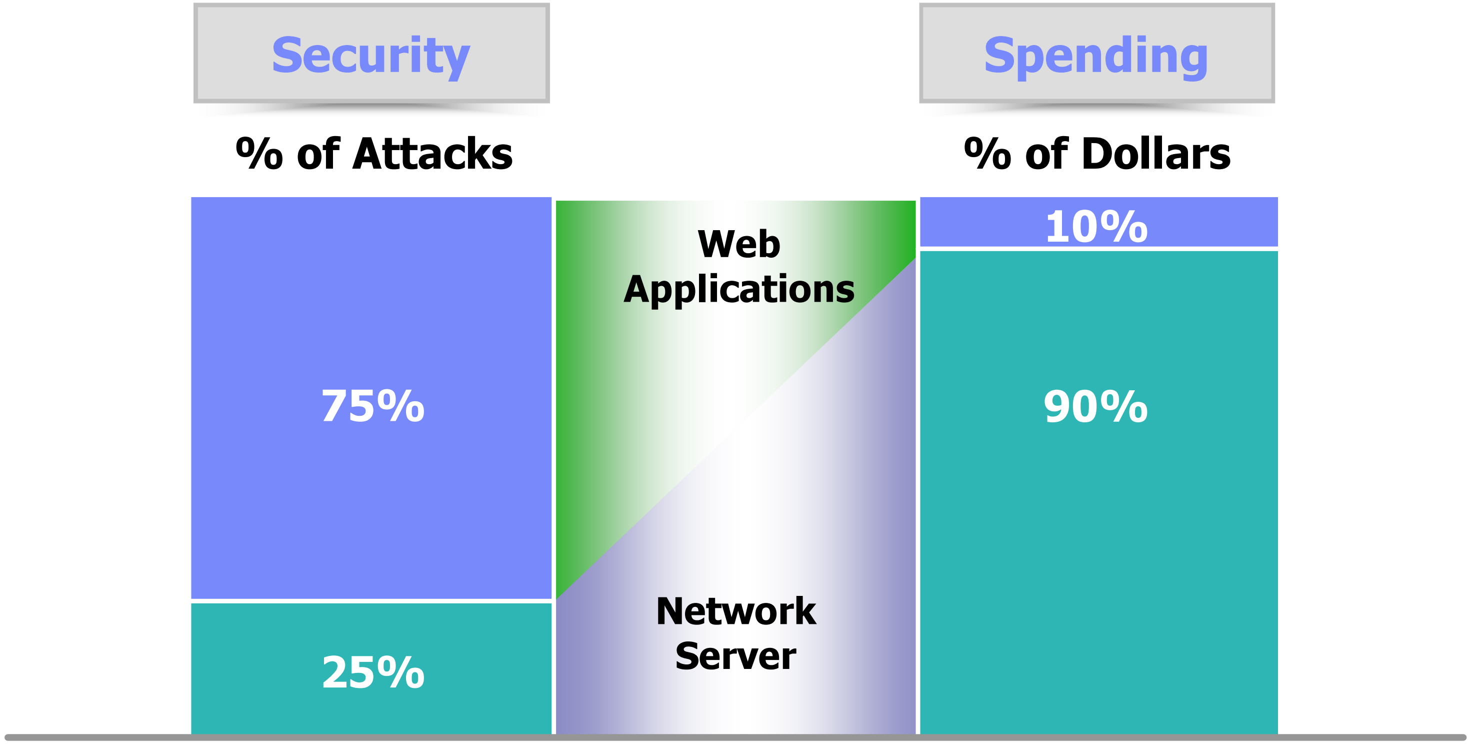
IBM

Rational. software

Agenda

- Current State of Web Application Security
- Understanding Web Application Attacks
  - Demo of the 4 top vulnerabilities affecting Web Application and how they can be exploited
- How to automatically find these vulnerabilities on your Web Site

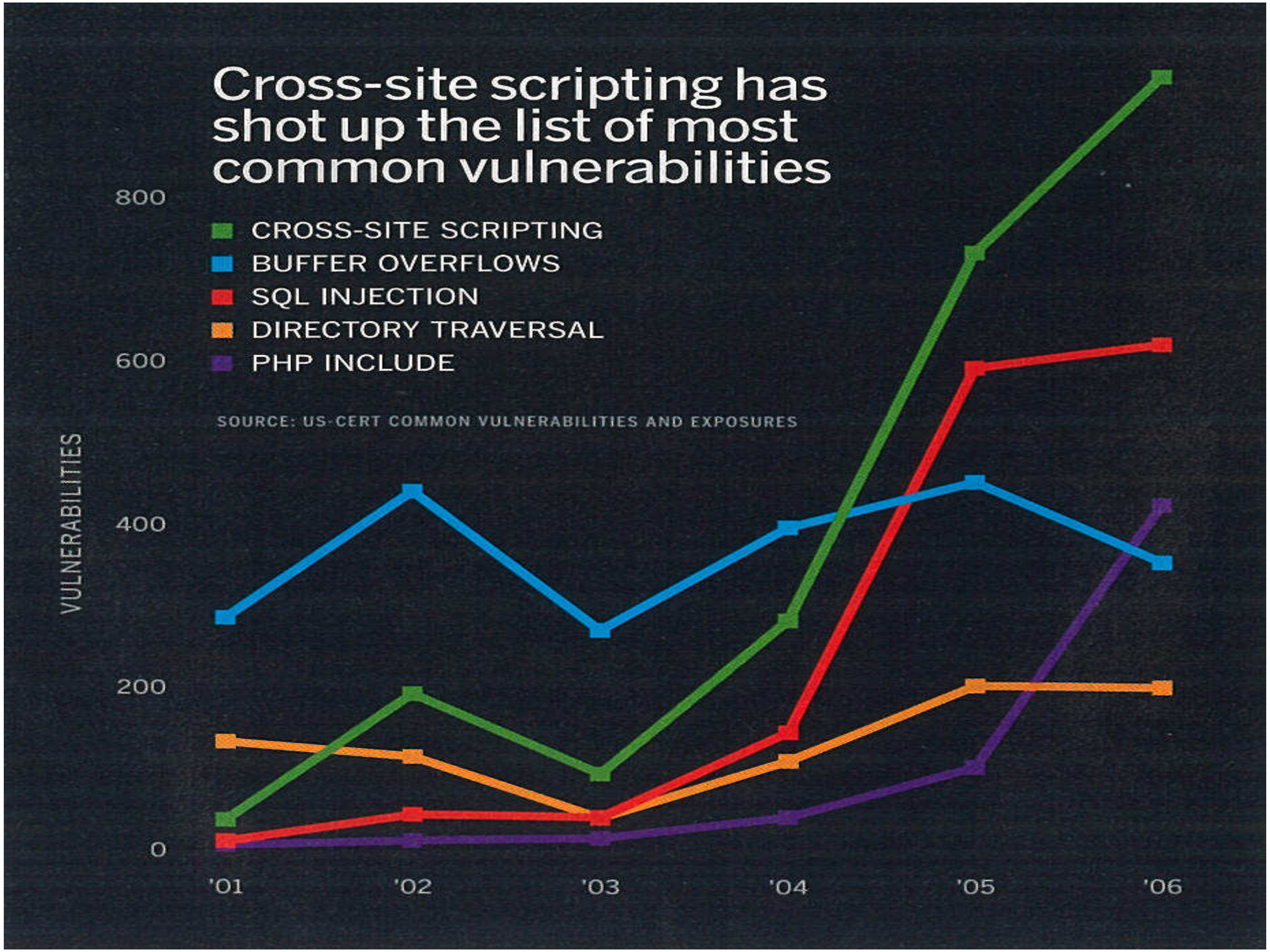
The Challenge for Organizations



75% of All Attacks on Information Security Are Directed to the Web Application Layer

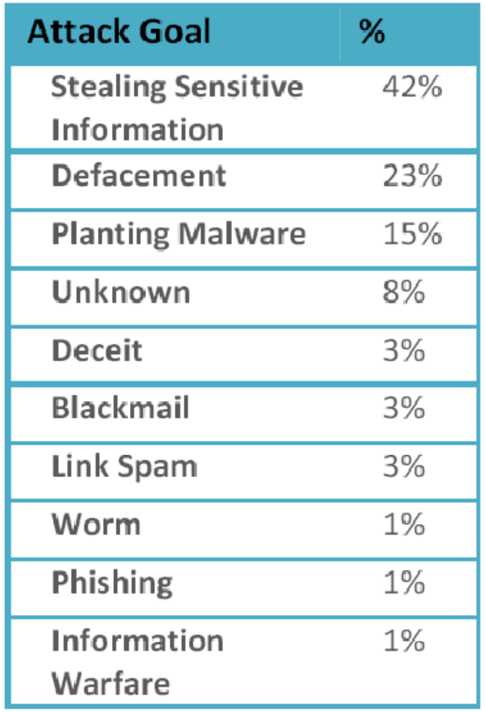
Sources: Gartner, Watchfire

Application Security Defects #1 & #2 Vulnerabilities



SOURCE: US-CERT COMMON VULNERABILITIES AND EXPOSURES

Drivers for Web Application Attacks*



Attack Goal	%
Stealing Sensitive Information	42%
Defacement	23%
Planting Malware	15%
Unknown	8%
Deceit	3%
Blackmail	3%
Link Spam	3%
Worm	1%
Phishing	1%
Information Warfare	1%

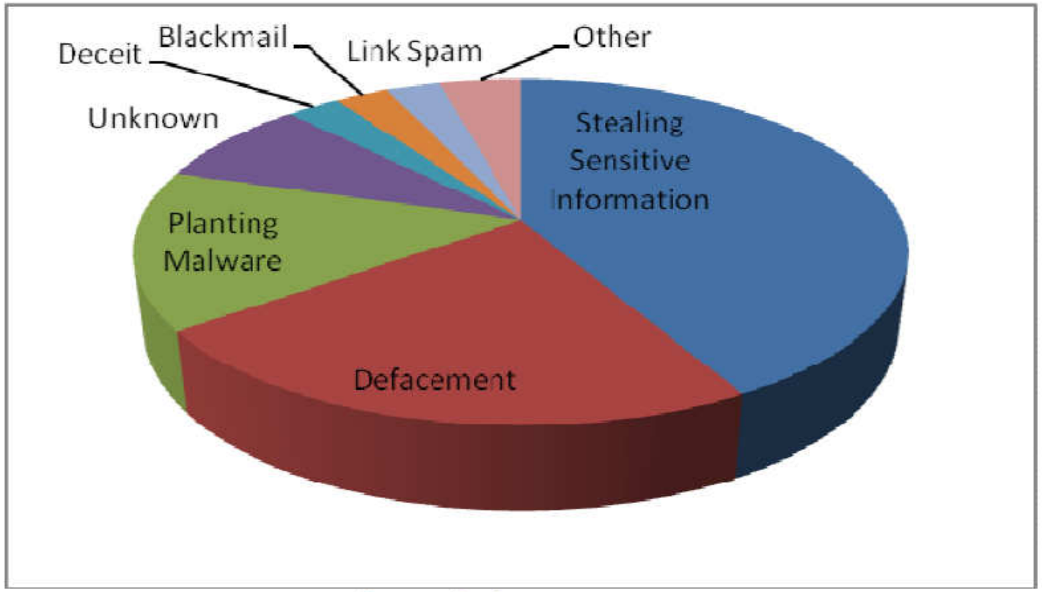


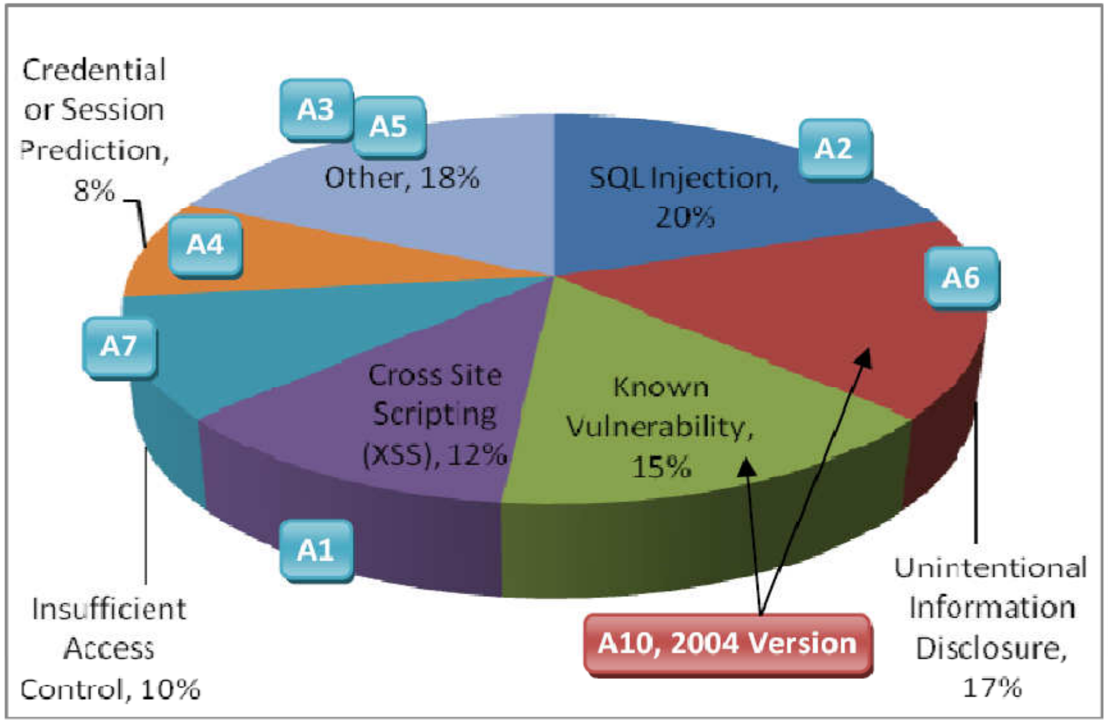
FIGURE 1 - INCIDENT BY OUTCOME

Web Applications hacks have replaced email as the preferred delivery method of Malware (viruses, root kits and trojans)

*From Web Hacking Incidents Database 2007

Incident by Attack Type (2007)

Attack/Vulnerability Used	%
SQL Injection	20%
Unintentional Information Disclosure	17%
Known Vulnerability	15%
Cross Site Scripting (XSS)	12%
Insufficient Access Control	10%
Credential/Session Prediction	8%
OS Commanding	3%
Misconfiguration	3%
Insufficient Anti-automation	3%
Denial of Service	3%
Redirection	2%
Insufficient Session Expiration	2%
Cross Site Request Forgery (CSRF)	2%



Cost of an Application Security Breach

Media attention/ Brand damage

Sharp decline in Stock Prices

Communication/Monitoring Service Costs

Legal Fees (Reported $3-4 million/incident)

FTC Penalties (Fines can range up to 15 million/incident)

Additional 3rd party Audits

New Security Spending

Customer Lawsuits

Customer Loss

TJ Maxx’s Application Security Breach cost them over 45 million dollars!!

Why Application Security Problems Exist

Root Cause:

- **Developers are not trained to write or test for secure code**
- Firewalls and IDS/IPS systems don’t block application attacks.
  - Port 80/443 is wide open for attack.
- Network scanners won’t find application vulnerabilities.
- Network security (firewall, IDS, etc) do nothing once an organization web enables an application.

Current State:

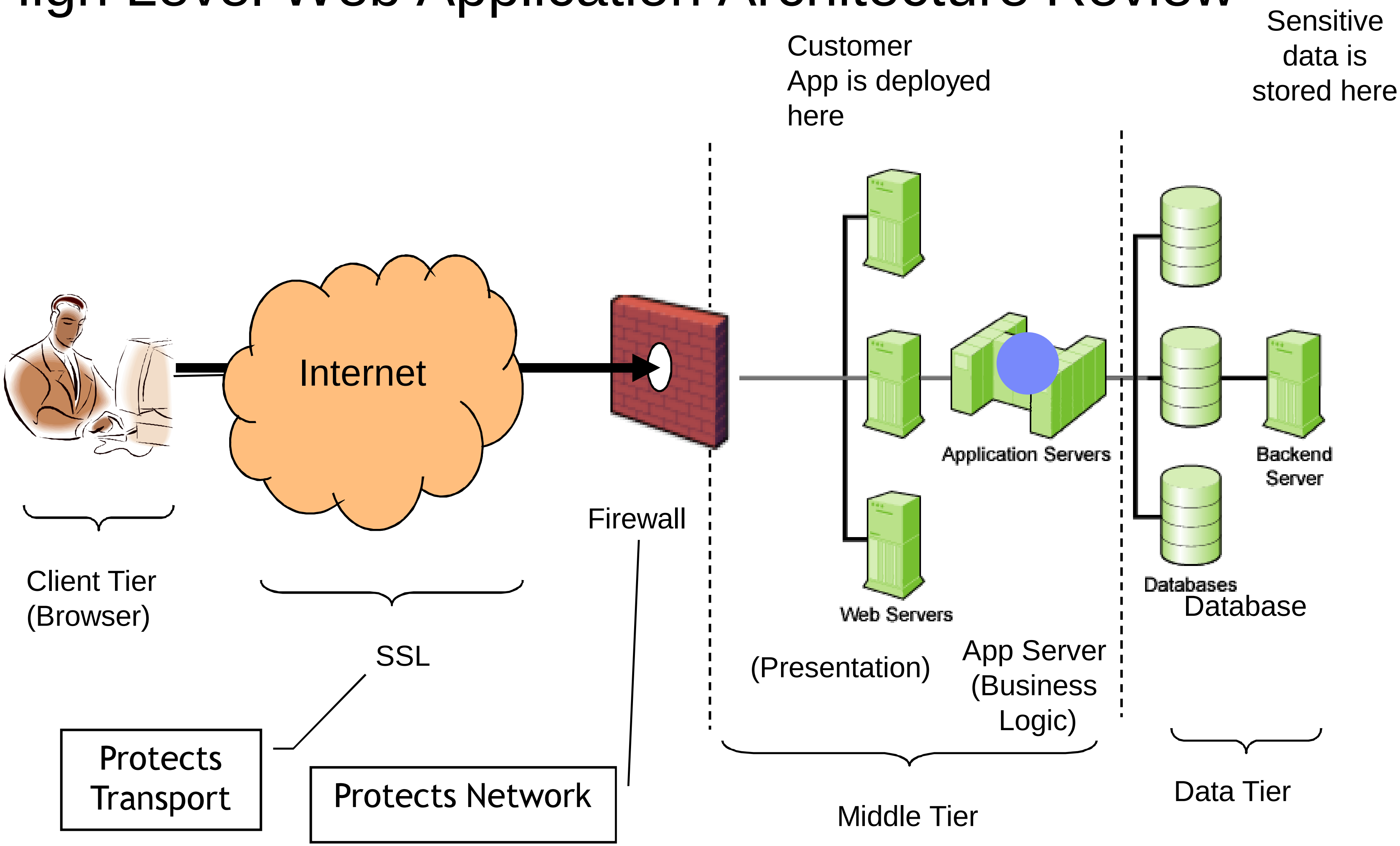
- Organizations test tactically at a late & costly stage in the SDLC
- A communication gap exists between security and development as such vulnerabilities are not fixed
- Testing coverage is incomplete

Goal:

- To build better and more secure applications/websites

Understanding Web Application Attacks

High Level Web Application Architecture Review



The Myth: "Our Site Is Safe"

We Have Firewalls in Place

We Audit It Once a Quarter with Pen Testers

We use SSL

We Use Network Vulnerability Scanners

OWASP and the OWASP Top 10 list

Open Web Application Security Project – an open organization dedicated to fight insecure software

“The OWASP Top Ten document represents a broad consensus about what the most critical web application security flaws are”

We will use the Top 10 list to cover some of the most common security issues in web applications

OWASP Top 10 Application Attacks

Application Threat	Negative Impact	Example Impact
Cross Site scripting	Identity Theft, Sensitive Information Leakage, …	Hackers can impersonate legitimate users, and control their accounts.
Injection Flaws	Attacker can manipulate queries to the DB / LDAP / Other system	Hackers can access backend database information, alter it or steal it.
Malicious File Execution	Execute shell commands on server, up to full control	Site modified to transfer all interactions to the hacker.
Insecure Direct Object Reference	Attacker can access sensitive files and resources	Web application returns contents of sensitive file (instead of harmless one)
Cross-Site Request Forgery	Attacker can invoke "blind" actions on web applications, impersonating as a trusted user	Blind requests to bank account transfer money to hacker
Information Leakage and Improper Error Handling	Attackers can gain detailed system information	Malicious system reconnaissance may assist in developing further attacks
Broken Authentication & Session Management	Session tokens not guarded or invalidated properly	Hacker can "force" session token on victim; session tokens can be stolen after logout
Insecure Cryptographic Storage	Weak encryption techniques may lead to broken encryption	Confidential information (SSN, Credit Cards) can be decrypted by malicious users
Insecure Communications	Sensitive info sent unencrypted over insecure channel	Unencrypted credentials "sniffed" and used by hacker to impersonate user
Failure to Restrict URL Access	Hacker can access unauthorized resources	Hacker can forcefully browse and access a page past the login page

1. Cross-Site Scripting (XSS)

What is it?

- Malicious script echoed back into HTML returned from a trusted site, and runs under trusted context

What are the implications?

- Session Tokens stolen (browser security circumvented)
- Complete page content compromised
- Future pages in browser compromised

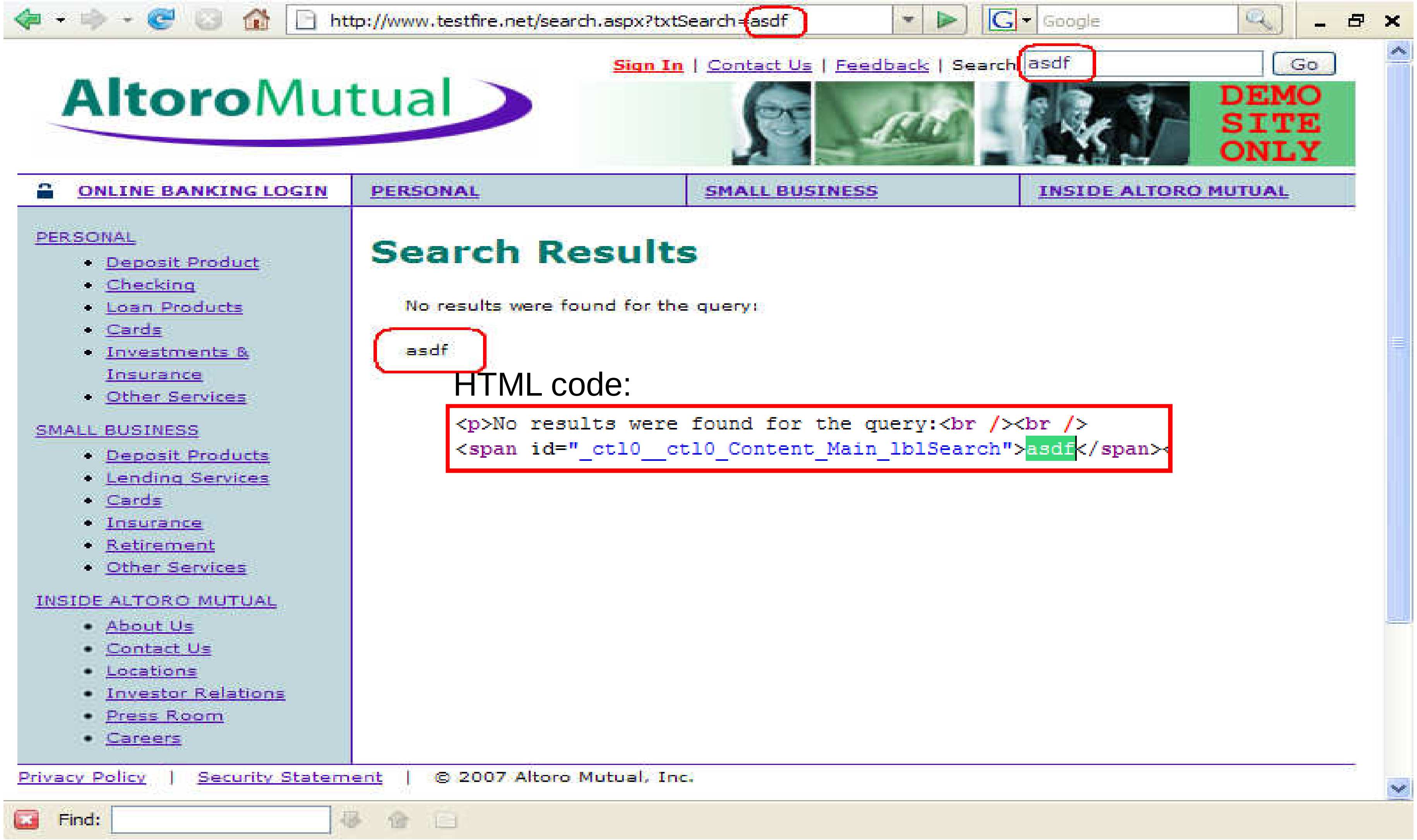
Demonstration – Cross Site Scripting

Main points covered in the demo or video:

- Locating an a place where user input which is echoed back to the browser
- Seeing if the user input is echoed back 'as-is' or if it is properly encoded
- Exploiting the vulnerability

CLICK HERE TO LAUNCH DEMO

XSS Example I



http://www.testfire.net/search.aspx?txtSearch=asdf

Sign In | Contact Us | Feedback | Search asdf Go

AltoroMutual

DEMO SITE ONLY

ONLINE BANKING LOGIN | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

PERSONAL
- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS
- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

INSIDE ALTORO MUTUAL
- About Us
- Contact Us
- Locations
- Investor Relations
- Press Room
- Careers

Search Results

No results were found for the query:

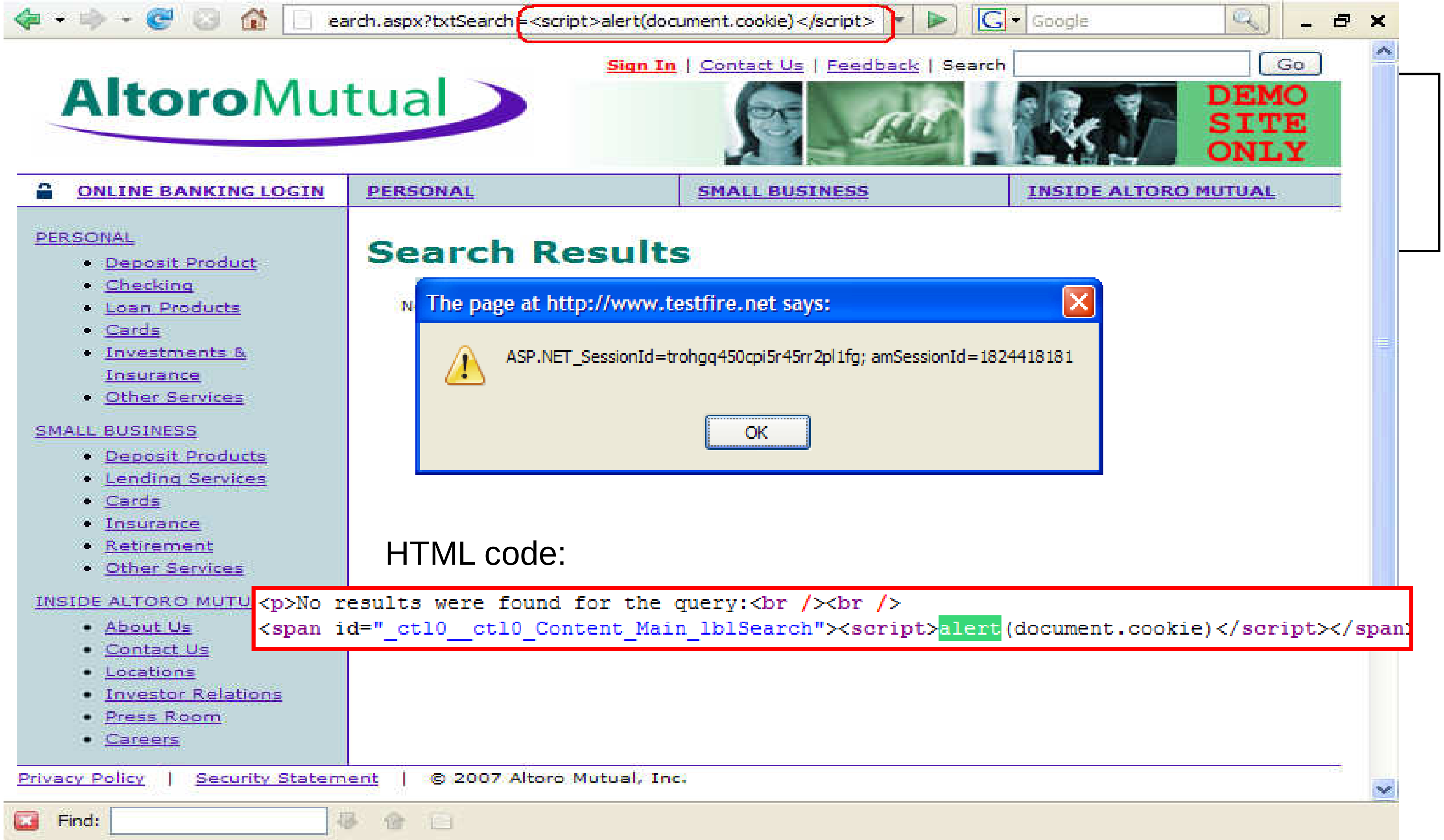
asdf

HTML code:

```
<p>No results were found for the query:<br /><br />
<span id="_ctl0__ctl0_Content_Main_lblSearch">asdf</span><
```

Privacy Policy | Security Statement | © 2007 Altoro Mutual, Inc.

XSS Example II



Search Results

The page at http://www.testfire.net says:

ASP.NET_SessionId=trohgq450cpi5r45rr2pl1fg; amSessionId=1824418181

OK

HTML code:

```
<p>No results were found for the query:<br /><br />
<span id="_ctl0__ctl0_Content_Main_lblSearch"><script>alert(document.cookie)</script></span
```

XSS – Details

Common in Search, Error Pages and returned forms.

- But can be found on any type of page

Any input may be echoed back

- Path, Query, Post-data, Cookie, Header, etc.

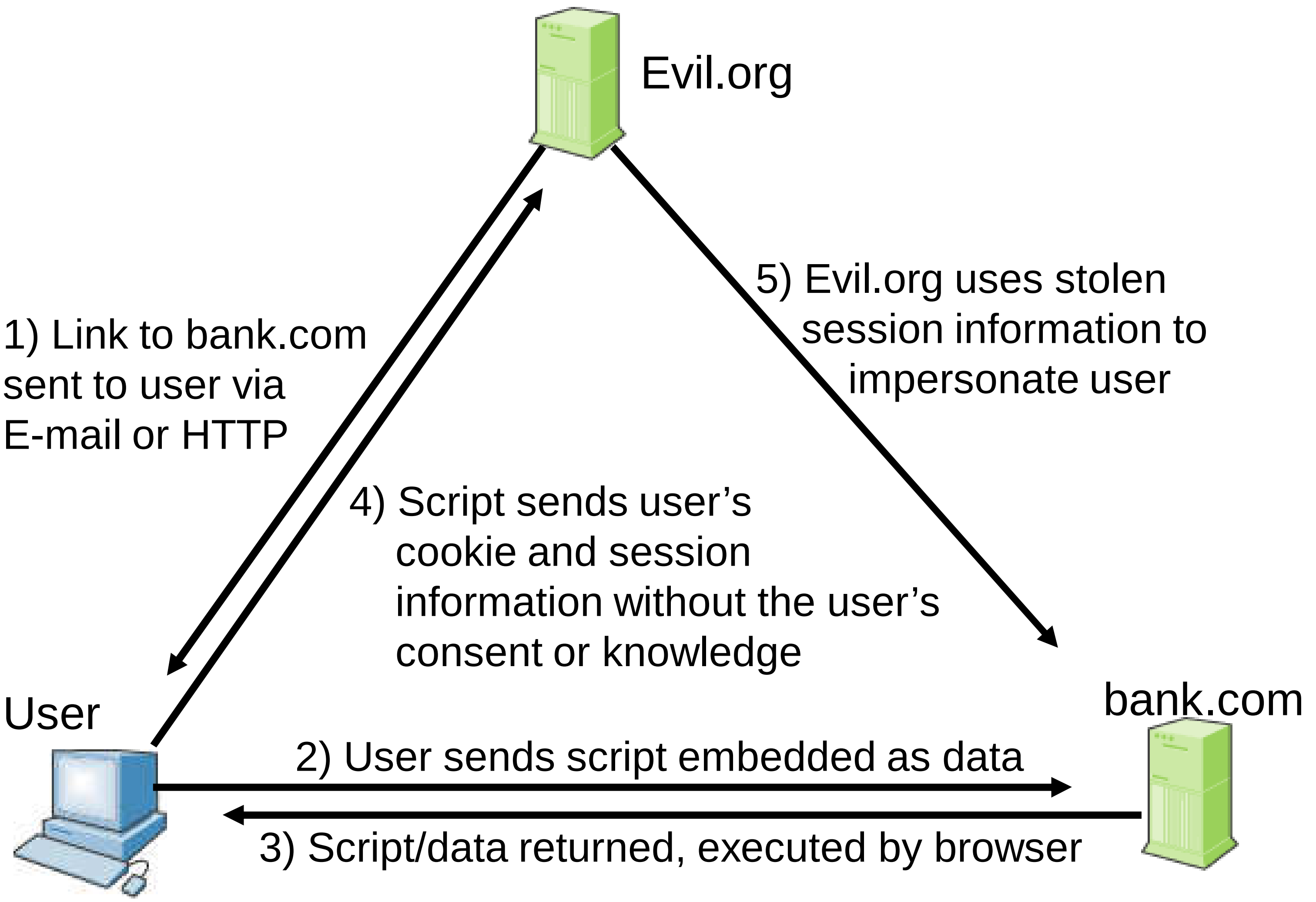
Browser technology used to aid attack

- XMLHttpRequest (AJAX), Flash, IFrame…

Has many variations

- XSS in attribute, DOM Based XSS, etc.

Cross Site Scripting – The Exploit Process



Evil.org

1) Link to bank.com sent to user via E-mail or HTTP

5) Evil.org uses stolen session information to impersonate user

4) Script sends user's cookie and session information without the user's consent or knowledge

User

bank.com

2) User sends script embedded as data

3) Script/data returned, executed by browser

Exploiting XSS

If I can get you to run my JavaScript, I can…

- Steal your cookies for the domain you're browsing
- Track every action you do in that browser from now on
- Redirect you to a Phishing site
- Completely modify the content of any page you see on this domain
- Exploit browser vulnerabilities to take over machine
- …

2 - Injection Flaws

What is it?

User-supplied data is sent to an interpreter as part of a command, query or data.

What are the implications?

SQL Injection – Access/modify data in DB

SSI Injection – Execute commands on server and access sensitive data

LDAP Injection – Bypass authentication

…

SQL Injection

User input inserted into SQL Command:

Get product details by id:
Select * from products where id=‘$REQUEST[“id”]’;

Hack: send param id with value ‘ or ‘1’=‘1

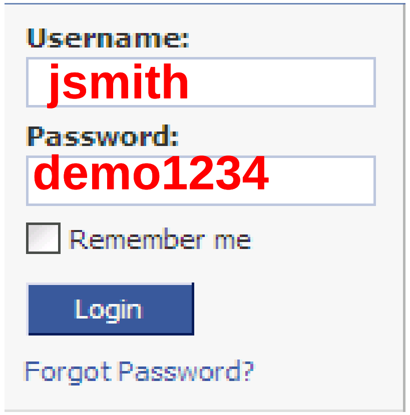
Resulting executed SQL:
Select * from products where id=‘’ or ‘1’=‘1’

All products returned

SQL Injection

User input is embedded as-is in predefined SQL statements:

```
query = "SELECT * from Users where
        userid=' +   iUserID + ' AND
        password=' +   iPassword + '";
```



Username:
jsmith

Password:
demo1234

□ Remember me

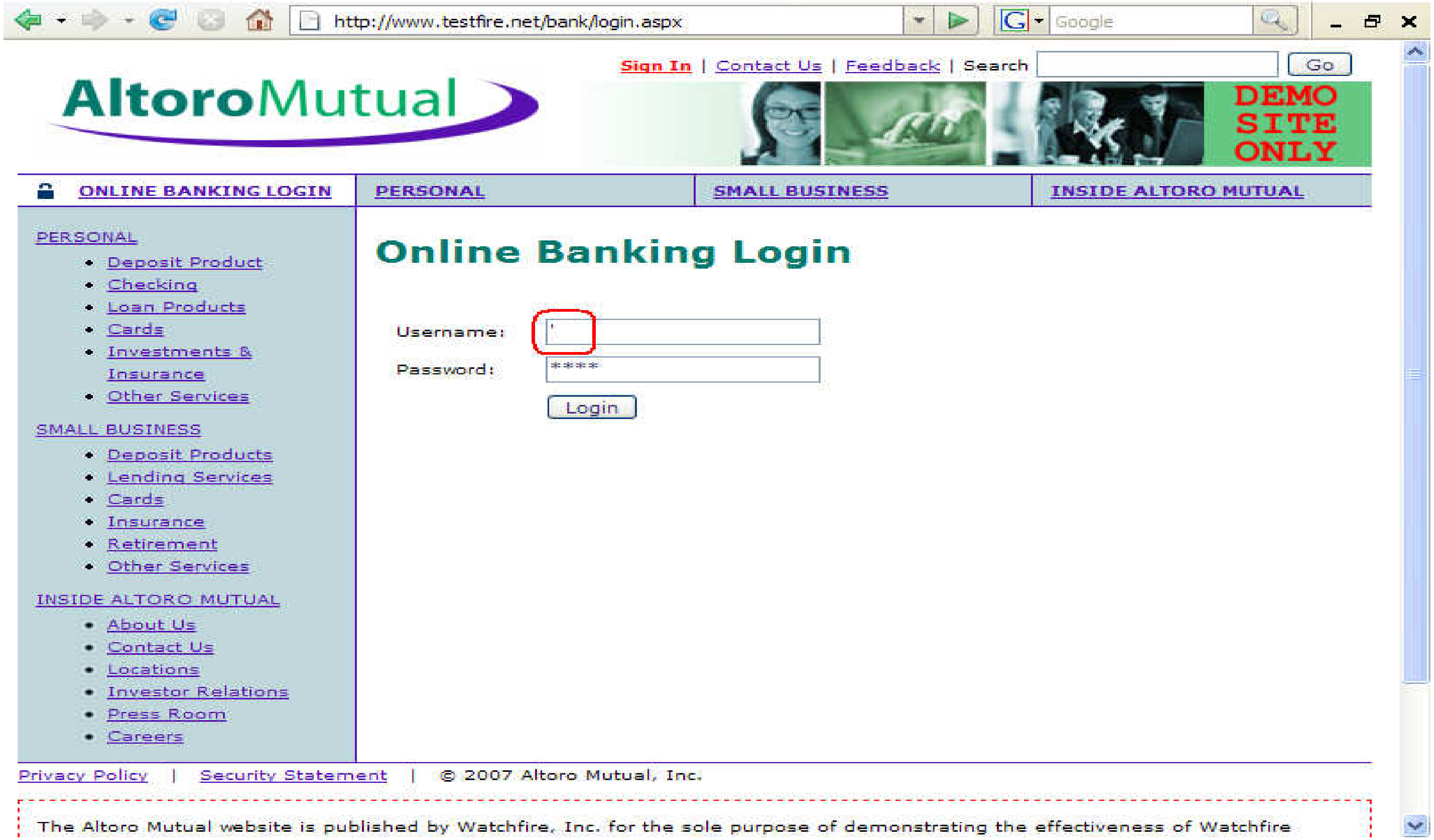
UserID	Username	Password	Name
1824	jsmith	demo1234	John Smith

Hacker supplies input that modifies the original SQL statement, for example:

```
iUserID = ' or 1=1 --
```

UserID	Username	Password	Name
1	admin	$#kaoeFor56	Administrator

SQL Injection Example I



http://www.testfire.net/bank/login.aspx

Sign In | Contact Us | Feedback | Search

PERSONAL

- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS

- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

INSIDE ALTORO MUTUAL

- About Us
- Contact Us
- Locations
- Investor Relations
- Press Room
- Careers

Online Banking Login

Username: '

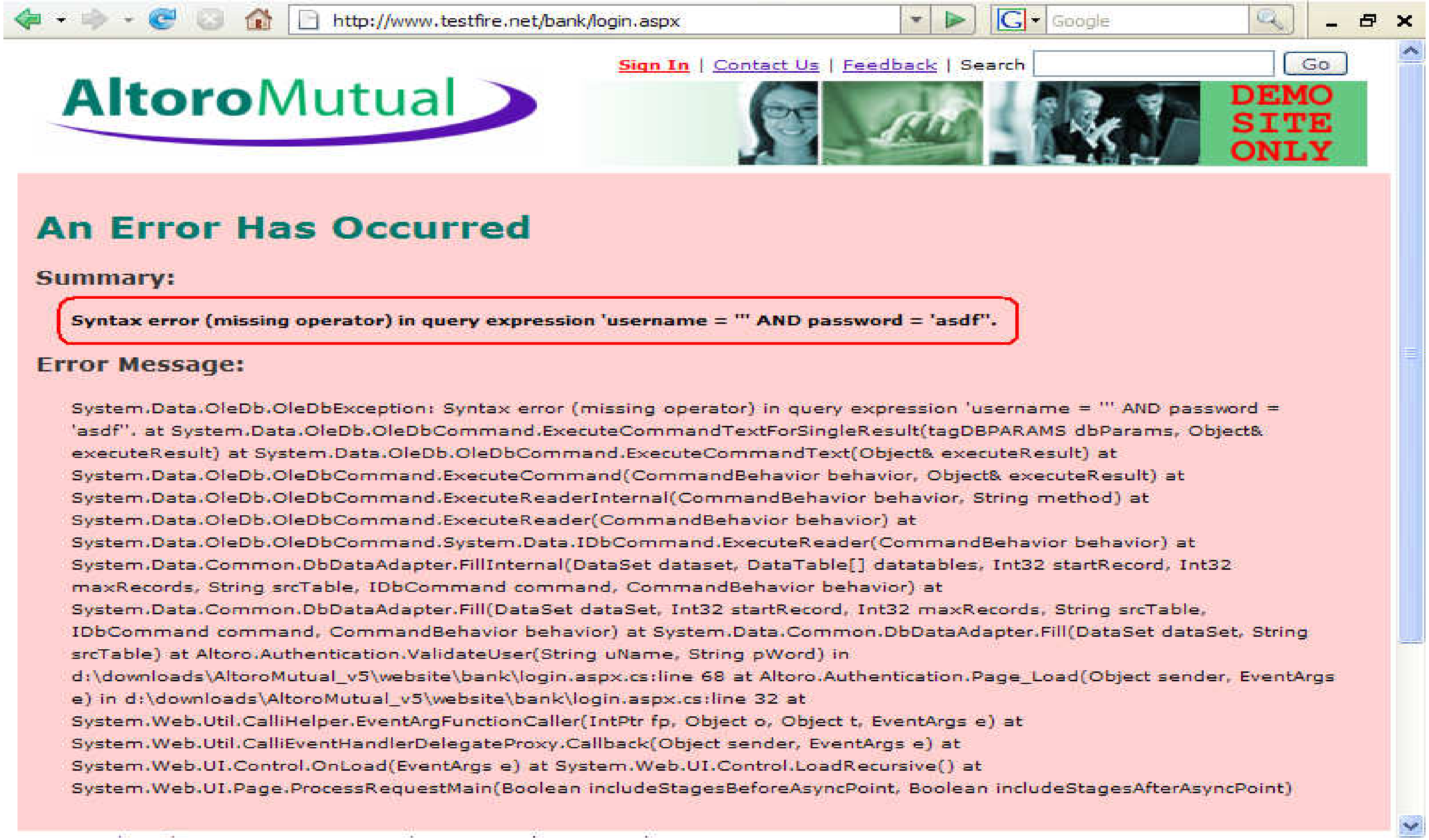
Password: ****

Login

Privacy Policy | Security Statement | © 2007 Altoro Mutual, Inc.

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire

SQL Injection Example II


http://www.testfire.net/bank/login.aspx
Google
Sign In | Contact Us | Feedback | Search
Go
AltoroMutual
DEMO SITE ONLY

An Error Has Occurred

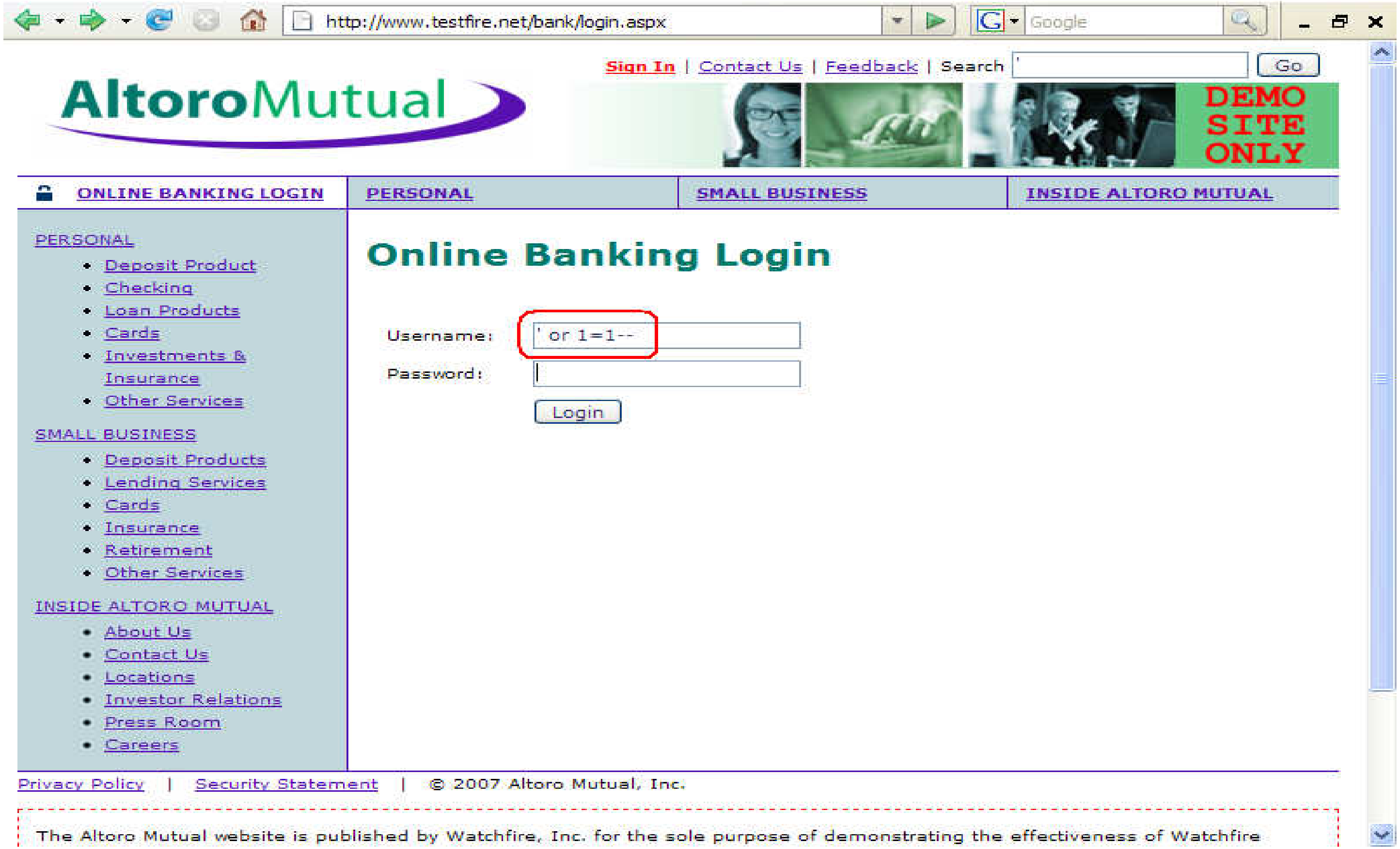
Summary:

Syntax error (missing operator) in query expression 'username = ''' AND password = 'asdf''.

Error Message:

```
System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression 'username = ''' AND password =
'asdf''. at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&
executeResult) at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object& executeResult) at
System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object& executeResult) at
System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method) at
System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior) at
System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.ExecuteReader(CommandBehavior behavior) at
System.Data.Common.DbDataAdapter.FillInternal(DataSet dataset, DataTable[] datatables, Int32 startRecord, Int32
maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior) at
System.Data.Common.DbDataAdapter.Fill(DataSet dataSet, Int32 startRecord, Int32 maxRecords, String srcTable,
IDbCommand command, CommandBehavior behavior) at System.Data.Common.DbDataAdapter.Fill(DataSet dataSet, String
srcTable) at Altoro.Authentication.ValidateUser(String uName, String pWord) in
d:\downloads\AltoroMutual_v5\website\bank\login.aspx.cs:line 68 at Altoro.Authentication.Page_Load(Object sender, EventArgs
e) in d:\downloads\AltoroMutual_v5\website\bank\login.aspx.cs:line 32 at
System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e) at
System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e) at
System.Web.UI.Control.OnLoad(EventArgs e) at System.Web.UI.Control.LoadRecursive() at
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
```

SQL Injection Example - Exploit



http://www.testfire.net/bank/login.aspx

Online Banking Login

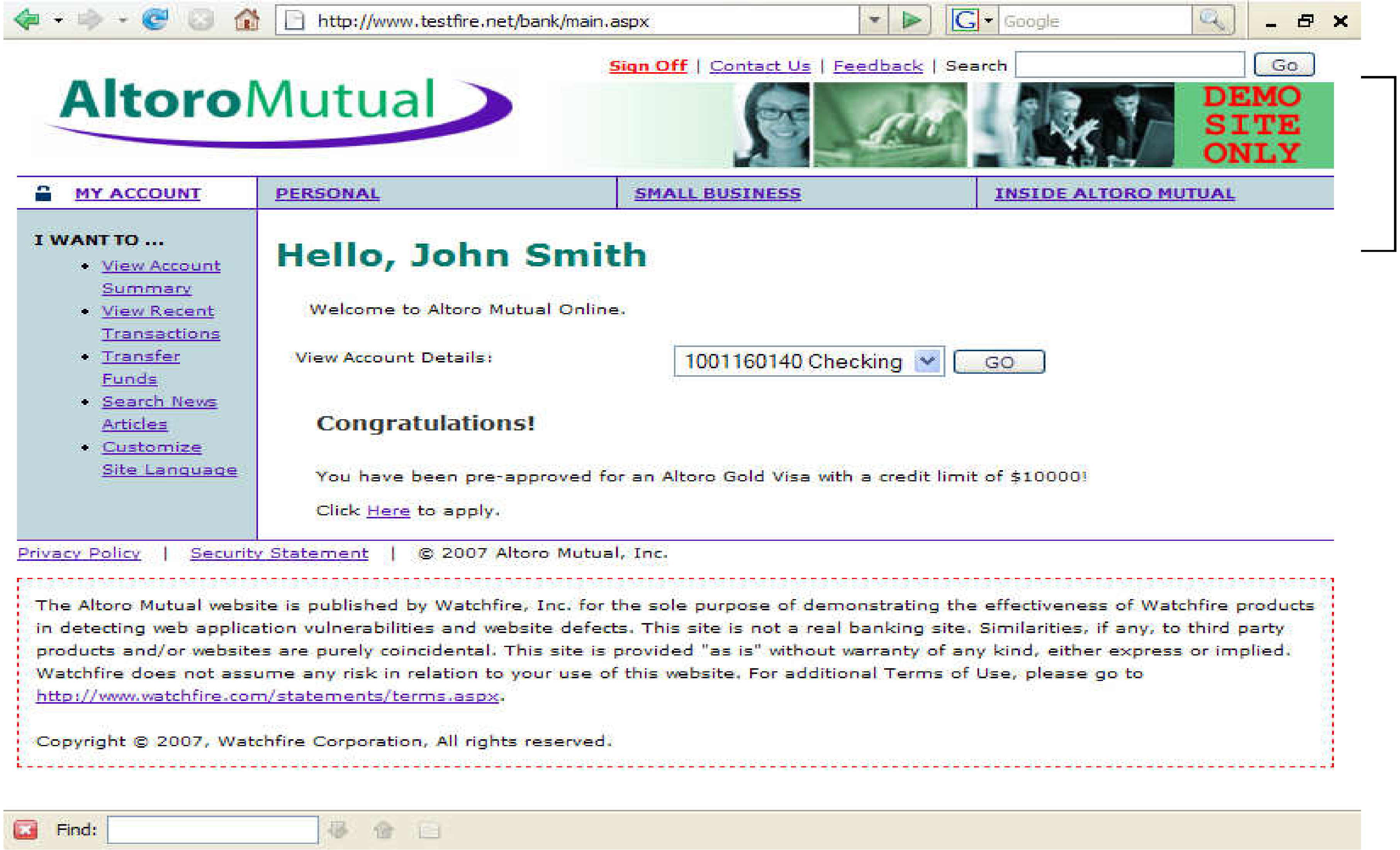
Username: ' or 1=1--

Password:

© 2007 Altoro Mutual, Inc.

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire

SQL Injection Example - Outcome



I WANT TO ...

- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

Hello, John Smith

Welcome to Altoro Mutual Online.

View Account Details: 1001160140 Checking GO

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

Privacy Policy | Security Statement | © 2007 Altoro Mutual, Inc.

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2007, Watchfire Corporation, All rights reserved.

Demonstration – SQL Injection

Main points covered in the demo or video:

- How to find a SQL injection vulnerability
- How to exploit a SQL injection vulnerability

3 - Malicious File Execution

What is it?

- Application tricked into executing commands or creating files on server

What are the implications?

- Command execution on server – complete takeover
- Site Defacement, including XSS option

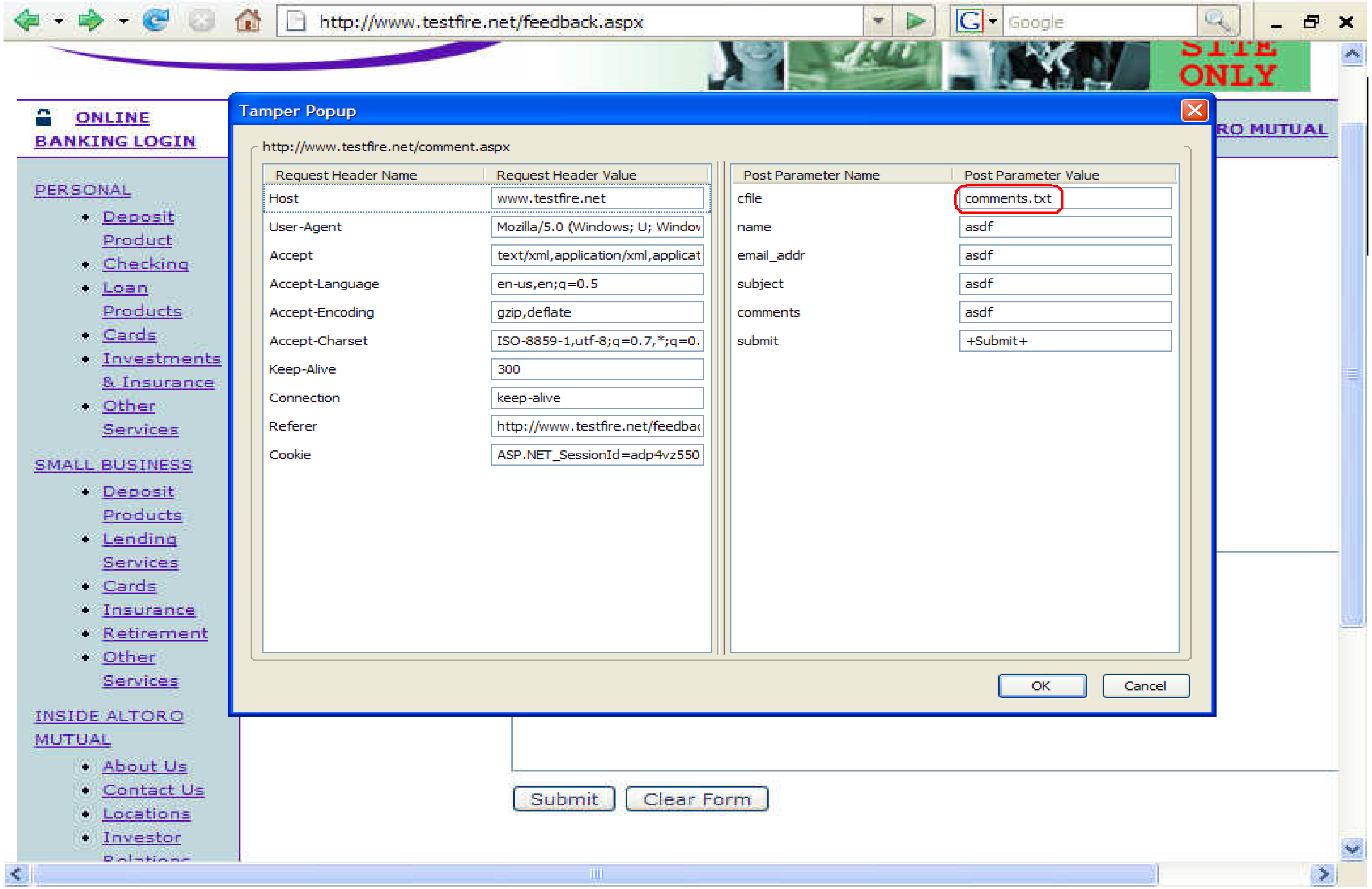
Demonstration – Malicious File

Main points covered in the demo or video:

- Demonstrating how a Malicious File Exploit attack can be used to get access to system files

CLICK HERE TO LAUNCH DEMO

Malicious File Execution – Example I

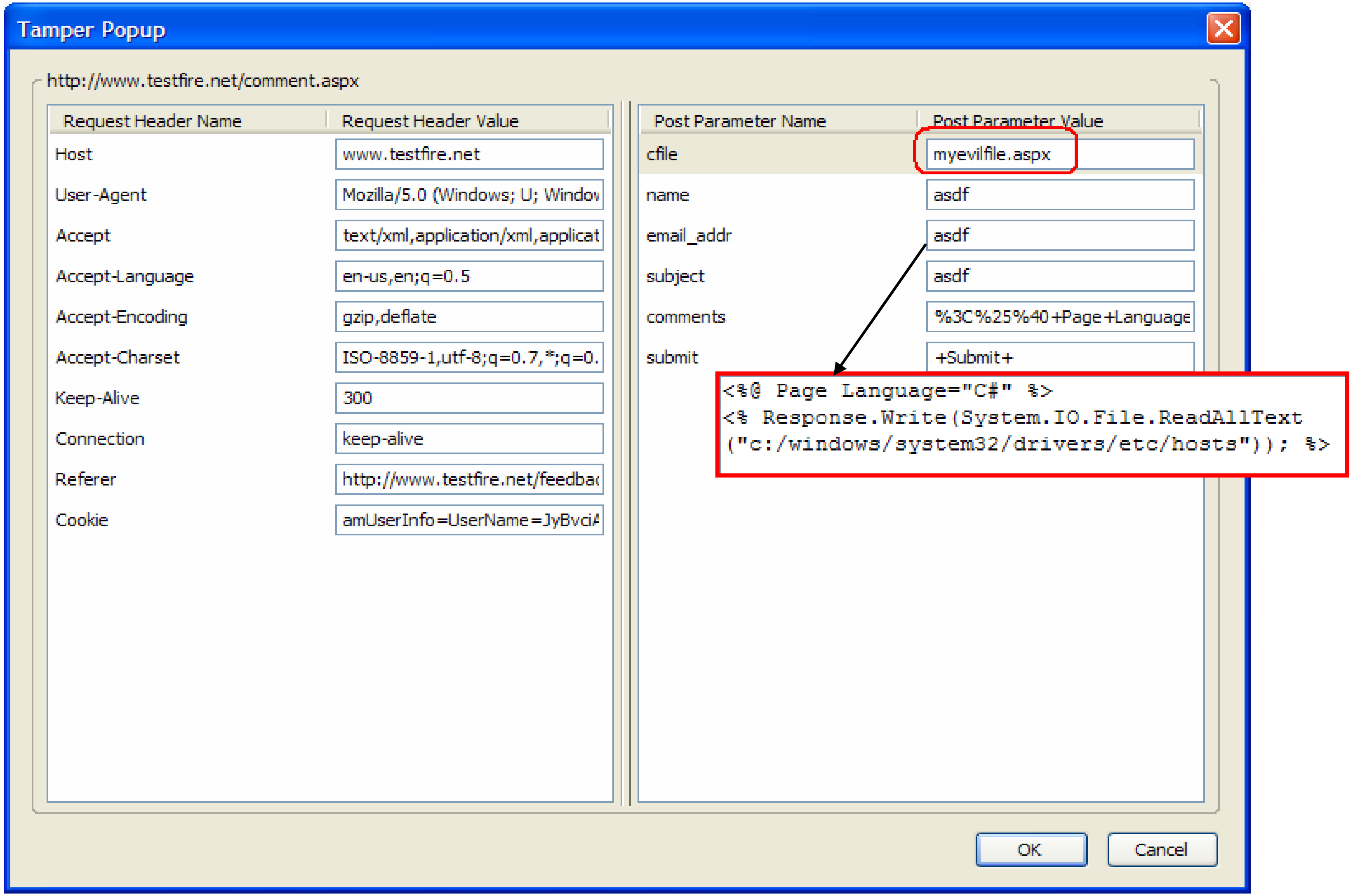


http://www.testfire.net/comment.aspx

Request Header Name	Request Header Value
Host	www.testfire.net
User-Agent	Mozilla/5.0 (Windows; U; Windov
Accept	text/xml,application/xml,applicat
Accept-Language	en-us,en;q=0.5
Accept-Encoding	gzip,deflate
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.
Keep-Alive	300
Connection	keep-alive
Referer	http://www.testfire.net/feedbac
Cookie	ASP.NET_SessionId=adp4vz550

Post Parameter Name	Post Parameter Value
cfile	comments.txt
name	asdf
email_addr	asdf
subject	asdf
comments	asdf
submit	+Submit+

Malicious File Execution – Example cont.



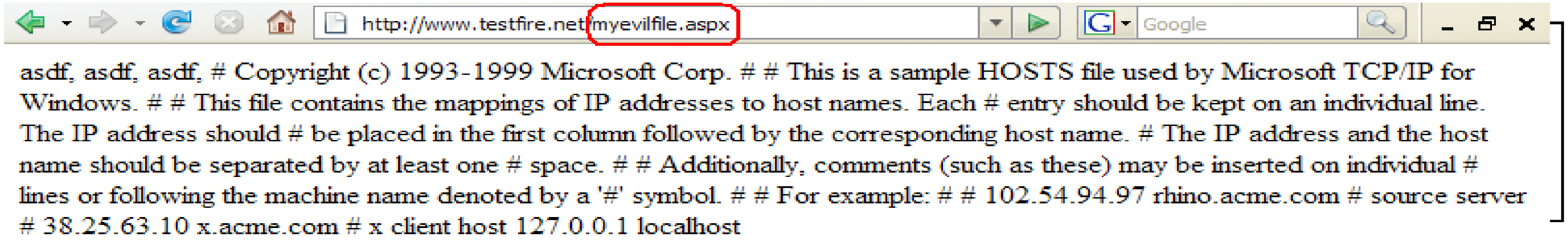
http://www.testfire.net/comment.aspx

Request Header Name	Request Header Value
Host	www.testfire.net
User-Agent	Mozilla/5.0 (Windows; U; Window
Accept	text/xml,application/xml,applicat
Accept-Language	en-us,en;q=0.5
Accept-Encoding	gzip,deflate
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.
Keep-Alive	300
Connection	keep-alive
Referer	http://www.testfire.net/feedbac
Cookie	amUserInfo=UserName=JyBvciA

Post Parameter Name	Post Parameter Value
cfile	myevilfile.aspx
name	asdf
email_addr	asdf
subject	asdf
comments	%3C%25%40+Page+Language
submit	+Submit+

```
<%@ Page Language="C#" %>
<% Response.Write(System.IO.File.ReadAllText
("c:/windows/system32/drivers/etc/hosts")); %>
```

Malicious File Execution – Example cont.



asdf, asdf, asdf, # Copyright (c) 1993-1999 Microsoft Corp. # # This is a sample HOSTS file used by Microsoft TCP/IP for Windows. # # This file contains the mappings of IP addresses to host names. Each # entry should be kept on an individual line. The IP address should # be placed in the first column followed by the corresponding host name. # The IP address and the host name should be separated by at least one # space. # # Additionally, comments (such as these) may be inserted on individual # lines or following the machine name denoted by a '#' symbol. # # For example: # # 102.54.94.97 rhino.acme.com # source server # 38.25.63.10 x.acme.com # x client host 127.0.0.1 localhost

4 - Insecure Direct Object Reference

What is it?

- Part or all of a resource (file, table, etc.) name controlled by user input.

What are the implications?

- Access to sensitive resources
- Information Leakage, aids future hacks

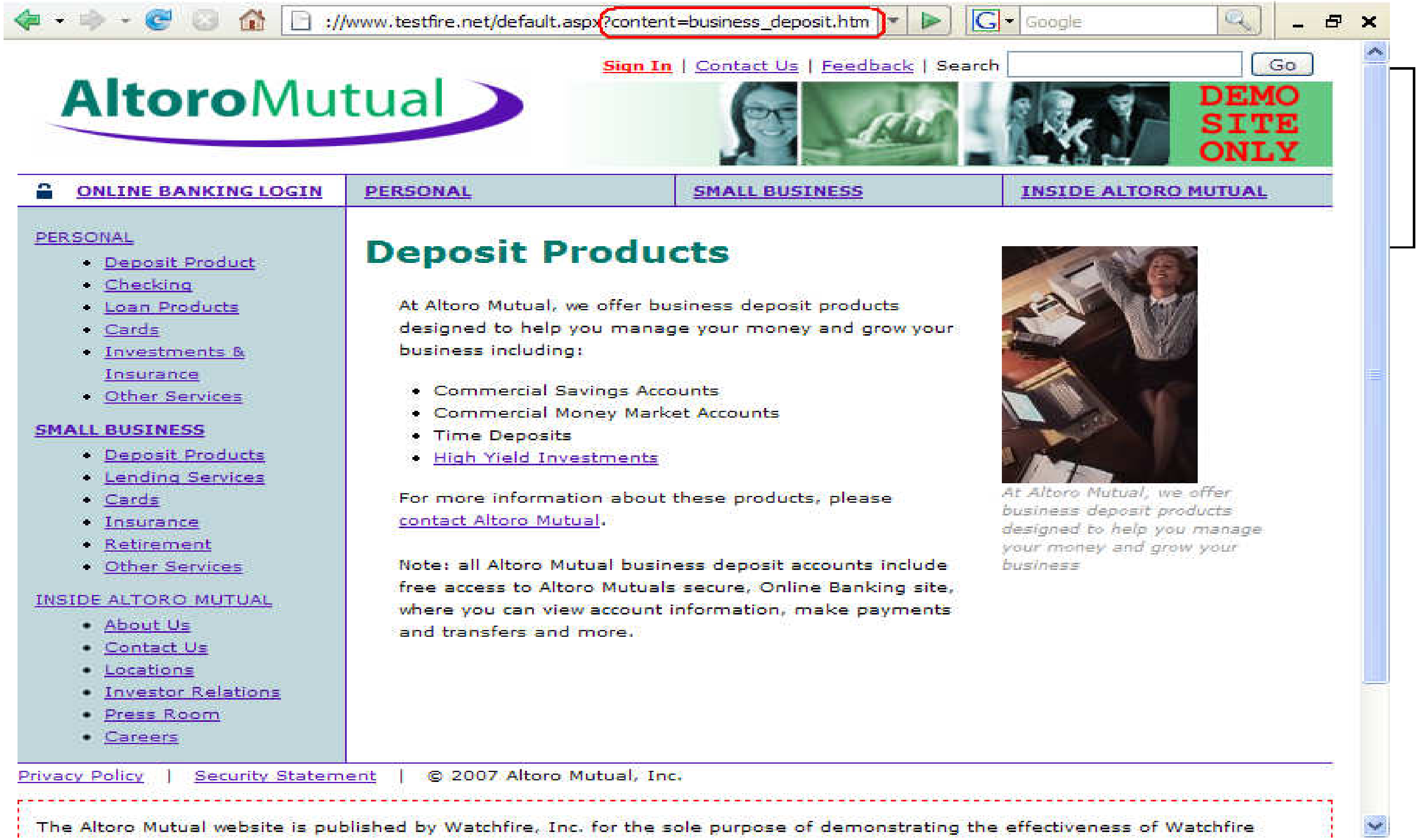
Demonstration – Insecure Direct Object References

Main points covered in the demo or video:

- Demonstrating how to extract files from the host system using the poison null byte attack

CLICK HERE TO LAUNCH DEMO

Insecure Direct Object Reference - Example



://www.testfire.net/default.aspx?content=business_deposit.htm
Google
Sign In | Contact Us | Feedback | Search
Go
AltoroMutual
DEMO SITE ONLY
ONLINE BANKING LOGIN
PERSONAL
SMALL BUSINESS
INSIDE ALTORO MUTUAL

PERSONAL

- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS

- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

INSIDE ALTORO MUTUAL

- About Us
- Contact Us
- Locations
- Investor Relations
- Press Room
- Careers

Deposit Products

At Altoro Mutual, we offer business deposit products designed to help you manage your money and grow your business including:

- Commercial Savings Accounts
- Commercial Money Market Accounts
- Time Deposits
- High Yield Investments

For more information about these products, please contact Altoro Mutual.

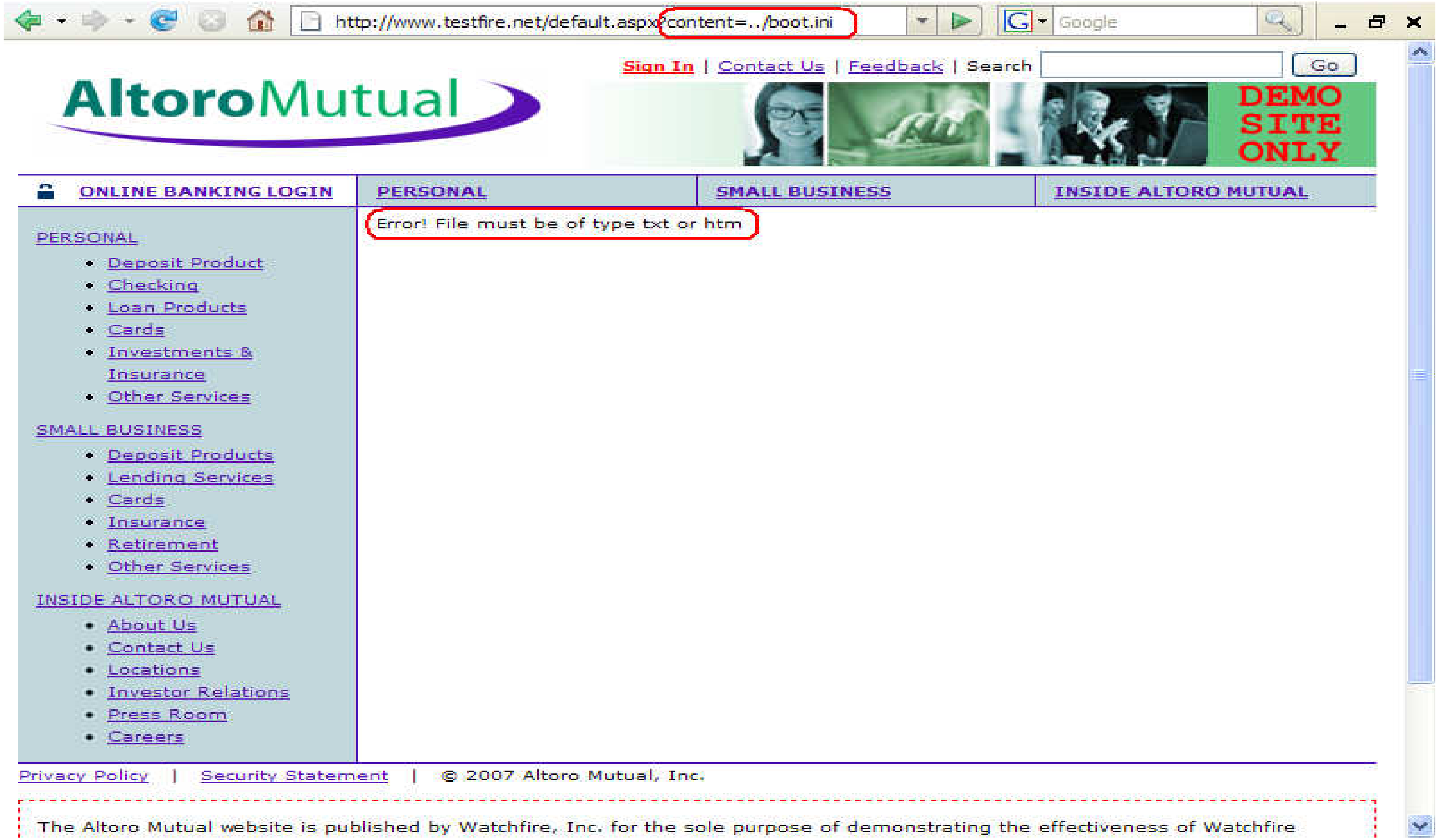
Note: all Altoro Mutual business deposit accounts include free access to Altoro Mutuals secure, Online Banking site, where you can view account information, make payments and transfers and more.

At Altoro Mutual, we offer business deposit products designed to help you manage your money and grow your business

Privacy Policy | Security Statement | © 2007 Altoro Mutual, Inc.

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire

Insecure Direct Object Reference – Example Cont.

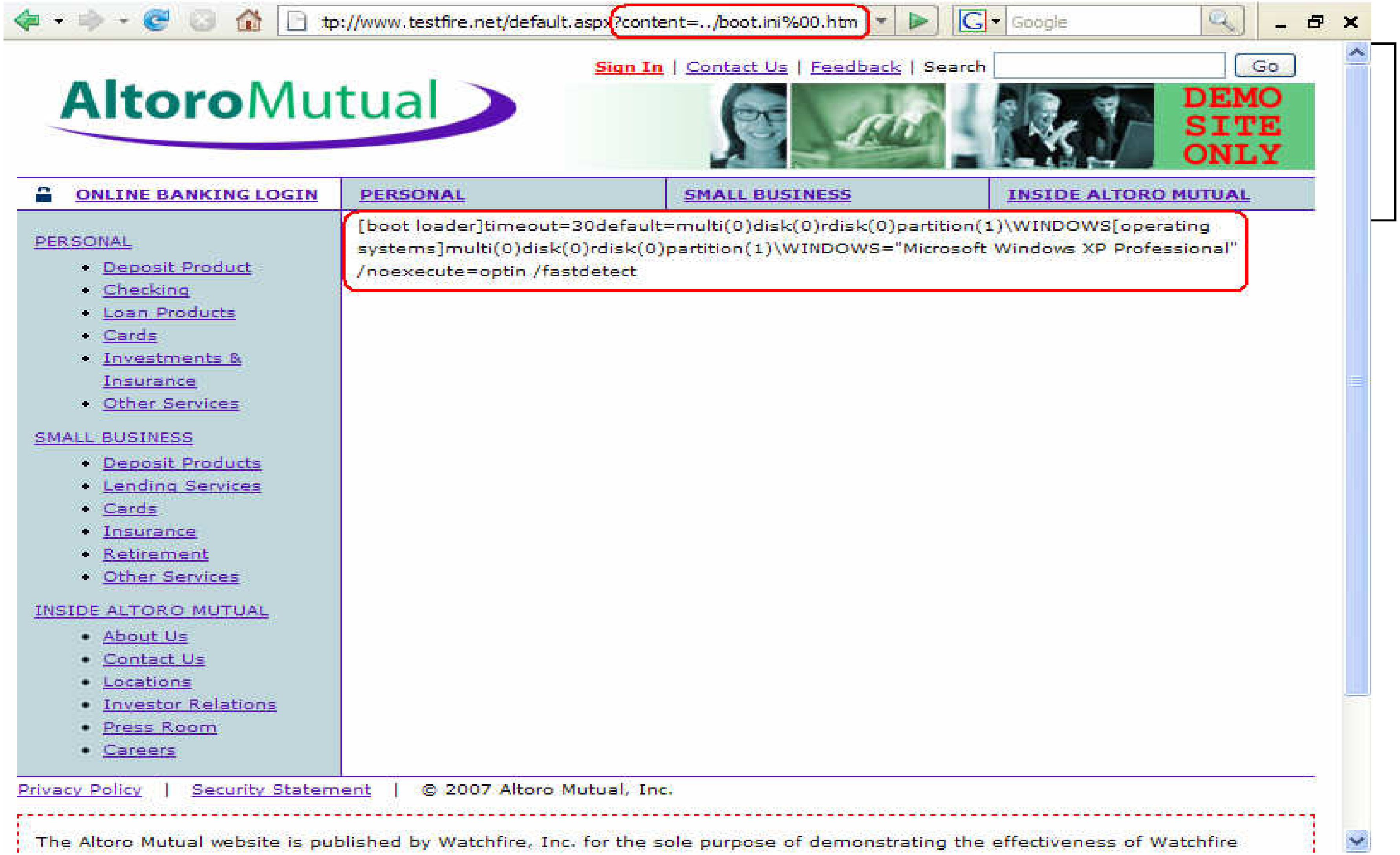


http://www.testfire.net/default.aspx?content=../boot.ini

Error! File must be of type txt or htm

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire

Insecure Direct Object Reference – Example Cont.



```
tp://www.testfire.net/default.aspx?content=../boot.ini%00.htm
```

```
[boot loader]timeout=30default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS[operating
systems]multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional"
/noexecute=optin /fastdetect
```

Privacy Policy | Security Statement | © 2007 Altoro Mutual, Inc.

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire

Organizations must mitigate the risk!

Organizations need to mitigate the risk of a Web Application Security breach!

They need to find and **remediate** vulnerabilities in their Web Applications before they are exploited by Hackers

IBM Rational AppScan is the tool to help them do this!

Demo of IBM Rational AppScan Empowering Organizations to mitigate risk

IBM Rational Software Development Conference 2008

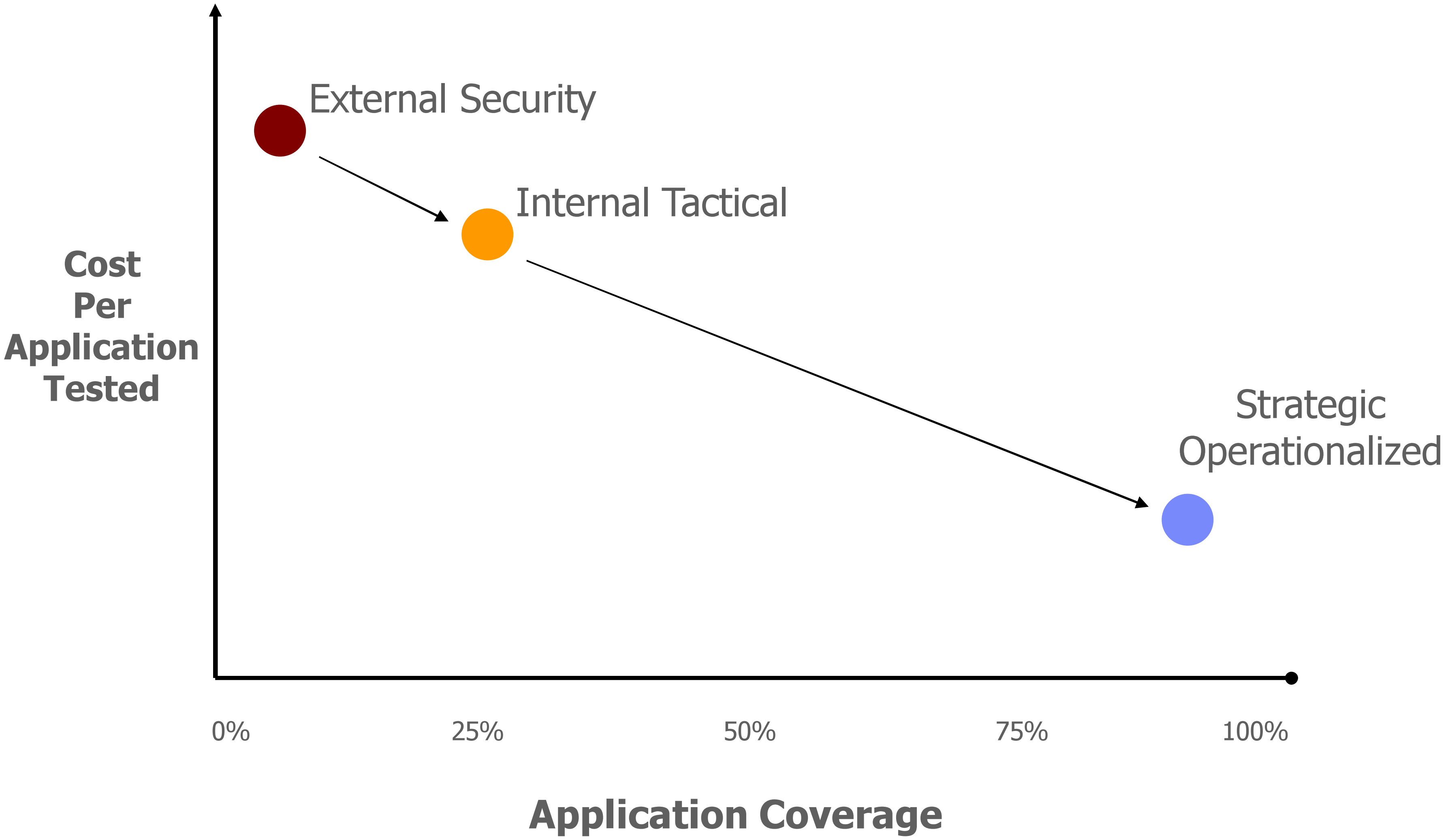
WHERE TEAMS ARE R-HEROES

IBM

Rational. software

When to test your Applications for Security Defects ??….

Application Security Maturity Model



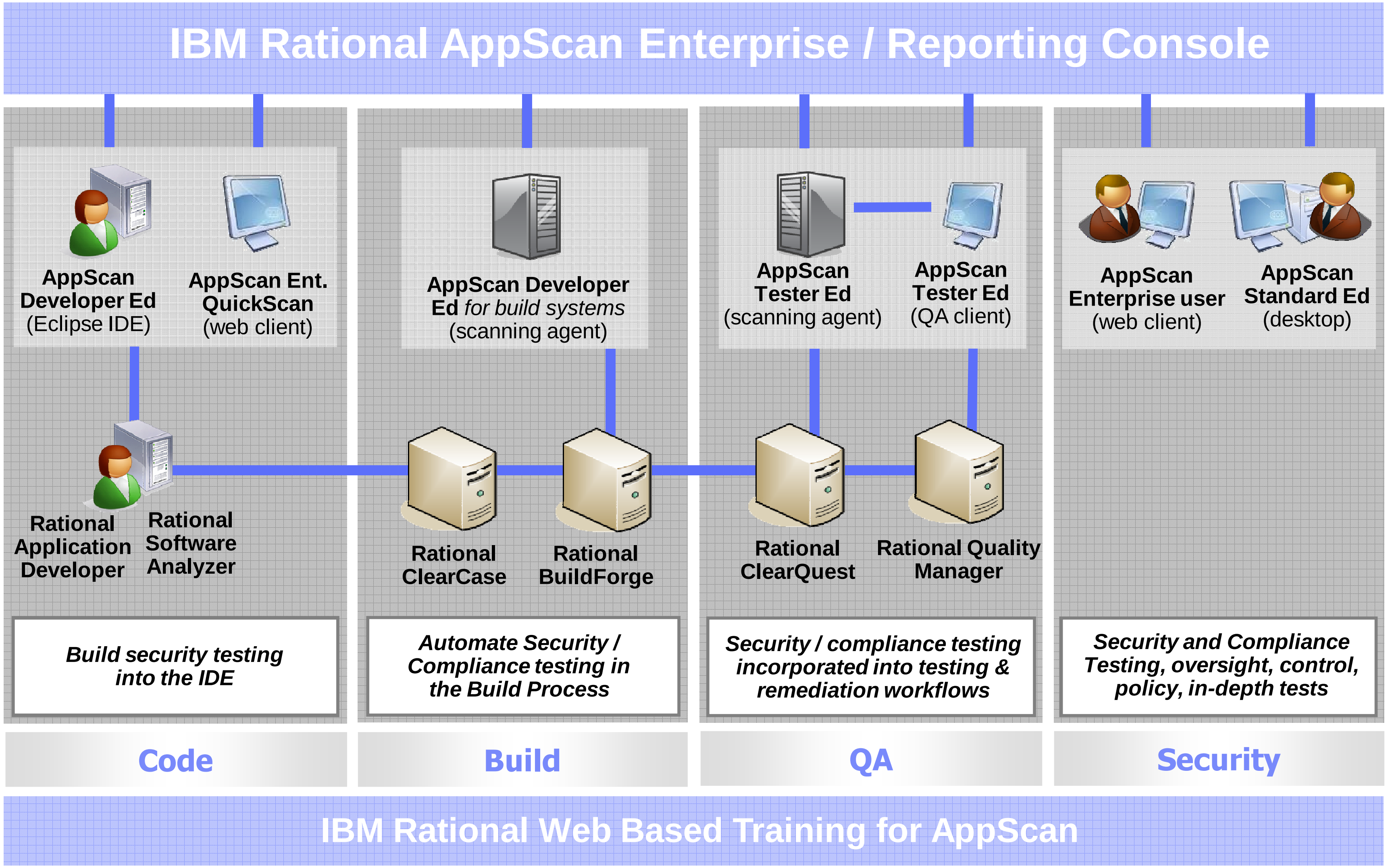
Testing early reduces cost and time to market

	Found in Design	Found in Coding	Found in Integration	Found in Beta	Found in GA
Design Errors	1x	5x	10x	15x	30x
Coding Errors		1x	10x	20x	30x
Integration Errors			1x	10x	20x

* *http://www.nist.gov/director/prog-ofc/report02-3.pdf*

IBM Rational AppScan SDLC Ecosystem

Includes Fall 2008 Releases



IBM Rational AppScan Enterprise / Reporting Console

AppScan Developer Ed (Eclipse IDE)

AppScan Ent. QuickScan (web client)

AppScan Developer Ed *for build systems* (scanning agent)

AppScan Tester Ed (scanning agent)

AppScan Tester Ed (QA client)

AppScan Enterprise user (web client)

AppScan Standard Ed (desktop)

Rational Application Developer

Rational Software Analyzer

Rational ClearCase

Rational BuildForge

Rational ClearQuest

Rational Quality Manager

Build security testing into the IDE

Automate Security / Compliance testing in the Build Process

Security / compliance testing incorporated into testing & remediation workflows

Security and Compliance Testing, oversight, control, policy, in-depth tests

Code

Build

QA

Security

IBM Rational Web Based Training for AppScan

Q&A

IBM Rational Software Development Conference 2008

WHERE TEAMS ARE R-HEROES

IBM

Rational. software