Software Survivability

Survivability Concepts

Survivability Motivation

- Growing societal dependence on complex, large-scale, networked systems
  - Sectors: commercial, government, defense, ...
  - Infrastructure: telecom, transportation, utilities, …
  - Interdependencies and cascade failures
- Serious consequences of system compromises and failures
- Presidential Commission on Critical Infrastructure Protection

The Changing System Environment

- Expanding network boundaries and connectivity
- Blurring of Intranets and Extranets
- Heterogeneous mix of participants with varying trust
- Lack of central administrative control
- Unknown components: COTS, Java, …
- Point security solutions: PKI, VPN, IDS, firewalls, ...

The fundamental limitation of security:

No amount of security can guarantee a system will not be penetrated

Survivability Defined I

Survivability is the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents.

- Focus is on the <u>continuity and recovery</u> of the <u>system mission</u>

- Imperfect defenses are assumed

Survivability Defined II

- Survivability differs from conventional security
  - Security focuses on static perimeter defenses
  - Survivability focuses on design and operation to maintain mission support in adverse environments

- Survivability differs from dependability
  - Dependability focuses on random faults
  - Survivability focuses on coordinated attacks by intelligent adversaries

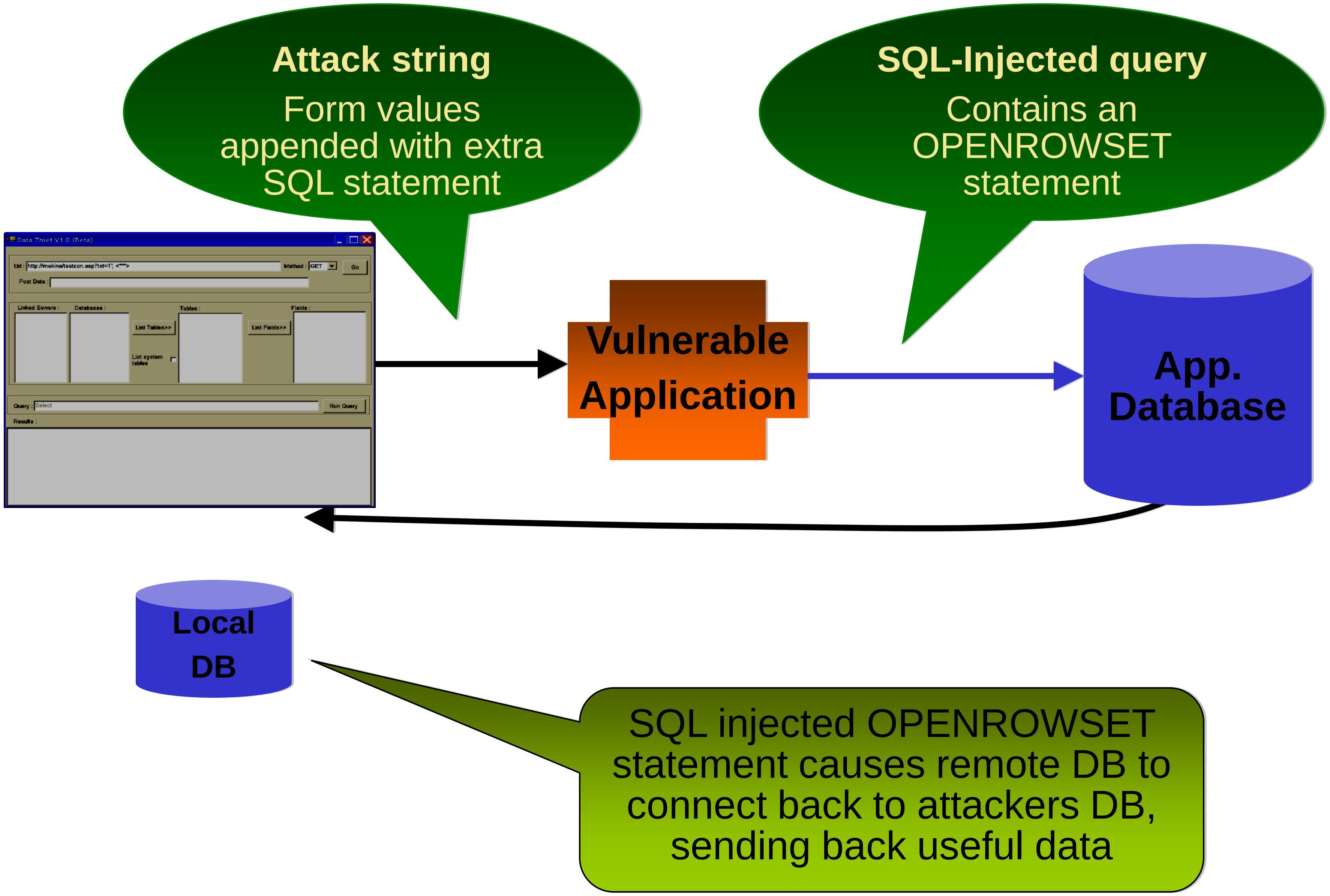
The "*Three R's*" of Survivability

- *Resistance*
  - Capability to deter attacks
- *Recognition*
  - Capability to recognize attacks and extent of damage
- *Recovery*
  - Capability to provide essential services/assets during attack and recover full services after attack

The Myth

“Our site is safe”:

- We have firewalls in place
- We encrypt our data
- We have a privacy policy


Url : http://makina/testcon.asp?txt=1'; <***>
Method : GET
Post Data :
Query : Select

Black Hat Community Sharing

Port Scanners

Brute Force pwd crackers

Cracker Tools

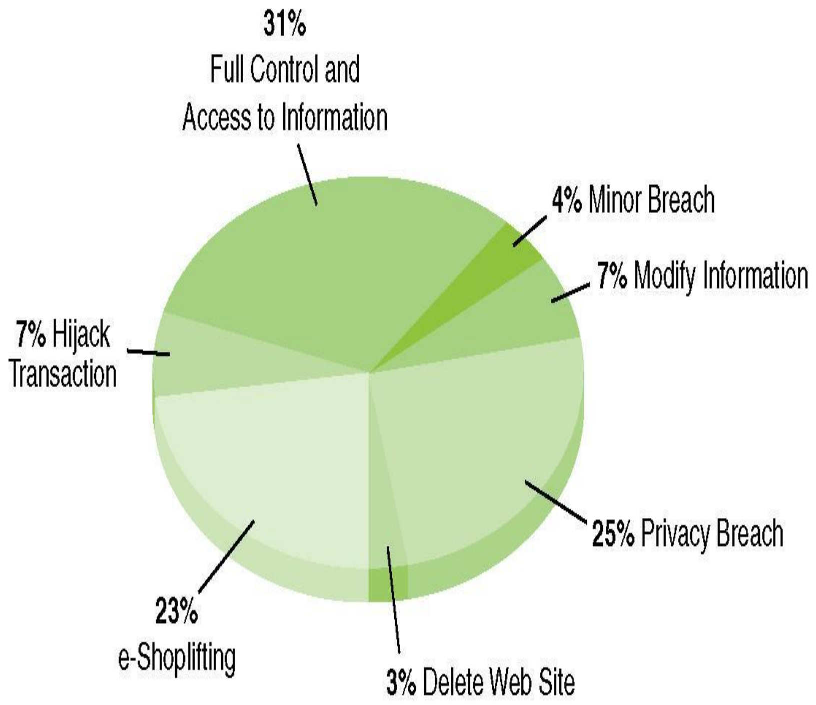
Network Sniffers

Dictionary Based pwd crackers

De-compilers

Debuggers

Application Security Defects



167 Audits conducted – 98% vulnerable: all had firewalls and encryption solutions in place…

Frequent

- 3 out of 4 business websites are vulnerable to attack (Gartner)

Pervasive

- 75% of hacks occur at the Application level (Gartner)

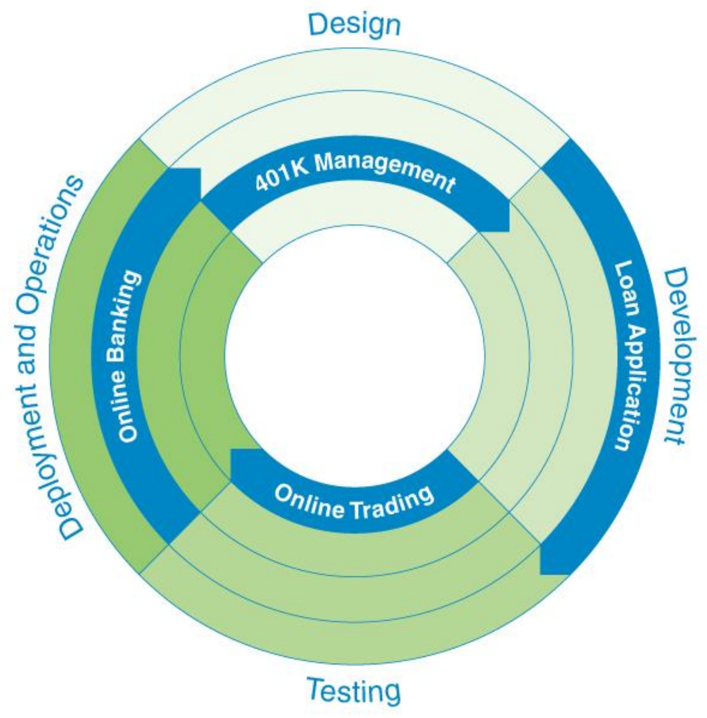
Undetected

- QA testing tools not designed to detect security defects in applications
- Manual patching - reactive, time consuming and *expensive*

Dangerous

- When exploited, security defects destroy company value and customer trust

Pressures on the Application Lifecycle



Financial Services Application

Time-to-Market

- Bringing new applications to market quickly

Complexity is Growing

- Increasing application lifecycle complexity

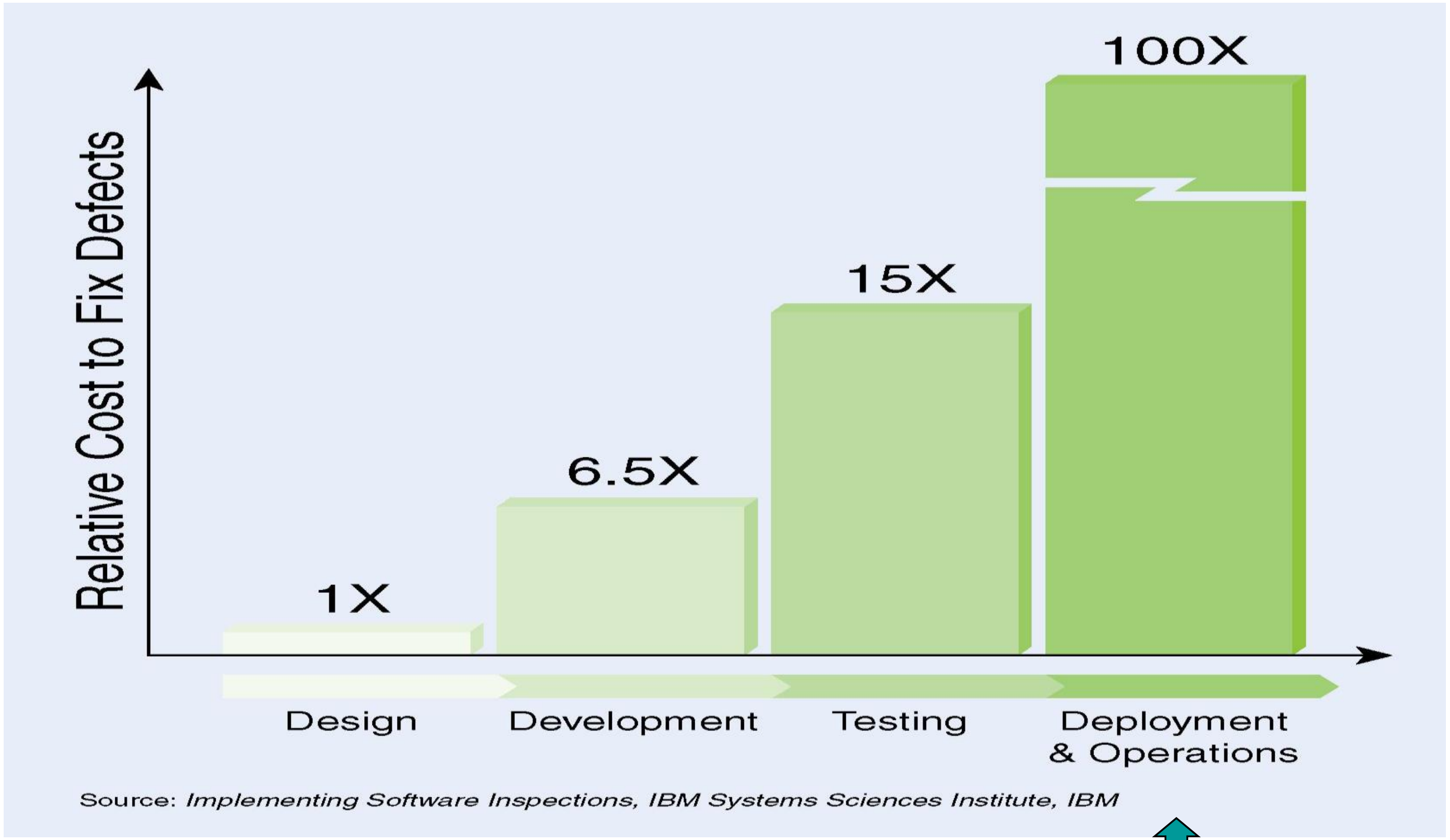
Increasing Business Risks Driven by Security Defects

- Hacker activity increasing
- Government scrutiny and regulation increasing
- Liability precedents for security defects

Costs Escalate Dramatically the longer you wait to Find and Fix

- Bad software costs the economy $59.5 billion a year- cost of breakdowns and repairs (Nat. Institute of Standards & Technology, May 2002)

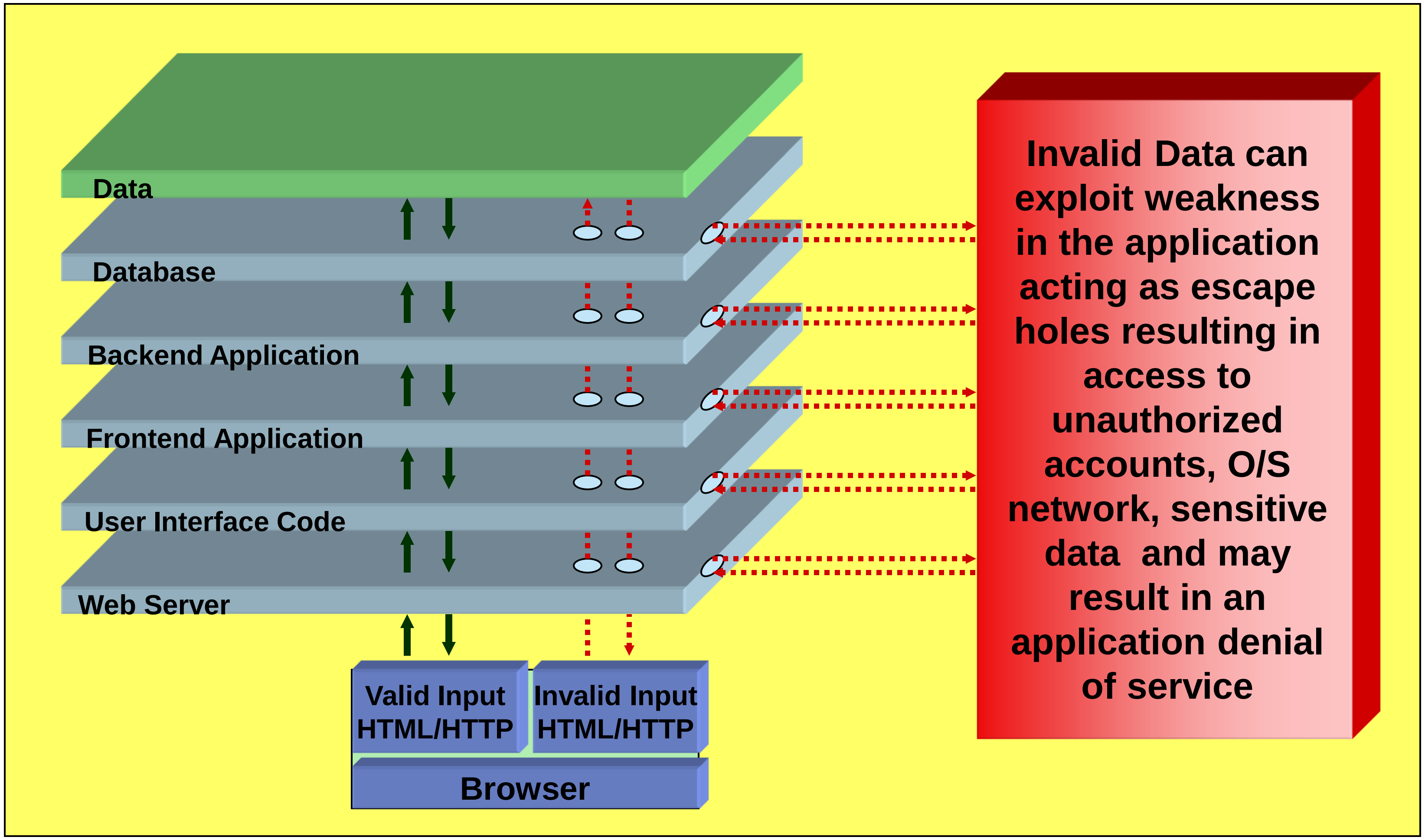
Cost Increases Later in the Lifecycle Security is Addressed



Source: *Implementing Software Inspections, IBM Systems Sciences Institute, IBM*

Cost to Fix dramatically increases the longer you wait to test

Web Application Vulnerabilities



Invalid Data can exploit weakness in the application acting as escape holes resulting in access to unauthorized accounts, O/S network, sensitive data and may result in an application denial of service

Without any protection,
holes and backdoors exist at every layer waiting to be exploited

Types of Application Hacks

Through a browser, a hacker can use the smallest bug or backdoor to change, or pervert, the intent of the application

Application	Attack Types	Negative Outcome Examples
Form field: collect data	Buffer overflow	Crash servers/close business
Customer account	Cookie poisoning	Identity theft/illegal transactions
Online shopping	Hidden fields	eShoplifting
Sloppy code	Backdoors/Debug options	Download proprietary database
Text Field: collect data	Cross Site scripting	eHijacking - Get account info
Database	Parameter Tampering/SQL injection	Fraud
Backend Apps	Stealth Commanding	Site defacement
Web Server	Published Vulnerabilities	Crash site
Front end Apps	3rd Party Misconfiquration	Admin access
Web Server	Forceful Browsing	Access sensitive data

10 Types of Attacks: Development Lifecycle

	Development	Operations	
APP. BUFFER OVERFLOW	●		
COOKIE POISONING	●	●	
CROSS SITE SCRIPTING	●		
HIDDEN MANIPULATION	●		
STEALTH COMMANDING	●		
3RD PARTY MISCONFIG.		●	
KNOWN VULNERABILITIES		●	
PARAMETER TAMPERING	●		
BACKDOORS & DEBUG OPT.	●	●	3rd party SW
FORCEFUL BROWSING		●	

Hidden Field Manipulation

- ***Vulnerability explanation***:

  The application sends data to the client using a hidden field in a form. Modifying the hidden field damages the data returning to the web application

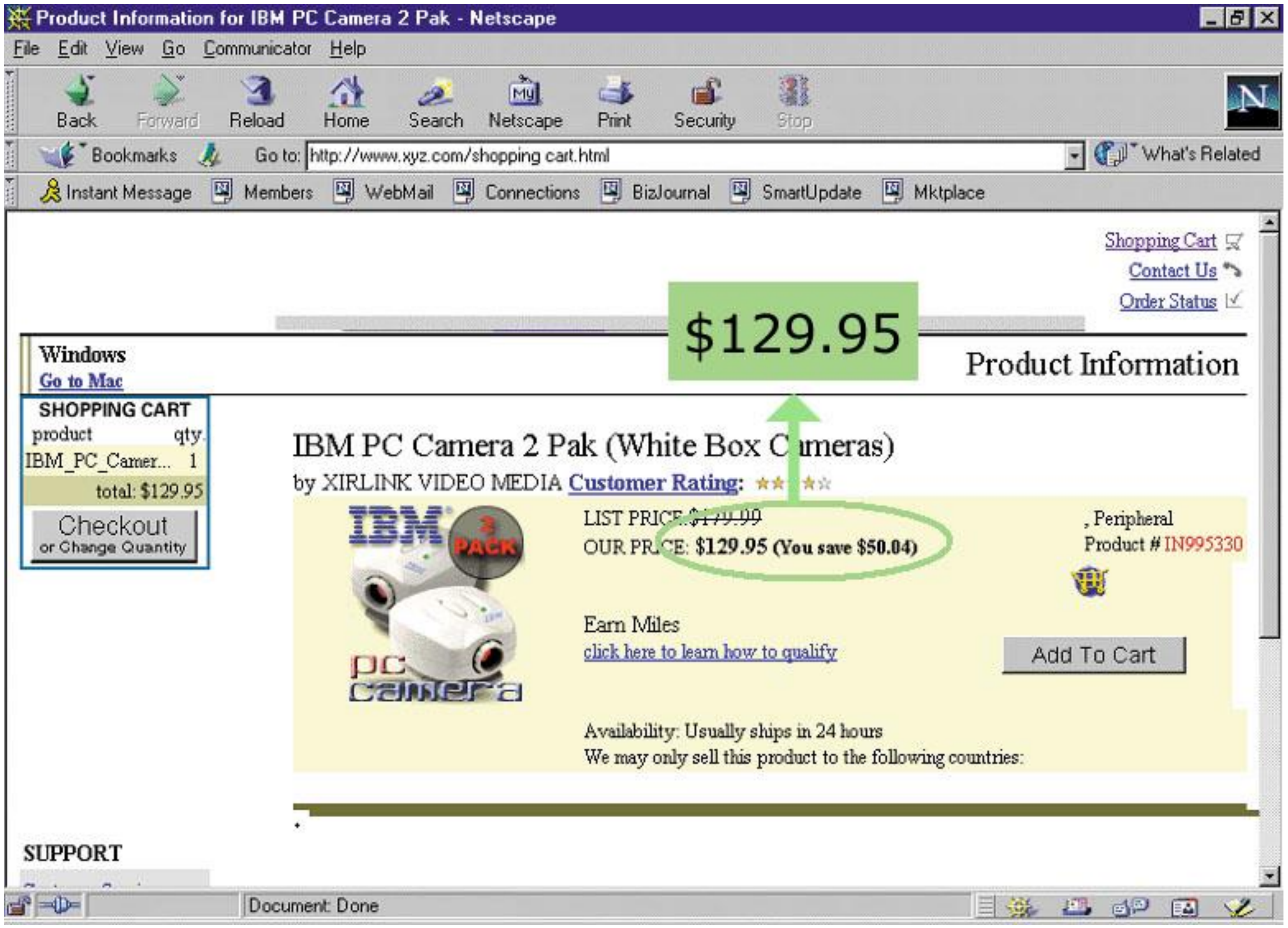
- ***Why Hidden Field Manipulation***:

  Passing hidden fields is a simple and efficient way to pass information from one part of the application to another (or between two applications) without the use of complex backend systems.

- ***As a result of this manipulation*** :

  The application acts according to the changed information and not according to the original data

Hidden Field Manipulation - Example



Product Information for IBM PC Camera 2 Pak - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Go to: http://www.xyz.com/shopping cart.html What's Related

Instant Message Members WebMail Connections BizJournal SmartUpdate Mktplace

Shopping Cart
Contact Us
Order Status

$129.95

Windows
Go to Mac

Product Information

SHOPPING CART

product	qty.
IBM_PC_Camer...	1
total: $129.95	

Checkout
or Change Quantity

IBM PC Camera 2 Pak (White Box Cameras)

by XIRLINK VIDEO MEDIA Customer Rating:

LIST PRICE: ~~$179.99~~
OUR PRICE: $129.95 (You save $50.04)

, Peripheral
Product # IN995330

Earn Miles
click here to learn how to qualify

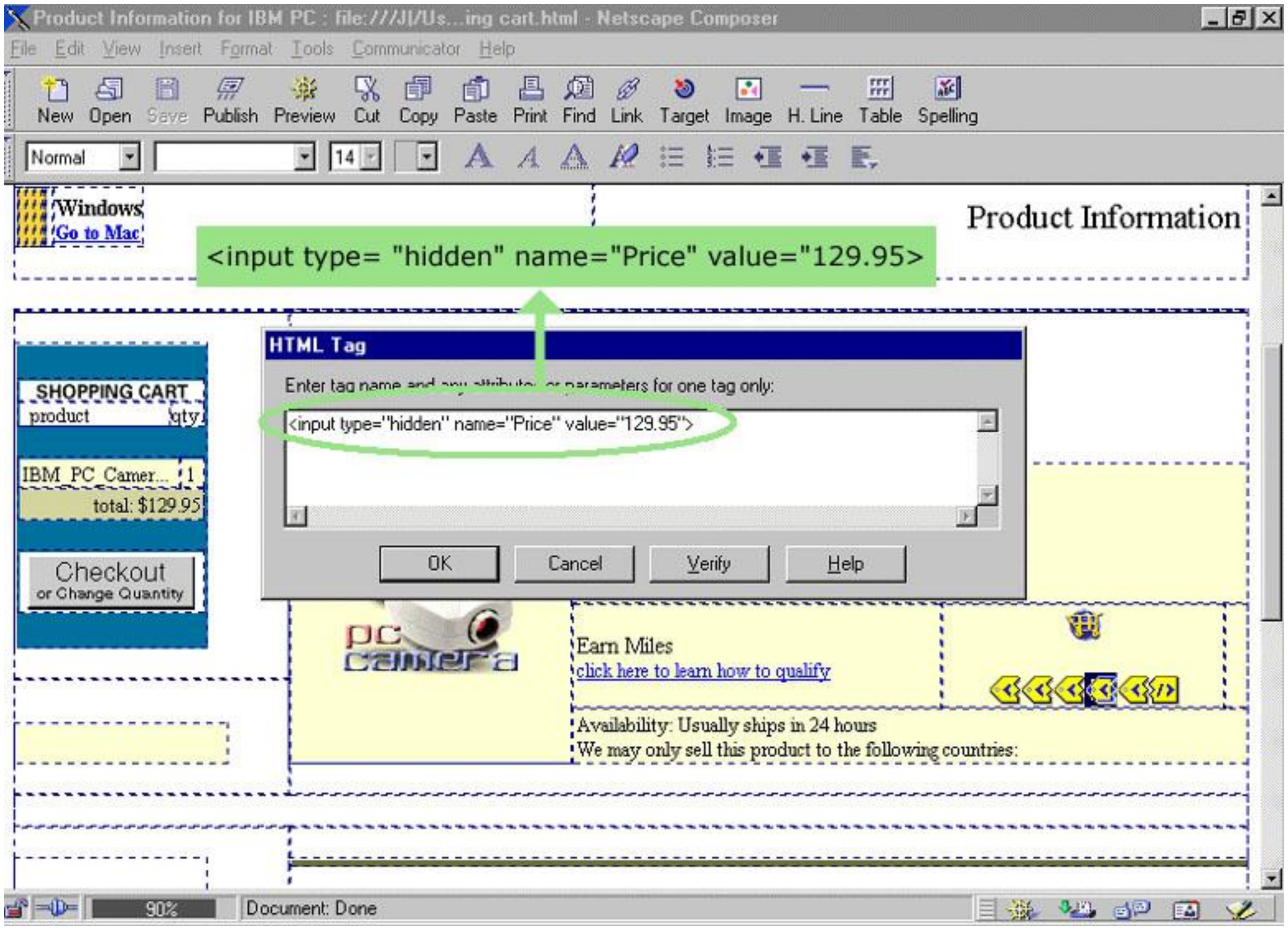
Add To Cart

Availability: Usually ships in 24 hours
We may only sell this product to the following countries:

SUPPORT

Document: Done

Hidden Field Manipulation - Example


Product Information for IBM PC : file:///J|/Us...ing cart.html - Netscape Composer
File Edit View Insert Format Tools Communicator Help
New Open Save Publish Preview Cut Copy Paste Print Find Link Target Image H. Line Table Spelling
Normal
14
Windows
Go to Mac
Product Information
SHOPPING CART
product qty
IBM_PC_Camer... 1
total: $129.95
Checkout
or Change Quantity
HTML Tag
OK
Cancel
Verify
Help
Earn Miles
click here to learn how to qualify
90%
Document: Done

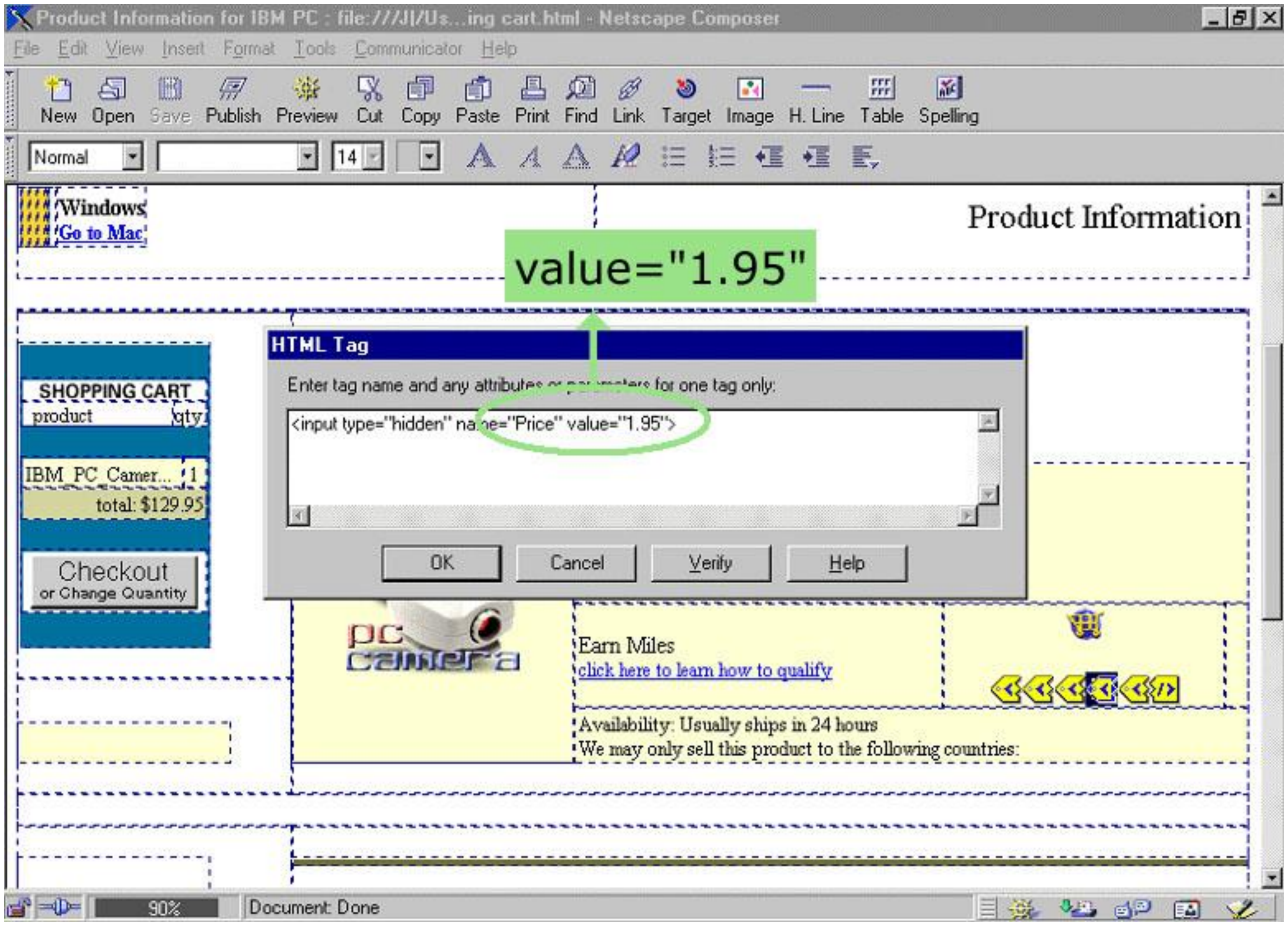
<input type= "hidden" name="Price" value="129.95>

Enter tag name and any attributes or parameters for one tag only:

<input type="hidden" name="Price" value="129.95">

Availability: Usually ships in 24 hours
We may only sell this product to the following countries:

Hidden Field Manipulation - Example

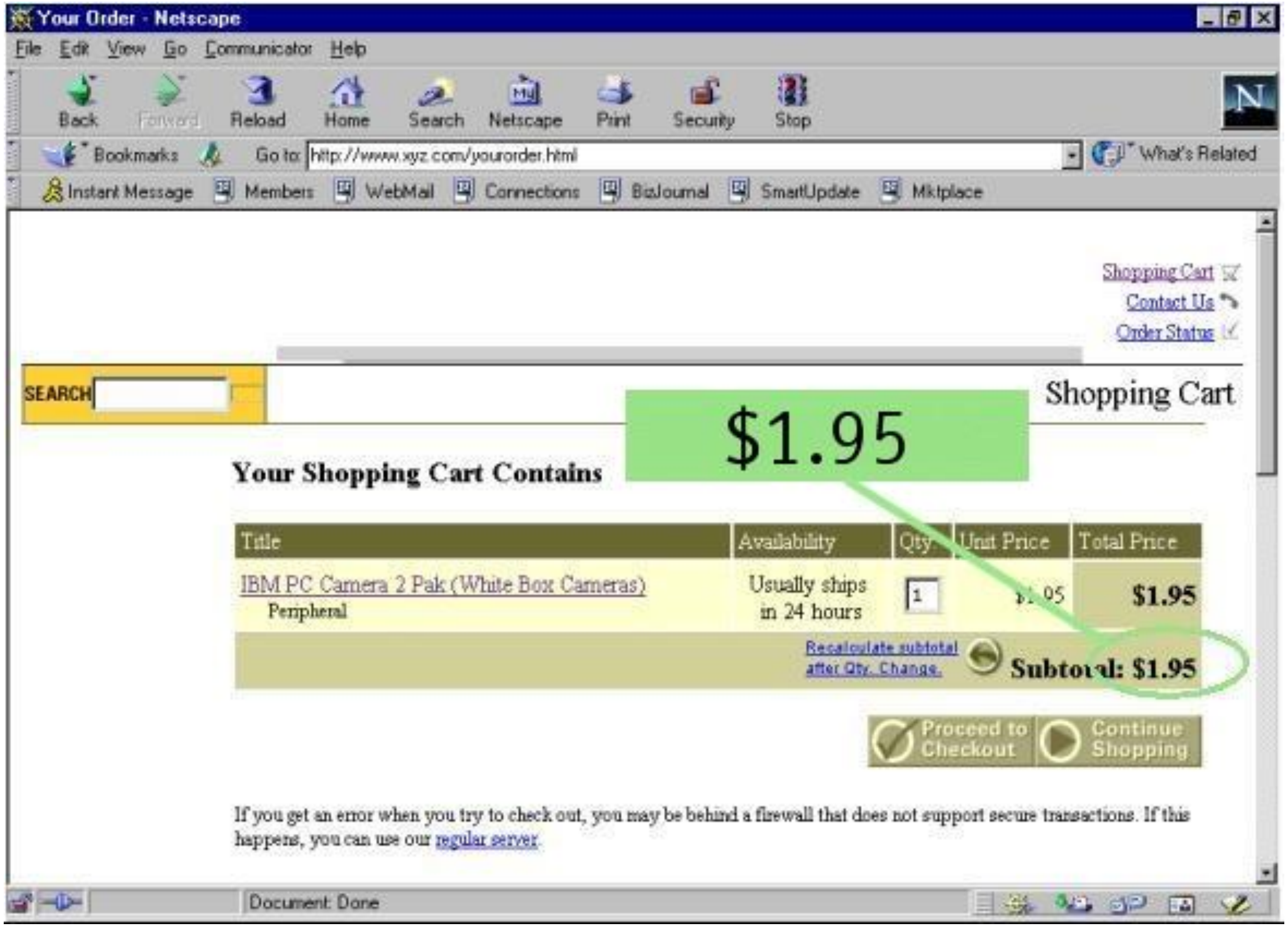

Product Information for IBM PC : file:///J|/Us...ing cart.html - Netscape Composer
File Edit View Insert Format Tools Communicator Help
New Open Save Publish Preview Cut Copy Paste Print Find Link Target Image H. Line Table Spelling
Normal
14
Windows
Go to Mac
Product Information
value="1.95"
HTML Tag
SHOPPING CART
product qty
IBM PC Camer... 1
total: $129.95
Checkout or Change Quantity
OK
Cancel
Verify
Help
Earn Miles
click here to learn how to qualify
90%
Document: Done

Enter tag name and any attributes or parameters for one tag only:

```
<input type="hidden" name="Price" value="1.95">
```

Availability: Usually ships in 24 hours
We may only sell this product to the following countries:

Hidden Field Manipulation - Example


Your Order - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go to: http://www.xyz.com/yourorder.html What's Related
Instant Message Members WebMail Connections BizJournal SmartUpdate Mktplace
Shopping Cart
Contact Us
Order Status
SEARCH
Shopping Cart
$1.95
Proceed to Checkout
Continue Shopping
Document: Done

Your Shopping Cart Contains

Title	Availability	Qty	Unit Price	Total Price
IBM PC Camera 2 Pak (White Box Cameras) Peripheral	Usually ships in 24 hours	1	$1.95	$1.95

Recalculate subtotal after Qty. Change. **Subtotal: $1.95**

If you get an error when you try to check out, you may be behind a firewall that does not support secure transactions. If this happens, you can use our regular server.

Backdoor & Debug options

- ***Vulnerability explanation***:

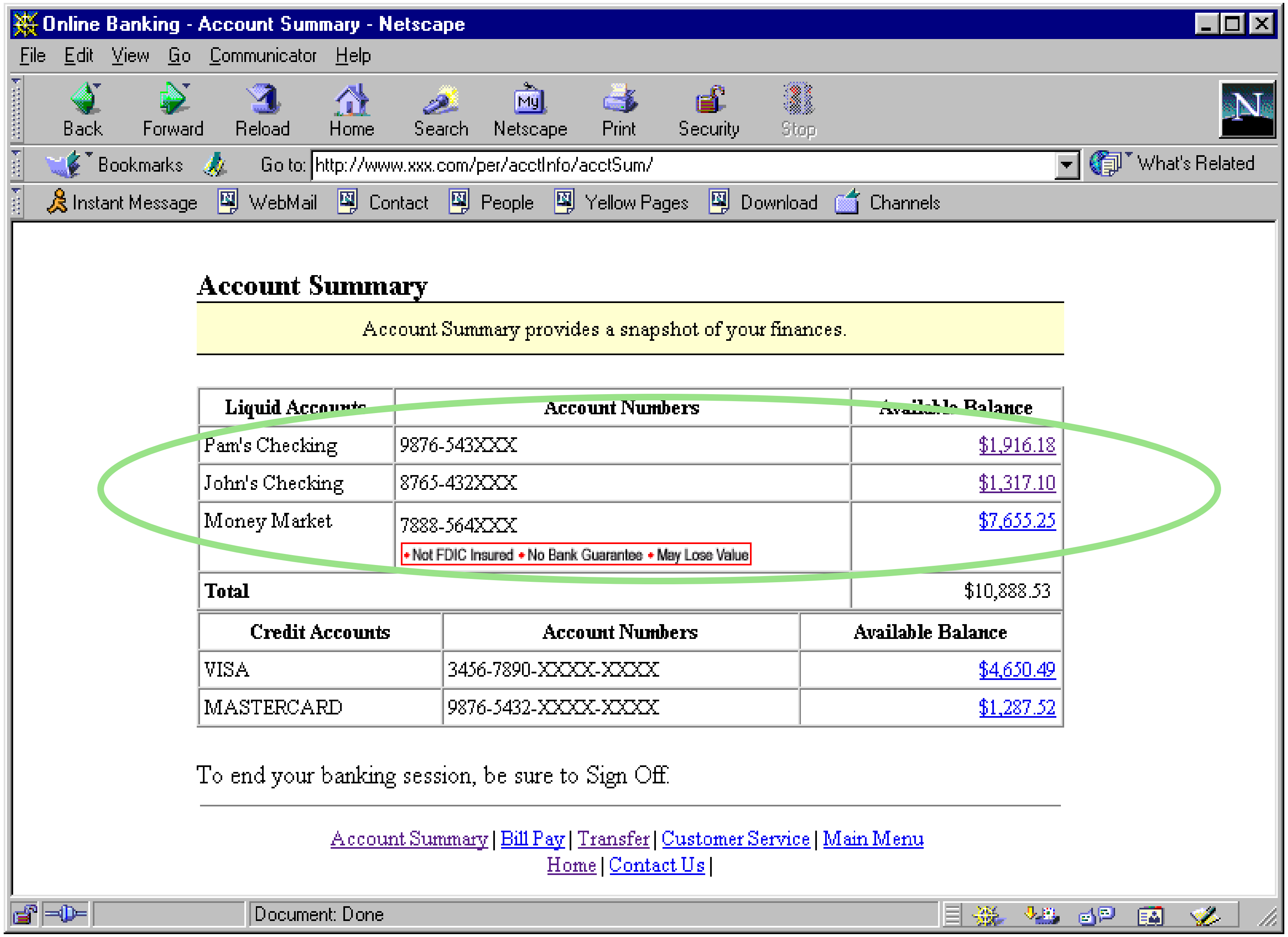
  The application has hidden debug options that can be activated by sending a specific parameter or sequence

- ***Why Backdoor and Debug options***:
  - Leaving debug options in the code enables developers to find and fix bugs faster
  - Developers leave backdoors as a way of guaranteeing their access to the system

- ***As a result of this manipulation*** :

  Activation of the hidden debug option allows the hacker to have extreme access to the application (usually unlimited).

Backdoor & Debug options - Example


Online Banking - Account Summary - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go to: http://www.xxx.com/per/acctInfo/acctSum/
What's Related
Instant Message WebMail Contact People Yellow Pages Download Channels
Document: Done

Account Summary

Account Summary provides a snapshot of your finances.

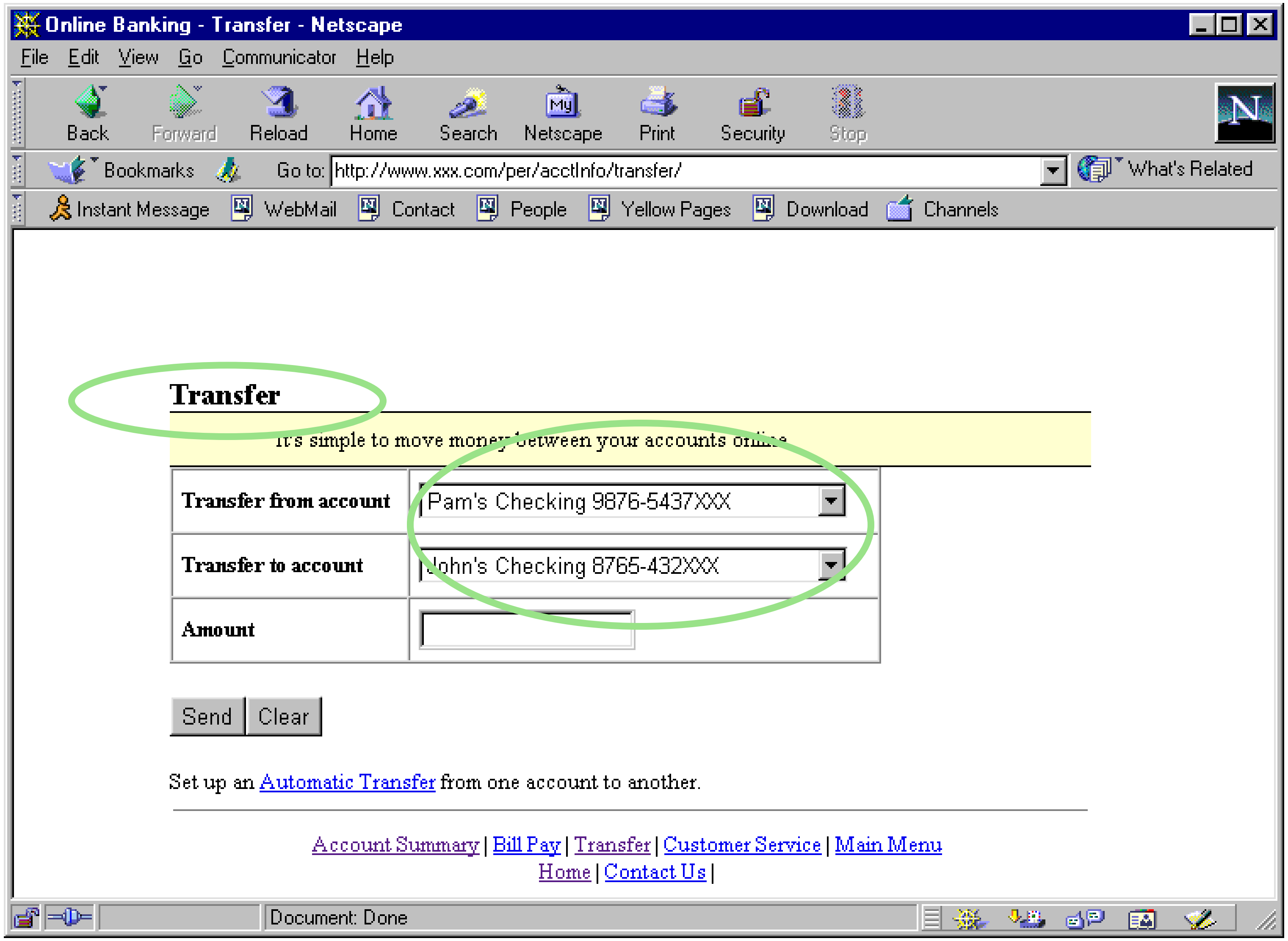
Liquid Accounts	Account Numbers	Available Balance
Pam's Checking	9876-543XXX	$1,916.18
John's Checking	8765-432XXX	$1,317.10
Money Market	7888-564XXX • Not FDIC Insured • No Bank Guarantee • May Lose Value	$7,655.25
Total		$10,888.53

Credit Accounts	Account Numbers	Available Balance
VISA	3456-7890-XXXX-XXXX	$4,650.49
MASTERCARD	9876-5432-XXXX-XXXX	$1,287.52

To end your banking session, be sure to Sign Off.

Account Summary | Bill Pay | Transfer | Customer Service | Main Menu
Home | Contact Us |

Backdoor & Debug options - Example



Online Banking - Transfer - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go to: http://www.xxx.com/per/acctInfo/transfer/ What's Related
Instant Message WebMail Contact People Yellow Pages Download Channels
Send Clear
Document: Done

Transfer

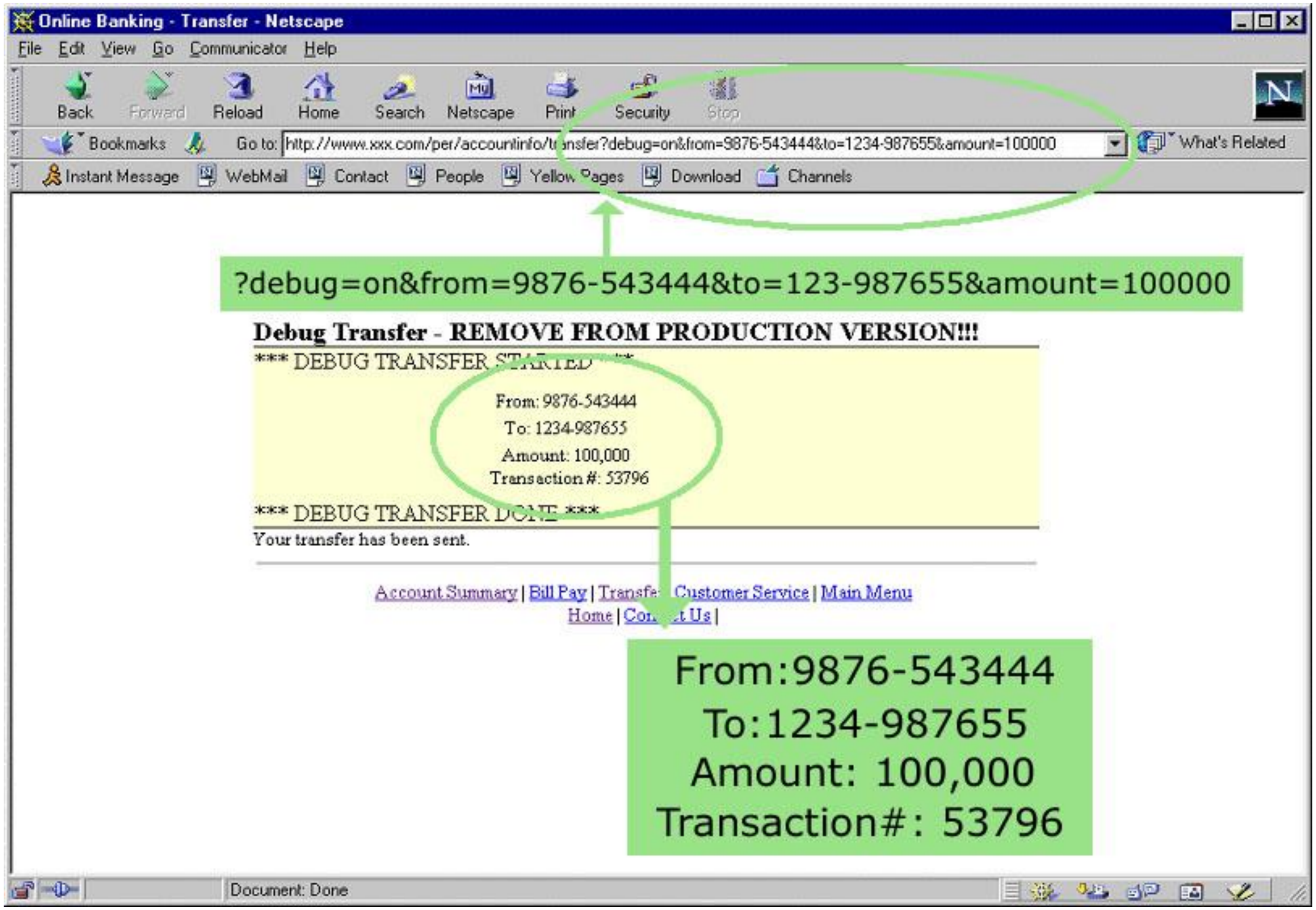
It's simple to move money between your accounts online

Transfer from account	Pam's Checking 9876-5437XXX
Transfer to account	John's Checking 8765-432XXX
Amount	

Set up an Automatic Transfer from one account to another.

Account Summary | Bill Pay | Transfer | Customer Service | Main Menu
Home | Contact Us |

Backdoor & Debug options - Example


Online Banking - Transfer - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks
Go to: http://www.xxx.com/per/accountinfo/transfer?debug=on&from=9876-543444&to=1234-987655&amount=100000
What's Related
Instant Message WebMail Contact People Yellow Pages Download Channels
Document: Done

?debug=on&from=9876-543444&to=123-987655&amount=100000

Debug Transfer - REMOVE FROM PRODUCTION VERSION!!!

*** DEBUG TRANSFER STARTED ***

From: 9876-543444
To: 1234-987655
Amount: 100,000
Transaction #: 53796

*** DEBUG TRANSFER DONE ***

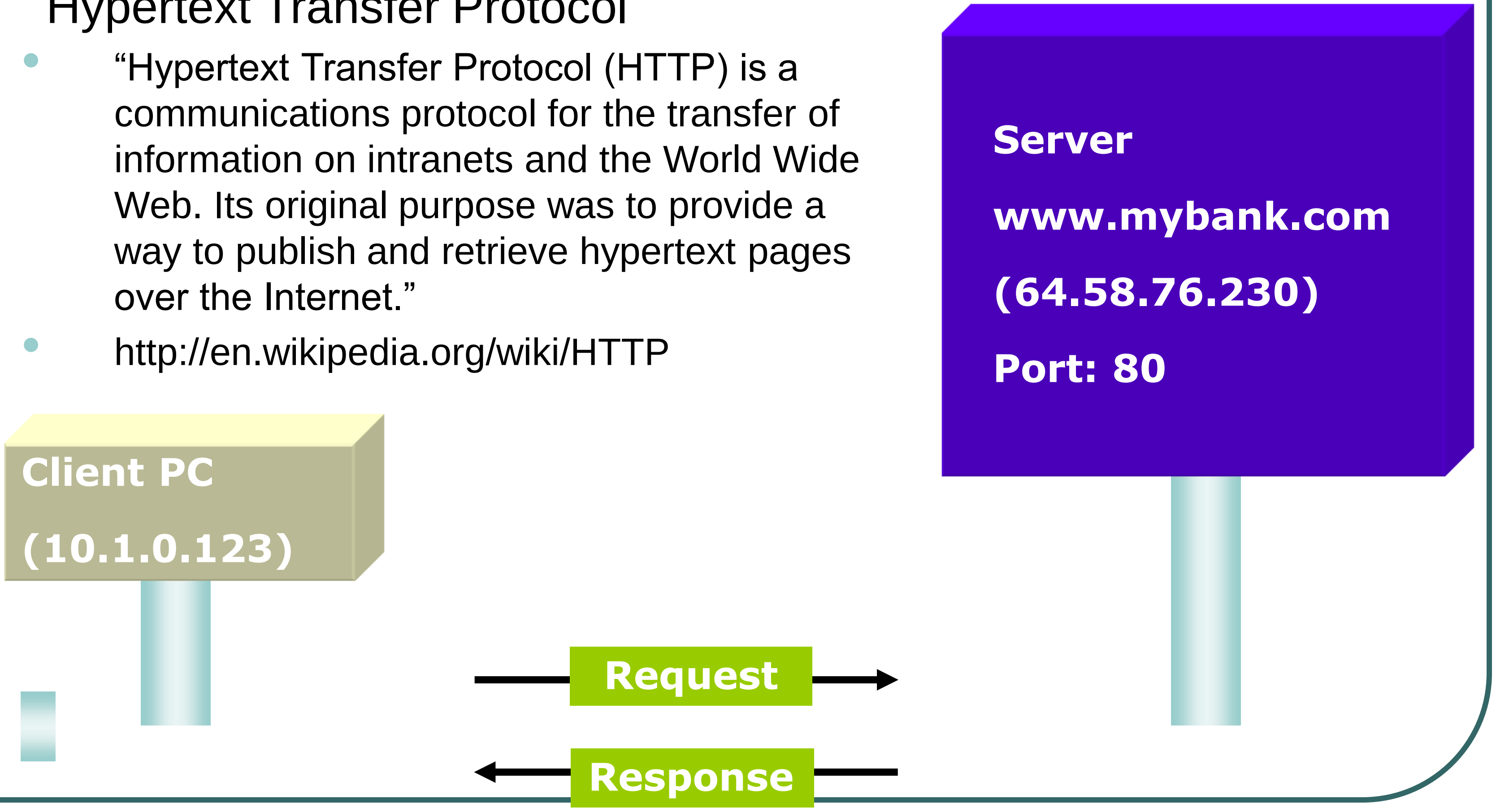
Your transfer has been sent.

Account Summary | Bill Pay | Transfer | Customer Service | Main Menu
Home | Contact Us |

From:9876-543444
To:1234-987655
Amount: 100,000
Transaction#: 53796

HTTP

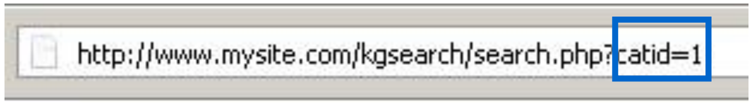
- Hypertext Transfer Protocol
  - "Hypertext Transfer Protocol (HTTP) is a communications protocol for the transfer of information on intranets and the World Wide Web. Its original purpose was to provide a way to publish and retrieve hypertext pages over the Internet."
  - http://en.wikipedia.org/wiki/HTTP



HTTP Request - GET

- Form data encoded in the URL
- Most common HTTP method used on the web
- Should be used to retrieve information, not for actions that have side-effects

HTTP Request - GET


http://www.mysite.com/kgsearch/search.php?catid=1

http://www.mysite.com/kgsearch/search.php?**catid=1**

text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;
q=0.5

HTTP Request - GET

- http://www.google.com/search?hl=en&lr=&c2coff=1&rls=GGLG%2CGGLG%3A2005-
26%2CGGLG%3Aen&q=http%3A%2F%2Fwww.google.com%2Fsearch%3Fhl%3Den%26lr%3D%26c2cof
f%3D1%26rls%3DGGLG%252CGGLG%253A2005-
26%252CGGLG%253Aen%26q%3Dhttp%253A%252F%252Fwww.google.com%252Fsearch%253Fhl%2
53Den%2526lr%253D%2526c2coff%253D1%2526rls%253DGGLG%25252CGGLG%25253A2005-
26%25252CGGLG%25253Aen%2526q%253Dhttp%25253A%25252F%25252Fwww.google.com%25252F
search%25253Fsourceid%25253Dnavclient%252526ie%25253DUTF-
8%252526rls%25253DGGLG%25252CGGLG%25253A2005-
26%25252CGGLG%25253Aen%252526q%25253Dhttp%2525253A%2525252F%2525252Fwww%25252
52Egoogle%2525252Ecom%2525252Fsearch%2525253Fsourceid%2525253Dnavclient%25252526ie%25
25253DUTF%2525252D8%25252526rls%2525253DGGLG%2525252CGGLG%2525253A2005%2525252
D26%2525252CGGLG%2525253Aen%25252526q%2525253Dhttp%252525253A%252525252F%252525
252Fuk2%252525252Emultimap%252525252Ecom%252525252Fmap%252525252Fbrowse%252525252
Ecgi%252525253Fclient%252525253Dpublic%2525252526GridE%252525253D%252525252D0%252525
252E12640%2525252526GridN%252525253D51%252525252E50860%2525252526lon%252525253D%2
52525252D0%252525252E12640%2525252526lat%252525253D51%252525252E50860%2525252526se
arch%252525255Fresult%252525253DLondon%25252525252CGreater%25252525252520London%252525
2526db%252525253Dfreegaz%2525252526cidr%252525255Fclient%252525253Dnone%2525252526lan
g%252525253D%2525252526place%252525253DLondon%252525252CGreater%252525252BLondon%2
525252526pc%252525253D%2525252526advanced%252525253D%2525252526client%252525253Dpub
lic%2525252526addr2%252525253D%2525252526quicksearch%252525253DLondon%2525252526addr3
%252525253D%2525252526scale%252525253D100000%2525252526addr1%252525253D%2526btnG%
253DSearch%26btnG%3DSearch&btnG=Search

HTTP Requests - POST

- Data is included in the body of the request.
- Should be used for any action that has side-effects
  - Storing/updating data, ordering a product, etc…

HTTP Requests - POST

http://www.mysite.com/kgsearch/search.php

http://www.mysite.com/kgsearch/search.php

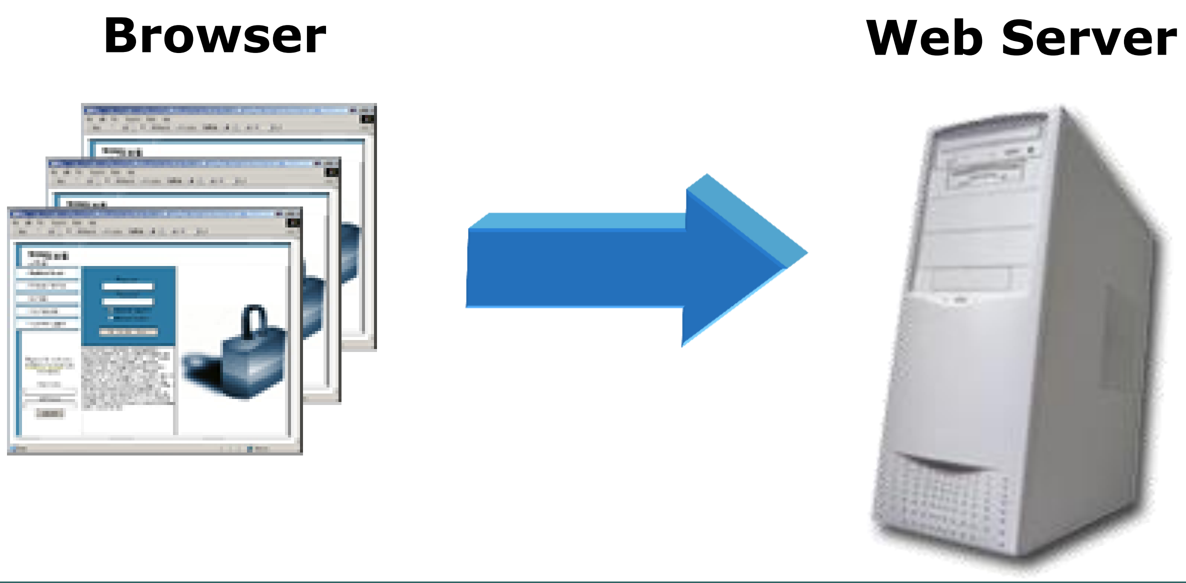
```
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
```

GET v. POST Security

- There information contained in parameters can tell a user a lot about how your application works
- GET parameters are easily visible in the address bar
- POST parameters are hidden from the *average* user
  - Users can still view source code
  - Users can still view the packets
  - Users can still intercept & modify web requests

Web Sites

- No applications
- Static pages
- Hard coded links

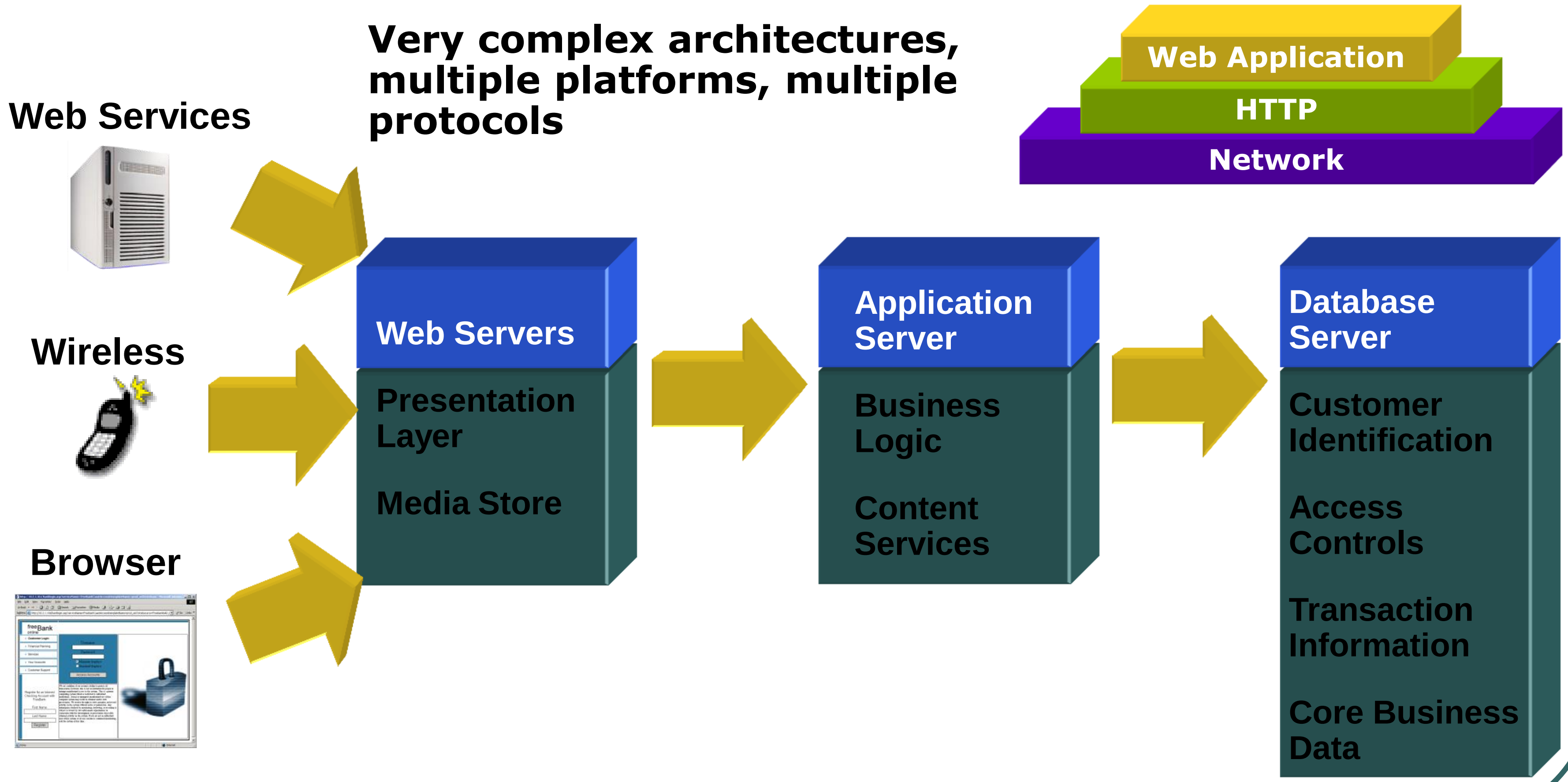


The Missing Piece

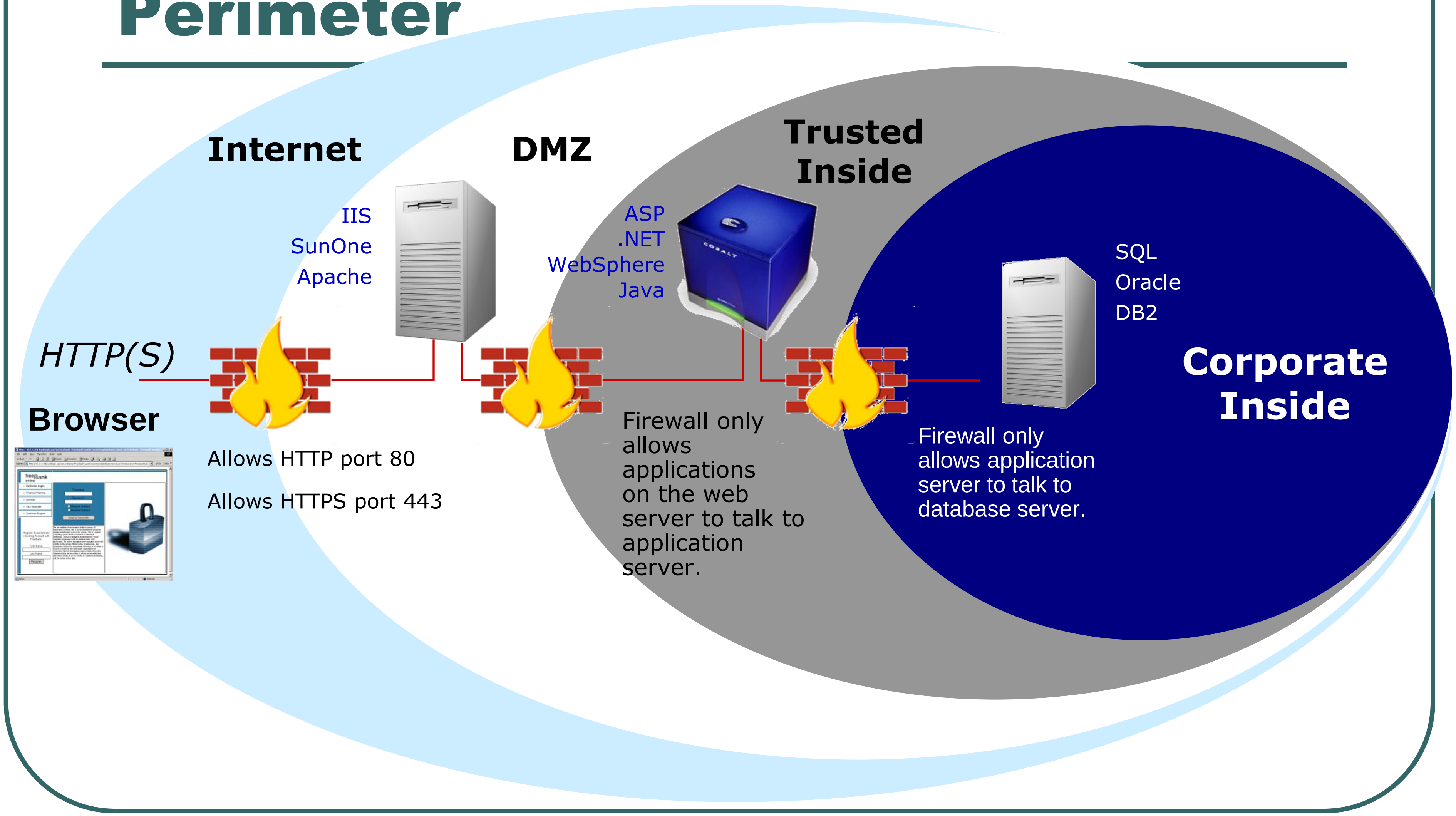
- Protection for the application itself
  - Applications are vulnerable
  - Developers lack tools and know how to build secure applications
  - No amount of QA testing will capture all the security vulnerabilities
  - Systematic failures in the application can be engineered by hackers

Web Applications

Very complex architectures, multiple platforms, multiple protocols



Web Applications Breach the Perimeter


Internet
DMZ
Trusted Inside
Corporate Inside
IIS
SunOne
Apache
ASP
.NET
WebSphere
Java
SQL
Oracle
DB2
HTTP(S)
Browser
Allows HTTP port 80
Allows HTTPS port 443
Firewall only allows applications on the web server to talk to application server.
Firewall only allows application server to talk to database server.

Why Web Application Vulnerabilities Occur

The Web Application Security Gap

Security Professionals Don’t Know The Applications

“As a Network Security Professional, I don’t know how my companies 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 Developers and QA 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 develop my web application with security as a feature.”

Web Application Vulnerabilities

“If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization.”

-Weinberg's Second Law

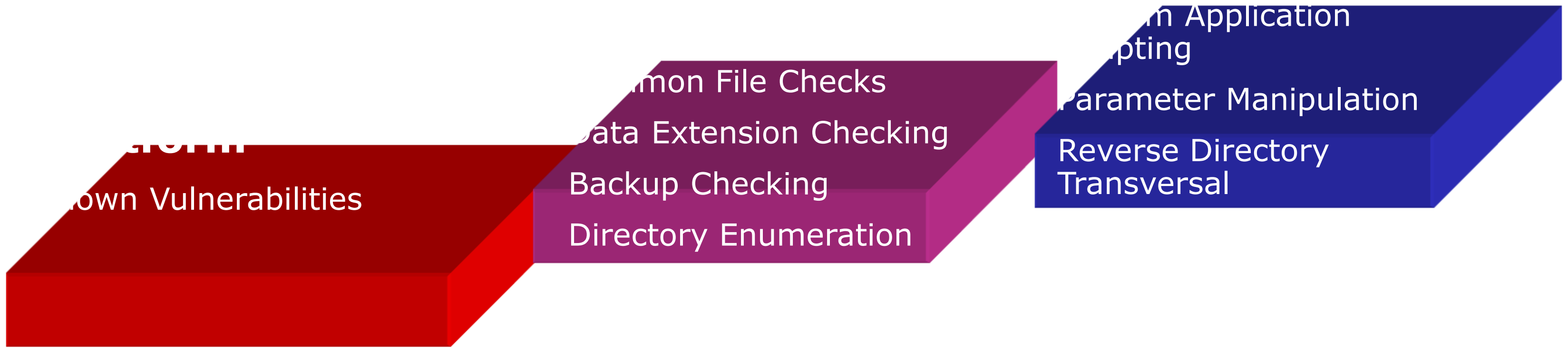
Web Application Vulnerabilities

- Technical (Syntactical) Vulnerabilities
  - Result of insecure programming techniques
  - Mitigation requires code changes
  - Detectable by scanners
  - http://example/order.asp?item=<script>alert(‘p0wned’)</script>&price=300.00

- Logical Vulnerabilities
  - Result of insecure program logic
  - Most often to due to poor decisions regarding trust
  - Mitigation often requires design/architecture changes
  - Detection often requires humans to understand the context
  - http://example/order.asp?item=toaster&price=30.00

Web Application Vulnerabilities

Web application vulnerabilities occur in multiple areas.

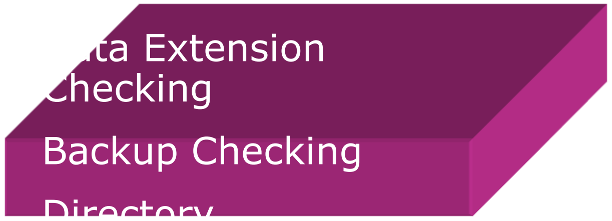


Web Application Vulnerabilities

Platform:

- Known vulnerabilities can be exploited immediately with a minimum amount of skill or experience – “script kiddies”
- Most easily defendable of all web vulnerabilities
- MUST have streamlined patching procedures

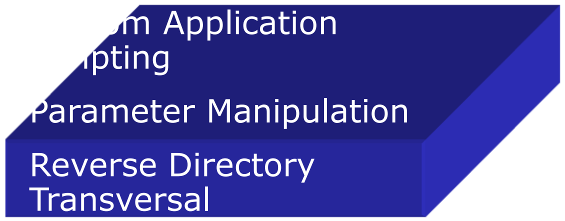
Web Application Vulnerabilities



Administration:

- Less easily corrected than known issues
- Require increased awareness
- More than just configuration, must be aware of security flaws in actual content
- Remnant files can reveal applications and versions in use
- Backup files can reveal source code and database connection strings

Web Application Vulnerabilities



Application Programming:

- Common coding techniques do not necessarily include security
- Input is assumed to be valid, but not tested
- Unexamined input from a browser can inject scripts into page for replay against later visitors
- Unhandled error messages reveal application and database structures
- Unchecked database calls can be ‘piggybacked’ with a hacker’s own database call, giving direct access to business data through a web browser

Web Application Hacking - Results

Type A
Stealing Company Assets
Type B
Falsifying B2B or B2C Buy/Sell Transactions
Type C
Obtaining Customer Information
Type D
Defacing a site

Auditing & Testing

The process

- Coverage of relevant business process
- Full inspection of client side scripts and comments
- Full inspection of application interfaces
- Analysis of potential vulnerabilities
- Testing of potential vulnerabilities
- Check for installation of known patches

The knowledge

- Complete understanding of the application logic
- Complete knowledge of application manipulation methods
- Awareness of all the known patches issues
- Complete understanding of most secure configuration of all tools

Auditing & Testing – The Problem

Multiple points of people failure

- Development, QA, Operations, Vendor software, Outsourcing

New third party bugs discovered every day

- site exposed during patch latency

Site Complexity

- many line of codes and application interactions

Compressed application development cycle

- time to market needs will impact development and QA

Distributed Knowledge

- Any single person does not have all the knowledge needed for a full audit.

What is a Viable Solution?

VIABLE = *Positive Security Model*:

- **Assessment**: bullet-proof applications before production
- **Application Firewalls**: block, log and alert against known/unknown attacks
- **Behavioral/ Policy-based**
  - Automatically builds a policy in real time for the site
  - Allows only intended business interactions
  - Maintains intended application behavior
- e.g., Code Red and Nimda blocked without updates or rules

Not Viable = *Negative Security Model*:

Signature/Rules-based – Blocks known attacks based on signatures, heuristics or rules

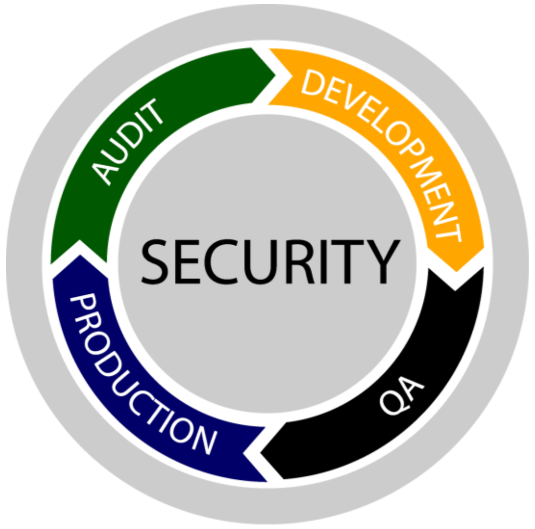
e.g., - need patch installed or signatures written to block Code Red & Nimda

How to Secure Web Applications

- Incorporate security into the lifecycle
  - Apply information security principles to all software development efforts
- Educate
  - Issue awareness, Training, etc…

How to Secure Web Applications

- Incorporating security into lifecycle
  - Integrate security into application requirements
  - Including information security professionals in software architecture/design review
  - Security APIs & libraries (e.g. ESAPI, Validator, etc.) when possible
  - Threat modeling
  - Web application vulnerability assessment tools



How to Secure Web Applications

Educate

- Developers – Software security best practices
- Testers – Methods for identifying vulnerabilities
- Security Professionals – Software development, Software coding best practices
- Executives, System Owners, etc. – Understanding the risk and why they should be concerned

Bespoke Applications Vs. Commercial Applications

Application Development internal use:

- Bespoke, customized, one-off application
  - Audience is not so great: (Users, developers, test)
    - ➢Vulnerabilities are not discovered too quickly by users.
    - ➢Vulnerabilities are discovered by hackers, they actively look for them.

Bespoke application = Small audience = Less chance of vulnerabilities being discovered
This is unlike, Say Microsoft XP 210 Million copies sold (http://www.forbes.com/ May2004)

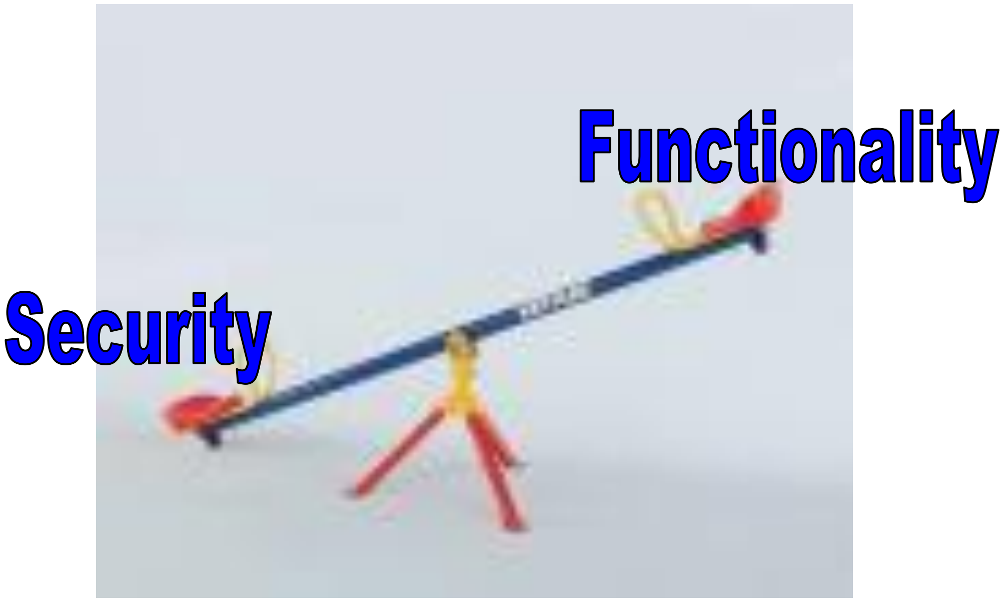
First Line of Defense:

The Developer:

- Writes the code.
- Understands the problem better than anyone!
- Has the skill set.
- More effective and efficient in providing a solution

Complexity Vs Security

As Functionality and
hence complexity increase
security decreases.



Integrating security into
functionality at design time
Is easier and cheaper.

"100 Times More Expensive to Fix Security Bug at Production Than Design"
– IBM Systems Sciences Institute

It also costs less in the long-term.
-maintenance cost

A Few Facts and figures (contd)

- Interesting Statistics – *Employing code review*
  - IBM Reduces 82% of Defects Before Testing Starts
  - HP Found 80% of Defects Found Were Not Likely To Be Caught in Testing
  - 100 Times More Expensive to Fix Security Bug at Production Than Design"
    – IBM Systems Sciences Institute
- Promoting People Looking at Code
  - Improvement Earlier in SDLC
  - Fix at Right Place; the Source
  - Takes 20% extra time – payoff is order of magnitude more.

Ref: http://ganssle.com/Inspections.pdf

If cars Were Built Like Applications….

1. 70% of all cars would be built without following the original designs and blueprints. The other 30% would not have designs.
2. Car design would assume that safety is a function of road design and that all drivers were considerate, sober and expert drivers.
3. Cars would have no airbags, mirrors, seat belts, doors, roll-bars, side-impact bars, or locks, because no-one had asked for them. But they *would* all have at least six cup holders.
4. Not all the components would be bolted together securely and many of them would not be built to tolerate even the slightest abuse.
5. Safety tests would assume frontal impact only. Cars would not be roll tested, or tested for stability in emergency maneuvers, brake effectiveness, side impact and resistance to theft.
6. Many safety features originally included might be removed before the car was completed, because they might adversely impact performance.
7. 70% of all cars would be subject to monthly recalls to add major components left out of the initial production. The other 30% wouldn't be recalled, because no-one would sue anyway.
8. The after-market for safety devices would include such useful products as training wheels, screen doors, elastic seatbelts and devices that would restrict the car's top speed to 3mph, if found to be unsafe (which would be always).
9. Useful safety could be found, but could only be custom retro-fitted, would take six months to fit and would cost more than the car itself.
10. A NCT/MOT inspection would consist of counting the wheels and making recommendations on wheel quantity.
11. Your only warning indicator would be large quantities of smoke and flame in the cab.
12. You could only get insurance from one provider, it would be extremely expensive, require a duplicate NCT/MOT inspection, and you might still never be able to claim against the policy.

- Denis Verdon

How do we do it?

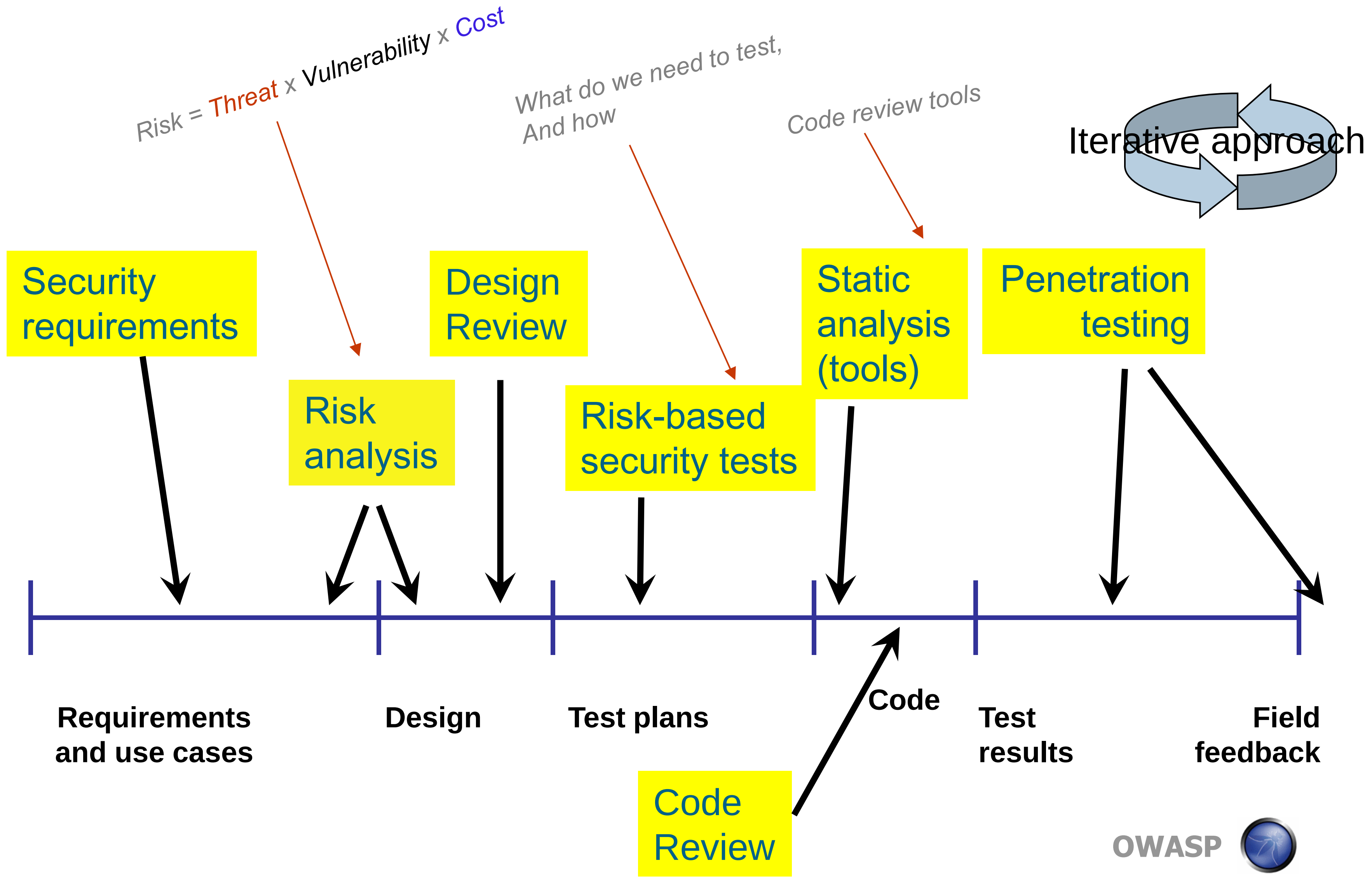
- Security Analyst:
  - Get involved early in SDLC. Security is a function of the asset we want to secure, what's it worth?
  - Understanding the information held in the application and the types of users is half the battle.
  - Involve an analyst in the design phase and thereafter.
- Developer:
  - Embrace secure application development. (Educate)
  - Quality is not just “Does it work” Security is a measure of quality also.

How do we do it? (contd)

- QA:
  - Security vulnerabilities are to be considered bugs, the same way as a functional bug, and tracked in the same manner.

- Managers:
  - Factor some time into the project plan for security.
  - Consider security as added value in an application.
    - $1 spent up front saves $10 during development and $100 after release

Software security tollgates in the SDLC



OWASP

Application Security Risk Categorization

- Goal
  - More security for riskier applications
  - Ensures that you work the most critical issues first
  - Scales to hundreds or thousands of applications

- Tools and Methodology
  - Security profiling tools can gather facts
    - Size, complexity, security mechanisms, dangerous calls
  - Questionnaire to gather risk information
    - Asset value, available functions, users, environment, threats
  - Risk-based approach
    - Evaluates likelihood and consequences of successful attack

Application Security Project Plan

- Define the plan to ensure security at the end
  - Ideally done at start of project
  - Can also be started before or after development is complete
- Based on the risk category
  - Identify activities at each phase
  - Necessary people and expertise required
  - Who has responsibility for risks
  - Ensure time and budget for security activities
  - Establish framework for establishing the “line of sight”

Application Security Requirements Tailoring

- Get the security requirements and policy right

- Start with a generic set of security requirements
  - Must include all security mechanisms
  - Must address all common vulnerabilities
  - Can be use (or misuse) cases
  - Should address all driving requirements (regulation, standards, best practices, etc.)

- Tailoring examples…
  - Specify how authentication will work
  - Detail the access control matrix (roles, assets, functions, permissions)
  - Define the input validation rules
  - Choose an error handling and logging approach

Design Reviews

- Better to find flaws early
- Security design reviews
  - Check to ensure design meets requirements
  - Also check to make sure you didn’t miss a requirement
- Assemble a team
  - Experts in the technology
  - Security-minded team members
  - Do a high-level penetration test against the design
  - Be sure to do root cause analysis on any flaws identified

Software Vulnerability Analysis

- Find flaws in the code early
- Many different techniques
  - Static (against source or compiled code)
    - Security focused static analysis tools
    - Peer review process
    - Formal security code review
  - Dynamic (against running code)
    - Scanning
    - Penetration testing
- Goal
  - Ensure completeness (across all vulnerability areas)
  - Ensure accuracy (minimize false alarms)

Application Security Testing

- Identify security flaws during testing
- Develop security test cases
  - Based on requirements
  - Be sure to include “negative” tests
  - Test all security mechanisms and common vulnerabilities
- Flaws feed into defect tracking and root cause analysis

Application Security Defect Tracking and Metrics

- "Every security flaw is a process problem"
- Tracking security defects
  - Find the source of the problem
    - Bad or missed requirement, design flaw, poor implementation, etc…
  - ISSUE: can you track security defects the same way as other defects
- Metrics
  - What lifecycle stage are most flaws originating in?
  - What security mechanisms are we having trouble implementing?
  - What security vulnerabilities are we having trouble avoiding?

Configuration Management and Deployment

- Ensure the application configuration is secure
- Security is increasingly "data-driven"
  - XML files, property files, scripts, databases, directories
- How do you control and audit this data?
  - Design configuration data for audit
  - Put all configuration data in CM
  - Audit configuration data regularly
  - Don't allow configuration changes in the field

What now?

"So now, when we face a choice between adding features and resolving security issues, we need to choose security."

-Bill Gates

If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology.

-Bruce Schneier

Using encryption on the Internet is the equivalent of arranging an armored car to deliver credit-card information from someone living in a cardboard box to someone living on a park bench.

-Gene Spafford

REFERENCES

Carnegie Mellon University
Software Engineering Institute

Lecture 3a

The Survivable Network Analysis Method: Evaluating Survivability of Critical Systems

Web Application Security

Presented by:
Colin English
Zerflow

Web Application Security 101

Steve Carter

(special thanks to SPI Dynamics)

Integration into the SDLC
(Software Development Life Cycle)

With *Eoin Keary*
Eoin.keary@owasp.org

OWASP

The OWASP Foundation
http://www.owasp.org