



build | integrate | secure

# Introduction to Web Application Security

Microsoft CSO Roundtable – Houston, TX

September 13<sup>th</sup>, 2006

# Overview

- Background
- What is Application Security and Why Is It Important?
- Examples
- Where Do We Go From Here?
  - *Assess Critical Infrastructure*
  - *Integrate Security Into the Software Development Lifecycle*
- Questions

# Background

- Denim Group
  - *San Antonio-based consultancy*
  - *Custom software development*
  - *Systems integration*
  - ***Application security***
- Management Team Experience
  - *Large-scale software development*
  - *Air Force information warfare*
  - *Client service for DoD, Big 4, Fortune 500*
- Presenter: Dan Cornell
  - *Software developer (MCSD, Java 2 Certified Programmer)*

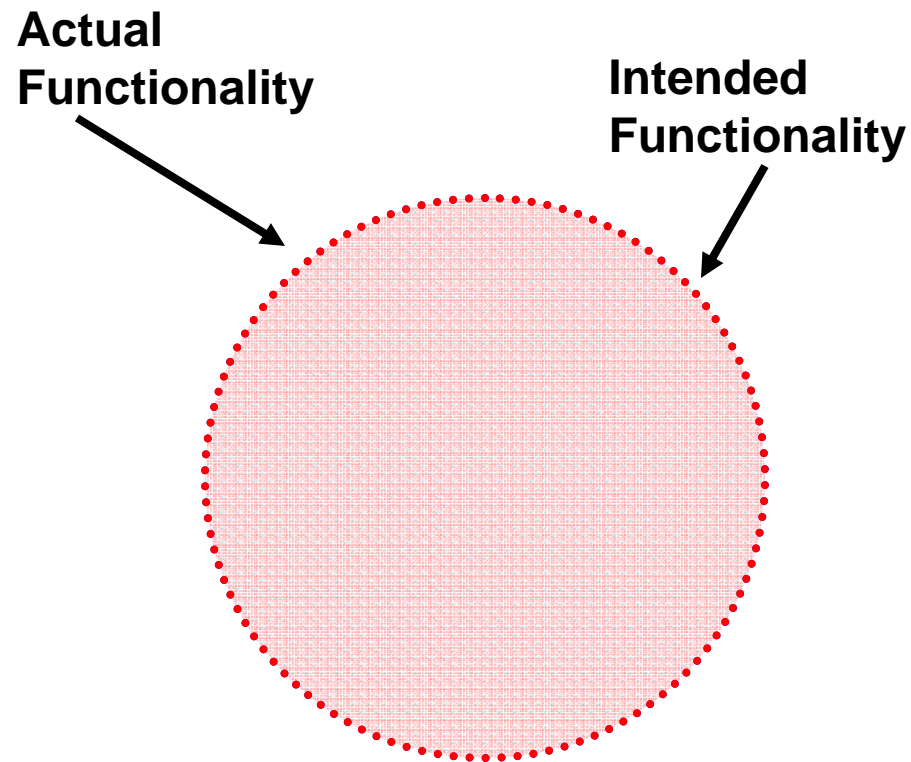
# What is Application Security and Why is it Important?

- Application Security Defined
- Fit with General Information Security Landscape
- Why Does Application Security Matter

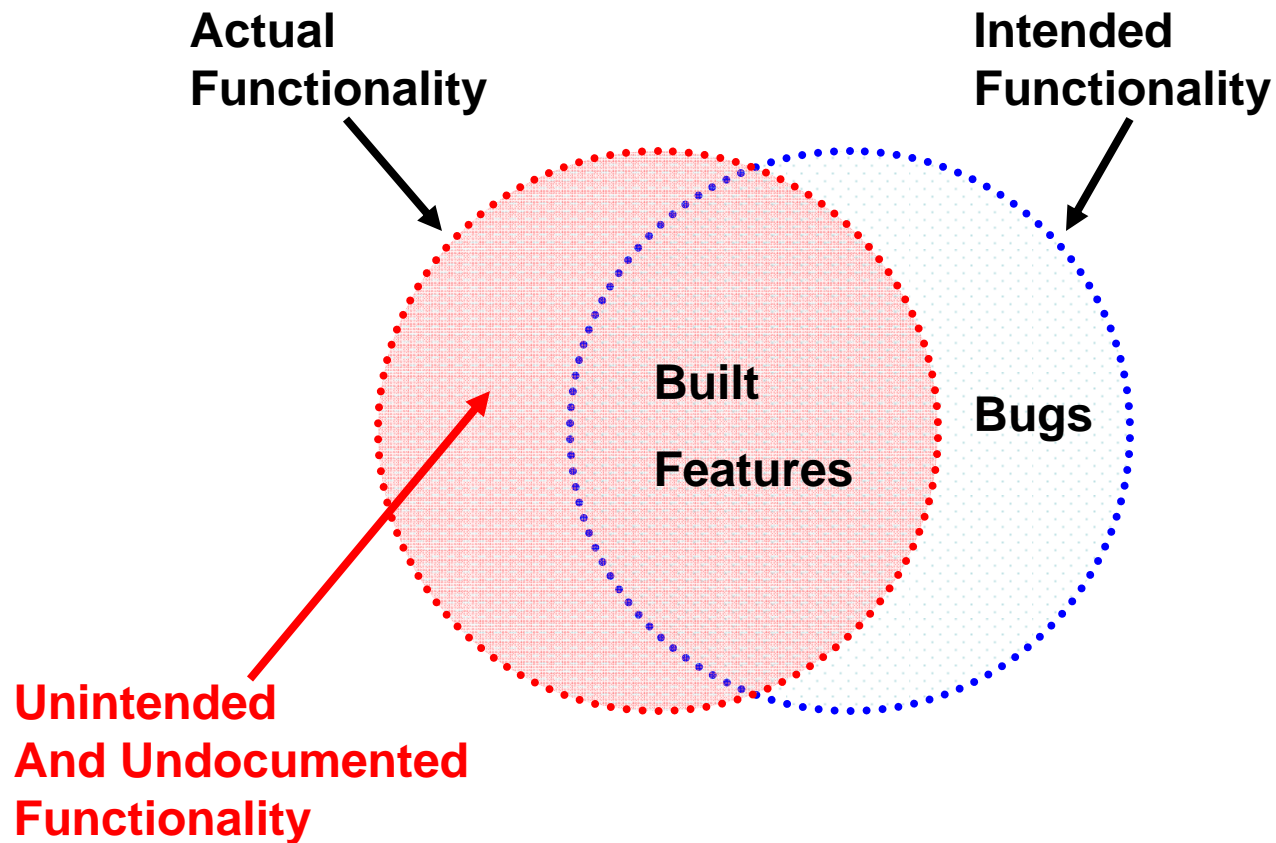
## Application Security Defined

- Ensuring that custom application code performs as expected under the entire range of possible inputs
- Goals:
  - *Confidentiality*
  - *Integrity*
  - *Availability*
- Relationship to Software Quality Assurance
  - *Really a sub-area of SQA*
  - *SQA typically verifies that software does what it is supposed to do*
  - *Application security is concerned that software does not do what it should not do*

## Software Implementation – Perfect World



## Software Implementation – Real World



## Infrastructure Security

- Software Development: Features, functions and timelines
- Traditional Information Security: Audit, measure and maintain
- Application security applies information security principles to custom software development efforts
- Many traditional information security practitioners are ill-equipped to mitigate application security issues
  - *Little to no experience coding*
  - *No experience coding in “modern” enterprise environments like .NET and J2EE*
  - *Understand that there are risks, but not in a position to address them*

## Why Does This Matter?

- Business-critical web applications are Internet-facing
  - *An increasing number of assets are exposed*
- Most applications have serious flaws
  - *Foundstone and @Stake studies*
- The regulatory environment has changed
  - *Sarbanes Oxley*
  - *GLB*
  - *California SB-1386*
  - *PCI*

# Types of Vulnerabilities

- Technical Vulnerabilities
  - *Surface due to insecure programming techniques*
  - *Typically due to poor input handling and input validation*
  - *Most “scanner” tools primarily find technical vulnerabilities*
  - *Remediation: coding changes*
- Logical Vulnerabilities
  - *Surface due to insecure program logic*
  - *Typically due to poor decisions about trust*
  - *Most “scanner” tools are powerless to find logical vulnerabilities*
  - ***Most “scanner” tools are powerless to find logical vulnerabilities***
  - *Remediation: architecture and design changes*

## OWASP Top 10 Critical Web Application Security Vulnerabilities

- Unvalidated Input
- Broken Access Control
- Broken Authentication and Session Management
- Cross Site Scripting (XSS)
- Buffer Overflows
- Injection Flaws
- Improper Error Handling
- Insecure Storage
- Application Denial of Service
- Insecure Configuration Management

(source: [http://www.owasp.org/index.php/OWASP\\_Top\\_Ten\\_Project](http://www.owasp.org/index.php/OWASP_Top_Ten_Project))

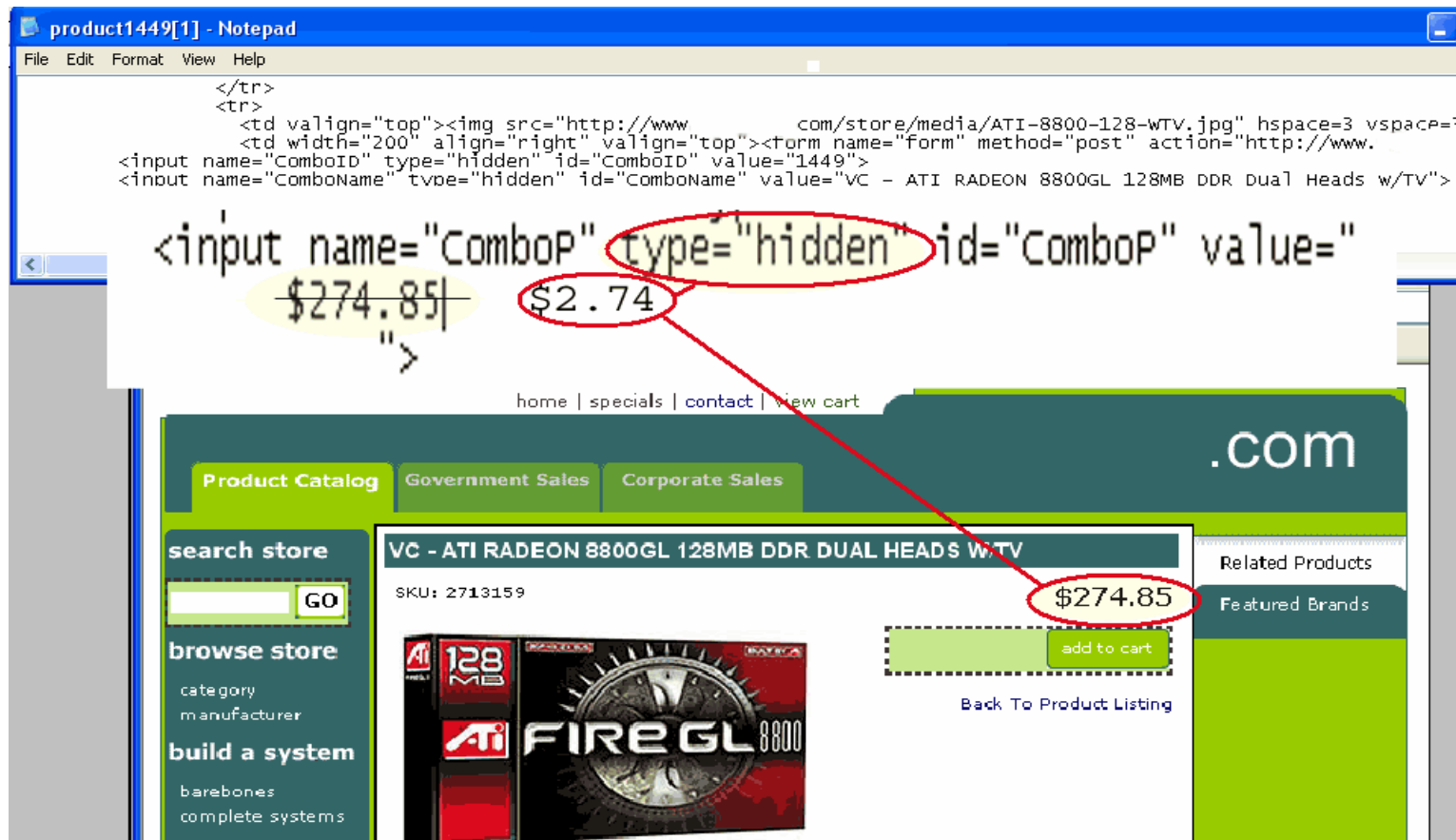
## Examples

- Hidden parameter tampering
- Cookie manipulation
- SQL injection

## Hidden Parameter Tampering

- Price information is stored in hidden HTML form field
- Assumption: hidden field won't be edited
- Attacker edits price parameter
- Attacker submits altered web page with new "price"
- Application trusts the price parameter from the user
- Still widespread in many web stores

# The Attack



The image illustrates a price manipulation attack on an e-commerce website. It consists of two parts:

**Top Part: Notepad Window**  
The Notepad window shows the HTML code for a product page. The code includes a form with hidden input fields for 'ComboID' and 'ComboName'. A new hidden input field, `<input name="Combop" type="hidden" id="Combop" value="`, is being injected into the code. The value of this field is `$274.85|`, which is highlighted in yellow. The original price `$2.74` is also highlighted in yellow and circled in red. A red arrow points from this circled price to the price on the product page below.

**Bottom Part: Product Page**  
The product page shows the product 'VC - ATI RADEON 8800GL 128MB DDR DUAL HEADS W/TV' with SKU 2713159. The price is displayed as **\$274.85**, which is circled in red. The 'add to cart' button is visible, and the price is significantly higher than the original price shown in the code above.

# The Result

home | specials | contact | view cart

Product Catalog Government Sales Corporate Sales

search store  GO

browse store  
category  
manufacturer

build a system  
barebones  
complete systems

customer care  
technical support  
returns  
order tracking  
open forum  
terms & conditions  
privacy pledge

FOLLOWING UPGRADES ARE IMPORTANT FOR YOUR VC - ATI RADEON 8800GL 128MB DDR DUAL HEADS W/TV

Price: \$2.74  
Price (with Selected Options): \$21.12

**Price: \$2.74**

**Thermal Management**  
Improve Heat Management . For Longer life and to get better Stability.  
Provide yourself with some peace of mind.

- Do not need recommended Heatsink and Fan Solutions
- thermaltake crystal orb for vga card cooling [+ \$15.95]
- thermaltake geforce 4 highest performance cooler [+ \$22.86]
- thermaltake g4-vga coolmod highest performance cooler [+ \$38.82]

Related Products

ATI RADEON 9800PRO 256MB

Graphics Controller:  
Radeon 9800 Pro  
Memory:  
256MB DDR  
W/TV-out & DVI  
Dual Head

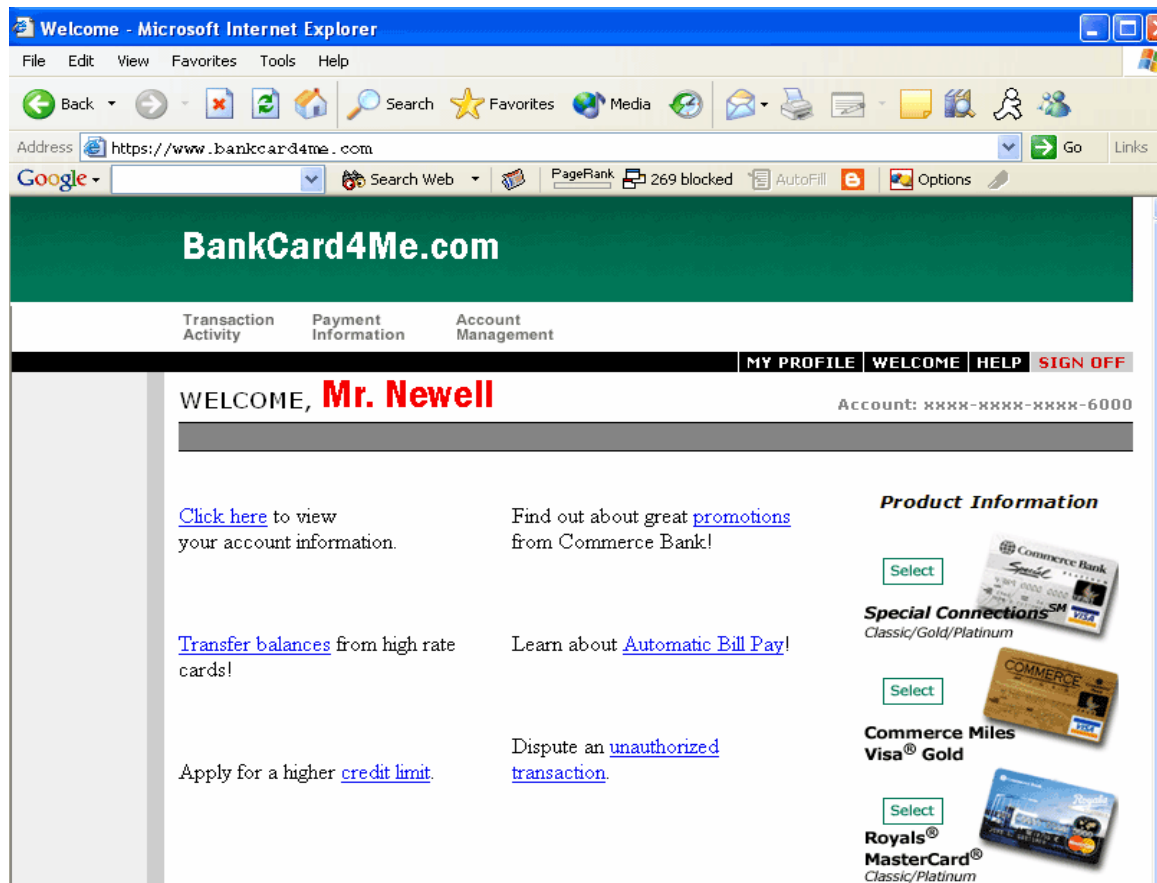
\$484.02 [ info ]

Samsung CD-RW

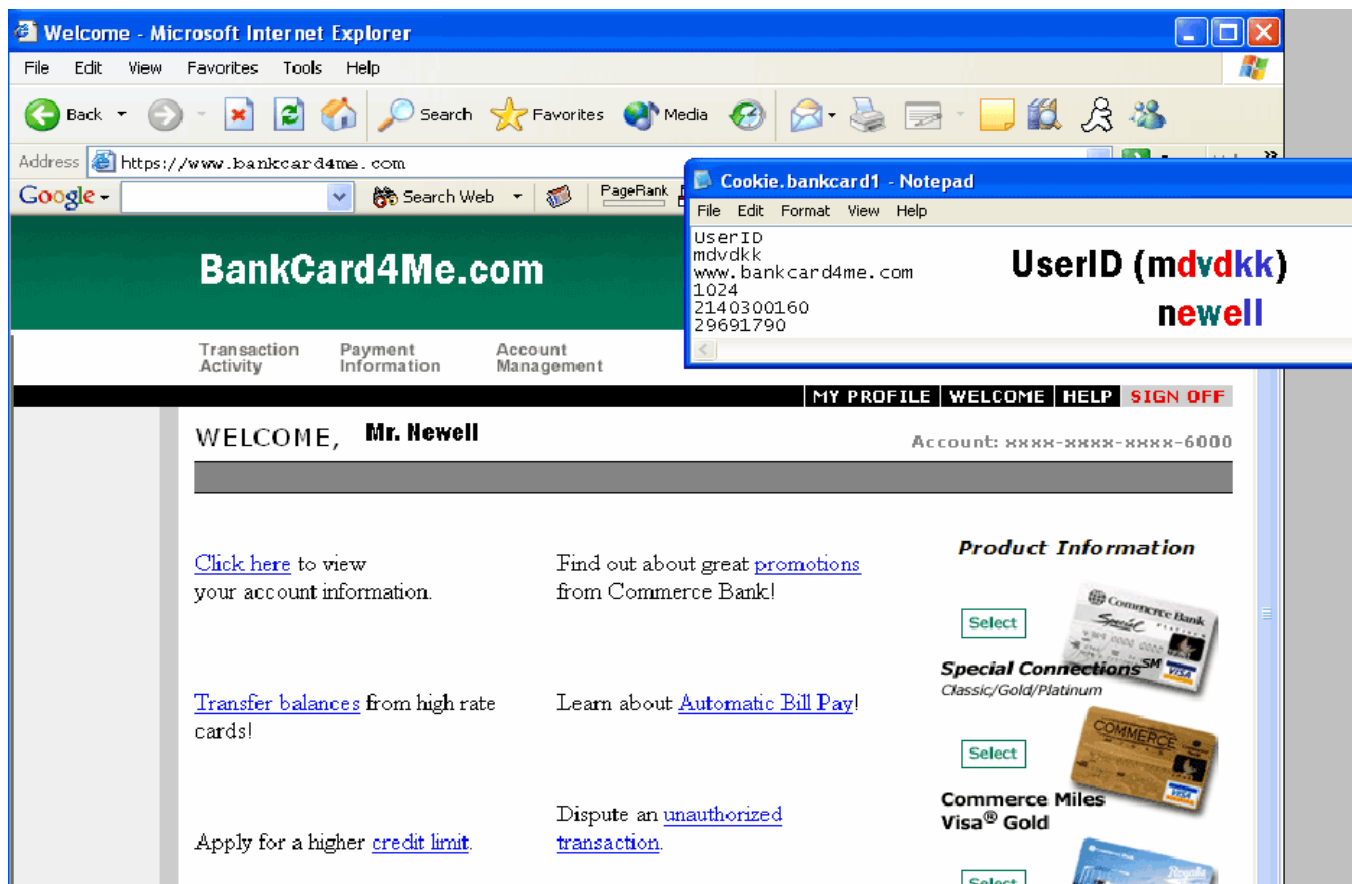
## Cookie Manipulation

- Browser cookie is used to store user identity information
- Assumption: cookies are set by server side code, handled by the browser automatically and not manipulated by users
- Attacker alters cookie
- Application trusts the browser cookie and allows attacker to assume identity of another user

# The Attack



# The Attack



Cookie.bankcard1 - Notepad

```
File Edit Format View Help
UserID
mdvdkk
www.bankcard4me.com
1024
2140300160
29691790
```

**UserID (mdvdkk)**  
**newell**

BankCard4Me.com

Transaction Activity Payment Information Account Management

MY PROFILE WELCOME HELP SIGN OFF

WELCOME, **Mr. Newell** Account: xxxx-xxxx-xxxx-6000

[Click here](#) to view your account information.

Find out about great [promotions](#) from Commerce Bank!


[Transfer balances](#) from high rate cards!

Learn about [Automatic Bill Pay!](#)


Apply for a higher [credit limit](#).

Dispute an [unauthorized transaction](#).


**Product Information**

Select 

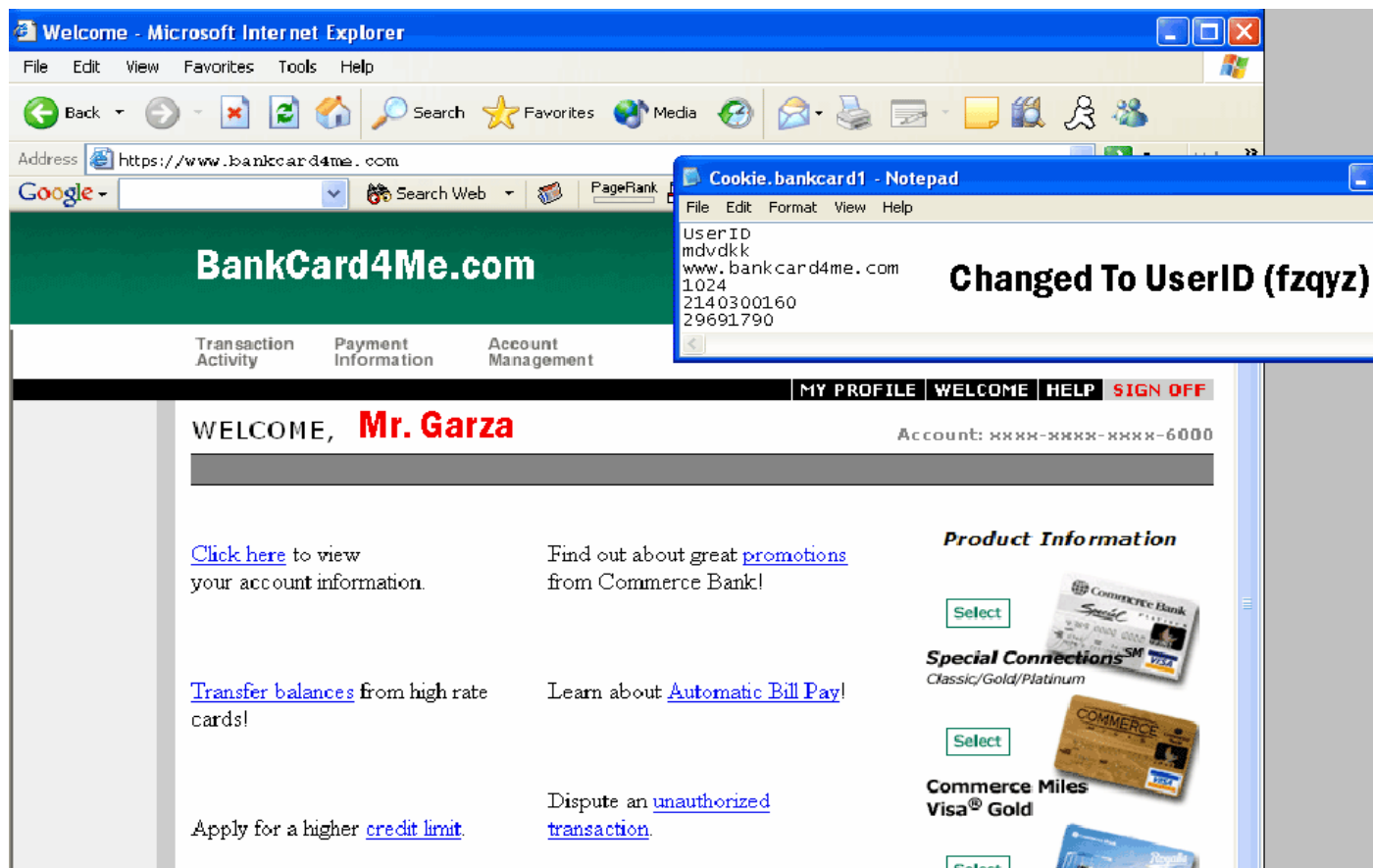
**Special Connections** SM  
Classic/Gold/Platinum

Select 

**Commerce Miles Visa® Gold**

Select 

# The Result



Welcome - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address <https://www.bankcard4me.com>

Google Search Web PageRank

**BankCard4Me.com**

Transaction Activity Payment Information Account Management

MY PROFILE WELCOME HELP SIGN OFF

WELCOME, **Mr. Garza** Account: xxxx-xxxx-xxxx-6000

[Click here](#) to view your account information.

[Transfer balances](#) from high rate cards!


Apply for a higher [credit limit](#).

Find out about great [promotions](#) from Commerce Bank!


Learn about [Automatic Bill Pay!](#)

Dispute an [unauthorized transaction](#).


**Product Information**

Select 

**Special Connections** Classic/Gold/Platinum

Select 

**Commerce Miles Visa® Gold**

Select 

## SQL Injection

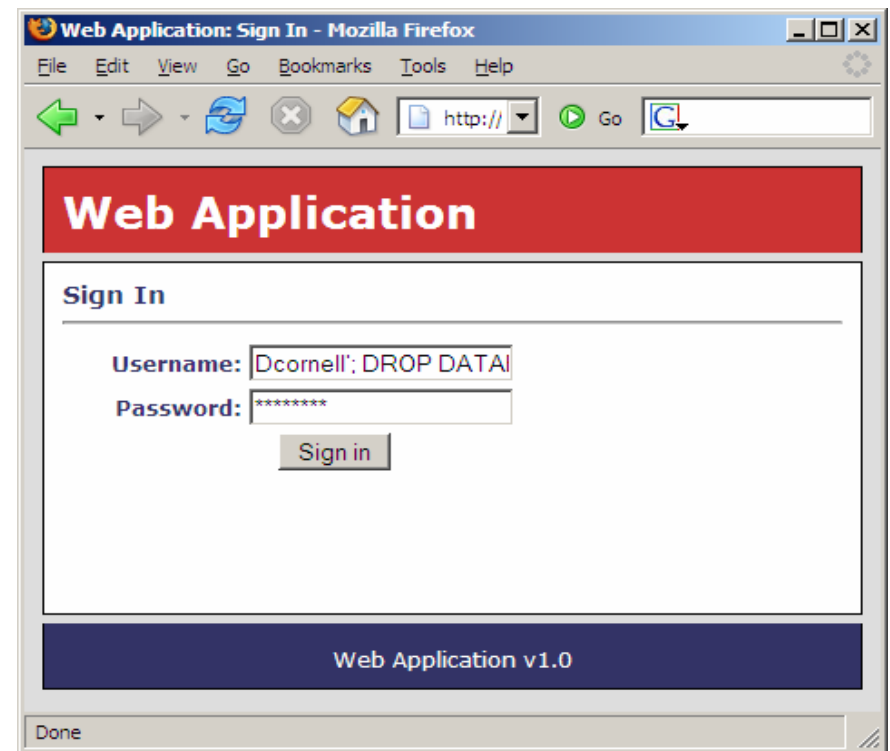
- SQL statements are created from a combination of static text and user inputs
- Assumption: users will enter well-formed inputs
- Attacker crafts a custom input to hijack control of the SQL interpreter and execute arbitrary code
- Very common flaw with tremendous security implications

## The Attack

```
try {
    string username = request.getParameter("username");
    string password = request.getParameter("password");
    string sSql = "SELECT * FROM User WHERE username =
    \' + username + \' AND password = \' + password +
    \'";
    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery(sSql);
    ...
} catch(Exception ex) {}
```

## The Attack

- Specially crafted input contains SQL control characters



## The Attack

- Malicious user sends in a username parameter of: Dcornell'; DROP DATABASE Ecommerce; --

SQL Executed is:

```
SELECT * FROM User WHERE username = 'Dcornell'; DROP
  DATABASE Ecommerce; -- AND password = 'whocares'
```

- Attacker can execute arbitrary database queries with the same permissions as the application
  - *View sensitive data (Confidentiality breach)*
  - *Modify data (Integrity breach)*
  - *Destroy data (Integrity, Availability breach)*

## Where Do We Go From Here

- Assess critical infrastructure applications
- Integrate security into the software development lifecycle
  - *If you want to reliably produce secure applications, you must create a secure application development process*
  - *Running a scanner at the end of the process is NOT a security program*
    - Limited scope of vulnerabilities detected
    - Finding vulnerabilities late

## Assess Critical Infrastructure

- Automated scanning tools
  - *Help find technical vulnerabilities*
  - *Very poor (often powerless) to find design and architecture flaws*
- Assessments and penetration tests to better evaluate the security state of applications
  - *Black box*
  - *White box*

# Integrate Security Into the Lifecycle

- Train personnel on application security issues:
  - *Business analysts*
  - *Architects*
  - *Developers*
  - *Quality assurance*
- Integrate security into application requirements
- Threat modeling
  - *STRIDE: threat classification*
  - *DREAD: vulnerability severity*
- Secure coding
- Peer review
  - *Architecture*
  - *Design*
  - *Code*
- Periodic penetration testing

## Questions

Dan Cornell

[dan@denimgroup.com](mailto:dan@denimgroup.com)

John Dickson

[john@denimgroup.com](mailto:john@denimgroup.com)

Denim Group, Ltd.

Web: [www.denimgroup.com](http://www.denimgroup.com)

Blog: [denimgroup.typepad.com](http://denimgroup.typepad.com)