



Application Security and the SDLC

Dan Cornell

Denim Group, Ltd.

www.denimgroup.com



Overview

- Background
- What is Application Security and Why is It Important?
- Specific Reference Examples
- Integrating Security into a Traditional Development Process
- Observations...



Background

- Denim Group is a San Antonio based software development firm specializing in:
 - Custom software development
 - Business systems integration
 - Application-level security
 - J2EE and .NET environments
- Management team has breadth of experience working in and servicing:
 - Air Force information warfare
 - DoD software development
 - Big 4 consulting experience
 - Fortune 500 to SMB



What is Application Security?

- Ensuring that custom application code performs as expected under the entire range of possible inputs
- Software development focus
 - Traditional InfoSec focuses at the TCP/IP layer
 - Many traditional InfoSec practitioners are ill-equipped to work in the application security space



Cultural Differences

- Traditional InfoSec tends to have a “measure” culture
 - Determine what is in place
 - Audit configurations
- Application development tends to have a “build” culture
 - Create something that did not exist before
 - Project-based deadline environment
 - Certainly the case for custom software development; also largely the case for systems integration work



Why Does AppSec Matter?

- Business critical web apps are Internet-facing
- New laws and regulations govern how data is stored and made available
 - HIPAA
 - Sarbanes Oxley
 - GLB
- 70+% of applications have serious design or coding flaws
 - Studies performed by @Stake and Foundstone



Top 10 Critical Web App Vulnerabilities

- Unvalidated Parameters
- Broken Access Controls
- Broken Account and Session Management
- Cross-site Scripting Flaws
- Buffer Overflows
- Command Injection Flaws
- Error Handling Problems
- Insecure Use of Cryptography
- Remote Administration Flaws
- Web and Application Server Misconfiguration



“Hidden” HTML Manipulation

- Price information is stored in hidden HTML field with assigned \$ value
- Assumption: hidden field won't be edited
- Attacker edits \$ value of product in HTML
- Attacker submits altered web page with new “price”
- Still widespread in many web stores

Price Changes via Hidden HTML tags

product1449[1] - Notepad

```
File Edit Format View Help
</tr>
<tr>
  <td valign="top"><form name="form" method="post" action="http://www.
  <input name="ComboID" type="hidden" id="ComboID" value="1449">
  <input name="ComboName" type="hidden" id="ComboName" value="VC - ATI RADEON 8800GL 128MB DDR Dual Heads w/TV">
  <input name="ComboP" type="hidden" id="ComboP" value="
  $274.85|
  $2.74
  ">
```

home | specials | contact | view cart

Product Catalog Government Sales Corporate Sales .com

search store
GO

browse store
category
manufacturer

build a system
barebones
complete systems

VC - ATI RADEON 8800GL 128MB DDR DUAL HEADS W/TV
SKU: 2713159

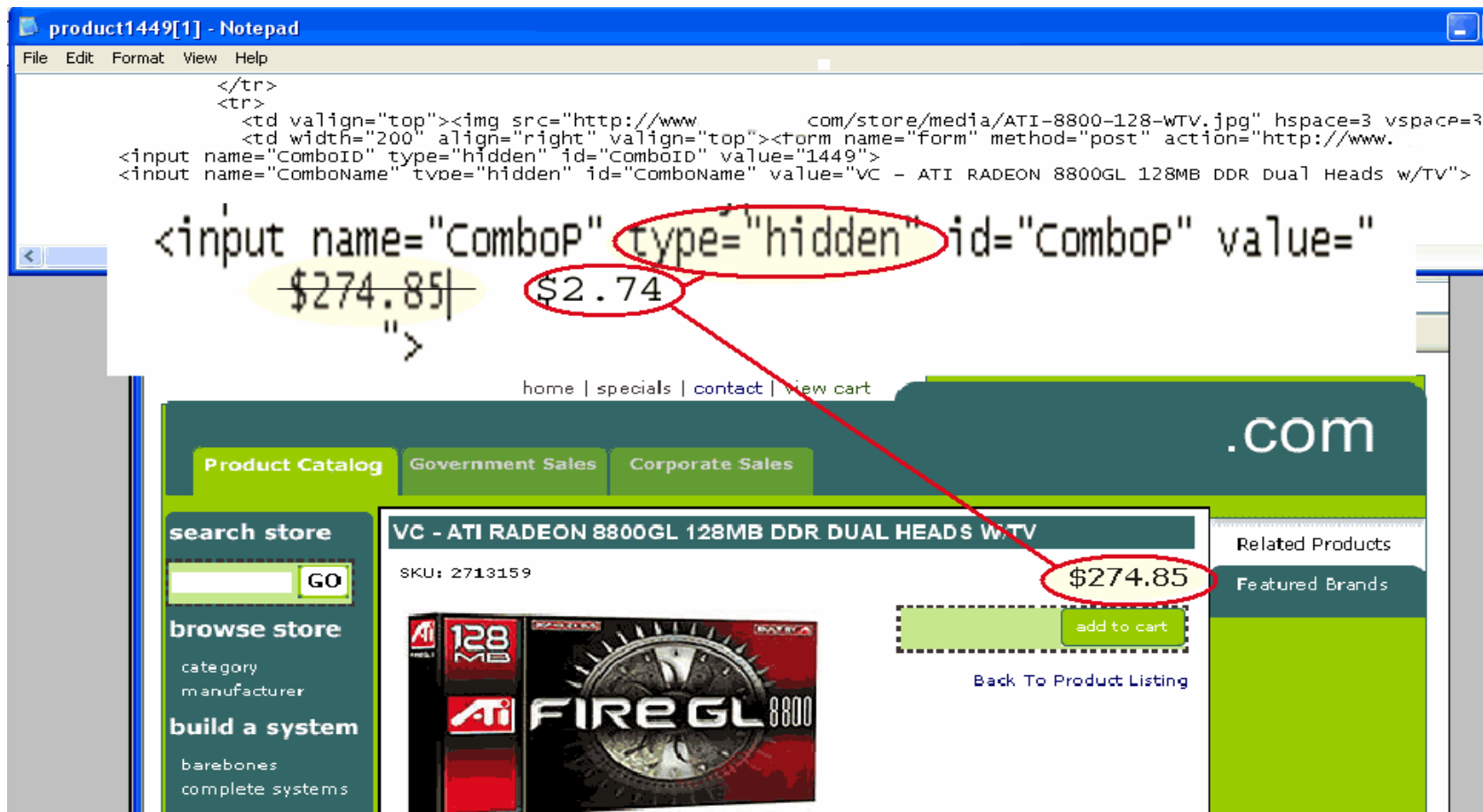
128 MB
ATI RADEON
ATI FIREGL 8800

Related Products
Featured Brands

add to cart

Back To Product Listing

\$274.85



Price Changes via Hidden HTML tags

The screenshot shows a web browser window with a Google search bar and various extensions. The browser address bar shows the URL "home | specials | contact | view cart" and ".com". The page content includes a navigation menu with "Product Catalog", "Government Sales", and "Corporate Sales". A sidebar on the left contains sections for "search store", "browse store", "build a system", and "customer care". The main content area features a product upgrade section titled "FOLLOWING UPGRADES ARE IMPORTANT FOR YOUR VC - ATI RADEON 8800GL 128MB DDR DUAL HEADS W/TV". The current price is \$2.74, and the price with selected options is \$21.12. A red circle highlights the price "\$2.74" with a red arrow pointing to it. Below the price, there is a section for "Thermal Management" with a description and a list of upgrade options, each with a radio button and a price increase. The options are: "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]", and "thermaltake g4-vga coolmod highest performance cooler [+\$38.82]". The "thermaltake g4-vga coolmod highest performance cooler" option is selected. To the right of the main content, there is a "Related Products" section featuring an "ATI RADEON 9800PRO 256MB" graphics card for \$484.02 and a "Samsung CD-RW" drive.

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
open forum
terms & conditions

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]

thermaltake g4-vga coolmod highest performance cooler [+\$38.82]

Related Products

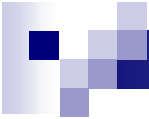
ATI RADEON 9800PRO 256MB

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

\$484.02 [info]

Samsung CD-RW

Samsung CD-RW



An Example: Insecure Design/Code

```
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) {}
```



Possible Exploits

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

SQL Executed is:

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

- Cracker breaks into database and has cleartext access to usernames and passwords, which may be reused on other sites
- Malicious user finds a way to cause an error condition and exploits the unexpected behavior in some manner



An Example: More Secure Design/Code

```
try {
    string username = request.getParameter("username");
    string password = request.getParameter("password");
    string passwordHash = MD5.hash(password);
    PreparedStatement stmt = con.prepareStatement("SELECT *
    FROM User WHERE username = ? AND passwordHash = ?");
    stmt.setString(1, username);
    stmt.setString(2, passwordHash);
    ResultSet rs = stmt.executeQuery();

    ...
} catch(SQLException sqlEx) {
    // Actually handle error condition...
}
```

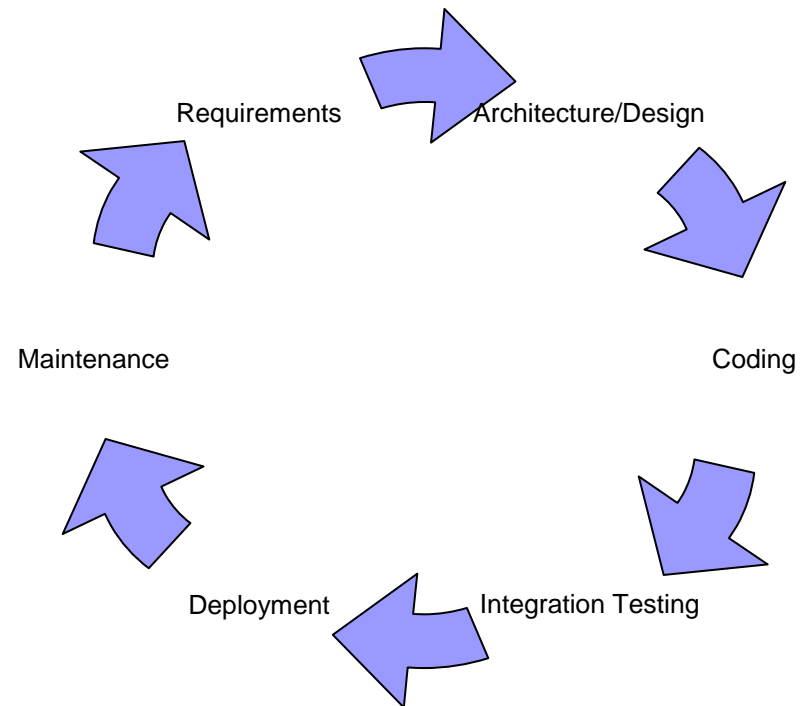


What to Do?

- To build better/more secure software, build a better software development process
- Security bugs are 100x more expensive to fix in deployment versus in design
 - IBM Systems Sciences Institute

Traditional Development SDLC

- Requirements
- Architecture/Design
- Coding
- Integration Testing
- Deployment
- Maintenance





Requirements

- Make security requirements explicit
 - Sometimes difficult to state positively
 - Are simultaneous logins allowed for the same account?
 - What data should not be cached in a user's browser?
- Perform threat modeling when doing use case analysis



Architecture/Design

- Centralizing security-critical functions into subsystems can make auditing and verification easier
 - User input validation
 - Data store access
- Explicitly define which messages between subsystems are “trusted” and which are not
- What facilities in the operating system can be used to implement security requirements?
- Evaluate the use of cryptography to protect data in case of a system compromise
 - Defense in depth



Development

- Often secure coding issues and quality coding issues overlap
 - Verifying user inputs
 - Proper error handling
- Perform code reviews with security in mind
 - Reference the threat models developed during use case analysis
- Integrate verification of security measures into unit testing



Integration Testing

- Use automated application scanners to test major functional areas of applications
 - Do not let this be the *only* effort made to develop the application securely...
- Application penetration testing
 - Be evil. Try to break the application.



Deployment

- Use automated server and application scanners to verify deployment servers are correctly secured
 - Web servers
 - Application servers
 - Application configuration
- This is the phase that has the greatest cross-over with traditional InfoSec



Maintenance

- Periodic reviews
 - New exploits and tests are added to automated scanners
- How often do you need to re-certify applications?
 - Depends on scope of changes and experience of team doing maintenance
- Integrate security impact reviews into normal change management procedures
- Analyze configuration management procedures for security implications
 - Attacks on configuration management infrastructure
 - Management of files in production environments



Observations

- Organizations need to focus on application level security in an increasingly web-enabled world
- Integrating application security into the SDLC will produce better, more consistent results
- Increased coordination between info security and development groups a must



Potential Areas of Assistance

- Application security assessments
 - Black box and white box testing
- Secure application development
 - Paired development
 - Outsourced security QA
 - Hands-on training
- .NET-specific transition services