

# MAX 2007

CONNECT. DISCOVER. INSPIRE.

Shlomy Gantz

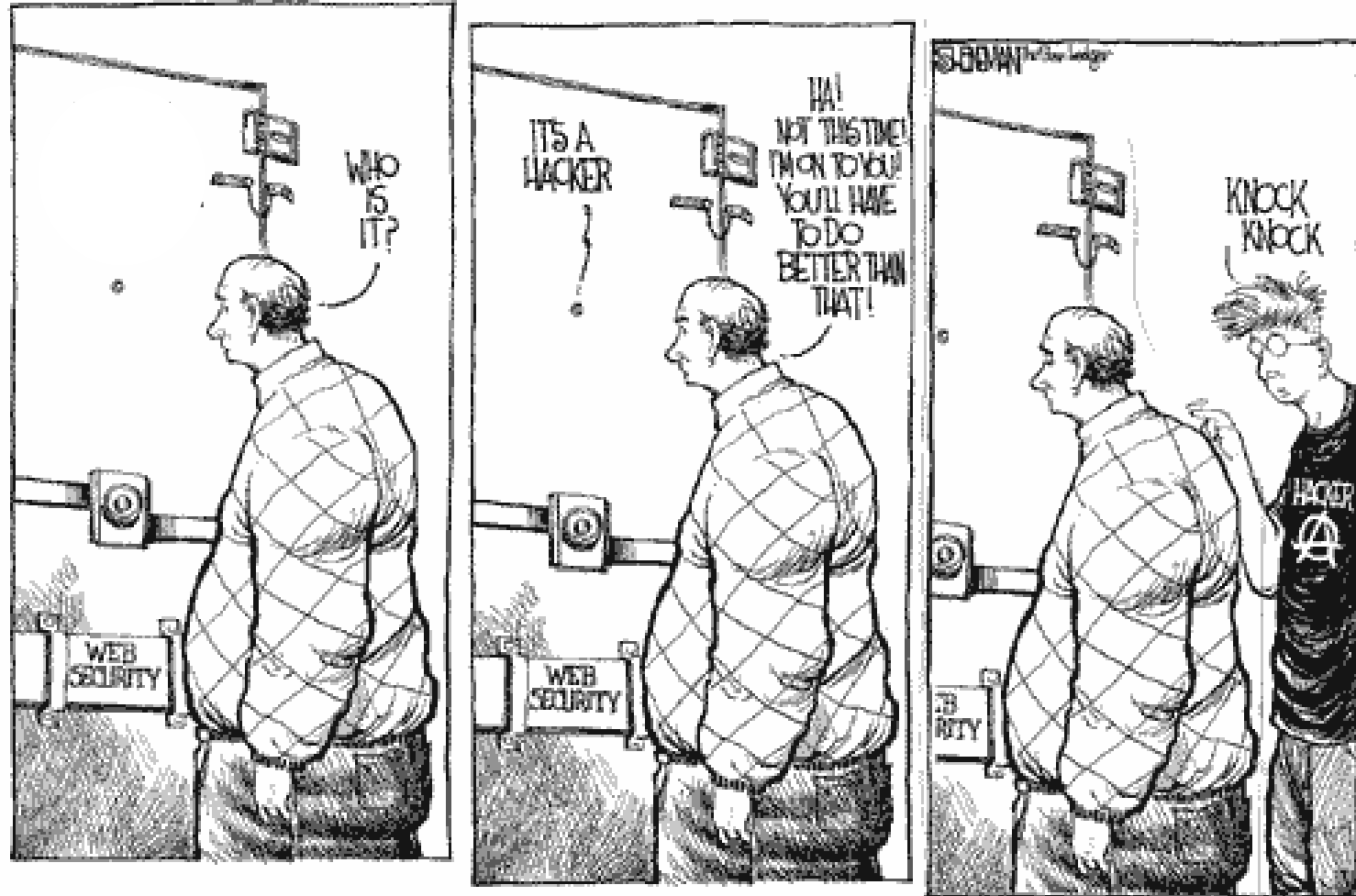
President,

BlueBrick Inc.

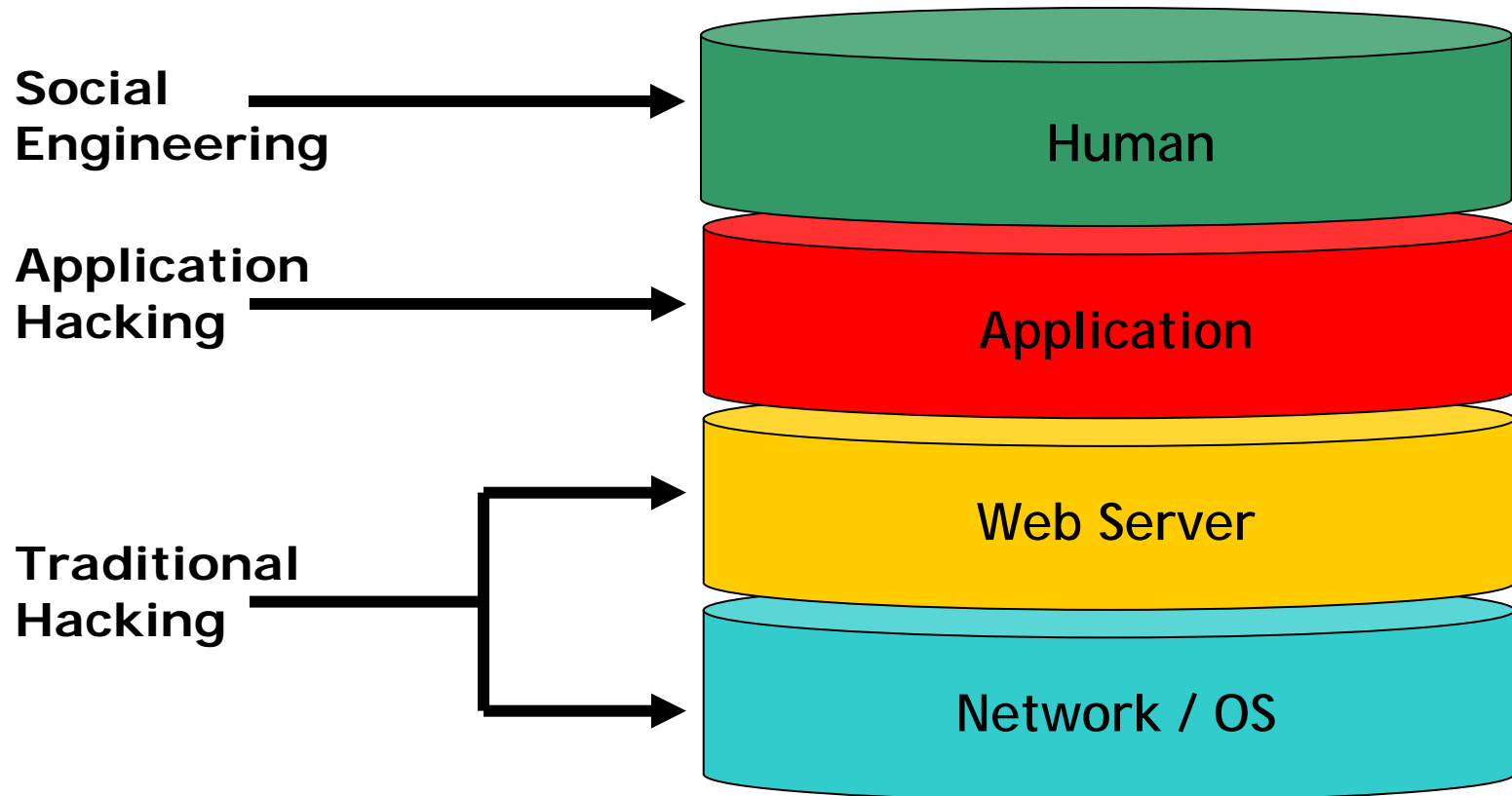


- President, BlueBrick Inc.
- 15 Years Software Development / Project Management
  - 12 Years of Web Experience
  - 11 Years of CF Development
  - Adobe Certified Instructor/Developer
  - Member, Adobe Community Expert
  - Manager, NYFLEX user group
- Author/Speaker
  - CFUNITED, MAX...
  - Co-Author - ColdFusion Developer's Handbook
  - ColdFusion Developer's Journal

- Progression Of Attack
  - Application Attacks
  - Application Vulnerabilities
  - Attack Progression
- OWASP Top 10 Vulnerabilities
- Beyond OWASP
- Q&A



# Layers Of Vulnerability



# But...

The logo for MAX, consisting of the letters 'M', 'A', and 'X' in a stylized, outlined font.

- I'm Using A Firewall...
- I'm Using SSL ...
- I've Installed The Latest Patches...



# Application Attacks

- Relate To The Meaning Of Application Messages:
  - Interpretation Of The HTTP Requests
  - Handling Of SQL Queries
  - Interpretation Of Application Specific Messages
  
- Harder To Identify Or Replicate
  - Requires Understanding Of Both Technology and Application Domain
  - Vulnerabilities Differ Between Applications
  
- Easier To Exploit...
  - Coding Is Simple
  - GUI Assisted

- Application Attacks Are Often More Dangerous
  - Involve Organization's Core Operation
  - (Infrastructure Attacks Usually Target The Servers Themselves Only)
  
- Harder To Repair...
  - May Require Code and Design Changes
  - Most Security Staff Has IT Background Rather Than Development Background

- Applications Assume Certain Client Behaviors
- Developers Anticipate Only “Real” User Will Input Data
- **ALL** Input Can Be Modified By The User
- Hackers Use A Variety Of Tools To Modify Data (Paros, ITR, Netcat...)

- **Footprinting**
  - Whois
  - DNS Lookup
  - HTTP Headers
- **Scanning -**
  - IP Addresses - Ping Sweep
  - Port Scan (TCP/UDP)
- **Enumeration**

- Responses May Contain Valuable Information Left Behind By Programmers Or System Administrators
- Information Can Be Found In:
  - Response Headers
  - Programmers Comments
  - Commented-out HTML/Javascript
  - Hidden Fields & Values
  - Client-side Scripts

- Do Not Include Any Redundant Code
  - Change All Comments To CFML
  - **Remove** Old Files
  - **Remove** Documentation
- Apply Global and Application Error Handler
  - <Cferror>
  - Onerror()
- Log, Alert and Review

- Gaining Access
  - Password/ Session Hijacking
  - Buffer Overflows
  - Application Vulnerabilities
- Escalating Privilege
  - Password Cracking
  - Application Vulnerabilities
- Data Theft/Pilfering
- Covering Tracks
  - Loading A "Root Kit"
  - Clear Log Files / Hide Footprints
  - Re-secure The System
  - Creating Back Doors For Re-entry



## Vulnerabilities - Application Top 10

- Unvalidated Input
- Broken Access Control
- Broken Account and Session Management
- Cross-site Scripting(xss) Flaws
- Buffer Overflows
- Command Injection Flaws
- Error Handling Problems
- Insecure Storage
- Application DoS
- Insecure Configuration

# 1. Unvalidated Input



- The Most Simple Form Of Application Attack
- Targets The Business Logic Of The Application
- Does Not Require Any Special Tools
- Can Be Done On Both Get and Post Variables

# 1. Unvalidated Input



- Forms
  - Hidden Fields
  - User Selection
  - Type and Range Validation
  
- URL
  - Master/Detail Pages
  - Attribute Parameters

# 1. Unvalidated Input - Mitigation



- Reduce Dependency On Hidden Fields By Using The Session Scope
- Do Not Rely On Client Side Validation Alone
- Check Validity Of User Selection and Input Type/Range
  - Use <CFPARAM> *Type*, *Pattern* and *Range* attributes
  - Use <CFINPUT> *validate* attribute onServer as well as onBlur

## 2. Broken Access Control



- Forceful Browsing
  - Static File Links
  - Hidden Files (Security By Obscurity)
  - File Name Predictions
    - Known System Files
    - .Log / .Old Files
- Path Traversal
- Client Side Caching
- File Permissions

## 2. Broken Access Control Mitigation



- Use Hash() To Perform Checksum Of URL
- Check Data Access Permissions On Every Request
- Control Access from a single location
  - Rely on session level variables rather than cookies
- Store Files to Download Outside Of Webroot
  - Use <CFCONTENT> To Serve Files To The User

Example: Hash()

- Account Authentication Bypassing
  - Login Tampering
  - Brute Force
  
- Session Hijacking
  - Brute Force
  - ID Predicting
  - Sniffing and Eavesdropping
  - Using HTTP\_REFERER When SessionID Is Passed On URL

### 3. Broken Account and Session Management - Mitigation



- Enforce At Least 8 Characters Password
- Require Numbers and Special Characters
- Do Not Sent Permanent Passwords Via Email
- Do Not Disclose Reason For Login Failure
- Expire Passwords
  
- Log, Alert and Restrict Access After Failed Login Attempts

### 3. Broken Account and Session Management - Mitigation



- Require Re-authentication On Email Change
- Consider Using SSL To Encrypt Transmissions
- Disable Browser Caching
- Use UUID For CFTOKEN
- Use J2EE Sessions
- Control Session Timeout

- Check CGI Variables
  - **CGI.HTTP\_REFERER**
  - **CGI.CF\_TEMPLATE\_PATH**
  - Note: They Can Be Spoofed!
- Check Cookies
- <CFLOGIN> functions
  - isUserInRole() , getAuthUser()
  - New in CF8 - IsUserInAnyRole()
  - New in CF8 - GetUserRoles()
  - New in CF8 - IsUserLoggedIn()
- <CFNTAuthenticate>

- **Stored**
  - Script Is Stored in Trusted Source
    - Forums
    - User Comments
    - Contact Forms
    - Online Web Mail System
- **Reflected**
  - Script Reflected Off The Web Server In
    - Error Messages
    - Search Results
    - ...

# 4. XSS - Mitigation



- Built in CF Protection
  - ColdFusion Admin Setting
  - `this.scriptprotect` in `Application.cfc`
- `HtmlTrans()`
  - <http://www.cflib.org/udf.cfm?id=945>
- `CF_XSSBLOCK`
  - <http://www.illumineti.com/documents/xssblock.txt>
- Log, Alert and Review Violations

Example: XSS

# 5. Buffer Overflow



- Application Fails To Allocate Sufficient Memory For Its Input
- May Allow The Attacker To Achieve:
  - Denial Of Service
  - Remote Command Execution
  - Data Alteration/ Leakage
- May Appear in Sub Components Executed By Non Vulnerable Code

# 6. Command Injection Flaws



- SQL Injection
  - Alter The Syntax Of The SQL Statements Sent By The Application To The Server
- Not Just SQL

Example: HTML Injection

## 6. Command Injection Flaws - Mitigation



- `<CFQUERYPARAM>`
- Consider Stored Procedures
- Limit DB Permissions On CF Admin and in Database
- Disable XP\_cmdshell and Equivalents
- Consider Server Sandboxing

Example: SQL Injection

- Error Messages Might Disclose Sensitive Information
  - Directory Structure
  - Code Snippets
  - Query Structure

# 7. Error Handling - Mitigation



- Disable Debugging on Production
- Define Site Wide Error and 404 Handler
- `<CFERROR>` / `OnError()`

- Storing Sensitive Information Using Inadequate Encryption Schemas
  - Failure to encrypt critical data
  - Insecure storage of keys, certificates, and passwords
  - Improper storage of secrets in memory
  - Poor sources of randomness
  - Poor choice of algorithm
  - Attempting to invent a new encryption algorithm
  - Failure to include support for encryption key changes and other required maintenance procedures

# 8. Insecure Storage - Mitigation



- Encrypt Sensitive Data
  - Encrypt()/Decrypt() – Two Way
    - Uses Symmetric Key
    - CF7
      - Additional Algorithms (AES,BLOWFISH,DES...)
      - Generatesecretkey
    - CF8
      - RSA BSafe encryption
  - Hash() – One Way
    - Impossible To Revert
    - Does Not Require Key
    - Best For Passwords

## Rendering A Service Offered By A Workstation Or Server Unavailable To Others

- Reasons:
  - To Get A System Reboot
  - Hacker Covering His/Her Tracks
  - Malicious Intent
  
- How It's Done:
  - Ping Of Death - ICMP Techniques
  - Syn (Network) Vulnerabilities
  - Data DoS

- Update With Latest Patches
  - ColdFusion
    - [http://www.adobe.com/support/coldfusion/downloads\\_updates.html](http://www.adobe.com/support/coldfusion/downloads_updates.html)
  - Web Server
  - OS
- Optimize Use Of Session Resources
  - Use Caching
  - Avoid Large Data Sets in Session

- Many Developers Do Not Configure Their Server Beyond The Initial Install
  - Missing Patches
  - Sample Files
  - Default Accounts

- **Configure CF Admin**
  - New in CF8
    - RDS sandbox support
    - User-based Administrator access
  - Secure Admin Directory With NT Authentication Or Completely Remove
  - Do Not Deploy Docs, Sample Apps and RDS To Production
  - Do Not Store DB Password in code
  - Disable Unused Services



## Beyond OWASP

- CFCs can be used as back-end for:
  - Flash/Flex
  - AJAX
  - SOAP
  - Non-browser based application
  
- New in CF8
  - VerifyClient
  - secureJSON

- **Simply Asking For :**
  - Information
  - Passwords
  - Assistance
  
- **Requires No Technical Skills**

- Integrate Security Into Your SDLC
  - Hack Test During/After Development
  - Create Anti-requirements
  - Review Code Regularly

- Open Web Application Security Project (OWASP)
  - <http://www.Owasp.org>
- CGI Security,
  - <http://www.Cgisecurity.Net>
- Web Application Security Mailing List,
  - <http://online.Securityfocus.Com/Archive/107>
- Hacktics
  - <http://www.Hacktics.Com/Presentations.Html>
- MIT Publications
  - <http://Pdos.Lcs.Mit.Edu/Cookies/Pubs.Html>  
“Dos and Don'ts Of Client Authentication On The Web”

Shlomy Gantz

[shlomy@bluebrick.com](mailto:shlomy@bluebrick.com)

<http://www.shlomygantz.com/blog>