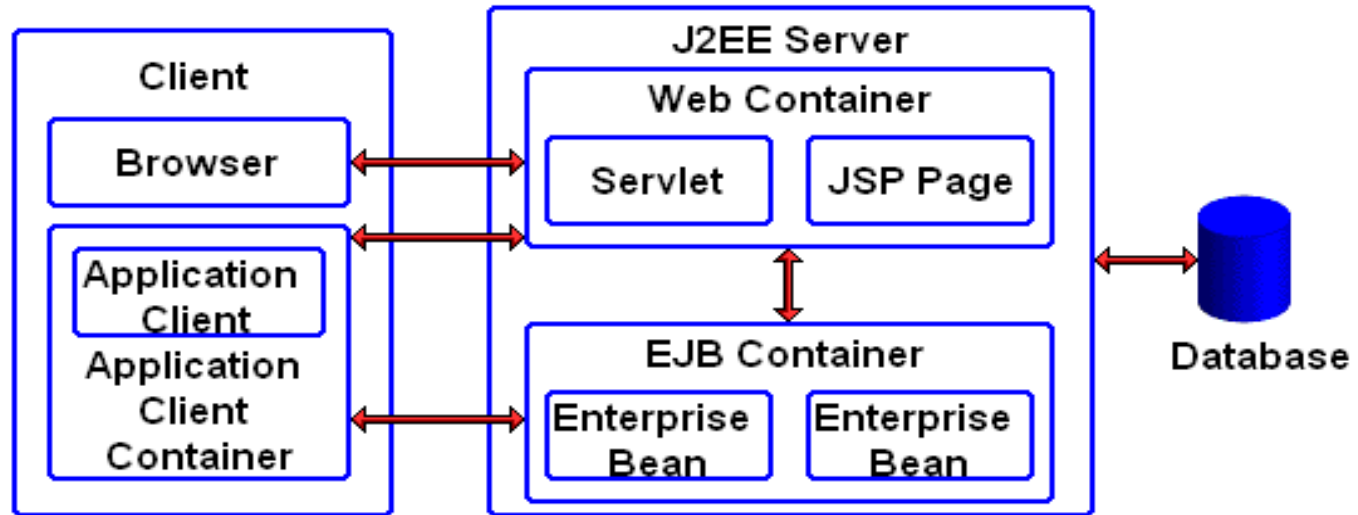




Web Application Gray Box Testing

Bob Grill
Wells Fargo Audit & Security

What are we talking About.



Presentation Layer

Consists of components that manage user interface and user interactions

Business Logic Layer

Consists of components that implement business rules and logic

Data Layer

Used for data persistence

Focus

- Java (Interpreted from Bytecode)
 - Beans
 - Servlets
 - JSP
 - Applets (No thick clients, or mobile code (Bytecode) in Presentation – Only Web Browser Clients)
- Containers
 - Websphere
 - Web Logic
- Web Servers
 - Iplanet (Sunone)
 - Apache (Free version of Iplanet)
- Not Covering Administrative Console controls and configuration files for Web Server or Application servers.

Gray Box Testing Definition

- Reviewing Configuration files on Web Server and Application Servers.
 - Not Looking At Source Code
 - Not Penetration Testing

Testing Approaches - From Inside Out or From Outside In

- From Inside Out
 - Configuration Files
 - Code Review
- Advantages – Saves Time, Focuses Black Box Testing on Proving Weaknesses.
- Disadvantages – Extreme Complexity, Harder To Learn.

Approaches – From Outside In

- From Outside In (Black Box Testing)
 - Hacking – Testing using proxy to overcome client side input controls and manipulate, cookies, hidden values and other parameters.
 - Intelligent Guessing based on Standard Naming Conventions for Software Installed.
 - Parameter Manipulation based on parameters noted during higher authorization levels.

Black Box Alone - Disadvantage

- Guessing can be time consuming
- If Guess Wrong can miss Exploit
- Can't tell developer what to fix, only that the symptoms say it is broken.

Web Application Overlap

Browser Side

Server Side

Input Controls - Only

Source Code Dynamically Created in Browser By Web & Application Server

Symptom

Web Server (IIS, SunONE, IIS, Etc.)

Presentation

- HTTP Methods
- Directory Indexing
- SSL
- Cookies
- Web Root File Permissions
- URL Filtering
- Tokens – From SAML or Siteminder
- Logging and Monitoring
- Patch Levels
- Change Control of Web Root
- Web Services
- Maintenance of Web Root

Infrastructure Only

- System Administration
- UNIX
- Performance Tuning

Application Server (Websphere, WebLogic, Dynamo, Etc.)

Root Cause

Presentation

- BEAN / Servlet Permissions and Coding
- Cookies
- Session Management
- SSL
- Autologin Settings
- Password Storage
- Web Services
- Pass through Web Root

Infrastructure

- System Administration Permissions
- Performance Monitoring
- Encryption Between Machines

- SAML
- SOR's
- Intermediate Databases

Input - One Basic Control

- Why so difficult
 - Abstraction layers use ASCII as Commands.
 - Can't filter all command codes and encoding schemes (O'Tool)
 - Thousands of Inputs per Web Site
 - Application will work without input controls
 - Costs Money / Time to Program
 - Reduces speed

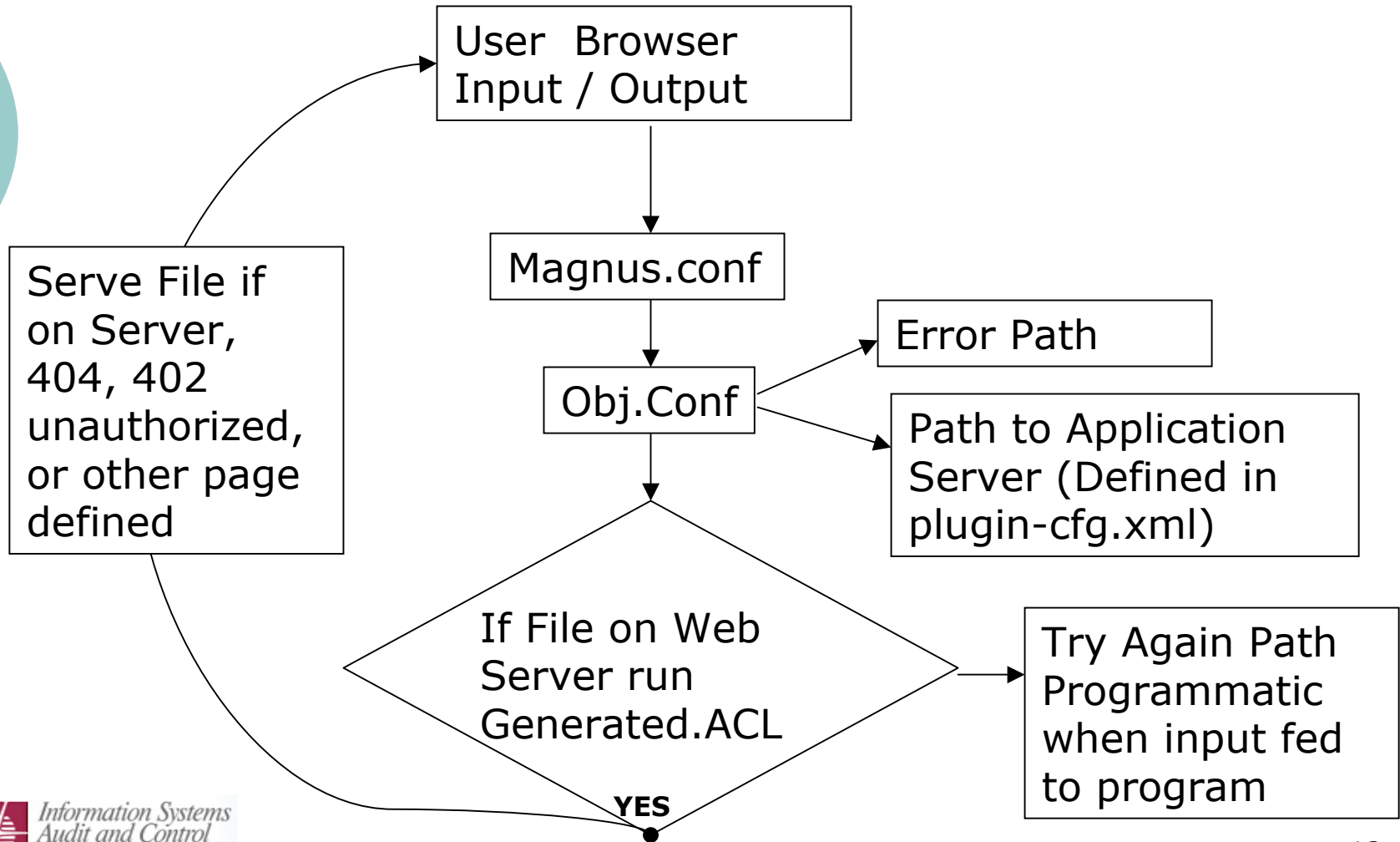
Scope of Audits – Methods Different, Same Controls Tested

	Presentation Layer Controls Testing Methods					
Testing Methodology and Who Tests Performed By	Audit	Method 1	Audit	Method 2	Audit-Method 3	
Alias	Stimulus / Response Analysis of Server Code		Configuration Review		Code Review	
Control Categories / Alias	Black Box Testing (of server side) and Client side code review.		Gray Box Testing - Root Cause of Black Box Testing Findings.		White Box Testing	Risk
Policies, Standards and Governance	No		Yes	Manual Processes	No	Low
Input Controls	Yes	Browser Source Code and Stimulus Response Analysis	Limited	Web Server (HTTP Headers)	Yes	High
File Permissions (from internet)	Limited	Browser Source Code and Stimulus Response Analysis	Yes	Web Server / Web Root	No	High
Object Permissions	Limited	Browser Source Code and Stimulus Response Analysis	Yes	Application Server	No	High
State Maintenance	Yes	Browser Source Code and Stimulus Response Analysis	Yes	Application Server	Limited	High
Privacy	Yes	Browser Source Code and Stimulus Response Analysis	Yes	Cache Control on Web Server	Limited	High
Change Control	No	N/A	Yes	Web Server / Web Root, Application Server Tables	No	Medium
Configuration (Presentation Layer)	Limited	Browser Source Code and Stimulus Response Analysis	Yes	Web Server / Application Server	No	High
Programming and QA Standards	Yes	Browser Source Code and Stimulus Response Analysis	No	N/A -	Yes	High
Logging (application layer)	No	N/A	Yes	Web Server / Manual Processes	No	High
Intrusion Detection (application)	Limited	Browser Source Code and Stimulus Response Analysis	Limited	Web Server / Manual Processes	No	High
Authentication / Authorization (Assertion vulnerability Only)	Limited	Browser Source Code and Stimulus Response Analysis	Limited	Site Minder/ LDAP / SAML Server	No	High
Web Services	Yes	Browser Source Code and Stimulus Response Analysis	Limited	Web Server / Application Server (Intra Application Web Services not Covered as Part of the presentation layer unless exposed to Internet.)	No	High
Encryption	Limited	Browser Source Code and Stimulus Response Analysis	Yes	Web Server / Application Server	No	High
Availability	No	N/A	Yes	Manual Processes	No	High
Administration of Web & Application Server.	Yes	Limited	Yes	Manual Processes	N/A	N/A

Main Web Server Configuration Files

- Magnus.conf – Port Instance is Running, SSL settings, ID Web Server Running as, Points to Obj.Conf, points to “generated.https-WebServername.Domain.com.acl
- Obj.Conf – Contents of Named Access Control Lists. Points to other configuration files, Doc Root.

Flow on Web Server – Simple File Server.



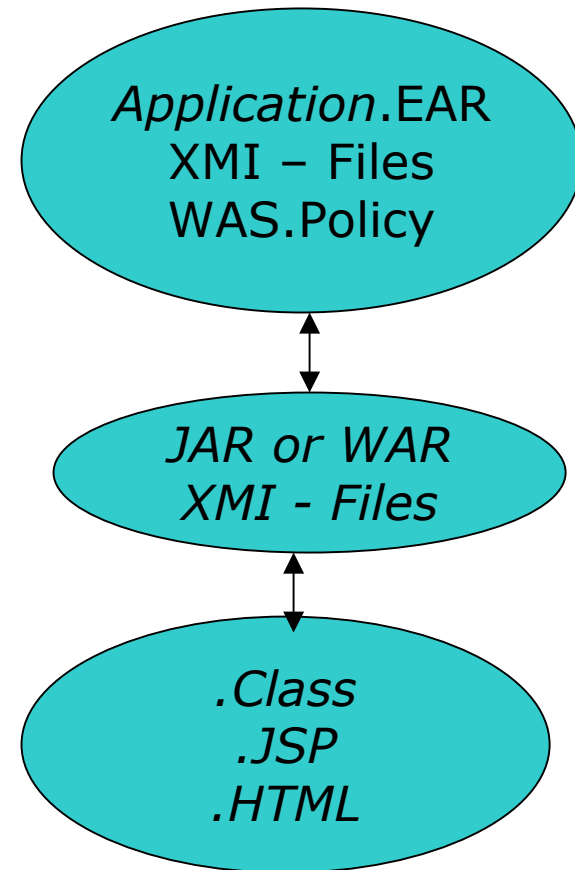
Generated.ACL File Sample - Generally HTTP Methods & File Path Restrictions

```
generated.https-Servername.DomainName.com.acl
version 3.0;
acl "ISACA1"; (Look in Obj.conf for Detail of This)
authenticate (user, group) {
    prompt = "ISACA1 Server Login"; - a prompt facing the
Internet.
};
allow (read, list, execute, info) user = "anyone";
allow (write, delete) user = "all"; All means all who are
authenticated.
```

J2EE Packaging - Application Server – Websphere Example

Java Containers Have the Same
Functionality because they are Based on Java Standards – How
Implemented is Different.

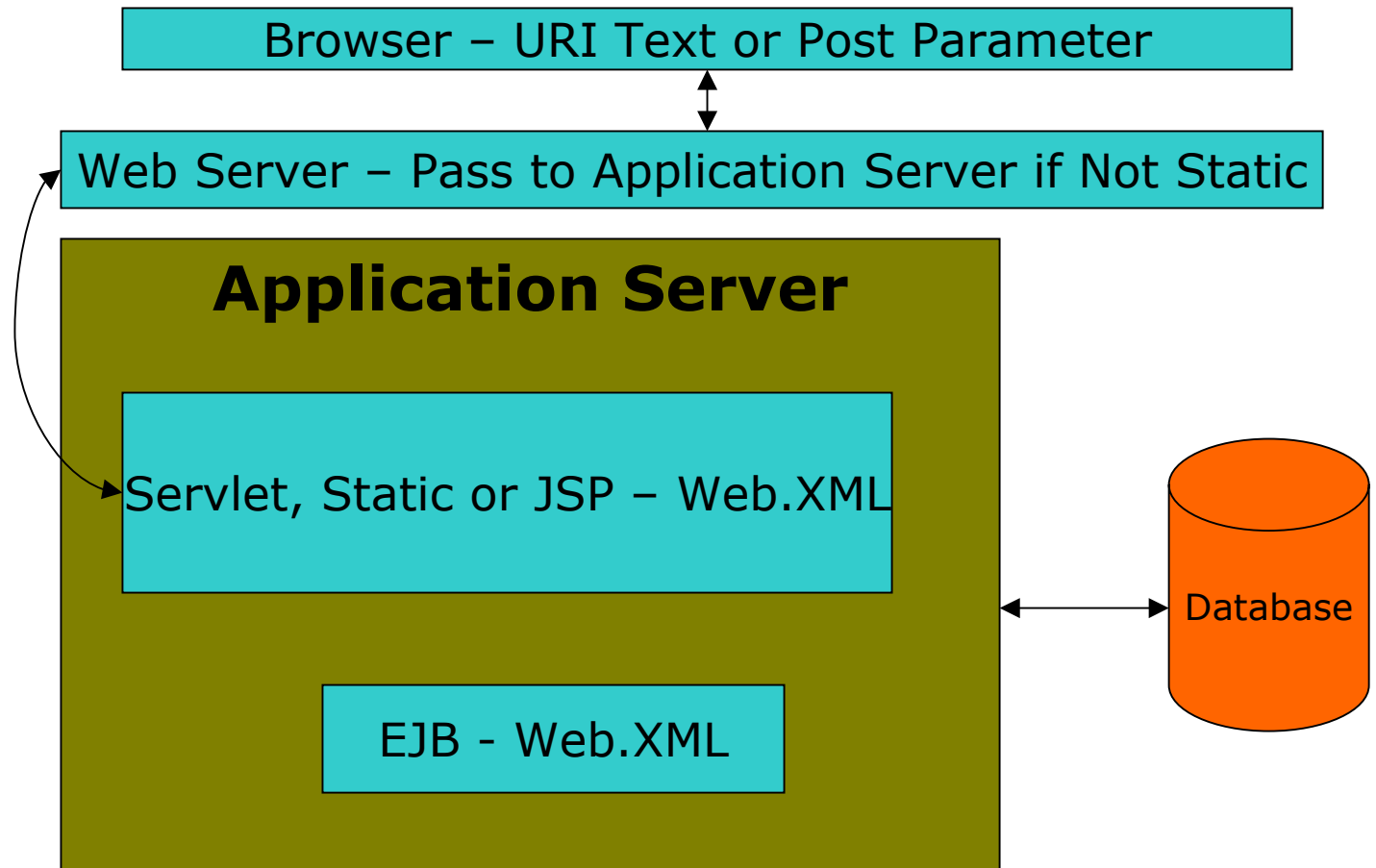
Tip: JAR and WAR are
Just Zip files of all the
Class Files. Think of
Class files as the code
for each business
transaction individually.



Configuration Files - Kept in Directory called /WEB-INF for each Application.

- Web.xml – Tags Describe Security Roles, Maps Role Names to Web Resources (Bean and Servlet names, Methods) using URI Patterns.
- Application.xmi – Security Roles used for methods.
- IBM-Application-bind.xmi - Security roles mapped to users.
- IBM-Application-ext.xmi – Serve Servlets by Classname.
- EJB-JAR.XML – Defines Method Permissions
- Struts-config.xml –shows the path to the validator.xml file
- Validator.xml – what input is filtered.
- sas.client.props, sas.server.props – Security Between Machines and thick clients.

Configuration File Decision Flow



Best Practice is Servlet for Incoming JSP for outgoing

Fast Auditing

- Session Timeouts
- SSL Settings
- Cookie Settings
- Servlet Permissions
- Method Permissions
- Web Paths
- Servlet Aliases

Let's Break It – Programming Taking the Obscurity out of Parameter Names

- Arbitrary Servlet Invocation
- Role Referencing
- Hidden Parameter Value Keys

Let's Break It - Configuration

- HTTP Method Insertion
- Web Root Obscure file Name / Path Invocation – Verify Anything Sensitive is secured by ACL and no Data or Code on Web Server.

Conclusion

- Black Box Testing Method Alone Not Sufficient
- Properties Files Provide Way to Change Java Program without Reprogramming.
- Inside Out Testing – More efficient and Effective than Outside in.
- Next Year - White Box Presentation.