Breaking is easy, Preventing is hard
Secure in 2010? Broken in 2011!

Matias Madou, Ph.D.
Principal Security Researcher
Matias Madou

- Principal Security Researcher, HP Enterprise Security (formerly Fortify)
  - Static Analysis: Standard rules + Customization
  - Insider Threat Research
  - Hybrid: Static and Dynamic result correlation
  - Gray-box analysis (HP WebInspect + Fortify SecurityScope)
- Presentations @ DefCon, BlackHat, RSA Conference, …
- Contributor to Building Security in Maturity Model (BSIMM) Europe
- History in code obfuscation (and binary rewriting)
Overview

- Introduction
- The Test Application: Secure in 2010
- What’s new in 2011?
  - New vulnerabilities
  - New analysis techniques
- Continues Testing
Introduction

History of the experiment: Gather empirical results while developing gray-box analysis.

Test Application, criteria:

- Extensively used
- Undergone security improvements
The Test Application

• Selection criteria for the project working on:
  – Open source, java or .NET
  – Widely used

• Apache
The Test Application

• End Users:
  – 1-800-Flowers
  – Olympus.de
  – United.com
  – BT.com
  – …
The Test Application

- Products and Projects based on Apache OFBiz:
  - OpenTaps

<table>
<thead>
<tr>
<th>Product/Project</th>
<th>License</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFBiz info</td>
<td>Free access</td>
<td>In2M</td>
</tr>
<tr>
<td>Mvelopes</td>
<td>Commercial, Free Trial</td>
<td>Opus Services</td>
</tr>
<tr>
<td>TurboPaye</td>
<td>Commercial, Free Trial</td>
<td>Emforium Group Inc.</td>
</tr>
<tr>
<td>ALL-IN Software</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td>Atlassian JIRA</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td>openTaps Open Source ERP + CRM</td>
<td>HPL and Commercial</td>
<td>Open Source Strategies</td>
</tr>
<tr>
<td>OZoom</td>
<td>GPL3</td>
<td>Mapea S.p.A., TD Group</td>
</tr>
<tr>
<td>Nexpia</td>
<td>GPL</td>
<td></td>
</tr>
<tr>
<td>SourceTap CRM</td>
<td>GPL and Commercial?</td>
<td></td>
</tr>
<tr>
<td>Npulion SAVANNA</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td>Codacase Helix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ova</td>
<td>GPL 3</td>
<td>C-Libre</td>
</tr>
<tr>
<td>@Strategic Power Office</td>
<td>Commercial</td>
<td>Businessesnetwork.com</td>
</tr>
<tr>
<td>myofbiz.com</td>
<td>n/a</td>
<td>Adaptive Enterprise Solutions, LLC</td>
</tr>
<tr>
<td>OrangeGears Project</td>
<td>Apache License Version 2.0</td>
<td>OrangeGears</td>
</tr>
<tr>
<td>Saas-Suite OFBiz</td>
<td>Commercial/APL</td>
<td>Corent Technology</td>
</tr>
</tbody>
</table>
The Test Application

- Security?
  - Multiple vulnerabilities found in CVE
  - Other (Exploit Search)

- … and an interesting video on how to become an admin by exploiting a XSS
The Test Application

Since the application does not properly sanitize the users input, we inject our payload. The payload will get executed when Administrator accesses his inbox.

The payload performs the following actions:

- The payload stays hidden from the users view by applying customized javascript techniques thus its completely stealth to the victim.
- Creates a new user with full privileges by the means of several XMLHttpRequests.
- Sends an HTTP request to the attacker’s server to inform successful exploitation.
The Test Application
The Test Application
The Test Application

- Bug Tracking: Security Issues grouped together
The Test Application

• In the end: All known issues are fixed in Apache OFBiz 10.04

Secure in 2010!
So... what’s new in 2011?

1) New vulnerabilities: Denial-of-service: Parse Double

2) Analysis techniques: Gray box analysis
So… what’s new in 2011?

1) New vulnerabilities:
   Denial-of-service:
       Parse Double

2) Analysis techniques:
   Gray box analysis
Denial-of-Service: Parse Double

CVE-2010-4476 (Feb 1, 2010)

- Value: \(2.2250738585072012 \times 10^{-308}\)
- API: `Double.parseDouble(value)`

Infinite loop!

Fixed: Feb 8, 2011
Denial-of-Service: Parse Double

- Feb 01, 2011? No, no. March 04, 2001!

- Why is this fixed within 1 month after the rediscover?
Denial-of-Service: Parse Double

Examples:

• Application: Apache Tomcat
• Usage: Tomcat uses parseDouble() on the value of the Accept-Language HTTP header when an application calls request.getLocale()

Big trouble!

http://blog.fortify.com/blog/2011/02/08/Double-Trouble
Denial-of-Service: Parse Double

How many issues in Apache OFBiz?

Used analysis techniques:

- Static Analysis (White Box)
- Penetration Testing (Black Box)
Denial-of-Service: Parse Double

Static Analysis (White Box)
Penetration Testing (Black Box):

http://yourofbiz.com/ecommerce/control/modifycart (update_0, update_1, …)
http://yourofbiz.com/ecommerce/control/additem/showcart (quantity, add_product_id)
http://yourofbiz.com/ecommerce/control/additem/quickadd (quantity)
http://yourofbiz.com/ecommerce/control/additem/keywordsearch (quantity)
http://yourofbiz.com/ecommerce/control/additem/advancedsearch (quantity)
http://yourofbiz.com/ecommerce/control/additem/showPromotionDetails (quantity)
http://yourofbiz.com/ecommerce/control/additem/product (quantity, add_amount)
http://yourofbiz.com/ecommerce/control/additem/lastViewedProduct (update_0)
http://yourofbiz.com/ecommerce/control/additem/showForum (quantity)
http://yourofbiz.com/ecommerce/control/additem/category (quantity)
http://yourofbiz.com/ecommerce/control/additem/main (quantity)
http://yourofbiz.com/ecommerce/control/additem (quantity)
http://yourofbiz.com/ecommerce/control/additem/setDesiredAlternateGwpProductID (…)
…
Denial-of-Service: Parse Double

What is the problem?

- Root case is a Java problem, not an application problem!
- Everybody uses the fixed java version, right? (Version Java 6 Update 24 or later)

Because of lack of updating java, Apache tomcat installed additional checking. (Tomcat 7.0.8, 6.0.32, 5.5.33 or later)

```java
int semi = entry.indexOf(";q=");
if (semi >= 0) {
    try {
        String strQuality = entry.substring(semi + 3);
        if (strQuality.length() <= 5)
            quality = Double.parseDouble(strQuality);
    }
```
So… what’s new in 2011?

1) New vulnerabilities:
   Denial-of-service:
   Parse Double

2) Analysis techniques:
   Gray box analysis
What is Gray-Box Analysis

• Well… what is black-box analysis?
  – Can see that something is truly wrong
  – No inside information
Black-box Testing (Penetration Testing)

Penetration testing

Application

http://www.testapp.com/index.html?q=a’or+1=1-

App crashes…
What is Gray-Box Analysis

• Get inside information
• Easier to find out what’s wrong and where exactly so easier to fix
Gray-box analysis: Integrated Analysis

- Find More
- Fix Faster

Penetration testing

Real-time link

Application

Monitoring agent
Find More

• Detect new types of vulnerabilities
  – Privacy violation, Log Forging

• Find more of all kinds of vulnerabilities
  – Automatic attack surface identification
  – Understand effects of attacks
Attack surface identification

- File system
- Configuration-driven
- Programmatic
Attack surface identification: Apache Ofbiz

**Black-box**

- https://10.100.60.50:8000/
  - accounting
  - ap
  - ar
  - as
  - btl
  - btl
  - catalog
  - catalog
  - content
  - content
  - db
  - db
  - ecommerce
  - ecommerce
  - example
  - example
  - images
  - images
  - marketing
  - marketing
  - privacy.html
  - privacy.html

- https://10.100.60.50:8000/
  - accounting
  - ap
  - ar
  - as
  - btl
  - btl
  - catalog
  - catalog
  - content
  - content
  - db
  - db
  - ecommerce
  - ecommerce
  - example
  - example
  - images
  - images
  - marketing
  - marketing
  - privacy.html
  - privacy.html

**Gray-box**
Understand effects of attacks

Command Injection

sysadmin$ ./sh
Fix Faster

- Provide Actionable Details
  - Stack trace
  - Line of code
- Group Symptoms with a Common Cause
Actionable Details

SecurityScope Stack Trace:

```
    at org.apache.tomcat.dbcp.PoolingDataSource.PoolGuardConnectionWrapper.prepareStatement(PropertySpec.java:46)
    at com.fortify.samplesركيزات. findAllLocations.execute(findAllLocations.java:45)
    at sun.reflect.GeneratedMethodAccessor64.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:555)
    at com.opensymphony.xwork2.DefaultActionInvocation.invokeAction(DefaultActionInvocation.java:404)
    at com.opensymphony.xwork2.DefaultActionInvocation.invokeAction(OnlyIfDefaultActionInvocation.java:20)
    at com.opensymphony.xwork2.DefaultActionInvocation.invoke(DelegatedActionInvocation.java:229)
    at com.opensymphony.xwork2.interceptor.DefaultWorkflowInterceptor.doIntercept(DefaultWorkflowInterceptor.java:221)
    at com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:150)
    at org.apache.struts2.interceptor.validation.AnnotationValidationInterceptor.doIntercept(AnnotationValidationInterceptor.java:150)
```

RWi Online Banking

Username:  
Password:  

Need to set up online access?  
Go to Help Center.
Fix Faster: Actionable details

- **Severity:** 1 Critical (17 items)
  - Duplicates: Guestserver Arbitrary Command Execution - http://10.100.60.50:8080/ecocommerce/products/products/guestbook.cgi (1 item)
  - Duplicates: Cross-Site Scripting - http://10.100.60.50:8080/webstinger/<Imgs Src=x OnErr0R=alert(54385)> (5 items)
  - Duplicates: SimplestMail Arbitrary Command Execution - http://10.100.60.50:8080/ecocommerce/products/products/simplestmail.cgi (1 item)
  - Duplicates: Blind SQL Injection (confirmed) - http://10.100.60.50:8080/ecommerce/control/additem/ (1 item)
  - Duplicates: Cross-Site Scripting - https://10.100.60.50:8443/ecocommerce/control/silentAddPromoCode (1 item)
- Duplicates: Host SMTP Web Application - Multiple Possible Vulnerabilities (mailform.exe) - http://10.100.60.50:8080/ecommerce/products/products/ad.cgi (1 item)
- Duplicates: Cross-Site Scripting - http://10.100.60.50:8080/cmsite/cms</CMSS_PPOINT/>--&gt;Imgs Src=x OnErr0R=alert(18718) (1 item)
- Duplicates: Hardcoded Mail Command Execution - http://10.100.60.50:8080/ecommerce/products/products/ad.cgi (1 item)
- Duplicates: Blind SQL Injection (confirmed) - http://10.100.60.50:8080/ecommerce/products/products/ad.cgi (1 item)
- Duplicates: Mail Spoofing Vulnerability - http://10.100.60.50:8080/ecommerce/products/products/mailsend.exe (1 item)
- Duplicates: Cross-Site Scripting - http://10.100.60.50:8080/cmsite/cms/CMS5_DEMO_PAGE/CreateImgs Src=x OnErr0R=alert(52528) (1 item)
Fix Faster: Actionable details

Cross-Site Scripting
This stack trace is from the running application and was returned by SecurityScope. It can be used to determine root cause.

SecurityScope Trigger:
<!-- no sub-content found with map-key [ ] -->\n\nSecurityScope Stack Trace:

```java
at org.apache.catalina.connector.CoyoteWriter.write(CoyoteWriter.java:171)
at java.io.PrintWriter.append(PrintWriter.java:960)
at java.io.PrintWriter.append(PrintWriter.java:35)
at org.ofbiz.content.content.ContentWorker.renderSubContentAsText(ContentWorker.java:358)
at org.ofbiz.content.cms.CmsEvents.cms(CmsEvents.java:291)
at sun.reflect.GeneratedMethodAccessor2982.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:597)
at org.ofbiz.webapp.event.JavaEventComponent.invoke(JavaEventComponent.java:317)
at org.ofbiz.webapp.event.JavaEventHandler.invoke(JavaEventHandler.java:92)
at org.ofbiz.webapp.event.JavaEventHandler.invoke(JavaEventHandler.java:78)
at org.ofbiz.webapp.control.RequestHandler.runEvent(RequestHandler.java:636)
at org.ofbiz.webapp.control.RequestHandler.doRequest(RequestHandler.java:382)
at org.ofbiz.webapp.control.ControlServlet.doGet(ControlServlet.java:227)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:617)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:208)
```

Sink: applications/content/src/org/ofbiz/content/content/ContentWorker.java:

```java
341 public static void renderSubContentAsText(Dispatcher dispatcher, Delegator delegator, String contentId, Appendable out, String mapKey, 358 out.append("<!-\n" -- no sub-content found with map-key [" + mapKey + "] for content [" + contentId + "] --"));
```
Group Symptoms with a common cause

- Counting issues seems to be hard!

36 vulnerabilities from the attackers perspective
1 vulnerability from the developers perspective
Fix Faster: Group symptoms

<table>
<thead>
<tr>
<th>Severity: 1 Critical (28 items)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicates: Cross-Site Scripting - [URL] (1 item)</td>
</tr>
<tr>
<td>Cross-Site Scripting - [URL] (GET)</td>
</tr>
<tr>
<td>Duplicates: Cross-Site Scripting - [URL] (GET)</td>
</tr>
<tr>
<td>Cross-Site Scripting - [URL] (GET)</td>
</tr>
<tr>
<td>Cross-Site Scripting - [URL] (GET)</td>
</tr>
<tr>
<td>Cross-Site Scripting - [URL] (GET)</td>
</tr>
<tr>
<td>Duplicates: SQL Injection (continued) - [URL] (1 item)</td>
</tr>
<tr>
<td>Duplicates: Cross-Site Scripting - [URL] (GET)</td>
</tr>
<tr>
<td>Cross-Site Scripting - [URL] (GET)</td>
</tr>
<tr>
<td>Duplicates: Cross-Site Scripting - [URL] (GET)</td>
</tr>
<tr>
<td>Cross-Site Scripting - [URL] (GET)</td>
</tr>
<tr>
<td>Duplicates: Cross-Site Scripting - [URL] (GET)</td>
</tr>
<tr>
<td>Cross-Site Scripting - [URL] (GET)</td>
</tr>
<tr>
<td>Duplicates: Cross-Site Scripting - [URL] (GET)</td>
</tr>
</tbody>
</table>
Group symptoms: details

- Detailed information on where to fix the issue
Wrap-up

Applications found secure in 2010

Broken in 2011 because:

1) New vulnerabilities:
   Denial-of-service:
     Parse Double

2) Analysis techniques:
   Gray box analysis

How to prevent?
Typical security throughout the SDLC

• How about security testing in production?
Pen testers find something, rely on WAF

- Seen in the field: adding the pattern to WAF
- Problems:
  1. Does not protect against persistent
  2. Are you sure your patterns cover everything?
     Pattern often used:
     - 2.2250738585072012e-308
     How about:
     - 0.22250738585072012e-307
Denial-of-Service: Parse Double

- Seen in the field: adding the pattern to WAF
- Problems:
  2. Are you sure your patterns cover everything?

Text: "Tomcat is vulnerable to a DoS if the accept-language header contains ';q=2.2250738585072012e-308' and other very small values. The"
So… what did we do?

Took a released application

- hit it with new analysis technique
- Search for vulnerabilities that were not known at the release day
Solution to keep it protected

• How about the application in production?
Solution to keep it protected

- Even if there are no code changes at all: keep scanning with updated security knowledge

- This way, you’ll find new ways of breaking your application
Solution to fix Apache Ofbiz?

- It’s still open source, so you can DIY

(quoted from the bug database)

- Scott Gray added a comment - 29/Apr/08 03:18
  I think the "policy" is a bit more like this:
  If you want it, either do it or pay someone else to do it.

(found in the bug database)
Solution to fix Apache Ofbiz?

• Run the Java 6 Update 24 or later (no DoS: Parse Double issues)
• XSS issues reported in CVE-2012-1621:
  Upgrade from version 10.04 to 10.04.02
THANK YOU!

QUESTIONS?

Matias Madou, mmadou@hp.com

https://www.surveymonkey.com/s/Research12_MatiasMadou