目录

序言：WebGoat 中文概述

WebGoat 是由著名的 OWASP 负责维护的一个漏洞百出的 J2EE Web 应用程序，这些漏洞并非程序中的 bug，而是故意设计用来讲授 Web 应用程序安全课程的。对于每堂课，用户须通过搞定 WebGoat 应用程序中的一个实际的安全漏洞来验证各自对于该安全问题的理解。举个例子，在其中一个课程中，用户必须使用 SQL 注入来窃取（杜撰的）信用卡号。这个应用程序提供了一个逼真的教学环境，为用户完成课程提供了有关的线索及代码，从而使用户能更深入地理解该课程。

为什么叫“WebGoat”呢？即使是最好的程序员也会写出产生安全漏洞的代码，因而开发人员不必因为不懂安全而自卑。他们需要的仅仅是一个“替罪羊”，不是吗？就让这只“山羊”背黑锅吧！

请参考“WebGoat 用户及安装指南”开始。

目标

在学习和实践 Web 应用程序安全知识时，我们所面临的一大难点是：到哪里去找可以练手的 web 应用程序呢？显然，明目张胆地扫描在线书店或者网络银行可不是个好主意，小心警察叔叔会找上门来。此外，安全专业人员经常需要测试某些安全工具，以检查它们的功能是否如厂商所鼓吹的那般，这时他们就需要一个具有确定漏洞的平台作为活靶子。但是，无论学习 web 测试，还是检查工具的性能，都要求在一个安全、合法的环境下进行。即使你的意图是好的，但是在未经许可的情况下企图查找安全漏洞也是绝不允许的。

WebGoat 项目的主要目标很简单，就是为 Web 应用程序安全学习创建一个生动的交互式教学环境。将来，项目研究小组希望将 WebGoat 发展成为一个安全性基准测试程序平台和一个基于 Java 的蜜罐网站。

概要

WebGoat 是用 Java 语言写成的，因此可以安装到所有带有 Java 虚拟机的平台之上。此外，它还分别为 Linux、OS X Tiger 和 Windows 系统提供了安装程序。部署该程序后，用户就可以进入课程了，该程序会自动通过记分卡来跟踪用户的进展。当前提供的训练课程有 30 多个，其中包括：

跨站点脚本攻击（XSS）

访问控制（Access Cnotrl）

线程安全（Thread Safety）

操作隐藏字段（Hidden Form Field Manipulation）

操纵参数(Parameter Manipulation)

弱会话 cookie(Weak Session Cookies)

SQL 盲注(Blind SQL Injection)

数字型 SQL 注入(Numeric SQL Injection)

字符串型 SQL 注入(String SQL Injection)

web 服务(Web Services)

Open Authentication 失效(Fail Open Authentication)

危险的 HTML 注释(Dangers of HTML Comments)

……等等

更多请参考“WebGoat 用户及安装指南”。

未来发展

近年来 WebGoat 已经比较稳定了。WebGoat 的问题页面仍然存在问题需要修复，如果您能帮助修复将不胜感激。

展望未来，WebGoat 应利用 OWASP 提供的培训材料的优势，并将其纳入教学计划的材料。现阶段的 WebGoat 已经证明对于安全人员了解攻击类型及如何利用之是有用的，WebGoat 应开始专注于教导安全和开发人员相关的缓解策略、方法。我期望计分卡功能和系统架构能有所完善，用于更好地跟踪课程的完成状况。WebGoat 也可用于企业引进安全编码实践。

马上看看项目路线图，找点任务来帮助我们吧。

下载

WebGoat 可从 Google code 下载区 http://code.google.com/p/webgoat/downloads/list 下载，您也可相应的同步取得当前的 WebGoat 代码树 http://code.google.com/p/webgoat/ 。

发行版

您可从 Sourceforge 的 OWASP 源代码中心

http://sourceforge.net/project/showfiles.php?group_id=64424&package_id=61824

取得 WebGoat 老版本，其中有包含 Java 的版本，也有不包含的版本。安装仅需解压下载到的安装包，并运行其中的开始脚本即可。为便于学习，其中也包含一个可直接部署在您 J2EE 应用服务器上的 war 包。

WebGoat 5.2 标准版

该标准发行版下载解压缩后，即可单击运行。它包含 Java 运行时环境及一个配置好的 Tomcat5.5 服务器。

双击执行 webgoat.bat - Tomcat 命令窗口即会启动，浏览器浏览 http://localhost/WebGoat/attack

WebGoat 5.2 开发版（位于 Sourceforge）

注意：此版旨在提供一个 WebGoat 实验室环境。如果您想开发自己的教学课程，请与 Google code 中的基线同步。

这个开发人员版本除了包含标准版本外，还多了一个已配置的 Eclipse 环境。这个开发人员版本使用也很简单，下载、解压缩然后单击脚本即可。如果您仅仅希望研究有关课程的话，它用起来跟标准版本没有什么区别。然而，如果希望组建实验室，或者在课堂上使用 WebGoat 的话，可以使用 eclipse.bat 脚本来启动一个预配置的 WebGoat 环境。具体的使用说明，请参见自带的_HOW TO create the WebGoat workspace.txt_文件：

1. 将 Eclipse-Workspace.zip 解压至工作目录；
2. 双击 eclipse.bat 文件；
3. 在 Eclipse 右上角的包资源管理器中，右键单击 WebGoat 项目，并刷新；
4. 在 Eclipse 右上角的包资源管理器中，右键单击 Servers 项目，并刷新；
5. 在 Eclipse 底部的服务器视图中，右键单击 localhost 服务器，并启动它；
6. 在浏览器中导航至 http://localhost/WebGoat/attack。
7. 源代码发生的任何变化，都会自动地引起编译操作，保存后会自动重新部署。

如对该版本有任何建议，请发送给 Bruce Mayhew（webgoat@owasp.org）。

演示解决方案

Aung Khant (YGN Ethical Hacker Group)创建了一系列的视频来演示 WebGoat 课程中可能的解决方案，它们可从 http://yehg.net/lab/pr0js/training/webgoat.php 处观看。如需关于 WebGoat 的帮助，可随时找他。

演示视频链接

1. 综合 http://yehg.net/lab/pr0js/training/webgoat.php#General
2. 代码品质 http://yehg.net/lab/pr0js/training/webgoat.php#Code_Quality
3. 协助 http://yehg.net/lab/pr0js/training/webgoat.php#Concurrency
4. 未认证参数 http://yehg.net/lab/pr0js/training/webgoat.php#Unvalidated_Parameters
5. 接入限制缺陷 http://yehg.net/lab/pr0js/training/webgoat.php#Access_Control_Flaws
6. 认证缺陷 http://yehg.net/lab/pr0js/training/webgoat.php#Authentication_Flaws
7. 会话管理缺陷 http://yehg.net/lab/pr0js/training/webgoat.php#Session_Management_Flaws
8. 跨站脚本攻击 http://yehg.net/lab/pr0js/training/webgoat.php#Cross-Site_Scripting_(XSS)
9. 缓冲区溢出 http://yehg.net/lab/pr0js/training/webgoat.php#Buffer_Overflows
10. 注入缺陷 http://yehg.net/lab/pr0js/training/webgoat.php#Injection_Flaws
11. 不安全的存储 http://yehg.net/lab/pr0js/training/webgoat.php#Insecure_Storage
12. 拒绝服务 http://yehg.net/lab/pr0js/training/webgoat.php#Denial_of_Service_(DOS)
13. 错误配置 http://yehg.net/lab/pr0js/training/webgoat.php#Insecure_Configuration_Insecure
14. web 服务 http://yehg.net/lab/pr0js/training/webgoat.php#Web_Services
15. AJAX 安全 http://yehg.net/lab/pr0js/training/webgoat.php#AJAX_Security
16. 挑战 http://yehg.net/lab/pr0js/training/webgoat.php#Challenge

项目贡献者

WebGoat 项目由 Bruce Mayhew.主持，可通过 webgoat@owasp.org 与之联系。WebGoat 通过 Sourceforge 及 Google 进行分发。WebGoat 框架使得添加其它课程变得非常容易。随着新的 web 技术不断涌现，我们正在积极寻找开发人员添加新的课程。如果你有兴趣为项目志愿服务，或有意见、问题及建议，请加入 WebGoat 邮件列表（http://lists.owasp.org/mailman/listinfo/owasp-webgoat）。

感谢 Ounce 实验室（http://www.ouncelabs.com/）允许我能在工作时间运行 WebGoat 项目。

1. Webgoat 用户指南卷首语

1.1 版权与许可

1.2 商标

1.3 贡献者

第 2 版本：匿名贡献者

第 4 版本原始草案：Robert Sullivan(msp.sullivan@gmail.com) 贡献作者：

1.4 文档版本历史

2004 年 1 月 Release v2

2006 年 3 月 Release v4 原始草案

2007 年 1 月 Release v5

2. 概述

WebGoatV5 应用程序是一个用来演示 Web 应用程序中的典型安全漏洞的应用程序，旨在在应用程序安全审计的上下文中系统地、有条理地讲解如何测试和利用这些安全漏洞。一个完整的应用程序安全性评估测试方法在文档 http://www.owasp.org/index.php/OWASP_Testing_Project 中有说明，它同时提供了一份 WebGoat 演示的扩展的说明。包括标准的设计和代码审计等等。WebGoat 课程旨在给大家一个能完成不同级别的 owasp web 应用程序安全测试方法真实的训练环境和实例。

WebGoatV5 应用程序提供了一个典型的应用程序安全评估的测试平台。测试人员在这个在线的应用程序中拥有和普通的客户或者客户端一样的权限和信息：

此应用程序是基于 web 的；

这些都是远程的攻击摸拟，所有有记录的攻击手法都可以从任何可以连接的地方进行连接演示；

这些测试都是基于黑盒的，源代码不提供，但是你可下载和浏览它；

凭证和操作信息是提供的，当然，WebGoat 也常常告诉我们可以显示出来的确定信息往往是你探测不到的。这也同时是在指导测试人员从头到尾完整的进行某项评估过程。

目前 WebGoat 课程计划中提供的内容如下：

HTTP Basics
HTTP Splitting and Cache Poisining
How to Exploit Thread Safety Problems

How to Discover Clues in the HTML
How to Exploit Hidden Fields
How to Exploit Unchecked Email
How to Bypass Client Side JavaScript Validation
How to Force Browser Web Resources
How to Bypass a Role Based Access Control Scheme
How to Bypass a Path Based Access Control Scheme
LAB: Role based Access Control
Using an Access Control Matrix
How to Exploit the Forgot Password Page
How to Spoof an Authentication Cookie
How to Hijack a Session
Basic Authentication
LAB: Cross Site Scripting
How to Perform Stored Cross Site Scripting (XSS)
How to Perform Reflected Cross Site Scripting (XSS)
How to Perform Cross Site Trace Attacks (XSS)
Buffer Overflow (TBD)
HTTPOnly Test
How to Perform Command Injection
How to Perform Parameter Injection

How to Perform Blind SQL Injection
How to Perform Numeric SQL Injection
How to Perform String SQL Injection
How to Perform Log Spoofing
How to Perform XPATH Injection Attacks
LAB: SQL Injection
How to Bypass a Fail Open Authentication Scheme
How to Peform Basic Encoding
Denial of Service from Multiple Logins
How to Create a SOAP Request
How to Perform WSDL Scanning
How to Perform Web Service SAX Injection
How to Perform Web Service SQL Injection
How to Perform DOM Injection Attack
How to Perform XML Injection Attacks
How to Perform JSON Injection Attack
How to Perform Silent Transactions Attacks
How to Add a New Lesson
The Challenge

未来的 WebGoat 新版本将会包含更多的课程和功能。如果你有任何的建议可以帮助改进或者有任何新课程想发布，请联系 bill@owasp.org 并说出你的想法。

3. 目的

在通过 WegGoat 课程体系的测试技术训练，测试人员应该掌握以下技能：

◆理解 web 应用程序中的各种高层次交互过程

◆确定客户信息的可见数据可用于攻击过程中

◆识别和理解能将应用程序暴露在攻击之下的数据和用户交互

◆对这些交互进行测试，以揭露它们的缺陷

◆利用漏洞对应用程序进行攻击，并能演示

4. 必备工具

对于有经验的应用程序安全审计人员来说，可用的辅助工具有很多。就我们这种类型的安全审计来说，最常用的工具就是本地代理和 Web 应用程序爬虫。为了完成全套 WebGoat 课程，Web 代理程序是必不可少的。

4.1 应用程序审计代理

一般的 web 代理通常都能接收、处理和转发客户和服务器之间的 HTTP 和 HTTPS 数据，这样就能让所有的 web 通信流量都流经某个点，以便通过高速缓存或者应用安全策略来监视利用率、提高性能，等等。

应用程序代理工具可用来拦截本地客户端的浏览器和服务器端之间所有的 HTTP 和 HTTPS 通信，它实际上充当了一个可以监视、检查和(最重要地)修改所有的交互的中间人角色。

通过这种工具，审计人员可以准确确定出在客户和服务器之间传递的到底是什么样的数据。此外，它们还可以对这些数据进行分析和修改，从而测试对应用程序的影响。

另外一个重要原因是使用 HTTP 代理是因为 WebGoat 要求基本的身份认证，当自动化工具被用于访问 WebGoat 时，他们可能没有足够的功能来验证 WebGoat。通过使用像 WebScarab 代理，测试人员可以设置基本身份验证，而且可以将认证的凭证透明地传输给 WebGoat 请求。

在 WebGoat 的许多课程中，应用程序审计代理或者具备同等功能的软件都是必不可少的。下列是我们推荐的工具：

◆WebScarab-https://www.owasp.org/index.php/Category:OWASP_WebScarab_Project

◆BurpProxy- http://portswigger.net/

◆ParosProxy - http://parosproxy.org

4.2 应用程序爬虫

所谓爬行一个站点，实际上就是识别和访问网站应用程序内所有预定的页面和链接，并建立本地副本；当然建立副本这一点通常是可选的。然后，我们就可以分析爬行结果，得到应用程序内目标脚本、表单、页面和字段等组成的明细表供后面的测试之用。镜像下来的内容也可以用来分析有关信息，这样做要比人工或者在线分析要快得多了。

下列是我们推荐的工具：

◆WebScarab-https://www.owasp.org/index.php/Category:OWASP_WebScarab_Project

◆BurpSpider - http://portswigger.net

◆ParosProxy - http://parosproxy.org

5. 安装

WebGoat 是一个平台无关的 Web 安全漏洞实验环境，该环境需要 Apache Tomcat 和 JAVA 开发环境的支持。它分别为 Microsoft Windows 和 UN*X 环境提供了相应的安装程序，下面我们将根据操作系统分别加以介绍。

需要注意，从版本 5 开始，这一步可以省略，因为它们自身带有 Java Development Kit 和 Tomcat 5.5。

5.1 安装 Java 和 Tomcat

5.1.1 安装 Java

首先安装 Java，您可以从 http://java.sun.com/downloads/安装和部署合适的版本，最低版本要求为 1.4.1。

5.1.2 安装 Tomcat

然后安装 Tomcat，您可以从 http://tomcat.apache.org/download-55.cgi 安装和部署 Tomcat。

5.2 安装到 Windows 系统

1. 将 WebGoat-OWASP_Standard-5.2.zip 解压至合适的目录中。

2. 若要启动 Tomcat，切换至前面存放解压后的 WebGoat 的目录，然后双击 webgoat.bat 即可。

3.启动浏览器，在地址栏输入 http://localhost/WebGoat/attack 。注意，这个链接地址是区分大小写的，务必确保其中使用的是大写字母 W 和 G。

5.3 安装到 Linux 系统

1. 将 WebGoat-OWASP_Standard-x.x.zip 解压至您的工作目录。

2. 将 webgoat.sh 文件中的第 17、19 和 23 行中的“1.5”改为“1.6”。

3. 因为最新版本运行在一个特权端口上，所以您需要使用下列命令来启/停 WebGoat Tomcat：

(1). 当作为 root 用户运行在 80 端口时，使用：

```
sudo sh webgoat.sh start80

sudo sh webgoat.sh stop
```

(2). 当运行在 8080 端口时，使用：

```
sh webgoat.sh start8080

sh webgoat.sh stop
```

5.4 安装至 OS X（Tiger 10.4+）系统

1. 将 WebGoat-OWASP_Standard-x.x.zip 解压至您的工作目录。

2. 将 webgoat.sh 文件中的第 10 行中的“1.5”改为“1.6”。

3. 因为最新版本运行在一个特权端口上，所以您需要使用下列命令来启/停 WebGoat Tomcat：

(1). 当作为 root 用户运行在 80 端口时，使用：

```
sudo sh webgoat.sh start80

sudo sh webgoat.sh stop
```

(2). 当运行在 8080 端口时，使用：

```
sh webgoat.sh start8080

sh webgoat.sh stop
```

5.5 安装至 FreeBSD 系统

1.从 Ports Collection 安装来安装 Tomcat 和 Java ：

```
cd /usr/ports/www/tomcat55

sudo make install
```

2. 安装 Java JDK 的时候，可能需要手工方式进行下载，届时系统会给出详细的提示。

3. 将 WebGoat-OWASP_Standard-x.x.zip 解压至您的工作目录。

4. 将 webgoat.sh 文件中的第 17、19 和 23 行中的“1.5”改为“1.6”。

5. 因为最新版本运行在一个特权端口上，所以您需要使用下列命令来启/停 WebGoat Tomcat：

(1).当作为 root 用户运行在 80 端口时，使用：

```
sudo sh webgoat.sh start80

sudo sh webgoat.sh stop
```

(2). 当运行在 8080 端口时，使用：

```
sh webgoat.sh start8080

sh webgoat.sh stop
```

5.6 运行

1. 启动浏览器，并在地址栏输入 http://localhost/WebGoat/attack，注意这里使用的大写的字母 W 和 G。

2. 登录时，用户帐号使用 guest，密码为 guest。

5.7 编译

如果你仅仅只是为了运行 WebGoat 的话，请跳过这一节。

WebGoat 可以用 eclipse WTP 1.5.x 来编译。请到 http://webgoat.googlecode.com/svn/trunk/webgoat/README.txt 去阅读相关的步骤来编译它，或者你也可以阅读下面的开发版安装方法：

WebGoat 开发版安装方法：

WebGoat 5.2 Developer 版（位于 SourceForge 网站），注意：这个版本旨在提供一个 WebGoat 实验室环境。如果您想开发自己的教学课程，请与 Google code 站点上的基线同步。

这个开发人员版本除了包含标准版本外，还多了一个已配置的 Eclipse 环境。这个开发人员版本使用也会简单，下载、解压缩然后单击脚本即可。如果您仅仅希望研究有关课程的话，它用起来跟标准版本没有什么区别。然而，如果希望组建实验室，或者在课堂上使用 WebGoat 的话，可以使用 eclipse.bat 脚本来启动一个预配置的 WebGoat 环境。具体的使用说明，请参见自带的_HOW TO create the WebGoat workspace.txt_文件。

1.将 Eclipse-Workspace.zip 抽取至工作目录

2.双击 eclipse.bat 文件

3.在 Eclipse 右上角的包资源管理器中，右键单击 WebGoat 项目，并刷新

4.在 Eclipse 右上角的包资源管理器中，右键单击 Servers 项目，并刷新

5.在 Eclipse 底部的服务器视图中，右键单击 localhost 服务器，并启动它

6.在浏览器中导航至 http://localhost/WebGoat/attack。

7.源代码发生的任何变化，都会自动地引起编译操作，保存后会自动重新部署。

5.8 安装 WAR 文件到已安装的 Tomcat 服务

这个版本将假定已经预先安装了 WebGoat Standard 版本，或者主机已经安装了 java 1.5（或更高版本）和 tomcat 5.5。如果您尚未安装 Standard 版本，那么就需要修改 tomcat/conf/tomcat-users.xml 文件来添加 WebGoat 用户，具体请参阅 http://code.google.com/p/webgoat/wiki/FAQ。

1.从 WebGoat Downloads 链接下载 WebGoat-OWASP_WAR-X.X.zip。

2.如果 Tomcat 正在运行的话，请先将其关闭——只需关闭 Tomcat 窗口即可。

3.将 war 文件拷贝至 WebGoat-X.X\tomcat\webapps\webgoat.war

4.删除现有的 WebGoat-X.X\tomcat\webapps\webgoat 目录

(1).这会导致所有的课程状态被丢失

(2).若要保存课程状态，请保留 webapps\webgoat\users 文件夹的副本

(3).重新启动 WebGoat 之后恢复这个用户目录

5.切换至 WebGoat-X.X 目录

6.双击 webgoat.bat 文件，这时 Tomcat 窗口就会启动。

7.在浏览器中导航至 http://localhost/WebGoat/attack。

6. WebGoat 入门

开始使用 WebGoat 之前，必须先启动 Tomcat，这可以通过 Tomcat 的/bin 目录中的 script/bat 程序 startup 来完成。此外，要想正常使用 WebGoat，它必须具备作为服务器运行所需的权限，并允许一些不常见的 web 行为。注意 WebGoat 运行时，安全漏洞会使主机很容易遭到攻击。

如果机器连接到了互联网，那么就应该将其断开。

运行的个人防火墙可能会阻止 WebGoat 的正常使用。所以，运行 WebGoat 时最好禁用所有的个人防火墙。

我们可以使用浏览器浏览 localhost 的 80 端口来访问 Tomcat 服务器，如 http://127.0.0.1。

WebGoat 位于 WebGoat 目录，你还可以在这里找到课程 http://127.0.0.1/WebGoat/attack。

WebGoat 的应用程序强制采用基于角色的安全机制。登录对话请求会要求输入身份凭证。可以使用 userid=guest,password=guest 登录。

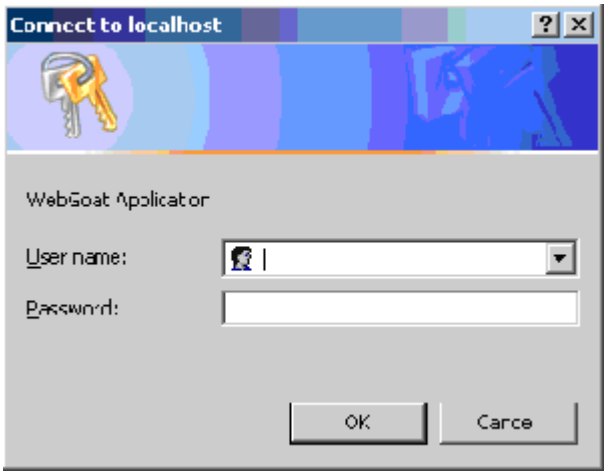


图 1　登录界面

成功登录之后，Tomcat 服务器将显示 WebGoat 的欢迎页面。

图 2 欢迎页面

7. 课程计划

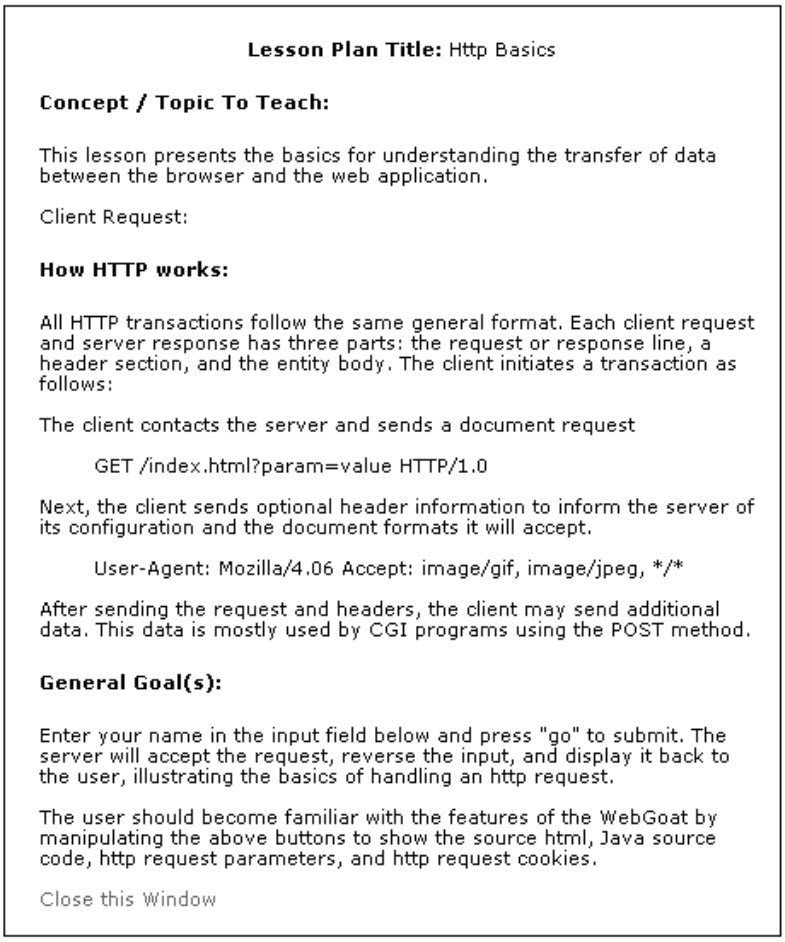
课程计划包含 WebGoat5.0(1/31/07)：

General	HTTP Basics
	HTTP Splitting and Cache Poisining
	How to Exploit Thread Safety Problems
	How to add a new WebGoat lesson
Code Quality	How to Discover Clues in the HTML
Unvalidated Parameters	How to Exploit Hidden Fields

	How to Exploit Unchecked Email
	How to Bypass Client Side JavaScript Validation
Broken Access Control	Using an Access Control Matrix
	How to Bypass a Path Based Access Control Scheme
	How to Perform Cross Site Request Forgery (CSRF)
	LAB: Role based Access Control
	Remote Admin Access
Broken Authentication	Forgot Password
	How to Spoof an Authentication Cookie
	How to Hijack a Session
	Basic Authentication
Cross Site Scripting (Xss)	LAB: Cross Site Scripting
	How to Perform Stored Cross Site Scripting (XSS)
	How to Perform Reflected Cross Site Scripting (XSS)
	HTTPOnly Test
	How to Perform Cross Site Tracing (XST) Attacks
Buffer Overflows	Buffer Overflow
Injection Flaws	How to Perform Command Injection
	How to Perform Blind SQL Injection
	How to Perform Numeric SQL Injection
	How to Perform Log Spoofing

	How to Perform XPATH Injection
	How to Perform String SQL Injection
	LAB: SQL Injection
	How to Use Database Backdoors
Improper Error Handling	How to Bypass a Fail Open Authentication Scheme
Insecure Storage	Encoding Basics
Denial of Service	Denial of Service From Multiple Logins
Insecure Configuration Management	Forced Browsing
Web Services	How to Create a SOAP Request
	WSDL Scanning
	Web Service SAX Injection
	Web Service SQL Injection
AJAX Security	DOM Injection
	XML Injection
	JSON Injection
	Silent Transactions Attacks
Challenge	The Challenge

每个 WebGoat 里的课程都提供了课程概述和目标。你可以点击显示课程计划按钮。



Lesson Plan Title: Http Basics

Concept / Topic To Teach:

This lesson presents the basics for understanding the transfer of data between the browser and the web application.

Client Request:

How HTTP works:

All HTTP transactions follow the same general format. Each client request and server response has three parts: the request or response line, a header section, and the entity body. The client initiates a transaction as follows:

The client contacts the server and sends a document request

GET /index.html?param=value HTTP/1.0

Next, the client sends optional header information to inform the server of its configuration and the document formats it will accept.

User-Agent: Mozilla/4.06 Accept: image/gif, image/jpeg, */*

After sending the request and headers, the client may send additional data. This data is mostly used by CGI programs using the POST method.

General Goal(s):

Enter your name in the input field below and press "go" to submit. The server will accept the request, reverse the input, and display it back to the user, illustrating the basics of handling an http request.

The user should become familiar with the features of the WebGoat by manipulating the above buttons to show the source html, Java source code, http request parameters, and http request cookies.

Close this Window

这些课程计划涵盖了目标应用程序的所有操作，以及感兴趣尝试的领域，包括有关的安全评估和攻击类型。

8. 基本操作

我们知道，在应用程序安全评估的每一个阶段，都需要对目标的运作机制有深入的了解。这通常包括：

考察客户端内容，如 HTML 和 script。

分析客户端和服务器之间的通讯。

检查 cookie 及其他本地数据。

浏览器已经使得查看 HTML 源代码变得非常轻松，而 WebGoat 又增加了多种操作，包括显示参数、显示 HTML、显示 Cookies 和显示 Java 等。

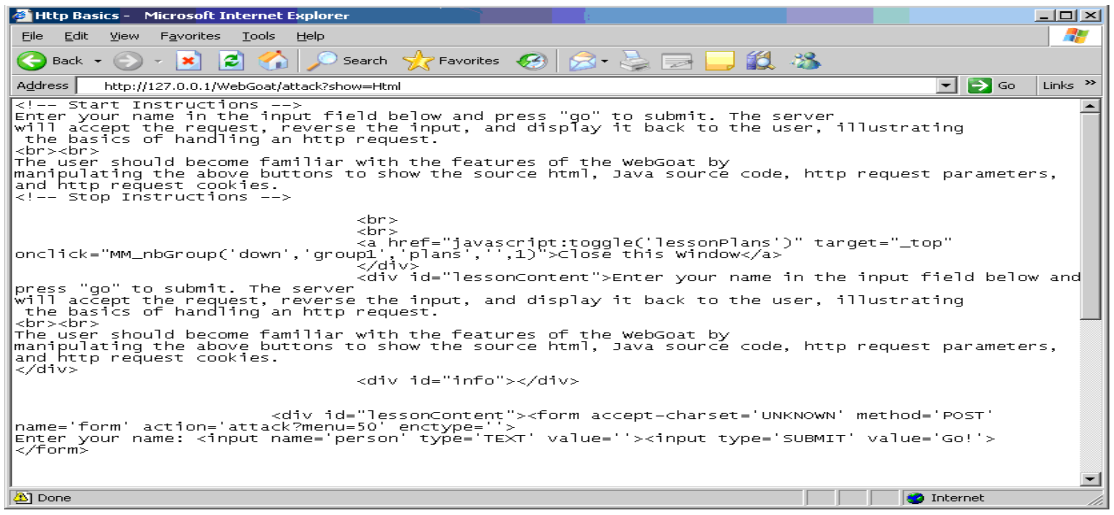


图 3 WebGoat 显示 HTML 源代码

通常情况下，浏览器提供了查看 HTML 源代码的功能。对于微软公司的 Internet Explorer 浏览器，可以通过“查看”菜单下的“源文件”选项来查看 HTML 源代码。对于 Firefox 浏览器来说，查看页面源码的功能同样位于“查看”菜单下的“页面源代码”下。WebGoat 显示 HTML 的功能仅仅展示当前的 HTML 代码，而不包括侧边栏和上边栏对应的 HTML 代码。

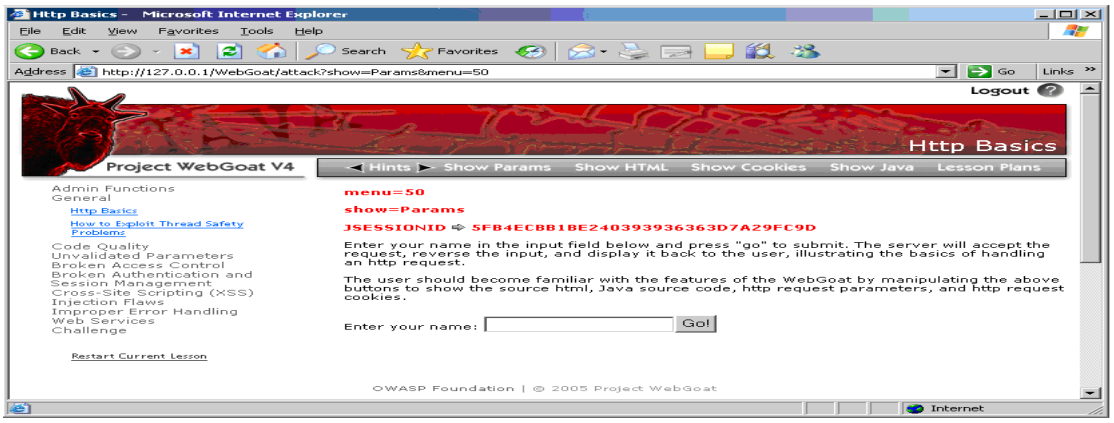


图 4 显示 HTML 源代码

这里，参数和 cookie 显示为红色。

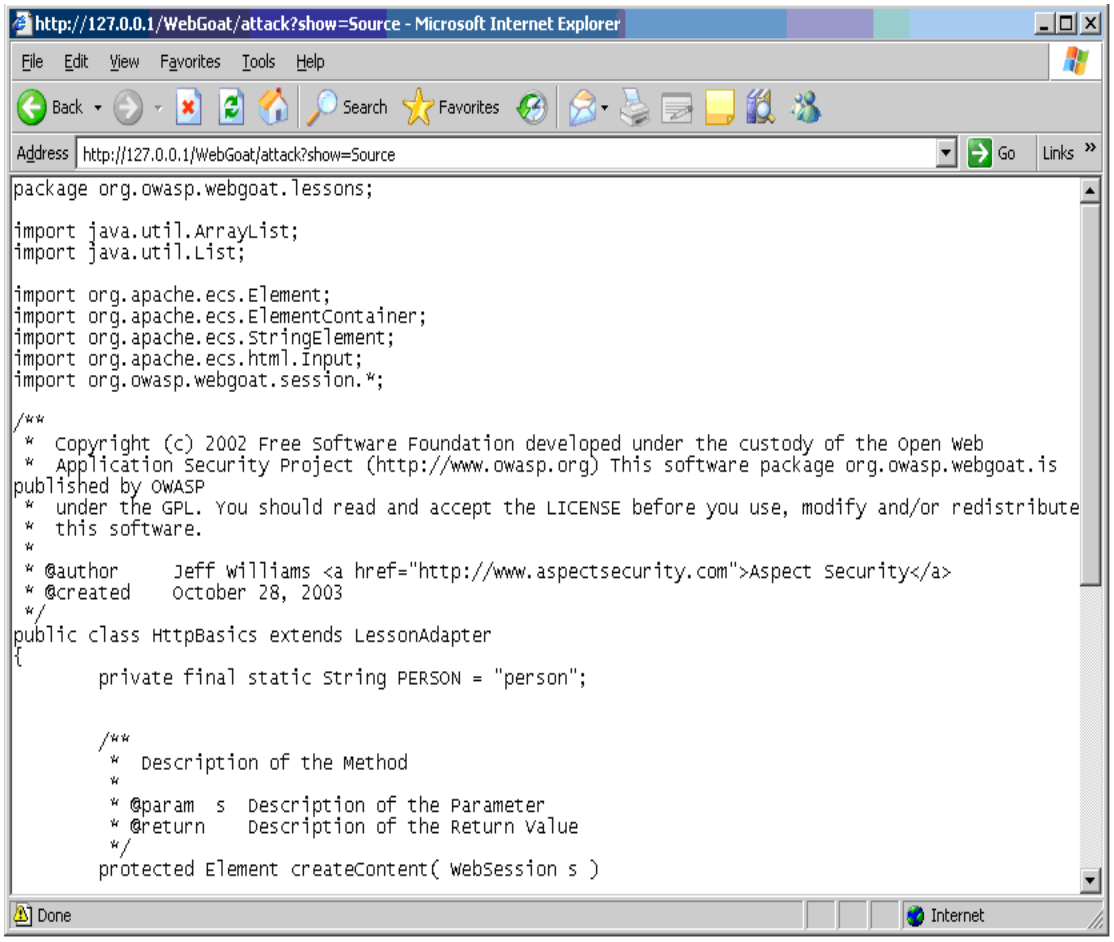


图 5 显示参数 Cookies

这里显示 Java 操作会弹出一个包含源代码的新窗口。

9. 利用代理

要想充分挖掘 WebGoat 的各种功能，我们还需要借助审计人员常用的应用程序审计代理工具。代理工具可以帮助我们进行更深入的分析，并能修改客户端-服务器的交互和传输过程中的数据。虽然不同的工具，它的使用和配置方法也不相同，但基本概念是一致的：

应用程序审计代理必须位于客户端的浏览器和远程服务器之间。

它应该允许显示和修改传输中的所有 HTTP 数据。

该工具通常会直接插入浏览器，或者在另外一个本机端口进行侦听。当代理程序直接插入浏览器的时候，需要在浏览器中键入一个特殊的 URL。当该工具侦听端口时，则需要对浏览器进行相应的配置，方可正常使用该工具。在微软公司的 Internet Explorer 中，可以通过工具菜单完成配置工作，如下所示：

选择工具菜单中的“Internet 选项”菜单项。

选择“连接”选项卡。

单击选项卡下方的“局域网设置…”按钮。

在局域网设置对话框中，选中为 LAN 使用代理服务器的复选框。

勾掉“对本地地址不使用代理服务器”框。

输入代理工具将要侦听的地址和端口。对于 WebScarab 而言，其默认侦听端口是 8008。

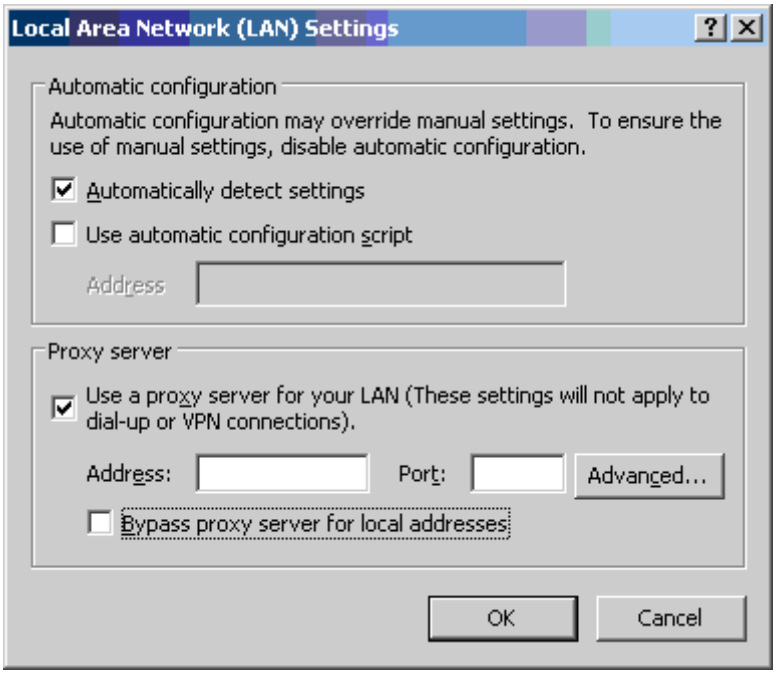


图 6 局域网设置

现在，每当从客户端的浏览器接收或者发送数据时，我们都能够通过拦截、分析和修改这些 HTTP 请求，来测试目标应用程序的安全性。

设置和运行 WebScarab 的教程请看这里：WebScarab 教程

借助这类代理，审计人员可以获得多种能力，包括：

- 所有的 GET/POST 参数都可以修改，不管它是不是隐藏的。
- 所有的 Cookie，不管是持久性还是非持久性的 Cookie，当它们进入和离开浏览器时，我们都可以对其进行修改。
- 所有的客户端验证都可以绕过，因为参数在发送给服务器之前可以立即进行修改。
- 能够暴露高速缓存的数据，以便于分析。
- 能够暴露出 Server 及其他报头，这对于获取远程 web 服务器类型和所用的应用程序-服务器技术非常有利。

10.起航

剩下的工作是开始 WebGoat lesson

最后一点：

如果你没有完整的手册，你也可以自己去收集一些帮助信息，这都能帮助你完成课程。不要太急于求成，应用测试靠的是 10%的技术和 90%的横向思维。如果你通过自己的努力去战胜课程设置的难题，你会学习并且掌握到更多的东西。当然，这一切要经过大量的尝试，一次次的失败，终于一个闪念把你引向成功。这个过程中，你可以去责怪 Goat，但是你不能去依赖他人。

11.怎样写一个新的 WebGoat lesson

你要做的是实现 LessonAdapter 的抽象方法。

WebGoat 使用了 Jakarta Element Construction Set，你可以去研读一下 ECS 的 API：

http://jakarta.apache.org/site/downloads/downloads_ecs.cgi。

WebGoat 使用了 WTP，更多 Eclipse WTP 的信息 http://www.eclipse.org/webtools/。

步骤 1：搭建 framework

NewLesson.java 类的源码。

步骤 2：实现 creatContent

创建一个 lesson 的内容相当简单，包含两个主要部分：

1. 处理输入，即来自于用户的最新请求
2. 为用户生成下一个交互页

这些都要通过 createContent 方法实现。每个 lesson 最好在一个单页上完成，因此设计的时候，把 lesson 的功能设计在一个页面上是很重要的。

实例 createContent 方法

下面是一个不错的通用范例 createContent 方法：

```
// define a constant for the filed name
Private static final String INPUT = "input";
Protected Element createContent(WebSession s)
{
 ElementContainer ec = new ElementContainer();
 try
 {
    // get some input from the user
    // see ParameterParser for details
    String userInput = s.getParser().getStringParameter(INPUT, "");
    // do something with the input
    // -- SQL query?
    // -- Runtime.exec?
    // -- Some other dangerous thing
    // generate some output - a string and an input field
    ec.addElement(new StringElement("Enter a string: "));
    ec.addElement( new Input(Input.TEXT, INPUT, userinput));
 }
 catch (Exception e)
 {
    s.setMessage("Error generating " + this.getClass().getName());
    e.printStackTrace();
 }
 return(ec);
}
```

ECS 功能十分强大。参看编码课程的示例，关于如何使用它创建一个行列输出的表格。

步骤 3：实现其他方法

要想让一个 lesson 有更丰富的功能，LessonAdapter 类需要更多的方法。这些方法可以导航课程或者显示课程信息给用户。所有的方法都应该相当简单，只需要花几分钟的时间就能实现。

其他 LessonAdapter 方法

	Method	*'Description'*
1	getHints	Return hints to the framework one at a time
2	getCredits	Return credits to the framework for display
3	getInstructions	This method will load the instructions HTML file from lesson_plans directory if you create one.
4	getRanking	Sets the order of the lessons within a category. The lowest ranked lesson appears at the top.
5	getTitle	The title is rendered as HTML

```
protected List getHints()
{
    // Hints will be returned to the user in the order they
    // appear below. The user must click on "next hint"
    // before the hint will be didplayed.
    List hints = new ArrayList();
    hints.add( "There are no hints defined." );
    return hints;
}
public Element getCredits()
{
    return new StringElement("");
}
/*
 * Gets the ranking attribute of the LessonAdapter object.
        * The ranking denotes the order in which
        * the menu item will appear in menu list for each category.
        * lowest number will appear as the first lesson.
 *
 * @return    The ranking value
 */
public Integer getRanking()
{
       return new Integer(10);
}
/**
 *  Fill in a descriptive title for this lesson.
        *  This will appear above the control area at the
        *  top of the page. This field will be rendered as html.
 *
 * @return    The title value
 */
public String getTitle()
{
       return "Untitled Lesson " + getScreenId();
}
```

步骤 4：创建和测试<v4 中有变动>

实现一个新的 lesson 后，ant 被用来创建、部署一个新的 web 应用。首先，移除 webapps 目录下的 webgoat.war 文件以及 webgoat 目录。然后，进入 webgoat 目录下输入：

```
ant install
```

用来编译新 lesson 并且“安装”到 tomcat 路径。Lesson 只需要一次安装，如果 web 应用和其他的测试有变动，可以输入：

```
ant reload
```

步骤 5：回馈

如果你完成了一个 lesson 并且认为在 web 应用安全方面有所帮助，请提供给 WegGoat 应用的维护人员。