



WEBGOAT and the Pantera Web Assessment Studio Project

Philippe Bogaerts

**OWASP
Belgium
Chapter**

Copyright © 2007 - The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License.

The OWASP Foundation
<http://www.owasp.org/>

Introduction

■ During the day

- ▶ Coming soon ... I hope ☺

■ During the night

- ▶ Independent trainer and consultant
- ▶ Trying to acquire a good understanding of
 - network security
 - web application, web services and XML security
 - Pen-testing

<mailto:philippe.bogaerts@radarhack.com>

<http://www.radarhack.com>



Why am I here ?

- A fascination for security...
- I like learning and exploring new things...
- Continuous education and awareness today is a must and must be kept big fun...

- ... and this resulted in writing a paper called

"Getting started with OWASP WebGoat4 and SOAPUI."

(The paper is available at <http://www.radarhack.com>)

... and thanks to *Erwin Geirnaert* from <http://www.zionsecurity.com> for reviewing the paper.

ZION SECURITY
SECURING YOUR BUSINESS VALUE



What is the paper about ?

- Explain in a **simple and easy way** what **SOAP and web services** are about.
- A unique opportunity to use **WebGoat 4.0** for what it is intended to do: **education and awareness**
- The paper is about how a web service can be exploited via simple and free available invocation tools.



Part 1: WebGoat



WebGoat

- **WebGoat** is a **deliberately insecure** J2EE web application maintained by [OWASP](#)
- Designed to teach web application security
- ... but also useful to test security products
 - ▶ IPS, Firewalls, Web Application Firewalls ...
 - ... against OWASP top 10 promise
 - ... against XML and AJAX security threats
- Who already played around with WebGoat ?



WebGoat versions

- Release Quality Projects

- Current stable version: 4.0

- ▶ http://www.owasp.org/index.php/Category:OWASP_WebGoat_Project

- A promising version 5.0 will be available 01/2007.

- ▶ Release candidate 1 is available since 17/01/2007



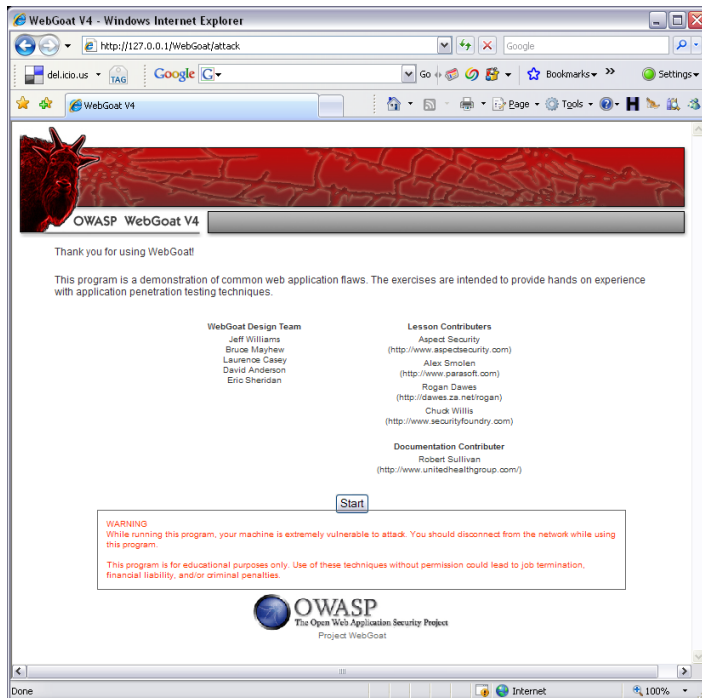
Installing WebGoat

- Download available via OWASP project pages
- Windows and Unix/Linux versions
- Today we are using
 - Windows_WebGoat-4.0_Release.zip*
 - Windows_WebGoat-5.0-RC1_Release.zip*
- Just unzip the archive and click *webgoat.bat*
 - ▶ *Some pitfalls*
 - *Make sure other web servers are stopped*
 - *Skype for some reason dares to use port 80*
 - *Verify with "netstat -an" port 80 is not used*



Connecting the first time

- http://webgoat_server/WebGoat/attack
- login with usn:guest and pwd:guest



Configuration tuning

- ...Windows_WebGoat-4.0_Release\tomcat\conf\server.xml
 - ▶ Port numbers of the web server
- ...Windows_WebGoat-4.0_Release\tomcat\conf\tomcat-users.xml
 - ▶ Tomcat usernames, passwords and role



WebGoat V4

■ A set of lessons and exercises to learn about basic and advanced web application security issues.

- ▶ Coverage OWASP TOP 10
- ▶ ... and more



Admin Functions
General
Code Quality

[How to Discover Clues in the HTML](#)

Unvalidated Parameters
Broken Access Control
Broken Authentication and Session Management
Cross-Site Scripting (XSS)
Buffer Overflows
Injection Flaws
Improper Error Handling
Insecure Storage
Denial of Service
Insecure Configuration Management
Web Services
Challenge



WebGoat is a training tool

■ Tools to assist

▶ Hints

- Starting tips up to the solutions of the problem
- Scroll through the hints.

▶ Show Cookies

▶ Show Java

▶ Show Params

▶ Report Card

[Restart this Lesson](#)

You can view the HTML source by selecting 'view source' in the browser menu.

menu=70

show=Cookies

JSESSIONID ➡ F43F1B58E8F2DF328C83B607092F9DF3

Below is an example of a forms based authentication form. Look for clues to help you log in.

Sign In

Please sign in to your account. See the OWASP admin if you do not have an account.

*Required
Fields

*User Name:

*Password:

Login

Sponsored by **ASPECT** SECURITY
Application Security Specialists



Example 1

■ Code Quality

- ▶ Look in the source code
- ▶ Use WebScarab !!!
 - Fragments module

Search for the word **HIDDEN**, look at URLs, look for comments.

JSESSIONID ➡ **F43F1B58E8F2DF328C83B607092F9DF3**

Below is an example of a forms based authentication form. Look for clues to help you log in.

Sign In

Please sign in to your account. See the OWASP admin if you do not have an account.

*Required
Fields

*User Name:


*Password:

Login



Example 2

■ Stored XSS



OWASP WebGoat V4

Logout ?

How to Perform Stored Cross Site Scripting (XSS)

< Hints > Show Params Show Cookies Show Java Lesson Plans

Admin Functions

General

Code Quality

Unvalidated Parameters

Broken Access Control

Broken Authentication and Session Management

Cross-Site Scripting (XSS)

Buffer Overflows

Injection Flaws

Improper Error Handling

Insecure Storage

Denial of Service

Insecure Configuration Management

Web Services

Challenge

Restart this Lesson

It is always a good practice to scrub all inputs, especially those inputs that will later be used as parameters to OS commands, scripts, and database queries. It is particularly important for content that will be permanently stored somewhere. Users should not be able to create message content that could cause another user to load an undesirable page or undesirable content when the user's message is retrieved.

* Congratulations. You have successfully completed this lesson

Title:

Message:

Submit

The page at http://127.0.0.1 says:
! hackerertje hack
OK


Message Contents For: owasp
Title: owasp
Message:



Example 3

■ Exploiting Hidden Fields

► Web Developer plug-in Firefox

**OWASP WebGoat V4**

Logout ?

How to Exploit Hidden Fields

◀ Hints ▶ Show Params Show Cookies Show Java Lesson Plans

Admin Functions
General
Code Quality
Unvalidated Parameters
[How to Exploit Hidden Fields](#)
[How to Exploit Unchecked Email](#)
[How to Bypass Client Side JavaScript Validation](#)
Broken Access Control
Broken Authentication and Session Management
Cross-Site Scripting (XSS)
Buffer Overflows
Injection Flaws
Improper Error Handling
Insecure Storage
Denial of Service
Insecure Configuration Management
Web Services
Challenge

Restart this Lesson

Try to purchase the HDTV for less than the purchase price, if you have not done so already.

<form enctype="" method="post" name="form">

Shopping Cart

Shopping Cart Items -- To Buy Now	Price:	Quantity:	Total
56 inch HDTV (model KTV-551)	2999.99	<input type="text" value="1"/>	\$2999.99

The total charged to your credit card: \$2999.99

<input name="Price" value="1"/>

Sponsored by **ASPECT SECURITY**
Application Security Specialists

OWASP Foundation | Project WebGoat



Example 4

■ Exploiting Web Services with SQL Injection

► WebScarab

The screenshot shows the WebScarab application running in Mozilla Firefox. The interface displays a SOAP message being sent to a service named 'WsSqlInjectionService'. The message is a 'getCreditCardRequest' with a single node 'id' of type 'string' and value '555 or 1=1'. The response is a 'getCreditCardResponse' with a 'getCreditCardReturn' node containing the value '987654321'.

Web Service SQL Injection - Mozilla Firefox

File Edit View History Bookmarks Tools Help

WebScarab

File View Tools Help

Getting Started Disable Co

WebServices Spider Extensions SessionID Analysis Scripted Fragments Fuzzer Compare Search

Summary Messages Proxy Manual Request

127.0.0.1 WSDL: 1 - GET http://127.0.0.1:80/WebGoat/services/WsSqlInjection 200 OK

WSDL URL: http://127.0.0.1/WebGoat/services/WsSqlInjection?WSDL Load

Service: WsSqlInjectionService

Operation: getCreditCard

Node	Type	Nillable	Value
getCreditCardRequest			
id	string		555 or 1=1

Parsed Raw

Version	Status	Message
HTTP/1.1	200	OK

Header	Value
Server	Apache-Coyote/1.1
Content-Type	text/xml; charset=utf-8
Date	Fri, 19 Jan 2007 20:32:42 GMT
Connection	close

XML Text Hex

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <xsi:type="xsd:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" />
  </soapenv:Body>
</soapenv:Envelope>
```

Done

Headers Execute

Used 4.24 of 63.56MB



WebGoat V5 (rc1)

■ What new ?

- ▶ More XSS
 - Forced Browsing
 - How to Perform CSRF
- ▶ More on SQL Injection
 - Blind SQL Injection
 - XPATH Injection
- ▶ Web Services
 - SAX parser injection
- ▶ AJAX security lessons
- ▶ ... and much more



Admin Functions
General
Code Quality
Unvalidated Parameters
Broken Access Control
Broken Authentication and
Session Management
Cross-Site Scripting (XSS)
Buffer Overflows
Injection Flaws
Improper Error Handling
Insecure Storage
Denial of Service
Insecure Configuration
Management
Web Services
AJAX Security
New Lessons
Challenge



Example 5

■ Web Service SAX injection

t V5

◀ Hints ▶ Show Params Show Cookies Show Java Lesson Plans

Restart this Lesson

Web Services communicate through the use of SOAP requests. These requests are submitted to a web service in an attempt to execute a function listed in the web service definition language (WSDL).

General Goal(s):

Some web interfaces make use of Web Services in the background. If the frontend relies on the web service for all input validation, it may be possible to corrupt the XML that the web interface sends.

In this exercise, try to change the password for a user other than 101.

Please change your password:

Go!

```
<?xml version='1.0' encoding='UTF-8'?>
<wsns0:Envelope
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:wsns0='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns:wsnsl='http://lessons.webgoat.owasp.org'>
  <wsns0:Body>
    <wsnsl:changePassword>
      <id xsi:type='xsd:int'>101</id>
      <password xsi:type='xsd:string'>[password]</password>
    </wsnsl:changePassword>
  </wsns0:Body>
</wsns0:Envelope>
```



Part 2: Pantera Web Assessment Studio Project



Pantera WASP, what is it ?

- “The primary goal of Pantera is to **combine automated capabilities with complete manual testing** to get the best penetration testing results.”
- penetration testing facilitation
 - ▶ Project management
 - ▶ Data mining
- Beta Status Project



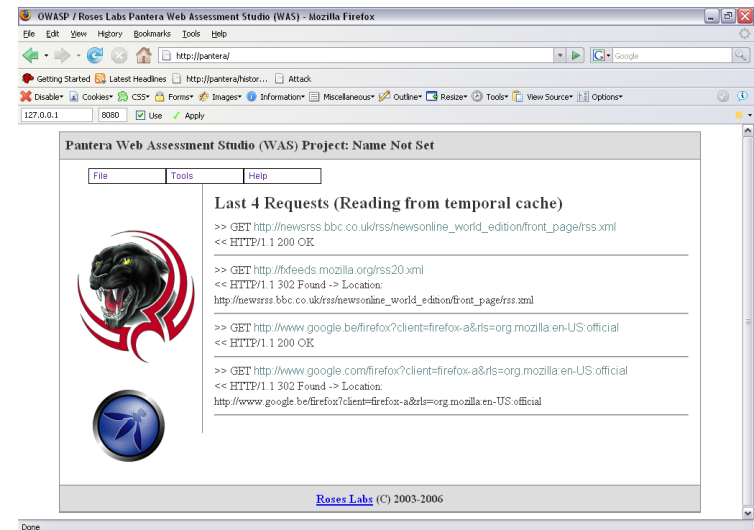
Pantera

■ (local) proxy

- ▶ monitors and intercepts web traffic
- ▶ Traffic is analyzed/modified by **Pantera Passive Analyzer Plugins (PPA)**

■ Web based management interface

- ▶ Project management
- ▶ Notes



How to install ?

- Pantera is available via the OWASP project pages on <http://www.owasp.org>
 - ▶ Current version 0.1.2
- Install the **correct versions** of the required software.
 - Python, MySQL, pyOpenSSL, Formbuild...



Install problems

■ Installation is difficult, but it works and is well described

- Read the INSTALL.TXT
- Very good step by step installation instructions

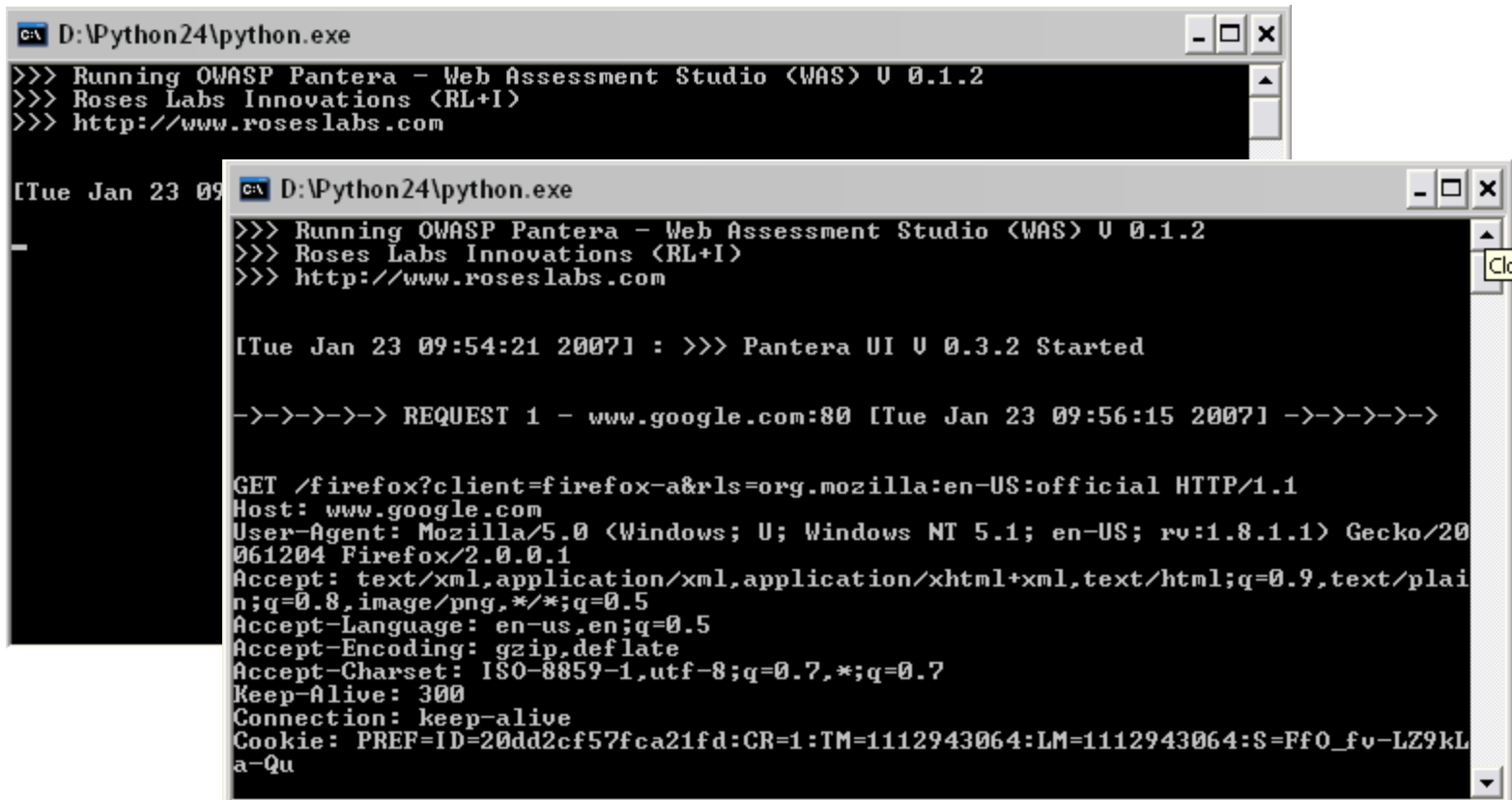
■ Problems ?

- ▶ Contact the mailing list
 - VERY good response.
 - Subscribe via the project page



Starting Pantera

■ python pantera.py



```
D:\Python24\python.exe
>>> Running OWASP Pantera - Web Assessment Studio (WAS) V 0.1.2
>>> Roses Labs Innovations (RL+I)
>>> http://www.roseslabs.com

[Tue Jan 23 09:54:21 2007] : >>> Pantera UI V 0.3.2 Started

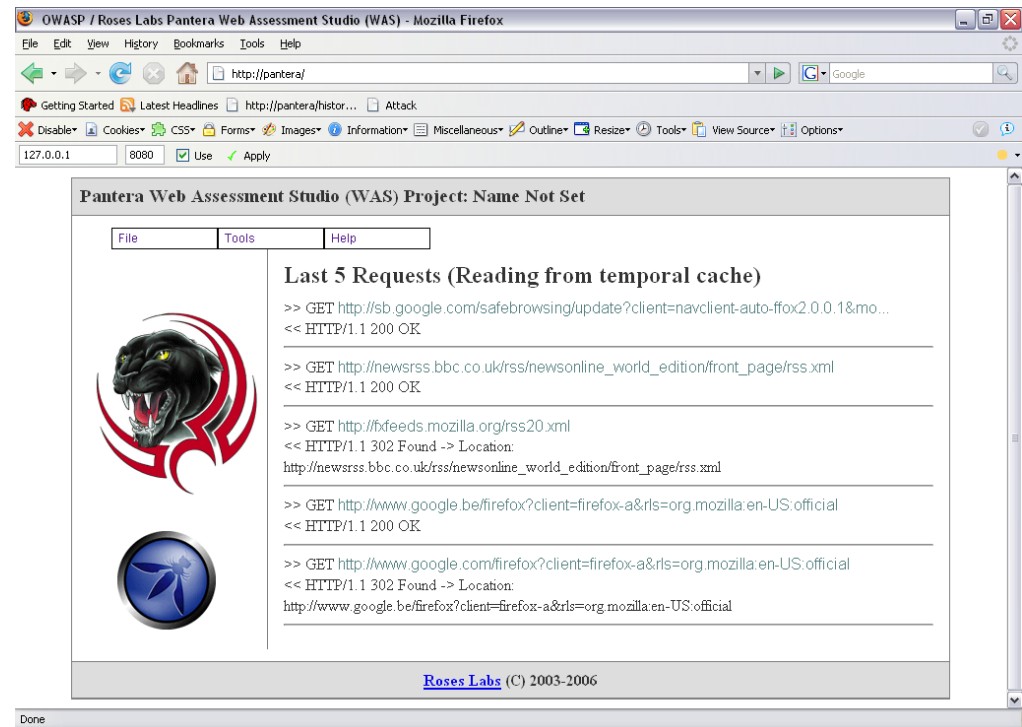
->->->-> REQUEST 1 - www.google.com:80 [Tue Jan 23 09:56:15 2007] ->->->->

GET /firefox?client=firefox-a&rls=org.mozilla:en-US:official HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.1) Gecko/20061204 Firefox/2.0.0.1
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: PREF=ID=20dd2cf57fca21fd:CR=1:TM=1112943064:LM=1112943064:S=Ff0_fv-LZ9kLa-Qu
```



Managing Pantera

- Point your browser to the Pantera proxy instance at 127.0.0.1:8080
- Browse to <http://pantera>



Create a project

The screenshot shows the OWASP Pantera Web Assessment Studio (WAS) running in a Mozilla Firefox browser. The browser window has a title bar that reads "OWASP / Roses Labs Pantera Web Assessment Studio (WAS) - Mozilla Firefox". The address bar shows the URL "http://pantera/project_link". The browser's status bar at the bottom displays "Done".

The application interface is divided into several sections:

- Left Sidebar:** Contains a menu with the following items: File, Configuration, Project Management, Close Project, Domain Management, PPA Management, Import Export, Project Notes, Cleaner, Error Console, and Pantera Status. A red arrow points to the "Project Management" item.
- Top Bar:** Includes a "Pantera Web Ass" label, a "127.0.0.1" address field, a "8080" port field, and a "Use" checkbox.
- Project Management Section:** Features a navigation bar with "Home", "File", "Tools", and "Help" tabs. Below this, there are two main sections:
 - New Project:** Includes a "Project Name:" text input field and a "Submit" button.
 - Available Projects:** Includes a "Select Project:" dropdown menu currently showing "radarhack", radio buttons for "Open" (selected) and "Delete", and a "Submit" button.
- Footer:** A blue circular logo with a stylized insect is located at the bottom left. The bottom right corner of the application area contains the text "Roses Labs (C) 2003-2006".

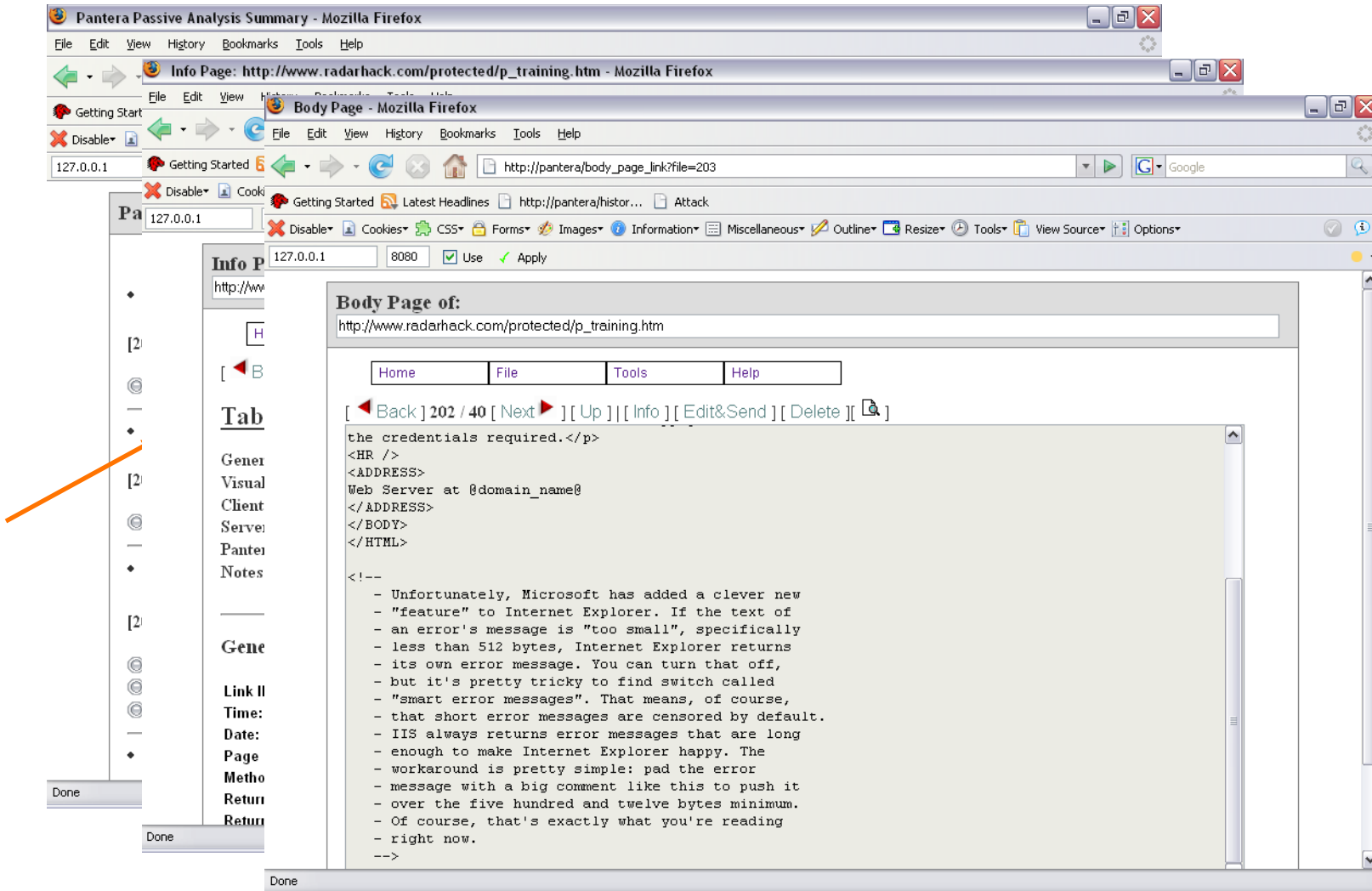


PPA plug-in

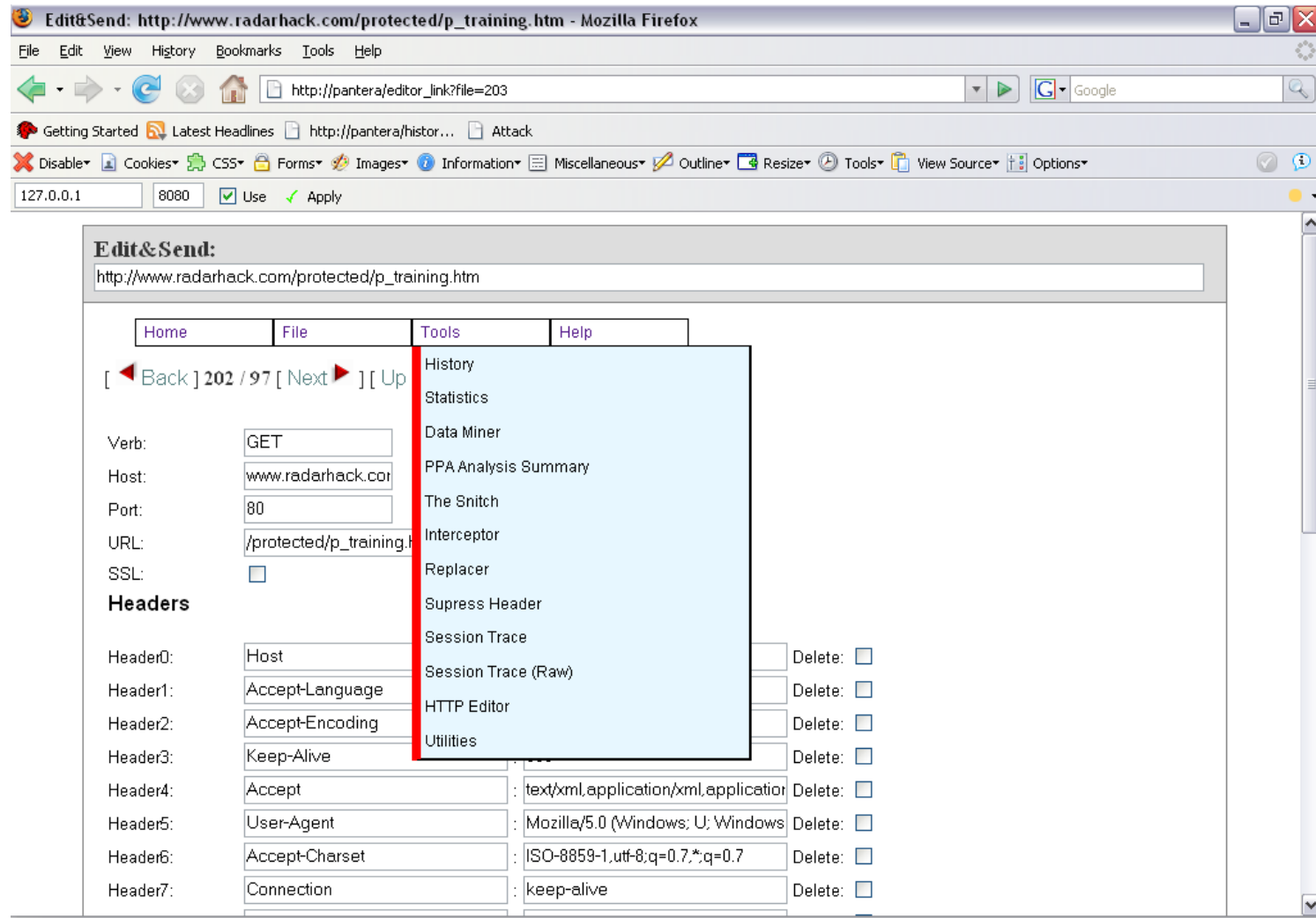
- PPA plug-ins are used to analyze PASSIVELY all web traffic for
 - ▶ Authentication
 - ▶ Vulnerabilities
 - ▶ Comments
 - ▶ ...
- File -> Configuration
- Results are shown in
 - Tools -> PPA Analysis summary



Pantera Passive Analysis Summary



Tools



Tools

- Stats and Data Mining
- Interceptor, Replacer, Suppress Headers
- Session Trace and HTTP Editor
- Utilities
 - ▶ En/decode, Hashing...
- Demo



Thank You

