Exploring the WebScarab Intercept Proxy

Chris Shuster  
University of Colorado at Colorado Springs  
Colorado Springs, CO  
cshuster@uccs.edu

# Abstract

Major threats on the Internet are increasingly based around web hacking. Web hacking doesn't require a server's operating system and installed software to be compromised to achieve the goal of the hacker. One of the major tools used by both white hat and black hat hackers to perform web hacking is an intercept proxy. An intercept proxy allows a request from a browser to be intercepted and altered in order to exploit inherent weaknesses in the server and web technologies. One such intercept proxy is WebScarab created by the Open Web Application Security Project. As such the focus of this paper is to explore and understand the capabilities and uses of the WebScarab intercept proxy.

# Overview

This section will provide an overview of intercept proxies, the attacks using intercept proxies and any additional software used. Additionally this section will give an overview of the constraints of this research report.

## Hacking

Hacking is often associated with criminal activity and those engaged in hacking are themselves often considered criminals. Hacking however comes in two forms, black hat and white hat. A black hat hacker or more correctly referred to as a cracker is the hacker most think of when referring to hacking. A black hat hacker hacks for personal gain and generally engages in illegal hacking actives. On the other hand a white hat hacker is a hacker that hacks for exploration, learning and testing. This paper is written from the white hat hacking perspective, such that the goal of this paper is to learn, explore and understand how to secure web systems be using the WebScarab intercept proxy. It is however worth noting that both white hat and black hat hacking generally use the same tools and techniques but as stated earlier the key difference between the two is the reason and goal of the hacking activity.

### Web Hacking

Web hacking is a sub category of hacking which involves attacking a web application instead of an entire system. Web hacking is often performed without even needed to compromise the underlying server system of the web application being attacked. Web hacking activities are normally directed towards looking for vulnerabilities in the web applications that the developer may have overlooked such as not validating inputs properly or validating on the client side. Clients can never be trusted as they have the ability to alter their response thus bypassing any validation that was performed on the client side. Validation done incorrect, such as validating on the client side, or not validating inputs at all are why web hacking is extremely effect and very common to this very day.

## Research Constraints

Hacking is a very large subject and is part of the even larger subject of computer and network security. It is for this reason that this paper is constrained to web hacking only and more specifically the exploration of a web hacking tool. Furthermore since there are many web hacking tools this paper will focus on just the WebScarab intercept proxy tool.

## Intercept Proxies

Intercept proxies are type of hacking and security test tool designed to allow the hacker to alter their request and response traffic with a application server. The ability to alter ones request and response traffic with a server allows the hacker to take advantages of weaknesses in the web applications. These weaknesses generally center around, as mentioned in the previous section, missing or improper validation. The intercept proxy allows the hacker to mangle their request in such as way that they can get the server to perform unintended actions or operations. There are many intercept proxies available and likewise there is a wide range of uses for them and because of this intercept proxies range from fairly simple to very advanced. WebScarab which this paper will be focusing on is a very advanced intercept proxy with many advanced features not found in more basic tools.

# Open Web Application Security Project (OWASP)

The Open Web Application Security Project (OWASP) is a community dedicated to enabling organization to develop secure applications [1]. OWASP provides tools and documentation geared towards teaching and enabling individual contributors to develop in more security conscious web applications. OWASP has too many projects to detail here and many are outside the scope of this paper as such this section will focus on the WebScarab project and the related WebGoat project.

## WebScarab

WebScarab at its core is an intercept proxy. However what sets WebScarab apart from other intercept proxies is its advance capabilities that build on the underlying intercept proxy functionality. Because of WebScarab’s abilities to do more than intercept traffic OWASP describes it as an analysis framework for web applications [2]. WebScarab is capable of manually or automatically intercepting, analyzing and altering request and response traffic between a client and server. Additionally WebScarab provides other tools beyond intercept proxy related activities such as being able to base 64 encoding/decoding, MD5 hash and authenticate against web applications using SSL. All which can be achieved using multiple applications but WebScarab brings these features into a single convenient location.

## WebGoat

Tools like WebScarab are generally meant to perform security testing on one’s own application servers but this is not always the case. WebScarab can also be used to explore different security vulnerabilities and cracking techniques. The issues with this however is if you’re not testing a specific target then in order to be able to explorer and test attacks you need a target. Attacking an arbitrary server on the internet however is not advisable and not only that it is in most circumstances illegal. This is where WebGoat comes in. WebGoat is an intentionally insecure web application [4]. WebGoat provides a legal way to explorer web hacking. Additionally WebGoat also serves as a web security training tool as it includes lessons on performing web hacks and then explains how to secure against those web hacks.

# Intercept Proxy Setup

The following section will detail setting up and configuring WebScarab as well the configuring a web browser to use WebScarab and the optional step of installing and configuring WebGoat.

## Prerequisites

WebScarab is a Java based application and as such is operating system independent, with the caveat that there must be a Java virtual machine available for the operating system. With that in mind it is assumed that an appropriate operating system is already installed and configured. Additionally Java virtual machine version 1.4 or newer as well as an unzipping application such as 7zip must be installed. It’s also assumed that the necessary zip files for WebScarab have already been downloaded.

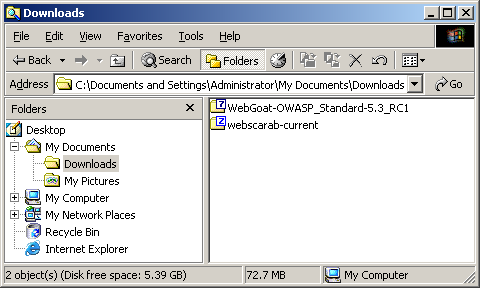
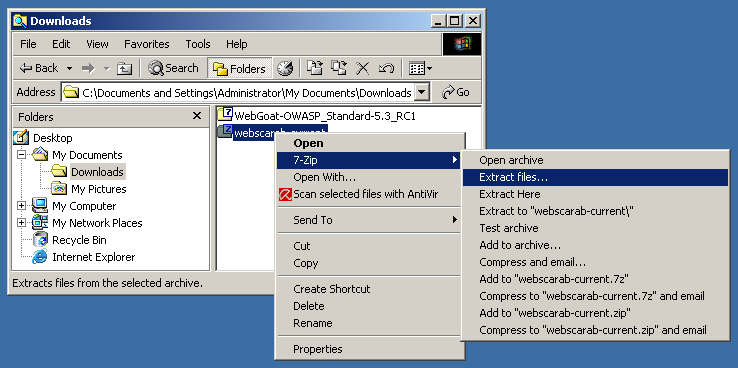
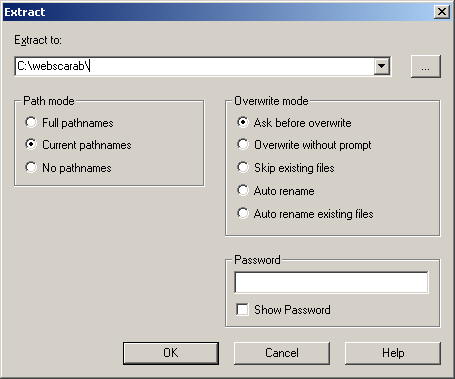
## Notes

This exploration was done using a virtual machine with Windows 2000 Professional installed. Additionally the Windows 2000 Professional install was updated with Service Pack 4 and Updates Rollup 1. The Windows installation was also installed with additionally third party software, including Sun Java virtual machine version 1.6.0 update 16, Mozilla Firefox and 7zip.

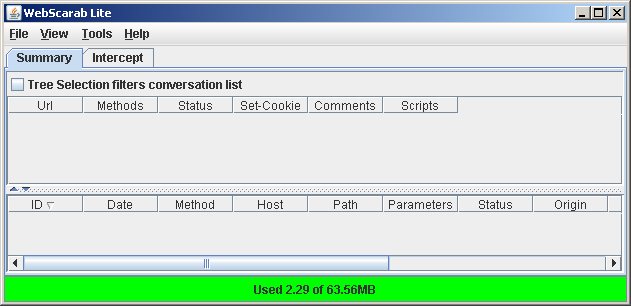
As such this setup guide is geared towards setting up WebScarab on a Windows platform. However this setup guide can be easily adapted to any platform. Additionally this setup guide is based on guides available from Open Web Application Security Project and as such their guide will be more generalized[1][3].

It’s important to be aware that due to WebScarab being run under a Java virtual machine and the requirement to have multiple applications running simultaneously memory demands placed on the virtual machine with be high. As such it is important to ensure there is enough available memory available to the virtual machine for both the guest operating system and the applications running within it. It’s recommended to have at least 512MB of memory allocated to the virtual machine.

## Installing WebScarab

1. Open an Explorer window to the directory WebScarab was downloaded to.  
   
2. Right click and navigate though the context menu to the unzipping application, in this case 7zip, and select “Extract files…”.  
   
3. In the “Extract to:” field enter the location to unzip WebScarab to, in this case “C:\WebScarab\” and click the “OK” button.  
   
4. The unzipping tool will display a progress bar as it unzips the archive and then exits when the unzipping activity has completed successfully.

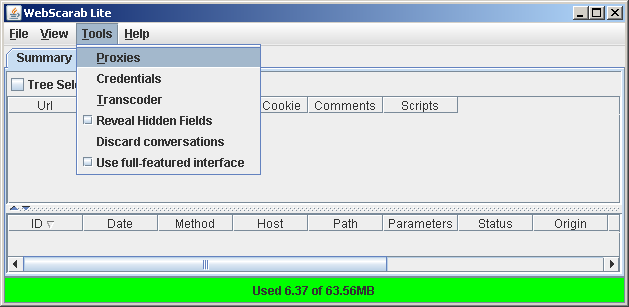
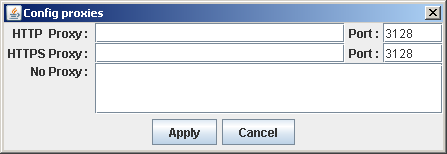
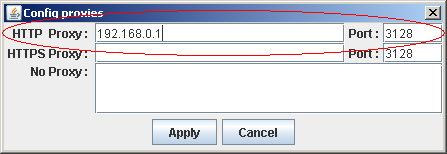
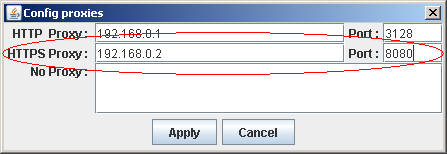
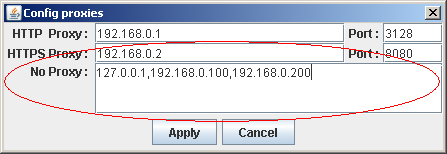
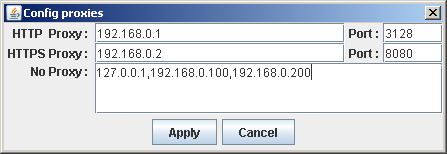
## Launching WebScarab

1. Open a command prompt window and navigate the WebScarab folder created after installing WebScarab.  
   
2. In the WebScarab folder enter the command.  
   
3. Once WebScarab has started the main window will be displayed.  
   

## Configuring WebScarab

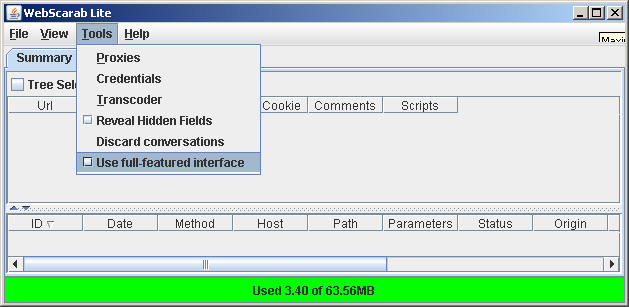
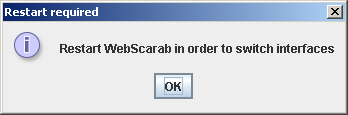
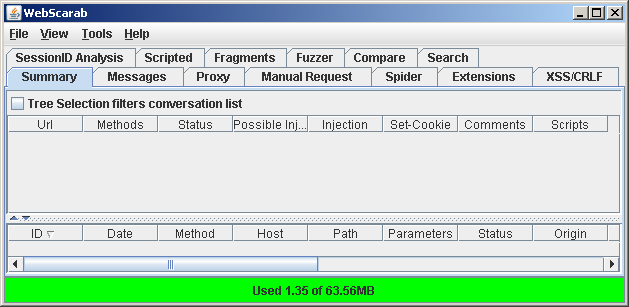
WebScarab has configuration options that accommodate different network options and different usages pattern. The configuration options generally only need to be changed if the user’s situation necessitates it. As such the following section covers a select subset of these configuration options.

### Proxy

1. It is often common, particularly in corporate networks, to be required to use a proxy to access the internet.
2. To configure WebScarab to use a proxy server select the “Proxies” under the “Tools” menu.  
   
3. The “Config proxies” window is displayed which has three sets of fields.  
   
   1. The first set of fields is the “HTTP Proxy:” fields. These two fields are for configuring the address and port of a proxy to use for HTTP requests.  
      
   2. The second set of fields is the “HTTPS Proxy:” fields. These two fields are for configuring the address and port of a proxy to use for HTTPS requests.  
      
   3. The final set of fields is the “No Proxy:” fields. This field is for configuring network address that should not be routed through a proxy. Addresses for nodes on a local network and the local host would be good candidates for the no proxy field.  
      
4. Once the appropriate values have been entered click the “Apply” button to accept the changes.  
   

### Interface Modes

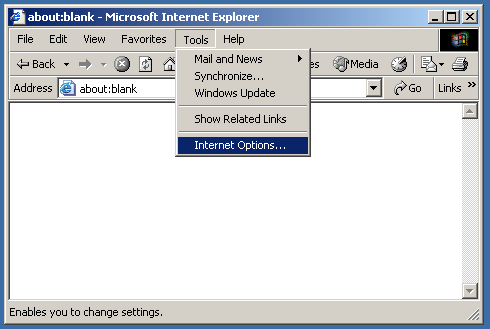
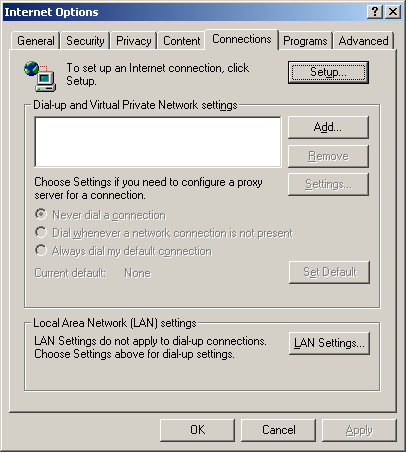
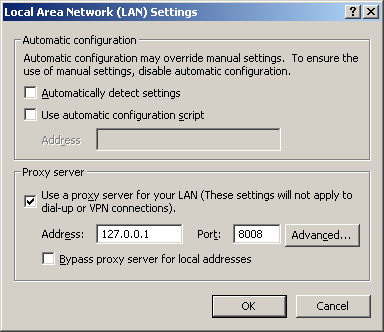
WebScarab has two different interface modes. The first which is known as the “Lite” mode is a subset of WebScarab’s full feature set. This subset of features can be thought of as WebScarab’s core feature set. WebScarab’s core feature set is just the ability to intercept communications between a browser and a web service. WebScarab’s other mode is known as the “full-featured” interface which displays all functions WebScarab is capable of performing. WebScarab in full-featured mode is capable of performing many tasks including but not limited to “Spider” a target site and scan the target site for unintentionally left files such as “.bak” files [1].

1. Check or uncheck, depending on the desired effect, the “Use full-featured interface” option under the “Tools” menus to change the interface mode.  
   
2. Once the full-feature interface setting has been changed WebScarab will prompt to be restarted. Exit WebScarab after clicking the “OK” button on the restart prompt.  
   
3. Re-launch WebScarab. WebScarab will launch its main window in the appropriate mode.  
   

## Configuring Web Browser

In order to use WebScarab a web browser most be configured to use WebScarab as a proxy. By doing this the browser sends its requests to WebScarab and the responses from the web server are sent to WebScarab. This enables WebScarab to intercept, monitor and alter the traffic as described earlier in this paper. This section will only cover one of the most popular web browsers, Microsoft Internet Explorer. Other browsers have similar methods for setting a proxy and the procedure to do so can be found in the respective browser’s documentation.

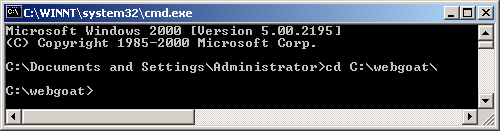
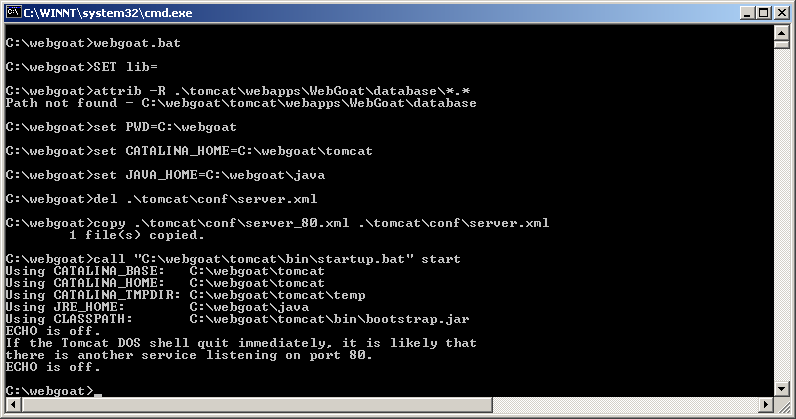
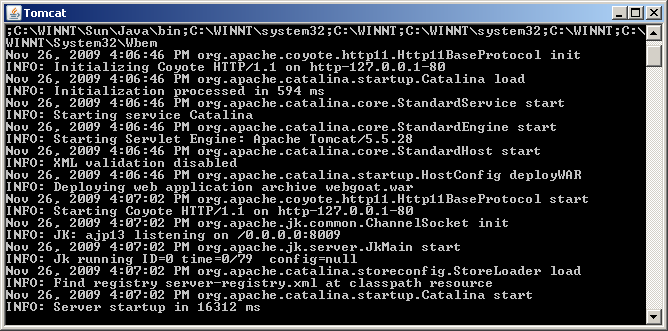
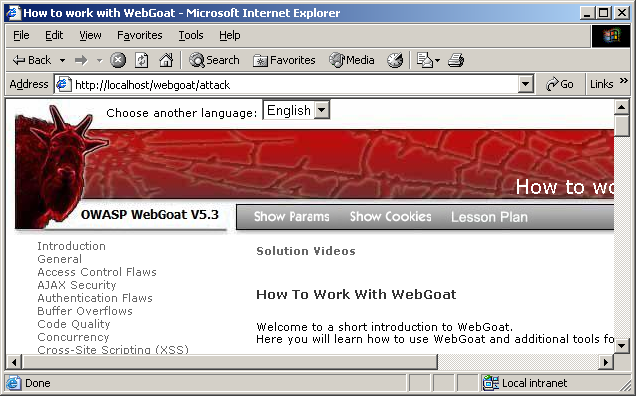
### Internet Explorer

1. Launch Internet Explorer and select “Internet Options…” from the “Tools” mean.  
   
2. Select the “Connections” tab and click the “LAN Settings…” button under that tab on the “Internet Options” window.  
   
3. There are two sections in the “Local Area Network Settings” window. Under the second section which is labeled “Proxy server” select the “Use a proxy server for your LAN” checkbox and enter “127.0.0.1” as the address and “8008” as the port.  
   
4. Select the “OK” button on both the “Local Area Network Settings” and the “Internet Options” windows to save the settings.
5. With WebScarab running navigate to a web page, such as Google, to verify the settings are correct and working properly. If the settings are working correctly the selected web page will be displayed normally as WebScarab operates as a transparent proxy when first launched.

## WebGoat

This section briefly describes the optional step of installing and launching WebGoat.

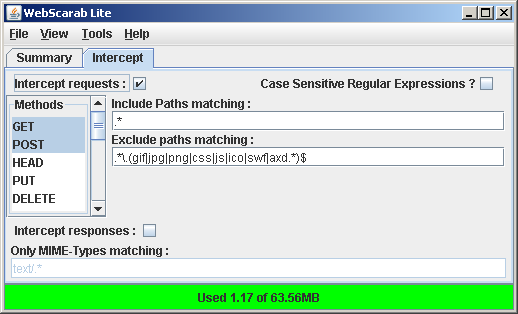
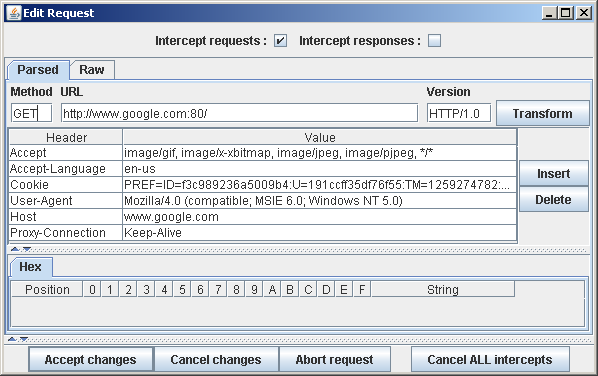
Be warned that while WebGoat is running the system will be insecure and vulnerable to attack. It is because of this it is highly recommended that the system be isolated from untrustworthy networks such as the internet. When using a virtual machine the guest operating system’s connection can be configured as NAT which allows the guest operating system to access the internet but disallows any remote system on the internet to connect back to the guest operating system.

1. Just like WebScarab unzip the WebGoat zip to the desired directory, open a command prompt and navigate to the newly created directory.  
   
2. Enter the command “webgoat.bat” to launch WebGoat.  
   
3. Once WebGoat is running a new “Tomcat” console window is display.  
   
4. Navigate to “http://localhost/webgoat/attack” in the browser configured to use WebScarab as a proxy, logging in as “guest/guest” for the username and password, to test if WebGoat is functioning properly.  
   

# Exploring WebScarab

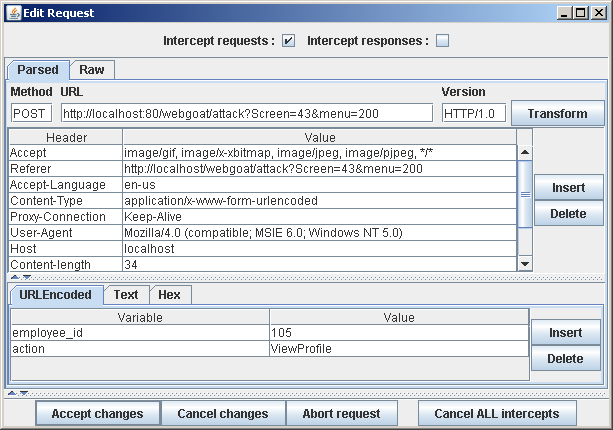
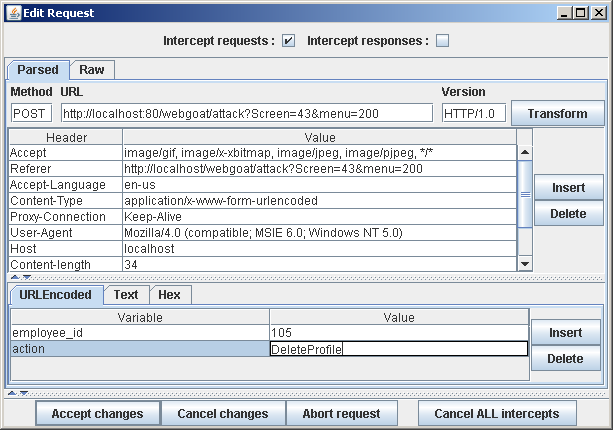
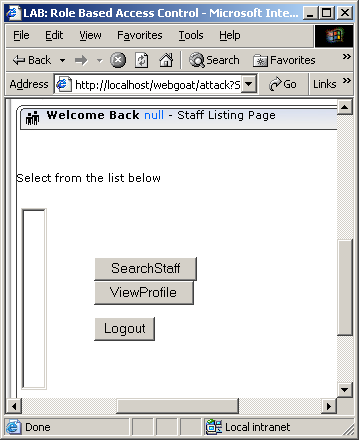
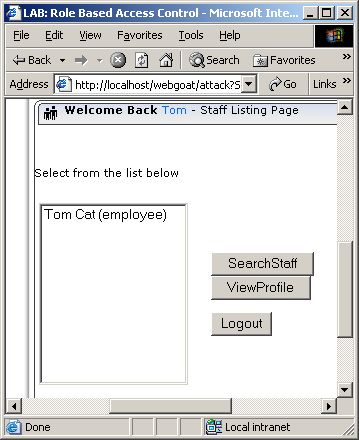
WebScarab at its core is an intercept proxy but WebScarab has many features beyond being just an intercept proxy. These features are, however, centered on WebScarab’s core functionality of being an intercept proxy. As such this section will first explore WebScarab’s intercept proxy functionality to then building on that ground work explore some of WebScarab’s more advanced features.

## Intercepting Requests

1. With WebScarab running select the “Intercept” tab and select the “Intercept requests:” checkbox. Verify that at list “GET” and “POST” methods are highlighted.  
   
2. In the browser configured to use WebScarab as a proxy server navigate to “http://www.google.com/” or any site. WebScarab will intercept the request and display a window allowing the request to be edited.  
   
3. The window displayed by WebScarab shows all details of the request, including the type of request, the user agent and any cookies. In this case it’s a simply GET request but it illustrates what WebScarab is doing.

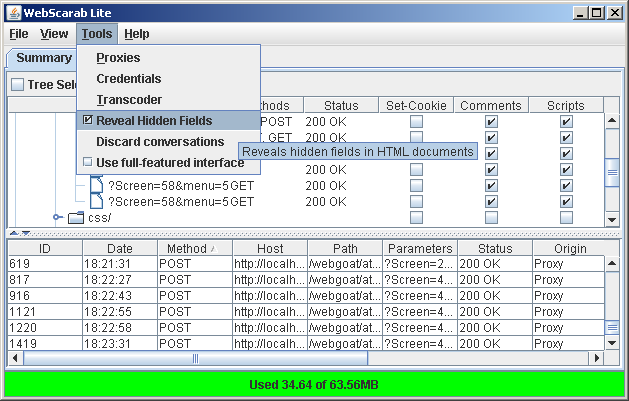
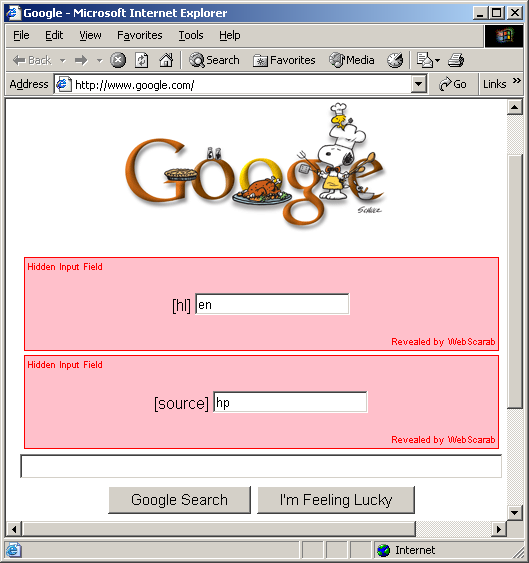
## Altering Requests

Intercepting requests is a very useful tool for hacking a web application but it’s nothing that a simple packet sniffer can’t do. Where intercept proxies like WebScarab are truly powerful is their ability to alter the requests before sending them to the server. In the example below using WebScarab to attack WebGoat the request is altered such that a user with insufficient privileges to delete the profile of another user is able to delete a user’s profile.

1. Just like when intercepting request with WebScarab running select the “Intercept” tab and select the “Intercept requests:” checkbox. Verify that at list “GET” and “POST” methods are highlighted.
2. WebScarab will once again display a window with all the details of the request. This time however it’s a POST request and the request is to view a profile.  
   
3. Assuming we’ve already figured out that the command to delete a profile is “DeleteProfile” we will alter the request to delete the user’s profile instead of viewing it. Once the request has been alter click the “Accept Changes” button to forward the altered request to the server.  
   
4. As can be seen in the before (left) and after (right) screenshots of the listing page that the after screenshot is now blank indicating that we successfully deleted the users profile.  
   

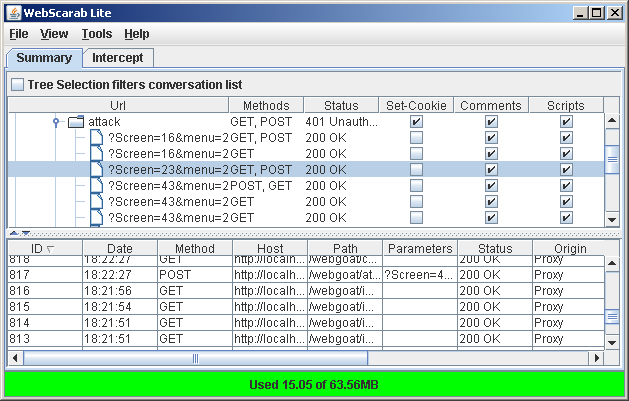
## Reveal Hidden Fields

Hidden fields are commonly used in web pages to store information, particularly settings. Thus it is very useful to view the source of the web pages being attacked and look for the hidden fields. The problem with this is it is an awkward and slow process to look through web page source code to find hidden fields that may provide useful information. WebScarab deals with this by providing a feature that intercepts the responses from the server and alters them so that the hidden fields are clearly visible on the web page itself. Making it much easier to find and exploit the hidden fields.

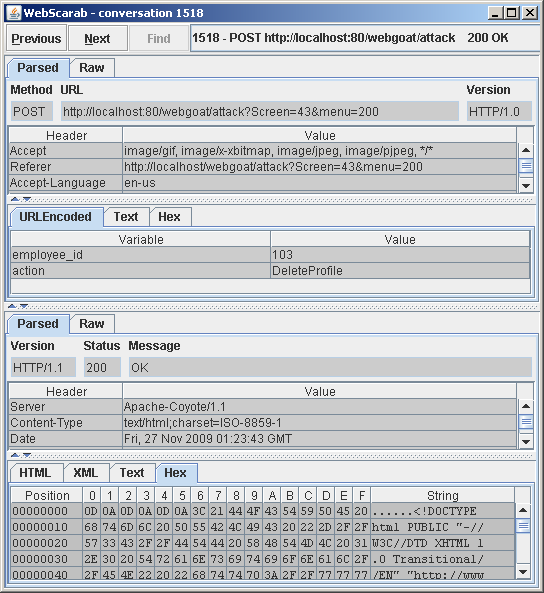
1. With WebScarab running select the “Reveal hidden fields” checkbox under the “Tools” menu.  
   
2. Using the browser configured to use WebScarab as an intercept proxy navigate to a web page that has hidden field to be revealed, in this example Google was used.  
   

## Communication History

WebScarab keeps a history of all communications between the browser and the server that were routed through it. This is a very useful feature as it allows the behavior of a web application as well as its communication and command methodology.



Additionally there is the possibility of intercepting a useful piece of information. An example is in the earlier section about altering requests WebScarab intercepted the correct value to use to issue a delete command to the web application being hacked. As can been seen in the screenshot the delete request was intercepted which gives use the correct command to use when attempting to execute an unprivileged delete.



# Future Research

There are many aspects to web hacking and hacking in general that were not discussed in this paper. Additionally the WebScarab capabilities explored in this paper are only a very small subset of what WebScarab is capable of. This is because there are far more capabilities available in WebScarab then the limited scope of this paper explored. Furthermore OWASP has many other projects related to web application security that either was not discussed or only discussed briefly in this paper, such as WebGoat. As such the following topics are for future research.

* OWASP Projects
  + Explore the remaining features not covered of WebScarab.
  + Fully explore the insecurities of WebGoat.
* Web Hacking
  + Fully explore the web hacking activities not covered that an intercept proxy can facilitate.
  + Explorer other web hacking topics outside of intercept proxies such as SQL injection.

# References

1. Open Web Application Security Project. 2009.  
   About The Open Web Application Security Project  
   http://www.owasp.org/index.php/About\_The\_Open\_Web\_Application\_Security\_Project
2. Open Web Application Security Project. 2009.  
   OWASP WebScarab Project  
   http://www.owasp.org/index.php/Category:OWASP\_WebScarab\_Project
3. Dr. Holger Peine, Fraunhofer IESE. 2007  
   The interactive HTTP proxy WebScarab – Installation and basic use  
   http://www.acsac.org/2007/downloads/t5-webscarab-instructions.pdf
4. Open Web Application Security Project. 2009.  
   OWASP WebGoat Project  
   http://www.owasp.org/index.php/Category:OWASP\_WebGoat\_Project
5. Open Web Application Security Project. 2009.  
   WebGoat Installation  
   http://www.owasp.org/index.php/WebGoat\_Installation
6. Open Web Application Security Project. 2009.  
   WebGoat 5.3 README