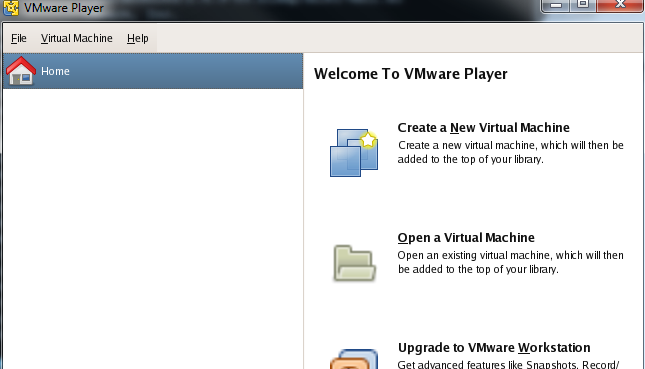
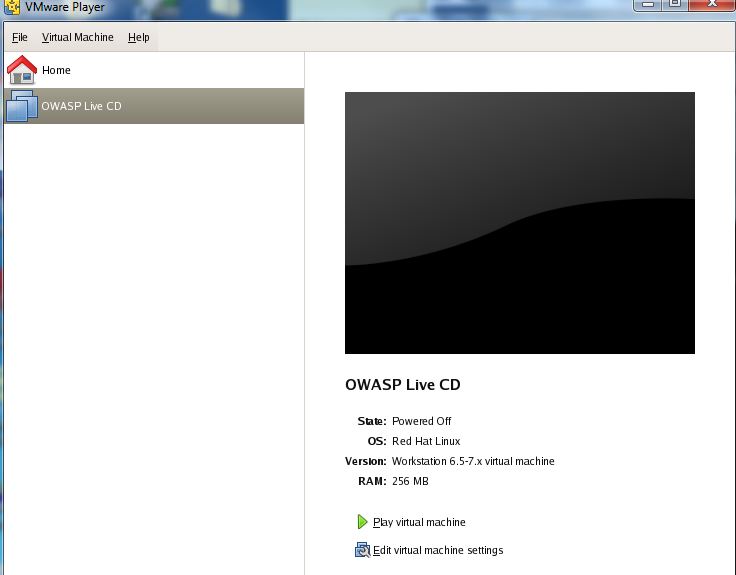
ITEC 345  
Testing web vulnerabilities  
hw-ec-owasp  
Due: 2013.Dec.12 (Thu) 17:00

**Notes:**

1. The lab requires you to start up a virtual linux machine. On a PC (windows/linux) you can download VMPlayer for free (search for VMplayer on [www.vmware.com](http://www.vmware.com) website). On Mac, download “VMPlayer Fusion”, which comes with a 30-day trial version.
2. You can use any computer in Davis Hall (they all have VMware Player installed on them), or you can install VMplayer or VMware fusion on your personal computer. You will also need to download the Owasp virtual image required for this project. (Start the 630MB download early, since it can take a while. You’ll also need enough disk space to decompress the ~2.5 GB image. If you decompress it to a (large enough) thumb drive, you won’t be tethered to a particular disk.) To download the disk image:
   1. Go to <http://mtesauro.com/livecd/>, and browse down to *Downloads > VMWare and Virtual Box Installs*.
   2. Download: Austin Terrier VMware Install: owasp-livecd-AustinTerrier-Feb2009.vmdk.rar. *Note*: There are several images on the above website – make sure you download the Austin Terrier Feb2009 .vmdk file, and not something else. This download might take a while (≥ 20min on a *fast* connection); the compressed file is 630MB.
   3. Un-archive/un-compress this file into a folder of your choice. Note that this is a .rar file. If it is not automatically de-compressed on your machine, you can use programs such as Stuff-It (Mac), or WinRAR or 7-zip (<http://www.7-zip.org/>) (Windows) to un-archive this file.
   4. Download the file Owasp.vmx file from *D2L > Dropbox > hw-ec-owasp* and save it to the same folder as in step (c).
   5. Start up the VMPlayer. You will now see the following window, click on **Open a Virtual Machine.** 
   6. Select the file: Owasp.vmx (browse to the folder from step (c).)
   7. **Select “open” at the bottom of the screen. You will see the following screen.**
   8. Select “**Play Virtual Machine**”.
   9. Ignore any warning (just say OK). **For one of the pop up windows, you will have to click on “I copied it” option.**
   10. You will be logged in. (The system may reboot sometimes, but after some time you should see a blue screen with a wasp symbol on it.
   11. **Next: the actual assignment.**

**Assignment**

*(You can discuss this homework with others, and work side-by-side with them. However,* ***you must finish your own homework and submit it independently****. You cannot copy or exchange screen shots. In your submission clearly cite any others who assisted you.)*

**STEP 1:** Make sure you have started up the Linux virtual machine (you will see a WASP symbol). This is the OWASP live CD (the software that I left on this is from the OWASP group)**.** W**e** will now use this to run a few attacks on a web application and a database management system (MySQL - www.mysql.com).

**STEP 2:** Start up a terminal on OWASP (look for the black terminal-window icon on the bottom of the screen).

**STEP 3:** Start a program called “webgoat”, which is a buggy webserver. **On the terminal,** **simply type:**

webgoat start80

*Note: If you have previously started webgoat and are not sure if it is still running or not, simply check the status by typing:* **webgoat status**

**STEP 4:** Start up two software applications: **WebScarab** and the **Mozilla Firefox** browser.

1. **Firefox:** A transliteration of the Chinese name for a Red Panda(which is not actually a panda, but is still [danged cute](https://www.google.com/search?q=red+panda)).
2. **WebScarab:** This is the software at the bottom of the screen with a spider on its icon (next to the Firefox icon)**.** This is aweb browser *proxy* – it is used to intercept all the requests as well as data that a web client such as Firefox sends or receives. You will be using this to intercept and add malicious modifications to the web requests from the Firefox browser during the course of an attack.

**STEP 5:** Get familiar with WebScarab. Check out the various options. Specifically,

1. Change the WebScarab interface to a “Lite” interface. Search through the menu options to find this selection. The Lite interface is easier to work with. Once you select “Lite” interface close and restart WebScarab.
2. Once WebScarab restarts in Lite interface, select the “intercept” tab, check the box “intercept requests”.

**STEP 6:** in Firefox, visit the URL **http://localhost/WebGoat/attack**

**STEP 7**: Login in with the highly secure, unbreakable, inviolable, impregnable username / password:  
 **guest** / **guest**

**STEP 8:** Read the warning at the bottom of the page. Commit it to memory.

**STEP 9:** Click on **Start WebGoat**.

**STEP 10:** Try to get a feel for the website. The menu bar on the left has various lessons that teach attacks. Read at least the following:

1. Introduction – How to work with WebGoat. (click on lesson plan on the top menu after selecting a topic on the right side).
2. General (which covers HTTP basics).

**STEP 11: Get familiar on how to intercept requests from the Firefox browser with WebScarab:**

1. On Firefox browser, search for the FoxyProxy plugin.
2. Once you click on this plugin, it will give you options to pick the proxy server to use.
3. **Caution:** Once you select WebScarab as your preferred proxy, remember that every web data between Firefox and any website will now be intercepted and STOPPED by WebScarab. You will have to go to the WebScarab interface and manually click on “Accept Changes” before either the data is allowed to pass to either the website or from the website to Firefox. To avoid spending too much time on this, you could only select WebScarab as the proxy when you are about to start the attack. All other times, keep FoxyProxy disabled.

**STEP 11:** Execute the following attacks. If an attack asks you to enter someone’s user name (e.g., say Tom Katz’s username and password), it will be tom/tom. When running some of these attacks, you may have to enable FoxyProxy plugin on Firefox to use WebScarab. *(Part of this exercise is self-learning – you can make use of the solution provided on the site but I strongly urge you to first see the hint, try if you can figure out how to attack, and only then see solution (if needed)).*

1. Parameter Tampering – Exploit Hidden Fields
2. Cross Cross Site Scripting – Stage 1 Stored XSS

Hint: <script>alert("haha, made you look");</script> is valid, if poor, javascript.

1. Inject Injection Flaws – Blind SQL injection

Hint: in SQL, strings are delimited with single-quote characters. Also, you can compare strings alphabetically using “>=”. So “first\_name >= 'Mjj'” will evaluate to true or false, accordingly.

1. Choose two additional attacks from the menus, and complete them. To get full credit, *also* provide a ½ page write-up briefly describing the vulnerability that the attack exploits, and how the attack exploits that vulnerability. As always, clear writing is part of what is graded. Attach a plain-text .txt file – *not a* .docx file.

***Note: After each attack is done, a green check mark will be shown next to your successfully-completed attacks. At the end of the homework, you will simply take screenshot(s) of these, to submit as your deliverable.*** Screenshots showing multiple green check-marks at once are preferred over multiple screenshots.

**STEP 13:** Once you are done**, don’t kill the window**; instead shutdown**.** Once you are done with the security attacks or if you want to exit – click on the icon with a “K” (for KDE) on the OWASP Linux, at the lower right corner and select *Log Out > Shutdown*. This will ensure that your virtual machine is not damaged (not doing this but simply killing the window or closing putty SSH is like pulling the power chord to shut down a desktop or a workstation). An attack may have left the data on the disk in an unstable state and hence a shutdown is necessary.

**DELIVERABLES** Submit screen shot(s) on D2L showing the left-hand-side’s green check-marks for each attack you completed, and one .txt file with the brief description of the other two attacks you completed.