Lab - Learning to Hack Linux Using Metasploitable2

Overview

In this lab, you will be introduced to hacking Linux using a vulnerable install of Linux called Metasploitable2. Metasploitable2 is an intentionally vulnerable Linux virtual machine. This VM can be used to conduct security training, test security tools, and practice common penetration testing techniques.

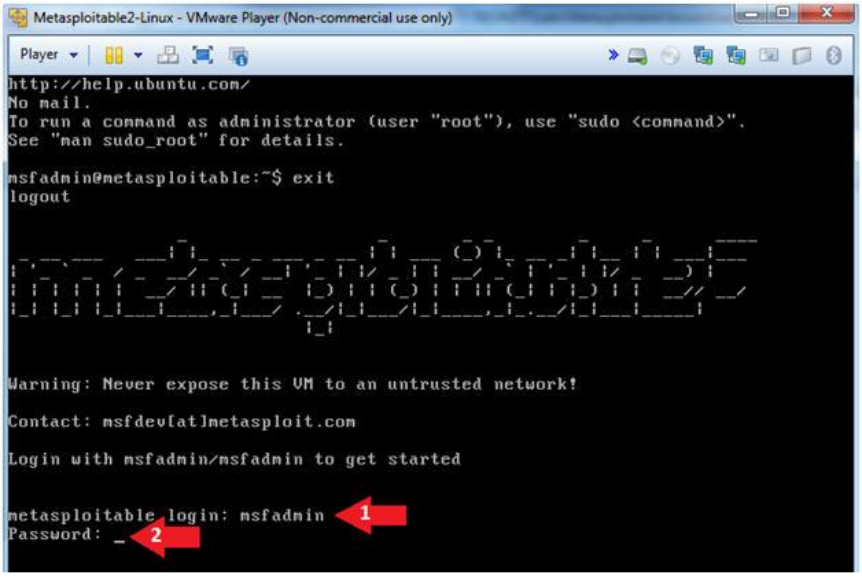
Hardware requirements for these labs:

1. Do not use a Wi-Fi connection. Use an Ethernet cable to connect to the network. Wi-Fi is configured for IPSec which can impede the labs from working. The additional transport and tunneling protocols do not play well with Kali or Metasploit.

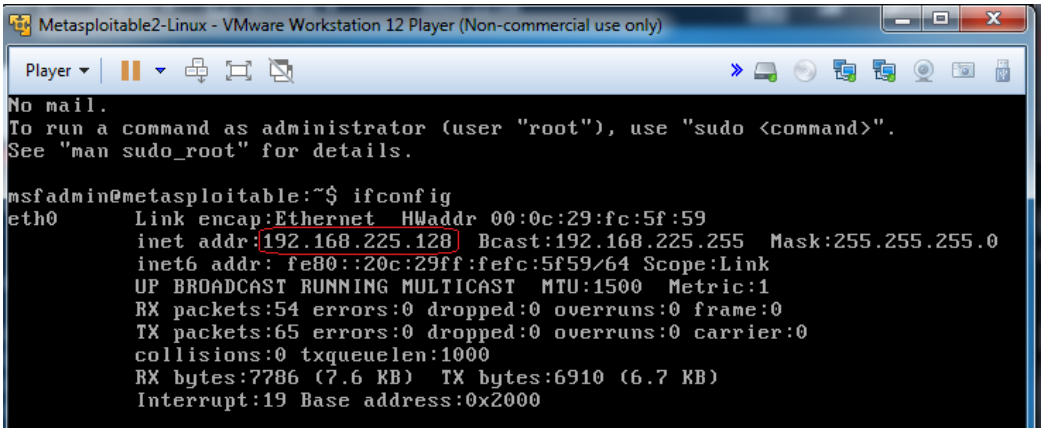
Getting Started

After Metasploitable2 boots, login to console with username ***msfadmin*** and password ***msfadmin***. There is no GUI.

(For security purposes, the password will not be visible when you type it in.)



From the shell, run the ifconfig command to identify the IP address of your install of Metasploitable. Write down or remember the IP address of this machine. Make sure it is up and running. Make sure you can ping Metasploitable from your Kali install.

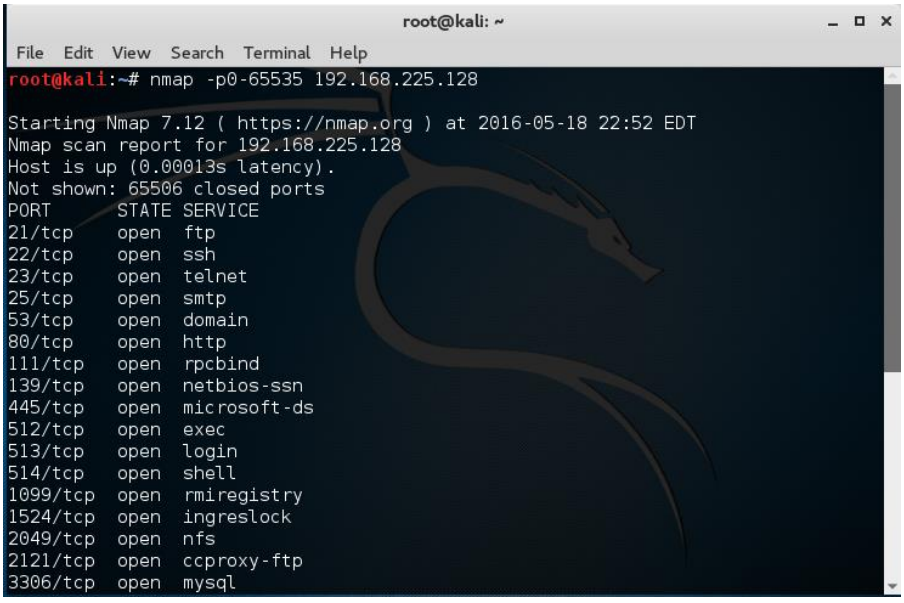


Services

From our attack system (Kali), we will identify the open network services on this virtual machine using the Nmap Security Scanner. The following nmap command will scan all TCP ports on the Metasploitable2 target.

```
root@ubuntu:~# nmap -p0-65535 192.168.225.12
```

This the IP of my Metasploitable victim! Not yours!

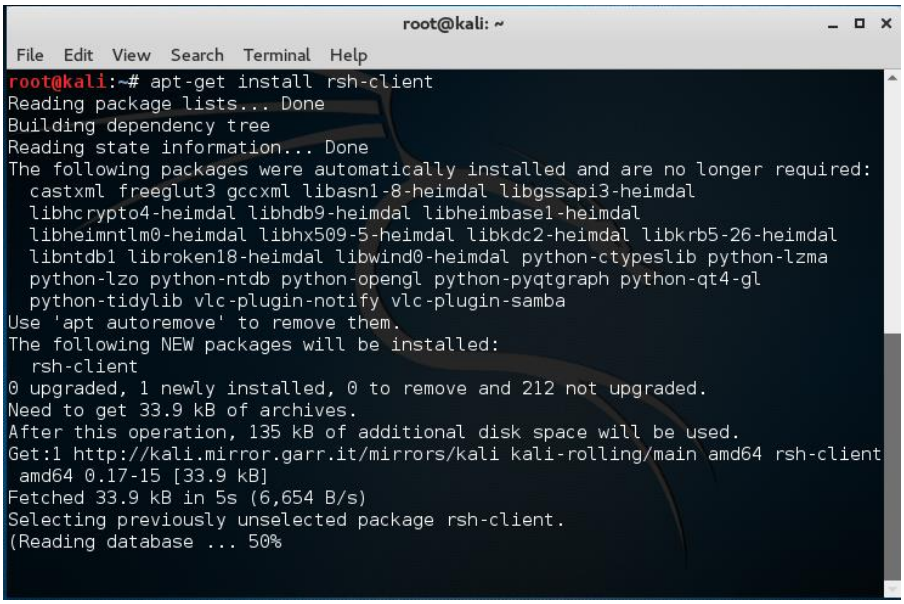


Nearly every one of these listening services provides a remote entry point into the system. In the next section, we will walk through some of these vectors.

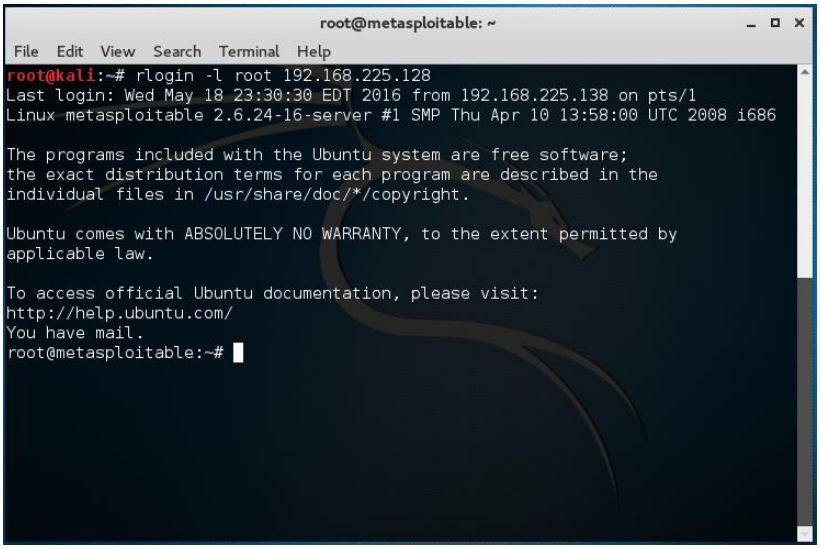
Services: Linux Basics

TCP ports 512, 513, and 514 are known as "r" services and have been misconfigured to allow remote access from any host (a standard ".rhosts + +" situation). To take advantage of this, make sure the "rsh-client" client is installed (on Kali), and run the following command as your local root user. If you are prompted for an SSH key, this means the rsh-client tools have not been installed and kali is defaulting to using SSH.

We first need to install the RSH tools using **apt-get install rsh-client**



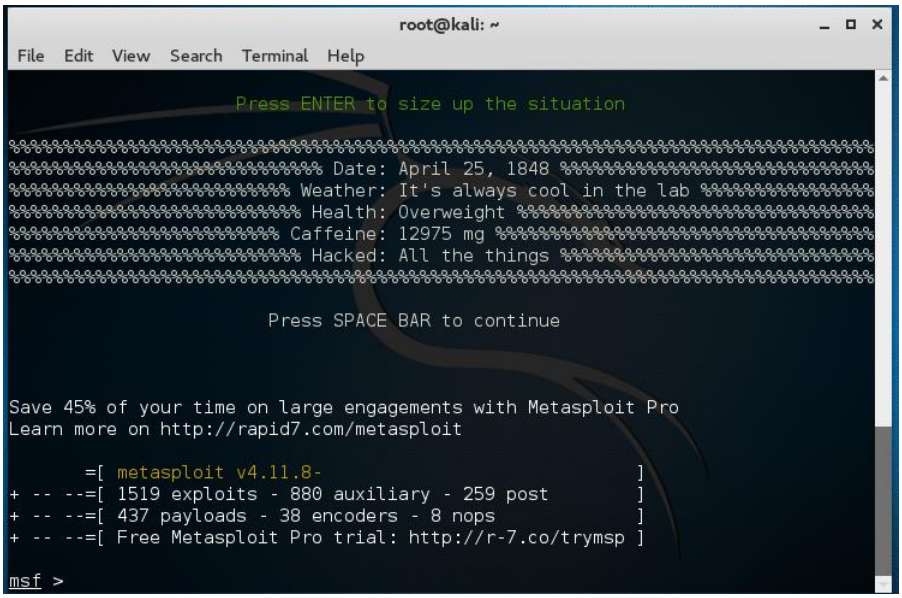
After the RSH client has completely installed, you should be able to log in without being prompted for any password.



Back to your Kali install…….

On port 6667, Metasploitable2 runs the UnrealRCD IRC daemon. This version contains a backdoor that went unnoticed for months - triggered by sending the letters "AB" following by a system command to the server on any listening port. Metasploit has a module to exploit this in order to gain an interactive shell, as shown below. The right exploit will do the leg work for us…

At the terminal prompt type `msfconsole` to start the Metasploit program.



Searching the Exploit Database using Searchsploit

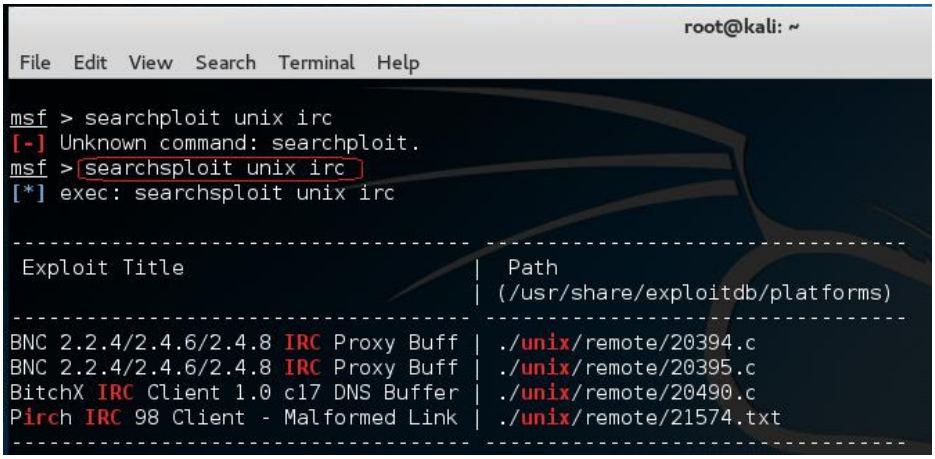
Exploits in Metasploit come and go. They are either updated or replaced or removed from the database. An example would be the old exploit **exploit/unix/irc/unreal_ircd_3281_backdoor** is no longer available and has been replaced with:

exploit/unix/irc/unreal_ircd_3281_backdoor

This happens quite a bit, but the solution is to search the MSF database for the updated exploit.

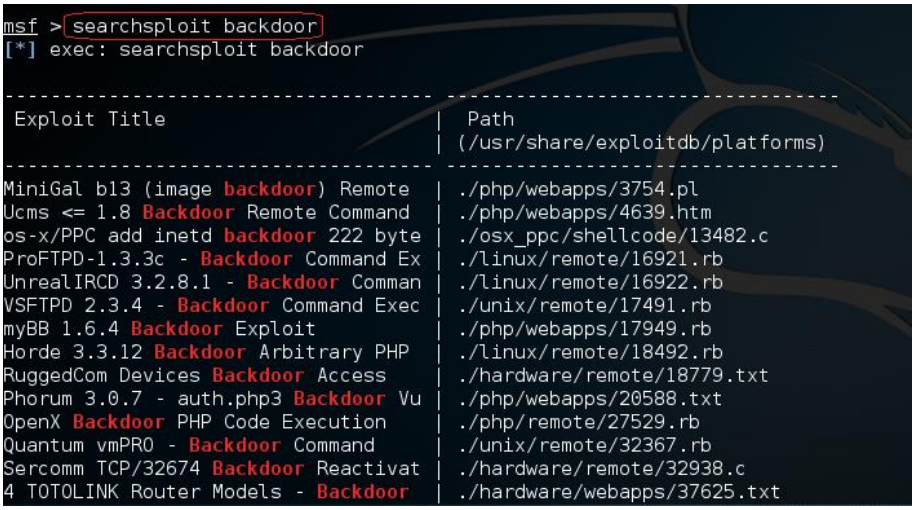
In Metasploit, you can use the **searchsploit** command to drill down until you find what you are looking for.

In this example, I based my search on keywords from the old command…. I started out looking for …. **unix irc**



Nothing useful here!

I next search for just the word **backdoor….**to many results!



Finally, I did a search for **irc backdoor….** I found the updated exploit using the same exploit ID, 3.2.8.1! Success!

Use the exploit!

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
```

(Pay attention to the underscores!)

You can use the `options` command to see what settings have to be configured.

Set the remote host using the IP address of our Metasploitable victim.

Attack!

What you end up with is access to the victim using a console shell. You can now have your way with the victim. Try typing in ifconfig. You're seeing the adapters located on the victim.

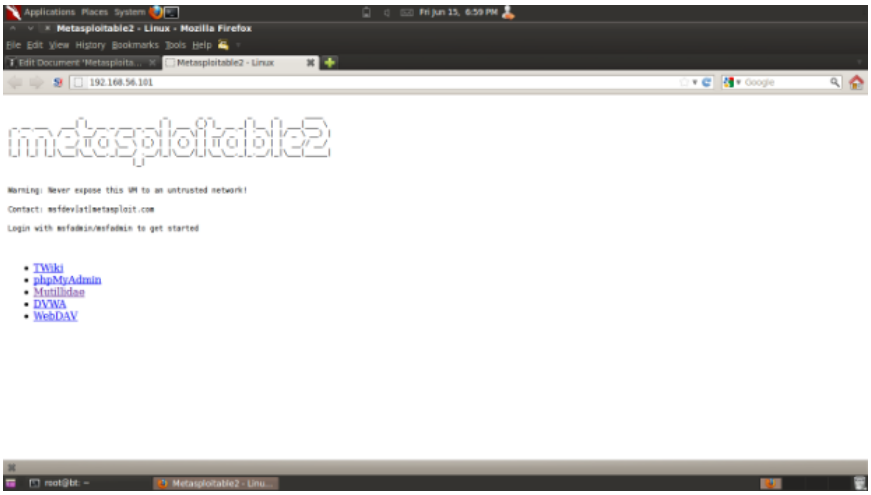
You can list the contents of the victim's directory you are in by typing ls at the prompt.

Vulnerable Web Services

Stop!!! Read this carefully.

Metasploitable2 comes with vulnerable web applications pre-installed. The web server starts automatically when Metasploitable2 is booted. To access the web applications, open a web browser and enter the URL http://<IP> where <IP> is the IP address of Metasploitable2. One way to accomplish this is to install Metasploitable2 as a guest operating system in Virtual Box and change the network interface settings from "NAT" to "Host Only".

In this example, Metasploitable2 is running at IP 192.168.56.101. Browsing to http://192.168.56.101/ shows the web application home page.



Note: 192.168.56/24 is the default "host only" network in Virtual Box. IP addresses are assigned starting from "101". Depending on the order in which guest operating systems are started, the IP address of Metasploitable 2 will vary.

Stop!!! Read this Carefully. The following application comes preinstalled with Metasploitable2!

To access a particular web application, click on one of the links provided. Individual web applications may additionally be accessed by appending the application directory name onto http://<IP of your Metasploitable install> to create URL http://<IP>/<Application Folder>/.

For example, the Mutillidae application may be accessed (in this example) at address http://192.168.56.101/mutillidae/. (You IP address will vary)

The applications are installed in Metasploitable 2 in the /var/www directory. (Note: See a list with command "ls /var/www".) In the current version as of this writing, the applications are

- mutillidae (NOWASP Mutillidae 2.1.19)
- dvwa (Damn Vulnerable Web Application)
- phpMyAdmin

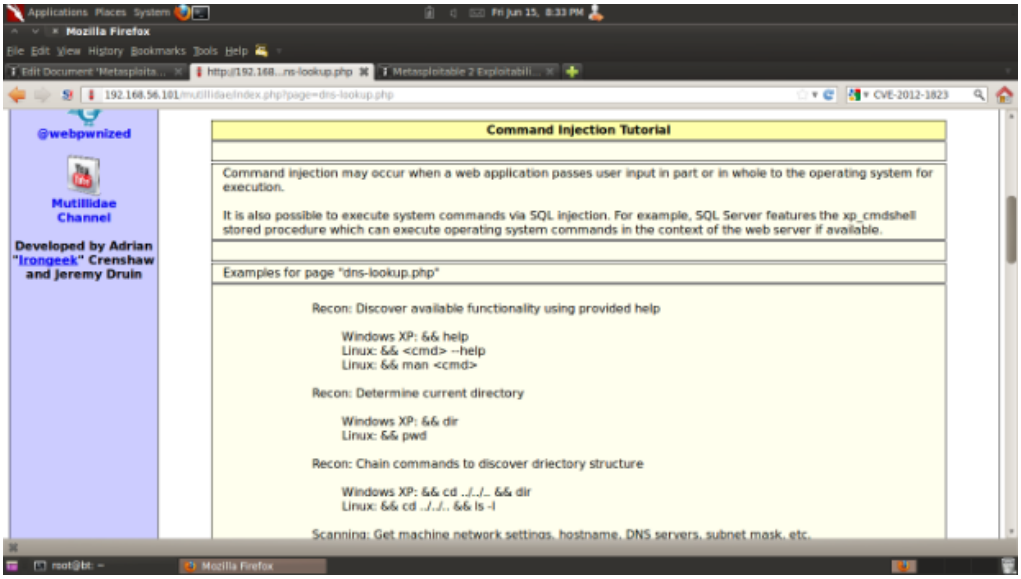
- tikiwiki (TWiki)
- tikiwiki-old
- dav (WebDAV)

Vulnerable Web Service: Mutillidae

The Mutillidae web application (NOWASP (Mutillidae) contains all of the vulnerabilities from the OWASP Top Ten plus a number of other vulnerabilities such as HTML-5 web storage, forms caching, and click-jacking. Inspired by DVWA, Mutillidae allows the user to change the "Security Level" from 0 (completely insecure) to 5 (secure). Additionally, three levels of hints are provided ranging from "Level 0 - I try harder" (no hints) to "Level 2 - noob" (Maximum hints). If the application is damaged by user injections and hacks, clicking the "Reset DB" button resets the application to its original state.

Note: Tutorials on using Mutillidae are available at the webpwnized YouTube Channel.

Enable hints in the application by click the "Toggle Hints" button on the menu bar:



The Mutillidae application contains at least the following vulnerabilities on these respective pages:

Page	Vulnerabilities
add-to-your-blog.php	SQL Injection on blog entry SQL Injection on logged in user name Cross site scripting on blog entry Cross site scripting on logged in user name Log injection on logged in user name CSRF JavaScript validation bypass XSS in the form title via logged in username The show-hints cookie can be changed by user to enable hints even though they are not suppose to show in secure mode

Page	Vulnerabilities
arbitrary-file-inclusion.php	System file compromise Load any page from any site
browser-info.php	XSS via referer HTTP header JS Injection via referer HTTP header XSS via user-agent string HTTP header
capture-data.php	XSS via any GET, POST, or Cookie
captured-data.php	XSS via any GET, POST, or Cookie
config.inc*	Contains unencrytped database credentials
credits.php	Unvalidated Redirects and Forwards
dns-lookup.php	Cross site scripting on the host/ip field O/S Command injection on the host/ip field This page writes to the log. SQLi and XSS on the log are possible GET for POST is possible because only reading POSTed variables is not enforced.
footer.php*	Cross site scripting via the HTTP_USER_AGENT HTTP header.
framing.php	Click-jacking
header.php*	XSS via logged in user name and signature The Setup/reset the DB menu item canbe enabled by setting the uid value of the cookie to 1
html5-storage.php	DOM injection on the add-key error message because the key entered is output into the error message without being encoded
index.php*	You can XSS the hints-enabled output in the menu because it takes input from the hints-enabled cookie value.

Page	Vulnerabilities
	You can SQL injection the UID cookie value because it is used to do a lookup You can change your rank to admin by altering the UID value HTTP Response Splitting via the logged in user name because it is used to create an HTTP Header This page is responsible for cache-control but fails to do so This page allows the X-Powered-By HTTP header HTML comments There are secret pages that if browsed to will redirect user to the phpinfo.php page. This can be done via brute forcing
log-visit.php	SQL injection and XSS via referer HTTP header SQL injection and XSS via user-agent string
login.php	Authentication bypass SQL injection via the username field and password field SQL injection via the username field and password field XSS via username field JavaScript validation bypass
password-generator.php	JavaScript injection
pen-test-tool-lookup.php	JSON injection
phpinfo.php	This page gives away the PHP server configuration Application path disclosure Platform path disclosure

Page	Vulnerabilities
process-commands.php	Creates cookies but does not make them HTML only
process-login-attempt.php	Same as login.php. This is the action page.
redirectandlog.php	Same as credits.php. This is the action page
register.php	SQL injection and XSS via the username, signature and password field
rene-magritte.php	Click-jacking
robots.txt	Contains directories that are supposed to be private
secret-administrative-pages.php	This page gives hints about how to discover the server configuration
set-background-color.php	Cascading style sheet injection and XSS via the color field
show-log.php	Denial of Service if you fill up the log XSS via the hostname, client IP, browser HTTP header, Referer HTTP header, and date fields
site-footer-xss-discusson.php	XSS via the user agent string HTTP header
source-viewer.php	Loading of any arbitrary file including operating system files.
text-file-viewer.php	Loading of any arbitrary web page on the Interet or locally including the sites password files. Phishing
user-info.php	SQL injection to dump all usernames and passwords via the username field or the password field XSS via any of the displayed fields. Inject the XSS on the register.php page. XSS via the username field
user-poll.php	Parameter pollution

Page	Vulnerabilities
	GET for POST XSS via the choice parameter Cross site request forgery to force user choice
view-someones-blog.php	XSS via any of the displayed fields. They are input on the add to your blog page.

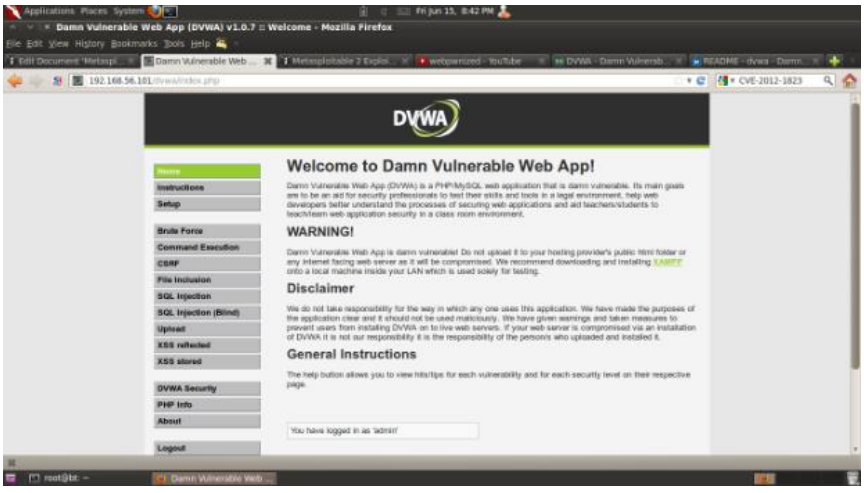
Vulnerable Web Services: DVWA

From the DVWA homepage: "Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a classroom environment."

DVWA contains instructions on the home page, and additional information is available at Wiki Pages - Damn Vulnerable Web App

Default username = admin

Default password = password



Vulnerable Web Services: Information Disclosure

Additionally, an ill-advised PHP information disclosure page can be found at http://<IP>/phpinfo.php. In this example, the URL would be http://192.168.56.101/phpinfo.php.

Your URL url will differ! It's the IP address of your Metasploitable2 machine.

The PHP info information disclosure vulnerability provides internal system information and service version information that can be used to look up vulnerabilities. For example, noting that the version of PHP disclosed in the screenshot is version 5.2.4, it may be possible that the system is vulnerable to CVE -CVE-2012-1823 and CVE -CVE-2012-2311 which affected PHP before 5.3.12 and 5.4.x before 5.4.2.

End of the lab!