Network Penetration Testing Methodology-Internal

6 Hr 36 Min Remaining

Instructions Resources Help  100%

Exercise 18: Adding an Exploit to Metasploit

Scenario

In this lab, you will

• Take an exploit from the exploit database and add it to Metasploit

**Lab Duration**: **5** Minutes

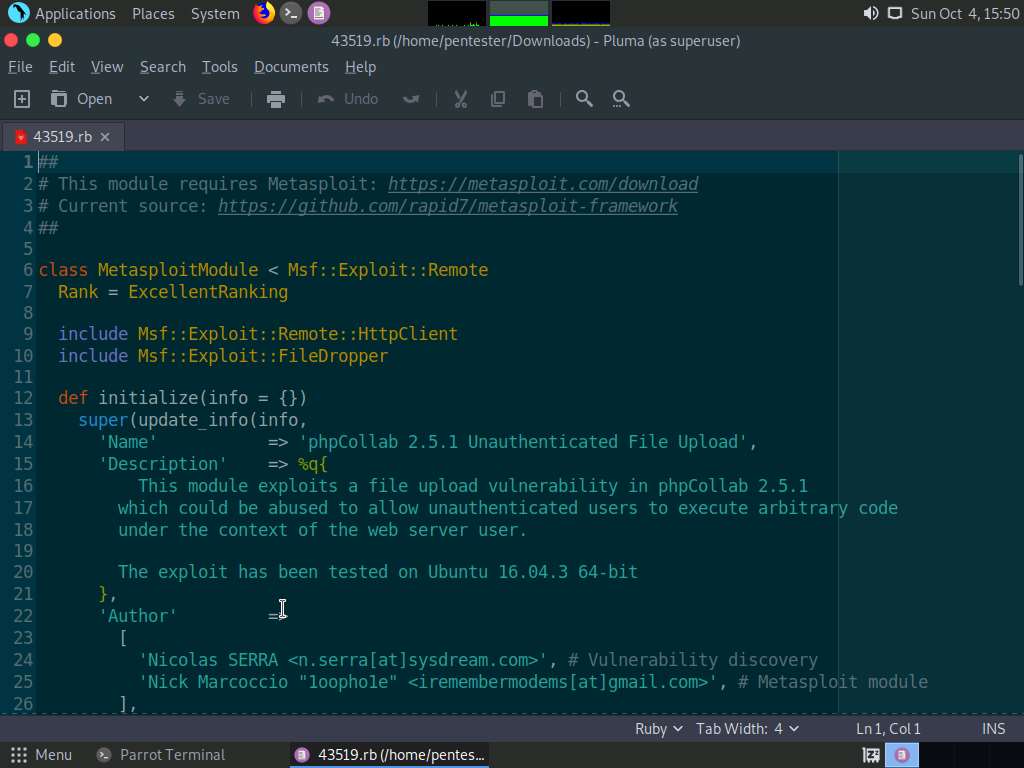
1. Click [Parrot](https://labclient.labondemand.com/Instructions/52f4d542-434e-4a10-8f51-0c2b8ca1d32b?rc=10). Parrot lock screen appears.



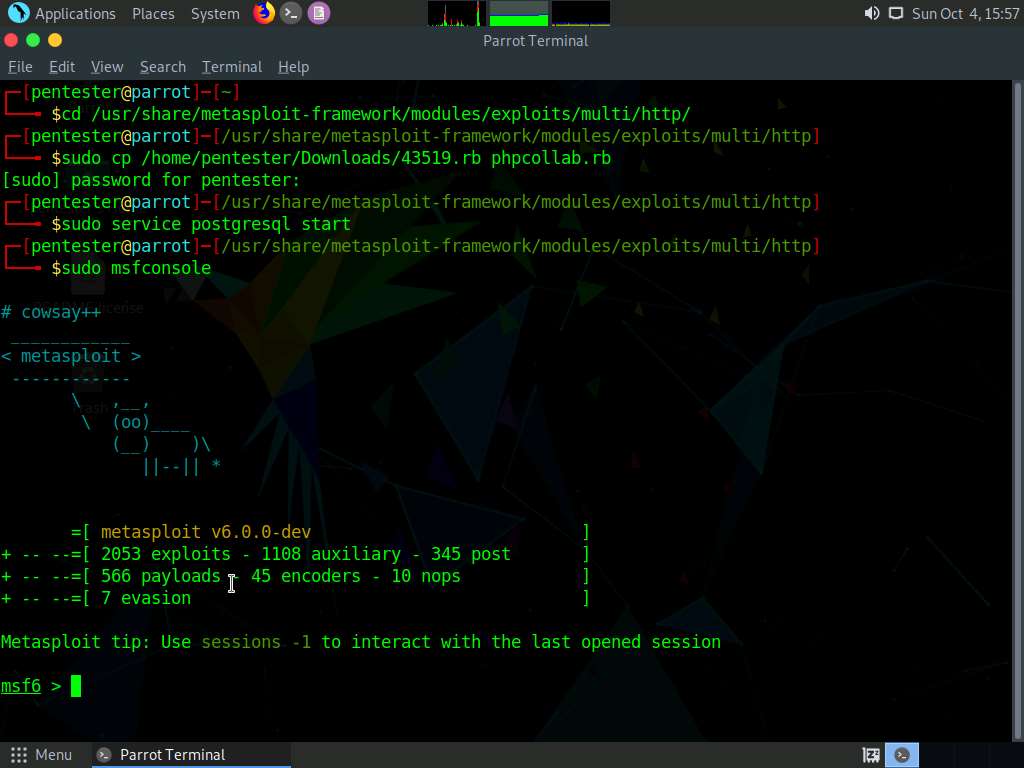
1. By default **pentester** is selected as the **user**. Type **toor** in the Password field and press **Enter**.



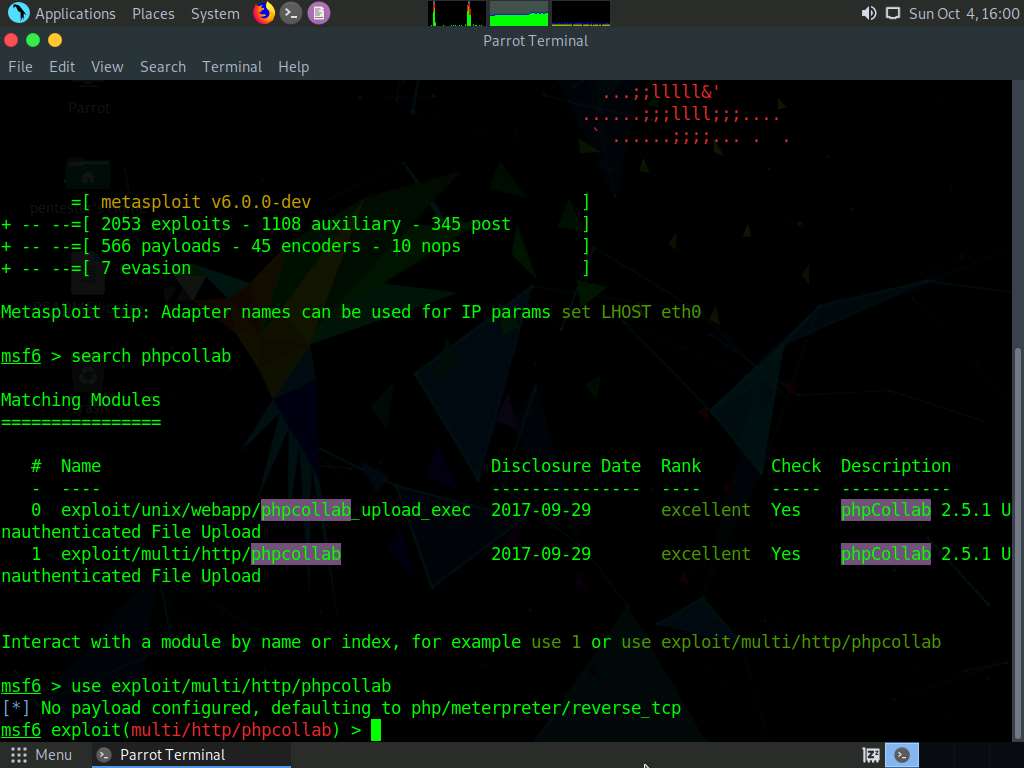
1. In the Parrot machine, open a terminal window.
2. There are times when a new exploit will be added to the exploit database, or you might even want to build and customize one on your own. You will work through this process.
3. A Ruby exploit is downloaded in the **Downloads** folder in the Parrot machine. Work with the **43519** script file; this is the **phpCollab 2.5.1 - Unauthenticated File Upload exploit** that has a CVE number of **CVE-2017-6090**, published **1-11-2018**.
4. Open a terminal window, type **sudo pluma /home/pentester/Downloads/43519.rb**, and press **Enter**. If you are asked to enter a password, enter **toor**.
5. This command will open the file in the gedit text editor window.
6. The **43519.rb** file opens in the text editor window.



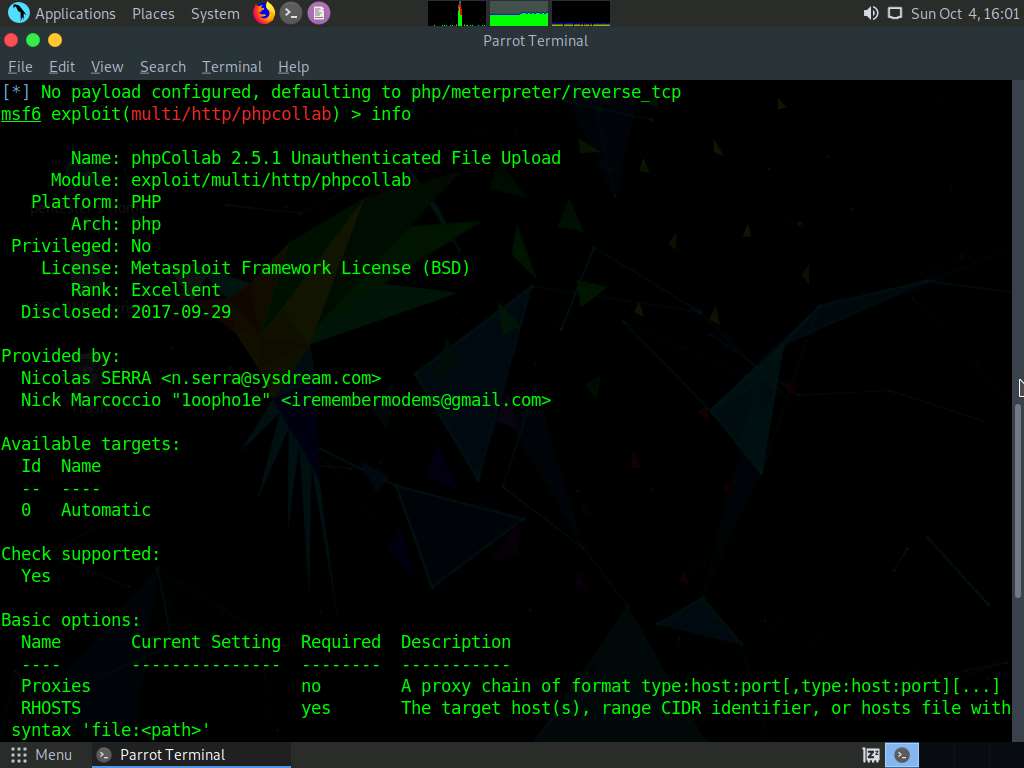
1. This is the structure of Metasploit module and contains the information that you can view with the info command in the modules. Carefully review the code within the module. Note that you need to vet all scripts before using them in the testing. If you are new to Ruby, please do your research on the details of running these modules against clients who are contracting you as a professional penetration tester (Please see the appendix for writing Ruby scripts).
2. Close the text editor window after your review.
3. Next, copy the module by entering the Metasploit directory. Then, copy the code into it; type **cd usr/share/metasploit-framework/modules/exploits/multi/http** and press **Enter**.
4. This will place you in the directory that contains the HTTP exploit modules. Since you are working with PHP, this is a good location to place it.
5. Copy the code with the following command **sudo cp /home/pentester/Downloads/43519.rb phpcollab.rb** and press **Enter**. If you are asked to enter a password, enter **toor**. This will copy the exploit to the directory and rename it.
6. Type **sudo service postgresql start** and press **Enter**.
7. Type **sudo msfconsole** and press **Enter**. Then, msfconsole appears; note the number of the exploits.



1. Type **reload\_all** and press **Enter**.
2. Find the added exploit by searching for it; type **search phpcollab** and press **Enter**.
3. You should now see your exploit. You have just updated exploits in Metasploit; type **use exploit/multi/http/phpcollab** and press **Enter**.



1. Once you are in the exploit, type **info** and press **Enter** to read about the exploit.



1. Once you have finished reading, type **show options** and press **Enter**. Review the options that are available.
2. To view the Targets type **show targets** and press **Enter**. In this case, you only have one target.
3. You will be able to find more information about this application. phpCollab is an open-source Internet-enabled collaboration workspace for project teams. Modeled on Macromedia Sitespring, phpCollab's architecture allows for the consulting team to share information with each other in one space and publish that information, when desired, to another space for the client. phpCollab encompasses the most important aspects of project management such as task planning and document sharing. It hooks into other open source applications such as Mantis for bug tracking as well as PhPNuke (using phpCollabPublisher) for content management and ongoing project support. phpCollab's community, part of the larger sourceforge open source software community, is highly active, with a dedicated volunteer team of developers, testers, and documenters constantly enhancing the application to meet user needs.
4. You could setup a lab, and then test your exploit as well.
5. This concludes the lab exercise.