|  |
| --- |
| **Scaffold: *Intrusion Testing*** |
| **Test 1: Phishing test**  To complete this test, you will need to input the user passwords, and store and display them in a variable. The skills to complete this are covered in this lesson: <https://digisoln.com/python/console/inputoutput>  **Test 2: Common password test**  There are multiple ways to complete this test, however there the obvious method is to 'hardcode' each username and password to test. So, testing passwords: "qwerty", "abc123", and "password" against three user accounts: "joe", "zoe" and "lee" would require 9 lines of code. In this case, use the previous link for help (<https://digisoln.com/python/console/inputoutput>).  The more eloquent solution to this test is to use a smaller loop within a bigger loop (iteration). There is an example here: <https://digisoln.com/python/console/iterationnested> that might help.  **Test 3: Brute force pin test**  There are too many PINs to code here, so you will need to use a simple loop if you are going to have any success with this. There is a sample here you can follow: <https://digisoln.com/python/console/iterationwhile>  **Test 4: Avoiding detection and compromise test**  The simplest method to complete this test is to generate 5 random numbers <https://digisoln.com/python/console/randomnumbers> and then use an if statement to check if the numbers are less than the value specified (<https://digisoln.com/python/console/branchingif>). Finally, sum the integer amounts (<https://digisoln.com/python/console/datatypes>).  The more eloquent to this solution is to generate the random numbers within a loop. This will assist you to scale the number of iterations (which is an extension activity).  **Test 5: Network Discovery**  This test seems easy, but it is actually tricky. Use a loop <https://digisoln.com/python/console/iterationwhile> to iterate over the IP addresses, and declare an integer variable for each quantity of network devices, to track how many workstations, printers etc. you have assigned. There are more eloquent solutions (perhaps) to this test. |