

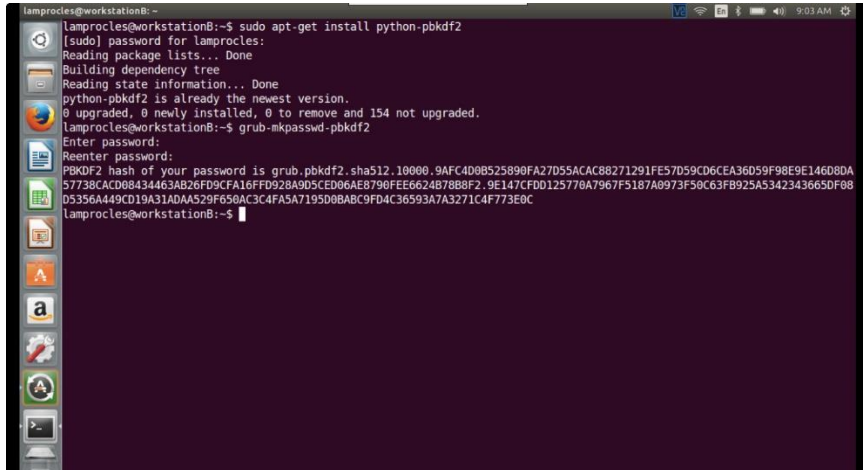
Real-World Scenario 14-1: Key Stretching

Scenario: You are working as a security administrator alongside your organization's development team. You are in charge of implementing key stretching whenever there is weak encryption.

Examine the figure below.

What is being encrypted here?

How many iterations (or rounds) were performed during the encryption process?



```
lamprocles@workstationB: ~  
lamprocles@workstationB:~$ sudo apt-get install python-pbkdf2  
[sudo] password for lamprocles:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python-pbkdf2 is already the newest version.  
0 upgraded, 0 newly installed, 0 to remove and 154 not upgraded.  
lamprocles@workstationB:~$ grub-mkpasswd-pbkdf2  
Enter password:  
Reenter password:  
PBKDF2 hash of your password is grub.pbkdf2.sha512.10000.9AFC4D0B525890FA27D55ACAC88271291FE57D59CD6CEA36D59F98E9E146D8DA  
57728CACD0843463AB28FD0CF416FFD928A905CED06AE8790FFE6624B78B8F2.9E147CFD0125770A7967F3187A0973F50C63FB925A5342343665DF08  
D5356A449CD19A31ADA529F650AC3C4FA5A7195D0BABC9FD4C36593A7A3271C4F773E0C  
lamprocles@workstationB:~$
```

Real-World Scenario 14-1 Solution

I created a password for the GRUB bootloader in Linux and encrypted it using PBKDF2. This utilized the SHA512 cryptographic hash, which ran for 10,000 rounds (iterations). The amount of rounds can be modified, but be careful when doing so because it might increase computational time considerably. Finally, be sure to test any key stretching technology before putting a system into production.

The video solution below describes key stretching, salting, and PBKDF2, and then demonstrates how the GRUB bootloader password in the figure was encrypted.

Video Solution: Watch the video solution “14-1: Introduction to Key Stretching.”