Spring-2017

Information Security and Assurance
Assignment-1

Submitted by:

Moulika Chadalavada

16234180

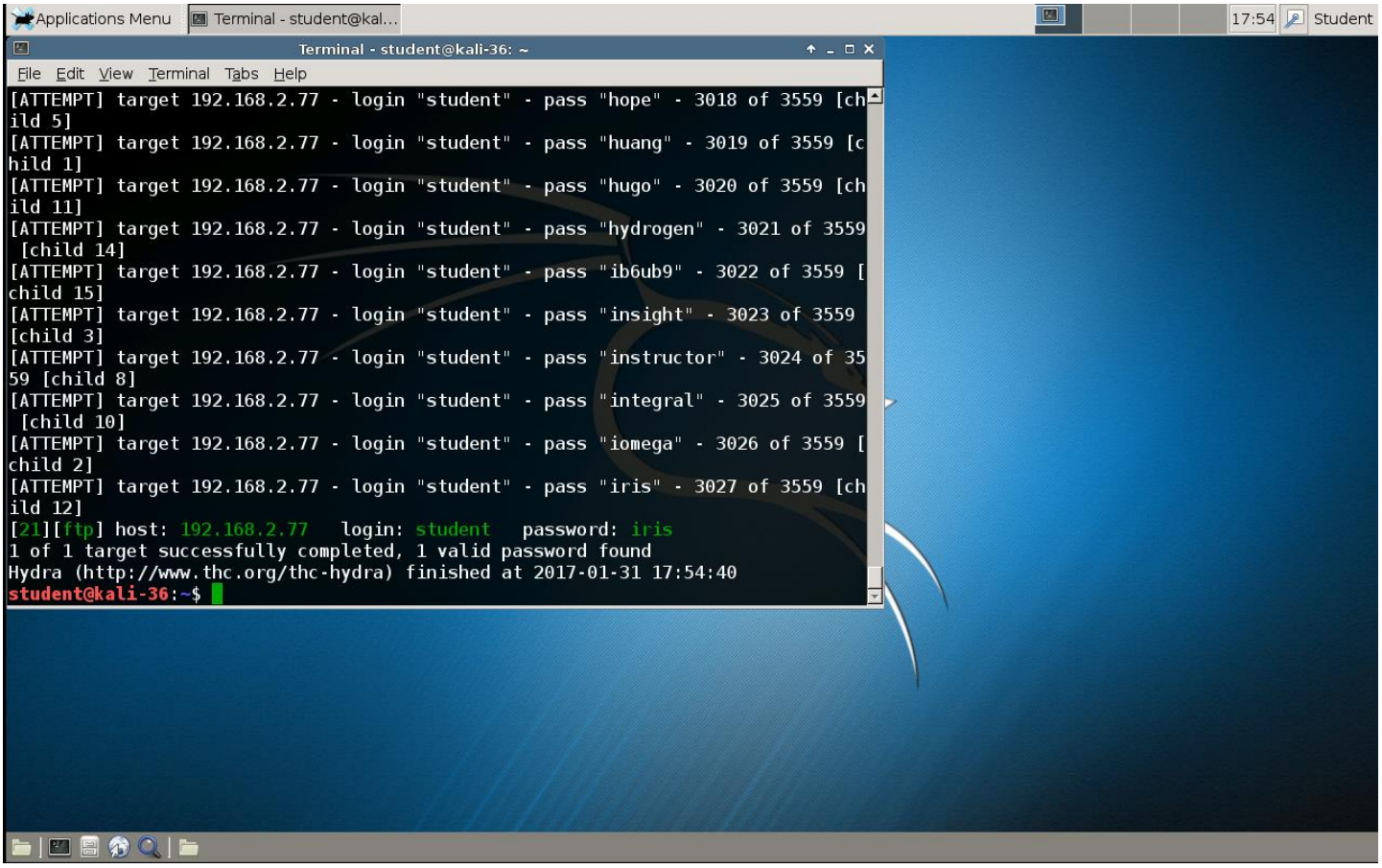
Password cracking using dictionary attack

Use the below command to crack the password of a user login account 'student' on the server 192.168.2.77 using dictionary attack.

Command: *hydra –l student –V –P /usr/share/john/password.lst* ftp://192.168.2.77

1. What is the password of the login account 'student'? (1 point)

Ans: The password of login account student is **iris**



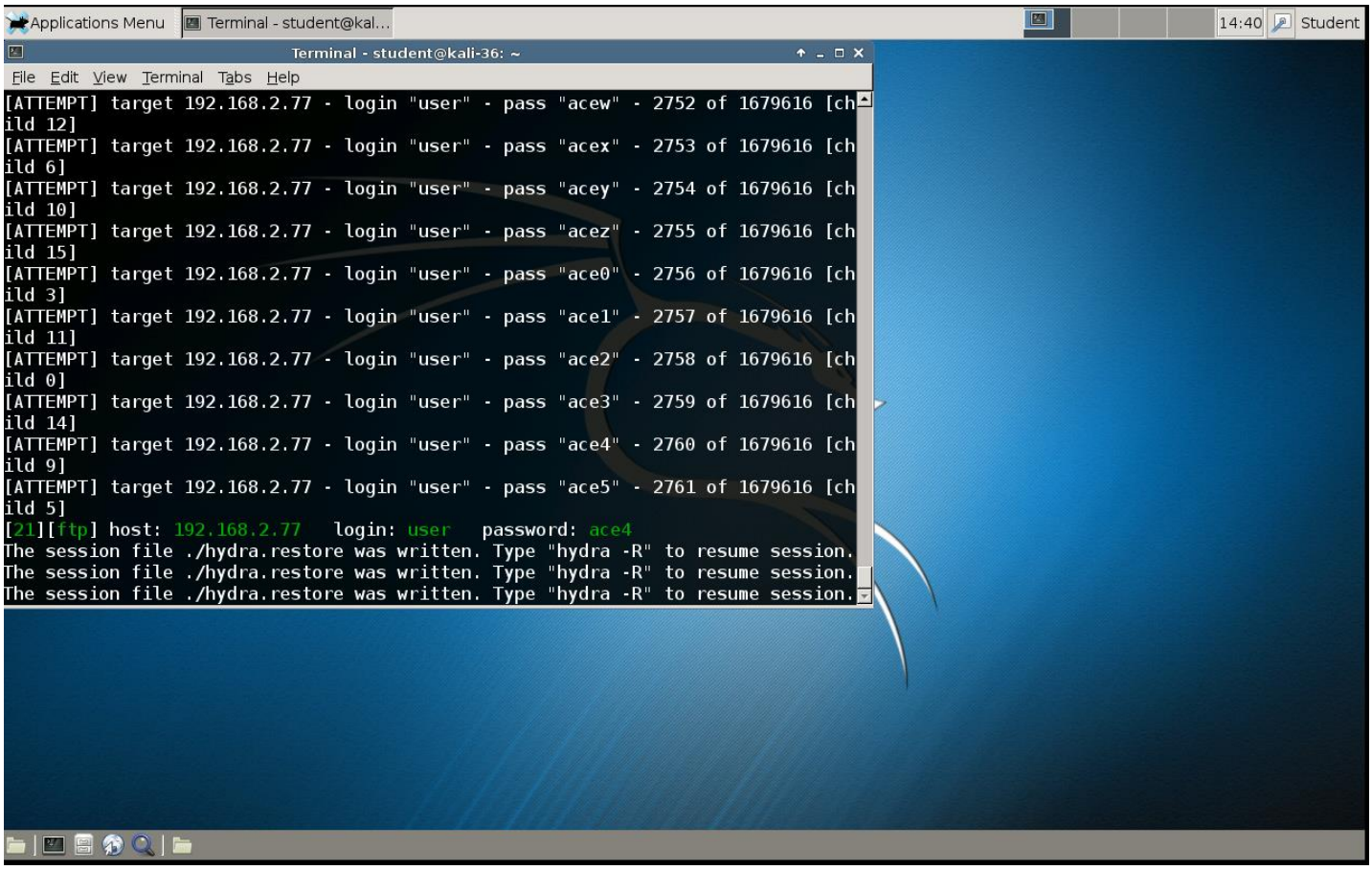
Password cracking using bruteforce attack

Use the below command to crack the password of a user login account ‘user’ on the server 192.168.2.77 using bruteforce attack.

Command: *hydra –l user –V –x 4:4:a1 ftp://192.168.2.77*

2. What is the password of the login account ‘user? (1 point)

Ans: The password of login account user is **ace4**



3. Explain the process of hashing and storing the passwords in Ubuntu/Linux operating systems? (1 point)

Ans:

The primary mechanism to get access to Linux machine is by having user account with corresponding password of that account. The passwords of all users in a system must be saved in file or database in encoded format, so user authentication can be done during user login. The encoded values are generated using hashing mechanism, which are not only used for storing passwords but also used to check data integrity. But by using dictionary attack against encoded values it is easy to detect real passwords. In any of the systems like Linux/Ubuntu, by using advanced computing attacker can try with millions of combinations in less time. Even though passwords are encoded, if attacker gets password file then it's easy to break password.

In Unix systems passwords are stored in a file called ***/etc/password*** but major loophole in this file is that it is readable by the user. This file is kept so because apart from passwords it contains critical user information. Many applications and tools need this information for proper functionality. To overcome this, loophole the passwords should be separated and has to be kept in file which is accessible only by root. This can be achieved in form of package called ***shadow-utils.***

shadow-utils is package which is installed in Linux by default for separating user passwords from /etc/password file. After implementing shadow-utils all the passwords are save in file ***/etc/shadow***. Unlike /etc/password, /etc/shadow file has read permissions only for root user which is not accessible by any other user. Along with encoded passwords /etc/shadow stores advanced features. The encoded hash value contains 3 different fields.

1. Numeric value that tells which algorithm is used in hashing
   $1: MD5, $2: Blowfish, $2a: eksblowfish, $5: SHA-256, $6: SHA-512
2. Salt Value : random data that is generated to increase strength of hash value
3. Hash Value of Salt + User_Password

For example 1Etg2ExUZ$F9NTP7omafhKIlqaBMqng1 is a has value where **$1** represents MD5 hash function , **Etg2ExUZ** is salt value and **F9NTP7omafhKIlqaBMqng1** is hash value of Salt + User Password.

4. Explain the process of hashing and storing the passwords in Windows 7 and Windows 10 operating systems? (1 point)

Ans:

A user identity is authenticated using a secret passphrase in any of the operating systems. To secure our network strong password is used which avoids threat of guessing weak passwords either by using manual methods or by using any tools. If we change password regularly the malicious attack is reduced to an extent.

In Windows, passwords are stored in many ways. The two different ways in which password stored by default is **LM OWF and NT OWF** for Windows networking. **One way function (OWF)** means one way mathematical transformation of data, in which transformed data can be converted only through encryption. One of the common one-way function is cryptographic hash.

In Windows LM OWF algorithm is used for software and hardware's backward compatibility. NT OWF is just a hash where password is hashed using MD5 algorithm and is also stored. This algorithm is used for authentication in Windows and its Active Directory Domains. Neither NT OWF nor LM OWF is salted. Salting is process that combines password with random numeric value before applying one-way function.

So, when user logs on, the password the is entered by user is converted into LM and NT one way functions. By using **Local Security Authority Subsystem Service** (LSASS) process it is stored in memory. If the user is accessing already stored account for authentication, NT OWF algorithm is compared with locally stored NT hash value, if these two values matches then the user can log in successfully. If user uses host name against Active Directory Domain to access resource, NT hash used against **Key Distribution Center** (KDC) in Kerberos logon. Password verifier computer via WINLOGON but not LSASS.

References:

https://technet.microsoft.com/en-us/library/hh994558(v=ws.10).aspx