Information Security and Assurance
Assignment - 1

Submitted by:

Anusha Malineni

16233382

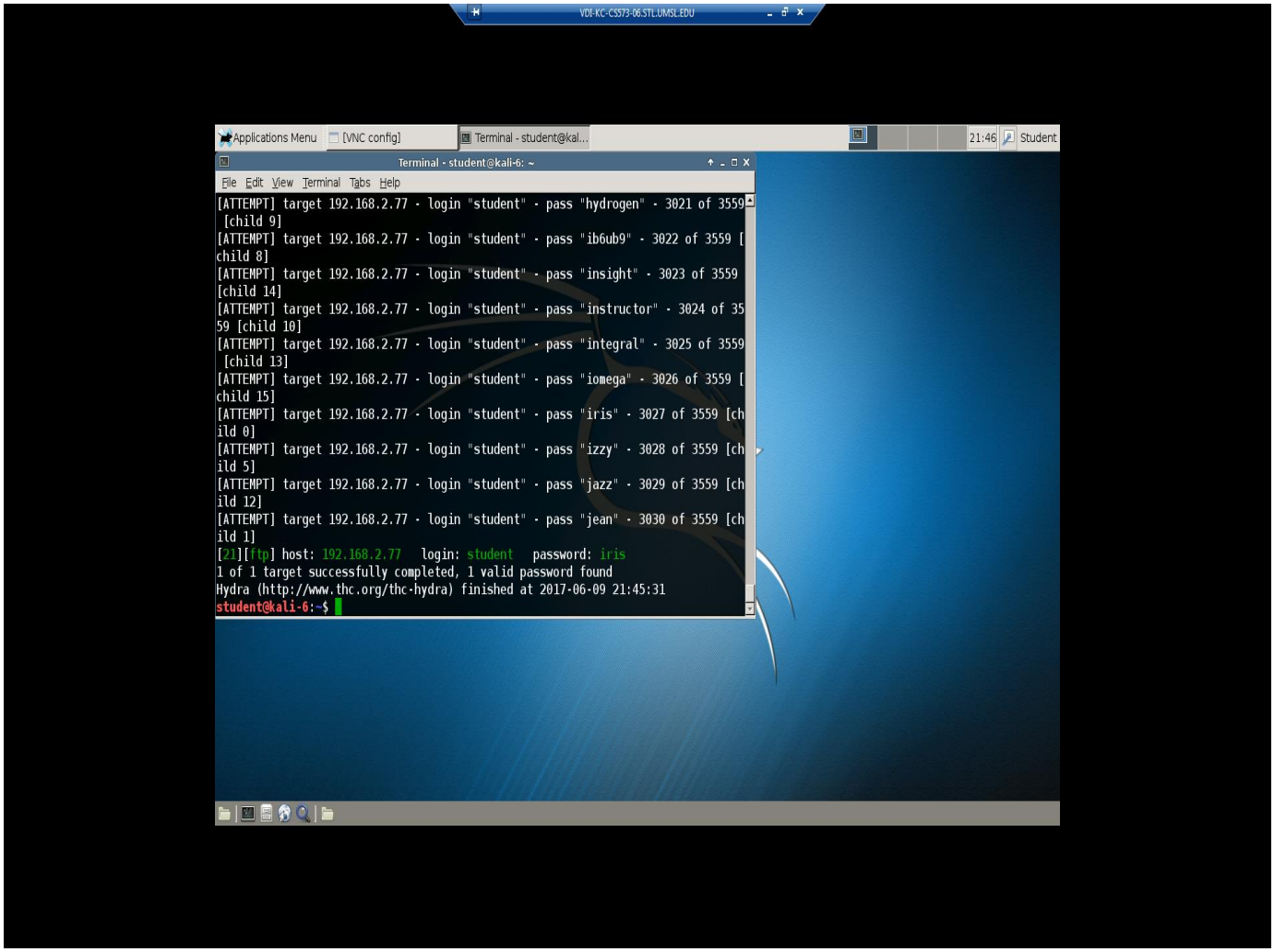
Password cracking using dictionary attack

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 the 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 is **XXXX.**

3. Explain the process of hashing and storing the passwords in Ubuntu/Linux and Windows 7 operating systems. (1 point)

Ans:

In Ubuntu/Linux operating systems:

A mechanism to have access to Linux machine is to have a user account with corresponding password for that account. User authentication is done when user logs in as the passwords of all users in a system are saved in a file or database in encoded format. The encoded values are not only used for storing passwords but also for data integrity and these are generated using a hashing mechanism. But it is easy to detect exact passwords by using dictionary attack against the encoded values. In Ubuntu/Linux systems, even though passwords are encoded, it's easy to break password if the attacker gets the password file. By using advanced computing techniques, attacker can try for many combinations in less time.

In Unix systems, passwords are stored in **/etc/password** file but the disadvantage is file is readable by the user. It is readable since it has user's critical information other than password. We have many tools and applications which depend on this information of user for functionality. In order to overcome this disadvantage, all the passwords should be separated and stored in a separate file which should be accessible only by root. **Shadow-utils** package is helpful in achieving this.

The shadow-utils package is default installed by Linux for separating passwords from the /etc/password file. All the passwords are saved in /etc/shadow file after implementing the package. The /etc/password file has read permissions only for root user. In /etc/shadow file, other than encoded passwords, some advanced features are stored. There are three different fields in the encoded hash value.

1. In the first field, we have the numeric value which helps in identifying which algorithm is used for hashing.

(a) $1: MD5
(b) $2: Blowfish
(c) $2a: eksblowfish
(d) $5: SHA-256
(e) $6: SHA-512

2. The second field is the salt value, this is a random data that is generated to increase the strength of hash value.

3. The third field is Hash value of salt + user-password.

For instance, let's consider an example **$1$4hx$XLjr/pkLpIjbwhrxmuQyb** is a hash value where **$1** represents MD5 hash algorithm, **$4hx** is a salt value and **$XLjr/pkLpIjbwhrxmuQyb** is a hash value of salt and user-password.

In Windows7 operating system:

Generally, in any operating system a user identity is authenticated by a secret passphrase. Strong passwords are used to avoid threat of guessing weak passwords either by any tools or by using any manual methods in order to secure the network. To reduce malicious attack, passwords are changed frequently.

Passwords are stored in many ways in windows operating systems. There are two different ways in which password can be stored i.e., **LM OWF** and **NT OWF** for windows networking. OWF is a One way function which is defined as one way mathematical transformation of data where in the transformed data is converted through encryption. One of the common OWF is cryptographic hash.

For software and hardware backward compatibility in windows LM OWF algorithm is used. Whereas NT OWF is just a hash value where password is hashed using MD5 algorithm and stored. Neither LM OWF nor NT OWF is salted. Salting is a process that combines password with random numeric value before applying OWF.

When user logs in, password given by user is first converted into LM and NT OWF's. It is stored in memory by using Local Security Authority Subsystem Service(LSASS). NT OWF algorithm compares with locally stored NT hash value, if the user access already stored account for authentication. The user can login if the two values are matched successfully. If the user uses host name against Active Directory Domain to access resource, NT hash used against Key Distribution Centre in Kerberos logon.

4. **Mention the salt value and hashed password value of the user account 'student' of your Kali Linux system. (Open the shadow file in your kali linux and locate the username 'student' and provide the above details)**

Ans:

References:

1) http://www.slashroot.in/how-are-passwords-stored-linux-understanding-hashing-shadow-utils
2) https://technet.microsoft.com/en-us/library/hh994558(v=ws.10).aspx
3) https://blog.aujas.com/2015/02/13/shadows-utils-linux-password-storage/