11.7.2 Password Attack Facts

Passwords are quite often the main defense against unauthorized access to computer systems and sensitive data. This makes passwords a prime target for attackers. A variety of attack methods have been developed to retrieve passwords.

This lesson covers the following topics:

- Social engineering
- Brute force attacks
- Rainbow attacks
- Cracking passwords using rainbow tables

Social Engineering

Social engineering is the art of manipulation. In most networks, the weakest link is the human element. Hackers can take advantage of this to gain access to sensitive information, including passwords.

The following table explains some social engineering techniques to be aware of and protect against.

Social Engineering Technique	Description

Password guessing	Password guessing is usually not a very efficient method to crack a password. An attacker may first attempt to use default login information, such as admin/admin or simple passwords like password123.
	If these don't work, the attacker can use publicly available information, such as on a target's social media, to make the process easier. Information such as the following can be used to guess a password or answer security questions and reset a user's password:
	▪ Birthday
	▪ First car
	▪ Family information
	  ▪ Spouse's name
	  ▪ Child's name
	  ▪ Important dates
	▪ Important locations

User manipulation	A common social engineering technique is user manipulation. This involves the attacker interacting with the user to trick the user into revealing the username and password. For example, the attacker may call the target pretending to be from tech support with a urgent problem. The attacker asks for the target's login information to remote in to resolve the issue. User manipulation is a very successful technique and is still used quite often. User training is the best prevention method.
Physical access	An attacker can use social engineering to gain physical access to an office building. Once inside, the attacker can look around for login information that users have written down. Many users have a tendency to write login information on sticky notes and stick the notes on the monitor or place them under the mouse pad.

Dumpster diving	An attacker may dumpster dive (go through the trash) to find important documents or information that has been thrown out. Many users will throw out papers without realizing the importance of the information. Documents should always be shredded to prevent data loss due to dumpster diving.
Shoulder surfing	Shoulder surfing is an eavesdropping technique in which the attacker obtains passwords or other confidential information by looking over the shoulder of a user typing a password.

User education is the best defense against any form of social engineering. Users should be trained that no one will ever ask for their login information and to always be aware of their surroundings.

Brute Force Attacks

In a brute force attack, the attacker attempts to guess the password by using a cracking tool that submits every possible letter, number, and symbol combination in a short amount of time. A brute force password attack can be a very time-consuming attack.

The following table describes some of the brute force attack methods.

Brute Force Attack Method	Description
Online attack	An online brute force attack requires the attacker to submit the passwords using the same user login interface while the target is up and running. For example: ▪ An attacker targeting a website will submit login attempts to the site interface. ▪ An attacker targeting a computer will submit login attempts to the login screen. The best defense against this method is to implement lock out policies. This means if the incorrect password is entered multiple times in a short period of time, the account will be locked for a specified amount of time.
Offline attack	Offline attacks require the attacker to somehow steal the password file. The attacker can then run attacks against that file with no limitations, such as lock out policies. This is the ideal method for the attacker, but is more difficult because it requires the attacker to somehow steal the password file.
Password spraying	Password spraying is another method that allows the attacker to avoid lock out policies. ▪ Instead of attempting multiple logins using a single user account and different passwords, the attacker will use the same password with multiple user accounts. ▪ The attacker will continue cycling through the user accounts submitting passwords until a match is found. ▪ Because there is a delay between submitting a password on each account, the lock out policy can be avoided.
Dictionary attack	In a dictionary attack, the hacker uses a list of words and phrases to try to guess the password. ▪ Dictionary attacks work well if weak passwords are used.

	▪ Using longer and uncommon passphrases is the best way to secure data against these attacks

Some common password cracking tools that can be used to carry out brute force attacks are:

- John the Ripper
- Hashcat
- Medusa
- Cain and Abel

Implementing proper password protocols is the best defense against password cracking attempts. A strong password should:

- Be at least 8 characters; more is better.
- Contain upper and lower case letters.
- Contain numbers.
- Contain symbols.
- Not use common words or phrases.

A passphrase is the best option to use instead of a password.

Rainbow Attacks

When a plaintext password is stored, it is encrypted and a hash is generated.

Rainbow attacks are similar to dictionary attacks, but instead of trying to match the words and phrases, a rainbow attack uses special tables called *rainbow tables* that are already filled with common passwords and their generated hashes. The attacker uses this table to match the hashes instead of the password itself. Rainbow attacks require less computing power and are much faster than brute force attacks.

Storing rainbow tables requires a lot of storage. A single rainbow table can range anywhere from 30GB to over 300GB. The character set (lower and/or upper case letters, numbers, symbols) being used will greatly increase the size. A different rainbow table needs to be generated for each encryption algorithm.

The best defense against rainbow attacks is *salting* the hashes. Salting the hash means that random characters are added at the beginning or end of the password. This generates a completely different hash. The login server is programmed to identify the part of the hash that is salted, but anyone intercepting the hash will have no idea; so, the hash can't be decrypted.

Cracking Passwords Using Rainbow Tables

An encrypted plaintext password stored in a hash file can be cracked using rainbow tables. There are several types of programs that can be used to create and crack these types of passwords, such as:

- Rtgen
- Winrtgen
- RainbowCrack
- Ophcrack

As an example, the following table lists a few examples of the commands needed to create and sort a rainbow crack table:

Command	Description

rtgen	This command generates a rainbow table based on the parameters specified by the user. The parameters are: rtgen ***hash_algorithm charset plaintext_len_min plaintext_len_max table_index chain_len chain_num part_index***
	
	Example: **rtgen md5 ascii-32-95 1 7 0 1000 1000 0**
	
	▪ *hash_algorithm* - A hashing algorithm is a mathematical algorithm which can convert an input data array of a certain type and arbitrary length to an output bit string of a fixed length. Rainbow table must be generated for the type of hash algorithm used. Although there are may hash algorithms that can be use, some of the more common are; ntlm, md5, and sha1.
	▪ *charset* - A charset specifies all the possible characters for the plaintext. Some of the possible charset that can be used include:
	  ▪ Numeric = [0123456789]
	  ▪ alpha = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
	  ▪ alpha-numeric = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
	  ▪ loweralpha = [abcdefghijklmnopqrstuvwxyz]
	  ▪ loweralpha-numeric = [abcdefghijklmnopqrstuvwxyz0123456789]
	  ▪ ascii-32-95 = ascii-32-95 = [!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_'abcdefghijklmngpqrstuvwxyz{|}~]
	▪ *plaintext_len_min* and *plaintext_len_max* - These two values, such as 1 7, specifies length of the plaintext.
	
	The next four parameters are advanced values and are beyond the scope of this lesson. Therefore, only a brief explanation is given here:
	
	▪ *table_index* - Specifies the reduction function. Examples are: 0, 1, 2, 3, 4. Zero is often used as the default.
	▪ *chain_len* - This specifies the rainbow chain length.
	▪ *chain_num* - This specifies the number of rainbow chains to generate.
	▪ *part_index* - The number of files used to store the rainbow table. If a value greater than zero is use, the rainbow table is saved in the number of smaller files specified by the value.
	
	As shown in the example above, common values for these four parameters are: **0 1000 1000 0**

rtsort	A rainbow table is an array of rainbow chains. Each rainbow chain has a start point and an end point. The rtsort program sorts the rainbow chains by end point to make binary search possible. To sort a rainbow table, use the following command (the period at the end is part of the command): **rtsort .**

After the rainbow table has been created, you are now ready to crack the passwords. This can be done using the **rcrack** command.

The **rcrack** syntax is: **rcrack** *path parameter*

The following table lists a few examples of how the **rcrack** command can be used:

Command	Description
rcrack . -l /root/hashes.txt	The **-l** parameter loads the hashes from a file and each hash is shown on its own line. The hash is shown followed by the cracked password. Example output: plaintext of 590cb9bZaC590/5b9b4b0/152d2321117 P@ssw0rd plaintext of 400238780e6c41f8f790161e6ed4aafc21 Test_Out@11_Last plaintext of 89BF04763BF91C9EE2DDBE23D735C73OBDD41FF2 NeverLAnd5
rcrack . -h *hash_value*	The **-h** parameter loads and displays the results for a single hash. Example command: **rcrack . -h 590cb9bZaC590/5b9b4b0/152d2321117** Example output: **590cb9bZaC590/5b9b4b0/152d232111a P@ssw0rd hex:444387**

Copyright © 2022 TestOut Corporation All rights reserved.