Password Hashing and Cracking Report

Task 01: Determining the Hash Type

In this task, I used the **Hash Lookup Analyzer** tool to identify the hash types of the following three hashes and determine the corresponding cleartext password.

Hashes:

1. 5366a90275a8fc64acbaab5d697e6d94
2. a9f44ff05b323527ca3a496c791faf050a1239c6
3. 462b0be44f0b7738bc83ac62ad132911e97a5cb6f60ffe6fc161f2f5acbb2ce7

Hash Types Identified:

- **Hash 1:** MD5
- **Hash 2:** SHA1
- **Hash 3:** SHA-256

After running the hashes through the **Hash Lookup Analyzer**, I discovered that the cleartext password for all three hashes was:

- **Cleartext Password:** ilovecit

```
a9f44ff05b323527ca3a496c791faf050a1239c6:ilovecit
```

Task 02: Password Cracking

For this task, I was given a collection of 15 hashed passwords from a security breach. My goal was to determine if these passwords were easily crackable using tools like **John the Ripper** or **Hashcat**, utilizing the **rockyou** wordlist.

List of Hashes:

1. b830bec01190a2a812c95fb31628b63ae933fd93
2. 5c58c6b2cf12b733b02b8d29e386d4119aac6db6
3. 2d0507eba5cb4d4523afd9c594db71152198467d
4. 2ab33fd22007e8d65d944c5d4e7ff84ba9c4a05e
5. 3a9cdab8ea9e98f0d997401fa3a1d73dce4e98e8
6. 3ceaeffd8a62e5c67fa99544b70770c86d8ba12f
7. 04602ae9d6f95db81251caaa2163c9e71ce1ef39
8. 2e9dcf9475e1a4560fd5927865bb013f5d02abe1
9. 2b9b3cf4a0a03662e7662dcc7476f1d65c62635a
10. 1833cffe02b4dd3e5130e5aa0e9b8abb82c08f59
11. 5f89fc9b6033aa210658489a14bf865a096dd702
12. f5c1f93238d45415a6313f3c7e233a6dcf45311e
13. 6c9709fce7b1b2dd11f8e424b868a8bfff44342f
14. 33cab4f0593def9f66c7a172c719c040a44701a3
15. 042a5a9bab02a1be32581db9013aabad43413a06

Using **John the Ripper** and **Hashcat**, I was able to crack several passwords from the list.

```
┌──(kali㉿kali)-[~]
└─$ hashcat -m 100  /home/kali/Desktop/TestHash /usr/share/wordlists/rockyou.txt.gz --show

5f89fc9b6033aa210658489a14bf865a096dd702:advent
b830bec01190a2a812c95fb31628b63ae933fd93:stainless
2e9dcf9475e1a4560fd5927865bb013f5d02abe1:wolfhound
3ceaeffd8a62e5c67fa99544b70770c86d8ba12f:jokeri
2d0507eba5cb4d4523afd9c594db71152198467d:provocation
f5c1f93238d45415a6313f3c7e233a6dcf45311e:spore
```

```
┌──(kali㉿kali)-[~]
└─$ hashcat -m 100  /home/kali/Desktop/TestHash /usr/share/wordlists/rockyou.txt.gz -r /usr/share/hashcat/rules/rockyou-30000.rule
```

```
2b9b3cf4a0a03662e7662dcc7476f1d65c62635a:cons.
```

```
33cab4f0593def9f66c7a172c719c040a44701a3:anja1988
```

```
5c58c6b2cf12b733b02b8d29e386d4119aac6db6:advantaged
```

```
6c9709fce7b1b2dd11f8e424b868a8bfff44342f:mystery66
```

Here are the cracked passwords:

Cracked Hashes:

1. **b830bec01190a2a812c95fb31628b63ae933fd93** – stainless
2. **5f89fc9b6033aa210658489a14bf865a096dd702** – advent
3. **5c58c6b2cf12b733b02b8d29e386d4119aac6db6** - advantaged
4. **2d0507eba5cb4d4523afd9c594db71152198467d** - provocation
5. **f5c1f93238d45415a6313f3c7e233a6dcf45311e** - spore
6. **6c9709fce7b1b2dd11f8e424b868a8bfff44342f** – mystery66
7. **3ceaeffd8a62e5c67fa99544b70770c86d8ba12f** - jokeri
8. **33cab4f0593def9f66c7a172c719c040a44701a3** – anja1988
9. **2e9dcf9475e1a4560fd5927865bb013f5d02abe1** - wolfhound
10. **2b9b3cf4a0a03662e7662dcc7476f1d65c62635a** – cons.

The remaining hashes were either not cracked within the time frame or required advanced cracking methods.

Task 03: Generating Custom Lists with Crunch and CeWL

For this task, I generated two custom wordlists using the **Crunch** and **CeWL** tools.

Using Crunch:

The task required generating a wordlist for 8-character passwords with the following pattern: four numbers followed by the word "cit" and a special character (e.g., 1234cit?).

The command I ran to generate the list was:

crunch 8 8 -t %%%%cit^ -0 crunchwordlist.txt

Here's a snippet of the generated wordlist:

```
6167cit%
6167cit^
6167cit&
6167cit*
6167cit(
6167cit)
6167cit-
6167cit_
6167cit+
6167cit=
6167cit~
6167cit`
6167cit[
6167cit]
6167cit{
6167cit}
6167cit|
6167cit\
6167cit:
6167cit;
6167cit"
6167cit'
6167cit<
6167cit>
6167cit,
6167cit.
6167cit?
6167cit/
6167cit
6168cit!
6168cit@
6168cit#
6168cit$
6168cit%
6168cit^
6168cit&
6168cit*
```

Using CeWL:

Next, I crawled the website livlab.org to a depth of 1 and generated a wordlist with terms relevant to the site.

The command I used was:

```
┌──(kali㉿kali)-[~]
└─$ cewl http://livlab.org -d 1 -w cewlwordlist.txt
CeWL 6.1 (Max Length) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
```

Here's a snippet of the generated wordlist:

Open

cewlwordlist.txt

```
and
the
Lab
Living
testimonials
for
The
entry
experience
widget
has
with
content
post
work
been
have
also
site
skills
that
learned
network
students
page
wrapper
you
meta
project
your
this
Search
from
projects
will
was
technical
```

Task 04: Calculating Brute Force Time

In this task, I was asked to calculate how long it would take to brute-force a 9-character NTDS password using all 95 different character types (A-Z, a-z, 0-9, special characters) and six GPUs.

Keyspace Calculation:

The available charset is 95 characters, and for a 9-character password, the keyspace is:

Keyspace=959=630,249,409,724,609,375\text{Keyspace} = 95^9 = 630,249,409,724,609,375Keyspace=959=630,249,409,724,609,375

Cracking Time in Seconds:

The hash rate of a single Nvidia RTX 3090 GPU is 121.8 GH/s. Using six GPUs, the total hash rate is calculated as follows:

Total Hash Rate = 121.8 GH/s × 6 = 730,800,000,000 hashes per second

Using the formula for cracking time:

Cracking Time = Keyspace / Total Hash Rate = 630,249,409,724,609,375 / 730,800,000,000 ≈ 862,410 seconds

Cracking Time in Weeks:

To convert the cracking time from seconds to weeks, we use the following calculation:

Cracking Time (in weeks) = 862,410 seconds / 604,800 seconds per week ≈ 1.43 weeks

Conclusion

Through the tasks in this report, I explored different aspects of password hashing and cracking. I successfully identified hash types, cracked several hashed passwords, and generated custom wordlists to assist in cracking. I also estimated the time it would take to brute-force a password using multiple GPUs, providing an understanding of the scale involved in password cracking.