# Nishant Baruah

# A053

# 70022100516

## Experiment 5 – Implement brute force attack by using dictionary

### What is Brute force attack

A brute force attack is a hacking method that uses trial and error to crack passwords, login credentials, and encryption keys. It is a simple yet reliable tactic for gaining unauthorized access to individual accounts and organizations' systems and networks. The hacker tries multiple usernames and passwords, often using a computer to test a wide range of combinations, until they find the correct login information.

The name "brute force" comes from attackers using excessively forceful attempts to gain access to user accounts. Despite being an old cyberattack method, brute force attacks are tried and tested and remain a popular tactic with hackers.

### Types of Brute Force Attacks

1. Simple Brute Force Attacks

A simple brute force attack occurs when a hacker attempts to guess a user's login credentials manually without using any software. This is typically through standard password combinations or personal identification number (PIN) codes.

These attacks are simple because many people still use weak passwords, such as "password123" or "1234," or practice poor password etiquette, such as using the same password for multiple websites. Passwords can also be guessed by hackers that do minimal reconnaissance work to crack an individual's potential password, such as the name of their favorite sports team.

2. Dictionary Attacks

A dictionary attack is a basic form of brute force hacking in which the attacker selects a target, then tests possible passwords against that individual's username. The attack method itself is not technically considered a brute force attack, but it can play an important role in a bad actor's password-cracking process.

The name "dictionary attack" comes from hackers running through dictionaries and amending words with special characters and numbers. This type of attack is typically time-consuming and has a low chance of success compared to newer, more effective attack methods.

3. Hybrid Brute Force Attacks

A hybrid brute force attack is when a hacker combines a dictionary attack method with a simple brute force attack. It begins with the hacker knowing a username, then carrying out a dictionary attack and simple brute force methods to discover an account login combination.

The attacker starts with a list of potential words, then experiments with character, letter, and number combinations to find the correct password. This approach allows hackers to discover passwords that combine common or popular words with numbers, years, or random characters, such as "SanDiego123" or "Rover2020."

4. Reverse Brute Force Attacks

A reverse brute force attack sees an attacker begin the process with a known password, which is typically discovered through a network breach. They use that password to search for a matching login credential using lists of millions of usernames. Attackers may also use a commonly used weak password, such as "Password123," to search through a database of usernames for a match.

5. Credential Stuffing

Credential stuffing preys on users' weak password etiquettes. Attackers collect username and password combinations they have stolen, which they then test on other websites to see if they can gain access to additional user accounts. This approach is successful if people use the same username and password combination or reuse passwords for various accounts and social media profiles.

## Brute Force Attack Tools

Guessing a user's email or social media website password can be a time-consuming process, especially if the accounts have strong passwords. To simplify the process, hackers have developed software and tools to help them crack passwords.

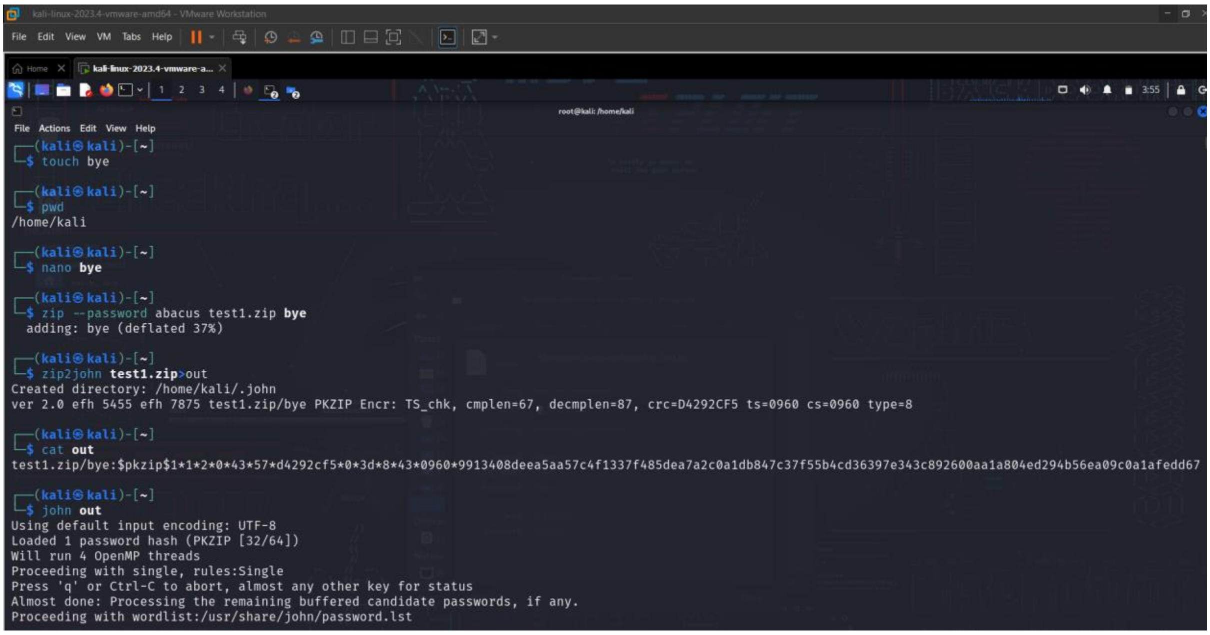
Brute force attack tools include password-cracking applications, which crack username and password combinations that would be extremely difficult for a person to crack on their own. Commonly used brute force attack tools include:

1. Aircrack-ng: A suite of tools that assess Wi-Fi network security to monitor and export data and attack an organization through methods like fake access points and packet injection.
2. John the Ripper: An open-source password recovery tool that supports hundreds of cipher and hash types, including user passwords for macOS, Unix, and Windows, database servers, web applications, network traffic, encrypted private keys, and document files.

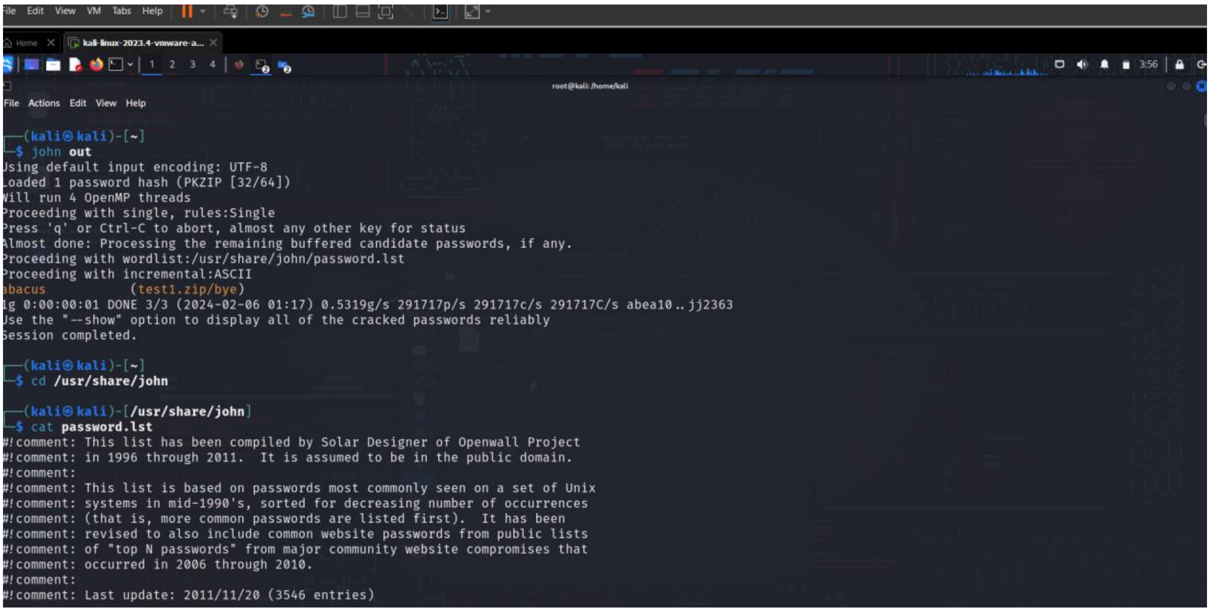
These types of software can rapidly guess combinations that identify weak passwords and crack multiple computer protocols, wireless modems, and encrypted storage devices.

A brute force attack can also demand huge amounts of computing power. To combat that, hackers have developed hardware solutions that simplify the process, such as combining a device's central processing unit (CPU) and graphics processing unit (GPU). Adding the computing core of the GPU enables a system to process several tasks simultaneously and the hackers to crack passwords significantly faster.

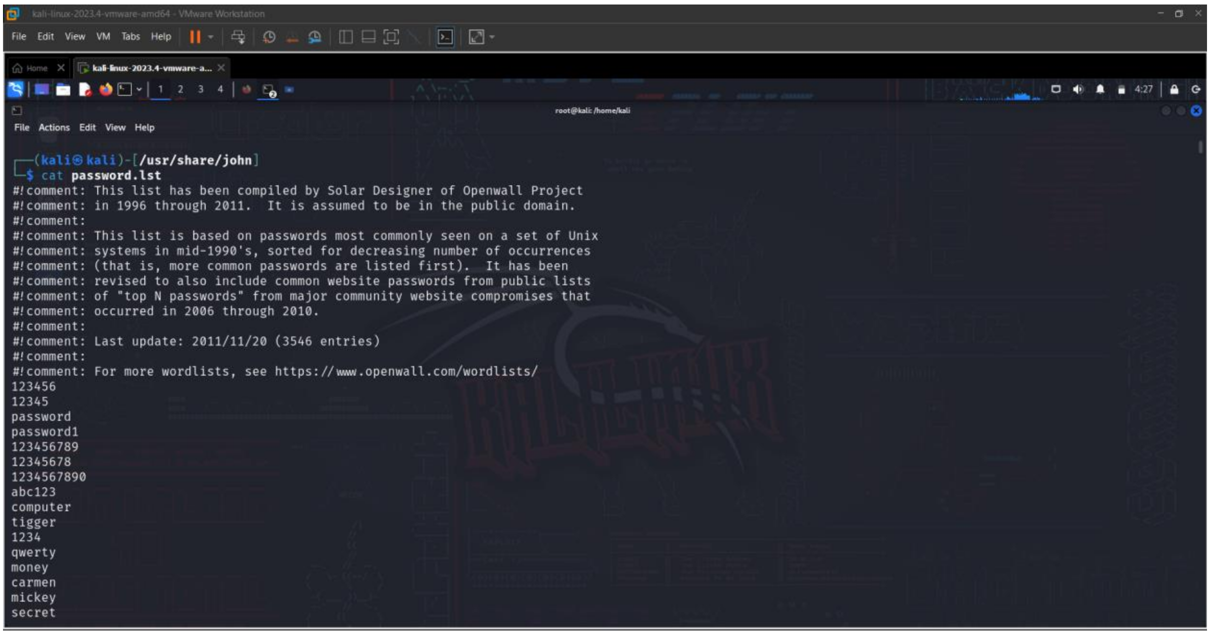
Implementation:


kali-linux-2023.4-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
root@kali: /home/kali
File Actions Edit View Help

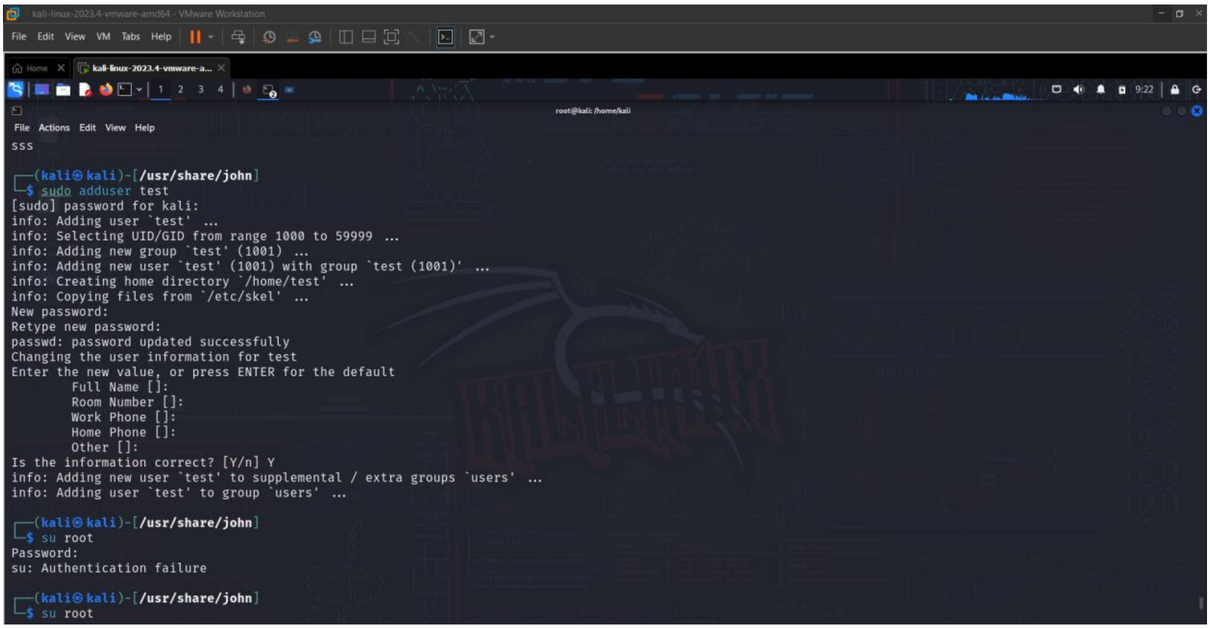
```
┌──(kali㉿kali)-[~]
└─$ touch bye

┌──(kali㉿kali)-[~]
└─$ pwd
/home/kali

┌──(kali㉿kali)-[~]
└─$ nano bye

┌──(kali㉿kali)-[~]
└─$ zip --password abacus test1.zip bye
  adding: bye (deflated 37%)

┌──(kali㉿kali)-[~]
└─$ zip2john test1.zip>out
Created directory: /home/kali/.john
ver 2.0 efh 5455 efh 7875 test1.zip/bye PKZIP Encr: TS_chk, cmplen=67, decmplen=87, crc=D4292CF5 ts=0960 cs=0960 type=8

┌──(kali㉿kali)-[~]
└─$ cat out
test1.zip/bye:$pkzip$1*1*2*0*43*57*d4292cf5*0*3d*8*43*0960*9913408deea5aa57c4f1337f485dea7a2c0a1db847c37f55b4cd36397e343c892600aa1a804ed294b56ea09c0a1afedd67

┌──(kali㉿kali)-[~]
└─$ john out
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
```



```
┌──(kali㉿kali)-[~]
└─$ john out
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:ASCII
abacus           (test1.zip/bye)
1g 0:00:00:01 DONE 3/3 (2024-02-06 01:17) 0.5319g/s 291717p/s 291717c/s 291717C/s abea10..jj2363
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~]
└─$ cd /usr/share/john

┌──(kali㉿kali)-[/usr/share/john]
└─$ cat password.lst
#!comment: This list has been compiled by Solar Designer of Openwall Project
#!comment: in 1996 through 2011.  It is assumed to be in the public domain.
#!comment:
#!comment: This list is based on passwords most commonly seen on a set of Unix
#!comment: systems in mid-1990's, sorted for decreasing number of occurrences
#!comment: (that is, more common passwords are listed first).  It has been
#!comment: revised to also include common website passwords from public lists
#!comment: of "top N passwords" from major community website compromises that
#!comment: occurred in 2006 through 2010.
#!comment:
#!comment: Last update: 2011/11/20 (3546 entries)
```

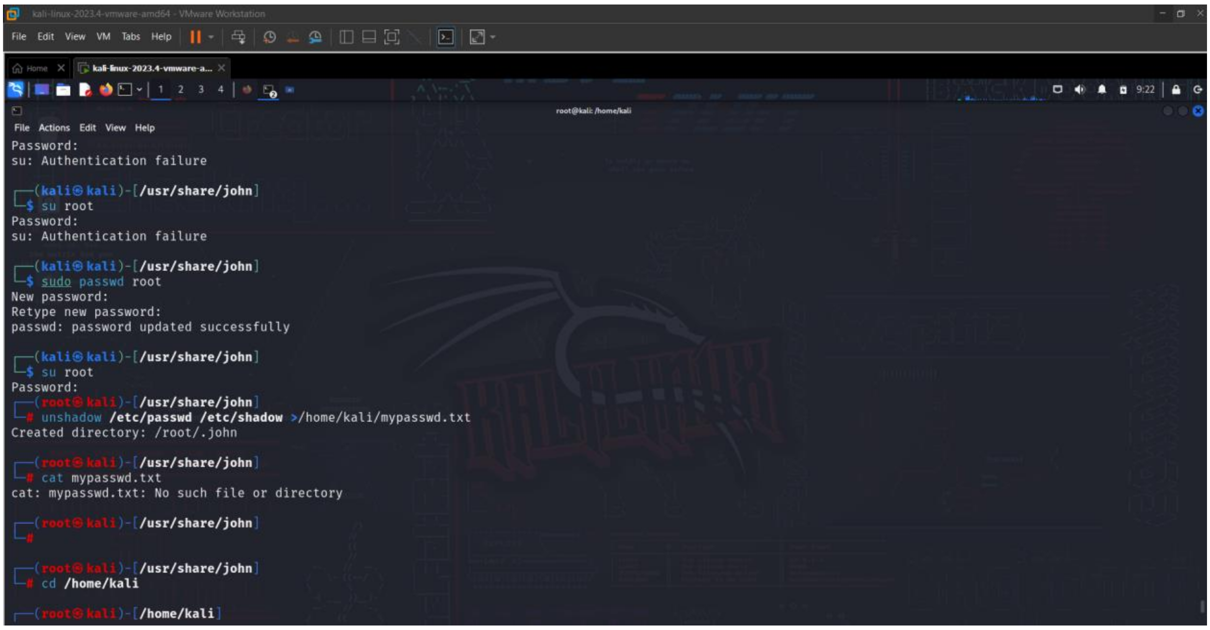

kali-linux-2023.4-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
root@kali: /home/kali
File Actions Edit View Help

```
┌──(kali㉿kali)-[/usr/share/john]
└─$ cat password.lst
#!comment: This list has been compiled by Solar Designer of Openwall Project
#!comment: in 1996 through 2011.  It is assumed to be in the public domain.
#!comment:
#!comment: This list is based on passwords most commonly seen on a set of Unix
#!comment: systems in mid-1990's, sorted for decreasing number of occurrences
#!comment: (that is, more common passwords are listed first).  It has been
#!comment: revised to also include common website passwords from public lists
#!comment: of "top N passwords" from major community website compromises that
#!comment: occurred in 2006 through 2010.
#!comment:
#!comment: Last update: 2011/11/20 (3546 entries)
#!comment:
#!comment: For more wordlists, see https://www.openwall.com/wordlists/
123456
12345
password
password1
123456789
12345678
1234567890
abc123
computer
tigger
1234
qwerty
money
carmen
mickey
secret
```



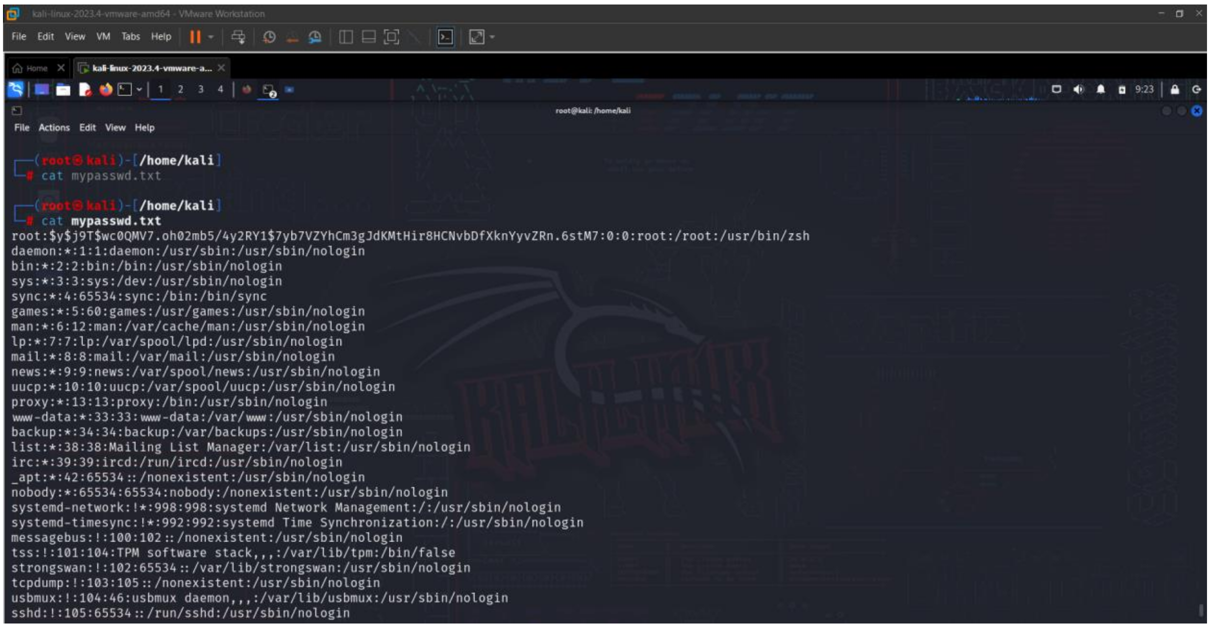
kali-linux-2023.4-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Home
kali-linux-2023.4-vmware-a...
9:22
root@kali: /home/kali
File Actions Edit View Help

```
sss

┌──(kali㉿kali)-[/usr/share/john]
└─$ sudo adduser test
[sudo] password for kali:
info: Adding user `test' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `test' (1001) ...
info: Adding new user `test' (1001) with group `test (1001)' ...
info: Creating home directory `/home/test' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for test
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
info: Adding new user `test' to supplemental / extra groups `users' ...
info: Adding user `test' to group `users' ...

┌──(kali㉿kali)-[/usr/share/john]
└─$ su root
Password:
su: Authentication failure

┌──(kali㉿kali)-[/usr/share/john]
└─$ su root
```



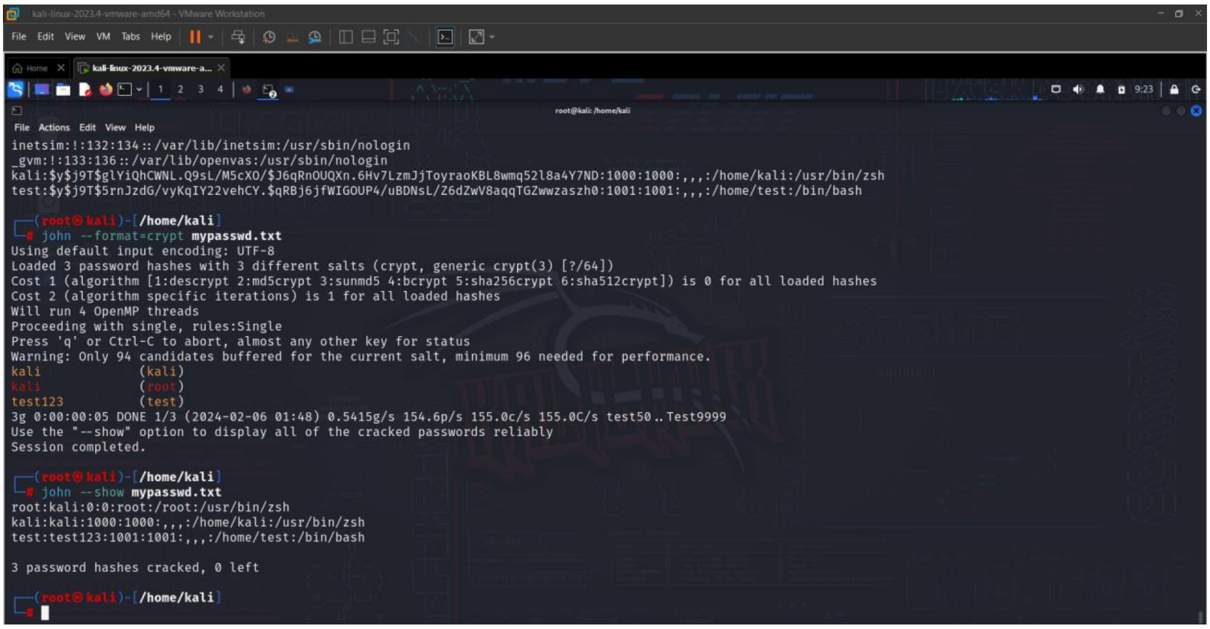
kali-linux-2023.4-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Home
kali-linux-2023.4-vmware-a...
9:22
root@kali: /home/kali
File Actions Edit View Help

```
Password:
su: Authentication failure

┌──(kali㉿kali)-[/usr/share/john]
└─$ su root
Password:
su: Authentication failure

┌──(kali㉿kali)-[/usr/share/john]
└─$ sudo passwd root
New password:
Retype new password:
passwd: password updated successfully

┌──(kali㉿kali)-[/usr/share/john]
└─$ su root
Password:
┌──(root㉿kali)-[/usr/share/john]
└─# unshadow /etc/passwd /etc/shadow >/home/kali/mypasswd.txt
Created directory: /root/.john

┌──(root㉿kali)-[/usr/share/john]
└─# cat mypasswd.txt
cat: mypasswd.txt: No such file or directory

┌──(root㉿kali)-[/usr/share/john]
└─#

┌──(root㉿kali)-[/usr/share/john]
└─# cd /home/kali

┌──(root㉿kali)-[/home/kali]
```



kali-linux-2023.4-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Home
kali-linux-2023.4-vmware-a...
9:23
root@kali: /home/kali
File Actions Edit View Help

```
┌──(root㉿kali)-[/home/kali]
└─# cat mypasswd.txt

┌──(root㉿kali)-[/home/kali]
└─# cat mypasswd.txt
root:$y$j9T$wc0QMV7.oh02mb5/4y2RY1$7yb7VZYhCm3gJdKMtHir8HCNvbDfXknYyvZRn.6stM7:0:0:root:/root:/usr/bin/zsh
daemon:*:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:*:2:2:bin:/bin:/usr/sbin/nologin
sys:*:3:3:sys:/dev:/usr/sbin/nologin
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/usr/sbin/nologin
man:*:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:*:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:*:8:8:mail:/var/mail:/usr/sbin/nologin
news:*:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:*:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:*:13:13:proxy:/bin:/usr/sbin/nologin
www-data:*:33:33:www-data:/var/www:/usr/sbin/nologin
backup:*:34:34:backup:/var/backups:/usr/sbin/nologin
list:*:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:*:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:*:42:65534::/nonexistent:/usr/sbin/nologin
nobody:*:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:!*:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:!*:992:992:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:!:100:102::/nonexistent:/usr/sbin/nologin
tss:!:101:104:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:!:102:65534::/var/lib/strongswan:/usr/sbin/nologin
tcpdump:!:103:105::/nonexistent:/usr/sbin/nologin
usbmux:!:104:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:!:105:65534::/run/sshd:/usr/sbin/nologin
```


kali-linux-2023.4-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Home
kali-linux-2023.4-vmware-a...
1 2 3 4
9:23
root@kali: /home/kali
File Actions Edit View Help

```
inetsim:!:132:134::/var/lib/inetsim:/usr/sbin/nologin
_gvm:!:133:136::/var/lib/openvas:/usr/sbin/nologin
kali:$y$j9T$glYiQhCWNL.Q9sL/M5cXO/$J6qRnOUQXn.6Hv7LzmJjToyraoKBL8wmq52l8a4Y7ND:1000:1000:,,,:/home/kali:/usr/bin/zsh
test:$y$j9T$5rnJzdG/vyKqIY22vehCY.$qRBj6jfWIGOUP4/uBDNsL/Z6dZwV8aqqTGZwwzaszh0:1001:1001:,,,:/home/test:/bin/bash

┌──(root㉿kali)-[/home/kali]
└─# john --format=crypt mypasswd.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 94 candidates buffered for the current salt, minimum 96 needed for performance.
kali             (kali)
kali             (root)
test123          (test)
3g 0:00:00:05 DONE 1/3 (2024-02-06 01:48) 0.5415g/s 154.6p/s 155.0c/s 155.0C/s test50..Test9999
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root㉿kali)-[/home/kali]
└─# john --show mypasswd.txt
root:kali:0:0:root:/root:/usr/bin/zsh
kali:kali:1000:1000:,,,:/home/kali:/usr/bin/zsh
test:test123:1001:1001:,,,:/home/test:/bin/bash

3 password hashes cracked, 0 left

┌──(root㉿kali)-[/home/kali]
└─# 
```

**Conclusion:**

In conclusion, the experiment involving the implementation of a brute force attack using a dictionary has provided valuable insights into the vulnerabilities of password security. The method demonstrated the effectiveness of systematically trying all possible combinations from a pre-built dictionary to gain unauthorized access. This underscores the importance of employing robust password policies, including the use of complex and unique passwords, as well as implementing additional security measures such as two-factor authentication. The findings of this experiment serve as a stark reminder for individuals and organizations to remain vigilant in their efforts to protect sensitive information and reinforce cybersecurity measures in an ever-evolving digital landscape.