

**Experiment No. 4**

**Title: Password Cracking Tools**



**(A Constituent College of Somaiya Vidyavihar University)**

**Roll No.: 16010423076****Experiments No.:4****Aim : To learn about password cracking tools**

---

**Resources : Internet access, web-browser, password cracking tools**

---

## **Theory:**

### **Introduction to Password Cracking**

Password cracking is the process of recovering passwords from stored data. It is used in cybersecurity to test the strength of passwords and improve security. When a user sets a password, it is not stored in its original form. Instead, it is converted into a scrambled version using a mathematical process called hashing. This scrambled version is called a hash. The main idea behind password cracking is to take a hash and find the original password that created it.

### **Types of Password Attacks**

There are different ways to crack passwords. One of the most common methods is the dictionary attack. In this method, a tool checks a long list of commonly used passwords one by one to see if any of them match the hash. If a weak password is used, this method can crack it in seconds. Another method is the brute-force attack, where every possible combination of letters, numbers, and symbols is tried until the correct password is found. This method is slow, but it works if enough time is given. A more advanced method is the rainbow table attack. This method uses precomputed tables of hashes and their matching passwords. Instead of calculating a hash for every possible password, the tool simply looks it up in the table, which makes the process much faster.

### **Common Password Cracking Tools**

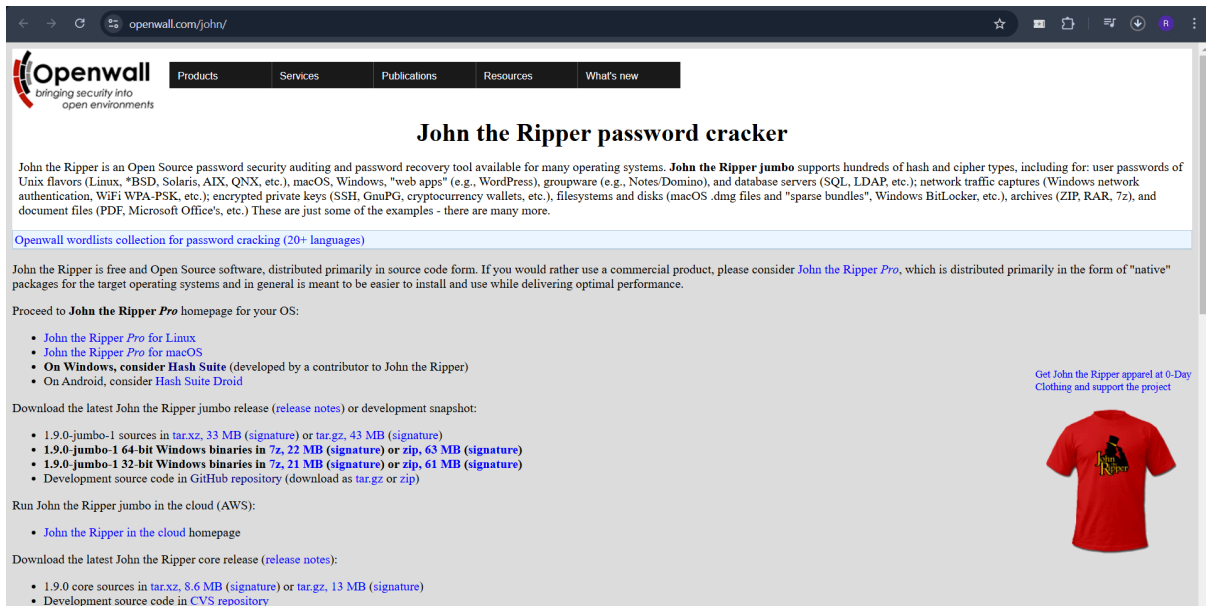
Password cracking tools are designed to automate these processes. John the Ripper is one of the most popular tools for this task. It works by taking a hash and testing passwords against it using different attack methods. It has a built-in list of common passwords, but users can also provide custom wordlists. Hashcat is another powerful tool. It can use the computer's processor and graphics card to speed up the cracking process. Hashcat supports many types of attacks, including dictionary and brute-force attacks. Another tool that was commonly used in the past is Cain and Abel, which could recover passwords stored on a Windows system. However, it is now outdated and not widely used.

## Importance of Strong Passwords

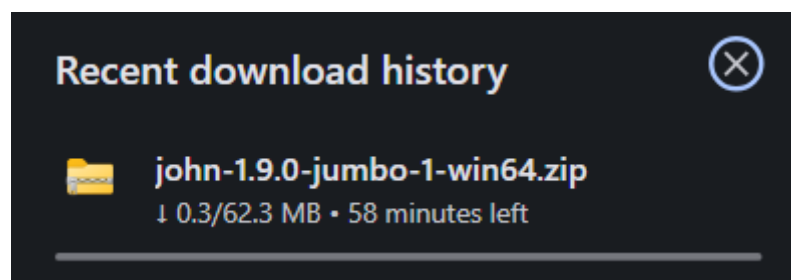
Strong passwords are difficult to crack. A password that is long and includes a mix of uppercase letters, lowercase letters, numbers, and symbols is much harder to break. Using unique passwords for different accounts also improves security. Many websites use salting to make password cracking harder. Salting means adding a random value to a password before hashing it. This prevents attackers from using precomputed tables like rainbow tables to find passwords easily.

## IMPLEMENTATION AND RESULTS:

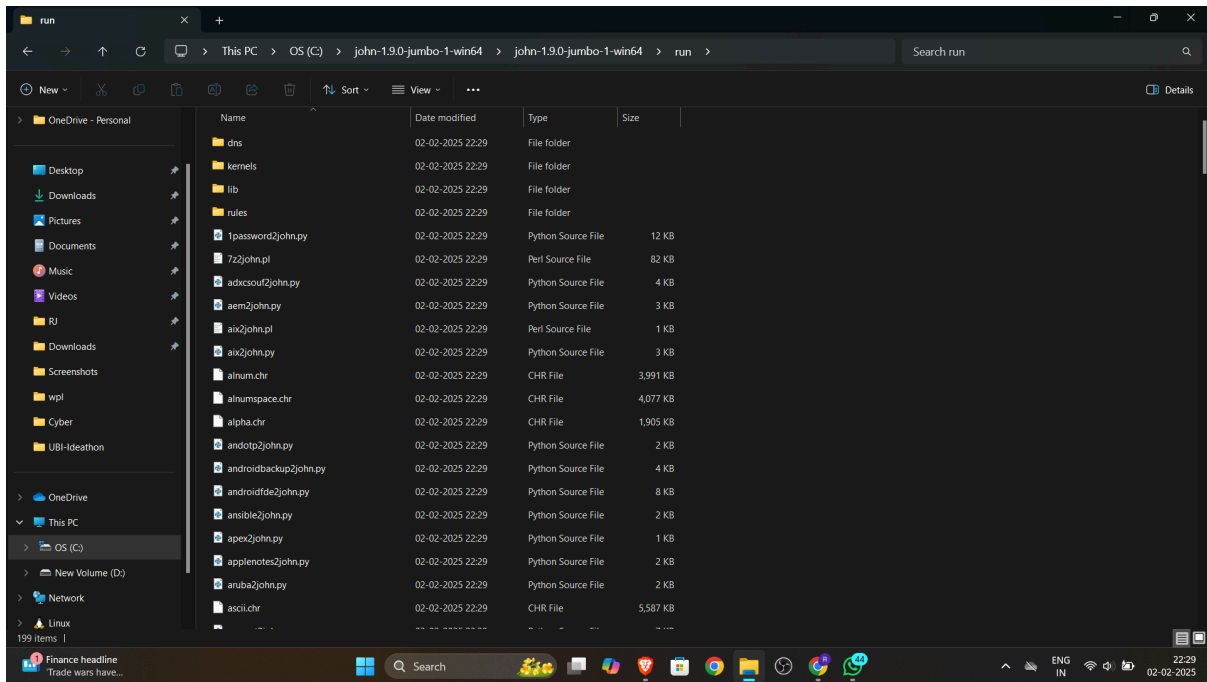
### Step 1 : Download and Install the Tool - John The Ripper



The screenshot shows the Openwall website for John the Ripper. The page title is "John the Ripper password cracker". The text describes it as an Open Source password security auditing and password recovery tool available for many operating systems. It lists supported hash and cipher types, including user passwords of Unix flavors (Linux, \*BSD, Solaris, AIX, QNX, etc.), macOS, Windows, "web apps" (e.g., WordPress), groupware (e.g., Notes/Domino), and database servers (SQL, LDAP, etc.); network traffic captures (Windows network authentication, WiFi WPA-PSK, etc.); encrypted private keys (SSH, GnuPG, cryptocurrency wallets, etc.); filesystems and disks (macOS .dmg files and "sparse bundles", Windows BitLocker, etc.); archives (ZIP, RAR, 7z), and document files (PDF, Microsoft Office's, etc.). It mentions that there are many more examples. A link to "Openwall wordlists collection for password cracking (20+ languages)" is provided. The text states that John the Ripper is free and Open Source software, distributed primarily in source code form. If a commercial product is preferred, "John the Ripper Pro" is suggested, which is distributed primarily in the form of "native" packages for the target operating systems and is generally easier to install and use while delivering optimal performance. A link to "John the Ripper Pro" is provided. Below this, it says "Proceed to John the Ripper Pro homepage for your OS:" and lists links for Linux, macOS, Windows (consider Hash Suite), and Android (consider Hash Suite Droid). It then says "Download the latest John the Ripper jumbo release (release notes) or development snapshot:" and lists links for 1.9.0-jumbo-1 sources in tar.xz (33 MB) or tar.gz (43 MB), 1.9.0-jumbo-1 64-bit Windows binaries in 7z (22 MB) or zip (63 MB), 1.9.0-jumbo-1 32-bit Windows binaries in 7z (21 MB) or zip (61 MB), and development source code in GitHub repository (download as tar.gz or zip). It then says "Run John the Ripper jumbo in the cloud (AWS):" and provides a link to "John the Ripper in the cloud" homepage. Finally, it says "Download the latest John the Ripper core release (release notes):" and lists links for 1.9.0 core sources in tar.xz (8.6 MB) or tar.gz (13 MB), and development source code in CVS repository. On the right side, there is a red t-shirt with "John the Ripper" text and a logo, with a link to "Get John the Ripper apparel at 0-Day Clothing and support the project".

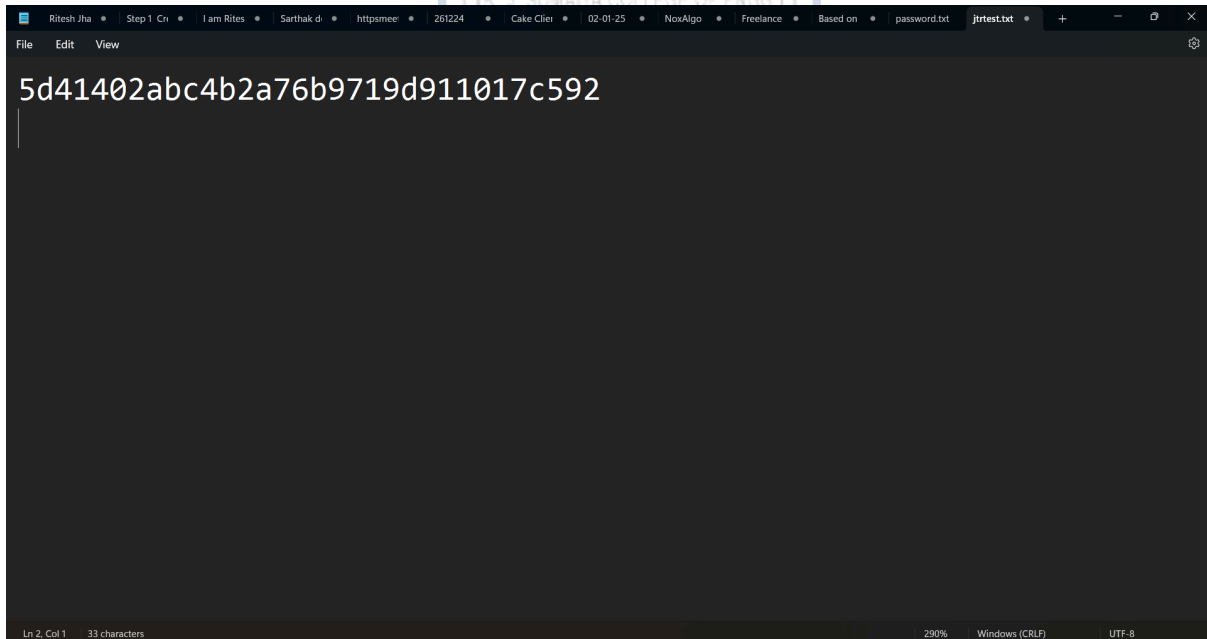


(A Constituent College of Somaiya Vidyavihar University)



## Step 2 : Prepare a Sample Hash File:

md5 hash for the word 'hello'



(A Constituent College of Somaiya Vidyavihar University)

### Step 3 : Run John the Ripper

Command : john

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22631.4830]
(c) Microsoft Corporation. All rights reserved.

C:\john-1.9.0-jumbo-1-win64>john-1.9.0-jumbo-1-win64\run-john
John the Ripper 1.9.0-jumbo-1-OMP [cypwin 64-bit x86_64 AVX2 AC]
Copyright (C) 1996-2019 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single=[SECTION,...]  "single crack" mode, using default or named rules
--single=rule[,...]     same, using "immediate" rule(s)
--wordlist=[FILE] --stdin wordlist mode, read words from FILE or stdin
--pipe                  like --stdin, but bulk reads, and allows rules
--loopback=[FILE]       like --wordlist, but extract words from a .pot file
--dupe-suppression      suppress all dupes in wordlist (and force preload)
--prince=[FILE]         PRINCE mode, read words from FILE
--encoding=NAME         input encoding (eg. UTF-8, ISO-8859-1). See also
                        doc/ENCODINGS and --list-hidden-options.
--rules=[SECTION,...]   enable word mangling rules (for wordlist or PRINCE
                        modes), using default or named rules
--rules=rule[,...]      same, using "immediate" rule(s)
--rules-stack=SECTION[,...] stacked rules, applied after regular rules or to
                        modes that otherwise don't support rules
--rules=rule[,...]      same, using "immediate" rule(s)
--incremental=[MODE]    "incremental" mode (using section MODE)
--mask=[MASK]           mask mode using MASK (or default from john.conf)
--markov=[OPTIONS]      "Markov" mode (see doc/MARKOV)
--external=MODE         external mode or word filter
--subsets=[CHARSET]     "subsets" mode (see doc/SUBSETS)
--stdout=[LENGTH]       just output candidate passwords [cut at LENGTH]
--restore=[NAME]        restore an interrupted session [called NAME]
--session=NAME          give a new session the NAME
--status=[NAME]         print status of a session [called NAME]
--make-charset=FILE     make a charset file. It will be overwritten
--show=[left]          show cracked passwords (if --left, then uncracked)
--test=[TIME]          run tests and benchmarks for TIME seconds each
--users=[-]LOGIN[UID,...] [do not] load this (these) user(s) only
--groups=[-]GID[,...]   load users [not] of this (these) group(s) only
--shells=[-]SHELL[,...] load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX] load salts with[out] COUNT [to MAX] hashes
--costs=[-]C[:W],...    load salts with[out] cost value on (to Wm). For
                        tunable cost parameters, see doc/OPTIONS
--save-memory=LEVEL     enable memory saving, at LEVEL 1..3
--mode=MIM[:MAX]/TOTAL this mode's number range out of TOTAL count
--fork=N               fork N processes
--pot=NAME             pot file to use
--list=WHAT            list capabilities, see --list-help or doc/OPTIONS
--devices=N[,...]      set OpenCL device(s) (see --list-opencl-devices)
--format=NAME          force hash of type NAME. The supported formats can
                        be seen with --list-formats and --list-subformats

C:\john-1.9.0-jumbo-1-win64>john-1.9.0-jumbo-1-win64\run>

```

### Step 4 : Crack the md5 hash

command : john.exe --format=raw-md5 C:\Users\Ritesh\Desktop\jtrtest.txt

```

C:\Windows\System32\cmd.exe
C:\john-1.9.0-jumbo-1-win64>john-1.9.0-jumbo-1-win64\run>john.exe --format=raw-md5 C:\Users\Ritesh\Desktop\jtrtest.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=16
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:password.lst, rules:WordList
hello      (?)
1g 0:00:00:00 DONE 2/3 (2025-02-02 22:44) 62.50g/s 24000p/s 24000c/s 24000C/s 123456..larry
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed

C:\john-1.9.0-jumbo-1-win64>john-1.9.0-jumbo-1-win64\run>

```

It has successfully cracked the password 'hello'.

---

**Outcomes: CO3: Understand Attack Methodology**

---

**Conclusion: (Conclusion to be based on the objectives and outcomes achieved)**

From this experiment I learned that password cracking tools such as John the Ripper and Hashcat are powerful resources for assessing the strength of password storage mechanisms. The experiment highlighted the importance of using strong, complex passwords and secure hashing algorithms, as even well-known and publicly available tools can successfully crack weak passwords. It also underscored the ethical responsibilities and limitations inherent in such tools, emphasizing that they should only be used in authorized environments for security testing purposes.

---

**Grade: AA / AB / BB / BC / CC / CD / DD****Signature of faculty in-charge with date**

---

**REFERENCES:**

<https://www.openwall.com/john/>

▶ John the Ripper in Action: Practical Steps to Crack Passwords

[Cain & Abel Overview](#)