SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE

This is to certify that **21CSE281T – Cryptography and Network Security**, Case study titled "**John the Ripper**" is the bonafide work of **MATHESH.M (Reg no RA2211030010053)** who undertook the task of completing the work within the allotted time.

Dept. of Networking and Communications
Kattankulathur
603 203
SRM IST

SIGNATURE

Dr. S. A. ANGAYARKANNI
Assistant Professor
Department of Networking and Communications
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SIGNATURE

Dr. ANNAPURANI. K
Professor and Head
Department of Networking and Communications
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

John the Ripper

MINI PROJECT REPORT
of
21CSE281T – Cryptography and Network Security

Submitted by

Mathesh.M [Reg No:RA2211030010053]

Under the Guidance of

Dr. Angayarkanni S A
Assistant Professor, NWC

in partial fulfillment of the requirementsfor the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING
with specialization in Cyber Security

DEPARTMENT OF NETWORKING AND COMMUNICATIONS
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE ANDTECHNOLOGY
KATTANKULATHUR- 603 203

MAY 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE

This is to certify that **21CSE281T – Cryptography and Network Security**, Case study titled "**John the Ripper**" is the bonafide work of **MATHESH.M (Reg no RA2211030010053)** who undertook the task of completing the work within the allotted time.

SIGNATURE

Dr. S. A. ANGAYARKANNI
Assistant Professor
Department of Networking and Communications
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SIGNATURE

Dr. ANNAPURANI. K
Professor and Head
Department of Networking and Communications
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	Abstract	4
1.	Introduction	5
2.	Objective	6
3.	Challenges	7
4.	Proposed Work	9
5.	Architecture Diagram	11
6.	Working model	12
7.	Results	19
8.	Conclusion	20
9.	References	21
10.	Appendix – I (Demo pic)	22
11.	Appendix – II (Sample Code)	23

Abstract

John the Ripper stands as a venerable titan in the realm of cybersecurity, offering a versatile and powerful suite of tools for password analysis, hash cracking, and vulnerability assessment. Developed by Solar Designer in the early 1990s, this open-source software has continually evolved to meet the challenges posed by ever-advancing encryption algorithms and security measures.

This abstract provides a detailed exploration of John the Ripper's capabilities, evolution, and impact on the field of information security. From its humble origins to its current status as an indispensable tool for cybersecurity professionals, John the Ripper has proven itself to be a formidable ally in the ongoing battle against cyber threats.

At its core, John the Ripper is designed to crack passwords efficiently across various platforms and operating systems. Its arsenal includes sophisticated algorithms and methodologies that enable it to analyze hashes, conduct dictionary attacks, and employ brute-force techniques with remarkable speed and accuracy. This versatility makes it invaluable for assessing the strength of password security measures and identifying potential vulnerabilities in digital infrastructures.

Furthermore, this abstract explores the ethical considerations surrounding the use of John the Ripper. While its primary purpose is to bolster cybersecurity defenses, its immense power also raises concerns about potential misuse and ethical implications. Responsible and lawful utilization is essential to ensure that John the Ripper is wielded for the greater good of protecting digital assets and preserving privacy.

In addition to its role in penetration testing and vulnerability assessment, John the Ripper has also found applications in forensic analysis and password recovery. Its ability to decrypt passwords from encrypted data sets has proven invaluable in criminal investigations and digital forensic examinations, aiding law enforcement agencies in their efforts to combat cybercrime.

Looking ahead, the continued development and refinement of John the Ripper promise to further enhance its capabilities and solidify its position as a cornerstone of cybersecurity. Community-driven collaboration and ongoing research efforts ensure that John the Ripper remains at the forefront of password security analysis, adapting to meet the evolving challenges of the digital landscape..

Introduction

In the realm of cybersecurity, the importance of robust password security cannot be overstated. As the digital landscape continues to expand, so too do the methods and tools used by malicious actors to compromise sensitive information. Amidst this perpetual arms race, one tool has remained a stalwart defender of digital fortresses: John the Ripper.

Initially conceived in the early 1990s by Solar Designer, John the Ripper has since established itself as a cornerstone of password security analysis and penetration testing. Its open-source nature, coupled with its adaptability and effectiveness, has made it a go-to solution for security professionals seeking to assess the resilience of their systems against unauthorized access.

This introduction sets the stage for an in-depth exploration of John the Ripper, examining its origins, evolution, and multifaceted capabilities. From humble beginnings to its current status as a ubiquitous presence in the cybersecurity toolkit, John the Ripper's journey reflects the ongoing struggle to stay one step ahead of those who seek to exploit vulnerabilities in digital infrastructure.

As we delve into the intricacies of John the Ripper, it becomes evident that its significance extends far beyond mere password cracking. Its versatility encompasses a wide array of functionalities, ranging from hash analysis and dictionary attacks to brute-force techniques, all aimed at identifying weaknesses and fortifying defenses against potential threats.

Moreover, this introduction highlights the ethical considerations surrounding the use of John the Ripper, emphasizing the importance of responsible and lawful application in cybersecurity practices. While its utility in penetration testing and vulnerability assessment is undeniable, adherence to ethical guidelines ensures that its power is wielded for the greater good of safeguarding digital assets and preserving privacy.

In essence, John the Ripper stands as a testament to the ever-evolving nature of cybersecurity, embodying the relentless pursuit of innovation and resilience in the face of emerging threats. By unraveling the intricacies of this indispensable tool, we gain valuable insights into the intricacies of password security and the ongoing battle to secure the digital realm.

Objective

The primary objective of John the Ripper is to provide cybersecurity professionals with a comprehensive and versatile tool for password security analysis, hash cracking, and vulnerability assessment. Developed with the goal of addressing the ever-present threat of unauthorized access to digital assets, John the Ripper aims to empower users with the means to identify and mitigate potential weaknesses in password security measures.

At its core, John the Ripper seeks to achieve several key objectives:

Efficient Password Cracking: One of the primary functions of John the Ripper is to crack passwords efficiently across various platforms and operating systems. By employing advanced algorithms and methodologies, the tool aims to rapidly decrypt passwords stored in encrypted data sets, thereby assessing the strength of password security measures and identifying potential vulnerabilities.

Versatile Hash Analysis: John the Ripper provides users with the ability to analyze hashes derived from password files, enabling them to understand the underlying encryption algorithms and identify patterns or weaknesses that may be exploited by malicious actors. This functionality is crucial for evaluating the effectiveness of cryptographic techniques and strengthening password security measures accordingly.

Comprehensive Dictionary Attacks: Another key objective of John the Ripper is to conduct dictionary attacks, wherein it systematically tests a predetermined list of words and phrases as potential passwords. By leveraging extensive wordlists and customizable parameters, the tool aims to identify common or easily guessable passwords that may pose security risks to digital assets.

Brute-Force Techniques: In addition to dictionary attacks, John the Ripper is equipped with brute-force capabilities, enabling it to systematically generate and test all possible password combinations within a specified character set and length. This exhaustive approach ensures thorough coverage of password space, allowing users to identify weak passwords that may be susceptible to brute-force attacks.

Ethical and Responsible Utilization: While empowering users with powerful password cracking capabilities, John the Ripper emphasizes the importance of ethical and responsible utilization. The tool encourages adherence to legal and ethical guidelines to ensure that its power is wielded for the legitimate purpose of strengthening cybersecurity defense

Challenges

While John the Ripper stands as a formidable tool in the arsenal of cybersecurity professionals, it is not without its challenges. These challenges span technical limitations, ethical considerations, and legal constraints, all of which must be navigated effectively to ensure responsible and effective use of the software. Some of the key challenges associated with John the Ripper include:

Complexity of Encryption Algorithms: As encryption algorithms continue to evolve and become more sophisticated, cracking passwords becomes increasingly challenging. John the Ripper must constantly adapt to keep pace with advancements in encryption technology, requiring ongoing research and development efforts to maintain its effectiveness.

Resource Intensive Operations: Cracking passwords using techniques such as brute-force attacks or exhaustive dictionary searches can be highly resource-intensive, requiring significant computational power and time. This poses challenges for users with limited computing resources or tight time constraints, potentially hindering the efficiency of password cracking operations.

Legal and Ethical Considerations: The use of John the Ripper raises important legal and ethical considerations, particularly regarding privacy, consent, and lawful access to digital assets. Users must navigate complex legal frameworks and adhere to ethical guidelines to ensure that the software is used responsibly and lawfully, avoiding unauthorized access to sensitive information.

False Positives and False Negatives: In password cracking operations, there is always a risk of encountering false positives (incorrectly identified passwords) or false negatives (missed opportunities to crack passwords). Balancing the trade-off between false positives and false negatives poses a significant challenge for users, requiring careful parameter tuning and validation processes to minimize errors.

Resistance to Cracking Techniques: Some password security measures employ techniques such as salting, key stretching, or multi-factor authentication to enhance resilience against cracking attempts. These measures introduce additional challenges for password cracking tools like John the Ripper, necessitating innovative approaches and techniques to overcome resistance and successfully crack passwords.

Detection and Mitigation: In many scenarios, the use of password cracking tools like John the Ripper may be detected by intrusion detection systems or security monitoring mechanisms. Detecting and mitigating such activities poses challenges for users seeking to maintain stealth and avoid detection during penetration testing or vulnerability assessment engagements.

Continual Evolution of Threat Landscape: The threat landscape in cybersecurity is constantly evolving, with new attack techniques, vulnerabilities, and security challenges emerging regularly. Staying ahead of these evolving threats requires

constant vigilance and adaptation on the part of users employing tools like John the Ripper, posing ongoing challenges for cybersecurity professionals.
s and protecting digital assets.

Proposed Work

The proposed work on John the Ripper aims to address several key areas to enhance its functionality, effectiveness, and usability as a password security analysis tool. This proposed work encompasses both technical improvements and research initiatives aimed at advancing the capabilities of John the Ripper and addressing emerging challenges in the field of cybersecurity. The following outlines the proposed areas of focus:

Algorithmic Enhancements: Research and development efforts will focus on improving the underlying algorithms used by John the Ripper for password cracking, hash analysis, and dictionary attacks. This includes exploring new cryptographic techniques, optimizing existing algorithms for performance and efficiency, and developing novel approaches to overcome challenges posed by complex encryption algorithms.

Scalability and Performance Optimization: Efforts will be directed towards enhancing the scalability and performance of John the Ripper, particularly for resource-intensive operations such as brute-force attacks and exhaustive dictionary searches. This may involve parallelizing computations, optimizing memory usage, and leveraging hardware acceleration techniques to maximize efficiency and reduce processing times.

Integration with Other Tools and Frameworks: The proposed work will explore opportunities to integrate John the Ripper with other cybersecurity tools and frameworks, enabling seamless interoperability and enhancing its utility in diverse security contexts. This may include integration with vulnerability scanners, penetration testing frameworks, and forensic analysis tools to streamline workflows and facilitate comprehensive security assessments.

Enhanced User Interface and Usability: Improvements to the user interface and overall usability of John the Ripper will be prioritized to enhance user experience and facilitate ease of use for both novice and experienced users. This may involve redesigning the interface for better clarity and intuitiveness, implementing interactive visualizations for analysis results, and providing comprehensive documentation and tutorials to support users.

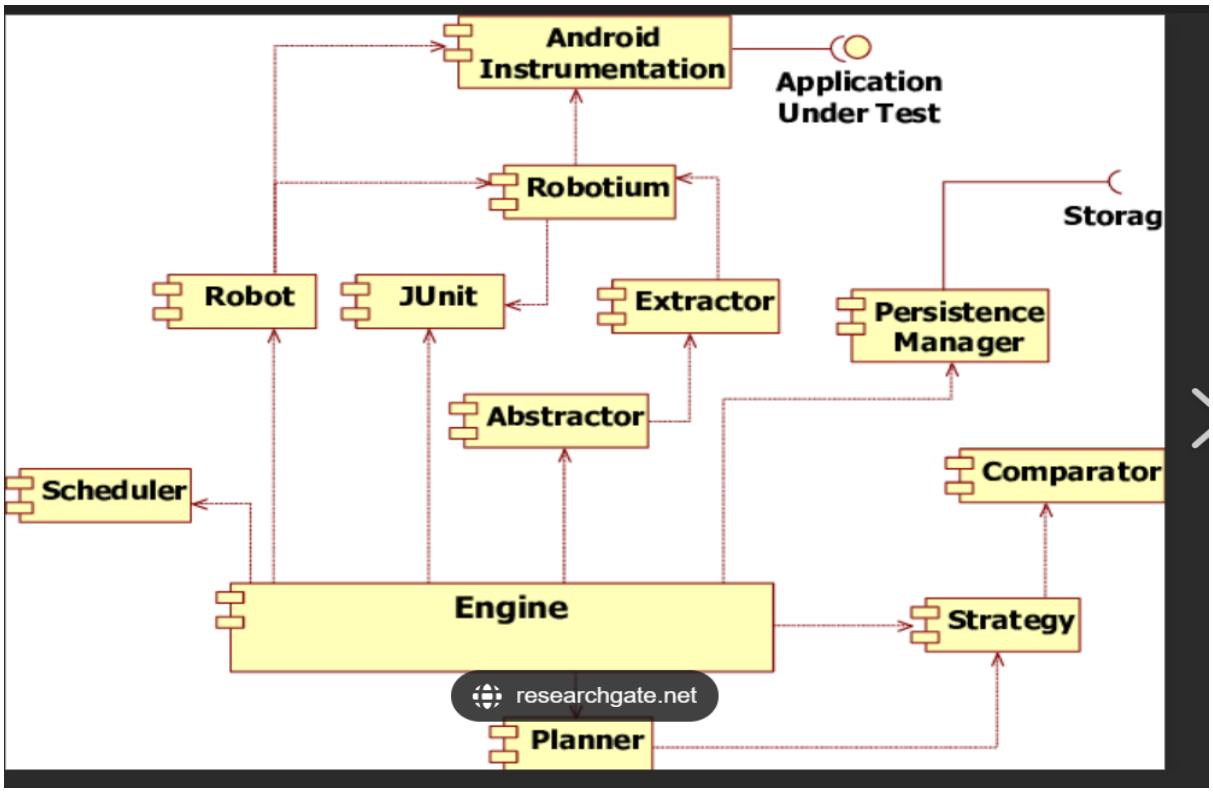
Advanced Threat Detection and Mitigation: Research efforts will focus on developing advanced threat detection and mitigation capabilities within John the Ripper to help users identify and respond to emerging cyber threats effectively. This may involve integrating anomaly detection algorithms, machine learning techniques, and behavioral analysis mechanisms to detect suspicious activities and mitigate potential risks in real-time.

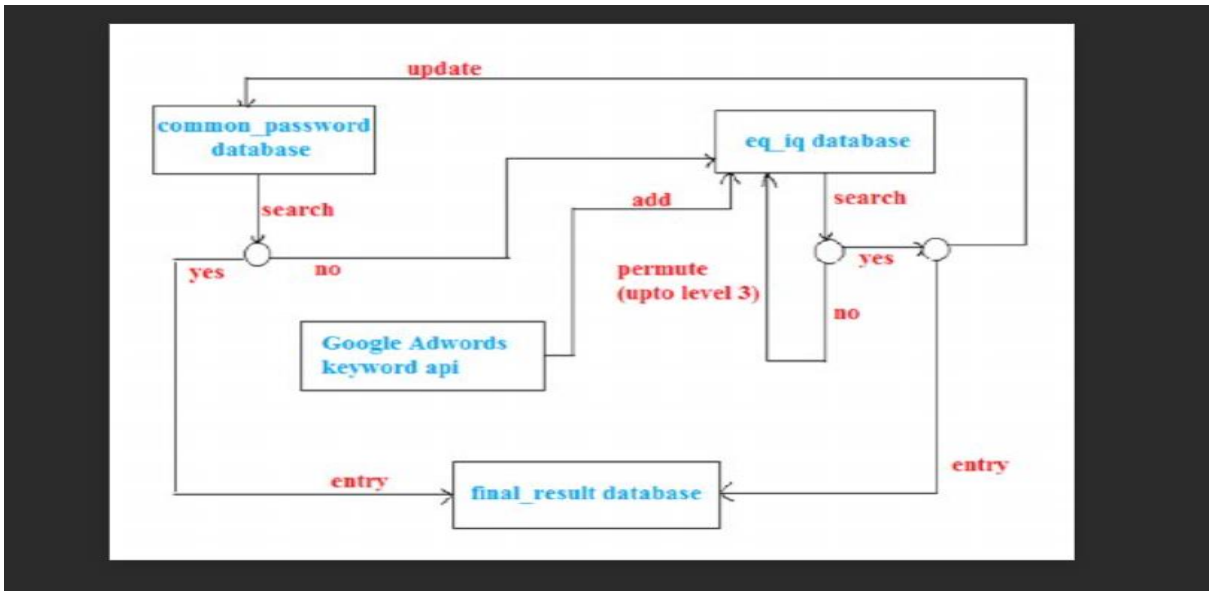
Ethical and Legal Compliance: The proposed work will emphasize the importance of ethical and legal compliance in the use of John the Ripper, providing guidance and best practices to ensure responsible and lawful utilization of the tool. This includes promoting transparency, consent, and accountability in password security analysis activities, as well as adhering to relevant regulations and standards governing cybersecurity practices.

Community Collaboration and Knowledge Sharing: Collaboration with the cybersecurity community will be encouraged to foster knowledge sharing, innovation, and continuous improvement of John the Ripper. This includes engaging with open-source contributors,

academic researchers, and industry experts to solicit feedback, share insights, and collaborate on research initiatives to advance the state-of-the-art in password security analysis.

Architecture Diagram





Working model

```
root@kali:~# john
John the Ripper password cracker, version 1.8.0.6-jumbo-1-
-64]
Copyright (c) 1996-2015 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single[=SECTION]        "single crack" mode
--wordlist[=FILE] --stdin wordlist mode, read words from F
                  --pipe  like --stdin, but bulk reads, an
--loopback[=FILE]         like --wordlist, but fetch words
--dupe-suppression        suppress all dupes in wordlist (
--prince[=FILE]           PRINCE mode, read words from FIL
--encoding=NAME           input encoding (eg. UTF-8, ISO-8
                          doc/ENCODING and --list=hidden-o
--rules[=SECTION]         enable word mangling rules for w
--incremental[=MODE]      "incremental" mode [using sectio
--mask=MASK               mask mode using MASK
--markov[=OPTIONS]        "Markov" mode (see doc/MARKOV)
--external=MODE           external mode or word filter
```

John the Ripper works in 3 distinct modes to crack the passwords:

1. Single Crack Mode
2. Wordlist Crack Mode
3. Incremental Mode

John the Ripper Single Crack Mode:

john --single --format=raw-sha1 crack.txt

```
root@kali:~# john --single --format=raw-sha1 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 SSE2 4x])
Press 'q' or Ctrl-C to abort, almost any other key for status
IgNiTe           (ignite)
1g 0:00:00:00 DONE (2018-06-04 20:29) 4.545g/s 1531p/s 1531c/s 1531C/s I
gite
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

John the Ripper Wordlist Crack Mode:

john --wordlist=/usr/share/john/password.lst --format=raw-sha1 crack.txt

```
root@kali:~# john --wordlist=/usr/share/john/password.lst
--format=raw-sha1 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 SSE2 4x])
Press 'q' or Ctrl-C to abort, almost any other key for sta
tus
asdfasdf         (pavan)
1g 0:00:00:00 DONE (2018-06-04 21:07) 1.562g/s 1175p/s 117
5c/s 1175C/s arizona..asdfasdf
Use the "--show" option to display all of the cracked pass
words reliably
Session completed
```

```
root@kali:~# cat /etc/shadow
root:$6$QizMF3Ej$W7m6QbPmvRb4eyjt.Ic6KiwjCy/FU86vUucgdc/Z
.THObbp2VvMCEDJXAEt0ibpL0sV6Fxps.8k9FpmKKY1FJ.:17569:0:99
999:7:::
daemon:*:17557:0:99999:7:::
bin:*:17557:0:99999:7:::
sys:*:17557:0:99999:7:::
sync:*:17557:0:99999:7:::
games:*:17557:0:99999:7:::
man:*:17557:0:99999:7:::
lp:*:17557:0:99999:7:::
mail:*:17557:0:99999:7:::
news:*:17557:0:99999:7:::
uucp:*:17557:0:99999:7:::
proxy:*:17557:0:99999:7:::
www-data:*:17557:0:99999:7:::
backup:*:17557:0:99999:7:::
list:*:17557:0:99999:7:::
irc:*:17557:0:99999:7:::
```

```
saned:*:17557:0:99999:7:::
speech-dispatcher:!:17557:0:99999:7:::
avahi:*:17557:0:99999:7:::
pulse:*:17557:0:99999:7:::
Debian-gdm:*:17557:0:99999:7:::
king-phisher:*:17557:0:99999:7:::
dradis:*:17557:0:99999:7:::
beef-xss:*:17557:0:99999:7:::
pavan:$6$oTuUxWEX$i4QeRmbUN4PfAF0fVRu6HMCHSUorO630R8tmIzi
DNVjY3jKKcVac9pWNfGKS/3SD1pF3UKr89HL01h51Q/nCu.:17686:0:9
9999:7:::
```

john crack.txt

```
root@kali:~# john crack.txt
Warning: detected hash type "sha512crypt", but
 is also recognized as "crypt"
Use the "--format=crypt" option to force loadi
 that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3)
 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other
atus
asdfasdf         (pavan)
1g 0:00:00:15 DONE 2/3 (2018-06-04 21:24) 0.06
9p/s 237.9c/s 237.9C/s valentine..bigben
Use the "--show" option to display all of the
swords reliably
Session completed
```

unshadow /etc/passwd /etc/shadow > crack.txt

```
root@kali:~# unshadow /etc/passwd /etc/shadow > crack.txt
```

Open | crack.txt ~/ | Save

```
root:$6$QizMF3Ej$W7m6QbPmvRb4eyjt.Ic6KiwjCy/FU86vUucgdc
Z.THObbp2VvMCEDJXAEt0ibpL0sV6Fxps.8k9FpmKKY1FJ.:
0:0:root:/root:/bin/bash
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
```

john --wordlist=/usr/share/john/password.lst crack.txt

```
root@kali:~# john --wordlist=/usr/share/john/password.lst
 crack.txt
Warning: detected hash type "sha512crypt", but the string
 is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as
 that type instead
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512cr
ypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other key for st
atus
123              (raj)
asdfasdf         (pavan)
yellow           (ignite)
3g 0:00:00:21 DONE (2018-06-04 21:32) 0.1419g/s 167.7p/s
243.4c/s 243.4C/s paagal..sss
Use the "--show" option to display all of the cracked pas
swords reliably
Session completed
```

```
root@kali:~# john --wordlist=/usr/share/john/password.lst /root/Desktop/cra
.txt
Warning: detected hash type "sha512crypt", but the string is also recognize
as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$
HA512 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:21 78.28% (ETA: 08:40:51) 0g/s 120.3p/s 243.5c/s 243.5C/s bull..
rmal
Session aborted
```

john --restore

```
root@kali:~# john --restore
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3
HA512 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:22 78.28% (ETA: 08:41:23) 0g/s 119.2p/s 241.4c/s 241.4C/s
0g 0:00:00:29 DONE (2018-06-04 08:41) 0g/s 122.2p/s 246.7c/s 246.7C/s
.sss
```

john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-sha1 crack.txt

```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-
sha1 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 SSE2 4x])
Press 'q' or Ctrl-C to abort, almost any other key for status
Hacker           (pavan)
1g 0:00:00:00 DONE (2018-06-04 23:11) 3.225g/s 810541p/s 810541c/s 810541C/
s Hannah12..Hacker
Use the "--show" option to display all of the cracked passwords reliably
```

john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 rack.txt

```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5
rack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
P@ssword         (pavan)
1g 0:00:00:00 DONE (2018-06-04 23:09) 4.761g/s 352971p/s 352971c/s 352971C/s P
hbear1..Morgan1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md4 crack.txt

```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-
md4 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD4 [MD4 128/128 SSE2 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
Rockyou          (pavan)
1g 0:00:00:00 DONE (2018-06-04 23:12) 4.166g/s 30200p/s 30200c/s 30200C/s b
eyonce1..soccer09
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-sha256 crack.txt

john --wordlist=/usr/share/wordlists/rockyou.txt --format=ripemd-128 crack.txt

```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-
sha256 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 SSE2 4x])
Press 'q' or Ctrl-C to abort, almost any other key for status
pAsSwOrD         (pavan)
1g 0:00:00:02 DONE (2018-06-04 23:14) 0.4166g/s 2018Kp/s 2018Kc/s 2018KC/s
pAsik..pAsSWORD
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

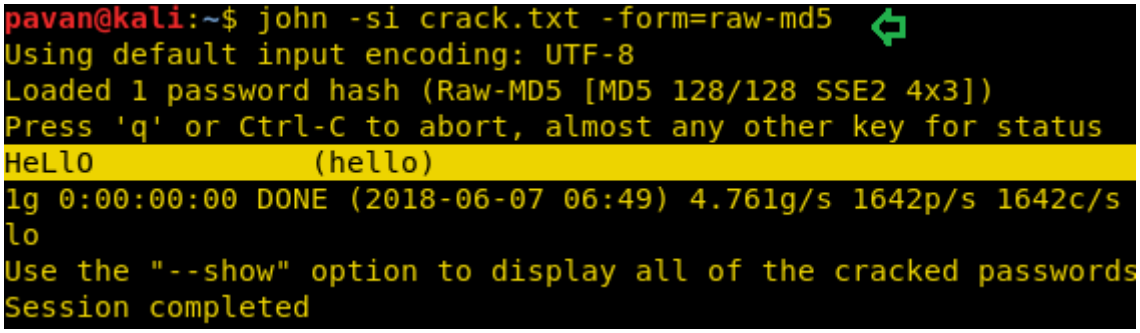
john --wordlist=/usr/share/wordlists/rockyou.txt --format=whirlpool crack.txt

```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=whir
lpool crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (whirlpool [WHIRLPOOL 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
password666      (pavan)
1g 0:00:00:00 DONE (2018-06-04 23:20) 3.225g/s 284241p/s 284241c/s 284241C/
s password666
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

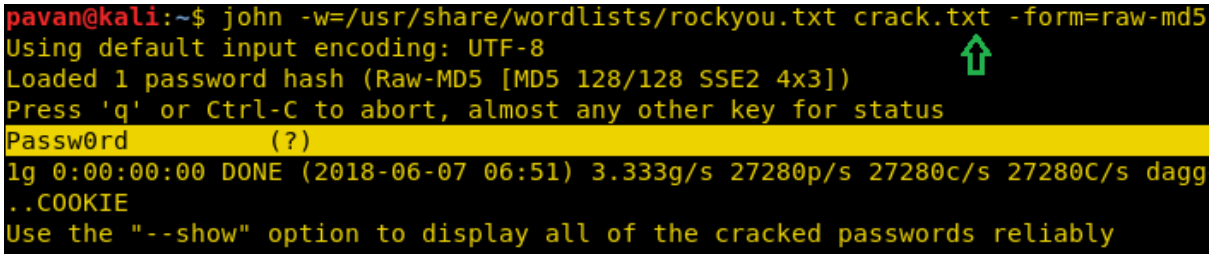
john --list=formats

```
root@kali:~# john --list=formats
descrypt, bsdicrypt, md5crypt, bcrypt, scrypt, LM, AFS, tripcode, dummy,
dynamic_n, bfegg, dmd5, dominosec, dominosec8, EPI, Fortigate, FormSpring,
has-160, hdaa, ipb2, krb4, krb5, KeePass, MSCHAPv2, mschapv2-naive, mysql,
nethalflm, netlm, netlmv2, netntlm, netntlm-naive, netntlmv2, md5ns, NT, osc,

PHPS, po, skey, SybaseASE, xsha, xsha512, agilekeychain, aix-ssha1,
aix-ssha256, aix-ssha512, asa-md5, Bitcoin, Blackberry-ES10, WoWSRP,
Blockchain, chap, Clipperz, cloudkeychain, cq, CRC32, sha1crypt, sha256crypt,

sha512crypt, Citrix_NS10, dahua, Django, django-scrypt, dmg, dragonfly3-32,
dragonfly3-64, dragonfly4-32, dragonfly4-64, Drupal7, eCryptfs, EFS, eigrp,
EncFS, EPiServer, fde, gost, gpg, HAVAL-128-4, HAVAL-256-3, HMAC-MD5,
HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, hMailServer,
hsrp, IKE, keychain, keyring, keystore, known_hosts, krb5-18, krb5pa-sha1,
kwallet, lp, lotus5, lotus85, LUKS, MD2, md4-gen, mdc2, MediaWiki, MongoDB,
Mozilla, mscash, mscash2, krb5pa-md5, mssql, mssql05, mssql12, mysql-sha1,
mysqlna, net-md5, net-sha1, nk, nsldap, o5logon, ODF, Office, oldoffice,
OpenBSD-SoftRAID, openssl-enc, oracle, oracle11, Oracle12C, Panama,
pbkdf2-hmac-md5, PBKDF2-HMAC-SHA1, PBKDF2-HMAC-SHA256, PBKDF2-HMAC-SHA512,
PDF, PFX, phpass, pix-md5, plaintext, pomelo, postgres, PST, PuTTY, pwsafe,
RACF, RAdmin, RAKP, rar, RAR5, Raw-SHA512, Raw-Blake2, Raw-Keccak,
Raw-Keccak-256, Raw-MD4, Raw-MD5, Raw-SHA1, Raw-SHA1-Linkedin, Raw-SHA224,
Raw-SHA256, Raw-SHA256-ng, Raw-SHA3, Raw-SHA384, Raw-SHA512-ng, Raw-SHA,
Raw-MD5u, ripemd-128, ripemd-160, rsvp, Siemens-S7, Salted-SHA1, SSHA512,
sapb, sapg, saph, 7z, sha1-gen, Raw-SHA1-ng, SIP, skein-256, skein-512,
aix-smd5, Snefru-128, Snefru-256, LastPass, SSH, SSH-ng, STRIP, SunMD5, sxc,
Sybase-PROP, tcp-md5, Tiger, tc_aes_xts, tc_ripemd160, tc_sha512,
tc_whirlpool, VNC, vtp, wbb3, whirlpool, whirlpool0, whirlpool1, wpapsk, ZIP,
```

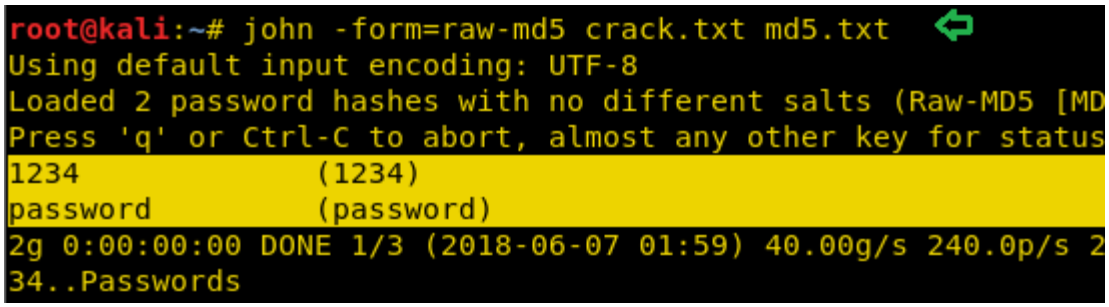
john -si crack.txt -form=raw-md5



john -w=/usr/share/wordlists/rockyou.txt crack.txt -form=raw-md5



john -form=raw-md5 crack.txt md5.txt



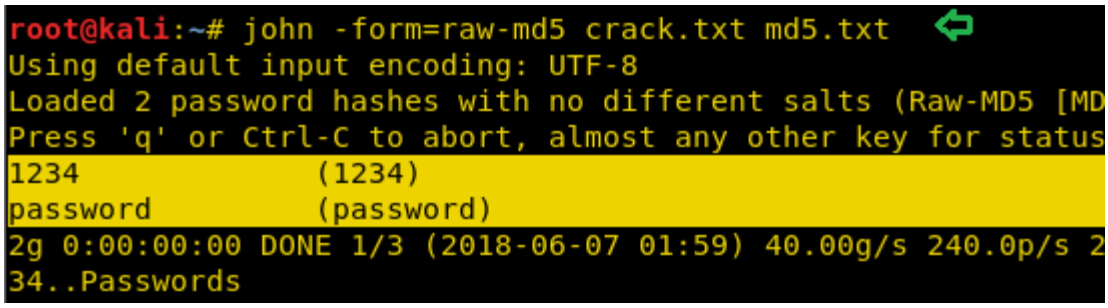
Results

John the Ripper is a powerful password cracking tool primarily used in the field of cybersecurity. It's designed to detect weak Unix passwords and employs various methods like dictionary attacks, brute force attacks, and rainbow tables to crack passwords.

The results you'll get from John the Ripper depend on factors like the strength of the passwords being targeted, the complexity of the attack, and the computing power available for the cracking process. In many cases, John the Ripper can successfully crack weak or poorly chosen passwords relatively quickly, while stronger passwords may take significantly longer or even be resistant to cracking altogether.

If you're using John the Ripper for security purposes, it's essential to ensure that you're authorized to test the passwords you're attempting to crack and that you're following all relevant laws and regulations regarding cybersecurity and ethical hacking.

Example output:



Conclusion

John the Ripper stands as a formidable force in the realm of cybersecurity, renowned for its prowess in password cracking. Offering a wide array of techniques, including dictionary attacks, brute force methods, and the utilization of pre-computed hash tables (rainbow tables), it provides a comprehensive toolkit for assessing the security of password-protected systems.

At its core, John the Ripper's strength lies in its versatility and adaptability. It supports an extensive range of hash types and algorithms, making it applicable across diverse systems and scenarios. Its extensibility allows for the integration of new features and enhancements, ensuring its relevance and effectiveness in an ever-evolving cybersecurity landscape.

One of its standout features is its customization options, which enable users to tailor the cracking process to specific targets. Users can define character sets, implement word mangling rules, and fine-tune parallelization settings, optimizing performance and maximizing the chances of success.

Furthermore, John the Ripper benefits from active maintenance and development by a dedicated community of contributors. This ongoing support ensures that the tool remains up-to-date and capable of handling emerging challenges and advancements in password security.

However, it's crucial to emphasize the importance of responsible and ethical usage. Unauthorized or unethical use of John the Ripper to crack passwords without proper authorization or consent is not only illegal but also undermines trust and integrity in cybersecurity practices. Security professionals and researchers must adhere to relevant laws and ethical guidelines, obtaining appropriate authorization before employing John the Ripper in security assessments or penetration testing.

In conclusion, John the Ripper's reputation as a powerful and versatile password cracking tool is well-deserved. With its diverse range of techniques, extensive support for hash types, and customizable features, it remains a go-to solution for assessing password security. Nevertheless, its use must be accompanied by a commitment to ethical conduct and legal compliance, ensuring that it serves as a tool for enhancing security rather than compromising it.

References

1. Official Website: The official John the Ripper website (https://www.openwall.com/john/) provides comprehensive documentation, including user guides, tutorials, and FAQs.
2. GitHub Repository: The project's GitHub repository (https://github.com/openwall/john) contains the source code, issue tracker, and discussions related to the development of John the Ripper.
3. Online Forums and Communities: Websites like Stack Overflow, Reddit (e.g., r/netsec), and specialized cybersecurity forums often have discussions, tips, and tutorials related to John the Ripper.
4. Books and Online Courses: Various cybersecurity books and online courses cover password cracking techniques, including the use of tools like John the Ripper. Searching for resources on cybersecurity education platforms like Cybrary or Udemy may yield valuable learning materials.
5. White Papers and Conference Proceedings: Researchers and cybersecurity professionals may publish white papers or present findings related to password security and cracking techniques at conferences like DEF CON, Black Hat, or academic conferences in cybersecurity.
6. Blog Posts and Tutorials: Many cybersecurity professionals and enthusiasts share their experiences and insights into using John the Ripper through blog posts and tutorials. Searching for blogs dedicated to cybersecurity or ethical hacking may yield valuable step-by-step guides and tips.
7. YouTube Videos: Video tutorials on platforms like YouTube can be a valuable resource for visual learners. Many cybersecurity experts create video content demonstrating how to use John the Ripper effectively and discussing best practices for password cracking.
8. Online Security Communities: Websites and forums dedicated to cybersecurity, such as Security Stack Exchange, Null Byte, and Hack Forums, often have dedicated sections or threads where users discuss tools like John the Ripper, share tips, and ask for advice.
9. Academic Papers: While John the Ripper itself may not have academic references, research papers in the field of cybersecurity often discuss password cracking techniques and tools. Searching academic databases like IEEE Xplore or Google Scholar for papers related to password security and cracking may yield valuable insights.
10. Capture The Flag (CTF) Challenges: Participation in cybersecurity CTF competitions can provide hands-on experience with tools like John the Ripper in simulated environments. Many CTF challenges involve password cracking scenarios where participants use tools like John the Ripper to recover passwords from encrypted files or services.

Appendix – I (Demo pic)

John the Ripper:

Appendix – II (Sample Code)

John the Ripper:

```
$ john
$ apt install John
$ brew install john
$ john -h
$ john --single --format=raw-sha1 crack.txt
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-sha1 crack.txt
$ john -i:digits passwordfile.txt
$ john --format=lm crack.txt
$ unshadow /etc/passwd /etc/shadow > output.db
$ john output.db
$ zip2john file.zip > zip.hashes
$john zip.hashes
```