HỌC VIỆN KỸ THUẬT MẬT MÃ

KHOA AN TOÀN THÔNG TIN

----֎----



CƠ SỞ AN TOÀN THÔNG TIN

BÁO CÁO HỌC PHẦN

**ĐỀ TÀI**

**TÌM HIỂU VỀ JOHN THE RIPPER**

**SINH VIÊN THỰC HIỆN :**

**Nguyễn Minh Tài AT180143**

**Nguyễn Thùy Linh AT180130**

**Nguyễn Tuấn Anh AT180103**

Giảng viên hướng dẫn T.S. Nguyễn Đức Ngân

Khoa An toàn thông tin, Học viện Kỹ Thuật Mật Mã

**Hà Nội, 2024**

Nội dung

[LỜI NÓI ĐẦU 5](#_Toc170314222)

[DANH SÁCH CÁC KÍ HIỆU ,CHỮ VIẾT TẮT 6](#_Toc170314223)

[CHƯƠNG I. TỔNG QUAN VỀ JOHN THE RIPPER 7](#_Toc170314224)

[1. Giới thiệu chung vê John The Ripper 7](#_Toc170314225)

[1.1. Khái niệm 7](#_Toc170314226)

[1.2.Lịch sử phát triển 7](#_Toc170314227)

[1.3.Mục đích sử dụng 8](#_Toc170314228)

[2. Các tính năng chính 8](#_Toc170314229)

[3.Cách cài đặt 8](#_Toc170314230)

[3.1.Trên Linux 9](#_Toc170314231)

[3.2.Trên MacOS 9](#_Toc170314232)

[3.2.Trên Windows 10](#_Toc170314233)

[4.Cách sử dụng cơ bản 10](#_Toc170314234)

[4.1. Chạy John the Ripper: 10](#_Toc170314235)

[4.2. Kiểm tra mật khẩu bằng wordlists: 10](#_Toc170314236)

[4.3. Kiểm tra mật khẩu bằng Incremental Mode: 11](#_Toc170314237)

[4.4. Kiểm tra kết quả bẻ khóa: 11](#_Toc170314238)

[4.5. Cấu hình tùy chỉnh: 11](#_Toc170314239)

[5.Cấu hình và tùy chỉnh 11](#_Toc170314240)

[6.Các Plugin và module bổ sung 12](#_Toc170314241)

[7.Các kịch bản tấn công 13](#_Toc170314242)

[8.Bảo mật và biện pháp phòng ngừa 14](#_Toc170314243)

[CHƯƠNG 2: CÁC CHẾ ĐỘ BẺ KHÓA MẬT KHẨU TRONG 15](#_Toc170314244)

[JOHN THE RIPPER 15](#_Toc170314245)

[1. Single Crack Mode 15](#_Toc170314246)

[1.1. Tổng quan về Single Crack Mode : 15](#_Toc170314247)

[1.2. Cách thức hoạt động : 16](#_Toc170314248)

[1.3. Thực nghiệm : 17](#_Toc170314249)

[1.4. Ưu điểm : 18](#_Toc170314250)

[1.5.Hạn chế : 18](#_Toc170314251)

[2. Wordlist Mode 18](#_Toc170314252)

[2.1. Tổng quan về Wordlist Mode : 18](#_Toc170314253)

[2.2. Cách thức hoạt động: 19](#_Toc170314254)

[2.3. Thực nghiệm : 20](#_Toc170314255)

[2.4. Ưu điểm : 22](#_Toc170314256)

[2.5. Hạn chế : 23](#_Toc170314257)

[3. Incremental Mode 24](#_Toc170314258)

[3.1. Tổng quan về Incremental Mode: 24](#_Toc170314259)

[ Định nghĩa: 24](#_Toc170314260)

[ Mục đích và ứng dụng của Incremental Mode: 24](#_Toc170314261)

[3.2. Cách thức hoạt động : 25](#_Toc170314262)

[3.3. Thực nghiệm 26](#_Toc170314263)

[3.4.Ưu điểm : 27](#_Toc170314264)

[3.5 Hạn chế : 28](#_Toc170314265)

[CHƯƠNG 3: BẺ KHÓA MẬT KHẨU VỚI JOHN THE RIPPER 28](#_Toc170314266)

[1.Choose a wordlist 28](#_Toc170314267)

[1.1. Chọn Wordlist Có Sẵn 28](#_Toc170314268)

[1.2. Tải và Chuẩn Bị Wordlist 28](#_Toc170314269)

[1.3. Sử dụng Wordlist trong John the Ripper 29](#_Toc170314270)

[2. choosing specific hashes to crack 29](#_Toc170314271)

[2.1. Xác định loại băm: 29](#_Toc170314272)

[2.2 Tạo tệp chứa băm: 30](#_Toc170314273)

[2.3. Chọn định dạng băm để bẻ khóa: 30](#_Toc170314274)

[3. Cracking Linux Password 31](#_Toc170314275)

[4. cracking windows passwords 33](#_Toc170314276)

[5. cracking zip file 34](#_Toc170314277)

[6. cracking rar file 35](#_Toc170314278)

[6.1.Chuẩn bị: 35](#_Toc170314279)

[6.2.Tạo tệp file rar đã được mã hóa nén 36](#_Toc170314280)

[6.3. Trích Xuất Hash từ Tệp RAR: 37](#_Toc170314281)

[6.4.Bẻ khóa Hash 37](#_Toc170314282)

[7.Cracking SSH keys 38](#_Toc170314283)

[8.Cracking PDF file 40](#_Toc170314284)

[8.1. Cài đặt và chuẩn bị 40](#_Toc170314285)

[8.2. Trích xuất hash từ file PDF 41](#_Toc170314286)

[8.3.Bẻ khóa mật khẩu PDF 42](#_Toc170314287)

[9.Cracking Putty 42](#_Toc170314288)

[CHƯƠNG 4: OTHER USEFUL COMMAND 44](#_Toc170314289)

[1. Cú pháp dòng lệnh của John the Ripper 44](#_Toc170314290)

[2. Các tiện ích bổ sung 53](#_Toc170314291)

[KẾT LUẬN 57](#_Toc170314292)

[TÀI LIỆU THAM KHẢO 57](#_Toc170314293)

[LỜI CẢM ƠN 58](#_Toc170314294)

# LỜI NÓI ĐẦU

Trong bối cảnh an ninh mạng ngày càng trở nên phức tạp và đa dạng, bảo mật mật khẩu đóng vai trò then chốt trong việc bảo vệ thông tin cá nhân và dữ liệu doanh nghiệp. Mật khẩu là hàng rào bảo vệ đầu tiên trước các cuộc tấn công mạng, nhưng đồng thời cũng là điểm yếu nếu không được quản lý và bảo vệ đúng cách. Để đảm bảo an toàn cho hệ thống và thông tin, việc kiểm tra và tăng cường độ mạnh của mật khẩu là vô cùng cần thiết.

John the Ripper, một công cụ mã nguồn mở nổi tiếng trong lĩnh vực bẻ khóa mật khẩu, đã chứng tỏ được sức mạnh và tính linh hoạt của mình. Được sử dụng rộng rãi bởi các chuyên gia an ninh mạng, quản trị viên hệ thống và cả các nhà nghiên cứu bảo mật, John the Ripper không chỉ giúp phát hiện những mật khẩu yếu, dễ bị tấn công mà còn cung cấp các phương pháp để cải thiện độ an toàn của hệ thống.

Báo cáo này nhằm mục đích cung cấp một cái nhìn tổng quan và chi tiết về John the Ripper. Chúng tôi sẽ đi sâu vào các khía cạnh khác nhau của công cụ này, từ lịch sử phát triển, các tính năng chính, đến cách cài đặt và sử dụng. Bên cạnh đó, chúng em cũng sẽ trình bày các kỹ thuật bẻ khóa mật khẩu, cách cấu hình và tùy chỉnh, cũng như các biện pháp bảo vệ để ngăn chặn các tấn công sử dụng John the Ripper.

Bằng việc nghiên cứu và trình bày về John the Ripper, chúng em hy vọng sẽ mang lại một cái nhìn tổng quan và chi tiết về công cụ này, đồng thời giúp Thầy Cô và các bạn có thêm tài liệu tham khảo hữu ích trong việc học tập và nghiên cứu về an ninh mạng.

Chương 1 : Tổng quan về John The Ripper

Chương 2: Các chế độ bẻ khóa mật khẩu trong John The Ripper

Chương 3: Bẻ khóa mật khẩu với John The Ripper

# DANH SÁCH CÁC KÍ HIỆU ,CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| CPU | Central Processing Unit |
| GPU | Graphics Processing Unit |
| SSH | Secure Shell |
| LDAP | Lightweight Directory Access Protocol |
|  |  |

# CHƯƠNG I. TỔNG QUAN VỀ JOHN THE RIPPER

## 1. Giới thiệu chung vê John The Ripper

### 1.1. Khái niệm

John the Ripper là một công cụ phần mềm bẻ khóa mật khẩu miễn phí được phát triển bởi Openwall. Ban đầu được phát triển cho Hệ điều hành Unix nhưng sau đó được phát triển cho các nền tảng khác.

Đây là một trong những chương trình kiểm tra và phá mật khẩu phổ biến nhất vì nó kết hợp một số công cụ bẻ mật khẩu vào một gói, tự động phát hiện các loại băm mật khẩu và bao gồm một công cụ bẻ khóa có thể tùy chỉnh.

Nó có thể chạy với nhiều định dạng mật khẩu được mã hóa khác nhau, bao gồm một số loại băm mật khẩu mật mã thường thấy trong Linux hoặc Windows. Nó cũng có thể là bẻ khóa mật khẩu của các tệp nén như ZIP và cả các tệp Tài liệu như PDF.

### 1.2.Lịch sử phát triển

John the Ripper được phát triển lần đầu tiên bởi Alexander Peslyak, còn được biết đến với biệt danh Solar Designer. Phiên bản đầu tiên của John the Ripper ra mắt vào năm 1996.

Kể từ đó, công cụ này đã trải qua nhiều phiên bản cập nhật và cải tiến, không ngừng mở rộng các tính năng và hỗ trợ nhiều định dạng mật khẩu khác nhau.

### 1.3.Mục đích sử dụng

Mục đích chính của John the Ripper là kiểm tra và bẻ khóa mật khẩu nhằm xác định các điểm yếu trong hệ thống bảo mật. Công cụ này giúp các quản trị viên hệ thống và các chuyên gia bảo mật kiểm tra độ an toàn của mật khẩu được sử dụng trong hệ thống, từ đó đưa ra các biện pháp cải thiện để bảo vệ dữ liệu và tài nguyên quan trọng.

## 2. Các tính năng chính

* **Hỗ trợ nhiều định dạng mật khẩu:** John the Ripper có khả năng xử lý nhiều định dạng mật khẩu khác nhau, bao gồm các mật khẩu Unix (DES, MD5, Blowfish, và nhiều loại hash khác), Windows LM hash, và các mật khẩu được mã hóa bởi các ứng dụng khác như AFS và Kerberos.
* **Kỹ thuật bẻ khóa mật khẩu tiên tiến:** Công cụ này sử dụng nhiều phương pháp bẻ khóa mật khẩu như tấn công từ điển (dictionary attack), tấn công brute force, và các phương pháp lai kết hợp cả hai. Điều này giúp tăng hiệu quả và tốc độ bẻ khóa mật khẩu.
* **Tùy chỉnh cao:** Người dùng có thể tùy chỉnh các quy tắc bẻ khóa, thêm từ điển mới, và cấu hình các tham số khác để phù hợp với nhu cầu cụ thể. John the Ripper cho phép người dùng chỉnh sửa tập tin cấu hình để tối ưu hóa quá trình bẻ khóa mật khẩu.
* **Khả năng mở rộng:** Với sự hỗ trợ của các module bổ sung, John the Ripper có thể được mở rộng để hỗ trợ thêm nhiều định dạng mật khẩu và kỹ thuật bẻ khóa mới.

## 3.Cách cài đặt

John the Ripper có thể được cài đặt trên nhiều hệ điều hành khác nhau, bao gồm Linux, Windows, và macOS. Người dùng có thể tải xuống phiên bản mã nguồn hoặc các gói phần mềm có sẵn để cài đặt. Quá trình cài đặt thường khá đơn giản và được hỗ trợ bởi các tài liệu hướng dẫn chi tiết.

### 3.1.Trên Linux

#### *3.1.1Cài đặt qua trình quản lý gói:*

* Với Ubuntu/Debian:

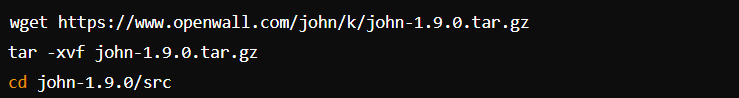


* Với Fedora:



#### *3.1.2.Cài đặt từ mã nguồn:*

* Tải xuống và giải nén:



* Biên dịch:



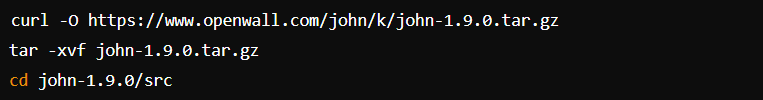
### 3.2.Trên MacOS

#### *3.2.1.Cài đặt qua Homebrew:*



#### *3.2.2. Cài đặt từ mã nguồn:*

* Tải xuống và giải nén:



* Biên dịch:



### 3.2.Trên Windows

**Tải xuống phiên bản Windows:**

* Tải file zip từ trang chủ: Openwall
* Giải nén file zip vào thư mục mong muốn.

**Chạy John the Ripper:**

* Mở Command Prompt và điều hướng đến thư mục John the Ripper đã giải nén.
* Chạy lệnh:



## 4.Cách sử dụng cơ bản

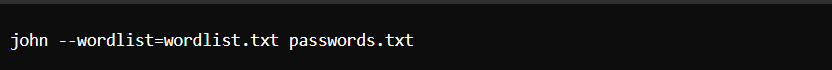
### 4.1. Chạy John the Ripper:

* Mở terminal hoặc command prompt và nhập lệnh sau để kiểm tra xem John the Ripper đã được cài đặt đúng cách chưa:



### 4.2. Kiểm tra mật khẩu bằng wordlists:

Giả sử bạn có một file mật khẩu được mã hóa (hash) tên là passwords.txt và một file từ điển wordlist.txt.



### 4.3. Kiểm tra mật khẩu bằng Incremental Mode:

Nếu không có file từ điển, bạn có thể sử dụng chế độ tấn công brute-force (Incremental Mode):

******

### 4.4. Kiểm tra kết quả bẻ khóa:

Để xem các mật khẩu đã bẻ khóa được, sử dụng lệnh sau:



### 4.5. Cấu hình tùy chỉnh:

* John the Ripper có thể được tùy chỉnh thông qua file cấu hình john.conf.
* Thêm từ điển và các quy tắc bẻ khóa tùy chỉnh vào file cấu hình để cải thiện hiệu quả bẻ khóa.

## 5.Cấu hình và tùy chỉnh

John the Ripper sử dụng một tập tin cấu hình chính thường được gọi là john.conf. Tập tin này chứa các thiết lập mặc định và các quy tắc bẻ khóa mật khẩu. Người dùng có thể chỉnh sửa tập tin này để thay đổi hành vi của John the Ripper.

Để chỉnh sửa tập tin cấu hình (john.conf), bạn có thể làm theo các bước sau:

1. **Định vị tập tin cấu hình:** Tập tin cấu hình thường được đặt trong thư mục gốc của John the Ripper hoặc trong một thư mục cấu hình riêng biệt.
2. **Mở tập tin cấu hình:** Sử dụng trình soạn thảo văn bản để mở tập tin cấu hình (ví dụ: nano, vi, hoặc gedit).
3. **Thực hiện các chỉnh sửa:** Trong tập tin cấu hình, bạn có thể tìm thấy các tùy chọn và cài đặt khác nhau cho John the Ripper. Một số tùy chọn quan trọng bao gồm:
   * Incremental: Điều chỉnh cài đặt cho chế độ tăng dần, bao gồm các quy tắc và bộ ký tự được sử dụng.
   * External: Cấu hình các công cụ hoặc phương pháp bên ngoài được sử dụng để tạo ra các mật khẩu hoặc tập từ điển.
   * Options: Các tùy chọn chung khác như cấu hình về vùng nhớ, hiển thị kết quả, và các cài đặt khác.
4. **Lưu và đóng tập tin:** Lưu các thay đổi bạn đã thực hiện trong tập tin cấu hình và đóng trình soạn thảo văn bản.

## 6.Các Plugin và module bổ sung

John the Ripper hỗ trợ nhiều plugin và module bổ sung để mở rộng khả năng và tính linh hoạt của công cụ. Dưới đây là một số plugin và module phổ biến:

1. **Jumbo Patch:** Được xây dựng trên phiên bản chính thức của John the Ripper, Jumbo Patch bao gồm nhiều tính năng và cải tiến mới, bao gồm hỗ trợ cho các định dạng mật khẩu mới, các tùy chọn tùy chỉnh mạnh mẽ, và nhiều tính năng khác.
2. **GPU Acceleration Module:** Module này cho phép sử dụng GPU để tăng tốc quá trình crack mật khẩu, giúp tăng hiệu suất đáng kể so với việc chỉ sử dụng CPU.
3. **Distributed Computing Module:** Module này cho phép phân tán công việc crack mật khẩu trên nhiều máy tính, tận dụng sức mạnh tính toán của một mạng lưới máy tính để giảm thời gian cần thiết để tìm kiếm mật khẩu.
4. **Incremental Mode and Markov Mode:** Đây là các module cho phép thử các mật khẩu bằng cách tạo ra các mẫu dựa trên các dấu hiệu thống kê từ tập dữ liệu mật khẩu đã biết.
5. **External Mode:** Cho phép sử dụng các chương trình bên ngoài để sinh mật khẩu hoặc các tập từ điển, mở rộng khả năng tùy chỉnh và sử dụng các tài nguyên bên ngoài của John the Ripper.
6. **Phần mở rộng cho định dạng mật khẩu:** Cung cấp hỗ trợ cho các định dạng mật khẩu cụ thể, bao gồm các định dạng hiện đại như bcrypt, PBKDF2, và Argon2.

Các plugin và module này cung cấp cho người dùng nhiều tùy chọn và tính năng để tối ưu hóa việc crack mật khẩu và tăng hiệu suất của John the Ripper.

## 7.Các kịch bản tấn công

* **Tấn công mật khẩu hệ thống (Unix, Windows, LDAP)**:
* Đây là kịch bản trong đó kẻ tấn công cố gắng đoán hoặc tìm cách thu thập thông tin về các tài khoản người dùng và mật khẩu hệ thống, chẳng hạn như sử dụng phương pháp "brute force" (liệt kê các mật khẩu có thể) hoặc "dictionary attack" (sử dụng từ điển mật khẩu phổ biến).
* Khi kẻ tấn công thu được quyền truy cập vào hệ thống, họ có thể thực hiện các hành động gây hại như lấy thông tin nhạy cảm, cài đặt phần mềm độc hại hoặc thậm chí kiểm soát hệ thống.
* **Tấn công mật khẩu ứng dụng (cơ sở dữ liệu, các dịch vụ web)**:
* Tương tự như tấn công mật khẩu hệ thống, nhưng nhắm vào các ứng dụng cụ thể như cơ sở dữ liệu (như MySQL, PostgreSQL) hoặc các dịch vụ web (như Apache, nginx).
* Kẻ tấn công thường sử dụng các kỹ thuật giống như brute force, dictionary attacks hoặc thậm chí là sử dụng các lỗ hổng trong phần mềm để lấy được thông tin đăng nhập.

## 8.Bảo mật và biện pháp phòng ngừa

để bảo vệ mật khẩu chống lại các công cụ như John the Ripper, bạn có thể áp dụng một số biện pháp sau:

1. **Sử dụng mật khẩu mạnh**:
   * Mật khẩu mạnh là một yếu tố quan trọng trong việc chống lại các cuộc tấn công brute force hoặc dictionary attacks. Sử dụng mật khẩu bao gồm cả chữ hoa, chữ thường, ký tự đặc biệt và số, và có độ dài đủ lớn (ít nhất 12 ký tự) có thể làm khó khăn việc thực hiện các cuộc tấn công này.
2. **Sử dụng các thuật toán băm mật khẩu mạnh mẽ**:
   * Các hệ thống thường sử dụng các thuật toán băm mật khẩu như bcrypt, scrypt hoặc PBKDF2 để lưu trữ mật khẩu người dùng. Các thuật toán này tạo ra các mã băm mật khẩu có độ dài lớn và khó để giải mã ngược lại thành mật khẩu gốc.
3. **Sử dụng salt**:
   * Salt là một chuỗi ngẫu nhiên được thêm vào mật khẩu trước khi thực hiện quá trình băm. Việc này làm tăng tính ngẫu nhiên và độ khó cho các cuộc tấn công dựa trên bảng rainbow (rainbow table) hoặc các kỹ thuật tương tự.
4. **Thực hiện chính sách mật khẩu mạnh**:
   * Thiết lập chính sách mật khẩu mạnh trong tổ chức của bạn, bao gồm việc yêu cầu người dùng thay đổi mật khẩu định kỳ, không sử dụng mật khẩu cũ, và hạn chế việc sử dụng mật khẩu dễ đoán.

Đối với việc sử dụng John the Ripper để tăng cường bảo mật hệ thống, bạn có thể sử dụng công cụ này như một phương tiện kiểm thử mật khẩu. Bằng cách chạy John the Ripper trên cơ sở dữ liệu mật khẩu của bạn, bạn có thể phát hiện ra mật khẩu yếu và thiếu an toàn và từ đó cập nhật hoặc thay thế chúng bằng các mật khẩu mạnh hơn.

Tuy nhiên, việc sử dụng John the Ripper phải được thực hiện với sự cho phép của người quản trị hệ thống và tuân thủ các quy định và luật lệ bảo mật.

Việc nắm rõ cái nhìn tổng quan sẽ giúp người dùng khai thác tối đa sức mạnh của John the Ripper, đồng thời áp dụng được các biện pháp bảo mật phù hợp để bảo vệ hệ thống của mình.

# CHƯƠNG 2: CÁC CHẾ ĐỘ BẺ KHÓA MẬT KHẨU TRONG

# JOHN THE RIPPER

## 1. Single Crack Mode

### 1.1. Tổng quan về Single Crack Mode :

Single Crack Mode là chế độ tấn công mặc định trong John the Ripper. Khác với các phương pháp tấn công dựa trên từ điển hoặc brute force, Single Crack Mode khai thác các thông tin cụ thể về người dùng được lấy từ các tập tin hệ thống như /etc/passwd và /etc/shadow trên Unix/Linux. Các thông tin này bao gồm tên người dùng, tên đầy đủ, tên miền, và các thông tin cá nhân khác có thể được sử dụng để dự đoán mật khẩu đồng thời áp dụng một bộ lớn các quy tắc xáo trộn. Mật khẩu đoán thành công cũng được thử dựa trên tất cả các hàm băm mật khẩu đã tải để đề phòng trường hợp có nhiều người dùng có cùng mật khẩu.

### 1.2. Cách thức hoạt động :

#### 1.2.1 Sử dụng thông tin người dùng:

John the Ripper lấy thông tin từ các trường trong tập tin mật khẩu, chẳng hạn như tên người dùng (username), tên thật (real name), và các thông tin khác.

Dựa trên thông tin này, công cụ tạo ra các biến thể của mật khẩu có khả năng được người dùng sử dụng. Ví dụ, nếu tên người dùng là "john", các mật khẩu như "john123", "johnsmith", "smithjohn", v.v. sẽ được thử.

#### 1.2.2 Áp dụng quy tắc biến đổi:

John the Ripper sử dụng một tập hợp các quy tắc biến đổi để tạo ra nhiều biến thể khác nhau từ các từ gốc. Các quy tắc này có thể bao gồm việc thêm số, thêm ký tự đặc biệt, thay đổi chữ hoa chữ thường, đảo ngược chữ, v.v.

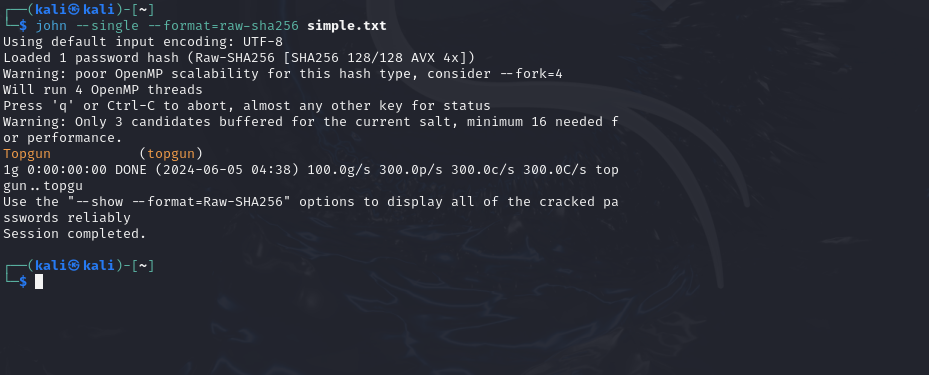
Ví dụ, với từ gốc "john", các quy tắc có thể tạo ra "john1", "John123", "nhoj", "john!", "JOHN", v.v.

#### 1.2.3.Tận dụng mật khẩu đã bẻ khóa:

Nếu John the Ripper đã bẻ khóa được một số mật khẩu, các mật khẩu này sẽ được sử dụng như các từ gốc để tạo ra các biến thể mới. Điều này giúp cải thiện hiệu suất bẻ khóa, đặc biệt trong trường hợp nhiều người dùng sử dụng mật khẩu tương tự.

### 1.3. Thực nghiệm :





* Chỉ định một chuỗi ngắn (topgun) làm tên người dùng và các biến thể viết hoa của nó làm mật khẩu (chẳng hạn như Topgun).
* Hiển thị đầu ra của mật khẩu băm SHA-256: echo -n 'Topgun' | sha256sum
* Tạo tệp văn bản mới (simple.txt) để lưu trữ tên người dùng và giá trị băm mật khẩu từ các bước trước: echo -n 'topgun:4558ce5abe3b1e70bbadc3b95f2ff84f54d0a5c30fb524ceebfd401f8233fda7' > simple.txt
* Chạy simple.txt qua Chế độ Single Crack của John the Ripper (thay đổi đối số --format tùy theo ý bạn): john --single --format=raw-sha256 simple.txt
* Nhận kết quả.

### 1.4. Ưu điểm :

* **Hiệu quả cao:** Chế độ này thường rất hiệu quả khi bẻ khóa các mật khẩu đơn giản và phổ biến do người dùng đặt dựa trên thông tin cá nhân.
* **Nhanh chóng:** Vì tập trung vào các mật khẩu có khả năng cao, Single Crack Mode thường nhanh hơn so với các chế độ khác như brute-force hoặc tấn công từ điển.

### 1.5.Hạn chế :

* **Giới hạn bởi thông tin có sẵn:** Hiệu quả của Single Crack Mode phụ thuộc nhiều vào chất lượng và số lượng thông tin người dùng có sẵn trong tập tin mật khẩu. Nếu thông tin này hạn chế hoặc không đầy đủ, chế độ này có thể không đạt hiệu quả cao.
* **Không hiệu quả với mật khẩu phức tạp:** Với các mật khẩu ngẫu nhiên hoặc rất phức tạp không dựa trên thông tin cá nhân, Single Crack Mode thường kém hiệu quả hơn.

## 2. Wordlist Mode

### 2.1. Tổng quan về Wordlist Mode :

Wordlist Mode là một trong những chế độ quan trọng và phổ biến nhất của John the Ripper, được sử dụng để bẻ khóa mật khẩu bằng cách so sánh chúng với một danh sách từ điển có sẵn.

* **Định nghĩa:** Wordlist Mode sử dụng một danh sách các từ (wordlist) để kiểm tra mật khẩu. Mỗi từ trong danh sách sẽ được thử làm mật khẩu cho đến khi tìm thấy mật khẩu đúng hoặc hết danh sách.
* **Ứng dụng:** Thường được sử dụng khi có một từ điển các mật khẩu phổ biến hoặc các mật khẩu có thể đoán trước.

### 2.2. Cách thức hoạt động:

#### 2.2.1.Chuẩn bị danh sách từ (wordlist):

* **Tạo hoặc chọn wordlist:** Wordlist có thể là một file chứa danh sách các mật khẩu phổ biến, được lấy từ các nguồn trên Internet hoặc tự tạo.
* **Định dạng:** Mỗi từ trong wordlist phải nằm trên một dòng riêng biệt. File wordlist thường có định dạng .txt.

#### 2.2.2.Chuẩn bị file mật khẩu mã hóa (hash):

* **Thu thập hashes:** Hashes của các mật khẩu có thể được thu thập từ các hệ thống hoặc ứng dụng cần kiểm tra.
* **Định dạng file:** File chứa hashes thường có định dạng đơn giản, mỗi hash nằm trên một dòng hoặc theo cấu trúc nhất định (như file /etc/shadow trên hệ thống Unix).

#### 2.2.3.Chạy John The Ripper với Wordlist Mode:

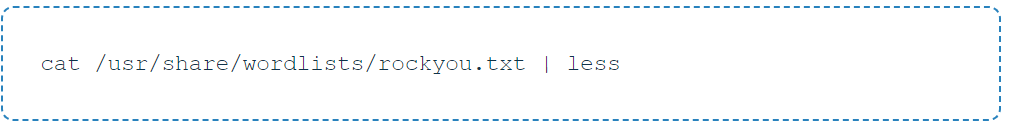
* Cú pháp lệnh cơ bản:



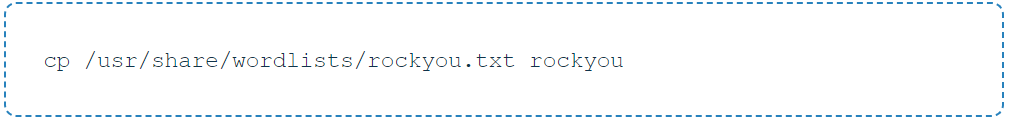
* **Giải thích cú pháp:**
* john: Lệnh chạy John the Ripper.
* --wordlist=wordlist.txt: Chỉ định file wordlist chứa các mật khẩu cần thử.
* passwords.txt: File chứa các hashes cần bẻ khóa.

### 2.3. Thực nghiệm :

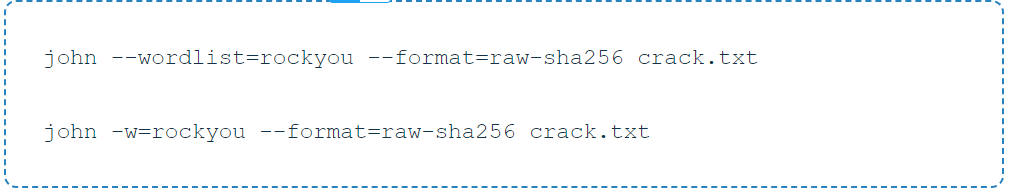
* Trong Chế độ danh sách từ, chúng tôi sẽ cung cấp cho John một danh sách mật khẩu. John sẽ tạo các hàm băm cho chúng theo thời gian thực và so sánh chúng với hàm băm mật khẩu của chúng tôi. Trong ví dụ này, chúng tôi sẽ sử dụng danh sách từ RockYou nổi tiếng, mà bạn có thể xem trước tại

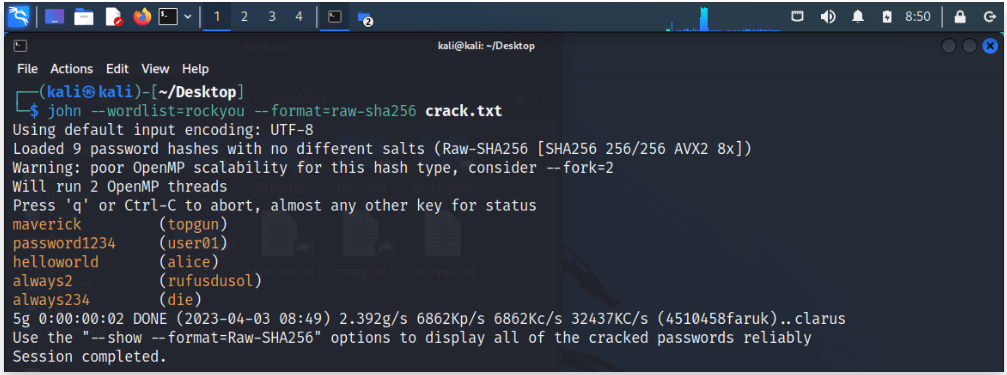


* Bạn có thể sao chép nó vào thư mục làm việc hiện tại của mình để đơn giản hóa các lệnh bằng cách sử dụng cờ --wordlist (-w):

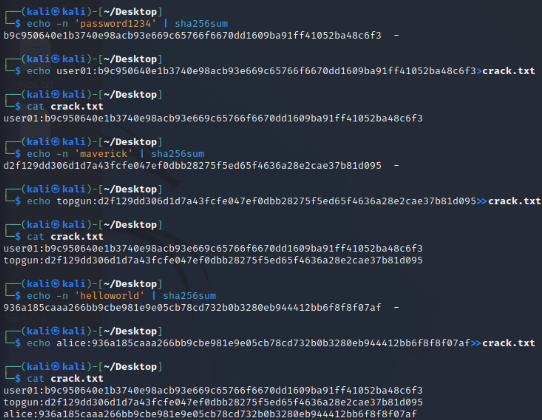


* Bây giờ chúng ta hãy truyền các tệp văn bản chứa mã băm mật khẩu qua John:





* Chuyển một hàm băm dựa trên một hoặc nhiều từ trong từ điển (tùy chọn có số) sang SHA-256: echo -n 'password1234' | sha256sum
* Ghi tên người dùng, dấu hai chấm (:) và hàm băm thành một chuỗi dài duy nhất vào tệp văn bản mới (crack.txt): echo user01:b9c950640e1b3740e98acb93e669c65766f6670dd1609ba91ff41052ba48c6f3>>crack.txt
* Lặp lại Bước 1 và 2 để tạo nhiều cặp tên người dùng-mật khẩu tùy ý và thêm chúng vào crack.txt.
* Chạy crack.txt qua Chế độ danh sách từ của John the Ripper: john --wordlist=rockyou --format=raw-sha256 crack.txt





### 2.4. Ưu điểm :

* **Hiệu quả cao đối với mật khẩu phổ biến:**

Wordlist Mode sử dụng một danh sách các mật khẩu phổ biến (từ điển), vì vậy nó có khả năng tìm thấy mật khẩu nhanh chóng nếu mật khẩu nằm trong danh sách này. Điều này đặc biệt hiệu quả đối với các mật khẩu đơn giản hoặc phổ biến.

* **Thời gian bẻ khóa ngắn:**

So với các phương pháp bẻ khóa như Brute Force (thử mọi kết hợp có thể), Wordlist Mode có thể nhanh hơn rất nhiều vì nó chỉ thử các mật khẩu có khả năng cao.

* **Dễ dàng tùy chỉnh và mở rộng:**

Người dùng có thể dễ dàng tạo hoặc tải xuống các tệp từ điển tùy chỉnh chứa các mật khẩu dự đoán hoặc thu thập từ các nguồn khác nhau. Điều này giúp cải thiện khả năng bẻ khóa.

* **Khả năng kết hợp với các quy tắc (Rules):**

John the Ripper cho phép áp dụng các quy tắc để biến đổi các mục trong từ điển, như thêm số, thay đổi chữ hoa/thường, và thêm ký tự đặc biệt. Điều này tăng khả năng thành công khi mật khẩu là biến thể của các từ trong từ điển.

* **Dễ dàng sử dụng:**

Cách sử dụng Wordlist Mode khá đơn giản và thân thiện với người dùng. Chỉ cần một lệnh cơ bản với chỉ định tệp từ điển và tệp chứa mật khẩu mã hóa.

* **Tiết kiệm tài nguyên:**

Do chỉ thử các mật khẩu trong danh sách từ điển, Wordlist Mode tiết kiệm tài nguyên hơn so với các phương pháp khác như Brute Force, vốn yêu cầu thử rất nhiều kết hợp khác nhau.

### 2.5. Hạn chế :

* **Phụ thuộc vào danh sách từ điển:** Wordlist Mode hoạt động tốt khi bạn có một danh sách từ điển phong phú và phù hợp với ngữ cảnh. Nếu danh sách từ điển của bạn hạn chế hoặc không chứa mật khẩu mong muốn, nó sẽ không hiệu quả.
* **Không hiệu quả đối với mật khẩu mạnh:** Đối với mật khẩu có độ dài lớn hoặc sử dụng kỹ thuật phức tạp như ký tự đặc biệt, chữ hoa, chữ thường xen kẽ, Wordlist Mode có thể không hiệu quả. Cần phải sử dụng các phương pháp khác như Brute Force để tìm ra mật khẩu trong trường hợp này.
* **Không thể tìm ra mật khẩu không nằm trong danh sách từ điển:** Nếu mật khẩu không nằm trong danh sách từ điển thì Wordlist Mode sẽ không có khả năng tìm ra mật khẩu đó. Điều này đặc biệt đúng khi người dùng sử dụng mật khẩu ngẫu nhiên hoặc không phổ biến.
* **Đòi hỏi kiến thức trước:** Wordlist Mode không tự động tạo ra các từ điển. Người dùng cần phải có kiến thức trước hoặc sử dụng các công cụ khác để tạo ra các danh sách từ điển phù hợp.
* **Yêu cầu thời gian:** Quá trình thử các mật khẩu từ danh sách từ điển có thể mất rất nhiều thời gian, đặc biệt là nếu danh sách từ điển lớn và mật khẩu nằm ở cuối danh sách.
* **Khả năng phát hiện bị hạn chế:** Nếu mật khẩu được bảo vệ bằng các biện pháp bảo mật như khóa tài khoản sau một số lần thử sai, Wordlist Mode có thể trở nên vô ích sau một số lần thử sai.

## 3. Incremental Mode

### 3.1. Tổng quan về Incremental Mode:

### Định nghĩa:

Incremental Mode là một phương pháp trong việc bẻ khóa mật khẩu sử dụng trong công cụ John The Ripper. Trong phương pháp này, John The Ripper sẽ thử tất cả các ký tự, từ hoặc số có thể có trong mật khẩu một cách tuần tự hoặc theo một quy tắc được định nghĩa trước, nhằm tăng hiệu suất tìm ra mật khẩu đúng.

### Mục đích và ứng dụng của Incremental Mode:

1. **Tìm kiếm mật khẩu một cách tự động**: Incremental Mode cho phép John The Ripper tự động thử các ký tự, từ hoặc số khác nhau để tìm ra mật khẩu chính xác. Điều này giúp tiết kiệm thời gian so với việc thử từng ký tự một cách thủ công.
2. **Dùng cho việc phục hồi mật khẩu đã quên**: Incremental Mode được sử dụng phổ biến trong việc phục hồi mật khẩu đã quên trong các hệ thống máy tính hoặc ứng dụng. Bằng cách thử các ký tự một cách tự động, người dùng có thể tìm lại mật khẩu mà họ đã quên mà không cần phải nhớ lại hoặc dùng các phương pháp khác như từ điển hay tấn công tập hợp mật khẩu.
3. **Tăng cơ hội thành công**: Incremental Mode tăng cơ hội tìm ra mật khẩu chính xác bằng cách thử nghiệm một loạt các ký tự, từ hoặc số, không giới hạn bởi những hạn chế về độ dài hoặc định dạng của mật khẩu.
4. **Phục hồi mật khẩu trong các tập tin mã hóa**: Khi áp dụng cho việc phục hồi mật khẩu trong các tập tin hoặc hệ thống mã hóa, Incremental Mode cung cấp một phương tiện hiệu quả để thử nghiệm các giả định về mật khẩu có thể có.

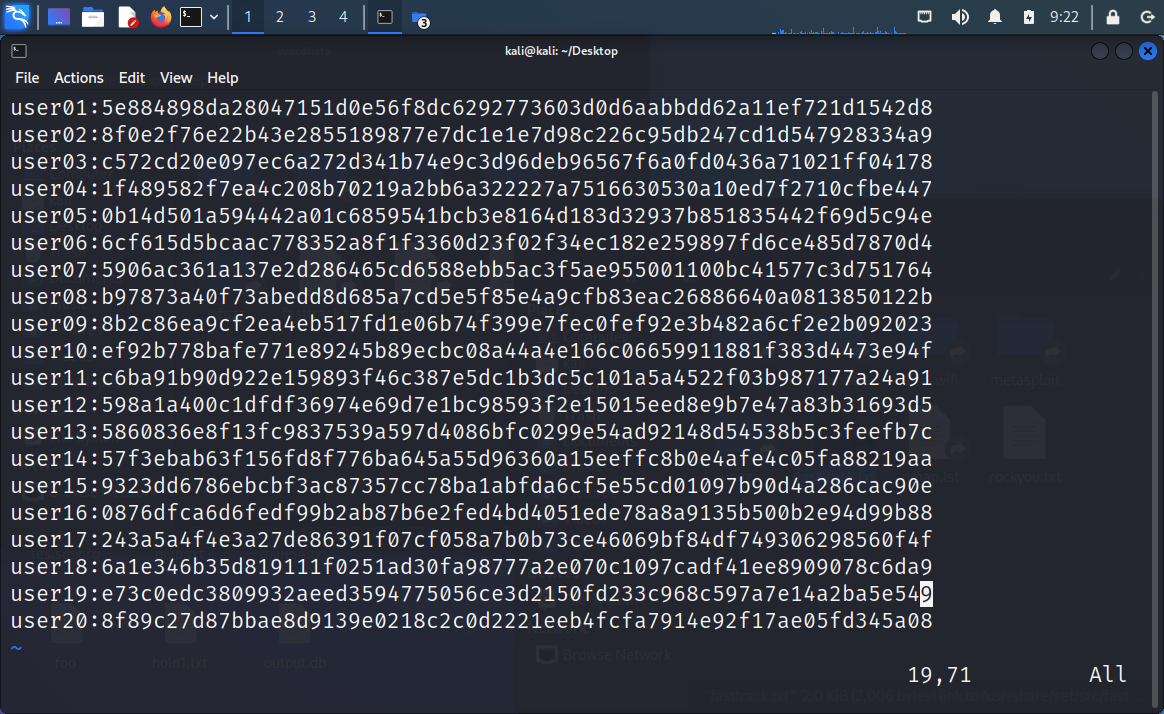
### 3.2. Cách thức hoạt động :

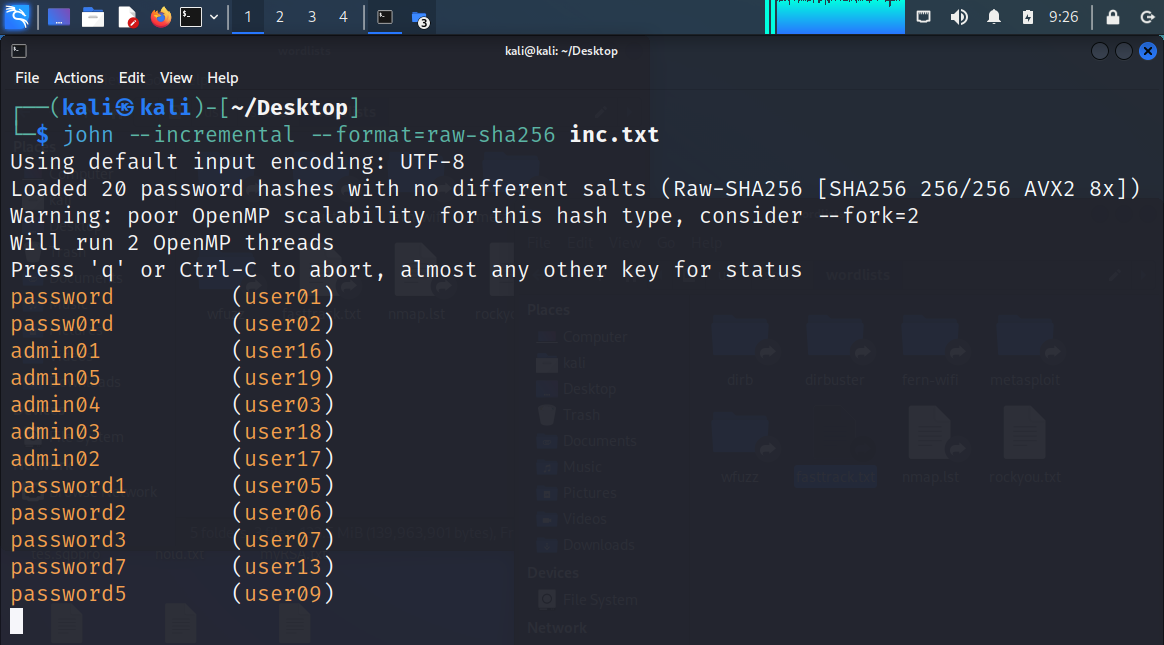
Incremental Mode trong John the Rippe hoạt động bằng cách thử tất cả các kết hợp ký tự có thể theo một trình tự được xác định trước. Đây là một phương pháp tấn công brute force mạnh mẽ nhất nhưng cũng rất tốn thời gian và tài nguyên. Dưới đây là cách thức hoạt động cụ thể của Incremental Mode:

* Tạo tất cả các kết hợp ký tự có thể Incremental Mode thử tất cả các kết hợp của một bộ ký tự đã xác định. Ví dụ: Nếu bộ ký tự là "abc", nó sẽ thử các mật khẩu như "a", "b", "c", "aa", "ab", "ac", "ba", "bb", "bc", v.v.
* Độ dài mật khẩu tăng dần John bắt đầu với các mật khẩu có độ dài ngắn nhất và dần dần tăng độ dài lên. Điều này có nghĩa là nó sẽ thử tất cả các mật khẩu có độ dài 1 trước, sau đó đến độ dài 2, rồi 3, và cứ tiếp tục như vậy.
* Tối ưu hóa bằng tần suất xuất hiện của các bộ ba ký tự (trigraph frequencies) John sử dụng tần suất xuất hiện của các bộ ba ký tự để quyết định thứ tự thử các kết hợp. Điều này giúp John thử các mật khẩu có khả năng đúng cao hơn trước.
* Cấu hình chi tiết trong file cấu hình Người dùng có thể tùy chỉnh các thông số của Incremental Mode trong file cấu hình john.conf. Các thông số cấu hình bao gồm bộ ký tự, độ dài tối thiểu và tối đa của mật khẩu, và các quy tắc thử mật khẩu.
* Sử dụng các chế độ incremental có sẵn hoặc tự định nghĩa John cung cấp các chế độ incremental có sẵn như "ASCII", "LM\_ASCII", "Alnum", "Alpha", v.v. Người dùng cũng có thể tự định nghĩa các chế độ incremental tùy chỉnh bằng cách tạo các mục trong file cấu hình với tên [Incremental:MODE].
* Thực hiện tấn công Khi chạy John the Ripper với Incremental Mode, công cụ sẽ bắt đầu thử các kết hợp ký tự theo cấu hình đã định. Quá trình này có thể mất rất nhiều thời gian, đặc biệt nếu mật khẩu dài hoặc bộ ký tự lớn.

### 3.3. Thực nghiệm

* Dùng lệnh sau để băm một mật khẩu đơn giản chứa chữ và số bằng SHA-256: echo -n 'passw0rd' | sha256sum
* Viết tên người dùng của bạn, dấu hai chấm (:), và mã băm thành một chuỗi dài duy nhất vào một tệp văn bản mới (inc.txt): echo user02:8f0e2f76e22b43e2855189877e7dc1e1e7d98c226c95db247cd1d547928334a9>>inc.txt
* Chạy tệp inc.txt qua chế độ Wordlist của John the Ripper: john --incremental --format=raw-sha256 inc.txt
* Nhận kết quả.





### 3.4.Ưu điểm :

* **Hiệu quả:** Khả năng bẻ khóa cao vì thử tất cả các kết hợp ký tự.
* **Linh hoạt:** Có thể tùy chỉnh để phù hợp với các yêu cầu cụ thể của người dùng.
* **Tự động:** Không cần chuẩn bị từ điển từ, phù hợp cho các mật khẩu phức tạp và không phổ biến.

### 3.5 Hạn chế :

* **Thời gian:** Vì phải thử tất cả các kết hợp ký tự, Incremental Mode có thể mất rất nhiều thời gian, đặc biệt đối với mật khẩu dài và phức tạp.
* **Tài nguyên:** Sử dụng nhiều tài nguyên hệ thống (CPU, GPU), có thể ảnh hưởng đến hiệu suất của các ứng dụng khác.

# CHƯƠNG 3: BẺ KHÓA MẬT KHẨU VỚI JOHN THE RIPPER

## 1.Choose a wordlist

Khi sử dụng John the Ripper, việc chọn danh sách từ (wordlist) phù hợp là rất quan trọng để nâng cao khả năng bẻ khóa mật khẩu thành công. Một wordlist tốt chứa nhiều mật khẩu phổ biến, biến thể và các từ khóa thường được sử dụng. Dưới đây là các bước để chọn và sử dụng wordlist trong John the Ripper:

### 1.1. Chọn Wordlist Có Sẵn

John the Ripper đi kèm với một số wordlist cơ bản, nhưng bạn cũng có thể tìm thêm các wordlist trực tuyến. Một số wordlist phổ biến bao gồm:

* **rockyou.txt:** Một trong những wordlist phổ biến nhất, chứa hàng triệu mật khẩu đã bị lộ.
* **SecLists:** Một tập hợp wordlist phong phú từ nhiều nguồn khác nhau.
* **John the Ripper Wordlists:** Đi kèm theo John the Ripper, thường nằm trong thư mục run.

### 1.2. Tải và Chuẩn Bị Wordlist

Bạn có thể tải các wordlist từ các nguồn như GitHub hoặc các trang web bảo mật. Dưới đây là cách tải một số wordlist phổ biến:

* rockyou.txt:



* SecLists:



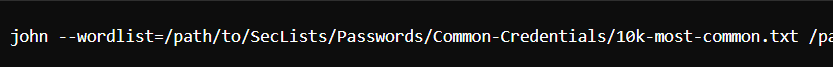
1.3. Sử dụng Wordlist trong John the Ripper

Khi đã có wordlist, bạn có thể sử dụng nó với John the Ripper bằng cách chỉ định đường dẫn đến file wordlist trong lệnh. Ví dụ:

* **Sử dụng wordlist cơ bản:**



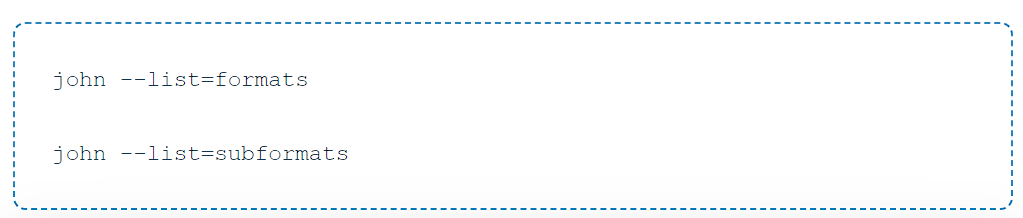
* **Sử dụng wordlist từ SecLists:**



**2. choosing specific hashes to crack**

**2.1.** Xác định loại băm:

Trước khi bắt đầu bẻ khóa, cần xác định loại hàm băm mà bạn đang đối phó. Điều này có thể được thực hiện bằng cách kiểm tra định dạng của hàm băm hoặc sử dụng các công cụ nhận dạng hàm băm trực tuyến.



 Tùy chọn dòng lệnh xem các loại hàm băm John the Ripper hỗ trợ

Hàm băm mật khẩu là một hàm toán học chuyển đổi một mật khẩu thành một chuỗi ký tự cố định, không thể đảo ngược. Các hàm băm phổ biến bao gồm MD5, SHA-1, SHA-256, bcrypt, và nhiều loại khác. Mục tiêu của việc băm mật khẩu là làm cho việc khôi phục mật khẩu gốc trở nên cực kỳ khó khăn nếu chỉ biết hàm băm.

### 2.2 Tạo tệp chứa băm:

Giả sử bạn có một tệp chứa nhiều loại băm khác nhau. Ví dụ, hashes.txt:

### 2.3. Chọn định dạng băm để bẻ khóa:

Sử dụng tùy chọn --format để chỉ định định dạng băm cụ thể mà bạn muốn bẻ khóa.

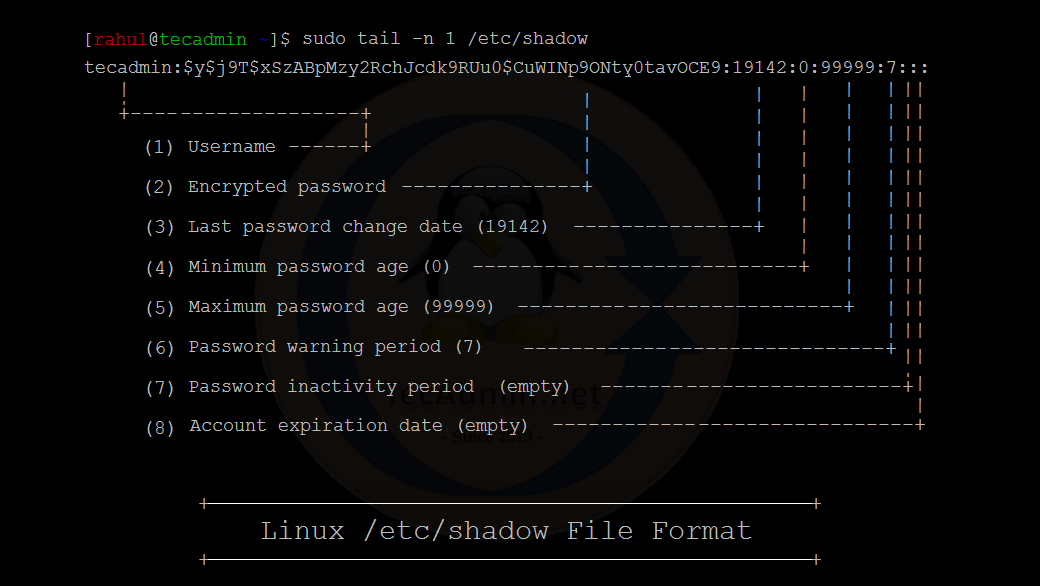
John the Ripper hỗ trợ nhiều loại hàm băm khác nhau và cho phép người dùng chọn lọc cụ thể loại hàm băm mà họ muốn bẻ khóa. Điều này có thể thực hiện bằng cách sử dụng các tùy chọn dòng lệnh hoặc chỉnh sửa các tập tin cấu hình. Việc tập trung vào các hàm băm cụ thể giúp tối ưu hóa quá trình bẻ khóa và giảm thời gian thực hiện.



Hình n. Tùy chọn dòng lệnh xác định loại hàm băm

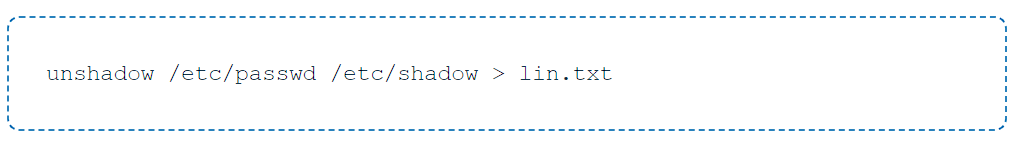
**3**. Cracking Linux Password

Trong hệ thống Linux, mật khẩu của người dùng thường được lưu trữ dưới dạng băm (hash) trong tập tin /etc/shadow. Tập tin này chỉ có quyền truy cập bởi người dùng root và chứa các mục bao gồm tên người dùng, hàm băm mật khẩu, và các thông tin khác như thời gian thay đổi mật khẩu và thời gian hết hạn.



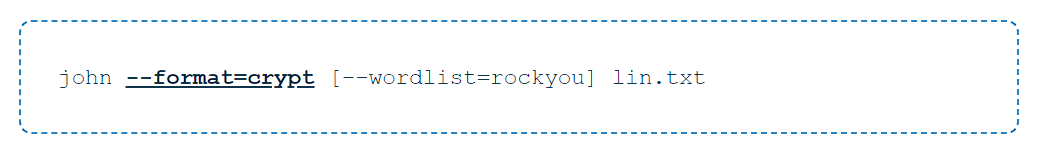
Hình n. Cấu trúc trong file /etc/shadow

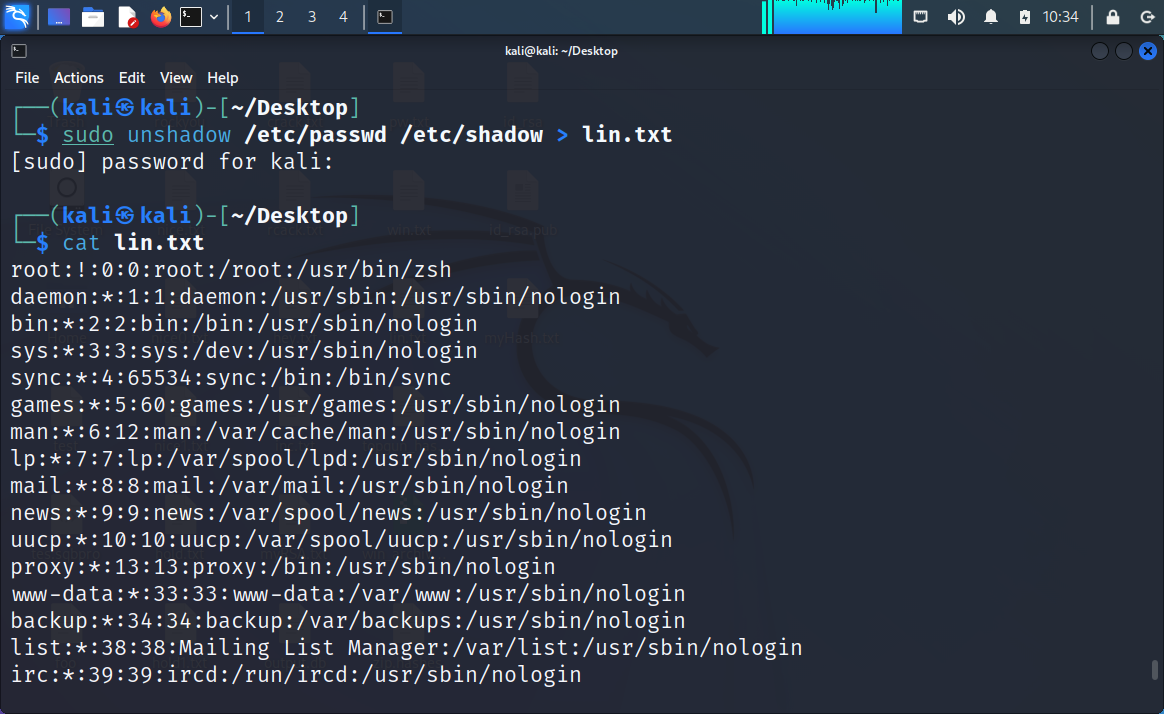
Một tiện ích đi kèm với John the Ripper có tên là ‘unshadow’ kết hợp các tệp /etc/passwd và /etc/shadow để John có thể sử dụng chúng để bẻ khóa chúng. Chúng ta sử dụng cả hai tệp để John có thể sử dụng thông tin được cung cấp nhằm bẻ khóa thông tin đăng nhập một cách hiệu quả cho mọi mục đích sử dụng

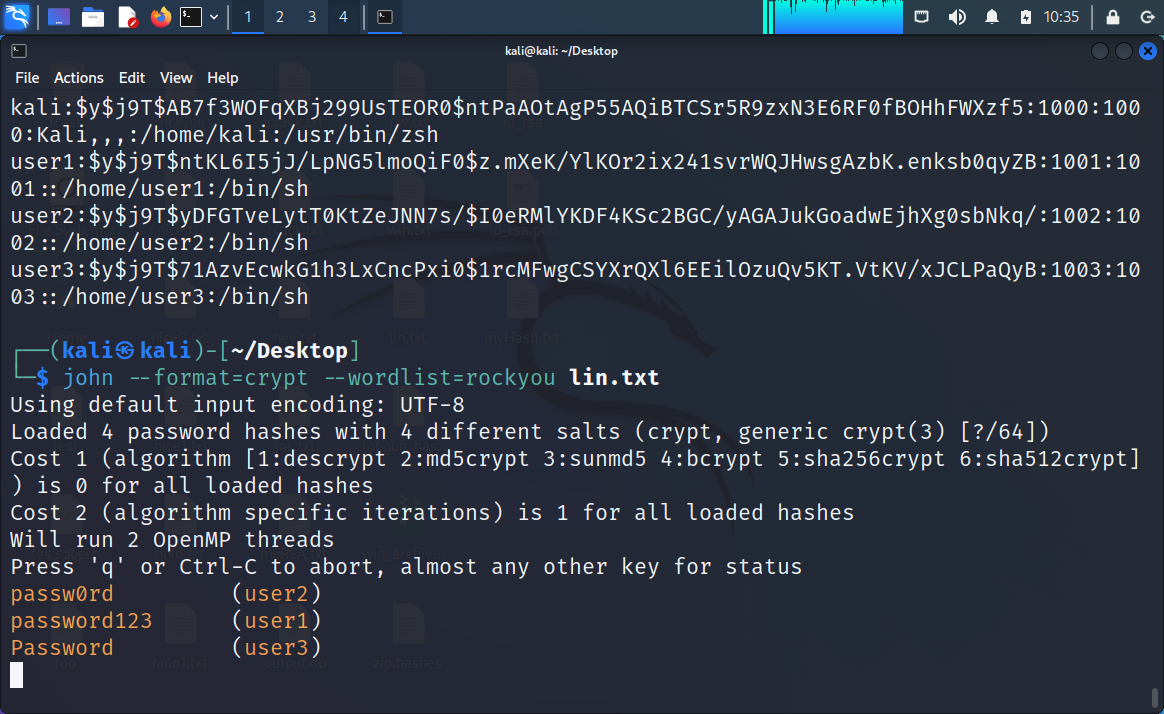


Hình n. Câu lệnh unshadow kết hợp 2 tệp /etc/passwd và /etc/shadow

Việc bẻ khóa băm Linux rất phức tạp; John the Ripper của Kali Linux không dễ dàng phát hiện loại băm của Linux (mật mã), trong đó cờ --wordlist là tùy chọn. Nếu bỏ cờ --format bên dưới, John sẽ không bẻ khóa được gì cả

 Hình n. Câu lệnh John the Ripper bẻ khóa file có nội dung của tệp /etc/shadow





**4. cracking windows passwords**

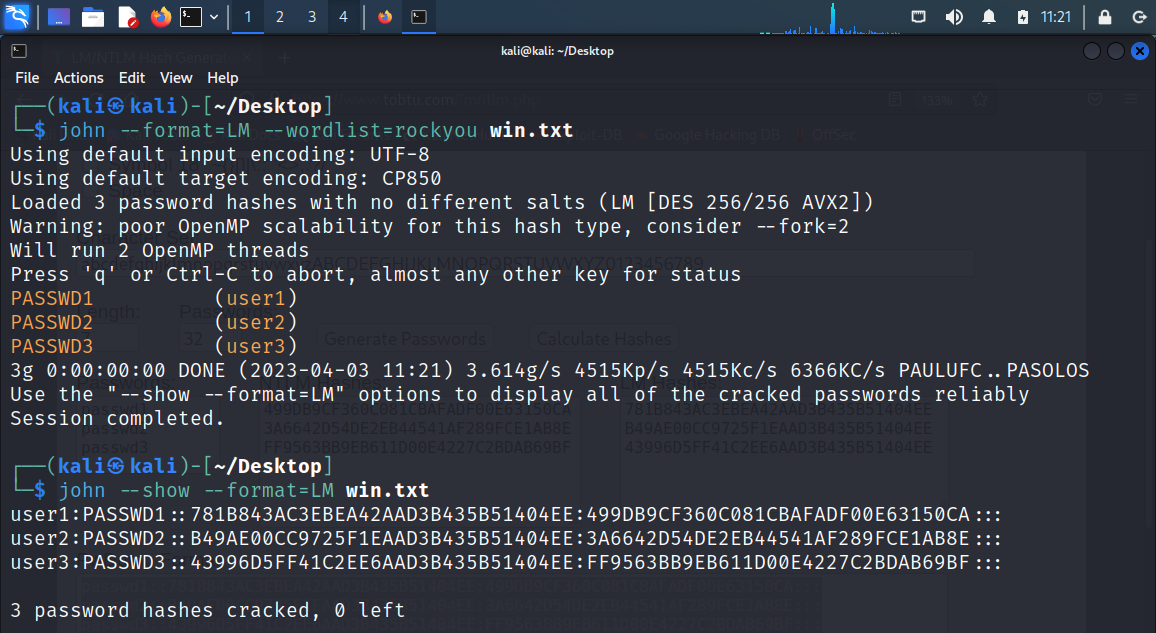
Windows lưu trữ các mật khẩu đã được băm trong cơ sở dữ liệu SAM. SAM sử dụng định dạng băm LM/NTLM cho mật khẩu. Vì việc lấy mật khẩu từ cơ sở dữ liệu SAM vượt quá phạm vi của bài báo này, chúng tôi đề xuất tạo các băm LM/NTLM của riêng bạn để kiểm tra chức năng này và ghi chúng vào một tệp văn bản, ví dụ như win.txt, như minh họa dưới đây:

Cú pháp để bẻ khóa tệp chứa các băm LM/NTLM là như sau, trong đó cờ --wordlist là tùy chọn:



Sau khi John đã tìm ra các mật khẩu, bạn có thể xem chúng bằng lệnh dưới đây:





**5. cracking zip file**

John có một tiện ích gọi là zip2john. zip2john giúp chúng ta lấy hash từ các tệp ZIP. Các tiện ích "2john" khác cũng tồn tại, chẳng hạn như tiện ích rar2john để bẻ khóa tệp RAR.

Để bẻ khóa một tệp ZIP được bảo vệ bằng mật khẩu, trước tiên chúng ta lấy hash của mật khẩu tệp ZIP:

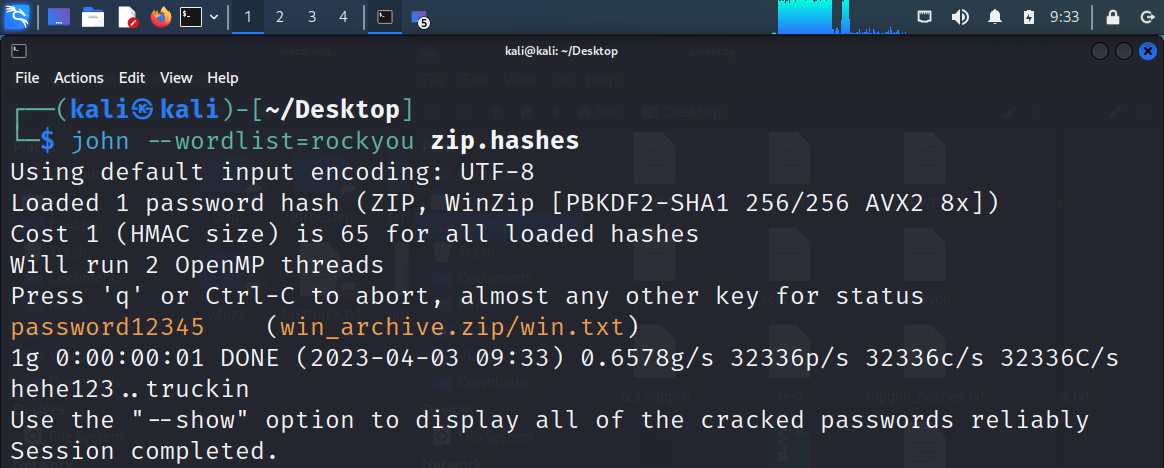
zip2john file.zip > zip.hashes

Lệnh này lấy hash từ tệp ZIP và lưu nó vào tệp zip.hashes.

Bây giờ bạn có thể bẻ khóa hash với John:

john zip.hashes # Chế độ Bẻ Khóa Đơn

john --wordlist=rockyou zip.hashes # Sử dụng danh sách mật khẩu RockYou

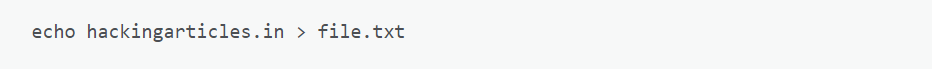


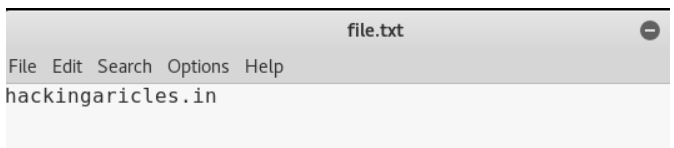
**6. cracking rar file**

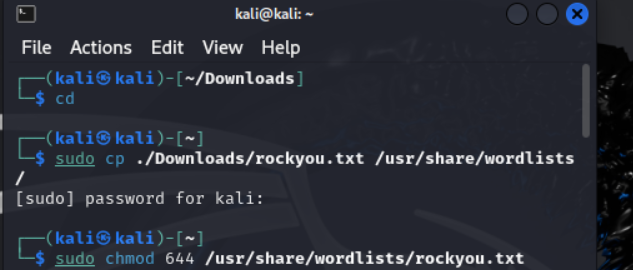
6.1.Chuẩn bị:

Chúng ta sẽ crack một số file nén, để làm được điều đó chúng ta sẽ phải tạo một file để nén, vì vậy hãy thực hiện bằng lệnh echo như trong ảnh chụp màn hình đã cho.

Bạn có thể thấy rằng chúng ta đã tạo một file.txt mà chúng ta sẽ sử dụng để tạo các file nén.



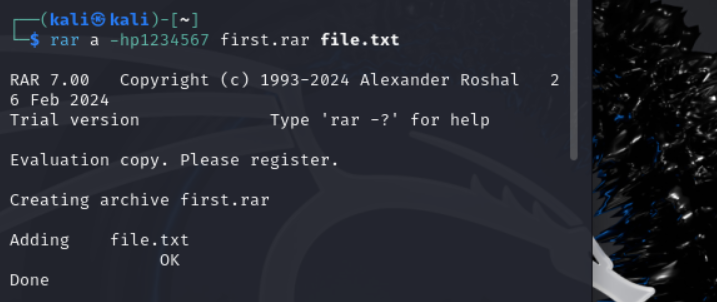




Hình n. Chuẩn bị wordlist rockyou.txt

6.2.Tạo tệp file rar đã được mã hóa nén

John the Ripper có thể bẻ khóa mật khẩu tệp RAR. Để kiểm tra khả năng bẻ khóa mật khẩu, trước tiên, hãy tạo một tệp rar được mã hóa nén.



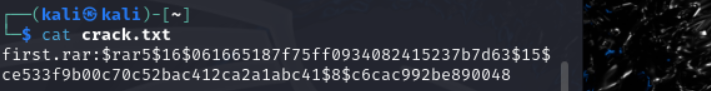
* a = Thêm tệp vào kho lưu trữ
* hp[password] = Mã hóa cả dữ liệu tệp và tiêu đề

Thao tác này sẽ nén và mã hóa file.txt của chúng tôi thành file.rar. Vì vậy, khi bạn thử mở file, bạn sẽ được chào đón bằng lời nhắc sau.

6.3. Trích Xuất Hash từ Tệp RAR:

John the Ripper không làm việc trực tiếp với các tệp RAR. Thay vào đó, bạn cần trích xuất hash từ tệp RAR để John có thể xử lý. Bạn có thể sử dụng công cụ rar2john được cung cấp cùng với John the Ripper cho việc này. Dưới đây là cách thực hiện:

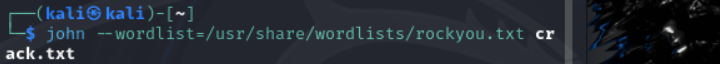


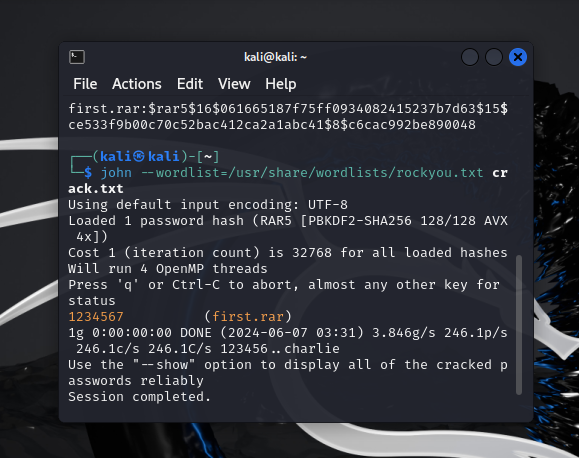


Hình n. File crack.txt giờ đã chứa hash của tệp Rar

6.4.Bẻ khóa Hash

Sau khi có được hash, bạn có thể bắt đầu bẻ khóa mật khẩu bằng John the Ripper. John sẽ bắt đầu quá trình bẻ khóa mật khẩu bằng cách thử các mật khẩu từ từ điển mật khẩu mặc định hoặc bạn có thể chỉ định một từ điển cụ thể bằng cách sử dụng tùy chọn --wordlist.





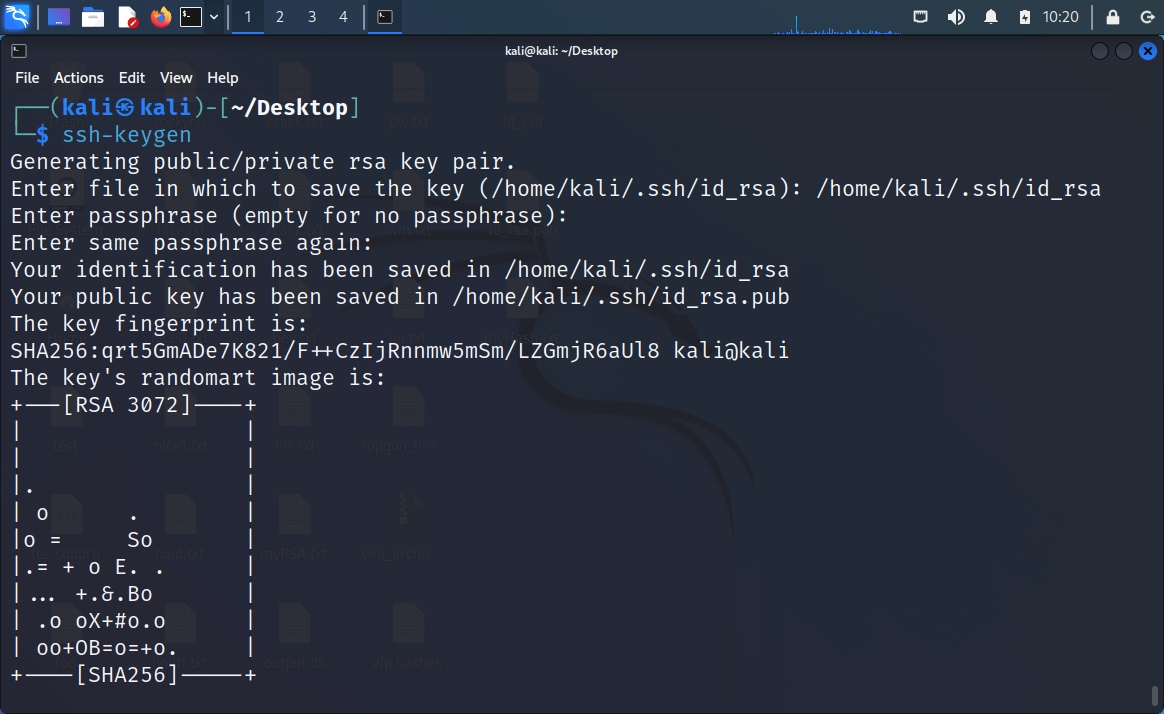
Hình n. Kết quả thu được là mật khẩu của tệp first.rar

## 7.Cracking SSH keys

SSH (Secure Shell) là một giao thức mạng được sử dụng để thực hiện các hoạt động đăng nhập và truyền dữ liệu an toàn giữa các hệ thống máy tính. Các khóa SSH (SSH keys) thường được sử dụng để xác thực người dùng và đảm bảo kết nối an toàn. Tuy nhiên, trong một số tình huống bảo mật, việc bẻ khóa các khóa SSH có thể cần thiết để kiểm tra độ an toàn hoặc khôi phục truy cập. John the Ripper là một công cụ mạnh mẽ giúp bẻ khóa các mật khẩu bảo vệ khóa SSH

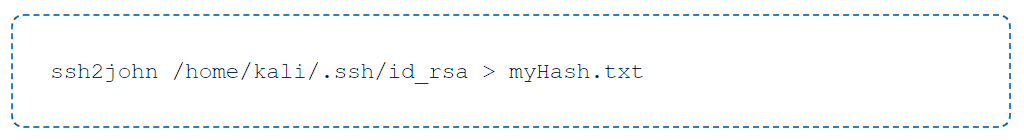
SSH keys gồm hai phần: khóa công khai (public key) và khóa bí mật (private key). Khóa công khai được lưu trữ trên máy chủ và khóa bí mật được giữ bởi người dùng. Khi kết nối SSH được thiết lập, máy chủ sẽ sử dụng khóa công khai để xác minh người dùng sở hữu khóa bí mật tương ứng. Để bảo mật, khóa bí mật thường được bảo vệ bằng mật khẩu (passphrase).

John the Ripper hoạt động bằng cách thử nhiều mật khẩu (passphrase) khác nhau để giải mã khóa bí mật SSH. Nếu mật khẩu đúng được tìm thấy, khóa bí mật có thể được sử dụng để thiết lập kết nối SSH.



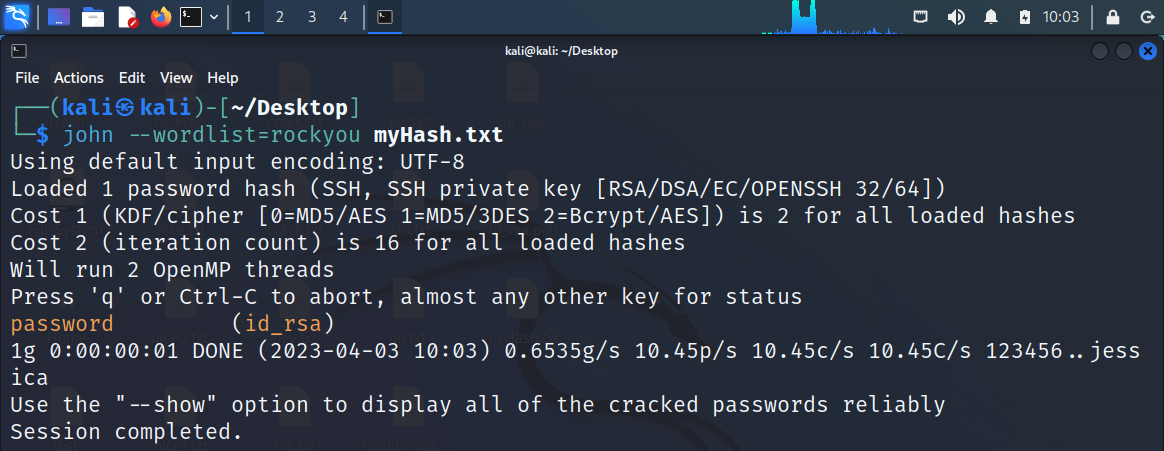
Hình n. Tạo cặp khóa SHH

John the Ripper yêu cầu khóa SSH ở định dạng John the Ripper hiểu được. Tiện ích ‘ssh2john’ có thể giúp chuyển đổi khóa SSH sang định dạng này. Nếu đường dẫn tệp khóa riêng của bạn là /home/kali/.ssh/id\_rsa và bạn muốn lưu trữ hàm băm dưới dạng myHash.txt



Hình n. Tiện ích ssh2john để chuyển đổi định dạng

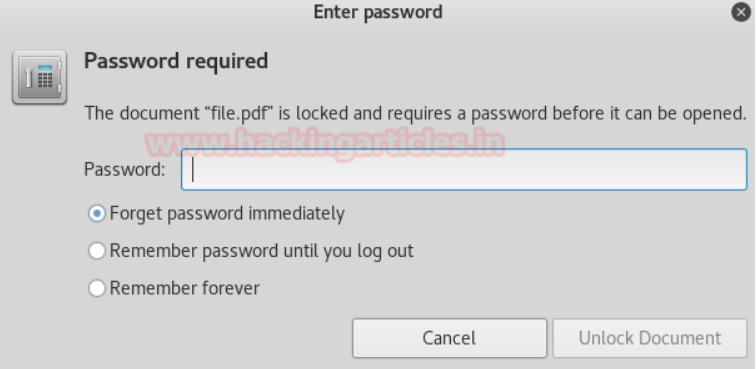
Bây giờ chúng ta sẽ dùng John the Ripper để bẻ khóa SSH



Hì nh n. John the Ripper bẻ khóa thành công khóa SSH

## 8.Cracking PDF file

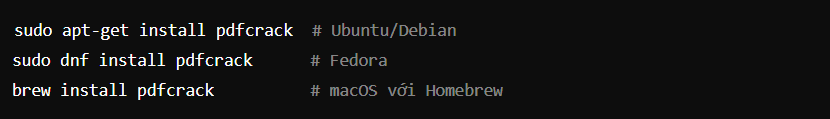
### 8.1. Cài đặt và chuẩn bị

John the Ripper có thể bẻ khóa mật khẩu tệp PDF. Bạn có thể mã hóa tệp PDF trực tuyến bằng cách sử dụng trang web này https://www.sodapdf.com/password-protect-pdf/. Thao tác này sẽ nén và mã hóa tệp PDF của chúng tôi thành tệp được bảo vệ bằng mật khẩu.pdf. Vì vậy, khi bạn thử mở tệp, bạn sẽ được chào đón bằng lời nhắc sau. 

John the Ripper sử dụng một công cụ bổ sung gọi là PDF2John để trích xuất hash từ file PDF.

#### Trên Linux/macOS

* Tải xuống và cài đặt gói **pdfcrack**:



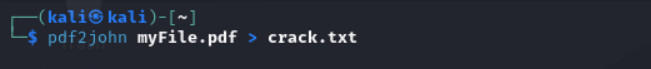
#### Trên Windows

* Tải xuống và cài đặt từ trang chủ của PDF2John: pdfcrack.

### 8.2. Trích xuất hash từ file PDF

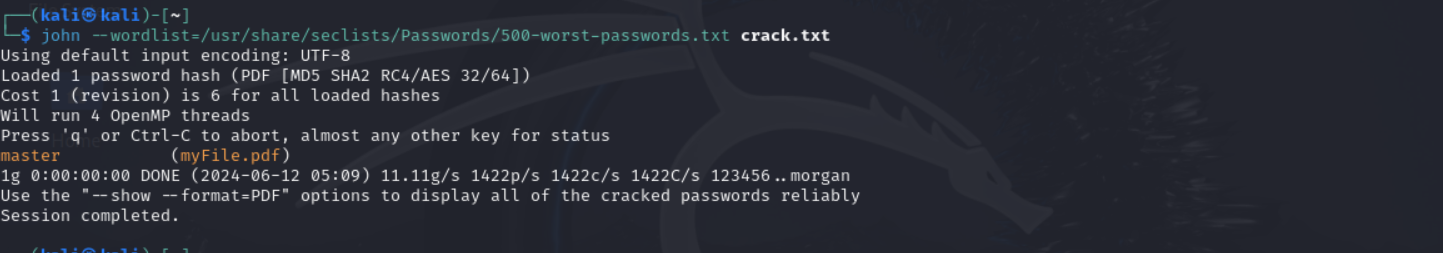
**Sử dụng pdf2john.py**:

* Tải xuống **pdf2john.py** từ [GitHub của John the Ripper](https://github.com/openwall/john/blob/bleeding-jumbo/run/pdf2john.py).
* Chạy script để trích xuất hash từ file PDF:



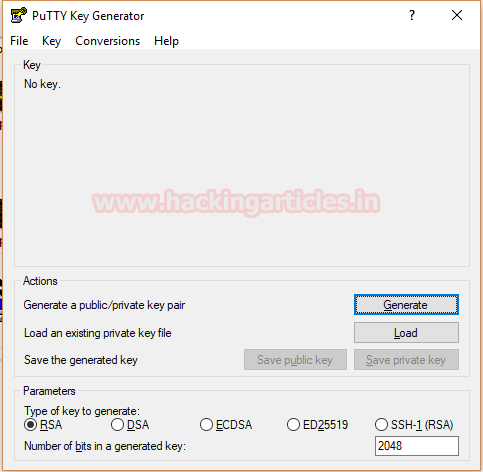


### 8.3.Bẻ khóa mật khẩu PDF

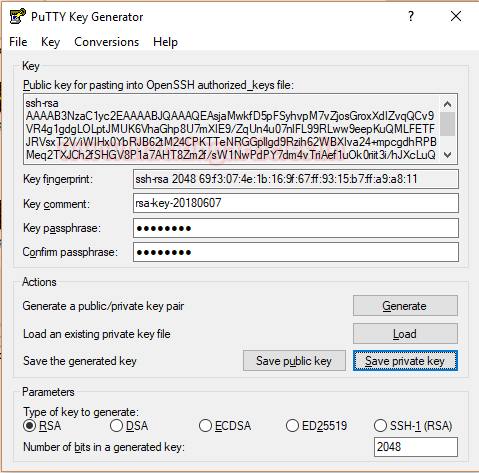


## 9.Cracking Putty

John the Ripper có thể bẻ khóa khóa riêng PuTTY được tạo bằng mã hóa RSA. Để kiểm tra việc bẻ khóa khóa riêng, trước tiên, chúng ta sẽ phải tạo một bộ khóa riêng mới. Để làm điều này, chúng ta sẽ sử dụng một tiện ích đi kèm với PuTTY, gọi là "PuTTY Key Generator".



Nhấp vào "Generate". Sau khi tạo khóa, chúng ta sẽ thấy một cửa sổ nơi chúng ta sẽ nhập mật khẩu cho khóa như trong hình minh họa.



Sau khi nhập mật khẩu, nhấp vào "Save private key" để lưu khóa riêng dưới dạng tệp .ppk.

Sau khi tạo xong, chuyển tệp .ppk này sang Kali Linux.

Bây giờ John không thể trực tiếp bẻ khóa khóa này, trước tiên chúng ta sẽ phải thay đổi định dạng của nó, điều này có thể thực hiện bằng cách sử dụng một tiện ích của john gọi là "putty2john".

Cú pháp:

putty2john [vị trí của khóa]

Ví dụ:

putty2john file.ppk > crack.txt

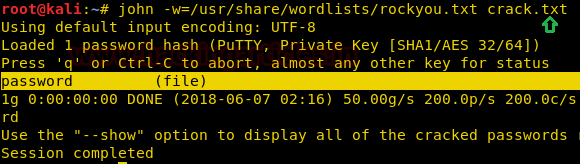


Bạn có thể thấy rằng chúng ta đã chuyển đổi khóa sang một hash có thể bẻ khóa và sau đó nhập nó vào một tệp văn bản tên là crack.txt.

Bây giờ, hãy sử dụng John the Ripper để bẻ khóa hash này.

john –w=/usr/share/wordlists/rockyou.txt crack.txt

Tuyệt vời! Chúng ta đã bẻ khóa thành công mật khẩu được sử dụng để tạo khóa riêng PuTTY, đó là “password”.



# CHƯƠNG 4: OTHER USEFUL COMMAND

## 1. Cú pháp dòng lệnh của John the Ripper

Khi được gọi mà không có bất kỳ tham số dòng lệnh nào, "john" sẽ hiển thị bản tóm tắt cách sử dụng.

Các tham số dòng lệnh được hỗ trợ bao gồm tên tệp mật khẩu và các tùy chọn. Nhiều tùy chọn hỗ trợ các tham số bổ sung.

Bạn có thể liệt kê bất kỳ số lượng tệp mật khẩu nào trực tiếp trên dòng lệnh của "john". Bạn không cần phải chỉ định bất kỳ tùy chọn nào. Nếu các tệp mật khẩu hợp lệ được chỉ định nhưng không có tùy chọn nào được đưa ra, John sẽ sử dụng các chế độ bẻ khóa mặc định với các thiết lập mặc định của chúng.

Các tùy chọn có thể được chỉ định cùng với các tệp mật khẩu hoặc tự chúng, mặc dù một số yêu cầu phải chỉ định các tệp mật khẩu và một số không hỗ trợ hoạt động trên các tệp mật khẩu.

Tất cả các tùy chọn đều phân biệt chữ hoa chữ thường, có thể viết tắt miễn là các viết tắt đó không bị nhầm lẫn, có thể được đặt trước bằng hai dấu gạch ngang (kiểu GNU) hoặc bằng một dấu gạch ngang, và có thể sử dụng "=" hoặc ":" để chỉ ra tham số (nếu được hỗ trợ cho tùy chọn đó).

Các tùy chọn được hỗ trợ như sau, các dấu ngoặc vuông biểu thị các tham số tùy chọn :

* --single - chế độ "bẻ khóa đơn"

Để kích hoạt chế độ "bẻ khóa đơn", sử dụng các quy tắc từ phần cấu hình [List.Rules:Single].

* --wordlist=FILE - chế độ từ điển, đọc các từ từ FILE,
* --stdin - hoặc từ stdin

Các tùy chọn này được sử dụng để kích hoạt chế độ từ điển :

* --rules - kích hoạt các quy tắc thay đổi từ cho chế độ từ điển
* Kích hoạt các quy tắc thay đổi từ được đọc từ [List.Rules:Wordlist].
* --incremental[=MODE] - chế độ "gia tăng" [sử dụng phần MODE]

Để kích hoạt chế độ incremental, sử dụng định nghĩa cấu hình được chỉ định (section [Incremental:MODE]). Nếu MODE bị bỏ qua, mặc định là "ASCII" cho hầu hết các loại hash và "LM\_ASCII" cho các hash LM.

* --external=MODE - chế độ ngoài hoặc bộ lọc từ

Để kích hoạt chế độ external, sử dụng các chức năng ngoài được định nghĩa trong phần [List.External:MODE].

* --stdout[=LENGTH] - chỉ xuất các mật khẩu thử nghiệm

Khi được sử dụng với một chế độ bẻ khóa, ngoại trừ "bẻ khóa đơn", làm cho John xuất các mật khẩu thử nghiệm mà nó tạo ra ra stdout thay vì thực sự thử chúng với các hash mật khẩu; không tệp mật khẩu nào được chỉ định khi sử dụng tùy chọn này. Nếu LENGTH được đưa ra, John cho rằng đó là độ dài mật khẩu quan trọng và chỉ tạo ra các mật khẩu lên đến độ dài đó.

* --restore[=NAME] - khôi phục một phiên làm việc bị gián đoạn

Tiếp tục một phiên làm việc bị gián đoạn, đọc thông tin trạng thái từ tệp phiên đã chỉ định hoặc từ $JOHN/john.rec theo mặc định.

* --session=NAME - đặt tên cho một phiên làm việc mới

Tùy chọn này chỉ có thể được sử dụng khi bắt đầu một phiên làm việc mới và mục đích của nó là đặt tên cho phiên làm việc mới (để John sẽ thêm hậu tố ".rec" để tạo tên tệp phiên). Điều này hữu ích cho việc chạy nhiều phiên bản John song song hoặc để có thể khôi phục một phiên làm việc khác với phiên làm việc cuối cùng bạn gián đoạn.

* --status[=NAME] - in trạng thái của một phiên làm việc [được gọi là NAME]

In trạng thái của một phiên làm việc bị gián đoạn hoặc đang chạy. Lưu ý rằng trên hệ thống Unix-like, bạn có thể cập nhật tệp phiên của một phiên làm việc chạy ngầm bằng cách gửi một tín hiệu SIGHUP đến quá trình "john" thích hợp; sau đó sử dụng tùy chọn này để đọc và hiển thị trạng thái.

* --make-charset=FILE - tạo một tập ký tự, ghi đè FILE

Tạo một tập ký tự dựa trên tần suất ký tự từ $JOHN/john.pot, để sử dụng với chế độ "incremental". Toàn bộ $JOHN/john.pot sẽ được sử dụng để tạo tập ký tự theo mặc định. Bạn có thể giới hạn tập mật khẩu được sử dụng bằng cách chỉ định một số tệp mật khẩu (trong trường hợp này chỉ các mật khẩu đã được bẻ khóa tương ứng với các tệp mật khẩu đó sẽ được sử dụng), "--format", hoặc/ và "--external" (với một chế độ ngoài định nghĩa một hàm filter()).

* --show - hiển thị mật khẩu đã bẻ khóa

Hiển thị các mật khẩu đã bẻ khóa cho các tệp mật khẩu đã cho (bạn phải chỉ định). Bạn có thể sử dụng tùy chọn này trong khi một phiên bản khác của John đang bẻ khóa để xem John đã làm gì cho đến nay; để có thông tin cập nhật nhất, trước tiên hãy gửi một tín hiệu SIGHUP đến quá trình "john" thích hợp.

* --test[=TIME] - chạy các bài kiểm tra và đánh giá trong TIME giây mỗi lần

Kiểm tra tất cả các thuật toán hash đã biên dịch để đảm bảo hoạt động đúng và đánh giá hiệu suất của chúng. Tùy chọn "--format" có thể được sử dụng để giới hạn việc này cho một thuật toán cụ thể.

* --users=[-]LOGIN|UID[,..] - [không] tải người dùng này (hoặc những người dùng này)

Cho phép bạn chọn chỉ một vài tài khoản để bẻ khóa hoặc cho các hoạt động khác. Một dấu gạch ngang trước danh sách có thể được sử dụng để đảo ngược kiểm tra (tức là, tải thông tin cho tất cả các tài khoản không được liệt kê).

* --groups=[-]GID[,..] - tải người dùng [không] của nhóm này (những nhóm này)

Yêu cầu John tải (hoặc không tải) thông tin cho các tài khoản trong nhóm được chỉ định.

* --shells=[-]SHELL[,..] - tải người dùng có [không] shell này (những shell này)

Tùy chọn này hữu ích để tải các tài khoản chỉ có shell hợp lệ hoặc không tải các tài khoản có shell không hợp lệ. Bạn có thể bỏ qua đường dẫn trước tên shell, vì vậy "--shells=csh" sẽ khớp với cả "/bin/csh" và "/usr/bin/csh", trong khi "--shells=/bin/csh" chỉ khớp với "/bin/csh".

* --salts=[-]N - tải các salts có [không] ít nhất N mật khẩu

Đây là một tính năng cho phép đạt được hiệu suất tốt hơn trong một số trường hợp đặc biệt. Ví dụ: bạn có thể bẻ khóa chỉ một số salts sử dụng "--salts=2" nhanh hơn và sau đó bẻ khóa phần còn lại sử dụng "--salts=-2". Tổng thời gian bẻ khóa sẽ tương đương, nhưng bạn có thể nhận được một số mật khẩu bị bẻ khóa sớm hơn.

* --save-memory=LEVEL - bật tiết kiệm bộ nhớ, ở cấp độ 1..3

Bạn có thể cần tùy chọn này nếu bạn không có đủ bộ nhớ hoặc không muốn John ảnh hưởng quá nhiều đến các quá trình khác hoặc không cần John tải và in tên đăng nhập cùng với các mật khẩu đã bị bẻ khóa. Mức 1 yêu cầu John không lãng phí bộ nhớ cho các tên đăng nhập; nó chỉ được hỗ trợ khi một chế độ bẻ khóa khác ngoài "bẻ khóa đơn" được yêu cầu rõ ràng. Nó không có ảnh hưởng tiêu cực đến hiệu suất - thực tế, đôi khi nó làm tăng tốc độ. Lưu ý rằng nếu không có tùy chọn --save-memory=1 (hoặc cao hơn), John sẽ lãng phí một số bộ nhớ cho các tên đăng nhập tiềm năng ngay cả khi các tệp hash mật khẩu không có tên đăng nhập. (Việc thiếu tên đăng nhập hoàn toàn không được biết đến khi John bắt đầu phân tích các tệp, vì vậy nó phải ghi lại thực tế rằng mỗi mục nhập riêng lẻ không có tên đăng nhập trừ khi bạn chỉ định tùy chọn này.) Các mức 2 và 3 giảm sử dụng các tối ưu hóa hiệu suất liên quan đến các bảng tra cứu lớn, do đó có tác động tiêu cực đến hiệu suất. Bạn có thể tránh sử dụng chúng trừ khi John không hoạt động hoặc bị chuyển sang swap.

* --node=MIN[-MAX]/TOTAL - phạm vi số của node này trong tổng số node

Tùy chọn này được thiết kế để cho phép xử lý song song và phân tán theo cách thủ công. Ví dụ, để chia khối lượng công việc trên hai node (có thể là các máy tính, lõi CPU, v.v.), bạn có thể chỉ định --node=1/2 trên một phiên chạy của John và --node=2/2 trên phiên chạy khác. (Nếu bạn thực hiện việc này trên cùng một máy và với cùng một bản dựng của John, bạn cũng cần chỉ định các tên phiên khác nhau với tùy chọn --session cho hai phiên chạy đồng thời.) Các node được giả định là có tốc độ tương đương (nếu không phải như vậy, một node sẽ hoàn thành trước node kia, trừ khi bạn đang sử dụng chế độ bẻ khóa và cài đặt mà phiên không được mong đợi hoàn thành). Nếu các node của bạn có tốc độ rất khác nhau, bạn có thể điều chỉnh bằng cách phân bổ phạm vi số node cho từng phiên chạy. Ví dụ, nếu bạn sử dụng các bản dựng John có hỗ trợ OpenMP trên hai máy, OpenMP được hỗ trợ (với khả năng mở rộng tốt) cho loại hàm băm mà bạn đang bẻ khóa, và một trong các máy có gấp đôi số lõi CPU có tốc độ tương đương so với máy kia, bạn có thể sử dụng --node=1-2/3 trên máy lớn hơn (cho phép nó là các node 1 và 2 trong tổng số 3 node) và --node=3/3 trên máy nhỏ hơn.

Hiệu quả của phương pháp này đối với xử lý song song, như được triển khai hiện nay, thay đổi tùy theo chế độ bẻ khóa và cài đặt của nó (hiệu quả cao hơn cho chế độ bẻ khóa incremental và chế độ danh sách từ với nhiều quy tắc, và thấp hơn cho các chế độ bẻ khóa khác và chế độ danh sách từ không có quy tắc hoặc có ít quy tắc), loại hàm băm (hiệu quả cao hơn cho các hàm băm tính toán chậm), số lượng salt (hiệu quả cao hơn cho số lượng salt lớn hơn), và số lượng node (hiệu quả cao hơn cho số lượng node thấp hơn). Khả năng mở rộng có thể bị hạn chế. Số lượng node cao nhất mà bạn có thể sử dụng hợp lý thay đổi tùy theo chế độ bẻ khóa, cài đặt của nó, loại hàm băm, và số lượng salt. Với chế độ bẻ khóa incremental, hiệu quả về tốc độ c/s gần như hoàn hảo (không có chi phí phụ), nhưng một số node có thể nhận được quá ít công việc - và vấn đề này trở nên nghiêm trọng hơn với số lượng node cao (chẳng hạn 100 hoặc hơn) và/hoặc các cài đặt hạn chế (chẳng hạn như MinLen và MaxLen được đặt cùng một giá trị hoặc trong phạm vi hẹp, và/hoặc tệp charset chỉ sử dụng ít ký tự). Với chế độ danh sách từ, để đạt hiệu quả cao, số lượng quy tắc (sau khi mở rộng tiền xử lý) cần phải cao gấp nhiều lần số lượng node, trừ khi tốc độ p/s thấp (do loại hàm băm chậm và/hoặc số lượng salt cao).

Do không có sự giao tiếp giữa các node, các hàm băm đã bị bẻ khóa bởi một node sẽ tiếp tục được bẻ khóa bởi các node khác. Điều này chủ yếu là ổn đối với các loại hàm băm không có salt hoặc khi chỉ có một salt (vì cùng một số lần tính toán hàm băm sẽ được thực hiện - cụ thể là chỉ một lần cho mỗi mật khẩu thử nghiệm), nhưng đây là một nhược điểm nghiêm trọng khi có nhiều salt khác nhau và số lượng của chúng có thể giảm dần khi một số hàm băm được bẻ khóa.

* --fork=N - fork N tiến trình

Tùy chọn này chỉ có sẵn trên các hệ thống giống Unix. Đây là một cách dễ dàng để sử dụng nhiều CPU hoặc lõi CPU - bạn chỉ cần chỉ định số lượng tiến trình John mà bạn muốn chạy. Bạn có thể sử dụng --fork như một lựa chọn thay thế cho OpenMP, cho các định dạng hiện tại không hỗ trợ OpenMP, hoặc cùng với OpenMP (ví dụ, trên một máy có 64 CPU logic, bạn có thể chọn chạy với --fork=8 cho 8 tiến trình và sử dụng OpenMP để chạy 8 luồng cho mỗi tiến trình).

Bạn có thể sử dụng --fork cùng với --node để sử dụng nhiều máy đồng thời chạy nhiều tiến trình John trên mỗi máy. Ví dụ, để sử dụng hai máy 8 lõi tương tự nhau, bạn có thể chạy --fork=8 --node=1-8/16 trên một máy và --fork=8 --node=9-16/16 trên máy kia. Đối với một ví dụ phức tạp hơn, nếu bạn có một máy 8 lõi và một máy 64 lõi với hiệu suất tương tự trên mỗi lõi, bạn có thể chạy một bản dựng hỗ trợ OpenMP trên cả hai máy và sử dụng --node=1/9 (không có --fork) trên máy đầu tiên (8 luồng trong 1 tiến trình) và --fork=8 --node=2-9/9 trên máy 64 lõi (8 luồng trong 8 tiến trình, tổng cộng 64 luồng trên máy này). Với triển khai hiện tại, phạm vi số node được gán cho mỗi phiên John phải khớp với số lượng tiến trình --fork.

Khi chạy với --fork, nhiều tệp .rec được tạo ra, sau đó được đọc lại bằng --status và --restore nếu bạn sử dụng các tùy chọn này. Giống như với các tùy chọn khác, bạn không được chỉ định --fork cùng với --status hoặc --restore, vì chúng đọc tệp .rec chính (không được đánh số) đầu tiên, chứa tùy chọn --fork đúng trong đó, dẫn đến việc đọc thêm các tệp .rec được đánh số khi cần.

Phía dưới, --fork sử dụng cùng chức năng với --node, vì vậy cùng những hạn chế về hiệu quả và khả năng mở rộng áp dụng. Mặc dù vậy, --fork thường hiệu quả hơn nhiều so với OpenMP - đặc biệt đối với các loại hàm băm tính toán nhanh (chẳng hạn như hàm băm LM), nơi mà chi phí phụ của OpenMP thường không chấp nhận được.

Tương tự như --node, gần như không có sự giao tiếp giữa các tiến trình với --fork. Các hàm băm đã bị bẻ khóa bởi một tiến trình sẽ tiếp tục được bẻ khóa bởi các tiến trình khác. Giống như với --node, điều này chủ yếu là ổn đối với các loại hàm băm không có salt hoặc khi chỉ có một salt, nhưng đây là một nhược điểm nghiêm trọng khi có nhiều salt khác nhau và số lượng của chúng có thể giảm dần khi một số hàm băm được bẻ khóa. Để loại bỏ các hàm băm (và có thể là salt) đã bị bẻ khóa khỏi tất cả các tiến trình, bạn có thể tạm dừng và khôi phục phiên một lúc.

* --format=NAME - ép buộc loại hàm băm NAME

Cho phép bạn ghi đè việc phát hiện loại hàm băm. Từ phiên bản John the Ripper 1.8.0, các "tên định dạng" hợp lệ bao gồm descrypt, bsdicrypt, md5crypt, bcrypt, LM, AFS, tripcode, dummy, và crypt (và nhiều hơn nữa trong phiên bản jumbo). Bạn có thể sử dụng tùy chọn này khi bắt đầu một phiên bẻ khóa hoặc cùng với một trong các tùy chọn: --test, --show, --make-charset. Lưu ý rằng John không thể bẻ khóa các hàm băm khác loại cùng một lúc. Nếu bạn có một tệp mật khẩu sử dụng nhiều loại hàm băm, thì bạn phải chạy John một lần cho mỗi loại hàm băm và cần sử dụng tùy chọn này để làm cho John bẻ khóa các hàm băm khác với loại mà nó sẽ tự động phát hiện theo mặc định.

* --format=crypt

Tùy chọn này có thể hoặc không được hỗ trợ trong một bản build của John. Trong các bản build mặc định của John, hỗ trợ này hiện chỉ được bao gồm trên Linux và Solaris. Khi được chỉ định (và được hỗ trợ), tùy chọn này khiến John sử dụng hàm crypt(3) hoặc crypt\_r(3) của hệ thống. Điều này có thể cần thiết để kiểm tra các băm mật khẩu được hệ thống hỗ trợ, nhưng chưa được các hàm mật mã tối ưu của John hỗ trợ. Hiện tại, đây là trường hợp của các băm SHA-crypt từ glibc 2.7+ như được sử dụng trong các phiên bản gần đây của Fedora và Ubuntu, và các băm SunMD5 được hỗ trợ (nhưng không được sử dụng mặc định) trên các phiên bản gần đây của Solaris. Thực tế, bạn không cần phải chỉ định rõ "--format=crypt" cho các băm thuộc các loại cụ thể này trừ khi bạn có các loại băm khác (được John hỗ trợ natively) trong file mật khẩu, trong trường hợp này một loại băm khác có thể được phát hiện trừ khi bạn chỉ định tùy chọn này.

"--format=crypt" cũng là cách để John crack các băm crypt(3) thuộc các loại khác nhau cùng một lúc, nhưng làm như vậy sẽ dẫn đến hiệu suất kém và kết quả kém không cần thiết (về số mật khẩu được crack) đối với các loại băm "nhanh hơn" (so với các loại "chậm hơn" được tải để crack cùng một lúc). Vì vậy, bạn được khuyên nên sử dụng các lần triệu gọi riêng biệt của John, mỗi lần cho một loại băm.

## 2. Các tiện ích bổ sung

Có một số tiện ích liên quan trong thư mục chạy của John. (Tùy thuộc vào nền tảng, những tiện ích này có thể là các liên kết biểu tượng đến tệp nhị phân chương trình chính của John.) :

***unshadow PASSWORD-FILE SHADOW-FILE***

Kết hợp các tệp "passwd" và "shadow" (khi bạn đã có quyền truy cập vào cả hai) để sử dụng với John. Bạn có thể cần điều này vì nếu bạn chỉ sử dụng tệp shadow, thông tin "Tên đầy đủ" hoặc "GECOS" sẽ không được sử dụng bởi "chế độ bẻ khóa đơn" (do đó giảm hiệu quả của nó) và bạn sẽ không thể sử dụng các tùy chọn "--groups" và "--shells" và để chọn theo UID với "--users". Bạn có thể cũng muốn xem tất cả các trường của tệp passwd với "--show". Bạn thường muốn chuyển hướng đầu ra của "unshadow" đến một tệp mà bạn sẽ chuyển cho John:

***unafs DATABASE-FILE CELL-NAME***

Lấy các băm mật khẩu ra khỏi cơ sở dữ liệu nhị phân AFS và tạo đầu ra có thể sử dụng được bởi John (bạn nên chuyển hướng đầu ra vào một tệp).

***unique OUTPUT-FILE***

Loại bỏ các mục trùng lặp khỏi danh sách từ (đọc từ stdin) mà không thay đổi thứ tự của các mục. Bạn có thể muốn sử dụng điều này với tùy chọn "--stdout" của John nếu bạn có nhiều không gian đĩa để đánh đổi lấy thời gian crack giảm (có thể thử một số trùng lặp như chúng có thể được tạo ra với các quy tắc xử lý từ).

## 3. Các tập lệnh

Cũng có một số tập lệnh liên quan được cung cấp trong thư mục chạy của John. (Các gói nhị phân của John có thể chọn cài đặt chúng cùng với tài liệu thay thế.)

***relbench BENCHMARK-FILE-1 BENCHMARK-FILE-2***

relbench là một tập lệnh Perl để so sánh hai lần chạy benchmark "john --test", chẳng hạn như cho các máy khác nhau, mục tiêu "make", trình biên dịch C, tùy chọn tối ưu hóa, hoặc/và các phiên bản của John the Ripper. Để sử dụng nó, hãy chuyển hướng đầu ra của mỗi lần chạy "john --test" vào một tệp, sau đó chạy tập lệnh trên hai tệp đó. Hầu hết các giá trị đầu ra bởi tập lệnh chỉ ra hiệu suất tương đối được thấy trên lần chạy benchmark thứ hai so với lần đầu tiên, với giá trị 1.0 chỉ ra không thay đổi, các giá trị cao hơn 1.0 chỉ ra tăng tốc, và các giá trị thấp hơn 1.0 chỉ ra giảm tốc. Cụ thể, tập lệnh đầu ra giá trị tối thiểu, tối đa, trung bình, và trung bình hình học cho sự tăng tốc (hoặc giảm tốc) được thấy trên nhiều benchmark cá nhân mà "john --test" thực hiện. Nó cũng đầu ra độ lệch tuyệt đối trung bình (so với trung bình) và độ lệch chuẩn hình học (so với trung bình hình học). Trong số hai giá trị này, độ lệch tuyệt đối trung bình là 0.0 sẽ chỉ ra rằng không có độ lệch so với trung bình, trong khi độ lệch chuẩn hình học là 1.0 sẽ chỉ ra rằng tất cả các benchmark đều được tăng tốc hoặc giảm tốc theo tỷ lệ chính xác hoặc tốc độ của chúng không thay đổi. Trong thực tế, các giá trị này sẽ có xu hướng lệch khỏi 0.0 và 1.0, tương ứng.

***mailer PASSWORD-FILE***

Một tập lệnh shell để gửi thư đến tất cả người dùng có mật khẩu bị crack. Điều này không phải lúc nào cũng là một ý tưởng tốt, vì nhiều người không kiểm tra email của họ hoặc bỏ qua những tin nhắn như vậy, và các tin nhắn có thể là gợi ý cho kẻ crack.

Bạn nên triển khai kiểm tra độ mạnh mật khẩu chủ động, chẳng hạn như với passwdqc, trước khi yêu cầu người dùng thay đổi mật khẩu của họ - cho dù sử dụng tập lệnh này hay cách khác. Và bạn nên chỉnh sửa tin nhắn bên trong tập lệnh trước khi có thể sử dụng nó.

***makechr***

Đây là một tập lệnh để (tái) tạo các tệp .chr bằng cách sử dụng mật khẩu trong john.pot và áp dụng tất cả các bộ lọc chế độ ngoài đã được định nghĩa. Để tăng tốc trên một máy tính đủ lớn (RAM và CPU), "&" có thể được thêm sau lần triệu gọi John bên trong tập lệnh (để chạy tất cả chúng song song).

# KẾT LUẬN

Trong quá trình nghiên cứu về John The Ripper, chúng em đã khám phá một thế giới đầy thách thức và cơ hội. Việc tìm hiểu về cách hoạt động của công cụ này và cách mà chúng có thể được sử dụng để tiến hành các cuộc tấn công đã là một hành trình đầy thú vị.

Chúng em nhận thấy rằng John The Ripper không chỉ đơn thuần là một công cụ công cụ mã nguồn mở được sử dụng phổ biến để kiểm tra độ mạnh của mật khẩu và bẻ khóa mật khẩu mà còn là một mối đe dọa lớn đối với bảo mật hệ thống. Việc sử dụng mở ra những khả năng mới cho việc đánh cắp thông tin nhạy cảm và đe dọa tính riêng tư của người dùng.

Mặc dù chúng em đã đạt được một số tiến bộ trong việc hiểu biết và phòng tránh, nhưng còn nhiều thách thức phức tạp phía trước. Trong đó, việc nắm bắt và ngăn chặn các cuộc tấn công vẫn là một vấn đề đầy thách thức và đòi hỏi sự chú ý đặc biệt.

Chúng em nhận thấy rằng việc tiếp tục nghiên cứu và phát triển các biện pháp phòng tránh tấn công bằng John The Ripper là rất cần thiết để bảo vệ hệ thống và dữ liệu của người dùng. Hy vọng rằng những công trình nghiên cứu tiếp theo sẽ tiếp tục đóng góp vào việc nâng cao sự hiểu biết và bảo mật trong lĩnh vực này.

# TÀI LIỆU THAM KHẢO

1. <https://www.hackingarticles.in/beginner-guide-john-the-ripper-part-1/?fbclid=IwZXh0bgNhZW0CMTAAAR0OYggBDOexTmKwxPEAcqb_-FavebUyz9kmAiCGhDeIzPLVOGINug8M7Xw_aem_AbKHxpG_eD1efvECCc76PJ9V4l2fE8uJx3WDi5r8y7DIsWtvqsvIFI9iImZEoHyd0603TANjQiepMc9FkPQheTJR>
2. <https://www.hackingarticles.in/beginners-guide-for-john-the-ripper-part-2/?fbclid=IwZXh0bgNhZW0CMTAAAR0vnnc-ipzjEe01CLrFE-mbrJrscTMbAuxrYbZZBS9O4YVuaYuSYK2wuKE_aem_AbLW9jDNFyuKn4dNACfLa_6qrXh0T9_OXecDV_5d955XOpIvYMZHEaJyZK8d-3L9FFe1dX705ycdUBQG_pzIJiTw>
3. <https://www.varonis.com/blog/john-the-ripper?fbclid=IwZXh0bgNhZW0CMTAAAR2BrZqEPod9onE3TrGGRJSZgkc_5bGE72hYsdi47pmVK1erKryZzSPDGMY_aem_AbIcGmaQ6koUm5FTWb5mluyKn8ATGYQpCLCEh0zrO9TleN9PhR3T8V4tSE53WT8JyfyooLVtY59cvF4tx5cQX207>

# LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến quý thầy cô và các bạn đã dành thời gian quý báu của mình để lắng nghe và theo dõi bài báo cáo của chúng em. Sự quan tâm và sự ủng hộ mạnh mẽ từ phía thầy cô không chỉ là một niềm động viên lớn lao mà còn là nguồn động viên không ngừng để chúng em tiếp tục nỗ lực và phát triển.

Chúng em không thể không bày tỏ lòng biết ơn sâu sắc về sự chia sẻ của thầy cô trong việc đóng góp ý kiến và phản hồi xây dựng. Những ý kiến này không chỉ giúp chúng em hiểu rõ hơn về chủ đề mà còn là nguồn động viên lớn lao để chúng em tiếp tục phát triển và cải thiện kỹ năng của mình. Sự trân trọng và tôn trọng chúng em dành cho thầy cô không thể diễn tả hết bằng lời.

Chúng em hiểu rằng không có sự thành công nào là độc lập và chúng em muốn bày tỏ lòng biết ơn đặc biệt đến sự hỗ trợ của quý thầy cô và các bạn trong suốt quá trình làm báo cáo này. Những góp ý, sự chỉ dẫn và sự khích lệ từ phía thầy cô đã chính là động lực mạnh mẽ giúp chúng em vượt qua những thách thức và đạt được kết quả tốt nhất.

Cuối cùng, chúng em xin kính chúc quý thầy cô và các bạn sức khỏe dồi dào, thành công trong sự nghiệp giảng dạy và học tập, và hy vọng được tiếp tục hợp tác và nhận được sự giúp đỡ từ thầy cô trong những dự án sắp tới.

Xin chân thành cảm ơn thầy cô và các bạn!

### 