

THM-John The Ripper(hash破解工具)-学习

本文相关的TryHackMe实验房间链接：<https://tryhackme.com/room/johntheripper0>

H2 基本概念

开膛手约翰是世界上最著名的、广受欢迎的多功能hash破解工具之一，它有非常快速的破解速度，也兼容极大范围的哈希类型。

哈希是一种获取任意长度的数据并以固定长度的另一种形式表示它的方法。这个计算过程将屏蔽数据的原始值，是通过哈希算法运行原始数据来完成的。有许多流行的哈希（散列）算法，如 MD4、MD5、SHA1 和 NTLM等。

哈希算法本身是不可逆的，但是这并不意味着破解哈希值是不可能的。如果你现在知道一个密码的哈希值（你也知道用了什么类型的哈希算法），那么就可以使用该哈希算法对大量数据进行哈希计算处理，然后你可以将这些通过计算得到的哈希值与你试图破解的密码的哈希值进行比较，以查看其中是否有相互匹配的，如果有的话，你现在就知道哪个数据对应密码的哈希值了——你已经破解这个密码了！

这个过程被称为“字典式攻击”，“开膛手约翰”，或者通常简称为“约翰”，则是一个允许你对大量不同哈希类型进行快速暴力破解的工具。

H2 John工具的下载与安装

John the Ripper 在许多不同的操作系统上都得到了支持。John 有多个版本：标准的“核心”发行版，以及多个社区版本——它们扩展了原始 John 发行版的特性。这些发行版中最流行的是“Jumbo John”——这也是Kali系统预安装的John工具。

如果你正在使用kali系统，你可以使用以下命令安装 Jumbo John：



```
sudo apt install john
```

如果你正在使用 Blackarch系统，或者 Blackarch 存储库，你可能已经安装了 Jumbo John，也可能没有安装，请用以下命令来检测John或者进行John的安装：



```
pacman -Qe | grep "john"      #检测是否安装了Jumbo John
```

```
pacman -S john                #安装Jumbo John工具
```

关于John安装过程和如何从源代码构建软件包配置John，请参考以下网页：<https://github.com/openwall/john/blob/bleeding-jumbo/doc/INSTALL>

如果你在 Windows 上安装 Jumbo John，你只需要下载并安装压缩的二进制文件，有64位系统和32位系统的压缩包。

答题：

Answer the questions below 回答下面的问题

What is the most popular extended version of John the Ripper? 开膛手约翰最受欢迎的扩展版本是什么？

Correct Answer

H2 字典

为了对哈希值进行字典式攻击，你还需要一个好用的字典。在github存储库里有一个很好的字典集合可以使用：<https://github.com/danielmiessler/SecLists>

在 Parrot、Kali 等渗透系统中，你也可以在/usr/share/wordlists 目录下找到一些默认的字典。

还有一个比较强大的字典是rockyou.txt，你可以在上面提到的 [SecLists](https://github.com/danielmiessler/SecLists) 存储库里面的 /Passwords/Leaked-Databases 下面找到rockyou.txt，

下载后，使用命令tar xvzf rockyou.txt.tar.gz 进行解压即可。

答题：

Answer the questions below 回答下面的问题

What website was the rockyou.txt wordlist created from a breach on? rockyou.txt 字典是因为什么网站的泄露事件得以创建？

Correct Answer

H2 破解简单的哈希值

H3 理论

John的基本语法



```
john [options] [path to file]
```

john 调用开膛手约翰程序

[path to file] 这个路径包含了你要破解的hash文件，如果它在当前工作目录中，则不需要输入路径，只需要输入hash文件名称即可

自动破解

John 有一些内置的功能，可以检测给出的hash类型，并为你选择合适的规则和格式来破解它，这并不总是最好的主意，但是如果你不知道要破解的hash值是什么hash类型，你只是想尝试破解它，那么这个自动破解选项可以是一个很好的选择！为此，我们使用以下语法：



```
john --wordlist=[path to wordlist] [path to file]
```

--wordlist= 指定使用字典模式，从路径包含的字典文件中读取内容
[path to wordlist] 指定你将要使用的字典的路径

示例用法：



```
john --wordlist=/usr/share/wordlists/rockyou.txt hash_to_crack.txt
```

识别哈希

有时候 John 无法很好地自动识别和加载哈希，此时我们可以使用其他工具来识别哈希类型，然后再将 john 设置为使用特定的哈希格式进行破解。

有多种方法可以做到对哈希类型进行识别，例如使用在线哈希识别器：https://hashes.com/en/tools/hash_identifier

也可以使用hash-identifie: <https://gitlab.com/kalilinux/packages/hash-identifier/-/tree/kali/master>

hash-identifie是一个非常容易使用的 Python 工具,当你输入一个哈希值时,它会告诉你这个哈希值最有可能是什么哈希类型,将一系列哈希类型按可能性进行降序排列。

你可以使用以下方法从 gitlab 中提取hash-identifie的python 文件进行工具安装:



```
wget https://gitlab.com/kalilinux/packages/hash-identifier/-/raw/kali/master/hash-id.py
```

然后使用命令 `python3 hash-id.py` 简单地启动它,输入你想要识别的hash值,它就会给出这个hash值可能对应的hash类型!

指定类型的破解

一旦你确定你要破解的hash值的类型,你就可以在John中指定类型进行破解:



```
john --format=[format] --wordlist=[path to wordlist] [path to file]
```

`--format=[format]` 这个标志告诉 **John** 你给它一个特定类型的哈希,并使用对应的类型来破解它
哈希的格式

示例用法:



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

注意

当你告诉 john 使用指定类型进行破解时,如果处理的是标准哈希类型,例如上面的例子中的 md5,你必须以 raw- 作为前缀,告诉 john 你只是在处理一个标准的 hash 类型(尽管这并不总是适用)。要检查是否需要添加前缀,可以使用命令验证: `john --list=formats | grep -iF "md5"`

H3 答题

下载包含hash文件的附件，回答问题。

安装hash-identifie工具，输入命令python3 hash-id.py执行，将需要进行识别的hash值复制粘贴输入，再按回车即可开始识别

hash-id.py 和在线网站 https://hashes.com/en/tools/hash_identifier 两种方式都能识别hash值的类型，结合两种方式的识别结果：得出要破解的hash值的类型。

知道哈希类型之后，就可以运行john并使用参数指定hash类型进行hash破解，注意：表示hash类型的参数写法要符合john的语法

如果john破解失败，则通过以下在线网站破解哈希：

<https://hashes.com/en/decrypt/hash>

<https://crackstation.net/>

操作截图：

```
Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

```
Least Possible Hashes:
[+] RAdmin v2.x
[+] NTLM
[+] MD4
[+] MD2
[+] MD5(HMAC)
[+] MD4(HMAC)
[+] MD2(HMAC)
[+] MD5(HMAC Wordpress))
[+] Haval-128
[+] Haval-128(HMAC)
[+] RipeMD-128
[+] RipeMD-128(HMAC)
[+] SNEFRU-128
[+] SNEFRU-128(HMAC)
[+] Tiger-128
[+] Tiger-128(HMAC)
[+] md5($pass.$salt)
[+] md5($salt.$pass)
[+] md5($salt.$pass.$salt)
[+] md5($salt.$pass.$username)
[+] md5($salt.md5($pass))
[+] md5($salt.md5($pass))
[+] md5($salt.md5($pass.$salt))
[+] md5($salt.md5($pass.$salt))
[+] md5($salt.md5($salt.$pass))
[+] md5($salt.md5(md5($pass).$salt))
```

```
(root@hekeats)~[/home/hekeats/桌面]
# john --format=raw-sha1 --wordlist=/usr/share/wordlists/rockyou.txt ./first_task_hashes/hash2.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 256/256 AVX2 8x])
Warning: no OpenMP support for this hash type, consider --fork=8
Press 'q' or Ctrl-C to abort, almost any other key for status
kangeroo (?)
1g 0:00:00.00 DONE (2022-09-29 00:53) 50.00g/s 5857Kp/s 5857Kc/s 5857KC/s karate2..kalvin1
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed.
```

```
(root@hekeats)~[/home/hekeats/桌面]
# john --format=raw-sha256 --wordlist=/usr/share/wordlists/rockyou.txt ./first_task_hashes/hash3.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 256/256 AVX2 8x])
Warning: poor OpenMP scalability for this hash type, consider --fork=8
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
microphone (?)
1g 0:00:00.00 DONE (2022-09-29 01:05) 100.00g/s 13107Kp/s 13107Kc/s 13107KC/s 123456..kovacs
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.
```

```
(root@hekeats)~[/home/hekeats/桌面]
# john --format=whirlpool --wordlist=/usr/share/wordlists/rockyou.txt ./first_task_hashes/hash4.txt
Using default input encoding: UTF-8
Loaded 1 password hash (whirlpool [WHIRLPOOL 32/64])
Warning: poor OpenMP scalability for this hash type, consider --fork=8
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
colossal (?)
1g 0:00:00.00 DONE (2022-09-29 01:08) 7.692g/s 5293Kp/s 5293Kc/s 5293KC/s davita1..blah2007
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Answer the questions below 回答下面的问题

What type of hash is hash1.txt?

Hash1.txt 是什么类型的散列?

MD5

Correct Answer

What is the cracked value of hash1.txt?

Hash1.txt 的破解值是多少?

biscuit

Correct Answer

What type of hash is hash2.txt?

Hash2.txt 是什么类型的散列?

SHA1

Correct Answer

What is the cracked value of hash2.txt?

Hash2.txt 的破解值是多少?

kangeroo

Correct Answer

What type of hash is hash3.txt?

Hash3.txt 是什么类型的散列?

SHA256

Correct Answer

What is the cracked value of hash3.txt?

Hash3.txt 的破解值是多少?

microphone

Correct Answer

What type of hash is hash4.txt?

Hash4.txt 是什么类型的散列?

Whirlpool

Correct Answer

What is the cracked value of hash4.txt?

Hash4.txt 的破解值是多少?

colossal

Correct Answer

💡 Hint 提示

H2 破解 Windows 身份验证哈希

H3 理论

破解 Windows 哈希值

现在我们已经了解了开膛手约翰的基本语法和用法——让我们继续来解决一些稍微困难一点的问题。

Windows身份验证哈希是由操作系统存储的密码的哈希版本，有时可以使用暴力方法去破解它们，要获得这些哈希值，你必须已经是一个特权用户。

NTHash / NTLM

NThash 是现代Windows操作系统计算机存储用户和服务密码的哈希格式，它也通常被称为"NTLM"，它引用了以前的 Windows 版本用于哈希密码的格式("LM")，所以NThash也被称为" NT/LM "。

Windows 产品的 NT 命名方式，最初意味着" 新技术 (New Technology) "，从 Windows NT版本开始被使用，表示并非由MS-DOS 操作系统所构建的产品。

后面随着时间发展，最终，"NT" 成为了微软发布的标准操作系统类型，"NT" 这个名字就被删除了，但它仍然以微软的一些技术的名称存在于Windows系统中。

你可以通过在 Windows 机器上转储(dumping) SAM 数据库、通过使用像 Mimikatz 这样的工具或者从 ActiveDirectory的数据库NTDS.dit中获取NTHash/NTLM哈希值。

你可能不需要继续进行权限提升来破解哈希-因为你可以进行“传递哈希”攻击，但如果目标有一个薄弱的密码策略，直接进行哈希破解也是一个可行的选择。

H3 答题

下载附件，使用以下命令破解哈希值：

```
john --format=nt --wordlist=/usr/share/wordlists/rockyou.txt ntlm.txt
```

```
(root@helixcat) [/home/helixcat/桌面]
john --format=nt --wordlist=/usr/share/wordlists/rockyou.txt ntlm.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=8
Press 'q' or Ctrl-C to abort, almost any other key for status
mushroom (??)
1g 0:00:00.00 DONE (2022-09-29 01:50) 100.0g/s 307200p/s 307200c/s 307200C/s skater1..dangerous
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

结果是：mushroom

Answer the questions below 回答下面的问题

What do we need to set the "format" flag to, in order to crack this? 为了破解这个问题，我们需要将John中的指定“格式”标志设置为什么？

nt

Correct Answer 正确答案

What is the cracked value of this password? 这个密码的破解值是多少？

mushroom

Correct Answer 正确答案

H2 破解linux上的/etc/shadow 哈希

H3 理论

从/etc/shadow 中破解散列

etc/shadow 文件是 Linux 机器上存储密码哈希的文件，它还会存储其他信息，如上次密码更改日期和密码过期信息，shadow文件的每一行都为系统的每个用户或用户帐户包含一个条目。

这个文件通常只有 root 用户才能访问——因此，为了获得文件的哈希值，你必须拥有足够的特权，你才有机会破解一些哈希值。

Unshadowing (去除shadow)

John 对于需要使用的数据格式有一定要求，所以为了破解/etc/shadow密码你必须将/etc/shadow与/etc/passwd 文件组合起来，以便 John 能够理解所给出的数据。为此，我们需要使用一个内置在 John 工具套件中的名为 unshadow 的工具。unshadow 的基本语法如下：



```
unshadow [path to passwd] [path to shadow]
```

unshadow 调用 unshadow 工具

[path to passwd] 包含从目标计算机获取的/etc/passwd 文件副本的文件路径

[path to shadow] 包含从目标计算机获取的/etc/shadow 文件副本的文件路径

示例用法：



```
unshadow local_passwd local_shadow > unshadowed.txt
```

注意：

在使用 unshadow 时，你可以使用完整的/etc/passwd 和/etc/shadow 文件——如果它们可用的话，也可以使用每个文件中的相关行，例如：

FILE 1 - local_passwd

包含 root 用户的/etc/passwd 行:

root:x:0:0::/root:/bin/bash

FILE 2 - local_shadow

包含 root 用户的/etc/shadow 行:

root:\$6\$2nwjN454g.dv4HN/\$m9Z/r2xVfweYVkr.r.v5Ft8Ws3/YYksfNwq96UL1FX00JjY1L6L.DS3KEVsZ9
rOVLB/ldTeEL/0IhJZ4GMFMGA0:18576:::~:

在完成unshadow处理之后，我们将 unshadow 的输出直接提供给 John，在我们的示例中 这个输出结果被称为 “unshadow.txt”。

我们不需要在这里指定哈希格式，因为我们已经专门为 John 做了输入处理（使用unshadow），但是在某些情况下，你仍然需要指定哈希格式:

--format=sha512crypt

john --format=sha512crypt --wordlist=/usr/share/wordlists/rockyou.txt unshadowed.txt

H3 答题

下载附件（此附件可不经unshadow处理），先识别hash类型，再使用以下命令破解哈希:

john --wordlist=/usr/share/wordlists/rockyou.txt --format=sha512crypt etchashes.txt

```
(root@hekeats) ~ [ /home/hekeats/桌面 ]
# john --wordlist=/usr/share/wordlists/rockyou.txt --format=sha512crypt etchashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
1234 (root)
1g 0:00:00:00 DONE (2022-09-29 01:59) 4.34/g/s 8904p/s 8904c/s 8904C/s kucing..lovers1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

结果是: 1234 root

What is the root password? 根用户密码是什么?

1234

Correct Answer 正确答案

H2 single破解模式

H3 理论

single模式简介

john还有另一种模式，叫做single破解模式。在这种模式下，john 只使用用户名中提供的信息，通过稍微更改用户名中包含的字母和数字（字词混淆），试探性地计算出可能的密码值。

字词混淆

要展示什么是single破解模式，什么是字词混淆，最好的方法是通过一个实际的例子：



```
如果用户名是 Markus
那么一些可能的密码可以是：
Markus1, Markus2, Markus3
MArkus, MARkus, MARKus
Markus!, Markus$, Markus*
.....
```

这种技巧被称为字词混淆，在这个过程中，John 正在建立一个自己的字典，这个字典是基于它接收到的信息所建立的，并且使用了一系列“混淆规则”，这些规则定义了它如何根据你想要破解的目标的相关因素来变异它开始的单词，进而生成一个字典。这是对弱密码生成策略的一种利用，因为很多弱密码都是基于用户名信息，或者用户所登录的服务而产生。

GECOS

John 实现的字词混淆处理还具有与 UNIX 操作系统（以及其他类 UNIX 操作系统，如 Linux）的 Gecos 字段兼容的特性。

那么，什么是 Gecos？还记得在上文中我们提到的/etc/shadow 和/etc/passwd 的数据条目吗？

如果你仔细看，你会发现shadow文件和passwd文件中的每个字段都用冒号":" 分隔，这些记录被分割成的每一个字段都称为 Gecos 字段。

John 可以将存储在这些记录中的信息，比如全名和主目录名等信息，添加到它用single模式破解/etc/shadow哈希时生成的字符串中。

使用single模式

如果我们要破解用户名为Mike的密码，我们使用的语法如下：



```
john --single --format=[format] [path to file]
--single    这个标志让 john 知道你想使用单破解模式。
```

示例用法:



```
john --single --format=raw-sha256 hashes.txt
```

注意：如果你想在单破解模式下破解哈希，那么你需要更改提供给 john 的文件内容，以便john理解 根据哪些数据开始创建字典，为此，你可以将哈希文件所属的用户名添加到哈希值开头，根据上面的示例，我们将更改文件 hashes.txt



```
从
1efee03cdcb96d90ad48ccc7b8666033
改为
mike:1efee03cdcb96d90ad48ccc7b8666033
```

H3 答题

现在你已经熟悉了 John 的单破解模式的语法，下载附件的hash并破解它，假设它所属的用户名为 “Joker” 。

下载附件，识别hash类型，添加"Joker:"到附件的hash值的开头，再使用以下命令破解哈希：



```
john --single --format=raw-md5 hash7.txt
```

```
(root@hakeste) [/home/hakeste/桌面]
john --single --format=raw-md5 hash7.txt
Using default input encoding: utf-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=8
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 24 needed for performance.
Warning: Only 21 candidates buffered for the current salt, minimum 24 needed for performance.
Jok3r (Joker)
1g 0:00:00:00 DONE (2022-09-09 23:44) 100.0g/s 19600p/s 19600c/s 19600C/s joker..J0k3r
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Answer the questions below 回答下面的问题

What is Joker's password? 小丑的密码是什么?

Jok3r

Correct Answer

H2 自定义用户规则

H3 理论

什么是用户规则?

在我们探索 John 的单破解模式能做什么的过程中——你可能会有一些关于什么是适合的破解模式的想法，或者你的密码经常使用什么哈希模式加密——这些想法可以通过某种破解模式来复制。你可以定义自己的规则集，John 将使用这些规则来动态创建密码。当你足够了解关于你的目标密码结构的信息时，用户规则非常有用。

共同的用户规则

许多组织需要一定程度的密码复杂性来对抗字典攻击，也就是说，如果你在某个地方创建了一个帐户，你可以创建一个密码并输入：

polopassword（此处是举例）

然后你就可能会收到一个提示，告诉你密码必须至少包含以下内容之一：



大写字母

数字

符号

我们可以利用这样一个规则，即大多数用户在这些字符的输入位置上是可以预测的。对于上述密码标准要求，许多用户可能会使用以下内容：

Polopassword1!

密码首先是大写字母，最后是一个数字和一个符号，这种熟悉的密码模式，由修饰词(如大写字母或符号)附加和预置，是人们在创建密码时经常会使用和重用的令人难忘的模式。这种模式可以让我们利用密码复杂性的可预测性：从我们的字典中创建动态密码来进行密码破解。

如何创建自定义用户规则

用户规则在 `john.conf` 文件中定义，通常位于 `/etc/john/john.conf` 中，如果你已经使用包管理器安装了 John，或者使用 `make` 从源代码构建了 John，可以直接在路径中查找并查看 `john.conf` 文件。

让我们了解一下这些自定义用户规则的语法：

第一行：

`[List.Rules:THMRules]` 用于定义用户规则的名称，就是你将来作为 `John` 参数调用自定义规则的名称。

然后，我们使用正则表达式模式匹配来定义字词中的哪些地方将被修改，我们在这里只讨论基本的和最常见的修饰符：

Az 获取原字词并将你定义的字符附加在其后面

A0 获取原字词并将你定义的字符前置在其前面

c 使用大写字母

这些修饰符可以结合使用来定义你要修改的字词的位置和内容。

最后，我们要定义哪些字符应该被附加或者前置，我们通过添加字符集来实现这一点，而且 `[]` 要放在 " " 里面，下面是一些常见的例子：

<code>[0-9]</code>	包括 0-9 数字
<code>[0]</code>	只包括数字 0
<code>[A-z]</code>	包括大写和小写字母
<code>[A-Z]</code>	只包括大写字母
<code>[a-z]</code>	只包括小写字母
<code>[a]</code>	只包括字母 a
<code>[!£\$%@]</code>	包括符号 !£\$%@

将修饰符放在一起，以便根据与示例密码“Polopassword1!”匹配的规则生成一个 wordlist(假设字词 `polopassword` 在我们的 wordlist 中)，我们将创建一个如下所示的规则条目：



```
[List.Rules:PoloPassword]
cAz"[0-9] [!£$%@]"
```

把第一个字母大写
附加到原单词的末尾

数字在 0-9 之间

后面跟着一个符号（在指定范围内的符号）

```
c
Az
[0-9]
[!£$%@]
```

使用自定义规则

我们可以使用 `--rule=PoloPassword` 标志：



```
john --wordlist=[path to wordlist] --rule=PoloPassword [path to file]
```

注意：

Jumbo John 已经提供了一个自定义规则的集合，包含了很多可能需要的修饰符。如果遇到语法无法正常工作，请尝试查看这些规则[john.conf 的第678行左右]。

H3 答题

Answer the questions below 回答下面的问题

What do custom rules allow us to exploit? 自定义用户规则允许我们利用什么？

Password complexity predictability

Correct Answer 正确答案

What rule would we use to add all capital letters to the end of the word?

我们用什么规则把所有大写字母加到单词的末尾？

Az"[A-Z]"

Correct Answer 正确答案

What flag would we use to call a custom rule called "THMRules" 我们将使用什么标志来调用称为"THMRules"的自定义规则

--rule=THMRules

Correct Answer 正确答案

H2 破解受密码保护的Zip压缩文件

H3 理论

我们可以用 John 破解密码保护的 Zip 文件：首先使用 John 工具套件的一个独立部分能将 zip 文件转换成 John 能够理解的格式（提取hash值），然后再进行破解。

zip2john

与我们之前使用的 unshadow 工具类似，我们将使用 zip2john 工具将 zip 文件转换为 John 能够理解的哈希格式（提取hash值），zip2john 的基本用法是这样的：



```
zip2john [options] [zip file] > [output file]
```

[options]	允许你向 zip2john 传递特定的校验和选项，这通常是不必要的
[zip file]	希望提取hash值的 zip 文件的路径
>	这是输出指示器，我们用它来将这个文件的输出发送到 ..。
[output file]	这个文件将存储输出结果

示例用法

```
zip2john zipfile.zip > zip_hash.txt
```

开始破解

使用了zip2john之后，我们就能够在示例中获取从 zip2john 输出的名为“zip _ hash.txt”的文件，正如我们使用 unshadow 所做的那样，我们将“zip _ hash.txt”直接提供给 john，因为我们已经为该文件做了输入格式的处理：



```
john --wordlist=/usr/share/wordlists/rockyou.txt zip_hash.txt
```

H3 答题

下载附件secure.zip，进行破解。

先使用命令转换zip文件格式（提取hash值）：

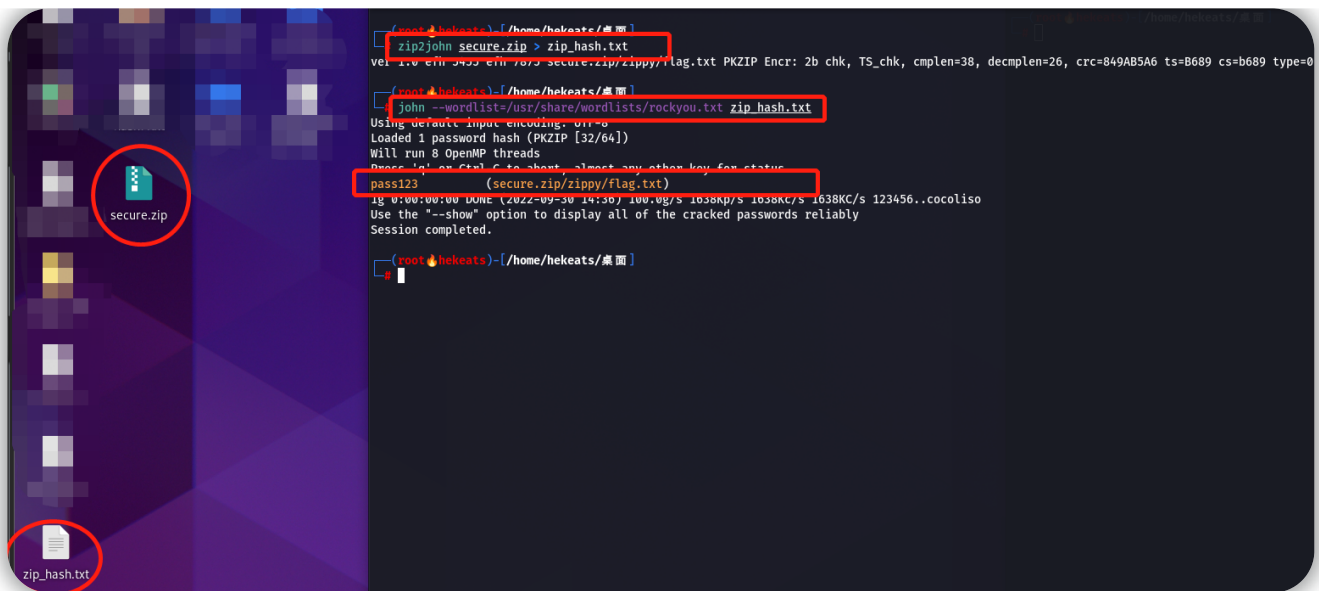


```
zip2john secure.zip > zip_hash.txt
```

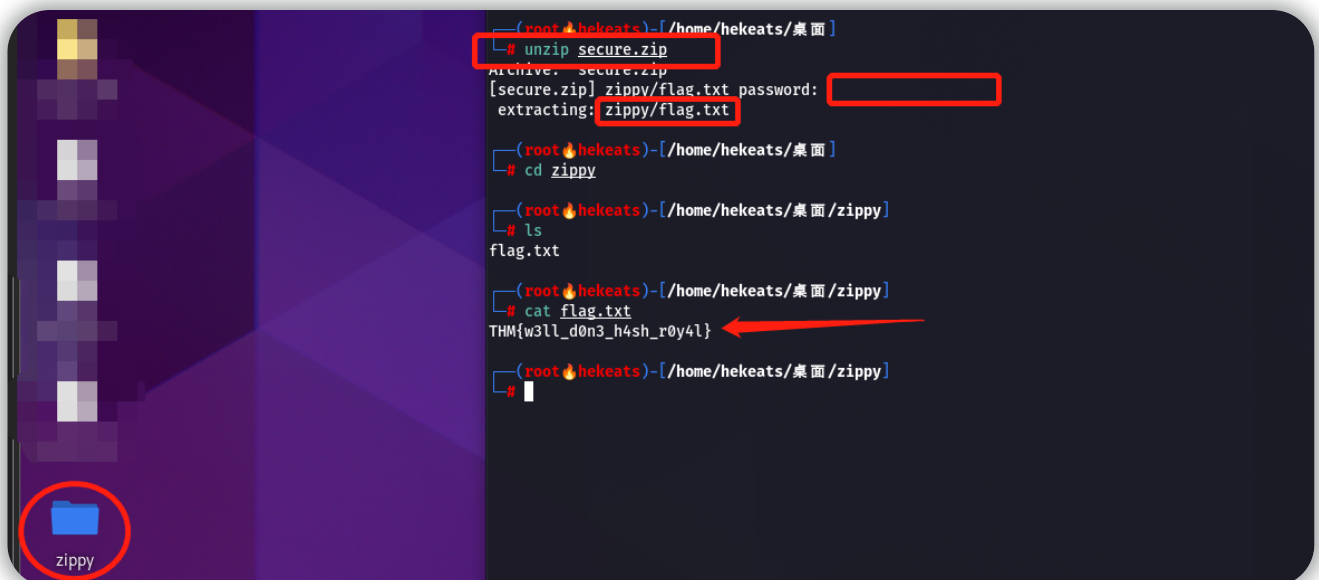
对 由zip2john处理后得到的文件进行破解：



```
john --wordlist=/usr/share/wordlists/rockyou.txt zip_hash.txt
```



使用破解得到的密码 解压zip文件并查看其中的flag文件内容：



Answer the questions below 回答下面的问题

What is the password for the secure.zip file? Zip 文件的密码是什么？

pass123

Correct Answer

What is the contents of the flag inside the zip file? 压缩文件中标志的内容是什么？

THM{w3ll_d0n3_h4sh_r0y4l}

Correct Answer

H2 破解受密码保护的 **RAR** 归档文件

H3 理论

我们可以使用与上一小节中类似的过程来破解rar归档文件的密码。rar 归档文件是由 Winrar 归档管理器创建的压缩文件，和zip一样，rar可以作为各种各样的文件夹和文件的压缩格式。

rar2john

rar2john几乎与我们刚才使用的 zip2john 工具相同，我们可以使用 rar2john 工具将 rar 文件转换为john能够理解的hash格式（提取hash值）。基本语法如下：

```
● ● ●

rar2john [rar file] > [output file]

rar2john          调用 rar2john 工具
[rar file]        想要提取hash值的rar文件的路径
>                这是输出指示器，我们用它来将这个文件的输出发送到..。
[output file]     这是存储输出结果的文件

示例用法
rar2john rarfile.rar > rar_hash.txt
```

开始破解

使用了rar2john之后，我们就能够在示例中获取从 rar2john输出的名为“rar_hash.txt”的文件，我们将“zip_hash.txt”直接提供给john，因为我们已经为该文件做了输入格式的处理：

```
● ● ●

john --wordlist=/usr/share/wordlists/rockyou.txt rar_hash.txt
```

H3 答题

下载附件secure.rar，进行破解。

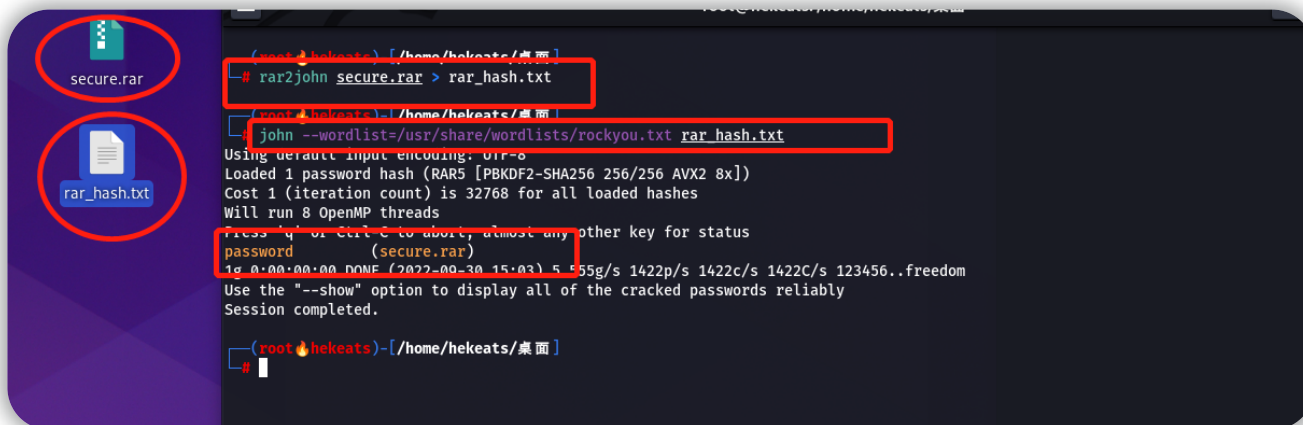
先使用命令转换rar文件格式（提取hash值）：

```
● ● ●

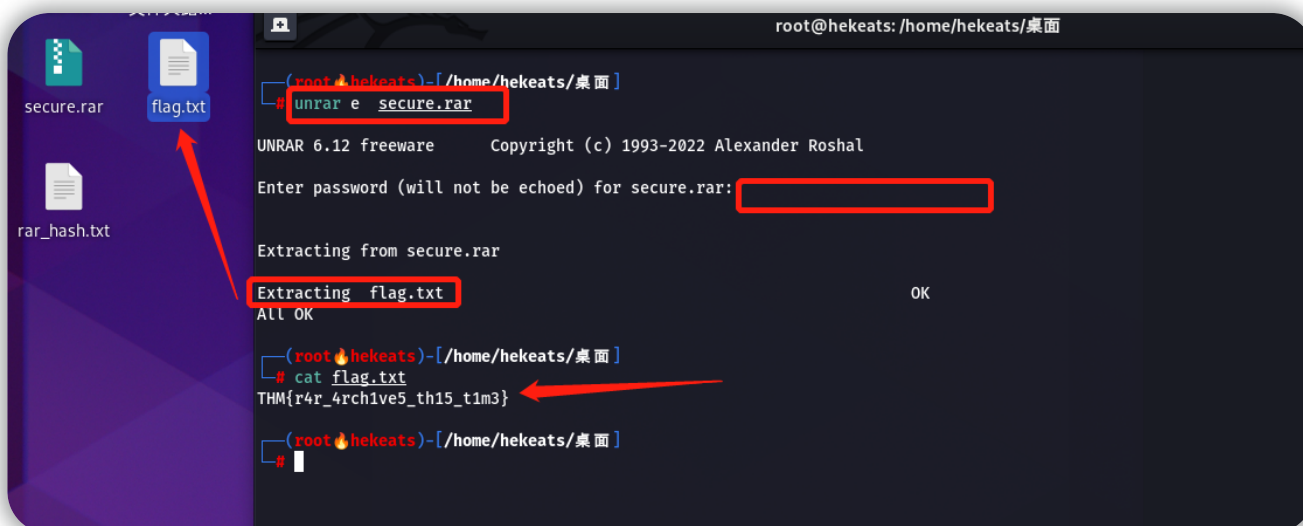
rar2john secure.rar > rar_hash.txt
```

对 由rar2john处理后得到的文件进行破解：

```
john --wordlist=/usr/share/wordlists/rockyou.txt rar_hash.txt
```



使用破解得到的密码 解压rar文件并查看其中的flag文件内容：



Answer the questions below 回答下面的问题

What is the password for the secure.rar file? Secure.rar 文件的密码是什么?

password

Correct Answer

What is the contents of the flag inside the zip file?

压缩文件中标志的内容是什么?

THM{r4r_4rch1ve5_th15_t1m3}

Correct Answer

Hint 提示

H2 使用john破解SSH密钥

H3 理论

破解 SSH 密码

使用 John 可以破解id _ rsa文件的SSH 私钥密码。除非另外配置，否则你将使用密码对 SSH 登录进行身份验证。你可以配置基于密钥的身份验证，这使你可以使用你的私钥id _ rsa作为通过 SSH 登录到远程计算机的身份验证密钥。但是，这样做通常需要设置一个密码-----在这里，我们将使用 john 来破解这个密码，以允许我们通过 SSH 使用密钥进行身份验证。

ssh2john

ssh2john 可以将用于登录到 SSH 会话的id _ rsa 私钥转换为 john能够识别的hash格式（提取hash值），使用语法如下：



```
ssh2john [id_rsa private key file] > [output file]
```

[id_rsa private key file]

希望提取hash值的id_rsa文件的路径

>

这是输出指示器，我们用它来将这个文件的输出发送到..。

[output file]

这是存储输出结果的文件

示例用法

```
ssh2john id_rsa > id_rsa_hash.txt
```

开始破解

使用了ssh2john之后，我们就能够在示例中获取从 ssh2john输出的名为“id_rsa_hash.txt”的文件，我们将“id_rsa_hash.txt”直接提供给 john，因为我们已经为该文件做了输入格式的处理：



```
john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa_hash.txt
```

H3 答题

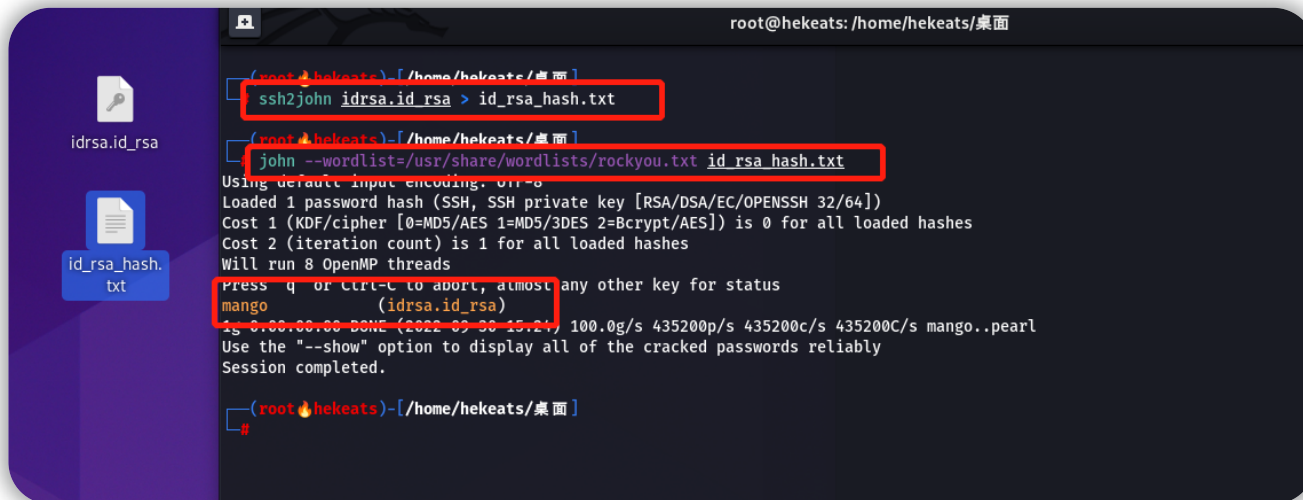
下载附件idrsa.id_rsa，进行破解。

先使用ssh2john将密钥文件转换成john能识别的格式（提取hash值）

```
ssh2john idrsa.id_rsa > id_rsa_hash.txt
```

再对ssh密钥的hash值进行破解

```
john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa_hash.txt
```



```
root@hekeats: /home/hekeats/桌面
(root@hekeats) - [/home/hekeats/桌面]
ssh2john idrsa.id_rsa > id_rsa_hash.txt
(root@hekeats) - [/home/hekeats/桌面]
john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa_hash.txt
Using default input encoding: utf-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 8 OpenMP threads
Press q or Ctrl-C to abort, almost any other key for status
mango (idrsa.id_rsa)
1g 0:00.00.00 DONE (2022-09-30-15:24) 100.0g/s 435200p/s 435200c/s 435200C/s mango..pearl
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
(root@hekeats) - [/home/hekeats/桌面]
#
```

Answer the questions below 回答下面的问题

What is the SSH private key password? SSH 私钥密码是什么?

mango

Correct Answer