THM-Encryption - Crypto 101(加密-密码)-学习

本文相关的TryHackMe实验房间链接： https://tryhackme.com/room/encryptioncrypto101

H2 基础关键词

Ciphertext： 密文

加密明文、加密数据的结果

Cipher：加密法

加密或解密数据的一种方法。现代密码是加密的，但也有许多非加密的密码，如凯撒密码。

Plaintext：明文

加密前的数据，通常是文本，但也可能是照片或其他文件。

Encryption：加密

用密码把数据转换成密文。

Encoding：编码

不是一种加密形式，只是一种类似 base64的数据表示形式。

Key：密钥

正确解密密文和获取明文所需的一些信息。

Passphrase：口令（密码短语）

与密钥分开，口令与密码类似，都能用于保护密钥。

tips：口令和密码的区别
当你输入一串字符，如果不经过任何处理直接送到服务器来验证，它一定不是密码，只是一个口令
如果输进去的字符，通过密码运算得出另外一个结果，当这个结果可以验证你是否是合法的用户时，这个口令就变成了密码

Asymmetric encryption：非对称加密

使用不同的密钥进行加密和解密。

Symmetric encryption：对称加密

使用相同的密钥进行加密和解密

Brute force：暴力破解

通过尝试每个不同的密码或每个不同的密钥来攻击加密

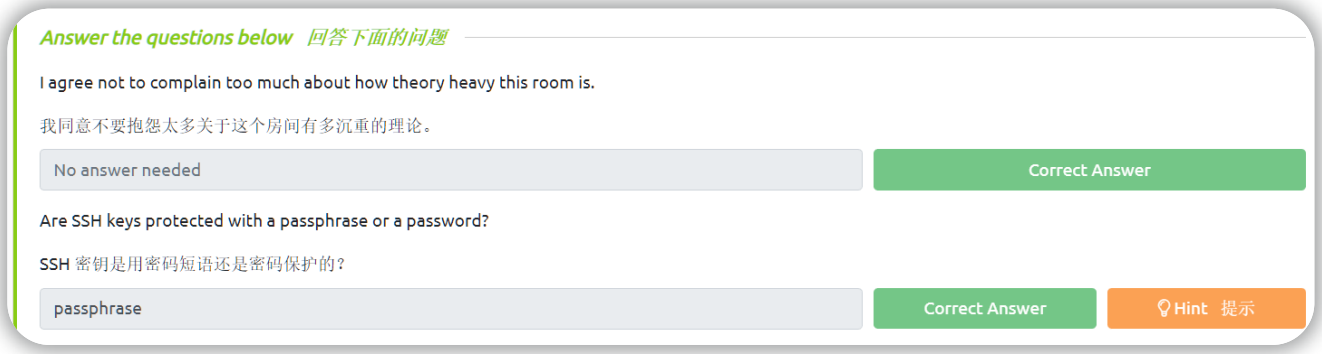
Cryptanalysis：密码分析

通过发现基础数学的弱点来攻击密码学（通过发现基数的弱点来攻击密码加密技术）

Alice and Bob（国外常用）

通常用来代表两个想要交流的人，他们被命名为爱丽丝和鲍勃，因为他们的首字母分别是 A 和 B。延伸开来也可以通过字母表上的其他字母，来代表许多不同的人参与交流。参考相关维基百科：https://en.wikipedia.org/wiki/Alice_and_Bob

答题



H2 为什么加密很重要？

加密是用来保护机密性，确保完整性，确保真实性的。你很可能每天都在使用加密技术，而且现在几乎可以肯定是通过加密连接来阅读这篇文章的（但是我开启了未登录也可观看）。

当你登录到 TryHackMe网站时，你的凭据被发送到服务器，这些都是加密的，否则有人可以通过窥探你的连接来捕获它们。

当你连接到 SSH 时，你的客户端和服务器将建立一个加密的通道，以便没有人可以窥探你的远程登录会话。

当你连接到你的银行时，会有一个证书使用加密技术来证明它实际上是你的银行而不是黑客搭建的钓鱼网站。

当你下载一个文件时，如何检查它是否下载正确？这里可以使用加密技术来验证数据的校验和（checksum ）。

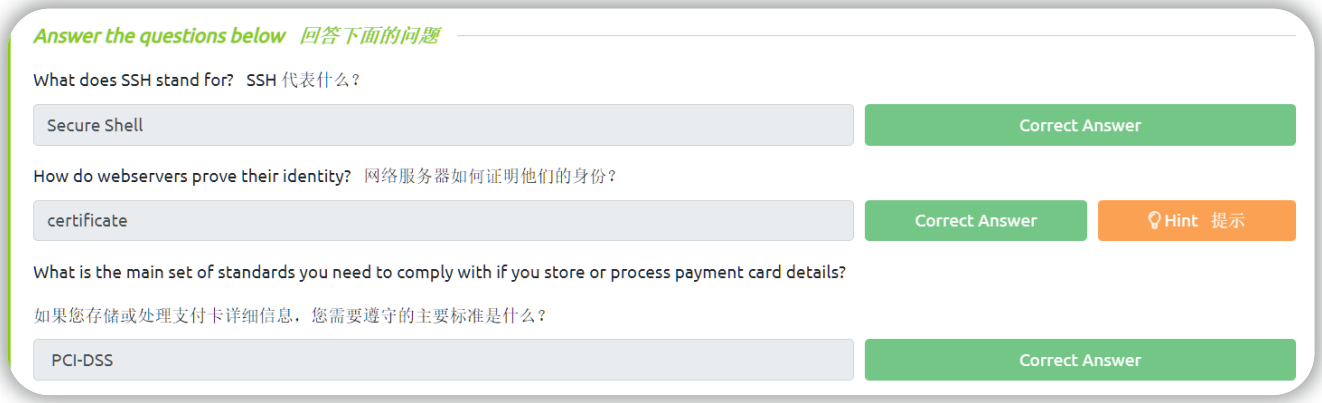
你很少需要直接与密码学进行交互，但它几乎可以静默地保护你以数字方式进行的所有操作。

无论何时需要存储敏感用户数据，都应该对其进行加密。像 PCI-DSS 这样的标准规定所提及的，数据应该在静止(存储)和传输时加密。如果你正在处理支付卡的详细信息，你就需要遵守这些 PCI 规则，医疗数据也有类似的标准。由于GDPR和加利福尼亚州（美国）的数据保护法的法律规定，数据泄露对于消费者和企业来说都是极其昂贵和危险的。

关于美国的PCI-DSS标准标准规定： https://listings.pcisecuritystandards.org/documents/PCI_DSS_for_Large_Organizations_v1.pdf

不要只是简单地加密密码，除非你使用密码管理器，密码也不应该存储在明文中，你应该使用哈希（散列）来安全地管理它们。

答题



H2 加密类型

加密的两个主要类别是对称加密和非对称加密。

对称加密使用相同的密钥对数据进行加密和解密。对称加密的例子有 DES算法 (一种脆弱的加密算法)和AES算法。这些加密算法往往比非对称加密算法运行速度更快，并且使用的密钥长度更短(128或256位密钥在 AES 中很常见，DES 密钥长度则为56位)。

非对称加密使用的是一对密钥，一个用于加密，另一个用于解密，例如 RSA算法和椭圆曲线密码学。通常，这些密钥会被称为公钥和私钥，用私钥加密的数据可以用公钥解密，反之亦然，你的私钥需要保持为私有状态，私钥因此得名。非对称加密往往速度较慢，使用的密钥位数也较长，例如 RSA 加密算法通常使用2048至4096位密钥。

答题

H2 RSA算法简介

数学中的RSA

RSA 是建立在求解大数因子这一数学难题的基础上的：把两个质数相乘很快，比如说17 * 23 = 391，但是很难算出是哪两个质数相乘能得到14351(113x127是参考答案)。

CTF中的RSA

RSA 背后的数学在 CTF（夺旗赛）中出现得比较频繁,此类题目通常要求你计算变量或者破解基于变量的加密。RSA 的维基百科页面看起来很复杂，但是它可以提供几乎所有你完成CTF挑战所需要的信息。

有一些很好的工具可以用于完成CTF比赛中关于RSA的挑战,主要是一些github项目：

> RsaCtfTool

> rsatool

对于CTF中出现的RSA，你需要了解的关键变量是p, q, m, n, e, d, 和 c

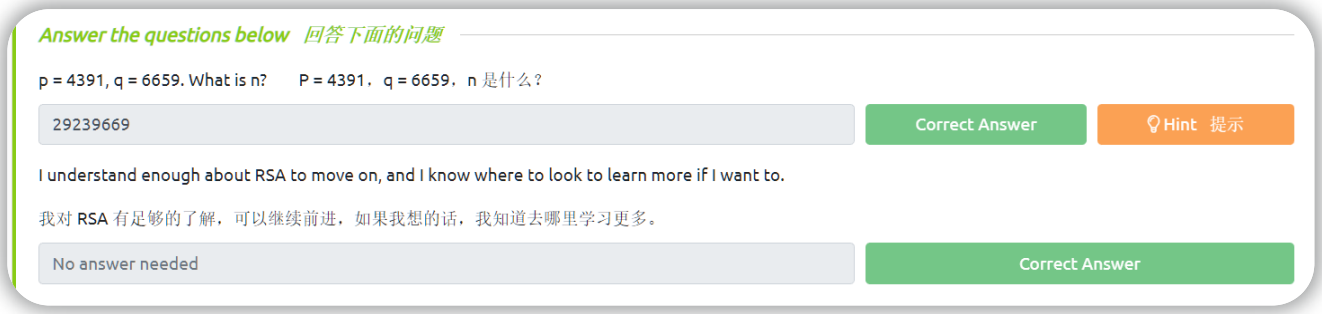
```
"p"和"q"是大质数，"n"是 p 和 q 的乘积。
公钥是 n 和 e，私钥是 n 和 d。
"m"用于表示消息(明文)。
"c"表示密文(加密文本)。
```

CTF中的加密挑战（Crypto题），通常会提供一组像上面这样的值，你需要破解加密、解密消息以拿到flag（标志）。

关于RSA 还有很多数学上的理论，而且它在以很快的速度变得相当复杂。

如果你想了解RSA背后的数学原理，我建议你阅读以下博客：https://muirlandoracle.co.uk/2020/01/29/rsa-encryption/

答题



> 在kali的终端调用计算器的命令是：calc

H2 使用非对称加密建立密钥

非对称加密的一个常见的用途是为对称加密交换密钥，因为非对称加密往往比较慢，所以对于 HTTPS 之类的东西来说，对称加密显然更好，但问题是，如何保证在传输密钥不被窥探者看到的情况下，与服务器达成密钥协议？

请看下面的比喻：

假设你有一个密码，以及如何使用这个密码的说明，如果你想在其他人无法读取的情况下将该密码说明书发送给你的朋友，你所能做的就是向你的朋友索要一把锁，只有他有这把锁的钥匙。

我们再假设你有一个坚不可摧的盒子，你可以用它来存放信息并能对它上锁，如果你把说明书放在一个锁着的盒子里发给你的朋友，他可以在收到盒子后立即解锁并阅读密码说明书。在那之后，你就可以用密码进行通信而不会被其他人所窥探。

在这个比喻中，密码表示对称加密密钥，锁表示服务器的公钥，朋友的钥匙表示服务器的私钥。你在这个过程中只使用过一次非对称加密，所以这个过程非常快，在非对称加密完成工作之后，你就可以使用对称加密私下进行通信。

实际上，你还需要更多的加密技术来验证与你交谈的人是否是你的朋友，这是通过使用数字签名和证书完成的。从以下这篇优秀的博客中，你可以找到关于 HTTPS (需要交换密钥的一个示例)如何真正工作的更多细节： https://robertheaton.com/2014/03/27/how-does-https-actually-work/

H2 数字签名和证书

什么是数字签名?

数字签名是证明文件真实性的一种方法，它能用来证明是谁创建或修改了这些文件。使用非对称加密技术时，可以利用私钥生成签名，并且可以利用公钥对签名进行验证。
由于只有你有权访问你的私人密钥，所以利用私钥生成的签名------能证明你签署了这个文件。在某些国家，数字签名和物理签名具有相同的法律价值。

最简单的数字签名形式是使用你的私钥对文档进行加密，如果有人想验证这个签名的真实性（该签名是否是你的签名），他们会使用你的公钥对文档进行解密，并检查文件是否匹配。

证书-证明你是谁

证书也是公钥加密的一个关键用途,证书能连接到数字签名，使用它们的一个常见地方是 HTTPS协议。

当你访问tryhackme.com时，你的网络浏览器怎么知道你正在与之交谈的服务器是真正的tryhackme.com网站服务器?

答案是证书，Web 服务器有一个证书，证明它是真正的 tryhackme.com。

这些证书有一个从根 CA (证书颁发机构)开始的信任链，你的设备、操作系统或浏览器从安装开始就将自动信任根 CA。从根CA往下颁发的一些组织的证书是可信的，因为根CA说他们信任这些组织，从这些组织往下颁发的证书也是受信任的，因为这些组织是受根 CA 信任的，以此类推，形成一条信任链。

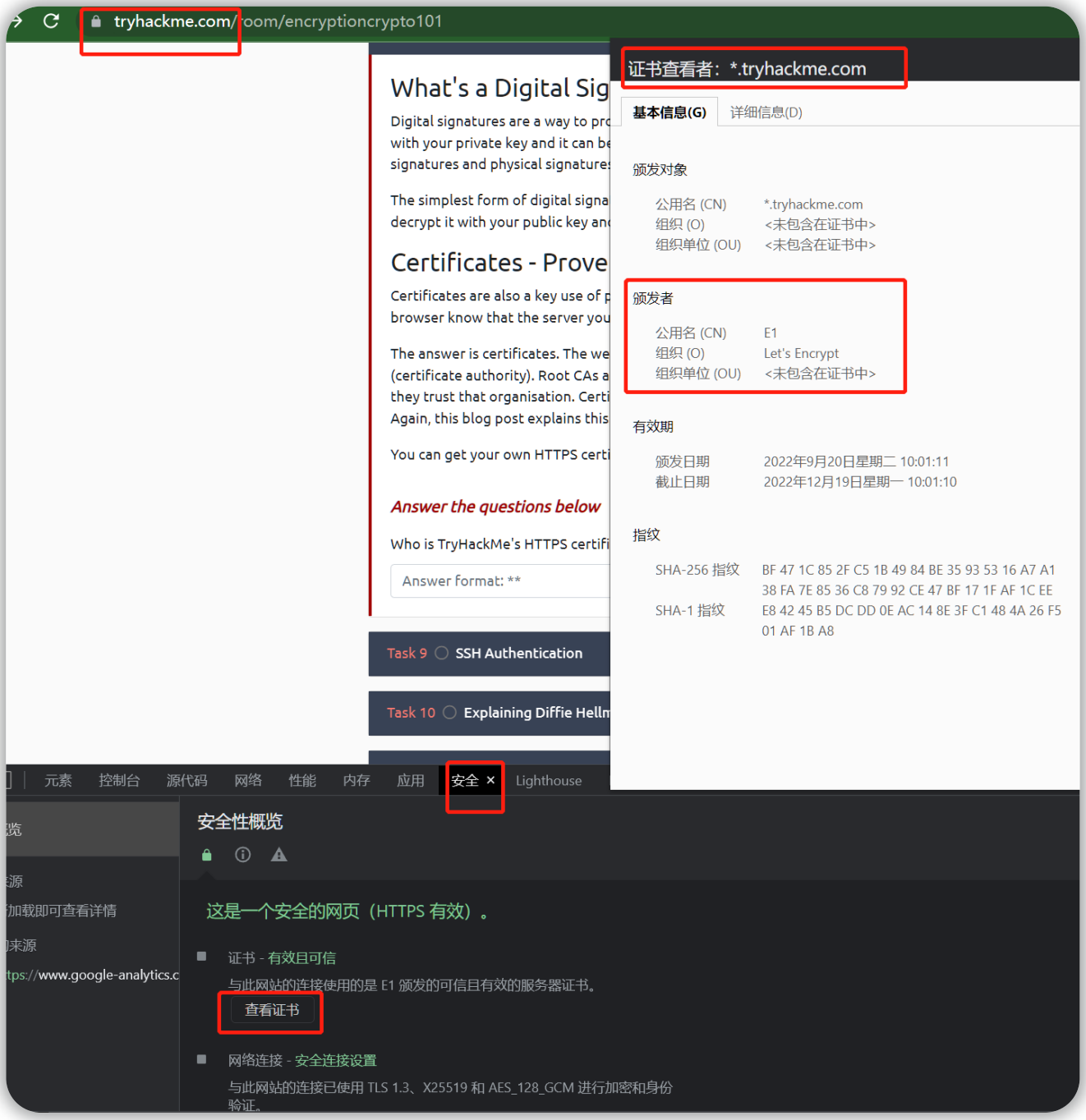
更多信息请参考博客： https://robertheaton.com/2014/03/27/how-does-https-actually-work/

你可以免费使用Let’s Encrypt，为自己拥有的域名获得自己的 HTTPS 证书，这有利于你运营一个网站。

答题

怎么知道网站所使用的HTTPS证书的颁发者：访问目标网站，按下F12查看“安全”项，点击查看证书即可。

参考链接： https://developer.aliyun.com/article/924924



H2 SSH身份认证

加密和SSH验证

在默认情况下，使用用户名和密码对 SSH 进行身份验证的方式与登录到物理计算机的方式相同。

但是你可能会遇到将SSH配置为密钥认证的机器，它要求使用公钥和私钥来证明客户端是服务器上的有效授权用户。在一般情况下，SSH 密钥会是 RSA 密钥（使用RSA算法加密），但是你也可以选择其他算法来生成SSH密钥，或者添加一个口令（密码短语）来加密 SSH 密钥。

在大多数情况下用来生成一对密钥的程序是 `ssh-keygen` ，在终端输入命令ssh-keygen即可生成密钥。

SSH私钥

如果有人拥有你的私钥，他们就可以使用该私钥登录到服务器，这些服务器将接受该私钥，除非该私钥被加密。

解密密钥的口令不是用来向服务器确认你的身份的，它所做的只是解密 SSH 密钥，它永远不会传输，也永远不会离开你的系统。

使用诸如“开膛手约翰”之类的工具，你可以攻击加密的 SSH 密钥以尝试找到口令，这在某种程度上强调了使用安全的口令和保护好私钥机密的重要性。

在生成登录到远程计算机的 SSH 密钥时,你应该在你的机器上生成密钥，然后复制公钥，这意味着私钥在目标机器上永远不存在。

怎么使用密钥

~/. ssh 文件夹是存储 OpenSSH密钥的默认位置， 这个目录中的authorized_keys（注意美国英语的拼写）文件包含公钥，如果启用了密钥认证，这些公钥就可以访问服务器。

默认情况下，许多发行版操作系统都启用了密钥认证，因为它比使用密码进行身份验证更加安全，对于 root 用户来说，通常只启用密钥认证。

要使用私有 SSH 密钥，必须正确设置权限，否则你的 SSH 客户端将忽略该文件并会提示一个警告信号。只有所有者才能读写私钥(chmod600权限或者更高的权限)。当你为标准的 Linux OpenSSH 客户端指定密钥时： `ssh -i keyNameGoesHere user@host`

使用 SSH 密钥获得更好的 shell

假设用户启用了登录(www-data 通常不启用登录，但是普通用户和 root 用户会启用登录)，那么SSH 密钥将会是“升级”反向 shell 的极好方法。

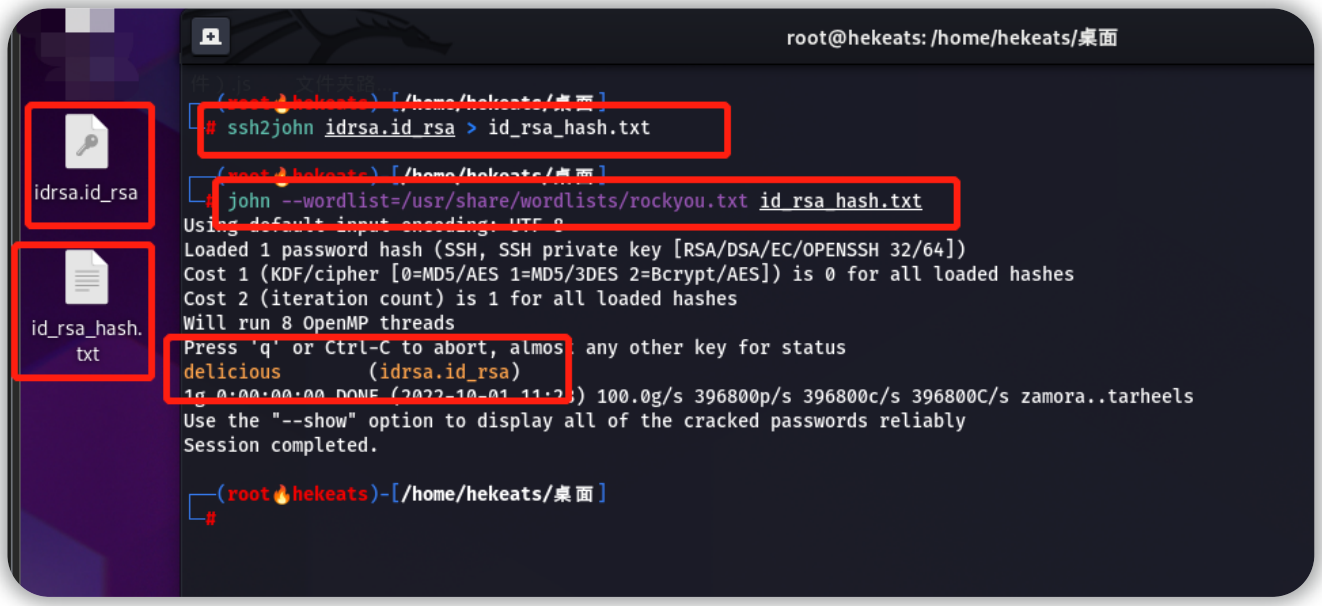
将一个 SSH 密钥保留在 authorized_keys文件中可能是一个有用的后门,而且你不需要处理任何不稳定的反向 shell 的问题，比如按Control-C会中断shell或者不能使用选项卡补全键。

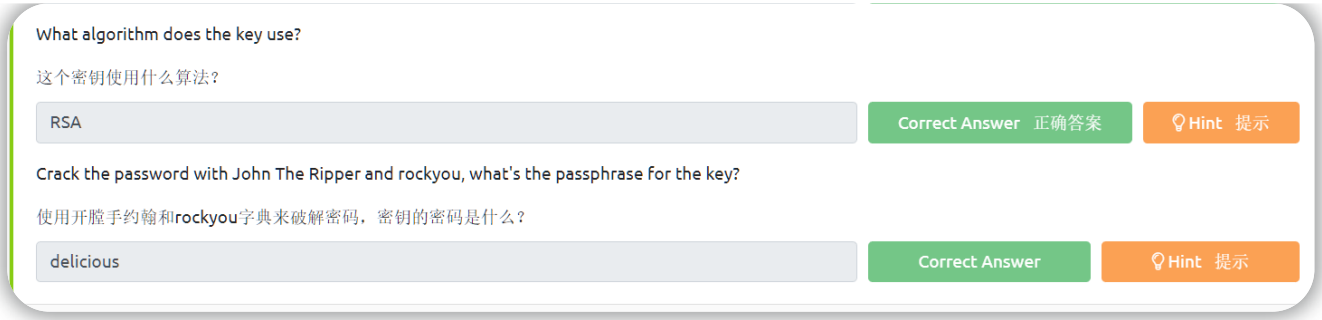
答题

下载附件，按步骤使用以下命令：

```
cat idrsa.id_rsa                                                    //查看密钥使用的加密算法

ssh2john idrsa.id_rsa > id_rsa_hash.txt                             //先使用ssh2john将密钥文件转换成john能识别的格式（提取hash值）

john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa_hash.txt    //再对提取得到的哈希值进行破解
```

```
┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# cat idrsa.id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,0B5AB4FEB69AFB92B2100435B42B7949

jc40IChbGadGmmQieKevqwq0DijIZc6T/vE1G65Umd9fvwTd9RDl5AckbkIhh2s/
u5OTGlJ2KBGCubrzjgw4pWVI8w53gcd+K/WUCtn3cmUQKrMou0xvf9BumjFTGR38
3c2WciVmCKW/8ET78zkBhJqiw0ZOJLsx1tZRYN9hhIlSp5zmYKl7MSP6U5dUoOX/
v7p5bJjBe0ykXu7uHhx6RUEuJv75uo7UihXCtg4jpaUl7iRR4DyFFF0DtxKXQLfs
O6vaLwEvGtIeqMnMrn6Or5Xlj+cxWdsxeF+DjenZYNPSpSir3a0DN0kMqnNWUEL/
jF3GctLlhALjRJzwUAsORnMAIgzuNbUo5xjrJf3H0mUELT27457VKkRb3XitSpRi
s3T2zofBvSjxFUtSxZ22AoGHwiyvpbAuq+J/mkFzOjW2z8c9g8Zf66/o51aNFbWl
ozQEcnlKK22lz/WTZJs1KZ7efooilM5YErtbyTlsxK5VJWIPToNELH4YExcILl/Z
Oyl3PdcgPiKUe5YLL+29CJ/7iHk1M9zxlSgSB+Ba2i0oTcabR159VhpH1DRw1JDs
nYR9gg6523lD3PEzNQtui2UT7S3uympRBetJYXD9I2ezY35zdYkaSDURFo/h8ykr
zWTiUmgoZefaHx8GriYaYqAVXTqTLMGXb0XB/qrxg62Gx86ReV/kU5WnMmjTwOIo
4k0CXJl6k2/LJ7sFmS/0sj4FDtqq50ixSoDE/zFF91Q2EA/IQNEH65fj2juBFIee
NzBT+MRDH/xv7s0WfymnUVKtLgm4vK9Or5KucVVoTJF14y/iFBtnaBw3+kHnkb1x
hy1J6lK96m9UrmxB61Oa0u0Mfe31Je4gRgoZOnBQHZvBj0I0ek3WLZTpysUEq7Ar
eilO+34ZRgFN6QUdmIw+I//88A9PW+s7GR+dAVVwectF6ZIZnRN3AGDlPwk4nKoG
LPxnWroCxpPvLEMmoUQ67xmH5Mj6EOEebSmV+vH4qpke//ys6iiWfyTqusVGfnAt
Oi4HpMVvZ4AYcPfNs02dgBFtbOJrPPu6mwbQaVeRG2wT43uHlZOvDDyynS9aOIlm
h2sKJsrdlOedl4aPlGTfbNZ0M3SPPau+XprA622s39DMQhnLvzuw/of85bkHYRvN
HpGmSxzas/JrifcDl+Xd1Y6SHbetaYcaZwUXC1hXPqypltbLmHIQ5NHqLgmJeFJb
442LBxdnHWUavqBSF2igPBAoVwp4UUcngS19F5Rs72qsoN3dHliF8Pf+rP56P3CR
Gm9CL4VbrC/SMQURSJj+RLUymS2EGlHggRG+LKpmqzCqPonNmRd6UycelADHmUTC
QG1gWghIdci0cw8QjioszmJRu0/Cem86/QPCiXRfsXYwqLD1ILp3DKFFXGOtHbey
EnL8ml0l+t/fI6ewIfbYBp6cqGMd0OgbGCUh57nvxGMmQ6wSPBv44s6EV2rgz8JH
MNBRcFvWiVjTSiMrEXQrzgXS24MCm9YxkTOS/FZebYrM7fH5wrqQxIp3O8xif5mr
GkSJcoDC2UWg2KEnAgZRXdL6CPjDSkFQoLo1/w09vCwhzQDgn3dKB0HShTTuxk6j
-----END RSA PRIVATE KEY-----
```





H2 介绍迪菲-赫尔曼密钥交换

什么是密钥交换

密钥交换允许2个人或者团体在没有其他窥探者能获得这些密钥的情况下建立一组通用的加密密钥，通常情况下建立的是公共对称密钥。

Defie Hellman密钥交换是怎么工作的

爱丽丝和鲍勃两人想要进行一个安全的谈话，他们想要建立一个公共密钥，他们可以使用对称加密的方式，但是他们又不想使用非对称加密的形式进行密钥交换，此时 DH 密钥交换就有了用武之地。

爱丽丝和鲍勃都有自己的秘密，我们称之为 A 和 B，同时他们也有一些公共的资料，我们称之为 C。

我们需要预先做一些假设：无论何时我们所结合的秘密或资料都不可能分离或者很难分离，其次，它们进行结合时的组合顺序并不重要。

然后爱丽丝和鲍勃将他们各自的秘密与共同的资料相结合，并形成 AC 和 BC，接着他们再把结合后的信息发送给对方，并把这些信息和他们各自的秘密再次结合起来，此时就形成两个完全相同的密钥： ABC。

完成以上过程之后，他们就可以用ABC这个密钥进行交流了。

tips

关于DH密钥交换的讲解视频： https://www.youtube.com/watch?v=NmM9HA2MQGI

DH密钥交换通常与 RSA 公钥加密一起使用，同时还可以用数字签名验证证明对方的身份，这样做可以防止有人假装自己是鲍勃：使用中间人攻击的方式攻击会话连接 。

H2 PGP, GPG and AES

什么是PGP?

PGP 代表“非常好的隐私”（Pretty Good Privacy），它是一个实现加密文件、执行数字签名等功能的软件。

什么是GPG?

GnuPG or GPG 是 GNU 项目中的PGP的一个开源实现，你可能需要使用 GPG 来解密 CTF 中的文件。在PGP/GPG中，可以使用与 SSH 私钥类似的口令来保护私钥。

如果密钥是受口令保护的，则可以尝试使用 John The Ripper 和 gpg2john 破解该口令。

关于GPG 的手册页可以通过在线找到： https://www.gnupg.org/gph/de/manual/r1023.html

什么是AES?

AES加密算法，有时会以其创造者的名字命名为 Rijndael算法，代表的是高级加密标准，它是 DES加密算法的替代品，而DES 具有短密钥缺陷和其他加密缺陷。

AES 和 DES 都对数据块进行操作（块是一系列固定大小的位）。

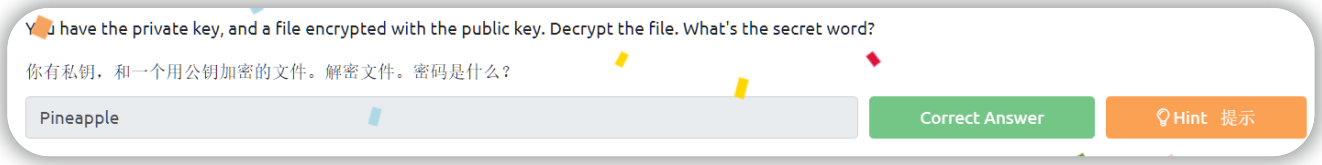
关于AES算法的视频讲解： https://www.youtube.com/watch?v=O4xNJsjtN6E

答题

下载附件并完成破解（此答题任务中提供的密钥不受口令保护）。

使用命令如下：

```
unzip gpg.zip

gpg --import tryhackme.key

gpg message.gpg

cat message
```



H2 未来——量子计算机与加密

量子计算机将很快成为许多加密类型的问题。

不对称加密与量子计算机

虽然在2030年之前，我们不太可能拥有足够强大的量子计算机，但一旦量子计算机发展起来，这些使用 RSA 或椭圆曲线密码学的加密技术，将很快被破解。这是因为量子计算机可以非常有效地解决这些算法所依赖的数学问题。

AES/DES 与量子计算机

具有128位密钥的 AES 在不久的将来也可能很容易就被量子计算机破解，三重 DES 也容易受到量子计算机的攻击，但是256位 AES 不会那么容易被破解。

现行建议

NSA 建议使用 RSA-3072或更好的非对称加密和 AES-256或更好的对称加密。目前已经有开展量子安全加密算法的竞赛，在量子计算机成为 RSA 和 AES 的威胁之前，我们可能会有一个新的加密标准。

如果你想了解更多关于这方面的信息，NIST 提供的资源详细说明了当前加密存在的问题以及当前提出的解决方案： https://doi.org/10.6028/NIST.IR.8105