链滴

从零开始 OpenSSL 之 (贰) - 使用 rsautl 解密文件

作者：adlered

原文链接：https://ld246.com/article/1552807405828

来源网站：链滴

前言

如果你没有看过第一章，请先学习第一章的内容：

点我跳转

回顾

上一章我们使用公钥public.key将text.txt加密为了encryptedFile.txt，这章我们将使用私钥private.key将ncryptedFile.txt中的原文提取出来。

解密

还是使用我们万能的**OpenSSL**。在这之前，让我们了解下**OpenSSL**中的**rsautl**命令，在命令行中输：

```
adler@localhost ~/keys: openssl rsautl --help
Usage: rsautl [options]
-in file        input file
-out file       output file
-inkey file     input key
-keyform arg    private key format - default PEM
```

```
-pubin         input is an RSA public
-certin        input is a certificate carrying an RSA public key
-ssl           use SSL v2 padding
-raw           use no padding
-pkcs          use PKCS#1 v1.5 padding (default)
-oaep          use PKCS#1 OAEP
-sign          sign with private key
-verify        verify with public key
-encrypt       encrypt with public key
-decrypt       decrypt with private key
-hexdump       hex dump output
```

让我们捋一捋，解密一个文件需要哪些参数：

```
解密 -decrypt
需要解密的文件 -in
导入私钥 -inkey
输出原文件 -out
```

你可能在想为什么不用指定-inkey给的是私钥还是公钥？因为解密必须用私钥，公钥是**无法解密**文件。

好的，那么组成我们的解密命令：

```
openssl rsautl -decrypt -in encryptedFile.txt -inkey private.key -out source.txt && cat source.t
t
```

由于我将cat命令与解密命令进行了拼接，我们能收到命令的返回值：

```
Hello world!
```

此时说明我们成功使用私钥进行了对**使用公钥加密后的文件**成功解密的处理。

后语

公私钥加密常被称为“**非对称加密**”，它很好地保护了我们在网络中传输信息的安全。这里我们只是了一个简单的实验，实际上还会有更复杂的算法存在。使用公钥加密后，黑客便无法获取明文数据，而保证了数据的安全性 - 只有拥有私钥的人才可以获取文件的内容。例如PGP Desktop软件就是基于钥而实现的文件加密传输，值得一试。