分析与总结常见勒索软件的加密算法

2016-11-27 zzz66686 FreeBuf

1、引言

1.1勒索软件

勒索软件(ransomware)是一种运行在计算机上的恶意软件，通过绑架用户文件，使用户数据资产或计算资源无法正常使用，并以此为条件向用户勒索钱财。这类用户数据资产包括文档、数据库、源代码、图片、压缩文件等多种文件。赎金形式通常为比特币，少数为真实货币或其他虚拟货币。

按照已有资料考证，早在1989年，第一款勒索软件就已经问世，但这与本文内容关系不大，不做讨论。而近期的勒索软件是从2012年开始流行，并于 2013年引起了广泛的重视。截止至当前，勒索软件的受害者已有上千万用户，这也使得勒索软件日益猖獗。

1.2本文内容

考虑到勒索软件所造成的深远影响，有必要写一些既可以给广大非技术人员了解勒索软件和加密算法又可以帮助技术人员深入分析的文章。笔者将自己近期在勒索软件方面的工作进行整理和汇总，并挑选出10款比较有代表意义的勒索软件进行深入探讨。

需要注意的是，本文着重讨论10款勒索软件样本的加密算法，而对其各自的加壳方法、反沙箱手段、提权方法和隐藏手段等不做讨论。所以本文定位于介绍这些勒索软件加密算法到底是怎样的，是否可以将其解密等相关问题。而其他的，诸如DLL 注入等问题，还是留给其他人再做分析吧。

出于让更多非技术人员读懂的目的，笔者倾向于用更通俗的语言和描述来整理勒索软件中的各种问题，但这可能导致一些不严谨或是不准确的地方，还请技术人员领会大意即可。

最后，笔者由衷地希望通过本文，可以帮助更多的受害者认识勒索软件，理性对待被加密的文件。也希望可以缩短其他技术人员的分析时间，提高分析效率。但限于笔者个人知识有限，才疏学浅，文中不当或错误之处还请各位读者包容和指正。

1.3行文结构

本文第一章为介绍本文的文字，读者可以通过第一章自行判断本文讨论的内容是否与读者需要的内容相关；第二章开始逐一介绍各个样本的加密算法；第三章则分析勒索软件被破解的原因，当然还有更多的勒索软件未能破解；第四章就目前勒索软件的发展趋势给出笔者自己的意见。

2、样本分析

从本章节开始，笔者将向各位读者逐一介绍各种勒索软件，被展示出来的10款勒索软件按字母顺序排序。针对每一款勒索软件着重介绍其加密流程，如需关注其他方面，可以单独找些相应的资料。

此外，各个勒索软件除了加密算法之后，还有大量的Hash算法，由于这类Hash算法会增加理解难度，所以在本文之中一概将其省略 。有兴趣的读者可以自行分析其中的各种Hash算法。

2.1 Apocalypse

2.1.1Apocalypse概述

Apocalypse勒索软件出现于2016年6月份，在得到广泛传播之前，就已经被破解了。但是笔者仍然决定把这款勒索软件拿出来进行分析，因为它同样可以作为某一类勒索软件加密算法的代表。一个被 Apocalypse勒索软件加密的文件夹内容如下：

从图中可以看到很明显的特征，所有被加密的文件都以.encrypted为扩展名，并生成了一个与其名称很类似的.txt文件，用于提醒用户感染了 Apocalypse勒索软件并索要赎金。

2.1.2Apocalypse加密流程

与其他所有勒索软件不同，Apocalypse勒索软件使用了一种其自定义的加密算法，并将其密钥也内置在样本之中：

其中，dl存着这该样本使用的密钥，cl为计数器。完成加密之后，将加密内容复写入用户文件，并更改其后缀名便完成其加密过程。

2.1.3Apocalypse解密流程

通过Apocalypse的加密算法，可以了解到其自定义加密算法为一种对称加密算法。所以，其解密算法与加密算法是完全相同的。

关于对称加密算法的更多细节，可以在wiki上找到：<点击原文查看链接>

2.2 Cerber

2.2.1Cerber概述

Cerber勒索软件是2016年3月份开始广泛传播的。被 Cerber加密的用户文件扩展名为Cerber，截止至笔者书写此章节（2016年 9 月）时，最新的Cerber版本已出现了Cerber2扩展名，可以通过其扩展名判断其从属于 Cerber 系列。本节所分析的Cerber勒索软件样本并非是最新样本，如需对最新的加密样本感兴趣，读者可以自行分析。

此外，被Cerber勒索软件感染之后，每个目录下还会生成额外的3个文件：# DECRYPT MYFILES #.txt 、# DECRYPT MY FILES #.html、# DECRYPT MY FILES #.vbs， 这三个文件的作用就是提示用户已经感染了 Cerber勒索软件，并索要赎金。

一个感染了Cerber勒索软件的文件夹内容如下图：

2.2.2Cerber加密流程

Cerber勒索软件的加密算法为RSA算法与RC4算法。其加密流程较为复杂，这里无法完全展示出来，只讨论几处关键点。关于 RSA算法和RC4算法的更多细节可以在wiki上找到：

RSA : <点击原文查看链接>

RC4 : <点击原文查看链接>

粗略地讲，Cerber采用三层加密方法：首先，用内置RSA密钥加密随机RSA 密钥；然后，利用随机RSA密钥加密随机RC4密钥；最后，利用随机RC4 密钥加密用户文件。

每个Cerber样本都内置了一份配置文件，该被加密存储在Cerber样本的资源段中，将其解密后可以发现一个RSA 公钥，截图如下：

此时，RSA公钥被base64编码了，使用时需要进行base64 解码。

利用该RSA公钥加密一对随机的RSA密钥。进而，利用这随机生成的RSA 密钥加密随机的RC4密钥。这随机的RC4密钥即为加密用户文件的最终密钥，其加密用户文件的RC4 算法核心代码如下：

此外，需要指出的是Cerber勒索软件并非对完整的用户文件进行加密，而是随机的把用户文件分为了几部分，并选择其中的一部分进行加密。在Cerber最终生成的文件比原用户文件大一些，其中保留了分块信息等其他在解密是需要使用到的密钥信息。

2.2.3Cerber解密流程

如果可以正确理解Cerber勒索软件的加密算法，那么就不难得出Cerber的解密算法。当忽略一些细节问题之后，其解密算法可以概述如下，首先从攻击者 C&C服务器获取RSA私钥，用该私钥解密RC4算法的密钥，进而利用 RC4 密钥解密用户文件即可。

2.3 CryptoWall

2.3.1CryptoWall概述

CryptoWall是一款在国外广泛传播的勒索软件，在2014年出现了CryptoWall的第一个版本，截止至笔者书写本小节时，CryptoWall勒索软件的最新版是第4版。本章节以CryptoWall的第三个版本为分析样本，一个感染了 CryptoWall的目录截图如下：

其中，文件扩展名为随机生成的3个字符。而文件夹内额外生成的3个文件为：HELP_DECRYPT.HTML、HELP_DECRYPT.PNG、HELP_DECRYPT.TXT，这3 个文件主要用于提示用户本机已感染了 CryptoWall勒索软件，并向用户索要赎金。

2.3.2CryptoWall加密流程

CryptoWall实现加密的方法为RSA算法和AES算法，其加密过程主要依靠微软提供的 CryptAPI实现。由于CryptoWall加密公钥需要从攻击者C&C服务器中获取，而攻击者的几个 C&C服务器很快就躺了，这在某种程度上抑制了CryptoWall的传播。

关于AES算法和CryptAPI的更多细节可以在wiki 上找到：

AES : <点击原文查看链接>

CryptoAPI :<点击原文查看链接>

正如之前所述，CryptoWall从C&C服务器获取RSA 公钥，用该公钥加密随机生成的AES密钥：

进而，通过随机生成的AES密钥加密受害者用户文件：

并将被加密的用户文件和其他的额外信息一同写入最终生成的文件中。

2.3.3CryptoWall解密流程

CryptoWall勒索软件的解密过程与加密过程是及其相似的，第一步同样是从攻击者C&C服务器获取密钥，不过这次获取的是RSA 私钥。然后用该私钥解密AES密钥。针对每个用户文件都有一个随机的AES私钥，通过该 AES 密钥解密最终的受害者用户文件即可。

2.4 CTB_Locker

2.4.1CTB_Locker概述

CTB_Locker勒索软件同样于2014年开始传播，被CTB_Locker勒索软件加密的用户文件具有相同的扩展名，均为随机的 7个字母文后缀。一个被CTB_Locker勒索软件加密的用户文件夹如下：

其中，最后两个文件分别以图片和文本的形式提示用户感染了CTB_Locker勒索软件，并要求支付赎金。

2.4.2CTB_Locker加密流程

CTB_Locker勒索软件的加密算法比较完善，也比较复杂，与Cerber类似，本小节选择用不严谨地语言描述其完整的加密算法，专业人员读到此处时，理解笔者的意思即可，不要做太深的推敲。

CTB_Locker勒索软件加密用户文件的所用算法主要是AES算法和ECDH算法。其 ECDH算法选用curve25519曲线。

关于ECDH算法和curve25519曲线的更多细节可以在wiki 上找到：

ECDH : <点击原文查看链接>

curve25519 : <点击原文查看链接>

必须指出的是，ECDH是一种密钥协商算法，但对于非技术人员，理解这种算法是有困难的。所以，笔者姑且将ECDH算法比作RSA算法 ，这种类比虽然不是完全正确，但是可以很好地帮助我们理解勒

索软件。

CTB_Locker勒索软件的加密过程可以粗略地理解为3层加密，第一层是运用内置在样本中的公钥通过ECDH算法加密随机生成的 ECDH密钥：

上图中的PblKey即为随机生成的ECDH密钥；进而第二层的结果通过随机生成的ECDH 密钥加密随机生成的AES密钥；

第三层为通过随机生成的AES密钥加密用户文件。

最后，将加密之后的用户文件，以及解密时需要用到的其他额外信息覆盖保存于用户文件。

2.4.3CTB_Locker解密流程

CTB_Locker勒索软件的正常解密流程是无法获得其主密钥的，即正常的CTB_Locker勒索软件的解密流程只包含两层，首先，通过从攻击者C&C 服务器拿到的关于随机生成的ECDH密钥相关信息，通过ECDH算法获得随机生成的AES 密钥；进而，通过该随机生成的 AES密钥解密用户文件即可。

2.5 Jigsaw

2.5.1Jigsaw概述

Jigsaw勒索软件也是一款.net托管类勒索软件，并与2016年 4月开始流传。关于.net框架的更多相关信息，可以在wiki上找到：

.net framework : <点击原文查看链接>

截止至笔者书写此章节是，Jigsaw勒索软件出现了2个版本，本小节分析的Jigsaw 勒索软件为第一个版本，其特征是被加密的用户文件以.fun为后缀。一个被Jigsaw勒索软件加密的文件夹如下：

此外，Jigsaw勒索软件并未生成额外的其他文件用于提醒用户，仅仅生成了一个窗口，显示了索要赎金等相关信息。

2.5.2Jigsaw加密流程

Jigsaw勒索软件选用AES算法作为其加密算法，其加密密钥和初始向量被编码存储于样本中：

Jigsaw勒索软件对文件的加密过程也是一目了然的，先是读取目标文件内容，然后加密，最后写入目标文件，即可完成其完全流程：

最后在增加一个文件名后缀，即完成了其加密流程。

2.5.3Jigsaw解密流程

Jigsaw勒索软件自身携带了其解密算法，当选择解密时，该软件会首先查询是提交了赎金。如果已经提交，那么则以加密时使用的密钥和初始向量完成解密流程；如果没有提交赎金，则继续计时。

2.6 Locky

2.6.1Locky概述

Locky勒索软件是另一款需要从C&C服务器申请公钥的勒索软件，是2016年 2月开始传播，由于Locky需要从攻击者的C&C服务器申请公钥，而 C&C 服务器很快就挂掉了，使Locky无法与其C&C服务器进行通信，导致无法申请到密钥，所以就不会继续运行下去。

截止至笔者书写本章时，已出现了多款不同的Locky版本，不同版本的Locky略有不同，这里选择了一个早期的 Locky 版本，一个感染了Locky勒索软件的文件夹如下：

其中，_HELP_instructions.bmp和_HELP_instructions.html是Locky 勒索软件生成的，用于提示用户感染了Locky并索要赎金。

2.6.2Locky加密流程

Locky勒索软件选用RSA算法和AES算法作为其加密算法。但不同的样本对 AES和RSA的实现略有不同，本小节分析的样本中，其RSA算法使用微软提供的 CryptAPI 实现：

其中，RSA算法用于加密随机生成的AES密钥，RSA 算法的公钥在Locky勒索软件运行时从攻击者C&C服务器中获取。AES 算法用于加密受害者的用户文件，其主要由 aesenc指令实现，该指令具有较高的执行效率：

完成加密之后，会将相关的信息与被加密的文件内容一同写入用户文件。

2.6.3Locky解密流程

Locky勒索软件的解密算法相对比较简单，首先需要从攻击者C&C服务器中拿到RSA私钥，用于解密 AES密钥，进而使用AES密钥完成用户文件的解密工作即可，此外，需要删除用户文件中额外的保留信息。

2.7 Petya

2.7.1Petya概述

Petya勒索软件于2016年3月开始流传。与其他勒索软件不同的是， Petya勒索软件并不对用户文件本身下手，其主要的恶意功能可概述为两点：其一，破坏计算机的主引导代码，使不能启动windows操作系统；其二，加密主文件表，使文件不能被访问。关于主引导代码和主文件表的更多描述可以在 wiki 中找到：

MBR : <点击原文查看链接>

NTFS MFT : <点击原文查看链接>

Petya勒索软件还有一款名为Mischa的改进版本，本小节以Petya勒索软件为分析样本。一个感染了 Petya勒索软件的计算机开机后，会显示如下内容：

其内容主要是提示用户感染了Petya勒索软件并索要赎金。

2.7.2Petya加密流程

Petya勒索软件的加密算法并不复杂，可简述为ECDH算法和SALSA20算法。其中， ECDH算法采用secp192k1曲线，用于加密SALSA20算法的密钥。关于 SALSA20 算法的更多描述可以在wiki中找到：

SALSA20 : <点击原文查看链接>

为了帮助更多的读者理解，本节中同样把ECDH算法类比为RSA算法，感兴趣的读者可以在此基础之上继续深入研究。下图展示了调用 secp192k1加密随机生成SALSA20密钥的截图：

而SALSA20算法用于加密主文件表，该算法运行在操作系统引导之前的16位环境之中，截图为引导程序调用 SALSA20 算法：

当完成上述加密步骤之后，程序会显示出其勒索页面并索要赎金。

2.7.3Petya解密流程

Petya勒索软件的解密算法仅包含一步，即从攻击者C&C服务器中拿到SALSA20的密钥，用之解密主文件表，最后将引导区还原为正常引导，即可完成其解密流程。

2.8 TeslaCrypt

2.8.1TeslaCrypt概述

TeslaCrypt勒索软件是在2015年2月首次出现。截止至笔者书写此章节时， TeslaCrypt总共有4个主要版本。不同版本的TeslaCrypt加密样本，拥有不同的扩展名，如 .ecc、.ezz、.zzz、 .vvv 、.abc等等。本文选择TeslaCrypt的第四个版本进行分析，其加密之后的文件名没有任何变化，一个感染了 TorrentLocker 勒索软件的文件夹内容如下图：

其中，-!RecOveR!-yjxka++.Htm和-!RecOveR!-yjxka++.Png以及-!RecOveR!-yjxka++.Txt 用于提示用户干扰了TeslaCrypt勒索软件，要求支付赎金。

2.8.2TeslaCrypt加密流程

不同版本的TeslaCrypt勒索软件所用的加密算法也略有不同，与CTB_Locker勒索软件类似的是，本小节分析的TeslaCrypt版本使用得同样是ECDH和AES两种，但其 ECDH 选用的secp256k1曲线。

同样的，为了让更多的非技术人员可以理解本小节所述的内容，本小节同样将ECDH算法类比为RSA算法。专业人员读到此处时，领会大意即可。

粗略地，TeslaCrypt勒索软件同样采用三层加密方法，第一层中，使用样本中内置ECDH公钥加密随机生成的 ECDH 密钥。第二层中，使用随机生成的ECDH密钥加密随机生成的AES密钥：

第三层中，使用随机生成的AES密钥加密用户文件：

加密完成之后，TeslaCrypt会将被加密的内容覆盖写入用户文件，解密需要用到的相关信息也一同被保存在其中。

2.8.3TeslaCrypt解密流程

与CTB_Locker勒索软件相同，正常情况下是无法获取TeslaCrypt勒索软件的主密钥，所以其解密过程

只包括两步，分别是通过 ECDH算法获得AES密钥，进而使用该AES密钥解密用户文件即可。

2.9 TorrentLocker

2.9.1TorrentLocker概述

TorrentLocker勒索软件是在2014年开始兴风作浪。TorrentLocker勒索软件的绝大部分版本都以 .encrypted为扩展名，但不同版本的TorrentLocker最终加密文件格式略有不同。本章节选取TorrentLocker 早期的一个版本进行分析，该样本加密用户文件之后，并不会改变原用户文件的体积，这一点可用于区分该版本与其他版本的 TorrentLocker勒索软件。

一个感染了TorrentLocker勒索软件的文件夹内容如下图：

其中，PLEASE_READ.txt为TorrentLocker勒索软件自动生成的。被 TorrentLocker勒索软件感染之后，每个目录下还会生成该文件，用于提示用户感染了TorrentLocker勒索软件，显示解密相关的信息。

2.9.2TorrentLocker加密流程

TorrentLocker勒索软件选用RSA和AES作为加密算法。其中 RSA算法用于加密AES密钥，AES算法用于加密用户文件。 AES 密钥通过Yarrow算法随机生成，关于Yarrow算法的具体描述可以在 wiki中找到：

Yarrow algorithm :<点击原文查看链接>

TorrentLocker勒索软件的随机过程中，随机种子涉及到多个函数的返回值，如下图：

截图内容不全，部分函数未能展示出来。

最终加密用户文件部分入如图：

每次加密0x10字节，直至结尾。

2.9.3TorrentLocker解密流程

需要指出的是，在TorrentLocker样本中，所有用户文件被同一AES密钥加密。在解密过程中，需要从 攻击者 C&C服务器获取AES密钥，通过该密钥即可解密所有的用户文件。

2.10 Unlock92

2.10.1Unlock92概述

Unlock92勒索软件于2016年6月开始流行，截止至笔者书写此章节时，已出现了若干个版本的 Unlock92勒索软件，该勒索软件加密用户文件之后，会增加不同的扩展名，其中以.CRRRT为后缀和以.CCCRRRPPP 为后缀的版本较为流行。相对于前者，后者的加密方法更科学，更严谨。本节选用以 .CCCRRRPPP为后缀的Unlock92勒索软件为分析样本。

Unlock92是一款.net托管类勒索软件，感染了Unlock92的用户文件夹如下：

其中，FBDX.jpg为勒索软件自动生成的图片文件，该名称不固定，图片内容为提示用户已经感染了Unlock92勒索软件。

2.10.2Unlock92加密流程

Unlock92勒索软件采用的勒索软件为两次RSA算法，每个Unlock92勒索软件都内置一个 RSA公钥，如下图：

该公钥用于加密一个随机生成的RSA私密。而这个随机生成的RSA公私密钥对，用于加密用户的全部个人文件：

需要指出的是，由于RSA算法运行速率较慢，Unlock92的作者并未对完整的用户文件全部进行加密，而是选择每个用户文件的前 0x300字节进行加密。

2.10.3Unlock92解密流程

根据Unlock92的加密过程，不难得出其解密算法。首先需要从攻击者手中拿到对应于样本内置RSA公钥的私钥，然后通过拿到的私钥解密随机 RSA私钥，最后用解密的来的RSA私钥解密用户文件即可。

3、破解原因

3.1加密流程总结

在上文中，已经概括地介绍了各个勒索软件的加密流程。而这10种勒索软件被选出并展示的主要原因在于他们可以分别代表某一类加密流程。而这10个不同的勒索软件，就涵盖了 10种不同的加密流程。其他的勒索软件的加密流程也不外乎是这十种之中的某一种。

这里在重新归纳一下各种加密算法：

> 1. 使用自定义加密算法，如2.1章节所述的勒索软件等。
>
> 2. 使用一层加密算法，如2.5章节所述的勒索软件等。
>
> 3. 使用二层加密算法，RSA+AES等，如2.9章节所述的勒索软件等。
>
> 4. 使用三层加密算法，ECDH+ECDH+AES等，如2.2章节所述的勒索软件等。
>
> 5. 借用其他正常软件的加密功能，如：CryptoHost 勒索软件借用winrar的加密功能、Vault勒索软件会借用gnupg的加密功能等等。笔者并未在本文中分析此类样本，因为这与勒索软件使用的加密算法关系不大。

在所有的标准加密和解密算法之中，AES算法的使用率是最高的，而RSA算法次之， ECDH 算法同样被一部分勒索软件采用。这些标准的加解密算法，可以认为其本身是不可解的，而造成其可解的原因完全是因为使用不当，这也是勒索软件被破解的主要原因。

3.2破解原因概括

勒索软件被破解的主要原因并不在于其加密算法不科学，而主要原因是未能正确使用其加密算法。目前，勒索软件被破解的原因可以归纳为以下几点：

> 1. 自定义算法，通过加密算法可以直接得到解密算法。在本文中第2.1章节所述的勒索软件即存在该问题。
>
> 2. 密钥保存方法不合适，如使用固定AES密钥加密用户文件。在本文中第2.5章节所述的勒索软件即存在该问题。
>
> 3. 加密强度不够，如RSA密钥长度不够，使得其被质数分解。在本文中第2.8章节所述的勒索软件的早期版本存在该问题。
>
> 4. 伪随机数生成算法不科学，使得可以预测密钥。在本文中第2.10章节所述的勒索软件的早期版本存在该问题。
>
> 5. 攻击者C&C服务器有漏洞，通过该漏洞可以获取密钥。在本文中第2.2 章节所述的勒索软件的早起版本存在该问题。

除了以上5点之外，还存在一些其他的原因可是实现破解勒索软件，但由于这些原因不具有通用性(比如， CoinVault勒索软件的作者被抓、TeslaCrypt主动放出密钥等) ，这里就不再介绍了。

4、趋势与建议

4.1趋势

按照笔者所了解的情况来看，目前勒索软件的种类和数量都有增多趋势。不仅如此，在黑市上已开始贩卖勒索软件的制作工具和勒索软件源代码。如果不采取有效措施遏制勒索软件的增长趋势，很难评估勒索软件最终会壮大到何种地步。

不过，反勒索软件的阵营也在不断成长，各种反病毒公司先后推出了对抗反病毒的软件与工具，在防范勒索软件的方法也不断的更新迭代，相信很快就会取得更丰硕的成果。

4.2建议

过去的勒索软件对于各种加解密算法的使用，或多或少的都存在一些问题。但随着勒索软件的版本更新，这些问题越来越少。一旦勒索软件的作者完全掌握了正确地使用这些加解密算法时，按目前的计算水平来看，想还原这些被加密的文件可以认为是不可能的。所以，针对于勒索软件的问题，笔者建议是以防范为主。即时地备份自己的个人文件，对于一些关键的文件，更要采取一些额外的手段进行

保护。而一旦真的感染了勒索软件也不要惊慌，更不要盲目的购买并支付比特币，咨询一些专业人员可以得到更为科学的处理手段。

此外，可以针对勒索软件的主要特征订制一些防御手段，如设置蜜罐文件等。相信通过这些方式可以有效地遏制勒索软件的传播。

ps：笔者目前处于无业状态，欢迎介绍工作。

阅读原文