基于API Hooking勒索软件WannaCry的解密方法

郭春生，程光
（东南大学计算机网络和信息集成教育部重点实验室、计算机科学与工程学院、
网络空间安全学院，江苏南京 211189）

摘 要： 勒索软件WannaCry的肆意扩散对网络用户造成了极大的危害，如何能够防范和解密WannaCry是目前的热点研究问题。为了实现对勒索软件WannaCry的解密，论文提出了基于API hooking的解密方法去恢复受害者的文件数据。首先，研究了WannaCry病毒执行时的详细加解密流程，发现其使用的加解密API函数；其次，针对这些加解密API函数实现Hook，使用自定义的钩子函数记录下密钥信息；最后，实现原型系统，监控操作系统中所有的进程。一旦勒索软件WannaCry感染主机，原型系统记录下勒索软件WannaCry使用的密钥信息，以此来完成文件的解密，结果表明系统能够完全的实现对勒索软件WannaCry的解密。

关键词： API Hooking；勒索软件；WannaCry；解密
中图分类号： TP393.08 **文献标识码：** A

An approach to decrypting ransomware wannacry based on api hooking

Guo Chun-sheng, Cheng Guang
(*Key Laboratory of Computer Network and Information Integration, Ministry of Education, School of Computer Science and Engineering, School of Cyber Science and Technology, Southeast University, JiangsuNanjing 211189*)

Abstract: Recently, the wanton outbreak of ransomware WannaCry caused great harm to the network users. How to prevent and decrypt ransomware WannaCry is a hot research issue at present. In order to implement the decryption of ransomware WannaCry, we proposes a novel method based on API hooking to decrypt and free the damaged data. Firstly, we study the encryption and decryption process of WannaCry virus and find the encryption / decryption API functions used by WannaCry. Secondly, performing API hooking for key operations, records key information with the customized hook functions. Finally, implementing the prototype system and monitoring all the processes in the operating system. When WannaCry infected the host computer, the prototype system recorded the key information to complete the decryption of the files. The result shows that the system can be effectively used to mitigate users from the damages caysed by WannaCry.

Key words: API hooking; ransomware; wannacry; decryption

1 引言

从勒索软件首次出现以来，已将近三十年时间，在这期间它并没有逐渐消失，而是发展到对各种平台进行感染，且更新速度越来越快，如今的勒索病毒软件已经具备更高强度的攻击性和更加广泛的传播性[1]。同时，由于勒索病毒软件普遍使用成熟的加密算法破坏用户文件[2]。被勒索的用户在无法恢复重要数据的情况下，被迫选择支付高额的赎金，这使得勒索者不断发动攻击以

获取更高的收益；而且，现阶段的勒索病毒软件大多是使用比特币来支付赎金，由于比特币其匿名性强、难以追踪的特点，也在一定程度上促使了勒索病毒软件爆发性的增长。

2017年5月12日起，勒索软件WannaCry发起了一场具有全球性质的网络攻击[3]，该勒索病毒使用了NSA黑客武器库泄露的工具Eternalblue，针对Windows操作系统中存在的SMB服务器漏洞发起攻击，勒索者在成功攻击后，会采取远程代码执行WannaCry勒索病毒软件。该病毒软件会对主机上存放的文件数据进行加密，同时弹出勒索页面，要求用户使用比特币进行赎金的支付以获取解密密码。随着WannaCry的爆发，一天的时间内，150多个国家的几十万台主机受到感染，其中我国部分教育机构、部分政府机关和企事业单位的网络系统受到不同程度的影响，造成严重的危机管理问题。因此，如何有效的防范WannaCry 勒索病毒软件并且顺利的恢复用户被加密的数据文件，是目前的重点研究问题。

本文提出了一种针对勒索软件WannaCry的解密方法，通过对该勒索病毒软件在Windows操作系统中运行的进程行为进行实时监测，采用API hooking方法[4]跟踪该勒索软件的进程行为操作，在钩子函数中截获其使用的密钥信息，并依此实现原型系统，对用户的数据文件完成解密。该原型系统针对WannaCry勒索病毒软件样本进行测试验证，可以完整地解密出被加密的用户数据文件；同时该系统可以在Windows操作系统上实时的进行后台监控，记录下每个使用相应API函数的进程的密钥信息，当勒索软件爆发的时候，可以根据记录下来的密钥信息实现对加密文件的解密。

2 背景技术

2.1 勒索软件WannaCry

WannaCry病毒软件使用强加密算法来加密用户的文件和数据，并且混合使用对称加密和非对称加密算法[5]，对每个文件的加密是使用AES加密算法，一次一密，同时对每个AES密钥使用RSA加密算法进行加密，并将加密后的AES密钥与加密的文件内容放置在同一个文件中。

WannaCry的加密操作关系，如图1所示[5]，WannaCry病毒在进行加密的时候，程序本身自带两个公钥（分别记为PK1和PK2）。PK2对部分文件使用的AES密钥进行加密处理，同时将该部分文件的访问路径保存在本地文件中，该部分文件是勒索软件用于展示解密操作的。该勒索软件每次运行时，都会使用RSA加密算法随机生成一组公私钥（分别记为PK3和DK3），其中PK3存储在本地文件中，主要用来加密其余用户文件的AES密钥，PK1对DK3进行加密并将加密后内容放入本地文件中。

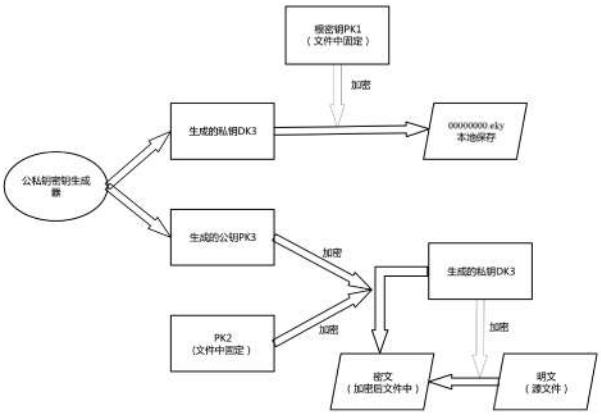

图1 WannaCry的加密操作关系

当受害者需要解密文件时，需要支付相应的赎金，并将本地保存的已加密的DK3密钥文件传送给勒索者，由勒索者使用私钥DK1对其进行解密获得DK3，并保存为00000000.dky文件，最后使用勒索病毒自带的解密程序即可完成对磁盘中文件的解密。

2.2 API Hooking技术

钩子(Hook) 在Windows操作系统中是一个独特且功能强大的特性，钩子能够在特定的消息到达目标窗口之前对其进行截获，并对该消息进行优先处理，之后再交给目标窗口的处理函数[7]。

而Windows操作系统下的应用程序一般都是使用系统API函数进行构建的，使用Hook技术时一般都是对这些API函数进行挂钩，在进程开始执行特定的API函数之前，截获系统调用消息并跳转到钩子函数进行处理，之后再交由原API函数执

行，即API Hooking，该技术可以监控API函数的执行，或者改变原有API函数的功能特性。

按照实现方法的不同， API Hooking能够分为两类：Inline Hooking(内联Hooking)和IAT Hooking(导入表Hooking)[8]。Inline Hooking是指：首先获取将要进行挂钩的API函数在内存中存放的入口地址，之后将其二进制代码替换为JMP指令，使其跳转到钩子函数所在的入口地址继续执行；IAT Hooking是指：因为IAT表中记录了进程所使用的DLL和API函数等信息，在对API函数挂钩之前，需要分析该API函数在IAT表中的相关信息，找出其在IAT表中的入口地址，将这些地址改为钩子函数所在的地址从而实现挂钩。

本文通过使用IAT Hooking方法来实现对API函数的Hook，在IAT表中更改目标API的地址，并在钩子函数中执行特定的操作，记录下勒索软件生成的密钥信息，之后继续执行程序的其他的操作。

2.3 相关工作

目前对于WannaCry病毒的破解方法是根据Adrien Guinet的成果[9]，Adrien Guinet研究员发现Windows操作系统自带的加解密API函数在生成公私钥的过程中，API函数CryptDestroyKey和CryptReleaseContext在执行时并不会从内存中删除生成公私钥的质数，通过读取内存可以恢复私钥所使用的质数，之后根据保存在本地的公钥来恢复私钥，写入00000000.dky文件中，使用WannaCry病毒自带的解密程序进行解密。该方法的局限性是要在受害者没有重新启动过计算机或没有将进行加密的进程杀掉的情况下，才有可能解密成功，否则将无法成功生成私钥文件并解密。

PayBreak[10]系统利用低开销的动态挂钩技术（Dynamic Hooking Techniques）对加密操作的API函数进行挂钩处理，获取勒索软件对每个文件加密使用的对称会话密钥，并使用非对称加密来实现密钥托管机制，使受害者能够利用保存下来的密钥信息恢复被勒索软件加密的文件。PayBreak可以完全的恢复出被加密破坏的文件，并且适用于防御和解密使用了对称加密算法的勒索软件，对于WannaCry来说也是适用的，但WannaCry有一个特别的地方，它不是用根密钥来加密对称会话密钥，而是使用非对称加密算法生成公私钥对，再用公私钥对会话密钥进行加密，因此不必要获取每个文件的会话密钥，只用获得非对称加密算法的私钥即可，更加便捷。

3 解密方法架构

根据2.1节中对勒索软件WannaCry加解密过程的详细分析可知，想要完成对WannaCry病毒的解密，需要得到运行过程中生成的私钥DK3。由于私钥DK3是使用非对称加密算法生成的，是不可逆向解密的，因此想到在病毒运行过程中，直接从内存中获取到私钥DK3的方法，同时API Hooking可以很容易实现对程序执行过程的监控和控制，只需要确定WannaCry的加解密函数以及函数的调用过程。因此本文利用API Hooking技术，监控Windows操作系统中kernel32.lib、user32.lib和advapi32.lib中的系统API函数，追踪WannaCry运行过程中详细的系统调用轨迹[11]，并打印到日志文件中。

在沙盒环境[12]中，运行WannaCry样本并对其进行API Hooking，记录下其详细的系统调用轨迹，在对其系统调用轨迹进行分析后，找出了勒索软件生成公私钥对、对密钥进行加密的操作过程，表1为WannaCry病毒生成公私钥（即PK3和DK3）的API函数调用过程（API函数的相关参数以十六进制形式显示）。

本文通过API Hooking技术追踪WannaCry病毒运行过程中的系统API调用轨迹，更加直观的发现其中对于公私钥对的生成过程、使用的API函数以及API函数的参数情况。针对加解密API函数操作过程分析发现，WannaCry使用CryptExportKey函数将公私钥（PK3和DK3）从容器中读取出来，并且将公私钥通过参数传递，由于公钥写在本地文件中，只需从此函数中获取到私钥即可，通过访问此函数的参数地址，读取出私钥。

为了防范和解密勒索软件WannaCry，本文实现的解密框架为：实时的监控操作系统中新进程的创建，对所有进程进行API Hooking并记录下每个使用相应API函数的进程的密钥信息，当主

表1 WannaCry病毒生成公私钥对的API函数调用过程

序号	API函数	描述
1	CryptAcquireContextA(,,,,)	申请CSP加密容器
2	CryptImportKey(,,,,,)	从病毒的DLL库中导入公钥PK1
3	CryptGenKey(,1,8000001,)	生成RSA公私钥对，生成的密钥长度为RSA 2048且可将公私钥对导出
4	CryptExportKey(,,6,,,)	从CSP容器中导出公钥PK3，类型值为6时为公钥
5	CreateFileA(00000000.pky,,,,,)	将公钥PK3写入00000000.pky文件中
6	WriteFile(,,114,,)	
7	CryptExportKey(,,7,,,)	从CSP容器中导出私钥DK3，类型值为7时为私钥
8	CryptGetKeyParam(,,,,)	获得了密钥的相关参数
9	CryptEncrypt(,,,,,100)	对私钥DK3使用公钥PK1进行五次加密
10	CryptEncrypt(,,,,,100)	
11	CryptEncrypt(,,,,,100)	
12	CryptEncrypt(,,,,,100)	
13	CryptEncrypt(,,,,,100)	
14	CreateFileA(00000000.eky,,,,,)	将被加密的DK3密钥写入到00000000.eky文件中
15	WriteFile(,,500,,)	
16	CryptDestroyKey()	将申请的密钥内存销毁
17	CryptReleaseContext(,)	释放掉申请的CSP加密容器

机被WannaCry感染时，将本地日志文件中记录的WannaCry病毒的密钥信息读取出来，以二进制形式写入00000000.dky文件中，即可以解密成功。本文对Windows系统中的CryptExportKey函数进行Hook，通过改变CryptExportKey函数在IAT表中的入口地址，从而跳转到钩子函数中，先执行真实的CryptExportKey函数，得到各个参数的值，之后根据CryptExportKey函数的第三个参数判断密钥类型是否为私钥，当参数密钥类型值为私钥时，访问进程中CryptExportKey函数的第五个参数的存储密钥的内存地址，读取出该进程中产生的私钥，并将私钥信息实时的写入日志文件中。解密方法框架如图2所示。

表2为算法中的自定义钩子函数伪代码显示：第1行函数即为自定义的钩子函数，第3行参数a2是密钥的类型值，其中6为公钥，7为私钥；第5行参数为密钥的存放地址指针，通过读取此指针地址的数据，可以获取到密钥数据；第6行参数为密钥的长度值；第9行执行真实的CryptExportKey函数操作，并获得相关的参数；第11行判断参数a2的密钥类型是否为私钥，若是私钥，则在第12行判断密钥的存放地址是否为空，若不为空，则在

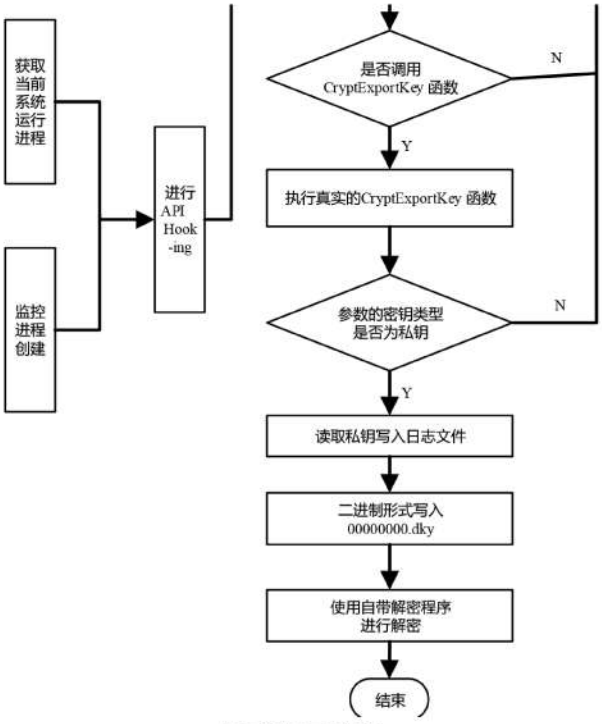

图2 解密方法框架

第13、14行以参数a5的密钥长度按字节输出密钥内容到日志文件中。

表2 算法中的自定义钩子函数伪代码显示

自定义钩子函数伪代码显示

```
1.BOOL __stdcall Mine_CryptExportKey(HCRYPTKEY a0,
2.                    HCRYPTKEY a1,
3.
 DWORD    a2,
4.
 DWORD    a3,
5.
 BYTE    *a4,
6.
 DWORD    *a5)
{
7.  BOOL rv = 0;
8.  __try {
9.     rv = Real_CryptExportKey(a0,a1,a2,a3,a4,a5);
10.  } __finally {
11.         if(a2==7){
12.          if(a4!=0){
13.                                  for(int i=0;i<*a5;i++){
14.                                   _Print("%02x",a4[i]);}
}
}
};
15.  return rv;
}
```

4 实验验证

本文在进行实验验证时，使用沙盒环境创建虚拟机，虚拟机配置为：CPU为1核，内存512MB，硬盘40GB，操作系统为Windows XP SP3，选用Windows XP操作系统的原因是由于其在此次WannaCry攻击过程中受影响的范围最大。在沙盒环境中的桌面、文档、所有人/桌面、所有人/文档、其他目录1及其他目录2下依次放置相同的样本文件，且这些样本文件的大小和类型是不一样的，并对样本源文件进行MD5哈希值[13]计算和保存，之后在沙盒中执行WannaCry病毒样本，感染系统中放置的样本文件，采用本文实现的原型工具监控WannaCry病毒的感染过程，并进行解密，对解密后的不同目录下的样本文件计算MD5哈希值，并与样本源文件的MD5哈希值进行比较，验证是否解密成功。

实验分析，依次对不同目录下的样本文件进行病毒感染以及解密，计算解密后文件的MD5哈希值，并与源文件的MD5哈希值进行比较分析，图3中为不同类型的文件解密情况表，14个doc文件、10个xlsx文件、9个ppt文件、9个txt文件、10个pdf文件以及10个rar文件全部被解密成功，对于不同类型的文件，使用本文的方法可以全部解密成功。图4中为不同大小文件的解密情况表，无论对于0K的文件还是大于200M的文件都是可以完全解密成功的。图5为桌面、文档、所有人\桌面、所有人/文档、其他目录1及其他目录2下的样本文件解密情况表，实验发现，不同目录下的样本文件都是可以完全解密成功的。图6为不同目录下对样本文件进行加解密时的时间情况，发现在桌面、文档、所有人/桌面、所有人/文档这四个目录中，对于相同的样本文件的加密时间相似，并要比其他目录1及其他目录2下的样本文件的加密时间用时久，而其他目录1及其他目录2下的样本文件的解密时间比桌面、文档、所有人\桌面、所有人/文档这四个目录中的样本文件的解密时间用时久，这是因为WannaCry勒索软件的作者根据文件的重要性对于重要目录下的文件不仅进行了加密操作，并且对源文件进行了擦写和删除，对于不重要目录下的文件仅做了加密操作，之后直接删除；解密时对于重要文件优先解密，不重要文件其次解密。

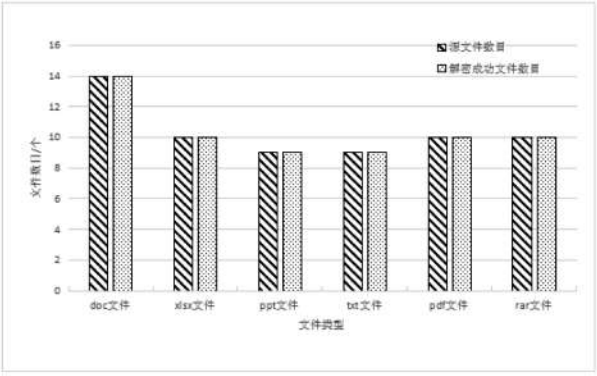

图3 不同类型文件的解密情况

本文实现的基于API Hooking的原型工具在Windows 系统下运行时的内存占用情况如图7所示：运行时内存占用为3700K左右，在后台运行

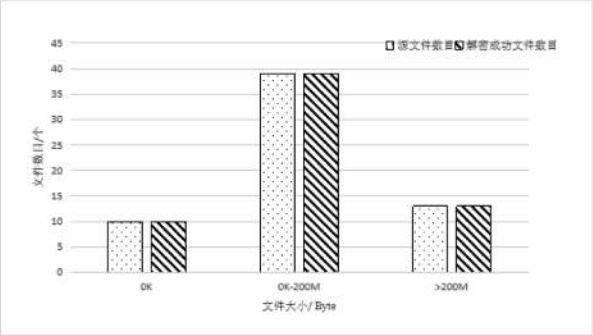

图4 不同大小文件的解密情况

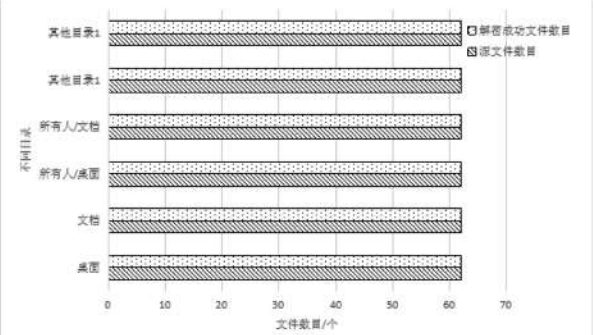

图5 不同目录下文件的解密情况

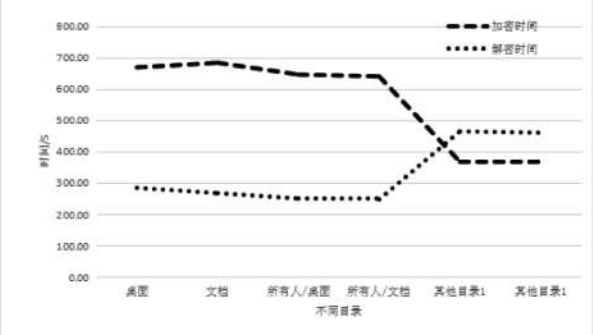

图6 不同目录下文件的加解密时间情况

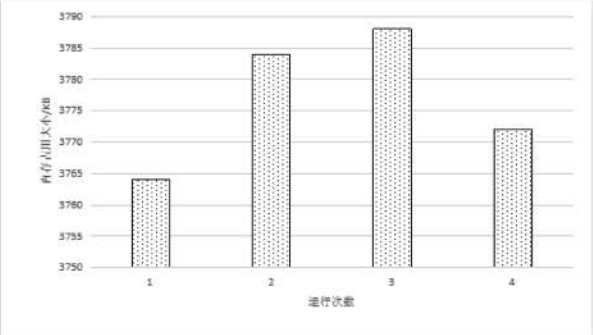

图7 基于API Hooking的原型工具运行时内存占用情况

时对系统的性能有较小的影响。

使用原型工具监控一些进程并测试对这些进程的性能影响情况，图8为这些进程进行API Hooking后增加的内存占用情况，增加的内存占用在400-500KB，对这些进程的性能影响较小。图9为WannaCry病毒和其他的使用相同加密API函数的进程在进行API Hooking后增加的运行时间情况，进行API Hooking后对于生成密钥信息的进程来讲，运行时间增加了200到300ms，而对那些没有使用加密API函数的进程来讲，运行时间几乎没有影响。

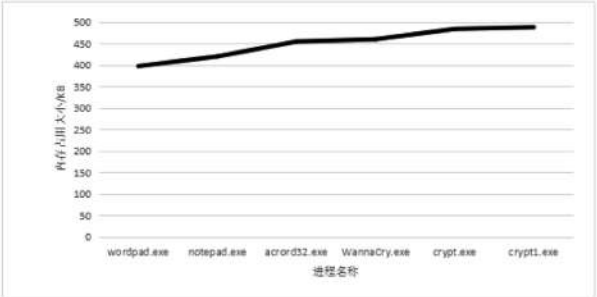

图8 进程被API Hooking后增加的内存占用情况

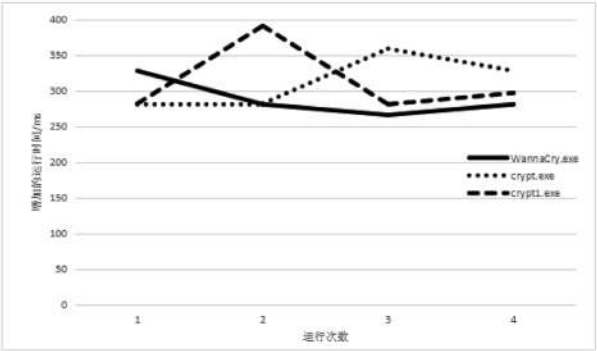

图9 加密进程进行API Hooking后增加的运行时间

根据实验验证，本文所实现的基于API Hooking技术的原型工具，可以监控到各种进程的运行，以及记录下使用CryptExportKey函数的加密程序的密钥信息，对于勒索软件WannaCry来说，经过实验验证可以完全的记录下它的密钥信息，并能够完全解密被勒索软件WannaCry加密的文件。本文实现的原型工具可以在后台实时监控系统中的进程，并对操作系统和进程的性能影响较小，可以被用户接受。

5 结束语

本文为了实现对勒索软件WannaCry的防范和解密，在仔细研究了WannaCry病毒执行时的系统调用轨迹，发现其使用的加解密API函数和解密需

要的条件，因此提出了基于API hooking的解密方法去恢复受害者的文件数据，实现了一个初步的原型系统，监控操作系统中所有的进程，针对这些加解密API函数进行挂钩，使用自定义的钩子函数记录下密钥信息。同时，对勒索软件WannaCry进行了实验验证，原型系统记录下WannaCry使用的密钥信息，并完成了对文件的解密，结果表明系统能够完全的实现对勒索软件WannaCry的解密，且对操作系统和进程的性能影响较小，可以被用户接受。此原型系统可以进一步扩展，以应对同样使用加解密API函数的勒索软件，一旦发生勒索软件感染，可以使用此原型系统记录下的密钥信息完成解密，减少用户的损失。

基金项目：

论文获得国家重点研发计划：SDN/NFV与NDN安全研究（项目编号：2017YFB0801703）和国家自然科学基金青年基金：基于网络编码的信息中心网络研究（项目编号：61602114）支持。

参考文献

[1] 安天安全研究与应急处理中心.勒索软件简史[J].中国信息安全,2016(4):50-58.

[2] 石磊,孙亮.勒索软件研究[J].无线互联科技, 2016 (21): 41-42.

[3] 瑞星安全公司.WannaCry勒索软件病毒分析报告[EB/OL]. http://www.freebuf.com/articles/paper/134637.html

[4] Kesheng L, Zhongshou W. The Analysis of API Hook Central Technique [J]. Network Security Technology & Application, 2006, 11: 48-50.

[5] Simmons G J. Symmetric and asymmetric encryption[J]. ACM Computing Surveys (CSUR), 1979, 11(4): 305-330.

[6] 腾讯电脑管家安全团队.WannaCry勒索病毒详细解读[EB/OL]. http://www.freebuf.com/articles/system/135196.html.

[7] Shaid S Z M, Maarof M A. In memory detection of Windows API call hooking technique[C]//Computer, Communications, and Control Technology (I4CT), 2015 International Conference on. IEEE, 2015: 294-298.

[8] 苏雪丽,袁丁. Windows 下两种 API 钩挂技术的研究与实现[J]. 计算机工程与设计, 2011, 32(7): 2548-2552.

[9] Adrien Guinet. A WannaCry Flaw Could Help Some Victim Get Files Back[EB/OL]. https://www.wired.com/2017/05/wannacry-flaw-help-windows-xp-victims-get-files-back/

[10] Kolodenker E, Koch W, Stringhini G, et al. PayBreak: Defense against cryptographic ransomware[C]// Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ACM, 2017: 599-611.

[11] Kharraz A, Arshad S, Mulliner C, et al. UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware[C]//USENIX Security Symposium. 2016: 757-772.

[12] Wright W, Schroh D, Proulx P, et al. The Sandbox for analysis: concepts and methods[C]//Proceedings of the SIGCHI conference on Human Factors in computing systems. ACM, 2006: 801-810.

[13] Wang X, Yu H. How to break MD5 and other hash functions[C]//Eurocrypt. 2005, 3494: 19-35.

作者简介：

郭春生（1994-），男，河南人，硕士研究生，东南大学计算机科学与工程学院，东南大学计算机网络和信息集成教育部重点实验室；主要研究方向和关注领域：网络安全。

程光（1973-），男，安徽人，工学博士，东南大学计算机科学与工程学院，教授、博士生导师，东南大学计算机网络和信息集成教育部重点实验室主任，东南大学网络空间安全学院常务副院长；主要研究方向和关注领域：网络空间安全监测和防护、网络大数据分析。