# Hidden Tear Project: Forbidden Fruit Is the Sweetest

tripwire.com/state-of-security/security-data-protection/cyber-security/hidden-tear-project-forbidden-fruit-is-the-sweetest/

Tripwire Guest Authors March 21, 2016

The scourge of ransomware is by far today's biggest computer security concern. By stepping into the crypto realm, cybercrooks have thrown down the gantlet to antivirus labs around the globe that are still mostly helpless in the face of this challenge.

While many experts have been busy reverse-engineering obtained ransomware samples and posting complex flowcharts to demonstrate their modus operandi, a Turkish programmer named Utku Sen made a very bold but questionable move.

Not only did he write code for a viable ransomware as a proof-of-concept, but he also made it publicly available on his GitHub page in mid-August 2015. The project, dubbed Hidden Tear, happens to be entirely open-source. To the author's credit, he added a disclaimer emphasizing the strictly educational goals of the initiative. This notice, predictably enough, didn't stop threat actors from taking advantage of the code in the worst way imaginable.

Since anyone with basic programming skills can use it to launch an extortion campaign of their own, the initially benign project resulted in the emergence of numerous real-world crypto Trojans with minor tweaks.

## HIDDEN TEAR 101

Utku Sen's proof-of-concept uses AES encryption to encode files located inside '\test' directory on the infected system's Desktop. The above acronym stands for Advanced Encryption Standard. Originally known as Rijndael, this algorithm is symmetric, which means that the encryption and decryption keys are identical. The key can be 128-, 192-, or 256-bits long. Ideally, either degree of entropy suffices to make brute-forcing virtually inefficient and keep a victim's files hostage.

The ransomware transmits the key to a remote server so that it's only available to the operator. To recover data, the infected person needs to have a specially crafted decryption program and the secret key at their disposal. These two prerequisites are the objects of negotiation, or rather, a bargain between the perpetrator and the user. The Trojan creates a document with detailed recovery instructions and relevant hyperlinks on the Desktop.

Owing to a lightweight payload of only 12 KB, the infection is easy to distribute through phishing emails that contain a booby-trapped attachment. Furthermore, Hidden Tear boasts antivirus evasion techniques that allow it to fly under the radar of popular AV engines. Extensive flexibility of the code makes it trivial for anybody interested to devise a custom variant of the program. The researcher also made a short video demonstrating his brainchild in action.

In a post published on his blog in late November, Sen explained his genuine motivations and responded to criticism regarding his project. In particular, he admitted that the abuse of Hidden Tear by script kiddies or other parties was a foreseeable but undesirable consequence. This is why the author deliberately incorporated a security flaw into the code, effectively turning it into a honeypot for likely offenders.

According to the researcher, the decryption key can be retrieved from the timestamp of an arbitrary ciphered file and the amount of time elapsed since the operating system launched. Once these values have been obtained via GetLastWriteTime method and Environment.TickCount property, all that's left to do is put two and two together. For the average computer expert, this shouldn't pose a difficulty.

## REAL-WORLD ABUSE INCIDENTS

The evolution of crypto malware gave birth to a new phenomenon known as Ransomware as a Service (RaaS). It denotes an affiliate framework where some criminals do the programming part and others distribute the readily available infection. Meanwhile, the latter have to share 20-25 percent of their revenue with the developer. No wonder the wannabe extortionists became interested in Utku Sen's project, which was completely free to use.

Scoundrels reportedly ended up coining more than 20 standalone strains based on Hidden Tear. In particular, the source code came in handy to the black hats responsible for the following notorious ransomware families:

1. **Encoder** is the first-ever ransom Trojan that targets Linux-based web servers. It surfaced at the beginning of November 2015. Luckily, this edition had a critical flaw that allowed researchers from Bitdefender to crack the crypto and obtain the AES key from the timestamp of any encoded file. And yet, this sample was revolutionary because never before had Linux undergone ransomware attacks.

1. Discovered by Trend Micro mid-January 2016, **B** turned out to be another incarnation of Hidden Tear. The distributor of this ransomware appears to operate in Brazil. The ransom instructions are written in Portuguese, and the racketeer demands the Brazilian currency equivalent of US$500 for decryption. Ultimately, Utku Sen was able to help the infected users since the sample was backdoored. Interestingly enough, the scammer never configured the Trojan to send the AES keys to a C&C server or simply save them anywhere. This means that victims had no chances to get their data back even if they paid the ransom.

1. **Magic Ransomware** is the most recent spin-off first spotted in late January this year. Unlike the earlier copycats, this one is based on EDA2, another POC created by Utku Sen. The malware appends .magic extension to filenames and extorts 1 Bitcoin for data restoration. For a number of reasons, which will be highlighted in the next section of this article, the whole campaign turned out an epic fail.

1. More than a dozen samples representing the **Trojan-Ransom.MSIL.Tear** family were found to also utilize Hidden Tear code. As per the in-depth analysis, however, these are script kiddies' experiments rather than professional ransomware plagues. Some of them, including Trojan-Ransom.MSIL.Tear.r and Trojan-Ransom.MSIL.Tear.t, sent AES keys to example.com domain, which the attackers configured as their Command and Control server. Obviously, the victims' data vanished for good.

## HIDDEN TEAR AUTHOR BLACKMAILED

The aforementioned Magic Ransomware case went terribly wrong. It was built with Sen's open-source EDA2 code. The researcher expected he could harness vulnerabilities in the control script to access the database of decryption keys. However, it turned out that the crooks behind the actual Trojan were using a C&C server located on a free hosting service. Someone submitted a complaint, which resulted in the takedown of the malicious Command and Control. The programmer was, therefore, unable to retrieve the database even with his pre-injected backdoor.

What happened next was unexpected for everyone involved. Distributors of the Magic virus joined the discussion of their ransomware on a popular security forum. They asked Utku Sen to remove the source code for his projects from GitHub and send them 3 Bitcoins. If these demands were met, the criminals promised they would assist everyone infected in data

recovery for free. At the end of the day, Sen abandoned Hidden Tear and EDA2, making both unavailable to the public. The hackers, in their turn, provided decryption details to the victims who asked for help.

It's unclear why exactly the perpetrators did this, but the infected users got their files back, which is a win.

## RECAP

Utku Sen's original motivations were to demonstrate researchers the ins and outs of how ransomware works. He also adopted measures to mitigate possible damage by injecting backdoors into his code. However, the emergence of Hidden Tear caused a spike in ransomware incidents. Providing a fully functional free extortion tool and expecting it to never go beyond the educational framework is wishful thinking.

Now that Hidden Tear is no longer available on official resources, there's no guarantee that interested parties will discontinue using it in new rip-off campaigns. It's naive to believe that cybercriminals failed to make and distribute copies of the code. Meanwhile, security professionals should think twice before publishing similar POCs. Even with backdoors under the hood, they may get out of hand.

**About the Author:** *David Balaban is a computer security researcher with over 10 years of experience in malware analysis and antivirus software evaluation. David runs the www.Privacy-PC.com project which presents expert opinions on the contemporary information security matters, including social engineering, penetration testing, threat intelligence, online privacy and white hat hacking. As part of his work at Privacy-PC, Mr. Balaban has interviewed such security celebrities as Dave Kennedy, Jay Jacobs and Robert David Steele to get firsthand perspectives on hot InfoSec issues. David has a strong malware troubleshooting background, with the recent focus on ransomware countermeasures.*

**Editor's Note:** *The opinions expressed in this guest author article are solely those of the contributor, and do not necessarily reflect those of Tripwire, Inc.*

*Title image courtesy of ShutterStock*