SamSam - The Evolution Continues Netting Over $325,000 in 4 Weeks

By Vitor Ventura

Published: 2018-01-22 · Archived: 2026-04-05 12:50:54 UTC

Monday, January 22, 2018 12:29

This post was written by Vitor Ventura

Introduction

Talos has been working in conjunction with Cisco IR Services on what we believe to be a new variant of the SamSam ransomware. This ransomware has been observed across multiple industries including Government, Healthcare and ICS. These attacks do not appear to be highly targeted, and appear to be more opportunistic in nature.

Given SamSam's victimology, its impacts are not just felt within the business world, they are also impacting people, especially if we consider the Healthcare sector. Non-urgent surgeries can always be rescheduled but if we take as an example patients where the medical history and former medical treatment are crucial the impact may be more severe. Furthermore, many critical life savings medical devices are now highly computerized. Ransomware can impact the operation of these devices making it very difficult for medical personnel to diagnose and treat patients leading to potentially life threatening situations. Equipment that might be needed in time-sensitive operations may be made unavailable due to the computer used to operate the equipment being unavailable.

The initial infection vector for these ongoing attacks is currently unknown and Talos is investigating this in order to identify it. The history of SamSam indicates that attackers may follow their previous modus operandi of exploiting a host and then laterally moving within their target environment to plant and later run the SamSam ransomware. Previously, we observed the adversaries attacking vulnerable JBoss hosts during a previous wave of SamSam attacks in 2016. Although the infection vector for the new variant is not yet confirmed, there is a possibility that compromised RDP/VNC servers have played a part in allowing the attackers to obtain an initial foothold.

There are no differences between the encryption mechanism used by this current SamSam variant compared to older versions. However, this time the adversaries have added some string obfuscation and improved the anti-analysis techniques used to make detection and analysis marginally more difficult.

This new variant is deployed using a loader which decrypts and executes an encrypted ransomware payload, this loader/payload model represents an improvement in the anti-forensic methods used by the malware. Samples containing this loader mechanism have been found as far back as October 2017. The wallet used by SamSam for this wave is shared by multiple infected victims as observed by monitoring the wallet at

1MddNhqRCJe825ywjdbjbAQpstWBpKHmFR. We are also able to confirm the first payment into this wallet was received on 25th December 2017 - a nice holiday gift for this adversary. This can be confirmed by observing the first wallet transaction found on the Bitcoin blockchain here. There is a possibility that other Bitcoin wallets are also used but currently Talos is currently unaware of any others.

Similar to the previous variants, we believe the deployment of this SamSam variant to be highly manual, meaning an adversary must take manual action in order to execute the malware. The symmetric encryption keys are randomly generated for each file. The Tor onion service and the Bitcoin wallet address are hardcoded into the payload whilst the public key is stored in an external file with the extension .keyxml.

Additionally, code analysis didn't find any kind of automated mechanism for contacting the Tor Service address which means that the victim identification with the associated RSA private key must be done either manually or by another adversary tool.

Ransom note displayed by SamSam new variant

In most ransomware the attackers try to convince affected users that they have the ability to decrypt the data after the payment is made. SamSam is no different here and even displays a disclaimer as seen in the above screenshot, stating 'we don't want to damage our reliability' and 'we are honest'.

To this end SamSam adversaries offer free decryption of two files and an additional free key to decrypt one server. Once again SamSam actors show their ability to monitor and laterally move through the network by pointing out they will only provide a key if they believe the server is not an important piece of infrastructure. As with previous versions of SamSam they are advising that messaging the attackers can be performed via their site.

The "Runner"

The adversary has changed their deployment methodology and now they use a loader mechanism called "runner" to execute the payload. Upon execution, the loader will search for files with the extension .stubbin in its execution

directory, this file contains the SamSam encrypted .NET Assembly payload. Upon reading the file, the loader decrypts the payload with the password supplied as the first argument and executes it, passing the remaining arguments.

The loader is a very simple .NET assembly with no obfuscation. Comparing both the Initialization Vector (IV) and the code structure it seems like it may have been derived from an example posted on the Codeproject.com website.

As you can seen in the images below, the IV used for the Rijndael encryption is the same in both implementations (posted code in hexadecimal, reversed code in decimal due to decompiler implementation).

`PasswordDeriveBytes pdb = new PasswordDeriveBytes(Password, new byte[] {0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76});`	`PasswordDeriveBytes passwordDeriveBytes = new PasswordDeriveBytes(Password, new byte[13]{73,118,97,110,32,77,101,100,118,101,100,101,118});`
Posted code	Reversed code

At the code level looking specifically at the function 'Decrypt', it is obvious that the code structure in the Codeproject source and the latest SamSam runner sample is the same (comments from the posted code were removed).

```
    MemoryStream memoryStream = new MemoryStream();
    Rijndael rijndael = Rijndael.Create();
    rijndael.Key = Key;
    rijndael.IV = IV;
    CryptoStream cryptoStream = new CryptoStream(memoryStream,
    rijndael.CreateDecryptor(), CryptoStreamMode.Write);
    cryptoStream.Write(cipherData, 0, cipherData.Length);
    cryptoStream.Close();
    return memoryStream.ToArray();
}
```

Reversed code

```
blic static byte[] Decrypt(byte[] cipherData, byte[] Key, byte[] IV)
{
    MemoryStream ms = new MemoryStream();
    Rijndael alg = Rijndael.Create();
    alg.Key = Key;
    alg.IV = IV;
    CryptoStream cs = new CryptoStream(ms, alg.CreateDecryptor(),
    CryptoStreamMode.Write);
    cs.Write(cipherData, 0, cipherData.Length);
    cs.Close();
    byte[] decryptedData = ms.ToArray();
    return decryptedData;
}
```

Posted code

Encryption routine source code comparison

The Payload

Previous versions of SamSam put some effort into the obfuscation of the malware code by encrypting strings with AES. The new version also obfuscates functions, class names and strings, including the list of targeted file extensions, the help file contents and environment variables, this time using DES encryption with a fixed hard-coded key and the IV.

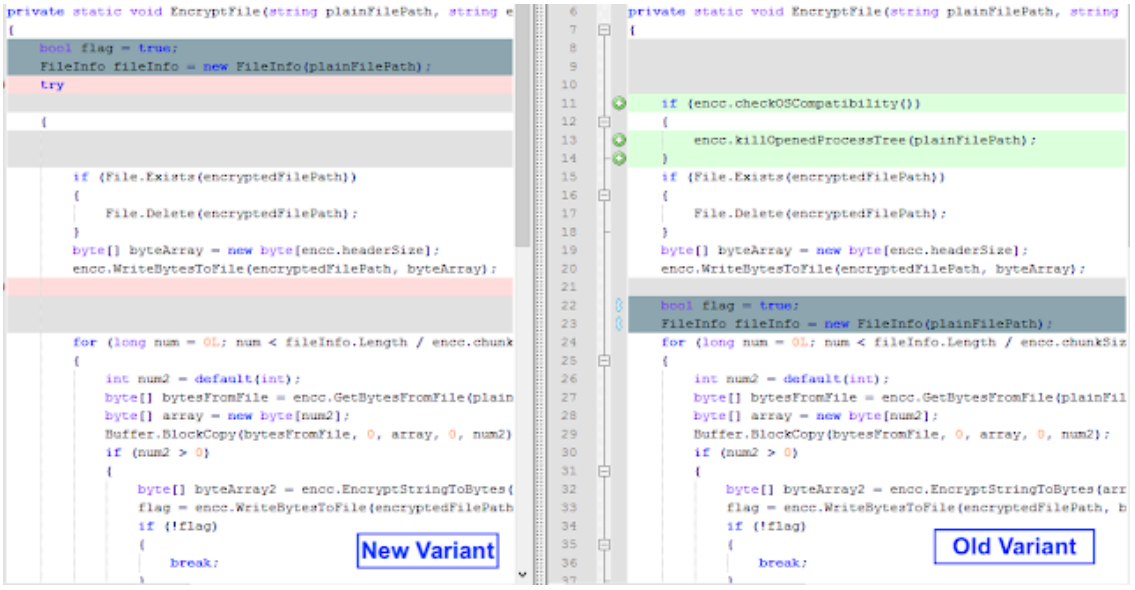
```
array = Encoding.UTF8.GetBytes(s);
byte[] array2 = new byte[inputString.Length];
array2 = Convert.FromBase64String(inputString);
DESCryptoServiceProvider dESCryptoServiceProvider = new DESCryptoServiceProvider();
memoryStream = new MemoryStream();
ICryptoTransform transform = dESCryptoServiceProvider.CreateDecryptor(array, rgbIV);
CryptoStream cryptoStream = new CryptoStream(memoryStream, transform, CryptoStreamMode.Write);
cryptoStream.Write(array2, 0, array2.Length);
cryptoStream.FlushFinalBlock();
```

Once again, the adversary has put more effort into preventing the forensic recovery of the malware sample itself rather than only relying on the obfuscation the running malware code, which allowed us to reverse engineer this sample.

As mentioned before, the password to decrypt the payload is passed as a parameter to the loader, which reduces the chances of obtaining the payload for analysis.

Previous versions of SamSam had an equivalent method for making payload access difficult by launching a thread that would wait 1 second before deleting itself from the hard disk.

The comparison of the main encryption routines between the old and the new samples indicates that this version of SamSam is similar enough to have high confidence that it belongs to the same malware family.



Encryption Routine Comparison

While previous SamSam versions used the API call DriveInfo.GetDrives() to obtain the list of available drives, this new version has the drive letters hardcoded. After checking that a drive is ready it starts a search for targeted files on the non-blocklisted folder paths.

The new variant keeps the same list of targeted file extensions as some of the previous ones. It adds a few new entries to the list of paths not to encrypt, which includes user profiles "All Users", "default" and the boot directory.

This is in tune with most ransomware which attempt to preserve the operability of the victim's machine. If the machine operation is so damaged that the system cannot be booted then the victim will be unable to pay, whereas if they keep the machine able to function, with limited access to files/folders, then they have a greater chance of a victim paying for recovering their important files and documents.

```
public static bool cond1(string file_path)
{
    if (file_path.ToLower() != Program.win_drive_installed.ToLower() + "windows" && file_path.ToLower() != Program.win_drive_installed.ToLower() +
    "winnt" && !file_path.ToLower().Contains("reference assemblies\\microsoft") && !file_path.ToLower().Contains("recycle.bin") && !
    file_path.ToLower().Contains(Program.win_drive_installed.ToLower() + "users\\all users") && !file_path.ToLower().Contains
    (Program.win_drive_installed.ToLower() + "documents and settings\\all users") && !file_path.ToLower().Contains
    (Program.win_drive_installed.ToLower() + "boot") && !file_path.ToLower().Contains(Program.win_drive_installed.ToLower() + "users\\default"))
    {
        return true;
    }
    return false;
}
```

Just like previous versions of SamSam the new version is especially careful to make sure that there is enough space on the current drive to create the encrypted document, thus avoiding any corruption that would lead to irrecoverable encryption.

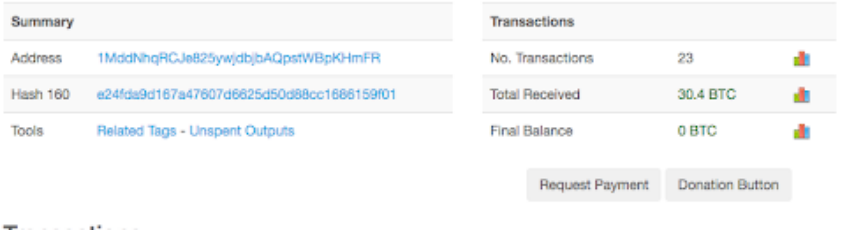
```
public static void my_func_ready_for_enk__(string _fullname_file__)
{
    FileInfo fileInfo = new FileInfo(_fullname_file__);
    try
    {
        long availableFreeSpace = new DriveInfo(_fullname_file__).AvailableFreeSpace;
        long length = fileInfo.Length;
        if (length < availableFreeSpace && new FileInfo(_fullname_file__).Length > 0 && !File.Exists(fileInfo.DirectoryName + "\\" + fileInfo.Name
        + Program.my_exnt_rns) && !string.IsNullOrEmpty(Program.my_pbb_keyy_))
        {
            Program.my_func_for_one_file___(_fullname_file__, Program.my_pbb_keyy_);
            if (File.Exists(fileInfo.DirectoryName + "\\" + fileInfo.Name + Program.my_exnt_rns) && new FileInfo(fileInfo.DirectoryName + "\\" +
            fileInfo.Name + Program.my_exnt_rns).Length > length)
            {
                if (!File.Exists(fileInfo.DirectoryName + "\\" + Program.my_hlp_name_ + Program.my_extn_hlp_))
                {
                    File.WriteAllText(fileInfo.DirectoryName + "\\" + Program.my_hlp_name_ + Program.my_extn_hlp_,
                    Program.my_content_of_hlp_file__);
                    for (int i = 0; i < 10; i++)
                    {
                        File.WriteAllText(fileInfo.DirectoryName + "\\000" + i.ToString() + "-" + Program.my_hlp_name_ + Program.my_extn_hlp_,
                        Program.my_content_of_hlp_file__);
                    }
                }
                fileInfo.Attributes = FileAttributes.Normal;
                File.Delete(_fullname_file__);
            }
        }
    }
```

Makes sure there is enough free space

Unlike most ransomware, SamSam does not delete Volume Shadow Copies and creates an encrypted version of the original file which is then deleted using the regular Windows API. Although unlikely, due to block overwriting, recovery of the original files from the versions of affected folders saved by the operating system may be possible.

Profitability

In identifying the scope of this SamSam campaign, Talos analyzed the Bitcoin wallet addresses used by the attackers in each of these attacks. As of the time of this writing, the attackers have received approximately 30.4 BTC which equals $325,217.07. As previously mentioned, it is possible that the attackers are leveraging multiple bitcoin wallets, however Talos has not observed any other than the one listed here being used in these attacks.



Recommendations

As the specific initial threat vector is not known at this time, best practices should be implemented to minimize risk to organizations. Talos has outlined several best practices that should be considered in a previous blog related to defending against ransomware related threats. In accordance with best practices protocols like SMB or RDP should never be internet facing.

IOCs SHA256s

0785bb93fdb219ea8cb1673de1166bea839da8ba6d7312284d2a08bd41e38cb9
338fdf3626aa4a48a5972f291aacf3d6172dd920fe16ac4da4dd6c5b999d2f13
3531bb1077c64840b9c95c45d382448abffa4f386ad88e125c96a38166832252
4856f898cd27fd2fed1ea33b4d463a6ae89a9ccee49b134ea8b5492cb447fb75
516fb821ee6c19cf2873e637c21be7603e7a39720c7d6d71a8c19d8d717a2495
72832db9b951663b8f322778440b8720ea95cde0349a1d26477edd95b3915479
754fab056e0319408227ad07670b77dde2414597ff5e154856ecae5e14415e1a
88d24b497cfeb47ec6719752f2af00c802c38e7d4b5d526311d552c6d5f4ad34
88e344977bf6451e15fe202d65471a5f75d22370050fe6ba4dfa2c2d0fae7828
8eabfa74d88e439cfca9ccabd0ee34422892d8e58331a63bea94a7c4140cf7ab
8f803b66f6c6bc4da9211a2c4c4c5b46a113201ecaf056d35cad325ec4054656
dabc0f171b55f4aff88f32871374bf09da83668e1db2d2c18b0cd58ed04f0707
e7bebd1b1419f42293732c70095f35c8310fa3afee55f1df68d4fe6bbee5397e

BTC Wallet

1MddNhqRCJe825ywjdbjbAQpstWBpKHmFR

Tor onion service

jcmi5n4c3mvgtyt5.onion

Detection

Snort Rules: 45484-45486

AMP for Endpoints: Ensure the TETRA engine, ‘Command Line Capture’, “System Process Protection” are enabled, and client is v6.05+

Source: http://blog.talosintelligence.com/2018/01/samsam-evolution-continues-netting-over.html