



# How Ransomware Attacks

What defenders should know about the most prevalent and persistent malware families

Ransomware's behavior is its Achilles' heel, which is why Sophos spends so much time studying it. In this report, we've assembled some of the behavioral patterns of the ten most common, damaging, and persistent ransomware families. Our goal is to give security operators a guideline to understand the core behaviors that underlie ransomware attacks, which we also use to convict ransomware with Sophos' behavioral engine, Intercept X.

By **Mark Loman**, Director, Engineering

# Contents

Introduction	3		
Ransomware	4		
Traits	4		
Dividing ransomware into categories	4		
Cryptographically signed code	4		
Privilege escalation (and lateral movement)	5		
Network first	7		
Multi-threaded	7		
File encryption	8		
Rename	8		
Key blob	9		
Wallpaper	9		
Vssadmin	9		
BCDEdit	10		
Cipher	10		
0 allocation	10		
Flush buffers	11		
Encryption by proxy	11		
Overview	12		
Color coding	13		
WannaCry	13		
Characteristics	13		
WannaCry file system activity	14		
Matrix	15		
Characteristics	15		
Matrix file system activity	16		
		GandCrab	17
		Characteristics	17
		GandCrab file system activity	18
		SamSam	19
		Characteristics	19
		SamSam file system activity	20
		Dharma	21
		Characteristics	21
		Dharma file system activity	21
		BitPaymer	22
		Characteristics	22
		BitPaymer file system activity	22
		Ryuk	23
		Characteristics	23
		Ryuk file system activity	23
		LockerGoga	24
		Characteristics	24
		LockerGoga file system activity	24
		Partial encryption	24
		MegaCortex	25
		Characteristics	25
		MegaCortex file system activity	25
		RobbinHood	26
		Characteristics	26
		RobbinHood file system activity	26
		Sodinokibi	27
		Characteristics	27
		Sodinokibi file system activity	28
		Indicators of Compromise (IOCs)	28

### Introduction

Most blogs or papers about crypto-ransomware typically focus on the threat's delivery, encryption algorithms and communication, with associated indicators of compromise (IOCs). This research paper takes a different approach: an analysis of the file system activity or behaviors of prominent crypto-ransomware families (hereafter, simply called ransomware).

Ransomware creators are acutely aware that network or endpoint security controls pose a fatal threat to any operation, so they've developed a fixation on detection logic. Modern ransomware spends an inordinate amount of time attempting to thwart security controls, tilling the field for a future harvest.

It's a lot easier to change a malware's appearance (obfuscate its code) than to change its purpose or behavior, and ransomware always shows its tell when it strikes. The increasing frequency with which we hear of large ransomware incidents indicates that the code obfuscation techniques ransomware now routinely employs, such as the use of runtime packers, must continue to be fairly effective against some security tools, otherwise the ransomware makers wouldn't use them.

It's important to recognize there's hope in this fight, and a number of ways admins can resist: Windows 10 Controlled Folder Access (CFA) whitelisting is one such way, allowing only trusted applications to edit documents and files in a specified location. But whitelisting isn't perfect – it requires active maintenance, and gaps or errors in coverage can result in failure when it's most needed.

## Ransomware

### Traits

Criminals are constantly releasing new ransomware variants. To endpoint protection products that rely on static analysis, these new variants bear no resemblance to earlier samples. As with other forms of malware, ransomware creators apply runtime packers to the ransomware program, helping to conceal its purpose and avoid detection until it has completed its core task.

In most cases, ransomware creators use proprietary, non-commercial packers that thwart automated unpacking routines used by endpoint protection software, making it harder to classify and determine the intention of the packed executable, as well as more difficult for human analysts to reverse engineer.

There are behavioral traits that ransomware routinely exhibits that security software can use to decide whether the program is malicious. Some traits – such as the successive encryption of documents – are hard for attackers to change, but others may be more malleable. Mixing it up, behaviorally speaking, can help ransomware to confuse some anti-ransomware protection.

### Dividing ransomware into categories

For this report we investigated several prominent ransomware families, and have categorized them into three categories, distinguishing them by the method attackers use to spread the infection:

1. **Cryptoworm** - A standalone ransomware that replicates itself to other computers for maximum reach and impact.
2. **Ransomware-as-a-Service (RaaS)** – A ransomware sold on the dark web as a distribution kit to anyone who can afford it. These RaaS packages allow people with little technical skill to attack with relative ease. They are typically deployed via malicious spam e-mails (malspam), via exploit kits as a drive-by download, or semi-manually by automated active adversaries.
3. **Automated Active Adversary** – Here, the ransomware is deployed by attackers who use tools to automatically scan the internet for IT systems with weak protection. When such systems are found, the attackers establish a foothold and from there carefully plan the ransomware attack for maximum damage. For example, services that are openly exposed to the internet – like the Remote Desktop Protocol (RDP) – are a sought-after entry point as they are susceptible to a brute-force password-guessing attack. Although victims may believe they are targeted, the attack is usually opportunistic.

### Cryptographically signed code

Attackers may attempt to minimize detection by security software by signing<sup>1</sup> their ransomware with an Authenticode certificate, which anyone can buy (or steal). Signed programs are supposed to offer assurance that the code has not been modified since the software company released it, but it offers no assurance that the software should even be running in the first place. Unfortunately, some security tools conflate "digitally signed" with "should be allowed to run." When ransomware is properly code-signed, anti-malware or anti-ransomware defenses might not analyze the ransomware as rigorously as they would other executables that lack a valid digital signature. Endpoint protection software may even choose to trust the malicious code.

<sup>1</sup> [https://en.wikipedia.org/wiki/Code\\_signing](https://en.wikipedia.org/wiki/Code_signing)

New code-signing certificates typically cost around \$50. In addition to sharing payment information, the certificate authority requires the person or organization purchasing the certificate to supply contact details. The certificate authority contacts the purchaser via email and phone to validate their existence. While this is a hurdle and risk for many malware authors, more organized criminals make the effort to ensure their malware is code-signed with a valid Authenticode certificate to prevent detection and help ensure success.

Certificate issuers act quickly to revoke a signing certificate when they're notified that the certificate is being used in the commission of a cybercrime. Once the certificate authority revokes the digital certificate, and it becomes very easy for endpoint protection software to locate and quarantine all malware signed with the revoked certificate.

### **Privilege escalation (and lateral movement)**

While it is good practice to give user accounts – and therefore the applications they run – limited access rights, in today's threat landscape that doesn't help much. Even if the logged-in user has standard limited privileges and permissions, today's ransomware uses exploits to elevate their own privileges and abuse stolen administrator credentials to make sure the attack is performed using a privileged account. Some examples:

- ▶ EternalBlue is an exploit<sup>2</sup> developed by the U.S. National Security Agency (NSA). It was leaked and, later, used as part of the worldwide WannaCry ransomware attack in 2017. In conjunction with the DoublePulsar code injection technique, the exploit allows the installation of malware with the highest privileges on an endpoint, regardless of the privileges of the logged in user.
- ▶ To suppress a User Access Control (UAC) prompt that normally occurs during privilege elevation, some ransomware employs a UAC bypass exploit<sup>3</sup> that sets the path to the ransomware in a specific registry key. When this is set, running the Windows Event Viewer process (eventvwr.msc, a Microsoft Saved Console file) will inadvertently launch the ransomware (for example, Dharma and BitPaymer) with elevated privileges, regardless of the privileges of the logged in user. This exploit works for every version of Windows until Windows 10 Creators Update (April 2017).
- ▶ CVE-2018-8453 is a Win32k Elevation of Privilege (EoP) use-after-free vulnerability. Malware that successfully exploits this vulnerability can run arbitrary code in kernel mode. For example, the malware can install programs; view, change, or delete data; or create new accounts with full user rights, regardless of the privileges of the logged in user. The Sodinokibi ransomware, for example, exploits this vulnerability to elevate its privileges.

Once the attackers compromise a server or endpoint, many active adversaries abuse existing Windows tools, as well as open-source security or penetration testing tools. For instance, they might use TASKKILL.EXE to terminate processes belonging to endpoint protection software. They may even introduce a tool like Process Hacker to interactively make the machine their own. They'll install a remote access tool (RAT) like CobaltStrike, Meterpreter, or PowerShell Empire for flexibility and to maintain persistence on their foothold machine until the mission is complete.

<sup>2</sup> <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010>

<sup>3</sup> <https://enigma0x3.net/2016/08/15/fileless-uac-bypass-using-eventvwr-exe-and-registry-hijacking/>

Once the machine is owned, many attackers attempt to harvest a local domain administrator's credentials using a post-exploitation tool like Mimikatz. The attackers may add a new domain administrator account to the Active Directory (AD), just for them to use, in case the real domain admins change passwords.

After the keys to the kingdom have been obtained, it's time to use them. Some active adversaries use a tool called BloodHound<sup>4</sup> to map the Active Directory domain and determine where the metaphorical crown jewels – servers or other high-value targets—are stored. Many attackers spend the time interactively looking for file servers and those used for data backups, as a corrupted or damaged backup leaves the victim more likely to agree to pay the ransom.

We've observed ransomware threat actors take over a server via the Remote Desktop Protocol (RDP), and destroy the backups via ransom encryption, or sometimes just by deleting them normally. Lastly, they will distribute ransomware to peer endpoints and file servers using those same domain admin credentials and the Windows software management utility WMI.

To automatically distribute ransomware to peer endpoints and servers, adversaries may leverage a trusted dual-use utility like PsExec from Microsoft SysInternals. The attacker crafts a script that lists the collected targeted machines and incorporates them together with PsExec, a privileged domain account, and the ransomware. This script successively copies and executes the ransomware onto peer machines. This takes less than an hour to complete, depending on the number of machines targeted. By the time the victim spots what's going on it is too late, as these attacks typically happen in the middle of the night when the IT staff is sleeping.

As an alternative to PsExec, active adversaries have also been seen leveraging a logon and logoff script via a Group Policy Object (GPO), or abusing the Windows Management Interface (WMI) to mass-distribute ransomware inside the network.

Attackers have been observed leveraging stolen credentials for, or exploiting vulnerabilities in, remote monitoring and management (RMM) solutions like Kaseya<sup>5</sup>, ScreenConnect<sup>6</sup>, and Bomgar<sup>7</sup>. These RMM solutions are typically used by a managed service provider (MSP) that remotely manages its customers' IT infrastructure and/or end-user systems. RMM solutions typically run with high privileges and, once breached, offer a remote attacker "hands on keyboard" access, resulting in unwanted data hostage situations. With such access, attackers can easily distribute ransomware into networks remotely, potentially hitting multiple MSP customers at once.

It is important to enable multi-factor authentication (MFA) on central management tools and leave Tamper Protection on endpoint protection software enabled. Active adversaries will attempt to disable local protection services via tools like Process Hacker, but also try to log in into central security portals to disable protection across the network.

<sup>4</sup> <https://github.com/BloodHoundAD/BloodHound>

<sup>5</sup> <https://www.darkreading.com/attacks-breaches/customers-of-3-msps-hit-in-ransomware-attacks/d/d-id/1335025>

<sup>6</sup> <https://www.bleepingcomputer.com/news/security/sodinokibi-ransomware-spreads-wide-via-hacked-msps-sites-and-spam/>

<sup>7</sup> <https://www.darkreading.com/risk/how-ransomware-criminals-turn-friends-into-enemies/a/d-id/1335778>

### **Network first**

To ensure victims pay the ransom money, ransomware will try to encrypt as many documents as possible, sometimes even risking, or purposely crippling, the endpoint. These documents can be stored on local fixed and removable drives, as well as on mapped remote shared drives. The ransomware might even prioritize certain drives or document sizes first to ensure success before being caught by endpoint protection software or noticed by victims. For example, ransomware may be programmed to encrypt several documents at the same time via multiple threads, prioritize smaller documents, or even attack documents on mapped remote shared drives first.

In networked environments, endpoints – in both local and remote offices – are typically connected to a file server park. Business documents, including drawings and other files, are stored on one or more central file servers, which are accessible from client endpoints via one or more shared folders. Usually, an endpoint could have several drive mappings to different file server shares, containing documents from individual offices, departments, teams, and projects to segregate the data.

Ransomware causes the most immediate damage to an organization when it encrypts these mapped network drives first, as it immediately affects most employees no matter where they are geographically located. When employees cannot do their work, it disrupts the entire organization, putting pressure on management to pay the ransom demand. And even though most businesses do create backups of their data, most backups are made periodically and are not always up to date. Further, restoring multiple servers from backup can take many days, depending on the size of the data and number of affected servers. The financial impact of such delays can mount up quickly.

It is important to mention that the file servers themselves are often not infected with the ransomware. The threat typically runs on one or more compromised endpoints, abusing a privileged user account with administrator-level permissions to remotely attack the documents. So even if the file server is protected by antivirus software, the threat itself is not actually running on the server.

To make matters worse, many endpoint protection solutions do not inspect document changes on remote shared storage drives, even when they offer anti-ransomware. Poor implementation and partial drive coverage – out of performance concerns or technical debt – are the main reasons why most security products and even anti-ransomware solutions fail to detect a ransomware attack.

### **Multi-threaded**

Computers now have one or more multi-core CPUs with Simultaneous Multithreading (SMT) or Hyper-Threading (HT) technology. Such advances in microprocessor hardware offer huge performance benefits for day-to-day business operations, as they allow parallel execution and better system utilization to speed up productivity.

Some ransomware is specifically designed to make efficient use of modern CPU hardware and parallelizes individual tasks to ensure faster and, subsequently, more harmful impact before victims discover they're under attack. These attacks can achieve higher throughput and lower latency since data in a faster medium (such as memory) can be retrieved by one thread while another thread retrieves data from a slower medium (such as storage), with neither thread waiting for the other to finish.

For example, the Sodinokibi ransomware has storage access (reading original document, writing encrypted document), key-blob embedding, and document renaming on multiple individual threads. And the LockerGoga and MegaCortex ransomware launch sub-processes for every few documents, to both accelerate and make it harder to detect and stop their attacks.

### File encryption

From a file system activity view, based on how it encrypts documents, ransomware can be divided into two groups: overwrite and copy.

OVERWRITE (IN-PLACE)	COPY
Encrypted document is stored on same disk sectors as original document. Steps:	Encrypted document is stored on free available disk sectors. Steps:
<ol style="list-style-type: none"> <li>1. Read original document; opened for <b>Read/Write</b></li> <li>2. Write encrypted version over original document</li> <li>3. Rename document</li> </ol>	<ol style="list-style-type: none"> <li>1. Read original document; opened for <b>Read</b> only</li> <li>2. Write (create) encrypted copy (with different file name or file extension)</li> <li>3. Delete original document</li> </ol>
Remarks	
<ul style="list-style-type: none"> <li>▸ Impossible to recover original documents with data recovery tools</li> <li>▸ Some ransomware, like LockerGoga, rename the original document before encryption</li> </ul>	<ul style="list-style-type: none"> <li>▸ The file name of the encrypted copy is like the original document</li> <li>▸ Without additional “wipe” actions afterwards, it is possible to recover some original documents with data recovery tools</li> <li>▸ Some ransomware, like WannaCry, may delete the original documents via another application or process</li> </ul>

### Rename

Ransomware typically renames the documents at the time of encryption. Precisely when the rename process takes place (it can happen before or after the ransomware encrypts the file) usually remains consistent within a ransomware family. And some ransomware (notably RobbinHood) even changes the entire file name of the document. Either way, the act of renaming the files helps the attacker in several ways:

1. Makes the harm more visible to victims, including both administrators and users, as the document file type icon changes in Explorer and applications.
2. Prevents double encryption of documents should the ransomware run again on the same machine. If ransomware were able to encrypt a document twice, a decryption tool may not be able to recover the original file.
3. Prevents other ransomware from encrypting the file, as ransomware typically encrypts documents with a certain file name extension only. This would certainly complicate decryption, even if the victim paid both attackers, as the victim would need to know which ransomware ran first.
4. Breaks the filetype relationship to the file's parent application, and also prevents the user from recovering their files from earlier versions in the Windows Volume Shadow Copy Service.



### 5. Complicates the salvage of deleted documents using special recovery software.

Recovery software typically scans the Master File Table (MFT) for deleted files that have a certain file name extension. In situations where the ransomware has encrypted a copy of the document instead of performing in-place encryption, the recovery software needs to do a full disk surface analysis instead, which takes a lot more time. But since a filesystem will reuse sectors previously occupied by now-deleted files, recovery results may be limited.

### 6. Some ransomware changes the file name extension to an email address of the attacker so victims can clearly see whom to contact for the ransom payment and the decryption tool – although there is no guarantee all encrypted documents can be correctly decrypted to their original state. Many ransomware attackers make poor software developers.

### **Key blob**

In order to recover the encrypted documents, ransomware stores a key that a decryption tool with the private key can use to revert the damage. This blob of data is prepended, appended, or stored in one or more separate files, depending on the ransomware family.

### **Wallpaper**

To ensure victims immediately understand what has happened and urge them to pay the ransom demand, some ransomware like WannaCry and Sodinokibi replace the Windows desktop wallpaper with a message that confronts the victim.

### **Vssadmin**

Microsoft Windows offers the ability to perform recovery “rollbacks.” It accomplishes this with the Volume Shadow Service (VSS). Windows maintains prior version copies of documents (and system files) that have changed – not just a single previous version, but potentially many previous versions. These previous version files are easily recoverable even if the current version has been successfully wiped. The VSS requires the volumes it shields to be formatted with the NTFS file system.

Typically, ransomware will rename your documents during its attack, breaking the relationship with a previous version file stored by the VSS. Theoretically, you could rename an encrypted document back to its original file name – thus restoring the relationship – and then restore the original from VSS to undo the effects of a ransomware attack. Unfortunately, attackers routinely delete the volume shadow copies during the attack, via a Windows utility called VSSADMIN.EXE. This utility provides a command-line interface with the volume shadow copy service and requires elevated administrator privileges. Attackers typically steal or already possess privileged credentials or use an exploit to elevate privileges.

An alternative method to delete the volume shadow copies is via Windows Management Instrumentation (WMI). Windows includes a command-line utility called WMIC.EXE to access WMI, which is used by many applications as a scripting interface. Simply blocking or using Group Policy to restrict WMIC.EXE will likely break some day-to-day tasks.

### BCDEdit

Along with removing backups of documents and other files stored by the Windows Volume Shadow Service, ransomware can also try to prevent victims and Windows from embarking on a recovery procedure to repair the computer. To achieve this, it abuses the Windows command BCDEDIT.EXE, which allows manipulation of the Windows Boot Configuration Data (BCD). Such ransomware typically set the following options:

1. **recoveryenabled No** – Disable the Windows diagnostic and repair feature, so it no longer runs automatically after a third unsuccessful boot of your computer.
2. **bootstatuspolicy ignoreAllFailures** – Ignore errors if there is a failed boot, failed shutdown, or failed checkpoint. The computer will attempt to boot normally after an error occurs.

### Cipher

Ransomware like LockerGoga and MegaCortex abuse the CIPHER.EXE command-line tool from Microsoft<sup>8</sup> to make sure ransomware victims cannot recover deleted documents from their storage drives. This tool has been part of Windows since Windows 2000 and is intended to manage legitimately encrypted data using the Encrypting File System (EFS).

CIPHER.EXE also provides the ability to permanently overwrite (or "wipe") all of the deleted data on a storage drive – as abused by some ransomware. The feature is meant to improve security by ensuring that even an attacker who gained complete physical control of a Windows computer would be unable to recover previously deleted data. In the hands of ransomware attackers, CIPHER.EXE adds to an already big problem.

### 0 allocation

Instead of performing an in-place encryption, some ransomware first creates an encrypted copy of the document it attacks and then deletes the original file. Compared to in-place encryption, in this case the encrypted copies are stored elsewhere on the storage drive. A data recovery tool could lift the original files as long as the marked-free sectors of the deleted files have not yet been overwritten. To frustrate this recovery avenue, the Dharma ransomware sets the file size of each of the attacked documents to 0 bytes before deletion.

Data recovery tools like *Recuva* can quickly list deleted files and their cluster allocations and can lift (copy) these files from the affected storage drive. It can do so thanks to the Master File Table (MFT), which often still holds records of the deleted documents, including the physical location of these files on the storage drive. The records of the deleted documents have the deleted bit set and a recovery tool can focus on this bit to help victims to get some data back. But setting the file size to 0 after deletion will update the record of the file in the Master File Table (MFT), making it more difficult for data recovery tools to recover deleted documents.

<sup>8</sup> <https://support.microsoft.com/en-us/help/298009/cipher-exe-security-tool-for-the-encrypting-file-system>

### Flush buffers

Windows improves system performance by using write-caching on fixed, removable, and network mapped storage devices. But ransomware victims who are in a panic and physically power off their machine might cause data loss or data corruption (which is still better than allowing all your documents to be encrypted and held for ransom). This is also the reason why, before Windows 10 v1809<sup>9</sup>, you had to follow the Safely Remove Hardware procedure for e.g. portable USB storage devices that hold user data and files, such as external hard drives and thumb drives.

File servers are typically set up with RAID<sup>10</sup> storage, often with a battery-backed write cache. On these machines, write-cache buffer flushing is usually disabled to further optimize performance, as the battery-backed write cache allows the machine to flush its buffer in case of power failure without risking data loss or corruption.

At the cost of some performance, some ransomware (like WannaCry, GandCrab and BitPaymer) make certain that the written data of the encrypted documents are immediately persisted to the storage drive. They do so by either calling the FlushFileBuffers function or using Write Through.

### Encryption by proxy

Some ransomware – like GandCrab and Sodinokibi – abuse Windows PowerShell to hoist in a PowerShell script from the internet, which is set to automatically start the ransomware after several days, making the attack appear to come out of nowhere. In this scenario, the actual file encryption attack itself is performed by the trusted Windows POWERSHELL.EXE process, making endpoint protection software believe a trusted application is modifying the documents. To achieve the same goal, ransomware like Ryuk may inject its malicious code into a trusted running process like SVCHOST.EXE. And the MegaCortex ransomware uses the Windows RUNDLL32.EXE application to encrypt documents from a trusted process.

Ransomware like BitPaymer may run from a NTFS Alternate Data Stream (ADS) in an attempt to hide from both victim users and endpoint protection software.

<sup>9</sup> <https://support.microsoft.com/en-us/help/4495263/windows-10-1809-change-in-default-removal-policy-for-external-media>

<sup>10</sup> <https://en.wikipedia.org/wiki/RAID>






## Overview

	WANNACRY	GANDCRAB	SAMSAM	DHARMA	BITPAYMER
<b>Type</b>	Cryptoworm	Ransomware-as-a-Service (Raas)	Automated Active Adversary	Automated Active Adversary	Automated Active Adversary
<b>Code-signed</b>	-	-	-	-	-
<b>Privilege escalation</b>	Exploit	Credentials	Credentials	Credentials	Exploit
<b>Network first</b>	-	-	-	Yes	Yes
<b>Multi-threaded</b>	-	-	-	Yes	-
<b>File encryption</b>	Copy, In-place	In-place	Copy	Copy	In-place
<b>Rename</b>	After	After	After	After	After
<b>Key blob</b>	Header	End of file	Header	End of file	Ransom note
<b>Wallpaper</b>	Yes	Yes	-	-	-
<b>Vssadmin</b>	After	After	Before	Before, After	Before
<b>BCDEdit</b>	After	-	-	-	-
<b>Cipher</b>	-	-	-	-	-
<b>0 allocation</b>	-	-	-	Yes	-
<b>Flush buffers</b>	Yes	Write Through	-	-	Yes
<b>Encryption by proxy</b>	-	Yes	-	-	-

	RYUK	LOCKERGOGA	MEGACORTEX	ROBBINHOOD	SODINOKIBI
<b>Type</b>	Automated Active Adversary	Automated Active Adversary	Automated Active Adversary	Automated Active Adversary	Ransomware-as-a-Service (Raas)
<b>Code-signed</b>	-	Yes	Yes	-	-
<b>Privilege escalation</b>	Credentials	Credentials	Credentials	Credentials	Exploit
<b>Network first</b>	-	-	-	-	-
<b>Multi-threaded</b>	Yes	-	-	-	Yes
<b>File encryption</b>	In-place	In-place	In-place	Copy	In-place
<b>Rename</b>	After	Before	Before	After	After
<b>Key blob</b>	End of file	End of file	Separate file	New	End of file
<b>Wallpaper</b>	-	-	-	-	Yes
<b>Vssadmin</b>	After	-	After	Before	Before
<b>BCDEdit</b>	-	-	-	Before	After
<b>Cipher</b>	-	After	After	-	-
<b>0 allocation</b>	-	-	-	-	-
<b>Flush buffers</b>	-	-	-	-	-
<b>Encryption by proxy</b>	Yes	-	Yes	-	-

### Color coding

In the following chapters, the term 'documents' also covers other productivity file types like spreadsheets, drawings, images, and photos.

FILE SYSTEM ACTIVITY COLOR CODING CLARIFICATION:	
	Query file information or activity, not directly associated with the encryption attack itself
	Open a file for read only
	Create a new file on free available disk sectors
	Open an existing file for reading and writing to make file changes
	Open file for delete or rename

### WannaCry

WannaCry is a crypto-ransomware network worm (or cryptoworm) which targets computers running the Microsoft Windows 7 operating system, encrypting documents and demanding ransom payments in the bitcoin cryptocurrency. Between May 12 and 15, 2017, an outbreak<sup>11</sup> of this crypto-ransomware took place, infecting more than 230,000 computers in 150 countries.

WannaCry is considered a network worm because it also includes its own transport. This transport code scans for vulnerable systems, then uses the EternalBlue<sup>12</sup> exploit to gain access, and the DoublePulsar<sup>13</sup> code injection method to install and execute a copy of itself.

More than two years on, modified WannaCry variants still cause headaches for IT admins and security analysts: <https://news.sophos.com/en-us/2019/09/18/the-wannacry-hangover/>

### Characteristics

- Sample [SHA-256]  
ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA
- Single-threaded, i.e. it encrypts one document at a time
- Configures the Discretionary Access Control List (DACL) on each document to give group Everyone full access permissions
- First creates encrypted copies of all documents, thus increasing [doubling] disk usage
- Encrypts original document in place and in chunks of 256 KB (262,144 bytes) and moves it to the %temp% folder with a new file name and file extension
- A separate application TASKDL.EXE deletes the scrambled originals in the %temp% folder, after all documents are encrypted
- Deletes volume shadow copies via VSSADMIN.EXE, after the documents are encrypted
- Deletes the backup catalog on the local computer via WBADMIN.EXE, after the documents are encrypted
- Changes the desktop wallpaper

<sup>11</sup> [https://en.wikipedia.org/wiki/WannaCry\\_ransomware\\_attack](https://en.wikipedia.org/wiki/WannaCry_ransomware_attack)

<sup>12</sup> <https://en.wikipedia.org/wiki/EternalBlue>

<sup>13</sup> <https://en.wikipedia.org/wiki/DoublePulsar>

## WannaCry file system activity

STEP	OPERATION	PURPOSE
1	SetSecurityFile	Modify discretionary access control list (DACL) of original document to Full for group Everyone, via the Windows application ICACLS.EXE.
2	CreateFile	Check if encrypted document with 'WNCRY' file extension exists.
3	CreateFile (Generic Read)	Open original document for read only.
4	QueryBasicInformationFile	Record timestamps on original document.
5	ReadFile	Read first 8 bytes of original document.
6	CreateFile (Generic Write)	Create encrypted file with 'WNCRYT' file extension, for write only.
7	WriteFile	Write 'WANACRY!' string (8 bytes) in encrypted file.
8	WriteFile	Write 4 bytes, at offset 8 bytes, in encrypted file.
9	WriteFile	Write 256 bytes, at offset 12 bytes, in encrypted file.
10	WriteFile	Write 4 bytes, at offset 268 bytes, in encrypted file.
11	WriteFile	Write 8 bytes, at offset 272 bytes, in encrypted file.
12	ReadFile	Read original document, entirely (0 bytes to EndOfFile).
13	WriteFile	Write encrypted file, entirely, at offset 280 bytes.
14	SetBasicInformationFile	Give encrypted file same timestamps as original document.
15	CloseFile	Close original document.
16	CloseFile	Close encrypted file.
17	SetRenameInformationFile	Change file extension of encrypted file from 'WNCRYT' to 'WNCRY'.
18	CreateFile (Generic Write)	Open original document for write only.
20	WriteFile	Write 1,024 bytes (1 KB) in original document. <b>At offset EndOfFile -1,024 bytes.</b>
21	FlushBuffersFile	Commit all buffered data to be written to disk.
21	WriteFile (Non-cached)	Write 4,096 bytes (4 KB) in original document, at offset AllocationSize on disk -4,096 bytes.
22	WriteFile	Write in chunks of 262,144 bytes (256 KB) in original document.
23	CloseFile	Close original document, now encrypted file.
24	OpenFile (Read Attributes)	Open encrypted file.
25	SetRenameInformationFile	Rename file to %temp%\<num>.WNCRYT. ReplaceIfExists: True.
26	CloseFile	Close encrypted file.
#	SetDispositionInformationFile	Once all documents on the disk are encrypted, a separate application TASKDL.EXE is run to delete %temp%\*.WNCRYT (i.e. all 'WNCRYT' files).

### Matrix

The ransomware we call Matrix highlights another growing trend within the cybercriminal community: to engage in active, targeted attacks against victim networks with the goal of delivering malware inside the victim's network. This threat vector has been gaining prominence since the widely publicized SamSam ransomware began to capitalize on it. The malware is delivered, in most cases, by the attackers performing an active brute-force attack against the passwords for Windows machines accessible through a firewall that has the remote desktop protocol (RDP) enabled.

The malware executable bundles within itself several payload executables it needs to accomplish its tasks. Once it has gained a foothold inside the network, Matrix uses the RDP within the networks it has infected. Among the embedded components are some free, legitimate system administrator tools the malware uses to achieve some of its goals.

A comprehensive but more traditional malware research report on this ransomware is available here: <https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-matrix-report.pdf>

### Characteristics

- Sample [SHA-256]  
2A12EEB58AC0A2A3E9CD1DDBBF1752086EE19387CAA0E1232EAA13CBFED2C80A
- Single-threaded, i.e. it encrypts one document at a time
- Opens documents for read/write for in-place encryption
- Encrypts 65,536 bytes at the beginning and 65,536 bytes of data at the end of the document; not entire document
- Renames the document by adding an identifier to the file type extension
- Deletes volume shadow copies via WMIC.EXE and VSSADMIN.EXE, before the documents are encrypted
- Changes the desktop wallpaper

### Matrix file system activity

STEP	OPERATION	PURPOSE
1	QueryStandardInformationFile	Determine file size of original document on disk.
2	QueryStandardInformationFile	Determine file size of original document on disk.
3	QueryStandardInformationFile	Determine file size of original document on disk.
4	QueryStandardInformationFile	Determine file size of original document on disk.
5	QueryStandardInformationFile	Determine file size of original document on disk.
6	QueryStandardInformationFile	Determine file size of original document on disk.
7	QueryStandardInformationFile	Determine file size of original document on disk.
8	QueryStandardInformationFile	Determine file size of original document on disk.
9	CreateFile (Read/Write)	Open original document for read and write.
10	QueryStandardInformationFile	Determine file size of original document on disk.
11	ReadFile	Read 61,440 bytes (60 KB) from original document, at offset 0 bytes.
12	ReadFile	Read 4,096 bytes (4 KB) from original document, at offset 61,440 bytes (60 KB).
13	QueryStandardInformationFile	Determine file size of original document on disk.
14	ReadFile	Read 61,440 bytes (60 KB) from original document, at offset EndOfFile -65,536 bytes (64 KB).
15	ReadFile	Read 4,096 bytes (4 KB) from original document, at offset EndOfFile -4,096 bytes (4 KB).
16	QueryStandardInformationFile	Determine file size of original document on disk.
17	WriteFile	Write 61,440 encrypted bytes (60 KB), in file, at offset 0 bytes.
18	WriteFile	Write 4,096 (4 KB) encrypted bytes in file, at offset 61,440 bytes (60 KB).
19	QueryStandardInformationFile	Determine file size of – now partially encrypted – file on disk.
20	WriteFile	Write 61,440 encrypted bytes (60 KB) in file. At offset EndOfFile -65,536 bytes (64 KB).
21	WriteFile	Write 4,096 encrypted bytes (4 KB) in file. At offset EndOfFile -4,096 bytes (4 KB).
22	QueryStandardInformationFile	Determine file size of encrypted file on disk.
23	WriteFile	Write 1,384 bytes, at offset EndOfFile, in encrypted file.
24	CloseFile	Close original document, now encrypted file.
25	CreateFile (Read Attributes)	Open encrypted file.
26	SetRenameInformationFile	Rename File. ReplaceIfExists: True. Example: BEM1kkKW-iLpTMM6H.[barboza40@yahoo.com]
27	CloseFile	Close encrypted file.



### GandCrab

In the first half of 2019, GandCrab was the most popular ransomware used in large scale, untargeted attacks that use malicious websites or email attachments to infect as many victims as possible. Its creators peddled it to anyone who wanted to use it using the Ransomware-as-a-Service (RaaS) model, which netted them a percentage of each ransom it extorted. GandCrab operators choose the ransom they want to demand, typically somewhere between a few hundred to a few thousand dollars per computer.

In the last couple of years, a new template for ransomware attacks has emerged. Some criminals are turning away from “fire and forget” distribution in favor of highly focused, guided attacks, generally referred to as targeted attacks. The distribution is carried out by an adversary, who uses tools to automatically scan the internet for IT systems with weak protection. Although victims may believe they have been targeted, the attack is usually opportunistic, thanks to the vulnerability scanners used.

Ransomware attacks carried out by automated active adversaries have typically been associated with BitPaymer, SamSam, or Ryuk. RaaS offerings like GandCrab allow crooks with less technical ability to also get in on the act without needing to create their own ransomware, command and control infrastructure, or payment handling.

In February 2019 we revealed how automated active adversaries delivered a GandCrab ransomware by hand to attack a hospital: <https://nakedsecurity.sophos.com/2019/02/14/inside-a-gandcrab-targeted-ransomware-attack-on-a-hospital/>

### Characteristics

- Sample (SHA-256)  
6FBA19BF0CC1BB764E063C1DE51CAF0CF0A6CC90FA76B592BCDE28CEEE161BDC
- Single-threaded, i.e. it encrypts one document at a time
- Opens documents for read/write for in-place encryption
- Uses Write Through to ensure the write is persisted to disk without potential caching delays
- Deletes volume shadow copies via VSSADMIN.EXE after documents are encrypted
- Changes the desktop wallpaper

### GandCrab file system activity

STEP	OPERATION	PURPOSE
1	CreateFile (Read/Write)	Open original document for read and write.
2	QueryStandardInformationFile	Determine file size of original document on disk.
3	ReadFile	Read 540 bytes from original document. At offset EndOfFile -540 bytes.
4	ReadFile	Read original document, entirely (0 bytes to EndOfFile).
5	WriteFile (Write Through)	Overwrite original document with encrypted version, entirely. Length: EndOfFile.
6	WriteFile (Non-cached)	Overwrite original document with encrypted version. Length: AllocationSize.
7	QueryStandardInformationFile	Determine file size of now encrypted document on disk.
8	WriteFile	Write 540 bytes to encrypted file, at offset EndOfFile.
9	WriteFile	Write 4,096 (4 KB) to encrypted file. At offset AllocationSize -4,096 bytes (4 KB).
10	CloseFile	Close original document, now encrypted file.
11	CreateFile (Read Attributes)	Open encrypted file.
12	SetRenameInformationFile	Rename encrypted file (add file extension, e.g. 'gxjei'). ReplaceIfExists: True.
13	CloseFile	Close document.

### SamSam

SamSam malware first appeared in December 2015, and since then more than \$6 million has been deposited into SamSam bitcoin wallets. The pattern of attacks and the evolution of the SamSam malware suggests that it's the work of a small group at most. Attacks are infrequent (around one a day) compared to other kinds of ransomware, but they are devastating.

After breaking in via the Remote Desktop Protocol (RDP), the attacker attempts to escalate their privileges to the level of Domain Admin so that they can deploy SamSam malware across an entire network, just like a sysadmin deploying regular software. The attacker seems to wait until victims are likely to be asleep before unleashing the malware on every infected machine simultaneously, giving the victim little time to react.

A ransom note demands payment of about \$50,000 in bitcoins and directs the victims to a dark web site (a hidden service on The Onion Router (TOR) network).

The alleged attackers behind the SamSam ransomware, who operated from Iran, have been identified and are wanted by the FBI<sup>13</sup>, the federal law enforcement agency of the United States. Since the suspects were indicted<sup>14</sup> in December 2018, the SamSam ransomware has not been seen again. Before then, more than 230 entities were infected, \$6 million in ransom payments was extorted, and an estimated \$30 billion in damages affected private and public institutions, including hospitals and schools.

An in-depth report on the SamSam ransomware is available here: <https://news.sophos.com/en-us/2018/07/31/sophoslabs-releases-samsam-ransomware-report/>

### Characteristics

- ▶ Sample (SHA-256)  
8C0425ECA81E1EEAF8043764EB38A2BC103598163D3307E583F4E5AD7EB0E708
- ▶ Single-threaded, i.e. it encrypts one document at a time
- ▶ Uses a 3,072 bytes header with an XML structure, i.e. based on ASCII-only characters, that can skew analysis of the encrypted binary file
- ▶ Creates an encrypted copy of the original document on free available disk sectors. Theoretically, if not overwritten by other data, the original document can be recovered from disk
- ▶ Deletes volume shadow copies via VSSADMIN.EXE, before the documents are encrypted

<sup>13</sup> <https://www.justice.gov/opa/press-release/file/1114746/download>

<sup>14</sup> <https://www.fbi.gov/news/stories/iranian-ransomware-suspects-indicted-112818>

### SamSam file system activity

STEP	OPERATION	PURPOSE
1	CreateFile (Generic Write)	Create encrypted file based on original document file name with added file extension, e.g. 'weapologize.'
2	WriteFile	Write 3,072 bytes in encrypted file, at offset 0 bytes.
3	CloseFile	Close encrypted file.
4	CreateFile (Generic Read)	Open original document for read only.
5	ReadFile	Read original document, entirely (0 bytes to EndOfFile).
6	CloseFile	Close original document.
7	CreateFile (Generic Write)	Open encrypted file for write only.
8	WriteFile	Write encrypted version of original document, entirely, at offset 3,072 bytes.
9	CloseFile	Close encrypted file.
10	CreateFile (Generic Write)	Open encrypted file.
11	WriteFile	Write 2,492 bytes in encrypted file, at offset 0 bytes.
12	CloseFile	Close encrypted file.
13	CreateFile (Read Attributes)	Open original document.
14	SetDispositionInformationFile	Delete: True.
15	CloseFile	Close original document, committing delete.

## Dharma

### Characteristics

- Sample [SHA-256]  
B8D32ED92E3227836054ED6BB4E53AD2E0ABE4617F1215D5E81162F9F5513EC2
- Dharma is also known as CrySIS
- Deletes volume shadow copies via VSSADMIN.EXE, both before and after the documents are encrypted
- Multi-threaded, i.e. it encrypts multiple documents at a time
- Opens original document for read/write but doesn't change contents. Instead, it sets the file size of original document to 0 bytes before it is deleted
- Creates an encrypted copy of the original document on free available disk sectors; theoretically, if not overwritten by other data, the original document would be recoverable from disk. However, this is complicated as the file size of the document is set to 0 bytes before it is deleted
- Setting the file size of the original document to 0 bytes may hinder behavior-based detection in anti-ransomware technology

### Dharma file system activity

STEP	OPERATION	PURPOSE
1	CreateFile (Generic Read/Write)	Open original document for reading and writing.
3	CreateFile (Generic Write)	Create encrypted document based on original document file name with added file extension. Example: Desert.jpg.id-<VolumeSerial>.[veracrypt@foxmail.com].adobe
4	ReadFile	Read original document, entirely.
5	WriteFile	Write encrypted file, entirely, + 11 bytes at EndOfFile.
6	ReadFile	Read original document, at offset EndOfFile, 1,048,560 bytes.
7	WriteFile	Write 212 bytes + (length original file name * 2 bytes), in encrypted file, at offset size of original document.
8	SetEndOfFileInformationFile	Set EndOfFile of encrypted file to size of original document + added bytes.
9	SetAllocationInformationFile	Set AllocationSize of encrypted file to EndOfFile of encrypted file.
10	CloseFile	Close encrypted file.
10	SetEndOfFileInformationFile	Set EndOfFile of original document to 0 bytes.
11	SetAllocationInformationFile	Set AllocationSize of original document to 0 bytes.
12	CloseFile	Close original document.
13	CreateFile (Read Attributes)	Open original document.
14	SetDispositionInformationFile	Delete: True.
15	CloseFile	Close original document, committing delete.

## BitPaymer

### Characteristics

- Sample [SHA-256]  
655C44BEBB2A642E665316236A082C94F88A028721C19BD28B5F25E1C40A13B8
- Deletes volume shadow copies via VSSADMIN.EXE, before the documents are encrypted
- Single-threaded, i.e. it encrypts one document at a time
- Known to abuse an alternate data stream (ADS), a feature of the NTFS file system that allows the ransomware to hide itself from plain sight and evade security tools that are not able to look into an ADS
- Employs FlushBuffersFile to ensure buffered data is immediately committed to the disk
- Renames document after encryption
- For each encrypted document, BitPaymer creates a ransom note text file that also contains the key blob in base64 for decryption

### BitPaymer file system activity

STEP	OPERATION	PURPOSE
1	CreateFile [Generic Read/Write]	Open document for reading and writing.
2	ReadFile	Read data from document.
3	WriteFile	Write encrypted data into document.
4	FlushBuffersFile	Commit changes to disk.
5	CloseFile	Close now encrypted document.
6	CreateFile [Read Attributes]	Open encrypted document.
7	SetRenameInformationFile	Rename encrypted document: add '<identifier>' file extension.
8	CloseFile	Close encrypted document.
9	CreateFile [Generic Read/Write]	Create ransom note text file '<document filename>_readme'.
10	WriteFile	Write data, including key blob, into ransom note.
11	CloseFile	Close ransom note text file.

## Ryuk

### Characteristics

- Sample (SHA-256)  
830F83578F3A5593B103EA4A682788DC376E96247CD790417F2630884D686E9F
- Based on Hermes ransomware
- Multi-threaded, i.e. it encrypts multiple documents simultaneously
- Encrypts data on mapped network drives
- Adds the key blob at the end of the encrypted document

### Ryuk file system activity

STEP	OPERATION	PURPOSE
1	CreateFile (Generic Read/Write)	Open document for reading and writing.
2	ReadFile	Read data from document.
3	WriteFile	Write encrypted data into document.
4	CloseFile	Close now encrypted document.
5	CreateFile (Read Attributes)	Open encrypted document.
6	SetRenameInformationFile	Rename encrypted document: add '.RYK' file extension.
7	CloseFile	Close encrypted document.

## LockerGoga

### Characteristics

- Sample [SHA-256]:  
2CE4984A74A36DCDC380C435C9495241DB4CA7E107FC2BA50D2FE775FB6B73CE
- The sample is digitally code-signed with a valid Authenticode certificate issued to 'ALISA LTD'; this is an attempt by the malware author to minimize anti-malware detection, as executables that are signed using valid certificates may not be analyzed as rigorously as executables without signature verification
- Single-threaded, i.e. it encrypts one document at a time
- Renames document before encrypting it
- Encrypts entire documents in chunks of 64 kilobytes
- Does not encrypt mapped network drives, but LockerGoga is usually distributed across the network to other endpoints and servers via PsExec or WMI using stolen [domain] credentials
- Adds the key blob at the end of the encrypted document
- Runs the Windows standard application CIPHER.EXE with the /w switch to wipe unused disk space, after the documents are encrypted
- Changes the passwords of all user accounts

### LockerGoga file system activity

STEP	OPERATION	PURPOSE
1	CreateFile (Read Attributes)	Open document.
2	SetRenameInformationFile	Rename document: add 'locked' file extension.
3	CloseFile	Close renamed original document.
4	CreateFile (Generic Read/Write)	Open renamed document for reading and writing.
5	ReadFile	Read data from document.
6	WriteFile	Write encrypted data into document.
7	CloseFile	Close now encrypted document.

### Partial encryption

A blog<sup>15</sup> on LockerGoga by Unit 42 in March 2019 discusses partial encryption of files by a sample from January 2019. SophosLab's sample is from a previously undiscussed targeted attack that encrypts documents entirely.

<sup>15</sup> <https://unit42.paloaltonetworks.com/born-this-way-origins-of-lockergoga/>



## MegaCortex

### Characteristics

- Sample [SHA-256]  
F5D39E20D406C846041343FE8FBD30069FD50886D7D3D0CCE07C44008925D434
- The sample is digitally code-signed with a valid Authenticode certificate issued to '3AN LIMITED'; If this was an attempt by the malware author to minimize anti-malware detection, it was unsuccessful. Analysts quickly discovered that the same certificate had been used to sign executables used by the [completely unrelated] Rietspoof malware family
- MegaCortex drops a dynamic link library (DLL) module that is run via RUNDLL32.EXE, an application part of Windows that loads 32-bit DLLs; this may thwart some anti-ransomware solutions that do not monitor or are configured to ignore encryption activity by default Windows applications
- The MegaCortex sample is protected by a unique base64 password and only runs in a certain timeframe; this frustrates both threat researchers and sandbox analysis that attempt to reveal the purpose of the sample
- Renames document before encrypting it
- Does not encrypt mapped network drives, but MegaCortex is usually distributed across the network to other endpoints and servers via PsExec or WMI using stolen [domain] credentials
- For every ten documents to be encrypted, a new RUNDLL32.EXE process is spawned – one process at a time – controlled via shared memory allocated by MegaCortex. This may thwart some anti-ransomware solutions that do not track encryption of a few documents by a single process id. Also, on infected endpoints and servers with many documents this can generate a lot of activity for Endpoint Detection and Response (EDR) solutions to monitor and record
- Deletes volume shadow copies via VSSADMIN.EXE, after the documents are encrypted, to avoid recovery of earlier versions of the affected documents
- Runs the Windows standard application CIPHER.EXE with the /w switch to wipe unused disk space, after the documents are encrypted
- Stores the key blobs in one randomly named single file

### MegaCortex file system activity

STEP	OPERATION	PURPOSE
1	CreateFile (Read/Write)	Open document for reading and writing.
2	SetRenameInformationFile	Rename document: add 'aes128ctr' file extension.
3	ReadFile	Read data from document.
4	WriteFile	Write encrypted data into document.
5	CloseFile	Close now encrypted document.

## RobbinHood

### Characteristics

- Sample [SHA-256]

3BC78141FF3F742C5E942993ADFB EF39C2127F9682A303B5E786ED7F9A8D184B

### RobbinHood file system activity

STEP	OPERATION	PURPOSE
1	CreateFile (Generic Read)	Open document for read only.
2	CreateFile (Generic Write)	Create encrypted file named 'Encrypted_' + random string and add '.enc_robbinhood' file extension.
#	ReadFile	Read data from original document, in chunks of 10 KB.
#	WriteFile	Write encrypted data into document, in chunks of 10 KB.
5	WriteFile	Add key blob to encrypted document at end of file.
6	CloseFile	Close original document.
7	CloseFile	Close encrypted document.
8	CreateFile (Read Attributes)	Open original document.
9	SetDispositionInformationFile	Delete: True.
10	CloseFile	Close original document, committing delete.

## Sodinokibi

### Characteristics

- Sample [SHA-256]  
06B323E0B626DC4F051596A39F52C46B35F88EA6F85A56DE0FD76EC73C7F3851
- Appeared on the threat landscape in April 2019, also known as Sodin and REvil ransomware
- Distribution and delivery via malicious spam e-mails (malspam) and automated active adversary (targeted) attacks:
  - Malicious spam e-mails (malspam) with a compressed (zip) archive attached that holds a Word document. This document contains an obfuscated macro written in Visual Basic for Applications (VBA) which uses Windows PowerShell to download the ransomware in text form (a script with the ransomware encoded in base64) from Pastebin.com, a popular website where anyone can store text online for a set period of time; the text is deleted after a few hours making investigation afterwards more difficult. The ransomware is loaded straight into memory by PowerShell without dropping a Portable Executable (PE) file on the disk, making it more complicated for some disk-based anti-malware to scan it with e.g. virus signatures and machine learning (ML). Note: The recipient is expected to open the spam e-mail, open the attached zip archive, open the Word document that is inside, and enable the macros in Word
  - Automated active adversary (targeted) attacks that use Windows PowerShell to hoist in a PowerShell script from Pastebin.com, which is set to automatically start the ransomware after one million seconds (10.5 days), making the attack seem to come out of nowhere. No interaction with the victim is required. Initial access is via:
    - Vulnerable Oracle Weblogic servers (CVE-2019-2725)
    - Brute-forced/purchased/stolen credentials for internet-exposed Windows servers that advertise RDP
    - Hacked managed service providers (MSPs) that use Kaseya, ScreenConnect, Bomgar and other remote monitoring and management products
- Elevates privileges to SYSTEM by exploiting CVE-2018-8453, a Win32k elevation of privilege vulnerability. Elevation of privileges (EoP) is required to encrypt documents in case the victim is a standard user without administrative privileges. It is also required to delete volume shadow copies and disable Windows Startup Repair
- The ransomware is multi-threaded; document reading and writing the encrypted version to disk, key-blob embedding and document renaming are on individual threads
- Opens documents for read/write for in-place encryption to impede recovery via data recovery tools
- Key blob for decryption is stored at the end of the encrypted document
- Renames document after encrypting it

- Deletes volume shadow copies via VSSADMIN.EXE before the documents are encrypted to avoid recovery of earlier versions of the affected documents
- Disables the Windows Startup Repair tool to prevent victims from attempting to fix system errors that may have been caused by the ransomware
- Changes the desktop wallpaper on affected systems into a notice informing users that their files have been encrypted
- Ransom is typically \$2,500 - \$5,000 USD, to be paid in bitcoin. Payment via automated site on TOR, aka the dark web

### Sodinokibi file system activity

STEP	OPERATION	PURPOSE
1	CreateFile (Generic Read/Write)	Open document for reading and writing.
2	ReadFile	Read data from document.
4	WriteFile	Write encrypted data into document.
5	WriteFile	Add key blob to encrypted document at end of file.
6	CloseFile	Close now encrypted document.
7	CreateFile (Read Attributes)	Open encrypted document.
8	SetRenameInformationFile	Rename document: add '4pqrk340' file extension.
9	CloseFile	Close encrypted document.

## Indicators of Compromise (IOCs)

RANSOMWARE	SHA-256 HASH OF ANALYZED SAMPLE
WannaCry	ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA
Matrix	2A12EEB58AC0A2A3E9CD1DBBF1752086EE19387CAAA0E1232EAA13CBFED2C80A
GandCrab	6FBA19BF0CC1BB764E063C1DE51CAF0CF0A6CC90FA76B592BCDE28CEEE161BDC
SamSam	8C0425ECA81E1EEAF8043764EB38A2BC103598163D3307E583F4E5AD7EB0E708
Dharma	B8D32ED92E3227836054ED6BB4E53AD2E0ABE4617F1215D5E81162F9F5513EC2
BitPaymer	655C44BEBB2A642E665316236A082C94F88A028721C19BD28B5F25E1C40A13B8
Ryuk	830F83578F3A5593B103EA4A682788DC376E96247CD790417F2630884D686E9F
LockerGoga	2CE4984A74A36DCDC380C435C9495241DB4CA7E107FC2BA50D2FE775FB6B73CE
MegaCortex	F5D39E20D406C846041343FE8FBD30069FD50886D7D3D0CCE07C44008925D434
RobbinHood	3BC78141FF3F742C5E942993ADFB EF39C2127F9682A303B5E786ED7F9A8D184B
Sodinokibi	06B323E0B626DC4F051596A39F52C46B35F88EA6F85A56DE0FD76EC73C7F3851

To discover more content from  
the SophosLabs team visit:

[Sophos.com/Labs](https://sophos.com/Labs)

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

North American Sales  
Toll Free: 1-866-866-2802  
Email: [nasales@sophos.com](mailto:nasales@sophos.com)

Australia and New Zealand Sales  
Tel: +61 2 9409 9100  
Email: [sales@sophos.com.au](mailto:sales@sophos.com.au)

Asia Sales  
Tel: +65 62244168  
Email: [salesasia@sophos.com](mailto:salesasia@sophos.com)

© Copyright 2019. Sophos Ltd. All rights reserved.  
Registered in England and Wales No. 2096520, The Pentagon, Abingdon Science Park, Abingdon, OX14 3YP, UK  
Sophos is the registered trademark of Sophos Ltd. All other product and company names mentioned are  
trademarks or registered trademarks of their respective owners.

191104 WPEN [DD]

**SOPHOS**