

GandCrab Ransomware-as-a-Service (RaaS) analysed in depth

Archived: 2026-04-06 01:02:57 UTC

VB2019 paper: Different ways to cook a crab: GandCrab ransomware-as-a-service (RaaS) analysed in depth

Alexandre Mundo Alguacil & John Fokker

McAfee, Spain & The Netherlands

Table of contents

Abstract

This paper examines the GandCrab ransomware, the biggest Ransomware-as-a-Service (RaaS) threat seen in 2018 and the first half of 2019. Through technical analysis, we discovered several mistakes and indicators in the malware. Armed with these findings, we were able to exploit those mistakes and build a publicly available vaccine against GandCrab.

The hard-coded indicators gave us a method to link individual ransomware samples to affiliates and, by looking at hundreds of GandCrab samples at once, we gained even more interesting insights into the service model dynamics. Subsequently, to learn more about the actor behind GandCrab and its affiliates, we carried out extensive underground forum research. This multi-angled approach gave us different ways to cook a GandCrab.

Our research was fuelled by a sense that, as an industry, we must realize that we cannot stop cybercrime alone and that we should aim to do more than just malware analysis and the writing of detection rules, especially when it comes to fighting RaaS-type threats. Unfortunately, we find ourselves in a situation where most of the cybercriminals involved in ransomware can operate with a certain degree of impunity – ransomware developers are often in countries that make legal prosecution difficult, and affiliates are hard to catch and can easily move from one RaaS to another, continuing their extortion operations.

While law enforcement faces a daunting challenge to bring the individuals responsible to justice, our industry's knowledge, data and tooling should help with this task.

Introduction

The GandCrab malware made its first appearance at the end of January 2018 and it didn't take long for it to be discovered by the security community [1]. At that time, no one imagined that GandCrab would eventually grow to become the most prolific Ransomware-as-a-Service threat of 2018 and the first half of 2019. Its growth continued almost right up until it ceased to operate in mid-2019.

Looking back, we believe that its success was due to a combination of factors, from technical to partnering, marketing and servicing skills. GandCrab had a large underground forum presence and enjoyed attention from both fellow cybercriminals and security researchers. The GandCrab crew operated predominantly on one of the largest underground forums, where it acted with a sense of impunity, openly mocked the security industry, and boasted about its income.

Its growth and open communication style sparked our research, with GandCrab giving us a great opportunity to link our malware analysis (capability) with the adversary's activity (GandCrab and its affiliates) on the underground forums.

We used the Diamond Threat Model [2] as our guideline to structure our research.

We started out by closely examining the GandCrab ransomware code and all its versions. We looked for mistakes and to see if it contained any clues or special indicators. That was the basis for building the various vaccines. From an adversary point of view, we closely monitored GandCrab's activity on the underground forums and investigated other forum users who showed an interest in the ransomware.

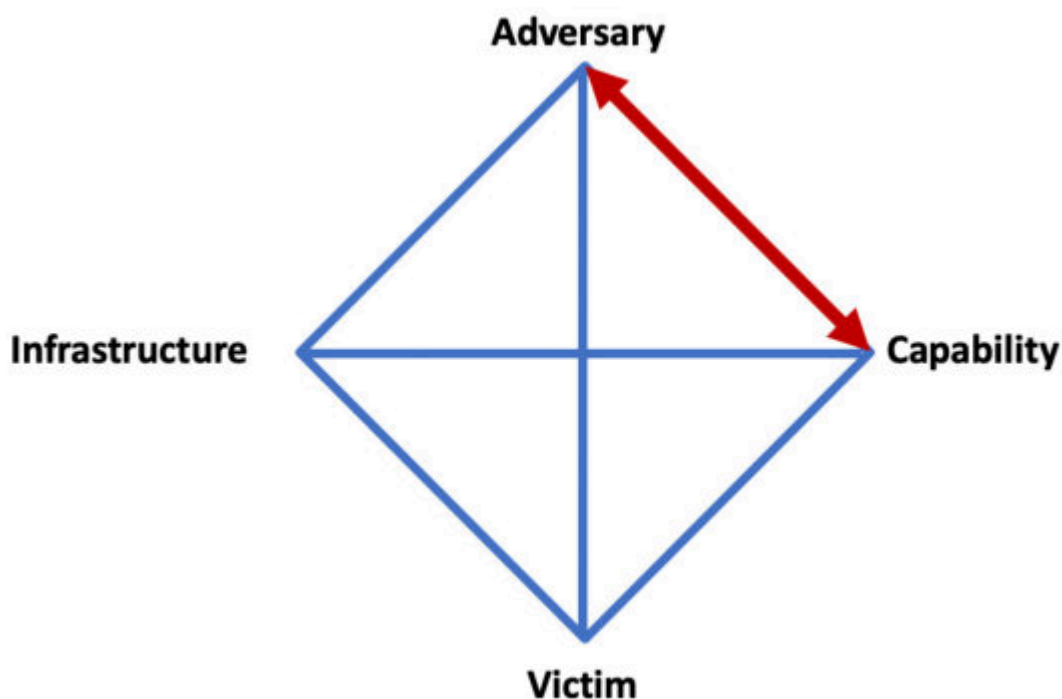


Figure 1: Diamond Threat Model: link between the adversary and its capability.

Furthermore, we worked on gathering victim information via telemetry detections from the *McAfee* backend. Even though we had developed a vaccine that prevented encryption, customers could still be exposed to the virus executable and this telemetry allowed us to monitor hits based on hash values. By doing so, we linked potential victims not only to the specific malware but also, as discussed in our research, we were also able to link groups of GandCrab samples to a single adversary or, in this case, an affiliate.

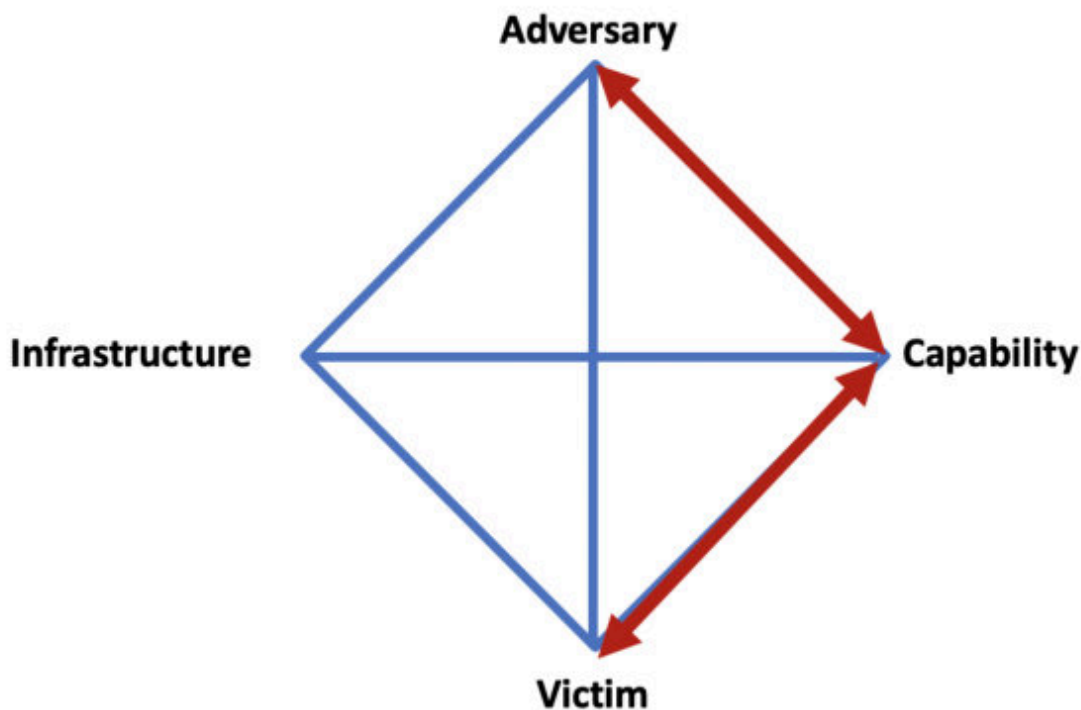


Figure 2: Linking victims to an adversary via the malware (capability).

This paper presents the highlights of our extensive research and details our methodology in examining the service model and the top GandCrab affiliates. Before we start with the technical analysis, it is important to understand the different versions of the ransomware that have been released.

Version overview

In order to be able to identify possible mistakes in the malware, and/or to find ways to disrupt the service model, it was important for us to have a clear understanding of the different versions of the malware, its development speed, and the agility of the actors behind it.

Version 1

The GandCrab malware was first discovered in late January 2018 by David Montenegro [1]. At the beginning, GandCrab only accepted payment of the ransom using Dash, but later included BTC as another method of payment. GandCrab was not made flawless; the first version and its infrastructure contained several mistakes that resulted in the development of a free decryptor.

Version 2

On 5 March 2018, one week after the release of the decryptor, a new version of GandCrab was discovered by the research collective *MalwareHunterTeam* [3]. GandCrab had fixed earlier mistakes, rendering the previously developed decryptor useless. This version had a new extension for the crypted files, different hard-coded domains,

and was offered in both .EXE and DLL formats, allowing affiliates to choose their preferred method of spreading the ransomware [4].

The GandCrab crew's swift response to the release of the decryptor showed that they were agile and determined to remain active.

There was already a hard-coded version number in the GandCrab malware, but this was not always accurate. We came across numerous strange version numbers that we believe were an attempt to mislead the research community.

Version 3

The third version of GandCrab was spotted around 23 April 2018 by a researcher using the *Twitter* handle *nao_sec* [5]. A later iteration of version 3 was subsequently discovered on 9 May 2018 by a researcher with the *Twitter* handle *zswei* [6].

Version 3 was the first to have a different wallpaper and we discovered signs of sub-versions of this iteration, identified by subtle changes in the wallpaper and, in some cases, the use of process injection.

Version 4

At the beginning of July 2018, version 4 was released. Version 4 featured some significant changes from the earlier versions. An important difference in version 4.0 was a change in the algorithm used to encrypt files. Earlier versions used RSA and AES; in version 4 GandCrab switched to Salsa20. We believe that this was mostly done for speed. RSA is a powerful but slow algorithm, whereas Salsa20 can encrypt much faster and the implementation is smaller. The ransomware generates a pair of RSA keys before encrypting any file. The public key encrypts the Salsa20 key and the random initialization vector (IV, or nonce) generated later for each file. GandCrab also used the registry to keep the generated RSA keys; this later proved vital in making the different vaccines.

To further increase speed, GandCrab started using another thread to look for network shares besides the normal encryption thread. The wallpaper introduced in version 3 disappeared.

One of the most important changes that helped our research was the introduction of more stable administration using hard-coded ID and SUB_ID numbers in the ransomware. We believe the GandCrab developers introduced this in order to have a tight accounting method for all the affiliate infections, to cope with the growth of the RaaS. Eventually, these hard-coded values proved to be vital in our research to understand the service model through large-scale ID number tracking and linking ID numbers to victims and affiliates.

Version 5

In version 5, GandCrab showed real signs of stepping up its game by showcasing new alliances with other criminal services to strengthen its supply and distribution networks. One of these alliances became obvious during version 4, in which the ransomware started being distributed through the new Fallout exploit kit. In the announcement of version 5, the GandCrab crew openly endorsed working with the Fallout exploit kit.

This was not the only partnership, though: another alliance was formed with a malware crypter service called NTCrypt and, later, AlexCrypt. A crypter service provides malware obfuscation to evade detection by anti-malware products. However, the GandCrab crew were not completely ruthless – after receiving a Tweet from a Syrian victim they decided to unlock all victims in that locale.

On 25 October a decryptor for GandCrab up to version 5.03 was made available via the *NoMoreRansom* [7] platform. This was a huge blow for the almost untouchable GandCrab ransomware and gave rise to some interesting conversations in the cybercriminal underground. Interestingly, it was the user behind the Kraken ransomware, ThisWasKraken, who broke the news first on the forum. Shortly after the release of the decryptor, GandCrab came out with version 5.04. Based on the compilation dates of the first 5.04 samples we believe that this new version was not a direct reaction to the release of the decryptor, but a new version that they had already been planning on spreading. At the end of 2018 the activity slowed down. Judging by GandCrab’s underground posts (an example of which is shown in Figure 3), the cybercriminal group was working mostly in the background, trying to fix and improve the ransomware.

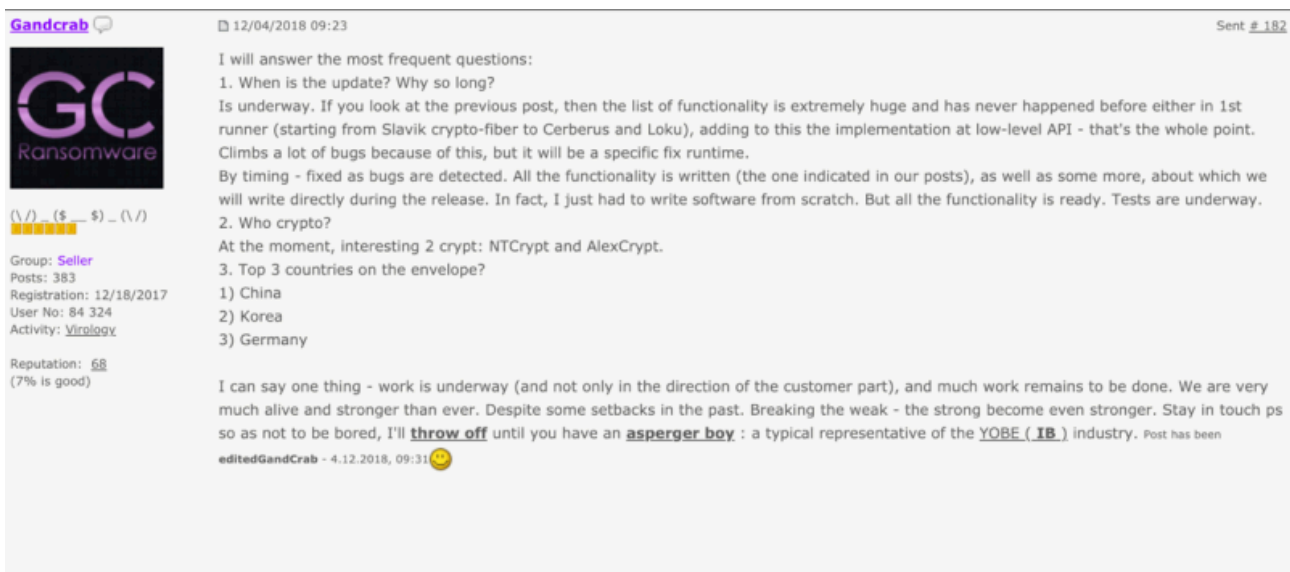


Figure 3: GandCrab explaining the delays.

GandCrab came back with the release of version 5.1 on 16 January 2019, two days after the Orthodox new year [8]. However, this victory was short-lived because another decryptor was released at the end of January. In response, the criminals behind GandCrab released version 5.2 a couple of days after the publication of the new decryptor. Despite the new version, Europol announced in mid-March [9] that, with the new decryptor, more than 14,000 people had been able to save their encrypted files – a significant blow for the criminals. GandCrab announced that it was stopping its business on 31 May 2019, claiming it had made hundreds of millions of dollars along the way. Despite the amount of money made, no mercy was shown to the remaining victims and they were urged to pay the ransom – GandCrab did not release the last keys for free. Luckily for those victims, there came a final version of the GandCrab decryptor, meaning they could also get their files back without paying the ransom.

Technical malware analysis

As discussed, several versions of GandCrab have been released since its initial appearance in January 2018. In this part we will highlight some technical insights into the two major versions: 4 and 5. Although some behaviours

were the same, some significant changes were also observed.

Algorithms and languages

Versions 4 and 5 encrypted victim files using the Salsa20 algorithm. Previous versions of GandCrab had used AES as an encryption algorithm. The Salsa20 algorithm has some advantages over AES:

- Small code and very fast. Speed in relation to encrypting files is critical for ransomware. A faster algorithm is better than a slower one. AES uses the CPU extension behind the scenes.
- Open-source implementation. AES also has open-source implementation, but the Salsa20 implementation is smaller in code and there are a lot of variants of the algorithm in source code format in C and other languages.
- Robust and with no known failures in the algorithm.
- Support for 16- and 32-bit keys with an initialization vector. GandCrab used the 32-bit version.

GandCrab obtained all processes of the system and searched for the common process names, just like other ransomware families such as Cerber. If it detected any of them, it would try to open the process and terminate it using the 'TerminateProcess' function.

GandCrab checked the language of the victim system before starting the process of gathering information. For this it used the 'GetUserDefaultUILanguage' and 'GetSystemDefaultUILanguage' functions. These functions would get the local language of the victim system and compare it with a hard-coded list that included all CIS countries. If the victim system was based in one of the CIS countries GandCrab would terminate.

In the earlier versions of GandCrab there was no function for detection of the Syrian language. Later, after disclosing the decryption keys to victims in that locale, the GandCrab crew added the Syrian language to the hard-coded list.

GandCrab determined the language of the victim machine by reading from a registry entry:

```
[HKEY_CURRENT_USER\Keyboard Layout\Preload]
```

It read the value and compared it against the hard-coded language list, checking for the Russian language value (0x419), for example.

```
mov     [ebp+var_40], 419h ; Russian language
mov     [ebp+var_3C], 422h ; Ukranian Language
mov     [ebp+var_38], 423h ; Belarusian language
mov     [ebp+var_34], 428h ; Tajik language
mov     [ebp+var_30], 42Bh ; Armenian Language
mov     [ebp+var_2C], 42Ch ; Arzerbaijan Language
mov     [ebp+var_28], 437h ; Georgian Language
mov     [ebp+var_24], 43Fh ; Kazakhstan Language
mov     [ebp+var_20], 440h ; Kyrgyzstan Language
mov     [ebp+var_1C], 442h ; Turkmenistan Language
mov     [ebp+var_18], 443h ; Uzbekistan Language
mov     [ebp+var_14], 444h ; Tatar Language
mov     [ebp+var_10], 818h ; Moldova Language (Romanian Based)
mov     [ebp+var_C], 819h ; Moldova Language (Russian Based)
mov     [ebp+var_8], 82Ch ; Azerbaijan Language
mov     [ebp+var_4], 843h ; Uzbekistan Language
call    ds:GetUserDefaultUILanguage
movzx   esi, ax
call    ds:GetSystemDefaultUILanguage
movzx   edx, ax
xor     eax, eax ; clear counter
jmp     short _loop_against_languages
```

Figure 4: Language check on the victim’s system.

After the language check, GandCrab prepared a string based on the serial number of the main installation of *Windows* and some calculation (as a division by 2) in hexadecimal format. To this string the malware added the extension ‘.lock’ and checked for this file. If it already existed, the malware quit. If not, it continued.

Following the release by *AhnLab* of a program [10] that made this file act as a vaccine, the GandCrab crew responded by changing this check – a sign that they were closely monitoring the industry.

After that, the malware prepared the RSA public key, which was embedded and crypted with two layers (except in the last version of the malware (5.2), which used three layers to protect it). The first layer was a simple XOR operation with the value 0x5 and the second layer was decryption using the Salsa20 algorithm with a hard-coded key (see Figure 5). In v5.2 the third layer was a custom algorithm (the first layer, followed by the two other steps):

- Version 4 to 5.1: XOR + Salsa20
- Version 5.2: custom algorithm + XOR + Salsa20

```

mov     eax, [ebp+var_1C]
mov     [ebp+var_60], eax
mov     eax, [ebp+var_18]
mov     [ebp+var_44], eax
mov     eax, [ebp+var_14]
push    4 ; flProtect
mov     [ebp+var_40], eax
mov     eax, [ebp+var_10]
push    3000h ; flAllocationType
mov     [ebp+var_3C], eax
mov     eax, [ebp-0Ch]
push    114h ; dwSize
mov     [ebp+var_38], eax
mov     eax, [ebp+var_4]
push    0 ; lpAddress
mov     dword_417964, offset GandCrabGlobalExpand32ByteKStringForSalsa20 ; "expand 32-byte k\\"
mov     [ebp+var_70], 'apxe'
mov     [ebp+var_5C], '3 dn'
mov     [ebp+var_48], 'yb-2'
mov     [ebp+var_34], 'k et' ; expand 32-byte k
mov     [ebp+var_58], ecx
mov     [ebp+var_54], eax
mov     [ebp+var_50], 0
mov     [ebp+var_4C], 0
call    ds:VirtualAlloc ; reserve memory to keep the RSA1 decrypted blob
push    114h
push    eax
mov     edx, offset GandCrabRSA1KeyCryptedBlob
mov     pbData, eax
lea    ecx, [ebp+var_70]
call    GandCrabManageCryptotRSAXKeyBlob
add    esp, 8
mov    esp, ebp
pop    ebp
retn

```

Figure 5: Decrypting the master RSA key in the second layer.

The key of the Salsa20 algorithm is ‘expa@hashbreaker Dannd 3@hashbr’ (without quotes). It is based on the name of the author of the Salsa20 algorithm and his *Twitter* nickname (‘hashbreaker’).

If the malware could not get the RSA key it would terminate since it was needed later. The RSA key was saved in a global var in a buffer to keep it in blob format.

Victim information and hard-coded values

After preparing the RSA key, the ransomware would get information from the victim machine and save it as a big string that would later be ciphered with the RC4 (sometimes a custom XOR) algorithm and encoded in Base64 to save it into the ransom note.

The information and fields were as follows:

pc_user	Name of the user logged into the machine.
pc_name	Name of the endpoint infected.
pc_group	Name of the domain or workgroup of the endpoint.
AV	Name or names of anti-virus product(s) in the endpoint.
pc_lang	Name of the language or languages of the endpoint.
pc_keyb	The type of keyboard on the endpoint.

os_major	The name of the operating system of the endpoint.
os_bit	The type of CPU of the infected endpoint.
ransom_id	Unique value for the victim for the ransom note and Onion web page to pay.
hdd	Information about the logic units.
ip	The IP address of the endpoint.

After preparing this big string, it would concat more special fields hard coded in the malware sample. These values were:

id	The affiliate id number.
sub_id	The sub id of the affiliate id.
version	The internal version number of the malware. In the example below, it is '4.0' but the last version would have shown '5.2'.

The malware concated these fields with the previously mentioned big string to finally get all the information about the system and malware information in plain text.

The hard-coded values above proved to be of great importance in analysing the RaaS model of GandCrab.

For example, in this case the final string is as follows (some fields are altered to fool the malware to fill the fields):

```
pc_user=IDC_UNIT56&pc_name=IDC_SEAT_56&pc_group=WORKGROUP&av=[System Process],  
smss.exe&pc_keyb=0&os_major=MicrosoftWindowsXP&os_bit=x86&ransom_id=6deb15d  
d9c2e5c79&hdd=C:FIXED_10725732352/8747036672,E:REMOTE_511503020032/2577879613  
44&id=15&sub_id=15&version=4.0
```

This big string would be crypted and a Base64 string prepared from it to put in the ransom note between the marks:

- ---BEGIN PC DATA---
- ---END PC DATA---

Payload execution

After these steps the malware started its critical malicious payload that was executed in a number of steps.

The first step was to verify the file extensions that the malware was not allowed to encrypt. These were stored in a hard-coded buffer crypted with the value 0x5.

The extensions that were blacklisted were as follows:

.ani

.cab

.cpl

.cur

.diagcab

.diagpkg

.dll

.drv

.lock

.hlp

.ldf

.icl

.icns

.ico

.ics

.lnk

.key

.idx

.mod

.mpa

.msc

.msp

.msstyles

.msu

.nomedia

.ocx

.prf
.rom
.rtp
.scr
.shs
.spl
.sys
.theme
.themepack
.exe
.bat
.cmd
.gandcrab
.KRAB
.CRAB
.zerophage_i_like_your_pictures

The above list is an example of common extensions to avoid, just as other ransomware families use. However, GandCrab's list includes some additions that are interesting:

- .lock: this extension belongs to the files that the malware creates in each folder to indicate that it is crypting them, as a measure to avoid another instance of GandCrab affecting the same folder. Usually this file is empty; for GandCrab only its presence/absence is important.
This file extension changed in some samples to '.luck' or '.fuck', so not all versions have the same name.
- .zerophage_i_like_your_pictures: this extension does not exist by default in *Windows*; it is a joke against a malware analyst, zerophage.
- .KRAB: this extension is included to avoid crypting the files that had already been encrypted.
- .CRAB: again, this extension is included to avoid crypting the files that had already been encrypted.
- The newest versions of GandCrab did not use a special extension, instead calculating a random extension that could be from 5 to 10 characters in length, depending on the version. In this case GandCrab saved the randomly created extension in the registry too, in a special subfolder (with the exception of the last version where this registry write code was removed to destroy the vaccine that had been created). In any event, the extension was included in the ransom note so the official decryptor (given to the victims when they paid the ransom) had the correct extension to search for.

The next action of the malware was to create a pair of RSA keys (one public and one private) to protect the keys that would later be used to crypt the files.

The malware reserved two memory buffers with 'VirtualAlloc' and acquired the context of the cryptosystem of Windows using the 'CryptAcquireContextW' function with the parameter 'Microsoft Enhanced Cryptographic Provider 1.0'.

In v1 of GandCrab there was a bug at this point where keys could be extracted from memory. After the release of a decryptor, the actors resolved this in version 2 using the value 0xF0000000 as a flag.

After creating the key pair, GandCrab exported them in two RSA blobs using the 'CryptExportKey' function with the argument '6' for the public RSA1 key blob and with the argument '7' for the private RSA2 key blob. After the exports, it destroyed the key pair from memory using the 'CryptDestroyKey' function and released the context using the 'CryptReleaseContext' function. In this case the function returned with TRUE as a result to cause the malware to follow the normal flow to crypt shares and files later.

It is important to understand that the memory reserved for both blobs was not released at this point as it would be needed later. By taking a memory dump at this point, both blobs could be retrieved.

Before version 4.0, GandCrab was creating registry settings that we could use to create a vaccine. If these registry settings were already present, and the ransomware failed to create them, the malware would not encrypt and would terminate. However, this was adjusted in version 5.2.

Another protection mechanism that worked with v4.0 and some of v5 (not including 5.2) was to create subkeys in the registry and remove all rights for all users (including SYSTEM). This way the subkey could not open because of the ERROR_ACCESS_DENIED (5) error that not is checked. The malware would create the Salsa20 key, etc. anyway, but could not save them in the registry and would return FALSE, causing the code flow to delete itself without crypting anything in the endpoint.

The next action by the malware was to prepare the strings of information about the infected endpoint (that was crypted in the previous layer) and the Salsa20 key, IV and RSA2 private key blob (all crypted in the first layer).

For this, the malware prepared some strings that were hard coded in the code:

- --- BEGIN GANDCRAB KEY ---
- --- END GANDCRAB KEY ---

Between these two strings the malware kept in memory the Salsa20, IV and crypted RSA2 buffer but, before that, it crypted the RSA2 buffer again with another layer.

The next action was to prepare the ransom note in memory. The malware decrypted the ransom note text with a XOR value of 0x10. After decrypting the full ransom note, it once again got the system information for the 'random_id' and 'pc_group' fields. Next, it wrote the previous string with the information of the endpoint and malware sample crypted between the two marks ' --- BEGIN PC DATA --- ' and ' --- END PC DATA --- ' and the Salsa20 key, IV and RSA2 private key blob crypted between the two marks ' --- BEGIN GANDCRAB KEY --- ' and ' --- END GANDCRAB KEY --- '.

code to delete the shadow volumes in a quiet way:

```
cmd.exe /c vssadmin delete shadows /all /quiet
```

If the OS version was *Windows Vista* or above, it would prepare the hard-coded text in the code:

```
\wbem\wmic.exe shadowcopy delete
```

Finally, the malware deleted itself without warning the user or awaiting user interaction.

Changes in GandCrab Version 5

Version 5 of GandCrab included a lot of changes to make analysis more complex. The authors also fixed a lot of code in the last version, '5.2', to avoid problems and vaccines, but that did not stop us from creating working vaccines for all versions.

One important change in the 5.x version was the inclusion of exploits to elevate privileges. The exploits were CVE-2018-8440 [11] by SandboxEscaper [12], and CVE-2018-8120 [13, 14].

The first version of GandCrab v5 was faulty because it used one exploit directly with IAT calls that do not exist in *Windows XP*. This prevented it from working on that OS, but the issue was quickly fixed in the next version.

Both exploits were used in *Windows 7* and newer OS versions to try to get SYSTEM privileges. With CVE-2018-8120 [15], it tried to steal the system token of the SYSTEM idle process. In the other exploit, the malware tried to load a special DLL that had crypted code inside, one for 32-bit and another for 64-bit systems.

These exploits could be blocked if a mutex with a hard-coded name exists in the future infected machine (see Figure 7).

```
sub_4059EF      proc near                                ; CODE XREF: .text:loc_405A57↓p
               push    offset Name                ; "Global\\X1AKFoxSKG0FSG0oSF00FNOLPE"
               push    0                          ; bInheritHandle
               push    100000h                    ; dwDesiredAccess
               call    ds:OpenMutexW
               test    eax, eax
               jz     short loc_405A10
               push    eax                        ; hObject
               call    ds:CloseHandle
               xor    eax, eax
               inc    eax
               retn

; -----
loc_405A10:    xor    eax, eax                                ; CODE XREF: sub_4059EF+14↑j
sub_4059EF      retn
sub_4059EF      endp
```

Figure 7: Check of the mutex – if it already exists, the exploits are blocked.

Some changes in the v5.2 family included:

- Instead of using a hard-coded extension after encryption, it created a name with 5 to 10 random characters.
- The use of two exploits.
- Changing the desktop wallpaper on the infected system to a bitmap generated in runtime with the name of the active user and the extension used for the encrypted files. This wallpaper was created in a faulty way if the vaccine (explained later) was used – in that case the extension did not appear.
- The wallpaper was saved on the hard disk in the %TEMP% folder with the hard-coded name ‘pidor.bmp’ (a very rude word in the Russian language).
- The username was checked with ‘SYSTEM.’ If the user was ‘SYSTEM’, the malware put the name ‘USER’ in the wallpaper. This check was made to avoid changing the wallpaper on an account where GandCrab had achieved system privileges, presumably because that would have alerted the user to what had happened.

The last version of GandCrab also removed a lot of useless code that been inserted to try to obfuscate some macros and static disassembling.

Besides all the layers that GandCrab put in the last version (5.2), an unofficial version ‘5.3’ was uploaded to *VirusTotal* with a different ransom note and another RSA public key. We do not know for sure why this happened.

Building a vaccine

During its development history, GandCrab showed clear differences in programming styles and mistakes made, which supports our hypothesis that multiple programmers worked on the code.

One of our goals with this ransomware threat was to create a vaccine that could work against it and protect our customers’ systems.

Our approach was to reverse the inner workings of the GandCrab family in order to discover failures in the design of the malware so that could create effective vaccines.

In total, six vaccines were crafted. The first was the most interesting because it was clear that a flaw existed in the logic of the malware. The GandCrab crew did not fix this until version 5.

The vaccine was based on code stored in the registry [16, 17]. The information, without Base64, was the same as that which appeared in the ransom note in the key part (but in the ransom note it was encoded in Base64 to print all characters). One registry subkey was stored in HKLM or HKCU, based on the privileges of the user that ran the malware. It used the public RSA key generated in runtime to protect the private RSA key of the victim.

The flaw in the design was that it didn’t use these registry values in later versions, even in the official decryptor. The malware checked for the existence of this subkey and values. If it found the public RSA key called ‘public’ it did not check the content to see if it was correct. If the value in the subkey was empty, or contained some random content, the malware believed that the victim machine was already infected and skipped all processes to encrypt files. In this case the malware launched the network thread anyway, if the victim had Internet connectivity, but the critical and most dangerous part was never launched. For the wallpaper change, a bitmap file was created in

runtime and stored with the hard-coded name ‘pidor.bmp’ in the %TEMP% folder of the infected system. Another vaccine that we made cleaned this file and removed the wallpaper, exchanging it for a clean wallpaper [18].

This vaccine worked for at least six months after its release to the public. It took until early this year, and the release of version 5 of GandCrab, for this part of the code to be cleaned and the vaccine thus rendered useless.

Another vaccine [19] came about from a clear design flaw, and an indication that the GandCrab crew did not care about the coding: in version 5 a hidden window was created with a hard-coded class name. When it was reversed and analysed, it only took us five minutes to make a new vaccine in a program that searched the full system, on an x-period basis, for the window with this class name (see Figure 8).

```
loc_401070:                                ; CODE XREF: sub_401000+6C↑j
      push    offset szWindow ; lpszWindow
      push    offset szClass  ; "AnaLab_sucks"
      push    0                ; hWndChildAfter
      push    0                ; hWndParent
      call   ds:FindWindowExW
      test   eax, eax
      jz     short loc_401062
      lea   ecx, [ebp+dwProcessId]
      mov   [ebp+dwProcessId], 0
      push  ecx                ; lpdwProcessId
      push  eax                ; hWnd
      call  ds:GetWindowThreadProcessId
      test  eax, eax
      jz     short loc_401062
      push  [ebp+dwProcessId] ; dwProcessId
      push  0                ; bInheritHandle
      push  1FFFFFFh         ; dwDesiredAccess
      call  ds:OpenProcess
      test  eax, eax
      jz     short loc_401062
      push  0                ; uExitCode
      push  eax                ; hProcess
      call  ds:TerminateProcess
      jmp   short loc_401062
```

Figure

8: Searching for GandCrab window.

This was because the chosen class name was not present in *Windows*, thus action could be taken against it without any risk of harm to the operating system.

When the program detected the window, it would get the PID of the process linked to that window and, with the PID, it was able to close and terminate it. This meant that, with this vaccine, it was impossible for GandCrab to run – when it created and ‘showed’ the window, the vaccine would discover it and terminate it.

This vaccine was fixed more quickly than others, in the cleaning up of code that GandCrab undertook, making the vaccine useless.

Other vaccines [20] included a search for a mutex name and creation of it in the system before infection even occurred; the flaw here was that the mutex name was always the same, regardless of what machine was affected. Even if it had been variable, it was very easy to mimic the same behaviour and create a successful vaccine (the last vaccine that worked with version 5.2 protected the system in this way). The GandCrab gang knew this, and

sometimes changed the mutex name, but the vaccine was able to create the last two mutex names, meaning the last and previous versions were covered.

Prior to this vaccine, the GandCrab crew made another mistake that allowed us to create a new vaccine in a matter of minutes. They made a global atom in the machine, so our vaccine only needed to make the atom beforehand to protect the system. That was fixed when the mistake was discovered.

Figure 9 gives an overview of the timeline of the different GandCrab versions, our vaccines and the public decryptors.

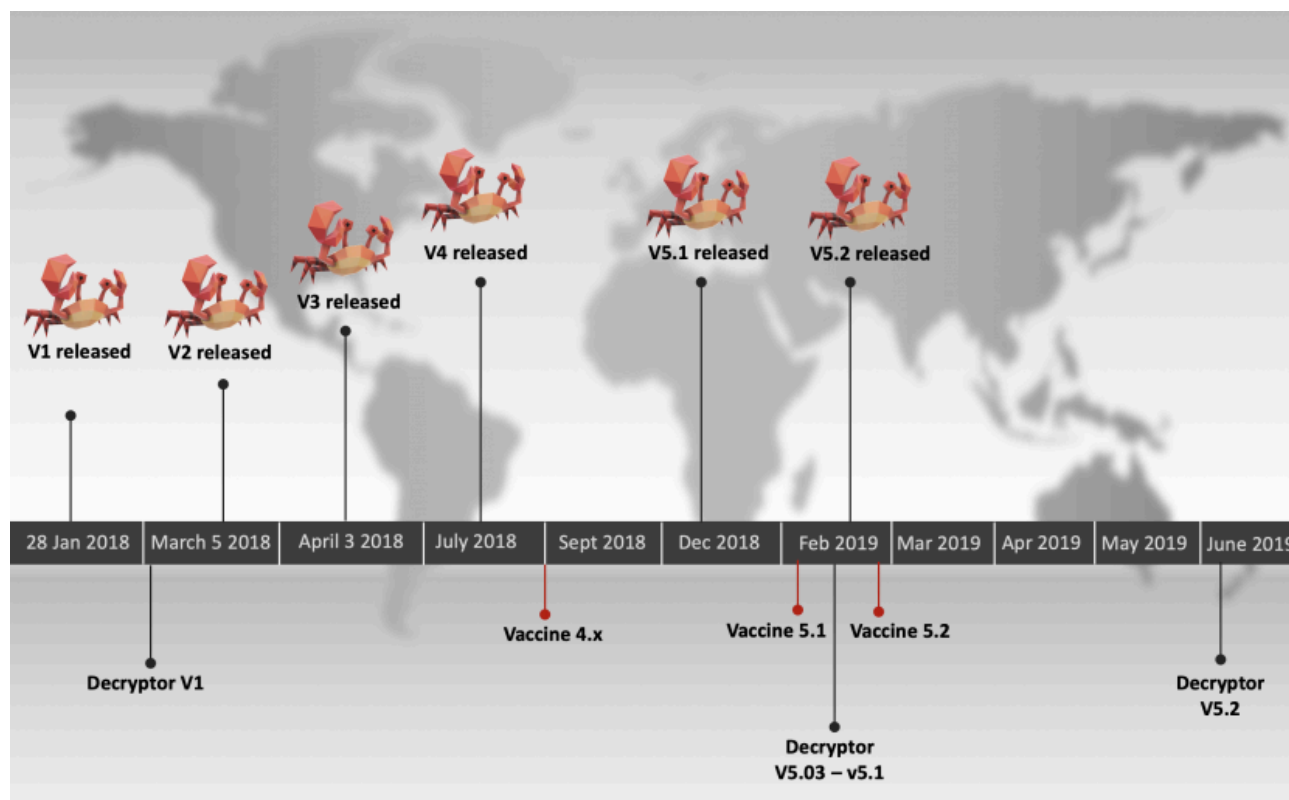


Figure 9: Version and vaccine timeline.

For malware to be successful, it needs to be effective, but it does not have to be flawless. As we have discussed, the various versions of GandCrab were full of little mistakes and errors that allowed us to build several different vaccines.

Thus, an important part of GandCrab's success was its service model and marketing.

GandCrab is a prime example of a Ransomware-as-a-Service threat. RaaS follows a structure where the developers offer their product to individuals, affiliates or partners, who are responsible for spreading the ransomware and generating infections. The developers take a percentage of the earned income and the rest goes to the affiliates.

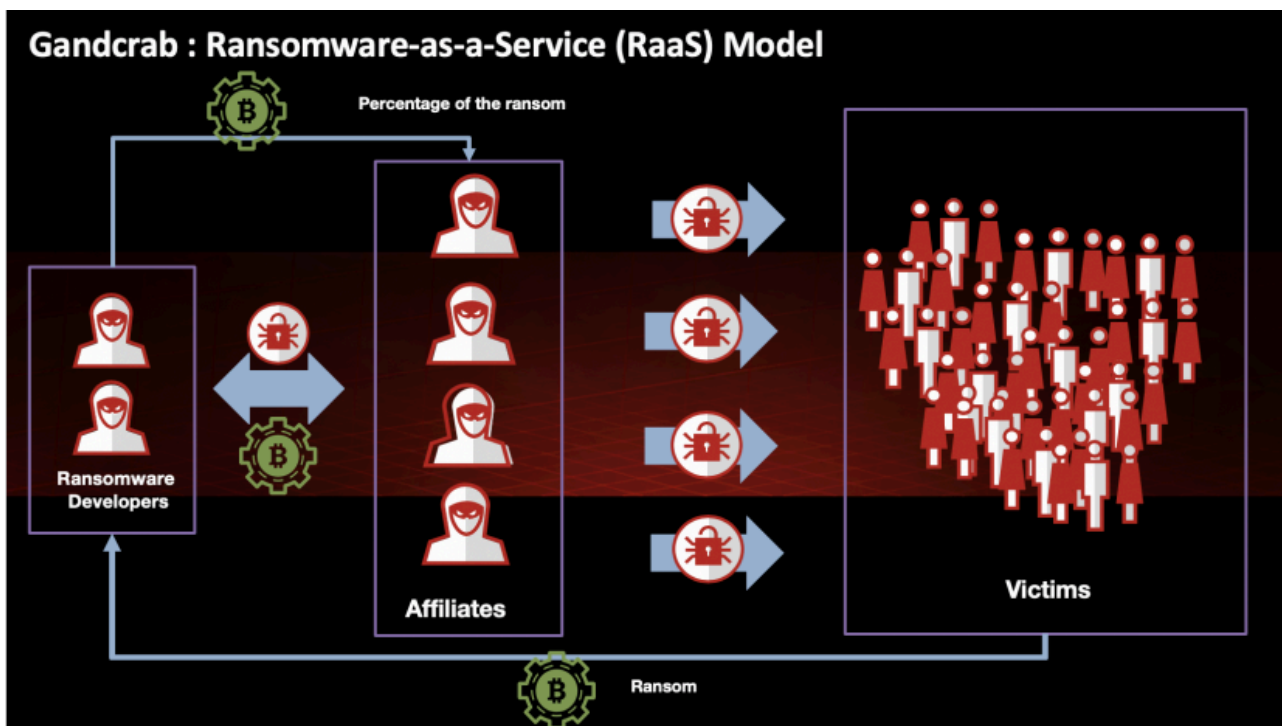


Figure 10: High-level overview of the GandCrab RaaS model.

Operating a RaaS model can be lucrative for both parties involved:

- Developer’s perspective

The malware author(s) request a percentage per payment for use of their ransomware product. This way the developers have less risk than the affiliates spreading the malware. The developers can set certain targets for their affiliates regarding the number of infections they need to produce. In a way, this is very similar to a modern sales organization in the corporate world.

Subsequently, a RaaS model offers malware authors a safe haven when they operate from a country that does not regard developing malware as a crime. If their own nation’s citizens are not victimized, the developers will not be prosecuted.

- Affiliate’s perspective

As an affiliate you do not have to write the ransomware code yourself; less technical skill is involved. RaaS makes ransomware more accessible to a greater number of users. An affiliate just needs to be accepted in the criminal network and reach the targets set by the developers. As a service model it also offers a level of decentralization, where each party sticks to their own area of expertise.

If proper administration of infections per affiliate is kept, a RaaS business model (developer/affiliates percentages) ensures that everyone gets a piece of the proverbial ‘pie’.

Partnerships to ensure growth

During its lifetime we observed several essential partnerships being established between the GandCrab ransomware and other facilitating services. The fact that cybercriminals were working together was not a new

thing.

In the cybercriminal underground there are several specialized services that can facilitate the preparation, pre-activity, activity and post-activity of financially driven cybercrime, as described by Erik van de Sandt [21].

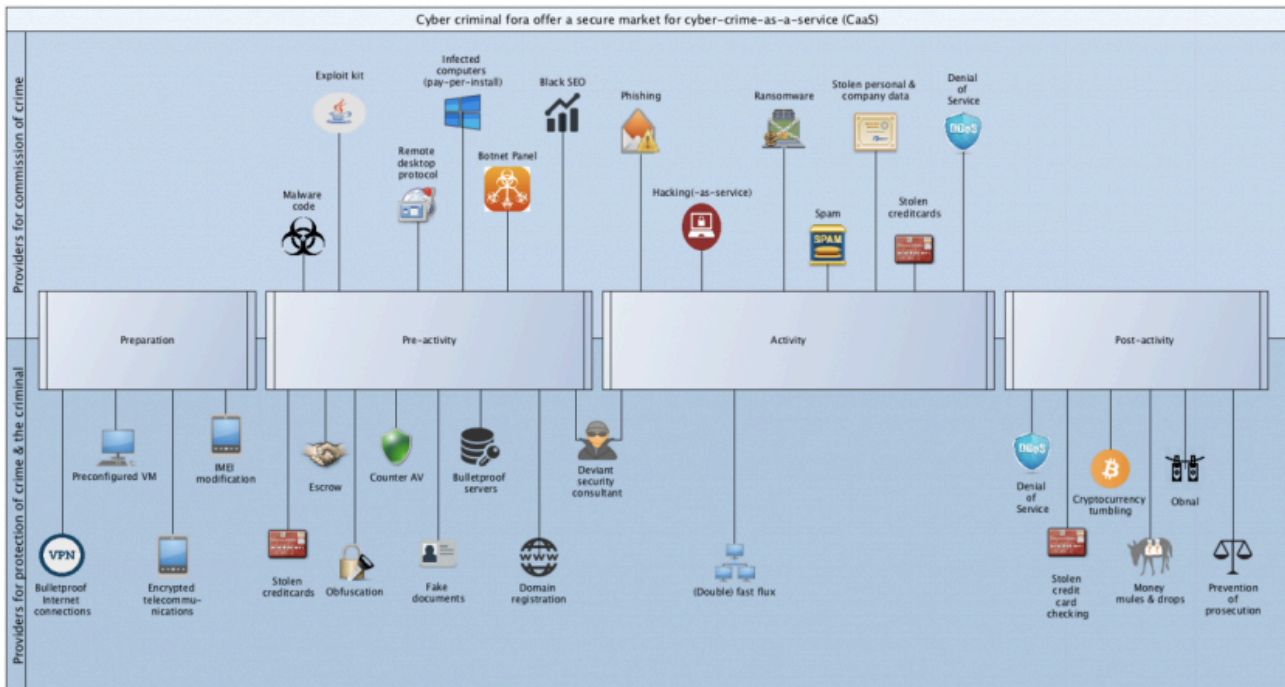


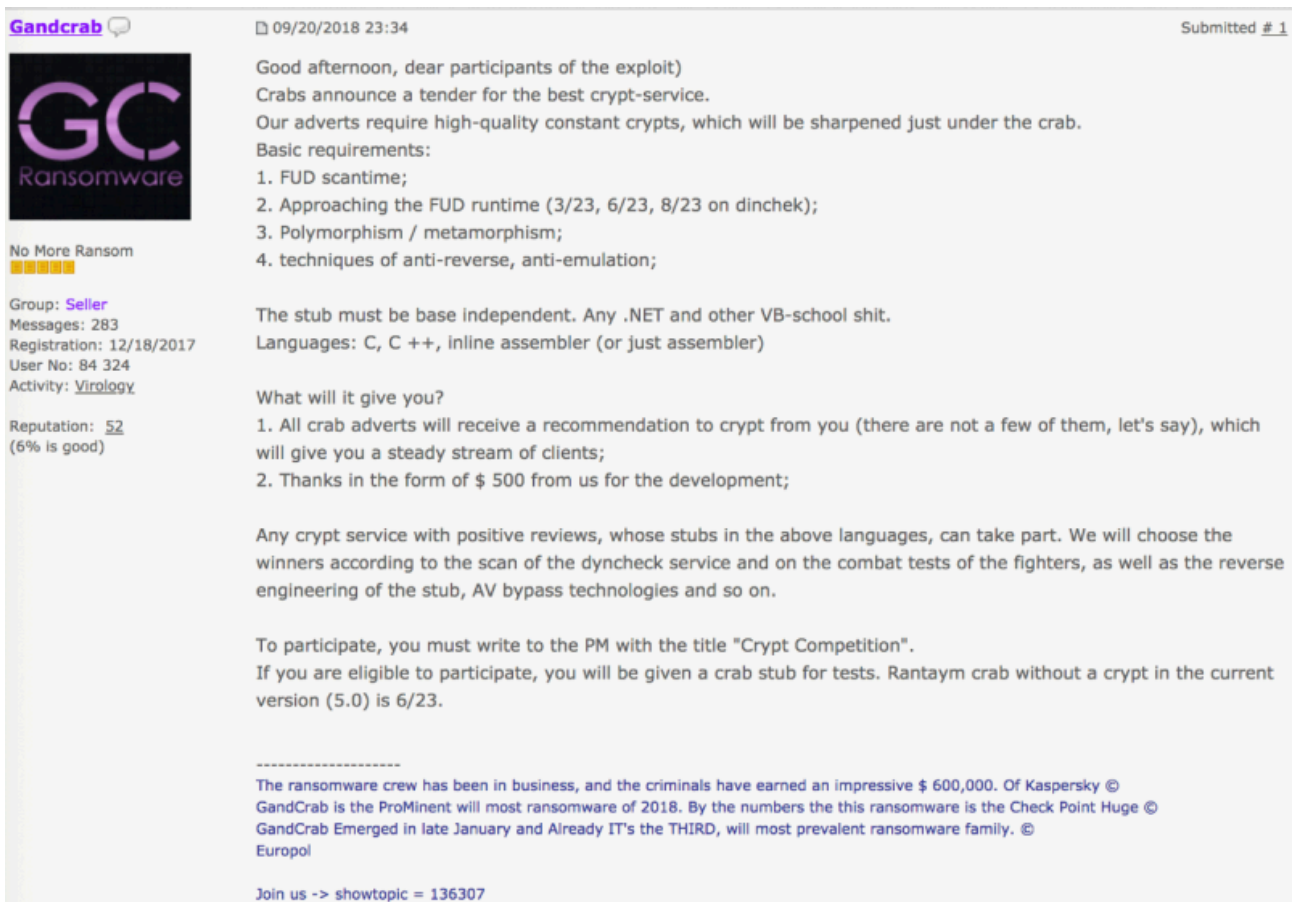
Figure 11: Overview by Van de Sandt [21], displaying the services offered on Russian-language cybercriminal forums that predominantly facilitate financially driven computer-focused crimes. Each serves a specific purpose in the preparation, pre-activity, activity and post-activity of the commission of crime, and the protection of crime and the criminal.

Often, cybercriminals will choose to interact with several facilitating services to ensure that they have all the necessary elements in place to commit their intended crime. However, it is less common to see facilitating services forming alliances amongst themselves to become more successful. Recently, Goznym was another good example of cybercriminal services working together to gain more revenue [22].

By choosing to work together, facilitating services expose themselves to a certain risk since they must trust their newly formed alliance with their partner. However, having a good reputation in the underground and an overall feeling of impunity helps providers of those services to form partnerships.

GandCrab was a perfect example of a service, in this case ransomware, that teamed up with other services such as RIG and the Fallout exploit kit. These alliances helped GandCrab’s customers spread ransomware on a larger scale, thus generating more income and traffic for both services.

GandCrab even used its popularity to issue an underground tender to find a new crypter service [23]. A crypter service provides malware obfuscation to evade detection by security products.



Gandcrab 09/20/2018 23:34 Submitted # 1

GC Ransomware

No More Ransom

Group: **Seller**
Messages: 283
Registration: 12/18/2017
User No: 84 324
Activity: **Virology**

Reputation: **52**
(6% is good)

Good afternoon, dear participants of the exploit)
Crabs announce a tender for the best crypt-service.
Our adverts require high-quality constant crypts, which will be sharpened just under the crab.
Basic requirements:
1. FUD scantime;
2. Approaching the FUD runtime (3/23, 6/23, 8/23 on dinchek);
3. Polymorphism / metamorphism;
4. techniques of anti-reverse, anti-emulation;

The stub must be base independent. Any .NET and other VB-school shit.
Languages: C, C ++, inline assembler (or just assembler)

What will it give you?
1. All crab adverts will receive a recommendation to crypt from you (there are not a few of them, let's say), which will give you a steady stream of clients;
2. Thanks in the form of \$ 500 from us for the development;

Any crypt service with positive reviews, whose stubs in the above languages, can take part. We will choose the winners according to the scan of the dyncheck service and on the combat tests of the fighters, as well as the reverse engineering of the stub, AV bypass technologies and so on.

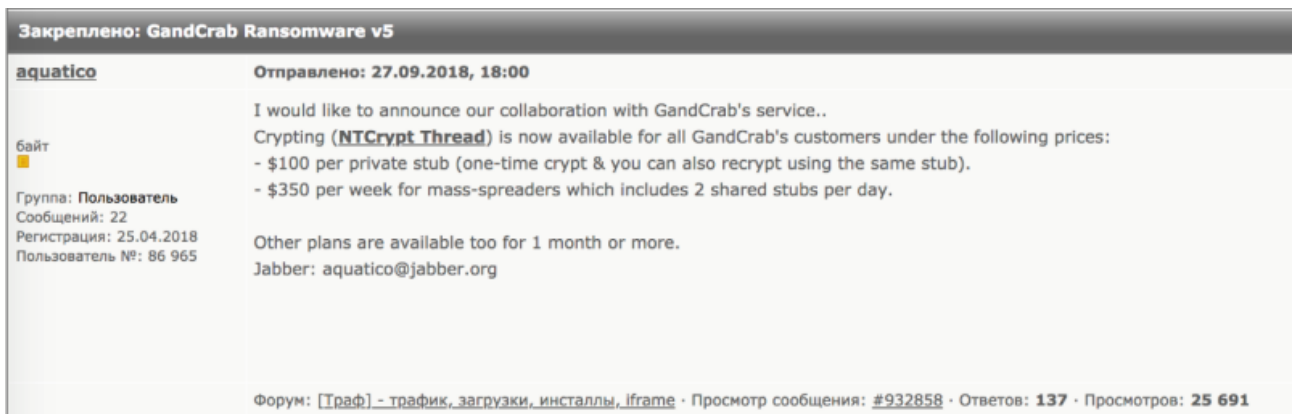
To participate, you must write to the PM with the title "Crypt Competition".
If you are eligible to participate, you will be given a crab stub for tests. Rantaym crab without a crypt in the current version (5.0) is 6/23.

The ransomware crew has been in business, and the criminals have earned an impressive \$ 600,000. Of Kaspersky © GandCrab is the ProMinent will most ransomware of 2018. By the numbers the this ransomware is the Check Point Huge © GandCrab Emerged in late January and Already IT's the THIRD, will most prevalent ransomware family. © Europol

Join us -> showtopic = 136307

Figure 12: Underground tender announcement for a new crypter service.

Eventually, NTCrypt won the tender and from then on offered a special price for customers of GandCrab.



Закреплено: GandCrab Ransomware v5

aquatico Отправлено: 27.09.2018, 18:00

байт

Группа: Пользователь
Сообщений: 22
Регистрация: 25.04.2018
Пользователь №: 86 965

I would like to announce our collaboration with GandCrab's service..
Crypting (**NTCrypt Thread**) is now available for all GandCrab's customers under the following prices:
- \$100 per private stub (one-time crypt & you can also recrypt using the same stub).
- \$350 per week for mass-spreaders which includes 2 shared stubs per day.

Other plans are available too for 1 month or more.
Jabber: aquatico@jabber.org

Форум: [Траф] - трафик, загрузки, инсталлы, iframe · Просмотр сообщения: #932858 · Ответов: 137 · Просмотров: 25 691

Figure 13: NTCrypt announcing that it had won GandCrab's crypter tender.

This behaviour is very similar to legitimate companies forming strategic partnerships and undertaking mergers and acquisitions to stimulate growth, gain a competitive advantage and increase market share. Therefore, observing the formation of alliances between facilitating services and a Ransomware-as-a-Service provider can be a sign of significant growth of operations.

Linking the ransomware to affiliates

Through our technical analysis, we established that, starting from version 4, GandCrab included certain hard-coded values in the ransomware source code:

- id: the affiliate id number.
- sub_id: the sub id of the affiliate id – we suspect that affiliates can sub rent infections to their own partners, identifiable via the sub_id number. However, more research is needed to confirm this.
- version: the internal version number of the malware.

Version 4 included a significant number of changes overall and we believe that these changes were made by the authors partly to improve administration and make GandCrab more scalable to cope with its increased popularity.

A successful service model is dependent on a tight administration of earnings because every party needs to feel that they receive what they have earned.

Based on the hard-coded values it was possible for us, to a certain extent, to extract the administration information and create our own overview.

We hunted for as many different GandCrab samples as we could find using YARA rules, industry contacts and customer submissions. The sample list we gathered is quite extensive but not exhaustive.

From the collected samples we extracted the hard-coded values and compilation times automatically, using a custom-built tool. We aggregated all these values together in one giant timeline from version 4, all the way up to version 5.2 (Figure 14).

1992-06-19	VERSION: 5.04	140	SUBID: 720	8ec87fd3ea777fa8d5160dc957e6683e	EXE	Timestomping
1992-06-19	VERSION: 5.04	140	SUBID: 763	9916e107b3d501c60d4baaf1b8f8a77a	EXE	Timestomping
2011-07-03	VERSION: 4.3	163	SUBID: 535	03915be56034b6ad7f66b5cfe1974f5e	EXE	Timestomping
2011-07-03	VERSION: 4.3	163	SUBID: 535	5abc4fea45e090ecda76cb5a56dafbc	EXE	Timestomping
2018-06-29	VERSION: 4.0	117	SUBID: 397	8d604e3c567aab3c8cfa2d2c424c09c4	EXE	
2018-06-30	VERSION: 4.0	9	SUBID: 9	cbdb4aebbb984096ee54c9eb2b1c128c	EXE	
2018-06-30	VERSION: 4.0	15	SUBID: 15	9be5102484d60f074499a8fd4403819c	EXE	
2018-06-30	VERSION: 4.0	15	SUBID: 15	77a7573a20dbf141a0ff1e5fade2eae0	EXE	
2018-06-30	VERSION: 4.0	41	SUBID: 62	19aa2a0f61f8b928b44de28d16f31174	EXE	
2018-07-03	VERSION: 4.1	9	SUBID: 9	946aa4b8273be8d4984e14d6c8c9d3b4	EXE	
2018-07-03	VERSION: 4.1	15	SUBID: 15	86613ae664b74c3a464f473408352635	EXE	
2018-07-03	VERSION: 4.1	44	SUBID: 83	3eed6f11720e53756db16de1b9a8d561	EXE	
2018-07-03	VERSION: 4.1	106	SUBID: 363	b6d06a87b35d15a1b3d9d76aced96f4e	EXE	
2018-07-03	VERSION: 4.1	114	SUBID: 383	7fd94b59280d80bcfdd1b3970c4189ed	EXE	
2018-07-04	VERSION: 4.1	15	SUBID: 15	fd602a6ae269d8fbc3b2c996678825b7	EXE	
2018-07-04	VERSION: 4.1	95	SUBID: 331	2212fac7fcded7f06042d0a0ca67898f	EXE	
2018-07-04	VERSION: 4.1	100	SUBID: 411	903f8718a1c3c12042fc44bac6a4c786	EXE	
2018-07-04	VERSION: 4.1	106	SUBID: 363	c24b3cf9336e7e994625d223fa7b5f4f	EXE	
2018-07-04	VERSION: 4.1	122	SUBID: 403	340117038ae2ef8d07c90c614d652934	EXE	
2018-07-05	VERSION: 4.1	110	SUBID: 374	9a680a7ff23746d92f4bb274c50be4a5	DLL	
2018-07-05	VERSION: 4.1.1	41	SUBID: 62	24fdd71ded0b9ffecfe3387002f7d361	EXE	
2018-07-05	VERSION: 4.1.1	73	SUBID: 414	ef50f5c2d6d8d10d8adc1efb840518d0	EXE	
2018-07-05	VERSION: 4.1.1	99	SUBID: 386	cce23a33a5a78b24ef7f5ced7d715d95	EXE	
2018-07-05	VERSION: 4.1.1	99	SUBID: 386	f876735f6d4f076dfb148c63c4ba5a3a	EXE	
2018-07-05	VERSION: 4.1.1	99	SUBID: 386	9b785e93d9ce42f6213d2c2a1ecc8293	EXE	
2018-07-05	VERSION: 4.1.1	99	SUBID: 417	c6b0fbf5190d3850b212c53e6ed56886	EXE	
2018-07-05	VERSION: 4.1.1	99	SUBID: 417	9c973702f8b40793c3ab60329dad7263	EXE	
2018-07-05	VERSION: 4.1.1	99	SUBID: 417	62fd133df8be543900aabe64b707896b	EXE	
2018-07-05	VERSION: 4.1.1	124	SUBID: 413	a74c335c0ee5958929b99cb14726cd9a	EXE	
2018-07-05	VERSION: 4.1.1	128	SUBID: 423	e8e19525aa73d1714f15552d166aaa84	EXE	
2018-07-06	VERSION: 4.1.1	111	SUBID: 375	6b8872624f0427deaf8df292038e657a	DLL	
2018-07-13	VERSION: 4.1.2	41	SUBID: 62	75f215c1f086c47ee45392bd188909d0	EXE	

Figure 14: Small portion of the timeline of collected samples (note the first four may be timestomped).

At the time of writing we have collected 314 different samples. The collected samples were run simultaneously on the *McAfee* backend to find any internal detection telemetry.

Of all our collected samples we only found four that had an irregular compile time. This anomaly might indicate deliberate timestomping [24], or it could be a defect from unpacking. The rest of the samples had compile times that correlated closely with the release dates mentioned on the forums and the security product detection dates.

ID and SUB_ID characteristics observed

Parent-child relationship

The extracted IDs and Sub_IDs showed a parent-child relationship, meaning that every ID could have more than one SUB_ID (child), but every SUB_ID only had one ID (parent).

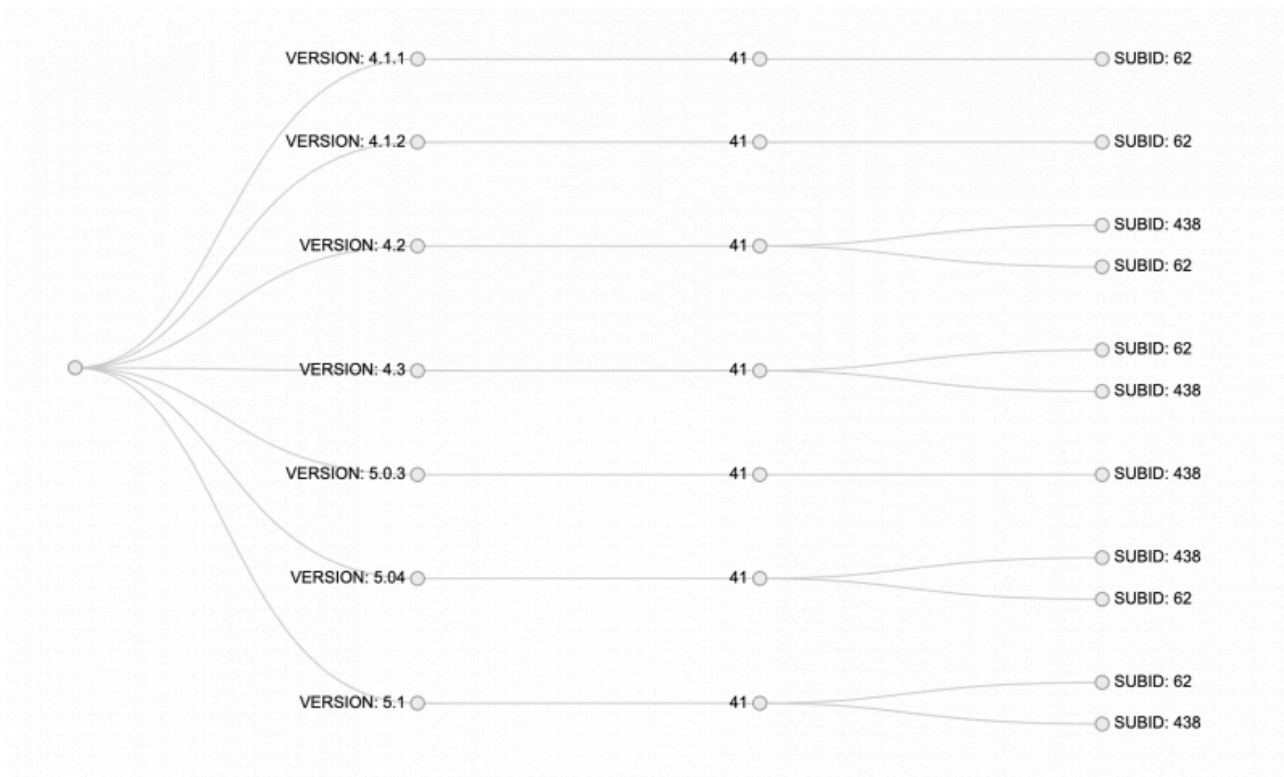


Figure 15: The activity of ID number 41 (parent) and its corresponding SUB_IDs (children).

ID increments

Overall, we observed a gradual increment in the ID number over time. The earlier versions generally had lower ID numbers and higher ID numbers appeared in the later versions.

However, there were relatively low ID numbers that appeared in many versions.

This observation aligned with our theory that the ID number corresponds with a particular affiliate. Certain affiliates remained partners for a long period of time, spreading different versions of GandCrab; this explains the ID number appearing over a longer period and in different versions. This theory has also been acknowledged by several (anonymous) sources.

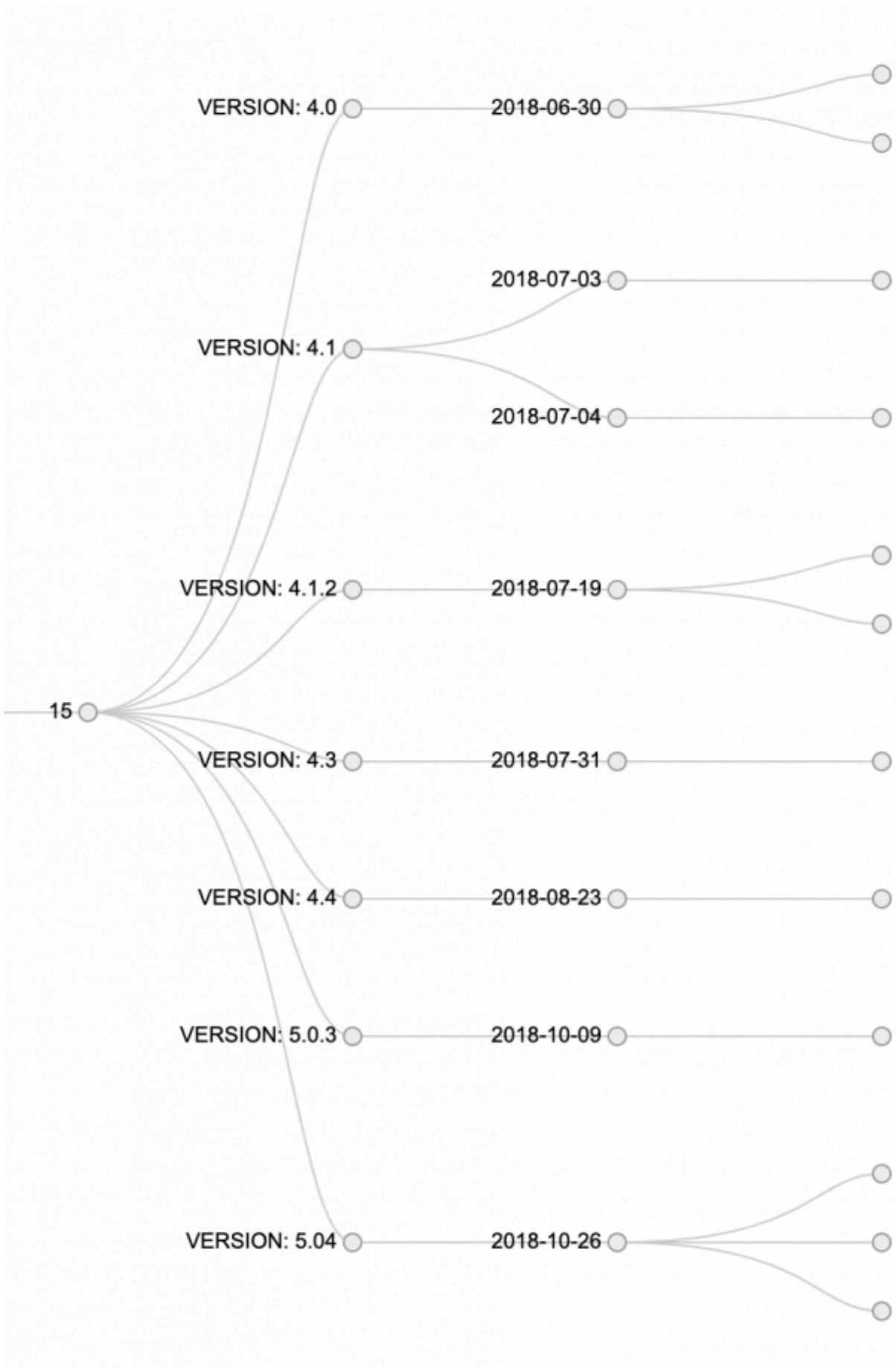


Figure 16: Activity of ID number 15, from version 4.0 to version 5.04.

Determining top IDs/affiliates

When we applied the theory that the ID corresponded with an affiliate, we observed different activity amongst the affiliates. There are some affiliates/IDs that were only linked to a single sample that we found. In some cases that specific sample was only found on a single source, like *VirusTotal*, and there were not any detections on our *McAfee* backend. This could occur when a sample was not spread, for instance, to a *McAfee*-protected system or if a single sample was, perhaps, indicative of a security researcher, undercover as an affiliate, only uploading their sample to *VirusTotal*. Another reason why affiliates might appear only for a short period is failure to perform. The GandCrab developers had a strict policy of expelling affiliates that underperformed. Expelling an affiliate would open a new slot that would receive a new incremented ID number.

On the other hand, we observed several very active affiliates, of which ID number 99 was by far the most active. We first observed ID 99 in six different samples of version 4.1.1, growing to 35 different samples in version 5.04. Based on our dataset we observed 71 unique unpacked samples linked to ID 99.

Being involved with several versions (consistency over time), in combination with the number of unique samples (volume) and the number of infections (based on industry malware detections) could effectively show which affiliate was the most aggressive and possibly the most important to the RaaS network.

This can be compared to a top sales person in any normal commercial organization. Given that the income of the RaaS network is partly dependent on the performance of its top affiliates, disrupting a top affiliate would have a crippling effect on the income of the RaaS network, internal morale and overall RaaS performance.

SUB_ID role

Based on the child relationship of the SUB_ID we believe that this number might represent a build number or a method for the affiliate to run its own partner program for other individuals. Unfortunately, based on the information available, we are unable to determine its role with absolute certainty at this time.

Overview versions and ID numbers

Using an online tool called *RAWGraphs* [25] we created an alluvial graphic display of the entire dataset, showing the relationship between the versions and the ID numbers. This is shown in Figure 17 – a more detailed overview can be supplied on request.

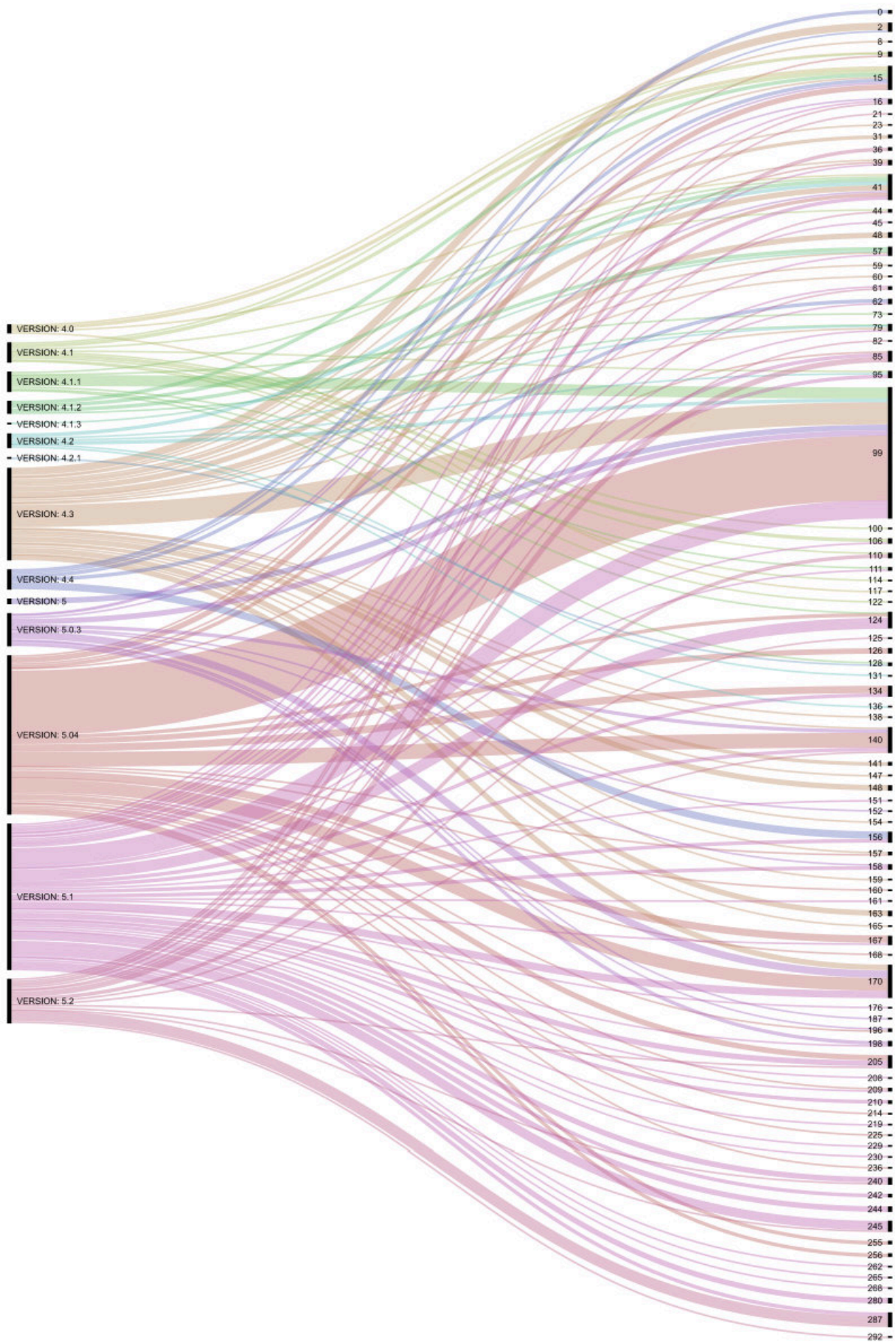


Figure 17: Overview of versions and IDs.

Top performing affiliates immediately stood out from the rest as the lines were thicker and more spread out. Information like this can help law enforcement decide where to focus their valuable resources. From a security industry perspective, affiliate analysis will further ensure chain of custody, since a direct link from victim, sample and responsible affiliate can be drawn.

Top affiliates missing in 5.2

When looking at the overview it does stand out that none of the top affiliates/ID numbers were present in version 5.2. We are unable to explain the exact cause of the absence of these IDs, but this might have been an early indicator that the end of GandCrab was imminent.

Combining the sample timeline with a timeline of forum postings

One of the other key factors that made GandCrab popular was its forum presence and marketing strategy. Every new version was announced in a grand fashion (see Figure 18), almost comparable to those of major software companies.



Figure 18: Announcement of GandCrab v5.

The timely announcements of new versions on the Exploit.in forum offered a way of checking if the compile times and hard-coded version numbers matched up with the announcements by the actor. Based on all the samples we collected, only four had a different compile date; the rest of the samples found had a compile date a couple of days before the new version announcement, or shortly after. We proceeded to add the timestamps of the announcements to the sample timeline and colour-coded them in yellow, as shown in Figure 19.

2018-08-29		VERSION: 4.4	1caaa8248f56c8d0bd592863d20f4d51
2018-08-29		VERSION: 4.4	c86704ab231c213a55564080f072323b
2018-08-29		VERSION: 4.4	1caaa8248f56c8d0bd592863d20f4d51
2018-08-29		VERSION: 4.4	c86704ab231c213a55564080f072323b
2018-09-24		VERSION: 5	96ead54f6aacd7c40e2d060cb303fa83
2018-09-24		VERSION: 5	884f86d79065d97244eea7ab68b129ce
2018-09-24		VERSION: 5	e168e9e0f4f631bafc47ddf23c9848d7
2018-09-27	Version 5 Forum announcement		
2018-10-09	Versions 5.0.3		
2018-10-09		VERSION: 5.0.3	f71ba2b07bd5041976385ce01deb1c42
2018-10-09		VERSION: 5.0.3	c5adb96a16aec327e9ff24ec947cfc78
2018-10-09		VERSION: 5.0.3	5df167c9027489bacc5a43ff9b769d6b
2018-10-09		VERSION: 5.0.3	5902ebd8b4aade42654b249ee3c615b5
2018-10-09		VERSION: 5.0.3	d40a530582e67ed1e8f7fa46cd4049d6
2018-10-09		VERSION: 5.0.3	87ac8c14c76a7c240fd7f03a847ac3e4
2018-10-09		VERSION: 5.0.3	10aa719c5ac18719e2dbcf4d86be4ae9
2018-10-09		VERSION: 5.0.3	310bd85d449eef8470b2b7135802893d

Figure 19: GandCrab forum announcement of version 5.03.

In addition to version announcements, the GandCrab Exploit.in forum thread also formed a lively discussion platform for individuals supporting and interested in the RaaS. Several forum users posted openly about their affiliation with GandCrab and spoke highly of the profits they earned.

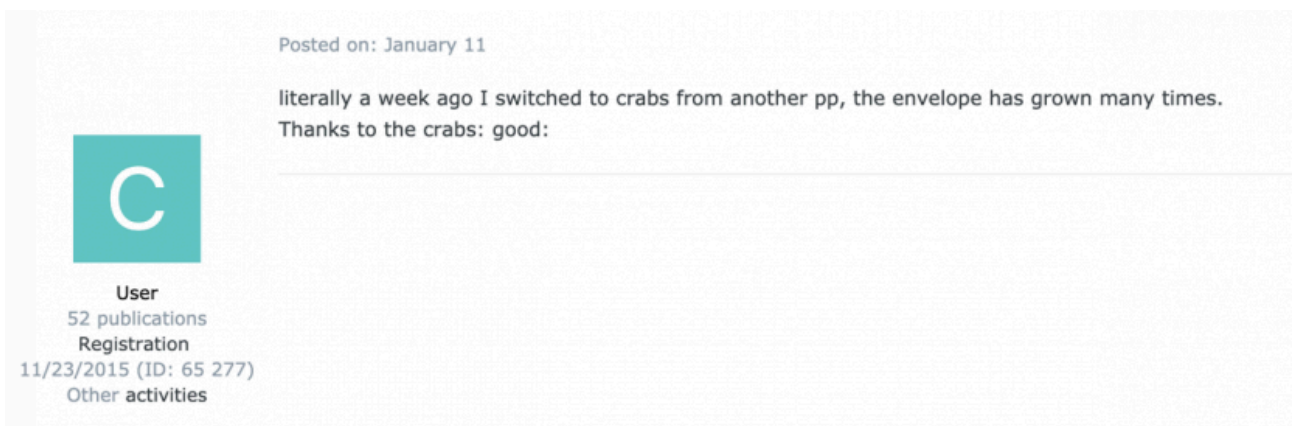


Figure 20: Forum posting of a user stating their affiliation with GandCrab ransomware.

This type of endorsement bears similarities to the social marketing of direct sales-based products. We are uncertain as to whether the affiliates endorsing GandCrab were doing this out of free will or if it was part of an internal marketing scheme. Overall, looking at the forum postings, one cannot help but notice that GandCrab gained a cult status amongst its followers.

Colour coding and affiliate research

We added all the relevant forum thread postings to the existing timeline and colour-coded the postings based on their content. Forum users that stated that they were an affiliate were coded blue. Posts expressing positive sentiment towards GandCrab were colour-coded green and those expressing negative sentiment were coloured red. By adding the forum postings, the timeline gave an accurate representation of the actors interested in GandCrab and the evolution of the actual ransomware over time.

2018-08-28	Deleting inactive users and sept 1st changing to old slot model. Version 5 mentioned		
2018-08-29		VERSION: 4.4	156
2018-08-29		VERSION: 4.4	156
2018-08-29		VERSION: 4.4	156
2018-08-29		VERSION: 4.4	156
2018-09-10	Talking about Fallout EK integration		
2018-09-15	interested in joining		
2018-09-15	Affiliate		
2018-09-20	Affiliate		
2018-09-23	Affiliate		
2018-09-24		VERSION: 5	99
2018-09-24		VERSION: 5	99
2018-09-24		VERSION: 5	99
2018-09-25	Affiliate		
2018-09-27	Affiliate running Dedic/RDP infections?		
2018-09-27	NTCrypt group announcement that they are working with Gandcrab		
2018-09-27	Version 5 Forum announcement		
2018-09-27	Affiliate		
2018-09-27	talking about upgrading the EK		
2018-10-02	Affiliate running Dedic/RDP infections?		
2018-10-09	Versions 5.0.3		
2018-10-09		VERSION: 5.0.3	152
2018-10-09		VERSION: 5.0.3	198
2018-10-09		VERSION: 5.0.3	198

Figure 21: Timeline of version 5.03 announcement, enhanced with colour-coded forum postings.

Continued analysis

The colour-coded timeline overview provided several options for deeper affiliate research:

- A user stating that they were affiliated with the RaaS at a moment X in time could serve as a reference marker that the person must have joined the RaaS prior to their statement. When we compared the statement timestamp to the ransomware ID numbers around the same time, we could estimate that the user must have had an ID number that was no higher than the highest ID number at the time of their statement. This offered a coarse method to link affiliates to a reduced number of ransomware samples.
- Creating an overview of all the interests per affiliate/username on this and other forums gave us insight into a person’s interests (good or bad), which could be used for identification of the individual. Subsequently, analysing previous user activity could provide insight into their cybercriminal skill progression. Generally speaking, cybercriminals make more operational security (OpSec) mistakes earlier on in their career.

The end of GandCrab

On Friday, 31 May 2019, the GandCrab crew released a statement saying that they were closing their business. That a RaaS was closing was not unusual, but GandCrab did it in a fashion true to its nature – overt, and with a lot of bravado.

The screenshot shows a forum post from a user named 'Gandcrab' with a profile picture of a purple and black logo. The post is titled 'All the good things come to an end.' and contains the following text:

Posted Friday at 09:44 PM

All the good things come to an end.
 For the year of working with us, people have earned more than **\$ 2 billion**, we have become a nominal name in the field of the underground in the direction of crypto-fiber. Earnings with us per week averaged **\$ 2,500,000** .
 We personally earned more than **150 million** dollars per year. We successfully cashed this money and legalized it in various spheres of white business both in real life and on the Internet.
 We were glad to work with you. But, as it is written above, all good things will ever end.

We are leaving for a well-deserved retirement . We have proven that by doing evil deeds, retribution does not come. We proved that in a year you can earn money for a lifetime. We have proven that it is possible to become number one not in our own words, but in recognition of other people.

In this regard, we:

1. Stop the set of adverts;
2. We ask the adverts to suspend the flows;
3. Within 20 days from this date, we ask adverts to monetize their bots by any means;
4. Victims - if you buy, now. Then your data no one will recover. Keys will be deleted.

That's all. The topic will be deleted in a month. Thank you all for the work.

Figure 22: The GandCrab crew announces the end of its operations.

Looking closely at this statement, there are some interesting observations to be made:

'We successfully cashed this money and legalized it in various spheres of white business both in real life and on the Internet.'

This means that they could have had a money laundering system in place to mix their earnings from the criminal underworld with legitimate businesses.

'For a year of working with us, people have earned more than \$ 2 billion.'

'We personally earned more than 150 million dollars per year.'

We think that this amount is largely exaggerated. Over the year, the *NoMoreRansom* decryptors helped a lot of victims to get their files back and prevented millions of dollars from falling into the hands of the GandCrab crew. However, based on the number of infections, we do believe that the individuals would have enough money to retire. Subsequently, our observations of the top affiliates being absent in version 5.2 might indicate an internal issue as a reason to stop.

'We have proven that by doing evil deeds, retribution does not come.'

This again emphasizes the strong sense of impunity felt by the GandCrab crew and its affiliates. This is a worrisome thought since the space that GandCrab left will probably be filled quickly by a new RaaS system. Punishment always come after the crime, but we hope that, in this case, it will come sooner rather than later.

'Victims – if you buy, now. Then your data no one will recover. Keys will be deleted.'

This got a bit mangled through *Google Translate* but the original statement urged victims to pay up because the GandCrab crew were planning to delete all the decryption keys. It is kind of strange that the self-proclaimed millionaires did not show the slightest compassion. This statement evoked a lot of reaction in the forum thread where other well-respected users urged publication of the remaining keys to the public.



Figure 23: Forum moderator urging release of the remaining keys.

Eventually, the GandCrab account requested suspension and deletion of all posts on the forum. The moderators did suspend the account but left the posts intact.

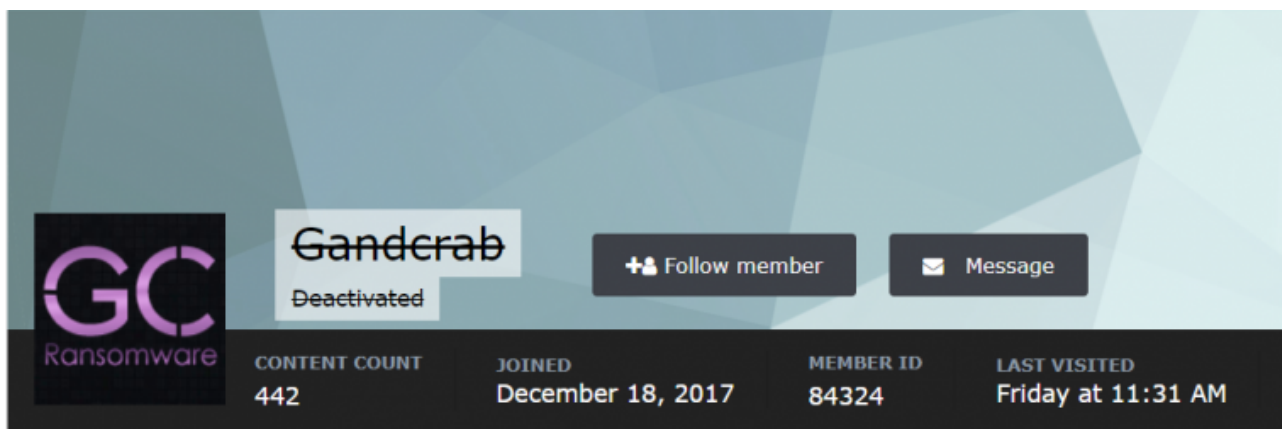


Figure 24: Deactivation of the GandCrab account.

Conclusion

We started our research on GandCrab by carefully dissecting the malware and discovering its inner workings and secrets. The hard-coded indicators in the malware and GandCrab crew’s forum presence led us to dig deeper into the inner workings of the RaaS model. By doing so we gained some valuable insights:

- Successful ransomware does not have to be the best coded ransomware, but the developers do have to be agile.
- In order to grow, a RaaS model needs good accounting – to make sure that everyone gets their share.
- In order to grow, a RaaS system needs strong alliances with complimentary services to drive infections and profits.
- The success of a RaaS system is dependent on its affiliates; strong affiliates have a large influence. Disruption of top affiliates can have a crippling effect on the income, morale and overall success of the RaaS.
- A timeline analysis of a RaaS system can offer a method to single out top affiliates and spot potential events early on.

As an industry we must realize that we cannot stop cybercrime alone; we should aim to do more than just malware analysis, especially when it comes to fighting RaaS-type threats. Unfortunately, we live in a situation where most of the cybercriminals involved in ransomware can operate with a level of impunity; the ransomware developers are often in countries that make legal prosecution difficult and affiliates that are not caught can easily move from one RaaS to another and continue their extortion operations.

Law enforcement faces a daunting task to bring the individuals responsible to justice, but our industry’s knowledge, data and tooling can help with this task. The best way to cook a Crab is together.

GandCrab might have shut down its operations but its developers and affiliates have still not been arrested and will probably continue to be active in cybercrime in one way or another.

References

- [1] Montenegro, D (@CryptoInsane). Twitter status. 26 January 2018. <https://twitter.com/CryptoInsane/status/956803455833853952>.
- [2] Caltagirone, S.; Pendergast, A.; Betz, C. The Diamond Model of Intrusion Analysis. <https://apps.dtic.mil/dtic/tr/fulltext/u2/a586960.pdf>.
- [3] MalwareHunterTeam (@malwrhunterteam). Twitter presence. <https://twitter.com/malwrhunterteam>.
- [4] MalwareHunterTeam (@malwrhunterteam). Twitter status. 5 March 2018. <https://twitter.com/malwrhunterteam/status/970746661231263745>.
- [5] nao_sec (@nao_sec). Twitter status. 23 April 2018. https://twitter.com/nao_sec/status/988451194573017088.
- [6] Jawe (@zsawei). Twitter status. 10 May 2018. <https://twitter.com/zsawei/status/994454718406578176>.
- [7] NoMoreRansom. <https://www.nomoreransom.org/>.
- [8] Orthodox New Year 2019. Calendar Date.com. https://www.calendardate.com/orthodox_new_year_2019.htm.
- [9] EC3 (@EC3Europol). Twitter status. 19 March 2019. <https://twitter.com/EC3Europol/status/1107984687253868545>.
- [10] Cimpanu, C. Vaccine Available for GandCrab Ransomware v4.1.2. Bleeping Computer. 19 July 2018. <https://www.bleepingcomputer.com/news/security/vaccine-available-for-gandcrab-ransomware-v412/>.
- [11] CVE-2018-8440. GitHub. <https://github.com/sourceincite/CVE-2018-8440>.
- [12] SandboxEscaper (@sandboxescaper). Twitter presence. <https://twitter.com/sandboxescaper> (suspended by Twitter).
- [13] CVE-2018-8120. GitHub. <https://github.com/unamer/CVE-2018-8120>.
- [14] Leeqwind. Win32k NULL-Pointer-Dereference Analysis by Matching the May Update. <https://xiaodaozhi.com/exploit/156.html>.
- [15] Threat Landscape Dashboard CVE-2018-8120. McAfee. <https://www.mcafee.com/enterprise/es-es/threat-center/threat-landscape-dashboard/vulnerabilities-details.cve-2018-8120.html>.
- [16] ValtheK. AntiCrab. <http://29wspy.ru/reversing/AntiCrab.zip>.
- [17] ValtheK. AntiCrab32. <http://29wspy.ru/reversing/AntiCrab32.zip>.
- [18] ValtheK. AntiCrabWithoutPersistenceAndRemoveWallpaper32. <http://29wspy.ru/reversing/AntiCrabWithoutPersistenceAndRemoveWallpaper32.zip>.
- [19] ValtheK. GandCrabSucksVaccine. <http://29wspy.ru/reversing/GandCrabSucksVaccine.zip>.
- [20] ValtheK. GandAtom. <http://29wspy.ru/reversing/GandAtom.zip>.

[21] van der Sandt, E. Deviant Security: The Technical Computer Security Practices of Cyber Criminals. 2019. https://research-information.bristol.ac.uk/files/194364696/DEVIANT_SECURITY_EHAVANDESANDT.pdf.

[22] Europol. Goznym malware: cybercriminal network dismantled in international operation. 16 May 2019. <https://www.europol.europa.eu/newsroom/news/goznym-malware-cybercriminal-network-dismantled-in-international-operation>.

[23] Mundo, A.; Fokker, J.; Roccia, T. Rapidly Evolving Ransomware GandCrab Version 5 Partners With Crypter Service for Obfuscation. McAfee. 10 October 2018. <https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/rapidly-evolving-ransomware-gandcrab-version-5-partners-with-crypter-service-for-obfuscation/>.

[24] MITRE ATT&CK. Timestomp. <https://attack.mitre.org/techniques/T1099/>.

[25] RAWGraphs. <https://rawgraphs.io/>.

Source: <https://www.virusbulletin.com/virusbulletin/2019/11/vb2019-paper-different-ways-cook-crab-gandcrab-ransomware-service-raas-analysed-indepth/>