Malicious ratatouille 🐀

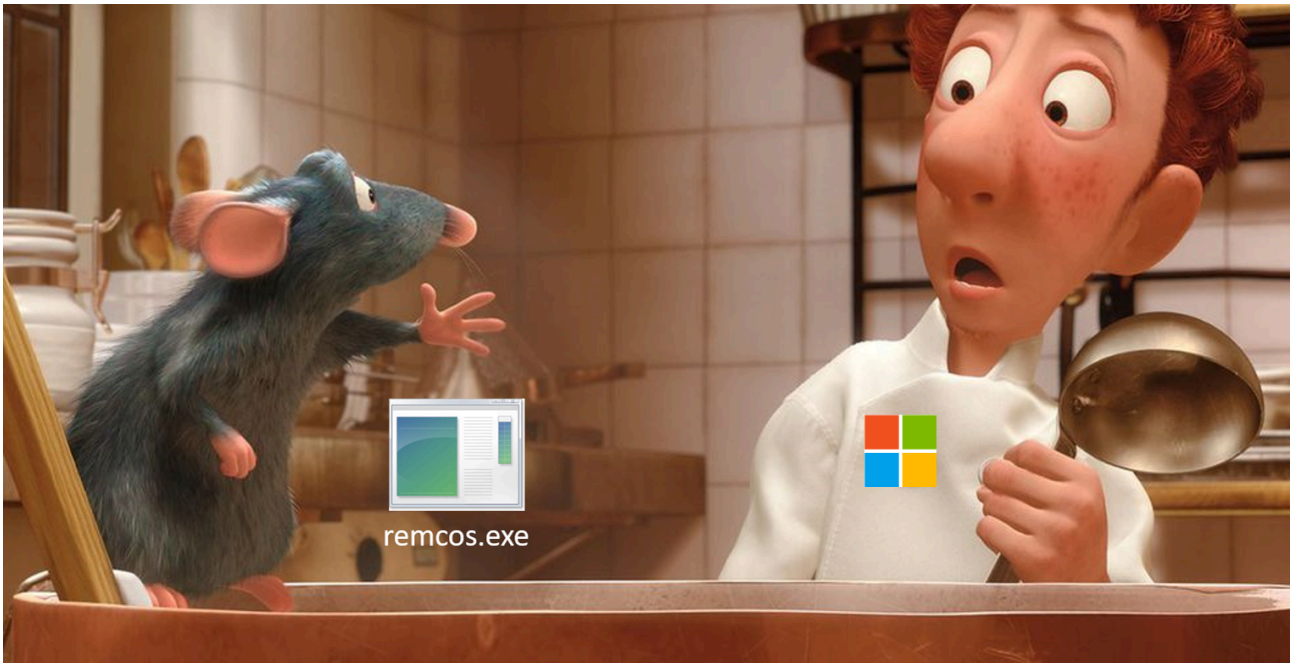
By f0wL

Published: 2019-09-07 · Archived: 2026-04-10 02:41:53 UTC

Sat 07 September 2019 in RATs

Remcos is a commercially sold Remote Adiministration Toolkit (RAT) that is regularly distributed as Spyware

Depending on the licensing model and capabilities Remcos is sold for 58$ to 389$ by the company (with the pretty fitting name) Breaking Security. Feature-wise the manfacturer's website lists: Remote Administration, Support, Surveillance, Anti-Theft and Proxy. In most cases the executable is dropped via a boobytrapped Office or XML Document. Of course I will not link to any of their webpages or products since shilling out for cybercriminals would be the last thing I'd do.



Inspiration for this blog post came from @wwp96 on Twitter:

> #remcos
>
> jkharding2014.myddns[.]rocks
> tomharry.ddns[.]net
>
> 2c8b1cca4ee54428dffc203b76c4dc30 - Dhl protected.iso
> 06469856a9bdecae989b64daf9db09c7 - carved exehttps://t.co/YtsJYbhle9
>
> — wwp96 (@wwp96) September 7, 2019

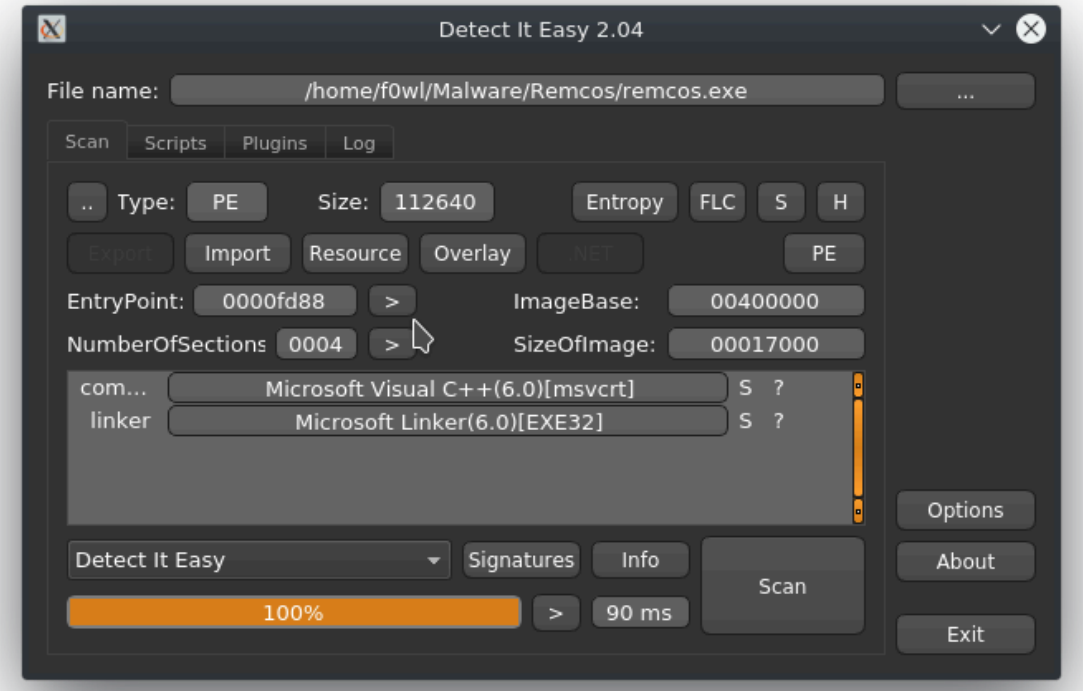
Remcos uses a Control instance (the C&C) and the so-called Agent (the executable that is delivered to the victim). It was first spotted in 2016 when it was being sold on HackForums. Since then it was being used in targeted attacks (mostly spear-phishing) against turkish government/military contractors or other businesses/individuals in the European Union. The Agent is written in C++ (while the Control application is written in Borland Delphi) and is 110KB in size. Click here for the [AnyRun Analysis](#).

```
WRITE   Key:    HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
+344ms  Name:   remcos
        Value:  "C:\Users\admin\AppData\Roaming\remcos\remcos.exe"

WRITE   Key:    HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap
+578ms  Name:   UNCAsIntranet
        Value:  0

WRITE   Key:    HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap
+578ms  Name:   AutoDetect
        Value:  1
```

Of course it fiddles around in the registry as well. It uses the Key in HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run to bind to the system startup.



Although there are versions of Remcos that are packed with UPX and MPRESS1 this sample is not obfuscated in any way.

```
    27 3.459844    192.168.100.96   192.168.100.2    DNS      86 Standard query 0xc4da A jkharding2014.myddns.rocks
    28 3.565811    192.168.100.2    192.168.100.96   DNS     102 Standard query response 0xc4da A jkharding2014.myddns.rocks A 66.154.113.142
    35 3.881521    192.168.100.96   192.168.100.2    DNS      77 Standard query 0xeccb A tomharry.ddns.net
    36 3.900117    192.168.100.2    192.168.100.96   DNS      93 Standard query response 0xeccb A tomharry.ddns.net A 66.154.113.142
```

In terms of network interactions it queries two Dynamic DNS URLs that both point to the same host at *66.154.113[.]142*

```
int32_t v16 = g1 ? -1 : 1; // 0x40fb7a
int32_t v17 = (int32_t)"1.7 Pro"; // 0x40fb83
int32_t v18; // 0x40fb85
```

With Version 1.7 Pro we've got an old Version of the RAT in our hands which dates back to 5th of January 2017. The most recent version of the malware according to the changelog is V2.4.7. Another thing one usually doesn't get with malware: a 31-page manual. It goes over the features and configuation points the malware has to offer and even includes a "Terms of Service" chapter which states that *users have to be notified that there is surveillance software in place* and that *the use of remcos for illegal activities* is forbidden. As if they would care that their software was probably used in >95% of malicious acts. Judging by the typos and a few screenshots I'd attribute this malware to eastern european threat-actors.

The following Screenshots were captured after decompiling the executable with the retargetable Decompiler retdec by Avast. The decompiled result can be found here.

```
int32_t v56 = &v55; // 0x40674d
*(int32_t *)(g9 - 4) = (int32_t)"\\install.bat";
*(int32_t *)(g9 - 8) = v56;
*(int32_t *)(g9 - 12) = (int32_t)"Temp";
*(int32_t *)g9 = (int32_t)getenv((char *)&g196);
int32_t v57 = _3f__3f_0_3f__24_basic_string_40_DU_3f__24_char_trait
*(int32_t *)(g9 - 4) = v57;
int32_t v58 = &v25; // 0x40676d
*(int32_t *)(g9 - 8) = v58;
*(int32_t *)(g9 + 8) = function_40fc20();
_3f__3f_0_3f__24_basic_string_40_DU_3f__24_char_traits_40_D_40_std_
_3f__3f_1_3f__24_basic_string_40_DU_3f__24_char_traits_40_D_40_std_
_3f__3f_1_3f__24_basic_string_40_DU_3f__24_char_traits_40_D_40_std_
*(int32_t *)(g9 - 4) = 1;
*(int32_t *)(g9 - 8) = 16;
int32_t v59 = _3f_c_str_40__3f__24_basic_string_40_DU_3f__24_char_t
*(int32_t *)(g9 - 4) = v59;
_3f__3f_0_3f__24_basic_ofstream_40_DU_3f__24_char_traits_40_D_40_st
*(int32_t *)(g9 - 4) = (int32_t)"PING 127.0.0.1 -n 2 \n";
*(int32_t *)(g9 - 8) = v2;
```

As a first step it runs its dropped install script called *install.bat* and uses a ping to localhost to stall the process and make sure it is finished before proceding.

```
// Address range: 0x401178 - 0x4011a3
int32_t function_401178(void) {
    int32_t phkResult = 0; // bp-8
    int32_t v1 = RegOpenKeyExA(HKEY_LOCAL_MACHINE, "HARDWARE\\ACPI\\DSDT\\VBOX__", 0, 0x20019, (int32_t **)&phkResult); // 0x401195
    return (int32_t)(v1 != 0) + 1 & 255 | -v1 & -256;
}
```

```
// Address range: 0x4010f1 - 0x401102
int32_t function_4010f1(void) {
    int32_t * moduleHandle = GetModuleHandleA("SbieDll.dll"); // 0x4010f6
    return (int32_t)(moduleHandle != NULL) | (int32_t)moduleHandle & -256;
}
```

In terms of Evasion techniques Remcos turns up with detection methods for both Virtualbox and Sandboxie.The above example shows the method it employs for Virtualbox via a registry key that is set if the Guest Additions are in place on the guest system. In the same manner it tries to call *SbieDll.dll* to check if Sandboxie is present.

```
*(int32_t *)(g9 - 20) = (int32_t)"Initializing connection to C&C...\n";
*(int32_t *)(g9 - 24) = v11;
*(int32_t *)(g9 - 28) = (int32_t)"%02i:%02i:%02i:%03i [INFO] ";
```

The Remcos Agent also has debugging functionality via a console window, for example for the communication with the C&C Server.

```
int32_t v8 = 0; // bp-12
int32_t * hHandle = OpenMutexA(0x100000, false, "Remcos_Mutex_Inj"); // 0x407506
if ((int32_t)hHandle != g8) {
    // 0x407510
    WaitForSingleObject(hHandle, 0xea60);
}
// 0x40751c
*(int32_t *)(g9 - 4) = (int32_t)&v8;
*(int32_t *)(g9 - 8) = (int32_t)"Inj";
int32_t v9 = _3f_c_str_40__3f__24_basic_string_40_DU_3f__24_char_traits_40_D_40_std_40__40_V_3f__24_allocator
*(int32_t *)(g9 - 4) = v9;
*(int32_t *)(g9 - 8) = -0x7fffffff;
int32_t v10 = function_408e97((int32_t)&g196, (int32_t)&g196, (int32_t)&g196, (int32_t)&g196); // 0x407533
int32_t v11; // 0x40755a
if ((char)v10 != 0) {
    // 0x40753f
    *(int32_t *)(g9 + 12) = (int32_t)"Inj";
    int32_t v12 = _3f_c_str_40__3f__24_basic_string_40_DU_3f__24_char_traits_40_D_40_std_40__40_V_3f__24_all
    *(int32_t *)(g9 - 4) = v12;
    *(int32_t *)(g9 - 8) = -0x7fffffff;
    function_409132((int32_t)&g196, (int32_t)&g196, (int32_t)&g196);
    v11 = g9 + 12;
} else {
    v11 = g9 + 16;
}
```

Remcos also employs Process Injection via a static Mutex. This behaviour is often used as a simple way of achieving persistence and to decrease the risk of a possible detection. Most versions of the RAT seem to inject into *svchost.exe*.

```
if ((char)v32 == 0) {
    // 0x40a999
    v6 = "filemgr";
    str_as_i = (char *)&v26;
    if ((char)_3f__3f_8std_40__40_YA_NABV_3f__24_basic_string_40_DU_3f__24_char_traits_40_D_40_std_40__40_
        // 0x40a9cd
        v6 = "downloadfromurltofile";
        str_as_i = (char *)&v26;
        if ((char)_3f__3f_8std_40__40_YA_NABV_3f__24_basic_string_40_DU_3f__24_char_traits_40_D_40_std_40_
            // 0x40aa4e
            v6 = "downloadfromlocaltofile";
            str_as_i = (char *)&v26;
            if ((char)_3f__3f_8std_40__40_YA_NABV_3f__24_basic_string_40_DU_3f__24_char_traits_40_D_40_st
                // 0x40ab1c
                v6 = "getproclist";
                str_as_i = (char *)&v26;
                char * v33 = v6; // 0x40ab2a
                if ((char)_3f__3f_8std_40__40_YA_NABV_3f__24_basic_string_40_DU_3f__24_char_traits_40_D_4
                    // 0x40abd2
                    v6 = "prockill";
                    str_as_i = (char *)&v26;
                    if ((char)_3f__3f_8std_40__40_YA_NABV_3f__24_basic_string_40_DU_3f__24_char_traits_40_
                        // 0x40ac05
                        v6 = "getwindows";
                        str_as_i = (char *)&v26;
                        if ((char)_3f__3f_8std_40__40_YA_NABV_3f__24_basic_string_40_DU_3f__24_char_traits
                            // 0x40ac20
                            v6 = "closewindow";
                            str_as_i = (char *)&v26;
                            if ((char)_3f__3f_8std_40__40_YA_NABV_3f__24_basic_string_40_DU_3f__24_char_t
                                // 0x40ac53
                                v6 = "maxwindow";
                                str_as_i = (char *)&v26;
                                if ((char)_3f__3f_8std_40__40_YA_NABV_3f__24_basic_string_40_DU_3f__24_cha
                                    // 0x40ac88
                                    v6 = "restorewindow";
                                    str_as_i = (char *)&v26;
                                    if ((char)_3f__3f_8std_40__40_YA_NABV_3f__24_basic_string_40_DU_3f__24
                                        // 0x40ac9d
                                        v6 = "closeprocfromwindow";
                                        str_as_i = (char *)&v26;
```

Via the command & control structure we also get a pretty good look at all the features the malware supports. In this screenshot we can see the file operations, process manipulations and window interactions it has to offer to the operator.

```
int32_t v82 = v10; // 0x405a30
char v83 = v82; // 0x405a30
if (v79 != v83) {
    // 0x405a38
    if (g127 != v83) {
        // 0x405ab2
        *(int32_t *)(g9 - 4) = v82;
        *(int32_t *)(g9 - 24) = v3;
        *(int32_t *)(g9 - 28) = (int32_t)"\n[Firefox StoredLogins cleared!]";
        g2 = _3f__3f_0_3f__24_basic_string_40_DU_3f__24_char_traits_40_D_40_std_40__40_V_3f
        g5 = &g116;
        function_403a9a((int32_t)&g196);
        v10 = v10 & -256 | 1;
        goto lab_0x405ad3;
```

Another "standard" feature for RATs is accessing Browser History, cache and password stores. In this case Remcos is trying to manipulate user data in Mozilla Firefox.

```
else {
  // 0x401d21
  g5 = v4;
  function_401289(1);
  int32_t v7 = function_4052ec(_3f_data_40__3f__24_basic_string_40_DU_3f__24_char_traits_40_D_40_s
  g8 = v7;
  if (v7 != 0) {
      int32_t v8 = function_405546(v7, "OpenCamera"); // 0x401d4a
      v1 = (char *)v7;
      g102 = v8;
      g103 = function_405546(v7, "CloseCamera");
      g104 = function_405546(v7, "GetFrame");
      g105 = function_405546(v7, "FreeFrame");
      g8 = v2;
      g100 = 1;
      function_40fc1a();
      function_40fc14();
      g5 = a1;
      function_402198();
      _3f__3f_1_3f__24_basic_string_40_DU_3f__24_char_traits_40_D_40_std_40__40_V_3f__24_allocator
  }
  goto lab_0x401f1a;
```

We also get a Look at the webcam capture module of the RAT which seems to support different camera modes. Additionally it also supports audio capture via a built-in microphone.

```
// Address range: 0x40ca41 - 0x40ca9b
int32_t function_40ca41(void) {
    int32_t * processHandle = GetCurrentProcess(); // 0x40ca4e
    OpenProcessToken(processHandle, (int32_t)&g196, (int32_t **)&g196);
    int32_t lpLuid; // bp-20
    LookupPrivilegeValueA(NULL, "SeShutdownPrivilege", (struct _LUID *)&lpLuid);
    int32_t NewState = 1; // bp-24
    int32_t TokenHandle; // bp-8
    AdjustTokenPrivileges((int32_t *)TokenHandle, false, (struct _TOKEN_PRIVILEGES *)&NewState, 0, NULL, NULL);
    return GetLastError() != 0;
}
```

Lastly the malware also has the capabilities to manipulte the system power state depending on the current priviledges.

Although Remcos is not a "new" malware by today's definition it is still a serious threat to look out for. In my test it scores *53/68* on VirusTotal.

IOCs

Remcos RAT (SHA256)

```
1c3a298dd32da9de457842613dd4f07e0e57131a94bc13d868ffcbbebfab6d63
11535ea0ba3bf9ed0691b850955ef2613475dfdce7d8a32fa3d2d7ae066de73d
```

C&C URLs

```
httx://tomharry.ddns[.]net
httx://jkharding2014.myddns[.]rocks
```

```
httx://gratefulheart.ddns[.]net
httx://uaeoffice999.warzonedns[.]com
```

IPs

```
66.154.113[.]142
79.134.225[.]77
79.134.225[.]81
```

Modified Registry Keys

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
remcos --> "C:\Users\admin\AppData\Roaming\remcos\remcos.exe"

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap
UNCAsIntranet --> 0
AutoDetect --> 1
```
