

THREAT RESEARCH

AI RedScam: Infostealer Malware in AI Offensive Tools



Services Performed By:

UltraViolet Cyber TIDE Team
tide@uvcyber.com

Published Date:

November 25, 2025

TLP:AMBER-STRICK



Executive Snapshot

UltraViolet Cyber (UVCyber) Threat Intelligence and Detection Engineering (TIDE) Team have worked closely with the UVCyber Application Security Testing (AST) Network Security Team to uncover, analyze, and reverse engineer a novel social engineering and malware campaign directed at Red Team and Offensive Cybersecurity professionals. TIDE and AST Teams have dubbed this newly discovered campaign “AIRedScam”. This novel SmartLoader campaign closely follows the Tactics, Techniques, and Procedures (TTPs) of similar SmartLoader and Infostealer deployments, where a known technical search-engine topic is selected and then leveraged by an attacker through an official looking Git repository to con junior to mid skill personnel into downloading and deploying malware. Normally, the search topics selected by threat actors running SmartLoader campaigns like this are surrounding Software License Cracks, Systems and Network Administration Automation Tasks, and Chat Service Bots. What sets AIRedScam apart is its choice in targeting Offensive Cybersecurity professionals looking for tools that can automate their enumeration and recon.

The “AIScan-N” tool, originally built by the Chinese Developer “SecNN” to provide a reverse Model Context Protocol (MCP) service, was repurposed by the threat actor running AIRedScam. “SecNN/AIScan-N” does appear to be suspicious due to its use of precompiled binaries and installation methods. However, “AIScan-N” was deemed out of scope for the purposes of this report.

This report covers initial discovery, social and business intelligence analysis, dynamic malware analysis, and solutions for enterprise environments to protect against this social engineering and malware threat moving forward. Please reach out to your UVCyber Account Executive if you have questions or concerns about AIRedScam or need any clarification on the contents of the technical assessment.



TIDE and AST Team Strategic Analysis

SmartLoader is a modular, stealth-focused malware loader that surged in prevalence lately due to its flexible architecture, heavy use of LuaJIT scripting, and broad distribution across compromised GitHub repositories. Threat actors have increasingly used it as a delivery platform for high-value payloads—including RedLine, Rhadamanthys, Lumma, and other credential and data-theft tools—by embedding it within ostensibly legitimate utilities, gaming “mods,” and automation scripts. SmartLoader’s design allows it to perform extensive host reconnaissance, environmental checks, and staged execution before retrieving live payloads from attacker-controlled infrastructure. Its combination of obfuscation, evasive execution logic, and multi-stage Lua components makes it particularly difficult for traditional detection engines to analyze and classify at scale.

LuaJIT gives SmartLoader a powerful avenue for evading AV and EDR because it blurs the line between legitimate scripting and malicious execution while operating inside a highly dynamic runtime environment. Unlike traditional compiled malware, LuaJIT executes bytecode within an embedded Just-In-Time virtual machine, allowing malicious logic to be delivered in encrypted fragments, decoded in memory, and executed without ever writing a stable binary artifact to disk. This JIT-compiled execution model produces ephemeral machine code that changes across runs, undermining static signatures and reducing the usefulness of behavioral baselines built on repeatable opcode patterns. Additionally, LuaJIT’s foreign function interface (FFI) allows the script to call native Windows APIs directly—without spawning suspicious subprocesses—enabling reconnaissance, injection, and payload staging to occur inside what appears to be a benign scripting engine. The result is a loader that is far more difficult for endpoint tools to fingerprint, trace, or confidently classify, particularly when paired with SmartLoader’s environment checks and modular staging logic.

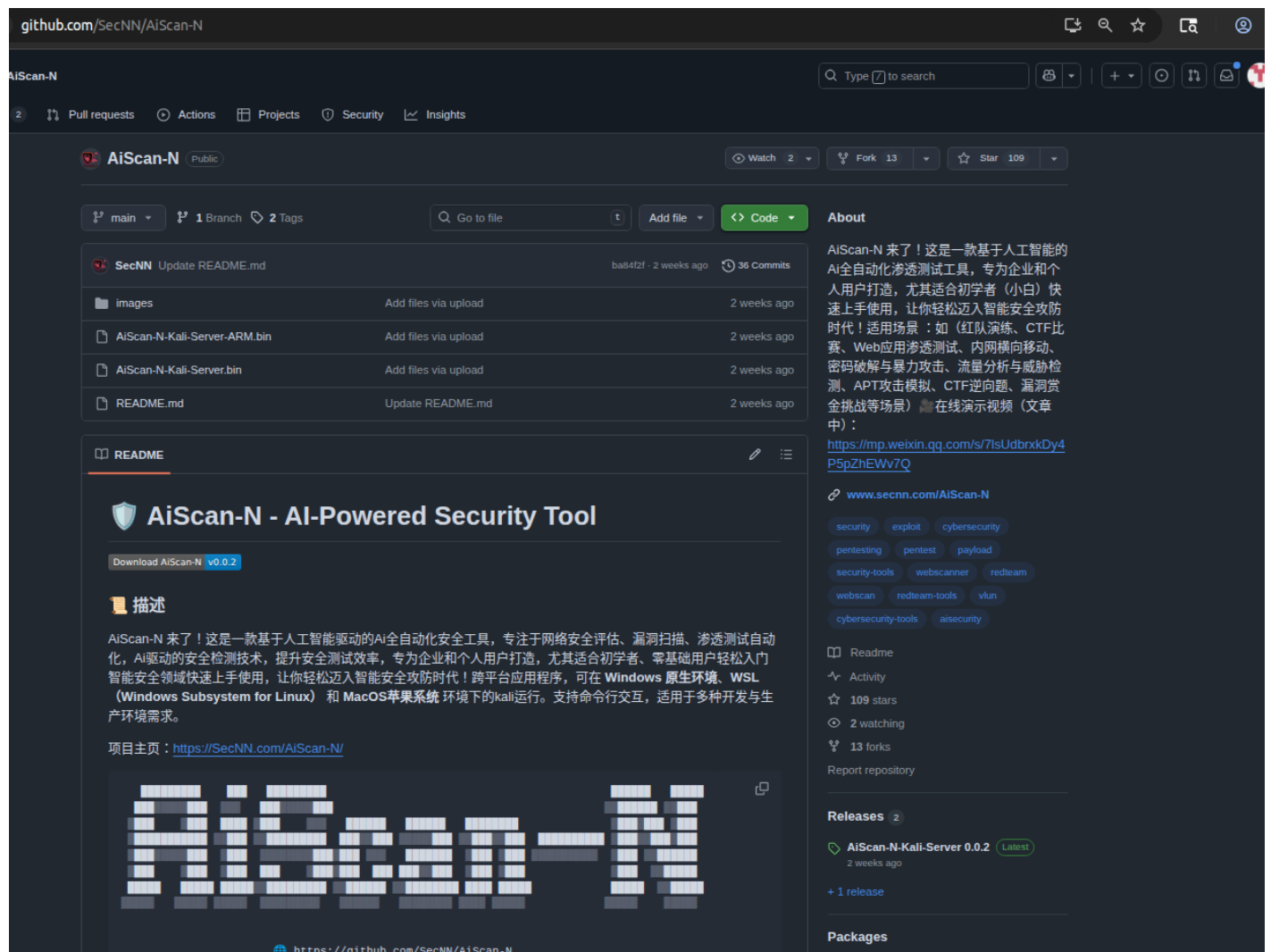
Throughout 2025, SmartLoader became a favored initial-access mechanism for both financially motivated groups and state sponsored threat actors because its distribution pipeline and staging logic were easily adapted across multiple campaigns. The move toward supply-chain style seeding—particularly through GitHub projects, third-party mirrors, and SEO-driven lure content—expanded its reach into enterprise developer environments and administrative workstations. Detection complexity increased as adversaries continually refreshed repository control using burner identities, rotated delivery artifacts, and embedded SmartLoader into rapidly updated open-source project forks. For enterprise defenders, the threat now represents a persistent infiltration vector that blends code-repository abuse, malware-as-a-service payload delivery, and highly adaptive staging behavior, requiring tightened developer hygiene, enhanced repository provenance controls, and continuous validation of downloaded tooling across engineering and IT teams.



Technical Analysis

UVCyber AST discovered the AIRedScam repository (<https://github.com/Rbxolexc8405/AiScan-N>) through their ongoing surveillance of the “AI Enabled Agentic Software Utilities” space. The offending repository was shared with TIDE Team for further review. On November 21st, 2025, both teams came to consensus that this was in fact a malicious Github Repository due to the following traits:

- The Repo commit chain shows the original Chinese language README.md being repurposed and changed to suit the attackers needs.
- README.md contains links that all go to the same path, including links that should normally point to official Github features; in this case all links pointed to the compressed archive which holds the malware payload.
- README.md was clearly AI generated, due to the rampant use of Emojis and language used.
- Hosting a much older version of the original SecNN/AIScan-N tool.



The above screenshot shows the original and current SecNN/AIScan-N repository.

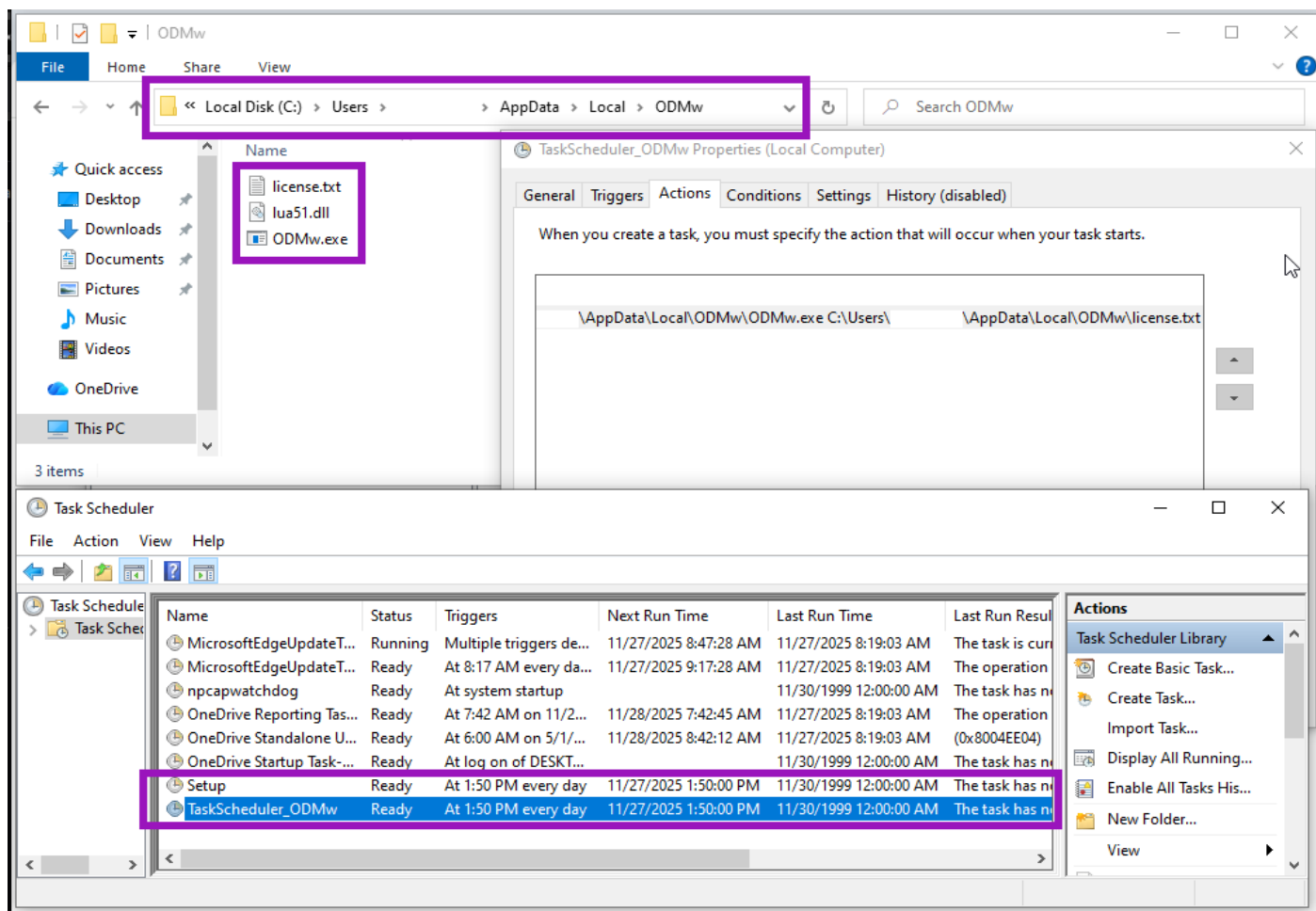


The above screenshot demonstrates the original Chinese language README.md from the original SecNN/AIScan-N repo being repurposed to target English speaking Offensive Cybersecurity personnel. Repository reuse through a fork, as demonstrated in the AIRedScam repository, is another hallmark of SmartLoader deployment and TTPs.

TIDE and AST Teams then downloaded the compressed malware archive and found four files inside. These files were:

- Launcher.cmd - A simple one-liner CLI command: “start luajit.exe license.txt”
- license.txt - The SmartLoader payload which is fired inside of the LuaJIT environment
- lua51.dll - A helper DLL for LuaJit.exe
- Luajit.exe - The LuaJIT binary itself

Using a tightly controlled and custom sandbox environment, TIDE and AST teams further confirmed that this payload was a variant of SmartLoader. A Windows 10.0.19045 build was selected for the Sandbox VM. Upon manually executing Launcher.cmd the luajit.exe binary immediately copied itself into the “C:\users\%USERNAME%\AppData\Local\ODMw\” directory, along with the license.txt and lua51.dll files. Immediately after copying itself, the SmartLoader payload running inside of the JIT VM creates two scheduled tasks by leveraging WinAPI calls. This scheduled task creation creates persistence in the infected host.



The above redacted screenshot captures the “Setup” and “TaskScheduler_ODMw” scheduled tasks created by the initial detonation of the payload inside of the high interaction Sandbox. Also shown is the copied SmartLoader payload, with *lua51.exe* being renamed to *ODMw.exe*. TIDE and AST teams found that the four-letter pattern is common for the SmartLoader family, all in caps, except for the last character. This four-letter code is generated when the LuaJIT payload is generated by the threat actor and will be used to track future SmartLoader infections by UVCyber TIDE and AST Teams and the UVCyber Managed Detection and Response (MDR) Team.

Alongside the Scheduled Task creation, a “Geo Fence” check is performed using the free “ip-api.com” service. This check ensures that the payload does not deploy and operate within the same country as the attacker. Then a series of plaintext HTTP requests were made to 85.209.129[.]236 on TCP/80. The 85.209.129.0/24 range is shown to be geographically located in Helsinki, Finland. The offending WAN IP is part of AS213702 as of November 2025. This ASN is owned by Qwins LTD, a Russian language Virtual Private Server (VPS) provider. Qwins LTD and the TIN/INN number listed on the website relate to a “Koshkin Matvey Vyacheslavovich” from Bolotnoye, Novosibirsk Oblast.

The Tax Payer ID Number (TIN) related to Koshkin Matvey Vyacheslavovich was established on March 27th, 2025. Other ASN ranges owned by QWINs were found to be geographically located in Germany, The Netherlands, Estonia, Poland, and the United Kingdom. These AS Ranges were all /24’s, which are often used by state sponsored threat actors to quickly rearrange their WAN facing infrastructure to best suit their own national interests. The upstream provider for the offending ASN Range is AS209693, “OC Networks LTD”, a Russian owned VPS provider known to take Monero/XMR as payment.



IP Koshkin Matvey Vyacheslavovich Отслеживать

[Summary](#) [Sanctions](#) [Story](#) [Types of activities](#) [Extract from the Unified State Register of Individual Entrepreneurs](#) [More](#)

Koshkin Matvey Vyacheslavovich

Citizenship
Russian Federation

OGRNIP
325547600053099
from March 27, 2025

Date of registration
March 27, 2025

Contacts
[Your company? Specify details](#)

[Summary report](#)
general information about the organization

[Extract from the Unified State Register of Individual Entrepreneurs](#)
with the signature of the Federal Tax Service on the required date

[Follow the organization](#)
[How](#) does it work and [why](#) is it needed?

Current as of 11/27/2025

Details

OGRNIP 325547600053099
Taxpayer Identification Number (INN) 541300541971
Type of business Sole proprietor
Date of registration March 27, 2025
Registrar Interdistrict Inspectorate of the Federal Tax Service No. 16 in Novosibirsk Oblast
Date of registration March 27, 2025
Name of the tax authority Interdistrict Inspectorate of the Federal Tax Service of Russia No. 23 for Novosibirsk Oblast

Rosstat data

OKPO 2041057920
OKATO 50206501000
OKTMO 50606101001

Information on registration with the Pension Fund

Registration number 1376658272
Date of registration March 28, 2025
Name of the territorial body Novosibirsk Region Office of the Pension and Social Insurance Fund of the Russian Federation

Special tax regime Новое

Applies Simplified Tax System

SME information

Date of inclusion April 10, 2025
Subject category Microenterprise

The above screenshot is an automated translation of the TIN information found attributed to the owner of Qwins LTD.

85.209.129.236 IP address Information

The IP address 85.209.129.236 was found in Helsinki, Uusimaa, Finland. , QWINS LTD. Additional IP location information, as well as network tools are available below.

IP address: **85.209.129.236**

Organization: QWINS LTD

ASN: AS213702

City: Helsinki

Region: Uusimaa (18)

Country: Finland (FI)

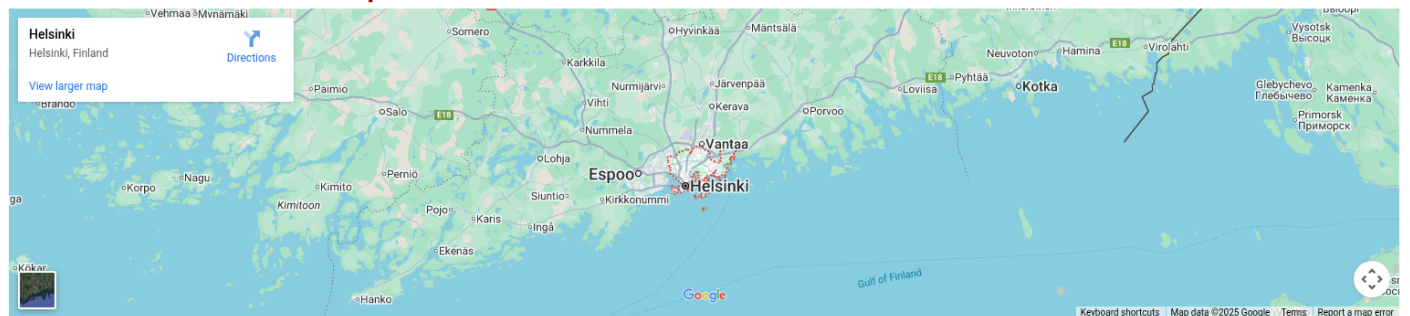
Postal code: 00121

latitude: 60.1719

longitude: 24.9347

[traceroute](#) [check latency](#) [whois](#) [BGP routing info](#)

85.209.129.236 Location Map



The above screenshot provides GeoINT on the location of the offending WAN IP.



File Machine View Input Devices Help

*Ethernet

File Edit View Go Capture Analyze

tcp.stream eq 3

No.	Time	Source
25	4.861356	10.0.2.15
26	4.874024	208.95.112.1
27	4.874240	10.0.2.15
28	4.874583	10.0.2.15
29	4.874759	208.95.112.1
30	4.890589	208.95.112.1
31	4.890679	10.0.2.15
1817	12.475821	10.0.2.15

Frame 28: Packet, 220 bytes on wire
Ethernet II, Src: PCSSystemtec14:8
Internet Protocol Version 4, Src: 1
Transmission Control Protocol, Src
Hypertext Transfer Protocol

Wireshark · Follow TCP Stream (tcp.stream eq 3) · Ethernet

```
GET /json/ HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/14
2.0.0.0 Safari/537.36
Host: ip-api.com

HTTP/1.1 200 OK
Date: Wed, 26 Nov 2025 20:48:08 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 353
Access-Control-Allow-Origin: *
X-Ttl: 60
X-RI: 44

{"status": "success", "country": "United States", "countryCode": "US", "region": "OR", "regionName": "Oregon", "
city": "Portland", "zip": "97209", "lat": 45.5288, "lon": -122.6821, "timezone": "America/Los_Angeles", "isp": "C
omcast Cable Communications, LLC", "org":
, "as":
, "query": }
```

1 client pkt, 1 server pkt, 1 turn.

Entire conversation (696 bytes) Show as ASCII No delta times Stream 3

Find: Case sensitive Find Next

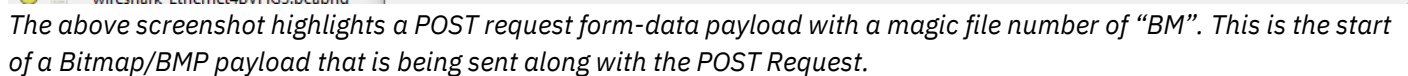
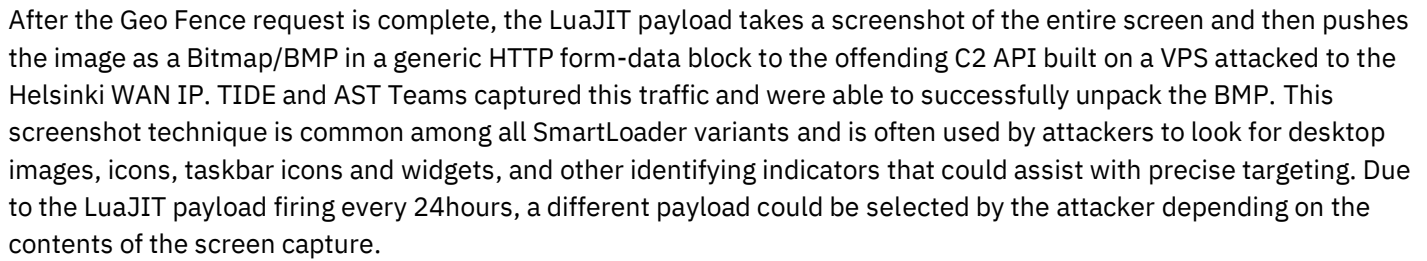
Filter Out This Stream Print Save as... Back Close Help

wireshark_Ethernet4BVG3.pcapng

The above redacted screenshot demonstrates the round trip HTTP plaintext request from ip-api.com/json/ back to the infect host with its own geographic information.

TYPE	VALUE	AUTHOR	CREATOR	LABELS	PLATFORM CREATION D/	ANALYSES	MARKING
INDICATOR	85.209.129.19	Threat Fox Abuse.ch	admin	botnet_cc c2 rhadamanthys	Oct 21, 2025	0	TLP-CLEAR
INDICATOR	http://85.209.129.105:2020/test112	Threat Fox Abuse.ch	admin	payload_delivery landupdate808 ko	Sep 2, 2025	0	TLP-CLEAR
INDICATOR	85.209.129.29	Threat Fox Abuse.ch	admin	botnet_cc rhadamanthys opendgam	Nov 13, 2025	0	TLP-CLEAR
INDICATOR	http://85.209.129.159	Threat Fox Abuse.ch	admin	botnet_cc c2 stealer	Nov 25, 2025	0	TLP-CLEAR
INDICATOR	http://85.209.129.105:2020/19	Threat Fox Abuse.ch	admin	payload_delivery kongtuke	Sep 1, 2025	0	TLP-CLEAR
INDICATOR	http://85.209.129.105:6060/capcha9856	Threat Fox Abuse.ch	admin	payload_delivery kongtuke	Sep 5, 2025	0	TLP-CLEAR
IPV4 ADDR...	85.209.129.19	Threat Fox Abuse.ch	admin	botnet_cc c2 rhadamanthys	Oct 21, 2025	0	TLP-CLEAR
IPV4 ADDR...	85.209.129.29	Threat Fox Abuse.ch	admin	botnet_cc rhadamanthys opendgam	Nov 13, 2025	0	TLP-CLEAR
URL	http://85.209.129.105:6060/capcha9856	Threat Fox Abuse.ch	admin	payload_delivery kongtuke	Sep 5, 2025	0	TLP-CLEAR
URL	http://85.209.129.159	Threat Fox Abuse.ch	admin	botnet_cc c2 stealer	Nov 25, 2025	0	TLP-CLEAR
URL	http://85.209.129.105:2020/test112	Threat Fox Abuse.ch	admin	payload_delivery landupdate808 ko	Sep 2, 2025	0	TLP-CLEAR
URL	http://85.209.129.105:2020/19	Threat Fox Abuse.ch	admin	payload_delivery kongtuke	Sep 1, 2025	0	TLP-CLEAR

The above screenshot shows similar malicious activity on the same range throughout 2025.





```
(root@kali-burner)-[/home/nowayjose/Downloads/aiscan-n-screencap]
# binwalk --dd=".*" -e NTEsN2QsN2UsNTgsNWIsNjAsNjIsNjcsYyw30Sw= --run-as=root

DECIMAL      HEXADECIMAL    DESCRIPTION
-----
236          0xEC          PC bitmap, Windows 3.x format,, 1024 x 768 x 24

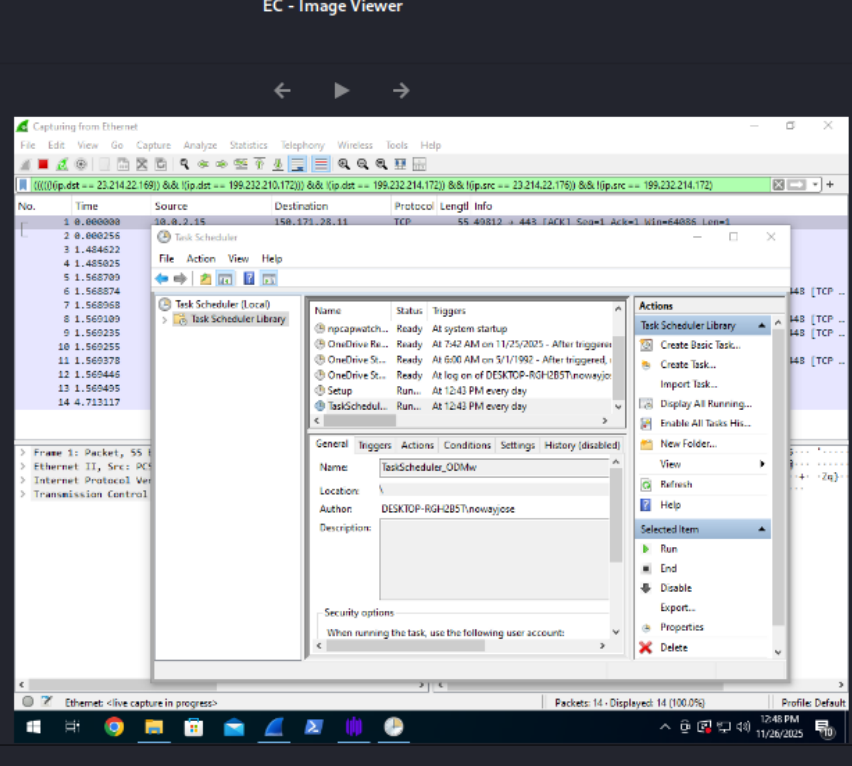
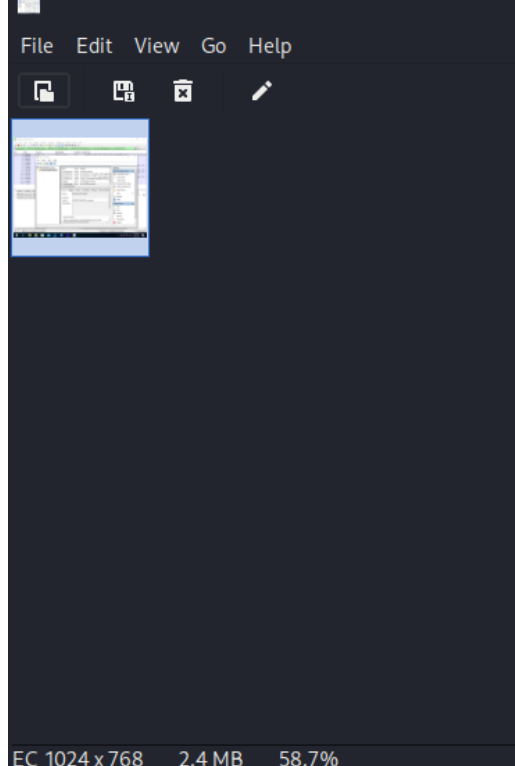
(root@kali-burner)-[/home/nowayjose/Downloads/aiscan-n-screencap]
# ls
ai-scan-n-malware  'NTEsN2QsN2UsNTgsNWIsNjAsNjIsNjcsYyw30Sw='  '_NTEsN2QsN2UsNTgsNWIsNjAsNjIsNjcsYyw30Sw=.extracted'
json              'NTEsN2QsN2UsNTgsNWIsNjAsNjIsNjcsYyw30Sw=(1)'  ODMw

(root@kali-burner)-[/home/nowayjose/Downloads/aiscan-n-screencap]
# cd _NTEsN2QsN2UsNTgsNWIsNjAsNjIsNjcsYyw30Sw=.extracted

(root@kali-burner)-[/home/nowayjose/Downloads/aiscan-n-screencap/_NTEsN2QsN2UsNTgsNWIsNjAsNjIsNjcsYyw30Sw=.extracted]
# ls
EC

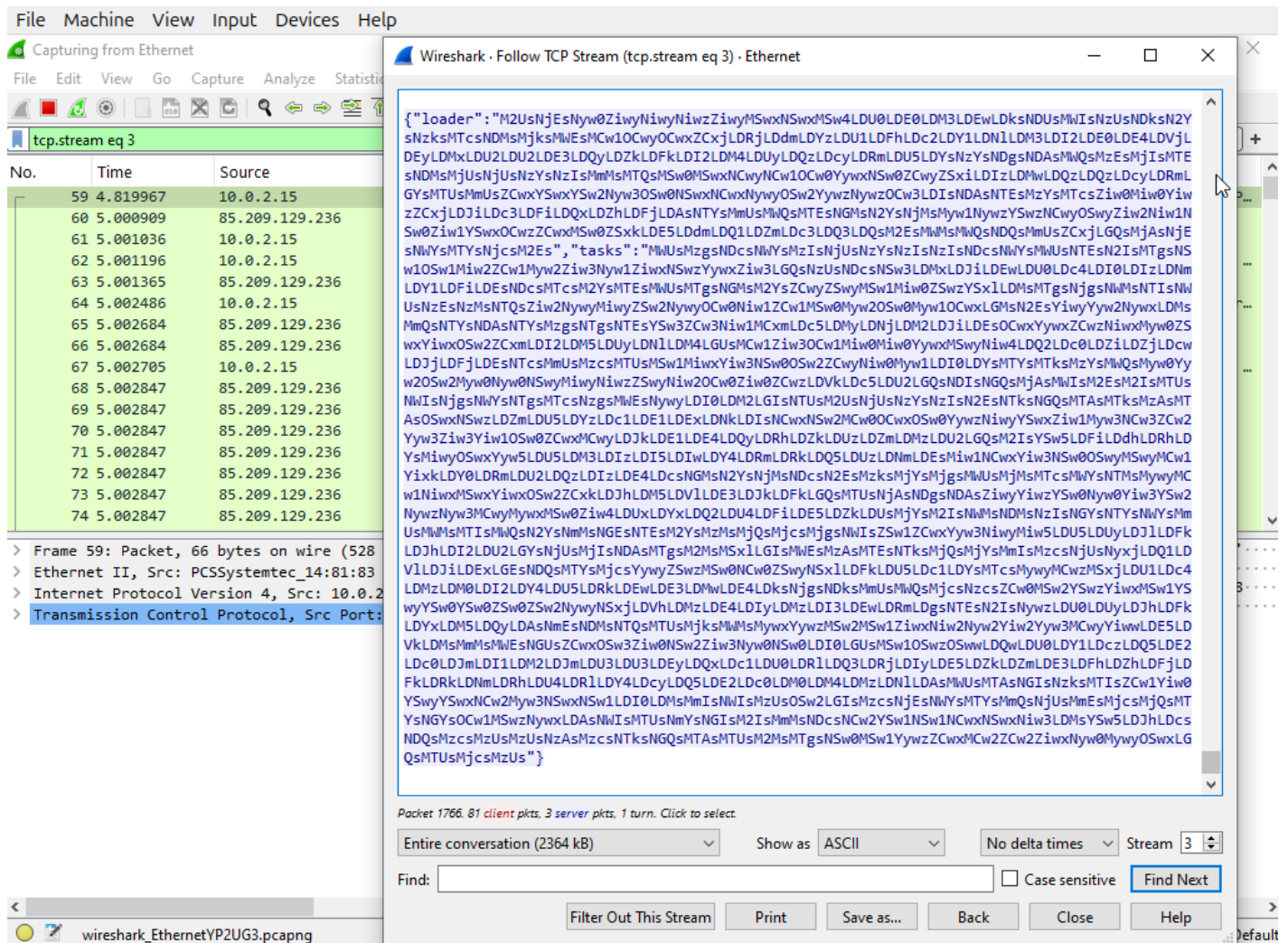
(root@kali-burner)-[/home/nowayjose/Downloads/aiscan-n-screencap/_NTEsN2QsN2UsNTgsNWIsNjAsNjIsNjcsYyw30Sw=.extracted]
# file EC
EC: PC bitmap, Windows 3.x format, 1024 x 768 x 24, cbSize 0, bits offset 54

(root@kali-burner)-[/home/nowayjose/Downloads/aiscan-n-screencap/_NTEsN2QsN2UsNTgsNWIsNjAsNjIsNjcsYyw30Sw=.extracted]
#
```



The above screenshot demonstrates binwalk being used to extract the BMP payload from the captured raw HTTP frame. On the top is the binwalk extraction, on the bottom is the “EC” file being opened in an image viewer, which shows the Win10 sandbox screencapture from the LuaJIT payload.

After the screen capture was sent, the offending API responded with a JSON payload. This activity also closely matches other known SmartLoader variants.



The above screenshot shows the entire JSON response from the offending API.

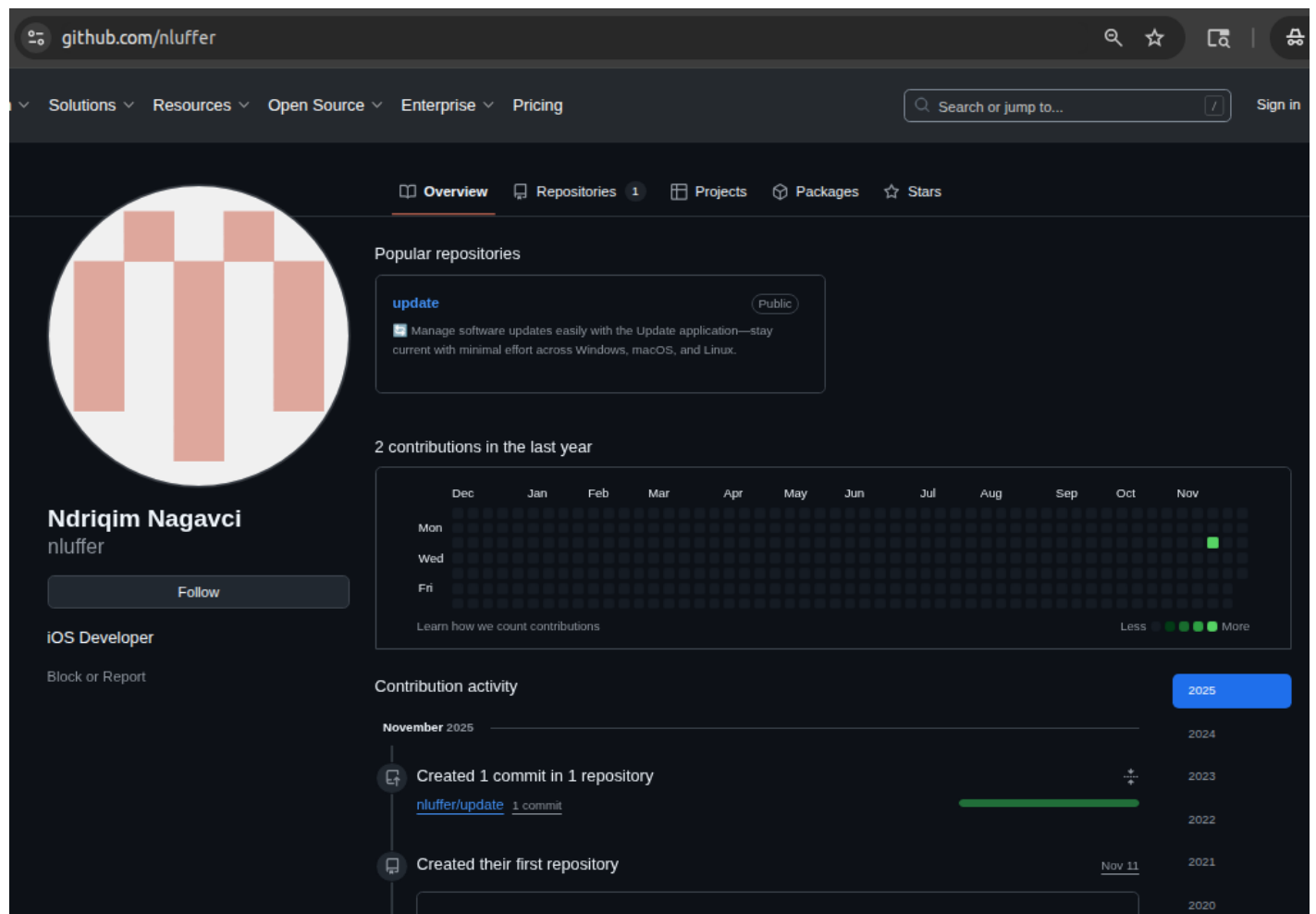
Instead of tackling the payload obfuscation and encryption directly, TIDE and AST Teams decided to rely on additional network traffic monitoring to understand what the offending Russia-Nexus C2 API was sending to the sandbox victim. While TIDE and AST Teams are actively performing static malware analysis on the provided artifacts, it was deemed more important to observe the infection chain holistically and dynamically instead.

After the Geo Fence check and initial C2 API requests were made, the LuaJIT payload resolved and accessed two additional Github hosted files:

- <https://github.com/nluffer/update/raw/240040191eb5b30c2074655bc26be94ab03d8559/update.log>
- https://raw.githubusercontent.com/Venkatesan-M4/file_storage/refs/heads/main/numeric.txt



Both Github accounts were created in November 2025 and have a single repository. This is a clear indicator of a “burner account”, meaning that the actor behind the account is treating it as a disposable asset.



The above screenshot is one of the offending Github malware hosts. The Venkatesan-M4 account mentioned previously was created on November 16.



The above screenshot demonstrates the README.md being clearly AI/LLM generated. Each of the links in the README.md point to files.com, a legitimate organization in Tempe, Arizona, US.

TIDE and AST Teams analyzed both of the files hosted in the offending Github accounts and once again found heavily obfuscated and encrypted payloads. OSINT collection was performed on the Github usernames, but nothing was found on either that relates to Malware Development or Russian-Nexus actors; both usernames and profile names appear randomly generated.



Time	Process	PID	Operation	Path	Result	Details
6:54:47.53695...	luajit.exe	9736	TCP Receive	DESKTOP-RGH2B5T...50884 -> cdn-185-199-111-133.github.com/https	SUCCESS	Length: 1153, seqnum: 0, connid:
6:54:47.53723...	luajit.exe	9736	WriteFile	C:\Users\nowayjose\AppData\Local\Microsoft\Windows\NetCache\IE\G4TVHU3P\numeric[1].txt	SUCCESS	Offset: 1,052,415, Length: 4,095
6:54:47.53777...	luajit.exe	9736	WriteFile	C:\Users\nowayjose\AppData\Local\Microsoft\Windows\NetCache\IE\G4TVHU3P\numeric[1].txt	SUCCESS	Offset: 1,056,510, Length: 4,095
6:54:47.53887...	luajit.exe	9736	WriteFile	C:\Users\nowayjose\AppData\Local\Microsoft\Windows\NetCache\IE\G4TVHU3P\numeric[1].txt	SUCCESS	Offset: 1,060,605, Length: 4,095
6:54:47.53913...	luajit.exe	9736	WriteFile	C:\Users\nowayjose\AppData\Local\Microsoft\Windows\NetCache\IE\G4TVHU3P\numeric[1].txt	SUCCESS	Offset: 1,064,700, Length: 2,736
6:54:47.53937...	luajit.exe	9736	QueryBasicInfor...	C:\Users\nowayjose\AppData\Local\Microsoft\Windows\NetCache\IE\G4TVHU3P\numeric[1].txt	SUCCESS	CreationTime: 11/26/2025 6:54:47
6:54:47.53939...	luajit.exe	9736	CloseFile	C:\Users\nowayjose\AppData\Local\Microsoft\Windows\NetCache\IE\G4TVHU3P\numeric[1].txt	SUCCESS	
6:54:48.58835...	luajit.exe	9736	CreateFile	C:\Users\nowayjose\AppData\Local\Temp\socket.luaODMw	SUCCESS	Desired Access: Generic Write, Re
6:54:48.58862...	luajit.exe	9736	CloseFile	C:\Users\nowayjose\AppData\Local\Temp\socket.luaODMw	SUCCESS	
6:54:48.58881...	luajit.exe	9736	CreateFile	C:\Users\nowayjose\AppData\Local\Temp\socket.lua	NAME COLL...	Desired Access: Read Data/List D
6:54:48.58902...	luajit.exe	9736	CreateFile	C:\Users\nowayjose\AppData\Local\Temp\socket.lua	NAME NOT ...	Desired Access: Read Attributes, I
6:54:48.58914...	luajit.exe	9736	CreateFile	C:\Users\nowayjose\AppData\Local\Temp\socket.lua	SUCCESS	Desired Access: Generic Write, Re
6:54:48.58941...	luajit.exe	9736	WriteFile	C:\Users\nowayjose\AppData\Local\Temp\socket.lua	SUCCESS	Offset: 0, Length: 352,256, Priority
6:54:48.58977...	luajit.exe	9736	WriteFile	C:\Users\nowayjose\AppData\Local\Temp\socket.lua	SUCCESS	Offset: 352,256, Length: 3,556, Pri

File Explorer window showing the Temp directory. The directory contains the following files:

Name	Date modified	Type	Size
msedge_installer	11/24/2025 8:19 AM	Text Document	16 KB
offline	11/26/2025 6:44 AM	File	24 KB
offline.session64	11/26/2025 6:44 AM	SESSION64 File	65 KB
socket.lua	11/26/2025 6:54 AM	LUA File	348 KB
socket.luaODMw	11/26/2025 6:54 AM	LUAODMW File	0 KB
socket2.lua	11/26/2025 6:54 AM	LUA File	285 KB
socket2.luaODMw	11/26/2025 6:54 AM	LUAODMW File	0 KB
stdlib.lua	11/26/2025 6:54 AM	LUA File	347 KB
stdlib.luaODMw	11/26/2025 6:54 AM	LUAODMW File	0 KB
wct5A8D.tmp	11/24/2025 3:10 PM	TMP File	101 KB

The above screenshot shows the “numeric.txt” and “update.log” obfuscated files being downloaded by LuaJIT from Github and written into the “%APPDATA%\Local\Temp” directory as “socket.lua” and “stdlib.lua” respectively. Interestingly, the Socket2.lua file was the raw HTML of the Github 404 page, which points to further misconfiguration in this SmartLoader payload.



Time of Day	Process ...	PID	Operation	Path	Result	Detail
6:54:46.66779...	luajit.exe	9776	RegOpenKey	HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions	SUCCESS	Desired Access: Read
6:54:46.66788...	luajit.exe	9776	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	REPARSE	Desired Access: Query Value, Enum
6:54:46.66789...	luajit.exe	9776	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Access: Query Value, Enum
6:54:46.67505...	luajit.exe	9776	RegOpenKey	HKLM\System\CurrentControlSet\Control\Lsa\FipsAlgorithmPolicy	REPARSE	Desired Access: Query Value
6:54:46.67507...	luajit.exe	9776	RegOpenKey	HKLM\System\CurrentControlSet\Control\Lsa\FipsAlgorithmPolicy	SUCCESS	Desired Access: Query Value
6:54:46.67511...	luajit.exe	9776	RegOpenKey	HKLM\System\CurrentControlSet\Control\Lsa\FipsAlgorithmPolicy	REPARSE	Desired Access: Query Value
6:54:46.67512...	luajit.exe	9776	RegOpenKey	HKLM\System\CurrentControlSet\Control\Lsa\FipsAlgorithmPolicy	SUCCESS	Desired Access: Query Value
6:54:46.67515...	luajit.exe	9776	RegOpenKey	HKLM\System\CurrentControlSet\Control\Lsa	REPARSE	Desired Access: Query Value
6:54:46.67516...	luajit.exe	9776	RegOpenKey	HKLM\System\CurrentControlSet\Control\Lsa	SUCCESS	Desired Access: Query Value
6:54:46.67521...	luajit.exe	9776	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Policies\Microsoft\Cryptography\Configuration	REPARSE	Desired Access: Query Value
6:54:46.67523...	luajit.exe	9776	RegOpenKey	HKLM\System\CurrentControlSet\Policies\Microsoft\Cryptography\Configuration	NAME NOT ...	Desired Access: Query Value
6:54:46.68325...	luajit.exe	9776	RegOpenKey	HKLM\System\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-3969147434-17154...	SUCCESS	Desired Access: All Access
6:54:48.61122...	luajit.exe	9736	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\luajit.exe	NAME NOT ...	Desired Access: Query Value, Enum
6:54:48.61126...	luajit.exe	9736	RegOpenKey	HKLM\Software\Microsoft\Wow64\xtajit	NAME NOT ...	Desired Access: Query Value
6:54:48.61165...	luajit.exe	9736	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\luajit.exe	NAME NOT ...	Desired Access: Query Value, Enum
6:54:48.61165...	luajit.exe	9736	RegOpenKey	HKLM\System\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-3969147434-17154...	SUCCESS	Desired Access: All Access
6:54:48.61647...	luajit.exe	9736	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\BAM	REPARSE	Desired Access: Query Value
6:54:48.61652...	luajit.exe	9736	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager\BAM	NAME NOT ...	Desired Access: Query Value
6:54:48.61680...	luajit.exe	9736	RegOpenKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders	SUCCESS	Desired Access: Query Value
6:54:48.61688...	luajit.exe	9736	RegOpenKey	HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Layers	SUCCESS	Desired Access: Query Value
6:54:48.61744...	luajit.exe	9736	RegOpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\SideBySide	SUCCESS	Desired Access: Read
6:54:48.61798...	luajit.exe	9736	RegOpenKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders	SUCCESS	Desired Access: Query Value
6:54:48.61897...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	REPARSE	Desired Access: Query Value
6:54:48.61899...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Access: Query Value
6:54:48.61904...	luajit.exe	7116	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap	REPARSE	Desired Access: Query Value
6:54:48.61904...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager\Segment Heap	NAME NOT ...	Desired Access: Query Value
6:54:48.61908...	luajit.exe	7116	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	REPARSE	Desired Access: Query Value, Enum
6:54:48.61909...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Access: Query Value, Enum
6:54:48.62161...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\StateSeparation\RedirectioMap\Keys	REPARSE	Desired Access: Read
6:54:48.62163...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\StateSeparation\RedirectioMap\Keys	NAME NOT ...	Desired Access: Read
6:54:48.62171...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	REPARSE	Desired Access: Read
6:54:48.62172...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Access: Read
6:54:48.62198...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	REPARSE	Desired Access: Query Value, Set
6:54:48.62200...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	NAME NOT ...	Desired Access: Query Value, Set
6:54:48.62201...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\Srp\GP\DLL	REPARSE	Desired Access: Read
6:54:48.62201...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\Srp\GP\DLL	NAME NOT ...	Desired Access: Read
6:54:48.62202...	luajit.exe	7116	RegOpenKey	HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCESS	Desired Access: Query Value
6:54:48.62207...	luajit.exe	7116	RegOpenKey	HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	NAME NOT ...	Desired Access: Query Value
6:54:48.62209...	luajit.exe	7116	RegOpenKey	HKLM\System\CurrentControlSet\Control\File System\	REPARSE	Desired Access: Read

Showing 1 681 of 14 301 518 events (0.011%)

Backed by virtual memory

The above screenshot demonstrates the LuaJIT binary payload checking registry keys that control system settings. It also looked for Image File Execution Options (IFEO) on itself, as that can often be a sign of reverse engineering or analysis.

TIDE and AST Teams monitored the sandbox environment with the live LuaJIT payload and scheduled tasks for over 24 hours. No additional C2 API callouts were made, and subsequent normally fired scheduled tasks performed the same steps. This lack of additional activity could be due to numerous reasons:

- The missing socket2.lua file is most likely integral to shipping collected data from the other infostealer payloads (socket.lua and stdlib.lua) back to the C2. With socket2.lua being the raw HTML 404 page from GitHub, it could be that the actor behind the AIRedScam payload was not done with designing their infrastructure when TIDE and AST began reversing their payloads.
- The actor behind the AIRedScam campaign noticed Wireshark, Procmon, and other tools that can assist in Malware analysis in screencaps sent from the victim sandbox back to the C2, and then set the WAN IPs and other unique identifiers of the victim sandbox into a denylist. This is common with SmartLoader campaigns, as the attackers have aggregate controls over what is deployed to victim machines when the scheduled task fires.
- The actor behind AIRedScam has not properly built and configured the SmartLoader and subsequent infostealer payloads.

As of November 27, 2025, TIDE and AST Teams have collected enough intelligence from this campaign to design bespoke detections to protect UVCyber customers and partners. TIDE and AST will continue to explore and analyze the AIRedScam campaign and will update this report if new and useful intelligence is collected.



TIDE and AST Team Technical Conclusion

The AIRedScam campaign is unique in its targeting of Red Team and Offensive professionals looking to automate their recon and enumeration stages during an engagement. However, AIRedScam is very average from a technical standpoint. The use of LuaJIT, the file names, the payload behavior, and the host targeting is almost identical to all other SmartLoader campaigns that have popped up throughout 2025. The AIRedScam payload, and similar SmartLoader payloads, are stealthy and rarely trigger Windows Defender and other AV/EDR products. Registry object checks and anti-reversing techniques all aligned with standard SmartLoader behavior as well.

While the offending C2 API WAN IP is hosted out of a Russia-Nexus owned ASN Range, this does not necessarily mean that the threat actor behind the AIRedScam campaign was associated with Russian state interests; especially with Geo Fence checks failing when presented with a geographically Russian WAN POP IP on the victim sandbox. As we have seen throughout the last decade, Threat Actors often maintain excellent OPSEC and FINSEC which make them extremely hard to enumerate and hunt individually. This adherence to OPSEC and FINSEC fundamentals is seen in the AIRedScam campaign through the use of burner accounts and use of VPS providers in a non-FVEY country who can accept “bullet-proof” payment options.

From a technical standpoint the AIRedScam payload, and subsequent malware it pulls down from burner Github accounts, is nothing more than a carbon copy of similar SmartLoader malware. What sets AIRedScam apart is its choice in human targeting. This campaign preys upon junior to mid skill Red Team Operators looking to automate their workflows, individuals interested in Offensive Capture the Flag, and curious SysAdmins. UVCyber TIDE and AST teams believe this is just the start in a new trend of targeting the Offensive Cybersecurity community, not only with Supply chain attacks against commonly used offensive tools, but also targeting the emergent need to automate and augment offensive workflows with AI/LLM backed infrastructure. UVCyber TIDE and AST Teams urge extreme caution when downloading and executing precompiled binaries and compressed archives from any source online, no matter how good the README.md appears.



Mitigation Analysis

Enterprise environments can reduce their exposure to SmartLoader by strengthening software provenance controls and narrowing the attack surface created by unmanaged code acquisition. Because SmartLoader spreads primarily through compromised GitHub repositories, fake tooling, and SEO-driven lure content, organizations should enforce strict policies around how administrators, developers, and engineers obtain utilities and scripts. Centralized software repositories, controlled package mirrors, and mandatory code-signing validation dramatically limit the risk of inadvertently introducing malicious loaders through routine workflow automation. Combined with browser isolation or application sandboxing for high-risk download activities, these controls prevent SmartLoader from exploiting open-source trust relationships and developer supply-chain habits.

Defense-in-depth across the endpoint is equally important, particularly because SmartLoader's LuaJIT architecture is difficult for traditional controls to classify. Enterprises should deploy EDR controls capable of monitoring script engines, foreign function interfaces, unusual in-memory JIT activity, and anomalous network beacons originating from nonstandard interpreters. Memory scanning, AMSI enhancement, and heuristic detection of dynamic call chains can help identify SmartLoader's staged execution even when the loader itself is obfuscated. Additionally, enforcing application allow-listing and restricting the execution of unsigned or unpackaged interpreters reduces the ability of SmartLoader to escalate from an initial Lua script to a fully functional staging platform.

Finally, organizations must treat SmartLoader as a broader supply-chain and developer-workflow threat, not merely a malware sample. Security teams should integrate repository validation, dependency checks, and real-time monitoring of cloned open-source projects into CI/CD pipelines to prevent poisoned artifacts from entering development ecosystems. Regular red-team exercises focused on GitHub impersonation, malicious tool repositories, and drive-by administrative tooling downloads can uncover blind spots in procurement and developer hygiene. By combining technical controls with cultural and process-oriented changes—such as mandatory training for engineers on source authenticity and tooling security—enterprises can significantly reduce the likelihood that SmartLoader infiltrates their environment and weaponizes trusted workflows.

UVCyber TIDE and AST Teams suggest heavily restricting LuaJIT within your environment if there is no official business need for this software.



IOCs

https://github.com/Rbxolexc8405/AiScan-N/
https://github.com/Rbxolexc8405/AiScan-N/blob/main/cteniform/AiScan-N-v2.4.zip
https://github.com/Venkatesan-M4/file_storage
https://raw.githubusercontent.com/Venkatesan-M4/file_storage/refs/heads/main/numeric.txt
https://github.com/nluffer/update
https://raw.githubusercontent.com/nluffer/update/240040191eb5b30c2074655bc26be94ab03d8559/update.log
b22dd8f01257d692b7a6d731a05a1a958596a2cb1844bafdfa339581cd7f57f8
4a37905fc01a4316507ec4f88b3b0aa8ee5f6afae2f85f77e48acd1923bf5eed
c3d1d08d383f5346febd9280e6796baa2208c5136bb7c3f66645b303f74d531f
0575bb1a77b956d6eac6c6aa7dfb53e611e8b52955436e6e73c0e802bfa23dba
fc88eaff5fea14f3aedc1bf6a4d147ee56b6b466c85fd0b17248a3a7ecd60c6f
5343326fb0b4f79c32276f08ffcc36bd88cde23aa19962bd1e8d8b80f5d33953
c7a657af5455812fb215a8888b7e3fd8fa1ba27672a3ed9021eb6004eff271ac
85.209.129[.]236
http://85.209.129[.]236/api/
93.123.39[.]74
http://93.123.39[.]74/api/
AS209693 - OC Networks LTD
AS213702 - Qwins LTD



About UltraViolet Cyber

UltraViolet Cyber is a leading tech-enabled managed security services provider, delivering unparalleled cybersecurity expertise that fills technology and talent gaps across Global 2000 and Federal Government customers. Founded and operated by security practitioners from the national intelligence community, UltraViolet Cyber connects offensive security, application security, detection and response, and security engineering to deliver a differentiated approach to cybersecurity operations. Transforming customers' security programs, UltraViolet Cyber's flagship security-as-a-service solution, UV Lens, removes complex operational silos, replacing them with integrated security capabilities. UltraViolet is headquartered in McLean, Virginia with technology centers across the world.

443.351.7630 / info@uvcyber.com |  UltraViolet Cyber |   @uv_cyber
