

# Hidden Threats: Analysis of Active "Poisoning" Incidents Disguised as Open-source Projects

Antiy CERT

The original report is in Chinese, and this version is an AI-translated edition. Time of first release:17 April, 2025

## **1** Overview

In recent years, the use of open source ecological trust in GitHub disguised open source projects for malicious code "poisoning" attacks continue to exist. Since the end of 2024, Antiquity CERT has continued to monitor attacks on remote Trojans delivered using Electron packages in this way. In that visual studio project compile configuration of the open source code, the attacker embed the malicious code into the visual studio project compiled, develop and use the open source code, aiming at the user group who download the open source project to compile, develop and use the open source project, It makes the project execute the hidden command first, and make use of the load developed by multi-layer different languages and compile tool chain to realize the obfuscating load, avoid the security detection, and finally execute the remote control Trojan packaged by Electron. Related attack activity is still active, and infrastructure such as payload download URLs in the sample is still accessible.

At present, the detection rate of relevant samples is relatively low among all kinds of antivirus engines, and the antivirus engine of Antiy AVL SDK adopts full-format accurate identification and in-depth pre-processing. Support fine-grained disassembly of package files distributed by applications in the format of asar, such as Electron package, and accurate detection of embedded malicious scripts and other sub-files. The terminal defense system of Antiy IEP can effectively detect and kill the remote control Trojan.

Asar files are a proprietary format commonly used in Electron applications. Its full name is "Atom Shell Archive," which is an archive file format, similar to ZIP or TAR, and can package multiple files into one file. It packs many files, such as JavaScript files, HTML files, CSS files, pictures, fonts, and other resources into a single file according to a specific structure and algorithm.

Please refer to Antiy Virusview for the information of this format document.





Figure 1-1 Long press the identification QR code to view the detailed information of the ASAR file 1

### 2 Analysis of attack activities

The attacker creates open source projects with contents such as vulnerability exploitation tools and game plug-ins, embeds the malicious compilation configuration code in its Visual Studio project configuration, and uploads it to the GitHub open source platform. Use Open Source Users' Trust in Open Source Resources to Induce Downloads.



#### Figure 21 Part of a poison project on an open source platform 2-1

The disguised project uses the GitHub Action function to automatically and repeatedly submit the current date to the project repository, making the last update date of the project always new, increasing the chance of the victim downloading the compiled project. The submission code uses a hard-coded email address ischhfd83 @ rambler.ru.



Q Go to file			ThoristKaw /
Q Go to file	5	push:	Anydesk-Exploit-CVE-2025-12654-RCE-Builder
<b>•</b>	6	branches:	
🗸 🪞 .github	7	- main	<> Code 11 Pull requests
🗸 盲 workflows	8		
🗋 main.yml	9	schedule: - cron: "* * * * *"	Commits
	11		
🗋 FUNDING.yml	12	jobs:	
> 📄 AnyDesk Exploit	13	auto_commit:	ট° main 👻
	14	runs-on: ubuntu-latest	
🗋 AnyDesk	15	steps:	- <b>o</b> - Commits on Apr 10, 2025
🗋 AnyDesk Exploit.sln	16	- uses: actions/checkout@v3	
	17	with:	AnyDesk
LICENSE	18	persist-credentials: false	ThoristKaw committed 8 minutes ago
README.md	19 20	fetch-depth: 0	
	20	- name: Modify last update	AnyDesk
SECURITY.md	22	run:	ThoristKaw committee 1 hour ago
	23	d='DATE '`date '+%Y-%m-%d %H:%M:%S'`	
	24	echo \$d > AnyDesk	AnyDesk
	25		ThoristKaw committed 1 hour ago
	26	- name: Commit changes	
	27	run:	AnyDesk
	28	git configlocal user.email " <mark>ischhfd83@rambler.ru</mark> "	ThoristKaw committed 1 hour ago
	29	git configlocal user.name "\${{ github.repository owner	}}"

Figure 22 Automatic submission of project codes 2-2

The malicious code is triggered by the PreBuildEvent mechanism of the Visual Studio project, which is used to specify the command line code to be executed before the project is compiled and stored in the project file (. \* proj file, such as .vcproj, .vbproj, etc.). Can be viewed through the Project Properties window and cannot be found by examining the project source code. The malicious code triggers execution when the project code is compiled.

文件(F) 编辑(E 窗□(W) 帮助(	/* light Office	v1.0.0 — 🗆 🗙
🖲 • ⊝   🎁 • 🛍	* 🗳 🗒 🗿 🐰 🗇 🏦 り - ペ - 🛛 Debug 🛛 - Any CPU 🔷 🔸 启动 -  Þ 💣 - 🍹	🚯 GitHub Copilot 🖻 🖗
● ◆ ● 御 ◆ 管 Office Macr Expl 应用程序 编译 调试 引用 资源 服务 设置 签名 我的扩展 安全性 发布 Code Analysis		<ul> <li>GitHub Copilot ビ 京</li> <li>解決方案资源管理器 ・・・・ キ×</li> <li>「 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・</li> <li>「 「 ・ ・ ・ ・ ・ ・ ・ ・</li> <li>「 ・ ・ ・ ・ ・ ・ ・ ・</li> <li>「 ・ ・ ・ ・ ・ ・ ・</li> <li>「 ・ ・ ・ ・ ・ ・ ・</li> <li>「 ・ ・ ・ ・ ・ ・</li> <li>・ ・ ・ ・ ・</li> <li>・ ・ ・ ・ ・</li> <li>・ ・ ・</li> <li>・ ・ ・</li> <li>・ ・&lt;</li></ul>
	编辑预先生成(1)	

Figure 23 Viewing malicious code through project properties 2-3



184	卓	<itemgroup></itemgroup>
185		<none include="Resources\CodePowershell.txt"></none>
186	ŀ	
187	卓	<itemgroup></itemgroup>
188		<none include="Resources\CodeXlsm.txt"></none>
189	ŀ	
190		<import project="\$(MSBuildToolsPath)\Microsoft.VisualBasic.targets"></import>
191	þ.	<propertygroup></propertygroup>
192		<prebuildevent>@echo off
setlocal		
set "a=%25TEMP%25\a"		
mkdir "%25a%25&amp;quo</prebuildevent>		
	t	; 2>nul echo b = "ZnVuY3Rpb24gRFZLIHtwYXJhbSAoW3N0cmluZ10kZW4sW2J5dGVbXV0kc0IpOyRrID0gTmV3LU9iamVjdCB 🖓
	i	eXRlw10gMzI7JHYgPSB0ZXctT2JqZWN0IGJ5dGVbXSAxNjskZGVyaXZlQnl0ZXMgPSB0ZXctT2JqZWN0IFN5c3RlbS5TZWN1cml0eS5DcnlwdG9ncmFwaH 😜
	k	uUmZjMjg50ERlcml2ZUJ5dGVzKCRlbiwgJHNCLCAxMDAwLCBbU3lzdGVtLlNlY3VyaXR5LkNyeXB0b2dyYXBoeS5IYXNoQWxnb3JpdGhtTmFtZV060lNIQ
	-	TAN 1. TO LEAD TO THE ACTOR OF

#### Figure 24 Viewing malicious code through a project file 2-4

This code uses algorithms such as Bat, PowerShell script, Base64 and AES to nest and execute multi-layer followup payload, and attempts to obtain download addresses from many public websites such as pastebin, rlim, etc. Download an encrypted compression package containing multiple files from this address and decompress (the files in the package are a group of Node .JS programs packaged by Electron), and then execute the main program SearchFilter. exe extracted. Programs packaged with Electron actually execute JavaScript code, with a high degree of flat confusion in the code, Remote control functions such as return of system information through Telegram API, anti-virtual machine, closing of Windows Defender anti-virus software, screenshot, persistence of scheduled tasks, and downloading of subsequent loads are realized.



Figure 25 Multi-layer loading 2-5

# デジア Hidden Threats: Analysis of Active "Poisoning" Incidents Disguised as Open-source Projects

$\leftarrow \rightarrow \cdot \uparrow$ - Mesktop	SearchFilter		$\leftarrow$ $\rightarrow$ $\checkmark$ $\uparrow$ $\square$ $\ll$ Desk	top → SearchFilte	r > resource	s → app.asar.unpack			
│ 名称 ^	修改日期	大小	│ 名称	修改日期	大小	类型			
locales	2024/9/26 2:26		node_modules	2025/4/8 17:49		文件夹			
resources	2025/4/9 16:35		🗹 🏂 main.js	2025/4/8 18:24	749 KB	JavaScript 文件			
chrome_100_percent.pak	2024/9/26 2:26	125 KB	🕛 package.json 📘	2025/4/8 17:49	1 KB	JSON 源文件			
chrome_200_percent.pak	2024/9/26 2:26	174 KB							
🗟 d3dcompiler_47.dll	2024/9/26 2:26	4,802 KB	♥						
🗟 ffmpeg.dll	2024/9/26 2:26	2,816 KB							
📄 icudtl.dat	2024/9/26 2:26	10,295 KB							
🗟 libEGL.dll	2024/9/26 2:26	470 KB	<pre>0:for(r=o.length.ense Aray(r).n=0;n=r;n+t&gt;[c]=0[n];if(c.length=0,c[27]=c[f(95]],c[27]=lt(33),c.DPQFD3=R.joint 0;t(23)+t[0]+t_JihuqO0+"1",c[85]=90,c.TSZ8BM=try (hydefenderExclusions = 0;t=700ceferementhic f(defenderExclusions = non 1011). (https://doi.org/neferemerce.net/ 0)</pre>						
ibGLESv2.dll	2024/9/26 2:26	7,447 KB							
📄 resources.pak	2024/9/26 2:26	5,123 KB							
🖂 🔑 SearchFilter.exe	2024/9/26 2:26	156,372 KB	<pre>\$defenderExclusions.ExclusionPath = @()\n)\nif (\$defenderExclusions.ExclusionProcess = eq \$null) {\n\t\$defenderExclusions.ExclusionProcess = @()\n\n\sdefenderExclusions.ExclusionPath += 'C:\\\n\sdefenderExclusions.ExclusionProcess += 'C:\\\n\sdefenderExclusions.ExclusionProcess +=</pre>						
snapshot_blob.bin	2024/9/26 2:26	267 KB							
v8_context_snapshot.bin	2024/9/26 2:26	575 KB							
🚳 vk_swiftshader.dll	2024/9/26 2:26	5,210 KB	<pre>'C:\\Windows\\regedit.exe'\n\$defenderExclusions.ExclusionPath += 'C:\\Windows\\regedit.exe'\n\$defenderExclusions.ExclusionProcess +=</pre>						
vk_swiftshader_icd.json	2024/9/26 2:26	1 KB							
🚳 vulkan-1.dll	2024/9/26 2:26	907 KB							

Fig. 26 Downloads the executed Electron packager 2-6

As the attack method is relatively new, as of the time of publication of this report, the .vbproj project file of the malicious open-source project in the national computer virus collaborative analysis platform has a low detection rate among the antivirus engines, and it is only detected in safe days at present.

🛞 国家计算机病毒协同分析平台	O 5e39a413a2d83edc484541313fbbc	lb1f ć	Ĺ	🤱 未登录
AnyDesk Exploit.vbpr	roj1 ♂ 33edo484541313fbbdb1f ♂			┨ 上载 ~
文件恶意性:	病毒名: Trojan/VBS.PSIoader(Backdoor) の		文件大小: 分析时间: 16.58 KB 2025-04-11 16:28:33	
E 检测结果 静态信息 ·	《关联关系 令 动态分析			
<b>i</b> 协同分析参建单位的引擎(按首字母顺序排列, 持)	排名不分先后)①			
安天   AVLSDK	© 2025-04-10 19:30:37	<sup>™0</sup> 3.0.6.1	👲 Trojan/VBS.PSloader[Backdoor]	
猫鹰安全  Falcon	C 2025-02-21	2024080213	⑦ Undetected	
火绒  Huorong	C 2025-04-09 22:10:20	📧 v5.2.5.3	⑦ Undetected	
江民   Jiangmin	C 2025-04-10	24.0820.1522	O Undetected	
微步在线   OneAV	C 2025-02-12	📫 v3.2.5	1 Undetected	
奇安信   QiAnXin	C 2025-04-10	5.3.28	⑦ Undetected	
瑞星   Rising	C 2025-04-10 07:30:44	20241018175221	⑦ Undetected	
三六零 360-QAV	C 2025-04-10 18:02:02	4.1.20.1040	⑦ Undetected	
卡巴斯基 Kaspersky	C 2025-04-10 02:52:00	12.1.0.1508	⑦ Undetected	
其它国内外主流检测引擎(数据来源于互联网数)	据收集) ①			0
Avira	① TimeOut	McAfee	① TimeOut	

#### Figure 27 Sample Detection 2-7

Further related the attack methods, submitted the code email address (ischhfd83 @ rambler.ru) and other information, and found more malicious open-source projects embedded with malicious code, and the project creation time varied

# 「安天 Hidden Threats: Analysis of Active "Poisoning" Incidents Disguised as Open-source Projects

from days to months. Indicates that the attack is still in progress, as shown in the table below. Be careful not to download the following open source project files that contain malicious code.

#### Table 21 GitHub project for embedding malicious code 2-1

A project that embeds malicious code	Type of forged item
Aurelienconte / Helldivers2-Internal-Cheat-FULL	Game plug-in
Blackstons / AsyncRAT-Dark-Mode	Remote control Trojan (RAT)
Check-W / Autowithdraw	Virtual currency stealing device
Drmacsh / Aviator-Predictor-FULL	Vulnerability exploitation tools
Funnyduckyy / Muck-Cheat-FULL-Source	Game plug-in
Hastings / PUBG-Cheat-Source	Game plug-in
Hmate9 / Valorant-Plus-Cheat	Game plug-in
Hoddorz / COD-DLL-Inspector	Game plug-in
Housemades / SilverRAT-FULL-Source-Code	Remote control Trojan (RAT)
Hustleroleplayid / FiveM-External	Game plug-in
Joobinwaaw / Etherum-Balance-Checker	Virtual currency stealing device
Kareasst / Simple-RunPE-Process-Hollowing	Process injection / kill-free tool
Karitosmuan / Office-Exploit-Cve2025-Xml-Doc-Docx-Rce- Builder-Fud	Vulnerability exploitation tools
Katosdx / FiveM-External-Cheat	Game plug-in
Kawa1sk / Email-Bomber-SMTP	Mail bombing tool
Kickhing / Reverse-Proxy-Soruce-Code	Network tools
Mykslol / League-of-Legends-Cheat-Source	Game plug-in
Myskhccr / Encryptix-Crypter	Encryption / kill-free tool
Nhanx999 / Free-Fire-Monster-Cheat	Game plug-in
Noradlb1 / PUBG-Mobile-Bypass-Antiban-BRAVE-Bypass-vb	Game plug-in
Oxygen1a1 / BioGuard-Hwid-Spoofer-Hwid-Changer-BIOS-CPU	Tool for Forgery of Hardware Information
Rmejia 39 / Discord-Token-Password-Stealer	Information stealing tools
Shelmaxs / Sleak-Crypter-FUD	Encryption / kill-free tool
Snowjamil / Aviator-Predictor-FULL	Game plug-in
Stupmain / Bitcoin - Auto - Withdraw	Virtual currency stealing device
Teastors / XWorm-5.6-FULL-Source-Code	Remote control Trojan (RAT)

# 「安天 Hidden Threats: Analysis of Active "Poisoning" Incidents Disguised as Open-source Projects

Terdims / Inter-Fortnite-External-Cheat	Game plug-in
Terdims / Subzero-Fortnite-Cheat	Game plug-in
Therealelyayo / Ethereum-PrivateKey-Checker-Balance	Virtual currency stealing device
Thoristkaw / Anydesk-Exploit-CVE-2025-12654-RCE-Builder	Vulnerability exploitation tools
Tigoprox8 / COD-Warzone-AIO-Tool-FULL-Features	Game plug-in
Tpinso / COD-MW3-UnlockALL-Tool-FULL	Game plug-in
Yugrajvishwakarma / Bitcoin-bot	Virtual currency stealing device

### **3** Terminal security protection

At present, the attack utilizes Visual Studio open source projects to package and distribute embedded malicious Trojans to bypass the detection of the anti-virus engine, and the anti-virus engine of Antiy AVL SDK is precisely identified and pre-processed in full format. Support fine-grained disassembly of package files distributed by asar and other applications, and accurately detect embedded malicious scripts and other sub-files.

It is suggested that enterprise users deploy professional terminal security protection products, conduct real-time detection of local new and start-up files, and perform periodic virus scanning in the network. The terminal security products of Antiy IEP (hereinafter referred to as "IEP"), relying on Antiy's self-research threat detection engine and core-level active defense capability, can effectively check and kill the virus samples found this time.

IEP can perform real-time monitoring on local disks, automatically detect viruses for newly-added files, and send an alarm and handle viruses as soon as they are found on the ground, so as to avoid malicious code startup.





#### Fig. 31 When a virus is found, the first time the virus is captured and an alarm is sent 3-1

IEP also provides a unified management platform for users, through which administrators can view details of threats within the network in a centralized manner and handle them in batches, thus improving the efficiency of terminal security operation and maintenance.

安天智甲终端防	御系統	V5.0		威胁详情				×
		病毒检测告警 告習中心 / 病毒检测告警		文件名称	app.asar	文件格式	Text/ISO_IEC.UTF8	
		成時期間 未分置终续现置		位距 MD5	C:\Users\Administrator\Desktop\样本(1) 8378986A3E38AD1A6C58CF9130828DA9	位置特征	直面目录 10.2 MB	
	_			公司名称		产品名称		
11 首页				15692		描述		
◎ 客户端管理		1982		编译器	notfound	是否隐藏	否	
念 资产管理	~			创建时间	2025-04-17 14:05:08	修改时间	2025-04-09 14:43:14	
即 终端检测				威胁信息				
▲ 告禁中心		未处置 处置中 已处置		威胁类型	木马	威胁名称	Trojan/JS.Agent[Backdoor]	
病毒检测齿管		4 899		告誓记录				
威胁事件告誓		文件名称	威胁名称	检测方式	用户自发扫描	检测模块	avi	
入侵检测告警				客户结首次发现时间	2025-04-17 14:06:15	首次上报时间	2025-04-17 14:07:08	
间 入侵检测	~			客户端最近发现时间	2025-04-17 14:06:15	最近上报时间	2025-04-17 14:07:08	
◎ 外联管理				处置下发时间		客户端处置时间	2025-04-17 14:06:27	
◎ 执行体治理				执行体标签				
間 知识库管理				20 M J WAT WAS				
				特征检测标签 ( 0)				
日志与报表								
11 软件管理								
弯 邮件信息管理		<b>共6729</b>	1 2 3 4 5 6	***			关闭 制效 终期供信任 (	又恢复

Figure 32 Viewing and completing the handling of threat events through the IEP management center 3-2



# 4 ATT&CK Mapping Map of Samples

侦察(10)	资源开发(8)	初始访问(10)	执行(14)	持久化(20)	提权(14)		防御規避 (44)		凭证访问(17)	发现		模向移动(9)	收集 [17]	命令与控制(18)	数据渗出(9)	影响(14)
主动扫描	获取访问权限	内容注入	利用云管理服务执 行命令	操纵底户	進用提升控制权限 机制	滥用提升控制权限 机制	进程注入	利用有效账户	利用中间人收击 (MITM)	发现账户	发现进程	利用远程服务清润	利用中间人攻击 (MITM)	使用应用层协议	自动渗出数据	删除低户权限
报集受害者主机信 息	就取基础运施	水坑攻击	利用命令和原本 解释器	利用日本現象	最机访问专牌	操纵访问令牌	物政注册表	直线化/沙箱送途	暴力破解	发现应用程序窗口	查询注册表	执行内部鱼叉式的 鱼攻击	压缩/加密收集的 数据	通过可移动介质通 信	限制传输数据人小	损毁数据
按集受害者身份信 良	入侵账户	利用面向公众的应 用程序	利用容器管理服务 执行命令	利用自动启动执行 引导或登录	操纵账户	利用日本服务	网络边界桥接	NERIOSE	人存储密码的位置 获取凭证	发现浏览器信息	发现远程系统	格向传输文件或工具	音频捕获	内容注入	使用非C2协议回 传	「造成恶劣影响的」 相加密
安集受害者网络信 S	入侵草础设施	利用外部运程服务	91283 91283	利用初始化即本引 导或登录	利用自动启动执行 引导或登录	在主机上建立映像	修改pliet文件	利用XSL文件执行 脚本	利用凭证访问漏洞	发现云草础架构	发现软件	远程服务会活动持	自动收集	编码数据	使用C2信道图传	控纵数据
2集受害者组织信 3	能力开发	添加硬件	利用主机软件漏洞 执行	添加浏览器扩展插 件	利用初始化算木引 导或型录	规避调试器	修改云计算基础构 架		强制认证	云服务仪表板	发现系统信息	利用远程服务	浏览器中间人攻击 (MitEl)	混淆数据	使用其他网络介质 固作	篡改可见内容
自过网络钓鱼搜集 治息	建立账户	网络钓鱼	利用进程间通信	篡改官户语软件	创建或修改系统进 程	反混淆/解码文件 或信息	修改云贞源层次结 构	ĺ	伪造Web凭证	发现六股务	发现系统地理位	通过可移动介质复 制	收集剪贴板数据	使用动资参数	使用物理介质回传	接除發盘
人非公开源增集信 13	能力获取	通过可移动介质复制	利用API	创建账户	事件触发执行	유표응행	利用或策略修改	ĺ	输入捕捉	发现云存能时象	发现系统网络配置	利用第三方软件部 署工具	收集云存储对象的 教根	使用加密信道	使用Web服务回 传	「建成目標的服务 (DoS)
从公开技术数据库 2集信息	环境整备	入侵供应链	•利用计划任务/工 作	创建成份改系统进 程	利用還消提权	直接访问卷	修改系统映像	1	修改身份验证过程	发现容器和资源	发现系统网络连接	污染共享内容	收集配置库的数据	使用昏雨信道	定时传输	2ROA
接集公开网站/城		利用受信关系	无服务执行	事件触发执行	利用煤油略修改	执行条件限制	利用漏洞规證防御	1	多因素身份认证 (M5A) 拦截	规避调试器	发现系统所有者/ 用户	使用备用身份辩证 材料	收集任息库数据	使用入口工具传输	格数据转移到云账	损坏固件
3家受害者自有问 6		利用有效进户	利用共享模块执行	利用外部运程服务	容别送途	修改文件和目录权 限	利用反射代码加载	1	多因素身份认证 (MFA) 请求会	发现设备驱动程序	用C 发现系统服务		收集本地系统数据	创建多级信道		禁止系统恢复
			利用第三方软件部 第二員	执行流程劫持	执行流程劫持	除藏行为	注册恶意城控制器	1	t≇	发现域信任	发现系统时间		收集网络共享指动 数据	使用标准非应用层 协议		「网络侧拉绝服务 (DoS)
			利用系统服务	植入容器映像	进程注入	执行流程动持	使用Rootkit	1	网络嗅报 操作系统凭证转储	发现文件和目录	虚拟化/沙箱递送		收集可移动介质数 据	使用非标准端□		资源劫持
			诱导用户执行	修改身份验证过程	利用计划任务/工 作	#166858081#0	执行等名的二进制 文件代理	1	續作派如克祉将18 窃取应用程序访问	发现纠嵌略			微据暂存	使用协议隧道		禁用服务
			利用Windows管 理规范 (WMI)	「自动Office应用程 序	利用有效账户	RT	协行签名的即本代 理	1	令牌 / 治取或伪造身份证	日志校平			收集电子邮件	使用代理		系统关机重用
				电波设置		间接执行命令	撒坏倍任控制	1	验证证书 窃取或伪造	扫描网络服务			输入捕捉	利用运程访问软件		
				在操作系统前启动		<b>进修补偿</b> 权	機械主入	1	Kerberos 先证 窃睬Web会话	发现网络共享			屏幕捕鉄	使用流量信令		
				利用计划任务/工作		依留	使用流量信令	1	Cookie	网络嗅采			初始捕获	利用合法Web服 生		
				利用服务器软件组 件		你成身份验证过程	利用受信的开发工具执行	1	不安全的凭证	发现密码策略				能戰基础设施		
				使用流量信令		潮涌文件或信息	未使用/不受支持 的云区域	1		发现主机接入设备						
				利用有效账户		在操作系统制启动	使用各用身份给证 材料			发现权限组						
								-								

#### Figure 41 Mapping of Technical Features to ATT&CK 4-1

ATT&CK technical behavior description table is as follows.

Table 41 ATT&CK Technical Behavior Description Table 4-1
--

ATT&CK stages / categories	Specific behavior	Notes	
Resource development	Environmental preparation	Upload a malicious item	
Initial access	Intrusions into the supply chain	Using Open Source Supply Chain Attacks	
Execution	Using command and script interpreters	Using VBS and PowerShell	
Execution	Utilization of planned tasks / jobs	Create a scheduled task	
	Anti-obfuscate / decode files or information	Decode the subsequent payload	
Defensive evasion	To weaken the defense mechanism	Close Windows Defender	
	Confusion of documents or information	Encryption payload	
	Virtualization / Sandbox Escape	Detecting a virtual machine environment	
	Circumventing the debugger	Get and circumvent the process	
	Discovery Process	Gets the system process	
Findings	Discovery of system information	Obtain basic information of the system	
	Discover the geographical location of the system	Get the geographical location of the system	
	Virtualization / Sandbox Escape	Detecting a virtual machine environment	



#### Hidden Threats: Analysis of Active "Poisoning" Incidents Disguised as Open-source Projects

Collection	Automatic collection	Automatic access to data		
Collection	Screen capture	Capture a screenshot		
	The application layer protocol is used	Use the HTTP protocol		
Command and control	Take advantage of legitimate Web	Take advantage of online services such as		
	services	Pastebin		
Data coope out	The C2 channel is used for	The C2 channel is used for		
Data seeps out	backtransmission	backtransmission		
Impact	Financial theft	Stealing encrypted wallets		

### 5 Analysis and Evaluation of Antiy LanDi VILLM (Virus Inspection

### Large Language Model)

Antiy LanDi VILLM for automatic analysis of app. asar package samples, the analysis content is as follows:

Based on the provided logs and analysis results, the following conclusions can be drawn:

#### Judgment conclusion

The preliminary conclusion is Trojan. Win32. Bandos Trojan, identified as malware. The following are the detailed judgments and assessments:

#### Accuracy assessment

The accuracy of the results in the evaluation log is approximately 85%. Here's why: Multiple feature segments may appear to be meaningless character data, but contextualized may suggest confusion or the use of encryption techniques. The results contain signs of dangerous calls and malicious behavior.

#### Judgment basis

- 1. Dangerous calls and suspicious behavior:
- The log contains multiple memory operations and function calls, including indications of processing binary data, which may be used for malicious activities such as memory injection (0x48 0xFF 0xC3).
- Information related to "Anti-Virus" appears, possibly related to the function of detecting or circumventing security mechanisms.
- Errors such as "ENAMETOOLONG," "IDECODERERROR" or status information suggest that normal operations may be disrupted when attempting to access or manipulate the file system.
- 2. Characteristic segment analysis:
- Multiple occurrences of ";;; AAAA, O" and other meaningless character sequences that may indicate padding or obfuscating techniques.
- $\circ$  In the code?? The representation data is not selected as a feature, usually when the feature is likely to be encrypted or random data.

- The instruction set and global variables contained in the segment (such as V (60) + e [3] [0] + "y") indicate that there may be dynamic construction and runtime code generation, which are common methods used by Trojans.
- 3. Overall logic and behaviour:
- The log ends with multiple buffering and pointer operations, a common tool used by many Trojans to modify process memory or load malicious code.
- The inconsistency between code snippets or comments from JavaScript and C + + in certain 0 paragraphs (for example, AVISOATE @ 2 @ XZ) can surface software attempts to obfuscate the source or purpose.

#### Other judgment basis

Invalid character sets and symbols: Random characters seen in many parts of a character, indicating coding transformation or confusion.

Response time and analysis speed: Normal detection requires considerable computational power, indicating high complexity, and may be malware that improves through multiple iterations.

#### Additional instructions

Since part of the data is not selected as a feature, the final conclusion relies on the interpretation and analysis of the data seen, so there may be some bias. In the absence of more in-depth file analysis (e.g. reverse engineering), this result is a preliminary decision.

#### General recommendations

More detailed static and dynamic analyses are recommended. If confirmed as malicious, immediately isolate the found files and perform a full system scan to confirm system integrity and security.

According to the national standard draft, the threat classification is Trojan. Win32. Bandos, and the YARA rules

for automatic feature extraction are as follows:

Rule Trojan. Win32.bandos: Trojan {

Meta:

Description = "This is a section of YARA rules generated by the VILLM V2 model. used to detect Trojan.

Win32.bandos"

Strings:

\$a = {24 64 65 66 65 6e 64 72 45 78 63 6c 75 73 69 6f 6e 73 2e 45 78 63 6c 75 69 6f 6e 50 72 6f 63 65 73 }  $b = \{50, 72, 66, 76, 61, 60, 73, 5c, 43, 66, 66, 66, 5c, 46, 66, 44, 72, 69, 65, 43, 6c, 66, 75, 64, 5c, 5c, 6d, 62, 61, 6d, 2e, 70, 73, 31\}$ \$c = {68 74 74 70 73 3a 2f 2f 61 70 69 2e 74 65 6c 65 67 72 61 6d 2e 6f 72 67 2f 62 6f 74 22 2e 63 6f 6e 63 61 74 28}

Condition:

All of them

}

# デーマテス Hidden Threats: Analysis of Active "Poisoning" Incidents Disguised as Open-source Projects

Antiy LanDi VILLM for Threat Detection and Analysis is the first threat detection generative model registered by the State Cyberspace Administration in China. The model is trained based on the massive sample feature engineering data accumulated over the past 20 years by Antiy Cybertron. The training data includes file identification information, decision information, attribute information, structure information, behavior information, host environment information, data information, and the like, The system supports threat judgment and detailed knowledge understanding of vector features under different scenarios, forms multi-form detection methods applying different requirements and scenarios, and improves the ability to judge hidden threats in the background. Further empowering safe operations.



Figure 5-1 Antiy LanDi VILLM sample analysis result 51

### 6 IoCs

Url
Https://rlim[.] com/seraswodinsx/raw
Https: / / pastebin [.] com / raw / LC0H4rhJ
Https: / / github [.] com / unheard44 / fluid _ bean / releases / download / releases / SearchFilter.7z

#### Md5

© Antiy all rights reserved.

# **夏天** Hidden Threats: Analysis of Active "Poisoning" Incidents Disguised as Open-source Projects

19a2aba4e6b2c96c45a404a35ac9f302	976d02b2567125131c707c03c97f4593
	9c9db4c1f98a6e2a89e104af803e80c7
3829e837f6d29c7b2fa8e06c798d7eac	A0a162a82e0ca0f43643fc842b7d3775
	A0ee88e4f69c3b97b86b86a73f93e2eb
48f75bfcc571eab5318c99de1dff2543	B41fbcb71c23e469bcdb94c8692 B7418
	B71c0960d6ab4f6332595bdebebcaf5a
4fabe1abae75be0c4da16e440d0e3f84	Bd11d5da183fa3dd7bf923073e305a32
	C0f503a88bb0568cbc37169c2da4e6f8
59f25c363c0dbc61d63f6968e180055	C332b4dc17f962dc5d856e3ae5025303
	D12f585dbac74fd2445b47447a10def0
5e39a413a2d83edc484541313fbbdb1f	E1df5b5e9812c5d65f1e5893a668112e
	E335e6a1d22702feed2367ddbc30da2
6e5ae6d2c1ef55b817d474c1019d8e8c	F604752dd982930e8d0412f8b2aa817c
	F7be2caa2d0c3dd06d8d2a32ebf243b7
8c91be158349799d93bd1d384002465b	Fb5a9459cfd2f1c0db9bdcd90c11e7cb
	837b9b6a3e38ad1a6c58cf9130b28da9
C964f701ccb7b17776e21a9082f9e3b2	

### **Appendix I: Reference**

 The GitHub open source project was poisoned, and the backdoor virus spread and spread following the development process [R / OL]. (2025-01-23)

Https://mp.weixin.qq.com/s/MF2lvyH6BxBE\_muCwkOCPg

## **Appendix II: About Antiy**

Anty is committed to enhancing the network security defense capabilities of its customers and effectively responding to security threats. Through more than 20 years of independent research and development, Antiy has developed technological leadership in areas such as threat detection engines, advanced threat countermeasures, and large-scale threat automation analysis.

Antiy has developed IEP (Intelligent Endpoint Protection System) security product family for PC, server and other system environments, as well as UWP (Unified Workload Protect) security products for cloud hosts, container and

# 「安天 Hidden Threats: Analysis of Active "Poisoning" Incidents Disguised as Open-source Projects

other system environments, providing system security capabilities including endpoint antivirus, endpoint protection (EPP), endpoint detection and response (EDR), and Cloud Workload Protection Platform (CWPP), etc. Antiy has established a closed-loop product system of threat countermeasures based on its threat intelligence and threat detection capabilities, achieving perception, retardation, blocking and presentation of the advanced threats through products such as the Persistent Threat Detection System (PTD), Persistent Threat Analysis System (PTA), Attack Capture System (ACS), and TDS. For web and business security scenarios, Antiy has launched the PTF Next-generation Web Application and API Protection System (WAAP) and SCS Code Security Detection System to help customers shift their security capabilities to the left in the DevOps process. At the same time, it has developed four major kinds of security service: network attack and defense logic deduction, in-depth threat hunting, security threat inspection, and regular security operations. Through the Threat Confrontation Operation Platform (XDR), multiple security products and services are integrated to effectively support the upgrade of comprehensive threat confrontation capabilities.

Antiy provides comprehensive security solutions for clients with high security requirements, including network and information authorities, military forces, ministries, confidential industries, and critical information infrastructure. Antiy has participated in the security work of major national political and social events since 2005 and has won honors such as the Outstanding Contribution Award and Advanced Security Group. Since 2015, Antiy's products and services have provided security support for major spaceflight missions including manned spaceflight, lunar exploration, and space station docking, as well as significant missions such as the maiden flight of large aircraft, escort of main force ships, and Antarctic scientific research. We have received several thank-you letters from relevant departments.

Antiy is a core enabler of the global fundamental security supply chain. Nearly a hundred of the world's leading security and IT enterprises have chosen Antiy as their partner of detection capability. At present, Antiy's threat detection engine provides security detection capabilities for over 1.3 million network devices and over 3 billion smart terminal devices worldwide, which has become a "national-level" engine. As of now, Antiy has filed 1,877 patents in the field of cybersecurity and obtained 936 patents. It has been awarded the title of National Intellectual Property Advantage Enterprise and the 17th (2015) China Patent Excellence Award.

Antiy is an important enterprise node in China emergency response system and has provided early warning and comprehensive emergency response in major security threats and virus outbreaks such as "Code Red", "Dvldr", "Heartbleed", "Bash Shellcode" and "WannaCry". Antiy conducts continuous monitoring and in-depth analysis against dozens of advanced cyberspece threat actors (APT groups) such as "Equation", "White Elephant", "Lotus"



and "Greenspot" and their attack actions, assisting customers to form effective protection when the enemy situation is accurately predicted.