# Cyberspike Villager – Cobalt Strike's AI-native Successor


☼ / ☾

☼ / ☾

## Signal Check ⚡

*Executive Summary of Findings*

> Straiker's AI Research (STAR) team recently uncovered Villager, an AI-native penetration testing framework in the wild by the Chinese-based group Cyberspike. Originally positioned as a red-team offering, Cyberspike has released an AI-enabled, MCP-supported automation tool called "Villager" that combines Kali Linux toolsets with DeepSeek AI models to fully automate testing workflows. The package is published in PyPI.org and has recorded ~10,000 downloads in two months. The rapid, public availability and automation capabilities create a realistic risk that Villager will follow the Cobalt Strike trajectory: commercially or legitimately developed tooling becoming widely adopted by threat actors for malicious campaigns.

### Key facts

1. **What?** Villager — an AI-driven automation layer for pentesting workflows (integrates with Kali and Deepseek).
2. **From where?** Developed from a Chinese-based, red-team project under the Cyberspike name.
3. **What is the scale?** Available on the official Python Package Index, Villager is freely accessible worldwide and has seen 10,000+ downloads within its first two months.
4. **Why does it matter?** Low barrier to access + powerful automation = high potential for dual-use abuse.

### How can Villager lead to AI-powered attacks?

1. Dual-use capability: Villager reduces skill and time required to run sophisticated offensive toolchains, enabling less-skilled actors to perform more advanced intrusions.
2. Distribution channel risk: Publishing on the official Python repository gives attackers a convenient, trusted supply chain vector to obtain and potentially integrate the tool.

3. Cobalt Strike precedent: Legitimate red-team tools have historically been repurposed by criminal/APT groups; the combination of automation + easy access raises the probability of similar misuse and broader dissemination.
4. Operational impact: Increased frequency and speed of automated reconnaissance, exploitation attempts, and follow-on activity could raise detection and response burdens across the enterprise.

### What are the potential impacts to enterprises?

1. More frequent and automated external scanning and exploitation attempts.
2. Faster attack lifecycles that compress detection and response windows.
3. Greater use of off-the-shelf tools in blended attacks, complicating attribution and response.
4. Increased supply-chain and developer environment exposure if packages are installed in CI/CD or dev workstations.

### Recommendations

1. **Implement MCP Protocol Security Gateway** - Deploy real-time inspection and filtering of Model Context Protocol communications to detect malicious tool invocation patterns and unauthorized AI agent behaviors
2. **Review Third-Party AI Integrations** - Audit all organizational use of AI tools, MCP implementations, and external AI services to identify potential attack vectors
3. **Establish AI Governance Framework** - Create policies governing internal AI tool usage, including approval processes for AI-powered security testing and development tools
4. **Develop AI Threat Intelligence Capabilities** - Build expertise to track AI-enhanced threat actor TTPs, emerging MCP-based attack tools, and World threat actor AI weaponization trends
5. **Create Incident Response Playbooks** - Develop specific procedures for AI-enhanced attacks, including rapid containment of automated exploitation and evidence preservation for AI-driven campaigns
6. **Build Continuous Agentic Security Testing** - Establish ongoing red team exercises specifically targeting MCP-enabled applications with AI-driven attack scenarios to validate defensive controls

## Who is Cyberspike?

Cyberspike first appeared on **November 27, 2023**, when the domain `cyberspike[.]top` was registered under `Changchun Anshanyuan Technology Co., Ltd.`, a Chinese company listed as an Artificial Intelligence and Application Software Development provider.

**cyberspike.top registration query**

Latest cyberspike.top ICP filing query

| | |
|---|---|
| Filing Type | enterprise |
| Filing Entity | Changchun Anshanyuan Technology Co., Ltd. |
| Registration Number | JICP No. 2023008137-2 |
| Filing time | 2024-01-26 to 2025-09-05 |

Figure 1: Src: https://icplishi.com/cyberspike.top/

Interestingly, when looking at a website similar to LinkedIn in China, there is no information about the website of the Company, only this description (See Figure 2).

Figure 2: https://www.liepin.com/company/gs109346865/

The complete absence of any legitimate business traces for "Changchun Anshanyuan Technology Co., Ltd" along with no web site available raise some concerns about who is behind running “Red Team Operations” with an automated tool.

Although Cyberspike currently lacks an official website, archived snapshots from the Wayback Machine provide insight into their 2023 product offerings. These historical records confirm that the company registration number displayed on their former webpage matches the official registration for Changchun Anshanyuan (see Figure 3). The archived content was in Chinese, requiring translation for analysis.

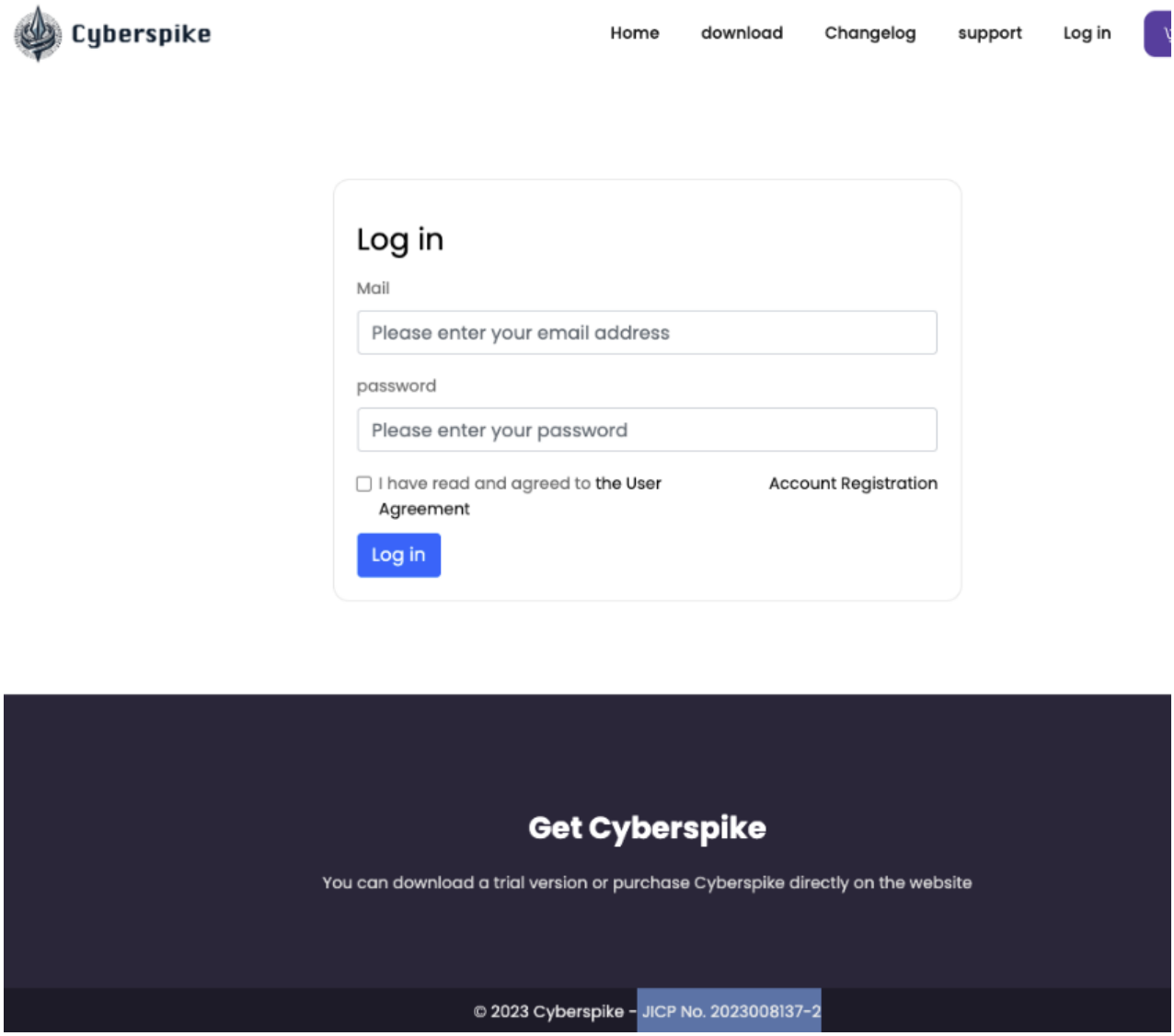


Figure 3: Registration number JICP

At Figure 4, we can see the company was selling a product named: **Cyberspike**.

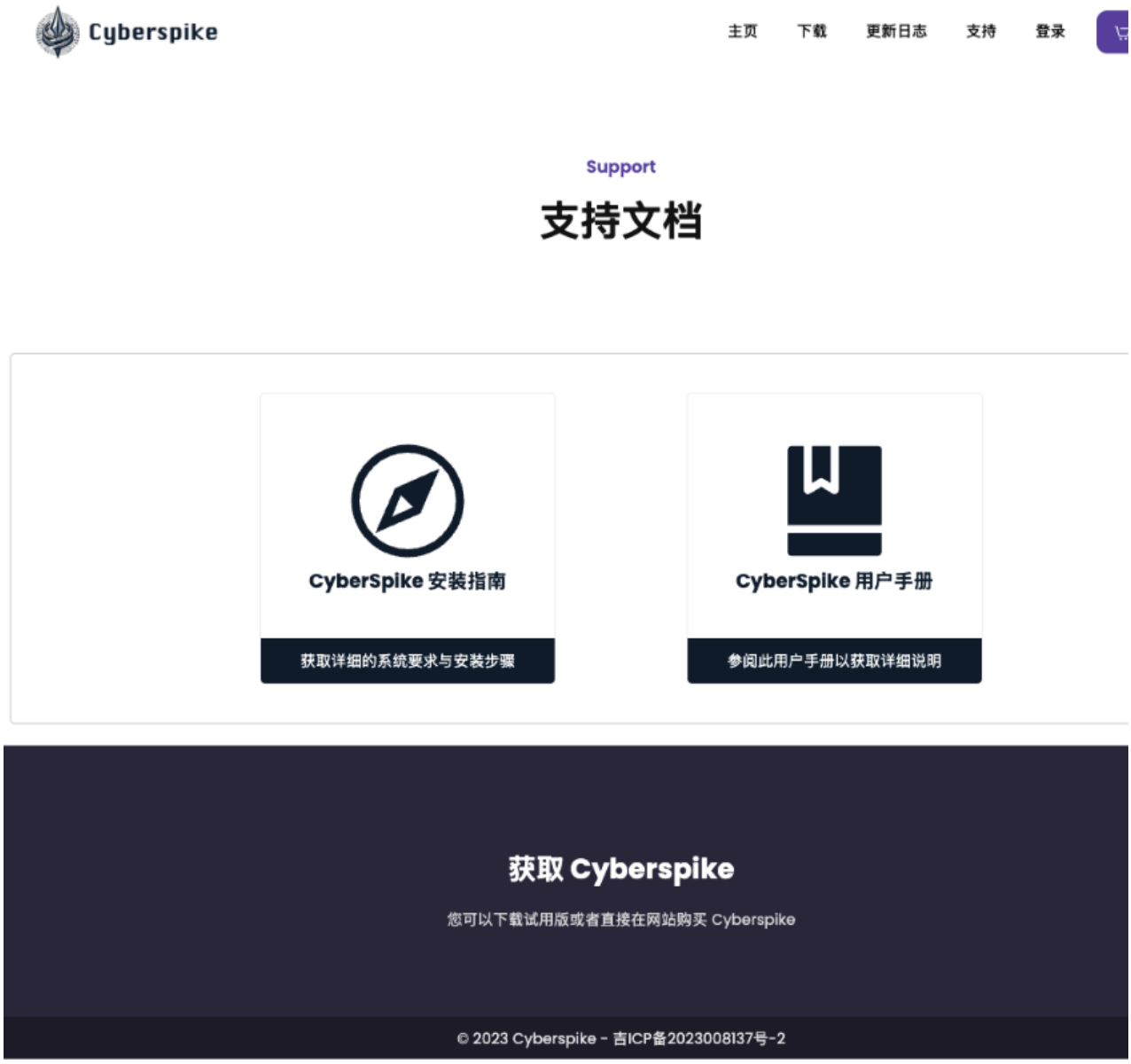


Figure 4: “You can download a trial version or purchase …”

We recovered the Dashboard view of the product, basically showing stats related to the machines being attacked, as shown at Figure 5.


Figure 5: Src: https://web.archive.org/web/20231211145137/https://www.cyberspike.top/static/images/cs_bg_screen.png

Then, the latest features from Cyberspike product can be seen, highlighting some related to a Remote Administration Tool (translated from Chinese) as follows:

- V1.1.7: Dec 10, 2023
  - Launched built-in reverse proxy
- V1.1.0: Nov 19, 2023:
  - Built-in multi-stage generator
  - Other common C2 Tools functions


Figure 6: Promoting latest RAT features

We traced the **Cyberspike Studio Installer v1.1.7** to a VirusTotal submission on December 10, 2023 (hash 40127d53ee0554fa997240fc37860a79).

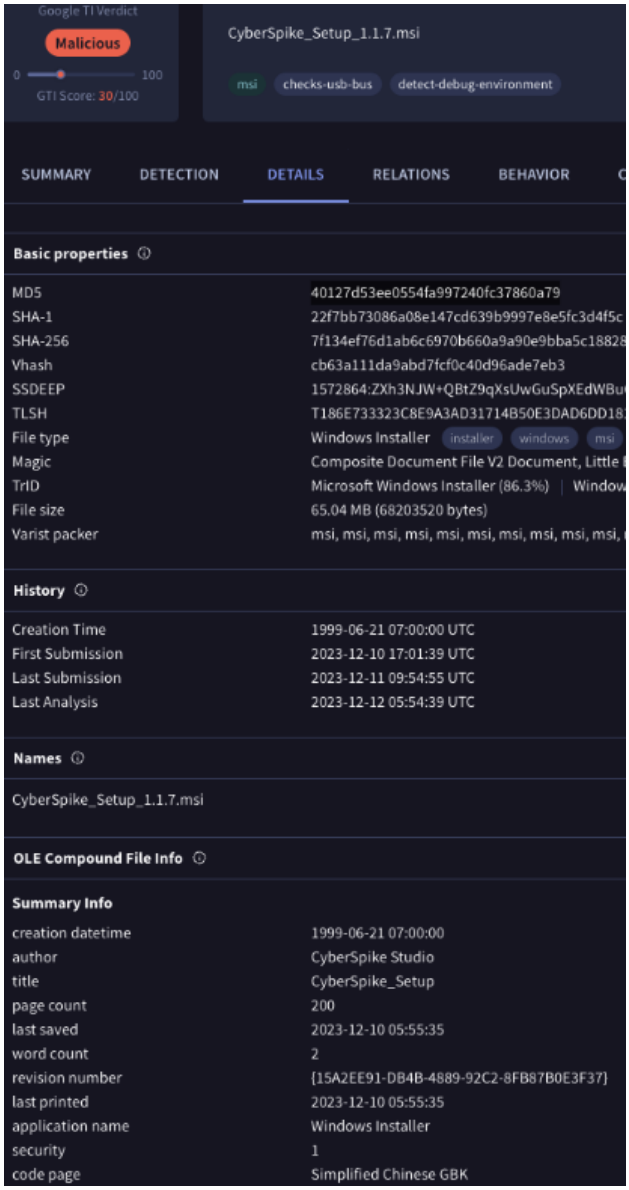


Figure 7: Cyberspike_Setup_1.1.7.msi

Upon installation and analysis of the Cyberspike Studio Installer v1.1.7 product, we discovered that all included plugins were components of a Remote Administration Tool (RAT) designed for comprehensive victim surveillance and control. The malicious capabilities included remote desktop access, Discord account compromise, keystroke logging, webcam hijacking, and other invasive monitoring functions, as demonstrated in Figure 8.

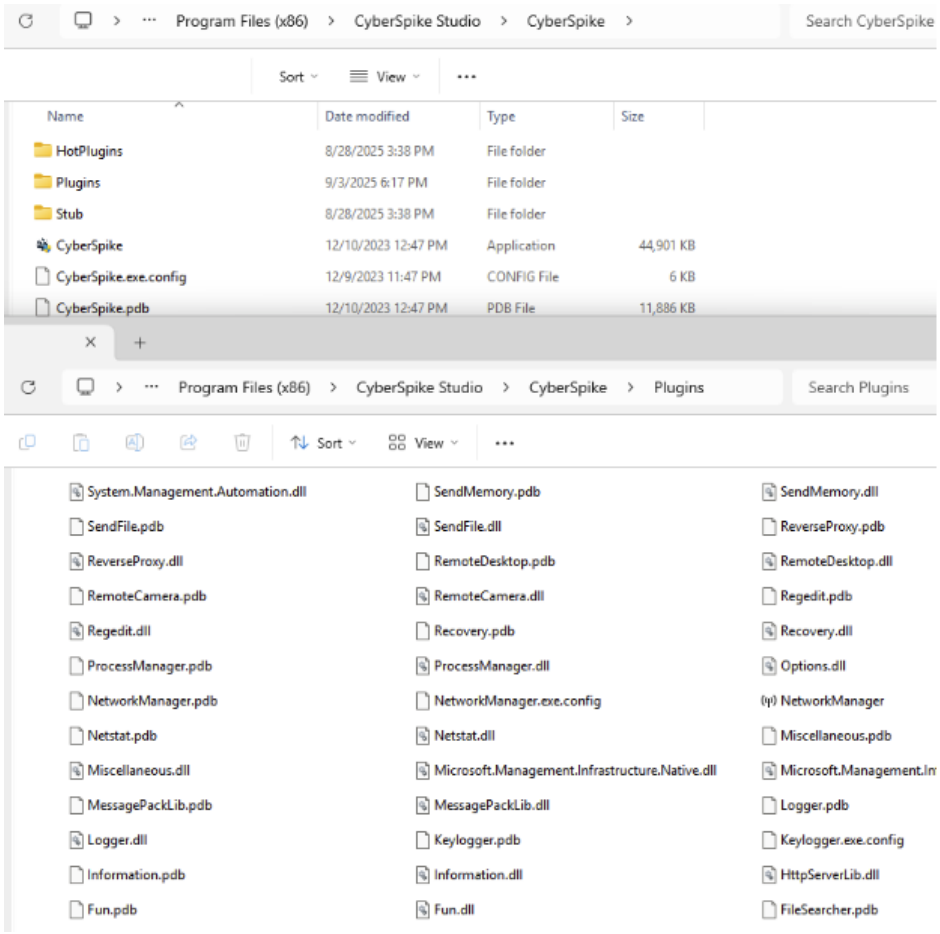


Figure 8: The Cyberspike Suite

When running the main program Cyberskpike.exe, an authentication form is displayed:

Figure 9: Access to Cyberspike Suite

The malware attempts authentication to a Chinese-hosted IP address (**111[.]173[.]106[.]69**) which was unresponsive during our testing period. However, passive DNS historical analysis reveals this IP previously resolved to the URL `www[.]cyberspike[.]top`.

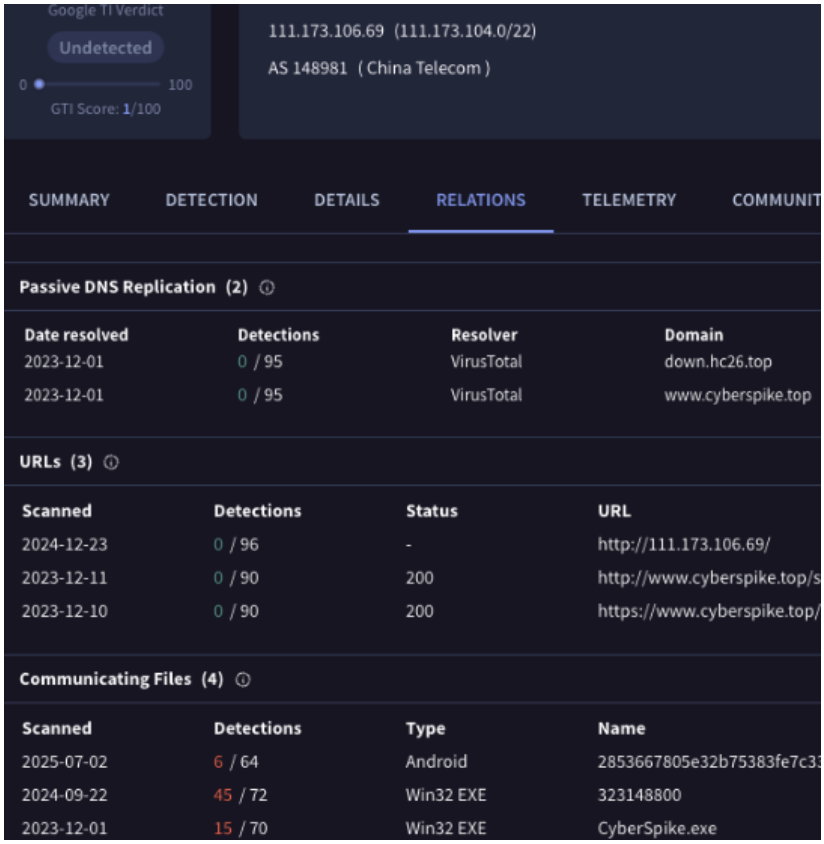
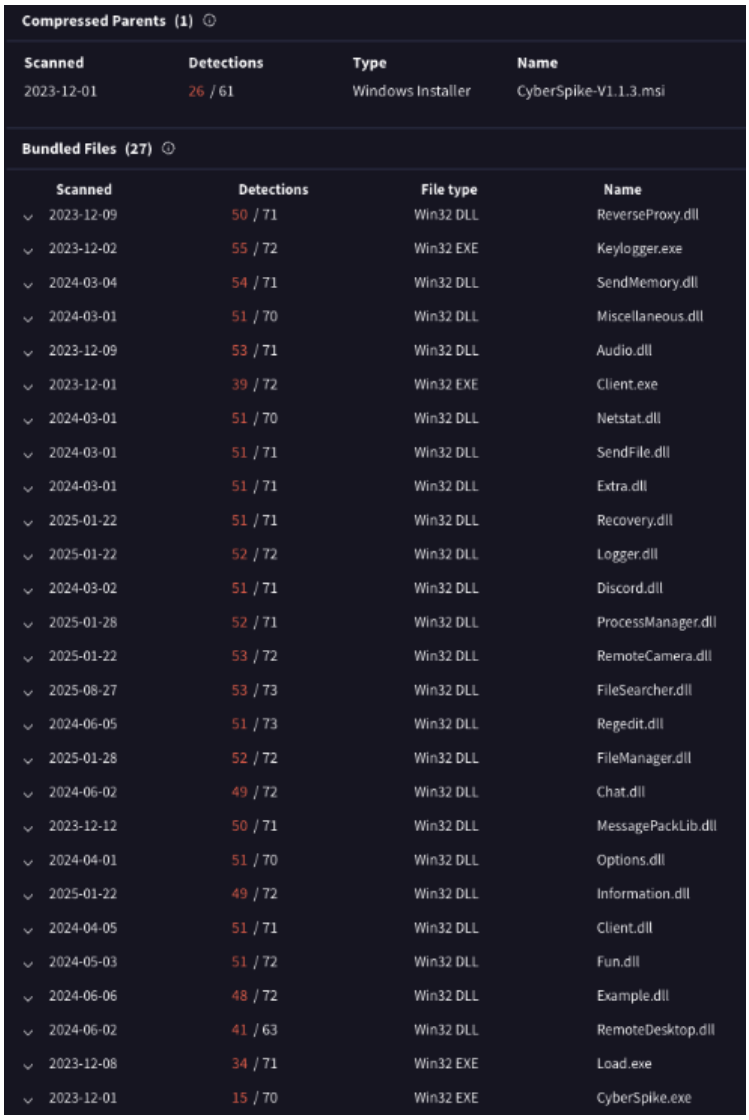


Figure 10

A deeper look into VirusTotal revealed that the entire Cyberspike toolset and arsenal had also been uploaded.



**Compressed Parents (1)**

| Scanned | Detections | Type | Name |
|---|---|---|---|
| 2023-12-01 | 26 / 61 | Windows Installer | CyberSpike-V1.1.3.msi |

**Bundled Files (27)**

| Scanned | Detections | File type | Name |
|---|---|---|---|
| 2023-12-09 | 50 / 71 | Win32 DLL | ReverseProxy.dll |
| 2023-12-02 | 55 / 72 | Win32 EXE | Keylogger.exe |
| 2024-03-04 | 54 / 71 | Win32 DLL | SendMemory.dll |
| 2024-03-01 | 51 / 70 | Win32 DLL | Miscellaneous.dll |
| 2023-12-09 | 53 / 71 | Win32 DLL | Audio.dll |
| 2023-12-01 | 39 / 72 | Win32 EXE | Client.exe |
| 2024-03-01 | 51 / 70 | Win32 DLL | Netstat.dll |
| 2024-03-01 | 51 / 71 | Win32 DLL | SendFile.dll |
| 2024-03-01 | 51 / 71 | Win32 DLL | Extra.dll |
| 2025-01-22 | 51 / 71 | Win32 DLL | Recovery.dll |
| 2025-01-22 | 52 / 72 | Win32 DLL | Logger.dll |
| 2024-03-02 | 51 / 71 | Win32 DLL | Discord.dll |
| 2025-01-28 | 52 / 71 | Win32 DLL | ProcessManager.dll |
| 2025-01-22 | 53 / 72 | Win32 DLL | RemoteCamera.dll |
| 2025-08-27 | 53 / 73 | Win32 DLL | FileSearcher.dll |
| 2024-06-05 | 51 / 73 | Win32 DLL | Regedit.dll |
| 2025-01-28 | 52 / 72 | Win32 DLL | FileManager.dll |
| 2024-06-02 | 49 / 72 | Win32 DLL | Chat.dll |
| 2023-12-12 | 50 / 71 | Win32 DLL | MessagePackLib.dll |
| 2024-04-01 | 51 / 70 | Win32 DLL | Options.dll |
| 2025-01-22 | 49 / 72 | Win32 DLL | Information.dll |
| 2024-04-05 | 51 / 71 | Win32 DLL | Client.dll |
| 2024-05-03 | 51 / 72 | Win32 DLL | Fun.dll |
| 2024-06-06 | 48 / 72 | Win32 DLL | Example.dll |
| 2024-06-02 | 41 / 63 | Win32 DLL | RemoteDesktop.dll |
| 2023-12-08 | 34 / 71 | Win32 EXE | Load.exe |
| 2023-12-01 | 15 / 70 | Win32 EXE | CyberSpike.exe |

Figure 11: Cyberspike Suite on VirusTotal

Subsequent similarity analysis of these binaries revealed that the entire software suite was actually version 1.0.7.0 related to AsyncRAT, a well-established and widely-recognized Remote Access Trojan.

AsyncRAT gained widespread attention following its 2019 release on GitHub. Since then, it has become a preferred tool among attackers and continues to actively threaten organizations worldwide. Numerous RAT variants have emerged based on AsyncRAT, including but not limited to: DCRat and VenomRAT. The similarities between these variants commonly cause confusion in anti-virus detection systems.

The comparative analysis table below clearly demonstrates that the **Cyberspike** and **AyncRAT** components are not only functionally identical but also physically the same binaries—sharing identical file formats, programming languages, file sizes, and operational purposes, as evidenced by our examination of three representative components from the software suite:

| Cyberspike | | | | AsyncRAT | | | |
|---|---|---|---|---|---|---|---|
| Discord.dll / 1.0.7.0 | | | | | | | |
| 93d9d7d0ad423f1b4ff607b14edbca48 | | | | 46ff79eacaa4e9cebceb87c57e9edb6e | | | |
| **Type** | **Size** | **First Seen** | **PDB Path** | **Type** | **Size** | **First Seen** | **PDB Path** |
| .NET | 24.50 KB | Dec 10, 2023 | D:\代码\VS\CyberSpike\Binaries\Release\Plugins\Discord.pdb | .NET | 24.50 KB | Jan 12, 2022 | D:\a\DcRat\DcRat\Base\Plugins\Discord.p |
| Keylogger / 1.0.7.0 | | | | | | | |
| 5f9d29dfb766d86d2f10cd57ac9f291c | | | | 1c751dfcd67807c28a861096abe90e7e | | | |
| .NET | 10 KB | Dec 10, 2023 | D:\代码\VS\CyberSpike\Binaries\Release\Plugins\Keylogger.pdb | .NET | 10 KB | May 6, 2021 | C:\Users\Eddie Toth\Desktop\DcRat Keylogger\Keylogge Keylogger.pdb |
| RemoteCamera / 1.0.7.0 | | | | | | | |
| 1cd2946849f3db9fdaed7998f23e931f | | | | 5666b21c4b89714e33d40f39e30fb28e | | | |
| .NET | 107 KB | Dec 01, 2023 | D:\代码\VS\CyberSpike\Binaries\Release\Plugins\RemoteCamera.pdb | .NET | 107 KB | Apr 17, 2021 | C:\Users\28718\Doc ub\DcRat\Binaries\R ns\RemoteCamera.p |

Figure 12: Connection between Cyperspike and AsyncRAT

Our analysis confirms that Cyberspike integrated AsyncRAT into its red teaming product, with additional plugins to well-known hacktools like Mimikatz as well (see Figure 13). These integrations demonstrate how Cyberspike repackaged established hacktools and offensive tools into a turnkey framework designed for penetration testing and probably malicious operations.

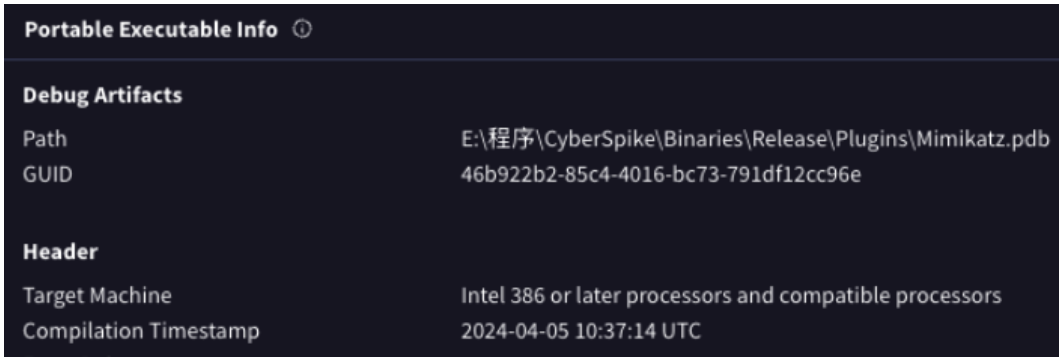



Figure 13: Mimikatz Plugin in Cyberspike

### Recent activities from Cyberspike

On **July 23, 2025,** the Cyperspike team released the **Villager** Pentesting Tool on Python Package Index, a legitimate repository of software for the Python programming language.

Through subdomain enumeration of cyberspike[.]top, we discovered the Python package, which automates penetration testing operations using Deepseek AI models. Within the package, a testing script references a custom model designated "**ai-1s-20250421**" hosted at:

`http://gpus[.]dev[.]cyberspike.top:8000/v1/chat/completions`

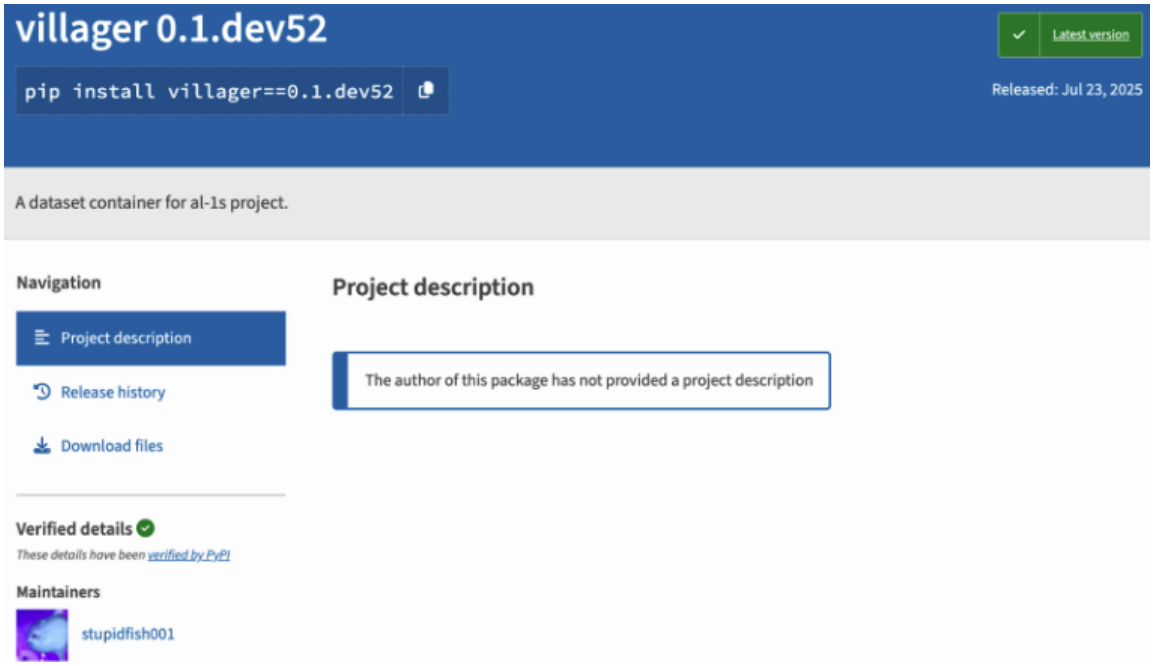



Figure 14. Python Villager released

The author `@stupidfish001` is a former CTF player for the Chinese HSCSEC Team. He is also the creator of multiple software related to a project name "**al-s1**":

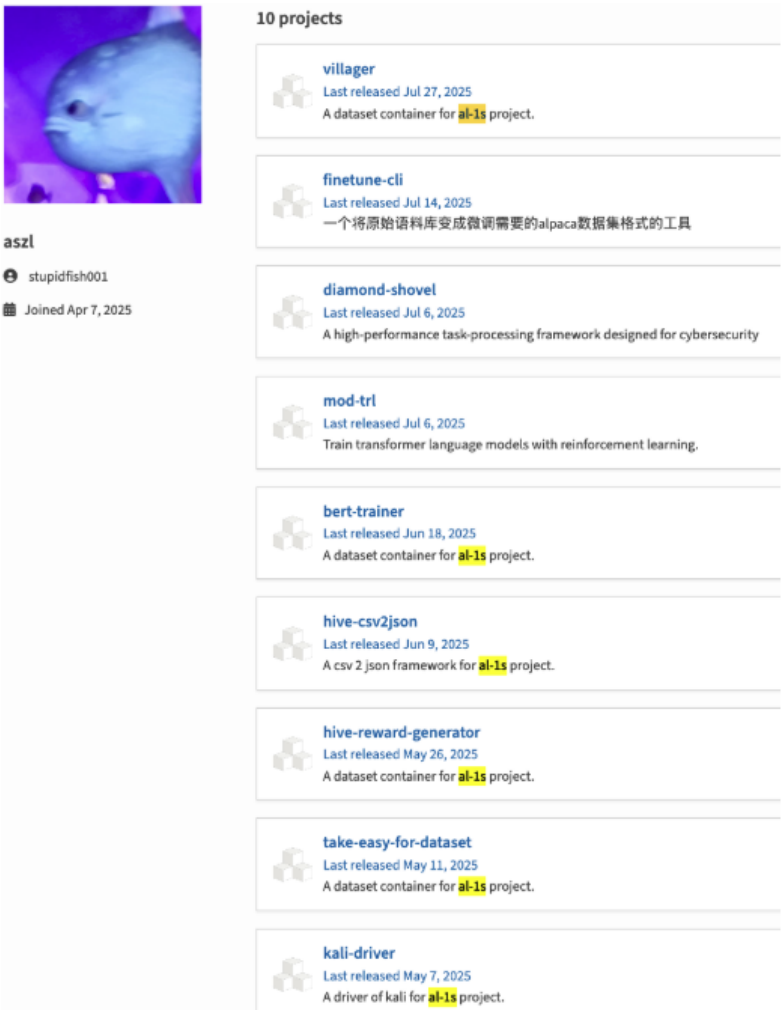



Figure 15

He is registered in Python repo with the email: `shovel@hscsec.cn` and also uses the email `diamondshovel@cyberspike.top` as one of the maintainers at replogy.org:


Figure 16: Src: https://repology.org/maintainer/diamondshovel%40cyberspike.top

Since its release two months ago, Villager has accumulated 10,030 downloads across multiple operating systems, including Linux, macOS, and Windows.:

```
pypistats overall villager
```

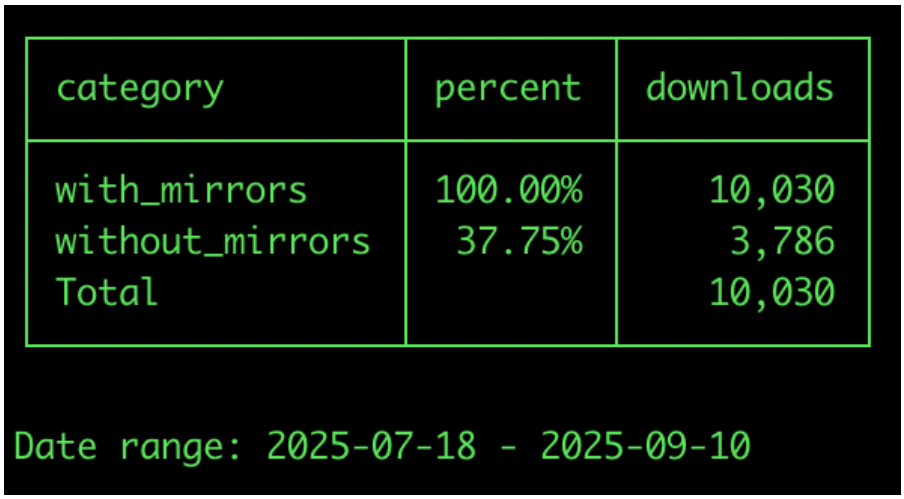



Figure 17: Villager downloads stats

We noticed an average rate of 200+ downloads every 3 days during our investigation.

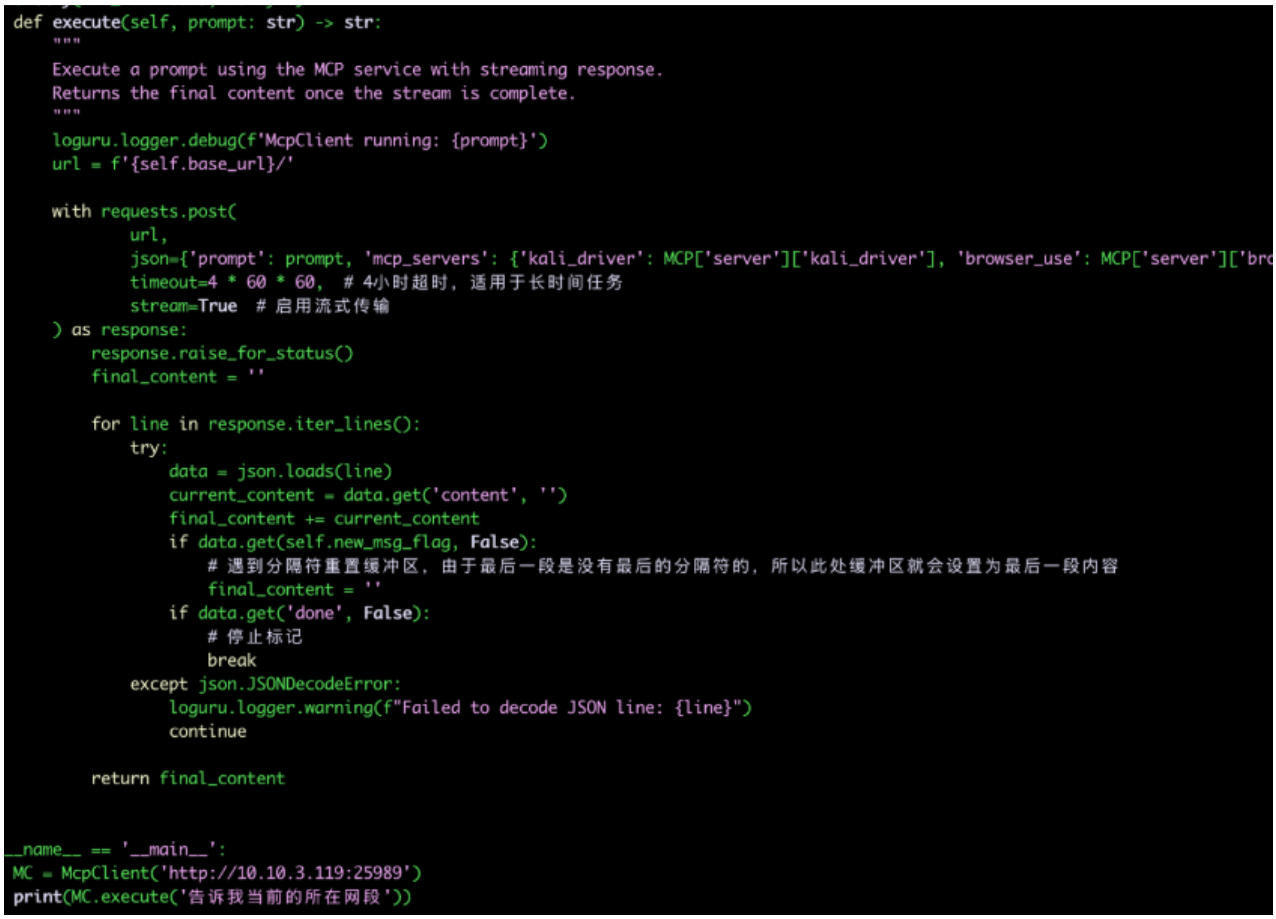
Villager operates as an MCP client, integrating multiple security tools. Most notably the 'Kali Driver' integration automates penetration testing using Kali Linux toolsets, and in adversarial hands, can unleash containerized Kali Linux attacks at scale. This component is deployed as a containerized Docker image hosted on GitLab at:

```
gitlab.cyberspike.top:5050/aszl/diamond-shovel/al-1s/kali-image:main
```

## Deep Dive into Villager's AI Pentesting Framework and Functionality

The Villager AI pentesting framework implements a distributed architecture consisting of multiple service components:

- **MCP Client Service** (Port 25989): Central message passing and coordination.



```
def execute(self, prompt: str) -> str:
    """
    Execute a prompt using the MCP service with streaming response.
    Returns the final content once the stream is complete.
    """
    loguru.logger.debug(f'McpClient running: {prompt}')
    url = f'{self.base_url}/'

    with requests.post(
            url,
            json={'prompt': prompt, 'mcp_servers': {'kali_driver': MCP['server']['kali_driver'], 'browser_use': MCP['server']['bro
            timeout=4 * 60 * 60,  # 4小时超时，适用于长时间任务
            stream=True  # 启用流式传输
    ) as response:
        response.raise_for_status()
        final_content = ''

        for line in response.iter_lines():
            try:
                data = json.loads(line)
                current_content = data.get('content', '')
                final_content += current_content
                if data.get(self.new_msg_flag, False):
                    # 遇到分隔符重置缓冲区，由于最后一段是没有最后的分隔符的，所以此处缓冲区就会设置为最后一段内容
                    final_content = ''
                if data.get('done', False):
                    # 停止标记
                    break
            except json.JSONDecodeError:
                loguru.logger.warning(f"Failed to decode JSON line: {line}")
                continue

        return final_content


__name__ == '__main__':
MC = McpClient('http://10.10.3.119:25989')
print(MC.execute('告诉我当前的所在网段'))
```

Figure 18. Villager's MCP Client

- **RAG Enhanced Decision Making:** Leverages a database of 4,201 AI system prompts to generate exploits and make real-time decisions in penetration testing.
- **On-Demand Container Creation**: Automatically creates isolated Kali Linux containers when it needs cybersecurity tools for network scanning, vulnerability assessment, and penetration testing.
- **Enhanced AI Orchestration with Pydantic:** Uses Pydantic AI to enforce strict formatting rules on AI outputs, ensuring reliable and predictable responses for task management and decision-making.
- **Forensic Evasion**: Containers are configured with a 24-hour self-destruct mechanism that wipes activity logs and evidence. The ephemeral nature of these containers, combined with randomized SSH ports, makes AI-powered attack containers difficult to detect, complicating forensic analysis and threat attribution.

## The Cyberspike Villager's Architecture Diagram

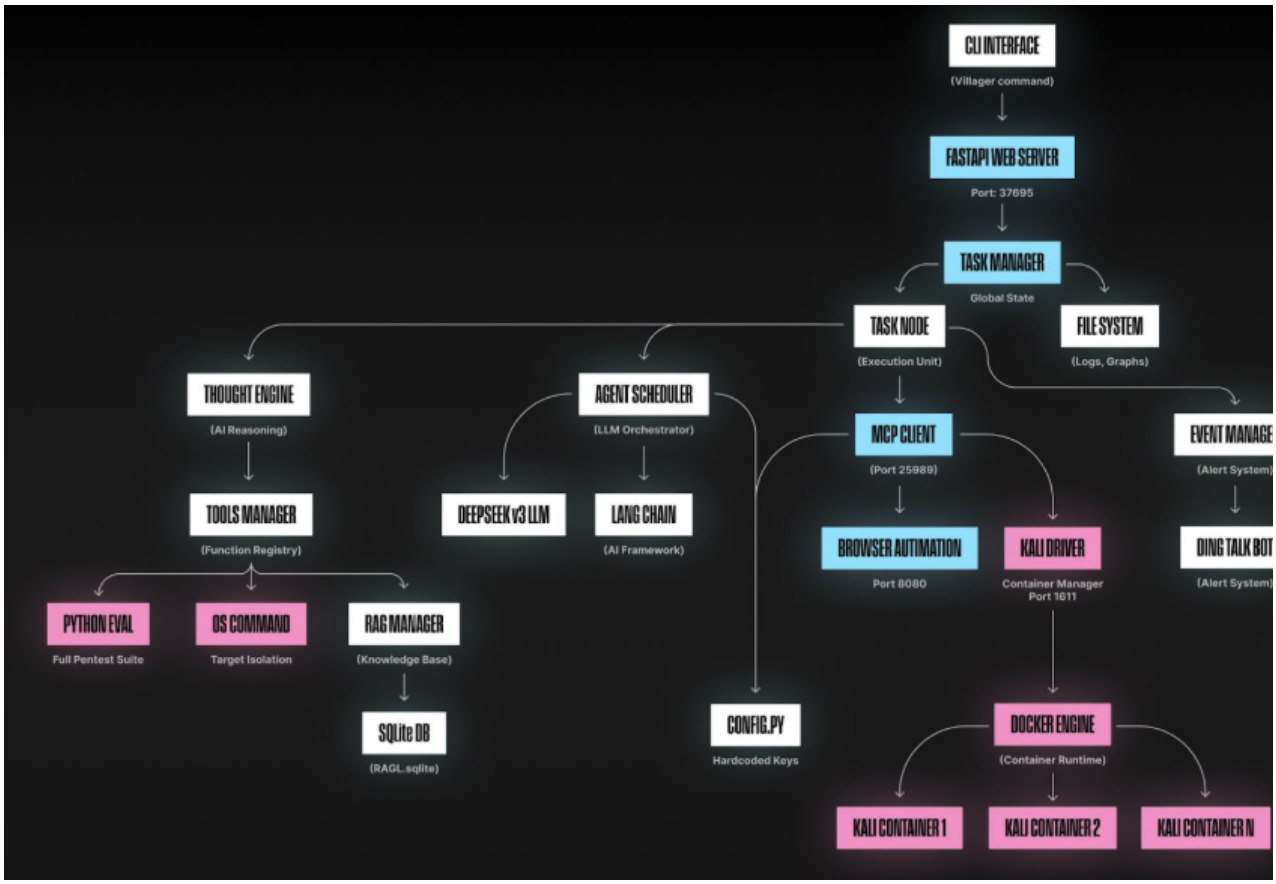


Figure 19. Villager orchestration overview

Unlike traditional pentesting frameworks like Cobalt Strike that rely on scripted playbooks, the Villager is an AI-native penetration testing framework and an early example of AI-powered exploitation. The integration with LangChain and the DeepSeek v3 language model (accessed through OpenAI-compatible API endpoints) provides natural language processing capabilities, allowing operators to issue commands in plain-text that are automatically translated into technical and dynamic attack sequences.

An API testing script comes with the python package pointing to a dev environment at `http://gpus[.]dev[.]cyberspike.top:8000/v1/chat/completions` using what looks like a custom model called “al-1s-20250421” using OpenAI GPT-3.5-turbo as the tokenizer.

```
API_URL = "http://gpus.dev.cyberspike.top:8000/v1/chat/completions"
MODEL_NAME = "al-1s-20250421/"
TEST_ROUNDS = 100
MAX_WORKERS = 10

enc = tiktoken.encoding_for_model("gpt-3.5-turbo")
```

*Src: villager-0.1.dev52/test/unitest/api_test.py*

Based on task requirements, the Villager MCP client dynamically interfaces with available tools:

- **Kali Driver** (Port 1611): Provides on-demand containerized Kali Linux environments for network operations. The image is loaded from `gitlab[.]cyberspike[.]top` domain:

```
class KaliContainer:
    def __init__(self, uuid, owner, host):
        self._uuid = uuid
        self._owner = owner
        self._host = host
```

```
        self._container =
owner._docker_client.containers.create(image="gitlab.cyberspike.top:5050/aszl/diamond-
shovel/al-1s/kali-image:main", command="/usr/sbin/sshd -D", ports={"22/tcp": None},
detach=True)
        self._ssh_connection = None
```

*Src: kali_driver-0.0.0/al1s/drivers/kali/driver.py*

- **Browser Automation** (Port 8080): Handles web-based interactions and client-side testing.
- **Direct Code Execution**: Uses pyeval() and os_execute_cmd() for system-level operations.

## A Look into Villager's Task-Based C2 Architecture

Villager implements a sophisticated task-based command and control (C2) system through its FastAPI interface at port 37695:

Operators submit high-level objectives and tasks through natural language commands:

```
POST /task
{
  "abstract": "Test example.com for vulnerabilities",
  "description": "Perform comprehensive security assessment",
  "verification": "Provide list of exploitable vulnerabilities"
}
```

### Intelligent Task Management

The framework's TaskRelationManager handles:

- **Task Decomposition**: Complex objectives are automatically broken into subtasks
- **Dependency Tracking**: Ensures subtasks execute in proper sequence
- **Failure Recovery**: Failed tasks trigger re-planning through the AI model
- **Parallel Execution**: Independent subtasks run simultaneously

Villager leverages Pydantic AI to standardize outputs, guaranteeing consistent responses for task management and multi-step decision-making, which is critical for AI-driven penetration testing workflows.

"If necessary, provide task chains in sequence and ensure the continuity of the task chain so that the results of the previous task can be directly used by the next task"

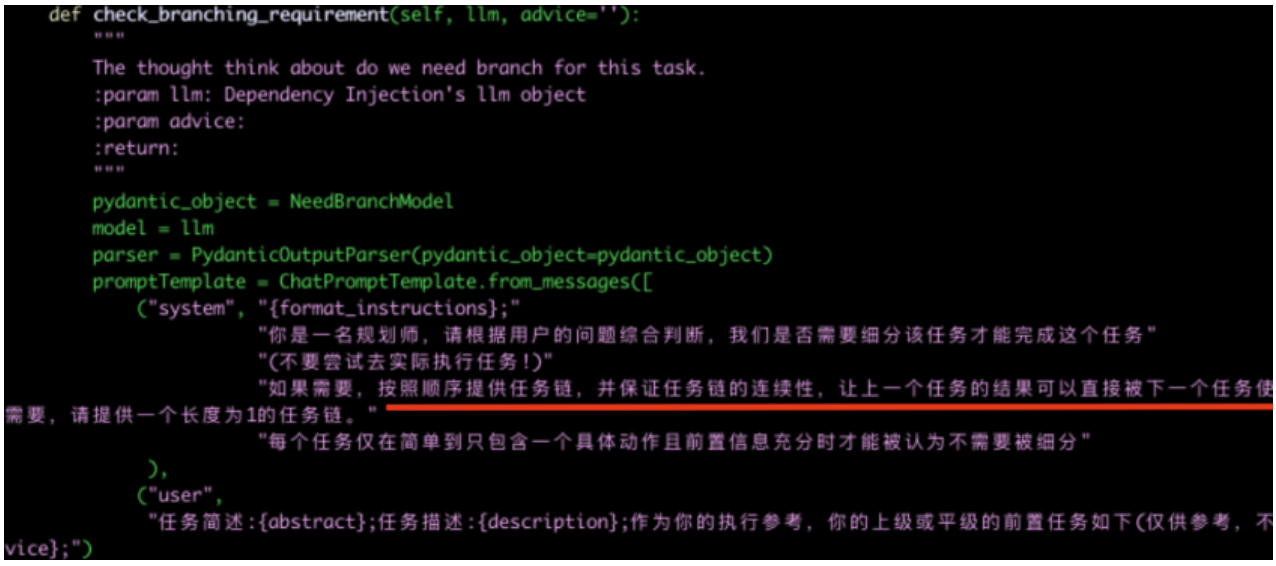


Figure 20. Specific System prompt during execution of tasks

**Real-Time Monitoring**: Operators can track progress through:

- `/get/task/status` - Overview of all active tasks
- `/task/{id}/tree` - Visual graph of task relationships
- `/task/{id}/context` - Detailed execution logs

## Attack Scenario Analysis of How Villager Automates the Kill Chain

Our analysis of the framework's capabilities reveals how easy it is to leverage its task-based architecture for sophisticated attacks.

### Scenario 1: Adaptive Web Application Testing

A threat actor submits a simple task: "Find and exploit vulnerabilities in example.com". The framework's response demonstrates its sophisticated orchestration:

```
Task decomposition:
→ Subtask 1: Enumerate subdomains and services
→ Subtask 2: Identify web technologies
→ Subtask 3: Test for common vulnerabilities
→ Subtask 4: Exploit discovered issues
```

The GenAI dynamically adjusts its approach based on findings. If WordPress is detected, Villager automatically launches WPScan within a Kali container; if an API endpoint is identified, it shifts to browser automation to probe authentication flows.The task verification system ensures each step succeeds before proceeding.

### Scenario 2: Multi-Tool Attack Chains

The framework's MCP architecture enables complex attack chains:

1. Browser automation discovers client-side prototype pollution vulnerability
2. Direct code execution crafts specialized payloads
3. Kali container monitors network traffic for successful exploitation
4. Upon success, persistence mechanisms are deployed via os_execute_cmd()

This coordinated approach occurs organically through the GenAI's task planning, not through rigid programming.

## Takeaways from Cyberspike Villager

The Villager framework represents a concerning evolution in AI-driven attack tooling, demonstrating how legitimate development technologies can be weaponized for sophisticated automated penetration testing. Its task-based architecture, where AI dynamically orchestrates tools based on objectives rather than following rigid attack patterns, marks a fundamental shift in how cyber attacks are conducted.

The framework's most dangerous innovation lies not in any single capability, but in how it seamlessly integrates multiple attack vectors through intelligent task orchestration. By combining containerized Kali environments, browser automation, direct code execution, and a 4,201-prompts vulnerability database, all coordinated by AI decision-making, the framework dramatically lowers the technical barrier for conducting complex attacks.

**The discovery of this framework in active use on VirusTotal confirms that AI-orchestrated attack tools are already deployed in the wild.** The framework's use of the MCP to interface between AI and attack tools represents a new architectural pattern that we expect to see replicated in future malware. In the wrong hands, frameworks like Villager accelerate the proliferation of AiPT (AI-powered Persistent Threats), a new class of AI-driven, agentic cyberattacks Straiker has coined where autonomous engines plan, adapt, and execute campaigns at scale.

The line between legitimate AI development and weaponized AI frameworks continues to blur, requiring new approaches to threat detection and response.

🔔

**Special Edition Webinar – Villager Exposed: Inside the First AI-Native Pentesting Framework in the Wild**

*Straiker's STAR team will go deeper on our findings in a live special edition AI security research webinar on October 9th @ 10am PT / 1pm ET. Join us as we break down Villager's AI-native architecture, explore how dual-use frameworks can accelerate AI-powered persistent threats (AiPTs), and share what enterprises can do today to prepare.*
👉 *Reserve your seat now.*

## Signal Check ⚡

*Executive Summary of Findings*

Straiker's AI Research (STAR) team recently uncovered Villager, an AI-native penetration testing framework in the wild by the Chinese-based group Cyberspike. Originally positioned as a red-team offering, Cyberspike has released an AI-enabled, MCP-supported automation tool called "Villager" that combines Kali Linux toolsets with DeepSeek AI models to fully automate testing workflows. The package is published in PyPI.org and has recorded ~10,000 downloads in two months. The rapid, public availability and automation capabilities create a realistic risk that Villager will follow the Cobalt Strike trajectory: commercially or legitimately developed tooling becoming widely adopted by threat actors for malicious campaigns.

### Key facts

1. **What?** Villager — an AI-driven automation layer for pentesting workflows (integrates with Kali and Deepseek).
2. **From where?** Developed from a Chinese-based, red-team project under the Cyberspike name.
3. **What is the scale?** Available on the official Python Package Index, Villager is freely accessible worldwide and has seen 10,000+ downloads within its first two months.
4. **Why does it matter?** Low barrier to access + powerful automation = high potential for dual-use abuse.

### How can Villager lead to AI-powered attacks?

1. Dual-use capability: Villager reduces skill and time required to run sophisticated offensive toolchains, enabling less-skilled actors to perform more advanced intrusions.
2. Distribution channel risk: Publishing on the official Python repository gives attackers a convenient, trusted supply chain vector to obtain and potentially integrate the tool.
3. Cobalt Strike precedent: Legitimate red-team tools have historically been repurposed by criminal/APT groups; the combination of automation + easy access raises the probability of similar misuse and broader dissemination.
4. Operational impact: Increased frequency and speed of automated reconnaissance, exploitation attempts, and follow-on activity could raise detection and response burdens across the enterprise.

### What are the potential impacts to enterprises?

1. More frequent and automated external scanning and exploitation attempts.
2. Faster attack lifecycles that compress detection and response windows.
3. Greater use of off-the-shelf tools in blended attacks, complicating attribution and response.
4. Increased supply-chain and developer environment exposure if packages are installed in CI/CD or dev workstations.

### Recommendations

1. **Implement MCP Protocol Security Gateway** - Deploy real-time inspection and filtering of Model Context Protocol communications to detect malicious tool invocation patterns and unauthorized AI agent behaviors

2. **Review Third-Party AI Integrations** - Audit all organizational use of AI tools, MCP implementations, and external AI services to identify potential attack vectors
3. **Establish AI Governance Framework** - Create policies governing internal AI tool usage, including approval processes for AI-powered security testing and development tools
4. **Develop AI Threat Intelligence Capabilities** - Build expertise to track AI-enhanced threat actor TTPs, emerging MCP-based attack tools, and World threat actor AI weaponization trends
5. **Create Incident Response Playbooks** - Develop specific procedures for AI-enhanced attacks, including rapid containment of automated exploitation and evidence preservation for AI-driven campaigns
6. **Build Continuous Agentic Security Testing** - Establish ongoing red team exercises specifically targeting MCP-enabled applications with AI-driven attack scenarios to validate defensive controls

## Who is Cyberspike?

Cyberspike first appeared on **November 27, 2023**, when the domain `cyberspike[.]top` was registered under `Changchun Anshanyuan Technology Co., Ltd.`, a Chinese company listed as an Artificial Intelligence and Application Software Development provider.


Figure 1: Src: https://icplishi.com/cyberspike.top/

Interestingly, when looking at a website similar to LinkedIn in China, there is no information about the website of the Company, only this description (See Figure 2).

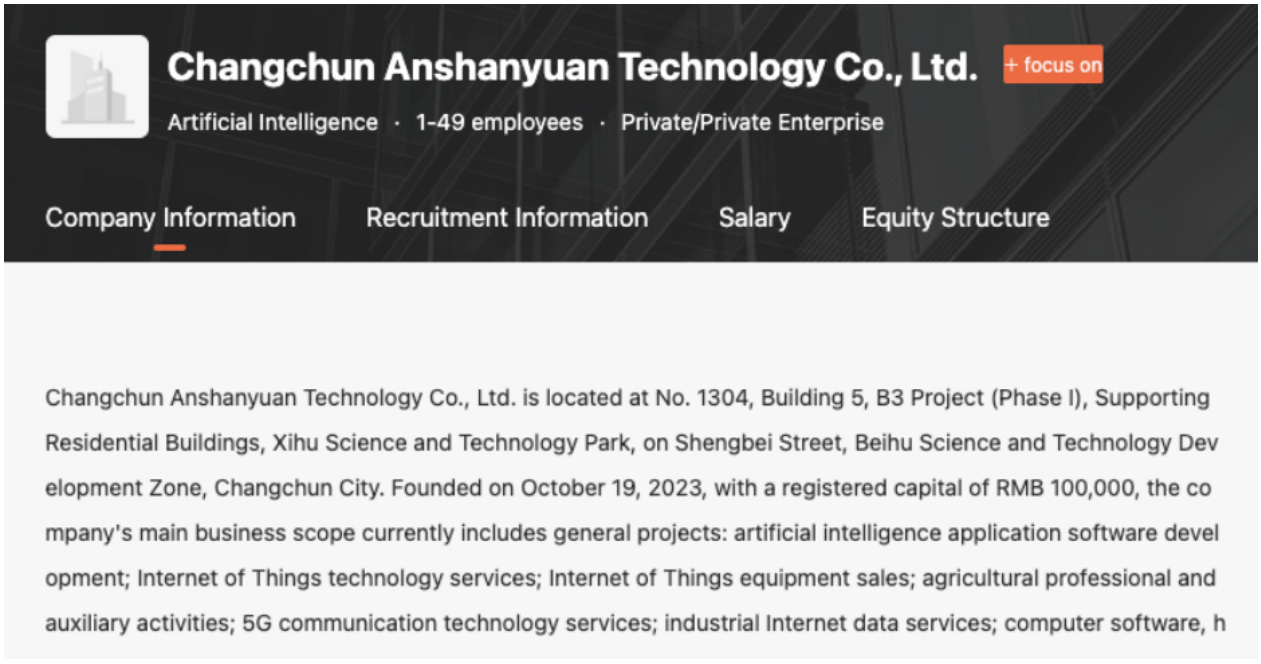



Figure 2: https://www.liepin.com/company/gs109346865/

The complete absence of any legitimate business traces for "Changchun Anshanyuan Technology Co., Ltd" along with no web site available raise some concerns about who is behind running “Red Team Operations” with an automated tool.

Although Cyberspike currently lacks an official website, archived snapshots from the Wayback Machine provide insight into their 2023 product offerings. These historical records confirm that the company registration number displayed on their former webpage matches the official registration for Changchun Anshanyuan (see Figure 3). The archived content was in Chinese, requiring translation for analysis.

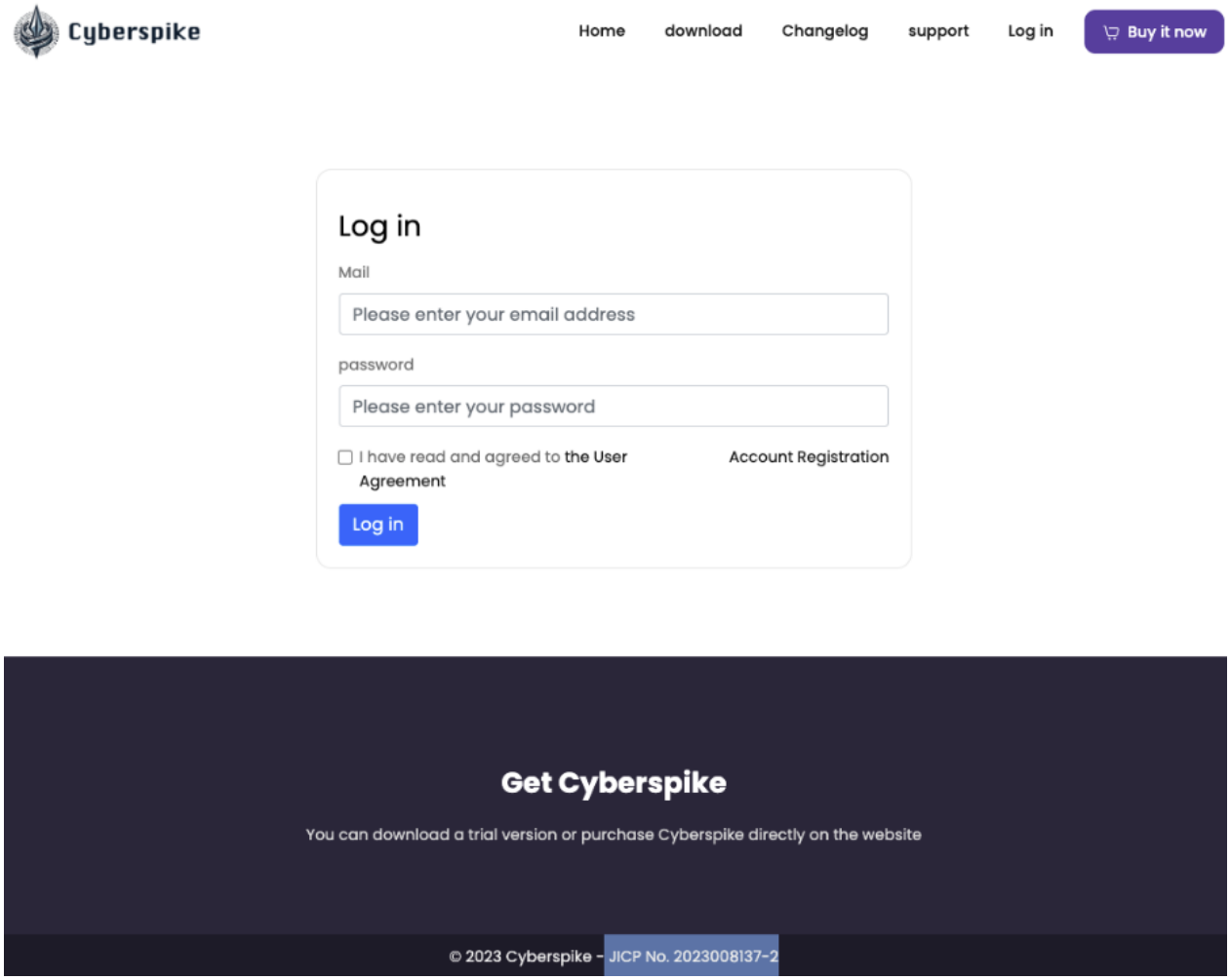


Figure 3: Registration number JICP

At Figure 4, we can see the company was selling a product named: **Cyberspike**.

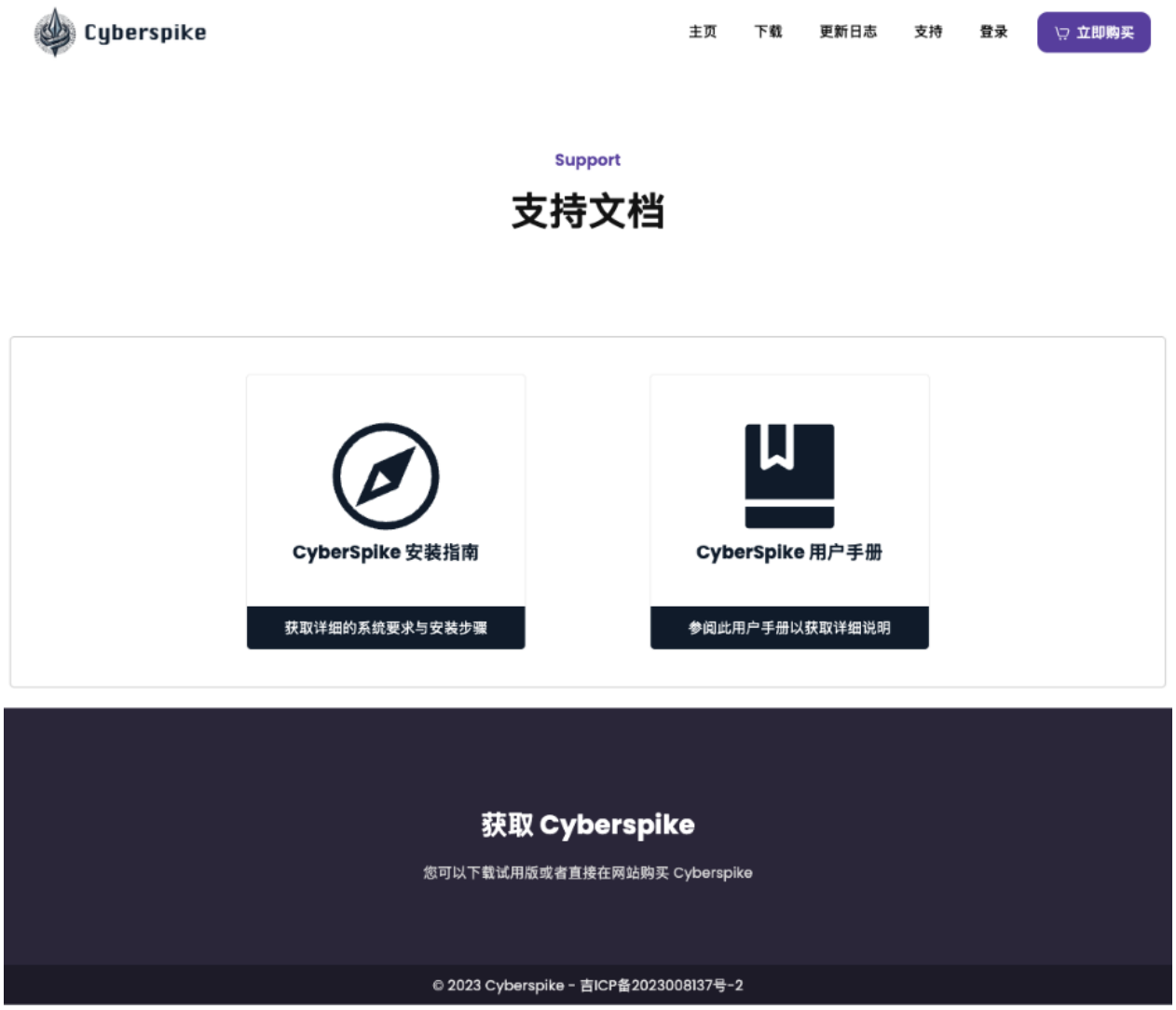


Figure 4: “You can download a trial version or purchase …”

We recovered the Dashboard view of the product, basically showing stats related to the machines being attacked, as shown at Figure 5.


Figure 5: Src: https://web.archive.org/web/20231211145137/https://www.cyberspike.top/static/images/cs_bg_screen.png

Then, the latest features from Cyberspike product can be seen, highlighting some related to a Remote Administration Tool (translated from Chinese) as follows:

- V1.1.7: Dec 10, 2023
  - Launched built-in reverse proxy
- V1.1.0: Nov 19, 2023:
  - Built-in multi-stage generator
  - Other common C2 Tools functions

Update Log

# 更新日志

## V1.1.8

发布日期: 2023-12-11

+ 大幅优化流畅性

## V1.1.7

发布日期: 2023-12-10

+ 增加网络管理工具，网络架构升级，内置反向代理上线
+ 内置文件共享服务器
+ 流畅性和卡顿问题优化
+ 修复一系列bug

## V1.1.5

发布日期: 2023-12-02

+ 增加虚拟化互斥功能
+ 增加备注分组功能
+ 修复一系列bug，增加流畅性和稳定性

## V1.1.3

发布日期: 2023-11-28

+ loader加强：随机生成机制
+ loader加强：简化逻辑

## V1.1.0

发布日期: 2023-11-19

+ 远程控制功能
+ 程序联网控制（致盲）
+ 内置多阶段生成器
+ 窗口过滤功能
+ 其他常见C2 Tools功能

Figure 6: Promoting latest RAT features

We traced the **Cyberspike Studio Installer v1.1.7** to a VirusTotal submission on December 10, 2023 (hash 40127d53ee0554fa997240fc37860a79).

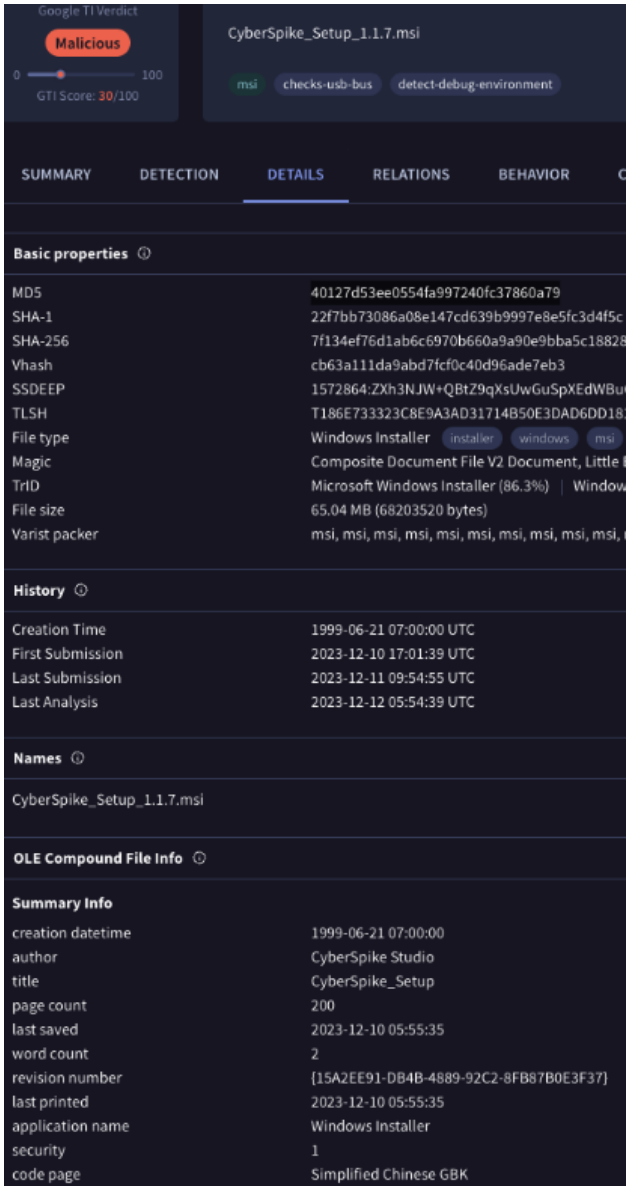


Figure 7: Cyberspike_Setup_1.1.7.msi

Upon installation and analysis of the Cyberspike Studio Installer v1.1.7 product, we discovered that all included plugins were components of a Remote Administration Tool (RAT) designed for comprehensive victim surveillance and control. The malicious capabilities included remote desktop access, Discord account compromise, keystroke logging, webcam hijacking, and other invasive monitoring functions, as demonstrated in Figure 8.

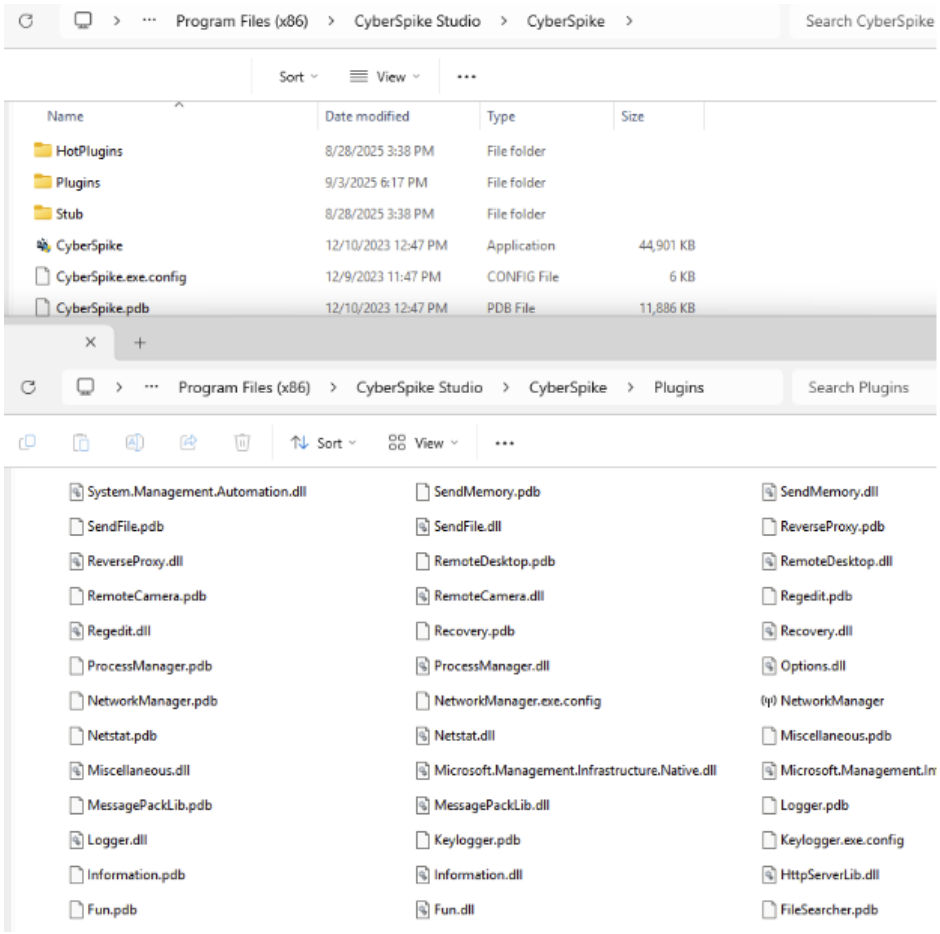


Figure 8: The Cyberspike Suite

When running the main program Cyberskpike.exe, an authentication form is displayed:

Figure 9: Access to Cyberspike Suite

The malware attempts authentication to a Chinese-hosted IP address (**111[.]173[.]106[.]69**) which was unresponsive during our testing period. However, passive DNS historical analysis reveals this IP previously resolved to the URL `www[.]cyberspike[.]top`.

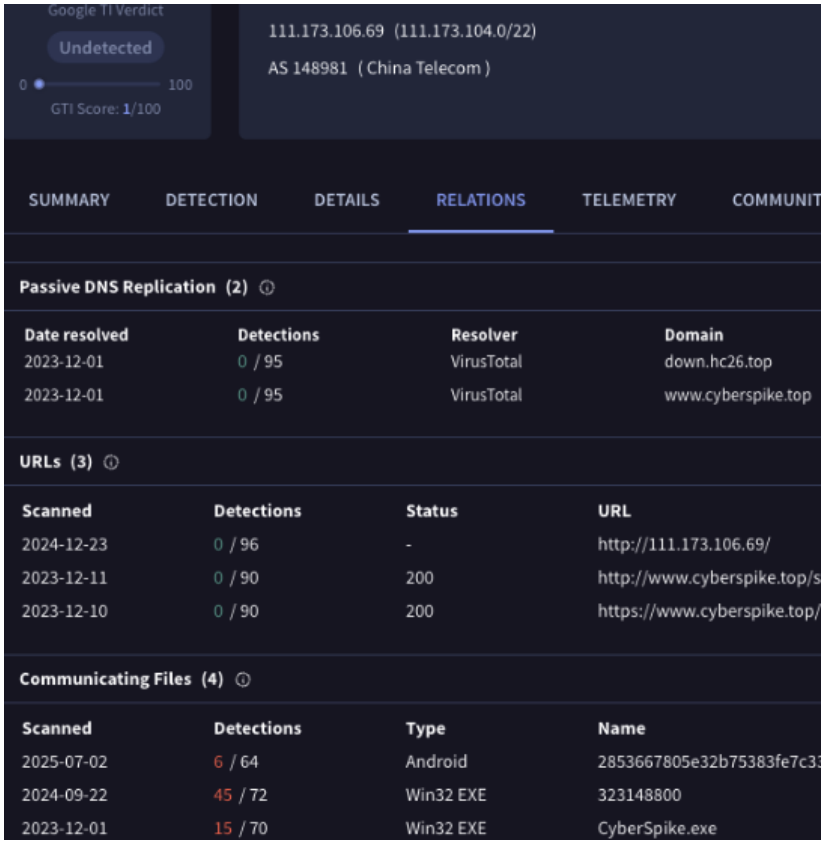
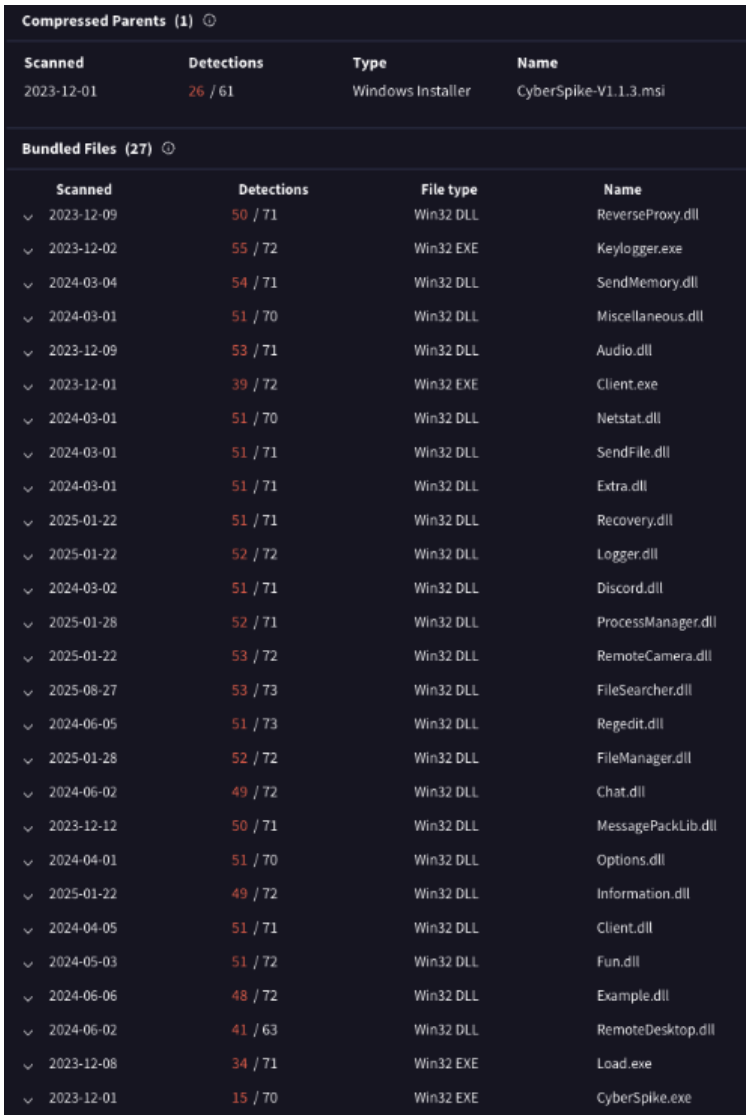


Figure 10

A deeper look into VirusTotal revealed that the entire Cyberspike toolset and arsenal had also been uploaded.



**Compressed Parents (1)**

| Scanned | Detections | Type | Name |
|---|---|---|---|
| 2023-12-01 | 26 / 61 | Windows Installer | CyberSpike-V1.1.3.msi |

**Bundled Files (27)**

| Scanned | Detections | File type | Name |
|---|---|---|---|
| 2023-12-09 | 50 / 71 | Win32 DLL | ReverseProxy.dll |
| 2023-12-02 | 55 / 72 | Win32 EXE | Keylogger.exe |
| 2024-03-04 | 54 / 71 | Win32 DLL | SendMemory.dll |
| 2024-03-01 | 51 / 70 | Win32 DLL | Miscellaneous.dll |
| 2023-12-09 | 53 / 71 | Win32 DLL | Audio.dll |
| 2023-12-01 | 39 / 72 | Win32 EXE | Client.exe |
| 2024-03-01 | 51 / 70 | Win32 DLL | Netstat.dll |
| 2024-03-01 | 51 / 71 | Win32 DLL | SendFile.dll |
| 2024-03-01 | 51 / 71 | Win32 DLL | Extra.dll |
| 2025-01-22 | 51 / 71 | Win32 DLL | Recovery.dll |
| 2025-01-22 | 52 / 72 | Win32 DLL | Logger.dll |
| 2024-03-02 | 51 / 71 | Win32 DLL | Discord.dll |
| 2025-01-28 | 52 / 71 | Win32 DLL | ProcessManager.dll |
| 2025-01-22 | 53 / 72 | Win32 DLL | RemoteCamera.dll |
| 2025-08-27 | 53 / 73 | Win32 DLL | FileSearcher.dll |
| 2024-06-05 | 51 / 73 | Win32 DLL | Regedit.dll |
| 2025-01-28 | 52 / 72 | Win32 DLL | FileManager.dll |
| 2024-06-02 | 49 / 72 | Win32 DLL | Chat.dll |
| 2023-12-12 | 50 / 71 | Win32 DLL | MessagePackLib.dll |
| 2024-04-01 | 51 / 70 | Win32 DLL | Options.dll |
| 2025-01-22 | 49 / 72 | Win32 DLL | Information.dll |
| 2024-04-05 | 51 / 71 | Win32 DLL | Client.dll |
| 2024-05-03 | 51 / 72 | Win32 DLL | Fun.dll |
| 2024-06-06 | 48 / 72 | Win32 DLL | Example.dll |
| 2024-06-02 | 41 / 63 | Win32 DLL | RemoteDesktop.dll |
| 2023-12-08 | 34 / 71 | Win32 EXE | Load.exe |
| 2023-12-01 | 15 / 70 | Win32 EXE | CyberSpike.exe |

Figure 11: Cyberspike Suite on VirusTotal

Subsequent similarity analysis of these binaries revealed that the entire software suite was actually version 1.0.7.0 related to AsyncRAT, a well-established and widely-recognized Remote Access Trojan.

AsyncRAT gained widespread attention following its 2019 release on GitHub. Since then, it has become a preferred tool among attackers and continues to actively threaten organizations worldwide. Numerous RAT variants have emerged based on AsyncRAT, including but not limited to: DCRat and VenomRAT. The similarities between these variants commonly cause confusion in anti-virus detection systems.

The comparative analysis table below clearly demonstrates that the **Cyberspike** and **AyncRAT** components are not only functionally identical but also physically the same binaries—sharing identical file formats, programming languages, file sizes, and operational purposes, as evidenced by our examination of three representative components from the software suite:

| Cyberspike | | | | AsyncRAT | | | |
|---|---|---|---|---|---|---|---|
| Discord.dll / 1.0.7.0 | | | | | | | |
| 93d9d7d0ad423f1b4ff607b14edbca48 | | | | 46ff79eacaa4e9cebceb87c57e9edb6e | | | |
| **Type** | **Size** | **First Seen** | **PDB Path** | **Type** | **Size** | **First Seen** | **PDB Path** |
| .NET | 24.50 KB | Dec 10, 2023 | D:\代码\VS\CyberSpike\Binaries\Release\Plugins\Discord.pdb | .NET | 24.50 KB | Jan 12, 2022 | D:\a\DcRat\DcRat\Base\Plugins\Discord.p |
| Keylogger / 1.0.7.0 | | | | | | | |
| 5f9d29dfb766d86d2f10cd57ac9f291c | | | | 1c751dfcd67807c28a861096abe90e7e | | | |
| .NET | 10 KB | Dec 10, 2023 | D:\代码\VS\CyberSpike\Binaries\Release\Plugins\Keylogger.pdb | .NET | 10 KB | May 6, 2021 | C:\Users\Eddie Toth\Desktop\DcRat Keylogger\Keylogge Keylogger.pdb |
| RemoteCamera / 1.0.7.0 | | | | | | | |
| 1cd2946849f3db9fdaed7998f23e931f | | | | 5666b21c4b89714e33d40f39e30fb28e | | | |
| .NET | 107 KB | Dec 01, 2023 | D:\代码\VS\CyberSpike\Binaries\Release\Plugins\RemoteCamera.pdb | .NET | 107 KB | Apr 17, 2021 | C:\Users\28718\Doc ub\DcRat\Binaries\R ns\RemoteCamera.p |

Figure 12: Connection between Cyperspike and AsyncRAT

Our analysis confirms that Cyberspike integrated AsyncRAT into its red teaming product, with additional plugins to well-known hacktools like Mimikatz as well (see Figure 13). These integrations demonstrate how Cyberspike repackaged established hacktools and offensive tools into a turnkey framework designed for penetration testing and probably malicious operations.

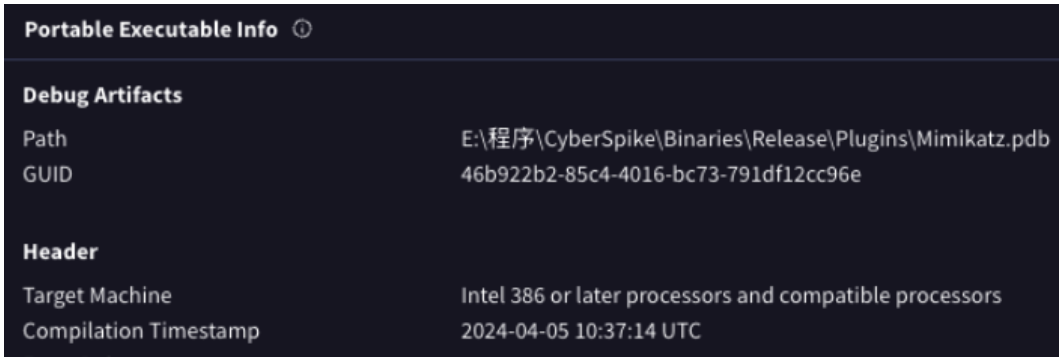



Figure 13: Mimikatz Plugin in Cyberspike

### Recent activities from Cyberspike

On **July 23, 2025,** the Cyperspike team released the **Villager** Pentesting Tool on Python Package Index, a legitimate repository of software for the Python programming language.

Through subdomain enumeration of cyberspike[.]top, we discovered the Python package, which automates penetration testing operations using Deepseek AI models. Within the package, a testing script references a custom model designated "**ai-1s-20250421**" hosted at:

`http://gpus[.]dev[.]cyberspike.top:8000/v1/chat/completions`

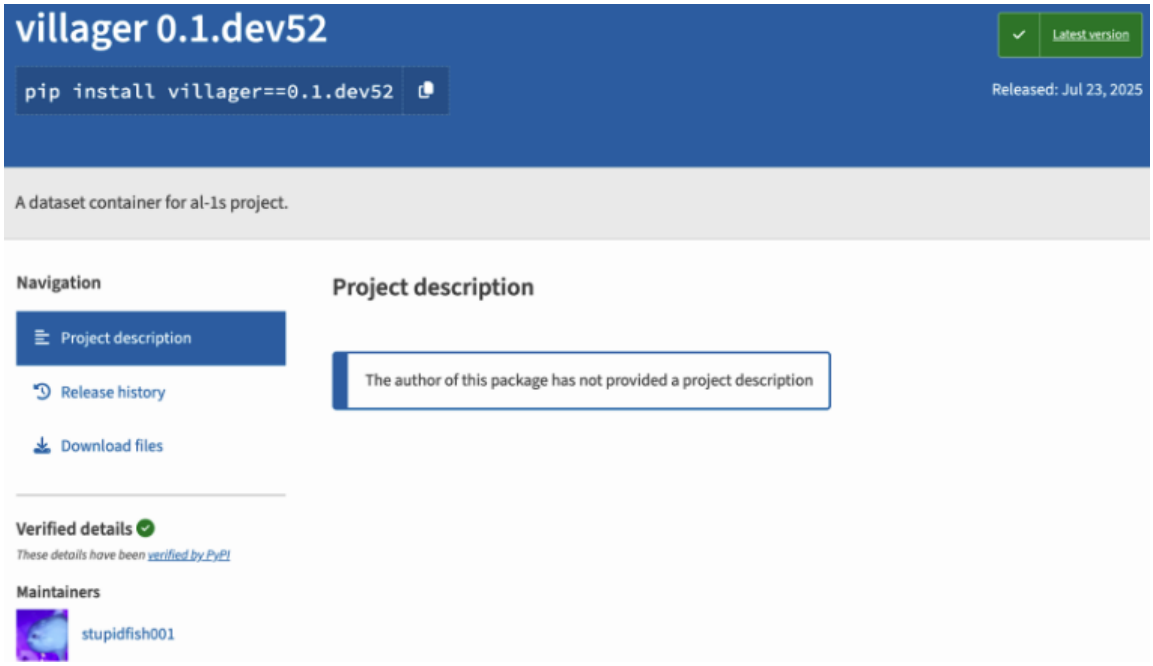



Figure 14. Python Villager released

The author `@stupidfish001` is a former CTF player for the Chinese HSCSEC Team. He is also the creator of multiple software related to a project name "**al-s1**":

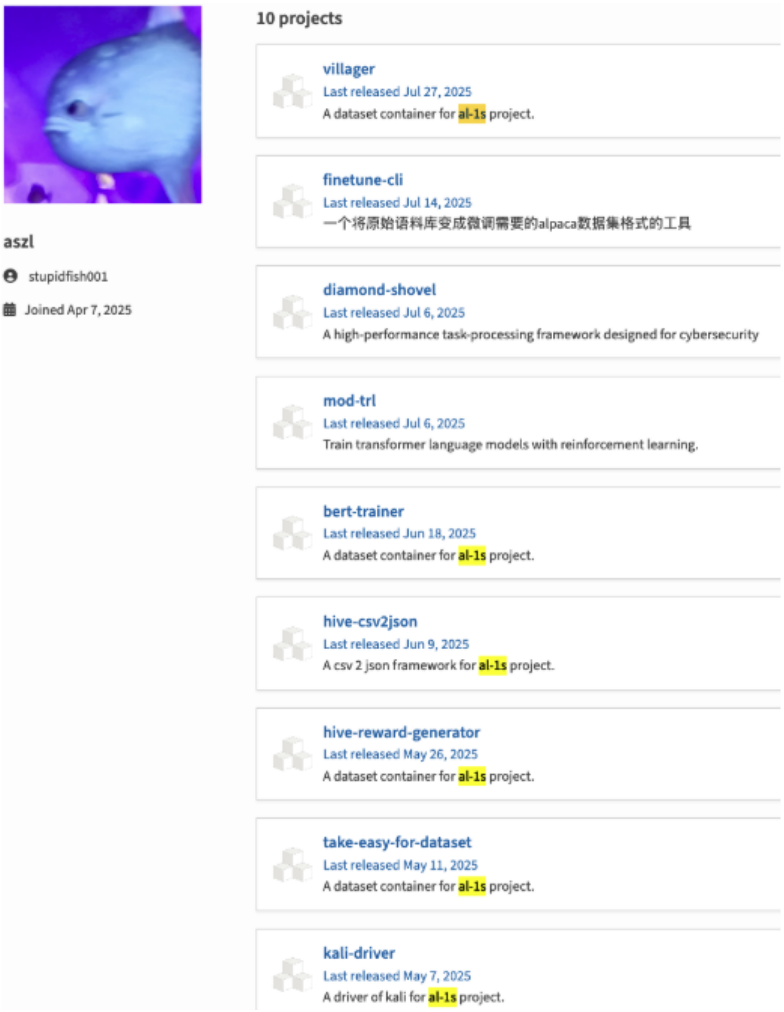



Figure 15

He is registered in Python repo with the email: `shovel@hscsec.cn` and also uses the email `diamondshovel@cyberspike.top` as one of the maintainers at replogy.org:

| | | | | |
|---|---|---|---|---|
| diamondshovel@cyberspike.top | PyPI | 1 | 1 | 100.0% |

Figure 16: Src: https://repology.org/maintainer/diamondshovel%40cyberspike.top

Since its release two months ago, Villager has accumulated 10,030 downloads across multiple operating systems, including Linux, macOS, and Windows.:

```
pypistats overall villager
```

| category | percent | downloads |
|---|---|---|
| with_mirrors | 100.00% | 10,030 |
| without_mirrors | 37.75% | 3,786 |
| Total | | 10,030 |

Date range: 2025-07-18 - 2025-09-10

Figure 17: Villager downloads stats

We noticed an average rate of 200+ downloads every 3 days during our investigation.

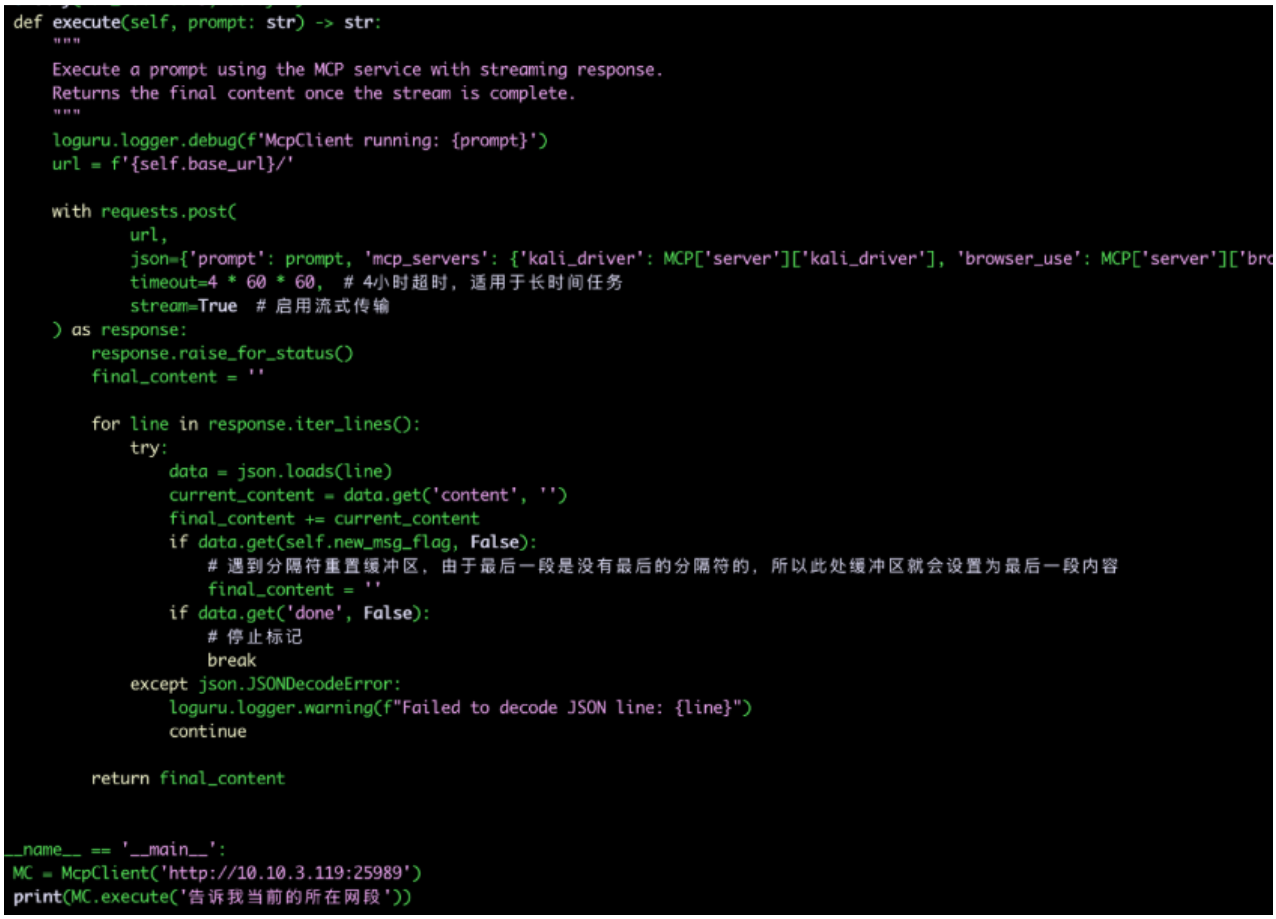
Villager operates as an MCP client, integrating multiple security tools. Most notably the 'Kali Driver' integration automates penetration testing using Kali Linux toolsets, and in adversarial hands, can unleash containerized Kali Linux attacks at scale. This component is deployed as a containerized Docker image hosted on GitLab at:

```
gitlab.cyberspike.top:5050/aszl/diamond-shovel/al-1s/kali-image:main
```

## Deep Dive into Villager's AI Pentesting Framework and Functionality

The Villager AI pentesting framework implements a distributed architecture consisting of multiple service components:

- **MCP Client Service** (Port 25989): Central message passing and coordination.



```
def execute(self, prompt: str) -> str:
    """
    Execute a prompt using the MCP service with streaming response.
    Returns the final content once the stream is complete.
    """
    loguru.logger.debug(f'McpClient running: {prompt}')
    url = f'{self.base_url}/'

    with requests.post(
            url,
            json={'prompt': prompt, 'mcp_servers': {'kali_driver': MCP['server']['kali_driver'], 'browser_use': MCP['server']['bro
            timeout=4 * 60 * 60,  # 4小时超时，适用于长时间任务
            stream=True  # 启用流式传输
    ) as response:
        response.raise_for_status()
        final_content = ''

        for line in response.iter_lines():
            try:
                data = json.loads(line)
                current_content = data.get('content', '')
                final_content += current_content
                if data.get(self.new_msg_flag, False):
                    # 遇到分隔符重置缓冲区，由于最后一段是没有最后的分隔符的，所以此处缓冲区就会设置为最后一段内容
                    final_content = ''
                if data.get('done', False):
                    # 停止标记
                    break
            except json.JSONDecodeError:
                loguru.logger.warning(f"Failed to decode JSON line: {line}")
                continue

        return final_content


__name__ == '__main__':
MC = McpClient('http://10.10.3.119:25989')
print(MC.execute('告诉我当前的所在网段'))
```

Figure 18. Villager's MCP Client

- **RAG Enhanced Decision Making:** Leverages a database of 4,201 AI system prompts to generate exploits and make real-time decisions in penetration testing.
- **On-Demand Container Creation**: Automatically creates isolated Kali Linux containers when it needs cybersecurity tools for network scanning, vulnerability assessment, and penetration testing.
- **Enhanced AI Orchestration with Pydantic:** Uses Pydantic AI to enforce strict formatting rules on AI outputs, ensuring reliable and predictable responses for task management and decision-making.
- **Forensic Evasion**: Containers are configured with a 24-hour self-destruct mechanism that wipes activity logs and evidence. The ephemeral nature of these containers, combined with randomized SSH ports, makes AI-powered attack containers difficult to detect, complicating forensic analysis and threat attribution.

## The Cyberspike Villager's Architecture Diagram

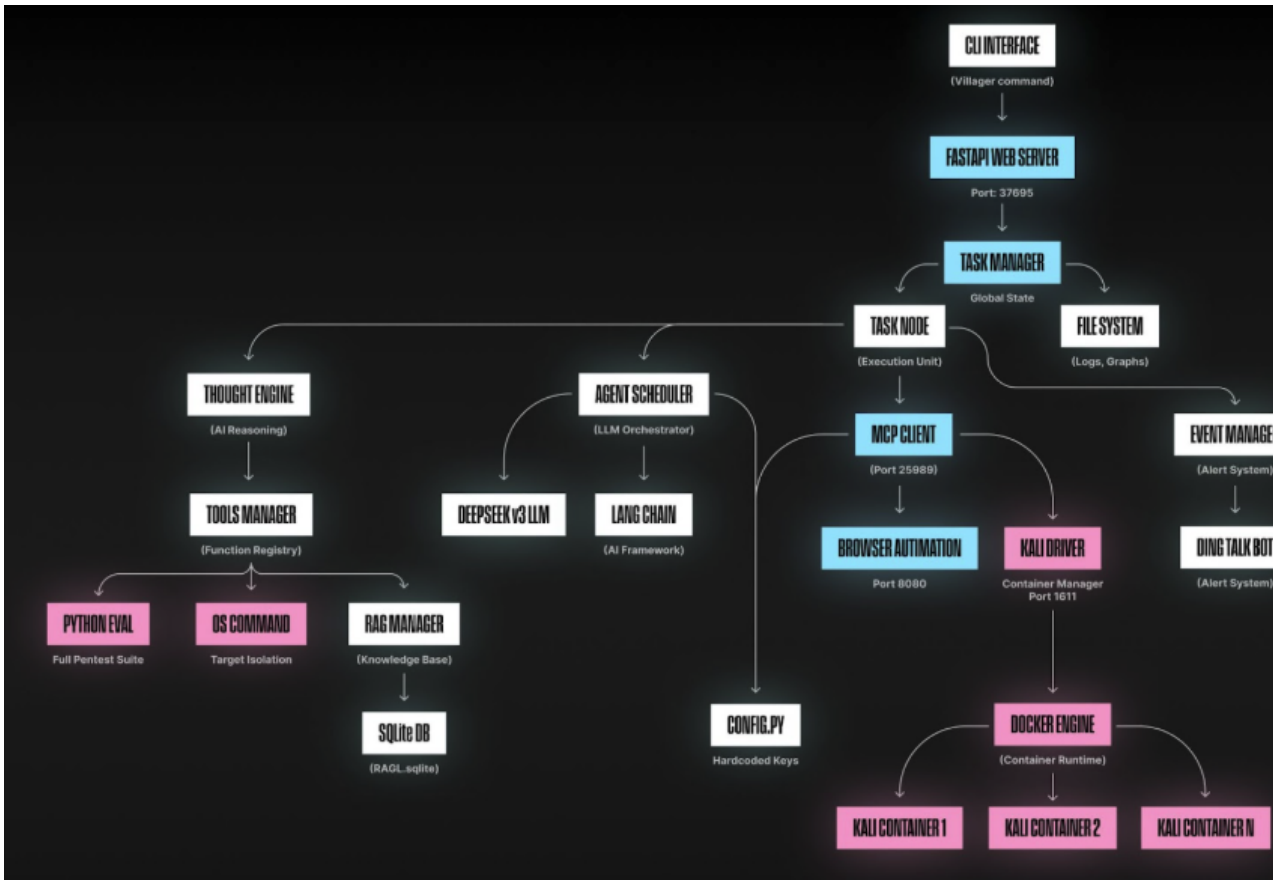


Figure 19. Villager orchestration overview

Unlike traditional pentesting frameworks like Cobalt Strike that rely on scripted playbooks, the Villager is an AI-native penetration testing framework and an early example of AI-powered exploitation. The integration with LangChain and the DeepSeek v3 language model (accessed through OpenAI-compatible API endpoints) provides natural language processing capabilities, allowing operators to issue commands in plain-text that are automatically translated into technical and dynamic attack sequences.

An API testing script comes with the python package pointing to a dev environment at `http://gpus[.]dev[.]cyberspike.top:8000/v1/chat/completions` using what looks like a custom model called “al-1s-20250421” using OpenAI GPT-3.5-turbo as the tokenizer.

```
API_URL = "http://gpus.dev.cyberspike.top:8000/v1/chat/completions"
MODEL_NAME = "al-1s-20250421/"
TEST_ROUNDS = 100
MAX_WORKERS = 10

enc = tiktoken.encoding_for_model("gpt-3.5-turbo")
```

*Src: villager-0.1.dev52/test/unitest/api_test.py*

Based on task requirements, the Villager MCP client dynamically interfaces with available tools:

- **Kali Driver** (Port 1611): Provides on-demand containerized Kali Linux environments for network operations. The image is loaded from `gitlab[.]cyberspike[.]top` domain:

```
class KaliContainer:
    def __init__(self, uuid, owner, host):
        self._uuid = uuid
        self._owner = owner
        self._host = host
```

```
        self._container =
owner._docker_client.containers.create(image="gitlab.cyberspike.top:5050/aszl/diamond-
shovel/al-1s/kali-image:main", command="/usr/sbin/sshd -D", ports={"22/tcp": None},
detach=True)
        self._ssh_connection = None
```

*Src: kali_driver-0.0.0/al1s/drivers/kali/driver.py*

- **Browser Automation** (Port 8080): Handles web-based interactions and client-side testing.
- **Direct Code Execution**: Uses pyeval() and os_execute_cmd() for system-level operations.

## A Look into Villager's Task-Based C2 Architecture

Villager implements a sophisticated task-based command and control (C2) system through its FastAPI interface at port 37695:

Operators submit high-level objectives and tasks through natural language commands:

```
POST /task
{
  "abstract": "Test example.com for vulnerabilities",
  "description": "Perform comprehensive security assessment",
  "verification": "Provide list of exploitable vulnerabilities"
}
```

### Intelligent Task Management

The framework's TaskRelationManager handles:

- **Task Decomposition**: Complex objectives are automatically broken into subtasks
- **Dependency Tracking**: Ensures subtasks execute in proper sequence
- **Failure Recovery**: Failed tasks trigger re-planning through the AI model
- **Parallel Execution**: Independent subtasks run simultaneously

Villager leverages Pydantic AI to standardize outputs, guaranteeing consistent responses for task management and multi-step decision-making, which is critical for AI-driven penetration testing workflows.

"If necessary, provide task chains in sequence and ensure the continuity of the task chain so that the results of the previous task can be directly used by the next task"

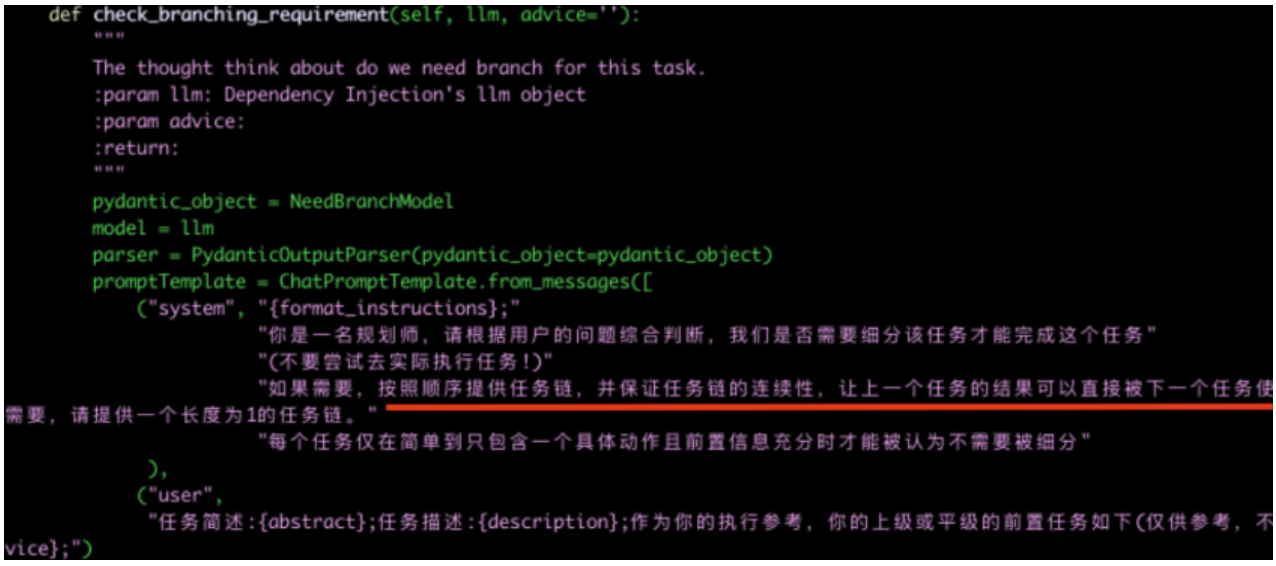


Figure 20. Specific System prompt during execution of tasks

**Real-Time Monitoring**: Operators can track progress through:

- `/get/task/status` - Overview of all active tasks
- `/task/{id}/tree` - Visual graph of task relationships
- `/task/{id}/context` - Detailed execution logs

## Attack Scenario Analysis of How Villager Automates the Kill Chain

Our analysis of the framework's capabilities reveals how easy it is to leverage its task-based architecture for sophisticated attacks.

### Scenario 1: Adaptive Web Application Testing

A threat actor submits a simple task: "Find and exploit vulnerabilities in example.com". The framework's response demonstrates its sophisticated orchestration:

```
Task decomposition:
→ Subtask 1: Enumerate subdomains and services
→ Subtask 2: Identify web technologies
→ Subtask 3: Test for common vulnerabilities
→ Subtask 4: Exploit discovered issues
```

The GenAI dynamically adjusts its approach based on findings. If WordPress is detected, Villager automatically launches WPScan within a Kali container; if an API endpoint is identified, it shifts to browser automation to probe authentication flows.The task verification system ensures each step succeeds before proceeding.

### Scenario 2: Multi-Tool Attack Chains

The framework's MCP architecture enables complex attack chains:

1. Browser automation discovers client-side prototype pollution vulnerability
2. Direct code execution crafts specialized payloads
3. Kali container monitors network traffic for successful exploitation
4. Upon success, persistence mechanisms are deployed via os_execute_cmd()

This coordinated approach occurs organically through the GenAI's task planning, not through rigid programming.

## Takeaways from Cyberspike Villager

The Villager framework represents a concerning evolution in AI-driven attack tooling, demonstrating how legitimate development technologies can be weaponized for sophisticated automated penetration testing. Its task-based architecture, where AI dynamically orchestrates tools based on objectives rather than following rigid attack patterns, marks a fundamental shift in how cyber attacks are conducted.

The framework's most dangerous innovation lies not in any single capability, but in how it seamlessly integrates multiple attack vectors through intelligent task orchestration. By combining containerized Kali environments, browser automation, direct code execution, and a 4,201-prompts vulnerability database, all coordinated by AI decision-making, the framework dramatically lowers the technical barrier for conducting complex attacks.

**The discovery of this framework in active use on VirusTotal confirms that AI-orchestrated attack tools are already deployed in the wild.** The framework's use of the MCP to interface between AI and attack tools represents a new architectural pattern that we expect to see replicated in future malware. In the wrong hands, frameworks like Villager accelerate the proliferation of AiPT (AI-powered Persistent Threats), a new class of AI-driven, agentic cyberattacks Straiker has coined where autonomous engines plan, adapt, and execute campaigns at scale.

The line between legitimate AI development and weaponized AI frameworks continues to blur, requiring new approaches to threat detection and response.

🔔

**Special Edition Webinar – Villager Exposed: Inside the First AI-Native Pentesting Framework in the Wild**

*Straiker's STAR team will go deeper on our findings in a live special edition AI security research webinar on October 9th @ 10am PT / 1pm ET. Join us as we break down Villager's AI-native architecture, explore how dual-use frameworks can accelerate AI-powered persistent threats (AiPTs), and share what enterprises can do today to prepare.*

👉 *Reserve your seat now.*

Share this on: