Earth Ammit Disrupts Drone Supply Chains Through Coordinated Multi-Wave Attacks in Taiwan

By By: Pierre Lee, Vickie Su, Philip Chen May 13, 2025 Read time: 14 min (3653 words)

Published: 2025-05-13 · Archived: 2026-04-05 14:37:18 UTC

Cyber Threats

Trend™ Research discusses the evolving tradecraft of threat actor Earth Ammit, proven by the advanced toolset used in its TIDRONE and VENOM campaigns that targeted the drone supply chain.

Summary

- Earth Ammit, a threat actor linked to Chinese-speaking APT groups, launched two waves of campaigns from 2023 to 2024. The first wave, VENOM, mainly targeted software service providers, and the second wave, TIDRONE mainly targeted the military industry. In its VENOM campaign, Earth Ammit's approach involved penetrating the upstream segment of the drone supply chain.
- In the VENOM campaign, the threat actors primarily relied on open-source tools due to low cost and difficult tracking. They shifted to custom-built tools like CXCLNT and CLNTEND in the TIDRONE campaign for cyberespionage purposes.
- Victims of the TIDRONE and VENOM campaigns primarily originated from Taiwan and South Korea, affecting a range of industries including military, satellite, heavy industry, media, technology, software services, and healthcare sectors. Earth Ammit’s long-term goal is to compromise trusted networks via supply chain attacks, allowing them to target high-value entities downstream and amplify their reach. Organizations that fall prey to these attacks are also at risk of data theft, including exfiltration of credentials and screenshots.
- Organizations can mitigate supply chain and fiber-based attacks by managing third-party risks, enforcing code signing, monitoring software behavior and fiber-related API usage, applying patches, segmenting vendor systems, adopting Zero Trust Architecture, and strengthening EDR and behavioral monitoring.
- The malicious elements of Earth Ammit's dual campaigns are detected and blocked by Trend Vision One™. Customers can also access hunting queries, threat insights, and threat intelligence reports to gain rich context and the latest updates on Earth Ammit.

In July 2024, we disclosed the TIDRONE campaign, in which threat actors targeted Taiwan’s military and satellite industries. During our investigation, we discovered that multiple compromised entities were using the same enterprise resource planning (ERP) software. This led us to engage with the ERP vendor, through which we uncovered additional details that pointed to an earlier, related campaign – VENOM. Our findings were also presented at Black Hat Asia 2025 last month, where we discussed in depth Earth Ammit's tactics in the TIDRONE

and VENOM campaigns, their targeted attacks on military sectors in Eastern Asia, and their possible ties to Chinese-speaking cyber-espionage groups.

The VENOM campaign focused on a wide range of upstream vendors, spanning the heavy industry, media, technology, software services, and healthcare sectors. Figure 1 presents a consolidated timeline and visual overview from the attackers' perspective, illustrating both TIDRONE and VENOM campaigns conducted by the intrusion set Earth Ammit.

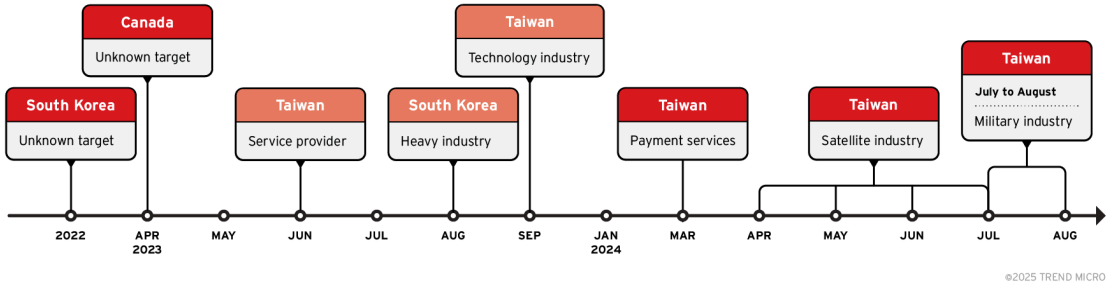

Figure 1. The timeline of operations conducted by Earth Ammit

- Orange blocks on the timeline correspond to the VENOM campaign, active from 2023 to 2024, which compromised service providers and technology companies in Taiwan, as well as heavy industry entities in South Korea. Earth Ammit’s strategy centered around infiltrating the upstream segment of the drone supply chain. By compromising trusted vendors, the group positioned itself to target downstream customers – demonstrating how supply chain attacks can ripple out and cause broad, global consequences.
- Red blocks in the timeline represent campaign TIDRONE, which targeted payment services, satellite industries, and military industries in Taiwan in 2024 through the upstream supply chain. As we observed the whole campaign, it could be traced back to 2022 for the earliest case that some unknown victim and community from South Korea and Canada submitted the samples to the VirusTotal.

Victimology

Figure 2. The victimology of Earth Ammit

Incorporating findings from the TIDRONE report published by AhnLab, the campaign's victimology was primarily concentrated in Taiwan and South Korea (Figure 2), affecting organizations across various sectors including heavy industry, media, technology, software services, healthcare, satellite and drone vendors, military-related suppliers, and payment service providers. In Taiwan, our telemetry indicated that several infected entities had close ties to the military and drone industry, leading to the initial assessment that the operation may have been specifically targeting the drone sector – an assumption that informed the direction of the subsequent investigation.

Supply chain attack

Supply chain attacks typically involve compromising trusted vendors or service providers to gain access to downstream targets. In our analysis of the VENOM and TIDRONE campaigns, we observed two distinct types of supply chain attack techniques, each with its own tactics and operational implications (Figure 3).

Path A: Classic supply chain attack

In a classic supply chain attack, threat actors inject malicious code into legitimate software or replace software update packages with tampered versions. These compromised executables are then delivered to downstream customers under the guise of legitimate software. This traditional approach relies on the attacker's ability to insert or replace code within the victim's supply chain pipeline.

Path B: General supply chain attack

However, when code injection or update replacement is not feasible, attackers may adopt an alternative strategy. By compromising upstream vendors, they can leverage trusted communication channels – such as remote monitoring or IT management tools – to distribute malware across connected environments. This method, which we refer to as a general supply chain attack, enables lateral movement from the upstream vendor to downstream targets without altering any software artifacts.

Both VENOM and TIDRONE campaigns employed a combination of these techniques. This underscores the evolving nature of supply chain threats and the importance of monitoring not only software integrity but also trusted network relationships and administrative access points within partner ecosystems.

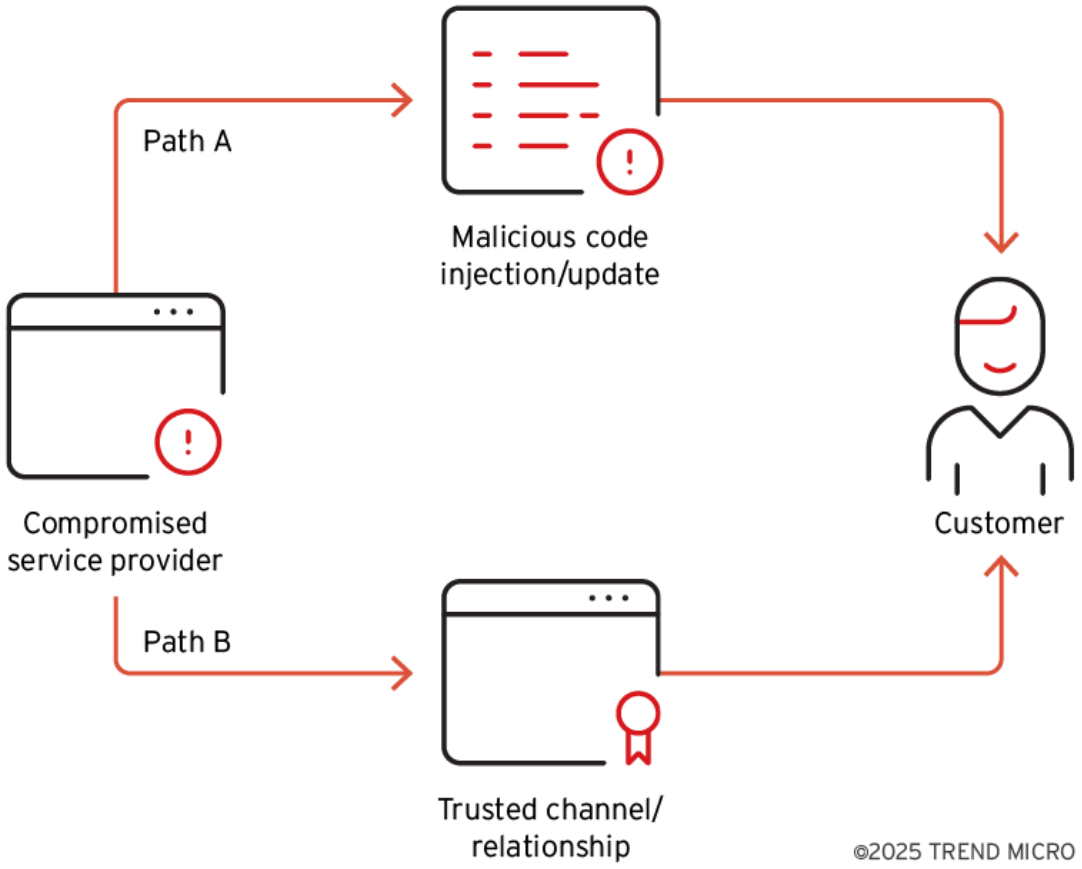

Figure 3. Two kinds of supply chain attacks were observed in Earth Ammit's activities

Campaign analysis - VENOM

Based on our telemetry, the attacker exploited web server vulnerabilities and uploaded web shells in the initial access phase. This method allowed the attackers to gain entry into the servers on the victim side. Following the successful breach, the attackers progressed to the command and control phase. They utilized open-sourced proxy tools and remote access tools (RAT) to maintain persistence within the system. As noted previously, the attackers prefer to implement open-sourced tools rather than their own malware, a characteristic that prevents attribution by concealing their activities (Figure 4).

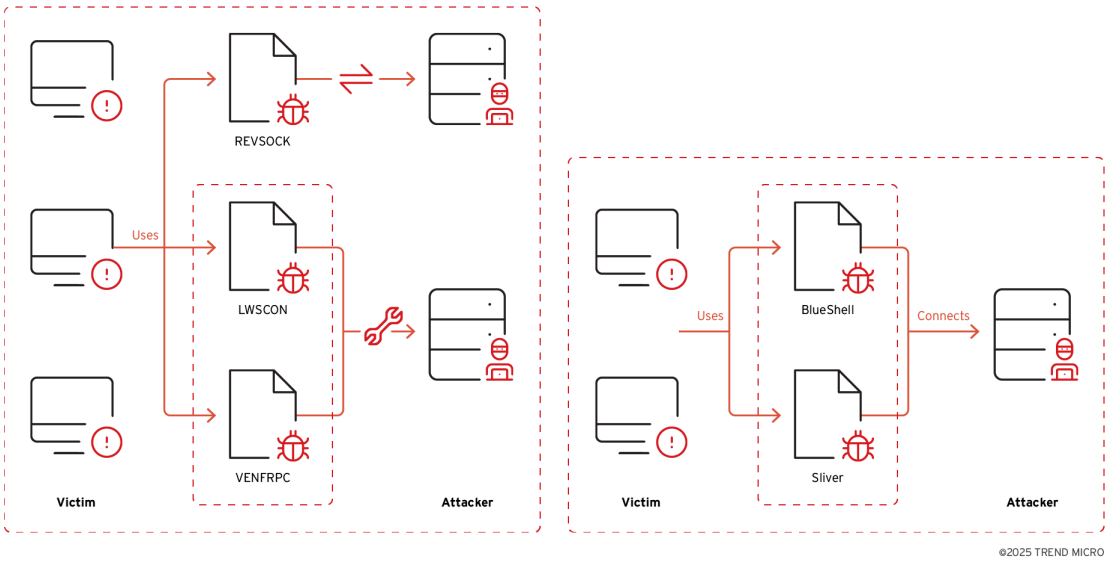

Figure 4. The threat actor utilized the open-source tools after access with the proxy tool (left) and backdoor (right)

Once they had established persistence on the victim's machine, their next objective was to steal credentials from the environment. In this stage, they targeted NTDS data from the victims. This data was leveraged to compromise the next stage, representing the downstream customers, which is linked to the campaign TIDRONE.

Campaign analysis - TIDRONE

The infection chain of the campaign TIDRONE is divided into three parts.

Initial access

Initially, the attackers targeted service providers, performing malicious code injection and distributing malware through trusted channels to downstream customers, much like in the campaign VENOM. This entire process serves as the initial access stage for the campaign TIDRONE (Figure 5).

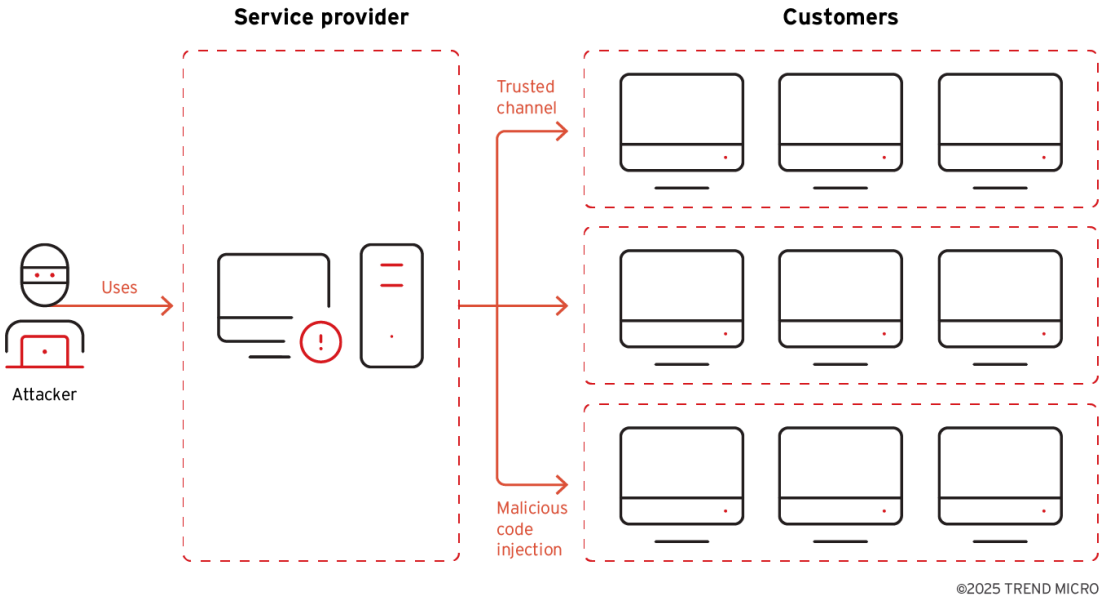

Figure 5. The campaign TIDRONE compromised the victim through a supply chain attack from the service provider or upstream vendor

Command and control

In the second stage, the threat actors spread the customized backdoor for cyberespionage. Our research supposed that the same loader can load two different kinds of payloads, which are backdoor CXCLNT and CLNTEND. Note that the flow chart in Figure 6 is just the rough version for illustration; the multiple layers of loading were discussed in the previous report on the TIDRONE campaign.

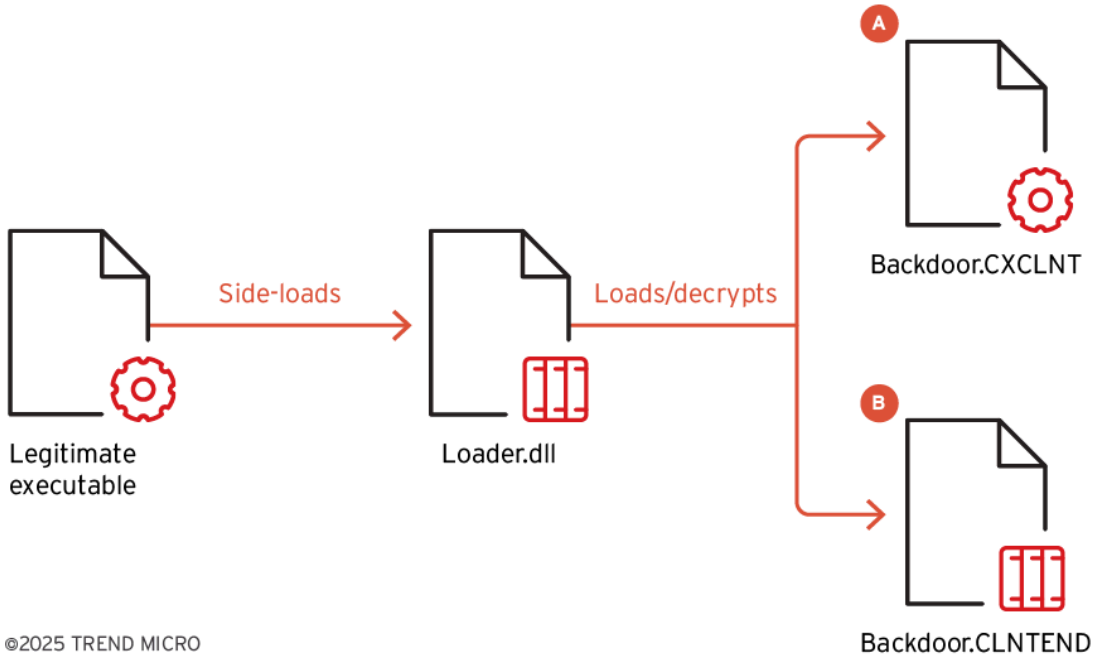

Figure 6. The rough infection chain in campaign TIDRONE

Post-exploitation

Table 1 shows the activities and related logs observed in the victim's environment. In the whole picture, threat actors mainly performed these behaviors.

Behavior	**Related log and description**
Privilege escalation	Perform UAC Bypass and restart the process with the Winlogon process token. • $ C:\Windows\SysWOW64\reg.exe: add HKCU\Software\Classes\ms-settings\Shell\Open\command /v DelegateExecute • $ C:\Windows\SysWOW64\reg.exe: add HKCU\Software\Classes\ms-settings\Shell\Open\command /t REG_SZ /d "C:\ProgramData\winword.exe" /f • $%APPDATA%\.temp\winsrv.exe
Persistence	• Run a scheduled task. • Replace the legitimate executable in a selected directory with an auto-run feature.
Credential dumping	The series of conventional commands to dump credentials via mimikatz. • $ C:\Windows\Temp\procdump.exe -accepteula -ma lsass.exe lsass.dmp • $ C:\Windows\SysWOW64\cmdkey.exe /list • $ C:\Temp\procwin.exe (Execute mimikatz)
Disabling antivirus software	TrueSightKiller is a tool designed to terminate antivirus (AV) and endpoint detection and response (EDR) processes. It allows attackers or red teamers to bypass security measures and disable targeted processes. • $ mytemp$\TrueSightKiller.exe -n smartscreen.exe
Install and run a customized tool to collect the victim's information	main.exe is a screenshot tool downloaded and installed by the CLNTEND backdoor via remote shell. • $ C:\Temp\main.exe

Table 1. The behaviors and corresponding logs in Earth Ammit's activities

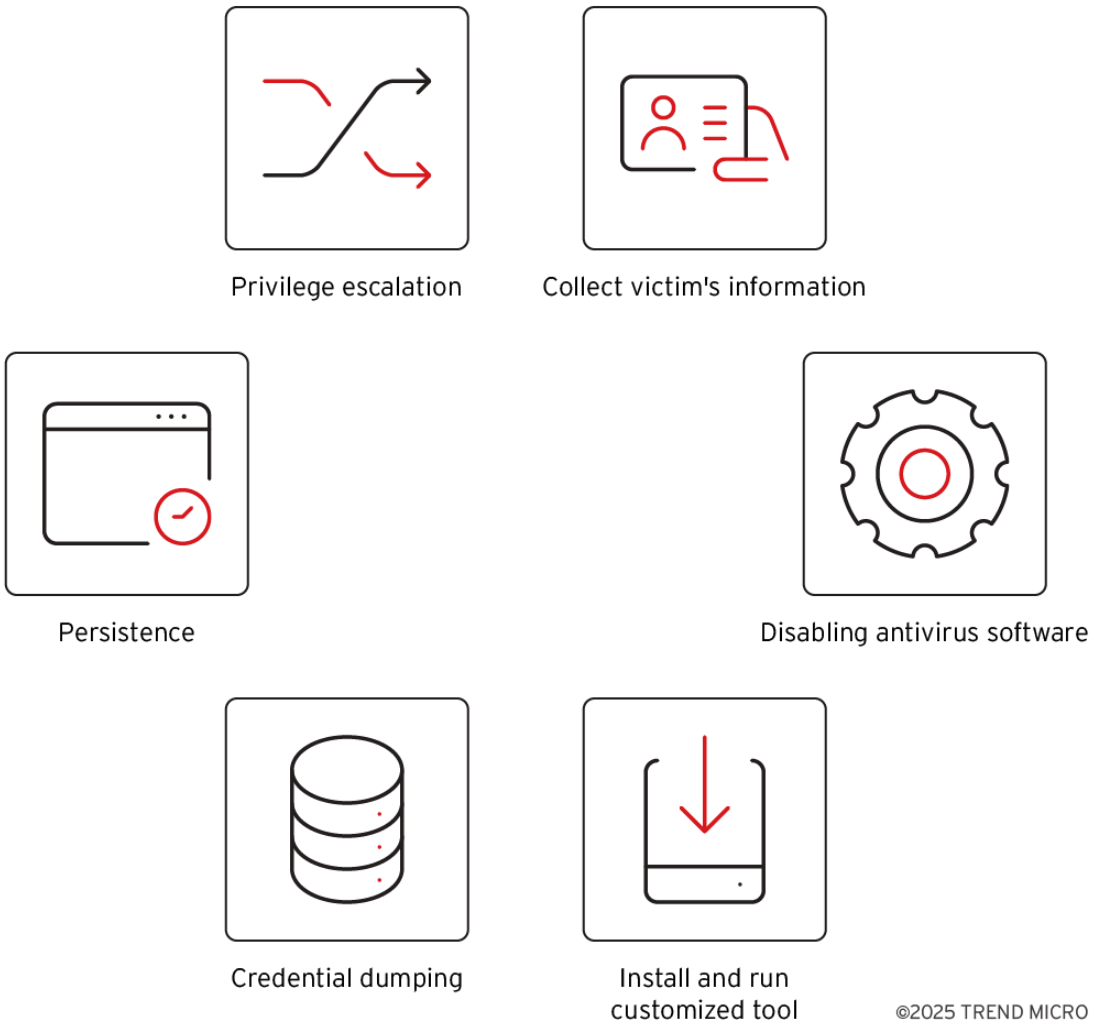

Figure 7. Post-exploitation was observed in the targeted machine

Malware analysis

As introduced in the previous section, we knew that VENOM campaign preferred using open-source tools instead of their own customized tools to hide their footprint (Figure 8). There's only one customized tool called VENFRPC. This could be a strong characteristic of the attribution to the attacker. For the arsenal of campaign TIDRONE, it used many customized tools, like CXCLNT, CLNTEND, and SCREENCAP (Figure 9).

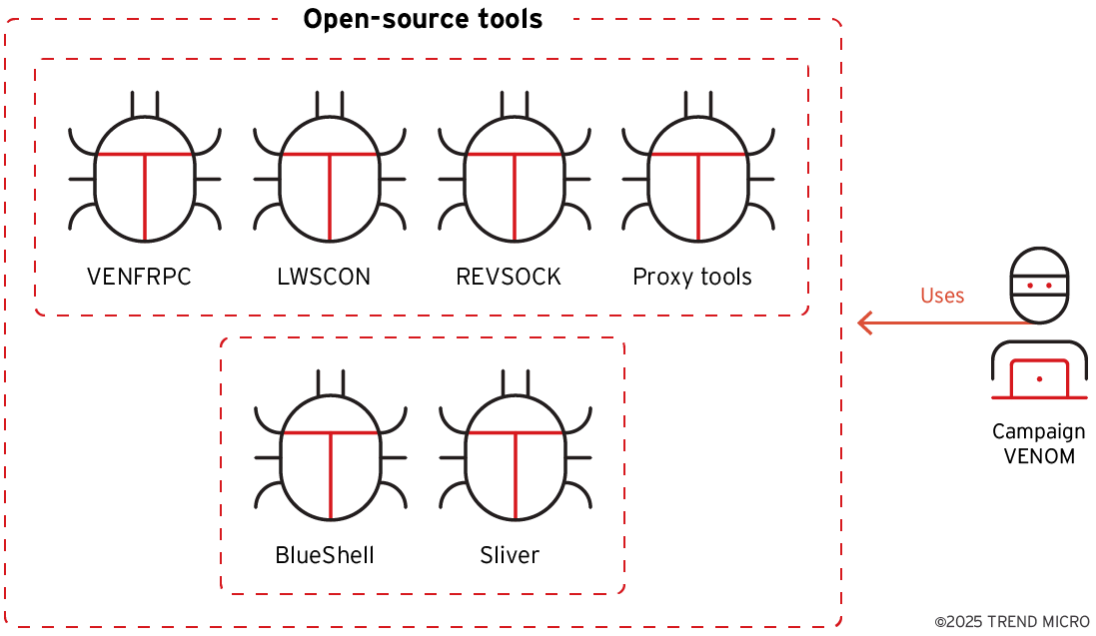

Figure 8. The arsenal of the campaign VENOM, with open-source tools

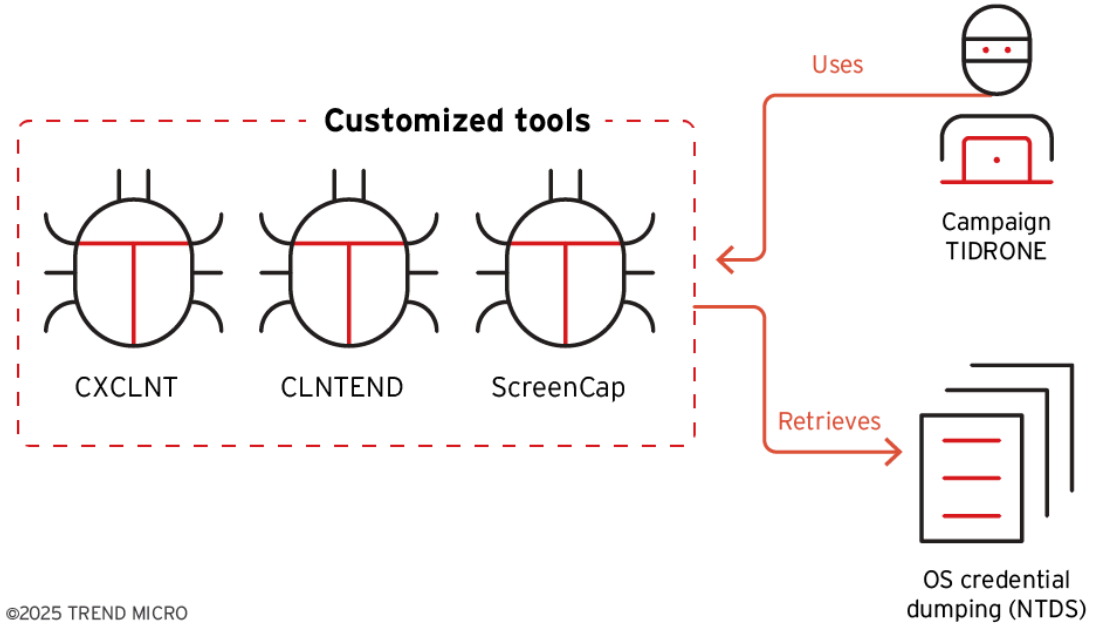

Figure 9. The arsenal of the campaign TIDRONE, with customized tools

Hacktool - VENFRPC

In the VENOM campaign, we observed a customized FRPC called VENFRPC that is slightly different from what we usually see on GitHub, as the configuration is directly embedded into the file itself. From this configuration format, we can see that the attacker tends to use the victim's identification details to make it easier to recognize their targets.

As shown in Figure 10, this GitHub repository has hosted multiple VENFRPC. Each VENFRPC has its own configuration and corresponds to different victims for easy management.

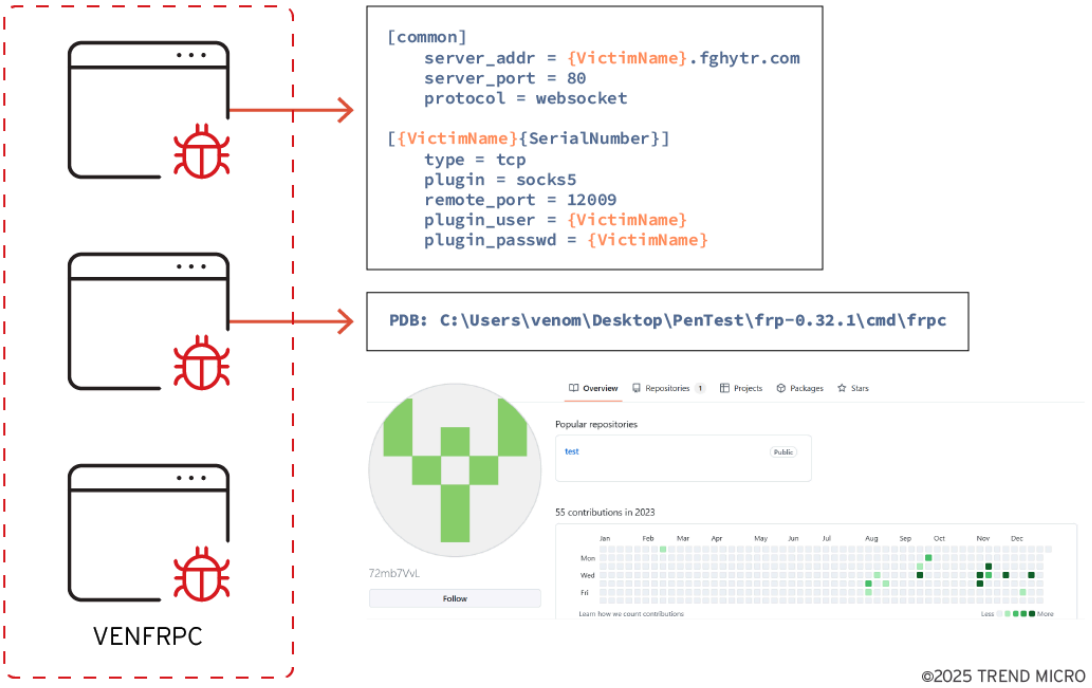

Figure 10. VENFRPC configuration and the host GitHub

Loader of CXCLNT/CLNTEND

Since our previous report, we have observed further evolution of the attacks. In 2023, the attacker started to use the fiber-based technique SwitchToFiber in their malware. In 2024, the loader switched to another fiber-based technique, FlsAlloc (Figure 11). Later the same year, the exception-handling technique also appeared in the malware. Interestingly, these fiber-based techniques appeared around the same time, and they were presented at BlackHat USA 2023 and BlackHat Asia 2024 by the same speaker, Daniel Jary. These talks likely inspired the threat actors to update their skill sets by developing fiber-based techniques to evade detection and monitoring.

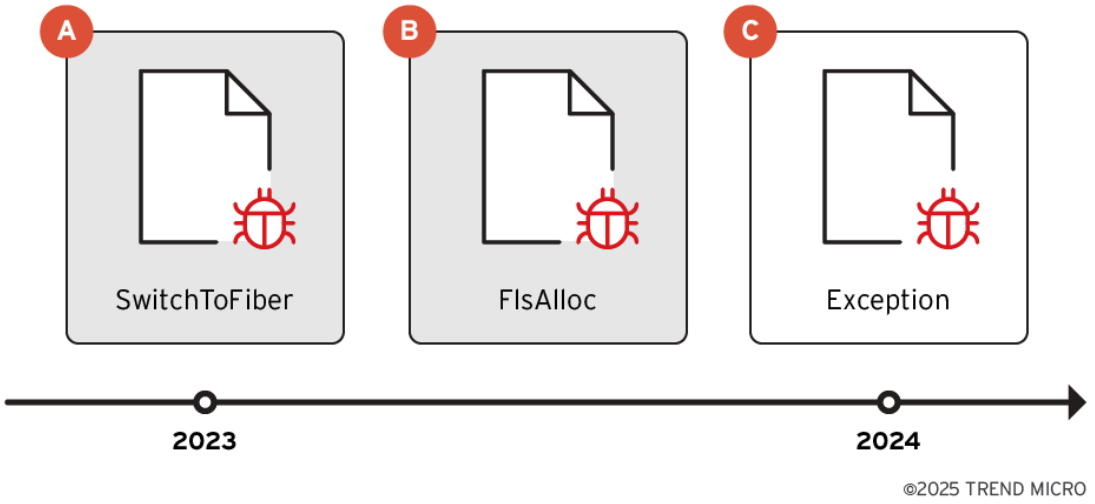

Figure 11. The evolution of the loader from 2023 to 2024

Based on our telemetry, we have identified three distinct versions of the loader.

Variant A - ConvertThreadToFiber

In this variant, the API ConvertThreadToFiber is applied to convert the current thread into a fiber, and allow it to switch to other fibers. Then, CreateFiber would create a new fiber within the same thread. The malicious code is placed at the fiber structure offset + 0xC4 in hex. Finally, SwitchToFiber switches execution to the new fiber and runs the malicious code (Figure 12).

```
hModule = hinstDLL;
ModuleHandleA = GetModuleHandleA(0);
dword_10013300 = *(_DWORD *)((char *)ModuleHandleA + *((_DWORD *)ModuleHandleA + 15) + 40);
lpFiber = ConvertThreadToFiber(0);
Fiber = (char *)CreateFiber(0, (LPFIBER_START_ROUTINE)StartAddress, 0);
dword_100132C4 = (int)Fiber;
*(_DWORD *)&Fiber[(dword_10013300 ^ 0x10EC) +0xC4]=(char *)sub_10001480 + (dword_10013300 ^ 0x10EC);
SwitchToFiber(Fiber);
```

Figure 12. The fiber-based technique in the loader with variant A

Variant B - FlsAlloc

FlsAlloc registers a fiber object's callback function. When the object is freed or deleted, the callback function will be triggered and execute the malicious code (Figure 13).

```
K32EnumProcesses(idProcess, 0x8Cu,
v3 = cbNeeded == 0x8C;
if ( cbNeeded == 0x8C )
  dwFlsIndex = FlsAlloc(Callback);
return v3;
```

```
FlsSetValue(dwFlsIndex + v11, Filename);
FlsFree(dwFlsIndex);
dword_74773308 = (int)CreateThread(0, 0, lpStartAddress, 0, 0, 0);
WaitForSingleObject((HANDLE)dword_74773308, 0xFFFFFFFF);
ExitProcess(0);
```

Figure 13. The fiber-based technique in the loader with variant B

Variant C - Exception

This technique leverages the exception handler, when the exception is triggered, the malicious code inside the custom handler function will be executed. As shown as Figure 14 and Figure 15, the custom exception handler would be triggered when the exception occurred, then it executes the payload by the callback function called by ImmEnumInputContext.

```
while ( v5 != -1 )
{
  v7 = *(int (__fastcall **)(_DWORD, _DWORD))(v6 + 12 * v5 + 4);// custom handler 1 -> loc_10001421
  if ( v7 )
  {
    v8 = v7(0, 0);
    v4 = (int *)v3[3];
    if ( v8 )
    {
      if ( v8 < 0 )
        return 0;
      CallDestructExceptionObject(v3[2], 1);
      v9 = v4[2];
      _global_unwind2(v4);
      v3 = v4 + 4;
      _local_unwind2((int)v4, v5);
      _NLG_Notify(1);
      v4[3] = *(_DWORD *)(v9 + 4 * v10);
      v4 = 0;
      v5 = 0;
      (*(void (__fastcall **)(_DWORD, _DWORD))(v9 + 4 * v10 + 8))(0, 0);// custom handler 2 -> loc_10001431
    }
  }
  v6 = v4[2];
  v5 = *(_DWORD *)(v6 + 12 * v5);
}
```

Figure 14. Custom exception handler installation

```
loc_10001421:                           ; DATA XREF: .rdata:stru_10011258↓o
;   __except filter // owned by 10001374
                mov     eax, [ebp+ms_exc.exc_ptr]
                mov     eax, [eax]
                mov     eax, [eax]
                mov     [ebp+var_2C], eax
                call    sub_10001150
                retn
; ---------------------------------------------------------------------------

loc_10001431:                           ; DATA XREF: .rdata:stru_10011258↓o
;   __except(loc_10001421) // owned by 10001374
                mov     esp, [ebp+ms_exc.old_esp]
                push    0               ; lParam
                push    lpfn            ; lpfn
                push    0               ; idThread
                call    ImmEnumInputContext
```

Figure 15. Custom exception handler

Anti-analysis

In addition to the fiber-based technique, there are two interesting anti-analysis techniques observed in the loader evolution.

Technique 1 - Entrypoint verification via GetModuleHandle and XOR check

This anti-analysis technique uses GetModuleHandle to retrieve information about the current process. Later, xor with specific bytes checks whether the entry point matched the expected target process (Figure 16).

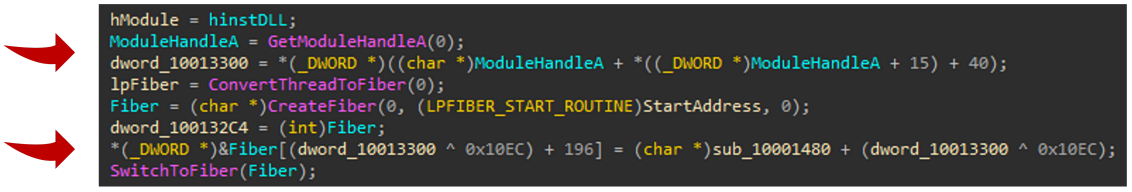

Figure 16. Anti-analysis through checking the expected parent process.

Technique 2 - Execution order dependency thwarts analysis attempts

This anti-analysis technique requires the correct order to execute the export functions (Figure 17). Since this loader distributes its decryption function and payload execution into different export functions, the process fails if the running order of export functions is wrong or applying rundll32.exe executes a specific export function.

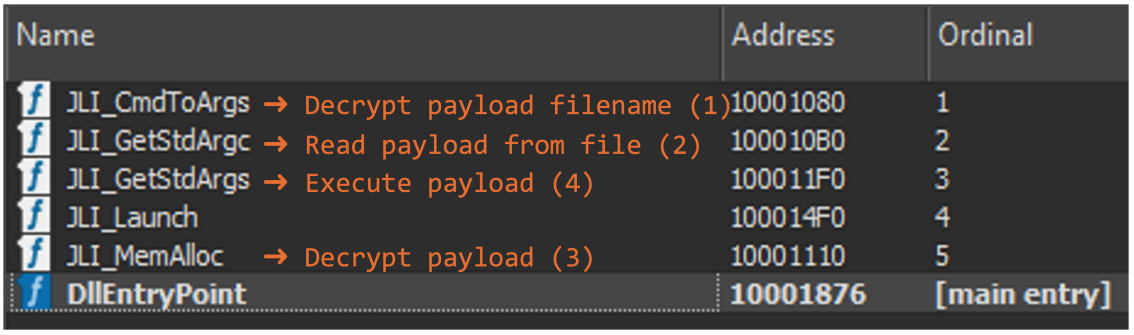

Figure 17. Export functions sequence defined by the legitimate host process

CXCLNT backdoor

Our telemetry data indicates that the CXCLNT backdoor has been applied since at least 2022. Notably, it operates entirely in memory with EXE format, never writing itself to disk, which enhances its stealth and makes detection significantly more challenging. For communication, it supports two traffic parsing methods: a custom protocol over SSL and standard HTTPS, allowing it to blend into legitimate encrypted traffic.

CXCLNT's core functionality is dependent on a modular plugin system. Upon execution, it retrieves additional plugins from its C&C server to extend its capabilities dynamically. This architecture not only obscures the backdoor's true purpose during static analysis but also enables flexible, on-demand operations based on the attacker's objectives.

Based on our hunting records, CXCLNT can be traced back to be used since 2022. It doesn't exist as a file; instead, it's decompressed and executed in memory. For network traffic, it supports two connection methods to parse traffic: one is SSL with custom protocol, and the other is using HTTPS. The main functionality depends on an extra plugin received from the C&C server. It makes analysis difficult to figure out the backdoor purpose, and easy to hide the intention.

Backdoor command

CXCLNT's command set is categorized into two main types: **general** and **plugin manipulation**

General manipulation

The commands shown in Table 2 cover fundamental backdoor functions commonly seen in other malware, such as system reconnaissance, updating embedded configurations, and executing shell commands on the compromised host.

Backdoor command	Behaviors
0x1001	Send victim information to C&C server, including: • BIOS • Computer name

	• config mark • host IP • OS
0x1002	Turn off backdoor
0x1003	SetEvent and turn off the backdoor
0x1004	Receive shellcode from C&C server
0x1005	Clear footprints • Delete loader and encrypted payload • Delete service
0x1006	Update the C&C server and write the encrypt C&C into registry software\\classes\\Licenses\\

Table 2. The backdoor command of CXCLNT in the general category

Plugin manipulation

CXCLNT supports runtime plugin installation, allowing the C&C server to deploy specialized modules as needed. These plugins can extend the backdoor's capabilities temporarily and are fully removable once their task is complete (Table 3). This plugin-based design supports a wide range of malicious operations while minimizing the backdoor's static footprint.

Backdoor command	**Behaviors**
0x2001	Receive the size of plugin
0x2002	Receive the payload of plugin
0x2003	Load plugin and write function into backdoor command: 0x2004-0x2007
0x2004	Unknown
0x2005	Call export function of plugin: Init
0x2006	Call export function of plugin: DeleteInstance
0x2007	Call export function of plugin: GetInstance

Table 3. The backdoor command of CXCLNT in the plugin manipulation category

CLNTEND backdoor

CLNTEND, first observed in 2024, is the evolved successor of the CXCLNT backdoor. Like its previous version, CLNTEND executes entirely in memory to evade detection, but it is delivered in the form of a DLL. This version implemented many features to adapt to various attack scenarios. One of CLNTEND's key improvements is its dual-mode design – supporting both client and server modes – based on the embedded configuration. It also supports a broader range of communication protocols, including:

- HTTP
- HTTPS
- SMB (port 445)
- TCP
- TLS
- UDP
- WebSocket

To hide its footprint, CLNTEND also includes anti-detection features such as process injection into dllhost.exe, a legitimate Windows process, and disabling EDR solutions.

CLNTEND organizes its capabilities into three primary command categories:

- Link - The link module provides the capability to choose one from seven kinds of connection methods and alternate the backdoor mode between client and server.
- Plugin - The plugin manipulation is similar to the first version CXCLNT, but only keeps two export functions, GetInstance and DeleteInstance.
- Session - It injected the remote shell into dllhost.exe. In one of our observed behaviors, we saw the commands are executed under winword.exe. In normal situations, winword.exe rarely executes cmd.exe directly, so we believe this injection is a technique used to evade detection or escalate privileges.

Comparison - CXCLNT vs CLNTEND

The comparison table for CXCLNT and CLNTED is shown in Table 4. CLNTEND does not only support more connection methods, but also equips more functionalities against AV solutions.

	CXCLNT	**CLNTEND**
Active time	2022 ~ 2024	2024 ~
Type	EXE	DLL
Victim information	• ComputerName • OS • Host IP • Net BIOS	• ComputerName • OS • UserName

Connection method	• HTTPS • SSL	TCP, HTTP, HTTPS, TLS, SMB (port:445), UDP, WebSocket
Anti-EDR	N/A	• EDRSilence • Blindside
Functionality	Client	• Server • Client
Backdoor module	General, Plugin	Plugin, Session, Link
Plugin export function	• Init • GetInstance • DeleteInstance	• GetInstance • DeleteInstance

Table 4. The comparison of features between CXCLNT and CLNTEND

We also found some similarities between the two backdoors. Both have a function that collects the victim's information for calculating a victim hash. This if-else statement indicates two modes: one is for testing, and another is for executing in the victim's environment. This flag is in the embedded configuration to control which mode is enabled (Figure 18).

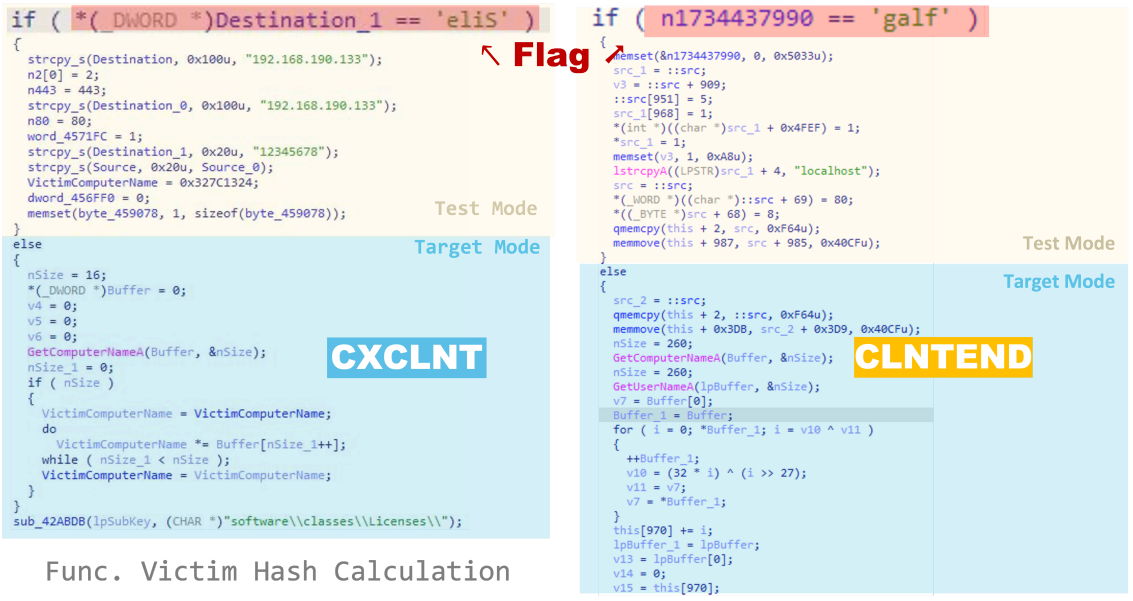

Figure 18. A similar code flows in the if-else statement to choose the mode in the infected environment

TrojanSpy - SCREENCAP

Another customized tool is ScreenCap, a screen capture tool installed by the CLNTEND backdoor through remote shell (Figure 19). It's adapted from an open-source tool, which can be found on the GitHub repository "vova616". It sends the victim's screenshots back to the C&C server.

```
if ( dword_8EAC70 )
{
  p_mjpeg_Stream = (mjpeg_Stream *)runtime_gcWriteBarrier1(p_mjpeg_Stream);
  *v12 = v16;
}
p_mjpeg_Stream->m = v16;
p_mjpeg_Stream->frame.len = 90LL;
p_mjpeg_Stream->frame.cap = 90LL;
if ( dword_8EAC70 )
{
  p_mjpeg_Stream = (mjpeg_Stream *)runtime_gcWriteBarrier1(p_mjpeg_Stream);
  *v11 = v15;
}
v13 = p_mjpeg_Stream;
p_mjpeg_Stream->frame.ptr = v15;
p_mjpeg_Stream->FrameInterval = 50000000LL;
v14 = runtime_newobject((const RTYPE *)&unk_668140);
*v14 = main_main_func1;
v14[2] = 23LL;
v14[1] = "[redacted]";
v14[4] = 4LL;
v14[3] = "8880jpeg";
v5 = runtime_newobject((const RTYPE *)&unk_6681E0);
*v5 = main_main_func2;
if ( dword_8EAC70 )
{
  v5 = (_QWORD *)runtime_gcWriteBarrier2();
  *v10 = v14;
  v10[1] = v13;
}
v5[1] = v14;
v5[2] = v13;
v9 = runtime_newproc((_DWORD)v5, 90, (_DWORD)v13, v0, v1, v6, v7, v8);
runtime_block(v9);
```

Figure 19. The main code structure inside the ScreenCap

Attribution

Our analysis links the VENOM and TIDRONE campaigns (Figure 20) through two primary indicators:

- **Shared victims and service providers** - Several organizations appear in both campaigns, indicating a sustained interest by the threat actor in specific entities across multiple operations.
- **Overlapping C&C infrastructure** - The use of common C&C domains, including the notably named *fuckeveryday[.]life*, further strengthening the connection.

These overlaps strongly suggest that both VENOM and TIDRONE were orchestrated by the same threat actor or group.

For attribution, the attacker might be launched by a Chinese-speaking threat actor with these observations:

- **Timestamps** from file compilation and command execution logs align with the GMT+8 time zone, which corresponds to regions such as China, Taiwan, and parts of Southeast Asia.

- The attacker's **tactics, techniques, and procedures (TTPs)** – as well as their target profile – bear resemblance to those used by Dalbit, a threat group previously reported by AhnLab. While we do not claim definitive attribution, the operational similarities are notable and suggest a potential connection or shared toolkit.

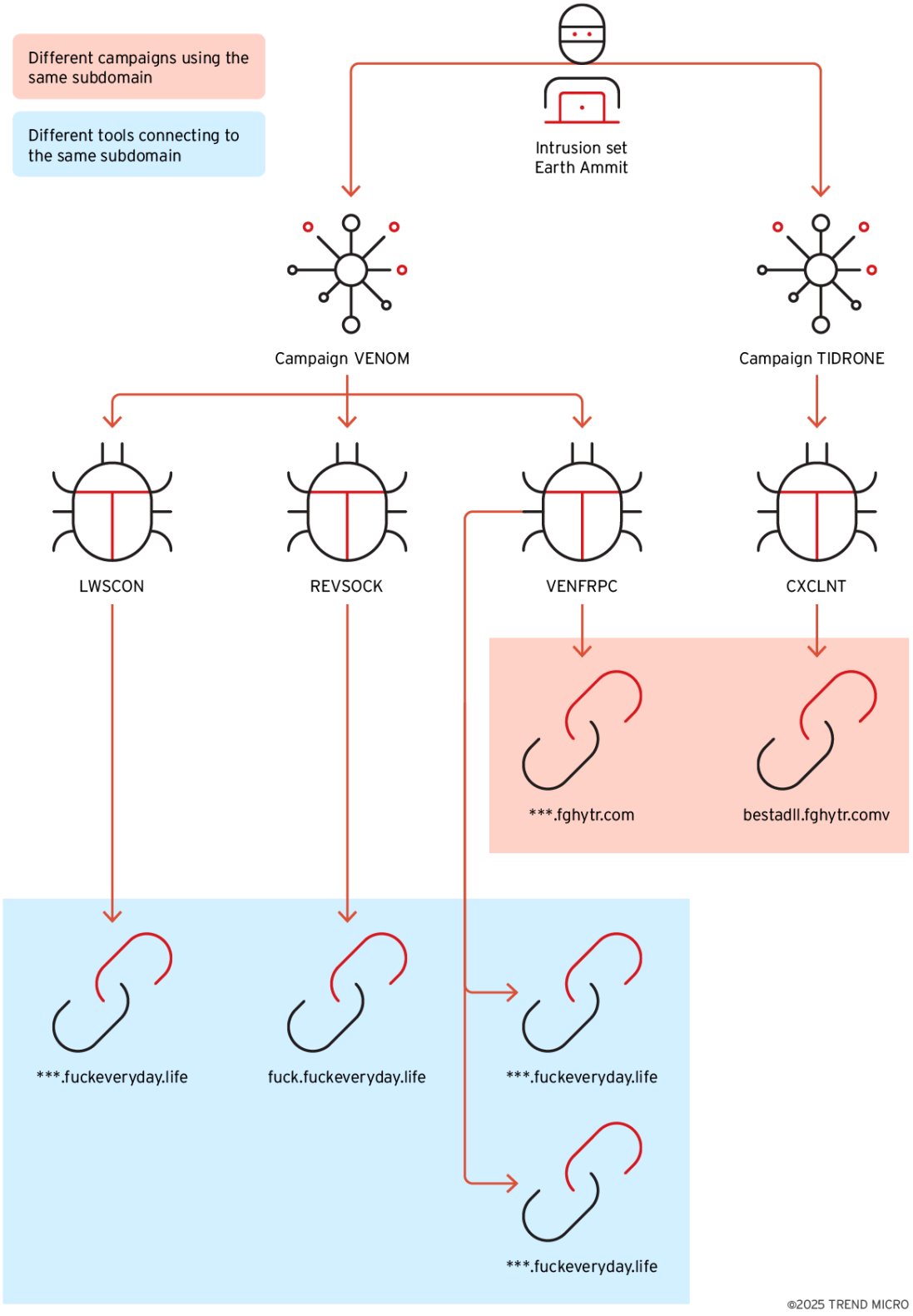

Figure 20. The relation and overlap connecting the VENOM and TIDRONE campaigns

Conclusion

Our investigation into the VENOM and TIDRONE campaigns reveals several key trends in Earth Ammit's evolving tradecraft. First, we observed a growing reliance on fiber-based evasion techniques across their malware arsenal – an approach designed to bypass traditional detection mechanisms more effectively. Second, both campaigns carry out supply chain attacks across two distinct attack waves. This highlights the adversary's long-term objective of infiltrating trusted networks to reach high-value targets. Continued monitoring of their infrastructure and toolset is essential to anticipate their next move.

In the VENOM campaign, Earth Ammit primarily leveraged open-source tools, likely due to their accessibility, low cost, and ability to blend in with legitimate activity. However, as the operation matured, they shifted toward deploying custom-built malware – notably in the TIDRONE campaign – to increase precision and stealth in targeting sensitive sectors.

This progression underscores a deliberate strategy: start broad with low-cost, low-risk tools to establish access, then pivot to tailored capabilities for more targeted and impactful intrusions. Understanding this operational pattern will be critical in predicting and defending against future threats from this actor.

To mitigate the risk of supply chain attacks, organizations may implement a third-party risk management program to assess vendors, verify software with Software Bills of Materials (SBOMs), enforce code signing, continuously monitor third-party software behavior, apply patches promptly, segment vendor systems, include third-party breach scenarios in incident response plans, and adopt Zero Trust Architecture to validate every connection.

Organizations may also better protect themselves from fiber-based techniques by monitoring the use of fiber-related APIs (such as ConvertThreadToFiber and CreateFiber) to detect abnormal behavior, strengthening EDR solutions to recognize fiber-based anomalies, and enhancing behavioral monitoring to identify unusual execution patterns typical of fiber-based malware.

Proactive security with Trend Vision One™

Organizations can protect themselves from threats like these with Trend Vision One™ – the only AI-powered enterprise cybersecurity platform that centralizes cyber risk exposure management, security operations, and robust layered protection. This comprehensive approach helps you predict and prevent threats, accelerating proactive security outcomes across your entire digital estate. Backed by decades of cybersecurity leadership and Trend Cybertron, the industry's first proactive cybersecurity AI, it delivers proven results: a 92% reduction in ransomware risk and a 99% reduction in detection time. Security leaders can benchmark their posture and showcase continuous improvement to stakeholders. With Trend Vision One, you're enabled to eliminate security blind spots, focus on what matters most, and elevate security into a strategic partner for innovation.

Trend Micro™ Threat Intelligence

To stay ahead of evolving threats, Trend customers can access Trend Vision One™ Threat Insights, which provides the latest insights from Trend Research on emerging threats and threat actors.

Trend Vision One Threat Insights

- Threat Actors: Earth Ammit
- Emerging Threats: Earth Ammit Disrupts Drone Supply Chains Through Coordinated Multi-Wave Attacks in Taiwan

Trend Vision One Intelligence Reports (IOC Sweeping)

- *Earth Ammit Disrupts Drone Supply Chains Through Coordinated Multi-Wave Attacks in Taiwan*

Hunting Queries

Trend Vision One Search App

Trend Vision One customers can use the Search App to match or hunt the malicious indicators mentioned in this blog post with data in their environment.

Malware Detection for Earth Ammit Activities

eventName:MALWARE_DETECTION AND (malName:*VENFRPC* OR malName:*CXCLNT* OR malName:*CLNTEND* OR malName :*SCREENCAP*)

More hunting queries are available for Trend Vision One customers with Threat Insights Entitlement enabled.

Indicators of Compromise (IOCs)

The indicators of compromise for this entry can be found here.

With additional insights from Cyris Tseng and Leon M Chang.

Tags
