New threat actor, UAT-9921, leverages VoidLink framework in campaigns

By Nick Biasini

Published: 2026-02-11 · Archived: 2026-04-05 13:33:59 UTC

Tuesday, February 10, 2026 19:00

- Cisco Talos recently discovered a new threat actor, UAT-9921, leveraging VoidLink in campaigns. Their activities may go as far back as 2019, even without VoidLink.
- The VoidLink compile-on-demand feature lays down the foundations for AI-enabled attack frameworks, which can create tools on-demand for their operators.
- Cisco Talos found clear indications that implants also exist for Windows, with the capability to load plugins.
- VoidLink is a near-production-ready proof of concept for an enterprise grade implant management framework, and features auditability and oversight for non-operators.

VoidLink is a new modular framework that targets Linux based systems. Modular frameworks are prevalent on the landscape today with the likes of Cobalt Strike, Manjusaka, Alchimist, and SuperShell among the many operating today. This framework is yet another implant management framework denoting a consistent and concerning evolution with shorter development cycles.

Cisco Talos is tracking the threat actor first seen to be using the VoidLink framework as UAT-9921. This threat actor seems to have been active since 2019, although they have not necessarily used VoidLink over the duration of their activity. UAT-9921 uses compromised hosts to install VoidLink command and control (C2) which are then used to launch scanning activities both internal and external to the network.

Who is UAT-9921?

Cisco Talos assesses that this threat actor has knowledge of Chinese language based on the language of the framework, code comments and code planning done using the AI enabled IDE. We also assess with medium confidence that they have been active since at least 2019, not necessarily using VoidLink.

VoidLink development appears to be a more recent addition with the aid of large language model (LLM) based integrated development environment (IDE). However, in their compromise and post-compromise operations, UAT-9921 does not seem to be using AI-enabled tools.

Cisco Talos was able to determine that the operators deploying VoidLink have access to the source code of some modules and some tools to interact with the implants without the C2. This indicates inner knowledge of the communication protocols of the implants.

While the development of VoidLink seems to be split into teams, it is unclear what level of compartmentalization exists between the development and the operation. We do know that UAT-9921 operators have access to VoidLink source code of kernel modules, as well as tools that enable interaction with the implant without the C2.

Talos assesses with high confidence that UAT-9921 compromises servers with the usage of pre-obtained credentials or exploiting Java serialization vulnerabilities which allow remote code execution, namely Apache Dubbo project. We also found indications of possible initial compromise via malicious documents, but no samples were obtained.

In their post-compromise activities, UAT-9921 deploys the VoidLink implant. This allows the threat actor to hide their presence and the VoidLink C2, once deployed.

To find new targets and perform lateral movement, UAT-9921 deploys a SOCKS server on their compromised servers, which is used by FSCAN to perform internal reconnaissance.

With regard to victimology, UAT-9921 appears to focus on the technology sector, but we have also seen victims from financial services. However, the cloud-aware nature of VoidLink and scanning of entire Class C networks indicates that there is no specific targeting.

Given VoidLink's auditability and oversight features, it is worth noting that even though UAT-9921 activity involves usage of exploits and pre-obtained credentials, Talos cannot discount the possibility that this activity is part of red team exercises.

Timeline

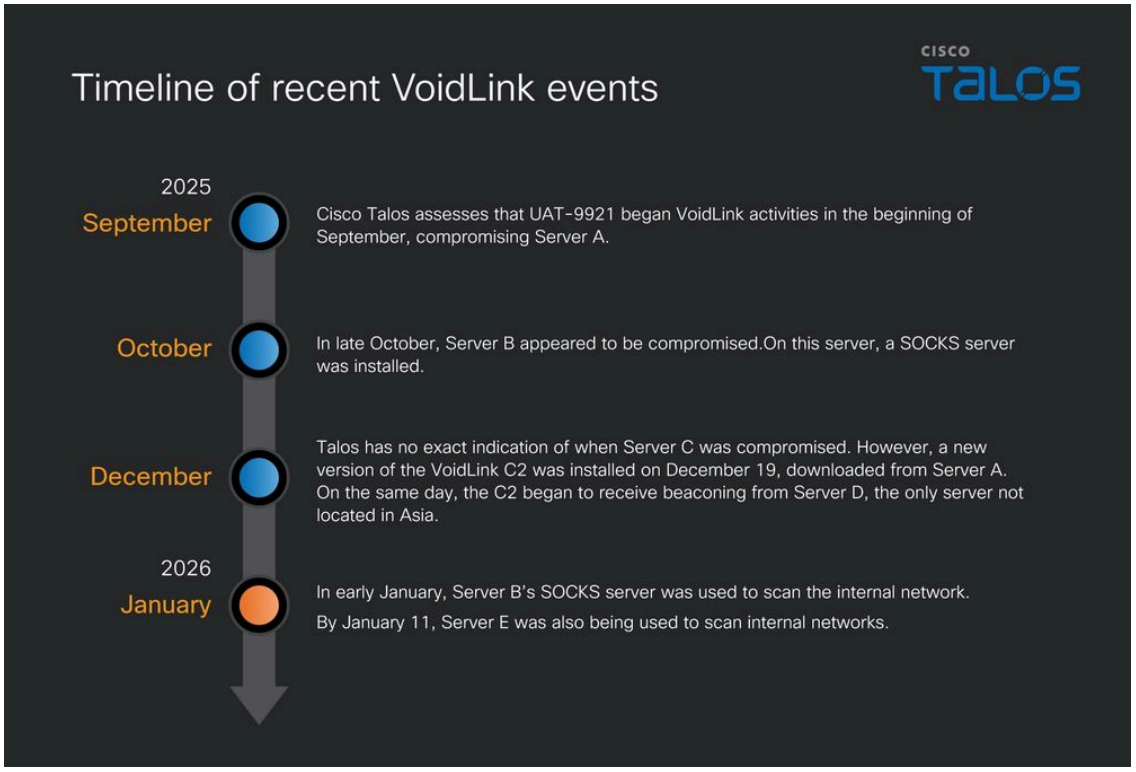

Figure 1. Timeline of activities involving UAT-9921 and VoidLink.

Talos is aware of multiple VoidLink-related victims dating back to September with the activity continuing through to January 2026. This finding does not necessarily contradict the Checkpoint Research mentions of late November since the presented documents show development dates from version 2.0 and Cisco Talos assesses that this was still version 1.0.

The future of attack frameworks

Talos has been tracking fast deployment frameworks since 2022, with reports on Manjusaka and Alchimist/Insekt. These two projects were tightly linked in their development philosophy, features, and architectural design. There were obvious inspirations from CobaltStrike and Sliver; however, one fundamental difference was the single file infrastructure and the lack of integrated initial infector vector.

The VoidLink framework represents a giant leap in this predictable evolution, while keeping the same, single file infrastructure philosophy. This is a clear example of a “defense contractor grade” implant management framework, which represents one natural next step of other single file infrastructure frameworks like Manjusaka and Alchimist.

The development of VoidLink was fast, supported on AI-enabled integrated development environments. It uses three different programing languages: ZigLang for the implant, C for the plugins and GoLang for the backend. It supports compilation on demand for plugins, providing support for the different Linux distributions that might be targeted. The reported development timeline of around two months would be hard to achieve by a small team of developers without the help of an AI-enabled IDE.

While Talos will discuss the framework in more detail below, it is important to reflect on what is to come in the framework landscape. With the current level of AI agents, it will not be surprising to find implants that ask their C2 for a “tool” that allows them to access certain resources.

The C2 will provide that implant with a plugin to read a specific database the operator has found or an exploit for a known vulnerability, which just happens to be on an internal web server. The C2 doesn’t necessarily need to have all these tools available — it may have an agent that will do its research and prepare the tool for the operator to use. With the current VoidLink compile-on-demand capability, integrating such feature should not be complex. Keep in mind that all of this will happen while the operator continues to explore the environment.

Of course, this may just be an intermediate step, assuming that there is a human operator managing the environment exploration. However, it likely will not be long before we begin to uncover malicious agents doing the initial stages of exploration and lateral movement before human intervention.

This has an impact of reducing compromise attack metrics — namely, the time to lateral movement and time to focused data exfiltration. It also allows the generation of never-before-seen tools and the constant change in the attacker’s behavior, making detection more difficult.

VoidLink Overview

VoidLink contains features that make it “defense contractor grade,” such as the auditability of all actions and the existence of a role-based access control (RBAC). The RBAC consists of three different levels of roles:

“SuperAdmin,” “Operator,” and “Viewer.” This feature is not often seen in other similar frameworks, but it is crucial when operations need to have legal and corporate oversight.

The mesh peer-to-peer (P2P) and dead-letter queue routing capabilities allow some implants to communicate with others, creating hidden networks with-in the same environment allowing the bypass of network access restrictions, as one implant may serve as external gateway for other implants.

The development timeline reported by Checkpoint Research indicates that this is a near-production-ready proof of concept. Most frameworks support Windows and MacOS from their early stages of development; VoidLink only appears to have implants developed for Linux, although the implant code is written in such a way that can easily be adapted to other languages. The main implant is written in ZigLang, a rather uncommon language; however the plugins are written in C. When needed these are loaded via an ELF linker and loader.

Talos has found clear indications that the main implant has been compiled for Windows and that it can load plugins via dynamic-link library (DLL) sideloading. Unfortunately, we were unable to obtain a sample to confirm these indications.

The Linux implants have advanced features, such as an eBPF or Loadable Kernel Module (LKM) based rootkit, container privilege escalation, and sandbox escape. These are often related with the server side, but there are a multitude of plugins in the implant targeting Linux as a desktop and not a server, something which is not often seen on malware since the Linux desktop base is not as prevalent as Windows or MacOS.

Most of the modular frameworks Talos observes support a wide variety of platforms typically inclusive of Linux, Windows, and MacOS — but VoidLink is different. The VoidLink framework specifically targets Linux devices without any current support for Windows or MacOS. Linux is a particularly large landscape, with the Internet of Things (IoT) and critical infrastructure heavily relying on the Linux OS.

As with most frameworks, VoidLink can generate implants consisting of a variety of plugins. The plugins themselves are standard, with the ability to interact and extract information from end systems, as well as capabilities allowing for lateral movement and anti-forensics. VoidLink is also cloud-aware and can determine if it is running in a Kubernetes or Docker environment, then gather additional information to make use of the vendor’s respective APIs. It has stealth mechanisms in place, including the ability to detect endpoint detection and response (EDR) solutions and create an evasion strategy based on the findings. There are also a variety of obfuscation and anti-analysis capabilities built into the framework designed to either obfuscate the data being exfiltrated or hinder the analysis and removal of the malware itself.

VoidLink is positioned to become an even more powerful framework based on its capabilities and flexibility, as demonstrated through this apparent proof of concept.

Coverage

The following Snort Rules (SIDs) detect and block this threat:

- Snort2: 1:65915 - 1:65922, 1:65834-65842
- Snort3: 1:65915 - 1:65922, 1:65834-65838, 1:310388-1:310389

The following ClamAV signature detects and blocks this threat:

- Unix.Trojan.VoidLink-10059283

More details on how Cisco detects threats like VoidLink is available here.
