Suspected China-Nexus Threat Actor Actively Exploiting Critical Ivanti Connect Secure Vulnerability (CVE-2025-22457)

By Mandiant

Published: 2025-04-03 · Archived: 2026-04-02 11:17:01 UTC

Written by: John Wolfram, Michael Edie, Jacob Thompson, Matt Lin, Josh Murchie

---

On Thursday, April 3, 2025, Ivanti disclosed a critical security vulnerability, CVE-2025-22457, impacting Ivanti Connect Secure ("ICS") VPN appliances version 22.7R2.5 and earlier. CVE-2025-22457 is a buffer overflow vulnerability, and successful exploitation would result in remote code execution. Mandiant and Ivanti have identified evidence of active exploitation in the wild against ICS 9.X (end of life) and 22.7R2.5 and earlier versions. Ivanti and Mandiant encourage all customers to upgrade as soon as possible.

The earliest evidence of observed CVE-2025-22457 exploitation occurred in mid-March 2025. Following successful exploitation, we observed the deployment of two newly identified malware families, the TRAILBLAZE in-memory only dropper and the BRUSHFIRE passive backdoor. Additionally, deployment of the previously reported SPAWN ecosystem of malware attributed to UNC5221 was also observed. UNC5221 is a suspected China-nexus espionage actor that we previously observed conducting zero-day exploitation of edge devices dating back to 2023.

A patch for CVE-2025-22457 was released in ICS 22.7R2.6 on February 11, 2025. The vulnerability is a buffer overflow with a limited character space, and therefore it was initially believed to be a low-risk denial-of-service vulnerability. We assess it is likely the threat actor studied the patch for the vulnerability in ICS 22.7R2.6 and uncovered through a complicated process, it was possible to exploit 22.7R2.5 and earlier to achieve remote code execution.

Ivanti released patches for the exploited vulnerability and Ivanti customers are urged to follow the actions in the Security Advisory to secure their systems as soon as possible.

Post-Exploitation Tactics, Techniques, and Procedures

Following successful exploitation, Mandiant observed the deployment of two newly identified malware families tracked as TRAILBLAZE and BRUSHFIRE through a shell script dropper. Mandiant has also observed the deployment of the SPAWN ecosystem of malware. Additionally, similar to previously observed behavior, the actor attempted to modify the Integrity Checker Tool (ICT) in an attempt to evade detection.

Shell-script Dropper

Following successful exploitation of CVE-2025-22457, Mandiant observed a shell script being leveraged that executes the TRAILBLAZE dropper. This dropper injects the BRUSHFIRE passive backdoor into a running

`/home/bin/web` process. The first stage begins by searching for a `/home/bin/web` process that is a child process of another `/home/bin/web` process (the point of this appears to be to inject into the `web` process that is actually listening for connections). It then creates the the following files and associated content:

- `/tmp/.p` : contains the PID of the `/home/bin/web` process.
- `/tmp/.m` : contains a memory map of that process (human-readable).
- `/tmp/.w` : contains the base address of the `web` binary from that process
- `/tmp/.s` : contains the base address of `libssl.so` from that process
- `/tmp/.r` : contains the BRUSHFIRE passive backdoor
- `/tmp/.i` : contains the TRAILBLAZE dropper

The shell script then executes `/tmp/.i` , which is the second stage in-memory only dropper tracked as TRAILBLAZE. It then deletes all of the temporary files previously created (except for `/tmp/.p`), as well as the contents of the `/data/var/cores` directory. Next, all child processes of the `/home/bin/web` process are killed and the `/tmp/.p` file is deleted. All of this behavior is non-persistent, and the dropper will need to be re-executed if the system or process is rebooted.

TRAILBLAZE

TRAILBLAZE is an in-memory only dropper written in bare C that uses raw syscalls and is designed to be as minimal as possible, likely to ensure it can fit within the shell script as Base64. TRAILBLAZE injects a hook into the identified `/home/bin/web` process. It will then inject the BRUSHFIRE passive backdoor into a code cave inside that process.

BRUSHFIRE

BRUSHFIRE is a passive backdoor written in bare C that acts as an `SSL_read` hook. It first executes the original `SSL_read` function, and checks to see if the returned data begins with a specific string. If the data begins with the string, it will XOR decrypt then execute shellcode contained in the data. If the received shellcode returns a value, the backdoor will call `SSL_write` to send the value back.

SPAWNSLOTH

As detailed in our previous blog post, SPAWNSLOTH acts as a log tampering component tied to the SPAWNSNAIL backdoor. It targets the `dslogserver` process to disable both local logging and remote syslog forwarding.

SPAWNSNARE

SPAWNSNARE is a utility that is written in C and targets Linux. It can be used to extract the uncompressed linux kernel image (vmlinux) into a file and encrypt it using AES without the need for any command line tools.

SPAWNWAVE

SPAWNWAVE is an evolved version of SPAWNANT that combines capabilities from other members of the SPAWN* malware ecosystem. SPAWNWAVE overlaps with the publicly reported SPAWNCHIMERA and RESURGE malware families.

Attribution

Google Threat Intelligence Group (GTIG) attributes the exploitation of CVE-2025-22457 and the subsequent deployment of the SPAWN ecosystem of malware to the suspected China-nexus espionage actor UNC5221. GTIG has previously reported UNC5221 conducting zero-day exploitation of CVE-2025-0282, as well as the exploitation CVE-2023-46805 and CVE-2024-21887.

Furthermore, GTIG has also previously observed UNC5221 conducting zero-day exploitation of CVE-2023-4966, impacting NetScaler ADC and NetScaler Gateway appliances. UNC5221 has targeted a wide range of countries and verticals during their operations, and has leveraged an extensive set of tooling, spanning passive backdoors to trojanized legitimate components on various edge appliances.

GTIG assesses that UNC5221 will continue pursuing zero-day exploitation of edge devices based on their consistent history of success and aggressive operational tempo. Additionally, as noted in our prior blog post detailing CVE-2025-0282 exploitation, GTIG has observed UNC5221 leveraging an obfuscation network of compromised Cyberoam appliances, QNAP devices, and ASUS routers to mask their true source during intrusion operations.

Conclusion

This latest activity from UNC5221 underscores the ongoing sophisticated threats targeting edge devices globally. This campaign, exploiting the n-day vulnerability CVE-2025-22457, also highlights the persistent focus of actors like UNC5221 on edge devices, leveraging deep device knowledge and adding to their history of using both zero-day and now n-day flaws. This activity aligns with the broader strategy GTIG has observed among suspected China-nexus espionage groups who invest significantly in exploits and custom malware for critical edge infrastructure.

Recommendations

Mandiant recommends organizations immediately apply the available patch by upgrading Ivanti Connect Secure (ICS) appliances to version 22.7R2.6 or later to address CVE-2025-22457. Additionally organizations should use the external and internal Integrity Checker Tool (“ICT”) and contact Ivanti Support if suspicious activity is identified. To supplement this, defenders should actively monitor for core dumps related to the web process, investigate ICT statedump files, and conduct anomaly detection of client TLS certificates presented to the appliance.

Acknowledgements

We would like to thank Daniel Spicer and the rest of the team at Ivanti for their continued partnership and support in this investigation. Additionally, this analysis would not have been possible without the assistance from analysts across Google Threat Intelligence Group and Mandiant's FLARE, we would like to specifically thank Christopher Gardner and Dhanesh Kizhakkinan of FLARE for their support.

Indicators of Compromise

To assist the security community in hunting and identifying activity outlined in this blog post, we have included indicators of compromise (IOCs) in a GTI Collection for registered users.

Code Family	MD5	Filename	Description
TRAILBLAZE	4628a501088c31f53b5c9ddf6788e835	/tmp/.i	In-memory dropper
BRUSHFIRE	e5192258c27e712c7acf80303e68980b	/tmp/.r	Passive backdoor
SPAWNSNARE	6e01ef1367ea81994578526b3bd331d6	/bin/dsmain	Kernel extractor & encryptor
SPAWNWAVE	ce2b6a554ae46b5eb7d79ca5e7f440da	/lib/libdsupgrade.so	Implant utility
SPAWNSLOTH	10659b392e7f5b30b375b94cae4fdca0	/tmp/.liblogblock.so	Log tampering utility

YARA Rules

```
 rule M_APT_Installer_SPAWNANT_1
{
    meta:
        author = "Mandiant"
        description = "Detects SPAWNANT. SPAWNANT is an
Installer targeting Ivanti devices. Its purpose is to persistently
install other malware from the SPAWN family (SPAWNSNAIL,
SPAWNMOLE) as well as drop additional webshells on the box."

    strings:
        $s1 = "dspkginstall" ascii fullword
        $s2 = "vsnprintf" ascii fullword
        $s3 = "bom_files" ascii fullword
```

```
        $s4 = "do-install" ascii
        $s5 = "ld.so.preload" ascii
        $s6 = "LD_PRELOAD" ascii
        $s7 = "scanner.py" ascii

    condition:
        uint32(0) == 0x464c457f and 5 of ($s*)
}
```

```
 rule M_Utility_SPAWNSNARE_1 {
    meta:
          author = "Mandiant"
        description = "SPAWNSNARE is a utility written in C that targets
Linux systems by extracting the uncompressed Linux kernel image
into a file and encrypting it with AES."

    strings:
        $s1 = "\x00extract_vmlinux\x00"
        $s2 = "\x00encrypt_file\x00"
        $s3 = "\x00decrypt_file\x00"
        $s4 = "\x00lbb_main\x00"
        $s5 = "\x00busybox\x00"
        $s6 = "\x00/etc/busybox.conf\x00"

    condition:
        uint32(0) == 0x464c457f
        and all of them

}
```

```
 rule M_APT_Utility_SPAWNSLOTH_2
{
    meta:
        author = "Mandiant"
        description = "Hunting rule to identify strings found in SPAWNSLOTH"

    strings:
        $dslog = "dslogserver" ascii fullword
        $hook1 = "g_do_syslog_servers_exist" ascii fullword
        $hook2 = "ZN5DSLog4File3addEPKci" ascii fullword
        $hook3 = "funchook" ascii fullword

    condition:
        uint32(0) == 0x464c457f and all of them
}
```

Posted in

- Threat Intelligence
