

CNCMachineRMS: The Undocumented RAT At the End of a BabaDeda Chain

levelblue.com/blogs/spiderlabs-blog/cncmachinerms-the-undocumented-rat-at-the-end-of-a-babadeda-chain

Rodel Mendrez

August 10, 2026

LevelB/ue
SpiderLabs Blog



- [Home](#)
- [SpiderLabs Blog](#)

This post is the result of an investigation into a case we worked on, in which we traced a loader chain that ended where we didn't expect.

The chain consists of a ClickFix lure, a legitimately signed IBM SPSS IDE, four decoy DLLs, and a date-formatting API used as a trampoline. What comes out at the other end is a 1.14 MB x64 remote administration implant. It has no imports. Every string is built on the stack at runtime instead of sitting in the binary. Its config and its command and control (C2) traffic travel in the same custom binary container, and the whole thing is driven by a custom scripting language the author wrote.

What it gives an operator is remote administration of the host: an interactive shell, a file manager, screen capture functionality, a local account backdoor, seven persistence mechanisms, and twenty typed commands for pulling down and running further payloads.

The implant names itself, buried in a path it builds at runtime using the path named CNCMachineRMS.

This is the short version. The full technical writeup, including the config format, the decoder, the complete command set and the reverse engineering detail, is available as a [PDF](#).

The Infection Flow

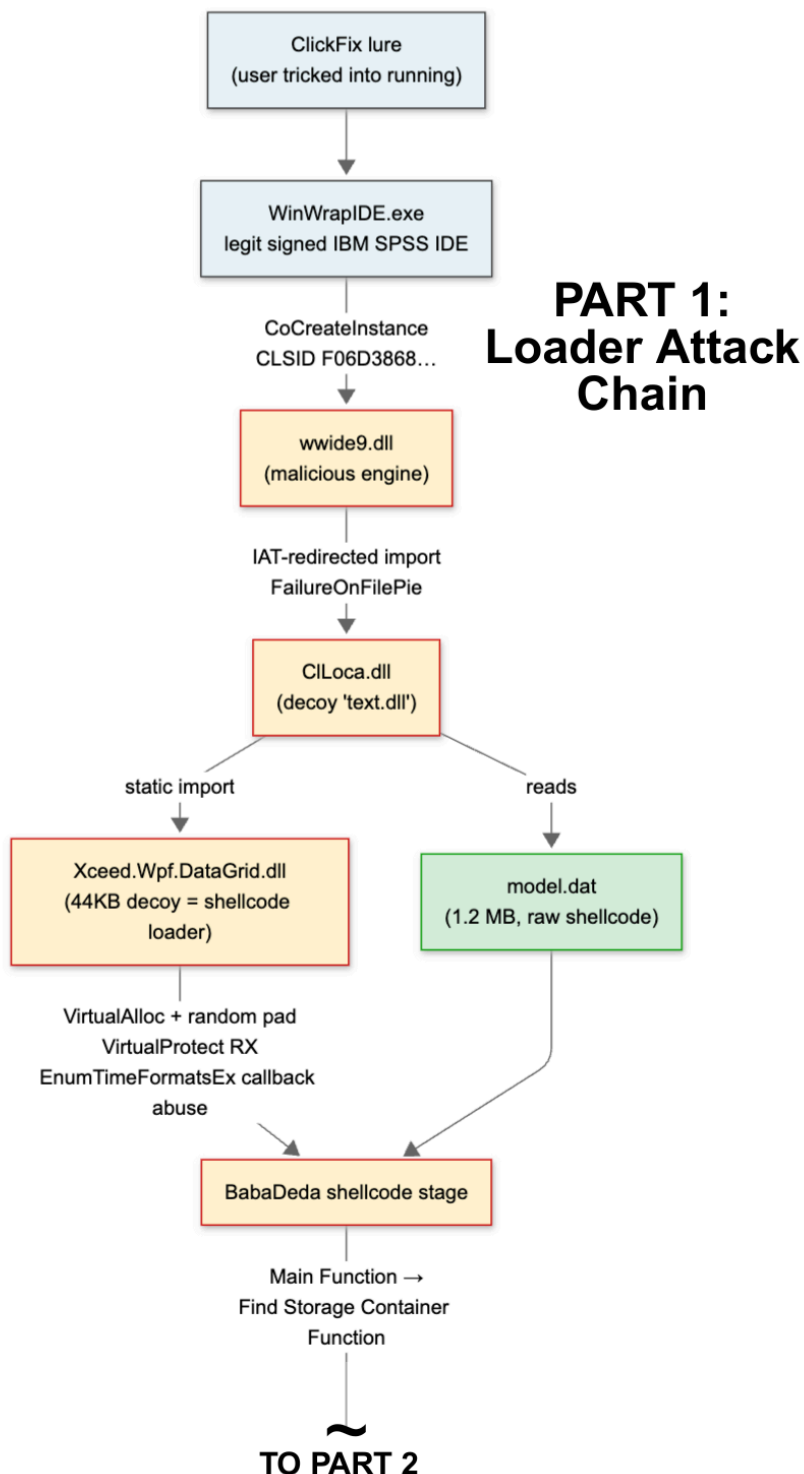


Figure 1. Part 1: The loader attack chain from the ClickFix lure to the final implant.

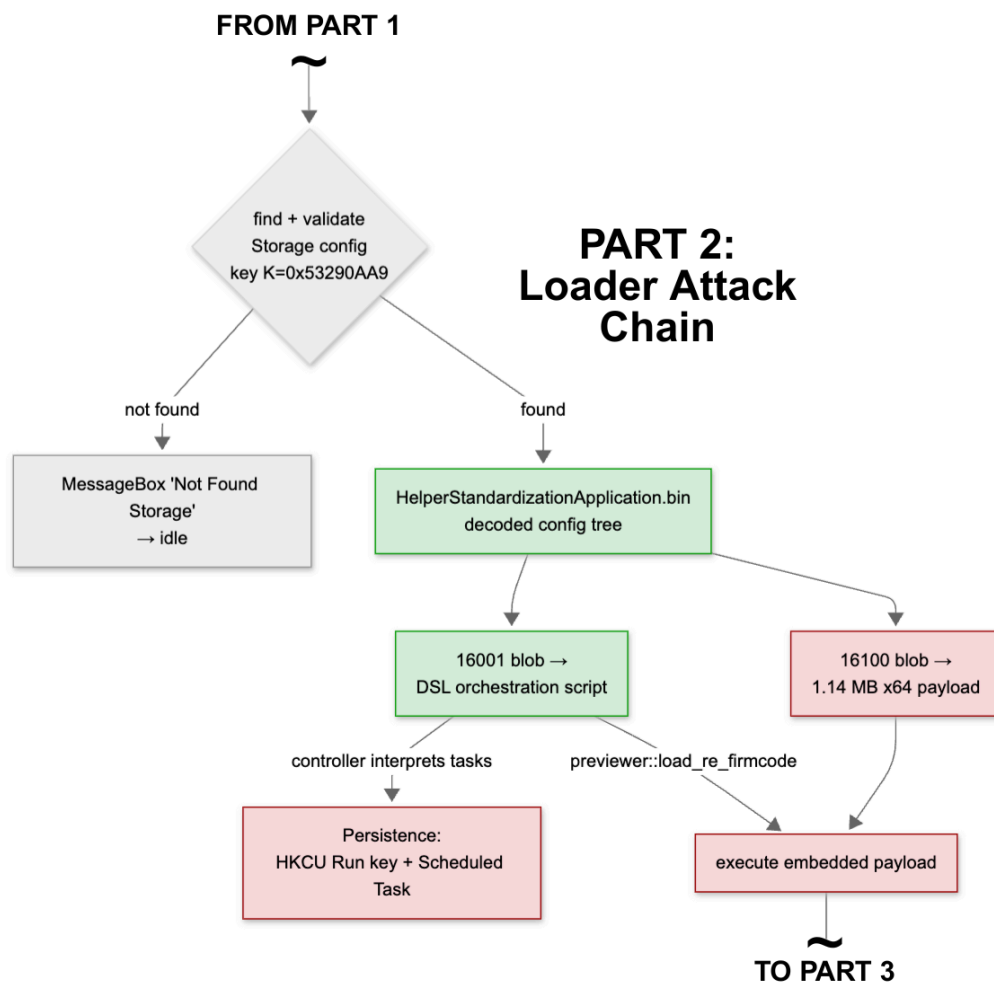


Figure 2. Part 2: The loader attack chain from the ClickFix lure to the final implant.

Dedicated to hunting and eradicating the world's most challenging threats.

[SpiderLabs](#)

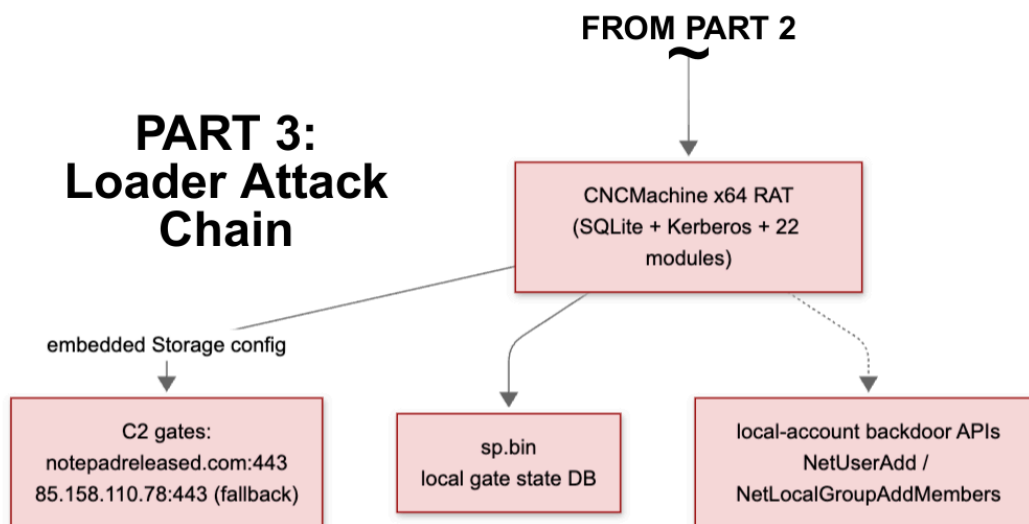


Figure 3. Part 3: The loader attack chain from the ClickFix lure to the final implant.

The initial infection was reported as ClickFix. At some later point in the chain, a legitimately signed IBM SPSS IDE, WinWrapIDE.exe, is launched. That binary is clean. It activates its scripting engine through COM, and the attacker has pointed that COM object at a dropped DLL. From there, a run of four decoy DLLs loads through ordinary import resolution, with no suspicious API calls and no odd LoadLibrary to flag.

The last decoy allocates a buffer, fills it with random noise to defeat signatures, writes shellcode into the middle of it, and marks it executable. It then passes the shellcode pointer as the callback argument to EnumTimeFormatsEx, a benign date formatting API. Windows calls the malware on the attacker's behalf, so the entry point is never invoked directly from the attacker's code.

That shellcode is the BabaDeda stage, and on its own, it does almost nothing. Run it in a sandbox and it pops a message box and idles, because it depends on a separate config file that was not there. That file, HelperStandardizationApplication.bin, looks like an encrypted blob at 8 bits of entropy per byte. It is not encrypted. It is a configuration tree under two layers of obfuscation, and it carries both a script and a 1.14 MB embedded payload.

That payload is the CNCMachineRMSimplant.

What the Attacker Gets

CNCMachineRMS is built for access, and everything it collects serves that. The 1,200+ strings we decoded from the binary describe an interactive shell, a file manager, screen capture functionality, a local account backdoor, seven persistence mechanisms, and twenty

typed commands for staging and running whatever the operator sends next. The reconnaissance it performs on first contact is there to help the attacker choose the next course of action.

That shapes how a defender may scope an infection. The damage is not bound by what the binary was compiled to take, because it was not compiled to take anything in particular. It is bound by what an operator decides to do across the dwell time. If you find this on a host, assume hands-on keyboard access and treat the follow-on payloads as the incident. Anything actually stolen arrived as a second stage.

A few other details worth noting about CNCMachineRMS:

- It takes a screenshot on first contact, then beacons every 600 seconds.
- Its host profile includes domain, SID, and elevation status, which is Active Directory context rather than fingerprinting, plus motherboard and BIOS serials and a list of installed antivirus software.
- It creates local accounts and adds them to privileged groups.
- The sandbox evasion is disabled in this build. The campaign is literally named novm.
- Creating a folder called C:\Intel switches the VM check off entirely; this is good for detonating the malware in a VM environment.

It Runs Its Own Scripting Language

CNCMachineRMS has a task engine with its own small language, stored as text in the config file, with namespaces like `delayer::`, `runtime::`, `autoruns::` and `communicator::` that map onto native handlers. Behavior that would normally be compiled in is instead scripted:

```
task autorun_to_registry[active=true,async=true]
{
  loop
  {
    delayer::delay_in_seconds(150);
    autorunToRegistry();
  }
}
```

The example script shown above is a persistence mechanism re-asserted every 150 seconds in a script, changeable per campaign without recompiling anything.

One container format for everything. The config on disk, a second config embedded in the payload's own memory image, the local state database, and the C2 messages are all the same serialized property tree. The implant even finds its own embedded config by scanning memory forward one byte at a time from a dummy function address until a header validates. There is no offset to hardcode and no magic value to grep for.

The 3.6 MB file that holds two booleans and a hostname. The state database at %LOCALAPPDATA%\SPProject\sp.bin reads at 8.00 bits per byte and looks exactly like an encrypted database. It is not encrypted. The real state is a hostname, a port, and two flags. The rest is a single property stuffed with BCryptGenRandom output, sized randomly between 512 KiB and 16 MiB, regenerated on every save. Every victim's copy differs in size and content, the hash changes roughly every 200 seconds.

Indicators of Compromise

Network

- Notepadreleased[.]com and 85[.]158.110.78, both TCP/443
- Beacon every 600 seconds, retry after 60 seconds on failure
- Domain resolution over DNS over HTTPS through dns.google, cloudflare-dns.com and dns.quad9.net, so internal resolver logs will show nothing

Host

- %TEMP%\CNCMachineRMS\tasks*\task_payload.bin
- %LOCALAPPDATA%\SPProject\sp.bin, a multi megabyte file of random size
- A Run key or scheduled task named IBM SPSS WinWrap Basic IDE
- An unexplained empty C:\Intel directory
- Windows event IDs 4720 and 4732, from the local account backdoor
- Scheduled tasks created with /SC ONLOGON /RU SYSTEM /F /RL HIGHEST

File hashes, SHA256

0562c588997fa9961d55c6ab1db2656344324bca8db9ea7b64fe309132b2399f
WinWrapIDE.exe (signed, abused)

5b71b49bad415643ff3e291ee3ba550e5edfd584eb913859dadb81634450e7e7 wwide9.dll

3d4e7187f74de912f9d52f5ba6a95bd41868348b16583d72957d732c0ed8f0e7 ClLoca.dll

b804b9dd2726e69fb4f496a6872f90edf9146ad8044c6d3a592040ce1074a5d0
Xceed.Wpf.DataGrid.dll

3466c3524f13cb3b7c87819e619bdb482ec9034a57bcdbe0240af6a9a530b02a
ComPDFKit.Viewer.dll

3b9b86feb3b789dda9cdfe5425ccb7e1feae9fde2cc158ff962991218a98f53f model.dat

2922837a8d049bf0b51f0f9b27340a377b27dd1912675e2cd82056db83ec7a19
HelperStandardizationApplication.bin

The two hashes below never touch disk. They are listed for anyone carving from memory:

744b8165b6161cbd1d5ecc423ecfdb8306485afdc80262000ab3c6908881ef9e BabaDeda
stage from model.dat

bb81786286be437b3ec6d562055767097c9277054e1d09172caa96376451481c CNCMachineRMS
payload

Read the Full Analysis

The complete write-up covers the sideloading chain in detail, the two obfuscation layers in the config file, the malware's own scripting language, the 20 command opcodes, the beacon format, and a working decoder you can run against your own samples.

[Read the full report.](#)

Stay Informed

Sign up to receive the latest security news and trends straight to your inbox from LevelBlue.

RESEARCH REPORT

[CNCMachineRMS: The Undocumented RAT At the End of a BabaDeda Chain](#)

ABOUT LEVELBLUE

LevelBlue secures what's next with intelligence-led security delivering visibility and speed to stop threats faster. As the world's largest and most analyst-recognized pure-play managed security services provider, our AI-powered managed services and cyber expertise across managed, advisory, and incident response services help clients operate with confidence.

Learn more [about us](#).

[Reports](#) [Security](#) [Research](#) [Threat Intelligence](#)