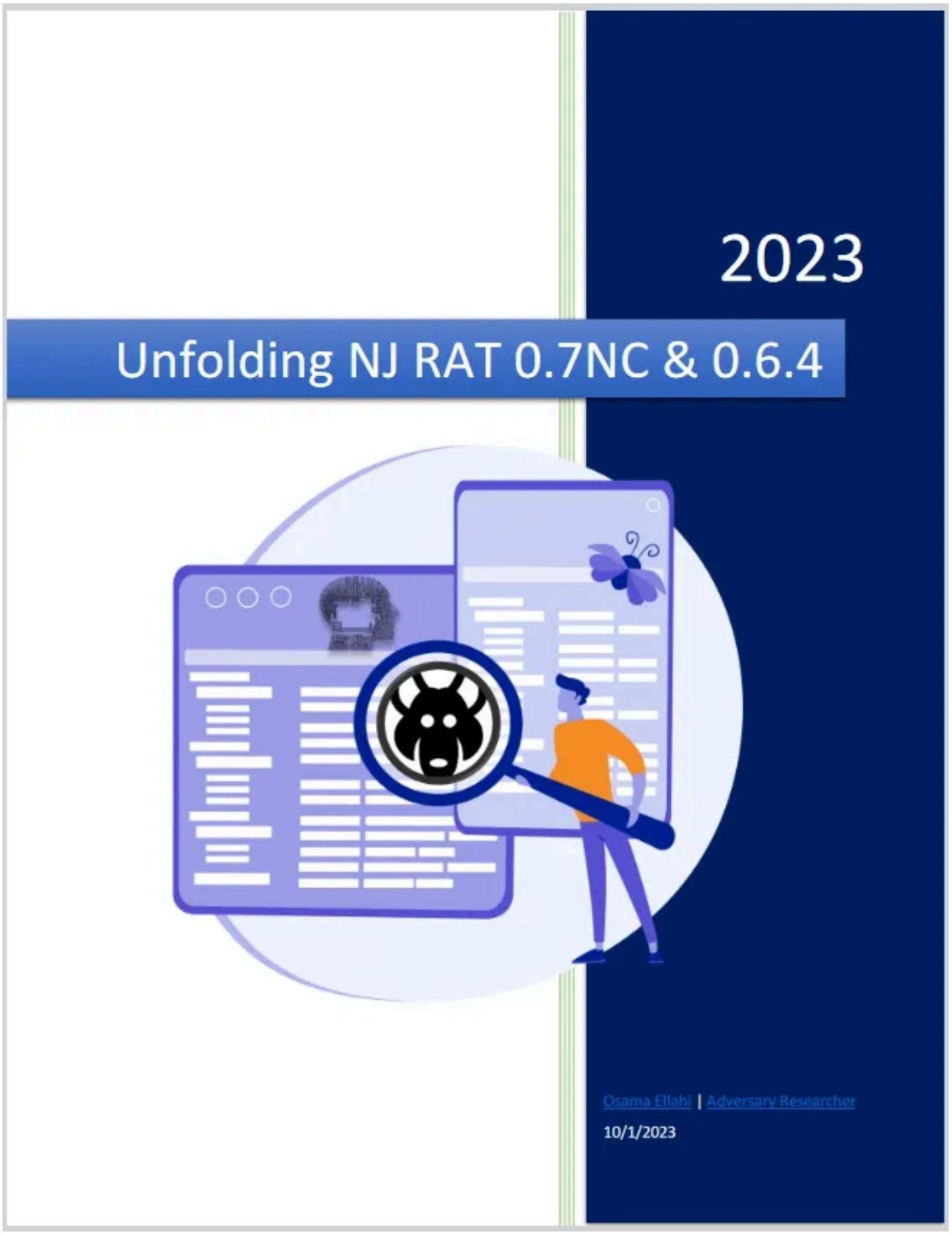
Unfolding NJ RAT 0.7NC & 0.6.4

By Osama Ellahi

Published: 2024-08-09 · Archived: 2026-04-05 18:02:34 UTC

| NJRAT Malware analysis



Executive Summary

This version {0.7NC} of **NJRat was first seen on 17 August 2023** with the name utah-Robert-magazine- speaker. It was delivered by email using phishing. Red Packet Security defines NJRat as a type of remote access trojan (RAT). This malicious software can do a range of things, like **recording keystrokes, accessing the victim's camera, stealing saved login information from web browsers**, creating a way for attackers to control the victim's computer from a remote location, transferring files to and from the victim's computer, **seeing what's on the victim's screen, making changes to files, processes, and the Windows registry, and even allowing the attacker to update, remove, restart, close, disconnect, or change the name of their attack campaign**.

This analysis comprises two samples labeled as NJ RAT 0.7NC and 0.6.4. The 0.7NC variant introduces a novel method for evading analysis, while 0.6.4 is responsible for managing all other malicious activities.

High-Level Technical Summary

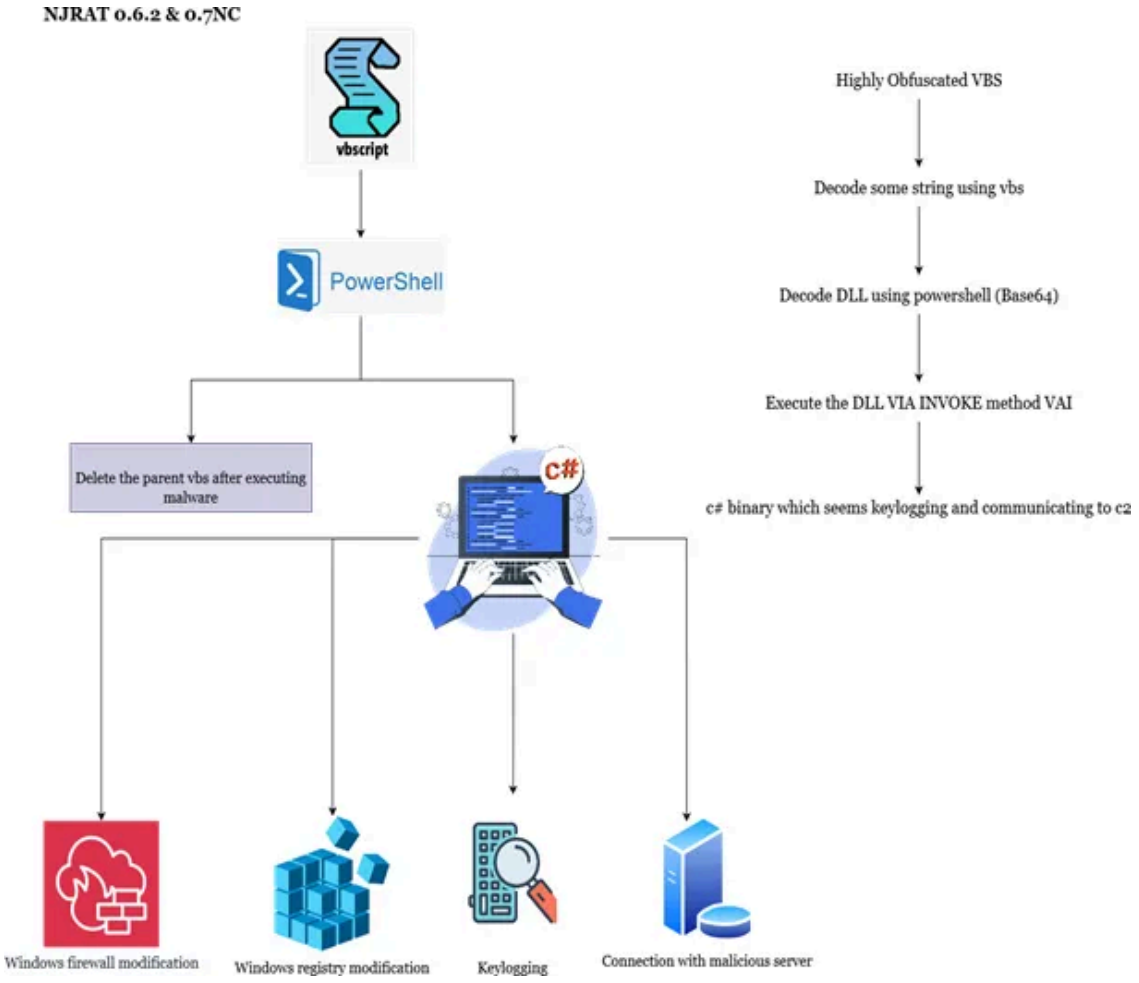
NJRAT is a sophisticated malware that operates in **two primary stages**. **The initial stage involves phishing and obfuscation tactics.** In August 2023, security experts first encountered malware, which was distributed via email in the form of a malicious and highly obfuscated VBS (Visual Basic Script) file embedded in documents.

Get Osama Ellahi's stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

Upon execution, **this VBS file performs deobfuscation and reveals a PowerShell script**. Within this script lies a base64-encoded DLL (Dynamic Link Library). Once the script successfully decodes the DLL, it proceeds to invoke the "VAI" method within the DLL. This marks the beginning of malware's further exploitation and malicious activities.



Initial Stage

This stage consists of deobfuscation and decoding of real dll and invoking the binary.2

> SHA256
>
> vbs = 5f66c7336f8469a6ab349a3f0f3f7aca1b483f2f2a8b4ad71af79ff51a8aad6b
>
> dll = 153c9ffe148909981900c59c2ccba8ef66f94688ce7ab5e01e3a541937a31294

.VBS

The initial executable comprises a VBS file containing obfuscated PowerShell code. After modifying the VBS file and revealing the de-obfuscated PowerShell code, we can observe its initial command in the terminal. This command involves **pinging localhost** for a dynamic delay, followed by the **self-copying of the executable to the startup folder**. This technique is employed to achieve persistence, ensuring that the executable runs every time the device starts up.

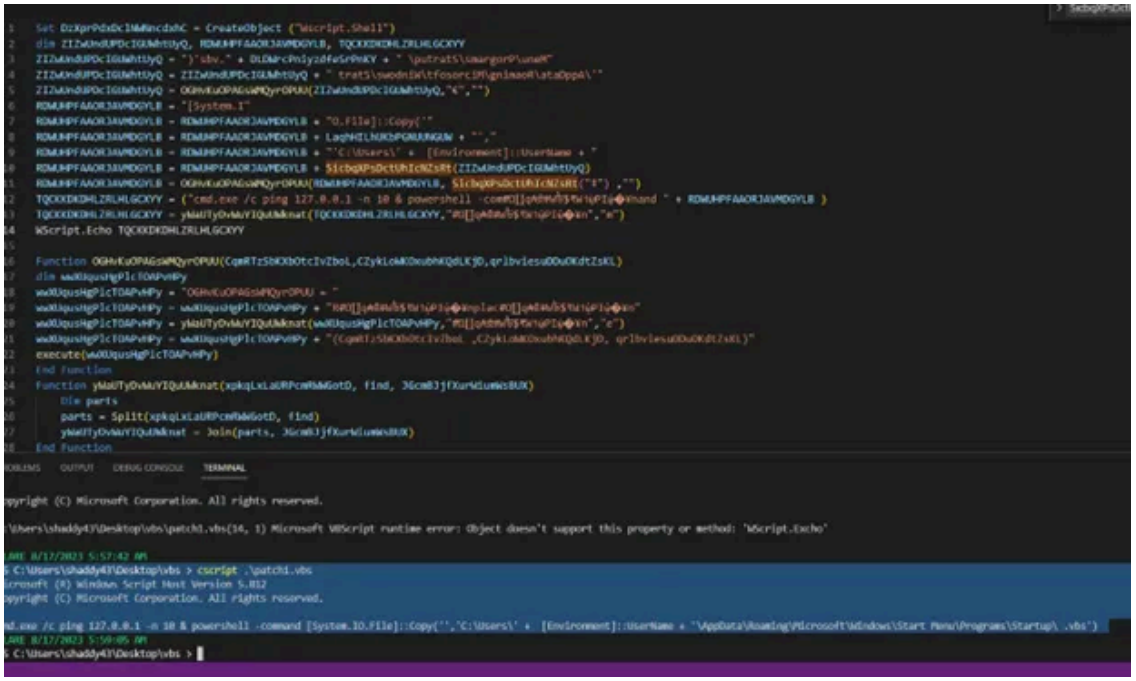


Figure 1 First Command of PowerShell which is responsible for persistence.

> This is the command which copy the malicious file in startup folder for future purposes.

> cmd.exe /c ping 127.0.0.1 -n 10 & powershell -command [System.IO.File]::Copy(‘’,’C:\Users\’ +
>
> [Environment]::UserName + ‘\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\.vbs’)

To See how it gains persistence and how it have its own language when it communicate with the C2 visit following link. I have moved this blog to my personal website.

https://breachnova.com/blog.php?id=27
