XWorm Malware: Exploring C&C Communication

By Igal Lytzki

Published: 2023-11-21 · Archived: 2026-04-05 22:32:10 UTC

In this article, our guest author Igal Lytzki (0xToxin on Twitter) will explore and understand the dynamics occurring when a successful connection is established between the XWorm operating server and a user who has fallen victim to executing this malware.

Throughout this article, Igal will investigate the encryption of the communication between the client and the server, uncover the methods to decrypt it, and identify the potential data and commands the server can transmit to the client.

Let's get started!

What is XWorm malware?

XWorm is a Remote Access Trojan (RAT) malware, specifically targeting Windows operating systems. It provides the operator with an extensive array of so-called "plugins" designed to infect users upon successful connection.

This malware has been active for quite a while now, a fact reflected in ANY.RUN's weekly upload analytics they share on twitter:

XWorm Initial Connection

As previously noted, the purpose of this article is an examination of the occurrences post-establishment of a new connection by the XWorm operating server.

For those seeking a deep understanding of the XWorm code, such as persistence techniques and configuration extraction, read this detailed XWorm technical analysis in ANY.RUN's blog.

The current article is based on this ANY.RUN analysis. Feel free to join us in real-time analysis by filtering with the Process **ID 2932**.

We’ll focus on a process with ID 2932

Analyzing the code of the XWorm payload, a specific class surfaces as the chief handler of the communication process with the server, termed the **ClientSocket** class in our context:

ClientSocket class interface detailing server communication methods in XWorm payload

On its initial run on the victim’s computer, XWorm initiates a connection to a remote server, the details of which are located in the **MalConf** section on the ANY.RUN scan:

XWorm malware configuration in ANY.RUN

The transmitted communication data appears as follows:

Transmitted communication data

Examining the code reveals that it can be splitted into two principal segments:

1. The data length (initial byte sequence up to the 0x00 byte)
2. The encrypted data

Illustrated in the above scenario, the data length stands at **272** (expressed in decimal value).

The encryption employed is AES-ECB (without padding), and the encryption key is the MD5 hash of a configuration variable decrypted during the malware's execution. In our case it's <Guage12>.

Decrypting the data XWorm transmits to the server

Equipped with this, we can hash the key and attempt to decrypt the data transmitted to the server:

A script to decrypt the communication

A script to decrypt the communication can be accessed here.

We can see that the first batch of data sent to the server has several fields. These can be split by using the splitter, found under the **MalConf** section. By comparing this with the malware's code, we can understand what each field represents:

- **ID**: This is the MD5 hash of the following values strung together:
- Processor Count
    - UserName
    - MachineName
    - OS Version
    - Total size of C: drive
- **UserName**: The user's identifier.
- **OS**: Information about the Operating System.
- **Version**: Details about the version.
- **Last Write Time to the Executable**: Shows the last time the executable file was changed.
- **Execution from Persistence**: Shows if it ran from persistence.
- **Admin Status**: Shows if it is running with administrative privileges.
- **Camera Check**: Checks for a camera's presence.
- **CPU Info**: Provides information about the Central Processing Unit.
- **GPU Info**: Provides details about the Graphics Processing Unit.
- **RAM Info**: Provides information on available Random Access Memory.
- **AV's Info**: Provides details about installed antivirus software.

After the necessary information has been sent to the XWorm server and the client is added to the infection panel, the attacker can use a wide range of plugins on the client.

Info Stealer Plugin

Examining the code structure of the XWorm binary, which was injected and operated under RegAsm.exe (PID 2932), reveals that it operates through the invocation of seven main classes during its execution:

- **AlgorithmAES**: Responsible for decrypting data.
- **ClientSocket**: Manages the establishment of connections to the remote server.
- **Helper**: A class populated with numerous functions invoked by other classes.
- **Messages**: Handles data received from the server, including the execution of plugins.
- **Xlogger**: Serves as the keylogger function.
- **Uninstaller**: Facilitates the uninstallation of the binary.

- **Main.**

Each of these classes plays a significant role in the operation and execution of the malware, collectively allowing it to function efficiently and achieve its malicious objectives.

A list of classes. Each plays a significant role in execution of the malware

When examining the traffic between the client and the server, an unusually large packet being transmitted from the server to the client becomes evident:

Snapshot of large data packet transfer in client-server traffic analysis

Upon downloading and decrypting this packet, it is revealed that the received command is to store a plugin. This plugin is stored as a .gz archive which, once uncompressed, unveils an executable:

Decryption of a .gz archive plugin command in malware payload analysis

Properties of a .NET DLL revealing its info-stealing capabilities post-decompression

This executable is a .NET based DLL. A closer inspection of the code discloses that this DLL is an extensive infostealer, endowed with several theft capabilities such as:

- Capturing Credit Card Information
- Harvesting Chromium Cookies
- Acquiring Discord Tokens
- Extracting FileZilla Credentials
- Accessing Browser Data
- Collecting Browser History
- Retrieving WiFi Passwords
- Compromising MetaMask
- Compromising Telegram

and more.

Class list from Recovery DLL showcasing various data extraction functions

Commands Plugin

Continuing our examination of the traffic between the client and the server, we identify another packet of interest(smaller than the previous one but still unusual) that warrants a closer look:

Data packet capture highlighting another notable command plugin in network traffic analysis.

By applying the same decryption and uncompression processes, we uncover another executable:

Decryption output revealing executable code

Details of a .NET DLL with potential to execute various malicious actions

This is another .NET based DLL, which suggests a range of potential actions that the adversary might want to execute:

- Disabling or terminating Windows Defender.
- Excluding a path from Windows Defender scans.
- Installing the .NET framework.
- Blanking the screen.

And more.

Options.dll functions revealing malicious capabilities including screen blanking and system disruption

Reviewing the Threats section in the ANY.RUN analysis, it is noted that numerous malicious activities were detected during the malware's execution, including the identification of XWorm's request commands for **sendPlugin** and **savePlugin**.

Command and control activity with sendPlugin and savePlugin requests in ANY.RUN

IOCs

FileName	SHA256
msbuilds.exe	f58193da4f61b45e375f5aa2978b08908578b5151dc779dc4b566e6a941e802b
Recovery.dll	0ee68c8008e2a8d6252db3d3b1a1b0179e1f868b0b3240bbcec3d1c29d5364fb
Options.dll	7df14d2929a500eec6a144ec8e687960bbea047f9a78a46ea64faa1fa28f8724

MITRE ATT&CK

Tactic	Technique
COLLECTION	Archive Collected Data::Archive via Library T1560.002
CREDENTIAL ACCESS	Credentials from Password Stores::Credentials from Web Browsers T1555.003
DEFENSE EVASION	Deobfuscate/Decode Files or Information T1140
DEFENSE EVASION	Obfuscated Files or Information T1027
DEFENSE EVASION	Reflective Code Loading T1620
DISCOVERY	File and Directory Discovery T1083
DISCOVERY	Process Discovery T1057
DISCOVERY	Query Registry T1012
DISCOVERY	System Information Discovery T1082
DISCOVERY	System Location Discovery T1614
EXECUTION	Shared Modules T1129
EXECUTION	Windows Management Instrumentation T1047

C2

- 140.228.29[.]162:7900

Igal Lytzki

Threat Analyst & team leader. Malware researcher in my spare time.

Igal Lytzki

Threat Analyst

Threat Analyst & team leader. Malware researcher in my spare time.
