AZORult Malware: Technical Analysis

By Mostafa ElSheimy

Published: 2024-09-04 · Archived: 2026-04-06 00:35:26 UTC

Editor's note: The current article is authored by Mostafa ElSheimy, a malware reverse engineer and threat intelligence analyst. You can find Mostafa on X and LinkedIn.

In this malware analysis report, we conduct an in-depth examination of AZORult, a sophisticated credential and payment card information stealer.

Our walk-through covers the malware's evolution, including its transition from Delphi to C++ and the introduction of .bit domain support. We will examine a sample of AZORult to uncover its behavior, evasion techniques, and operational tactics. This analysis aims to enhance understanding of AZORult's functionality and inform effective countermeasures.

Overview

AZORult is a sophisticated credential and payment card information stealer that can also act as a downloader for various malware families. Notably, version 2 introduced support for .bit domains, enhancing its capabilities.

AZORult has been observed operating alongside Chthonic and has been deployed by Ramnit. Originally developed in Delphi, the malware was ported to C++ in 2019, which shows its evolution and increased complexity.

Basic Analysis

Let's begin our analysis of a sample. Here's its key details:

Sample Hash	90a82defe606e51d2826265a43737130682b738241700782d7e41188475b7fb7
Creation Time	2013-12-25 05:01:38 UTC

It's important to note that the creation time has been edited by the author.

The sample was allegedly created on December 25, 2013

First we run the sample in the ANY.RUN sandbox to observe its behavior in a real-time and fully interactive virtual environment.

View the analysis session.

The initial sample analyzed in the ANY.RUN sandbox

The sample initiates two critical processes:

- Executes a PowerShell command
- Drops a file belonging to the Azorult malware family

The PowerShell command launches a script in a hidden window:

```
"powershell.exe" -windowstyle hidden "$Nummmeret=Get-Content 'C:\Users\admin\AppData\Local\Temp\forgrovelse\ko
```

This command performs the following:

- Reads the contents of a file located at C:\Users\admin\AppData\Local\Temp\forgrovelse\konstituerendes\Printermanualens.Ear and stores it in the variable $Nummmeret.
- Extracts a substring from $Nummmeret, starting at index 42833 with a length of 3 characters, and stores this substring in the variable $Trojanerens.
- Attempts to execute the content of $Trojanerens as a command or script, passing $Nummmeret as an argument to this command.

It also drops a file named Declinometer235.exe, the main AZORult payload.

ANY.RUN displays the SHA-256 hash of the malicious payload file

The malware tries to contact thirteen IP addresses and one malicious domain.

ANY.RUN provides IOCs for malware and phishing samples

An analysis of the sample using UnpacMe suggested that it was likely not packed.

The sample has no packer

Let’s see the imports.

AZORult malware actively modifies the Windows registry and attempts to delete data

The malware queries, deletes, and modifies some registry keys, as well as uses an anti-debugging technique.

The certificate is issued by Pretermit Brunbejdsedes

The sample has a digital certificate.

Advanced Analysis

Let’s now open the sample in IDA to take a closer look at its code.

Code of the load_SHGetFolderPathW function

We can see that it loads **SHGetFolderPathW**.

The malware loads SHGetFolderPathW

It gets TEMP path and sets an environment variable containing this path.

GetTempPathW API is used to to retrieve the temporary directory path

It uses **GetTickCount** API to detect if their malware is being debugged.

The malware is equipped with anti-debugging capabilities

Debugging often slows down the execution of a program. By checking the time taken between certain operations, the malware can detect anomalies.

GetTickCount retrieves the current system time in millisecond

If the time taken is unusually long, it might indicate the presence of a debugger.

The malware also creates, writes to, and reads a new file.

CreateFileW function creates or opens a file

WriteFile writes data to a specified file, while ReadFile reads data from a specified file

It returns the value of these functions to Buffer.

The value of the functions is returned to Buffer

It queries the value under the key **HKEY_CURRENT_USER\Control Panel\Desktop\ResourceLocale**.

The malware tries to identify the language ID of the UI

This code attempts to gain shutdown privileges by using **SeShutdownPrivilege** to either disrupt the system by forcing a shutdown or restart, or to ensure changes take effect after a restart.

The malware uses SeShutdownPrivilege to reboot the system

The function interacts with the clipboard, which could be used to steal or manipulate data.

The malware manipulates the clipboard

After looking at the strings section, we found the following:

AZORult uses several system functions

off_40940C contains these strings in .data section:

```
 "GetDiskFreeSpaceExW"

"MoveFileExW"

"RegDeleteKeyExW"

"OpenProcessToken"

"LookupPrivilegeValueW"

"AdjustTokenPrivileges"

"GetUserDefaultUILanguage"
```

```
"SHAutoComplete"

"SHFOLDER"

"SHGetFolderPathW"
```

Let’s see the xrefs of **off_40940C**.

GetProcAddress is used to resolve the APIs

It uses **LoadLibraryA and GetProcAddress** to resolve these APIs.

The malware uses **GetDiskFreeSpaceExW** to check if there is enough disk space available before attempting to install or execute.

If the disk is nearly full, the malware might avoid installation to prevent detection or impact.

LookupPrivilegeValueW/ AdjustTokenPrivileges

Malware uses **LookupPrivilegeValueW** to get the LUID for a privilege like SE_DEBUG_NAME or SE_SYSTEM_ENVIRONMENT_NAME, which allow it to perform actions like debugging other processes or modifying system settings.

It uses **AdjustTokenPrivileges** to:

- Modify Privileges: By adjusting token privileges, malware can avoid detection by security software or make modifications to the system that are not typically allowed under normal user privileges.
- Access Sensitive Operations: Malware might need elevated privileges to modify system settings, access protected files, or inject code into other processes.

GetUserDefaultUILanguage

This API provides the language used for the user interface of Windows.

It is used to tailor the malware's behavior or appearance based on the language of the system to avoid detection or appear more localized.

Conclusion

The AZORult malware represents a highly adaptable and sophisticated threat, evolving significantly since its initial development. As observed, AZORult employs various techniques to evade detection and maximize its impact, such as anti-debugging measures, use of environment variables, and privilege escalation.

The malware's ability to operate in hidden modes, drop additional malicious files, and interact with multiple IP addresses and domains underscores its potential for widespread damage.

The use of specific Windows API calls for tasks like checking disk space, adjusting token privileges, and manipulating system settings reflects a well-designed strategy to ensure persistence and effectiveness. The presence of digital certificates and obfuscation techniques further complicates detection and analysis.

About ANY.RUN

ANY.RUN helps more than 400,000 cybersecurity professionals worldwide. Our interactive sandbox simplifies malware analysis of threats that target both Windows and Linux systems. Our threat intelligence products, TI Lookup, YARA Search and Feeds, help you find IOCs or files to learn more about the threats and respond to incidents faster.

With ANY.RUN you can:

- Detect malware in seconds.
- Interact with samples in real time.

- Save time and money on sandbox setup and maintenance
- Record and study all aspects of malware behavior.
- Collaborate with your team
- Scale as you need.

Request free trial →

IOCs

MD5 Hash

0824428fdccf3c63fc1ca19a1dd7ef74

DNS requests

ehzwq[.]shop	fp-afd-nocache-ccp.azureedge[.]net
r10.o.lencr[.]org	a-ring-fallback[.]msedge[.]net
t-ring-fdv2[.]msedge[.]net	reap.skyestates[.]com[.]mt

IP connections

108.167.181.251	20.166.126.56	52.168.117.175	20.223.35.26
2.23.209.130	2.23.209.158	2.23.209.140	13.107.246.45
131.253.33.254	20.99.185.48	2.23.209.140	13.107.246.45
131.253.33.254	20.99.185.48		

Registry keys

HKEY_USERS\S-1-5-21-575823232-3065301323-1442773979-1000\fordjelsesbesvret\Uninstall\Spidsfindigeres22\luftrr

HKEY_CURRENT_USER\fordjelsesbesvret\Uninstall\Spidsfindigeres22\luftrr

HKEY_CURRENT_USER\fordjelsesbesvret\Uninstall\Spidsfindigeres22luftrr Spidsfindigeres22\luftrr

fordjelsesbesvret\Uninstall\Spidsfindigeres22\luftrr

HKEY_CURRENT_USER\fordjelsesbesvret

HKEY_CURRENT_USER\fordjelsesbesvret\Uninstall

HKEY_CURRENT_USER\fordjelsesbesvret\Uninstall\Spidsfindigeres22

Mutexes

Global\6b9d2ecb-1948-49c6-b61f-9cc3ad1d78d1
Global\AmiProviderMutex_InventoryApplicationFile
Global\OneSettingQueryMutex+compat+encapsulation
Local\WERReportingForProcess1284

MITRE ATT&CK TTPs

TACTIC	TECHNIQUE	MITRE ATT&CK ID
Execution	Windows Management Instrumentation	T1047
	Command and Scripting Interpreter	T1059
	PowerShell	T1059.001
	Scripting	T1064 (deprecated)
	Native API	T1106
	Shared Modules	T1129
Persistence	Boot or Logon Autostart Execution	T1547
	Shortcut Modification	T1547.009
	Hijack Execution Flow	T1574
	DLL Side-Loading	T1574.002
Privilege Escalation	Process Injection	T1055
	Boot or Logon Autostart Execution	T1547
	Shortcut Modification	T1547.009
	Hijack Execution Flow	T1574
	DLL Side-Loading	T1574.002
Defense Evasion	Obfuscated Files or Information	T1027
	Software Packing	T1027.002
	Embedded Payloads	T1027.009

TACTIC	TECHNIQUE	MITRE ATT&CK ID
	Masquerading	T1036
	Process Injection	T1055
	Scripting	T1064 (deprecated)
	Indicator Removal	T1070
	Timestomp	T1070.006
	Modify Registry	T1112
	Deobfuscate/Decode Files or Information	T1140
	File and Directory Permissions Modification	T1222
	Virtualization/Sandbox Evasion	T1497
	Hide Artifacts	T1564
	Hidden Window	T1564.003
	Hijack Execution Flow	T1574
	DLL Side-Loading	T1574.002
Credential Access	OS Credential Dumping	T1003
	Unsecured Credentials	T1552
	Credentials In Files	T1552.001
	Credentials in Registry	T1552.002
Discovery	Application Window Discovery	T1010
	Query Registry	T1012
	Remote System Discovery	T1018
	Process Discovery	T1057
	System Information Discovery	T1082
	File and Directory Discovery	T1083
	Virtualization/Sandbox Evasion	T1497
	Software Discovery	T1518
	Security Software Discovery	T1518.001

TACTIC	TECHNIQUE	MITRE ATT&CK ID
Collection	Data from Local System	T1005
	Email Collection	T1114
	Clipboard Data	T1115
	Video Capture	T1125
	Application Layer Protocol	T1071
	Non-Application Layer Protocol	T1095
	Encrypted Channel	T1573
Impact	System Shutdown/Reboot	T1529
	System Shutdown/Reboot	T1529

Mostafa ElSheimy

Mostafa ElSheimy is a malware reverse engineer and threat intelligence analyst, specializing in analyzing TTPs (Tactics, Techniques, and Procedures) and crafting YARA rules to detect and counter cyber threats. Mostafa's work focuses on dissecting malware to uncover hidden dangers and protect organizations from emerging threats.

Find him on X and LinkedIn.

Mostafa ElSheimy

Malware Analyst

Mostafa ElSheimy is a malware reverse engineer and threat intelligence analyst, specializing in analyzing TTPs (Tactics, Techniques, and Procedures) and crafting YARA rules to detect and counter cyber threats. Mostafa's work focuses on dissecting malware to uncover hidden dangers and protect organizations from emerging threats.

Find him on X and LinkedIn.
