TMPN (Skuld) Stealer: The dark side of open source

Archived: 2026-04-05 14:42:09 UTC

Summary

- TMPN Stealer is based on the open-source project ‘Skuld stealer’
- Uses Discord webhooks for communications
- Injects JS payload to Discord
- Steals browsers and cryptocurrency wallet data
- Steals local files and system information

Introduction

Skuld, also known as TMPN Stealer, is an information-stealing malware written in Golang (Go) that emerged in May 2023. Its developer, identified as "Deathined," appears to be a newcomer in the malware development scene, utilizing open-source projects as inspiration for Skuld's functionality. The malware is distributed through various means, including malicious links and compromised websites, aiming to infect systems globally.

Technical details

Overview

The analyzed sample is written in Go and has a fake compilation timestamp. The file size is 14.4 MB and is normal for programs that are written in Go.

GitHub contains a page that probably contained TMPN stealer source code, but it is no longer available: https://github.com/DextenXD/TMPN-Stealer

While this page is unavailable, during analysis, we discovered a link in the code that leads to another project: Skuld Stealer. It is an open-source project that demonstrates a Discord-oriented stealer. Even the description is similar to the TMPN one:

Configuration

At the start of execution, Skuld loads configuration to the memory, which contains multiple strings. The first one is ‘webhook’, and the second one contains the URL:

hxxps://discord.com/api/webhooks/1272963856322527274/PGGfe9V7To17wrSy0T7qE8EpNjFXfms2KY4A421gXmXwMcrPdaeG0Z3DB2

This refers to Discord webhooks, a low-effort way to post messages to channels in Discord. This mechanism is very useful to organize control, due to webhooks not requiring a bot user or authentication to use.

The next string that is loaded to the memory points to the BTC wallet:

In fact, the source code supports multiple cryptocurrency wallets:

Preparation

Before executing the main payload, Skuld makes some preparations:

First, Skuld checks if it is already running by checking the specific mutex name:

Next, it calls the UAC Bypass function. Here it checks if the process is elevated or whether it can it be elevated, and then finally calls the elevation function. This function sets the malware path to the fodhelper.exe utility registry key:

HKCU\\Software\\Classes\\ms-settings\\shell\\open\\command\\DelegateExecute

This will cause a fodhelper.exe to pop the UAC window, but since the registry key has been changed, upon accepting notification, it will execute a sample with elevated permissions. The old process then will clear the *Fodhelper* registry key and terminate.

To hide itself from user eyes, Skuld uses an '*attrib +h +s*' command, which will give sample '*hidden*' and '*system*' attributes. Additionally, it uses the 'GetConsoleWindow' function to obtain the current window descriptor and the 'ShowWindow' function to set the window show state to 'Hidden'.

It adds a registry key to 'Software\Microsoft\Windows\CurrentVersion\Run' to persist on the victim's systems. The key name will be 'Realtek HD Audio Universal Service' and have the next value: '%APPDATA%\Microsoft\Protect\SecurityHealthSystray.exe'. After this, it will check for file existence in this path. If it does not exist, it will copy itself there.

Next, Skuld checks if the sample is running on a Virtual Machine. To do this it checks the hostname, username, MAC address, IP address and HWID, and compares it with its own saved lists. If any string matches, it will terminate execution. Next, it checks if any debugger is attached to the process.

Finally, it tries to evade antivirus software. Here, it excludes its own path from Microsoft Windows Defender scanning and uses PowerShell commands to disable it:

powershell", "Set-MpPreference", "-DisableIntrusionPreventionSystem", "$true", "-DisableIOAVProtection", "$true", "-DisableRealtimeMonitoring", "$true", "-DisableScriptScanning", "$true", "-EnableControlledFolderAccess", "Disabled", "-EnableNetworkProtection", "AuditMode", "-Force", "-MAPSReporting", "Disabled", "-SubmitSamplesConsent", "NeverSend"

powershell", "Set-MpPreference", "-SubmitSamplesConsent", "2

"%s\\Windows Defender\\MpCmdRun.exe", os.Getenv("ProgramFiles")), "-RemoveDefinitions", "-All

The last step of this function is to block connection to the following sites:

"virustotal.com", "avast.com", "totalav.com", "scanguard.com", "totaladblock.com", "pcprotect.com", "mcafee.com", "bitdefender.com", "us.norton.com", "avg.com", "malwarebytes.com", "pandasecurity.com", "avira.com", "norton.com", "eset.com", "zillya.com", "kaspersky.com", "usa.kaspersky.com", "sophos.com", "home.sophos.com", "adaware.com", "bullguard.com", "clamav.net", "drweb.com", "emsisoft.com", "f-secure.com", "zonealarm.com", "trendmicro.com", "ccleaner.com"

Discord injection

Upon entering the injection function, Skuld first calls two bypass functions. The first is used to bypass BetterDiscord, which is a Discord client modification. Besides different plugins, emotes and developer tools, it also contains security enhancement. To perform a bypass, it searches and opens a '*AppData\Roaming\BetterDiscord\data\betterdiscord.asa*r' file and replaces all existing '*api/webhooks*' strings with '*ByHackirby*'.

It then tries to bypass the Discord Token Protector. First, it checks if this module is presented on the system. If yes, it opens a "*AppData\Roaming\DiscordTokenProtector\config.json*" file and changes the next values:

"auto_start" = false

"auto_start_discord" = false

"integrity" = false

"integrity_allowbetterdiscord" = false

"integrity_checkexecutable" = false

"integrity_checkhash" = false

"integrity_checkmodule" = false

"integrity_checkscripts" = false

"integrity_checkresource" = false

"integrity_redownloadhashes" = false

"iterations_iv" = 364

"iterations_key" = 457

"version" = 69420

It then calls an injection function. First, it checks the presence of Discord on the PC, loading multiplied data blocks and taking some data from them. It then joins all taken data and passes it to the '*filepath.Glob*' function as a search filter.

Next, it downloads the file 'injection.js' from Skuld Github. The file name will be changed to 'coreinedex.js' and written to the previously found Discord path. This script will set up hooks and intercept such data as login, register and 2FA requests, PayPal credits and email / password changes. When any info is captured, it will send the result in JSON format to the server.

Cryptocurrency wallets injection

The injection targets two cryptocurrency wallets: Exodus and Atomic. The point of this technique is to download and save files with the '*.asar*' extension, which is an archive that is used by cryptocurrency wallets, but contains attacker data.

System information discovery

After both injections are done, Skuld starts obtaining system information. Here, it calls a number of functions to obtain such information as CPU, disks, GPU, Network, OS, Windows license keys, RAM and others.

All this data will be saved in JSON format and sent to the server. Besides raw data, it appends a link to the picture to the 'avatar_url' field. This avatar is probably used in the attacker Discord bot.

Browsers

The browser's function targets two types of browsers and depends on their engine — Chromium or Gecko based. Both of these functions have different saved browser names and paths. Functions that extract data, such as logins, cookies, credit cards, downloads and history, are the same for all browsers.

Discord tokens

To get Discord tokens, Skuld again checks browsers and searches particular strings in their databases. Then, results are passed to the function that will interact with the Discord API and check if the found tokens are valid. All results are saved in JSON format and sent to the server.

Discord 2FA codes

Discord has a mechanism that is used in case a user loses access to the 2FA device. The file 'discord_backup_codes.txt' contains codes that can be used in such situations.

Sample will search this file in the next user folders: *Desktop, Downloads, Documents, Videos, Pictures, Music, OneDrive.*

Common files

This function will search for files with particular keywords in their names and extensions in the next folders: *Desktop, Downloads, Documents, Videos, Pictures, Music, OneDrive.* If both the keyword and extension are presented in the filename, the file will be copied to the new folder.

This folder will then be archived with a password, which contains 16 random symbols. The archive will be uploaded to the server alongside JSON data, which contains the archive server URL, password and archived file number. Here we can note that the server link that was provided in the 'Upload' function matches the link from the source code: 'https://api.gofile.io/getServer '. This link is invalid, meaning that this sample will collect data but not will not send it to the attacker.

Cryptocurrency wallets

Here it has two different functions for this purpose. The first will search for cryptocurrency wallet files on the local system in the '%APPDATA%\\Roaming' folder. All found files will be zipped to archive and sent to the server. The second will check for cryptocurrency wallet browser extensions and try to steal their profiles.

Game session stealer

To steal game data, Skuld loads a list that contains six names: *"Epic Games", "Minecraft", "Riot Games", "Uplay", "NationsGlory"* and *"Steam"*. Each field contains additional values, which include file paths and filenames that must be searched for. It will try to copy all files that are presented in this list to the temporary folder and exfiltrate them to the server as a zip archive.

Clipper

Finally, it starts a clipper function. It uses multiple regex values to filter clipboard data. This regex targets different cryptocurrency wallet addresses, and if there is a match, it will replace this data with its own cryptocurrency wallet address. The only attacker address that was spotted in the configuration is the BTC address.

Conclusion

Compiled from open-source project Skuld, TMPN stealer has a BTC wallet address and Discord API link in its configuration. There is no additional information added to the source code, such as a server to upload files, meaning that this sample primarily targets Discord, injecting a JS payload to retrieve information such as emails, passwords and tokens.

Discord is a popular target not only because it is the most popular communication platform for gamers, but also because it has popularity in the cryptocurrency community. It can even bypass some Discord plugins that are designed to enhance security. It can also steal browser data and user files, which may contain any credentials, and upload them to the attacker.

IoCs

Files

loader.exe

5a7e38a45533e0477c3868c49df16d307a3da80b97a27ac4261619ff31a219f8

Network indicators

https://raw.githubusercontent.com/hackirby/discord-injection/main/injection.js

https://discord.com/api/webhooks/1272963856322527274/PGGfe9V7To17wrSy0T7qE8EpNjFXfms2KY4A421gXmXwMcrPdaeG0Z3Dl

https://i.redd.it/68p07sk4976z.jpg

https://api.gofile.io/getServer
