# CoinLurker: The Stealer Powering the Next Generation of Fake Updates

By Nadav Lorber

Archived: 2026-04-06 00:54:17 UTC

The evolution of fake update campaigns has advanced significantly with the emergence of CoinLurker, a sophisticated stealer designed to exfiltrate data while evading detection. Written in Go, CoinLurker employs cutting-edge obfuscation and anti-analysis techniques, making it a highly effective tool in modern cyberattacks.

## Introduction

Building on the deceptive strategies of SocGolish, ClearFake, ClickFix and FakeCAPTCHA, attackers now combine highly convincing fake update prompts with stealthy payloads like CoinLurker. These campaigns leverage innovative methods, such as **EtherHiding** and in-memory execution, to bypass traditional security defenses and obscure the malware's origin.

In this blog, we examine the evolution of fake update campaigns, the techniques enabling CoinLurker's success, and actionable strategies for defending against this next-generation threat.

## Delivery Tactics and Techniques

Fake update campaigns initiate infections through various deceptive entry points that exploit user trust in common actions like:

### Fake Software Update Notifications

Malicious websites prompt users to download fake updates, disguised as essential software patches. This vector is often observed on compromised WordPress sites, where attackers exploit vulnerabilities to deliver fake update prompts.

### Malvertising Redirects

Compromised ads on legitimate sites redirect users to malicious pages, prompting fake updates or CAPTCHA verifications.

### Phishing Emails

Emails link to spoofed update or CAPTCHA pages, tricking users into downloading malware disguised as security updates.

### Fake CAPTCHA Prompts

FakeCAPTCHA introduces malicious CAPTCHA prompts that deliver malware instead of verifying users.

### Direct Downloads from Fake or Compromised Sites

Malicious actors host fake updates on compromised or deceptive download sites, luring users into installing malware.

### Social Media and Messaging Links

Links shared on social platforms lead to malicious sites disguised as update or verification pages.

Each of these vectors effectively disguises malware as routine actions, initiating the infection chain with minimal user suspicion.

## Leveraging Microsoft Edge Webview2 as a Stager

**Microsoft Edge Webview2** is utilized by the stager to execute the malware, presenting a **GUI** that mimics legitimate browser update tools. Any interaction with the GUI—clicking buttons or even closing the window—triggers the payload execution.

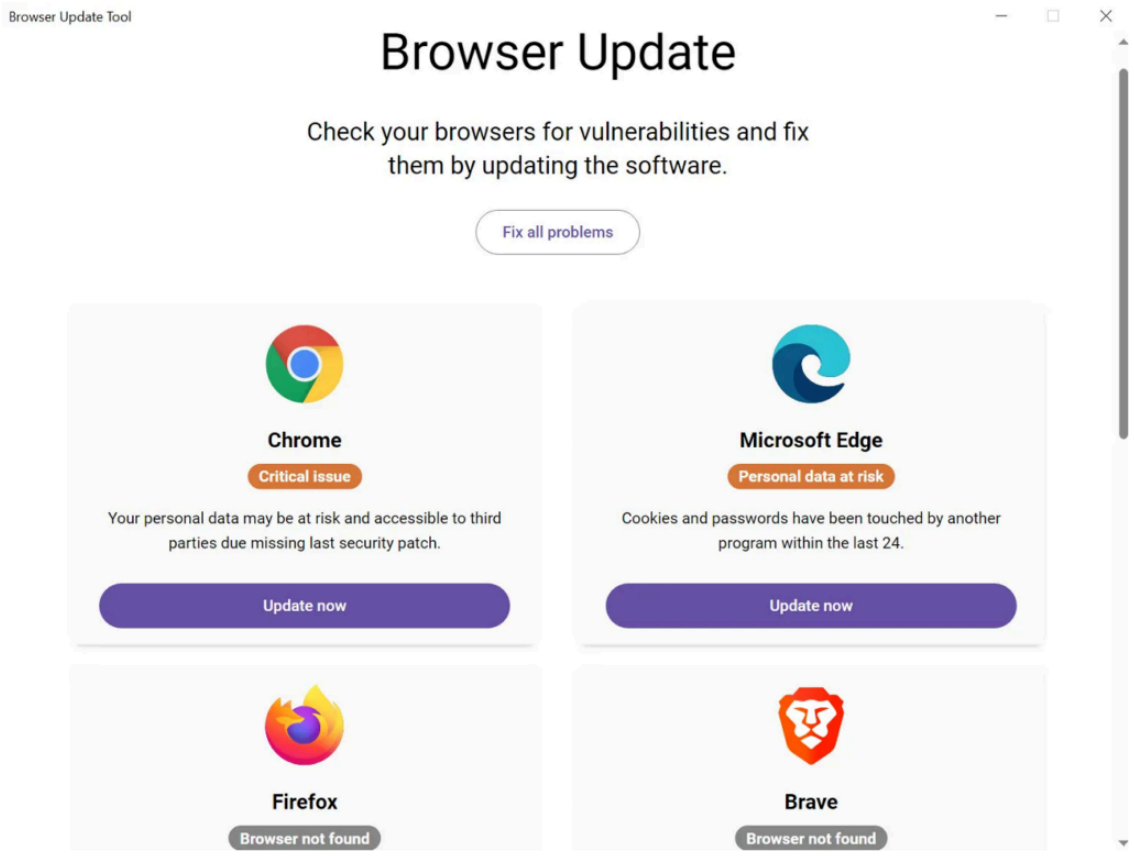



*Figure 1: Fake Browser Update Webview2 GUI*

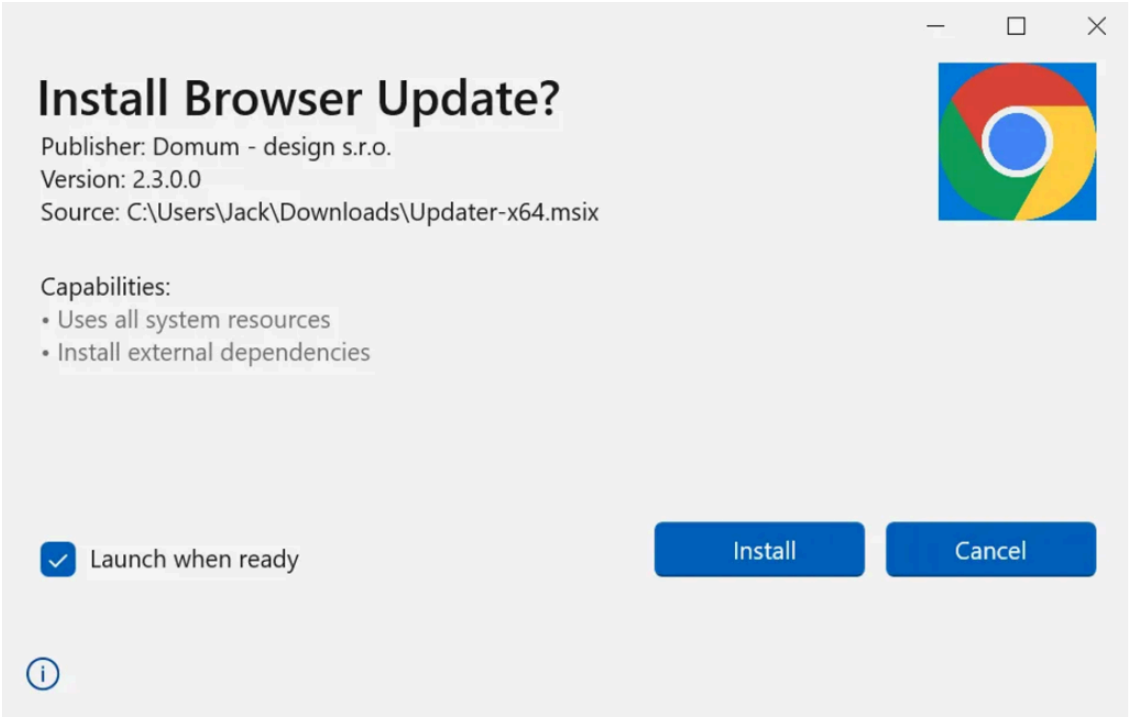



*Figure 2: Chrome fake update Webview2 GUI*

Webview2's dependency on pre-installed components and user interaction complicates dynamic and sandbox analysis. Sandboxes often lack Webview2 or fail to replicate user actions, allowing the malware to evade automated detection.

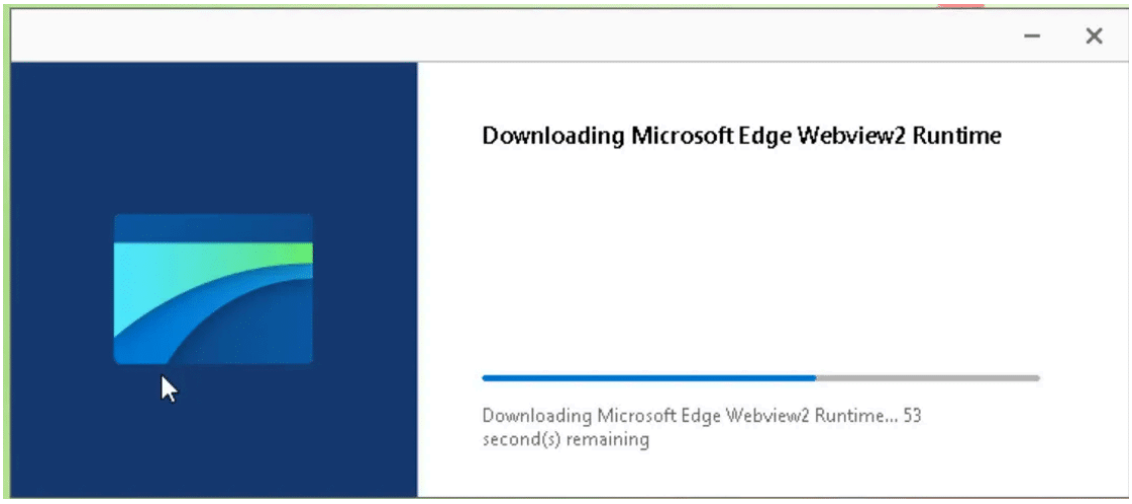



*Figure 3: Screenshot of Webview2 installation within Sandbox*

## The Obfuscation Chain: Smart Contracts to Trusted Platforms

**Binance Smart Contract → Actor-controlled C2 → Bitbucket Repository**

Fake update campaigns like those deploying CoinLurker have adopted advanced techniques to evade detection, including **EtherHiding**, which leverages Web3 infrastructure to conceal malicious payloads. This campaign employs a multi-stage chain to deliver its payload seamlessly while remaining under the radar.

1. **Binance Smart Contract**:
   This process begins with encoded data embedded within a Binance Smart Contract. By leveraging the decentralized and immutable properties of blockchain, attackers store payload instructions that are resistant to tampering or removal.
2. **Actor-controlled Command-and-Control (C2) Server**:
   The encoded data directs the malware to an actor-controlled C2 server, which serves as a pivot point in the chain. Here, the server dynamically fetches further instructions or payload links, ensuring the malware does not carry any static indicators that could trigger detection.
3. **Bitbucket Repository**
   The final stage involves a Bitbucket repository that initially hosts a benign executable. Once downloaded and deemed safe by security scans, this executable is later replaced by a malicious version. This tactic capitalizes on Bitbucket's reputation as a trusted platform while reducing the chances of immediate detection. The use of a clean file in the initial stage ensures the campaign avoids raising alarms during early stages of distribution.

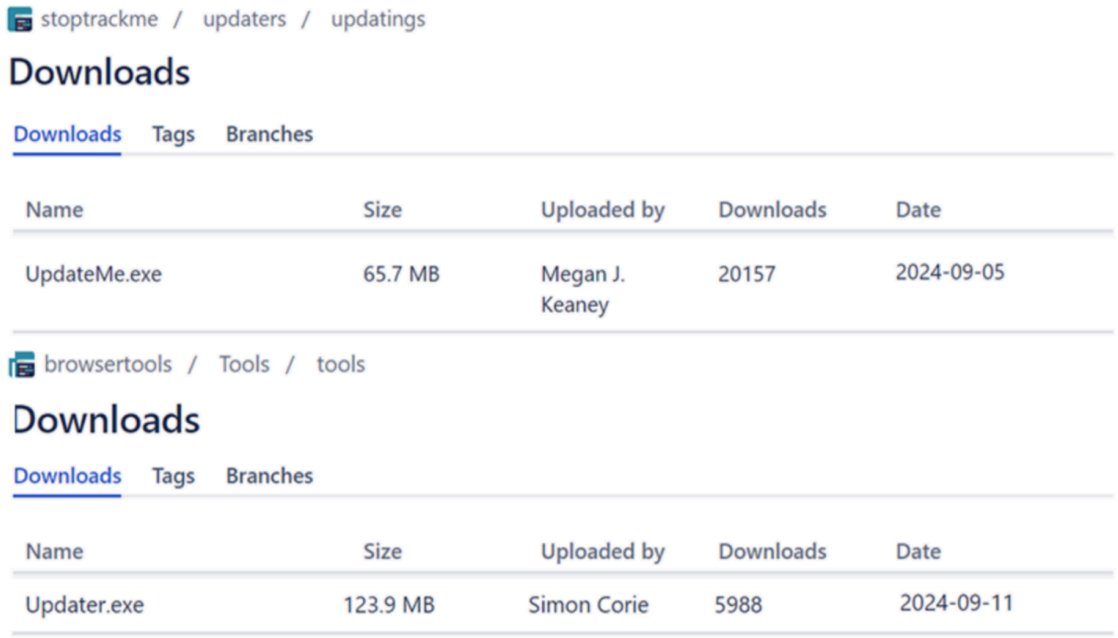



*Figure 4: Screenshot of repositories used by the actor with high downloads count*

## Timeline of Filenames (August to October 2024)

CoinLurker's evolution includes a notable timeline of filenames used in the Bitbucket repository, often masquerading as legitimate tools to enhance deception. From August to October 2024, the filenames observed include:

| | | |
|---|---|---|
| `BrowserUpdateTool.exe` | `BrowserTool.exe` | `BrowserUpdater.exe` |
| `UpdateNow.exe` | `UpdateMe.exe` | `Updater.exe` |
| `UpdaterSetup.exe` | `Updating.exe` | `SecurityPatch.exe` |

Each filename aligns with the fake update theme, designed to appear as genuine system utilities or browser update tools. Additionally, those executables are signed with a legitimate **Extended Validation (EV) certificate**, adding another layer of credibility. While the origin of the certificate cannot be confirmed, it is likely stolen, enabling the attackers to bypass security warnings and enhance the perceived legitimacy of the malicious files.

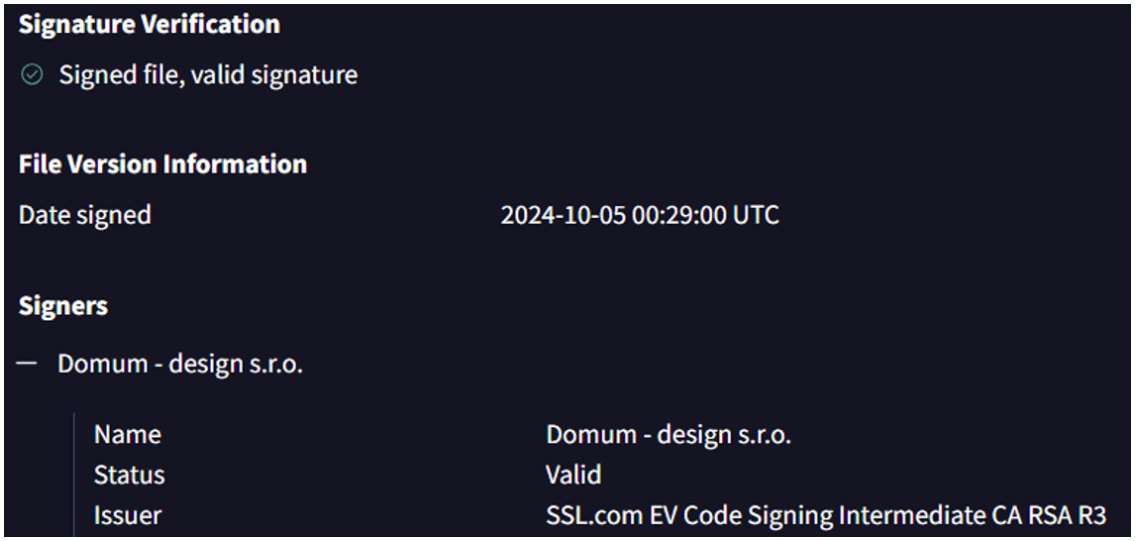



*Figure 5: EV Certificate parsed in VirusTotal*

## Layered Injection Tactics to Evade Detection

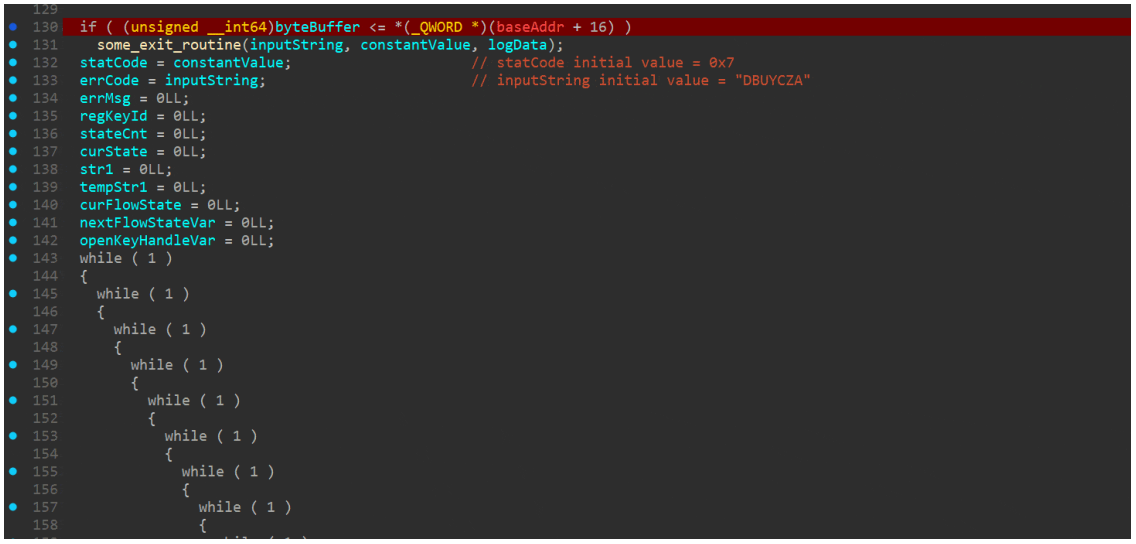
CoinLurker utilizes a sophisticated multi-layered injector to stealthily deploy malicious payloads into multiple instances of legitimate msedge.exe processes. This approach ensures that the malware evades detection, blends seamlessly into legitimate system activity, and bypasses network security rules that rely on process behavior for filtering. Below are the key obfuscation techniques observed during analysis.

## Infection Validation Through Registry Checks

CoinLurker employs a heavily obfuscated function to determine if the system has already been infected. This method dynamically constructs a unique registry key, such as `SOFTWARE\<GUID>-<ID>`, using system-specific data like the machine's GUID and custom input strings.

The malware then attempts to access the key using the Windows *OpenKey* API. If the key exists and contains the expected values, CoinLurker identifies the system as already infected and terminates its execution. If the key is missing or does not match the expected values, the malware proceeds with its infection routine.

While this technique serves as a mutex to prevent multiple infections, the obfuscation within the function—such as dynamic API resolution and a layered execution flow—makes it challenging for analysts to reverse-engineer the logic or identify the key construction process.



```
129
130 if ( (unsigned __int64)byteBuffer <= *(_QWORD *)(baseAddr + 16) )
131   some_exit_routine(inputString, constantValue, logData);
132 statCode = constantValue;                    // statCode initial value = 0x7
133 errCode = inputString;                       // inputString initial value = "DBUYCZA"
134 errMsg = 0LL;
135 regKeyId = 0LL;
136 stateCnt = 0LL;
137 curState = 0LL;
138 str1 = 0LL;
139 tempStr1 = 0LL;
140 curFlowState = 0LL;
141 nextFlowStateVar = 0LL;
142 openKeyHandleVar = 0LL;
143 while ( 1 )
144 {
145   while ( 1 )
146   {
147     while ( 1 )
148     {
149       while ( 1 )
150       {
151         while ( 1 )
152         {
153           while ( 1 )
154           {
155             while ( 1 )
156             {
157               while ( 1 )
158               {
```

*Figure 6: .gif – Runtime Validation Obfuscated Function*

## Runtime String Decoding and Injection

CoinLurker employs a sophisticated injection process that relies on dynamic string decoding and obfuscation to conceal its activities. The malware targets msedge.exe, launching each instance with unique, obfuscated command-line arguments. Examples include:

- WSCOGJJEZZWL
- NTOCBJPKZPNT
- XXEZGQVPKJGS
- PEQDTHUEORHX
- RLZXCUVFFESG

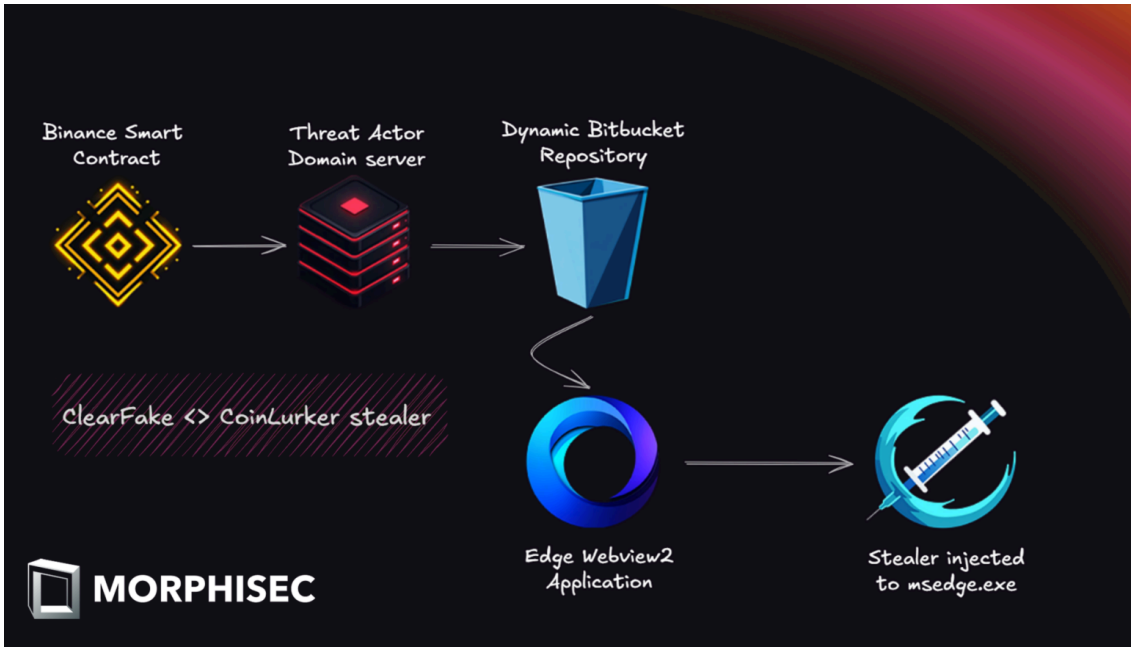
These arguments are dynamically generated and transformed at runtime, passing through layered transformations like Base64 decoding, UTF-16 conversion, and dynamic resource mapping. The final values only emerge during execution, leaving minimal static traces. The payload itself is decrypted in memory using obfuscated routines,

ensuring traditional detection methods are bypassed.

```
memoryManipPtr = transform_and_map_strings(
                   (__int64)&unk_7FF7CB778895,
                   28LL,
                   0LL,
                   0,
                   param5,
                   memoryParam21,
                   memoryParam22,
                   memoryParam23,
                   memoryParam24);          // in debug: v25 is pointer to 'DBUYCZA..
CreateRegistryKey(memoryManipPtr, 28LL, memoryTemp26);// v25 ptr DBU..
registryKeyPtr = seed2;
outputStrPtr = (char **)dataCap;
functionResult = (uint8_0 *)RC4EncryptWithGoSlice(
                              ptr_a1098n7326r54,
                              seed2,        // 0xC
                              data,
                              dataLen,      // 0x555600
                              dataCap,      // 0x555600
                              encryptionTemp33,
                              encryptionTemp34,
                              encryptionTemp35,
                              encryptionTemp36);
if ( registryKeyPtr )
{
  rc4EncryptionResult = functionResult;
  responseSize = encryptionSize;
  processStrMapPtr = processStringMap(
                       (__int64)&unk_7FF7CB7EA78B,
                       140LL,
                       0LL,
                       0,
                       outputStrPtr,
                       processTemp38,
                       processTemp39,
                       processTemp40,
                       processTemp41);      // in debug: v42 is pointer to '23697dd0-2018-4014-b1e0-9154710f0816'
  rc4EncryptedDataPtr = RC4EncryptWithGoSlice(
                          (int64_t)processStrMapPtr,
                          140LL,
                          rc4EncryptionResult,
                          registryKeyPtr,
                          responseSize,
                          processTemp43,
                          processTemp44,
                          processTemp45,
                          processTemp46);   // in debug: v47 is pointer (probably to PE EXE) to 'MZ....
  CreateProcessObfuscated_wrap(
    rc4EncryptedDataPtr,
    140LL,
    createProcTemp48,
    registryKeyPtr,
    responseSize,
    createProcTemp49,
```

*Figure 7: Main Loader Function*

The injection logic incorporates heavily obfuscated control flow, including nested state machines and conditional checks that obscure the actual execution path. Redundant resource assignments and iterative memory manipulations further complicate analysis, keeping critical data hidden until runtime.




## Socket-Based Communication for C2 Operations

CoinLurker communicates with its C2 servers using a socket-based framework. It employs functions like GetAddrInfoW for DNS resolution, WSASocketW for socket creation, and ConnectEx for establishing connections. Data exchange is handled via WSASend and WSARecv, with asynchronous operations using CreateIoCompletionPort to enhance efficiency.

Domains dynamically resolved by CoinLurker include:

- zovik[.]info
- analfucker[.]lol
- paveldurov[.]sbs

## File Enumeration Targeting Cryptocurrency Wallets

CoinLurker demonstrates a highly targeted approach to data collection, focusing on directories associated with cryptocurrency wallets and financial applications. Through systematic enumeration, it attempts to access a variety of locations that are commonly used for storing sensitive user data.

### Key targets include:

**Major Cryptocurrency Wallets:**

- Bitcoin\wallets
- Ethereum\keystore
- Ledger Live\Local Storage\leveldb
- Exodus\exodus.wallet

**Alternative Cryptocurrencies and Lesser-Known Wallets:**

- Examples include BBQCoin, Lucky7Coin, MemoryCoin, and many others, showcasing its effort to cover a wide range of cryptocurrencies.

**Related Applications:**

- Directories such as Telegram Desktop\tdata, Discord\Local Storage\leveldb, and FileZilla

This comprehensive scanning underscores CoinLurker's primary goal of harvesting valuable cryptocurrency-related data and user credentials. Its targeting of both mainstream and obscure wallets demonstrates its versatility and adaptability, making it a significant threat to users in the cryptocurrency ecosystem.

## How Morphisec Can Help

Morphisec's pioneering Automated Moving Target Defense (AMTD) technology stops sophisticated attacks at the earliest stage without relying on outdated signature or behavioral-based detection methods. By preemptively blocking memory and application-based attacks, Morphisec eliminates threats before they can take hold and become business impacting.

Schedule a demo today to see how Morphisec stops fake update campaigns like CoinLurker and other new and emerging threats.



## IOCs

### Fake Installers SHA256

- 324e1bf24f13d5a8f45cc5ee25d3dfe330a7e755b19901549976f2db02ca4fa4
- c8adb9bf6997a9fa2738a09600a60abc4fb6334aa54b24166cf042afdc5a1064
- 1f4624c44288f77327ec2e8d260399559b81c7cae442c31311736c2a2ec5f399
- a7eca930c2aa851cae3475cb4f5d599058816d51e1cc55a82ae976a030794aac
- be5e250168d37e7a9a4999d41a77cde19a6ac376a391f602b3496ace307ad0e8
- 93cc9759d86f8b087b71583f577a5534e975ce9ac19ec3ec140efa6bbfad6bd0
- 44521e1af289aa3473d7445d097766f1c3f3d8721d14b14ed6d5404994a03eb2
- 2198912e1a1f4a5b5f0dfe237b75d264c9be0b5b6f98f83a999117dd194e842c
- f79c62b820420bda78252197db842eabe63261a4e80fbdcec8d671ce3d0a43ef
- 8119a59487c6ffe5382c03e3de8c70b2c2e26899b51dcc4794066a8e1f358bcb
- 9a036f20d758107d9434bd3bed682ff7d81393dc9d49fd6fe70d4b549045eaa2

- a12809c76461d00760bef767c98baf5909a4aed48f2256d3c42eb1ca62835c14
- 487156ae20cc6d8e7d922cebe35b197c28ae43134f7e04c5f6bd0f3e164a7120
- 9116c7878f51e6d8173d41a5a0e63ca16105dac954afedeaf1d5e06594cc4d41
- cc2f65faf61154815b4fa151d9a27c01a160d7d46398c7e44169949a61c63c2b
- 7eede0e13ed9990afb465c2f612d85bc10c946dd2419323528a58707cef62899
- 2c8f611b0f2c157f010c20379d4fcd725a8c462a8d226ae0095e3e0fb110ddbe
- 6976c3e0ffbbbbb310995e70f24bf9501d017279d865ac4536aee25b316a92de
- 269c3b26b215d397f012a20e241c54b2c693667d4f64243ebf8dba1a5872c02d
- 397a0f6515a81f307b5289ff3e939a0e01a6c1a0f0515be9844ddc9c6031ad97
- 82cc0f3f4aa70a8215b62db7ee9deac1c3d4dd27cde25cf56ec2f82ca7d146a9
- 2181c60e8727d5cfe7e713aa9731018168660ad2c96f31b08a729d1503dfc19a
- 0b5fe211d558daa7d54207d2869f53d0a91ae16397343fd2605fd3a0f292dd21
- 9c0c9945f81977269542f941c10fa28dbefe91078b6df68e97d61b58318cac9a
- b761e91e77b67661db51d6b498ea39ccb6f143e51eeee18925a2dc4aab20adfa
- a612bca9b5cbda864f4b808992de3d616c67b9120d8b24cbfa8a836ccdde9142
- a3c7b289054635f5239d453fb4be718298037ea6c1f4bf16954af1e9da2a53e2
- 9ea70e081c13c4b0e30b43dd68a6a0e0cfb6926c990bbe8ddedd8d9693c953d6
- 0b420a565e5e6f6899ebcb1da2fc162b05f5a8b7bfe0f56f52a085f17abb253d
- 80b2950f1249d439105eac421660ddd15caab6de6afce3511f945deef1c0dd21
- c643c087c68e51dfe422ddb48614675ab8e6aaecbe5704759c9978ac22b15f83
- 3048030c0e3ff5e6e45bbb37e75d6e55fde8d77a928958dc34497177e077b69a
- 18f882b6c16641be3899f4e5123d10bb5c448ac7b7dafe7adb6144176acae304
- 15be79b09fa5efe3ca3440a94e436124d97232436af91f64917b7095b559a210
- 162e4277a4cb2e3703df74529d83d47b66a5b46b0a93b3ac902b56da3e588fe9
- 8d61f5b56f05daeef394dbc434abb96c1388aca8406e02445a72db1a65b9da3d
- 9374e1561a87a23b12ec586859661241b2eb5da822c0b4b874cdf9eda480363f
- Fff7637514c6238443100fbc4d1fef626cebf043eef1aefa3a0f5ab6d0103bf6

## Stager URLs

- md928zs[.]shop/endpoint
- smolcatkgi[.]shop/endpoint
- dais7nsa[.]shop/endpoint
- ajsdiaolke[.]shop/endpoint
- peskpdfgif[.]shop/endpoint
- ndas8m92[.]shop/endpoint
- test-1627838[.]shop/endpoint
- smkn1leuwimunding[.]com/Updating.zip
- bitbucket[.]org/browsertools/tools/downloads/
- bitbucket[.]org/targetfile/download/downloads/UpdateRequest.exe
- bitbucket[.]org/browserupdater/download/downloads/BrowserUpdater.exe
- bitbucket[.]org/cleopatrall/upds/downloads/updater.exe
- bitbucket[.]org/stoptrackme/updatings/downloads/UpdateMe.exe

- bitbucket[.]org/napoleon_bonaparte/browtool/downloads/BrowserUpdateTool.exe

## C2 Domains

- paveldurov[.]sbs
- zovik[.]info
- analfucker[.]lol

## Sensitive Data Discovery Paths

- c:\users\<username>\appdata\local\google
- c:\users\<username>\appdata\roaming\mozilla\firefox
- c:\users\<username>\appdata\local\microsoft\edge
- c:\users\<username>\appdata\local\bravesoftware\brave-browser
- c:\users\<username>\appdata\local\360chrome
- c:\users\<username>\appdata\roaming\opera software
- c:\users\<username>\appdata\local\vivaldi
- c:\users\<username>\appdata\local\coccoc
- c:\users\<username>\appdata\local\yandex
- c:\users\<username>\appdata\local\chromium
- c:\users\<username>\appdata\local\tencent
- c:\users\<username>\appdata\roaming\jupitercoin
- c:\users\<username>\appdata\roaming\memorycoin
- c:\users\<username>\appdata\roaming\ledger live\local storage\leveldb
- c:\users\<username>\appdata\roaming\bbqcoin
- c:\users\<username>\appdata\roaming\bitbar
- c:\users\<username>\appdata\roaming\crimecoin
- c:\users\<username>\appdata\roaming\globalcoin
- c:\users\<username>\appdata\roaming\grain
- c:\users\<username>\appdata\roaming\lucky7coin
- c:\users\<username>\appdata\roaming\maples
- c:\users\<username>\appdata\roaming\ethereum\keystore
- c:\users\<username>\appdata\roaming\bits
- c:\users\<username>\appdata\roaming\colossuscoin
- c:\users\<username>\appdata\roaming\frankocoin
- c:\users\<username>\appdata\roaming\freecoin
- c:\users\<username>\appdata\roaming\zccoin
- c:\users\<username>\appdata\roaming\zcash
- c:\users\<username>\appdata\roaming\bountycoin
- c:\users\<username>\appdata\roaming\earthcoin
- c:\users\<username>\appdata\roaming\androidstokens
- c:\users\<username>\appdata\roaming\peoplecoin
- c:\users\<username>\appdata\roaming\redcoin

- c:\users\<username>\appdata\roaming\florincoin
- c:\users\<username>\appdata\roaming\sexcoin
- c:\users\<username>\appdata\roaming\lebowskis
- c:\users\<username>\appdata\roaming\skycoin
- c:\users\<username>\appdata\roaming\ezcoin
- c:\users\<username>\appdata\roaming\joulecoin
- c:\users\<username>\appdata\roaming\last coin
- c:\users\<username>\appdata\roaming\dogecoin
- c:\users\<username>\appdata\roaming\megacoin
- c:\users\<username>\appdata\roaming\unobtanium
- c:\users\<username>\appdata\roaming\extremecoin
- c:\users\<username>\appdata\roaming\grandcoin
- c:\users\<username>\appdata\roaming\richcoin
- c:\users\<username>\appdata\roaming\infinitecoin
- c:\users\<username>\appdata\roaming\uscoin
- c:\users\<username>\appdata\roaming\exodus\exodus.wallet
- c:\users\<username>\appdata\roaming\avingcoin
- c:\users\<username>\appdata\roaming\goldcoin
- c:\users\<username>\appdata\roaming\atomic_qt
- c:\users\<username>\appdata\roaming\bitcoin\wallets
- c:\users\<username>\appdata\roaming\namecoin
- c:\users\<username>\appdata\roaming\primecoin
- c:\users\<username>\appdata\roaming\luckycoin
- c:\users\<username>\appdata\roaming\onecoin
- c:\users\<username>\appdata\roaming\quarkcoin
- c:\users\<username>\appdata\roaming\asiccoin
- c:\users\<username>\appdata\roaming\cosmoscoin
- c:\users\<username>\appdata\roaming\ticketscoin
- c:\users\<username>\appdata\roaming\cloudcoin
- c:\users\<username>\appdata\roaming\mavro
- c:\users\<username>\appdata\roaming\secondscoin
- c:\users\<username>\appdata\roaming\supercoin
- c:\users\<username>\appdata\roaming\tagcoin
- c:\users\<username>\appdata\roaming\armory
- c:\users\<username>\appdata\roaming\beaocoin
- c:\users\<username>\appdata\roaming\freicoin
- c:\users\<username>\appdata\roaming\nanotokens
- c:\users\<username>\appdata\roaming\orbitcoin
- c:\users\<username>\appdata\roaming\royalcoin
- c:\users\<username>\appdata\roaming\worldcoin
- c:\users\<username>\appdata\roaming\alphacoin
- c:\users\<username>\appdata\roaming\ferretcoin

- c:\users\<username>\appdata\roaming\galaxycoin
- c:\users\<username>\appdata\roaming\unitedscryptcoin
- c:\users\<username>\appdata\roaming\ybcoin
- c:\users\<username>\appdata\local\coinomi\coinomi\wallets
- c:\users\<username>\appdata\roaming\bottlecaps
- c:\users\<username>\appdata\roaming\neocoin
- c:\users\<username>\appdata\roaming\protosharescoin
- c:\users\<username>\appdata\roaming\novacoin
- c:\users\<username>\appdata\roaming\terracoin
- c:\users\<username>\appdata\roaming\com.liberty.jaxx\indexeddb\file__0.indexeddb.leveldb
- c:\users\<username>\appdata\roaming\americancoin
- c:\users\<username>\appdata\roaming\gamecoin
- c:\users\<username>\appdata\roaming\kingcoin
- c:\users\<username>\appdata\roaming\securecoin
- c:\users\<username>\appdata\roaming\franko
- c:\users\<username>\appdata\roaming\nxtcoin
- c:\users\<username>\appdata\roaming\walletwasabi\client\wallets
- c:\users\<username>\appdata\roaming\fastcoin
- c:\users\<username>\appdata\roaming\nuggets
- c:\users\<username>\appdata\roaming\sifcoin
- c:\users\<username>\appdata\roaming\argentum
- c:\users\<username>\appdata\roaming\philosopherstone
- c:\users\<username>\appdata\roaming\xencoin
- c:\users\<username>\appdata\roaming\devcoin
- c:\users\<username>\appdata\roaming\elephantcoin
- c:\users\<username>\appdata\roaming\hobonickels
- c:\users\<username>\appdata\roaming\protoshares
- c:\users\<username>\appdata\roaming\zetacoin
- c:\users\<username>\appdata\roaming\atomic\local storage\leveldb
- c:\users\<username>\appdata\roaming\craftcoin
- c:\users\<username>\appdata\roaming\cryptogenicbullion
- c:\users\<username>\appdata\roaming\krugercoin
- c:\users\<username>\appdata\roaming\guarda
- c:\users\<username>\appdata\roaming\valuecoin
- c:\users\<username>\appdata\roaming\bytecoin
- c:\users\<username>\appdata\roaming\diamond
- c:\users\<username>\appdata\roaming\feathercoin
- c:\users\<username>\appdata\roaming\pennies
- c:\users\<username>\appdata\roaming\realcoin
- c:\users\<username>\appdata\roaming\electrum\wallets
- c:\users\<username>\appdata\roaming\ixcoin
- c:\users\<username>\appdata\roaming\naanayam

- c:\users\<username>\appdata\roaming\zenithcoin
- c:\users\<username>\appdata\roaming\bitgem
- c:\users\<username>\appdata\roaming\digitalcoin
- c:\users\<username>\appdata\roaming\ppcoin
- c:\users\<username>\appdata\roaming\mincoin
- c:\users\<username>\appdata\roaming\peercoin
- c:\users\<username>\appdata\roaming\shitcoin
- c:\users\<username>\appdata\roaming\liquidcoin
- c:\users\<username>\appdata\roaming\mastercoin
- c:\users\<username>\appdata\roaming\memecoin
- c:\users\<username>\appdata\roaming\tekcoin
- c:\users\<username>\appdata\roaming\tumcoin
- c:\users\<username>\appdata\roaming\yacoin
- c:\users\<username>\appdata\roaming\netcoin
- c:\users\<username>\appdata\roaming\paycoin
- c:\users\<username>\appdata\roaming\spots
- c:\users\<username>\appdata\roaming\chncoin
- c:\users\<username>\appdata\roaming\dollarpounds
- c:\users\<username>\appdata\roaming\playtoken
- c:\users\<username>\appdata\roaming\cryptogenicbullionc
- c:\users\<username>\appdata\roaming\eaglecoin
- c:\users\<username>\appdata\roaming\opensourcecoin
- c:\users\<username>\appdata\roaming\phenixcoin
- c:\users\<username>\appdata\roaming\sauron rings
- c:\users\<username>\appdata\roaming\bitcoin
- c:\users\<username>\appdata\roaming\anoncoin
- c:\users\<username>\appdata\roaming\copper bars
- c:\users\<username>\appdata\roaming\growthcoin
- c:\users\<username>\appdata\roaming\italycoin
- c:\users\<username>\appdata\roaming\42coin
- c:\users\<username>\appdata\roaming\blakecoin
- c:\users\<username>\appdata\roaming\casinocoin
- c:\users\<username>\appdata\roaming\ghisler
- c:\users\<username>\appdata\roaming\psi+\profiles\default
- c:\users\<username>\appdata\roaming\telegram desktop\tdata
- c:\users\<username>\appdata\roaming\discord\local storage\leveldb
- c:\users\<username>\appdata\roaming\filezilla

## About the author

Nadav Lorber

Security Research Tech Lead

Nadav Lorber is a leader on Morphisec's cutting-edge threat research team. He began his career in threat intelligence in 2013, where he was a SOC Specialist for the Israeli government's military intelligence department. Since joining Morphisec, Nadav has helped uncover key insights on topics like Jupyter Infostealer, Log4j, and the Snip3 crypter.

---