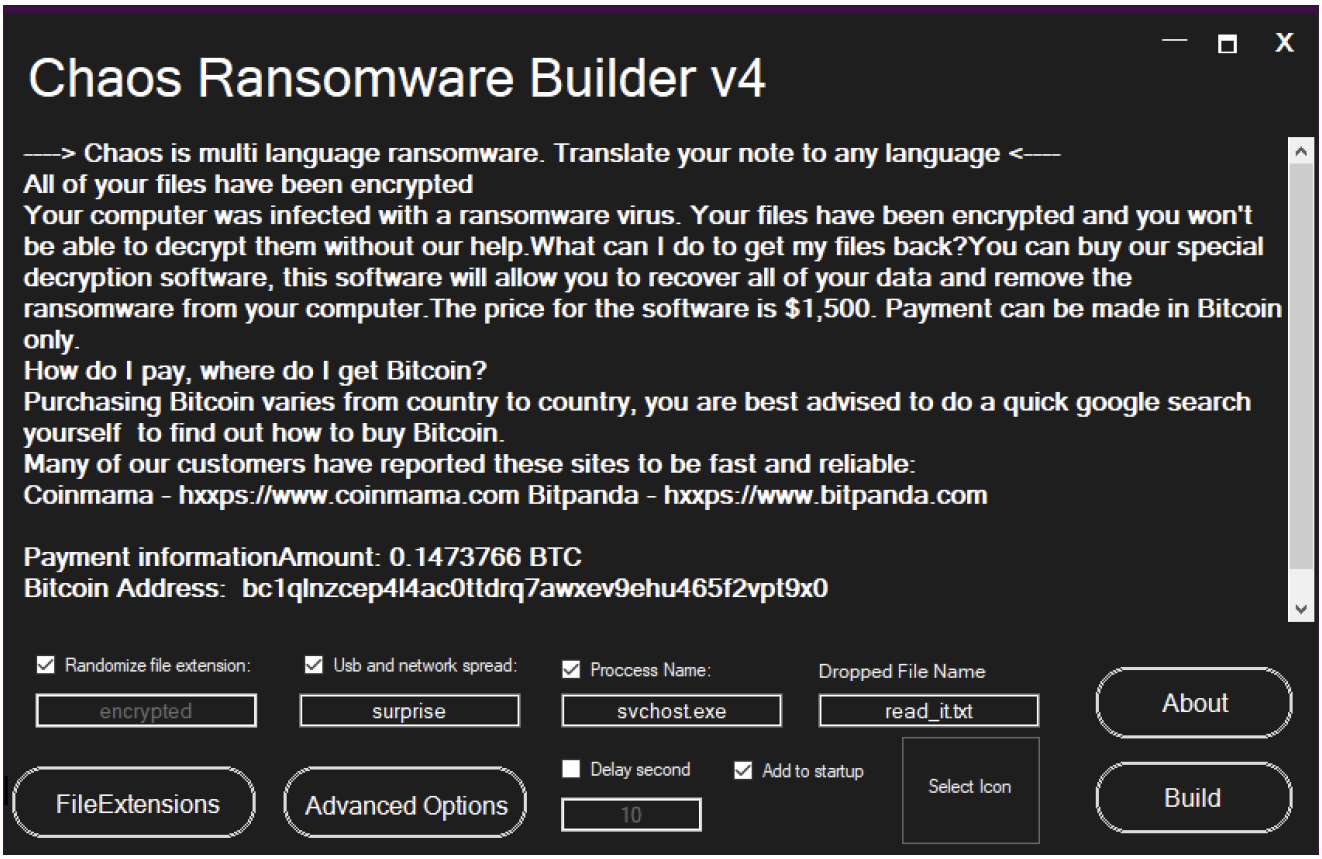
Quasar Chaos

research.openanalysis.net/quasar/chaos/rat/ransomware/2023/04/13/quasar-chaos.html

OALABS Research April 13, 2023



Overview

This sample appears to be a Chaos Ransomware builder but it is actually bound with Quasar RAT!!

- Binder: Celesty Binder
- PDB path: `C:\Users\DarkCoderSc\Desktop\Celesty Binder\Stub\STATIC\Stub.pdb`
- Chaos Ransomware malpedia

Samples

`141056b82cd0a20495822cd2bcd5fae5c989c6d24dac5a5e3c3916f1b406bdb9` UnpacMe

Chaos Builder

Chaos Ransomware builder is an open source project that can be found on GitHub ChaosRansomwareBuilderVersion4. It appears that this project was compiled then the Celesty Binder was used to bind the ransomware builder with Quasar RAT. Both the builder and the RAT can be found in the resources section of the binder exe.

The extracted builder is a clean build and will work on its own
`f2665f89ba53abd3deb81988c0d5194992214053e77fc89b98b64a31a7504d77`

Quasar RAT

Quasar is ostensibly a "remote administration tool" (RAT) that is open source and available on GitHub Quasar. Looking through the source this appears to be developed for the purpose of unauthorized remote access to victims and includes a configuration that could turn this into a malicious RAT.

The extracted RAT
`d8b36742b4c5cf9ce5ce58ac859c4162fb127298dfd3f15fa4f101c0cb878bda`

Analysis

The strings in the RAT (config) are encrypted using Base64 and AES CBC.

```
import base64
import malduck
from Crypto.Protocol.KDF import PBKDF2



string_data =
'muoBJw7vz107HYcI4tyRBz0XVW2kCA367J52yCDjuHUkVGWPKkpXUgV5Q1/s4HNhSAMJDhTJwYIa3MxqdMkg7


string_data_b64 = base64.b64decode(string_data)
string_data_b64 = string_data_b64[32:]
iv = string_data_b64[:16]
enc_data = string_data_b64[16:]

key_data = "SM73jcn259KtoJ4uPciZ"

iterations = 50000
salt =
bytes([191,235,30,86,251,205,151,59,178,25,2,36,48,165,120,67,0,61,86,68,210,30,98,185



key = PBKDF2(key_data, salt, count=iterations)


strings = [
"3DaXS6MYqYL9Q/3WF/cPdbdoy2NggCqoSmasPYwzkPD389j4IoSZZVQHHz196cPEy2h4VSsjy7se22/++XH89

"U2MkYAPUljFBQRO9iIkRZVGmxS2mOB+3klWr1xcKn3OqiosSod4C8iKk+GmogWRVZ6xUFktvHtwFnyOxg+ZSL

"1WvgEMPjdwfqIMeM9MclyQ==",
"NcFtjbDOcsw7Evd3coMC0y4koy/SRZGydhNmno81ZOWOvdfg7sv0Cj5ad2ROUfX4QMscAIjYJdjrrs41+qcQw

"NX2L76Nud+1o8CF2fRs8qiHu4v2wb0E701jiqZNY+WP0X+oOZUuIpza8zsipPF550Uz4XlYTbeon9njxoQ2MB

"DQSIoMapurAvRyZWC74v/c0E7zcV+8LgDPpOmChR453N+Cj+6Fwipe5tbYPbhkpNhwf9hEy/78hh8qB6c1B3n

"p56HD6/EQvRGDzCuDAjko6aJqVPRc/Mug3q2bslOWAZN8H2n4vy8m3x0RtwAUXh5C6kG15y+qrvsfs2s4qJHQ

"xf05S4o+UGg6gPS2slPSroORS4DLfYXnHiWz6VyhTQOpNKzIHxhEvDSTlPMFUIek3Wi3lCxroWOHJr9WeGvvH

"muoBJw7vz107HYcI4tyRBz0XVW2kCA367J52yCDjuHUkVGWPKkpXUgV5Q1/s4HNhSAMJDhTJwYIa3MxqdMkg7

"B0T3cryizrl4VOcnw40TDxor8c5ycs9chw7RjsLxM2h+rS/BlcPa2ZW4po/PpJXob3byyEj4GOuWUPn+M4Shc



for s in strings:
    try:
        string_data_b64 = base64.b64decode(s)
```

```
        string_data_b64 = string_data_b64[32:]
        iv = string_data_b64[:16]
        enc_data = string_data_b64[16:]
        out = malduck.aes.cbc.decrypt(key, iv, enc_data)
        print(out)
    except:
        pass
```

```
b'1.3.0.0\t\t\t\t\t\t\t\t\t'
b'66.63.167.164:55640;\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c'
b'\xf6H\xcf\xc9\x87`\x96\xb7N\xe7O\xf1i\x8d\xc1\xf7'
b'SubDir\n\n\n\n\n\n\n\n\n\n'
b'Client.exe\x06\x06\x06\x06\x06\x06'
b'QSR_MUTEX_M6ajmD3hhoJo7CTsvN\x04\x04\x04\x04'
b'Quasar Client Startup\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b'
b'Ransomware\x06\x06\x06\x06\x06\x06'
b'Logs\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c\x0c'
```