StrelaStealer

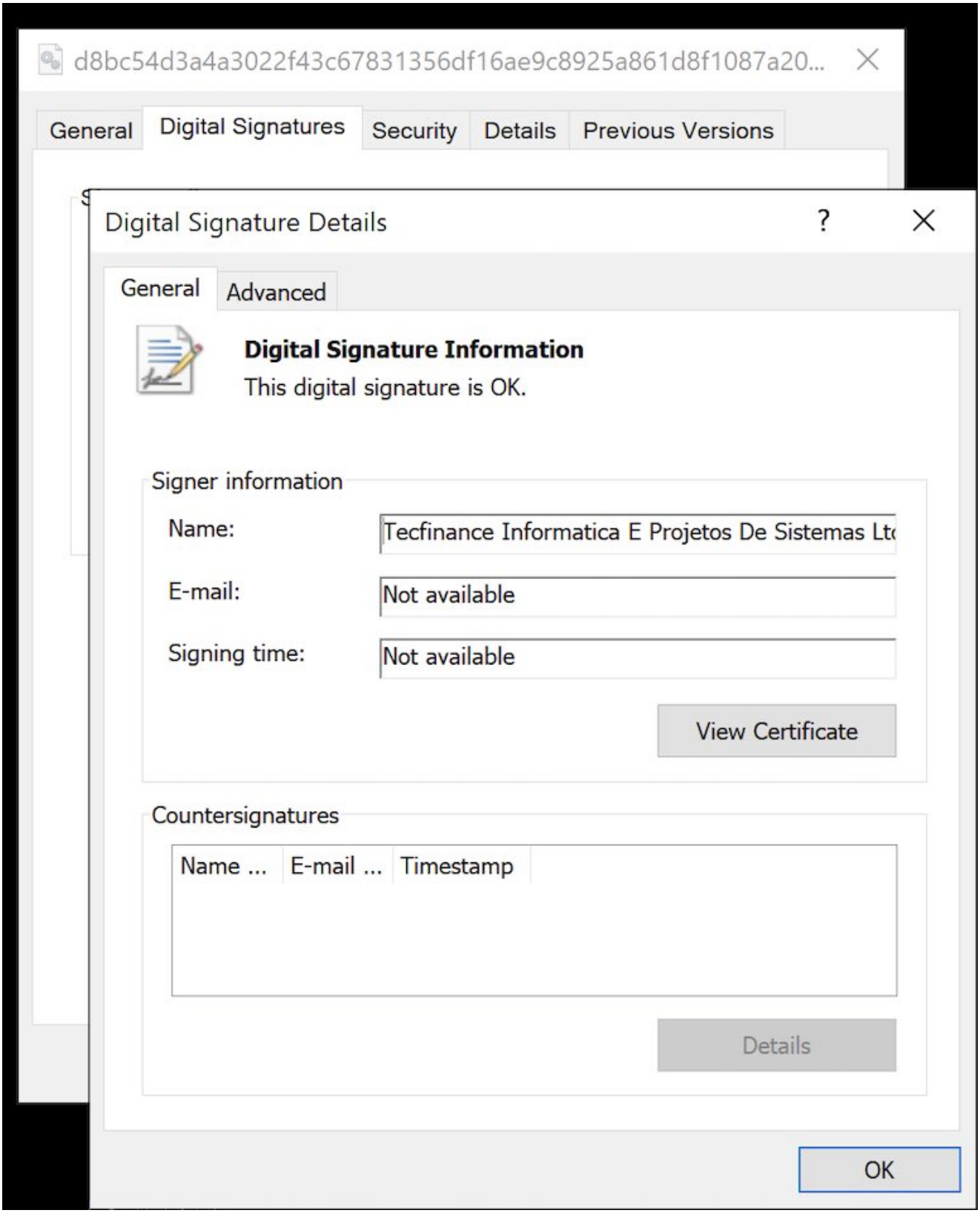
research.openanalysis.net/strelastealer/stealer/2023/05/07/streala.html

OALABS Research May 7, 2023

Overview

This is an email stealer that has been in operation since at least November 2022. The stealer is simple, it collects emails from the target and uploads them to a hard coded C2.

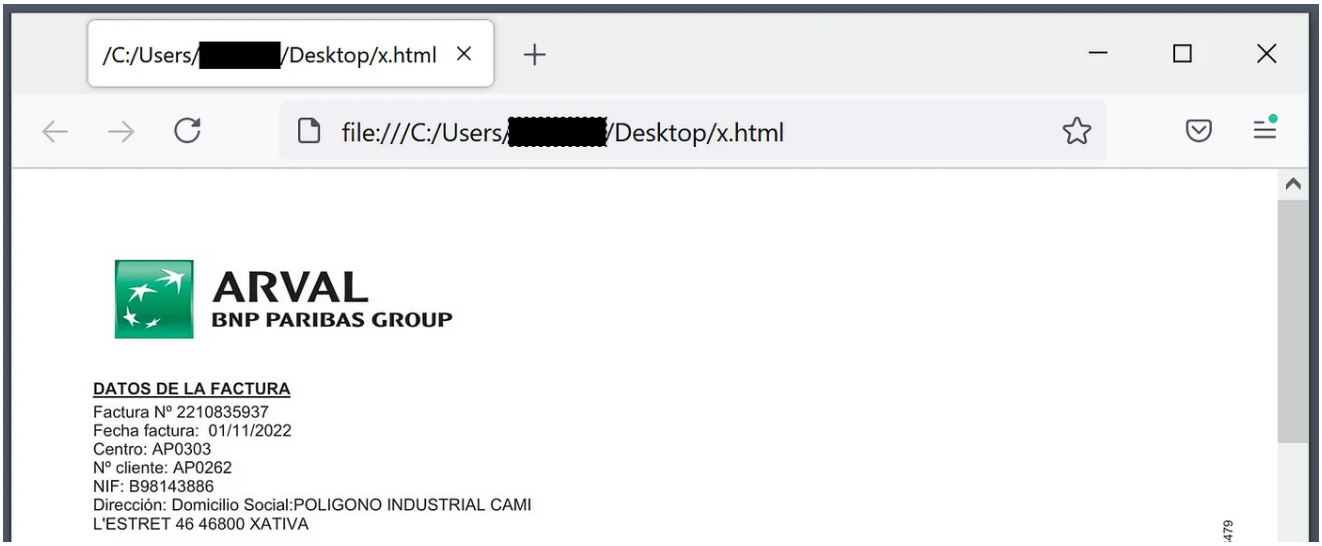
Recent versions have been signed, and there are both 32-bit and 64-bit variants.



New versions pop up a message box in Spanish that attempts to trick the user into thinking the program did not execute correctly.

> El archivo est ...

> https://twitter.com/seanmw/status/1654504780339859456?s=20 Earlier versions of this malware also appeared to target Spanish speaking victims based on the decoy documents that were dropped along with the delivery of the malware.



References

Samples

- `be9f84b19f02f16b7d8a9148a68ad8728cc169668f2c59f918d019bce400d90e` (older) unpacme
- `8b0d8651e035fcc91c39b3260c871342d1652c97b37c86f07a561828b652e907` (older) unpacme
- `3b3f2a92db0f19e96ba8a729709e357419e1aba1ccd48244f34fb74cc621ed51` (obfuscated) unpacme
- `61118d0f778c2f9b3a2bb3e37176ba6a13ee266c49b89dab7e187129f5c00887` (new 32-bit) unpacme
- `fbcfed6900eadd7d36a169400bfcc65a56778cf51152fa8cea0b74daa6cbcd60` (new 64-bit) unpacme

Analysis

Sample - November 2022

be9f84b19f02f16b7d8a9148a68ad8728cc169668f2c59f918d019bce400d90e

Wed Nov 2 08:50:41 2022 UTC

String Decryption

The strings are encrypted with XOR using a hard coded key that resembles a GUID
4f3855aa-af7e-4fd2-b04e-55e63653d2f7

```
key = b'4f3855aa-af7e-4fd2-b04e-55e63653d2f7'
data = bytes.fromhex('4712415d595400')

def xor(data, key):
    out = []
    for i in range(len(data)):
        out.append(data[i] ^ key[i%len(key)])
    return bytes(out)

xor(data, key)

b'strelaa'

table =
bytes.fromhex('46135d5c595952530d43434447016712165741035 70b57181a5641125900144345404 70
```

```
for s in table.split(b'\x00'):
    if len(s) == 0:
        continue
    try:
        tmp_str = xor(s, key)
        if tmp_str.isascii():
            print(tmp_str)
    except:
        pass
```

```
b'rundll32 "%s",Strelag?25/c$$j'
b' %vxr6'
b'a.q*$#n\t3|.f6'
b'\\slc.dllkr;k<tf5g630!!sf\\ka.*y7fu\'p|]?=>q:tx"?=?\r\x1fo+?c)q\x7f<q~\x0cH#?
1mt4\'Z'
b'%s,%s,%s\n'
b'IMAP P'
b'&8%1p4'
b'IMAP User'
b'IMAP Server'
b'SOFTWARE\\Microsoft\\Office\\1'
b')`ZD$sk9vl\t_"w3n%68\t\x1al%\'>h#\x08:1g+\x12AE13a5:`6ce[?mN`(d7}\x11y\x14c/f}\r'
b'POSTJ\x1b'
b'OL'
b'FF'
b'/server.php'
b'Mozill'
b'z~|n"x\x1b<%'
b'i<"(O\x04le(*`?sOnl``!v.1i1 \x15tk>5U'
b'{\x19<((-0f77}"x\x03J\x01\x1fK2&n;9|sA8`m8}>F<#=t7zj763\x7f*5z0~y3'
b'"\x00a:f!?z~'
b';}5o'
b'19'
b'"b6oz)9645}5'
b'strela'
```

IDA String Decrypt

The following can be used to highlight and decrypt strings in IDA.

```
key = b'4f3855aa-af7e-4fd2-b04e-55e63653d2f7'

def xor(data, key):
    out = []
    for i in range(len(data)):
        out.append(data[i] ^ key[i%len(key)])
    return bytes(out)


start = idc.read_selection_start()
end = idc.read_selection_end()
if idaapi.BADADDR in (start, end):
    ea = idc.here()
    start = idaapi.get_item_head(ea)
    end = idaapi.get_item_end(ea)
data = idc.get_bytes(start, end - start).encode('hex')

out = xor(data, key)
print(out)
```

Hot-key Bind.

```
import ida_expr import ida_kernwin
import idc
import ida_bytes
import ida_kernwin

ida_expr.compile_idc_text('static n_key() { RunPythonStatement("nopping()"); }')
ida_kernwin.add_idc_hotkey("Alt-N", "n_key")
```

C2 Comms

The harvested data is sent to the following hard coded C2 with a POST request.

`http[:]//193.106.191[.]166/server.php`

Decoy

The payload expects a `x.pdf` file to be present in the launch directory at runtime. This pdf is launched by the malware as a decoy to trick the user into thinking they have only opened a PDF not launched an executable. The PDF is not dropped by the payload, instead the payload relies on the previous stage to deploy the PDF. This directly ties the payload to delivery stage which is unusual for malware that is sold, and indicates that this malware is both developed and operated by the same actor.

```
cmd /c start msedge x.pdf
```

Sample - April 2023

`61118d0f778c2f9b3a2bb3e37176ba6a13ee266c49b89dab7e187129f5c00887`

`Sun Apr 2 22:42:15 2023 UTC`

Updates

- All strings are now plaintext
- The decoy has been simplified to launch an error message box rather than a PDF
- PE file not a DLL
- The C2 comms are still encrypted with a hard coded XOR string `7a7dd62b-c4ea-4bbb-9f3f-2e6d58aada40`

C2

`http[:]//91[.]215.85.209/server.php`

Decoy

The new "decoy" no longer requires a PDF instead a message box is launched that says...

```
El archivo está dañado y no se puede ejecutar
```

Translated from Spanish this reads `The file is damaged and cannot be executed`, an attempt to trick the user into thinking they can't open the lure. Based on the use of Spanish they are still targeting Spanish victims.

PDB Tracking

The following PDB paths have been found in versions of Strela.

```
C:\Users\Serhii\Documents\Visual Studio
2008\Projects\StrelaDLLCompile\Release\StrelaDLLCompile.pdb
C:\Users\admin\source\repos\Dll1\Release\Dll1.pdb
C:\Users\Serhii\Documents\Visual Studio 2008\Projects\dll1\Release\dll1.pdb
C:\Users\Serhii\source\repos\WindowsProject1\x64\Release\WindowsProject1.pdb
```

Pivoting from these PDB paths and searching through public malware repositories related samples have been recovered indicating that the developer has been working on similar projects since at least `2022-01-23 21:58:42` (sample `d091cb30b4c19b24249af2648d8c43abd5390118d502b5041b5d89d2152a0d7a`).

Malware in Development

Some related samples are not malware but rather test code apparently used to test features in development. These include a sample `2F3A2B18252E39C5B95A199412D97916E6E2611F3A83EF7160E74AA959A41933` that appears to be some type of `putty.exe` launcher using the local path `C:\\Users\\Serhii\\Downloads\\putty.exe`.

C2 Tracking

Performing a revers lookup for the C2 IP `91.215.85[.]209` the following domains have been registered and point to this IP. It is unclear what their purpose is.

```
posts-fi[.[com
carrefours-tw[.]com
directeredie[.]org
dkpostnord[.]com
fornying-skonto[.]com
redisimple[.]com
chunghwa-post[.]app
atuh-manor[.]com
post-chunghwa-tw[.]app
post-chunghwa-tw[.]com
tibouton[.]org
carrefour-tw[.]com
post-tw[.]app
```