

22 JUNE 2022 •

SAMIR BOUSSEADEN • DANIEL STEPANIC • ELASTIC SECURITY INTELLIGENCE & ANALYTICS TEAM

# A close look at the advanced techniques used in a Malaysian-focused APT campaign

Our Elastic Security research team has focused on advanced techniques used in a Malaysian-focused APT campaign. Learn who's behind it, how the attack works, observed MITRE attack® techniques, and indicators of compromise.

🕒 9 min read   📁 Campaigns






The Elastic Security Intelligence & Analytics Team researches adversary innovations of many kinds, and has recently focused on an activity group that leveraged remote templates, VBA code evasion, and DLL side-loading techniques. Based on code similarity and shared tactics, techniques, and procedures (TTPs), the team assessed this activity to be possibly linked to a Chinese-based group known as APT40, or Leviathan. The group's campaign appears to target Malaysian government officials with a lure regarding the 2020 Malaysian political crisis.

## Anatomy of the attack

To initiate their advanced persistent threat (APT) campaign, the group likely delivered a Microsoft Word document as a phishing lure attachment. The image used in the lure (Figure 2) appears to be crafted from a broadcast announcement shared by a Malaysian blogger (Figure 1). The lure image includes the same broadcast time, but the date and speech topic are removed. Once this attachment is opened, a decoy document is presented while behind the scenes, taking the following actions:

- The lure document downloads the remote template RemoteLoad.dotm
- The remote template executes VBA macro code
- The VBA macro code unpacks and executes two embedded base64-encoded DLLs (sl1.tmp and sl2.tmp) to c:\users\public\


[Gharedon Group blog post](#). This an effective approach used by adversaries to bypass perimeter controls such as email gateways.

Both embedded DLLs (sl1.tmp and sl2.tmp) are similar and export the same function names: RCT and RCP. The first DLL (sl1.tmp) is used to download a benign executable called LogiMailApp.exe and an associated library LogiMail.dll, and the second DLL (sl2.tmp) is used to execute LogiMailApp.exe, which automatically attempts to execute LogiMail.dll due to an inherent DLL search order vulnerability we'll cover shortly.

File name	File type	Size (bytes)	MD5	Compile time
LogiMailApp.exe	Win32 EXE	311656	850a163ce1f9cff0367854038d8cfa7e	2012-09-26 22:13:13+00:00
LogiMail.dll	Win32 DLL	105984	b5a5dc78fb392fae927e9461888f354d	2020-06-03 04:08:29+00:00
sl1.tmp	Win32 DLL	3072	ccbdda7217ba439dfb6bbc6c3bd594f8	2019-11-29 17:15:29+00:00
sl2.tmp	Win32 DLL	3072	dbfa006d64f39cde78b0efda1373309c	2019-11-29 21:23:44+00:00

Table 1: Dropped files metadata

This implementation stood out to our researchers due to a behavioral idiosyncrasy:

- The Microsoft Office application winword.exe loads sl1.tmp and sl2.tmp DLLs uses the LoadLibraryA method, which is moderately rare
- These DLLs run explicit commands or install a payload from a URL using the CallWindowProcA method, which appears to be exceptionally rare
- Both DLLs are deleted after execution



The embedded DLLs, `sl1.tmp` and `sl2.tmp`, have very limited functionality — exporting the RCP and RCT functions. The RCP function implements the `WinExec` method to execute commands where the RCT function uses the `URLDownloadToFileA` method to download a file from a specified URL.

## DLL side-loading a backdoor

`LogiMailApp.exe`, which is downloaded by `sl1.tmp` and executed by `sl2.tmp`, is vulnerable to a form of DLL search-order hijacking called side-loading, which automatically searches for and executes `LogiMail.dll` if found in the same directory. Forms of DLL search-order hijacking can be used with many third-party software applications. In this case, search-order hijacking was used to load a backdoor that exports the following notable functions:

The adversary-created binary `LogiMail.dll` exports the function `DllGetClassObject` that contains critical logic for the execution flow of this sample:

1. Download an AES-encrypted second stage object to `%TEMP%~liseces1.pcs`
2. Derive a 128-bit AES key and initialization vector from SHA256 of a hardcoded string
3. Read and decrypt `%TEMP%~liseces1.pcs` in memory using the `ReadFile` and `CryptDecrypt` functions
4. Delete `%TEMP%~liseces1.pcs` from disk

## Second stage backdoor

The decrypted second stage backdoor is mapped into memory and then its original entry point (OEP) is called, thus bypassing successful detections based on file system scanning.



This payload supports the following capabilities:

- Basic anti-debug checks
- System and user discovery
- Execution via command line
- File discovery, upload, and download
- Persistence via run registry
- Encrypt C2 traffic using same AES key

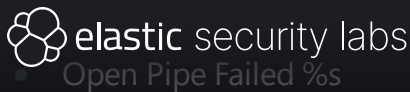
## Possible APT40/Leviathan connection

Earlier in the year, the Malaysian Computer Emergency Response Team (MyCERT) issued an [advisory](#) related to espionage activity targeting their country. The report listed different TTPs and included multiple samples and other technical indicators that align with a threat group known as APT40/Leviathan.

At a high level, this sample follows the continued trend of targeting Malaysian victims using specific TTPs such as remote templates, employing macros, using DLL side-loading techniques, and leveraging an in-memory implant with dynamic DNS for command and control. More specifically, the second stage implant from this lure shares unique strings and URL references and contains similar functionality that correlates with the previous reporting for APT40/Leviathan. With these similarities, our Intelligence & Analytics Team assesses with moderate confidence that this activity is linked to APT40/Leviathan.

Implant String Similarities with MyCERT Sample:

- /list\_direction
- /post\_document
- /post\_login



- Open Pipe Failed %s
- Download Read Path Failed %s
- %02X-%02X-%02X-%02X-%02X-%02X
- Software\Microsoft\Windows\CurrentVersion\Run
- ntkd

## Conclusion

In this post, we highlighted a recent sample that most likely represents the work of a highly organized adversary. Activity groups like this are significant for everyone to take notice of, if only because they represent a higher maturity level of post-exploit innovation. Their cutting edge TTPs today end up being everyone's run of the mill tomorrow; it's important to learn from these events.

We hope that by sharing some of these insights, we can help raise awareness and continue to focus on protecting the world's data from attack. To enable organizations further, we've added all the observed MITRE ATT&CK® techniques and indicators of compromise (IoCs) below.

## MITRE ATT&CK® techniques

- [T1193 - Spearphishing Attachment](#)
- [T1221 - Template Injection](#)
- [T1060 - Registry Run Keys / Startup Folder](#)
- [T1073 - DLL Side-Loading](#)
- [T1129 - Execution through Module Load](#)
- [T1055 - Process Injection](#)



- [T1140 - Deobfuscate/Decode Files or Information](#)
- [T1059 - Command-Line Interface](#)

## Indicators of Compromise (IOCs)

### File names and paths

```
Bubar Parlamen.zip
Bubar Parlamen.docx
RemoteLoad.dotm
C:\Users\Public\s11.tmp
C:\Users\Public\s12.tmp
C:\Users\*\AppData\Local\Temp\~1iseces1.pcs
C:\Users\*\AppData\Local\Microsoft\Office\LogiMailApp.exe
C:\Users\*\AppData\Local\Microsoft\Office\LogiMail.dll
```

### Registry keys

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\ntkd
```

### URLs

```
hxxps[ : ]//armybar[.]hopto[.]org/LogiMail.dll
hxxps[ : ]//armybar[.]hopto[.]org/LogiMailApp[.]exe
hxxps[ : ]//armybar[.]hopto[.]org/Encrypted
hxxp[ : ]//tomema.myddns[.]me/postlogin
hxxp[ : ]//tomema[.]myddns[.]me/list_direction
hxxp[ : ]//tomema[.]myddns[.]me/post_document
```

### IPs



## HTTPS certificate

```
74b5e317527c93539dbaaf84d6a61da92a56012a
```

## Hashes

```
523cbdaf31ddc920e5b6c873f3ab42fb791fb4c9d1f4d9e6a7f174105d4f72a1
ab541df861c6045a17006969dac074a7d300c0a8edd0a5815c8b871b62ecdda7
145daf50aefb7beec32556fd011e10c9eaa71e356649edfce4404409c1e8fa30
93810c5fd9a287d85c182d2ad13e7d30f99df76e55bb40e5bc7a486d259810c8
925f404b0207055f2a524d9825c48aa511199da95120ed7aafa52d3f7594b0c9
feca9ad5058bc8571d89c9d5a1eebce09e709cc82954f8dce1564e8cc6750a77
06a4246be400ad0347e71b3c4ecd607edda59fbf873791d3772ce001f580c1d3
77ef350639b767ce0a748f94f723a6a88609c67be485b9d8ff8401729b8003d2
```

## YARA

```
rule APT_APT40_Implant_June2020 {
  meta:
    version = "1.0"
    author = "Elastic Security"
    date_added = "2020-06-19"
    description = "APT40 second stage implant"
  strings:
    $a = "/list_direction" fullword wide
    $b = "/post_document" fullword wide
    $c = "/postlogin" fullword wide
    $d = "Download Read Path Failed %s" fullword ascii
    $e = "Open Pipe Failed %s" fullword ascii
```





elastic security labs



```
    "cmd.exe Read Path Failed %s" fullword ascii
    $h = "\\cmd.exe" fullword wide
    condition:
      all of them
  }
```

## References

- <https://www.mycert.org.my/portal/advisory?id=MA-774.022020>
- <https://prezi.com/view/jGyAzyy5dTOkDrtwsJi5/>
- <https://www.fireeye.com/blog/threat-research/2019/03/apt40-examining-a-china-nexus-espionage-actor.html>
- <https://malpedia.caad.fkie.fraunhofer.de/details/win.dadstache>

### Share this article



Twitter



Facebook



LinkedIn



Reddit