## THE NEXT-GEN PLUGX/SHADOWPAD? A DIVE INTO THE EMERGING CHINA-NEXUS MODULAR TROJAN, PANGOLIN8RAT

Silvia Yeh / Leon Chang





#### Silvia Yeh

- Threat Intelligence Analyst
- OSINT, APT, InfoOps in APAC
- 2022 SANS CTI Summit, 2021 CODE BLUE, 2021 HITCON Pacific, etc.



#### Leon Chang

- Threat Intelligence Researcher
- Reverse engineering, APT campaign tracking in APAC, IoT security
- 2022 JSAC, 2021 JSAC

# Speaker

## Outline

- 1. Intro: Modular shared tools among Chinese APTs
- 2. Anatomy of Pangolin8RAT
- 3. APT Tianwu: Activity Timeline, Target, Attribution
- 4. Conclusion and outlook

# **1. Background Information**

PlugX, ShadowPad, and modular tools shared among Chinese APTs

## PlugX

- First Seen: 2008
- A RAT with modular plugins
- Used by many Chinese APT groups:
  - Amoeba/APT41, APT27, DragonOK, Polaris, menuPass, LuoYu, and more...
- "PlugX" → plugin and malware module features
- Various PlugX variants (Some have other communication protocols, including P2P and DNS Tunneling.)

## ShadowPad

- First Seen: 2015
- A RAT with modular plugins
- Used by many Chinese APT groups (respectively limited):
  - Amoeba/APT41, Fishmaster, Sanyo, LuoYu, Naikon, more...
- Its functions are provided by interchangeable modules
- Protected by layers of encryption and heavy obfuscation

6

• The modular design resembles PlugX

## **Related malware families**

7

- PlugX
- ShadowPad
- Pangolin8RAT
- FFRAT
- KeyPlug
- Winnti 2.0
- FunnySwitch
- Natwalk (Sidewalk)
- Crosswalk

•

## Amoeba, Fishmaster, and Tianwu - Modular malware

## **Fishmaster**

aka Lusca Earth, TAG-22 - Amoeba's subgroup Amoeba

aka APT41, Barium - Notorious Chinese

- APT group
- Chengdu404
- Civilian hackers

- KCP protocol
- Multiple C2 protocol supported

8

- CobaltStrike technique

### Tianwu

- Tool: Pangolin8RAT

- An emerging China-nexus group that has been active since late-2020

ShadowPad, Winnti, FunnySwitch

## **Tianwu Profile**





Tools

#### Pangolin8RAT

#### Custom CobaltStrike Beacon

TTPs

Modular malware

**KCP** protocols

Phantom DLL hollowing

**Target Region** 



Target Industry

Gambling, Gaming, IT, Telecom, Gov, Transport, Dissident

# 2. Anatomy of Pangolin8RAT

Pangolin8RAT's evolution and its similarities with other malware families

10

## Malware Profile: Pangolin8RAT

| Category        | Description  |
|-----------------|--|
| Туре            | Modular backdoor   |
| Naming          | The PDB string contains "pangolin" and its RTTI contains "p8rat"                                   |
| First seen      | 2019/11  |
| Function        | supported 8 communication protocols, including TCP, HTTPS, UDP, DNS, ICMP, HTTPSIPV6, WEB, and SSH |
| Target industry | Online gaming, gambling, IT, Telecom, Transportation, Gov, Dissident                               |
| Linked APT      | Tianwu   |

## Naming

#### The PDB string

- Z:\Disk\pangolin\_reload\Release\core\ldr\Mfcldrx64.pdb
- D:\PangolinRev\Release\core\LiteCorex64.pdb
- D:\PangolinRev\Release\core\corex64.pdb

#### The RTTI

- P8RatCore
- P8CorePluginManager

## In-Depth Analysis of Pangolin8RAT

- The installer/dropper of Pangolin8RAT
- The evolution of Pangolin8RAT
- The code similarity with FFRAT and Winnti 2.0
- TTPs overlap with Amoeba group malware family

## The installer of Pangolin8RAT

- It's written in C++ using MFC application framework
- The pdb string:
  - Z:\Disk\pangolin\_reload\Release\core\ldr\Mfcldrx64.pdb
- Used COM Session Moniker Privilege Escalation (MS17-012)

## The installer of Pangolin8RAT

The installer contains an additional encrypted/encoded module in the resource section.

| 8b6a63e522fd6b3f23f476a101720 |  |
|-------------------------------|--|
|                               |  |
|                               |  |
|                               | Offeet 0 1 2 3 4 5 6 7 8 9 Å B C D F F Åenii   |
| ± Icons                       |  |
|                               |  |
| 🗄 🛅 String Tables             |  |
| 🖻 🛅 RCData                    |  |
|                               |  |
|                               | 00000050 00 00 00 00 00 00 00 00 00 00 0       |
|                               | 00000060   00 00 00 00 00 00 00 00 00 00 00 00 |
| 1829 - Dang: 1078]            |  |
| 1830 - Dang: 1078]            |  |
| 1830 - [lang, 1076]           |  |
| 1832 - [lang: 1078]           |  |
| E Cursor Groups               |  |
|                               |  |
| 🗄 🛅 Version Info              | 000000E0 00 00 00 00 00 00 00 00 00 00 0       |
| 🗄 🧰 Configuration Files       | 000000F0 00 00 00 00 00 00 00 00 00 00 0       |
|                               |  |
|                               |  |
|                               |  |
|                               |  |
|                               |  |
|                               |  |
|                               |  |
|                               |  |

Md5=8b6a63e522fd6b3f23f476a101720bf9

## **Resource format**





C2 Server

## **Module Decryption Procedure**



## The summary of resource ID

| Resource ID | Filename or in-memory module name | Description   |
|-------------|-----------------------------------|---|
| 1809        | N/A                               | must exist, check size >= 24 bytes                                    |
| 1817        | inst.dat                          | persistence path  |
| 1825        | smcache.dat                       | C2 config   |
| 1826        | newuac.dll                        | UAC bypass module   |
| 1829        | newwhite.dll                      | Drop files(launcher, loader, config) for<br>loading Pangolin8RAT core |
| 1830        | pkgx64.dll                        | Decrypt and decompress modules  |

## The summary of resource ID

| Resource ID | Filename or in-memory module name | Description                                   |
|-------------|-----------------------------------|---|
| 1832        | log.dll                           | Loader(corex64.dll, MainLdr.dll)              |
| 1840        | bdservicehost.exe                 | Legitimated launcher for dll hijacking        |
| 1841        | N/A                               | Filenames for DLL hijacking                   |
| *1816       | hostcfg.dat                       | Used in the Host header, in C&C communication |
| *1833       | bdservicehost.exe                 | Signed PE for sideloading 32bit – N/A         |
| *1831       | log.dll                           | Loader (32bit)                                |

## The evolution of Pangolin8RAT – Timeline



The timeline of TTPs used by Pangolin8RAT

\*1: https://github.com/forrest-orr/phantom-dll-hollower-poc \*2: https://gist.github.com/jthuraisamy/4c4c751df09f83d3620013f5d370d3b9

## The evolution of Pangolin8RAT - Case1



- 1. No longer uses hardcoded AES IV
  - The Resource\_BLOB/payload contains the AES KEY and IV value (use "#" as delimiter)
- 2. Used shellcode injection technique\* to bypass EDR detection
  - Call windows API EnumSystemCodePagesW

```
v71 = nNumberOfBytesToRead[0];
v72 = &lpBuffer;
if ( v78 >= 0x10 )
  v72 = lpBuffer.m128i i64[0];
shellcode = call NtAllocateVirtualMemory(0i64, *nNumberOfBytesToRead, 0x1000i64, 0x40i64)
*&PipeAttributes.nLength = 0x18i64;
*&PipeAttributes.bInheritHandle = 1i64;
hWritePipe = 0i64;
hReadPipe = 0i64;
PipeAttributes.lpSecurityDescriptor = 0i64;
NumberOfBytesRead = 0;
CreatePipe(&hReadPipe, &hWritePipe, &PipeAttributes, 2 * v71);
WriteFile(hWritePipe, v72, v71, &NumberOfBytesRead, 0i64);
NumberOfBytesRead = 0;
ReadFile(hReadPipe, shellcode, v71, &NumberOfBytesRead, 0i64);
CreateThread(0i64, 0i64, call EnumSystemCodePagesW, shellcode, 0, 0i64);// run shellcode
```

- 3. PE2Shellcode: Convert corex64.dll to shellcode
  - PDB:

"D:\pe2shellcode\x64\Release\native\_loader.pdb"

## The evolution of Pangolin8RAT - Case1



• Download from C2 Server

The workflow of Pangolin8RAT in Mar. 2021



The workflow of Pangolin8RAT in June 2021

## The evolution of Pangolin8RAT - Case2 (workflow)

- Extracted from process kwsprotect64.exe

- Download from C2 Server

![](_page_24_Figure_3.jpeg)

The workflow of Pangolin8RAT in June 2021

## The evolution of Pangolin8RAT - Case1 vs. Case2

![](_page_25_Figure_1.jpeg)

## The evolution of Pangolin8RAT - Case1 vs. Case2

![](_page_26_Figure_1.jpeg)

- 2. C2 config stored in resource ID:2321 and config is encoded.
- 3. Add persistence path in c2 config structure

base\_file\_mgr.dll

## The evolution of Pangolin8RAT - Case1 vs. Case2

![](_page_27_Figure_1.jpeg)

## The evolution of Pangolin8RAT - Case2 (persistence)

- 1. Uses known persistence method\*:
  - HKLM\Software\Microsoft\Cryptography\Offload\ExpoOffload = C:\ProgramData\SppTools\ess4c85b739.dll
- 2. Same shellcode injection technique

![](_page_28_Figure_4.jpeg)

- 4. Local privilege escalation through CVE-2019-16098
- 5. Create process kwsprotect64.exe with SYSTEM privilege

## The evolution of Pangolin8RAT - Case3 (Ketugya)

#### Ketugya: Malware Profile

| Category   | Description   |
|------------|---|
| Туре       | Loader  |
| Naming     | Ketugya is named after its PDB string of final stage dll<br>"E:\fud2\ <mark>Ketugya</mark> \bin\x64\test_msg.pdb".  |
| First seen | 2022/02   |
| Functions  | <ul> <li>Uses Tiny-AES algorithm to decrypt payload in-memory</li> <li>Kills EDR process</li> <li>Patches ETW, UAC bypass</li> <li>Anti-IDA Pro decompiler</li> <li>Searches NortonSecurity.exe, avp.exe</li> </ul> |
| Linked APT | Tianwu  |

## The evolution of Pangolin8RAT - Case3 (Ketugya)

#### The workflow of Ketugya

- Same decryption method with Case 2
  - Step1: Read payload from resource ID:1905
  - Step2: Uses Tiny-AES algorithm to decrypt the payload
  - Step3: Unzip the decrypted payload -> shellcode
  - Step4: shellcode will use Tiny-AES algorithm to decrypt payload again -> in-memory dll (first stage DLL)
  - Step5: run first-stage DLL in memory via reflective DLL injection techniques
  - Step6: go-to Step1 until the final-stage DLL/payload is running in-memory

## The evolution of Pangolin8RAT - Case3 (Ketugya)

#### TTPs used by Ketugya

- Kill EDR process\*
  - Modify the token Integrity of the PPL (Protected Process Light) process to Untrusted
  - Kill Windows defender process (MsMpEng.exe)
- Anti IDA Pro decompiler
  - Junk code will cause decompilation failure (stack frame is too big)

![](_page_31_Picture_7.jpeg)

\* ref: https://elastic.github.io/security-research/whitepapers/2022/02/02.sandboxing-antimalware-products-for-fun-and-profit/article/

![](_page_32_Figure_1.jpeg)

- Pangolin8RAT.FileMgr
  - 0879125ed34df60a70ed5bb8d58f3a19
- FFRAT
  - 1962a69c204289cb8214a30c15f05609
- Winnti 2.0
  - 5778178a1b259c3127b678a49cd23e53

![](_page_32_Figure_8.jpeg)

#### Code overlap/reuse

- 1. Pangolin8RAT.FileMgr vs. FFRAT
  - Code overlap just change XOR key: 0xBC vs. 0x57
  - Same debug string and proxy connector class reused

.?AVsocks4a\_connector@@ .?AVconnector@@ .?AVsocks4\_connector@@ .?AVhttp\_tunnel\_ntlm@@ .?AVhttp\_connect\_ntlm@@ .?AVhttp\_proxy\_connector@@ .?AVsocks5\_connector@@

#### 2. FFRAT vs. Winnti 2.0

- Same debug string
  - "m\_ServerComplete Continue\n"
  - "SrvCode", "DrvCode"

![](_page_33_Figure_10.jpeg)

| 😧 IDA - 0879125ed34df60a70ed5bb  | 98d58f3a19 C:\Users\user\Desktop\p8rat\0879125ed34df60a70ed5bb8d58f3a19 📃 🔍 🗶  | TDA - 1962a69c204289cb8214      | a30c15f05609 C:\Users\user\Desktop\FFRAT\1962a69c204289cb8214a30c15f05609                        |                            |
|--|--|---------------------------------|--|----------------------------|
| File Edit Jump Search View   | Debugger Lumina Options Windows Help   | File Edit Jump Search Vi        | iew Debugger Lumina Options Windows Help   |                            |
| 📑 📑 🖕 - 🔿 - 16, 16, 6,   |  | 1 📫 🚍 1 🍝 - 🚽 1 🐜 1 🏣           | 🚳 🚳 🖍 👘 👩 🗛 🚓 🚓 📾 👉 🛷 🧀 🖌 🖌 🖿 🔲 🖬 No debugger  | 🖌 🐜 📷 🗄 🚮 👫 🚧              |
|  |  |                                 |  |                            |
| Librar function Development of   | Lateria Die Rouder Betradundel Lanie Andre   |                                 | the Interaction Data Harmonical Technological Interior function                                  |                            |
| . Library function Kegousi reaction  |  | - Library function Kegular func | non haurenon bes onexpires Extensi syntoi Lumma runteon  |                            |
| Functions 🗖 🗧 🗙 🛄  | 📴 IDA 💌 📑 Facud 🔟 🔚 Strings 🐹 🔟 He 🔟 🚮 S 🔟 🛅 E 💌 🖓 E 💌   | Functions 🗖 🗗 🗙                 | 📭 IDA···· 🛛 📔 Pasud···· 🖾 😨 Stringg···· 🔀 🔘 He··· 🔟 🖪 S··· 💌                                     | 🛗 E··· 🗶 🛐 I··· 🗶 🛃 E··· 🗶 |
| Function name  | 1_int64tastcall sub_180016DA0(int64 a1)  | Function name                   | 1_int64fastcall sub_180014AF8(int64 a1)  |                            |
| 🗲 _acrt_EnumSystemLocale   | 3 SOCKET v2; // rbx  | 🗲 _chsize_nolock                | 3 unsigned int v2; // edi  |                            |
| 📝 _acrt_FlsAlloc   | 4 int v3; // esi   | 📝 _lseek_nolock                 | 4 int v3; // eax   |                            |
| 🗲 _acrt_FlsFree  | 5 int v4; // en14<br>6 int64 v5; // rcx  | 🗲 _setmode_nolock               | 5 01NT V4; // eax<br>6 int v5; // er13   |                            |
| 🖌 _vcrt_FlsFree_0  | 7int64 v6; // rdx  | 📝 sub_180029914                 | 7 SOCKET v6; // r12  |                            |
| F _acrt_FlsSetValue  | 8 _int64 v7; // r8   | 🗲 _tzset_nolock                 | 8int64 v7; // r11  |                            |
| vcrt_EventSetInformatic  | 9 Int vo; // er9   | 🗲 cvtdate                       | 10 char v10[8]; // [rsp+40h] [rbp-538h] BYREF  |                            |
| acrt_GetSystemTimePre  | 11 void "vi@; // rcx   | 🖌 _isindst_nolock               | 11 void *Block; // [rsp+48h] [rbp-530h]  |                            |
| acrt_GetUserDefaultLoc   | 12 char optval[4]; // [rsp+40h] [rbp-C6h] BYREF  | 🗲 _tzset                        | 12int64 v12; // [rsp+58h] [rbp-520h]   |                            |
| facrt_InitializeCriticalSec  | 13   | 🗲 _isindst                      | 13 unsignedinto4 vi3, // [rsp+508h]<br>14 int v14; // [rsp+70h] [rbp-508h]                       |                            |
| facrt_IsValidLocaleName  | 15 struct _RTL_CRITICAL_SECTION CriticalSection; // [rsp+58h] [rbp-A8h] BYREF  | F sub_18002A3BC                 | 15 int v15[55]; // [rsp+74h] [rbp-504h] BYREF  |                            |
| acrt_LCIDToLocaleNam   | 16 struct_RTL_CRITICAL_SECTION_V16; // [rsp+80h] [rbp-80h] BYREF   | 🗲 sub_18002A3FC                 | 16 char v16; // [rsp+150h] [rbp-428h] BYREF<br>17 char v17[1033]; // [rsp+151h] [rbp-427h] BYREF |                            |
| _acrt_LCMapStringEx  | 1/   | 🗲 sub_18002A43C                 | 18 18  |                            |
| _acrt_LocaleNameToLCI  | 19 struct WSAData WSAData; // [rsp+D0h] [rbp-30h] BYREF  | <i>f</i> sub_18002A47C          | 19 v2 = 0;   |                            |
| Facrt_can_use_vista_local  | 20 char v20[1024]; // [rsp+270h] [rbp+170h] BYREF  | F sub_18002A484                 | 20 memset(v15, 0, 0xD4ui64); 21 memory (0x95551, 0uple 190045A54, 0u40ui64);                     |                            |
| 🗗acrt_initialize_winapi_th   | 22 v13 = 0xFFFFFFFFFFFFFFFeui64;   | <u> </u>                        | 21 memmove(avij[5], aunk_1886+3Ac4, 8A+8010+); 22 v14 = 0x13572468;                              |                            |
| 📝 _acıt_is_packaged_app 🛛 🌒  | 23 v2 = *(_QWORD *)a1;   | 🖌 sub_18002A494                 | 23 v15[0] = 0x1000010;   |                            |
| 📝acrt_uninitialize_winapi_   | 24 memset(SMSAData, 0, 0xDSui64);  | 🖌 _toupper_l                    | <pre>24 LOWORD(v15[2]) = dword_180045AB4 != 0;</pre>   |                            |
| 📝 _fcloseall   | 2 * (WorkD *)&WSADITA:SZDESTIPTION[0x40] = *(WORD *)(a1 + 0x710); 2 * (WORD *)&WSADITA:SZDESTIPTION[0x40] = 0x2D74FFA7;  | 📝 toupper                       | 0 25 = (a1 + exoso);<br>0 26 = v15[1] = 4;   |                            |
| 📝acrt_stdio_free_buffer_n 💽 🍳  | 27 *(_DWORD *)&WSAData.iMaxSockets = 0x1002;   | 🗲 _getbuf                       | 27 v15[0x15] = v3;   |                            |
| 📝 _acrt_stdio_free_stream( 🖉 🎴   | <pre>#SAData.lpvendorInfo = (char *)qword_180062228;<br/>0 #/(DDDD_NESDate_concentration_c</pre> | 📝 _isatty                       | 28  v4 = GetACP();   |                            |
| 📝crt_seh_guarded_call <ir th="" 🏅<=""><th>23 v3 = dword 1806134C;</th><th>📝 _initp_misc_cfltcvt_tab</th><th><ul> <li>30 v15[3] = v4;</li> </ul></th><th></th></ir> | 23 v3 = dword 1806134C;  | 📝 _initp_misc_cfltcvt_tab       | <ul> <li>30 v15[3] = v4;</li> </ul>  |                            |
| 📝 sub_1800370D0 🔹  | <pre>31 *(_DWORD *)&amp;WSAData.szDescription[4] = dword_180061B4C;</pre>  | 📝 _get_printf_count_outpu       | <pre>31 v15[4] = dword_1800455FC;</pre>  |                            |
| 📝 _close_nolock  | 32  v4 = "(DWORD ")(a1 + 9x718);   | 📝 _wctomb_s_l                   | 0 32 v6 = a1;  |                            |
| icurch ma  | 33 weise(vz) v 31201(vz));   | - wetamb e                      | 34 memset(v17, 0, sizeof(v17));  |                            |
| Line 900 of 1065   | 35 v6 = 0xD8i64;   | Line 672 of 762                 | ● 35 v7 = 0i64;  |                            |
|  | 36 do  |                                 | 36 v8 = 0xD8164;<br>37 do  | 1                          |
| 👬 Graph ov: 🗖 🖅 🗙 🍵  | 38 v20[v5] = *((_BYTE *)&WSAData.wVersion + v5) ^ 0xBC;  | 🚠 Graphove 🗆 🖃 🗙                | 38 {   |                            |
|  | 39 ++vS;   |                                 | • 39 *( $\$v16 + v7$ ) = *( $\$v14 + v7$ ) ^ 0x57;   |                            |
|  | 40Yb;<br>41 }  |                                 | ● 40 ++V7;<br>● 41V8:  |                            |
|  | 42 while ( v6 );   |                                 | 42 }   |                            |
|  |  |                                 |  |                            |
| Ϋ́ς Ι  | 00016255[808_1800161409:38 (180016E55)]  |                                 | 00013845[BUD_180014AF8:39 (180014BF5)  |                            |

Pangolin8RAT.FileMgr vs. FFRAT Code overlap just change XOR key: 0xBC vs. 0x57

#### Code similarity - Dead Drop Resolver technique

- Step1: Get response from web server (first-stage c2)
- Step2: Parse encrypted/encoded string with hardcoded delimiters
  - Format: <start\_delimiter>binary\_data<end\_delimiter>
- Step3: covert data to bytes and decode string
  - C2 Format: "<ipv4 or domain>:<port>"
- Step4: resolve the second-stage C2 ip address

![](_page_35_Figure_8.jpeg)

| TDA - 0879125ed34df60a70ed       | 15bb8d58f3a19          | C\Users\user\Desktop\p8rat\0879125ed34df60a70ed5bb8d58f3a19   |                | 10A - 1962a69c204289cb8214      | a30c15f0560            | 09 C/Users/user/Desktop/FFRAT/1962a69c204289cb8214a30c15f05609                              | - • ×     |
|----------------------------------|------------------------|---|----------------|---------------------------------|------------------------|---|-----------|
| File Edit Jump Search V          | liew Debugge           | er Lumina Options Windows Help  |                | File Edit Jump Search Vi        | iew Debug              | gger Lumina Options Windows Help  |           |
| 📑 🚍 ( e - e) - 1 fin fin         | 6. 6. 1                | 👩 : 🔽 🌒 : 📾 📾 🔊 👉 - 🖈 🗹 🗙 i 🕨 🔲 🔲 No debugger   | · 🐜 🛃 👫 👫 🏁    | 📑 🚍 🧄 - 🚽 - 1 🌆 🖓               | 🦀 🕾 1                  | 🗙 🧔 🖉 🕘 🖬 📾 🕼 🦿 🖉 🚽 🗹 🗙 🕨 🔲 🔲 No debugger 💿 👻 🦏 🛃 🔐   | 🕂 🕬       |
|                                  |                        |   |                |                                 |                        |   |           |
| 📕 Library function 📃 Regular fun | etion 📕 Instructio     | on 📄 Data 📕 Unexplored 📕 External symbol 📕 Lumina function  |                | Library function 📃 Regular func | rtion 📕 Instru         | uction 📃 Data 🔤 Unexplored 🗾 External symbol 🔛 Lumina function                              |           |
| Functions 🗖 🗗 🗙                  | [ I 🗵                  | 📑 P··· 🗷 📣 F··· 🗷 🛐 ··· 🗷 🚺 F··· 🗷 🚺 ··· 🗷 🛗 F··· 🗷   | 🖬 I 🔟 🚮 E- 4 🕨 | Functions 5 ×                   | 📑 IDA…                 | 🛛 🚺 Pasud 👓 🗷 🐨 Strings 🕶 🗷 💽 Herr 🗷 🚺 Srr 🗷 🗮 Err 🗶 🏹 Irr 🕱                                | 🛃 E… 🗵    |
| Function name                    | • 107 v1               | <pre>17 = std::_Traits_find<std::char_traits<char>&gt;(v14, v40, 0i64, "1BFF2C7</std::char_traits<char></pre> | 7", 8164);     | Function name                   | 36 {                   |   |           |
| acrt InitializeCriticalSec       | 108 11<br>109 <i>1</i> | ( VI/ := 0xfffffffffffffff)   |                | Chsize polock                   | 37                     | start = strstr(v4, FCS19892);<br>v7 = start:  |           |
| f act kValidLocaleName           | • 110                  | v18 = v17 + 8;  |                |                                 | <ul> <li>39</li> </ul> | if (start)  |           |
| act I CIDTol ocaleNem            | • 111                  | v31 = v18;  |                | j _seck_noisek                  |                        |   |           |
|                                  | 112                    | v19 = Block;  |                | 5 ml 180030014                  | <b>41</b>              | <pre>end = strstr(start, "6419ED17");</pre>   |           |
| # _acrt_LCMapStringEx            | 113                    | 11 ( V16 >= 0×10 )<br>v19 = v15:  |                | f sub_180029914                 | • 42<br>• 43           | v9 = end;<br>if ( end )   |           |
| acrt_LocaleNameToLCI             | 0 115                  | <pre>v20 = std:: Traits find<std::char traits<char="">&gt;(v19, v11, v18, "3381ED</std::char></pre>           | FE", 8164):    | f _tzset_nolock                 | 44                     |   |           |
| acrt_can_use_vista_local         | 0 116                  | if ( v20 != 0xFFFFFFFFFFFFFFFii64 )   |                | f cvtdate                       | 0 45                   | v10 = v7 - v5 + strlen("FC519892");   |           |
| 🗲acrt_initialize_winapi_th       |                        |   |                | 🗲 _isindst_nolock               | 9 46                   | pszString = 0;  |           |
| 🗲acrt_is_packaged_app            | 118                    | sub_18000C90C(Block, Src, v31, v20 - v31);  |                | f _tzset                        | 47                     | memset(v21, 0, sizeof(v21));  |           |
| 🗲acrt_uninitialize_winapi        | 119                    | std::string::string(V32);   |                | 📝 _isindst                      | 40                     | obBinary = 0:   |           |
| 🗲 _fcloseall                     | 0 121                  | sub 180015484(Src);   |                | F sub_18002A3BC                 | 0 50                   | <pre>memset(v19, 0, sizeof(v19));</pre>   |           |
| 🖌 acrt stdio free buffer n       | 0 122                  | v33 = 0164;   |                | 7 sub 18002A3FC                 | • 51                   | <pre>pcbBinary = 0x104;</pre>   |           |
| f acrt stdio free stream(        | 0 123                  | v34 = 0164;   |                | sub 18002443C                   | 52                     | <pre>v11 = strlen(&amp;pszString);</pre>  |           |
| <pre></pre>                      | 124                    | <pre>std::string::_Construct_lv_contents(v32, Src); cub_199915508(v37);</pre>                                 |                | 7 sub 18002447C                 | 53                     | CryptStringToBinaryA(&pszString, v11, 1u, &pbBinary, &pcbBinary, 0i64, 0i64);               |           |
|                                  | 125                    | v21 = 0164:   |                | 7 sub_10002447C                 | 55                     | $v_{12} = 0.04;$<br>$v_{17} = 0.04;$  |           |
| f sub_100037000                  | 0 127                  | v30 = 1;  |                | 7 SUD_10002A404                 | • 56                   | while ( v12 < strlen(&pbBinary) )   |           |
|                                  | 128                    | for ( j = Size[0]; v21 < Size[0]; j = Size[0] )   |                | 7 sub_18002A48C                 |                        |   |           |
| <u>f</u> iswctype                | 129                    |   |                | <u>f</u> sub_18002A494          | 58                     | *(&pbBinary + v12++) ^= 0x57u;  |           |
| 🗲 _realloc_base                  | 130                    | V23 = V37;<br>if ( Simplify a going )   |                | 🚰 _toupper_l                    | • 59<br>60             | v17 = v12;  |           |
| 🖌 _isctype_l                     | 0 132                  | $v_{23} = v_{37}[0];$   |                | 📝 toupper                       | 60<br>61               | $v_{13} = 0x40i64$ :  |           |
| 📝 fegetround                     | • 133                  | *(v23 + v21++) ^= 0, 4Fu;   |                | 📝 _getbuf                       | 62                     | <pre>memset(lpThreadParameter + 0x210, 0, 0x40ui64);</pre>                                  |           |
| 🖌 _dsign                         |                        |   |                | 🖌 _isatty                       | 0 63                   | if ( strlen(&pbBinary) <= 0x40 )  |           |
| 🗲 _errcode                       | 135                    | <pre>memset(v2, 0, 0x100ui64);<br/>u24 = u27</pre>  |                | 🕝 _initp_misc_cfltcvt_tab       | 64                     | v13 = strlen(&pbBinary);  |           |
| f except1                        | 137                    | $v_{24} = v_{37}$ ;<br>$v_{25} = v_{37}$ [0]:   |                | get printf count outpu          | 66                     | if ( =(]oThreadParameter + 0x18()   |           |
| f handle exc                     | 0 138                  | v26 = Size[1];  |                | f wctomb s l                    |                        | <pre>   resolve_c2_ip(lpThreadParameter + 0x210, lpThreadParameter + 0x194, lpThreadP</pre> | Parameter |
|                                  | • 139                  | if ( Size[1] >= 0x10 )  |                | y _incomb_s_                    |                        |   |           |
|                                  | 0 140                  | v24 = v37[0];   |                |                                 | 69                     | v2 = 1;   |           |
| Line 900 of 1065                 | 141                    | <pre>it ( ) &gt; 0x100 ) invalid parameter mointo noneturn();</pre>   |                | Line 673 of 762                 |                        |   |           |
|                                  | 143                    | memmove(v2, v24, i):  |                |                                 |                        |   |           |
| Graph ov: 🗆 🖶 🗙                  | 144                    | if ( !*(lpThreadParameter + 0x1FB) )  |                | An Graph ov:                    |                        |   |           |
|                                  |                        |   |                |                                 | 074 if                 |   |           |
|                                  | 146                    | v30 = resolve_c2_ip(v2, lpThreadParameter + 0x1C4, lpThreadParameter + 0x1C4, lpThreadParameter               | er + 0x38A);   | <b>1</b>                        | 75                     | Tree(VS);   |           |
|                                  | • 148                  | $v_{25} = v_{37}[0];$   |                |                                 | 0 77 }                 | curri vz,   |           |
| 1                                |                        |   |                |                                 |                        |   |           |
| <u>3</u>                         | 00014                  | E9E[StartAddress:133 (180015A9E)]   |                |                                 | 0000                   | DEDE7 [sub_18000783C:58 (1800079E7)]  |           |

Pangolin8RAT.FileMgr vs. FFRAT Different hardcoded delimiters and XOR key: 0xAF vs. 0x57

![](_page_37_Figure_1.jpeg)

#### Phantom DLL hollowing

• ChatLoader<sup>[2]</sup> (aka. StealthVector)

#### Anti IDA Pro decompiler

- The linux variant of Natwalk
  - Specter botnet<sup>[3]</sup> is the predecessor of Natwalk.linux

#### **Dead Drop Resolver**

- Natwalk<sup>[2]</sup> (aka. Sidewalk<sup>[5]</sup>, ScrambleCross<sup>[9]</sup>)
  - Natwalk is one of the backdoors loaded by the ChatLoader
- KeyPlug<sup>[4]</sup> (tech community forums)
- ShadowPad (MSDN forums, github), PlugX(MSDN forums, pastebin)
- Winnti<sup>[13]</sup> (MSDN forums), FFRAT
- 9002 RAT

#### **KCP** Protocol

- KeyPlug
- Crosswalk<sup>[6]</sup>
- FunnySwitch<sup>[6]</sup> (unused)
- PseudoManuscrypt<sup>[8]</sup>(unknown adversary)

## Multiple c2 protocol supported & Modular designed

- KeyPlug (HTTP, KCP, TCP, WSS)
- Crosswalk (TCP, HTTP, KCP)
- FunySwitch (RPC, TCP, HTTP)
- Winnti (ICMP, UDP, TCP, Reuse port)
- PlugX<sup>[7]</sup> (DNS, ICMP, HTTP, TCP, UDP), ShadowPad<sup>[7]</sup> (TCP, UDP, HTTP, DNS)

![](_page_40_Figure_1.jpeg)

#### Targets online gaming/gambling industry

- Natwalk, Crosswalk, FunnySwitch, Spyder
- ShadowPad, Winnti, PlugX
- KeyPlug

#### CobaltStrike technique

- Abusing Cloudflare Workers to hide the real IP address
- Modify XOR-key
- Early bird code injection

![](_page_42_Figure_0.jpeg)

## The New Era of Chinese APT analysis?

- Increasing intricacy of malware families
- Increasing tendency of malware sharing

Malware-as-a-Service among APT groups?

![](_page_43_Picture_4.jpeg)

# 3. Tianwu

TTPs, Activity Timeline, Target, Attribution

![](_page_44_Picture_2.jpeg)

![](_page_45_Picture_0.jpeg)

- A beast with 8 human heads, 8 feet and 8 tails
  - Modular features of Pangolin8RAT
  - Amalgamation of different groups of actors
- The Classic of Mountains and Seas (山海經)

## **Target Industry and Region**

![](_page_46_Figure_1.jpeg)

Gambling Telecom Government Transport Dissident Other Taiwan The Philippines Kazahstan China Hong Kong Other

## **Activity Timeline**

![](_page_47_Figure_1.jpeg)

## TTPs

#### **Delivery Method**

• Social engineering, forum phishing, planting backdoor in NAS server

#### Malware abused

• Pangolin8RAT, custom CobaltStrike Beacon

#### **C2**

- C2 disguised as legitimate websites
- C2 hosted on VPS
- Recent C2 activity indicated possible abuse of Log4j

#### **Exploit**

- CVE-2022-24934
  - Exploit of WPS Office updater (wide Chinese user base)
- Possible Chromium exploit

## Case Study: Months-long campaign against KZ Telecom

#### Victim

Kazakhstan telecom

First attack spotted in 2021/10, latest attack spotted in 2022/01

#### Tools

- Pangolin8RAT
- CobaltStrike Beacon with specific watermark

#### C2

- C2 domain disguised as the victim's domain
- VPS provided by Leaseweb

![](_page_49_Figure_10.jpeg)

## Case Study: Months-long campaign against TW gambling firm

#### Victim

• Taiwanese gambling firm

First attack spotted in 2021/04, latest attack spotted in 2022/02

Tools

- Pangolin8RAT
  - hijacking, pipeline operation, local shellcode injection
  - Evaded detection of multiple anti-virus software
- CobaltStrike Beacon with specific watermark
- Hacking tool
  - Attempts to collect info of victims' browser and messaging software

![](_page_50_Picture_11.jpeg)

## Case Study: Attack against TW transport industry

#### Victim

Taiwanese public transport-related firm

Time: 2021/08

Tools

- Pangolin8RAT
- 8 C2 configs were populated in the RAT

#### C2

- Disguised as the enterprise management software used by the victim
- Registered with TUCOWS and shielded by privacy protection
- C2 infra also used in attack against PH gambling firms

![](_page_51_Figure_11.jpeg)

## Case Study: Campaign against Chinese-speaking dissident

#### Victim

- Chinese-speaking dissident
- Time: 2021/03-2021/04

Delivery

- Phishing via Forum
- Disguised as TW IT Company

|   | 论坛 导读 家园   | │ 排行榜 │ 新闻子站 │ 评论子站 │ 生活子站 │ 娱乐子站 │  |
|---|--|--------------------------------------|
|   | 查看: 656   回复: 4  | [复制链接]                               |
|   |  | 📃 发表于 2021-3-6 16:20:23   只看该作者 ▶    |
| 2         5         37           主题         帖子         积分 |  | 您好,我是                                |
| 2   | 新手上路   | "How to put ads on                   |
| 2   | 和分 37  | abuoluowang.com"                     |
| 0   | ‱收听TA I 发消息  |                                      |
| die.  |  |                                      |
|   |  |                                      |
|   |  | ▲ 楼主   发表于 2021-3-6 16:21:16   只看该作者 |
|   | 2       5       37         主题       站子       积分         新手上路       ☆ | 快來聯係我啊 工作人員                          |

## Case Study: Campaign against Chinese-speaking dissident (cont.)

#### Exploit

- Possible Chromium exploit targeting Chromium-based browser users
  - eg: QQ browser

#### Tools

- Malicious WeChat CRX (Chrome extension)
  - Pangolin8RAT
  - CobaltStrike

## **Directory listing for /**

- <u>bak.zip</u>
- <u>c2.zip</u>
- <u>c2\_x32.zip</u>
- load.html
- <u>notepad.zip</u>
- <u>p.py</u>
- pio.html
- vuln.html
- <u>xyz.html</u>

#### C2 with an open directory

## **Tianwu and Amoeba overlaps**

![](_page_54_Picture_1.jpeg)

![](_page_54_Picture_2.jpeg)

Delivery Method

- Forum phishing, planting backdoor in NAS devices
- Malware feature
  - KCP protocol
  - Utilization of multiple C2 protocols
  - Phantom DLL hollowing

#### • C2

- Abusing Cloudflare Workers to hide the real IP address
- Target Scope
  - Interests in online gaming/gambling industry

## **Attribution: Another Amoeba?**

Possible scenarios:

Amalgamation of civilian hackers

- Operation mode like Chengdu404
- Operate bid projects of the national/public security agencies
- Motive: espionage, domestic surveillance

## Subgroup of Amoeba

No shared infra and tools detected so far

## **Open Directory**

#### Information collected

- Staffs and operators' personal info
- Credentials
- Software source code
- Business info

# Threat Landscape: New APT Operation Mode

Difficulty of pinning down actors' motive

- Target scope spans different industry
- Espionage operations outsourced by MSS/MPS?

Chinese authorities' crackdown on online gaming/gambling industry

- Abundant money and data (personal info and cash flow)
- Data collection for authorities' crackdown campaign

Civilian hacker/front company aiming for personal gain

- Participation in cybercrime
- Software source code for sale in underground market

## Tianwu's Operations in Diamond Model

#### Technical Axis

- Tools and TTPs resemble APT41
- Proprietary malware possibly developed by the developers of Winnti and FFRAT

#### Capability

#### Tools:

- Pangolin8RAT
- Custom CobaltStrike Beacon TTPs:
- Social Engineering
- Planting backdoor in NAS server
- Malware with modular feature and KCP protocol
- Exploit: WPS Office, Chromium

#### Adversary

![](_page_58_Picture_13.jpeg)

Tianwu Origin:China

#### Social-Political Axis

- China's crackdown on its domestic gaming industry
  - Data collection of service
     providers
- China's crackdown on Macau gambling industry forced gambler move online
  - Data collection of gamblers and cash flow

#### Infrastructure

- C2 disguised as legitimate websites
- C2 hosted on VPS
- Abused Cloudflare Workers to hide the real IP address
- Recent C2 activity indicated possible abuse of Log4j
- Geography: Taiwan, the Philippines, Kazakhstan, Hong Kong, China
- Victim
- Industry: Gambling, gaming, IT, telecom, gov, transport, dissident

# 4. Conclusion

Outlook and suggestions

## **Conclusion and Outlook**

Pangolin8RAT could be the next gen PlugX/ShadowPad

- Modular-featured RATs become more popular
- Highly possible to be shared or even sold among Chinese threat groups
- Both espionage and financially driven operations

![](_page_60_Picture_5.jpeg)

## **Conclusion and Outlook**

New mode of APT operations

- Trends of malware sharing
- Malware with similar structure and techniques
- Malware-as-a-Service among APT groups

Tianwu might operate as: a collaborator of APT41, a subgroup of APT41, or a digital quartermaster of Chinese APTs

![](_page_61_Picture_6.jpeg)

## Countermeasures

Defend your organization with all-level Intelligence

- Tactical
  - Feed CTI vendor's IoCs to cybersecurity infra
- Operational
  - Patch servers in timely manner
  - Beware of new social engineering tactics
  - Apply in-memory detection
- Strategic
  - New operation mode of Chinese APTs makes attribution/group tracking more difficult

- China's policies/crackdown heavily affects cyberspace in APAC region

![](_page_62_Picture_11.jpeg)

## loC

#### Pangolin8RAT

- 0f44724d498f77a59bc542be7d17dc89
- 47b3627c3900e29bdef6d36cfdf61bbf
- ea76ad28a3916f52a748a4f475700987
- cfae9252291fdf63f0c3d485a162a444
- bfa657d3eca9df2b122d0908ac23c1ed
- 4fb9b38e9c4b3c98b6f13c153bbe6f6a
- bf421d42174edb2f31007cbede9cf5b9
- 8b6a63e522fd6b3f23f476a101720bf9
- ea2e29b351d4e07460e5955b8e1b4d5d
- 641d23463a53bcb29673d179379e1a8f
- 81d9be954a09774887eb75b5a23db9b4
- 9c4df895509a8906a09be0b19bf5c05a

#### CobaltStrike

- 3e08c0e69fc1bbd36b2bb09086fd30ad
- c4e31051dc80d87927d15d0fbed704d0
- 544a7746c87698665744520820551750

## loC

- www.tiger266[.]com
- help.tiger266[.]com
- new.mkdjgame[.]com
- help.mkdjgame[.]com
- www.ffyl-bet[.]com
- help.ffyl-bet[.]com
- zk.full-subscription[.]com
- cs.full-subscription[.]com
- yd.full-subscription[.]com
- www.animal777[.]com
- time.daytimegamers[.]com
- themerecord[.]com

- static.daytodayup[.]com
- mirrors.centos.8788912[.]com
- stat.8788912[.]com
- login.good-enough-8fe4[.]com
- cdn2.twmicrosoft[.]com
- cdn.1685810[.]com
- static.1685810[.]com
- cachedownload.goldenrose88[.]com
- backup.microsupdate[.]com
- api.gpk-demo[.]com
- static.gpk-demo[.]com
- api.geming8888[.]com

- 23.106.122[.]171
- 23.106.123[.]134
- 23.106.124[.]156
- 23.106.125[.]132
- 45.153.242[.]41
- 74.119.193[.]139

## Reference

- 1. Operation Dragon Castling: APT group targeting betting companies, March 22, 2022 <u>https://decoded.avast.io/luigicamastra/operation-dragon-castling-apt-group-targeting-betting-companies/</u>
- 2. Evolution after prosecution: Psychedelic APT41, November 27, 2021 https://vblocalhost.com/uploads/2021/09/VB2021-12.pdf
- 3. Ghost in action: the Specter botnet, September 25, 2020 https://blog.netlab.360.com/ghost-in-action-the-specter-botnet/
- 4. Does This Look Infected? A Summary of APT41 Targeting U.S. State Governments, March 08, 2022 https://www.mandiant.com/resources/apt41-us-state-governments
- 5. The SideWalk may be as dangerous as the CROSSWALK, August 24, 2021 https://www.welivesecurity.com/2021/08/24/sidewalk-may-be-as-dangerous-as-crosswalk/
- 6. Higaisa or Winnti? APT41 backdoors, old and new, January 14, 2021 https://www.ptsecurity.com/ww-en/analytics/pt-esc-threat-intelligence/higaisa-or-winnti-apt-41backdoors-old-and-new/
- 7. SHADOWPAD: A MASTERPIECE OF PRIVATELY SOLD MALWARE IN CHINESE ESPIONAG, August, 2021 <u>https://www.sentinelone.com/labs/shadowpad-a-masterpiece-of-privately-sold-malware-in-chinese-espionage/</u>

## Reference

- 8. PseudoManuscrypt: a mass-scale spyware attack campaign, December 16, 2021 <u>https://ics-cert.kaspersky.com/publications/reports/2021/12/16/pseudomanuscrypt-a-mass-scale-spyware-attack-campaign/</u>
- 9. Earth Baku Returns, August 24, 2021 https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/earth-baku-returns
- 10. Delving Deep: An Analysis of Earth Lusca's Operations , January 17, 2022 <u>https://www.trendmicro.com/content/dam/trendmicro/global/en/research/22/a/earth-lusca-employs-</u> <u>sophisticated-infrastructure-varied-tools-and-techniques/technical-brief-delving-deep-an-analysis-of-</u> <u>earth-lusca-operations.pdf</u>
- 11. This Is Not a Test: APT41 Initiates Global Intrusion Campaign Using Multiple Exploits, March 25, 2020 https://www.mandiant.com/resources/apt41-initiates-global-intrusion-campaign-using-multiple-exploits
- 12. LOWKEY: Hunting for the Missing Volume Serial ID, October 15, 2019 https://www.mandiant.com/resources/lowkey-hunting-missing-volume-serial-id
- 13. "Winnti" More than just a game, April 11, 2013 <u>https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/20134508/winnti-more-than-just-a-game-130410.pdf</u>

# Thank you!

![](_page_67_Picture_1.jpeg)

Persistent Cyber Threat Hunters