

[返回 TI 主页](#)

RESEARCH

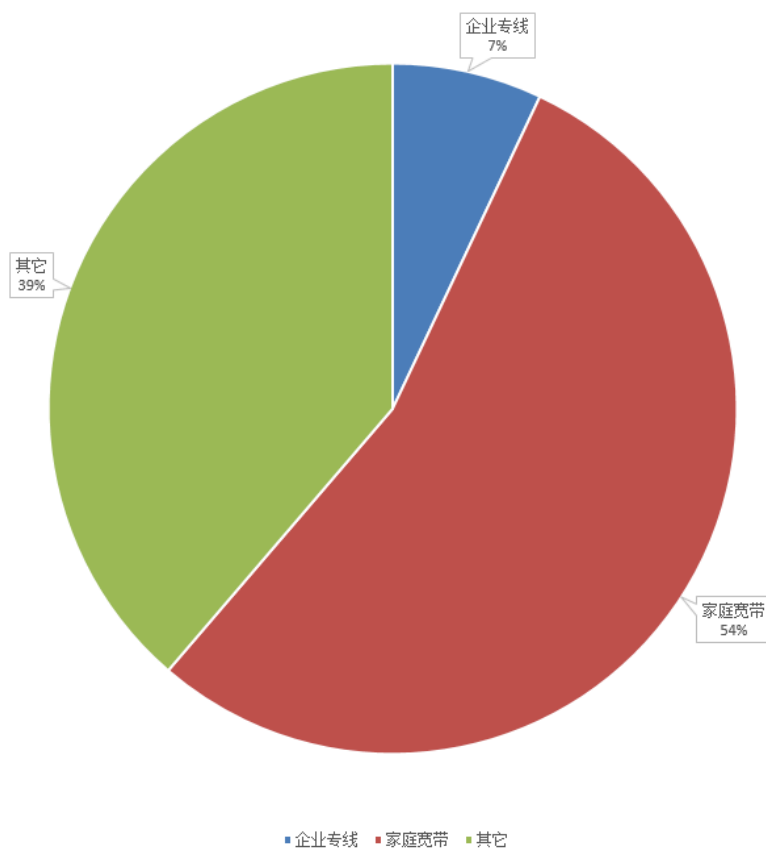
数据驱动安全

Background

Recently, the QiAnXin Threat Intelligence Center's Red Raindrop team has discovered a previously undisclosed special Trojan, named "SetcodeRat", that is being massively spread in the Chinese region during the operation of the private intelligence production process. The Trojan has a built-in customization for Telegram, and the attacker has successfully attracted our attention by spreading the Trojan against TGs in TG-blocked areas.

According to our observations, the earliest activity of SetcodeRat can be traced back to October 2025. It is mainly suspected to be spread through SEO means, and we have not found any other delivery methods. The malicious installation package will first verify the region of the victim. If it is not in the Chinese-speaking area, it will automatically exit. In just one month, the Trojan has infected hundreds of computers, affecting some government and enterprise units.

受害者网络属性分布图



QiAnXin Tianqing's "Liuhe" engine can now block this trojan by default. We recommend government and enterprise clients deploy the "Liuhe" engine to defend against unknown threats.



Sample Analysis

The sample is disguised as a normal software installation package.

 chrome-stallerwin64.exe	2025/11/10 15:45	应用程序	172,038 KB
 Huaseoaxz-youdao navig-tion.exe	2025/11/16 21:39	应用程序	148,089 KB

After the victim downloads and runs it, the malicious sample will first screen domestic users according to the language settings, and continue to execute when the settings are in Chinese, involving mainland China, Hong Kong, Macao and Taiwan.

Zh-CN Mainland China

Zh-TW Taiwan, China

Zh-HK Hong Kong, China

Zh-MO Macao, China

```

if ( !UILanguages._r0.len )
    runtime_panicIndex();
if ( UILanguages._r0.array->len == 5
    && ((str = UILanguages._r0.array->str, *(_DWORD *)UILanguages._r0.array->str == 'C-hz') && str[4] == 'N'
    || *(_DWORD *)str == 'T-hz' && str[4] == 'W'
    || *(_DWORD *)str == 'H-hz' && str[4] == 'K'
    || *(_DWORD *)str == 'M-hz' && str[4] == 'O') )
{

```

Secondly, it will access the bilibili related api and terminate the execution if the access is unsuccessful.

```

p_http_Client = (http_Client *)runtime_newobject((internal_abi_Type *)&RTYPE_http_Client);
p_http_Client->Timeout = 1000000000LL;
v18.str = (uint8 *)"https://api.bilibili.com/x/report/click/now";
v18.len = 43LL;
v19 = net_http_ptr_Client_Get((net_http_Client *)p_http_Client, v18);
if ( v19.err.tab )
{

```

After passing the verification, it will release the white plus black component named pnm2png.exe.

```

{
    v1.str = (uint8 *)"pnm2png.exe";
    v1.len = 11LL;
    main_KAMMZMdf(v1);
    main_Un7z();
    os_Exit(3LL);

```

Among them, zlib1.dll is the malicious loader and qt.conf is the encrypted payload.

 api-ms-win-crt-time-l1-1-0.dll	2025/11/10 15:47	应用程序扩展	14 KB
 api-ms-win-crt-utility-l1-1-0.dll	2025/11/10 15:47	应用程序扩展	12 KB
 libpng16-16.dll	2025/11/10 15:47	应用程序扩展	241 KB
 pnm2png.exe	2025/11/10 15:47	应用程序	93 KB
 qt.conf	2025/11/10 15:47	CONF 文件	7,070 KB
 vcruntime140.dll	2025/11/10 15:47	应用程序扩展	108 KB
 vcruntime140_1.dll	2025/11/10 15:47	应用程序扩展	50 KB
 zlib1.dll	2025/11/21 10:34	应用程序扩展	7,214 KB

When released, it first generates a .TMP configuration file based on the device uuid and writes it to the C:\ProgramData folder.

```

v1 = main_randStr(5LL);
elem[1] = 14LL;
elem[0] = "C:\\ProgramData";
elem_16 = v1;
v15.array = (string *)elem;
v15.len = 2LL;
v15.cap = 2LL;
v2 = path_filepath_join(v15);

```

The pnm2png.exe component is released to a randomly named folder under C:\ProgramData, and when it is released, it is executed by the cmd with a random parameter, after which it performs its fake normal program installation process.

```

main_zBomvNKD(v4, v14, *(string *)&v13.len);
v6 = main_randStr(5LL);
v13.array = &v6;
v13.len = 1LL;
v13.cap = 1LL;
v5 = os_exec_Command(v3, v13); // "C:\\ProgramData\\MCsot\\pnm2png.exe 71Drm"
os_exec_ptr_Cmd_Run((_ptr_exec_Cmd)v5);

```

The pnm2png.exe run loads zlib1.dll, which when loaded modifies the pnm2png.exe entry point to the gsgweTyv function.

```

ModuleHandleA = GetModuleHandleA(0LL);
v1 = *((int *)ModuleHandleA + 15);
*(_DWORD *)((char *)&v9 + 6) = 0xE0FF0000;
v2 = *(unsigned int *)((char *)ModuleHandleA + v1 + 40);
v8 = -18360;
v3 = (char *)ModuleHandleA + v2; // EntryPoint
*(_QWORD *)&v9 = gsgweTyv;
VirtualProtect((char *)ModuleHandleA + v2, 0xCuLL, 4u, flOldProtect);
v4 = v3;
do
{
    v5 = (v4++)[(char *)&v8 - v3];
    *(v4 - 1) = v5;
}
while ( v4 != v3 + 12 );
return VirtualProtect(v3, 0xCuLL, flOldProtect[0], flOldProtect);

```

The main function of gsgweTyv is to decrypt qt.conf and load it.

```

v9 = os_Executable(a1, a2, a3, a4, a5, a6, a7, a8, a9);
v27[1] = a2;
v27[0] = internal_filepathlite_Dir(v9, a2, v10, a4, a5, v11, v12, v13, v14);
v27[3] = 7LL;
v27[2] = "qt.conf";
v19 = path_filepath_join((unsigned int)v27, 2, 2, a4, a5, v15, v16, v17, v18);
return os_ReadFile(v19, 2, v20, a4, a5, v21, v22, v23, v24, v26);

```

qt.conf is essentially a zip file with a decryption code of RPNmUcoToJo5eR7AyHaQ.

```

v20 = github_com_bodgit_sevenzzip_NewReaderWithPassword(
    (unsigned int)off_7FF92EB30D60,
    (_DWORD)p_bytes_Reader,
    v16,
    (unsigned int)"RPNmUcoToJo5eR7AyHaQ",
    20,
    v12,
    v13,
    v14,
    (_DWORD)v15);
if ( v18 )
    return 0LL;

```

The decrypted payload is a dll file with its own loader. After gsgweTyv finishes loading, it jumps to the payload and finally loads the payload, client.dll, and executes the export function CodeRun.

```

if ( a3 == v90 && v89 )
{
    ((void (__fastcall *))(__int64, _QWORD, _QWORD, __int64))(v43
        + *(unsigned int *)v43
        + (unsigned int)v85[7]
        + 4LL * v89)) // go client.dll

    v87,
    a3,
    a6,
    a5);
return v43;

```

CodeRun first checks if the current module is Telegram.exe, if so, it executes Telegram plugin mode.

Telegram plugin function

The hook is first prepared for initialization.

```
result = main_Plugins_TGMsgHook_hook_MH_Initialize(a1, (__int64)a2, a3, (__int64)a4, (__int64)a5, a6, a7, a8, a9);
if ( !result )
{
    initd = main_Plugins_TGMsgHook_util_InitUtil(0LL, (__int64)a2, v10, (int)a4, (int)a5);
    main_Plugins_TGMsgHook_util_InitPackMap(initd, a2, v12, a4, a5);
    main_Plugins_TGMsgHook_userInfo_InitApplication();
}
```

The hook removes proxy hint related functions.

```
SignatureModule = main_Plugins_TGMsgHook_util_FindSignatureModuleEx(
    (int)"48 89 ?? ?? ?? 48 89 ?? ?? ?? 48 89 ?? ?? ?? 55 56 57 41 54 41 55 41 56 41 57 48 88 EC 48 ?? "
    "?? ?? 4D 8B F0 4C 8B FA 45 33 E4 44 89 ?? ?? 0F 57 C0 33 C0 0F 11 02 48 89 ?? ?? 48 89 02 48 "
    "?? ?? ?? ?? ?? ?? 48 89 ?? ?? 44 89 ?? ?? BF ?? ?? ?? ?? 89 7D ?? 41 0F ?? ?? ?? 88 42 ?? 45 "
    "39 20 0F 95 C0 88 42 ?? 45 38 ?? ?? 74 ?? 45 38",
    326LL,
    a3,
    a4,
    a5,
    a6,
    a7,
    a8,
    a9);
return main_Plugins_TGMsgHook_hook_MH_CreateHook(// main_RunMsgHook_RemoveProxyTips_func1
    SignatureModule,
    (__int64)&RTYPE_userinfo_Hooked_computeLayoutType,
    a1,
    (unsigned __int64)&qword_7FF932BE5CE0,
```

Set up a message hook with callbacks to receive, send and read/write data for stealing virtual currency.

```
v12 = (void (__golang *)(_ptr_PacketControl_RequestPackData))runtime_newobject((const RTYPE *)&unk_7FF931DB7CE0);
*(__QWORD *)v12 = main_Plugins_TGMsgHook_PacketControl_ptr_MsgHack_StartWrite_fm;
if ( dword_7FF932BE6430 )
{
    v12 = (void (__golang *)(_ptr_PacketControl_RequestPackData))runtime_gcWriteBarrier1(v12, 0LL);
    v13 = (__int64)v30;
    *v14 = v30;
}
else
{
    v13 = (__int64)v30;
}
*((__QWORD *)v12 + 1) = v13;
v15 = (__int64)main_Plugins_TGMsgHook_PacketControl_NewRequestPack(v12, 0LL, v13, a4, a5);
if ( dword_7FF932BE6430 )
{
    v15 = runtime_gcWriteBarrier2(v15, 0LL, v30);
    *v17 = v15;
    v17[1] = v18;
}
else
{
    p_Write = &v30->Write;
}
*p_Write = v15;
v19 = runtime_newobject((const RTYPE *)&unk_7FF931DB7CE0);
*v19 = main_Plugins_TGMsgHook_PacketControl_ptr_MsgHack_StartRead_fm;
if ( dword_7FF932BE6430 )
{
    v19 = (__QWORD *)runtime_gcWriteBarrier1(v19, 0LL);
```

This is done by fetching USDT related messages from telegram and replacing them with wallet addresses.

```

SendUSDT = main_Plugins_TGMsgHook_client_ptr_Config_GetSendUSDT*((_QWORD *) (a1 + 32), a2, a3, a4, a5, a6, a7, a8, a9);
if ( a4 )
    return a3;
if ( !SendUSDT )
    return a3;
v16 = v10;
v18 = *((_QWORD *) (a3 + 56));
v19 = main_Plugins_TGMsgHook_PacketControl_USDTReplace(
    *((_QWORD *) (a3 + 48)),
    v18,
    a2,
    v16,

```

Match the Tron USDT wallet address in the message using a regular pattern.

```

result = (regex_Regexp *) regex_MustCompile((unsigned int) "T[a-zA-Z0-9]{33,}", 17, a3, a4, a5, a6, a7, a8, a9); // Tron
if ( dword_7FF932BE6430 )
{
    result = (regex_Regexp *) runtime_gcWriteBarrier2(result, 17LL, qword_7FF932B991F8);
    *v10 = result;
    v10[1] = v11;
}
qword_7FF932B991F8 = result;
return result;

```

After that, it connects to C2 (see Remote Control Functions for details on how to get C2) and receives the sent packets.

```

while ( 1 )
{
    Packet = main_Plugins_TGMsgHook_protocol_ptr_ClientConn_ReadPacket(a2, a2, (_DWORD) v9, a4, a5, a6, a7, a8, a9, v23);
    if ( v11 )
        break;
    v24 = Packet;
    v16 = runtime_newobject((const RTYPE *) &unk_7FF931E3AA00);
    *v16 = main_Plugins_TGMsgHook_client_ptr_LinkControl_StartClient_func2;
    if ( dword_7FF932BE6430 )
    {
        v16 = (_QWORD *) runtime_gcWriteBarrier3(v16, a2, main_Plugins_TGMsgHook_client_ptr_LinkControl_StartClient_func2);
        v21 = a1;
        *v20 = a1;
        v20[1] = a2;
        a5 = v28;
        v20[2] = v28;
    }
}

```

Depending on the commands, the functions to be executed include getting user information, setting configuration, message operation and setting agent.

```

if ( v11 > 0x9F87804E )
{
    if ( v11 == 0xC85CA676 )
    {
        if ( (RTYPE **) a2 == off_7FF9322A0360 )
            return main_Plugins_TGMsgHook_client_ptr_LinkControl_GetUserData(a1, a3, a4, a4, a5, a6, a7, a8, a9);
        v10 = a2 == 0;
    }
    else if ( v11 == 0xCE680E4E )
    {
        if ( (RTYPE **) a2 == off_7FF9322A0340 )
            return main_Plugins_TGMsgHook_client_ptr_LinkControl_ConfigSet(a1, a3, a4, a4, a5, a6, a7, a8, a9);
        v10 = a2 == 0;
    }
    else
    {
        v10 = a2 == 0;
    }
}
else if ( v11 == 0x43479015 )
{
    if ( (RTYPE **) a2 == off_7FF9322A0380 )
        return main_Plugins_TGMsgHook_client_ptr_LinkControl_MessageEditorPack(a1, a3, a4, a4, a5, a6, a7, a8, a9);
    v10 = a2 == 0;
}
else if ( v11 == 0x9F87804E )
{
    if ( (RTYPE **) a2 == off_7FF9322A0320 )
        return main_Plugins_TGMsgHook_client_ptr_LinkControl_SetProxy(a1, a3, a4, a4, a5, a6, a7, a8, a9);
    v10 = a2 == 0;
}

```

Among them, the subcommands corresponding to message operation are as follows:

Delete specified message:

```

if ( (*(_QWORD *)v9 == 'eMeteleD' && *(_DWORD *)(v9 + 8) == 'gass' && *(_BYTE *)(v9 + 12) == 'e' )// deletemessage
{
    v15 = *(_QWORD *)(a2 + 8);
    a2 = *(_QWORD *)a2;
    main_Plugins_TGMsgHook_client_ptr_MessageEditor_DeleteMessage(
        *(_QWORD *)(a1 + 40),
        a2,
        v15,
        a4,
        a5,
        a6,
        a7,
        a8,
        v9,
        v23,
        v25,
        v26);
    a1 = v31;
}

```

Set all messages:

```

else if ( (*(_QWORD *)v9 == 'asseMteS' && *(_DWORD *)(v9 + 8) == 'lAeg' && *(_BYTE *)(v9 + 12) == 'l' )// setmessageall
{
    a2 = *(_QWORD *)(a2 + 72);
    main_Plugins_TGMsgHook_client_ptr_MessageEditor_SetMessageAll(
        *(_QWORD *)(a1 + 40),
        a2,
        a3,
        a4,
        a5,
        a6,
        a7,
        a8,
        v9,
        v23,
        v25);
    a1 = v31;
}

```

Delete all messages:

```

else if ( v8 == 16 && *(_QWORD *)v9 == 'eMeteleD' && *(_QWORD *)(v9 + 8) == 'llAegass' )// deleteallmessage
{
    a2 = *(_QWORD *)a2;
    main_Plugins_TGMsgHook_client_ptr_MessageEditor_DeleteMessageAll(*(_ptr_client_MessageEditor *) (a1 + 40), a2);
    a1 = v31;
}

```

Set the specified message:

```

else if ( v8 == '\n' && *(_QWORD *)v9 == 'asseMteS' && *(_WORD *)(v9 + 8) == 'eg' )// setmessage
{
    v10 = *(_QWORD *)(a2 + 8);
    v11 = *(_QWORD *)(a2 + 16);
    a4 = v11;
    v12 = *(_QWORD *)(a2 + 32);
    a5 = *(_QWORD *)(a2 + 24);
    v14 = *(_QWORD *)(a2 + 48);
    v13 = *(_QWORD *)(a2 + 40);
    a2 = *(_QWORD *)a2;
    main_Plugins_TGMsgHook_client_ptr_MessageEditor_SetMessage(

```

The logic of the setup proxy function is as follows, with three subcommands:

1.Get the proxy information from C2 provided by the attacker. 2.Set Proxy 3.Remove proxy

```

if ( *(_WORD *)v10 == 'eG' && *(_BYTE *) (v10 + 2) == 't' )
{
    Proxy = main_Plugins_TGMsgHook_userInfo_GetProxy(qword_7FF932BE5CC0, (_DWORD)a2, a3, a4, v10, a6, a7, a8, a9);
    p_packet_SetProxy = (packet_SetProxy *)runtime_newobject(&RTYPE_packet_SetProxy);
    p_packet_SetProxy->Type.len = 3LL;
    p_packet_SetProxy->Type.ptr = (char *)"Get";
    v16 = Proxy;
    if ( dword_7FF932BE6430 )
    {
        v23 = p_packet_SetProxy;
        runtime_wbMove(&RTYPE_util_ProxyData, &p_packet_SetProxy->Data, Proxy);
        p_packet_SetProxy = v23;
        v16 = Proxy;
    }
    *(_QWORD *)&p_packet_SetProxy->Data.Type = *(_QWORD *)v16;
    p_packet_SetProxy->Data.Host = *(string *) (v16 + 8);
    *(_QWORD *)&p_packet_SetProxy->Data.Port = *(_QWORD *) (v16 + 24);
    *(_QWORD *)&p_packet_SetProxy->Data.User.len = *(_QWORD *) (v16 + 40);
    *(_QWORD *)&p_packet_SetProxy->Data.Password.len = *(_QWORD *) (v16 + 56);
    main_Plugins_TGMsgHook_protocol_ptr_ClientConn_WritePacket(
        a3,
        (unsigned int)off_7FF9322A0320,
        (_DWORD)p_packet_SetProxy,

else if ( *(_WORD *)v10 == 'eS' && *(_BYTE *) (v10 + 2) == 't' )
{
    if ( !a2[4] )
    {
        p_errors_errorString = (errors_errorString *)runtime_newobject(&RTYPE_errors_errorString);
        p_errors_errorString->s.len = 21LL;
        p_errors_errorString->s.ptr = (char *)&unk_7FF931FC51BC;
        return off_7FF9322A0300;
    }
    main_Plugins_TGMsgHook_userInfo_SetCurrentProxy((_DWORD)a2 + 16, (_DWORD)a2, a3, a4, v10, a6, a7, a8, a9);
}
}
else if ( v9 == '\x0F'
    && *(_QWORD *)v10 == 'rPevomeR'
    && *(_DWORD *) (v10 + 8) == 'Tyxo'
    && *(_WORD *) (v10 + 12) == 'pi'
    && *(_BYTE *) (v10 + 14) == 's' )
{
    v19 = main_Plugins_TGMsgHook_userInfo_SetIsRemoveProxyTips(1, (_DWORD)a2, 'omeR', a4, v10, a6, a7, a8, a9);
    main_Plugins_TGMsgHook_userInfo_ResetProxy(v19);
}

```

Remote Control Function

If the current module is not Telegram.exe, the normal remote control mode is executed, first reading the .tmp files previously generated based on the device uuid.

The content of each .tmp file is the encryption configuration hardcoded in the SEO Poisoning Trojan, which is different for each different SEO Poisoning Trojan.

```

.rdata:0000000141111CBC encode_config db 0B0h ; DATA XREF: .rdata:000000014908B0E8lo
.rdata:0000000141111CBD db 8Ch
.rdata:0000000141111CBE db 10h
.rdata:0000000141111CBF db 10h
.rdata:0000000141111CC0 db 19h
.rdata:0000000141111CC1 db 0Fh
.rdata:0000000141111CC2 db 10h
.rdata:0000000141111CC3 db 11h
.rdata:0000000141111CC4 db 16h
.rdata:0000000141111CC5 db 0Fh
.rdata:0000000141111CC6 db 15h
.rdata:0000000141111CC7 db 14h
.rdata:0000000141111CC8 db 0Fh
.rdata:0000000141111CC9 db 15h
.rdata:0000000141111CCA db 13h

```

Decrypt the encrypted configuration heterodyne 0x21 and get C2 after decryption.

Connect to C2 and perform remote control functions.

```

p_net_TCPAddr->IP.ptr = v17;
v44 = v3;
v19 = (void *)((__int64 (__fastcall *) (__int128 *))loc_2CAF2CC99)(&v45 + 4);
v3 = v44;
v54.LocalAddr.tab = off_2CC283268;
v54.LocalAddr.data = v19;
v65.tab = (void *)&off_2CC29CCB0;
v65.data = &qword_2CCBC5B60;
v63.ptr = "tcp";
v63.len = 3LL;
v64.ptr = ptr;
v64.len = len;
*(retval_2CBD4FCC0 *)(&v2 - 3) = net_ptr_Dialer_DialContext(&v54, v65, v63, v64);
if ( v66.1.tab )
{
    time_Sleep(0xB2D05E00, v66.0.data, v66.1.tab, v2, 3, v20, v21, v22, v23, v45);
}

```

The remote control functions are as follows: screenshot, keystroke logging, reading folders, setting folders, starting processes, running cmd commands, setting socket connections, getting browser logs, getting network connection information, updating the Trojan horse, and getting system information.

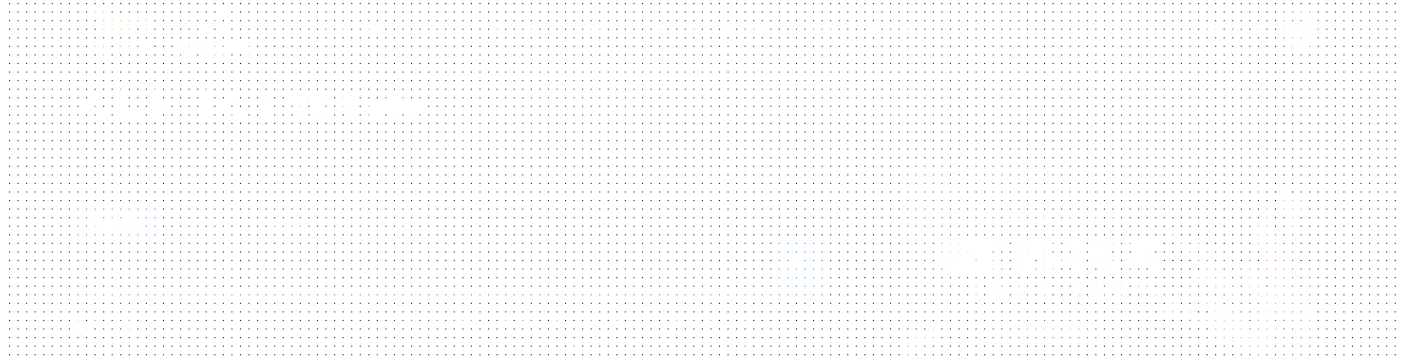
```

p_function_Screen1 = (function_Screen1 *)runtime_newobject(&RTYPE_function_Screen1);
p_function_Keyboard = (function_Keyboard *)runtime_newobject(&RTYPE_function_Keyboard);
p_function_Folders = (function_Folders *)runtime_newobject(&RTYPE_function_Folders);
p_function_FolderSet = (function_FolderSet *)runtime_newobject(&RTYPE_function_FolderSet);
p_function_Processes = (function_Processes *)runtime_newobject(&RTYPE_function_Processes);
p_function_RunCmd = (function_RunCmd *)runtime_newobject(&RTYPE_function_RunCmd);
p_function_Sockets = (function_Sockets *)runtime_newobject(&RTYPE_function_Sockets);
p_function_HackBrowser = (function_HackBrowser *)runtime_newobject(&RTYPE_function_HackBrowser);
p_function_ConnectionInfo = (function_ConnectionInfo *)runtime_newobject(&RTYPE_function_ConnectionInfo);
p_function_UpVersion = (function_UpVersion *)runtime_newobject(&RTYPE_function_UpVersion);
p_function_SystemSettings = (function_SystemSettings *)runtime_newobject(&RTYPE_function_SystemSettings);

```

Summary

Currently, all Qi'anxin products powered by threat intelligence from Qi'anxin Threat Intelligence Center—including Qi'anxin Threat Intelligence Platform (TIP), Tianqing Endpoint Security, Tianyan Advanced Threat Detection System, Qi'anxin NGSOC, and Qi'anxin Situational Awareness—support accurate detection of such attacks.



IOC

Md5

773aae5bd834b3de00f97f2f47204eb6

2273578c084a5730c80e37be276ece90

C2

118.107.45.42:443

xionger.cc

sslIndac.com

ndcwsw.com

38.45.122.162:443

xiongdaylf.com

MALWARE SETCODERAT

分享到：