关于国家授时中心遭受美国国家安全局网络攻击事件的技术分析报告

10月19日上午，国家安全机关披露了美国国家安全局（以下简称NSA）对国家授时中心（以下简称“授时中心”）实施重大网络攻击活动。国家互联网应急中心（CNCERT）通过分析研判和追踪溯源得出此次攻击事件的整体情况，现将具体技术细节公布如下：

一、攻击事件概貌

2022年3月起，NSA利用某国外品牌手机短信服务漏洞，秘密监控10余名国家授时中心工作人员，非法窃取手机通讯录、短信、相册、位置信息等数据。2023年4月起，NSA在“三角测量”行动曝光前，多次于北京时间凌晨，利用在某国外品牌手机中窃取的登录凭证入侵国家授时中心计算机，刺探内部网络建设情况。2023年8月至2024年6月，NSA针对性部署新型网络作战平台，对国家授时中心多个内部业务系统实施渗透活动，并企图向高精度地基授时导航系统等重大科技基础设施发动攻击。

纵观此次事件，NSA在战术理念、操作手法、加密通讯、免杀逃逸等方面依然表现出世界领先水准。**隐匿实施攻击，**NSA通过使用正常业务数字证书、伪装Windows系统模块、代理网络通信等方式隐蔽其攻击窃密行为，同时对杀毒软件机制的深入研究，可使其有效避免检测；**通讯多层加密，**NSA

使用网攻武器构建回环嵌套加密模式，加密强度远超常规TLS通讯，通信流量更加难以解密还原；**活动耐心谨慎，**在整个活动周期，NSA会对受控主机进行全面监控，文件变动、关机重启都会导致其全面排查异常原因；**功能动态扩展，**NSA会根据目标环境，动态组合不同网攻武器功能模块进行下发，表明其统一攻击平台具备灵活的可扩展性和目标适配能力。但其整体创新性缺失和部分环节乏力，显示出在被各类曝光事件围追堵截后，技术迭代升级面临瓶颈困境。

二、网络攻击过程

此次攻击事件中，NSA利用“三角测量行动”获取授时中心计算机终端的登录凭证，进而获取控制权限，部署定制化特种网攻武器，并针对授时中心网络环境不断升级网攻武器，进一步扩大网攻窃密范围，以达到对该单位内部网络及关键信息系统长期渗透窃密的目的。梳理发现，NSA使用的网攻武器共计42款，可分为三类：前哨控守（“eHome_0cx”）、隧道搭建（“Back_eleven”）和数据窃取（“New_Dsz_Implant”），以境外网络资产作为主控端控制服务器实施攻击活动共计千余次。具体分为以下四个阶段：

（一）获取控制权限

2022年3月24日至2023年4月11日，NSA通过“三角测量”行动对授时中心10余部设备进行攻击窃密。2022年9月，攻击者通过授时中心网络管理员某国外品牌手机，获取了办公计算机的登录凭证，并利用该凭证获得了办公计算机的远程控制权限。

2023 年 4 月 11 日至 8 月 3 日，攻击者利用匿名通信网络节点远程登录办公计算机共 80 余次，并以该计算机为据点探测授时中心网络环境。

序号	时间	事件
1	2023/08/03 03:44:54	远程登录
2	2023/08/03 04:00:23	远程登录
3	2023/08/03 04:06:32	网攻武器回连
4	2023/08/03 04:07:04	远程登录
5	2023/08/03 04:19:12	网攻武器心跳回连
6	2023/08/03 04:26:46	远程登录
7	2023/08/03 04:31:41	远程退出
8	2023/08/03 04:58:35	网攻武器心跳回连
9	2023/08/03 05:19:33	网攻武器心跳回连
10	2023/08/03 05:35:44	网攻武器心跳回连
11	2023/08/03 05:45:14	网攻武器心跳回连
12	2023/08/03 05:58:22	网攻武器心跳回连
13	2023/08/03 06:04:55	网攻武器回连并清除退出

表 2023 年 8 月 3 日攻击过程

（二）植入特种网攻武器

2023 年 8 月 3 日至 2024 年 3 月 24 日，攻击者向网管计算机植入了早期版本的“Back_eleven”，窃取网管计算机数据，并在每次攻击结束后清除网络攻击武器内存占用和操作痕迹。该阶段“Back_eleven”功能尚未成熟，攻击者每次启动前需远程控制关闭主机杀毒软件。

序号	时间	事件
1	2023/08/03 03:36:16	关闭杀毒软件
2	2023/08/09 02:50:35	关闭杀毒软件
3	2023/08/23 02:48:29	关闭杀毒软件
4	2023/08/25 02:52:04	关闭杀毒软件
5	2023/08/31 03:30:50	关闭杀毒软件
6	2023/09/15 04:28:58	关闭杀毒软件
7	2023/10/11 02:34:45	关闭杀毒软件
8	2023/10/18 03:19:26	关闭杀毒软件
9	2023/12/29 04:02:49	关闭杀毒软件

表 部分杀毒软件关闭记录

（三）升级特种网攻武器

2024 年 3 月至 4 月，攻击者针对授时中心网络环境，定制化升级网络攻击武器，植入多款新型网络攻击武器，实现对计算机的长期驻留和隐蔽控制。攻击者加载“eHome_0cx”“Back_eleven”“New_Dsz_Implant”，配套使用的 20 余款功能模块，以及 10 余个网络攻击武器配置文件。

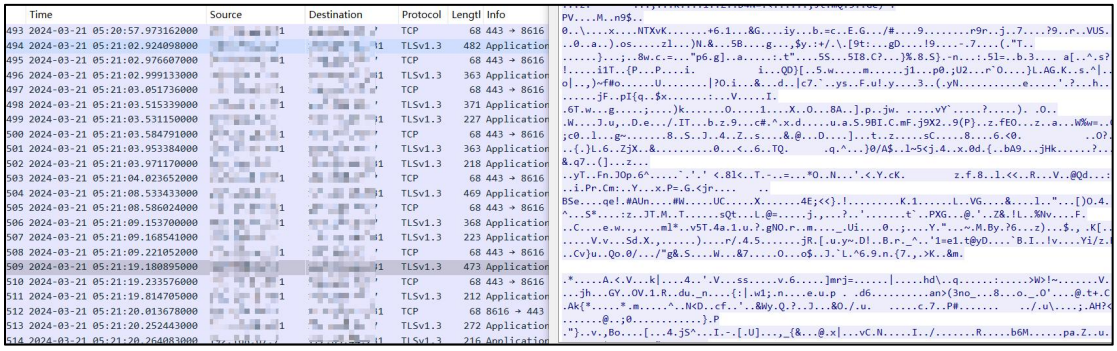

图 加载“eHome_0cx”数据包

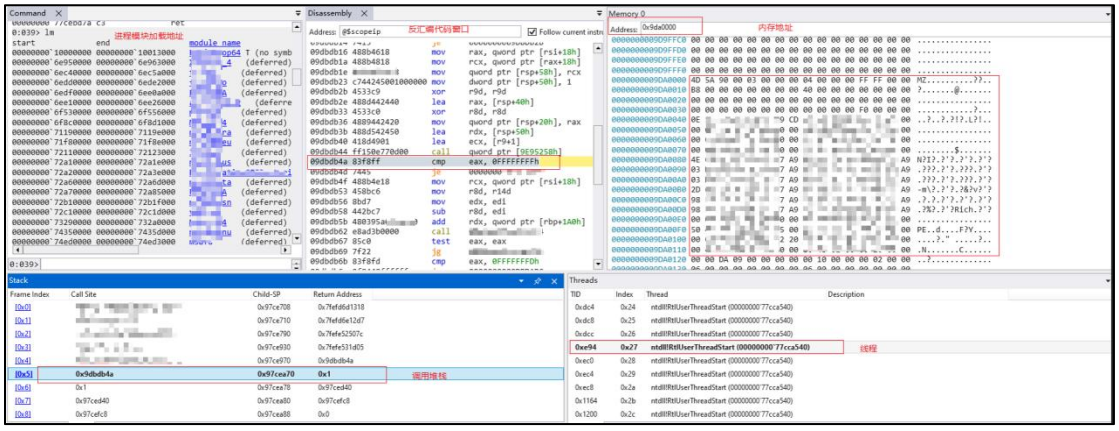

图 内存加载“Back_eleven”过程

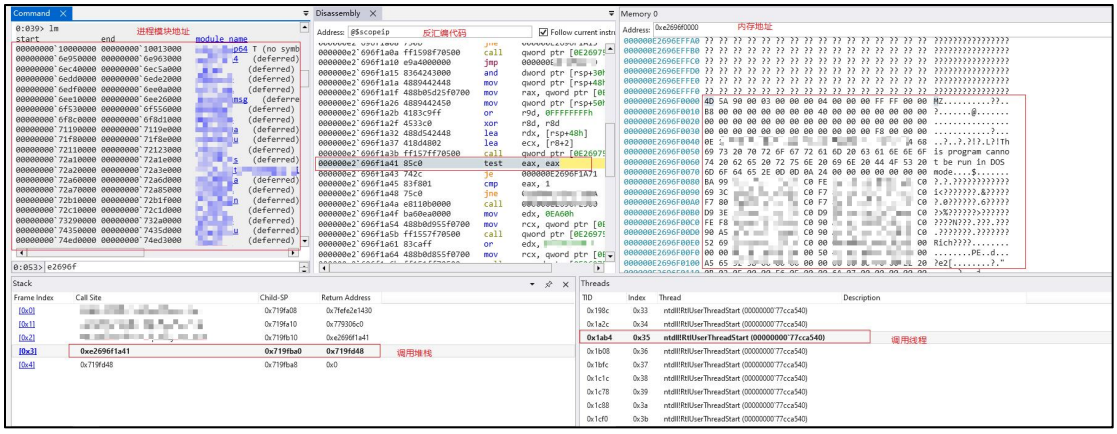

图 内存加载“New_Dsz_Implant”过程

攻击者利用多款网络攻击武器相互配合，搭建起 4 层加密隧道，形成隐蔽性极强且功能完善的网攻窃密平台。

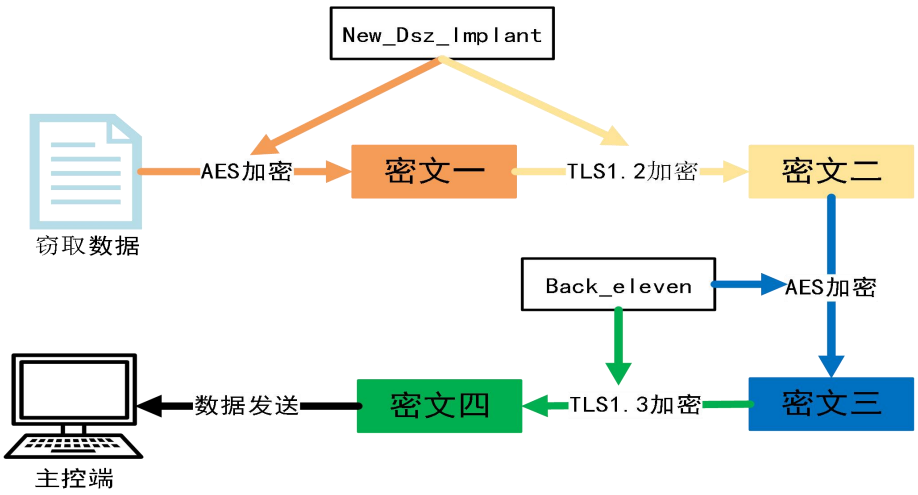

图 网攻武器加密模式

（四）内网横向渗透过程

2024 年 5 月至 6 月，攻击者利用“Back_eleven”以网管计算机为跳板，攻击上网认证服务器和防火墙。

6 月 13 日 9 时，攻击者激活网管计算机上的“eHome_0cx”，植入“Back_eleven”“New_Dsz_Implant”，并以此为跳板窃取认证服务器数据。

7 月 13 日 9 时，攻击者激活网管计算机上的“eHome_0c

x”，下发“Back_eleven”和“New_Dsz_Implant”窃取数据。

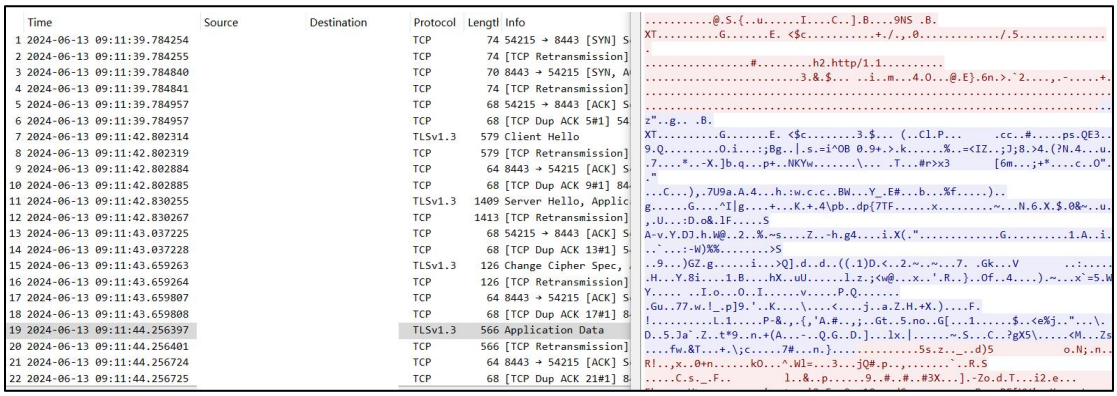

图 2024 年 6 月 13 日网攻窃密数据包

三、网攻武器库分析

攻击者在此次网络攻击事件中使用的网攻武器、功能模块、恶意文件等总计 42 个，主要网攻武器按照功能可分为前哨控守类武器、隧道搭建类武器、数据窃取类武器。

（一）前哨控守类武器

攻击者利用该类型网络攻击武器的隐蔽驻留和心跳回连功能，实现了长期控守目标计算机终端和加载后续网络攻击武器的目的。根据该类型主武器的资源加载路径，将其命名为“eHome_0cx”。

“eHome_0cx”由 4 个网攻模块组成，通过 DLL 劫持系统正常服务（如资源管理器和事件日志服务）实现自启动，在启动后抹除内存中可执行文件头数据，以隐藏网攻武器运行痕迹。

序号	驻留路径	MD5	时间信息
1	C:\windows\system32	786xxxxxxxxx7a9	创建时间：2009/05/06 10:19:10 修改时间：2017/12/10 14:32:24 访问时间：2024/03/15 08:19:25
2	C:\windows\system32	265xxxxxxxxx7bd	创建时间：2009/06/14 10:24:28 修改时间：2017/12/10 14:52:00

			访问时间：2024/03/15 04:29:45
3	C:\programdata\microsoft\search	4d5xxxxxxxxx968	创建时间：2009/06/12 10:25:38 修改时间：2009/06/12 10:25:38 访问时间：2024/03/15 04:29:45
4	C:\windows\system32	3eaxxxxxxxxxx571	创建时间：2009/06/12 09:25:16 修改时间：2009/06/12 09:43:34 访问时间：2009/06/12 09:52:36

表 “eHome_0cx”各网攻模块信息表

“eHome_0cx”以受控主机唯一标识 guid 作为解密武器资源的密钥，各网攻模块之间利用 LPC 端口进行通信，并以指令号的方式调用该武器的各项功能。

指令	功能
0	心跳时间控制
1	设置注册表 productState，ProductUpToDate，DeviceGroup,DeviceHandlers
2	设置注册表 Mask
3	预留
4	读取本地配置发送
5	设置事件
6	设置注册表 DisableProcessIsolation，Seed
7	接收文件，申请内存执行
8	设置注册表 enabled
13	接收文件，申请内存执行
14	接收文件，申请内存执行
15	解密 ehome 模块并执行相关函数
16	处理指定地址的数据，可能与其他模块相关
17	解密 ehome 模块并执行相关函数
18	设置注册表 Attributes
19	设置注册表 DeviceGroup,DeviceHandlers

表“eHome_0cx”功能

“eHome_0cx”使用 RSA 算法和 TLS 协议完成通信加密。“eHome_0cx”内置有与主控端通信使用的 RSA 公钥（见下图）。每次通信均随机产生一个会话密钥，使用 RSA 算法与主控端完成密钥交换过程，之后利用 TLS 协议传输使用会话密钥加密的数据。

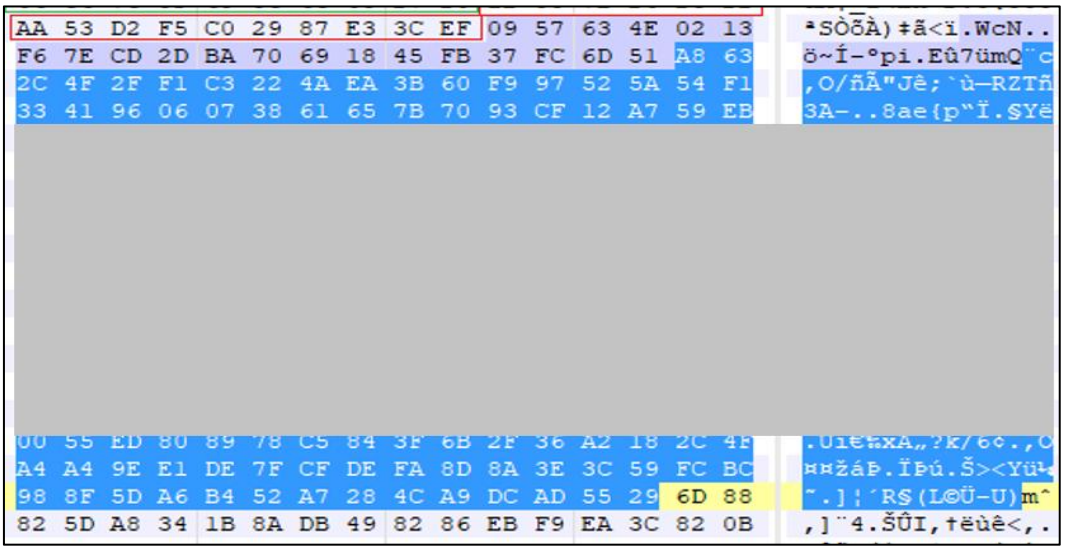

图 RSA 公钥

（二）隧道搭建类武器

攻击者利用该类型网络攻击武器搭建网络通信和数据传输隧道，实现了对其他类型网络攻击武器的远程控制和窃密数据的加密传输，同时还具备信息获取和命令执行功能，在初始连接阶段向主控端发送带有数字“11”标识，命名为“Back_Eleven”。

“Back_Eleven”由“eHome_0cx”加载运行，具有严格运行环境检测机制，若发现运行环境系统版本异常、调试程序正在运行等情况，将启动自删除功能。并且该武器在设计时加入了反调试功能，以防止被逆向分析。

```
memset(v7, 0, sizeof(v7));
v7[0] = 64i64;
v3 = (*(__int64 (**)(void))(a1 + 776))();
if ( !(*(unsigned int (__fastcall **)(__int64, _QWORD, _QWORD *, __int64, _QWORD))(a1
                                                                                  + 936))(// 查询进程信息，反调试
        v3,
        0i64,
        v7,
        64i64,
        0i64) )
{
  v4 = (v7[7] & 1) == 0;
  *(_BYTE *)(a1 + 1052) = v7[7] & 1;          // 0
  if ( v4 && *(int *)(a1 + 1048) >= 6 )
  {
```

图 “Back_Eleven”检测运行环境

“Back_Eleven”具有主动回连和被动监听两种工作模式：在主动回连模式下，“Back_Eleven”解密内置的主控端控制服务器 IP 地址，并使用内置的 RSA 加密算法公钥完成密钥交换，然后使用 AES 算法将上线信息加密后，利用 TLS 协议加密传输到主控端；在被动监听模式下，“Back_Eleven”通过监听 Windows 系统网卡流量，筛选主控端发送的特定条件数据包，实现主控端命令执行。

```
  v5 = *(_QWORD *)(a1 + 128);
  for ( i = *(_DWORD *)(a1 + 136); ; i = *(_DWORD *)(a1 + 136) )
  {
    v9 = recv(*(_QWORD *)(a3 + 16), (char *)(v5 + 9), i - 9, 0);
    v10 = v9;
    if ( v9 == -1 )
    {
      *(_DWORD *)(a3 + 84) = WSAGetLastError();
      *(_WORD *)(a3 + 80) = 257;
LABEL_7:
      a2[1] = 0;
      *(_BYTE *)a2 = 0;
      return a2;
    }
    v8 = *(_QWORD *)(a1 + 128);
    *(_DWORD *)v8 = *(_DWORD *)(a3 + 72);
    *(_DWORD *)(v8 + 5) = v9;
    *(_BYTE *)(v8 + 4) = 0;
    func_send_order_1(a1, v12, a1 + 120, v9 + 9, 0, *(_DWORD *)(a1 + 24), 0);// 接收响应内容，发回给CC
    if ( LOBYTE(v12[0]) )
      break;
    if ( !v10 )
      goto LABEL_7;
    v5 = *(_QWORD *)(a1 + 128);
  }
  if ( a2 != v12 )
  {
    *(_BYTE *)a2 = v12[0];
    a2[1] = v12[1];
  }
  return a2;
}
```

图 “Back_Eleven”向主控端转发数据

```
v16 = *(_DWORD *)(v9 + 36);                     // (v9+36) 的值为实际数据的长度
if ( (v16 & 0xF) != 0 )
  v16 = v16 - (*(_DWORD *)(v9 + 36) & 0xF) + 16;// 长度对齐
if ( v16 )
{
  if ( !func_HeapReAlloc(v9 + 8, v16) )         // 16字节包含长度信息，申请该长度大小的空间
  {
    *(_BYTE *)a2 = 3;
    a2[1] = 0;
    goto LABEL_47;
  }
  v17 = func_recvdata(a1 + 8, (int *)v32, *(char **)(v9 + 16), v16, &v27, a4, 0i64, 0i64);
  if ( &v25 != v17 )                            // 接收内容
  {
    v14 = *(_BYTE *)v17;
    v10 = v17[1];
    if ( *(_BYTE *)v17 )
      goto LABEL_19;
  }
  v18 = *(_DWORD *)(v9 + 36);
  if ( (v18 & 0xF) != 0 )
    v18 = v18 - (*(_DWORD *)(v9 + 36) & 0xF) + 16;// 解密内容
  v19 = (int *)func_decode_recvdata((__int64)a1, (__int64)v33, *(BYTE **)(v9 + 16), v18);
  if ( &v25 != v19 )
  {
    v14 = *(_BYTE *)v19;
    v10 = v19[1];
    if ( *(_BYTE *)v19 )
      goto LABEL_19;
  }
```

图 “Back_Eleven”接收主控端指令并解密

“Back_Eleven”以指令号的方式调用该武器的各项功能。

指令	功能
0	流量转发模块，内容转发
1，2，15	流量转发模块，功能开启连接控制
3	通信测试，发送标志数据，类似心跳
4	清理线程，退出执行
5	连接主控端地址，发送上线包，进行密钥协商等
6	查询本地的变量，然后发送
7	接收变量的值，然后配置本地变量
8	创建 8 位随机数，设置到本地变量，然后发送
9	创建 8 位随机数，设置到本地变量，然后发送
10	修改等待定时器
11	通信测试，发送标志数据
12	接收数据后，重新配置本地变量，然后重连主控端地址
13	刚连接上主控端地址时，发送 13 消息
14	当执行出错和检查变量出错时，发送错误原因
16	获取发送本地变量的值

表 “Back_Eleven”指令功能

（三）数据窃取类武器

攻击者利用此类网络攻击武器进行数据窃密。该武器运行时，通过启动模块化网攻武器框架，加载各种插件模块来实现具体的窃密功能。该武器与 NSA 网攻武器“DanderSpritz”（怒火喷射）具有高度同源性，将其命名为“New-Dsz-Implant”。

“New-Dsz-Implant”由“eHome_0cx”加载运行，在攻击活动中配合“Back_Eleven”所搭建的数据传输链路使用。其自身无具体窃密功能，需通过接收主控端指令加载功能模块，实现各项窃密功能。本次网攻事件中，攻击者使用“New-Dsz-Implant”加载了 25 个功能模块，各模块功能情况如下表所示。

文件名	功能	文件大小
module0	植入模块，加载其他模块	398 KB
module1	设置指定目录为当前目录	145 KB
module2	获取主机时间信息	69 KB

module3	域名和主机名解析 IP 地址	35 KB
module4	作为给定进程的子进程执行	74 KB
module5	获取操作注册表的函数	28.5 KB
module6	查询注册表键和值	41.5 KB
module7	查找 sid 号相关信息	67.0 KB
module8	指定进程申请内存	35.5 KB
module9	截断打开的文件	128 KB
module10	获取计算机开机后的运行时间	43.5 KB
module11	获取最近用户活动时间	49.0 KB
module12	获取系统版本信息	53.0 KB
module13	获取计算机的内存信息	36.5 KB
module14	获取计算机的驱动信息	71.5 KB
module15	磁盘空间查询	40.5 KB
module16	获取进程信息	104 KB
module17	获取服务列表	58.0 KB
module18	开启和关闭组策略配置	68.5 KB
module19	事件日志信息查询	87.5 KB
module20	获取路由表和 Mac 地址	50.5 KB
module21	获取加载删除驱动模块信息	59.5 KB
module22	设置获取管理环境变量	44.0 KB
module23	获取安装的软件列表	60.5 KB
module24	获取删除增加计划任务	104 KB

表 “New-Dsz-Implant”各模块功能

```
*((_QWORD *)Block + 7) = v7;
_InterlockedAdd(&dword_E269767A08, 1u);
v29 = ((__int64 (__fastcall *)(__int64, char *, __int64, _QWORD))v49)(a1, v50, 0x88i64, 0i64);// 执行此处，调用解压的dll的导出函数
v30 = a4;
*a4 = v29;
_InterlockedDecrement(&dword_E269767A08);
*a5 |= 4u;
v7 = 0i64;
if ( *a4 == 3 )
  *a5 |= 2u;
```

图 “New-Dsz-Implant”加载 module0 模块代码

```
if ( Src && v15 )
{
  if ( (v52 & 1) != 0 )
  {
    if ( !sub_7FEE39462C4(Src, v15, (__int64 *)&v25, (unsigned int *)v26) )// 解压缩module文件
    {
      sub_7FEE397D7C4((__int64)v40);
      sub_7FEE396B3BC(v55);                    // 调用module的导出函数执行
      v27 = off_7FEE398D448;
      v19 = off_7FEE398D448;
      return 20i64;
    }
    v16 = (__int64)v25;
  }
```

图 module0 加载其他模块代码

“New-Dsz-Implant”与主控端进行通信时，先使用 AES

和 TLS1.2 进行 2 层数据加密，再使用本地回环的方式利用“Back_Eleven”进行另外 2 层数据加密，最终实现 4 层嵌套加密。

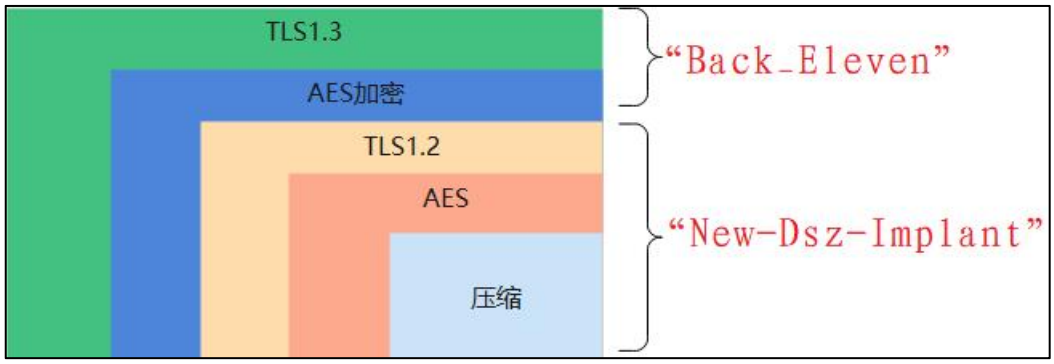

图 嵌套加密模式

```
    break;
  case
    v8 = (char *)func_malloc(0x70u);          // 创建本地回环连接，设置连接端口
    if ( v8 )
    {
      v9 = *(_QWORD *)(a3 + 8);
      v10 = *(_QWORD *)(v9 + 16);
      v11 = *(_DWORD *)(v9 + 40);
      *(_QWORD *)v8 = &off_   ;
      *((_QWORD *)v8 + 2) = a2;
      *((_DWORD *)v8 + 6) = v11;
      *((_DWORD *)v8 + 7) = -1;
      *((_QWORD *)v8 + 5) = 0i64;
      *((_QWORD *)v8 + 7) = 0i64;
      *((_QWORD *)v8 + 9) = 0i64;
      *((_QWORD *)v8 + 11) = 0i64;
      *(_QWORD *)v8 = &off_   ;
      *((_QWORD *)v8 + 12) = *(_QWORD *)v10;
      *((_DWORD *)v8 + 26) = *(_DWORD *)(v10 + 8);
    }
    else
    {
      v8 = 0i64;
    }
    break;
  default:
    *(_BYTE *)a1 = 21;
    goto LABEL_19;
}
```

图 创建本地回环通信

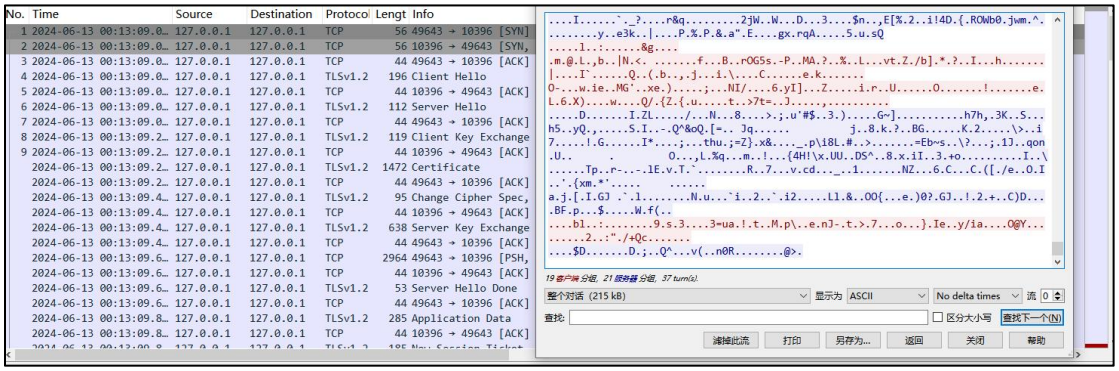

图 本地回环通信数据包

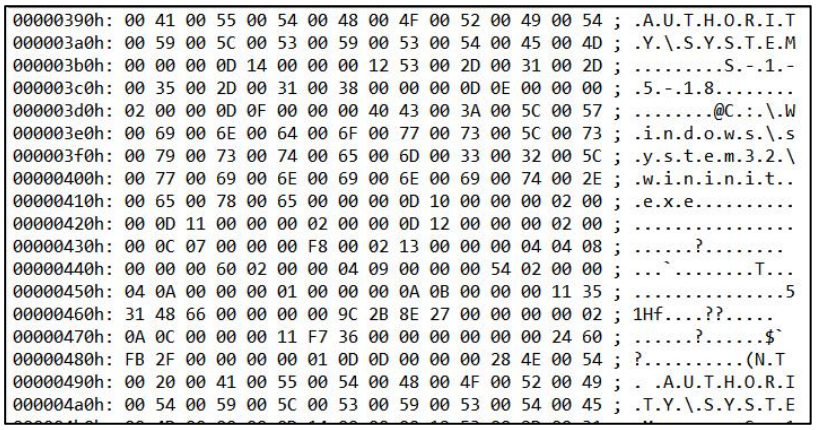

图 解密回环通信数据为进程信息

四、背景研判分析

（一）技术功能细节

“New-Dsz-Implant”是一个网攻武器框架，通过加载不同的模块实现具体功能,此种功能实现方式与NSA武器库中“DanderSpritz”网攻平台一致，且在代码细节上具有高度同源性，并进行了部分功能升级：**一是**加密了部分函数名称和字符串；**二是**使用系统的常规模块名称伪装功能模块；**三是**功能模块编译时间从2012至2013年更新至2016至2018年，各功能模块增加了模拟用户操作函数，伪装用户点击、登录等正常行为以迷惑杀毒软件的检测。

序号	New-Dsz-Implant 功能模块	DanderSpritz 功能模块	模块功能
1	module0	Dsz_Implant_Pc.dll	植入模块，加载其他模块
2	module1	Cd_Target.dll	设置指定目录为当前目录
3	module2	Time_Target.dll	获取主机时间信息
4	module3	NameServerLookup_Target.dll	域名和主机名解析IP地址
5	module4	RunAsChild_Target.dll	作为给定进程的子进程执行
6	module5	Mcl_NtNativeApi_Win32.dll	获取操作注册表的函数
7	module6	RegistryQuery_Target.dll	查询注册表键和值
8	module7	SidLookup_Target.dll	查找sid号相关信息
9	module8	Mcl_NtMemory_Std.dll	指定进程申请内存
10	module9	Papercut_Target.dll	截断打开的文件

11	module10	UpTime_Target.dll	获取计算机开机后的运行时间
12	module11	Activity_Target.dll	获取最近用户活动时间
13	module12	SystemVersion_Target.dll	获取系统版本信息
14	module13	Memory_Target.dll	获取计算机的内存信息
15	module14	Drives_Target.dll	获取计算机的驱动信息
16	module15	DiskSpace_Target.dll	磁盘空间查询
17	module16	Processes_Target.dll	获取进程信息
18	module17	Services_Target.dll	获取服务列表
19	module18	Audit_Target_Vista.dll	开启和关闭组策略配置
20	module19	EventLogQuery_Target.dll	事件日志信息查询
21	module20	Route_Target.dll	获取路由表和 Mac 地址
22	module21	Drivers_Target.dll	获取加载删除驱动模块信息
23	module22	Environment_Target.dll	设置获取管理环境变量
24	module23	Packages_Target.dll	获取安装的软件列表
25	module24	Scheduler_Target.dll	获取删除增加计划任务

表 “New-Dsz-Implant”和“DanderSpritz”所加载功能模块对比

图 module0（左）与 Dsz_Implant_Pc.dll（右）代码同源性对比

图 module1（左）与 Cd_Target.dll（右）代码同源性对比

图 module2（左）与 Time_Target.dll（右）代码同源性对比

图 module3（左）与 NameServerLookup_Target.dll（右）代码同源性对比

图 module4（左）与 RunAsChild_Target.dll（右）代码同源性对比

图 module5（左）与 Mcl_NtNativeApi_Win32.dll（右）代码同源性对比

图 module6（左）与 RegistryQuery_Target.dll（右）代码同源性对比

图 module7（左）与 SidLookup_Target.dll（右）代码同源性对比

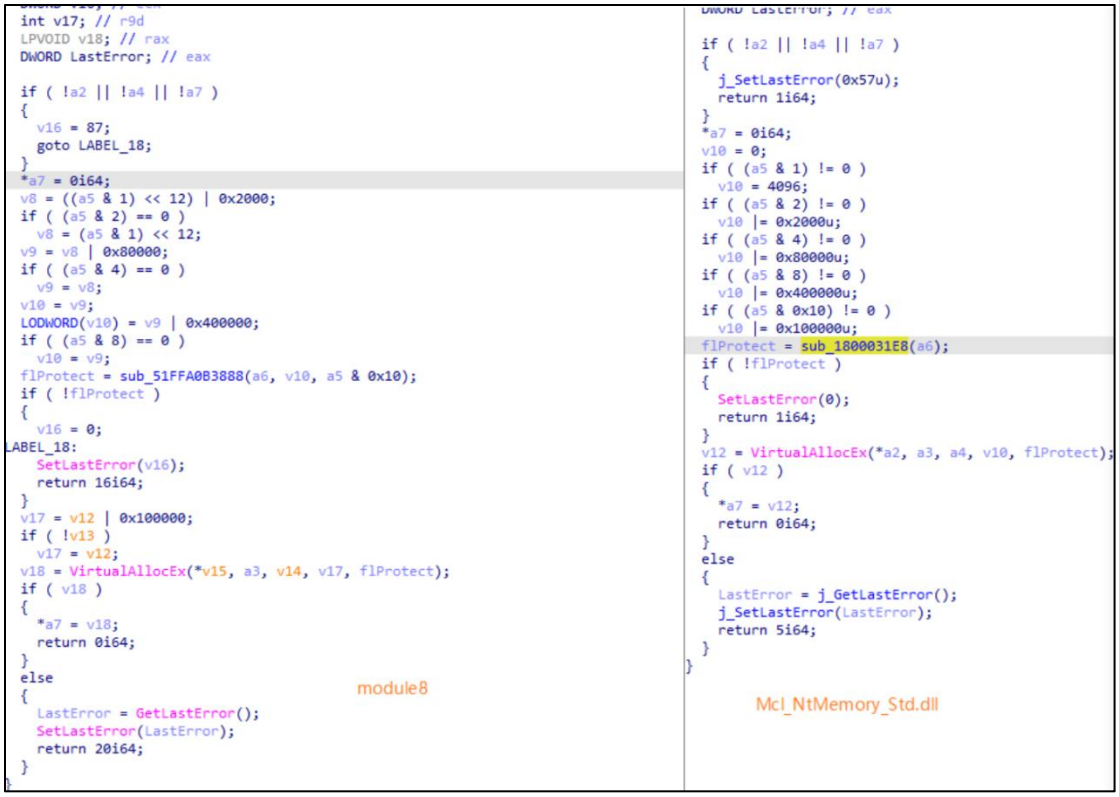

图 module8（左）与 Mcl_NtMemory_Std.dll（右）代码同源性对比

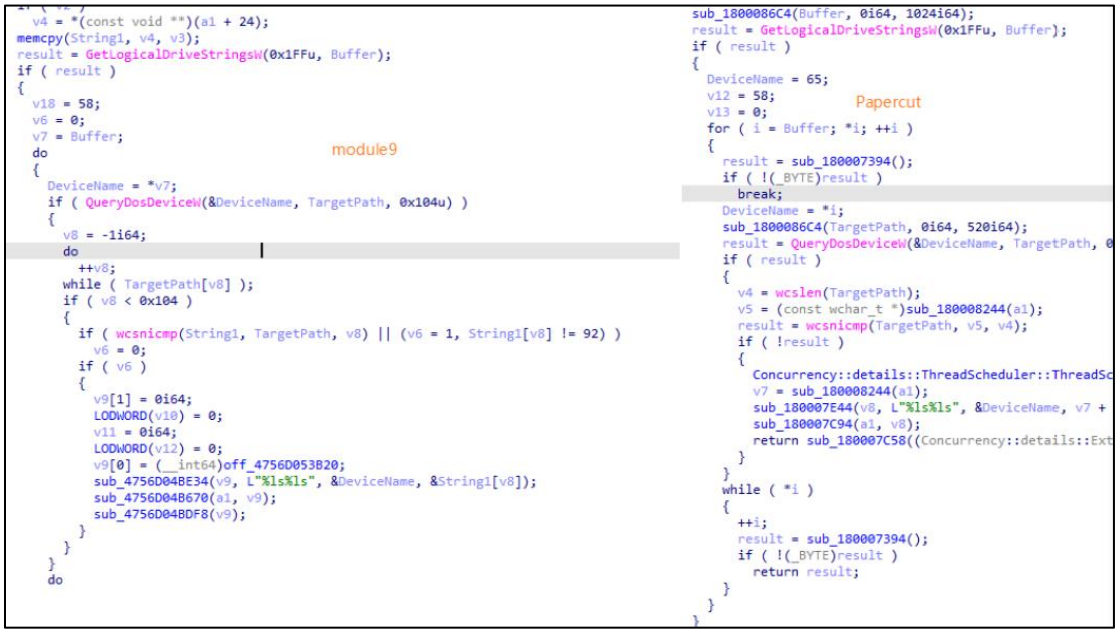

图 module9（左）与 Papercut_Target.dll（右）代码同源性对比

图 module10（左）与 UpTime_Target.dll（右）代码同源性对比

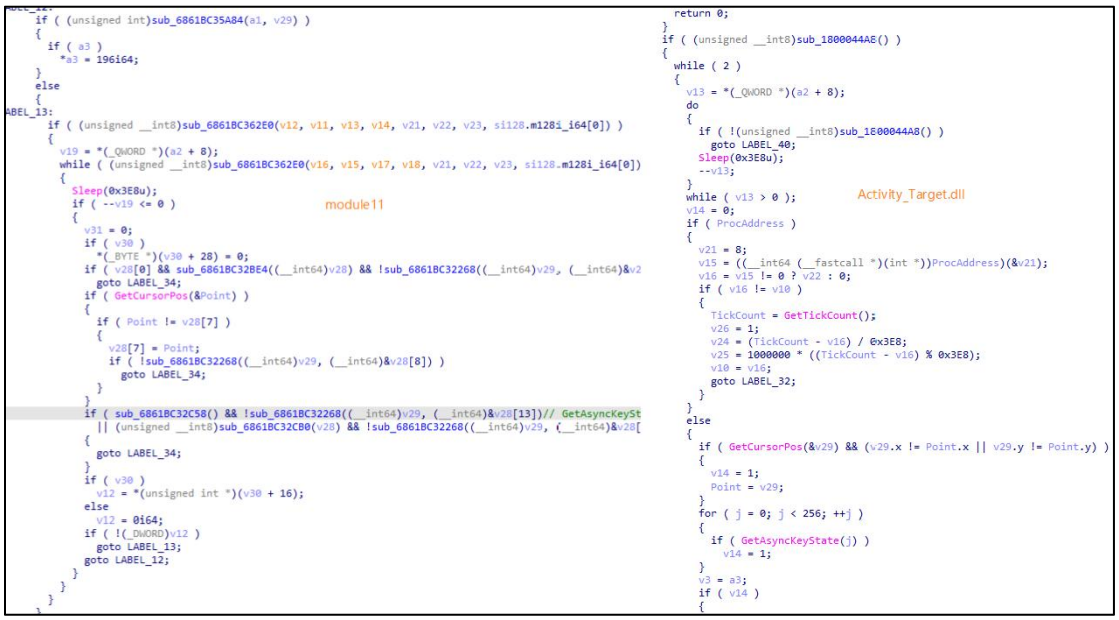

图 module11（左）与 Activity_Target.dll（右）代码同源性对比

图 module12（左）与 SystemVersion_Target.dll（右）代码同源性对比

图 module13（左）与 Memory_Target.dll（右）代码同源性对比

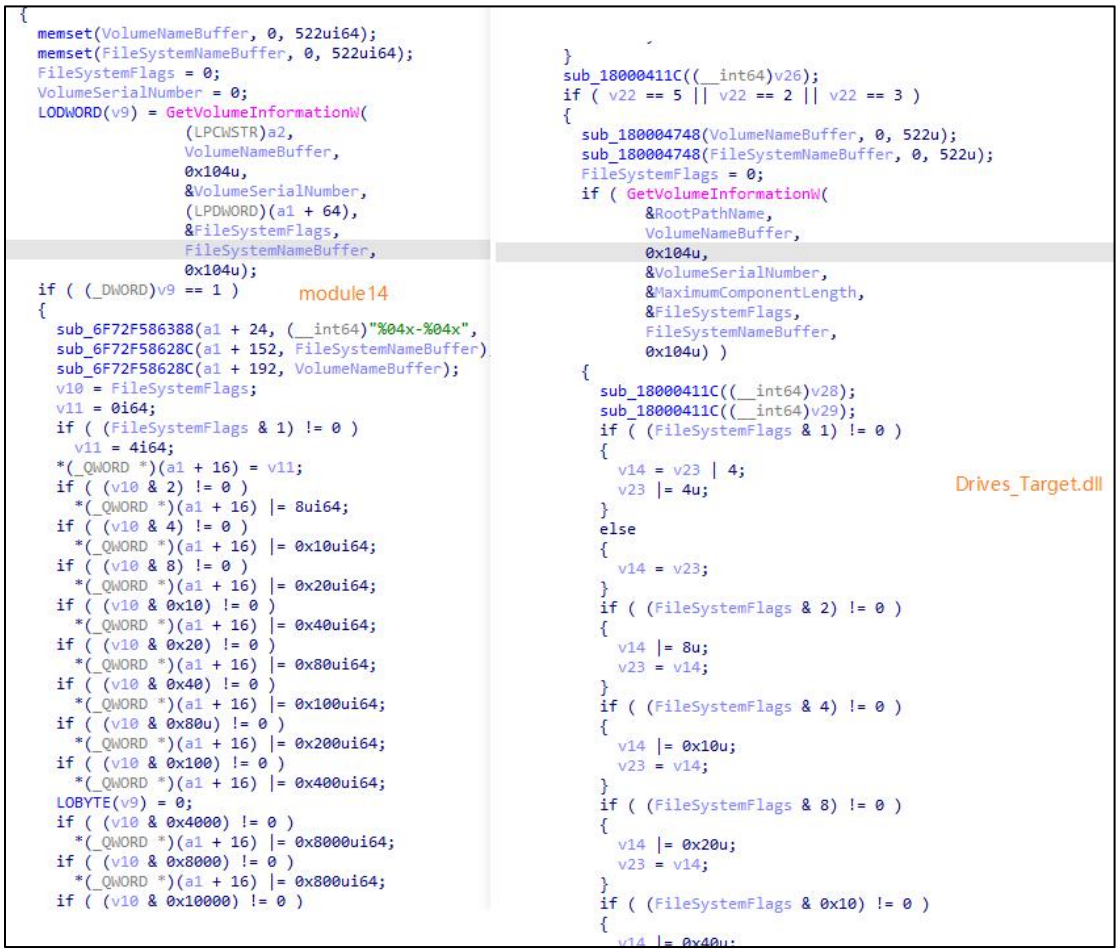

图 module14（左）与 Drives_Target.dll（右）代码同源性对比

图 module15（左）与 DiskSpace_Target.dll（右）代码同源性对比

图 module16（左）与 Processes_Target.dll（右）代码同源性对比

图 module17（左）与 Services_Target.dll（右）代码同源性对比

图 module18（左）与 Audit_Target_Vista.dll（右）代码同源性对比

图 module19（左）与 EventLogQuery_Target.dll（右）代码同源性对比

图 module20（左）与 Route_Target.dll（右）代码同源性对比

图 module21（左）与 Drivers_Target.dll（右）代码同源性对比

图 module22（左）与 Environment_Target.dll（右）代码同源性对比

图 module23（左）与 Packages_Target.dll（右）代码同源性对比

图 module24（左）与 Scheduler_Target.dll（右）代码同源性对比

（二）样本驻留方式

“eHome_0cx”的部分驻留文件通过修改注册表 InprocServer32 键值的方式，劫持了系统正常服务，在系统正常程序

启动前加载实现自启动。注册表修改位置与 NSA“方程式组织”所使用网攻武器相同，均位于 HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID 下随机 ID 项的 InProcServer32 子项。

序号	注册表路径	来源
1	HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{xxx}\InprocServer32	“eHome_0cx”
2	HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{xxx}\InProcServer32	“eHome_0cx”
3	HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{6AF33D21-9BC5-4F65-8654-B8059B822D91}\TypeLib	“方程式组织”木马
4	HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\1945f23e-0573-4e7e-9641-37215654bce4}\InprocServer32	“方程式组织”木马
5	HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{1945f23e-0573-4e7e-9641-37215654bce4}	“方程式组织”木马
6	HKEY_CLASSES_ROOT\CLSID\{B8DA6310-E19B-11D0-933C-00A0C90DCAA9}\InProcServer32	“方程式组织”木马

（三）数据加密模式

攻击者使用的 3 款网攻武器均采用 2 层加密方式，外层使用 TLS 协议加密，内层使用 RSA+AES 方式进行密钥协商和加密，在窃密数据传输、功能模块下发等关键阶段，各武器的相互配合实现了 4 层嵌套加密。此种多层嵌套数据加密模式与相比于“NOPEN”使用的 RSA+RC6 加密模式有了明显升级。

五、码址披露

2023 年 8 月至 2024 年 5 月，美方用于命令控制的部分

服务器 IP，如下表。

序号	IP 地址	归属地
1	172.xxx.xxx.57	美国
2	172.xxx.xxx.214	美国
3	172.xxx.xxx.235	美国
4	23.xxx.xxx.57	美国
5	23.xxx.xxx.86	美国
6	172.xxx.xxx.50	美国
7	70.xxx.xxx.116	波兰
8	70.xxx.xxx.167	波兰
9	70.xxx.xxx.164	波兰
10	193.xxx.xxx.106	丹麦
11	194.xxx.xxx.124	德国
12	91.xxx.xxx.179	德国
13	194.xxx.xxx.19	德国
14	194.xxx.xxx.160	德国
15	194.xxx.xxx.132	德国
16	159.xxx.xxx.116	德国
17	138.xxx.xxx.115	德国
18	95.xxx.xxx.14	法国
19	140.xxx.xxx.76	法国
20	107.xxx.xxx.14	法国
21	104.xxx.xxx.6	法国
22	65.xxx.xxx.89	芬兰
23	135.xxx.xxx.198	芬兰
24	95.xxx.xxx.173	芬兰
25	95.xxx.xxx.104	芬兰
26	158.xxx.xxx.126	韩国
27	158.xxx.xxx.65	韩国
28	158.xxx.xxx.153	韩国
29	141.xxx.xxx.19	韩国
30	103.xxx.xxx.72	日本
31	85.xxx.xxx.13	日本
32	103.xxx.xxx.168	日本
33	168.xxx.xxx.65	荷兰
34	192.xxx.xxx.58	荷兰
35	168.xxx.xxx.32	荷兰
36	206.xxx.xxx.54	荷兰
37	152.xxx.xxx.251	荷兰
38	178.xxx.xxx.106	荷兰
39	64.xxx.xxx.143	罗马尼亚
40	72.xxx.xxx.22	罗马尼亚

41	64.xxx.xxx.2	罗马尼亚
42	72.xxx.xxx.126	罗马尼亚
43	65.xxx.xxx.53	瑞典
44	70.xxx.xxx.40	瑞典
45	65.xxx.xxx.243	瑞典
46	70.xxx.xxx.192	瑞典
47	185.xxx.xxx.60	瑞士
48	159.xxx.xxx.90	瑞士
49	159.xxx.xxx.91	瑞士
50	140.xxx.xxx.140	台湾
51	65.xxx.xxx.30	西班牙
52	208.xxx.xxx.135	西班牙
53	65.xxx.xxx.210	西班牙
54	95.xxx.xxx.31	希腊
55	139.xxx.xxx.253	新加坡
56	139.xxx.xxx.9	新加坡
57	157.xxx.xxx.144	新加坡
58	167.xxx.xxx.159	英国