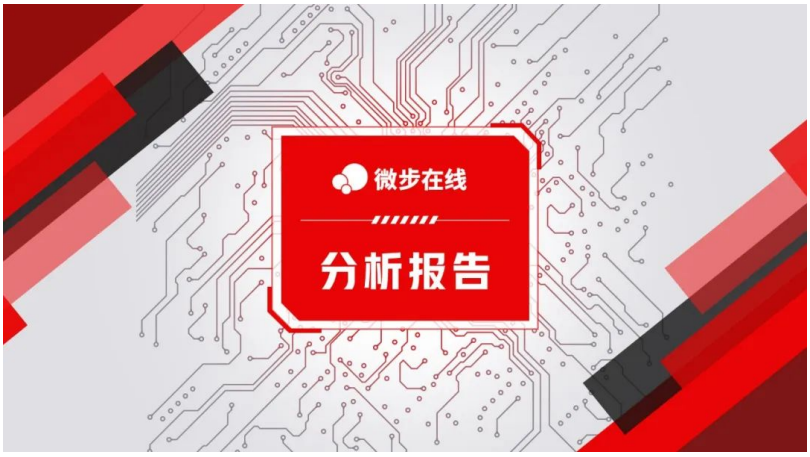
曝光！“海莲花”组织运营的物联网僵尸网络Torii

原创 睡不醒菜 微步在线研究响应中心 2022-11-02 15:19 发表于内蒙古

收录于合集

#安全报告 95 #APT 31 #海莲花 2



1 概述

“海莲花”，又名APT32和OceanLotus，是疑似越南背景的黑客组织。该组织至少自2012年开始活跃，长期针对中国能源行业、海事机构、边防机构、卫生部门、海域建设部门、科研院所和航运企业等进行网络攻击。除中国外，“海莲花”的目标还包含全球的政府、军事机构和大型企业，以及本国的媒体、人权和公民社会等相关的组织和个人。

在历史攻击手法中，“海莲花”一直在尝试不同方法以实现在目标系统上执行恶意代码和绕过安全检测。此外，在2020年“海莲花”组织也开始注重资产侧的流量隐藏，利用攻击手段拿下的国内外IoT、OA服务器作为流量中转。

微步情报局长期跟进“海莲花”组织，对其流量隐藏的手法深入调查分析后有以下发现：

- 攻击者利用1Day、NDay漏洞攻击国内外的IoT设备，且使用Torii特马长期控制，受害设备类型至少包括：三星、Vigor、Draytek路由器和物联网摄像头等；
- Torii木马最早由国外Avast安全厂商在2018年所披露，这表明Torii僵尸网络至少在2018年就开始部署，而后在2021年国内友商披露的RotaJakiro（双头龙）和Torii也存在相似的代码设计思路；
- 微步情报局研究结果表明，有足够的信心认为Torii僵尸网络是由“海莲花”组织所实际控制运营的，并且用于APT作战活动中；
- 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。微步在线威胁感知平台 TDP 、本地威胁情报管理平台 TIP 、威胁情报云 API 、互联网安全接入服务 OneDNS 、主机威胁检测与响应平台 OneEDR 等均已支持对此次攻击事件和团伙的检测。

2 详情

2020年至今，“海莲花”利用国内外失陷IoT设备做流量中转，微步情报局长期跟进“海莲花”组织，并且掌握其使用的大部分CobaltStrike通信C2资产。在此前提下，微步情报局对其流量隐藏的手法深入调查分析，于2021年找到证据确定Torii僵尸网络背后攻击者实际为“海莲花”组织，并且证实Torii僵尸网络控制的主机被用于APT攻击活动的流量隐藏&跳板，流量转发的方式包括iptables转发、tinyPortMapper等，失陷设备代理流量的木马类型包括CobaltStrike、Buni、Torii、Remy等。

Torii木马家族，最早在2018年由Avast安全厂商披露，而后在2021年国内友商报告中提到Torii木马和RotaJakiro（双头龙）存在相似的代码设计思路，木马最新版本和Avast安全厂商披露的版本存在通信流量加密算法上的细微变动。

微步在线深入分析且结合公开披露的信息，有足够的信心将Torii僵尸网络背后的攻击组织归因为“海莲花”，有以下证据：

1.Torii僵尸网络所控制的主机主要被用于“海莲花”攻击活动中的流量中转&隐藏，在少量攻击活动中，该木马也用于控制存储性质的主机窃取数据；

2.Torii木马、RotaJakiro（双头龙）和“海莲花”MacOS平台所使用的木马，三者存在相似的代码设计思路；

3.资产特点和APT32历史中所使用的一致，包括但不限于：域名注册服务商“Internet Domain Service BS Corp.”、NS服务器“he.net”、IP服务器提供商“EstNOC OY”。

3 样本分析

Torii木马拥有一组非常丰富的功能，用于（敏感）信息的窃取，结构化的流量能够多层加密通信，并且可以感染各种设备，并为各种目标架构提供支持（Linux木马的优点，交叉编译可以指定任何架构），包括MIPS、ARM、x86、x64、PowerPC、SuperH等。

木马在投递过程中具有Dropper、Core多个阶段，均为ELF类型并且在编译时将其ELF符号剥离，关键字符串加密存放（如：C2、命令等），通过算法解密，在一定程度上干扰静态分析，除此还具备基础的反沙箱、反调试功能。

其核心功能在Core阶段，此次分析为Core阶段的样本。

1.字符串解密算法如下：

IDA View-A | Pseudocode-A | Pseudocode-B | Hex

```
1 int __fastcall sub_9F68(int result, int a2)
2 {
3   int v2; // [sp+4h] [bp-Ch]
4   int i; // [sp+8h] [bp-8h]
5
6   v2 = result;
7   if ( result )
8   {
9     for ( i = 0; i < a2; ++i )
10    {
11      result = dword_36950;
12      *(_BYTE *)(i + v2) ^= (unsigned int)dword_36950 >> (8 * (i % 4));
13    }
14  }
15  return result;
16 }
```

2.Torii木马入口处延时60秒（可自定义参数，由进程参数决定），疑似反沙箱。

```
50  }
51  if ( argc <= 1 )
52  {
53    sleep(60);
54  }
55  else
56  {
57    v7 = sub_23BF4((int)argv[1]);                 // 可以传参（Sleep）
58    sleep(v7);
59    for ( i = 1; i < argc; ++i )
60    {
61      v8 = argv[i];
62      v9 = strlen((int)v8);
63      sub_1F620((int)v8, v9);
64    }
65  }
66  v34 = sub_C484((int)file_path);
```

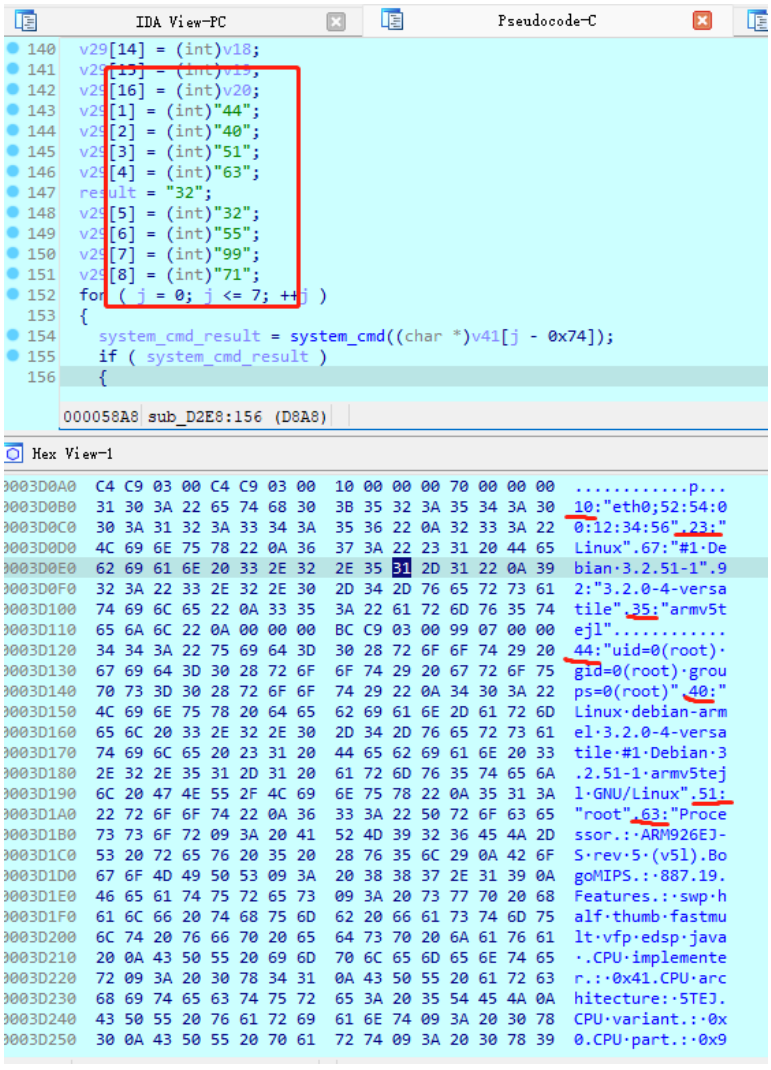
3.创建子进程，通过prctl(PR_SET_NAME)函数调用，设置随机化进程名称，例如“\[[az]{12,17}\]”（正则表达式）。

```
42  }
43  str2 = getpid(v7, v8, dword_3C928, (void *)v9);
44  v10 = memcpy(str1, &str2, sizeof(int));
45  if ( v17 )
46  {
47    if ( !fork(v10, v11, v12, (void *)v17) )
48    {
49      sleep(1);
50      v13 = (void *)sub_21CEC(17, 1);
51      v19 = getpid(v13, v14, v15, v16);
52      memcpy(str1 + 1, &v19, sizeof(int));
53      sub_1684C(1, (int)str1, argc);
54      sub_23E0C(0);
55    }
56    sub_16AA0(argc);                              // 实际功能入口
57  }
58  sub_16AA0(argc);                                // 实际功能入口
59 }
```

4.获取系统信息之后进入获取C2命令的循环。

```
1 void __fastcall __noreturn sub_16AA0(int a1)
2 {
3   int v2; // [sp+Ch] [bp-Ch]
4
5   if ( !sub_167C0(dword_3C924) )
6   {
7     byte_37710 = 0;
8     sub_D2E8();                                   // get system info
9     sub_166CC(a1);
10    v2 = 1;
11    while ( !byte_37710 )
12    {
13      dword_37714 = 0;
14      if ( sub_15214() )                          // C2通信、数据加解密、命令解析执行
15      {
16        if ( dword_37714 > 0 )
17          sleep(60 * dword_37714);                // 随机延时
18      }
19      else
20      {
21        sleep(10);
22        if ( v2 <= 2 )
23        {
24          ++v2;
25        }
26        else
27        {
28          v2 = 1;
29          if ( ++dword_3770C >= dword_36960 )
30            dword_3770C = 0;
31        }
32      }
33    }
34    sub_DD9C();
35  }
36  sub_23E0C(0);
0000EADC sub_16AA0:9 (16ADC)
```

5.通过cat XXXX命令获取系统信息，并且构造成特定格式，ID：“系统信息”，并且使用异或、字节变换等方法加密，之后在构造上线包中再次加密并发送，如下图：



6.主机上线信息构造&加密。

```
char *sub_DC8C()
{
  void *v0; // r0
  void *v1; // r1
  void *v2; // r2
  void *v3; // r3
  int v4; // r0
  char *a2; // [sp+0h] [bp-18h]
  char *v7; // [sp+4h] [bp-14h]
  int *a1; // [sp+8h] [bp-10h]
  char **v9; // [sp+Ch] [bp-Ch]

  a2 = sub_D23C();                              // uname 获取架构信息
  v7 = (char *)sub_DB24();
  a1 = malloc(0);
  v9 = sub_DABC();
  sub_12348(a1, (int *)v9);
  sub_CCF8((int *)v9);
  sub_120E8(a1, byte_36964);
  v0 = (void *)sub_123A8(a1, a2);
  v4 = sub_DB04(v0, v1, v2, v3);
  sub_12174(a1, v4);                            // PID
  sub_123A8(a1, file_path);                     // 文件路径
  sub_123A8(a1, v7);                            // CPU架构
  sub_123A8(a1, (char *)dword_376F8);           // 系统信息，由入口的cat获取
  sub_123A8(a1, (char *)dword_376FC);           // 系统信息，由入口的cat获取
  free((int)a2);
  free((int)v7);
  return (char *)a1;
}
```

7.等待C2服务器命令，按照命令类型分发。具体的功能和国外avast安全厂商所分析一致。

```
    goto LABEL_121;
  }
  while ( !v38 );
  if ( (((int)v72[((unsigned int)Comm_Data_Config->socket >> 5) - 0x5B] >> (Comm_Data_Config->socket & 0x1F)) & 1) == 0 )
    goto LABEL_79;
  v72[((unsigned int)Comm_Data_Config->socket >> 5) - 0x5B] &= ~(1 << (Comm_Data_Config->socket & 0x1F));
  ret_status = sub_183A8(Comm_Data_Config);   // recv command data
  if ( ret_status <= 0 )
  {
    if ( ret_status >= 0 )
      goto LABEL_79;
LABEL_121:
    sub_14DC8();
    goto LABEL_122;
  }
  command_type = (unsigned __int16)((Comm_Data_Config->packet_config_data->command_type[1] << 8) | Comm_Data_Config->packet_config_data->command_type[0]);
  if ( command_type == 0xB76E )
  {
    if ( v31 )
    {
      v28 = 1;
      Comm_Data_Config->unknow11 = 0xB4;
    }
    else
    {
      system_info_xor_data = sub_DC8C();
      if ( system_info_xor_data )
      {
        v5 = (packet_config_data *)sub_11F5C(
                                     (int)Comm_Data_Config->packet_config_data->random_id,
                                     (int)system_info_xor_data);
        Comm_Data_Config->packet_config_data = v5;
        system_info_xor_data = (char *)sub_CCF8((int *)system_info_xor_data);
        sub_18C88(Comm_Data_Config, Comm_Data_Config->packet_config_data);
        v31 = 1;
      }
    }
  }
  else if ( command_type == 0xF76F )
  {
    v50 = 0;
    v21[0] = 0;
    v21[1] = 0;
    v21[2] = 0;
    v21[3] = 0;
    v21[4] = 0;
```

命令类型

命令类型

8.除了加密算法变动，命令类型同avast所披露的一致，具体有以下：

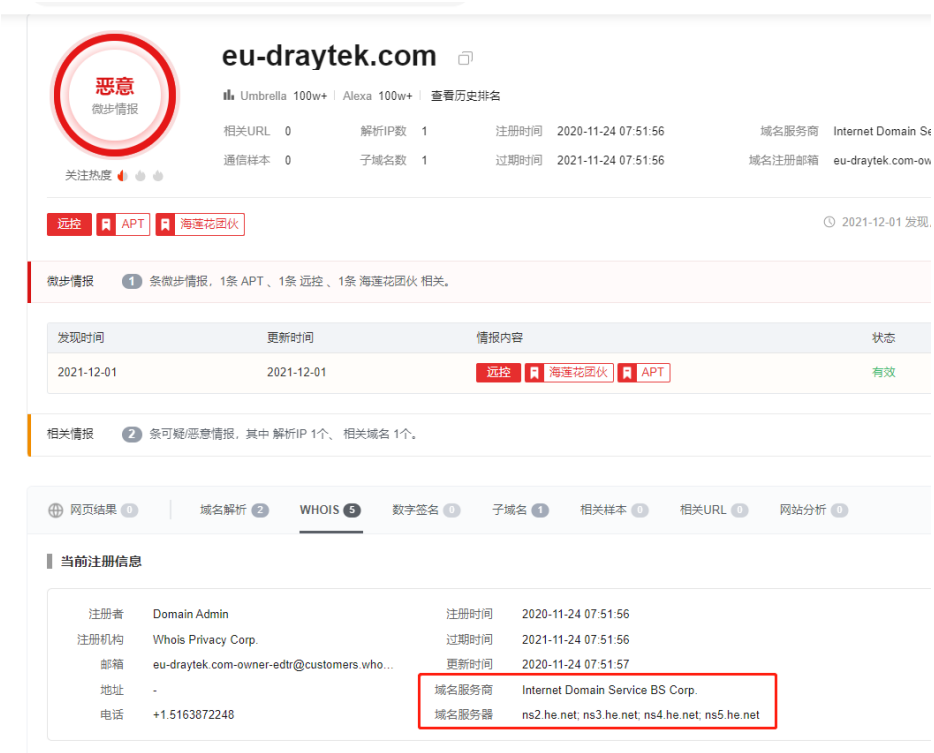
命令类型	**命令说明**
0xB76E	重发上线包
0xF76F	更新C2、Port
0xBB32	从C2服务器下载文件
0xE04B	检查本地系统上是否存在特定文件并返回其大小
0xF28C	从所选文件F的偏移O读取N个字节并发送到C2 服务器
0xC221	从URL下载文件
0xDEB7	删除指定文件
0xA16D	接收用于C2轮询的超时值

0xA863	从C2服务器下载文件，并更改为“rwxr-xr-x”权限，执行
0xAE35	执行Shell命令
0x5B77	
0x73BF 0xEBF0	

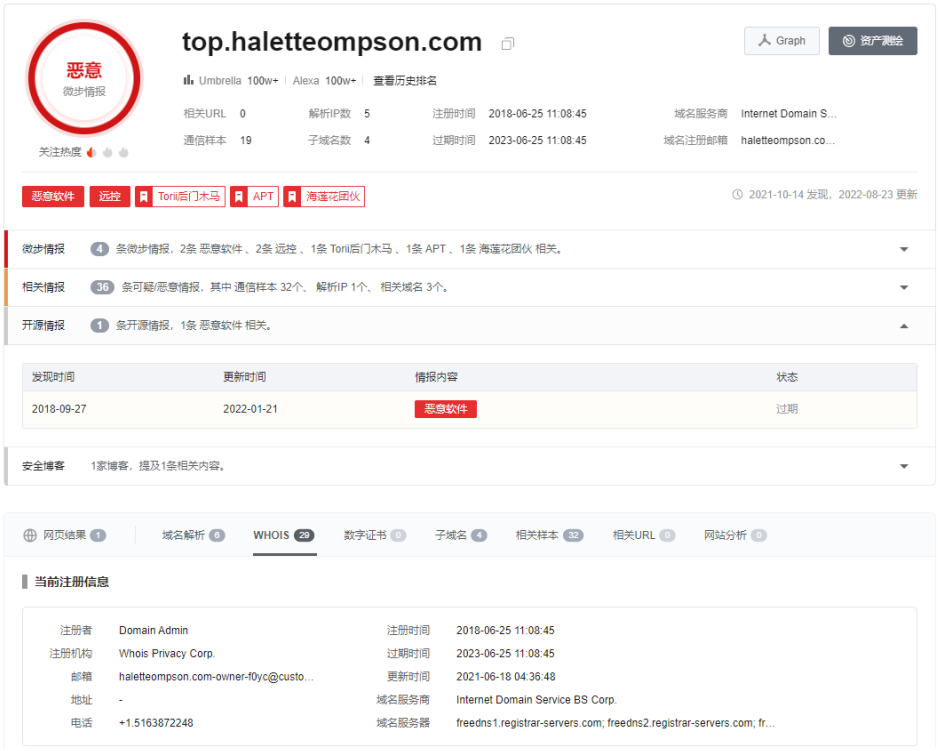
4 拓线关联

1.基于样本分析的研究结果和微步在线的大数据平台，拓线了部分存活的C2服务器，分析资产发现IP服务商和反查域名特点和APT32历史中所使用的几乎一致。

如：域名注册服务商“Internet Domain Service BS Corp.”、NS服务器“he.net”、IP服务器提供商“EstNOC OY”等。



2.除此，国外安全厂商18年所披露的“top.haletteompson.com”域名解析的服务器依旧存活，域名注册时间长达5年。



3.微步在线深入分析后，参考结合其他安全厂商所披露的报告，有足够的信心认为Torii僵尸网络是由“海莲花”组织所实际控制的，并且用于APT作战活动中。