日本の組織を狙うマルウェア「Thumtais」| LAC WATCH

By 石川 芳浩

Published: 2024-06-05 · Archived: 2026-04-05 20:24:03 UTC

ラックの石川です。

ラックの脅威分析チームでは、最新のサイバー攻撃に対処するため、日本の組織を標的とする多様な攻撃を日々調査しています。私たちは、2023年2月、日本のコンサルティング会社を対象とした標的型攻撃を観測しました。

この攻撃は、中国圏を拠点とする攻撃者グループによるものとみられ、Thumtais（別名：EAGERBEE）マルウェアやCython、Go、Nimといったプログラミング言語で開発された未知のマルウェアが利用されていました。Thumtaisは、Elastic Security Labsのブログ※1の中で紹介されているマルウェアの1つですが、調査する過程で新たに確認したThumtaisは、マルウェアの機能が追加およびアップデートされていました。

そこで今回は、この新しいThumtaisと背後に潜む攻撃者グループについて紹介します。

※1 Introducing the REF5961 intrusion set -- Elastic Security Labs

目次

攻撃の概要

Thumtaisは、IKE and AuthIP IPsec Keying Modules（IKEEXT）サービスを悪用したDLLハイジャックまたは正規のアプリケーションを利用して、DLLサイドローディングによって実行されます。図1は、DLLハイジャックを悪用して、このマルウェアが実行される一例です。以降では、DLLハイジャックによって読み込まれるThumtais Loaderを解説します。

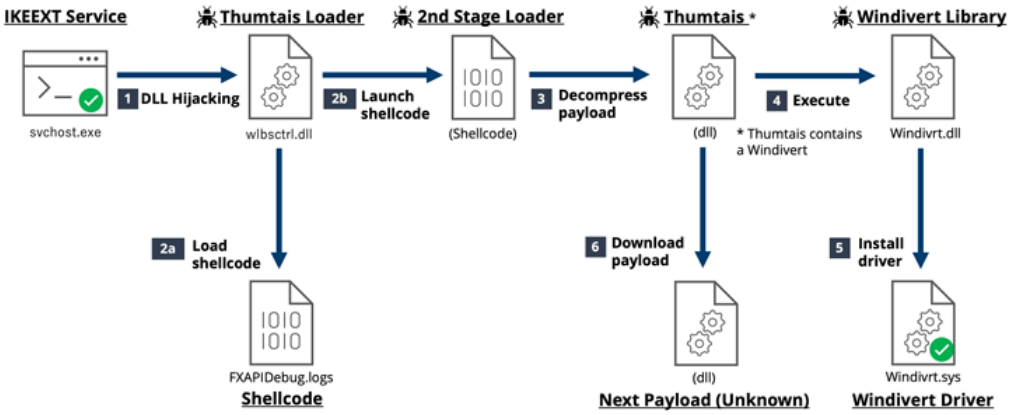


図1 Thumtaisの実行フロー

Thumtais Loader

Thumtais Loaderは、Thumtaisを読み込み、実行するローダ型マルウェアです。シェルコードを内包するタイプと別ファイルから読み込むタイプの2種類あります。

シェルコードを内包するタイプ

シェルコードを内包するThumtais Loaderは、実行時に図2に示すように、内包されたシェルコードを確保したメモリ領域に展開後、Call命令を利用して展開されたメモリ領域を呼び出し、シェルコードを実行します。

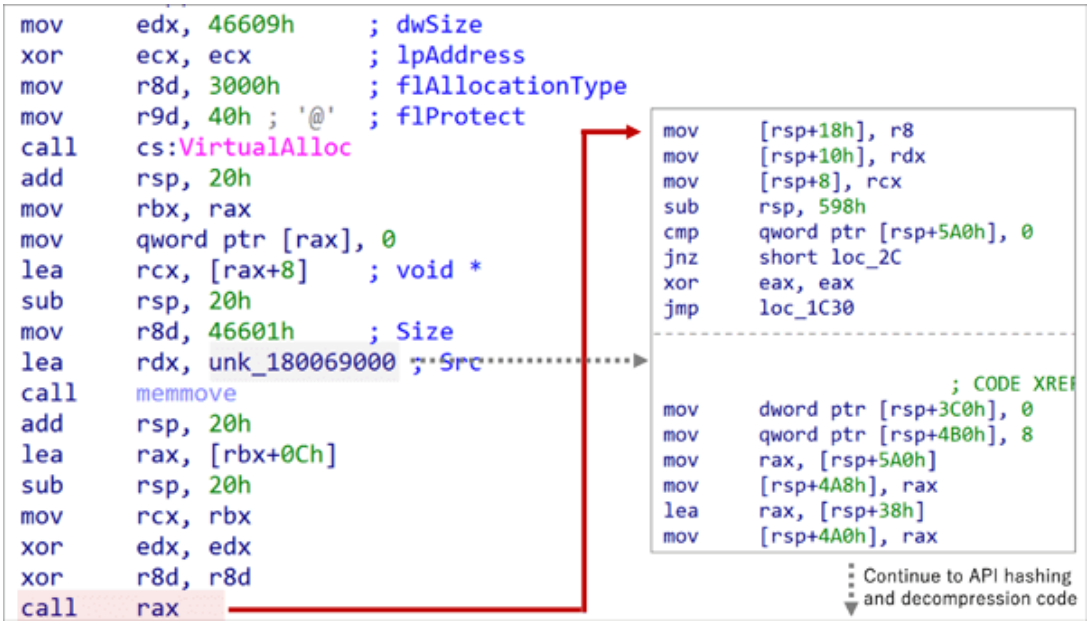


図2 シェルコードを内包するThumtais Loaderの実行処理

シェルコードを読み込むタイプ

別ファイルからシェルコードを読み込むThumtais Loaderは、図3に示すような方法でシェルコードを読み込み、メモリ領域上に展開し、内部に持つタイプと同様に実行します。読み込むファイルは表1に示すものが対象となり、ワイルドカードが利用されている具体的なファイル名は、"FXAPIDebug.logs"や"iconcaches.mui"などを確認しています。

```
ExpandEnvironmentStringsW(L"%tmp%\\*g.logs", Filename, 0x200u);
if ( FindFirstFileW(Filename, &FindFileData) == (HANDLE)-1LL )
  break;
v11 = Filename;
for ( i = 512LL; i; --i )
  *v11++ = 0;
ExpandEnvironmentStringsW(L"%tmp%\\", Filename, 0x200u);
lstrcatW(Filename, FindFileData.cFileName);
v13 = CreateFileW(Filename, GENERIC_READ, 1u, 0LL, 3u, 0, 0LL);
v14 = v13;
v15 = 0LL;
if ( v13 == (HANDLE)-1LL )
  return 0;
FileSize = GetFileSize(v13, 0LL);
v17 = FileSize;
if ( FileSize )
{
  v15 = VirtualAlloc(0LL, FileSize + 1, 0x3000u, 4u);
  v15[v17] = 0;
  ReadFile(v14, v15, v17, &NumberOfBytesWritten, 0LL);
}
CloseHandle(v14);
```

図3 シェルコードを読み込むThumtais Loaderの実行処理

%tmp%¥*g.logs
c:¥users¥public¥videos¥*.mui
c:¥users¥public¥nsdftuses.t
c:¥users¥public¥ntusess.t
c:¥users¥public¥ntuses.t
%tmp%¥*g.logs
c:¥users¥public¥videos¥*.mui
c:¥users¥public¥nsdftuses.t
c:¥users¥public¥ntusess.t
c:¥users¥public¥ntuses.t

表1 Thumtais loaderが読み込むシェルコードファイル例

また、Thumtais Loaderは、対解析機能としてControl Flow Flattening（CFF）により処理フローが難読化されたものもあります。（図4）

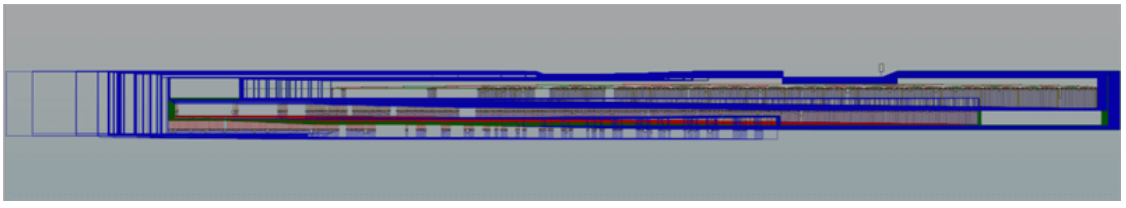


図4 Control Flow Flatteningで難読化されたコード

2nd Stage Loader（シェルコード）

展開されたシェルコードは、最初にROR13アルゴリズムを利用してハッシュ値を計算後、その値に10を加算するAPI HashingでWindows APIを解決します。（図5）

```
ror     eax, 0Dh
mov     [rsp+538h+var_70], eax
mov     rax, [rsp+538h+var_68]
movsx   ecx, byte ptr [rax]
mov     eax, [rsp+538h+var_70]
add     eax, ecx
mov     [rsp+538h+var_70], eax
mov     rax, [rsp+538h+var_68]
add     rax, 1
mov     [rsp+538h+var_68], rax
mov     rax, [rsp+538h+var_68]
movsx   eax, byte ptr [rax]
test    eax, eax
jnz     short loc_42F
mov     eax, [rsp+538h+var_70]
add     eax, 0Ah
mov     [rsp+538h+var_480], eax
cmp     [rsp+538h+var_480], 0EC0E4E98h ; LoadLibraryA
jz      short loc_4EE
cmp     [rsp+538h+var_480], 7C0DFCB4h ; GetProcAddress
jz      short loc_4EE
cmp     [rsp+538h+var_480], 91AFCA5Eh ; VirtualAlloc
jz      short loc_4EE
cmp     [rsp+538h+var_480], 30633B6h ; VirtualFree
jz      short loc_4EE
cmp     [rsp+538h+var_480], 45B06D80h ; GetModuleFileNameA
jz      short loc_4EE
cmp     [rsp+538h+var_480], 0A36DC680h ; IsDebuggerPresent
jz      short loc_4EE
cmp     [rsp+538h+var_480], 7946C625h ; VirtualProtect
jnz     loc_7C4
```

図5 Thumtais Loaderのシェルコードに実装されるAPI Hashing

マルウェア本体のThumtaisのコードは、LZNT1で圧縮された状態でシェルコードに埋め込まれています。このため、シェルコードは、図6に示すように、RtlDecompressBuffer APIを利用して、Thumtaisをメモリ領域上に展開後、当該ファイルのDLLEntryPointを呼び出し、実行します。

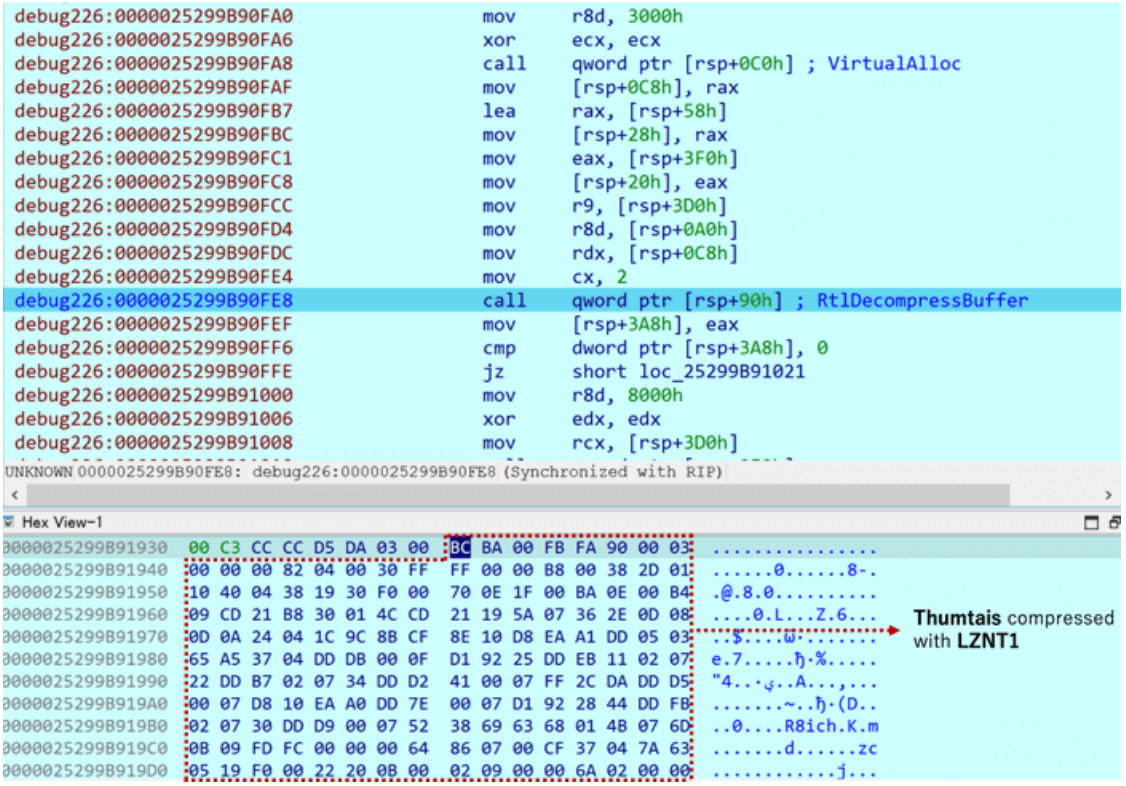


図6 シェルコードによるRtlDecompressBufferを利用したThumtaisの展開

展開されたThumtaisはDLLファイルですが、PEファイルヘッダのマジックナンバーがMZやPEといった文字列ではなく、"FB FA"や"FD FC"に変更されていました。（図7）

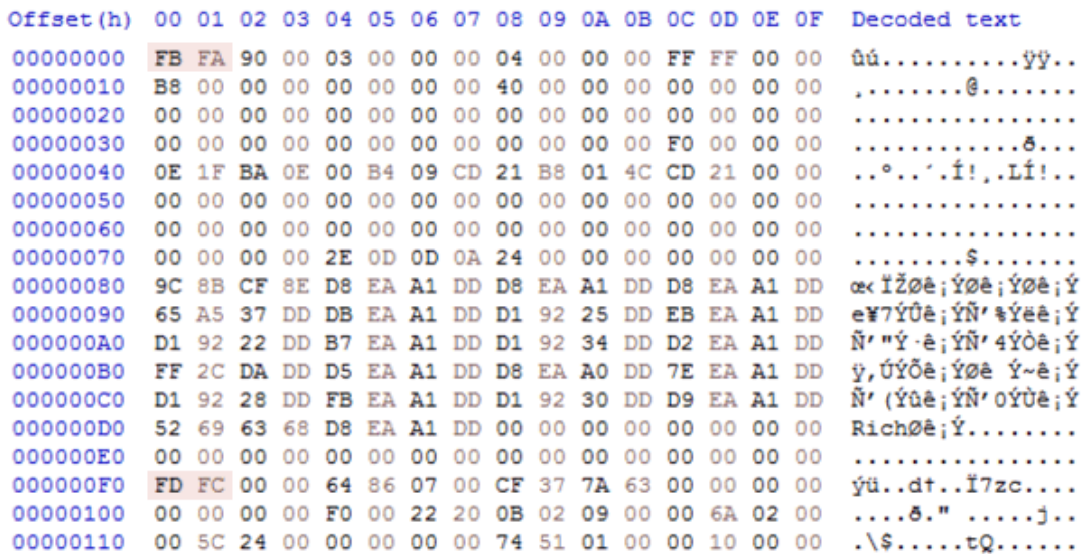


図7 LZNT1によって展開されたThumtais

Thumtais

Thumtaisは、Microsoft Visual C/C++で書かれたダウンローダ型のマルウェアであり、C2サーバからダウンロードしたDLLファイルをメモリ上で実行する機能を有します。脅威分析チームでは、このマルウェアが実行時に"thumtais2.dat"ファイルを操作することから、Thumtaisと命名しています。（図8）

```
v4 = (WCHAR *)xxx_readfile(L"c:\\users\\public\\thumtais2.dat", &v16)
Sleep(0xC8u);
if ( v4 )
{
  v5 = v4;
  do
  {
    v6 = *v5;
    *(WCHAR *)((char *)v5 + (char *)&FileName - (char *)v4) = *v5;
    ++v5;
  }
  while ( v6 );
  VirtualFree(v4, 0LL, 0x8000u);
  DeleteFileW(L"c:\\users\\public\\thumtais2.dat");
```

図8 thumtais2.datファイル

図9は、Thumtaisをコンパイルタイムスタンプの情報を基にタイムラインで時系列にまとめたものです。このマルウェアは、2022年5月ごろから継続的に攻撃に悪用されていることが確認できます。

以降では、2022年11月以降のThumtaisに追加された新たな機能や新旧の変更点を紹介します。なお、ここでは、2022年11月以降のThumtaisを新検体、それより前のものを旧検体とします。

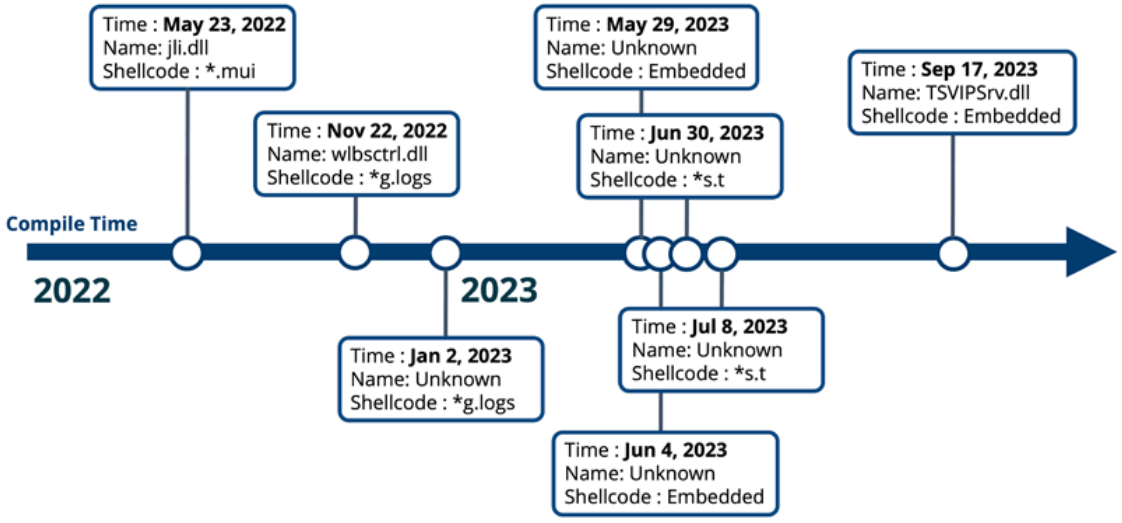


図9 Thumtaisのコンパイルタイムベースのタイムライン

Windows Packet Divert（WinDivert）

新しいThumtaisには、オープンソースで公開されるWinDivert※2が含まれていました。WinDivertは、Windows上でネットワークトラフィックを盗聴・操作することを可能にするツールであり、ユーザモードで動作するライブラリ"WinDivert.dll"とカーネルドライバである"WinDivert32.sys/WinDivert64.sys"の2つのモジュールで構成されます。

Thumtaisには、1つ目のライブラリとしてWinDivert バージョン2.2.1のソースコードをベースにカスタマイズされたDLLファイルが含まれており、このDLLファイルはThumtaisによってメモリ領域上に展開され、動作します。（図10）

※2 GitHub - basil00/Divert: WinDivert: Windows Packet Divert

```
00 00 00 00 52 53 44 53  1E AD 2D AB 95 E3 CC 46  ....RSDS..-.....
B3 0B 23 B1 36 95 C7 87  0E 00 00 00 44 3A 5C 77  ..#.6.Ǉ....D:\w
69 6E 64 69 76 65 72 74  5C 57 69 6E 44 69 76 65  indivert\WinDive
72 74 2D 32 2E 32 2E 31  2D 53 6F 75 72 63 65 5C  rt-2.2.1-Source\
57 69 6E 44 69 76 65 72  74 2D 32 2E 32 2E 31 5C  WinDivert-2.2.1\
64 6C 6C 5C 78 36 34 5C  52 65 6C 65 61 73 65 5C  dll\x64\Release\
57 69 6E 44 69 76 65 72  74 2E 70 64 62 00 00 00  WinDivert.pdb...
```

図10 Thumtaisに含まれるWinDivertライブラリ

2つ目のカーネルドライバは、前述するDLLファイルに含まれており、図11に示すように、実行時に感染端末の"%windir%¥Temp"配下にドロップし、インストールされます。なお、インストールされるカーネルドライバは、GitHubで公開されているWinDivert Version 2.2.0（WinDivert-2.2.0-C.zip）の"WinDivert64.sys"と同一のものでした。（図12）

```
ExpandEnvironmentStringsW(L"\\??\\c:\\windows\\temp\\tmpA8B4f6.tmp", Dst, 0x3E8u);
if ( GetFileAttributesW(L"c:\\windows\\temp\\tmpA8B4f6.tmp") == -1 )
{
  FileW = CreateFileW(L"c:\\windows\\temp\\tmpA8B4f6.tmp", 0x40000000u, 0, 0LL, 2u, 0x80u, 0LL);
  v1 = FileW;
  if ( FileW == (HANDLE)-1LL )
    return 0LL;
  Buffer = 77;
  WriteFile(FileW, &Buffer, 1u, &NumberOfBytesWritten, 0LL);
  Buffer = 90;
  WriteFile(v1, &Buffer, 1u, &NumberOfBytesWritten, 0LL);
  WriteFile(v1, &unk_180023000, 0x16156u, &NumberOfBytesWritten, 0LL);
  CloseHandle(v1);
}
sub_1800110F5((BYTE *)Dst);
v3 = CreateFileW(L"\\\\.\\WinDivert", 0xC0000000, 0, 0LL, 3u, 0x40000080u, (HANDLE)0xFFFFFFFFFFFFFFFFLL);
sub_180011109(L"WinDivert");
```

図11 DLLファイルに含まれるカーネルドライバ

```
PS C:¥> Get-FileHash C:¥windows¥temp¥tmpA8B4f6.tmp,C:¥WinDivert-2.2.0-C¥x64
¥WinDivert64.sys -Algorithm md5

Algorithm       Hash
---------       ----
MD5             3216155AA3B230EA8DB7D0FB27101713
MD5             3216155AA3B230EA8DB7D0FB27101713
```

図12 カーネルドライバのハッシュ値の比較

Thumtaisは、このWinDivertを介して、図13に示す文字列を含むドメイン名のパケットをカーネルドライバで監視し、ユーザモードのDLLファイルで該当するDNSパケットの一部を0に書き換え、セキュリティ対策製品が利用する名前解決の通信を妨害します。書き換えの対象ドメイン名については、Appendixの書き換え対象ドメイン名を参照ください。

```
strcpy(v29, "3 ");
si128 = _mm_load_si128((const __m128i *)"udp.DstPort == 5");
v0 = (void *)WinDivertOpen(&si128, 0LL, 404LL, 0LL);
if ( v0 != (void *)-1LL )
{
  sub_180011109((__int64)L"WinDivert");
  v36 = _mm_load_si128((const __m128i *)"augur.scanners.");
  strcpy(v8, "ksn-file");
  v31 = _mm_load_si128((const __m128i *)"checkappexec.mic");
  v26 = _mm_load_si128((const __m128i *)"networkdevice.sc");
  v40 = '1cd';
  strcpy(v15, "ortex.dat");
  strcpy(v4, "ksn-a");
  strcpy(v20, "alprotect1.m");
  strcpy(v19, "on.ccs.mcaf");
  v32 = 'r';
  strcpy(v21, "cloud.gti.mc");
  strcpy(v22, "protect1.mca");
  strcpy(v24, "adownload.mcaf");
  strcpy(v13, ".c.eset.");
  strcpy(v27, "anne");
  strcpy(v17, "edf.eset.");
  strcpy(v14, "ts.eset.");
  strcpy(v23, "tscreen.micros");
  strcpy(v9, "sn-verdi");
  strcpy(v6, "sn-url-");
  strcpy(v16, "sn-cinfo-");
  strcpy(v10, "crc.tren");
  strcpy(v11, "url.tren");
  strcpy(v18, "ensus.tren");
  strcpy(v7, "rx.tren");
  strcpy(v12, "dev.drwe");
  strcpy(v5, "f2.drw");
  v2 = VirtualAlloc(0LL, 0x10027uLL, 0x1000u, 4u);
```

図13 フィルタリングする文字列

マルウェアの動作時間

Thumtaisには、GetLocalTime APIを利用した動作時間のチェック機能が実装されています。新検体では、旧検体とDayOfWeekの値が異なっていました。具体的には、新検体ではDayOfWeekが0-6（日曜日-土曜日）かつHourが0-23（00時から23時）の条件が設定されており、いずれの日時でも動作する設計です。（図14）

```
memset(&String[1], 0, 0x1FFuLL);
memmove(String, "0-6:00:23;6:00:23;", 0x9DuLL);
strcpy(ProcName, "GetLocalTime");
LibraryA = LoadLibraryA("KERNEL32.dll");
ProcAddress = GetProcAddress(LibraryA, ProcName);
memset(v77, 0, sizeof(v77));
memset(v78, 0, sizeof(v78));
v21 = (void (__fastcall *)(char *))ProcAddress;
```

```
memset(&String[1], 0, 0x1FFuLL);
memmove(String, "0-5:00:23;6:00:23;", 0x9DuLL);
strcpy(ProcName, "GetLocalTime");
LibraryA = LoadLibraryA("KERNEL32.dll");
ProcAddress = GetProcAddress(LibraryA, ProcName);
memset(v63, 0, sizeof(v63));
memset(v64, 0, sizeof(v64));
v21 = (void (__fastcall *)(char *))ProcAddress;
```

図14 動作時間のチェック（上：新検体／下：旧検体）

認証Proxy機能

Thumtaisは、Proxy経由でC2サーバへ通信する機能を有していますが、認証プロキシサーバからエラー（407 Proxy Authentication Required）応答があった場合の処理に新旧で違いがありました。新検体では認証プロキシに対応しており、図15に示すようなハードコードされたリクエストヘッダを利用して、C2サーバに通信をします。この通信では、User-Agentヘッダに"My Service Enpoint 1.0"が含まれているところが特徴的です。

```
strlwr(Buffer);
strcpy(SubStr, "200 connec");
if ( !strstr(Buffer, SubStr) )
{
  ++dword_180266F2C;
  (*(void (__fastcall **)(__int64))(qword_180266F70 + 248))(v8);
  if ( strstr(Buffer, " 407 ") )
  {
    xxx_using_proxy_auth(cp, a2, a3, a4);
    exit(0);
  }
  return -1LL;
}
strlwr(Str);
strcpy(SubStr, "200 connec");
if ( !strstr(Str, SubStr) )
{
  ++dword_18002B4F4;
  (*(void (__fastcall **)(__int64))(qword_18002B508 + 248))(v8);
  if ( strstr(Str, " 407 ") )
    exit(0);
  return -1LL;
}
```

図15 認証プロキシ機能（上：新検体／下：旧検体）

```
"CONNECT %s:%d HTTP/1.1 \r\n"
"User-Agent: My Service Enpoint 1.0\r\n"
"Host: %s:%d\r\n"
"Proxy-Connection: Keep-Alive\r\n"
"Pragma: no-cache\r\n"
"Proxy-Authorization: Negotiate %s\r\n"
"Content-Length: 0\r\n"
"\r\n",
```

図16 認証Proxyに対応した通信を行う際のリクエストヘッダ例

他にもThumtaisには、C2サーバからダウンロードしたファイルコンテンツに意図する文字列が含まれるか確認する機能を有しています。この比較する際の文字列が旧検体では "zaq1xsw2cde3"でしたが、新検体では "a123456"または"z123456"と検体によって異なっている点もありました。

攻撃者グループの考察

次に、Thumtaisの通信先に目を向けると、マルウェアの種類やインフラなどの関連要素から"TA428"または"LuckyMouse"と呼ばれる攻撃者グループによる犯行である可能性が見えてきました。

図17は、ThumtaisがC2サーバとして利用する通信先を元に、Maltegoで関連する要素をマッピングしたものです。通信先であるC2サーバのIPアドレスに紐づくドメイン名がPhantomNetと呼ばれるマルウェアのC2サーバとして利用されていました。この紐づくドメイン名にC2通信を行うマルウェアを解析すると、PhantomNetが持つ特徴的な文字列"GetPluginInfomation", "GetRegisterCode", "GetPluginObject", "DeletePluginObject"が検体内に含まれていました。（図18）

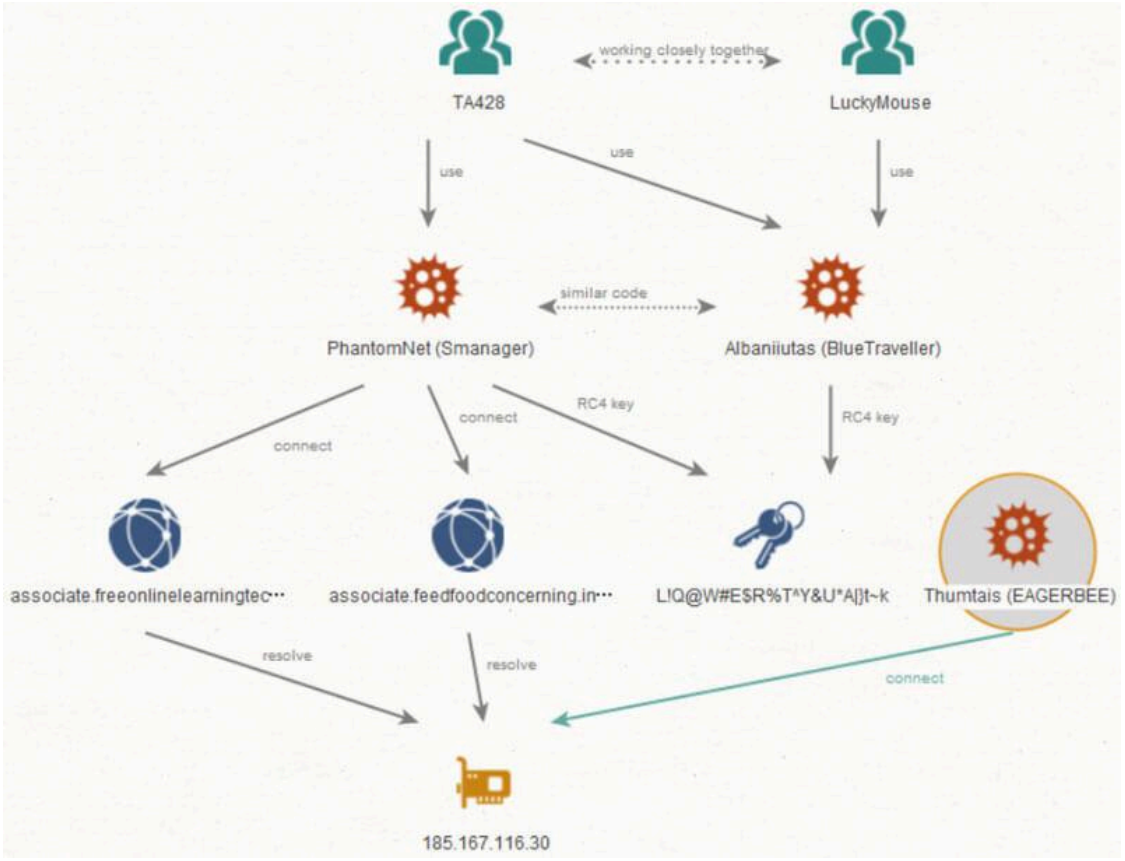


図17 Thumtaisと通信先の関連性

```
0000000018017AD40  47 65 74 50 6C 75 67 69  6E 49 6E 66 6F 6D 61 74  GetPluginInfomat
0000000018017AD50  69 6F 6E 00 00 00 00 00  47 65 74 52 65 67 69 73  ion.....GetRegis
0000000018017AD60  74 65 72 43 6F 64 65 00  00 00 00 00 00 00 00 00  terCode.........
0000000018017AD70  47 00 65 00 74 00 20 00  66 00 75 00 6E 00 63 00  G.e.t.·.f.u.n.c.
0000000018017AD80  74 00 69 00 6F 00 6E 00  20 00 61 00 64 00 64 00  t.i.o.n.·.a.d.d.
0000000018017AD90  72 00 65 00 73 00 73 00  20 00 65 00 72 00 72 00  r.e.s.s.·.e.r.r.
0000000018017ADA0  6F 00 72 00 20 00 2C 00  20 00 74 00 68 00 65 00  o.r.·.,.·.t.h.e.
0000000018017ADB0  20 00 70 00 6C 00 75 00  67 00 69 00 6E 00 20 00  ·.p.l.u.g.i.n.·.
0000000018017ADC0  69 00 73 00 20 00 69 00  6C 00 6C 00 65 00 67 00  i.s.·.i.l.l.e.g.
0000000018017ADD0  65 00 6C 00 21 00 00 00  31 00 32 00 37 00 2E 00  e.l.!...1.2.7...
0000000018017ADE0  30 00 2E 00 30 00 2E 00  31 00 00 00 00 00 00 00  0...0...1.......
0000000018017ADF0  34 00 35 00 34 00 35 00  00 00 00 00 00 00 00 00  4.5.4.5.........
0000000018017AE00  47 65 74 50 6C 75 67 69  6E 4F 62 6A 65 63 74 00  GetPluginObject.
0000000018017AE10  44 65 6C 65 74 65 50 6C  75 67 69 6E 4F 62 6A 65  DeletePluginObje
0000000018017AE20  63 74 00 00 00 00 00 00  6D 61 70 2F 73 65 74 3C  ct......map/set<
```

図18 Thumtaisの通信先に紐づくドメイン名へ通信するマルウェアに含まれる文字列

また、このPhantomNetは、通信先の設定情報がRC4で暗号化されており、暗号キー"L!Q@W#E$R%T^Y&U*A|}t~k"で復号できます。（図19）

この暗号キーの文字列は、Albaniiutasマルウェアがデータをデコードする際に利用するものと同じであり、2つのマルウェアは、同一の開発者によって作成された可能性が伺えます。

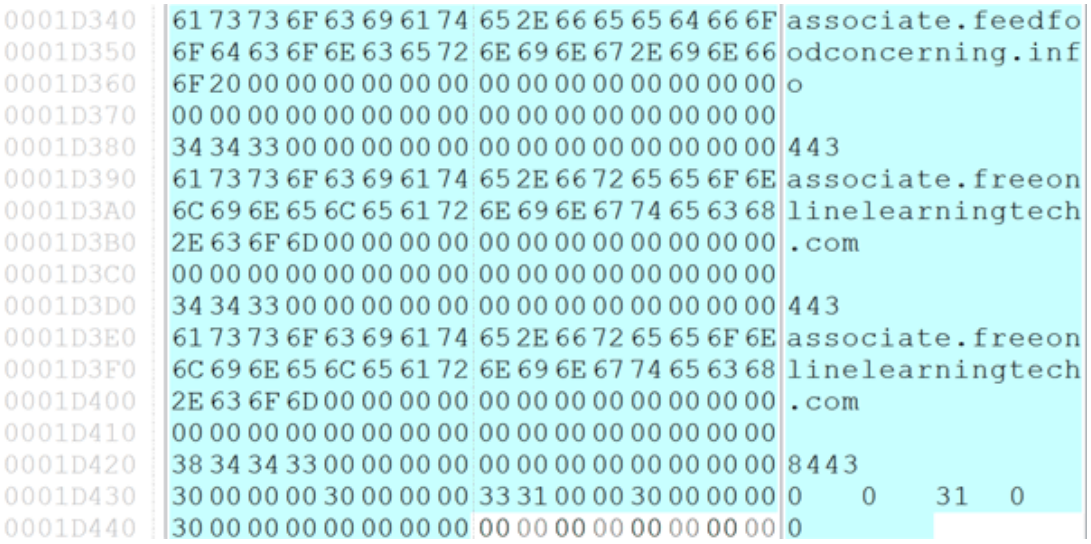


図19 PhantomNetの通信先の設定情報

その他、2022年5月で攻撃に悪用されたThumtaisは、利用されていたデコイファイルやインフラ、VirusTotalへのアップロード状況などを分析するとモンゴル政府関連の関係者を標的としたものであると推測でき、モンゴルを主な標的として狙う"TA428"と特徴が一致することもポイントとして挙げられそうです。

攻撃痕跡の確認と検出

今回紹介したThumtaisは、実行時にカーネルドライバをインストールするため、作成されるカーネルドライバの有無や、レジストリキーなどをチェックすることで、攻撃痕跡を調査することが可能です。以下にその痕跡を確認する方法を一例として紹介します。合わせてThumtaisを検出するための、Yaraルールも記載します。

カーネルドライバの確認

Thumtaisは「c:¥windows¥temp¥」配下に「tmpA8B4f6.tmp」としてカーネルドライバを作成するため、当該ファイルの有無を確認します。（図20）

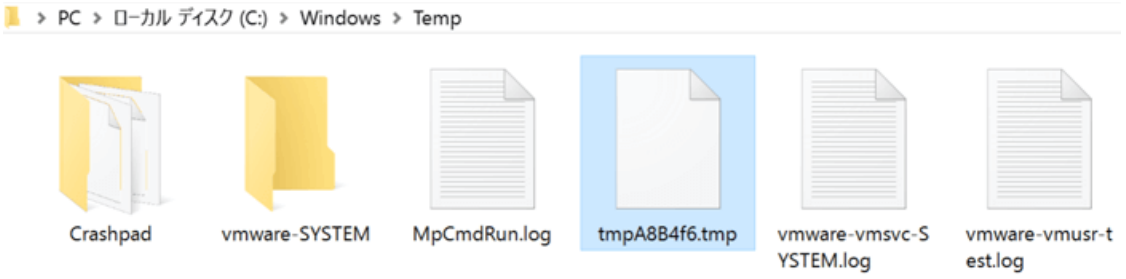
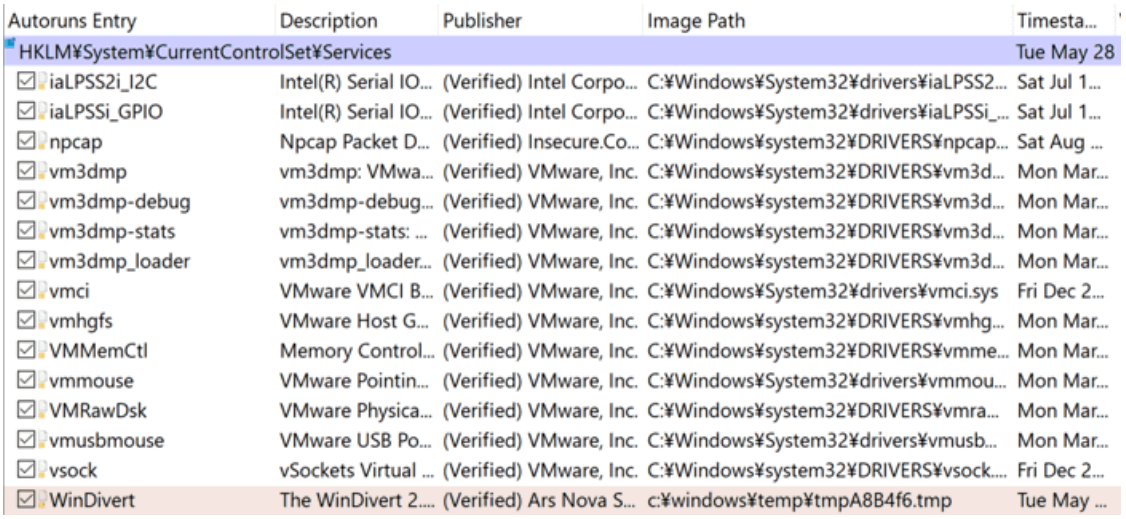


図20 カーネルドライバの作成

Autoruns

Autorunsを利用して、自動起動アプリケーションやレジストリ、ファイルを監査し、不審なプログラムが登録されていないか確認します。Thumtaisは、Windivertカーネルドライバをインストールするため、自動起動エントリにWindivertが登録されます。また、実行ファイルのパスがSystemディレクトリ配下など正しい場所であるかを確認します。（図21）



Autoruns Entry	Description	Publisher	Image Path	Timesta...
HKLM¥System¥CurrentControlSet¥Services				Tue May 28
iaLPSS2i_I2C	Intel(R) Serial IO...	(Verified) Intel Corpo...	C:¥Windows¥System32¥drivers¥iaLPSS2...	Sat Jul 1...
iaLPSSi_GPIO	Intel(R) Serial IO...	(Verified) Intel Corpo...	C:¥Windows¥System32¥drivers¥iaLPSSi_...	Sat Jul 1...
npcap	Npcap Packet D...	(Verified) Insecure.Co...	C:¥Windows¥system32¥DRIVERS¥npcap...	Sat Aug ...
vm3dmp	vm3dmp: VMwa...	(Verified) VMware, Inc.	C:¥Windows¥system32¥DRIVERS¥vm3d...	Mon Mar...
vm3dmp-debug	vm3dmp-debug...	(Verified) VMware, Inc.	C:¥Windows¥system32¥DRIVERS¥vm3d...	Mon Mar...
vm3dmp-stats	vm3dmp-stats: ...	(Verified) VMware, Inc.	C:¥Windows¥system32¥DRIVERS¥vm3d...	Mon Mar...
vm3dmp_loader	vm3dmp_loader...	(Verified) VMware, Inc.	C:¥Windows¥system32¥DRIVERS¥vm3d...	Mon Mar...
vmci	VMware VMCI B...	(Verified) VMware, Inc.	C:¥Windows¥System32¥drivers¥vmci.sys	Fri Dec 2...
vmhgfs	VMware Host G...	(Verified) VMware, Inc.	C:¥Windows¥system32¥DRIVERS¥vmhg...	Mon Mar...
VMMemCtl	Memory Control...	(Verified) VMware, Inc.	C:¥Windows¥system32¥DRIVERS¥vmme...	Mon Mar...
vmmouse	VMware Pointin...	(Verified) VMware, Inc.	C:¥Windows¥System32¥drivers¥vmmou...	Mon Mar...
VMRawDsk	VMware Physica...	(Verified) VMware, Inc.	C:¥Windows¥system32¥DRIVERS¥vmra...	Mon Mar...
vmusbmouse	VMware USB Po...	(Verified) VMware, Inc.	C:¥Windows¥System32¥drivers¥vmusb...	Mon Mar...
vsock	vSockets Virtual ...	(Verified) VMware, Inc.	C:¥Windows¥system32¥DRIVERS¥vsock....	Fri Dec 2...
WinDivert	The WinDivert 2....	(Verified) Ars Nova S...	c:¥windows¥temp¥tmpA8B4f6.tmp	Tue May ...

図21 Autoruns実行結果（Driversサービス登録の確認）

Yaraルール

下記に示すようなYaraルールを利用することで、Thumtaisを検出することが可能です。なお、本検知ルールの利用により過検出が発生する可能性があるため、本番システムへ導入する場合は、事前にテスト、チューニングいただくことをお勧めします。

```
rule Thumtais {
meta:
        description = "Detects Thumtais malware"
        author = "LAC Co., Ltd."

strings:
        $str1 = "mstoolFtip32W" wide
        $str2 = "thumtais2.dat" wide
        $str3 = "iconcache.mui" wide
        $str4 = "iconcache1.tts" wide
        $str5 = "yuijlkhrbgeagf" ascii

condition:
    uint16(0) == 0x5A4D and (4 of ($str*))
}
```

ThumtaisのYaraルール

おわりに

今回は、Thumtaisと背後に潜む攻撃者グループについて紹介しました。Thumtaisは、少なくとも2022年頃から機能をアップデートしながら攻撃に悪用されているマルウェアです。今後も、日本の組織を継続して攻撃してくる可能性も考えられますので、Thumtaisの活動を注視していく必要があると考えます。今回紹介したマルウェアの通信先やハッシュ値などについては、Appendixに記載していますので、対策にご活用いただければ幸いです。

ラックの脅威分析チームでは、今後もこの攻撃者グループおよび利用されたマルウェアについて、継続的に調査し、広く情報を提供していきます。

Appendix

書き換え対象ドメイン名

表2は、ThumtaisにWindivertにより含まれるDNS応答が書き換えられ、名前解決が不可になるドメイン名です。

文字列	文字列	想定されるドメイン名	内容
augur.scanners.	augur.scanners.eset.systems	ESET社の機械学習関連のドメイン名	
ksn-file	ksn-file-geo.kaspersky-labs.com	Kaspersky社のKSN関連のドメイン名	
checkappexec.micr	checkappexec.microsoft.com	Microsoft社製品関連のドメイン名	
networkdevice.scanne	不明	不明	
dc1	不明	不明	
ortex.dat	不明	不明	
ksn-a	ksn-a-stat-geo.kaspersky-labs.com	Kaspersky社のKSN関連のドメイン名	
alprotext1.m	realprotect1.mcafee.com	Trellix社のクラウドベーススキャン関連のドメイン名	
on.ccs.mcaf	ccs.mcafee.com	Trellix社関連のドメイン名	
cloud.gti.mc	cloud.gti.mcafee.com	Trellix社のレピュテーション関連のドメイン名	
protect1.mca	realprotect1.mcafee.com	Trellix社のクラウドベーススキャン関連のドメイン名	
adownload.mca	sadownload.mcafee.com	Trellix社のソフトウェアップデート関連のドメイン名	
.c.eset	c.eset.com	ESET社のLiveGrid関連のドメイン名	
edf.eset.	edf.eset.com	ESET社のアクティベーション関連のドメイン名	
ts.eset.	ts.eset.com	ESET社への情報送信関連のドメイン名	
tscreen.micros	smartscreen-prod.microsoft.com	Microsoft Defender SmartScreen関連のドメイン名	
sn-verdi	ksn-verdict-geo.kaspersky-labs.com	Kaspersky社のKSN関連のドメイン名	

文字列	文字列	想定されるドメイン名	内容
sn-url-	ksn-url-geo.kaspersky-labs.com	Kaspersky社のKSN関連のドメイン名	
sn-cinfo-	ksn-cinfo-geo.kaspersky-labs.com	Kaspersky社のKSN関連のドメイン名	
crc.tren	icrc.trendmicro.com	Trend Micro社のスマートスキャンサービス関連のドメイン名	
url.tren	url.trendmicro.com	Trend Micro社のWebレピュテーション関連のドメイン名	
ensus.tren	*-census.trendmicro.com	Trend Micro社のソフトウェア安全評価関連のドメイン名	
rx.tren	trx.trendmicro.com	Trend Micro社の機械学習検索関連のドメイン名	
dev.drwe	dev.drweb.com	Dr.Web社製品関連のドメイン名	
f2.drw	f2.drweb.com	Dr.Web社製品関連のドメイン名	

表2 書き換え対象ドメイン名

IOC（Indicator Of Compromised）

Indicator	Indicator	Type	Type	Context
8c9b5eca594f4d482f37b936dcdedfd8ea187e6f9361c11e2bcd2ab5625b5233	SHA256	Thumtais loader with an embedded shellcode		
e4d8a4aa2c5c9283c3633a49ce3292b259b3a5c3466706657214c6fb4921808b	SHA256	Thumtais loader with an embedded shellcode		
b01e8dae90ae03fa5f7be5966766fabba06a5dff60682342a6e21fd6c4e8d82e	SHA256	Thumtais loader with an embedded shellcode		
1a9d3e5b7a58419ae23036caacfaaaf0e5d2de0e0074fe0322fcd8395d836947	SHA256	Thumtais loader with an		

Indicator	Indicator	Type	Type	Context
		embedded shellcode		
74e7d0b47ea78f456f1ea1008b73dc3f16a7fbc8f81788215864764bfde9d367	SHA256	Thumtais loader without a shellcode		
499440d6786493664b1fb5ecc7c218e1420a6ec44ac6917d4cbcc6a15649d7b2	SHA256	Thumtais loader without a shellcode		
595d577cdbc24296898a3ab0f434edd6ea56e09f8f1bab5d11ddd968de982010	SHA256	Thumtais loader without a shellcode		
5df5b5cb0471d68abac9dcc7d96b2c090e747f051125214fa61bd0a20db5a2d9	SHA256	Thumtais loader without a shellcode		
9617a5e37b2f68bb14527c98676c0ea5585cbce82e718ce2ae106041b374fe10	SHA256	Thumtais loader without a shellcode		
1d0fd0d97b5753c0492332a1fd41607b473ed02852ce540487894a3ae9cacca8	SHA256	Thumtais loader without a shellcode		
2e72f9eea13d6ee6ff64d28b86cfa4801b5dd650990b15a76556839d9fcd8463	SHA256	Thumtais loader without a shellcode		
7b8284881993d3163ee64a8d400a160984304cdd213d28b4a2a9216a133405e3	SHA256	Thumtais loader without a shellcode		
c733bc7c3f0511b378f8d6b6e719c28677caad4991427eff3e8374a1ca4bf8a9	SHA256	Thumtais loader without a shellcode		

Indicator	Indicator	Type	Type	Context
c42f5650f04fa413266e4597d64bd8a0c7da10668420d8ec179225b43ac74878	SHA256	Thumtais loader without a shellcode		
5693a0d76211c32f777bce0daaa2cb3733ec9312296b95e43fdab87e06cb17b4	SHA256	Thumtais loader without a shellcode		
1d78b4de1b283fd622633d45e5c82bf02e03727f390e4fcfdc87a5190a49b8ff	SHA256	Thumtais loader without a shellcode		
d441eaed23ddfc8e0735175c438f55cc6e5de604aa20b2c0c7ef72f513c41d25	SHA256	Thumtais loader without a shellcode		
09325f01a5ab0fd811516d5204ba5f96c901b56c8e36196d074ccafe7745d733	SHA256	Thumtais loader without a shellcode		
f20cc0f6aa7a12e5925c76cdcde850d2d860a755151109bbae38ed38569d37e8	SHA256	Thumtais loader without a shellcode		
85cc39359d942df10d3455a45b51ab6e5843f8b5b56b010a6578c8e9d27d8938	SHA256	Thumtais loader without a shellcode		
99a3bf0cdef86e3e5bfdaca1478bacffdcb3b4fe569f9002891ec7c141f20e24	SHA256	Thumtais loader without a shellcode		
65387e02258e5e7e31c3974fdafdb0b627d65b89b9e107d47b82d4507d9e42c0	SHA256	Thumtais loader without a shellcode		

Indicator	Indicator	Type	Type	Context
52adc3140cc1fcbef0092132051d9a6fa4eed33dff65e9aa24d604d08603c91e	SHA256	Thumtais loader without a shellcode		
506e85fd089280f4b50038588262d08949e9e83c9142ada445c71e3ee2acf45c	SHA256	Thumtais loader without a shellcode		
35559bfc5c3266b3e01b9a8729c007eb2c56d50da7ae5315e6d55d176f1dd436	SHA256	Thumtais loader without a shellcode		
ab64999a4e523295ba54176d01799a2938b62917bf3288d6defe20634724feb1	SHA256	Thumtais loader without a shellcode		
b88ae29c829e2fb554369fd759b2de7512d49246680b8023aefa32846b1aac7c	SHA256	Thumtais loader without a shellcode		
a787707904f99cba013d37389c64ed8bb85c4b0bc58194dd592d99fe79067fcf	SHA256	Thumtais loader without a shellcode		
ce4dfda471f2d3fa4e000f9e3839c3d9fbf2d93ea7f89101161ce97faceadf9a	SHA256	Thumtais shellcode		
6b9883db8b58b06601bb2fe390f81f2fb6916a7a30ff463b89faf3f3389eae85	SHA256	Thumtais shellcode		
185.167.116[.]30	IP	C2 for Thumtais		
210.1.226[.]238	IP	C2 for Thumtais		
217.197.161[.]85	IP	C2 for Thumtais		
185.82.219[.]204	IP	C2 for Thumtais		