PUBLIC

30 YEARS OF CONTINUOUS IT SECURITY INNOVATION

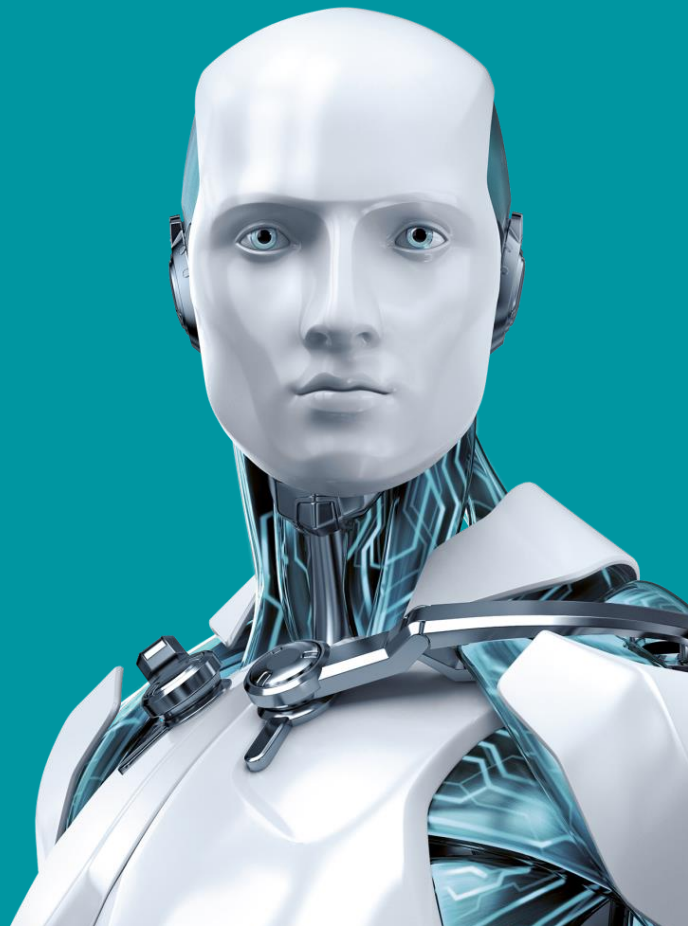
Herzlich Willkommen!

- Ton über „Communicate > Audio Connection“ aktivieren

- Auswahl PC-Ton oder Telefon
(man kann unter Angabe der eigenen Nummer sich zurückrufen lassen)

- WICHTIG! Fenster „Audio Options/Connection“ danach schließen

- Bitte eigenes Mikrofon stummschalten / muten



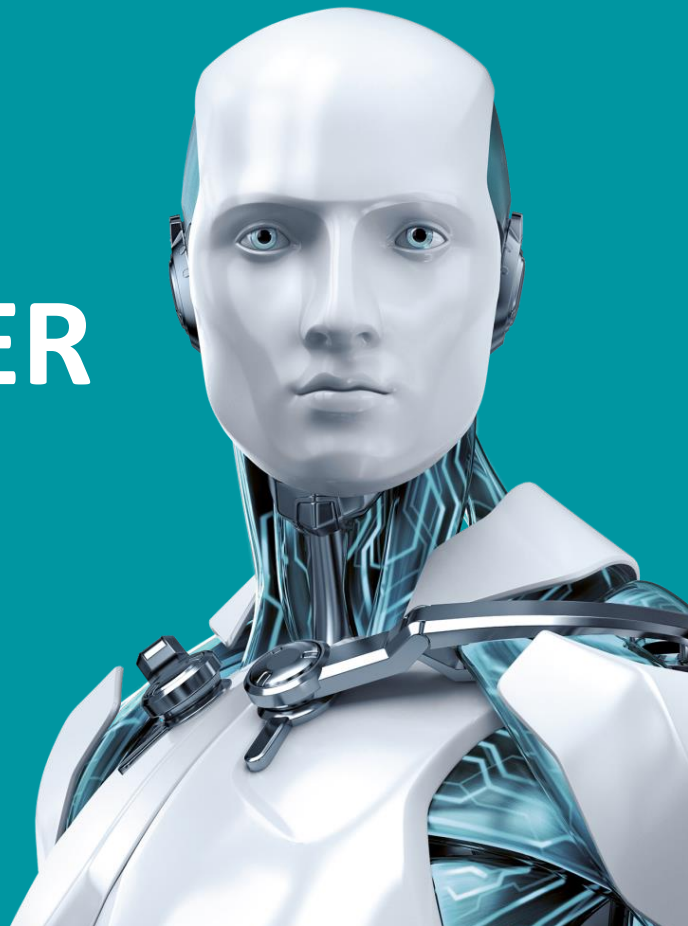
ENJOY SAFER TECHNOLOGY™

PUBLIC

30 YEARS OF CONTINUOUS IT SECURITY INNOVATION

WIN32/INDUSTROYER

Die größte Gefahr für KRITIS seit Stuxnet



ENJOY SAFER TECHNOLOGY™

Thomas Uhlemann

Security Specialist
ESET Deutschland GmbH

PUBLIC

PUBLIC

21 offices

HQ
Bratislava (SK)

Regional centres
San Diego (US)
Buenos Aires (AR)
Singapore (SG)

Branches
Kosice (SK)
Munich (DE)
Melbourne (AU)

Local Offices
Prague (CZ)
Jablonec nad Nisou (CZ)
Sao Paulo (BR)
Jena (DE)
Krakow (PL)
Sydney (AU)
Taunton (GB)
Bournemouth (GB)
Toronto (CA)
Montreal (CA)
Iași (RO)
Mexico City (Mexico)
Zilina (SK)
Brno (CZ)

Globale Präsenz, globaler Schutz

100.000.000+

GESCHÜTZTE ANWENDER WELTWEIT

INDUSTROYER

AGENDA

1. Einleitung und Übersicht
2. Backdoors
3. Launcher
4. Payload

AGENDA

5 Wiper

6 Zusätzliche Tools

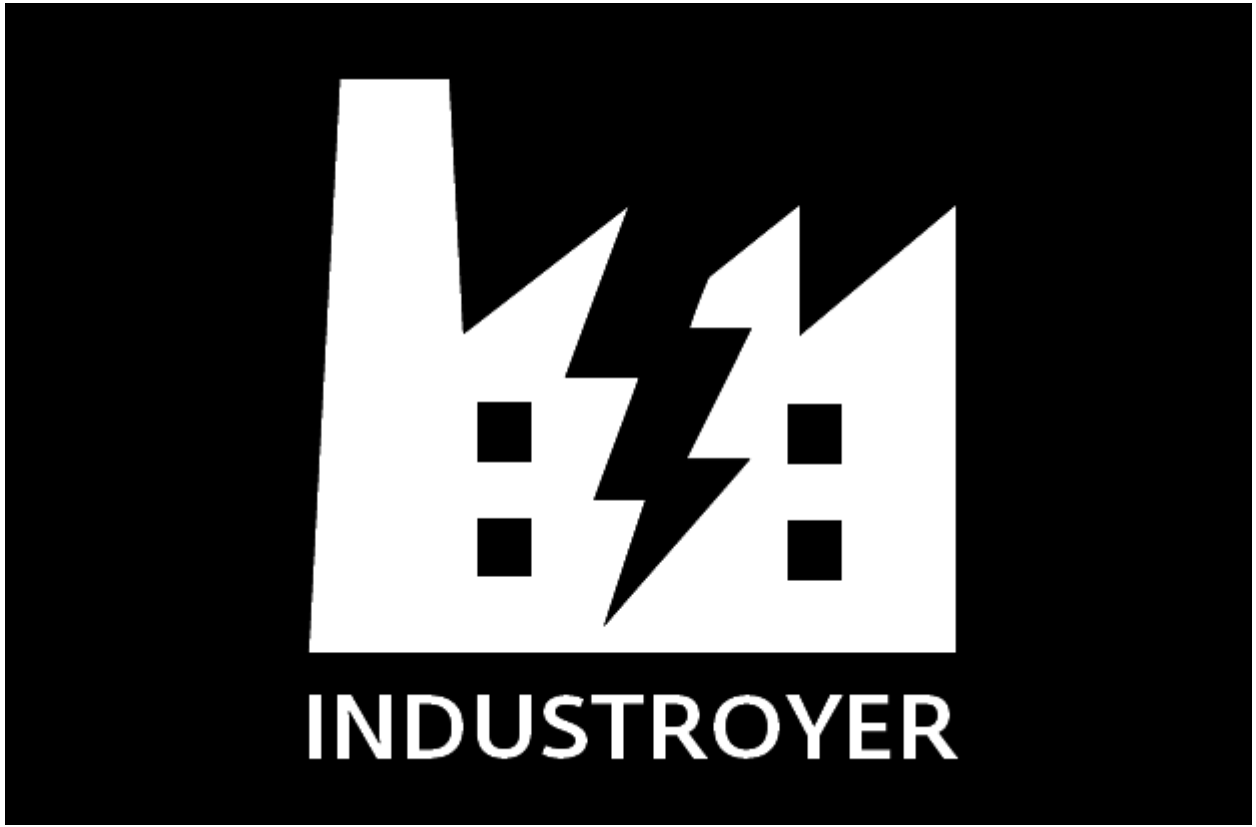
7 Fazit

8 IoC

EINLEITUNG

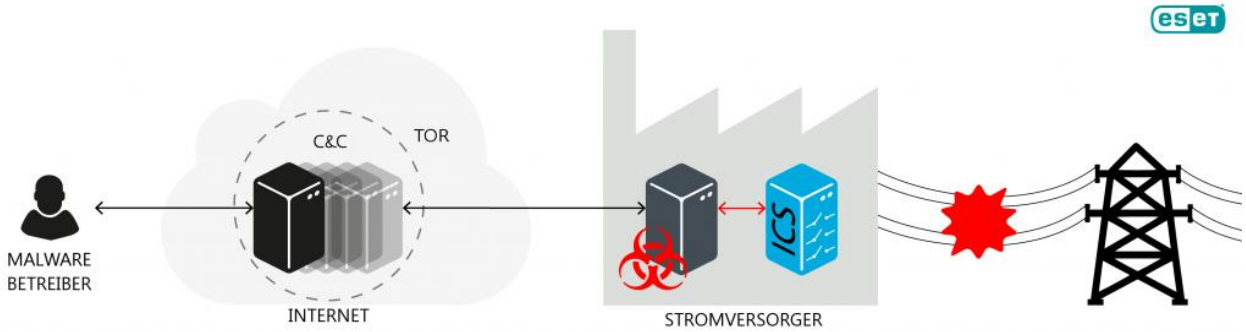
ÜBERSICHT

Win32/Industroyer – eine neue Bedrohung für industrielle Steuerungssysteme



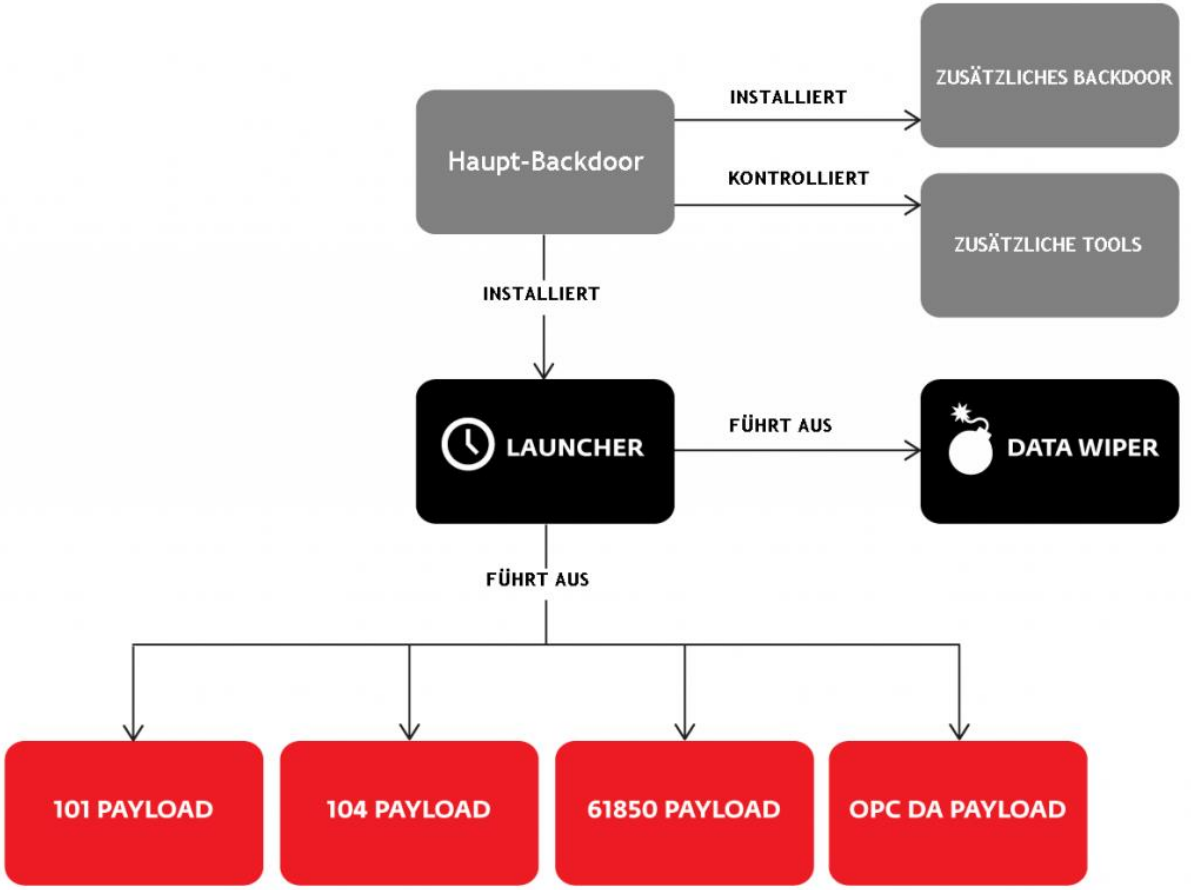
ÜBERSICHT

Win32/Industroyer – entwickelt, um Stromversorger aus der Ferne anzugreifen und vom Netz zu nehmen



ÜBERSICHT

Win32/Industroyer – verfügt über ein mehrstufiges Konzept mit mehreren Backdoors, Payload, Data Wiper & anderen Tools



WAS IST SO BESONDERS AN INDUSTROYER?

- Am 23.12.2015 kam es zu einem stundenlangen Stromausfall in mehreren Regionen der Ukraine.
  Verantwortlich war “Black Energy”.
- Am 17.12.2016 folgte ein neuer Angriff in Kiew, der für ca. 75min große Teile der Stadt im Dunkeln ließ.
- Darauf folgend analysierte ESET mehrere Samples einer neuen Kampagne, die sehr wahrscheinlich dafür verantwortlich sind: “Industroyer”.
- Der Infektionsweg ist bisher unbekannt.

WAS IST SO BESONDERS AN INDUSTROYER?

- Im Gegensatz zu Stuxnet und Black Energy finden sich bei Industroyer innerhalb der Malware Funktionen die direkt mit Schaltern und Stromkreis-Unterbrechern in Kraftwerken kommunizieren können. Zudem ist es in der Lage, längere Zeit unbemerkt im Netzwerk zu existieren.
- Durch die enthaltenen Module ist es Angreifern möglich, von lokaleren, kleinen Blackouts bishin zu Kaskaden oder noch größerem Schaden, die Kampagne maßzuschneidern.
- Die Gang hinter Industroyer muss über tiefes Wissen über industrielle Steuerungsanlagen und über entsprechende Testmöglichkeiten verfügen.

BACKDOORS

HAUPT-BACKDOOR

Die Kernkomponente von Industroyer.

- Hauptkomponente zur Kontrolle aller weiteren Tools
- Verbindung zum C&C per HTTPS
- Alle analysierten Samples besitzen hart gecodet die gleiche Proxy-Adresse im lokalen Netzwerk.
- Dadurch ist klar, dass es sich um einen zielgerichteten Angriff handelt.
- Die meisten der verwendeten C&C Server befinden sich im Tor Netzwerk.
- Angreifer können spezifische Zeiten hinterlegen, zu denen die Backdoor aktiv wird (wie etwa außerhalb der Bürozeiten).
- Alle analysierten Samples arbeiten jedoch 24h.
- Daten werden per POST gesendet, wie HW Identifier, Version der Malware und mehr.

HAUPT-BACKDOOR

Die Kernkomponente von Industroyer.

```
int main_loop()
{
  struct _SYSTEMTIME SystemTime; // [esp+0h] [ebp-14h]@4
  DWORD dwMilliseconds; // [esp+10h] [ebp-4h]@2

  SetLastError(0);
  if ( GetLastError() != ERROR_ALREADY_EXISTS )
  {
    dwMilliseconds = 5000;
    SetUnhandledExceptionFilter(TopLevelExceptionFilter);
    if ( !GetSystemMetrics(SM_CLEANBOOT) )
    {
      if ( create_imapi_handle() )
      {
        while ( 1 )
        {
          do
          {
            Sleep(dwMilliseconds);
            GetLocalTime(&SystemTime);
          }
          while ( SystemTime.wHour >= 24u );
          c2_connect_and_execute_cmd(&dwMilliseconds);
        }
      }
    }
  }
  return 0;
}
```

HAUPT-BACKDOOR

Die Kernkomponente von Industroyer.

Command ID	Purpose
0	Execute a process
1	Execute a process under a specific user account. Credentials for the account are supplied by the attacker
2	Download a file from C&C server
3	Copy a file
4	Execute a shell command
5	Execute a shell command under a specific user account. Credentials for the account are supplied by the attacker
6	Quit
7	Stop a service
8	Stop a service under a specific user account. Credentials for the account are supplied by the attacker
9	Start a service under a specific user account. Credentials for the account are supplied by the attacker
10	Replace "Image path" registry value for a service

HAUPT-BACKDOOR

Die Kernkomponente von Industroyer.

- Wenn die Angreifer Admin-Rechte erlangt haben, können sie die installierte Backdoor zu einem Windows Dienst „upgraden“.
- Dazu wird der *ImagePath* in der Registry eines bestehenden, nicht kritischen Windows Dienstes zum Pfad der Backdoor verändert.
- Anschließend ändert sich die Version der Backdoor und ihr Code wird mit „Junk“ Assmebly Instruktionen verschleiert.

```
.text:00403FD2 main_func proc near                 ; CODE XREF: WinMain(x,x,x,x)+14↑p
.text:00403FD2                                     ; .text:004038C4↑p
.text:00403FD2         call    $+5
.text:00403FD7
.text:00403FD7 loc_403FD7:                         ; CODE XREF: main_func+57↓j
.text:00403FD7                                     ; main_func+5F↓j
.text:00403FD7         add     esp, 4
.text:00403FDA         push    ebp                 ; lpOverlapped
.text:00403FDB         mov     ebp, esp
.text:00403FDD         cmp     edx, 142F9F9Ah
.text:00403FE3         jz      short loc_404023
.text:00403FE5         push    ecx
.text:00403FE6         push    ecx
.text:00403FE7         mov     eax, [ebp+10h]
.text:00403FEA         mov     dword_416190, eax
.text:00403FEF         mov     eax, [ebp+8]
.text:00403FF2         mov     dword_416194, eax
.text:00403FF7         mov     eax, [ebp+0Ch]
.text:00403FFA         cmp     edx, 0B5B93EF3h
.text:00404000         jz      short loc_404023
.text:00404002         mov     lpOverlapped, eax
.text:00404007         mov     [ebp-8], eax
.text:0040400A         lea     eax, [ebp-8]
.text:0040400D         push    eax                 ; lpServiceStartTable
```

ZUSATZ-BACKDOOR

„Ersatztür“, falls die Haupt-Backdoor erkannt oder blockiert wurde.

- Ersatz, falls die Haupt-Backdoor ausfällt (erkannt, blockiert, etc.)
- Trojanisierte Version von Notepad.
- Gleiche Funktionalität wie Notepad PLUS Schadcode, der bei jedem Start mit ausgeführt wird.
- Haben die Angreifer Admin-Rechte, tauschen sie Notepad.exe manuell aus.
- Der Schadcode ist stark verschleiert, aber wenn einmal entschlüsselt, kommuniziert die Malware mit einem C&C.
- Dieser C&C ist ein anderer als der der Haupt-Backdoor und von ihm wird Shellcode geladen, der direkt in den RAM geladen und von dort ausgeführt wird.

ZUSATZ-BACKDOOR

„Ersatztür“, falls die Haupt-Backdoor erkannt oder blockiert wurde.

Original Notepad

```
.text:01004AD5     lea     eax, [ebp+var_50]
.text:01004AD8     push    eax
.text:01004AD9     lea     eax, [ebp+h]
.text:01004ADC     push    eax
.text:01004ADD     push    0B0h
.text:01004AE2     push    hWnd
.text:01004AE8     mov     stru_100A680.lStructSize, 58h
.text:01004AF2     mov     stru_100A680.hwndOwner, edx
.text:01004AF8     mov     stru_100A680.nMaxFile, 104h
.text:01004B02     mov     stru_100A500.lStructSize, 28h
.text:01004B0C     mov     stru_100A500.hwndOwner, edx
.text:01004B12     call    esi ; SendMessageW
.text:01004B14     push    [ebp+var_50]
.text:01004B17     push    [ebp+h]
.text:01004B1A     push    0B1h
.text:01004B1F     push    hWnd
.text:01004B25     call    esi ; SendMessageW
.text:01004B27     push    ebx
.text:01004B28     push    ebx
.text:01004B29     push    0B7h
.text:01004B2E     push    hWnd
.text:01004B34     call    esi ; SendMessageW
.text:01004B36     push    ebx
.text:01004B37     call    ds:GetKeyboardLayout
.text:01004B3D     and     ax, 3FFh
.text:01004B41     cmp     ax, 11h
.text:01004B45     jnz     short loc_1004B58
.text:01004B47     push    1
.text:01004B49     push    1
.text:01004B4B     push    0D8h
.text:01004B50     push    hWnd
.text:01004B56     call    esi ; SendMessageW
```

Backdoored Notepad

```
.text:01004AD5     lea     eax, [ebp+var_50]
.text:01004AD8     push    eax
.text:01004AD9     lea     eax, [ebp+h]
.text:01004ADC     push    eax
.text:01004ADD     push    0B0h
.text:01004AE2     push    hWnd
.text:01004AE8     mov     stru_100A680.lStructSize, 58h
.text:01004AF2     mov     stru_100A680.hwndOwner, edx
.text:01004AF8     mov     stru_100A680.nMaxFile, 104h
.text:01004B02     mov     stru_100A500.lStructSize, 28h
.text:01004B0C     mov     stru_100A500.hwndOwner, edx
.text:01004B12     call    esi ; SendMessageW
.text:01004B14     pusha
.text:01004B15     pushf
.text:01004B16     neg     ebx
.text:01004B18     shr     eax, 1
.text:01004B1B     dec     ebx
.text:01004B1C     mov     eax, 17B200AFh
.text:01004B21     mov     edi, 71CFC28h
.text:01004B26     or      edi, dword_10095C7
.text:01004B2C     xor     esi, 1C779E91h
.text:01004B32     xor     eax, eax
.text:01004B34     dec     edi
.text:01004B35     rol     esi, 5
.text:01004B38     and     esi, edi
.text:01004B3A     and     esi, edi
.text:01004B3C     rol     edx, 6
.text:01004B3F     neg     eax
.text:01004B41     xor     esi, eax
.text:01004B43     neg     ebx
.text:01004B45     shr     ebx, 5
.text:01004B48     mov     ecx, 5E95422h
```

LAUNCHER

LAUNCHER

Eigenständige EXE, die die Payloads und den Wiper startet.

- Enthält spezifische Zeitangabe.
- In den analysierten Samples waren das der 17.12.2016 und der 20.12.2016.
- Startet 2 Threads.
- 1. versucht eine Payload DLL zu laden
- 2. wartet 1 oder 2 Stunden (je nach Version) und versucht dann die Wiper Komponente zu laden
- Beide haben höchste Thread Priorität in Windows
- Name der DLL wird per Parameter in einem der Backdoor-Shellcode-Kommandos mitgegeben
- Der Wiper ist immer *haslo.dat*

LAUNCHER

Eigenständige EXE, die die Payloads und den Wiper startet.

- Payload und Wiper sind Standard DLL
- Zum Laden durch den Launcher müssen sie eine *Crash* Funktion exportieren.

```
;
; Export directory for Crash101.dll
;
                dd 0                    ; Characteristics
                dd 5855F8EDh            ; TimeDateStamp: Sun Dec 18 02:48:13 2016
                dw 0                    ; MajorVersion
                dw 0                    ; MinorVersion
                dd rva aCrash101_dll    ; Name
                dd 1                    ; Base
                dd 1                    ; NumberOfFunctions
                dd 1                    ; NumberOfNames
                dd rva off_100355F8     ; AddressOfFunctions
                dd rva off_100355FC     ; AddressOfNames
                dd rva word_10035600    ; AddressOfNameOrdinals
;
; Export Address Table for Crash101.dll
;
off_100355F8    dd rva Crash            ; DATA XREF: .rdata:100355EC↑o
;
; Export Names Table for Crash101.dll
;
off_100355FC    dd rva aCrash           ; DATA XREF: .rdata:100355F0↑o
                                        ; "Crash"
;
; Export Ordinals Table for Crash101.dll
;
word_10035600   dw 0                    ; DATA XREF: .rdata:100355F4↑o
aCrash101_dll   db 'Crash101.dll',0     ; DATA XREF: .rdata:100355DC↑o
aCrash          db 'Crash',0            ; DATA XREF: .rdata:off_100355FC↑o
```

PAYLOAD

101 PAYLOAD

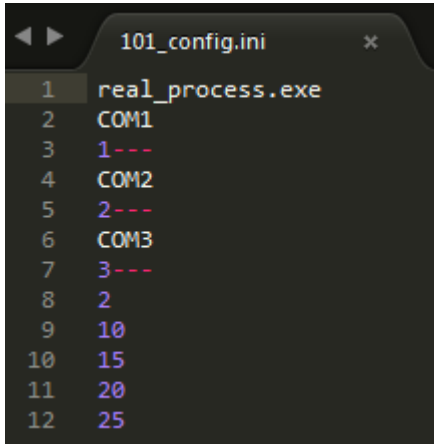
DLL zur seriellen Kommunikation zwischen den Steuerungen und Remote Terminal Units (RTU).

- 101.dll implementiert teilweise das IEC 101 Protokoll zur Steuerung und Überwachung von RTU.
- Kann somit direkt mit den RTU und anderen Geräten, die dieses Protokoll nutzen, kommunizieren.
- Nach Start parst die DLL die Konfiguration ihres INI Files.
- Diese beinhaltet Einträge wie Prozessnamen, Windows Geräte Namen (COM Ports), Information Object Address (IOA) Bereiche, Beginn und Ende der IOA Werte spezifischer IOA Bereiche.
- IOA identifiziert ein spezifisches Datenelement im Gerät.

101 PAYLOAD

DLL zur seriellen Kommunikation zwischen den Steuerungen und Remote Terminal Units (RTU).

Beispiel 101 Payload Config-File mit zwei definierten IOA Bereichen: *10-15* und *20-25*



101 PAYLOAD

DLL zur seriellen Kommunikation zwischen den Steuerungen und Remote Terminal Units (RTU).

- Der Prozessname in der Config entspricht dem Kommunikationsprozess, von dem die Angreifer ausgehen, dass dieser auf dem Zielsystem verwendet wird.
- 101 beendet diesen Prozess und nutzt Windows APIs um selbst mit den RTU zu kommunizieren.
- Kommunikation über den 1. COM Port. Die anderen werden genutzt, um Kommunikation anderer Prozesse zu verhindern.
- Dadurch volle Kontrolle über RTU.
- Hauptziel von 101 ist das Ändern der An/Aus Zustände von Single-Command IOA und Double-Command IOA.

101 PAYLOAD

DLL zur seriellen Kommunikation zwischen den Steuerungen und Remote Terminal Units (RTU).

hex viewer

```
          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F  0123456789ABCDEF
00000000 68 09 09 68 73 01 2e 01  06 00 0a 00 81 34 16     h..hs........4.
```

object tree

```
startByte1 = 0x68 = 104
blockLength = 0x9 = 9
blockLengthCopy = 0x9 = 9
startByte2 = 0x68 = 104
controlField [ControlField]
    dir = false
    prm = true
    fcb = true
    fcv = true
    functionCode = USER_DATA_CONFIRM_EXPECTED (0x3 = 3)
linkAddress = 0x1 = 1
typeIdentification = C_DC_NA_1 (0x2E = 46)
variableStructureQualifierField [StructureQualifierField]
    sq = false
    number = 0x1 = 1
causeOfTransmissionField [CauseOfTransmissionField]
    testBit = false
    positiveNegativeConfirmBit = false
    causeOfTransmission = ACTIVATION (0x6 = 6)
asduAddress = 0x0 = 0
informationObjectAddress = 0xA = 10
dco [DoubleCommandType]
    se = SELECT (0x1 = 1)
    qualifierOfCommand = NO_ADDITIONAL_DEFINITION (0x0 = 0)
    doubleCommandState = COMMAND_OFF (0x1 = 1)
checksum = 0x34 = 52
stopByte = 0x16 = 22
```

104 PAYLOAD

Erweitert 101 um die Möglichkeit der Kommunikation via TCP/IP.

- Erweiterung der Möglichkeiten und somit hohe Konfigurierbarkeit
- Dadurch können die Angreifer die Attacken für jede beliebige Infrastruktur maßschneidern.

104.ini

```
[STATION]
target_ip = 192.168.0.1
target_port = 2404
logfile = logfile.txt
asdu = 1
stop_comm_service = 0
change = 1
first_action = on
silence = 0
uselog = 1
stop_comm_service_name = process01.exe
command_type = def
operation = range
range = 10-15,
```

104 PAYLOAD

Erweitert 101 um die Möglichkeit der Kommunikation via TCP/IP.

- Beim Start liest die DLL ihre Config.
- Der Pfad zur Config wird vom Launcher geliefert.
- Config kann mehrere *Station* Einträge beinhalten, die das Verhalten definieren.

Property	Expected value	Purpose
target_ip	IP address	The IP address that will be used for the communication using IEC 104 protocol standard
target_port	Port number	Self-explanatory
uselog	1 or 0	Enables or disables logging to a file
logfile	Filename	Specifies the filename for the log, if enabled
stop_comm_service	1 or 0	Enables or disables termination of the process
stop_comm_service_name	Process name	Specifies the process name that will be terminated
timeout	Timeout in milliseconds	Specifies timeout between send and recv calls. Default value: 15000
socket_timeout	Timeout in milliseconds	Specify the receiving timeout. Default value: 15000
silence	1 or 0	Enables or disables console output
asdu	Integer	Specifies ASDU (Application Service Data Unit) address also known as sector
first_action	on or off	Specifies the Switch value in ASDU packet for first iteration
change	1 or 0	Specifies that the Switch value in ASDU packet should be inverted during iterations
command_type	def or short or long	Specifies command pulse duration for qualifier of command (QOC)
operation	range or sequence or shift	Specifies iteration type for Information Object Addresses (IOA)
range	Specific format of IOAs	Specifies range of Information Object Addresses (IOA)
sequence	Specific format of IOAs	Specifies sequence of Information Object Addresses (IOA)
shift	Specific format of IOAs	Specifies shift of Information Object Addresses (IOA)

104 PAYLOAD

Erweitert 101 um die Möglichkeit der Kommunikation via TCP/IP.

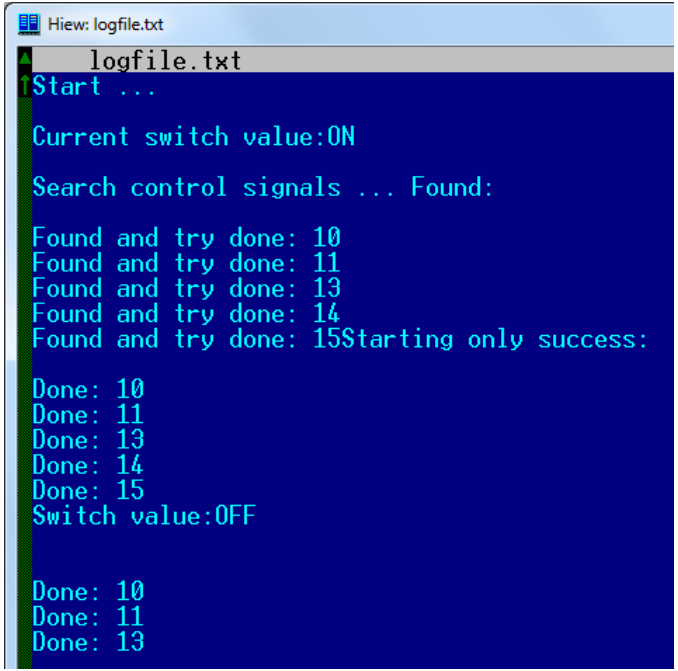
- Für jeden Eintrag wird ein Thread kreiert.
- Über jeden Thread wird kommuniziert.
- Vor der Verbindung wird *D2MultiCommService.exe* gestoppt und der Prozess aus der Config gestartet.
- Anschließend werden IOA auf der Ziel IP gesucht und angesprochen.

```
▷ Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2
▷ Transmission Control Protocol, Src Port: 2404, Dst Port: 49168, Seq: 39, Ack: 45, Len: 16
▷ IEC 60870-5-104-Apci: -> I (2,2)
◢ IEC 60870-5-104-Asdu: ASDU=1 C_SC_NA_1 ActTerm IOA=10 'single command'
    TypeId: C_SC_NA_1 (45)
    0... .... = SQ: False
    .000 0001 = NumIx: 1
    ..00 1010 = CauseTx: ActTerm (10)
    .0.. .... = Negative: False
    0... .... = Test: False
    OA: 0
    Addr: 1
  ◢ IOA: 10
      IOA: 10
    ◢ SCO: 0x01
          .... ...1 = ON/OFF: On
          .000 00.. = QU: No pulse defined (0)
          0... .... = S/E: Execute
```

104 PAYLOAD

Erweitert 101 um die Möglichkeit der Kommunikation via TCP/IP.

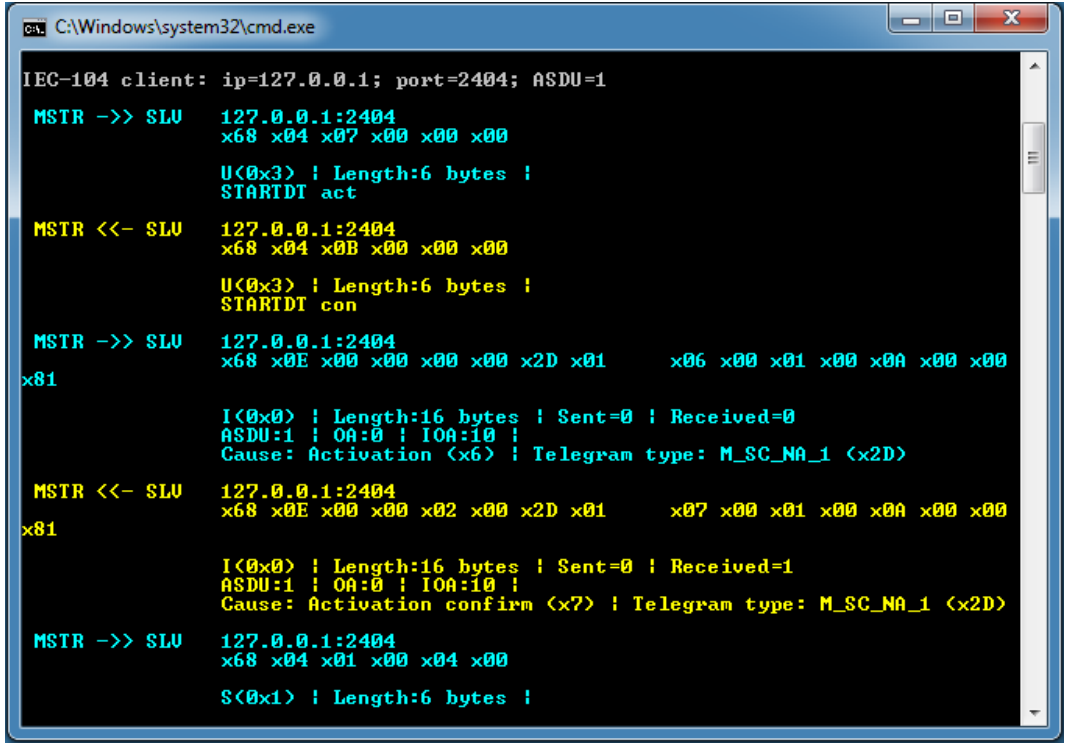
- Bei erfolgreicher Kommunikation wird je nach Config etwa ein Loop gestartet, der u.a. ständig zwischen An/Aus schaltet.
- Die Aktionen werden geloggt.



104 PAYLOAD

Erweitert 101 um die Möglichkeit der Kommunikation via TCP/IP.

- Verschiedene andere Aktionen sind, je nach Config, möglich.
- Unter anderem gibt es auch einen Debug-Output.



61850 PAYLOAD

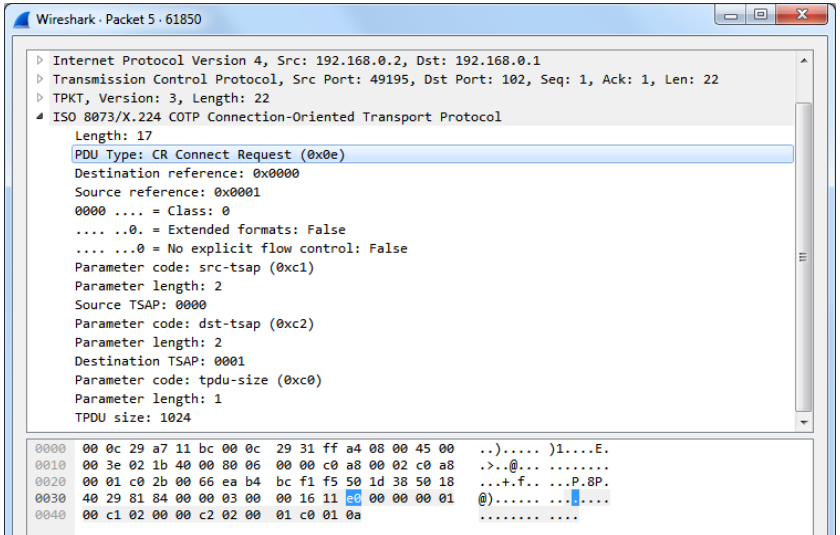
Eigenständige EXE und DLL

- IEC 61850 beschreibt ein Protokoll zur Multivendor-Kommunikation zwischen Geräten zum Schutz, Automatisierung, Messung, Überwachung und Kontrolle von Automatisierungssystemen.
- 61850 reicht ein kleines Subset des an sich recht robusten Protokolls für seinen disruptiven Effekt.
- Beim Start Lesen der Config – Pfad kommt vom Launcher: *i.ini*
- Config enthält eine Liste von IP-Adressen der Geräte, die üblicherweise über den IEC 61850 Standard kommunizieren.

61850 PAYLOAD

Eigenständige EXE und DLL

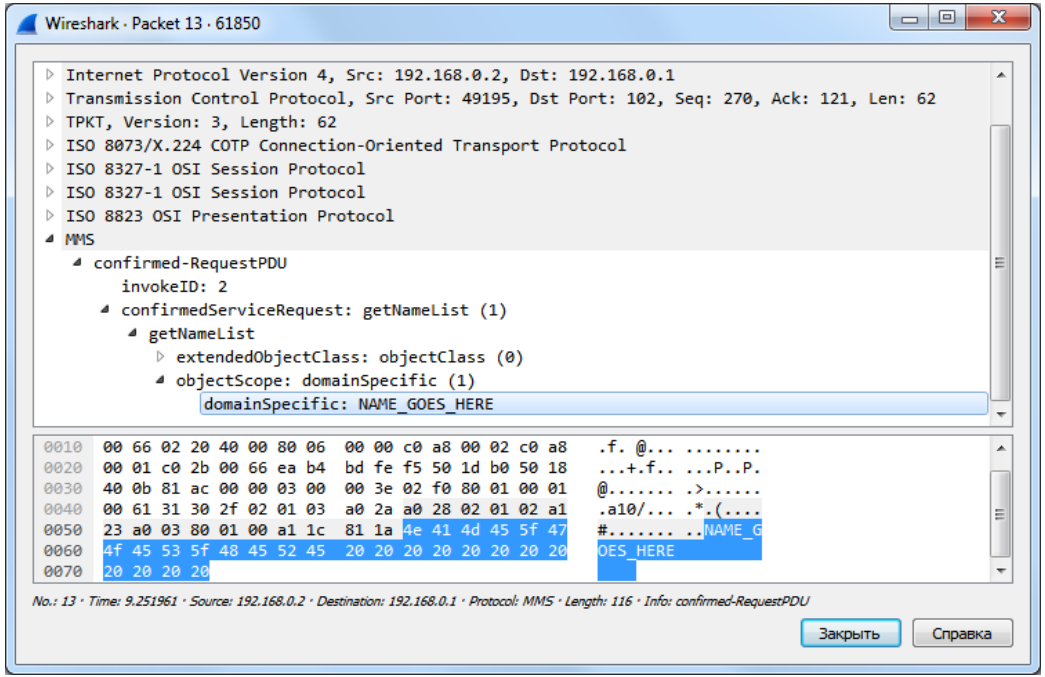
- Falls Config fehlt werden alle verbundenen Netzwerkadapter zur Bestimmung ihrer TCP/IP Subnetz Masken enumeriert.
- Anschließend werden alle möglichen IP-Adressen enumeriert und eine Verbindung über Port 102 versucht.
- Somit können relevante Geräte automatisch im Netzwerk erkannt werden.



61850 PAYLOAD

Eigenständige EXE und DLL

- Kommunikation ermittelt per MMS Variablen mit spezifischen Strings.
- Liest zudem die Stati der einzelnen Nodes.
- Erstellt ein Log dazu.



OPC DA PAYLOAD

Implementiert Client für OLE for Process Controll (OPC) Data Access (DA)

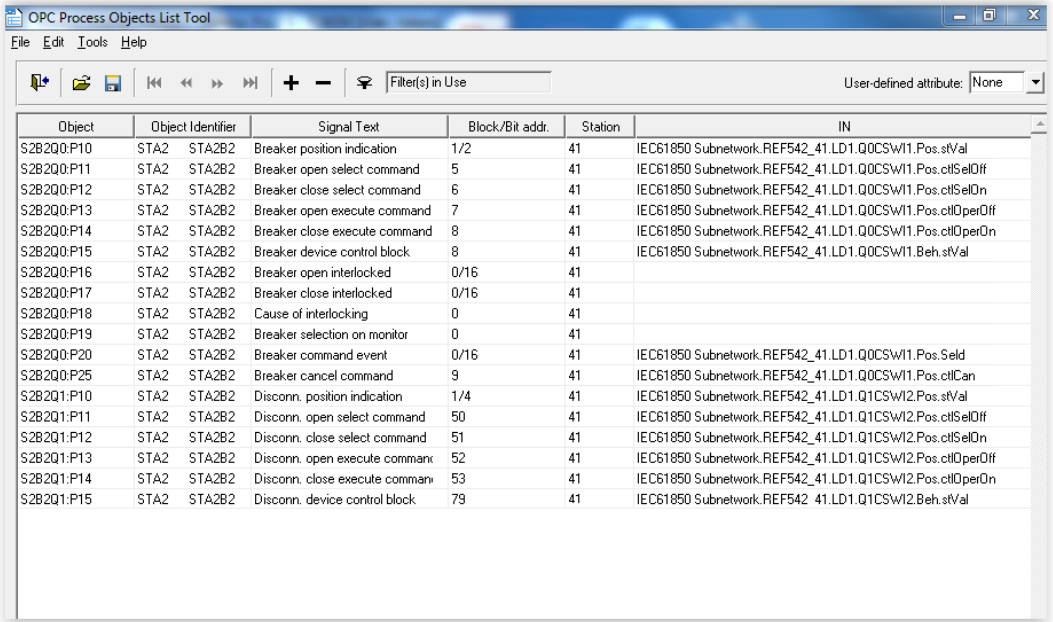
- OPC ist ein Standard basierend auf MS Technologien wie OLE, COM und DCOM
- DA erlaubt Echtzeit-Datenaustausch zwischen verteilten Komponenten, basierend auf Client/Server-Modell
- Malware = Standalone Tool *OPC.exe* und *OPCClientDemo.exe*

```
;
; Export Address Table for OPCClientDemo.dll
;
off_10039678    dd rva Crash            ; DATA XREF: .rdata:1003966C↑o
;
; Export Names Table for OPCClientDemo.dll
;
off_1003967C    dd rva aCrash           ; DATA XREF: .rdata:10039670↑o
                                        ; "Crash"
```

OPC DA PAYLOAD

Implementiert Client für OLE for Process Controll (OPC) Data Access (DA)

- Tool braucht keine Config – enumeriert einfach alle OPC Server und identifiziert die tatsächlich aktiven.
- Schaut nach spezifischen Items – vermutlich gezielt nach denen von ABB Geräten (MicroSCADA).

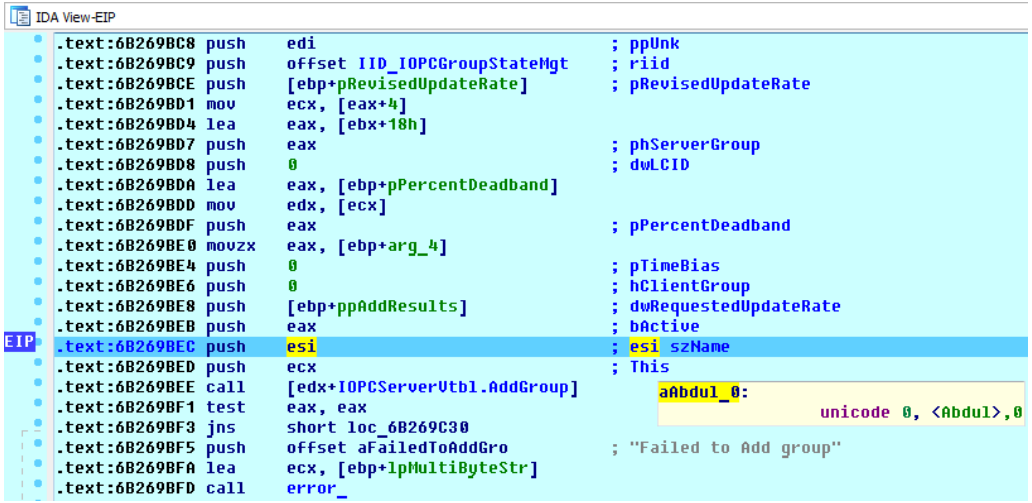


Object	Object Identifier		Signal Text	Block/Bit addr.	Station	IN
S2B2Q0:P10	STA2	STA2B2	Breaker position indication	1/2	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSWI1.Pos.stVal
S2B2Q0:P11	STA2	STA2B2	Breaker open select command	5	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSWI1.Pos.ctlSelOff
S2B2Q0:P12	STA2	STA2B2	Breaker close select command	6	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSWI1.Pos.ctlSelOn
S2B2Q0:P13	STA2	STA2B2	Breaker open execute command	7	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSWI1.Pos.ctlOperOff
S2B2Q0:P14	STA2	STA2B2	Breaker close execute command	8	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSWI1.Pos.ctlOperOn
S2B2Q0:P15	STA2	STA2B2	Breaker device control block	8	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSWI1.Beh.stVal
S2B2Q0:P16	STA2	STA2B2	Breaker open interlocked	0/16	41	
S2B2Q0:P17	STA2	STA2B2	Breaker close interlocked	0/16	41	
S2B2Q0:P18	STA2	STA2B2	Cause of interlocking	0	41	
S2B2Q0:P19	STA2	STA2B2	Breaker selection on monitor	0	41	
S2B2Q0:P20	STA2	STA2B2	Breaker command event	0/16	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSWI1.Pos.SelId
S2B2Q0:P25	STA2	STA2B2	Breaker cancel command	9	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSWI1.Pos.ctlCan
S2B2Q1:P10	STA2	STA2B2	Disconn. position indication	1/4	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSWI2.Pos.stVal
S2B2Q1:P11	STA2	STA2B2	Disconn. open select command	50	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSWI2.Pos.ctlSelOff
S2B2Q1:P12	STA2	STA2B2	Disconn. close select command	51	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSWI2.Pos.ctlSelOn
S2B2Q1:P13	STA2	STA2B2	Disconn. open execute comman	52	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSWI2.Pos.ctlOperOff
S2B2Q1:P14	STA2	STA2B2	Disconn. close execute comman	53	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSWI2.Pos.ctlOperOn
S2B2Q1:P15	STA2	STA2B2	Disconn. device control block	79	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSWI2.Beh.stVal

OPC DA PAYLOAD

Implementiert Client für OLE for Process Controll (OPC) Data Access (DA)

- Angreifer benutzen den String „Abdul“ beim Hinzufügen einer neuen OPC Gruppe.
- Möglicherweise interner Name für ABB



OPC DA PAYLOAD

Implementiert Client für OLE for Process Controll (OPC) Data Access (DA)

- Anschließend wird versucht die Stati der entdeckten OPC Items zu ändern.
- Logfile mit OPC Server Name, OPC Item Status, Qualitäts-Code und Wert wird geschrieben.

```
.text:004034FE        mov     eax, VT_I2
.text:00403503        mov     word ptr [ebp+pItemValues.anonymous_0], ax
.text:0040350A        mov     eax, 1
.text:0040350F        mov     word ptr [ebp+pItemValues.anonymous_0+8], ax
.text:00403516        lea     eax, [ebp+pItemValues]
.text:0040351C        push    eax                             ; pItemValues
.text:0040351D        mov     eax, [ebp+OPC_items]
.text:00403523        mov     ecx, [eax+esi*4]
.text:00403526        call    IOPCSyncIO_Write
.text:0040352B        cmp     esi, edi
.text:0040352D        jb      short loc_403539
.text:0040352F        push    80070057h
.text:00403534        call    throw_exception
```

WIPER

DATA WIPER

Destruktives Modul am Ende der Attacke, zum Verwischen der Spuren und zum Erschweren der Wiederherstellung des Betriebs.

- Komponente heißt *haslo.dat* oder *haslo.exe*
- Liest alle Werte aus *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services*
- Verändert alle *ImagePath* zu leerem String
- System kann so nicht starten
- Prüft alle Laufwerke von C:\ bis Z:\ nach spezifischen Dateitypen (Liste im White Paper)
- Überschreibt gewisse Bereiche dieser Dateien in 2 Versuchen
- Beendet Prozesse bis auf Auschlussliste

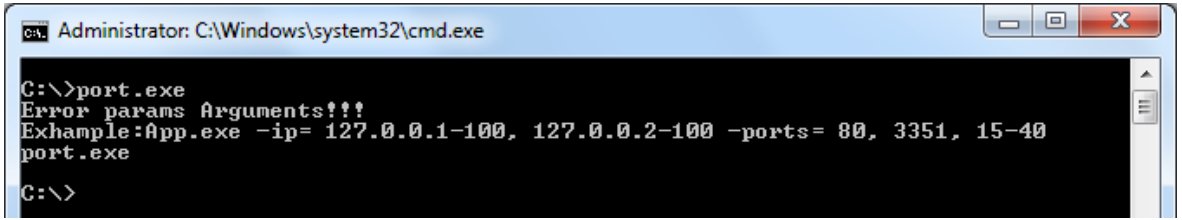
```
off_10010E88    dd offset aAudiodg_exe  ; DATA XREF: _terminate_processes:loc_10001470↑r
                                        ; "audiodg.exe"
                dd offset aConhost_exe  ; "conhost.exe"
                dd offset aCsrss_exe    ; "csrss.exe"
                dd offset aDwm_exe      ; "dwm.exe"
                dd offset aExplorer_exe ; "explorer.exe"
                dd offset aLsass_exe    ; "lsass.exe"
                dd offset aLsm_exe      ; "lsm.exe"
                dd offset aServices_exe ; "services.exe"
                dd offset aShutdown_exe ; "shutdown.exe"
                dd offset aSmss_exe     ; "smss.exe"
                dd offset aSpoolss_exe  ; "spoolss.exe"
                dd offset aSpoolsv_exe  ; "spoolsv.exe"
                dd offset aSvchost_exe  ; "svchost.exe"
                dd offset aTaskhost_exe ; "taskhost.exe"
                dd offset aWininit_exe  ; "wininit.exe"
                dd offset aWinlogon_exe ; "winlogon.exe"
                dd offset aWuauclt_exe  ; "wuauclt.exe"
```

ZUSÄTZLICHE TOOLS

TOOLS

Portscanner

- Das Arsenal der Angreifer beinhaltet einen Portscanner, der eine Übersicht über das angegriffene Netzwerk erzeugt.
- Statt auf vorhandene Scanner zu setzen, ist dies eine Spezialentwicklung.
- Die Angreifer können einen Bereich an IP-Adressen und Ports definieren, der vom Tool gescannt werden soll.



TOOLS

DoS Tool

- Denial of Service (DoS) Tool, dass gegen Siemens SIPROTECT Geräte eingesetzt werden kann.
- Unter Ausnutzung der CVE-2015-5374 Schwachstelle werden die Geräte dazu gebracht, bis zum Neustart keine Befehle mehr annehmen zu können.
- Wenn das Tool gestartet wird, sendet es speziell präparierte UDP Pakete zum Port 50000 der IP-Adresse des anzugreifenden Systems.
- Das UDP Paket enthält lediglich 18 Byte.

```
00000000:  11 49 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00000010:  28 9E
```

FAZIT

FAZIT

- Ziemlich hohe Wahrscheinlichkeit, dass analysierte Samples zum Angriff im Dezember 2016 gehören, da exakt gleicher Aktivierungsstempel und entsprechende Funktionen.
- Win32/Industroyer ist eine hochentwickelte Malware-Sammlung zum Angriff auf Industrielle Steuerungssysteme.
- Die Angreifer selbst können dank der Logs und hohen Konfigurierbarkeit der Payloads die Kampagne theoretisch auf jede beliebige andere Anlage anpassen.
- Die angegriffenen Komponenten wurden teils vor Jahrzehnten entwickelt, ohne dass Security eine Rolle gespielt hätte.
- Patches existieren nur teilweise und werden noch weniger eingespielt.

IOC

IOC

Indicators of Compromise

SHA-1 Hashes:

F6C21F8189CED6AE150F9EF2E82A3A57843B587D
CCCCE62996D578B984984426A024D9B250237533
8E39ECA1E48240C01EE570631AE8F0C9A9637187
2CB8230281B86FA944D3043AE906016C8B5984D9
79CA89711CDAEDB16B0CCCCFDCFBD6AA7E57120A
94488F214B165512D2FC0438A581F5C9E3BD4D4C
5A5FAFBC3FEC8D36FD57B075EBF34119BA3BFF04
B92149F046F00BB69DE329B8457D32C24726EE00
B335163E6EB854DF5E08E85026B2C3518891EDA8

C&C Server*:

195.16.88.6
46.28.200.132
188.42.253.43
5.39.218.152
93.115.27.57

*Da die meisten der Server Teil des Tor Netzwerks waren, kann die Suche nach den Adressen zu False Positives führen.

Basierend auf einer Arbeit von Anton Cherepanov, ESET

https://www.welivesecurity.com/wp-content/uploads/2017/06/Win32-Industroyer.pdf

PUBLIC

30 YEARS OF CONTINUOUS IT SECURITY INNOVATION

VIELEN DANK!

thomas.uhlemann@eset.de

@SecureTommi

WeLiveSecurity.de

ENJOY SAFER TECHNOLOGY™