Bundesamt
für Sicherheit in der
Informationstechnik

Nationales
IT-Lagezentrum
BSI

STUXNET STEP-7 Components and Infection Test

Version: 1.1

Date: 2010-09-30

Contact: stuxnet@bsi.bund.de

Introduction

The STUXNET malware contains three memory sections with executable code for the Siemens SIMATIC STEP-7 programmable logic controller (PLC). This document covers the current state of analysis of the code within those memory sections and resulting recommendations on how to identify infections on Siemens PLC systems.

All information within this document should be considered preliminary, as the analysis is current work in progress. Please also note that this analysis is independent from another publication (<http://www.symantec.com/connect/blogs/exploring-stuxnet-s-plc-infection-process>).

According to Siemens, the behavioural pattern of Stuxnet suggests that manipulation of PLCs apparently only takes place in plants with a specific configuration. The malware deliberately searches for certain modules and program patterns which apply to a specific production process. This means that STUXNET is apparently targeting a specific process or plant and not a particular brand or process technology and thus not the majority of industrial applications. Siemens offers more information and support on its website at <http://support.automation.siemens.com/WW/view/en/43876783>.

Contents of the Memory Sections

Of the three memory sections with STEP-7 PLC code, the largest is 80.889 bytes long and contains a number of PLC functions as well as what appears to be a large data section in the beginning. The other two memory sections are 17.740 and 19.163 bytes long respectively and both contain the same PLC functions with equal function numbers, function names and instruction sequences. Only immediate values and offsets for access to data blocks differ between the two versions.

It should also be noted that a subset of the functions have the same name as standard IEC library functions, but differ in their function number. Other functions are clearly part of the STUXNET payload code.

Functions

The following functions have been identified within the primary memory section of the STUXNET malware:

Function Number	Function Name	Offset
FC6055 (0x17A7)	SB_DT_TM	0x6FDC
FC6056 (0x17A8)	SB_DT_DT	0x76A8
FC6057 (0x17A9)	EQ_DT	0x7C5C
FC6058 (0x17AA)	DT_DATE	0x7D24
FC6059 (0x17AB)	NA_ME	0x7F4C
FC6060 (0x17AC)	CALC	0x803E
FC6061 (0x17AD)	DONE	0x8F36
FC6062 (0x17AE)	INIT	0x8FB0
FC6063 (0x17AF)	IO_ST	0x97FC
FC6064 (0x17B0)	RD_ST	0xAD2A
FC6065 (0x17B1)	DUMP_DT	0xB0B0

Function Number	Function Name	Offset
FC6066 (0x17B2)	MOD_NM	0xBDD4
FC6067 (0x17B3)	MAIN	0xBFFE
FC6068 (0x17B4)	GET_ST	0xC230
FC6069 (0x17B5)	RD_PH	0xC34E
FC6070 (0x17B6)	AFL_OP	0xC408
FC6071 (0x17B7)	AVERGE	0xCDCE
FC6072 (0x17B8)	PRM_DT	0xD286
FC6073 (0x17B9)	IS_OP	0xD568
FC6074 (0x17BA)	UP_STRING	0xDA32
FC6075 (0x17BB)	LGC_OP	0xE086
FC6076 (0x17BC)	SAV_MOVB	0x10C82
FC6077 (0x17BD)	RND_OP	0x1141E
FC6078 (0x17BE)	SB_DT_NM	0x11704
FC6079 (0x17BF)	CO_DAT	0x1197E
FC6080 (0x17C0)	ROD_NM	0x11AF4
FC6081 (0x17C1)	NR_DT	0x11DD0
FC6082 (0x17C2)	AD_OP	0x11EC0
FC6083 (0x17C3)	TMR_DB	0x12890
FC6084 (0x17C4)	RD_SK	0x12E9A

The following functions are present in both of the smaller memory sections of the STUXNET malware:

Function Number	Function Name	Offset
FC1865 (0x749)	S7_LV	0xD96
FC1866 (0x74A)	WE_TE	0xEA0
FC1867 (0x74B)	RF_GH	0x178A
FC1868 (0x74C)	AD_TT	0x1A8C
FC1870 (0x74E)	HA_FO	0x1EA6
FC1871 (0x74F)	DR_RN	0x1F50
FC1873 (0x751)	S7_WO	0x2284
FC1874 (0x752)	ADD_AC	0x2818
FC1876 (0x754)	DP_SEND (CP_300)	0x28AA
FC1877 (0x755)	RT_OS	0x2CDA
FC1878 (0x756)	SB_DT_TM	0x3852
FC1879 (0x757)	EQ_DT	0x3F1E
FC1880 (0x758)	SB_DT_DT	0x3FE6

Infection Test

The function ADD_AC, FC1874 is used by STUXNET to identify a successful infection. The following STL code shows the implementation of that function:

```
// set ACCU1 and 2 to 0xDEADF007 if DW888.16 is 3 or 4
// otherwise set ACCU1 and 2 to 0

ADD_AC:
          OPN DB888           // DataBlock 888
          L DBW16             // word 888.16
          L W#16#3            // word 3
          <I                  // ACCU2 is less than ACCU1
                              // 3 > 888.16
          JC loc_2840         // jump if RLO=1 (DW888.16 < 3)
                              // (do not jump if DW888.16 is 3 or more)
          TAK                 // exchange ACCU1 and ACCU2
          L W#16#4            // ACCU1 = 4
          >I                  // ACCU2 is greater than ACCU1
                              // 4 < 888.16
          JC loc_2840         // jump if RLO=1 (DW888.16 > 4 )
                              // (do not jump if DW888.16 is 4 or less)
          L DW#16#0DEADF007h
          PUSH                // copy ACCU1 into ACCU2
          BE

loc_2840: L DW#16#0
          PUSH                // copy ACCU1 into ACCU2
          BE
//
// Remark: The STL notation in this listing diverts slightly from the
// standard Siemens notation. The listing is produced by a custom
// CPU module for IDA Pro, which uses more a expressive notation for
// data types, memory references and immediate values.
//
```

The data block 888 is used all over the functions within the smaller memory sections as a central data storage structure. The data word 16 within the data block 888 appears to be a state variable for the malware's general operation. Other functions access this variable and select appropriate behaviour based on its value. Valid state values for this word are between 1 and 5 including.

As the listing above shows, the function returns the magic value 0xDEADF007 to signal a successful infection. The magic value is however only returned when the data word 16 of data block 888 is in state 3 or 4. Therefore, using this function or organization blocks calling this function as an infection test is only valid if the malware currently runs in one of these two states. If the malware is currently in another state (1, 2 or 5), the magic value is not returned.

Infection Verification of STEP-7 Systems

Based on the current results of the analysis, the following tests would be good indicators for a successful infection of a STEP-7 PLC by STUXNET:

- If all of the functions listed above are present on the PLC and were not part of the regular program code, the PLC most likely is infected with STUXNET code.
- If a data block 888 is present on the PLC and a data word at address 16 within this block has a value between 1 and 5 including, this might be an indication of an infection. One should however ensure that the regular program code on the PLC does not make use of a data block 888.

Important: Always consult with the vendor about the correct methods and procedures to determine the presence of the above indicators before executing any tests, and ensure that trained personnel is executing the same under supervision.

Infection of STEP-7 Project Folders

Besides making use of several vulnerabilities in the Microsoft Windows operating system for propagation, Stuxnet also spreads by infecting STEP-7 project folders. This is done by dropping a malicious **`s7hkimdb.dll`** file into all subfolders under the **`hOmSave7`** folder of STEP-7 projects and making use of an insecure library loading attack against the SIMATIC manager (see `<http://www.microsoft.com/technet/security/advisory/2269637.mspx>` for details on this attack method).

Additionally, the following files containing parts of the malware are dropped into subfolders of STEP-7 project folders:

- **`...\XUTILS\listen\XR000000.MDX`**
  (encrypted copy of the main malware DLL – like `%SystemRoot%\inf\oem7A.PNF`)
- **`...\XUTILS\listen\S7000001.MDX`**
  (encrypted configuration data – like `%SystemRoot%\inf\mdmcpq3.PNF`)
- **`...\XUTILS\links\S7P00001.DBF`**
  (encrypted data file – like `%SystemRoot%\inf\mdmeric3.PNF`)

During our analysis of a Stuxnet infection at an affected company in July 2010, it was noticed that all STEP-7 project folders stored on the local harddrive as well as on network shares had been manipulated by Stuxnet in the above mentioned way. Additionally, Stuxnet is also capable of modifying ZIP archives containing (backups of) STEP-7 projects.

We therefore strongly suggest to check all STEP-7 project folders for the presence of the above mentioned files and to be careful when importing STEP-7 projects from external sources.