Stuxnet - Infecting Industrial Control Systems

Liam O Murchu

Sep 2010

Operations Manager, Symantec Security Response

Stuxnet Features

- Attacks industrial control systems
- Spreads via
  - USB drives (autorun.inf / LNK vulnerability)
  - Network shares
  - Windows Printer Spooler vulnerability
  - Windows Server RPC vulnerability
  - WinCC Database servers
  - Step 7 Project files
  - P2P mechanism

Stuxnet Features

- Uses 4 0-day Microsoft vulnerabilities and 1 known Microsoft vulnerability
  - MS10-046 .LNK Vulnerability (autoexecution on USB drives)
  - MS10-061 Print Spooler Vulnerability (remote execution to shared print servers)
  - MS10-073 Win32k Keyboard Layout Vulnerability (local privilege escalation)
  - Unpatched - Task Scheduler Vulnerability (local privilege escalation)
  - MS08-067 Windows Server Service Vulnerability (used in Conficker/Downadup)
- Uses 2 Siemens 'vulnerabilities'
  - Hardcoded username and password in WinCC MSSQL database
  - DLL preloading attack in Step 7 Project files (S7P)
- Uses a Windows rootkit to hide Windows binaries
  - Signed by one of 2 stolen certificates from 'JMicron' and 'Realtek'

Stuxnet Features

- Injects STL code into Siemens PLCs (Progammable Logic Controllers)
- Uses rootkit techniques to hide injected PLC code
  - Patches Siemens Step 7 software, which is used to view PLC code
- Communicates with C&C servers using HTTP
  - www.mypremierfutbol.com
  - www.todaysfutbol.com
- Targeted system likely in Iran

Agenda

1. 60 second Intro to PLCs
2. Programming a PLC
3. How Stuxnet infects
4. What Stuxnet does
5. Demonstration

PLCs

Programmable Logic Controller

- Monitors Input and Output lines
  - Sensors on input
  - switches/equipment on outputs
  - Many different vendors
- Stuxnet seeks specific Models
  - s7-300 s7-400

Stuxnet is Targeted

Targeting a **Specific type of PLC**

Searches for a **Specific Configuration**

Hardware configuration

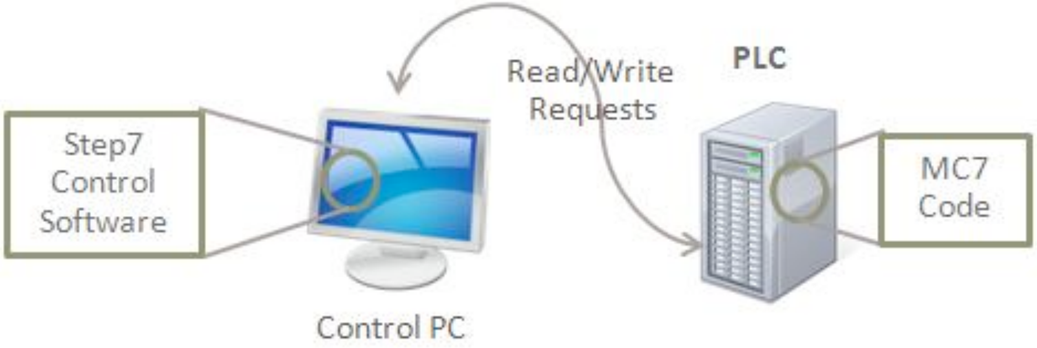
System Data Blocks

- Each PLC must be configured before use.
- Configuration is stored in **System Data Blocks (SDBs)**
- Stuxnet parses these blocks
- Looks for magic bytes **2C CB 00 01** at offset **50h**
- Signifies a Profibus network card attached - CP 342-5
- Looks for **7050h** and **9500h**
- Must have more than **33** of these values
- Injects different code based on number of occurrences

How Stuxnet Infects PLCs

Programming a PLC

Step7, STL and MC7

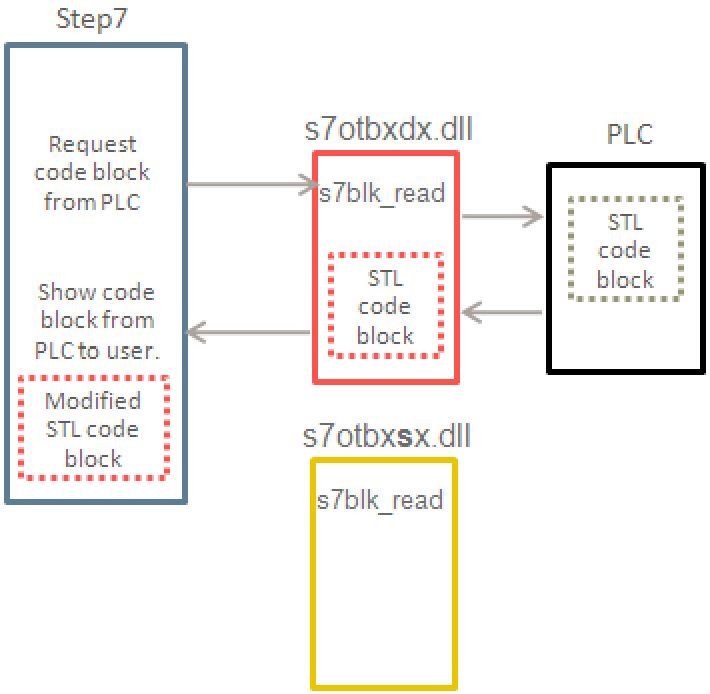


- Simatic or Step 7 software
  - Used to write code in STL or other languages
- STL code is compiled to MC7 byte code
- MC7 byte code is transferred to the PLC
- Control PC can now be disconnected

Stuxnet: Man in the Middle attack on PLCs

"Man in the App" attack

- Step7 uses a library to access the PLC
  - S7otbxdx.dll
- Stuxnet replaces that dll with its own version
- Stuxnet's version intercepts reads and writes to the PLC and changes the code at this point.



Stuxnet MC7 Byte code

- Stuxnet contains at least 70 binary blobs of data
- They are encoded and stored in the fake dll
- These are actually blocks of MC7 byte code
- This is the code that is injected onto the PLCs
- Must be converted back to STL to understand it
- Difficult task but we have now converted all the MC7 byte code to readable STL code
- Just unsure of real world effects of this code.

OB1 and OB35

Stuxnet changes these blocks

- OB1 = main() on PLCs
  - Stuxnet inserts its own code at the beginning of OB1 so it runs first.
- OB35 is a 100ms interrupt routine
  - Used to monitor inputs that would require fast action
  - Stuxnet infects OB35 too
- Stuxnet will return clean versions of these functions when they are read from the PLC.

Demo

Show Infection of a PLC

- Inflate a balloon for 5 seconds
- Infect the PLC
- Inflate balloon again for 5 seconds

Stuxnet’s PLC code

Complex and large amount of code

- Demo was just 8 lines of code.
- Stuxnet contains hundreds of lines of code
- It is difficult to understand the real world actions without knowing what is connected on the inputs and outputs.

```
UC   FC  1865;
POP   ;
L    DW#16#DEADF007;
==D   ;
BEC   ;
L    DW#16#0;
L    DW#16#0;
```

Call function 1865 return value is on the stack

Return value goes into Accu1

Load DEADF007 into Accu1 ACCU1 goes to ACCU2

Are Accu1 and Accu2 equal?

If true exit

Else continue to real OB35

Stuxnet’s PLC code

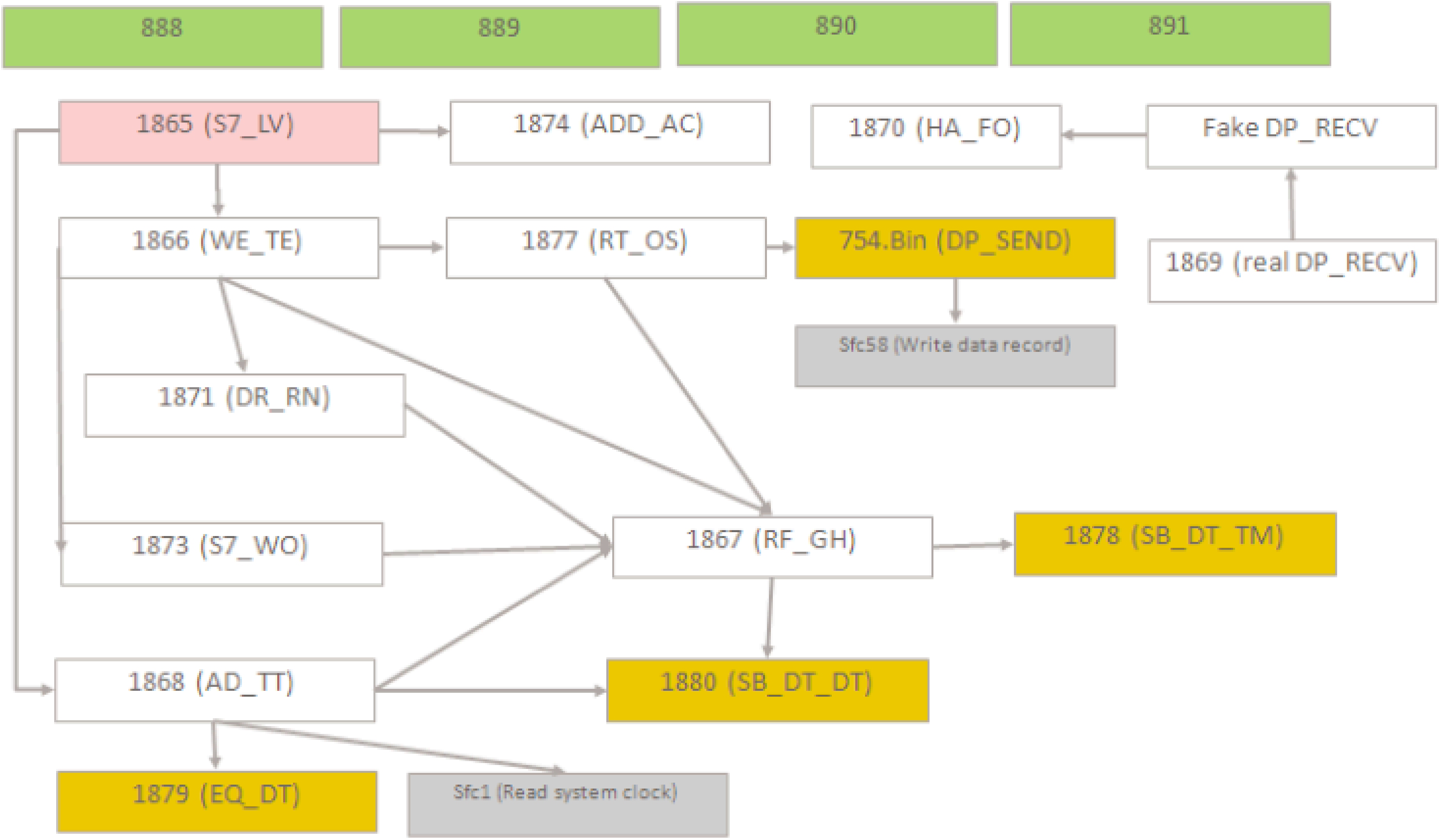
FC 1865

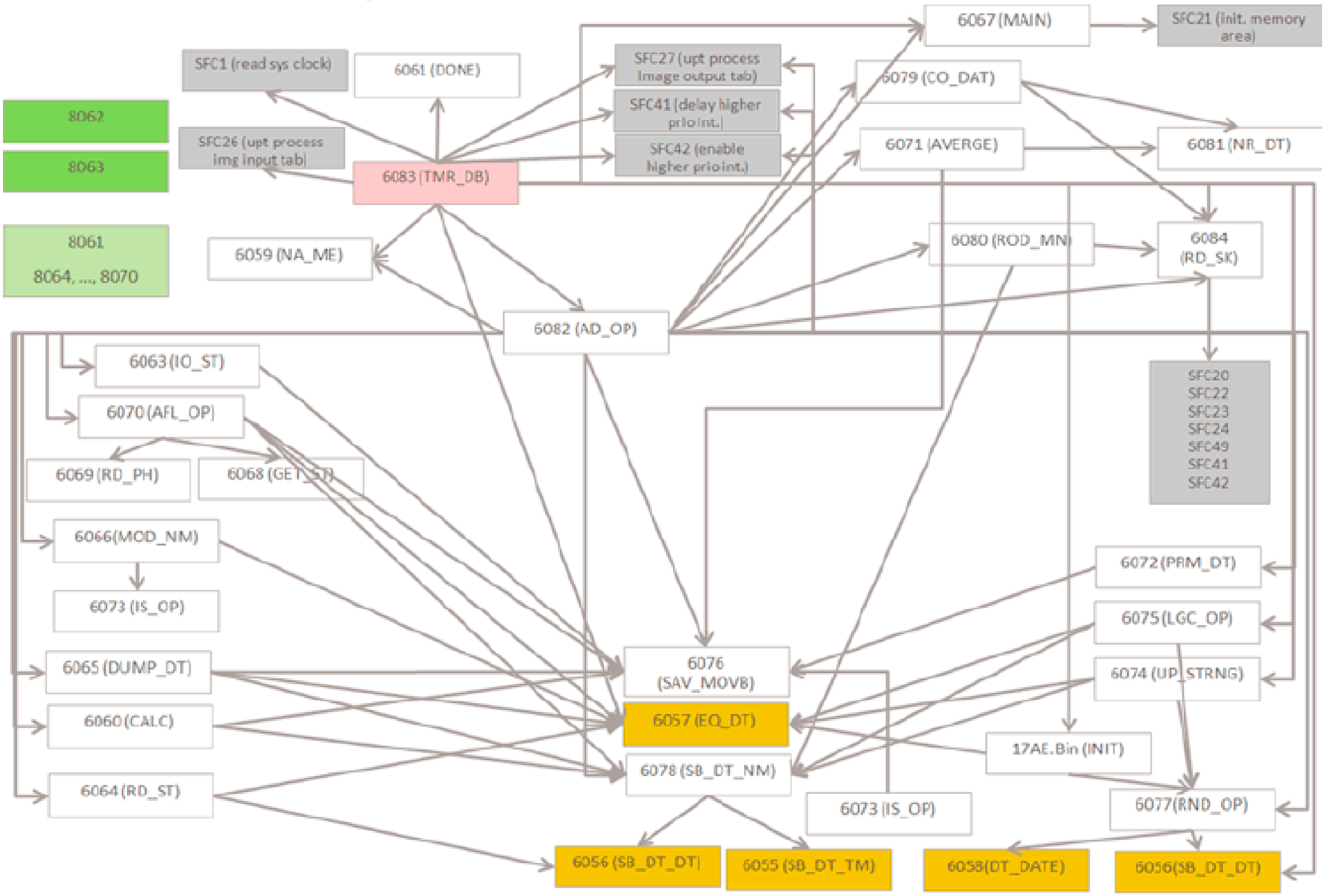
```
M004: CLR   ;
    =     DB888.DBX  642.4;
    UC    FC  1874;
    A     L      2.1;
    SAVE  ;
    BE    ;
END_FUNCTION
```

Stuxnet’s PLC code

FC 1874

```
 L    DB888.DBW   16;
      L    3;
      <I    ;
      JC   M001;
      TAK   ;
      L    4;
      >I    ;
      JC   M001;
      L    DW#16#DEADF007;
      PUSH  ;
      BE    ;
M001: L    DW#16#0;
      PUSH  ;
END_FUNCTION
```


888
889
890
891
1865 (S7_LV)
1874 (ADD_AC)
1870 (HA_FO)
Fake DP_RECV
1866 (WE_TE)
1877 (RT_OS)
754.Bin (DP_SEND)
1869 (real DP_RECV)
Sfc58 (Write data record)
1871 (DR_RN)
1873 (S7_WO)
1867 (RF_GH)
1878 (SB_DT_TM)
1868 (AD_TT)
1880 (SB_DT_DT)
1879 (EQ_DT)
Sfc1 (Read system clock)



6067 (MAIN)

SFC21 (init. memory area)

SFC1 (read sys clock)

6061 (DONE)

SFC27 (upt process image output tab)

6079 (CO_DAT)

8062

8063

SFC41 (delay higher prioint.)

SFC26 (upt process img input tab)

SFC42 (enable higher prioint.)

6071 (AVERGE)

6081 (NR_DT)

6083 (TMR_DB)

8061

8064, ..., 8070

6059 (NA_ME)

6080 (ROD_MN)

6084 (RD_SK)

6082 (AD_OP)

6063 (IO_ST)

6070 (AFL_OP)

SFC20
SFC22
SFC23
SFC24
SFC49
SFC41
SFC42

6069 (RD_PH)

6068 (GET_ST)

6066(MOD_NM)

6072 (PRM_DT)

6073 (IS_OP)

6075 (LGC_OP)

6065 (DUMP_DT)

6076 (SAV_MOVB)

6074 (UP_STRNG)

6057 (EQ_DT)

6060 (CALC)

17AE.Bin (INIT)

6078 (SB_DT_NM)

6064 (RD_ST)

6073 (IS_OP)

6077(RND_OP)

6056 (SB_DT_DT|

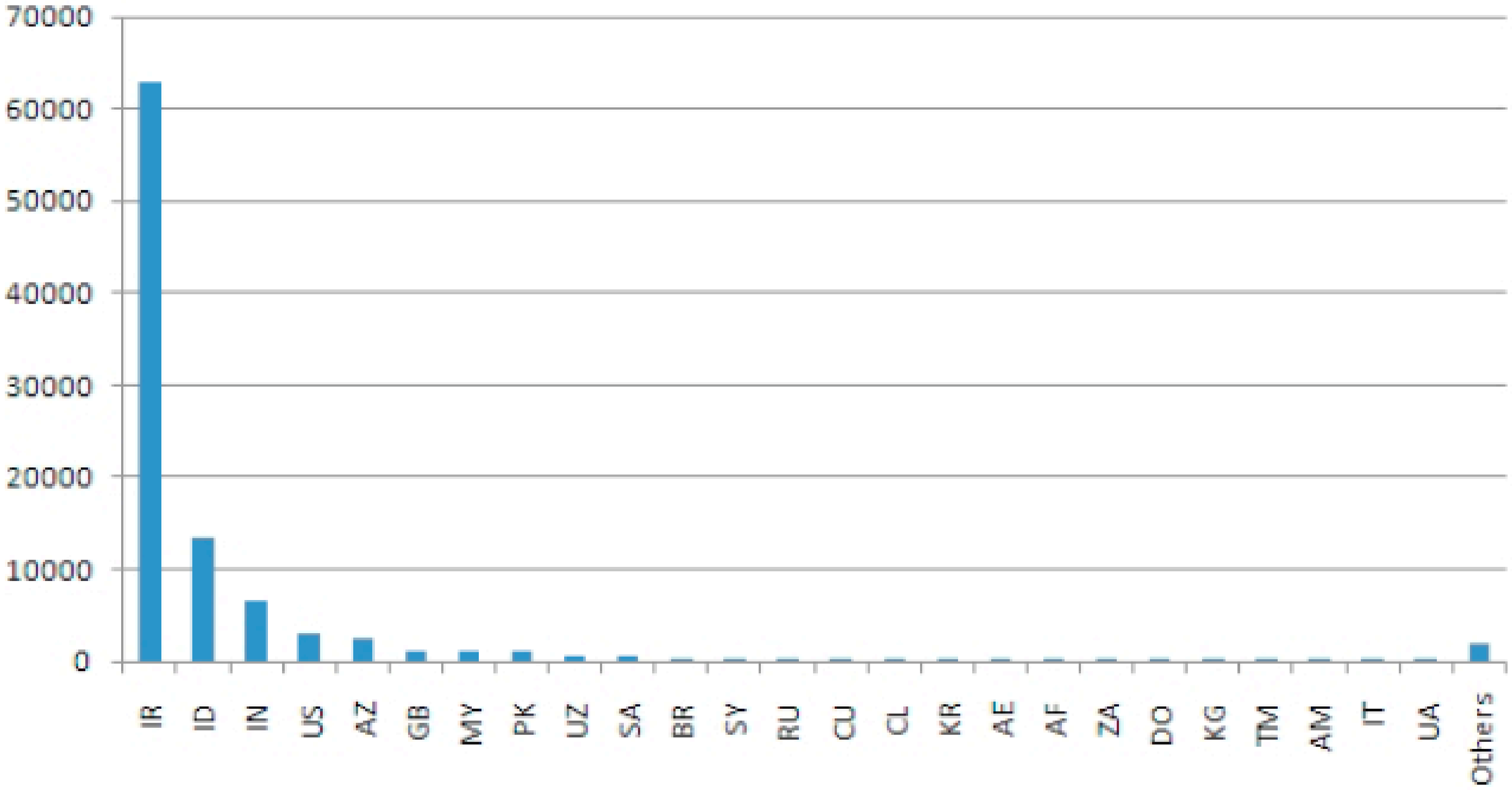
6055 (SB_DT_TM)

6058(DT_DATE)

6056(SB_DT_DT)

Targets

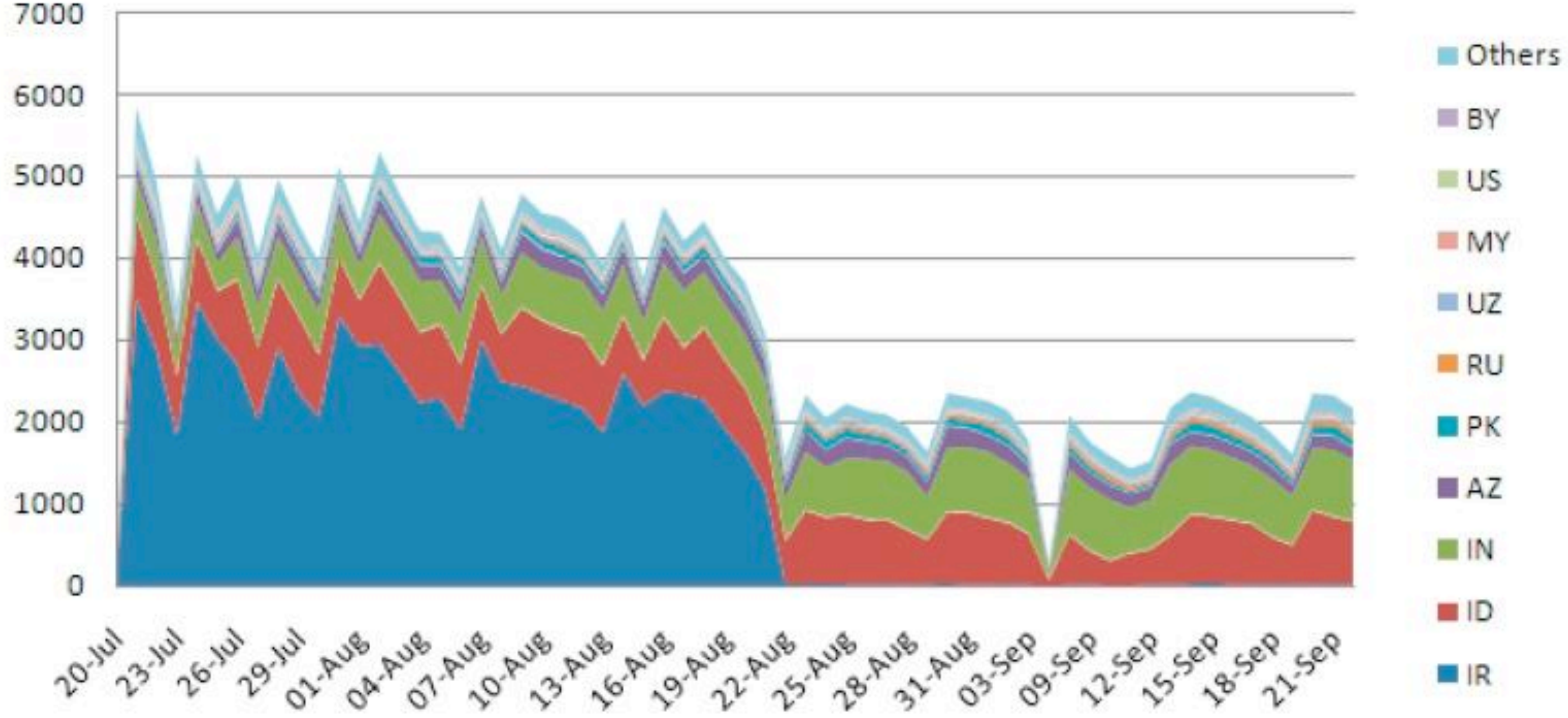
Stats for Command and Control Servers



Stuxnet Infections

Figure 5

Rate of Stuxnet infection of new IPs by Country



White Paper Available

W32.Stuxnet Dossier

- Stuxnet Technical Details Available here:
- http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

Thank you!

Liam O Murchu - liam_omurchu [at] symantec.com

Nicolas Falliere

Eric Chien

Threat Intelligence Team

All Stuxnet Reverse Engineers

This document is provided for informational purposes only and is not intended as advertising. All warranties relating to the information in this document, either express or implied, are disclaimed to the maximum extent allowed by law. The information in this document is subject to change without notice.