TippingPoint
a division of 3Com

Unraveling SCADA Protocols: Using Sulley Fuzzer

DVLabs

- Ganesh Devarajan

DEF CON

TippingPoint
a division of 3Com

Agenda

- Introduction to SCADA networks
  - Overview
  - SCADA Protocols
    - Modbus
    - DNP3
    - ICCP
    - UCA 2.0 and IEC 61850 Standards
- SCADA Security
  - Attack scenarios
  - Past known attacks
- SCADA Fuzzer
- Demo
- Conclusion
- Future work

SCADA Definition

SCADA

- ***Supervisory Control and Data Acquisition*** is defined as a common process control application that collects data from sensors on the shop floor or in remote locations and sends them to a central computer for management and control.
  - Large scale distributed measurement and control system (North America)
  - System that performs SCADA irrespective of the size and geographical distribution (Rest of the world)

    - Wikipedia
- SCADA is a technology that enables a user to collect data from one or more distant facilities and/or send limited control instructions to those facilities – Ronald L. Kurtz

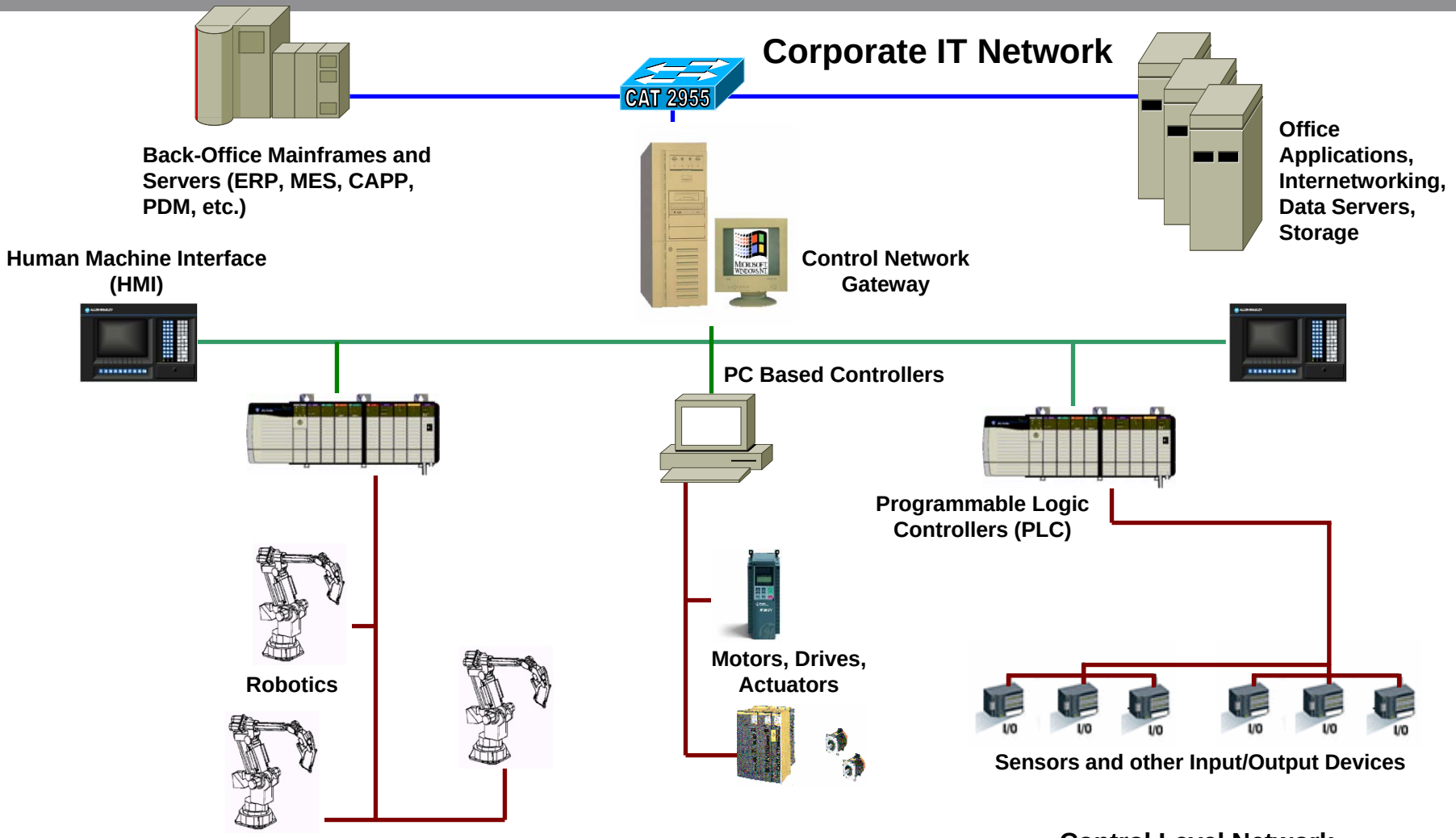
SCADA Use

- It is the vital component of many Critical Infrastructure
- They are used for sensing/managing real-time data
  - Water
  - Gas
  - Electricity
  - Refineries
  - Nuclear plants
  - Other manufacturing operations.

SCADA Infrastructure

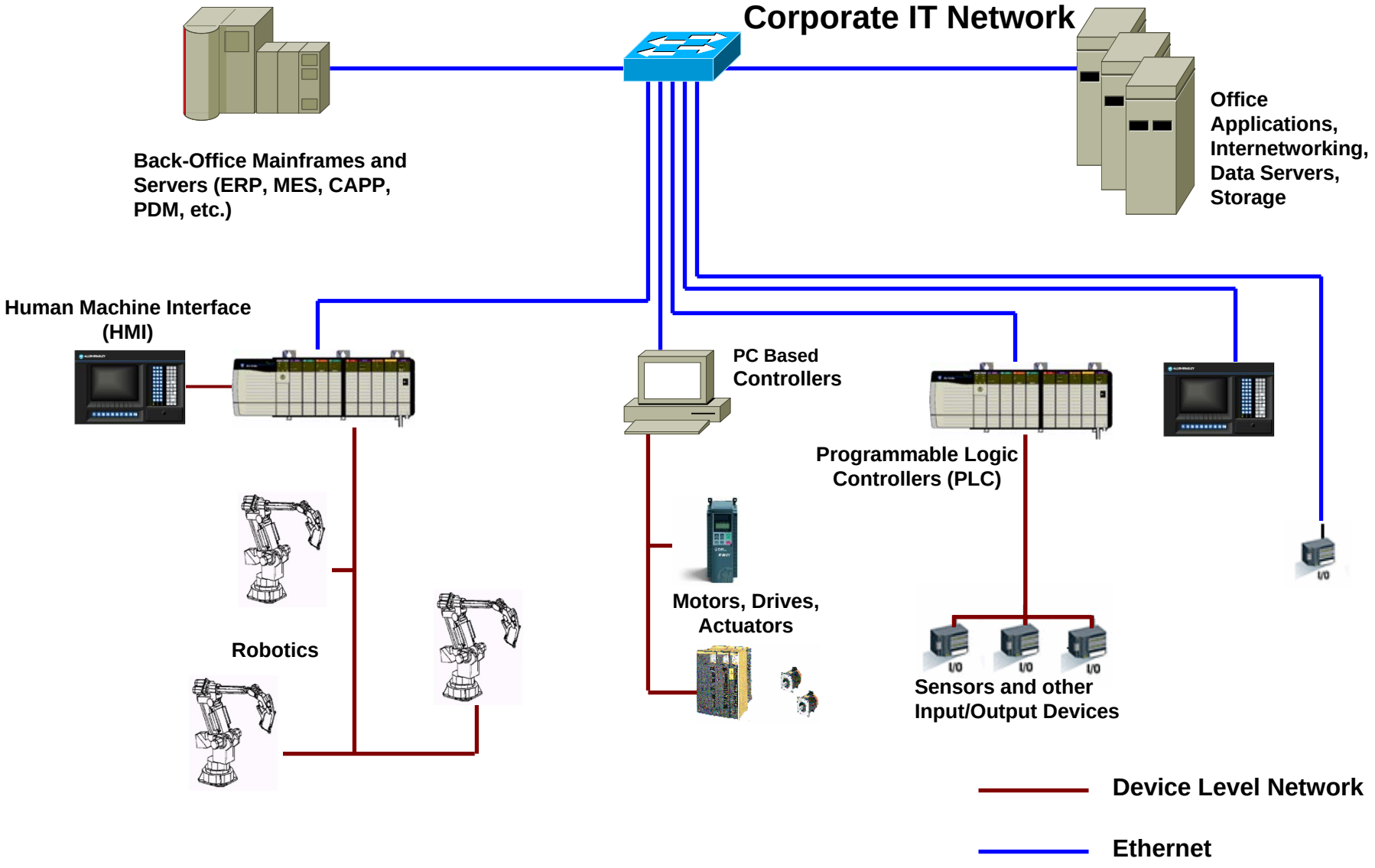
- SCADA System Components
  - Operator – Human
    - Person responsible to make the Supervisory control decisions
    - Might be present in the local or remote site.
  - Human Machine Interface (HMI)
    - Presents data to the user/operator
    - GUIs, Schematics, Windows
  - Master Terminal Unit (MTU)
    - Processes the data and presents it to HMI
    - Collects data from remote sites
    - Sends out queries and instructions to the remote nodes
  - Communication channel
    - Internet, wireless, switched network, etc
  - Remote Terminal Unit (RTU)
    - Sends the abstracted data to the MTU
    - RTU sends out control signals to the sensors and collects data values from them

Traditional SCADA Networks

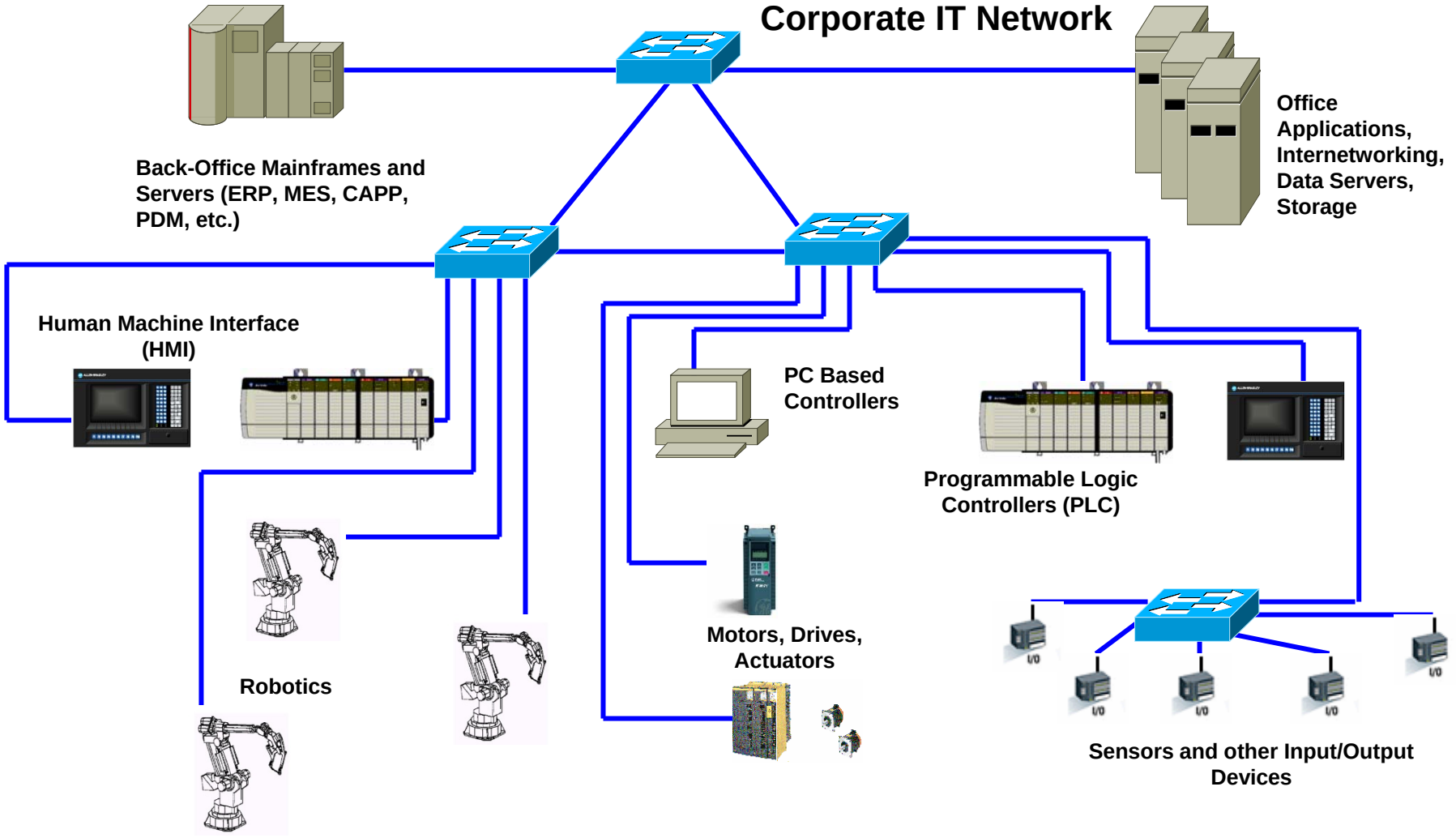


MES: Manufacturing Execution Systems
PDM: Process Data Management
MRO: Maintenance Repair and Overhaul
CAPP: Computer Assisted Process Planning

Off late SCADA Networks



Current/Future SCADA Networks



Need for SCADA Security

The need for security in SCADA systems

- When these protocols were initially created they were proprietary and were not linked to the outside world. But with the improved communication protocols they were exposed more to the Internet. The systems that control our day to day living is exposed to the outside world without any inbuilt security features.
- It is easier to take down the entire country's Critical Infrastructure.
  - Black out
- On a smaller scale you can take down the company's manufacturing plant.
  - The cooling system of the Server room
  - False reports at the manufacturing plant

TippingPoint
a division of 3Com

Current Security Scenario

- Poor Authentication or verification of the client nodes
  - There is no authentication scheme implemented in these protocols
  - Easy to introduce malicious nodes
- Platform Vulnerabilities
  - Windows & Linux Vulnerabilities
  - Not patched regularly – need for maximum uptime
- Vendors and Assert owners believes
  - Under the wrong impression that who is going to hack into these networks
  - Belief that their nodes are not exposed to the outside world
    - What if a malicious node was introduced into the network?
    - A few dangling nodes outside the protected area, that can be used to attack

SCADA Attack Scenarios

- Providing False Data - The functionality of the RTU is to either read or write data into the server and the compromised RTU can write false data into the server.
  - Sensors for Water pollutants
  - Temperature sensors in server rooms
- Denial of Service Attack
  - Continuous sting of reboot command
- Protocol anomalies

SCADA Attacks

- Cyber-Attacks by Al Qaeda Feared
  - Washington Post, June 27, 2002 Mountain View, Calif
- Information-technology contractor Vitek Boden who used his knowledge of control systems to release millions of liters of sewage into drinking water
- Slammer worm affected the operation of the corporate network at Ohio's inactive Davis-Besse nuclear plant and disabled a safety monitoring system for nearly five hours in January 2003
- An hacker took control of the gas pipelines run by Gazprom for around 24 hours in 1999 in Russia

SCADA Protocols

- Modbus
- Distributed Network Protocol 3 (DNP3)
- Inter-Control Center Communications Protocol (ICCP)
- Utility Communications Architecture 2.0 (UCA 2.0) and International Electrotechnical Commission (IEC) 61850 Standards
- Control Area Networks (CAN)
- Control Information Protocol (CIP)
- DeviceNet
- ControlNet
- OLE for Process Control (OPC)
- Profibus

SCADA Protocols - Modbus

- Modbus
  - Developed in the late 1970s by Modicon, Inc.
  - It is commonly used for connecting devices in the Industrial environment
  - It is free and open sourced
- Modbus RTU- Compact Binary format
- Modbus ASCII- more verbose and human readable
- Modbus/TCP is similar to Modbus RTU Data is stuffed into TCP/IP Data packets


```
0000  00 02 b3 ce 70 51 00 20  78 00 62 0d 08 00 45 00   ....pQ.  x.b...E.
0010  00 34 85 83 40 00 80 06  61 05 0a 00 00 39 0a 00   .4..@...  a....9..
0020  00 03 0a 12 01 f6 61 97  f1 83 70 f1 ad 1b 50 18   ......a.  ..p...P.
0030  fa f0 19 52 00 00 00 00  00 00 00 06 0a 08 00 04   ...R....  ........
0040  00 00                                             ..
```

Force Listen Mode

SCADA Protocols - Modbus

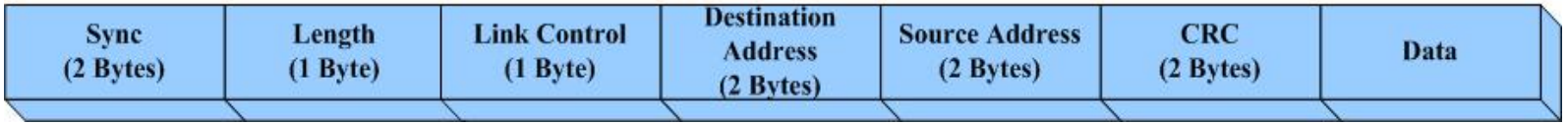
Function Code	Function Name
01	Read Coil Status
02	Read Input Status
03	Read Holding Registers
04	Read Input Registers
05	Force Single Coil
06	Preset Single Register
07	Read Exception Status
09	Program 484
0A	Poll 484
0B	Fetch Communication Event Counter
0C	Fetch Communication Event Log
0D	Program Controller
0E	Poll Controller
0F	Force Multiple Coils
10	Preset Multiple Registers
11	Report Slave ID
12	Program 884/M84
13	Reset Communication Link
14	Read General Reference
15	Write General Reference
16	Mask Write 4X Register
17	Read/Write 4X Registers
18	Read FIFO Queue

SCADA Protocols - Modbus

Function Code	Sub-Function Code	Function Name
08	00	Return Query Data
08	01	Restart Communication Option
08	02	Return Diagnostic Register
08	03	Change ASCII Input Delimiter
08	04	Force Listen Only Mode
08	05-09	Reserved
08	0A	Clear Counters and Diagnostic Reg.
08	0B	Return Bus Message Count
08	0C	Return Bus Communication Error Count
08	0D	Return Bus Exception Error Count
08	0E	Return Slave Message Count
08	0F	Return Slave No Response Count
08	10	Return Slave NAK Count
08	11	Return Slave Busy Count
08	12	Return Bus Char. Overrun Count
08	13	Return Overrun Error Count
08	14	Clear Overrun Counter and Flag
08	15	Get/Clear Modbus Plus Statistics
08	16-UP	Reserved

SCADA Protocols – DNP3

- Distributed Network Protocol 3
- Mainly used in the utility companies




```
0000   00 02 b3 ce 70 51 00 50   04 93 70 67 08 00 45 00    ....pQ.P ..pg..E.
0010   00 40 4c 82 40 00 80 06   9a 2b 0a 00 00 08 0a 00    .@L.@... .+......
0020   00 03 0a e5 4e 20 55 1c   1a fb 52 c6 38 a8 50 18    ....N U. ..R.8.P.
0030   ff dd 07 f3 00 00 05 64   11 c4 04 00 03 00 4e ef    .......d ......N.
0040   c2 c2 15 3c 02 06 3c 03   06 3c 04 06 b1 ec          ...<..<. .<....
```

Disable Spontaneous messages

SCADA Protocols – DNP3

Bit	Internal Indication Flag
0	Last received message was Broadcast message
1	Class 1 Data available
2	Class 2 Data available
3	Class 3 Data available
4	Time Synchronization Required
5	Digital Output in Local
6	Device Trouble
7	Device Restarted
8	Function Code (Not Implemented)
9	Requested Object Unknown or Application Error
10	Parameters Out of range
11	Even buffer overflowed
12	Operation already executing
13	Configuration Corrupt
14	Not used (returns 0)
15	Not used (returns 0)

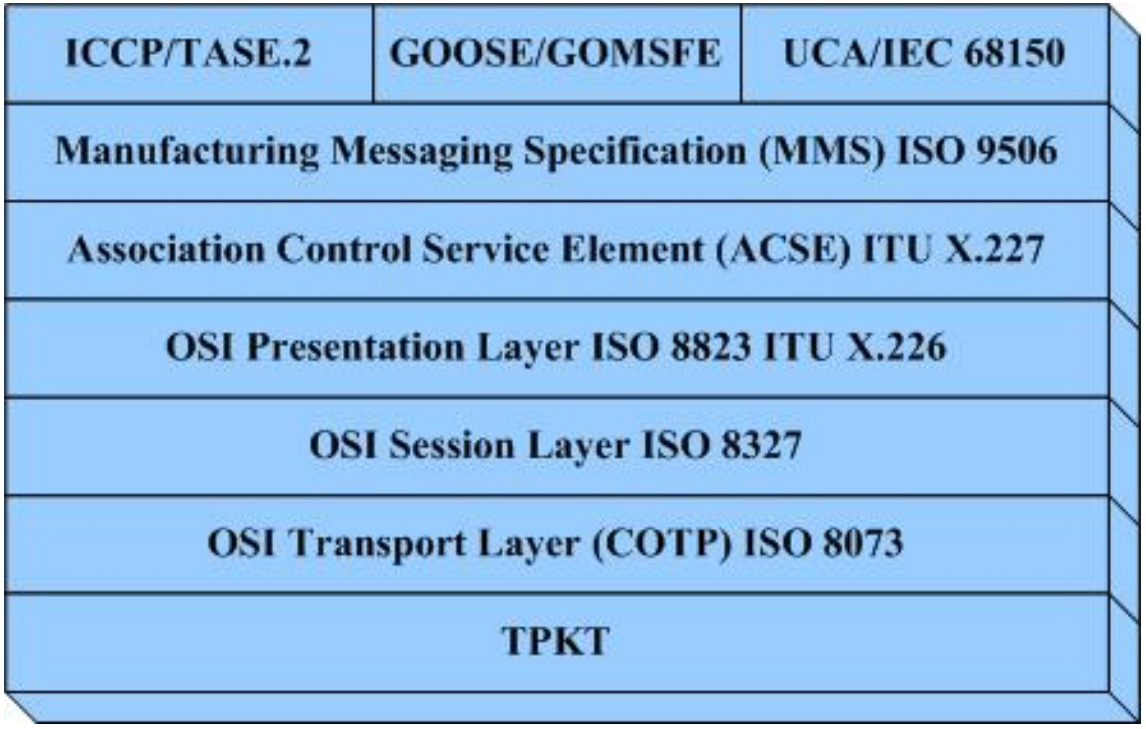
SCADA Protocols – DNP3

- Control Byte
  - Control function code
- Transport Layer byte
  - First-Final
  - Sequence Number
- Application Layer Control Byte
  - First-Final
  - Confirm
  - Sequence
- Data chunking
  - CRC DNP
  - 2 CRC bytes Every 16 bytes of data

SCADA Protocols – TASE2.0/ICCP

- Inter-Control Center Communications Protocol
- Telecontrol Application Service Element (TASE)
- Developed by Electric Power Research Institute(EPRI) and Northern States Power (NSP)
- Is used for communication between the Control centers and the WANs
- Used widely in the Utility organizations
- Associations – Connections
- Bilateral Table (BLT) – Stores the Associations and end point agreements
- Uses Application Control Service Element (ACSE) to Associate and mange connections
- Multiple Associations can be managed by ACSE

SCADA Protocols – TASE2.0/ICCP



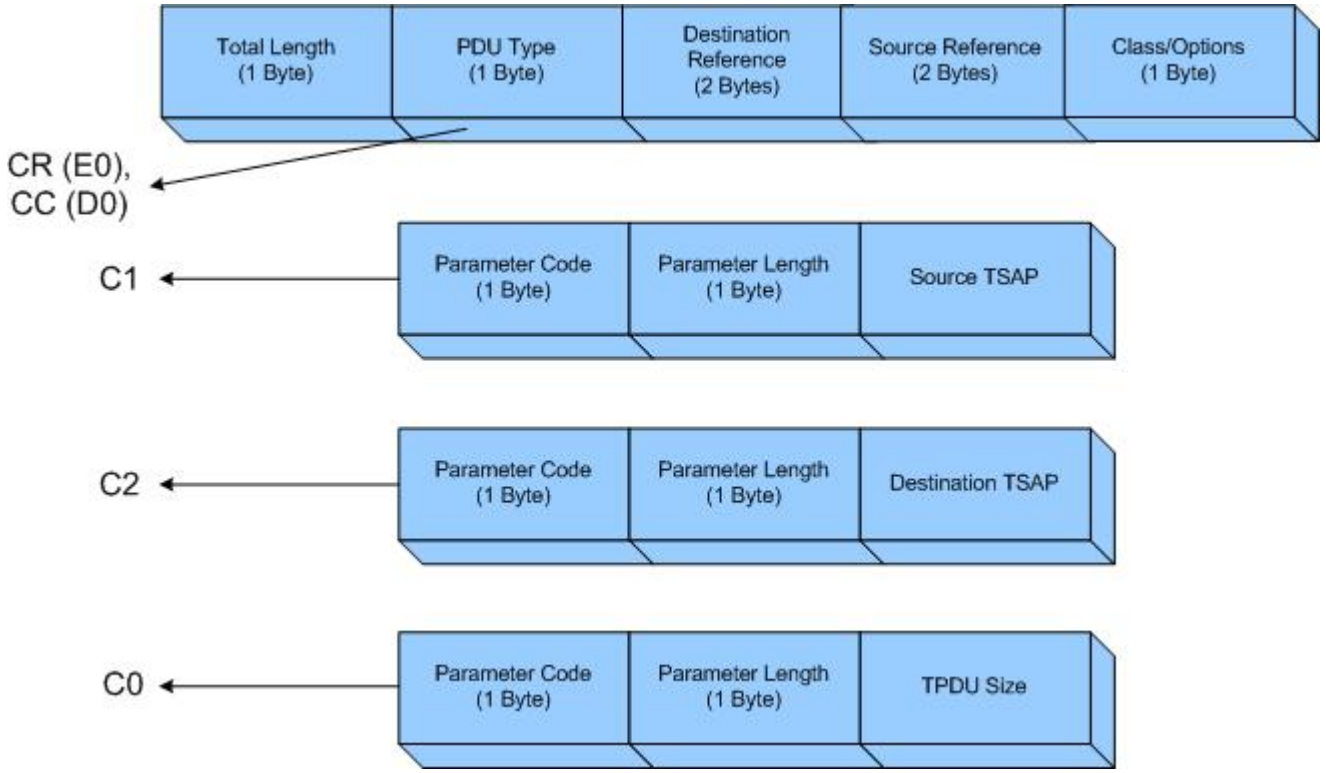
SCADA Protocols – ICCP – TPKT

- TPKT
  - Most software that I have seen have the version set to 03
  - The Reserved byte is 00
  - Finally the Length varies based on the other layers Information


```
0000  00 0c 29 d8 e9 f2 00 c0  4f 0c 7b 1d 08 00 45 00   ..)..... O.{...E.
0010  00 6c a2 9c 40 00 80 06  2f 11 0a 0a 0a 65 0a 0a   .l..@... /....e..
0020  0a 66 00 66 0c 03 04 ef  16 ed 4d 7c 55 b6 50 18   .f.f.... ..M|U.P.
0030  3f 6c c8 a1 00 00 03 00  00 44 02 f0 80 01 00 01   ?l.....  .D......
0040  00 61 37 30 35 02 01 01  a0 30 a3 2e a0 2c a1 1f   .a705... .0...,..
0050  a1 1d 1a 0e 41 41 41 41  41 41 41 41 41 41 41 41   ....AAAA AAAAAAAA
0060  41 41 1a 0b 41 41 41 41  41 41 41 41 41 41 41 a0   AA..AAAA AAAAAAA.
0070  09 80 01 02 80 01 02 80  01 02                     ........ ..
```

SCADA Protocols – ICCP – COTP

- Connection-Oriented Transport Protocol (COTP) – ISO 8073




```
0000   00 c0 4f 0c 7b 1d 00 0c   29 d8 e9 f2 08 00 45 00   ..O.{... )..... E.
0010   00 52 e5 de 40 00 80 06   eb e8 0a 0a 0a 66 0a 0a   .R..@... .....f..
0020   0a 65 0c 03 00 66 4d 7c   50 b2 04 ef 11 0a 50 18   .e...fM| P.....P.
0030   44 70 60 57 00 00 03 00   00 2a 25 e0 00 00 00 01   Dp`W.... .*%.....
0040   00 c1 0c 41 41 41 41 41   41 41 41 41 41 41 41 c2   ...AAAAA AAAAAAA.
0050   0c 42 42 42 42 42 42 42   42 42 42 42 42 42 c0 01   .BBBBBBB BBBBB...
```

SCADA Protocols – ICCP – COTP

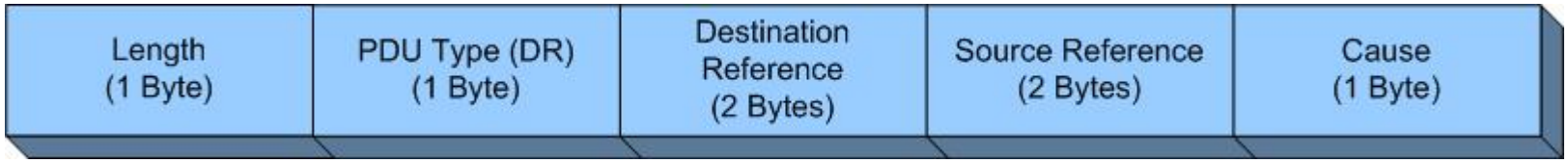
- Connection-Oriented Transport Protocol (COTP) – ISO 8073

Data Transfer F0

```
0000  00 0c 29 d8 e9 f2 00 c0  4f 0c 7b 1d 08 00 45 00   ..)..... O.{...E.
0010  00 6c a2 9c 40 00 80 06  2f 11 0a 0a 0a 65 0a 0a   .l..@... /....e..
0020  0a 66 00 66 0c 03 04 ef  16 ed 4d 7c 55 b6 50 18   .f.f.... ..M|U.P.
0030  3f 6c c8 a1 00 00 03 00  00 44 02 f0 80 01 00 01   ?l...... .D......
0040  00 61 37 30 35 02 01 01  a0 30 a3 2e a0 2c a1 1f   .a705... .0...,..
0050  a1 1d 1a 0e 41 41 41 41  41 41 41 41 41 41 41 41   ....AAAA AAAAAAAA
0060  41 41 1a 0b 41 41 41 41  41 41 41 41 41 41 41 a0   AA..AAAA AAAAAAA.
0070  09 80 01 02 80 01 02 80  01 02                     ........ ..
```

SCADA Protocols – ICCP – COTP

- Connection-Oriented Transport Protocol (COTP) – ISO 8073



Disconnect Request 80

```
0000  00 c0 4f 0c 7b 1d 00 0c  29 d8 e9 f2 08 00 45 00   ..O.{... )....E.
0010  00 33 e6 72 40 00 80 06  eb 73 0a 0a 0a 66 0a 0a   .3.r@... .s...f..
0020  0a 65 0c 03 00 66 4d 7c  55 b6 04 ef 17 31 50 18   .e...fM| U....1P.
0030  44 2c ed 6d 00 00 03 00  00 0b 06 80 00 01 00 01   D,.m.... ........
0040  81                                                .
```

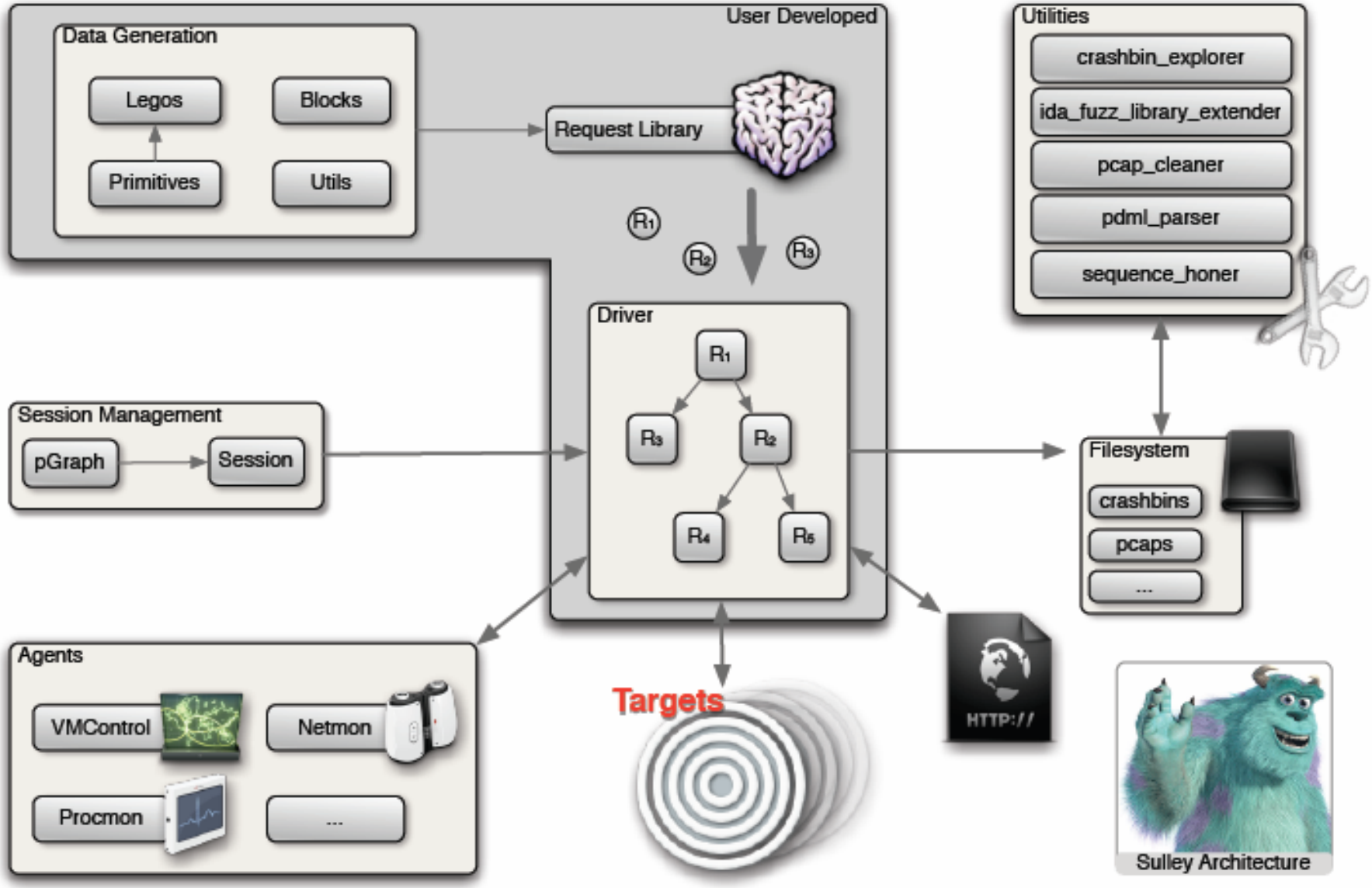
SCADA Fuzzer

- What does the SCADA Fuzzer detect?
  - Protocol anomalies
  - Unauthorized client/server communication
  - Unauthorized client/server command execution
  - Possible Denial of Service attacks
- What protocols are we covering today?
  - MODBUS
  - DNP3
  - ICCP
    - TPKT
    - COTP

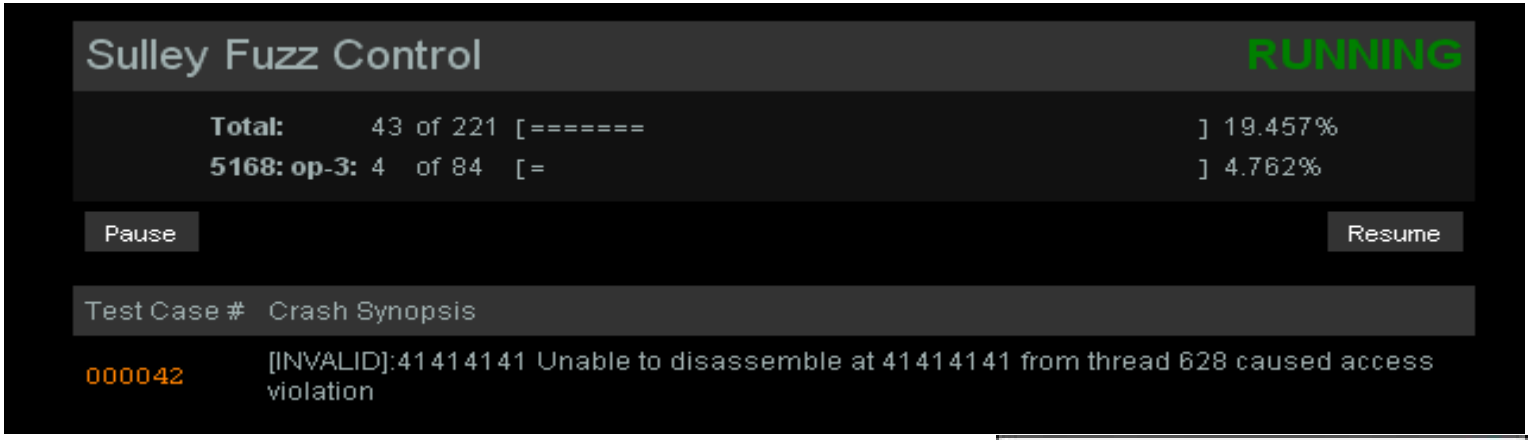
SCADA Fuzzer

- Fuzzer Components
  - __init.py – Defines all the aliases
  - blocks.py – Defines blocks and block helpers
  - pedrpc.py – Communication purposes and an interface with the main fuzzer
  - primitives.py – the fuzzer primitives includes string, static, etc
  - sessions.py – Functionality for building and executing session
  - sex.py – Sulley's exception Handler
- Agents
  - network_monitor.py – Monitors network communications and logs the pcap files
  - process_monitor.py – Detects the faults
  - vmcontrol.py – Interfaced with the VM image to start, stop, suspend and reset the image along with deleting and restoring the snapshots

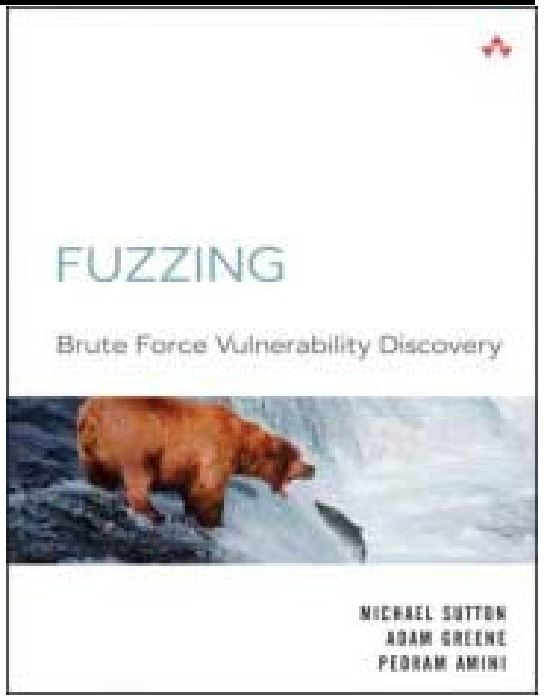
SCADA Fuzzer Architecture



Sulley Fuzzer

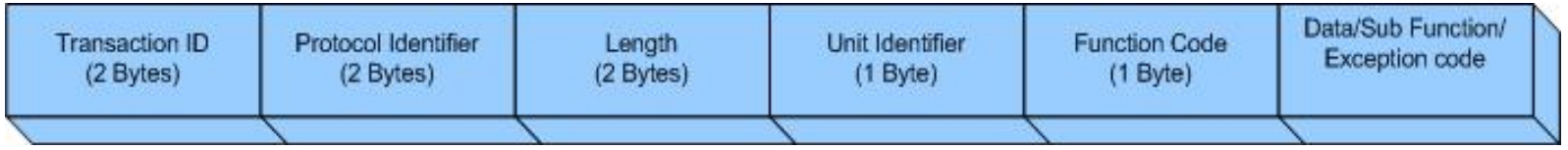


- Web GUI of Sulley
- www.fuzzing.org
- Fuzzing book



Modbus Code Snippet

```
s_initialize("MODBUSFUNCCODE01")
  # Transaction ID
  s_static("\x00\x01")
  # Modbus Protocol Identifier
  s_static("\x00\x00")
  # Length bytes
  s_sizer("modlength", length=2, name="length", endian=">", fuzzable=False)
  if s_block_start("modlength"):
      # Unity Identifier
      s_static("\x0D")
      # Function Code
      s_byte(0x01)
      # Data or Sub function Code
      s_dword(0x00000000)
  s_block_end()
```

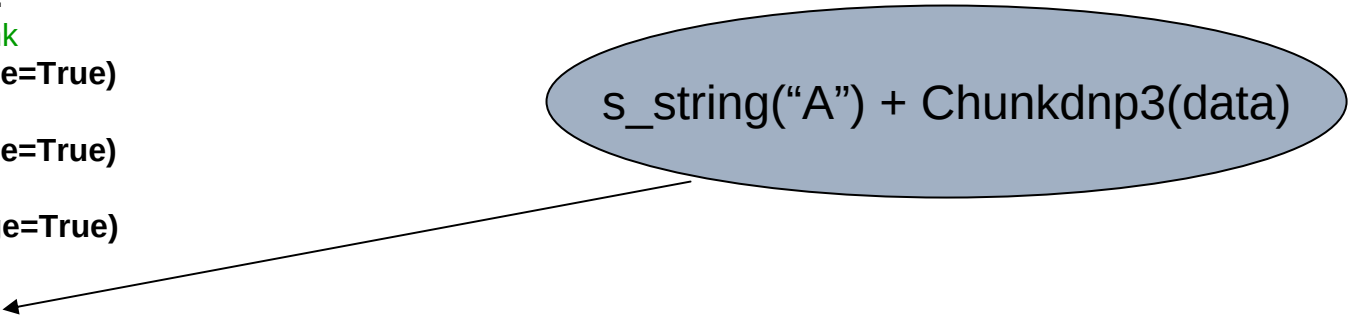


DNP3 Code Snippet

Static Length

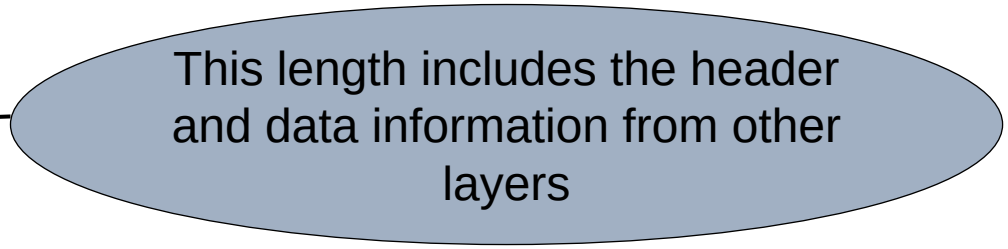
```
s_initialize("DNP3StaticLength")
if s_block_start("header"):
   s_static("\x05\x64") # Start Sync Bytes.
   # Length Bytes we are having it as a constant length at first
   s_static("\x12")
   # Control Byte
   s_byte(0xc4, full_range=True)
   # Destination Address
   s_short(0x0400)
   # Source Address
   s_short(0x300)
   s_block_end()
# Checksum of the DNP Header.
s_checksum("header", algorithm=dnp_crc16, length=2)

# The Data POrtion of the Packet
if s_block_start("Data"):
   # Transport Layer Chunk
   s_byte(0xc2, full_range=True)
   # Application Chunk
   s_byte(0xc2, full_range=True)
   # Function Code
   s_byte(0x0d, full_range=True)
   # Static Data for now..
   s_static("AAAAAAA")
   # This will fuzz a huge array of string cases..
   s_block_end()
s_checksum("Data", algorithm=dnp_crc16, length=2)
```



ICCP – TPKT Code Snippet

```
s_initialize("ICCP-TPKT")
if s_block_start("header"):
  s_byte(0x03, full_range=True)
    # Version
  s_byte(0x00, full_range=True)
    # Reserved
  s_short(0x0000)
    # Length
  s_block_end()
```



ICCP – COTP Code Snippet

```
s_initialize("ICCP-COTP")
s_sizer("header", length=1, name="length", fuzzable=True)
# Length
if s_block_start("header"):
    s_byte(0xE0, full_range=True)
                # PDU Type
    s_short(0x0000)
                # Destination Reference
    s_short(0x0000)
                # Source Reference
    s_byte(0x00, full_range=True)
                # Class/Options
    s_byte(0xc1)
       # Parameter Code Source TSAP
    s_sizer("Param1",length=1, name="ParamLength1", fuzzable=True)
    if s_block_start("Param1"):
       s_string("A")
       s_block_end()

    s_byte(0xc2)
       # Parameter Code Destination TSAP
    s_sizer("Param2",length=1, name="ParamLength2", fuzzable=True)
    if s_block_start("Param2"):
       s_string("A")
       s_block_end()

    s_byte(0xc0)
       # Parameter Code TPDU Size
    s_sizer("Param3",length=1, name="ParamLength3", fuzzable=True)
    if s_block_start("Param3"):
       s_byte(0x0b)
       s_block_end()
    s_block_end()
```

Source TSAP

Destination TSAP

TPDU Size

TippingPoint
a division of 3Com

Demo

Disclosure

- Please be responsible and use responsible disclosure methods when you do find some vulnerabilities

TippingPoint
a division of 3Com

Conclusion

- Basic SCADA network architecture
- Need for security in SCADA network
- SCADA protocol details
- Fuzzer details

TippingPoint
a division of 3Com

Future work

- Other SCADA Protocols.
- Two way fuzzing

References

- The SCADA Architecture and basic implementation details: ***Securing SCADA Systems – Ronald L. Krutz. PhD***
- Modbus***:*** ***www.modbus.org***
- DNP3***:*** ***www.dnp3.org***
- ICCP***:*** ***www.iccp.org***
- Attack Details**:** ***www.digitalbond.com***
- Modbus Protocol details***:*** ***http://www.modbustools.com/PI_MBUS_300.pdf***
- DNP3 Protocol Primer***:*** ***http://www.dnp.org/About/DNP3%20Primer%20Rev%20A.pdf***
- DNP3 User and Reference Manual by Control Microsystems***:*** ***https://dg.controlmicrosystems.com/Technical%20Support/Software,%20Manuals%20and%20Release%20Notes/Protocols/DNP3%20Protocol/Manuals/DNP3_User_and_Reference_Manual.pdf***
- ICCP Guide***:*** ***www.sisconet.com/downloads/usrguid5.doc***
- Matt Franz Wiki***:*** ***http://www.scadasec.net/secwiki/SecProducts***
- *Wikipedia*
- *SCADA Architecture slides - ??*

TippingPoint
a division of 3Com

Acknowledgements

- **Pedram Amini**, **Aaron Portnoy** and **Cody Pierce** for working on the Sulley Fuzzing Framework
- **Rohit Dhamankar** and **Dinesh Sequeira** for getting me psyched about SCADA systems
- **Matt Franz** for the support and guidance

TippingPoint
a division of 3Com

Questions?

Thank you

Ganesh Devarajan

gdevarajan@tippingpoint.com