60870-5-104 protocol snort rule customization

GIAC (GCIH) Gold Certification

Author: Adrian Aron, adaron@cisco.com
Advisor: Mohammed Haron
Accepted: 2020-08-04

Abstract

OT Security emerges as a necessity due to its flat network implementation and criticality of systems operated over the network. Supervisory Control And Data Acquisition (SCADA) 60870-5-104 is widely used in Europe by most Utility operators, making it a target for attackers. While IDS signatures for SCADA IEC104 have been developed, most of its signatures are generic and bind to the standard protocol itself, not to the specific implementation of each customer. For example, an interrogation command telegram in a customer environment might be harmless, while others might be critical information. This paper explains the underlying construct of an IEC104 telegram and how to customize standard snort rules for that specific telegram. In this way, each SCADA command can be interpreted, evaluated for permit/monitor/deny to any controlled device, for each particular SCADA implementation.

1. Introduction

Supervisory Control And Data Acquisition (SCADA) IEC 60870-5-104, renamed for convenience within this document as IEC104, provides the network component to IEC 60870-5-101 (for convenience IEC 101). "In simple terms, it delivers IEC 101 messages as application data (OSI L7) over TCP on port 2404. IEC 104 enables the communication between the control station and a substation via a standard TCP/IP network" (Petr Matousec, 2017, p.2). The communication of the SCADA application is following the client-server model in most of implementations (Thomas Teodorowicz, 2017, p.3)

The simplified diagram presented in Figure 1 intends to explain how and where the detection is proposed and other security controls that can protect the Industrial implementation; the diagram is following the Purdue Model for segmentation (ISA99 Committee, 2004):

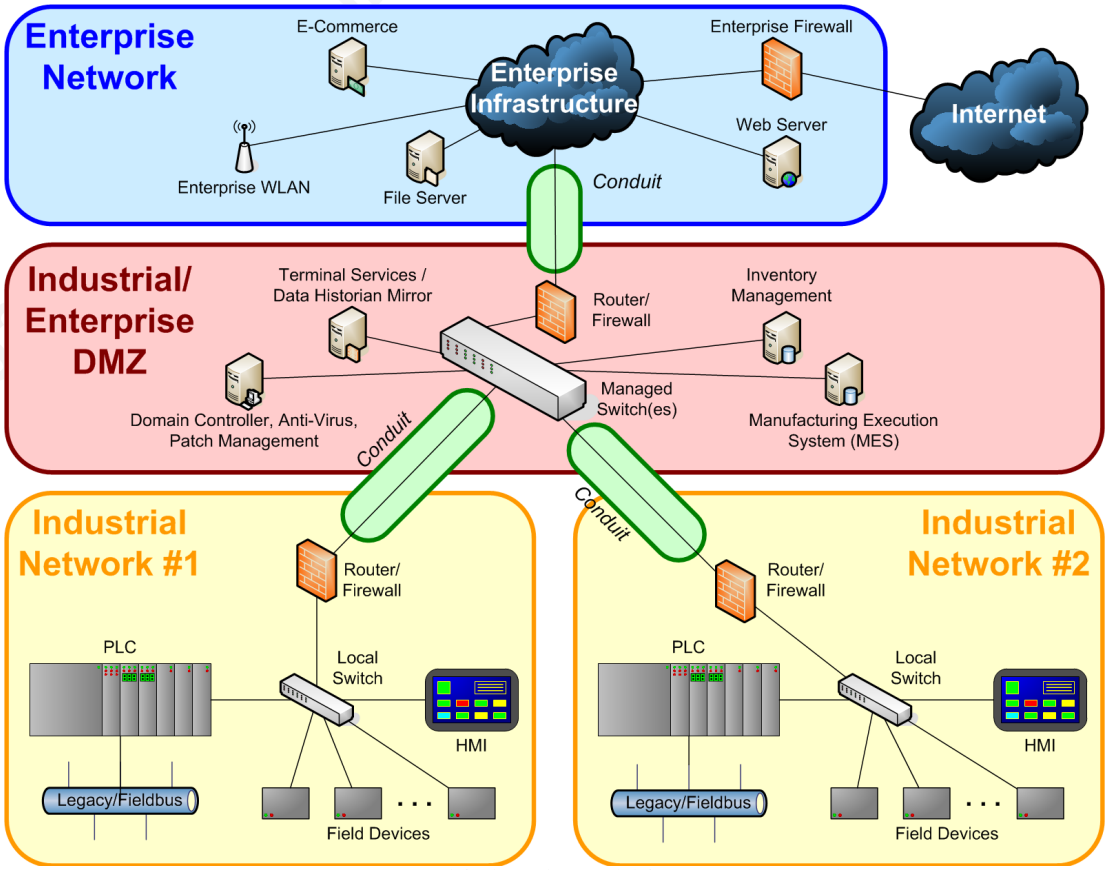


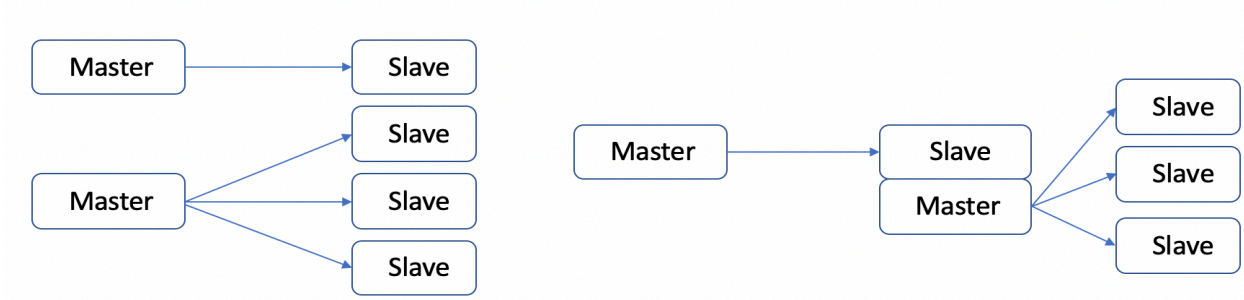
Figure 1 - Modified Purdue model for Control Hierarchy

Adrian Aron adaron@cisco.com

Each transit from one level to another is recommended to pass through a Router/Firewall as a segmentation control, following the guidance of isolated functional groups. (NIST SP800-82 Rev.2, p.5-6)

The WAN connections between Industrial Networks or Industrial Fields and Industrial Enterprise DMZ are considered to be encrypted and tunneled, so the transport layer between sites is secured. (NIST SP800-82 Rev.2, p.6-37)

Each Firewall in each Field Industrial Network can inspect and alert based on incoming messages for interrogations or commands from Enterprise DMZ, issued to its local Field devices. Each Firewall becomes a segmentation control point and, at the same time, as a protocol sensor, specific to its local Industrial function.
In typical IEC 104 architecture, there is a Master and a Slave, a Master and numerous Slaves, a Master, then a Master/Slave node, and numerous Slaves.



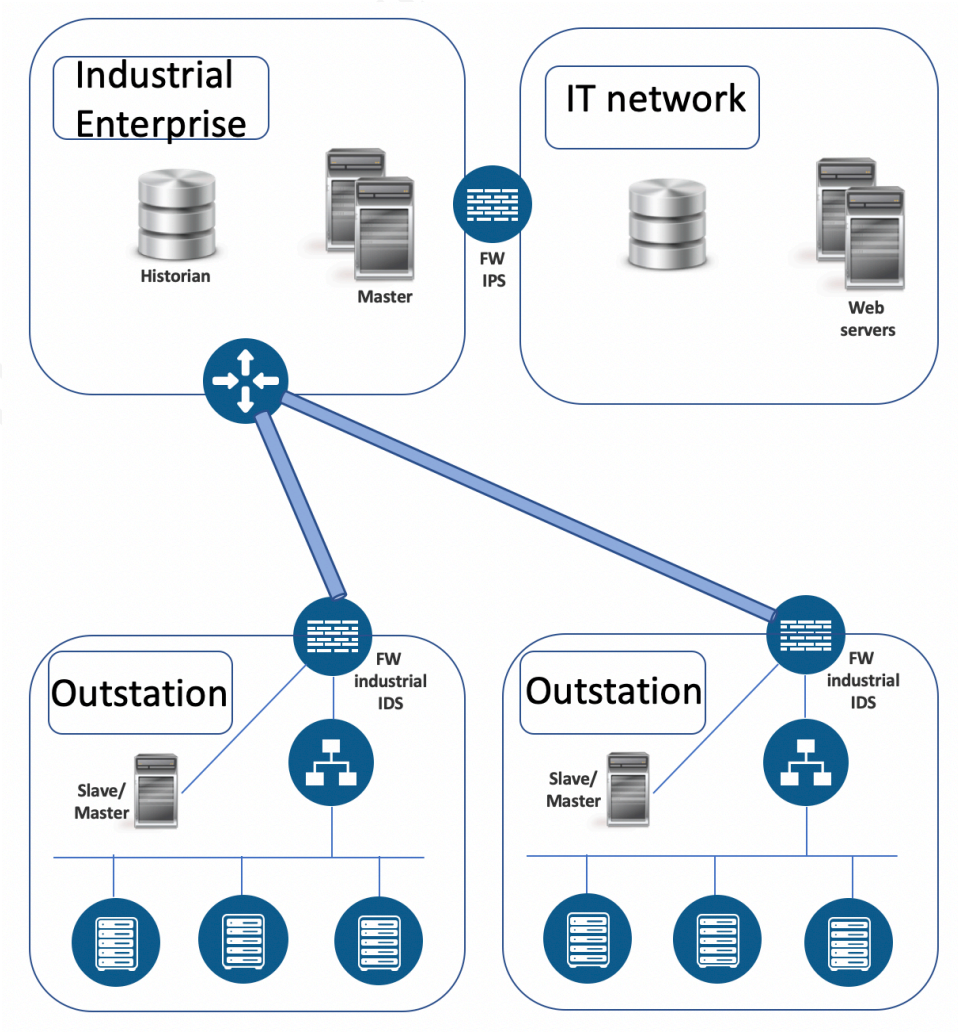
IEC 104 TCP port is standardized to 2404 but is not mandatory in practical implementations. In real scenarios could be observed IEC 104 servers running on TCP 2401, 2407, and adjacent ports. It becomes challenging to filter specific TCP ports on the WAN interfaces; it is not clear or documented what ports are used for Industrial implementation.

A Master/Slave node is typically located in Outstation (also known as a controlled station or remote station), relaying interrogations and commands to local controlled devices. This setup is used to either translate between multiple protocols within the outstation or to have local control in case the Wide Area Network (WAN) is inaccessible and local override of the outstation is needed. A local override by a user would be rare since the process is controlled at the Industrial Enterprise level. However, if it does happen, then local

temporary Master node issues the interrogation commands locally, similar to interrogation commands issued by Industrial Enterprise Master over WAN, whenever WAN is operational.

2. Open-source Snort signatures details

The following schema represents inspection points possible for SCADA's traffic. This document is focused on the Industrial Firewall function and its Intrusion Detection System (IDS) function located in the Outstation (remote Station) just before the Intelligent Electronic Devices (IED) or Programmable Logic Controllers (PLC) within that Outstation.



Adrian Aron adaron@cisco.com

Snort is an open source network intrusion prevention system, capable of performing real-time traffic analysis and packet logging on IP networks. Snort signatures for IEC 104 were developed and published on December 2016 and available for download on www.snort.org; IEC104 transmission over TCP is composed of packets following a certain data structure, named Application Protocol Data Unit (APDU) and Application Service Data Unit (ASDU) included in each APDU;

Snort rules follow the IEC 104 protocol description and alert or notify based on ASDU Type and its associated description;

For example, here are the bytes of a sample IEC 104 telegram, for convenience, stripped down to the bytes to IEC 104 APDU only :

```
68 15 08 00 02 00 1E 01 03 00 78 00 26 02 00 01 31 9C 2B 07 10 01 14,
representing IEC 60870-5-104-Asdu: ASDU=120 M_SP_TB_1 Spont IOA=550
'single-point information with time tag CP56Time2a'
```

When the telegram is inspected by default snort IDS, the snort rule matching it, is :

```
alert tcp any [1024:] <> any 2404 (msg:"PROTOCOL-SCADA IEC 104
M_SP_TB_1"; flow:established; content:"|68|"; depth:1;
content:"|1E|"; within:1; distance:5;
reference:url,blog.snort.org/2016/12/iec60870-5-104-protocol-
detection-rules.html; classtype:protocol-command-decode; sid:52160;
rev:1; gid:1;)
```

The bytes inspected by the snort rule are marked bold, APDU start byte x**68** and Type identification byte x**1E**; This is according to protocol a valid match, but the telegram has more data that can be useful but is very specific to an implementation and to a customer.

From the above telegram, examples of further information are :

- Common ASDU address (2Bytes) = **78 00** translating to ASDU address value, 120;

68 15 08 00 02 00 1E 01 03 00 **78 00** 26 02 00 01 31 9C 2B 07 10 01 14

- Information Object Address (IOA, 3bytes) = **26 02 00** translating to IOA value, 550;

68 15 08 00 02 00 1E 01 03 00 78 00 **26 02 00** 01 31 9C 2B 07 10 01 14

Adrian Aron adaron@cisco.com

With these new details extracted from the telegram, we can get an insight into **where** and to **what** device was the target of the initial "single-point information" interrogation.

Common ASDU address is typically associated with an Outstation, identifying the location address where the telegram has to reach. Information Object Address (IOA) is an Element identifier within that location, interrogating exactly a specific value or commanding a particular parameter to be changed within that location.

Here is where snort can help in limit dangerous commands to be sent or redundant or unnecessary interrogation reaching an Outstation. Of course, this means entering the realm of OT SCADA engineers, who have to support, confirm, and acknowledge the information detected from IEC 104 telegrams. Without the support of OT engineers, it is not recommended to customize snort rules, due to false positives and lack of more information on what IOA Elements are representing.

An IDS can sniff the common ASDU address and customize all rules to match just certain deployed ASDU values, then alert on unknown ASDU values. However, this remains generic customization despite the effort to deploy;
Below in Figure 2 are two diagrams presenting the structure of the IEC 104 Type I frame and the details of transported Elements. IEC 104 permits the transport of multiple IOAs of the same type on a single ASDU or the interrogation of various values or the same IOA in a single ASDU.

Adrian Aron adaron@cisco.com

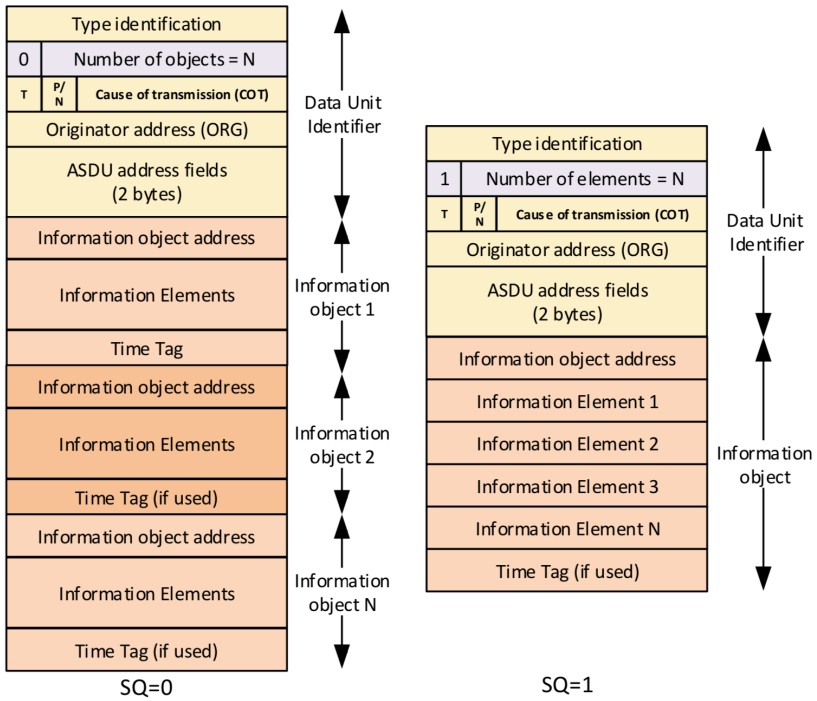


Figure 2 - ASDU detailed view (Petr Matousec, 2017, p.15)

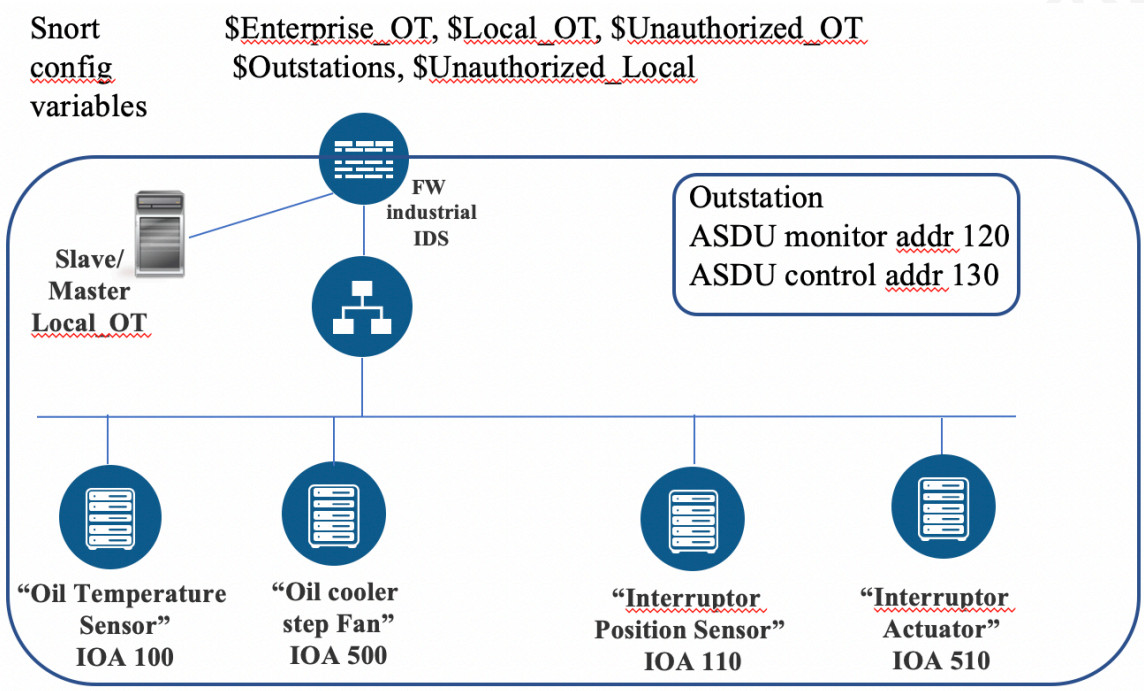
Details our example telegram in bytes structure, are in the figure below:

68 15 08 00 02 00 1E 01 03 00 78 00 26 02 00 01 31 9C 2B 07 10 01 14

Type identification	1E
0 Number of Objects	01
T P/N Cause of Transmision	03
Originator Adddress	00
ASDU Address	78 00
Info Object Address 1	26 02 00
IO1 Elements	01
IO1 Time Tag (if used)	31 9C 2B 07 10 01 14

A hypothetical Outstation diagram is presented below, for the following examples:

Adrian Aron adaron@cisco.com



It is a convenience to consider that SCADA servers in the Enterprise_OT are controlling the installation. So, as long as the WAN connectivity is operational, these servers are authorized to query and issue commands to Outstations;

The authorized servers are part of $Enterprise_OT snort variable, similarly were defined the variables for Local servers located within the Outstation, as variable $Local_OT, then for Outstations variable $Outstations; Snort accepts a variable that negates an object, so another variable is defined as $Unauthorized_OT, matching any IP other than $Enterprise_OT, similarly, $Unauthorised_Local matches any IP other than $Local_OT;

In this example, an alternative TCP port 2407 is used for IEC 104 communications; Also, Outstations use only ASDU addresses from 120 to 140; In OT networks, ASDU addresses are also known as Sectors.

It is also considered that OT engineering has translated the IOA Elements into actual roles performed within Outstation, IOA100 as "Oil temperature sensor," IOA500 is an "Oil cooler fan," IOA110 "Interrupter Position sensor" and IOA510 "Interrupter actuator." Without this information, it is tough to translate IOA values into meaning for the IDS

system.

2.1. Snort rule alarming on unknown or unused ASDU address

This example starts from a standard IEC 104 snort rule with SID 1:41073.

```
alert tcp any [1024:] <> any 2404 (msg:"PROTOCOL-SCADA IEC 104
bitstring of 32 bits"; flow:established; content:"|68|";
pcre:"/\x68.{5}[\x33\x40\x07\x21]/";
reference:url,blog.snort.org/2016/12/iec60870-5-104-protocol-
detection-rules.html; classtype:protocol-command-decode; sid:41073;
rev:4; gid:1; )
```

Strict requirements need to be followed when duplicating snort rules, the first one is to change the snort ID number to something unique (SID), so the changes made from the standard rule are:

- **SID** value of 1000080**;**
- change the Server known IP addresses with $**Enterprise_OT**
- known Outstations IP addresses to $**Outstations**.
- change the TCP Server port from standard 2404 to 2401-2409 range
- regex syntax searching for anything not in interval 120-140, in bytes reserved for ASDU address field in telegram;
- **msg** field can also be customized to be more specific, relevant to alert tools and SCADA implementation. In this example the message is "unknown ASDU address detected - in TCP range" since any ASDU address value used outside of 120-140 range will be considered not part of implementation;

The customized rule output is:

```
alert tcp $Enterprise_OT [1024:] <> $Outstations [2401:2409]
(msg:"PROTOCOL-SCADA IEC 104 unused ASDU address detected - in TCP
range"; flow:established; content:"|68|"; pcre:"/\x68.{5}[^\x{76}-
\x{8C}]/"; reference:url,blog.snort.org/2016/12/iec60870-5-104-
protocol-detection-rules.html; classtype:protocol-command-decode;
sid: 1000080; rev:8; gid:1; )
```

Adrian Aron adaron@cisco.com

This rule can raise False Positives (FP) during the implementation of a new SCADA process that can use the same Enterprise Servers, same TCP ports, same IEC 104 protocol, but different ASDU addresses. Or, Enterprise Servers are reaching out to newly installed Outstation with assigned ASDU address exceeding 120-140 range, requiring adjustments to snort rule PCRE field, to match the new range values.

2.2. Snort rule alarming on broadcast interrogations from the unknown servers

Broadcasts interrogations are valid within OT but expose a lot of data from Outstations when used. By using broadcasts, or general interrogations, a reconnaissance can be performed.

General interrogation command is matched on standard snort rule with SID: 52191.

```
alert tcp any [1024:] <> any 2404 (msg:"PROTOCOL-SCADA IEC 104 C_IC_NA_1"; flow:established; content:"|68|"; depth:1; content:"|64|"; within:1; distance:5; reference:url,blog.snort.org/2016/12/iec60870-5-104-protocol-detection-rules.html; classtype:protocol-command-decode; sid:52191; rev:1; gid:1; )
```

For the example presented the standard rule will not match because server TCP port is changed, also to identify unknown servers, authorized servers need to be excluded from matching, while General interrogations is frequently used;
Customizing the rule requires using the snort variable, named $**Unauthorized_OT**, where known servers are excluded;

alert tcp $**Unauthorized_OT** [1024:] <> $**Outstations** [**2401:2409**] (msg:"IEC 104 C_IC_NA_1 General Interrogation from non-authorized server, posible OT reconnaissance"; flow:established; content:"|68|"; depth:1; content:"|64|"; within:1; distance:5; reference:url,blog.snort.org/2016/12/iec60870-5-104-protocol-detection-rules.html; classtype:protocol-command-decode; sid: 1000081; rev:2; gid:1;)
Another method of interrogation is the use of ASDU broadcast address, 65535 (FFFF in hex). In this case the custom rule is:

```
alert tcp $Unauthorized_OT [1024:] <> $Outstations [2401:2409] (msg:"IEC 104 Broadcast ASDU from non-authorized server, posible OT reconnaissance"; flow:established; content:"|68|"; content:"|FFFF|"; distance:9; within:11; reference:url,blog.snort.org/2016/12/iec60870-5-104-protocol-detection-rules.html; classtype:protocol-command-decode; sid: 1000082; rev:2; gid:1; )
```

2.3. Snort rule alarming on queries made from unknown server within Outstation

This is a rule that alerts on connections performed by Servers, that are present locally in Outstation and are not explicitly defined as authorized in the **$Local_OT** variable. It can expose routine maintenance but with local variances of software running on engineer's laptops, or can expose unauthorized local access in OT networks and interrogations that might be reconnaissance data.

For this rule to match, it required that within the Outstation LAN, there is still segmentation between local server access and devices. In cases when the application server is directly connected to the device itself or an unmanaged switch, then the IDS will not detect it.

Let's start from standard rule with SID 41047:

```
alert tcp any [1024:] <> any 2404 (msg:"PROTOCOL-SCADA IEC 104 STARTDT ACT"; flow:established; content:"|68|"; depth:1; content:"|07|"; within:1; distance:1; reference:url,blog.snort.org/2016/12/iec60870-5-104-protocol-detection-rules.html; classtype:protocol-command-decode; sid:41047; rev:4; gid:1; )
```

```
alert tcp $Unauthorized_Local [1024:] <> $Outstations [2401:2409] (msg:"PROTOCOL-SCADA IEC 104 STARTDT ACT detected locally from unauthorized server"; flow:established; content:"|68|"; depth:1; content:"|07|"; within:1; distance:1; reference:url,blog.snort.org/2016/12/iec60870-5-104-protocol-detection-rules.html; classtype:protocol-command-decode; sid:1000083; rev:1; gid:1; )
```

$Unauthorized_Local and $Unauthorized_OT subnets should not overlap, in order for this approach to match correctly. Otherwise, a single $Unauthorized snort variable can be created and maintained to avoid false positives. Having a distinct alert on Local versus Remote might shorten the time on detecting in what location the alert triggered and by whom.

2.4. Snort rule alarming on high-risk commands made to Outstation

As an example, an alert is raised when a command is sent to a high-risk, high-value instrument, even the legitimate command is required. This alert has the role of network confirmation and audit that the command has been detected, confirming operations made by the engineers or by the automation itself.

Standard rule for regulating step pozition is SID: 52173;

```
alert tcp any [1024:] <> any 2404 (msg:"PROTOCOL-SCADA IEC 104
C_RC_NA_1"; flow:established; content:"|68|"; depth:1;
content:"|2F|"; within:1; distance:5;
reference:url,blog.snort.org/2016/12/iec60870-5-104-protocol-
detection-rules.html; classtype:protocol-command-decode; sid:52173;
rev:1; gid:1; )
```

An alert should be raised when a command is sent to the interrupter actuator in Outstation from Enterprise_OT; The object address for the interrupter actuator is 510 (HEX **FE 01** in reverse order in telegram); For this example the commands sent to Outstation have ASDU address 130 (HEX **82**), not 120;

Matching telegram is : **68** 0E 12 00 76 01 **2F** 01 06 04 **82 00 FE 01 00** 02

```
alert tcp $Enterprise_OT [1024:] <> $Outstations [2401:2409] (msg:"
IEC 104 C_RC_NA_1 Actuator position change"; flow:established;
content:"|68|"; depth:1; content:"|2F|"; within:1; distance:5;
content:"|8200FE01|"; within:4; distance:3;
reference:url,blog.snort.org/2016/12/iec60870-5-104-protocol-
detection-rules.html; classtype:protocol-command-decode; sid:1000084;
rev:1; gid:1; )
```

Adrian Aron adaron@cisco.com

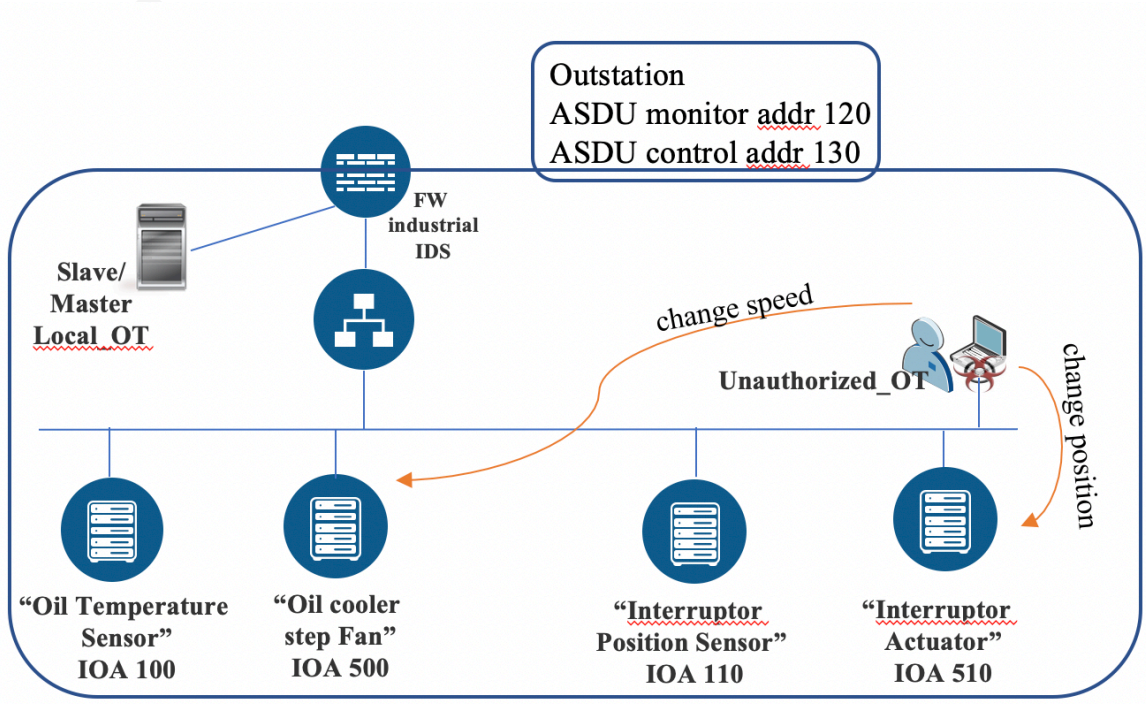
2.5. Snort rule alarming on specific dangerous commands sent from an unknown server within the Outstation

Continuing the example above, consider the situation described in the diagram below, where a malicious user has access to local Outstation and wants to change the speed of the oil cooler fan, shut it off, switch off the interrupter.

Despite the conventional idea of a network not interfering in OT communications, for this particular scenario blocking the command to the interrupter might be beneficial considering the user is not $Local_OT and the device is high-risk high-value.

As for the oil cooler fan malicious command, the alert would be enough to trigger, since the oil will not be heated faster than the operator correcting the speed of the fan.

For this type of event, rule 1000083 defined above at point 3.1.3 will also match;



Adrian Aron adaron@cisco.com

For the Interruptor actuator command alert, the rule should be :

```
alert tcp $Unauthorized_Local [1024:] <> $Outstations [2401:2409]
(msg:" IEC 104 C_RC_NA_1 Actuator position change from unauthorized
server"; flow:established; content:"|68|"; depth:1; content:"|2F|";
within:1; distance:5; content:"|8200FE01|"; within:4; distance:3;
reference:url,blog.snort.org/2016/12/iec60870-5-104-protocol-
detection-rules.html; classtype:protocol-command-decode; sid:1000084;
rev:1; gid:1; )
```

Similar for the Oil cooler fan command alert, the rule should be:

```
alert tcp $Unauthorized_Local [1024:] <> $Outstations [2401:2409]
(msg:" IEC 104 C_RC_NA_1 Oil cooler Fan speed change from unauthorized
server"; flow:established; content:"|68|"; depth:1; content:"|2F|";
within:1; distance:5; content:"|8200F401|"; within:4; distance:3;
reference:url,blog.snort.org/2016/12/iec60870-5-104-protocol-
detection-rules.html; classtype:protocol-command-decode; sid:1000084;
rev:1; gid:1; )
```

The only difference between the two proposed rules are the IOA bytes, addressing different devices within the Outstation for this specific command C_RC_NA_1;

Adrian Aron adaron@cisco.com

3. Conclusion

Snort as IDS proves flexible to deploy in OT environments, even the standard ruleset is not giving more information than the OT engineers already know about their environment. The benefits appear when the IT mindset applies as a detection mechanism in OT networks.

Still, it is not easy for an IT solution to add visibility to OT networks, without the help and cooperation of OT engineers. The industrial process is known to them and far less to IT. By understanding the OT process and OT engineers, then augmenting their data with network detection made by IT tools, can increase the visibility in both the process communication methods and increase the security posture of the OT network.

Using and customizing snort in the OT environment as IDS tool can also adapt the English alerting messages to a native language alert message that is displayed on OT monitoring screens. The native language alert message might be much more useful OT personnel, who might not be so proficient in English and might not react due to misunderstandings.
Customizing the alert text messages can speed up the recovery or the detection of who is doing what and from where based on the identity of devices that are accessed and their importance in OT networks.

Adrian Aron adaron@cisco.com

4. References

Petr Matousec, (2017), Brno University of Technology, *Description and analysis of IEC 104 protocol* - Petr Matousec, from https://www.fit.vut.cz/research/publication-file/11570/TR-IEC104.pdf

Critical Infrastructure Series: Electrical Grid, from https://bitvijays.github.io/LFF-CIS-ElectricalGrid.html#scada-architecture

Thomas Teodorowicz (2017), *Comparison of SCADA protocols andimplementation of IEC 104 and MQTT in MOSAIK* - Thomas Teodorowicz, from https://www.uni-muenster.de/imperia/md/content/informatik/agremke/comparison_of_scada_protocols_and_implementation_of_iec_104_and_mqtt_in_mosaik.pdf

Beckhoff Information System, from https://bkinfosys.beckhoff.com/english.php?content=../content/1033/tcplclibiec870_5_104master/html/tcplclibiec870_5_104master_samples.htm

Infosec Institute, *Basic Snort rules Syntax and Usage*, from https://resources.infosecinstitute.com/snort-rules-workshop-part-one/

Infosec Institute, *Snort Lab: Payload Detection Rules (PCRE)* - from https://resources.infosecinstitute.com/snort-lab-payload-detection-rules-pcre/

SourceFire, *Snort Offset, Depth, Distance and Within* - Joe Esler, from https://blog.joelesler.net/2010/03/offset-depth-distance-and-within.html

NIST SP800-82 Rev.2, *Guide to Industrial Control Systems (ICS) Security,* from https://csrc.nist.gov/publications/detail/sp/800-82/rev-2/final

Luciana Obregon, *Secure Architecture forIndustrial Control Systems,* from https://www.sans.org/reading-room/whitepapers/ICS/secure-architecture-industrial-control-systems-36327

ISA99 Committee (2004), Manufacturing and Control Systems Security Part 1: Models and Terminology. Retrieved from http://isa99.isa.org/

Adrian Aron adaron@cisco.com