International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1367

(09/2020)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Secure applications and services (2) – Internet of things (IoT) security

Standard format for Internet of things (IoT) error logs for security incident operations

Recommendation ITU-T X.1367

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES (1)	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols (1)	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES (2)	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1319
Smart grid security	X.1330–X.1339
Certified mail	X.1340–X.1349
Internet of things (IoT) security	**X.1360–X.1369**
Intelligent transportation system (ITS) security	X.1370–X.1389
Distributed ledger technology security	X.1400–X.1429
Distributed ledger technology security	X.1430–X.1449
Security protocols (2)	X.1450–X.1459
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589
CLOUD COMPUTING SECURITY	
Overview of cloud computing security	X.1600–X.1601
Cloud computing security design	X.1602–X.1639
Cloud computing security best practices and guidelines	X.1640–X.1659
Cloud computing security implementation	X.1660–X.1679
Other cloud computing security	X.1680–X.1699
QUANTUM COMMUNICATION	
Terminologies	X.1700–X.1701
Quantum random number generator	X.1702–X.1709
Framework of QKDN security	X.1710–X.1711
Security design for QKDN	X.1712–X.1719
Security techniques for QKDN	X.1720–X.1729
DATA SECURITY	
Big Data Security	X.1750–X.1759
5G SECURITY	X.1800–X.1819

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1367

Standard format for Internet of things (IoT) error logs for security incident operations

Summary

There are two issues to handle security incidents from the Internet of things (IoT) ecosystem: The first is the incompatibility of protocols between computer networks using transmission control protocol/Internet protocol (TCP/IP) and IoT edge devices. The second is the lack of compatibility of error codes among edge device manufacturers.

Recommendation ITU-T X.1367 specifies a standardized error log format that can be placed in a protocol payload, such as syslog (see IETF RFC 5424) to be used for converting an error log information issued by an edge device to the standard error log format.

This Recommendation also specifies a standardized error code table to solve the second issue. As a result, security incidents across computer networks and networks for IoT edge devices can be integrally managed.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.1367	2020-09-03	17	11.1002/1000/14263

Keywords

Edge device, error code, error log format, incident response, Internet of things (IoT), security operation.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

		Page
1	Scope	1
2	References	1
3	Definitions	1
3.1	Terms defined elsewhere	1
3.2	Terms defined in this Recommendation	2
4	Abbreviations and acronyms	2
5	Conventions	3
6	General	3
6.1	Current error handling on an IoT ecosystem	3
6.2	Overview	3
7	Standard error log format for the IoT environment	4
7.1	Basic structure of error log format	4
7.2	Basic attributes	4
Annex A – Error code and error message		6
Appendix I – Examples of how to utilize error logs for incident operations		7
I.1	Attacker sends a random binary string to an edge device through IoTGW	7
I.2	Attacker sends an incorrect certification to an edge device through a compromised IoTGW	7
I.3	Attacker physically breaks a sensor device that regularly sends data without any request	8
I.4	How to use the IoT logs in an incident response	9
Appendix II – Prospective security incident response using Internet of things error log		12
Bibliography		13

Recommendation ITU-T X.1367

Standard format for Internet of things (IoT) error logs for security incident operations

1 Scope

This Recommendation specifies a standardized error format for Internet of things (IoT) error logs that can be placed in a protocol payload, such as syslog [b-IETF RFC 5424] for converting an error log information issued by an edge device to the standard error log format.

This Recommendation also specifies a standardized error code table to solve the lack of compatibility of error codes among edge device manufacturers. As a result, security incidents across computer networks and networks for IoT edge devices can be integrally managed.

2 References

None.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 actuator [b-ITU-T Y.4109]: A device performing physical actions caused by an input signal.

3.1.2 attack [b-ISO 13491-1]: Attempt by an adversary on the device to obtain or modify sensitive information or a service they are not authorized to obtain or modify.

3.1.3 authentication [b-ITU-T X.1277]: Authentication is the process in which a user employs their FIDO authenticator to prove possession of a registered key to a relying party.

3.1.4 authorization [b-ITU-T X.800]: The granting of rights, which includes the granting of access based on access rights.

3.1.5 device [b-ITU-T Y.4000]: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.6 Internet of things (IoT) [b-ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

3.1.7 human/machine interface (HMI) [b-ITU-T H.320]: Human-machine interface between user and terminal/system which consists of a physical section (electro-acoustic, electro-optic transducer, keys, etc.) and a logical section dealing with functional operation states.

3.1.8 malware [b-ISO/IEC 27033-1]: Malicious software designed specifically to damage or disrupt a system, attacking confidentiality, integrity and/or availability.

3.1.9 sensor [b-ITU-T Y.4105]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.10 thing [b-ITU-T Y.4000]: With regard to the Internet of things, this is an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks.

3.1.11 vulnerability [b-ISO/IEC 27000]: Weakness of an asset or control that can be exploited by one or more threats.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 certification: Third-party attestation related to products, processes, systems or persons.

NOTE – Based on the definition in [b-ISO/IEC 17000].

3.2.2 command and control (C&C) server: A server that sends commands to and controls computers (botnets) that have become bots after being infected with malware.

3.2.3 encryption: The cryptographic transformation of data to produce ciphertext.

NOTE – Based on the definition of encipherment in [b-ITU-T X.800], which treats encryption as its synonym.

3.2.4 Internet of things edge device: A terminal device of the IoT ecosystem that collects data from the real world with sensors or effects the real world with actuators.

3.2.5 Internet of things gateway (IoTGW): A device that connects a network for IoT edge devices and widely available computer networks, such as the Internet.

3.2.6 microcontroller unit (MCU): an embedded microprocessor that integrates arithmetic units, memory units, and input/output ports into one integrated circuit.

3.2.7 security incident response team (SIRT): A team that receives, investigates, and responds to reports on "security incidents".

3.2.8 security operations centre (SOC): A division that monitors the status of computers and networks in the organization and responds after detecting signs of malicious activities.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

C&C	Command and Control
FW	Firewall
HMI	Human/Machine Interface
ID	Identifier
IDS	Intrusion Detection System
IoT	Internet of Things
IoTGW	Internet of Things Gateway
IP	Internet Protocol
JSON	JavaScript Object Notation
LAN	Local Area Network
MCU	Microcontroller Unit
PC	Personal Computer
SIRT	Security Incident Response Team
SOC	Security Operation Centre
TCP	Transmission Control Protocol

5 Conventions

None.

6 General

6.1 Current error handling on an IoT ecosystem

In the context of error handling in IoT, if one of the components in an IoT system such as sensor or actuator goes out of order, an error code will be issued and recorded in an error log. If the error rarely occurs, no correction is needed. However, if the error continues to occur, the component should be replaced to fix the problem.

Figure 1 shows a typical IoT ecosystem consisting of IoT components such as microcontroller units (MCUs) associated with sensors and actuators, an Internet of things gateway (IoTGW) and a cloud.

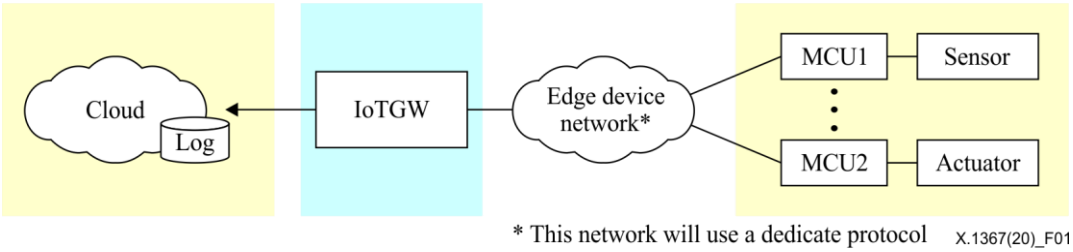

Figure 1 – Typical IoT ecosystem

The communication among IoTGW, MCU1, and MCU2 in Figure 1 is based on a dedicated protocol (see note) for the system. The MCU1-sensor and MCU2-actuator are examples of IoT edge devices. In this case, the IoTGW transmits a request to MCUs and the MCUs respond to the IoTGW with log information. The IoTGW also communicates with the cloud (a back-end system).

NOTE – A dedicated protocol can be one for big networks, such as oneM2M [b-oneM2M], ECHONET Lite [b-ISO/IEC 14543-4-3], or can be one for small networks such as SPI [b-SPI], I^2C [b-UM10204], Bluetooth [b-IEEE 802.15.1], Zigbee [b-IEEE 802.15.4], or a proprietary protocol designed only for this IoT system.

To manage an IoT ecosystem appropriately, because IoT edge devices sometimes break down, the IoT ecosystem needs to handle errors. Such systems handle error logs with error codes and initiate fixes. On the other hand, some systems analyse error logs stored with error codes to collate statistical information for their own improvement and even to handle security incidents. However, it is hard to unify the error logs of every IoT ecosystem and syslog [b-IETF RFC 5424].

6.2 Overview

To effectively respond to incidents, the provision of adequate capabilities for collecting and analysing error log information from components of the IoT ecosystem is key. However, the following concerns are recognized in the case of the IoT ecosystem.

1) There is a standardized procedure including the process for incident handing, by use of syslog [b-IETF RFC 5424], to collect log information from network devices. However, there is no such procedure for the IoT ecosystem.
2) IoT error log information cannot be stored in a single component of the IoT ecosystem, i.e., such log information should be collected by the back-end system (e.g., the cloud) or IoT edge system (e.g., an IoTGW) depending on the IoT ecosystem configuration.
3) Correlation analysis of error log information among different IoT components is difficult, because there is no standardized error log format.

4) Without a method of dealing with error log information, the IoT ecosystem will not be able to effectively maintain its IoT service.

This Recommendation describes a basic IoT system architecture for collecting IoT error logs to be used in incident-handling operations. It specifies standardized error codes and an error log format. By converting error codes adopted by each IoT ecosystem manufacturer into standardized error codes, the status of multiple IoT ecosystems can be more effectively monitored. Moreover, in the process of converting an error code to a standardized error code, the situation in which the error occurred is also recorded in associated error logs. As a result, security incidents can be handled across multiple IoT ecosystems. (see an example in Appendix I, clause I.4, multiple IoT ecosystem).

7 Standard error log format for the IoT environment

Because of poor computer resources, it can be difficult for IoT edge devices to implement new functions for handing IoT error logs. However, an IoTGW normally have better computer resources to do so. Requests and responses are often exchanged between IoTGW and cloud systems, and such communications include IoT error log information. Therefore, in this Recommendation it is required for an IoTGW to generate standardized error log information and to communicate it to cloud systems.

7.1 Basic structure of error log format

An IoT error log has the format described in Figure 2 by using JavaScript object notation (JSON) with regular expression. This format can be transformed for use by syslog or XML (see note). Examples of this format are described in Appendix I.

NOTE – This Recommendation does not define the length of every attribution value. The agreement between the sender and the receiver of the error log on the length of each attribute value will be needed in advance.

```
{
   "Timestamp":
        "^([0-9]{4})-(1[0-2]|0[1-9])-(3[01]|0[1-9]|[12][0-9])T
          (2[0-3]|[01][0-9]):([0-5][0-9]):([0-5][0-9])(\.[0-9]{+})Z$",
   "Reporter": {},
   "Protocol": String,
   "Requester": {},
   "Responder": {},
   "Error Code": "/^[0-9A-F]^{+}$/",
   "Error Message": String,
   "Description": String | {},
}
```

Figure 2 – Elements of an error record

In most cases, the reporter of this error log is the IoTGW. Attributes used in this format are defined in clause 7.2.

7.2 Basic attributes

The following attributes are elements for error log information.

a) Timestamp: A timestamp (see note) is required for the issue of an error log. For example, in the case of 20 September 2018 – 13 h 25 min and 51 s, the timestamp should be described as "2018-09-20T13:25:51.0Z." [b-ISO 8601-1].

NOTE – "Timestamp" is mandatory, even though the transmission protocol carrying an error log specified by this Recommendation also has a timestamp field, because timestamp usually changes between problem occurrence and the transfer time.

b) Reporter: The IoT component which converts a device's error code to a standard error code and sending it to the cloud. The detailed description is in Figure 3.

```
"Reporter":{
    "IP Address":
        "(^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}
              (?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$)
         |(^(?:[A-F0-9]{1,4}:){7}[A-F0-9]{1,4}$)"
}
```

Figure 3 – Elements of reporter

1) Internet protocol (IP) address: The unique identifier (ID) of the reporter on the Ethernet.

c) Protocol: Protocol name used among IoTGW and IoT edge devices.

d) Requester or responder: The IoT component which sends a request to an IoT edge device or an IoT component which replies to a request. The detailed description is in Figure 4.

```
"Requester(or Responder)": {
    "Unique ID": string,
    "Transmitted Code": "( [0-9A-F])^{+}}
```

Figure 4 – Elements of requester and responder

1) Unique ID (optional): This is a unique number or a unique code related to each device defined by the protocol of the IoT edge device network.
2) Transmitted code (optional): Content of data transmitted. It is expressed in hexadecimals.
3) It should be expressed as in Figure 5 when an error occurs without Requester or Responder, like a sensor device sending data periodically without being requested.

```
"Requester(or Responder)": {}
```

Figure 5 – The case of no requester or no responder

e) Error code, error message: Error code and error message are specified in Annex A.

f) Description (optional): Any sentences, phrases and sub-elements of JSON can be expressed, if notification of the situation of IoT edge devices or their networks is required.

The error message can be omitted if communication bandwidth is narrow or storage size is limited. The description can be omitted if it is not necessary to write any additional text.

Annex A

Error code and error message

(This annex forms an integral part of this Recommendation.)

Error code and error message are specified in Table A.1.

Table A.1 – Error code and error message

Code	Message	Description
	No Error (0x00-0x0F)	
0x00	No Error	No error occurs.
	Communication (0x10-0x1F)	
0x10	No Response	No response even though the request requires any responses.
0x11	Communication Failed	Some problems for failing communication.
0x12	Link Down	A network interface link is down.
0x1E	Extended Reasons	Prefix code for extended reasons.
0x1F	Other Communication Reasons	Other reasons related to communication.
	Security (0x20-0x2F)	
0x20	Authentication Failed	Some problems of the authentication.
0x21	Certification Failed	Some problems of the certification.
0x22	Encryption Failed	Some problems of the encryption.
0x23	Authorization Failed	Some problems of the authorization.
0x2E	Extended Reasons	Prefix code for extended reasons.
0x2F	Other Security Reasons	Other reasons related to the authentication, the certification, the encryption or the authorization.
	Command (0x30-0x3F)	
0x30	Invalid Command	The command is undefined or invalid.
0x31	Invalid Argument	The argument is out of range or invalid.
0x3E	Extended Reasons	Prefix code for extended reasons.
0x3F	Other Command Reasons	Other reasons related to the command.
	Device (0x40-0x4F)	
0x40	Device Broken	A part of the device has unrecoverably broken.
0x41	Device Failed	A part of the device has failed but is recoverable.
0x42	Out of Resources	Running out of storage, memory, and any computing resources.
0x4E	Extended Reasons	Prefix code for extended reasons.
0x4F	Other Device Reasons	Other reasons related to the device.
	Reserved for future extension (0x50-0xDF)	
	Reserved for private applications (0xE0-0xEF)	
	Others (0xF0-0xFF)	
0xFF	Other Reasons	Other reasons except for the above reasons.

Appendix I

Examples of how to utilize error logs for incident operations

(This appendix does not form an integral part of this Recommendation.)

I.1 Attacker sends a random binary string to an edge device through IoTGW

Figure I.1 shows an attacker sending a random binary string to MCU1 and its reaction. The head of the random binary string "01000001" identifies device No. 0001 and function No. 0100. In other words, the attacker tries to attack function No. 0100 of device No. 0001. MCU1 cannot understand "3A459187F43CDDE5." So it responds "000003FF (unknown command)" to device No. 0000 (IoTGW).

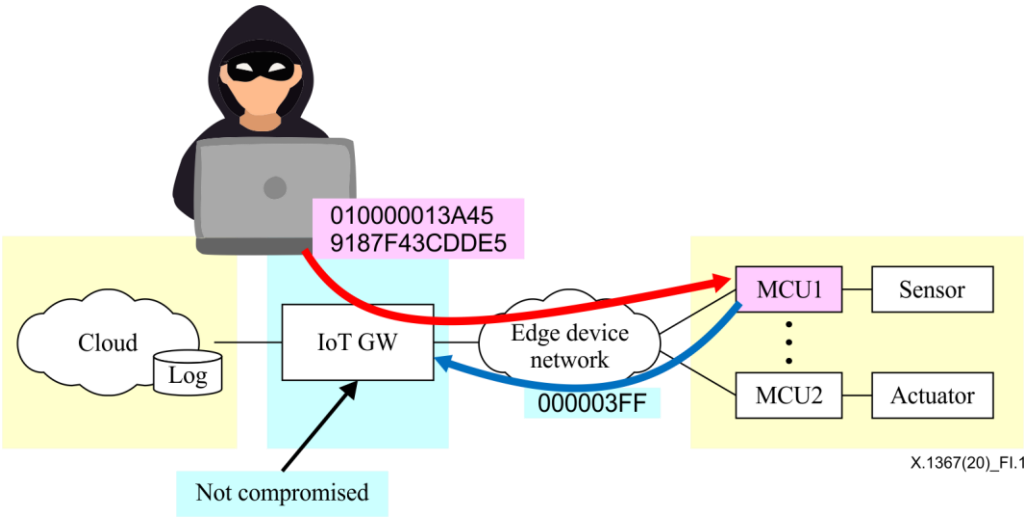

Figure I.1 – Attacker sends a random binary string to MCU1

After IoTGW receives "000003FF" from MCU1, IoTGW builds the following log and send it to a log server on the cloud. See Figure I.2.

```
{
    "Timestamp":"2018-08-28T09:34:55.0Z",
    "Reporter": { "IP Address": "192.168.2.11"},
    "Protocol": "ABC company protocol",
    "Requester": {
        "Unique ID": "0000",
        "Transmitted Code": "010000013A459187F43CdDE5"
    },
    "Responder": {
        "Unique ID": "0001",
        "Transmitted Code": "000003FF"
    },
    "Error Code": "30",
    "Error Message": "Invalid Command",
}
```

Figure I.2 – IoT error log for a bad request (example)

I.2 Attacker sends an incorrect certification to an edge device through a compromised IoTGW

Figure I.3 shows an attacker sending an incorrect certification to MCU2 through a compromised IoTGW and its reaction. MCU2 identifies that the certification is invalid. So, it responds

"00000010000c (certificate verification error)" to device No. 0000 (an IoTGW). However, the IoTGW is compromised. The IoTGW will never send an error log to the log server on the cloud. The intrusion detection system (IDS) sends an IoT error log instead of the IoTGW.

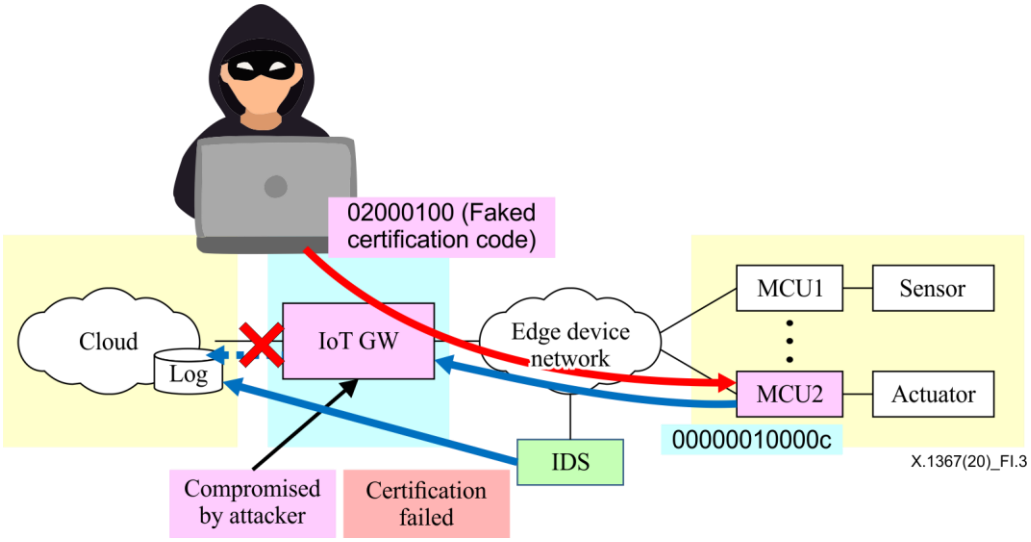

Figure I.3 – Attacker sends a faked certification code

The attacker sends a faked certification code to MCU2, but MCU2 cannot recognize it. So, MCU2 sends "00000010000c (Certificate verification error)." The IoTGW ignores the response message from MCU2, because the attacker has compromised it previously. However, if the system has an IDS, the IDS constructs the IoT error log in Figure I.4 instead of the IoTGW. In this case, IDS becomes a reporter.

```
{
    "Timestamp":"2018-08-28T09:34:55.0Z",
    "Reporter": { "IP Address": "192.168.100.249"},
    "Protocol": "EEE Company's protocol",
    "Requester": {
        "Unique Name": "0000",
        "Transmitted Code": "02000100... (Faked certification codes)"
    },
    "Responder": {
        "Unique Name": "0002",
        "Transmitted Code": "000000000010000c "
    },
    "Error Code": "21",
    "Error Message": "Certification Failed",
    "Description": {
        "Status": "This message was sent from IDS not IoTGW"
    },
}
```

Figure I.4 – IoT error log for a certificate verification error (example)

I.3 Attacker physically breaks a sensor device that regularly sends data without any request

Figure I.5 shows an attacker physically breaking MCU1, which regularly sends data without any request. MCU1 cannot send any data after that. In this case, the IoTGW detects an abnormal situation related to MCU1, and sends the IoT error log shown in Figure I.6.

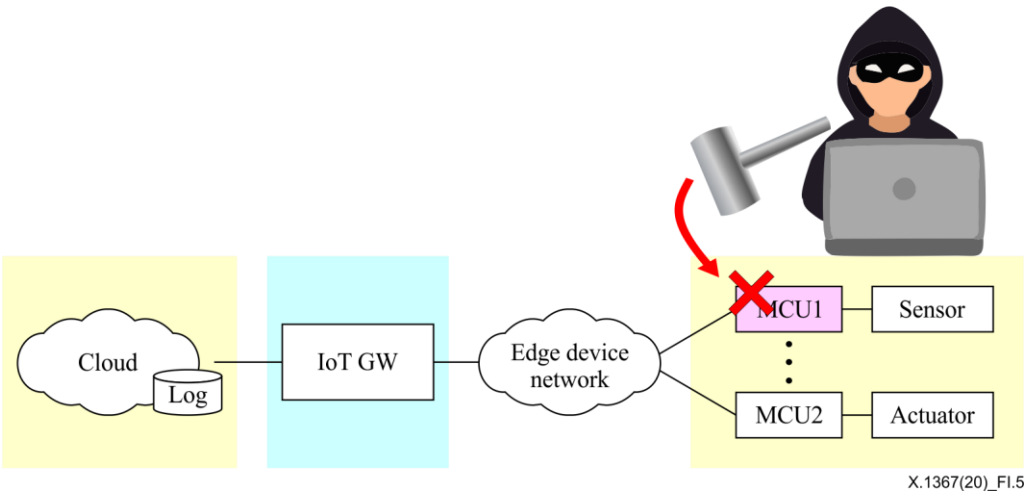

Figure I.5 – Attacker physically breaks MCU1

```
{
    "Timestamp":"2018-08-28T09:34:55.0Z",
    "Reporter": { "IP Address": "192.168.10.254"},
    "Protocol": "ZZZ Company's protocol",
    "Requester": {},
    "Responder": {
        "Unique Name": "0002",
        "Transmitted Code": ""
    },
    "Error Code": "11",
    "Error Message": "Communication Failed",
    "Description": {
        "Responder stopped sending data."
    },
}
```

Figure I.6 – IoT error log sent for losing a communication (example)

I.4 How to use the IoT logs in an incident response

A factory has three IoT ecosystems on the same local area network (LAN) as shown in Figure I.7.

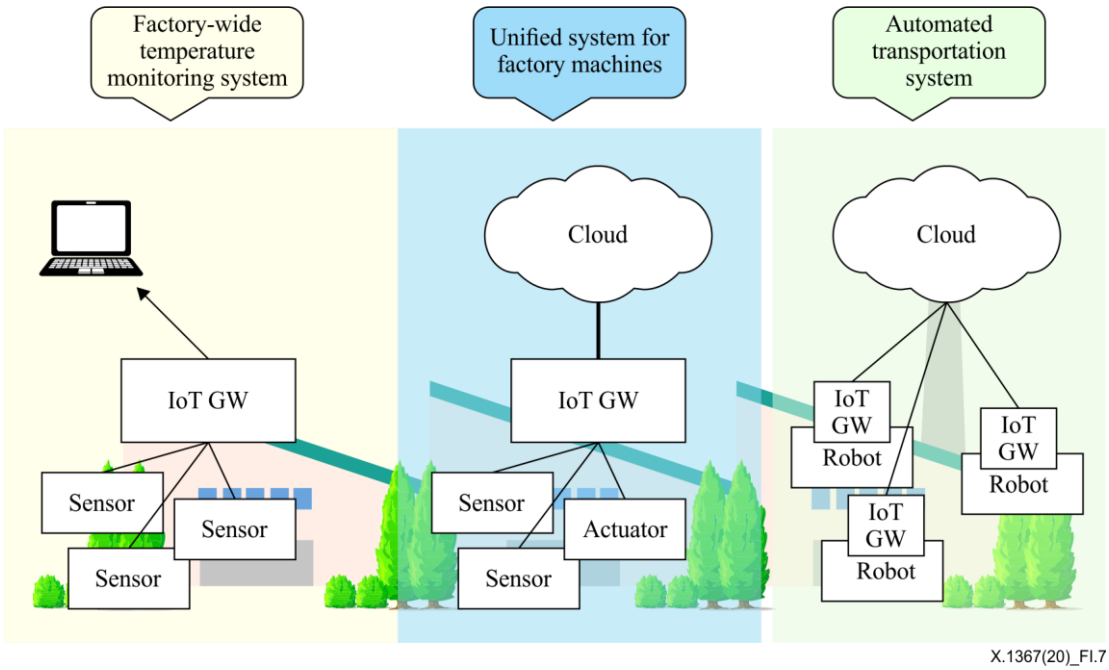

Figure I.7 – Multiple IoT ecosystems in a factory

In this case, an attacker can easily attack each device, as shown in Figure I.8. If there are no IoT error logs, it is hard for a security incident response team (SIRT) to identify the footprints of the attacker, because the SIRT cannot unify non-standard IoT logs from each IoT ecosystem.

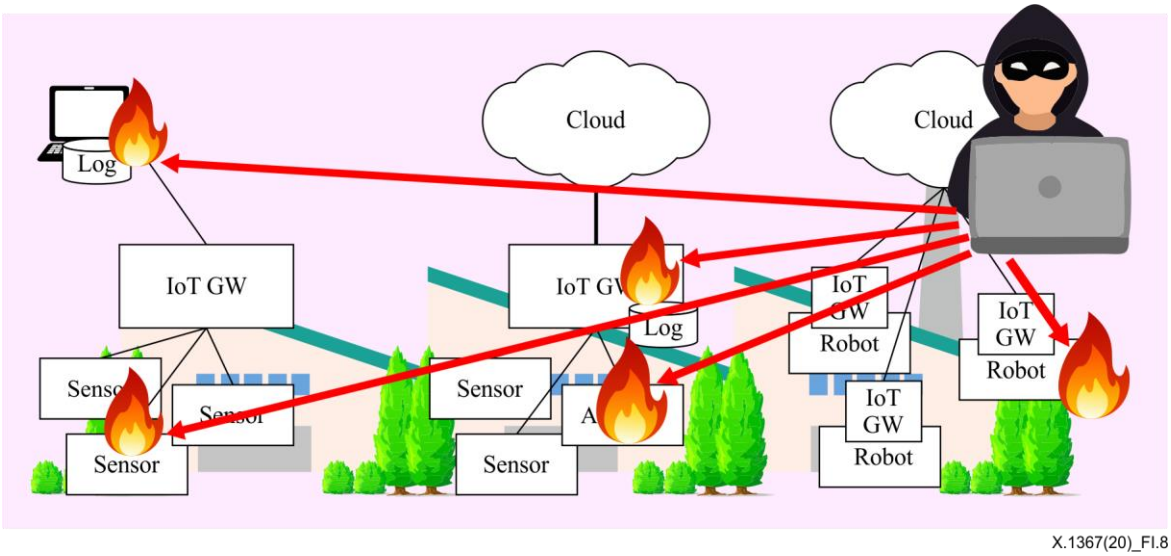

Figure I.8 – Attacker targets all IoT ecosystems at the same time

If these IoT ecosystems use the standard IoT error log format, the SIRT can easily unify all IoT error logs as shown in Figure I.9, as well as unifying syslog. The footprints of the attacker can then be identified.

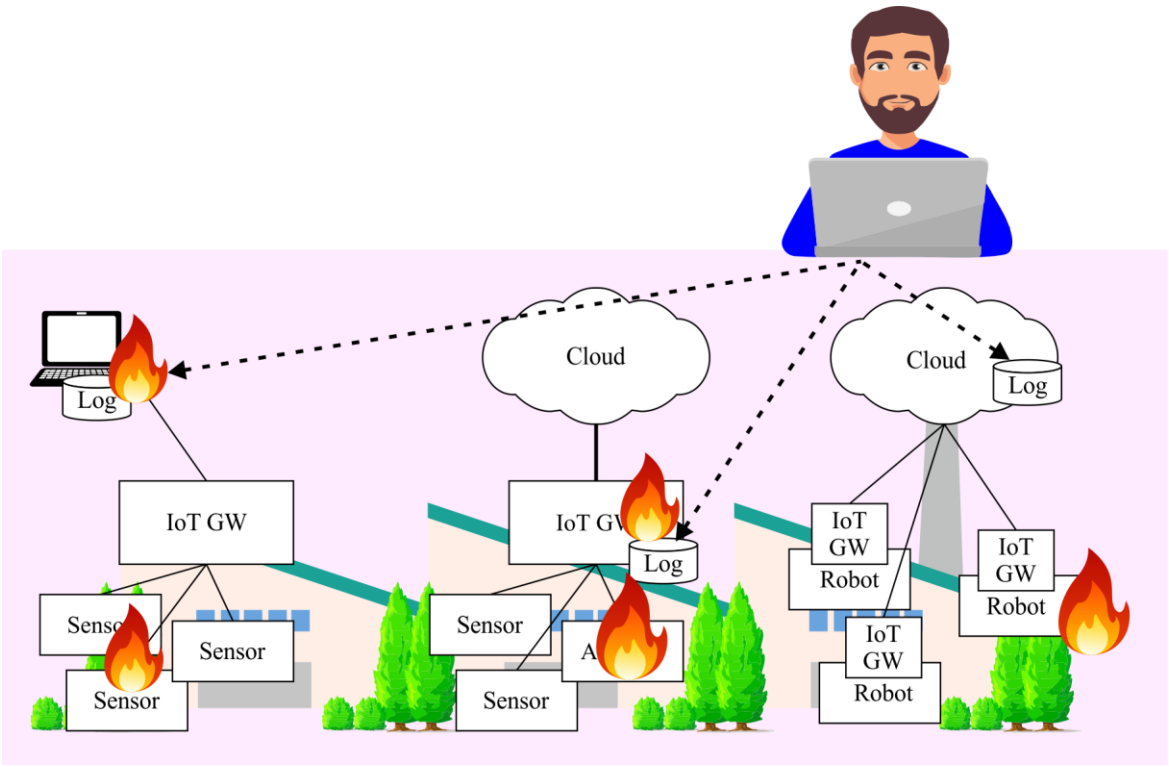

Figure I.9 – Unify and analyse all IoT error logs

Appendix II

Prospective security incident response using Internet of things error log

(This appendix does not form an integral part of this Recommendation.)

The standard IoT error log described in this Recommendation can be used for security incident response for an IoT ecosystem. Figure II.1 shows a flow of incident detection using IoT error log.

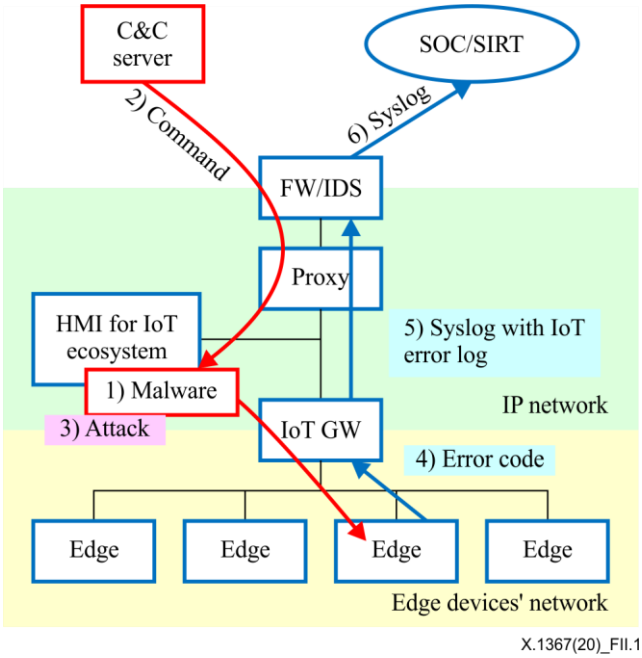

Figure II.1 – Incident detection using IoT error log

(1) Malware somehow infects a human/machine interface (HMI), e.g., a desktop personal computer (PC).

(2) Malware receives commands from the command and control (C&C) server. A firewall (FW), IDS or proxy may record a log of communication between the malware and the C&C server.

(3) Malware looks for a vulnerability of the IoT edge device. The IoT edge device sends many error codes to the IoTGW, because the malware tries to frequently send commands with various parameters. Almost always their parameters are incorrect for the IoT edge device.

(4) For every IoT error code sent, the IoTGW sends an IoT error log in the format specified in this Recommendation to the FW or IDS using the syslog protocol.

(5) A proxy also sends communication logs using the syslog protocol to the FW or IDS, which sends all logs, including the IoT error log, with the syslog protocol to a security operation centre (SOC) or SIRT.

Security analysts in SOC or SIRT may not respond to a single IoT error log which is not related to other IoT error logs. They, however, may notice an abnormal situation if the SOC or SIRT receives a large number of IoT error logs continuously, because this could be due to an attacker looking for an IoT system vulnerability.

This Recommendation allows SOC or SIRT analysts to detect attacks on IoT edge devices, because the SOC or SIRT receives IoT error codes in the standard IoT error log format. They will then check other related logs, including communications between the malware and the C&C server, and notify factory operator to separate this HMI desktop PC from a LAN or remove the malware from the HMI desktop PC.

Bibliography

[b-ITU-T X.800] Recommendation ITU-T X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications*.

[b-ITU-T X.1277] Recommendation ITU-T X.1277 (2018), *Universal authentication framework.*

[b-ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of things*.

[b-ITU-T Y.4105] Recommendation ITU-T Y.4105/Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

[b-ITU-T Y.4109] Recommendation ITU-T Y.4109/Y.2061 (2012), *Requirements for the support of machine-oriented communication applications in the next generation network environment*.

[b-ISO 8601-1] ISO 8601-1:2019, *Date and time – Representations for information interchange – Part 1: Basic rules*.

[b-ISO 13491-1] ISO 13491-1:2016, *Financial services – Secure cryptographic devices (retail) – Part 1: Concepts, requirements and evaluation methods*.

[b-ISO/IEC 14543-4-3] ISO/IEC 14543-4-3:2015, *Information technology – Home Electronic Systems (HES) architecture – Part 4-3: Application layer interface to lower communications layers for network enhanced control devices of HES Class 1*.

[b-ISO/IEC 17000] ISO/IEC 17000:2004, *Conformity assessment – Vocabulary and general principles*.

[b-ISO/IEC 27000] ISO/IEC 27000:2018, *Information technology – Security techniques – Information security management systems – Overview and vocabulary*.

[b-ISO/IEC 27033-1] ISO/IEC 27033-1:2015, *Information technology – Security techniques – Network security – Part 1: Overview and concepts*.

[b-ISO/IEC 27039] ISO/IEC 27039:2015, *Information technology – Security techniques – Selection, deployment and operations of intrusion detection and prevention systems (IDPS)*.

[b-IEEE 802.15.1] IEEE 802.15.1-2005, *IEEE Standard for information technology – Local and metropolitan area networks – Specific requirements – Part 15.1a: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPAN)*.

[b-IEEE 802.15.4] IEEE 802.15.4-2015, *IEEE Standard for low-rate wireless networks*.

[b-IETF RFC 5424] IETF RFC 5424 (2009), *The syslog protocol*.

[b-oneM2M] oneM2M Partners (2017), Standards for M2M and Internet of things. Available [viewed 2020-02-12] at: http://www.onem2m.org/

[b-SPI] Motorola (2001), SPI block guide, V04.01. Available [viewed 2020-02-12] at: https://www.nxp.com/files-static/microcontrollers/doc/ref_manual/S12SPIV4.pdf

[b-UM10204] UM10204 (2014), *I^2C-bus specification and user manual*, Rev.6.Available[viewed 2020-02-12] at: https://www.nxp.com/docs/en/user-guide/UM10204.pdf

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2020