IoT機器セキュリティログ検討WG

IoT機器のエラーログとSOC連携の標準化の最新動向

2020年1月21日

NPO 日本ネットワークセキュリティ協会
標準化部会　IoT機器セキュリティログ検討WG
渥美　清隆

アウトライン

- 問題の背景
- 想定される攻撃シナリオと対策
- ITU-T標準化の取り組み
- まとめ

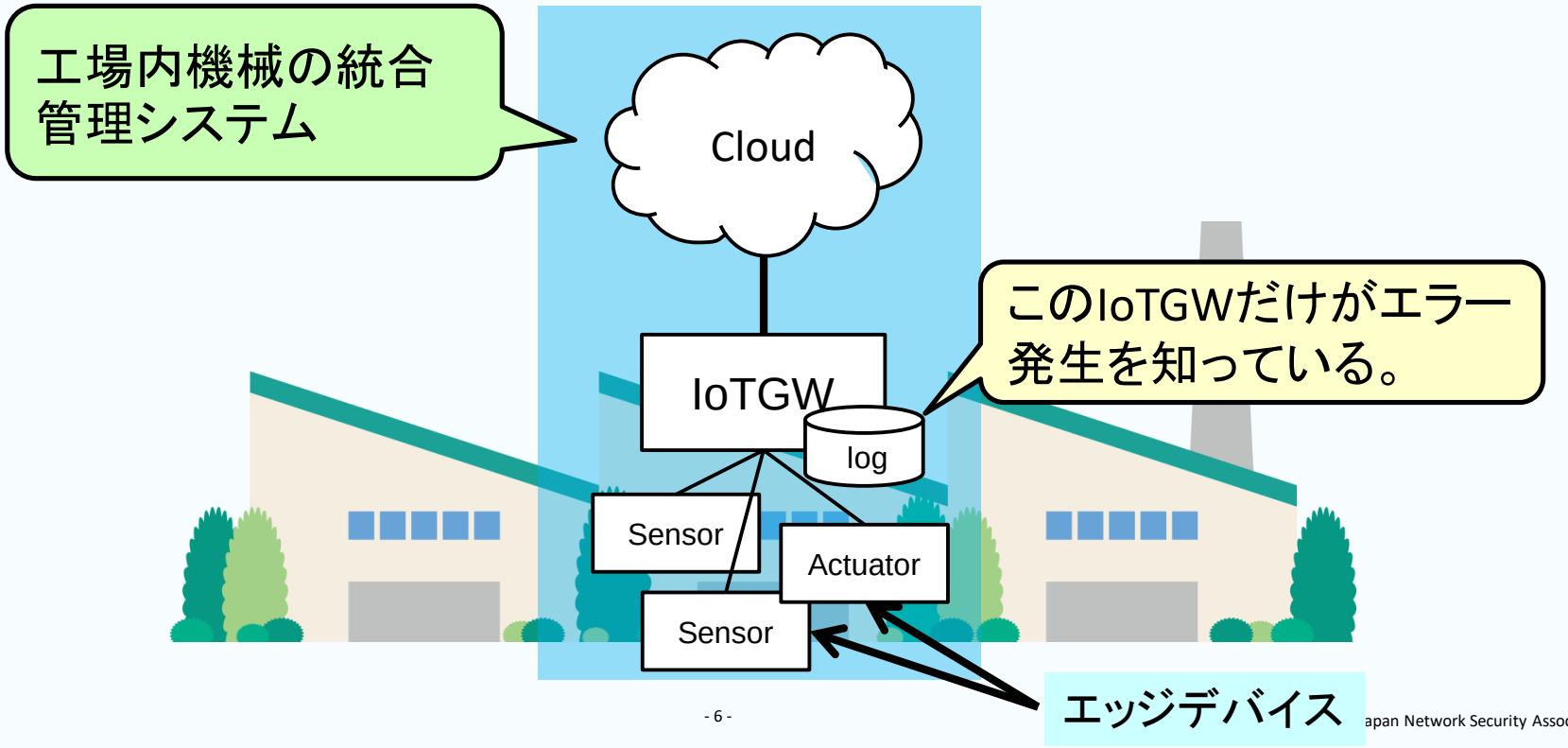
アウトライン

- 問題の背景
- 想定される攻撃シナリオと対策
- ITU-T標準化の取り組み
- まとめ

複数のIoTエコシステムが導入される過程

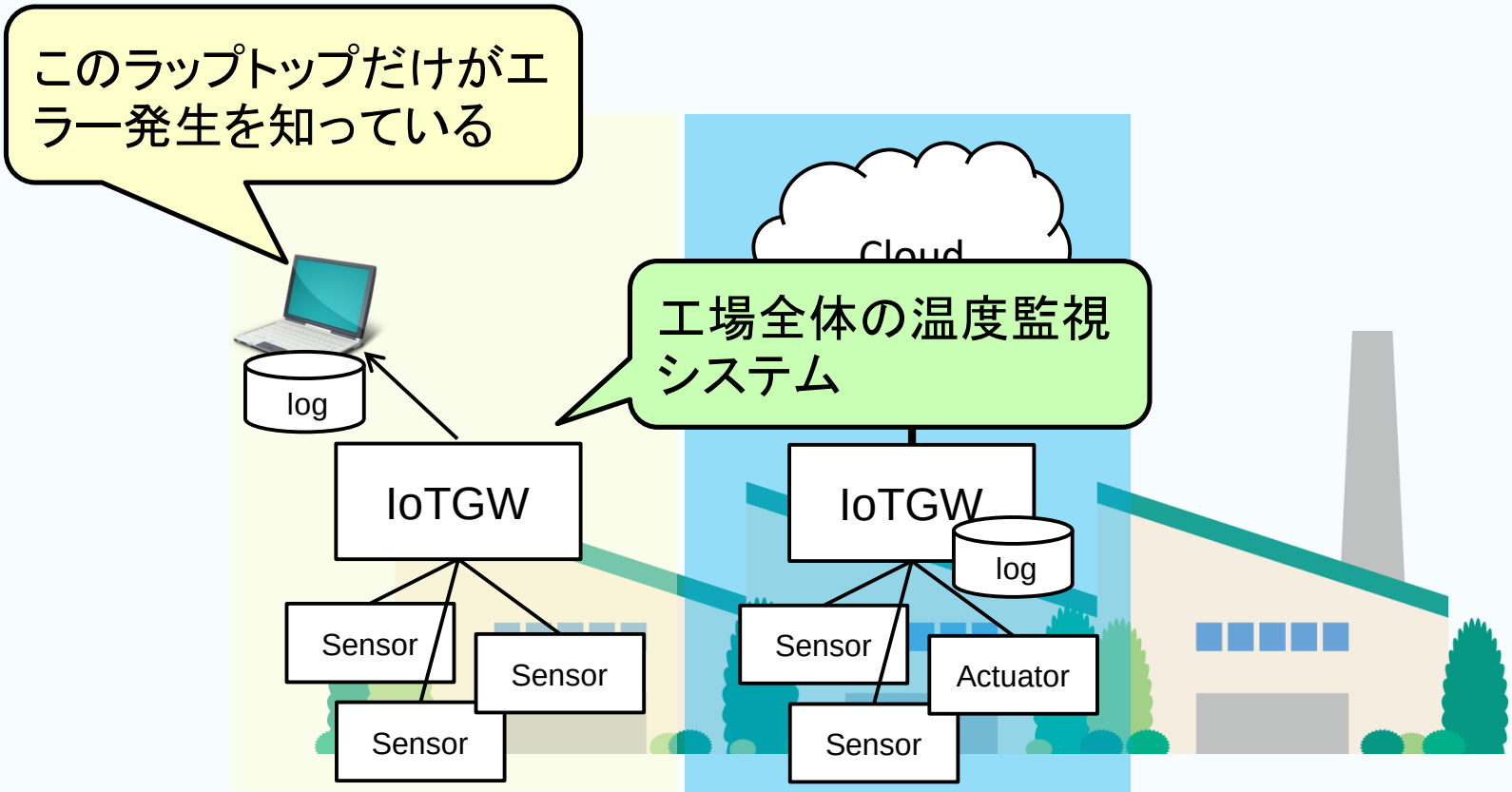
複数のIoTエコシステムが導入される過程

- 最初のIoTエコシステム導入



複数のIoTエコシステムが導入される過程

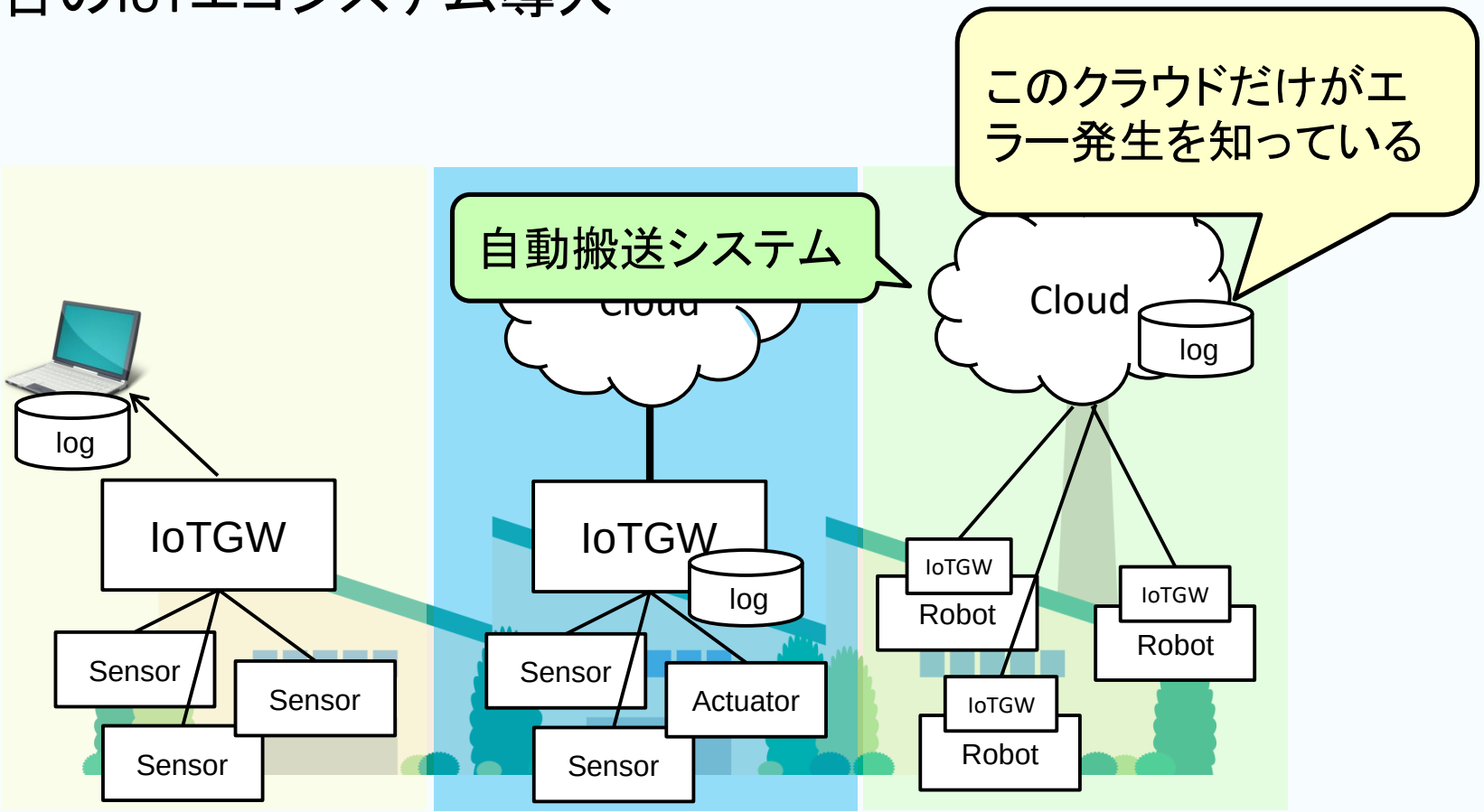
- 最初のIoTエコシステム導入
- 2つ目のIoTエコシステム導入



複数のIoTエコシステムが導入される過程

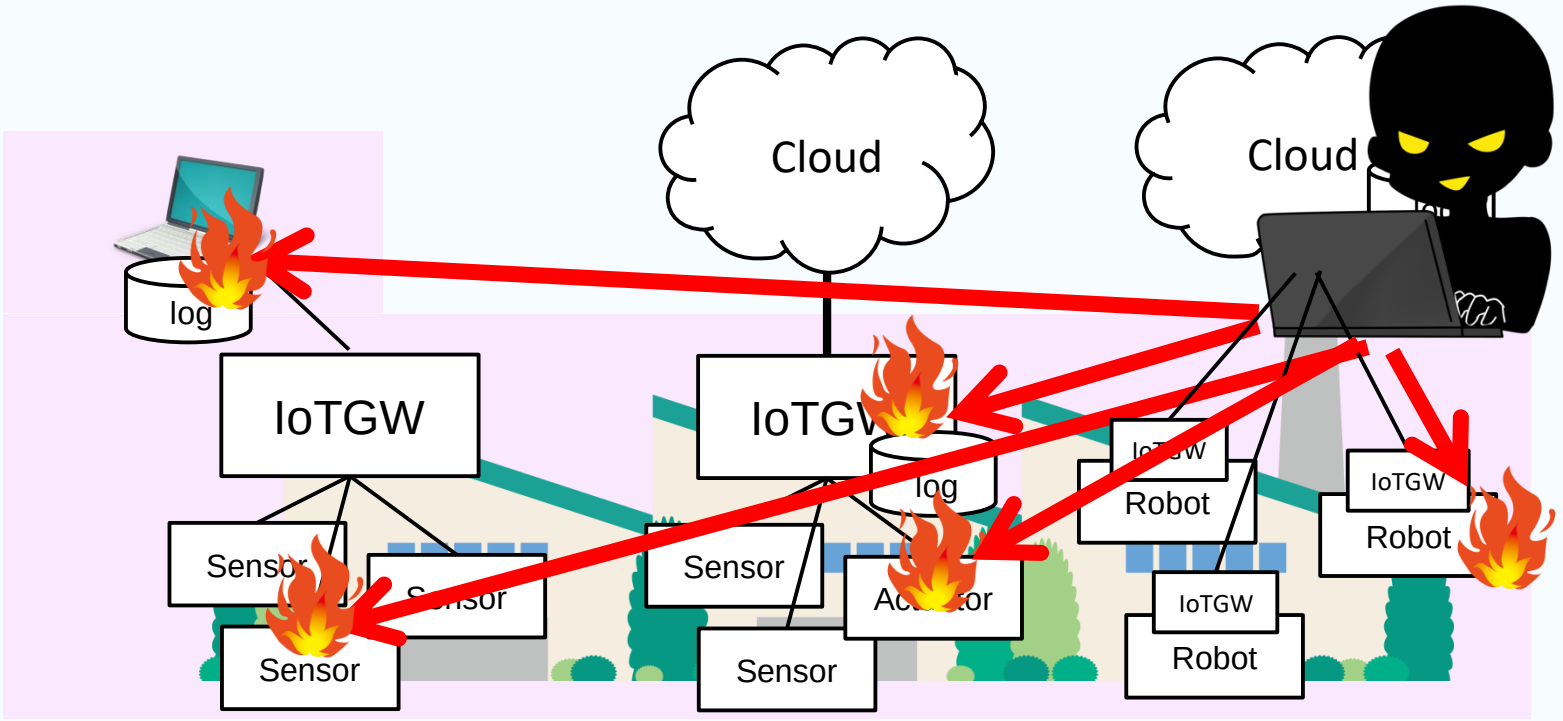
JNSA

- 最初のIoTエコシステム導入
- 2つ目のIoTエコシステム導入
- 3つ目のIoTエコシステム導入



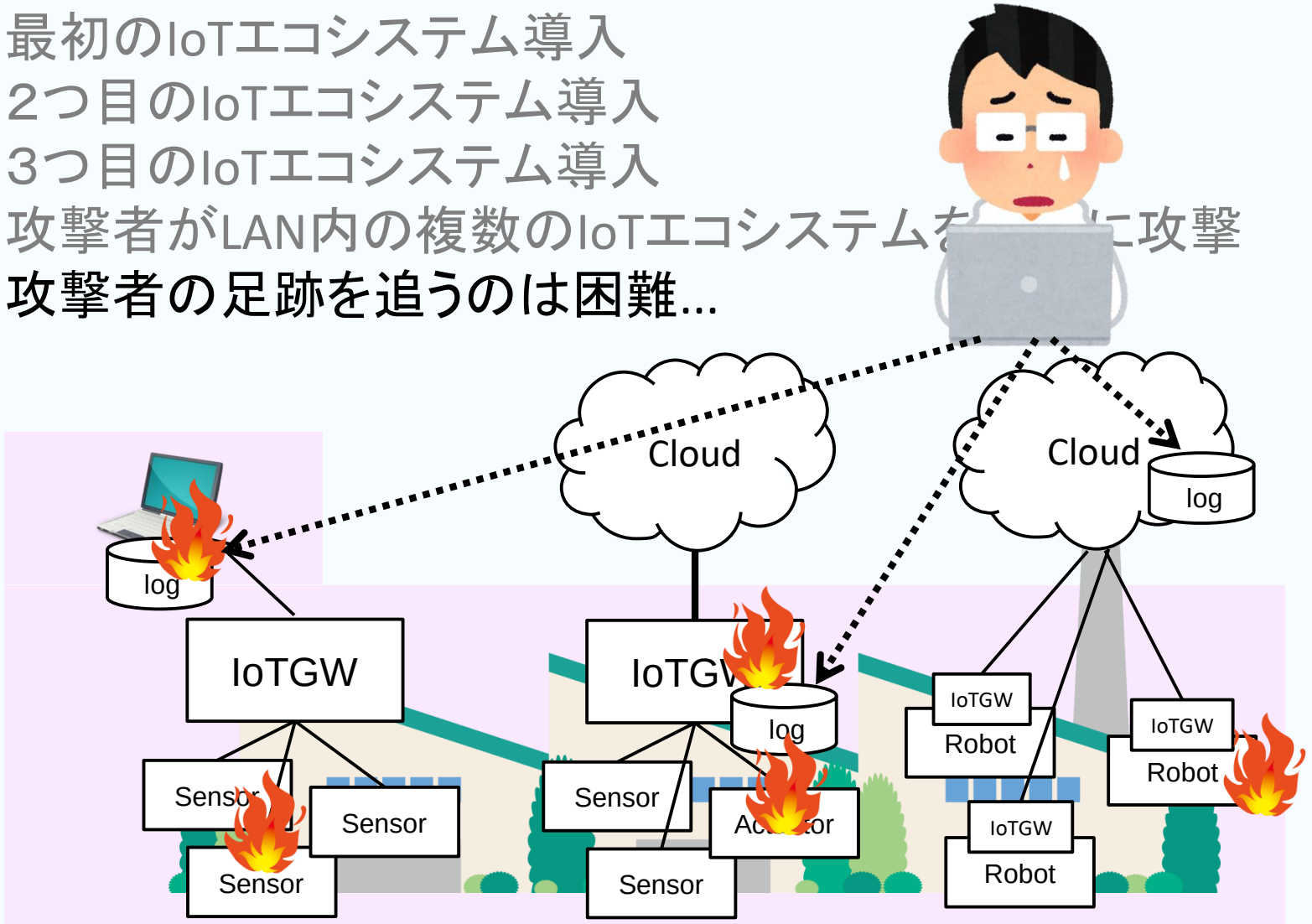
複数のIoTエコシステムが導入される過程

- 最初のIoTエコシステム導入
- 2つ目のIoTエコシステム導入
- 3つ目のIoTエコシステム導入
- 攻撃者がLAN内の複数のIoTエコシステムを自由に攻撃



複数のIoTエコシステムが導入される過程

- 最初のIoTエコシステム導入
- 2つ目のIoTエコシステム導入
- 3つ目のIoTエコシステム導入
- 攻撃者がLAN内の複数のIoTエコシステムを[illegible]に攻撃
- 攻撃者の足跡を追うのは困難…



何が起こったのか？

- 独立した縦割りのIoTエコシステムの導入
- 同一LANへの複数のIoTエコシステムの導入 or
- 論理的に接続された複数のLANへのIoTエコシステムの導入
- 内部侵入を果たした攻撃者は全てのIoTエコシステムに同時アクセス可能
- セキュリティオペレータは問題追跡が困難

アウトライン

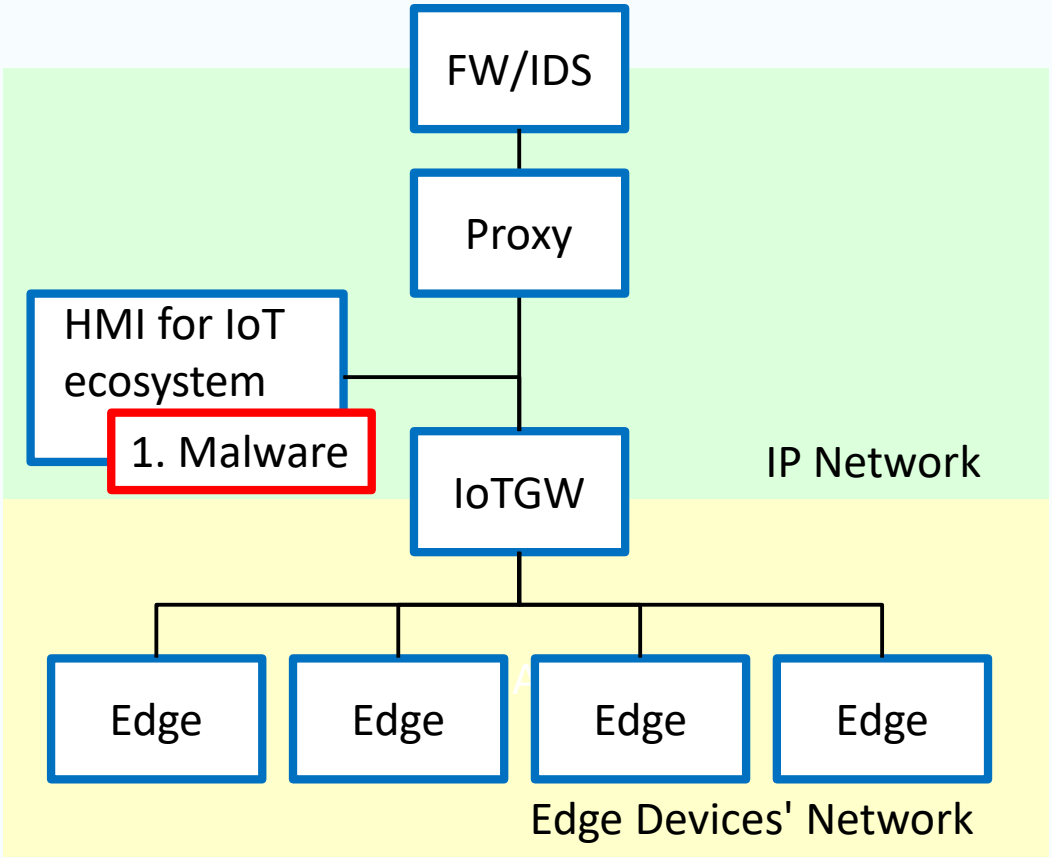
- 問題の背景
- 想定される攻撃シナリオと対策
- ITU-T標準化の取り組み
- まとめ

攻撃の流れ

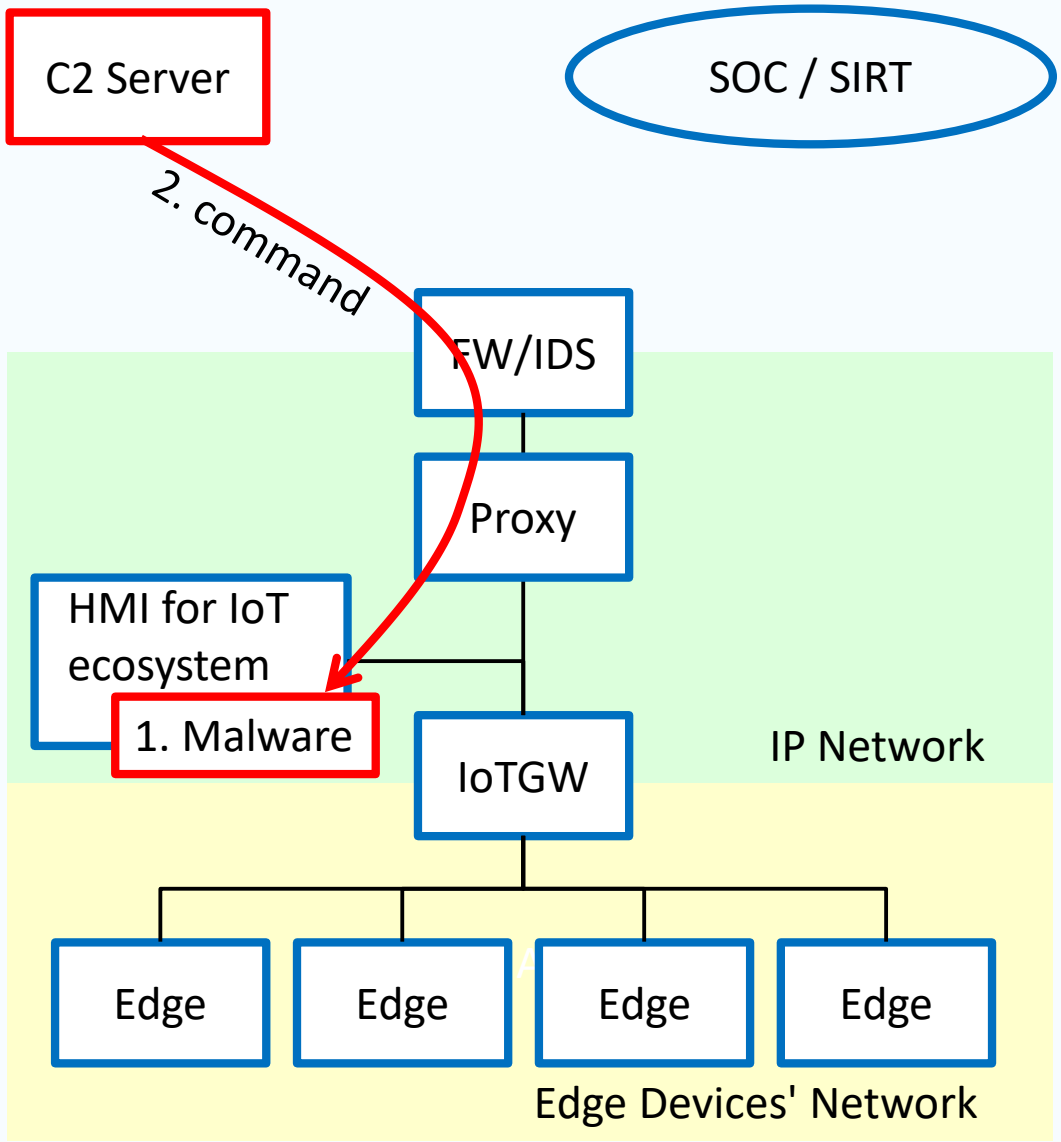
C2 Server

SOC / SIRT

1. IoTエコシステム制御卓にマルウェアの植付け

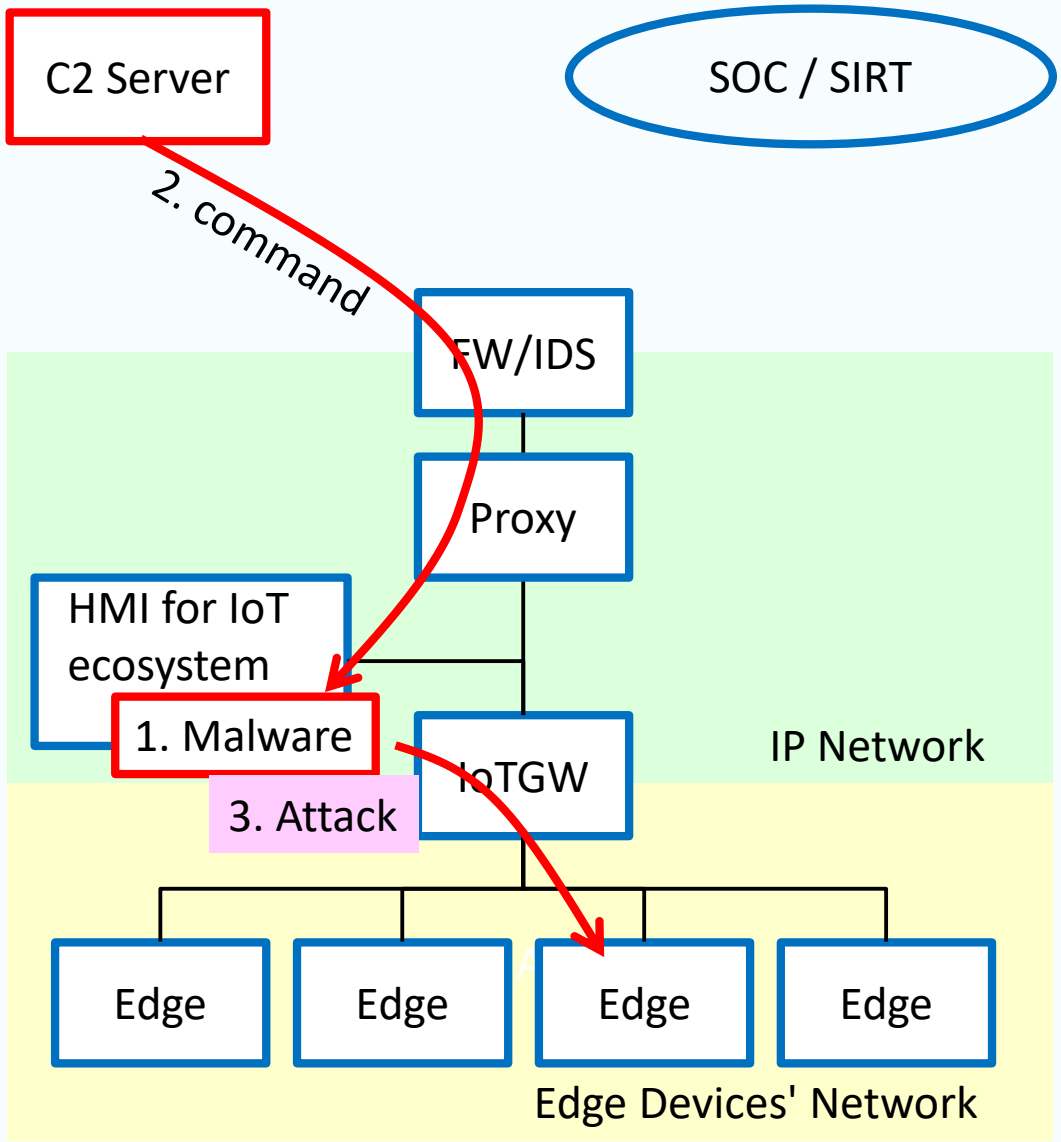


攻撃の流れ



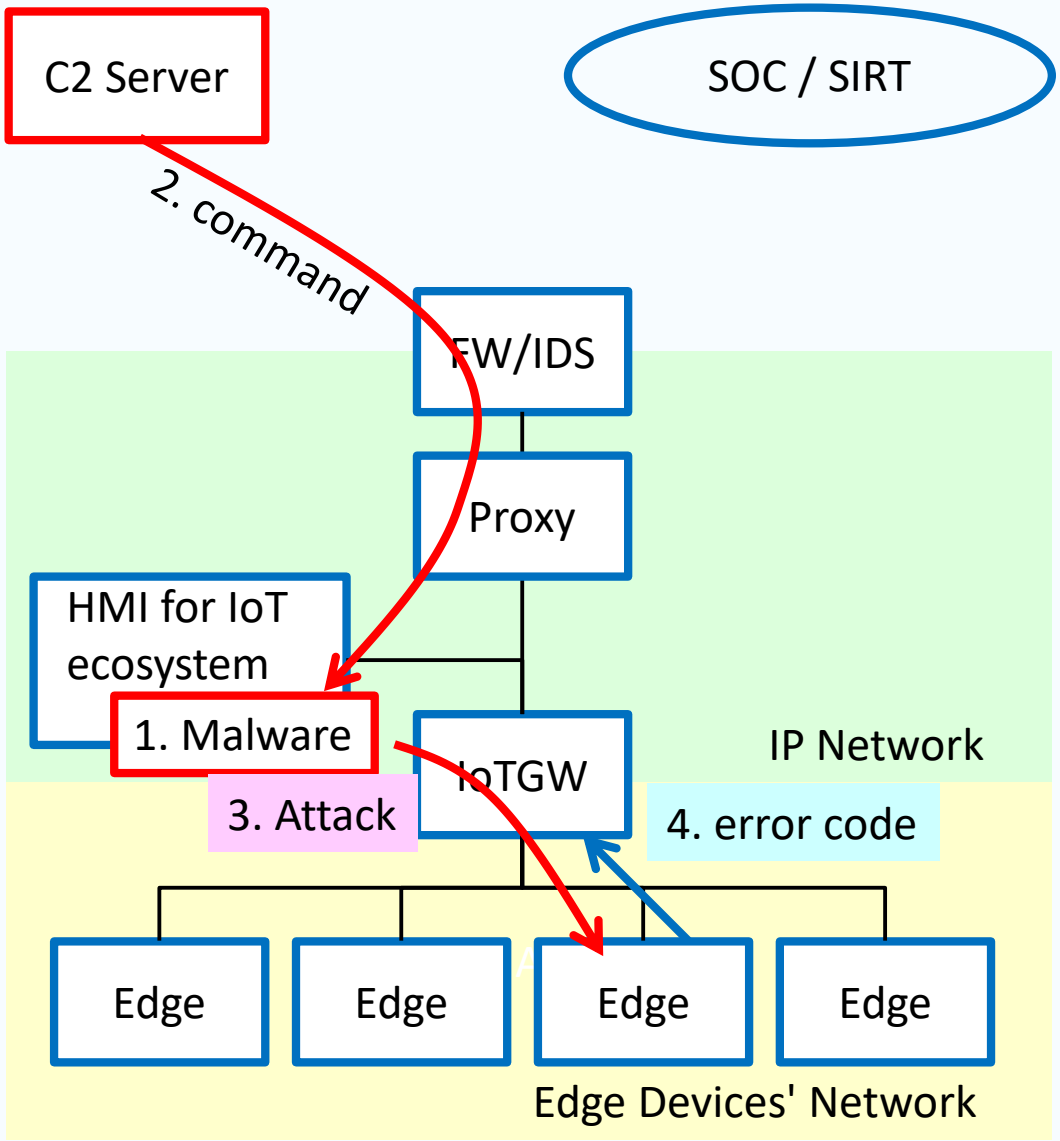
1. IoTエコシステム制御卓にマルウェアの植付け
2. マルウェアがC2サーバからコマンド取得

攻撃の流れ



1. IoTエコシステム制御卓にマルウェアの植付け
2. マルウェアがC2サーバからコマンド取得
3. マルウェアがIoTGWを通してエッジデバイスにアクセス

攻撃の流れ



1. IoTエコシステム制御卓にマルウェアの植付け
2. マルウェアがC2サーバからコマンド取得
3. マルウェアがIoTGWを通してエッジデバイスにアクセス
4. エッジデバイスでエラーが発生

貫通するパラメータを探索するために3，4を繰り返す。

エラーログの取り扱いの改善

- 点在するエラーログ
  - ✓ エラーログがどこにあるか分からない
  - ✓ 突合できるようにするまでが手間
- 異なるエラーログ取得方法
  - ✓ 標準的なエラーログ取得方法無し
  - ✓ HMIからキーボード・マウスを操作しないと得られない
- プロプライエタリのエラーコード
  - ✓ HEMS/BEMSなど標準化されたエラーコードは少数
  - ✓ バイナリコードの場合も多く可読性が低い
- エラー発生の原因特定のための情報消失
  - ✓ エラー発生の原因になったリクエストが分からない
  - ✓ エラー発生の原因になったリクエスタが分からない

エラーログの取り扱いの改善

- 点在するエラーログ
  - ✓ エラ…合も
  - ✓ 突合…
- 異なるエラーログ取得方法
  - ✓ 標準…
  - ✓ HM…得られないことも
- プロプライエタリのエラーコード
  - ✓ HEM…は少数
  - ✓ バイ…
- エラー発生の原因特定のための情報消失
  - ✓ エラ…いらない
  - ✓ エラ…いらない

Syslog Serverへの集約

Syslog Serverへの集約

標準エラーコードへのマッピング

標準エラーログフォーマットの定義と要素の充足方法の確立

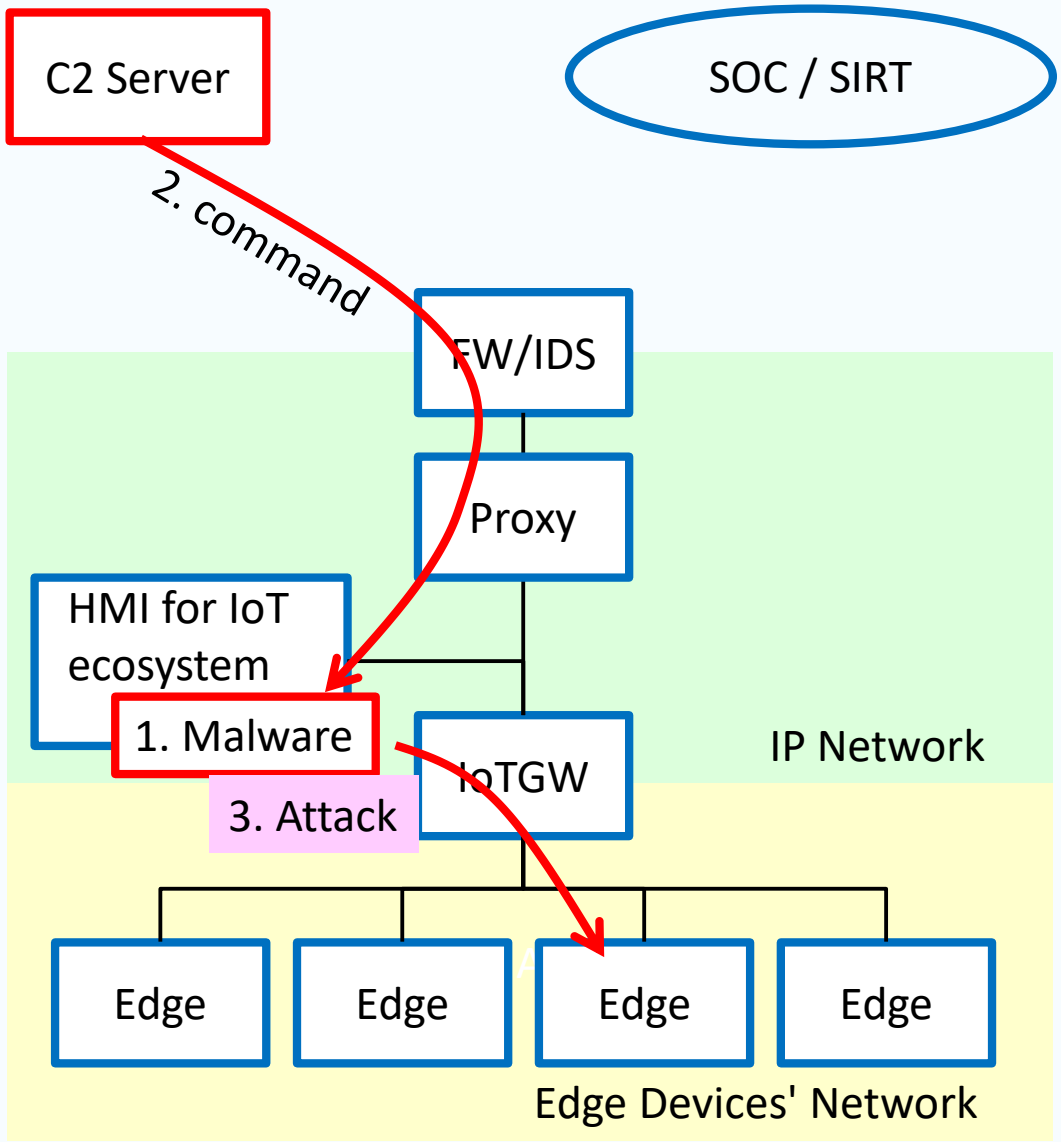
エラーログの取り扱いの改善

- Syslog Serverへの集約
  - ✓ Syslogに載せられるようにASCIIの文字列に変換
- 標準エラーコードへのマッピング
  - ✓ ごく少数の標準エラーコードを定義
  - ✓ エッジデバイス製造業者がプロプライエタリのエラーコードから標準エラコードへのマッピングテーブルを提供
- 標準エラーフォーマットの定義
  - ✓ リクエスタとリクエスト
  - ✓ レスポンダとエラーコード
  - ✓ レポータ
  - ✓ 標準エラーコード
  - ✓ その他の関連要素
- 要素の充足方法の確立
  - ✓ オペレーションの例示

エラーログフォーマットの要素

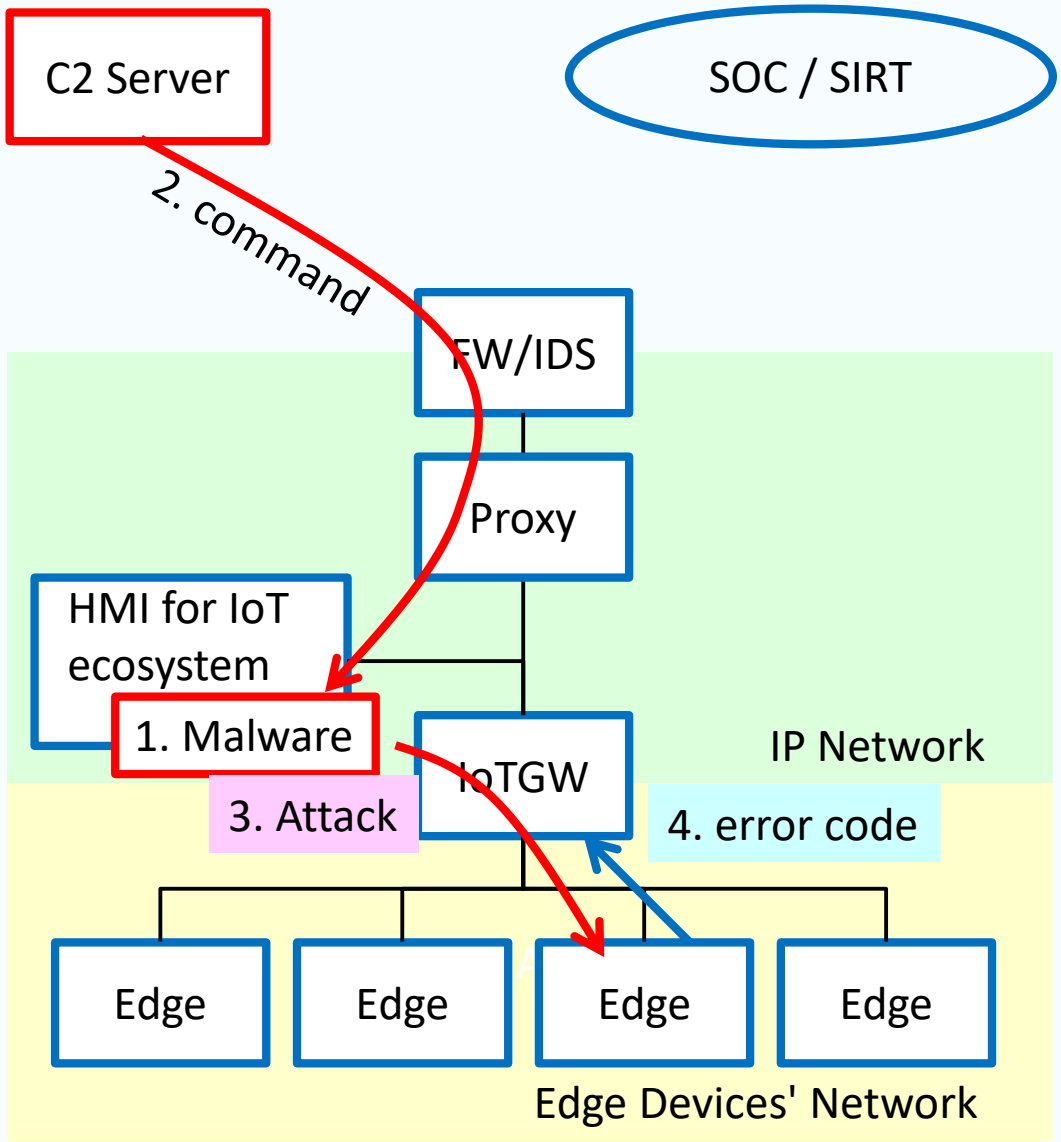
- リクエスタとリクエスト
  - リクエストの生データ
  - リクエストを送信したデバイスの識別子
- レスポンダとエラーコード
  - エラーコードの生データ
  - エラーコードを送信したデバイスの識別子
- レポータ
  - 標準エラーログフォーマットに従ってログを送信するデバイスの識別子
  - 多くの場合はリクエスタと同じ
  - IDSでも可
- 標準エラーコード
- その他の関連要素

Edge Deviceエラーの SOCへの流れ



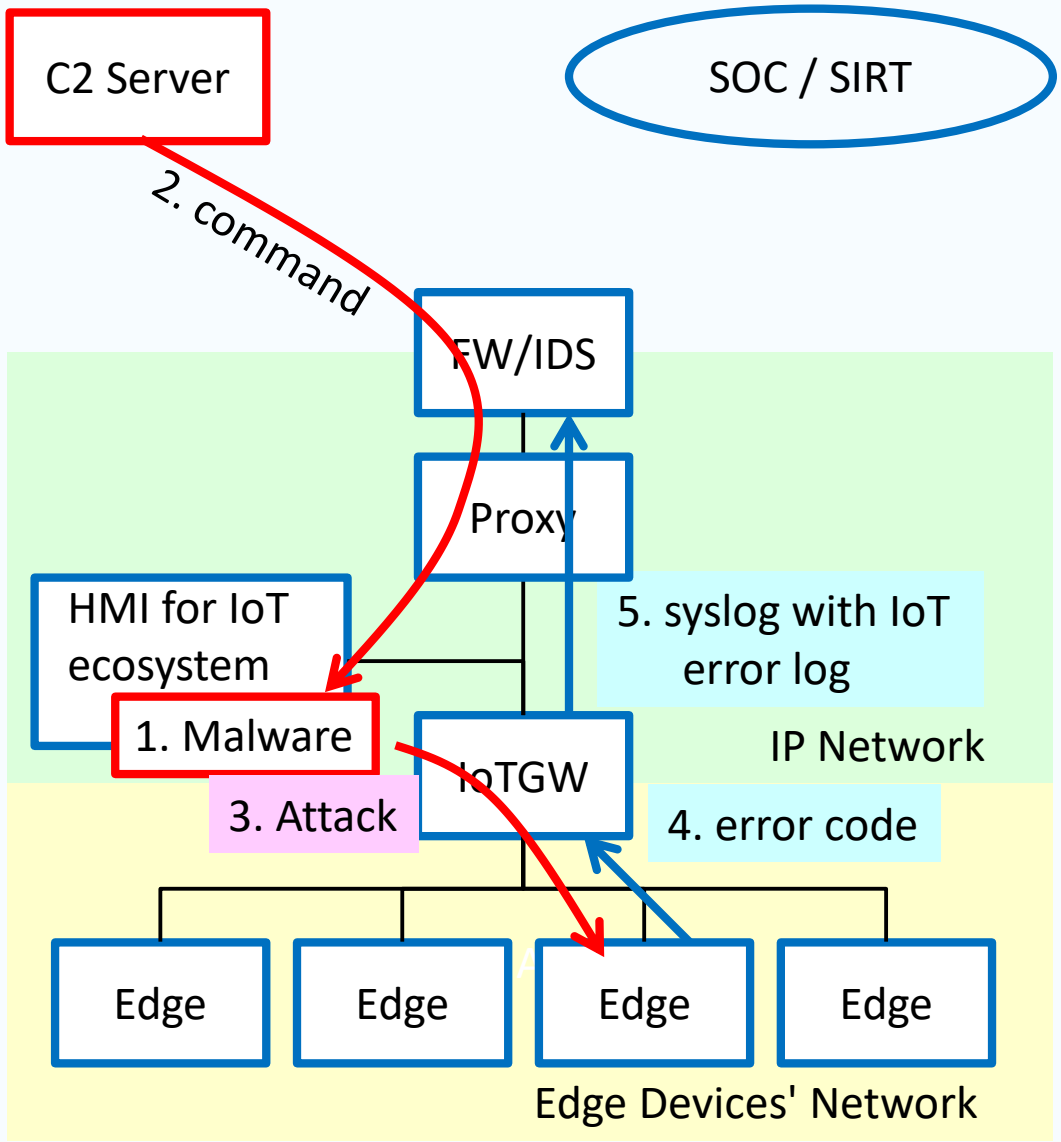
3. IoTGWがEdge Deviceにリクエストを送信する際、そのリクエストを一時的に保存

Edge Deviceエラーの SOCへの流れ



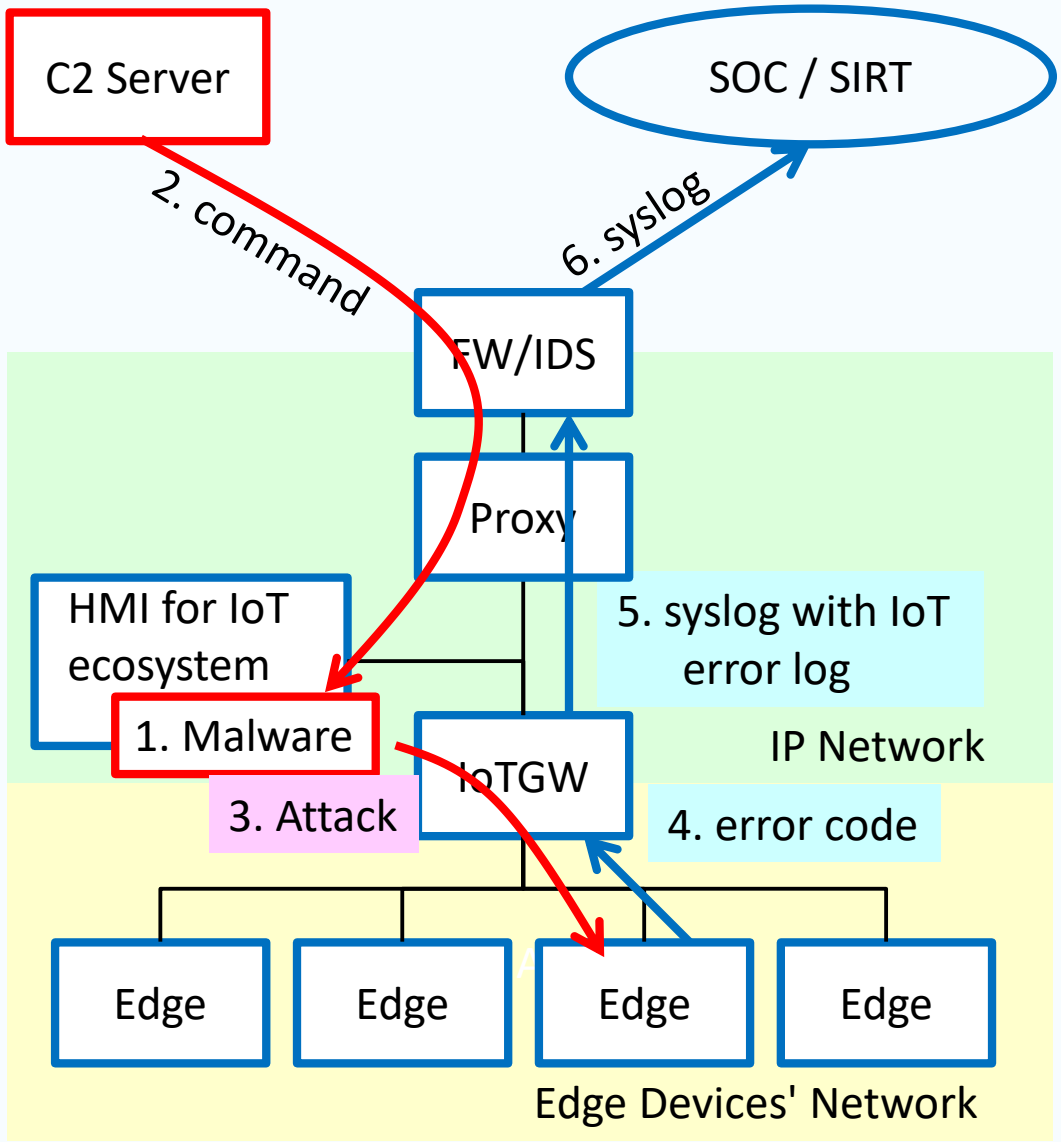
3. IoTGWがEdge Deviceにリクエストを送信する際、そのリクエストを一時的に保存
4. 得られたエラーコードから標準エラーコードへ変換、リクエスト、エラーコード、標準エラーコードでエラーログを構成

Edge Deviceエラーの SOCへの流れ



3. IoTGWがEdge Deviceにリクエストを送信する際、そのリクエストを一時的に保存
4. 得られたエラーコードから標準エラーコードへ変換、リクエスト、エラーコード、標準エラーコードでエラーログを構成
5. IoTGWがエラーログをsyslogでFWに転送

Edge Deviceエラーの SOCへの流れ



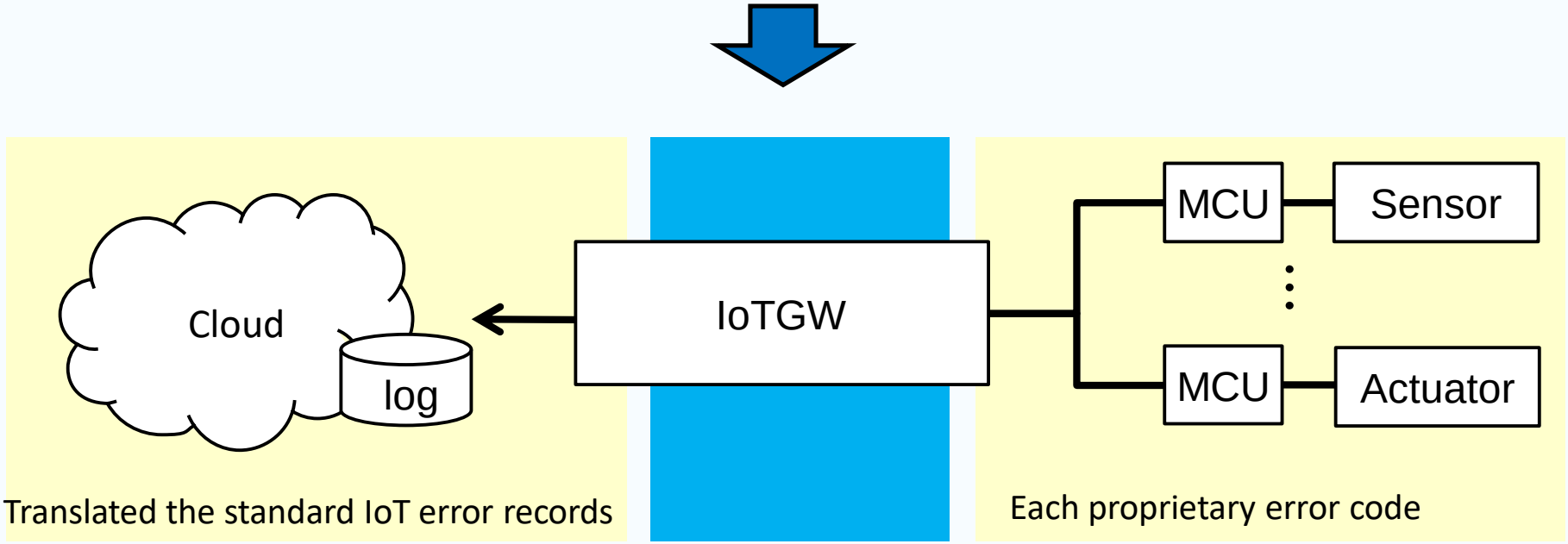
3. IoTGWがEdge Deviceにリクエストを送信する際、そのリクエストを一時的に保存
4. 得られたエラーコードから標準エラーコードへ変換、リクエスト、エラーコード、標準エラーコードでエラーログを構成
5. IoTGWがエラーログをsyslogでFWに転送
6. FWが他のログも併せてsyslogでSOC/SIRTに転送

アウトライン

- 問題の背景
- 想定される攻撃シナリオと対策
- ITU-T標準化の取り組み
- まとめ

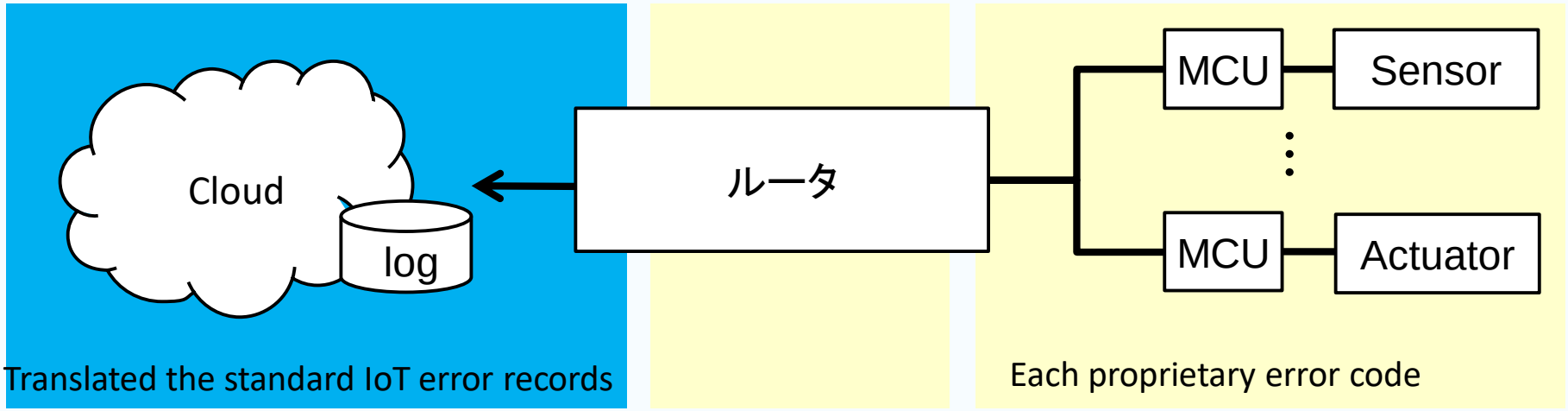
主張の範囲・パターン1

- IoTGWがリクエスタであり、プロプライエタリのエラーコードをIoTGWが受け取る場合
- IoTGWが標準エラーコードにマッピングし、標準エラーログフォーマットを用いてsyslogでsyslog serverに転送する。



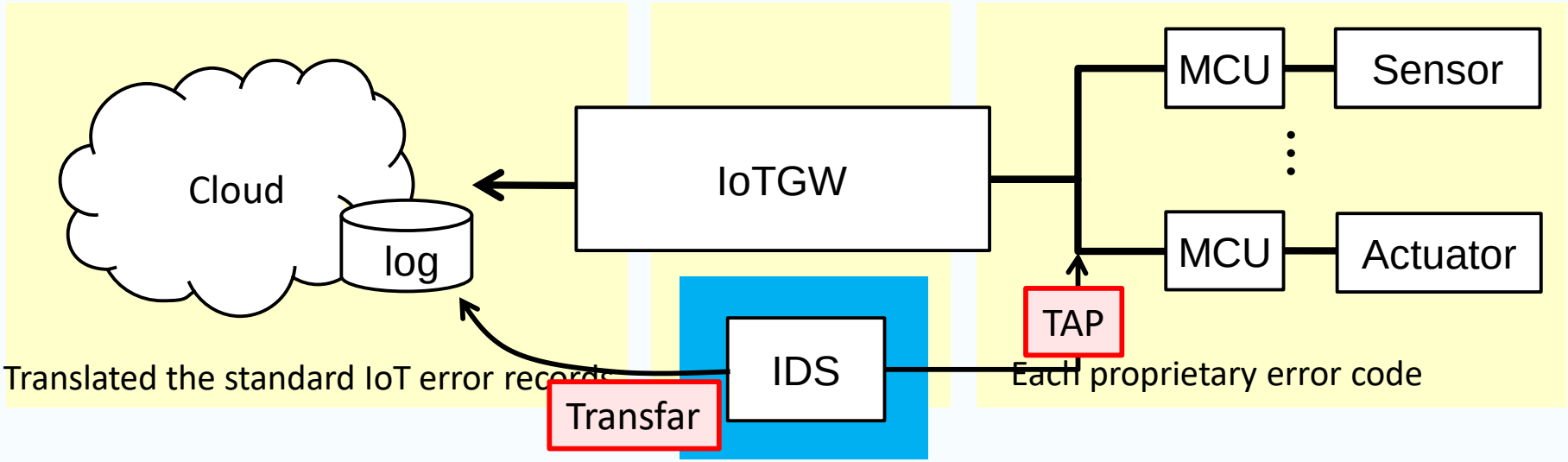
主張の範囲・パターン2

- クラウドがリクエスタであり、プロプライエタリのエラーコードをクラウド側が受け取る場合
- クラウド側で標準エラーコードにマッピングし、標準エラーログフォーマットを用いてsyslogで内部記録する。



主張の範囲・パターン3

- IDSがリクエスタ、リクエスト、レスポンダ、エラーを識別できる場合
- IDSが標準エラーコードにマッピングし、標準エラーログフォーマットを用いてsyslogでsyslog serverに転送する。



主張の範囲・手続き

1. IoTGWなどがエラーコードを収集する
2. エラーログに関する必要な情報の収集
   - 発生時刻
   - エッジデバイス間通信のプロトコル種類
   - リクエスタの識別子と要求コード
   - レスポンダの識別子とエラーコード
   - レポータの識別子
   - 標準化したエラーコード
   - 備考（必要であれば）
3. 上記の情報を標準化されたエラーログフォーマットに従ってエラーログを生成
4. エラーログをsyslog serverに転送

私たちの主張範囲

エラーログフォーマットの定義

現在はJSON形式で提案中

レポータは以下の情報を収集しエラーログを構成する

- 発生時刻
- レポータの識別子
- エッジデバイスのプロトコル
- リクエスタの識別子とリクエストの生データ
- レスポンダの識別子とエラーコードの生データ
- マッピングされた標準エラーコード
- 備考(Option)

```
{
 "Timestamp":
   "YYmmddTHHMMSS.sss...Z",
"Reporter": {},
"Protocol": String,
"Requester": {},
"Responder": {},
"Error Code": string,
"Description": {},
}
```

エラーコード表(1/2)

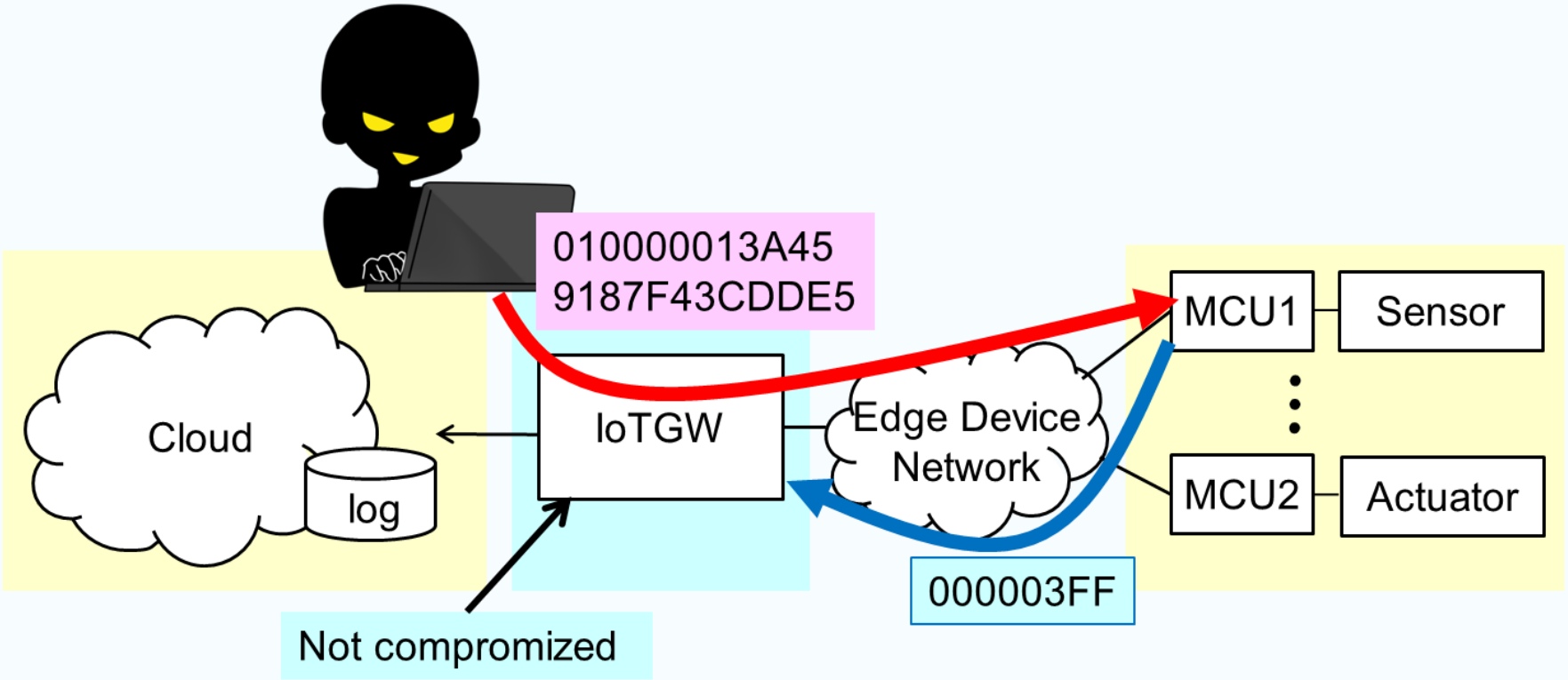
Code	Message	Description
	No Error (0x00-0x0F)	
0x00	No Error	No error occurs.
	Communication (0x10-0x1F)	
0x10	No Response	No response even though the request requires any responses.
0x11	Communication Failed	Some problems for failing communication.
0x12	Link Down	A network interface link is down.
0x1E	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.

エラーコード表(2/2)

Code	Message	Description
	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.

エラーログの構成例その１

状況：
攻撃者がエッジデバイスに誤ったリクエストを送り"Bad Request"のエラーが発生した。



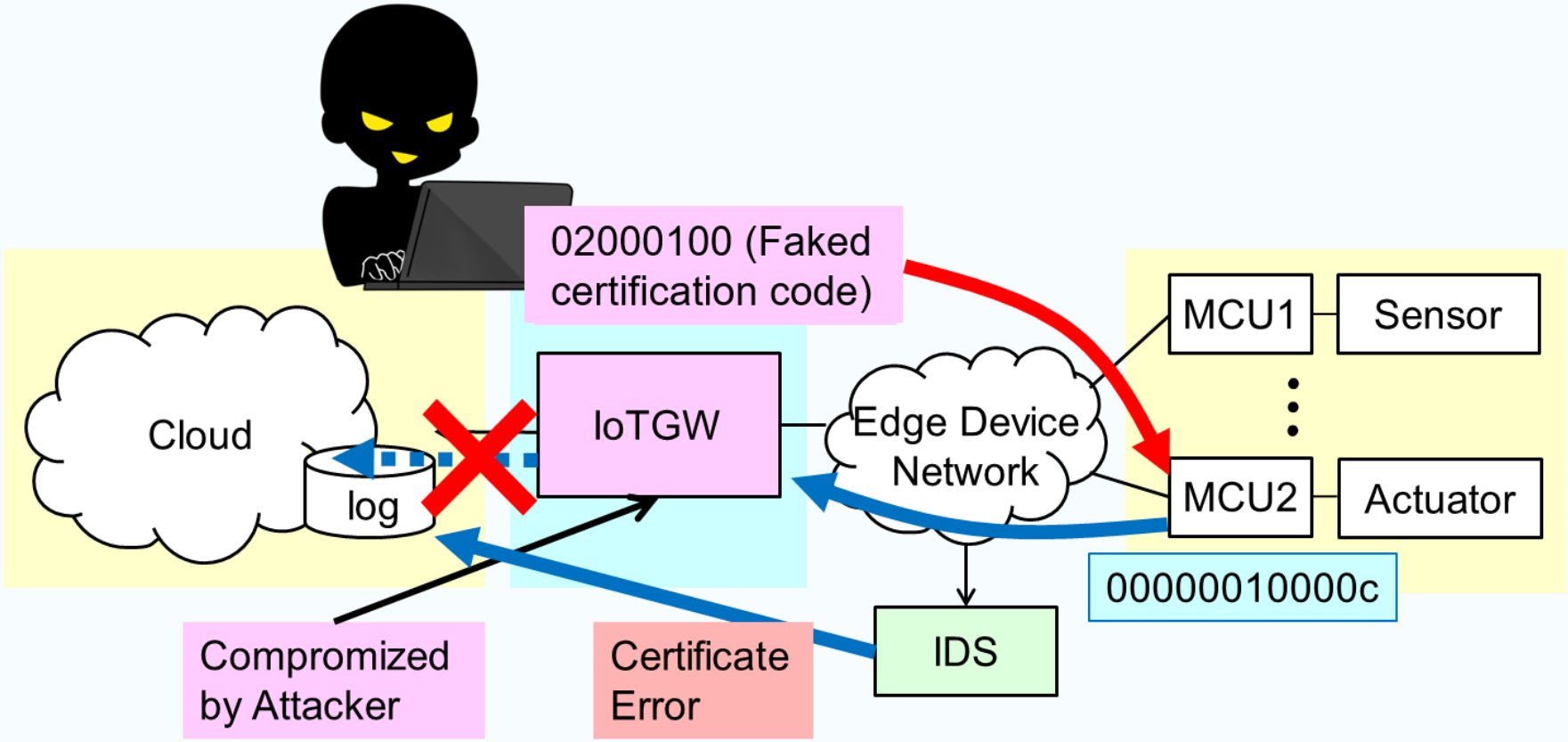
エラーログの構成例その１

状況：
攻撃者がエッジデバイスに誤ったリクエストを送り"Bad Request"のエラーが発生した。


```
{
   "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",
}
```

エラーログの構成例その2

状況：
攻撃者が先にIoTGWのファームウェア改ざんをした後、エッジデバイスに偽物の証明書を提示し、"Certification Error"のエラーが発生した。



エラーログの構成例その2

状況：
攻撃者が先にIoTGWのファームウェア改ざんをした後、エッジデバイスに偽物の証明書を提示し、"Certification Error"のエラーが発生した。


```
{
   "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”
   },
}
```

SOCの役割

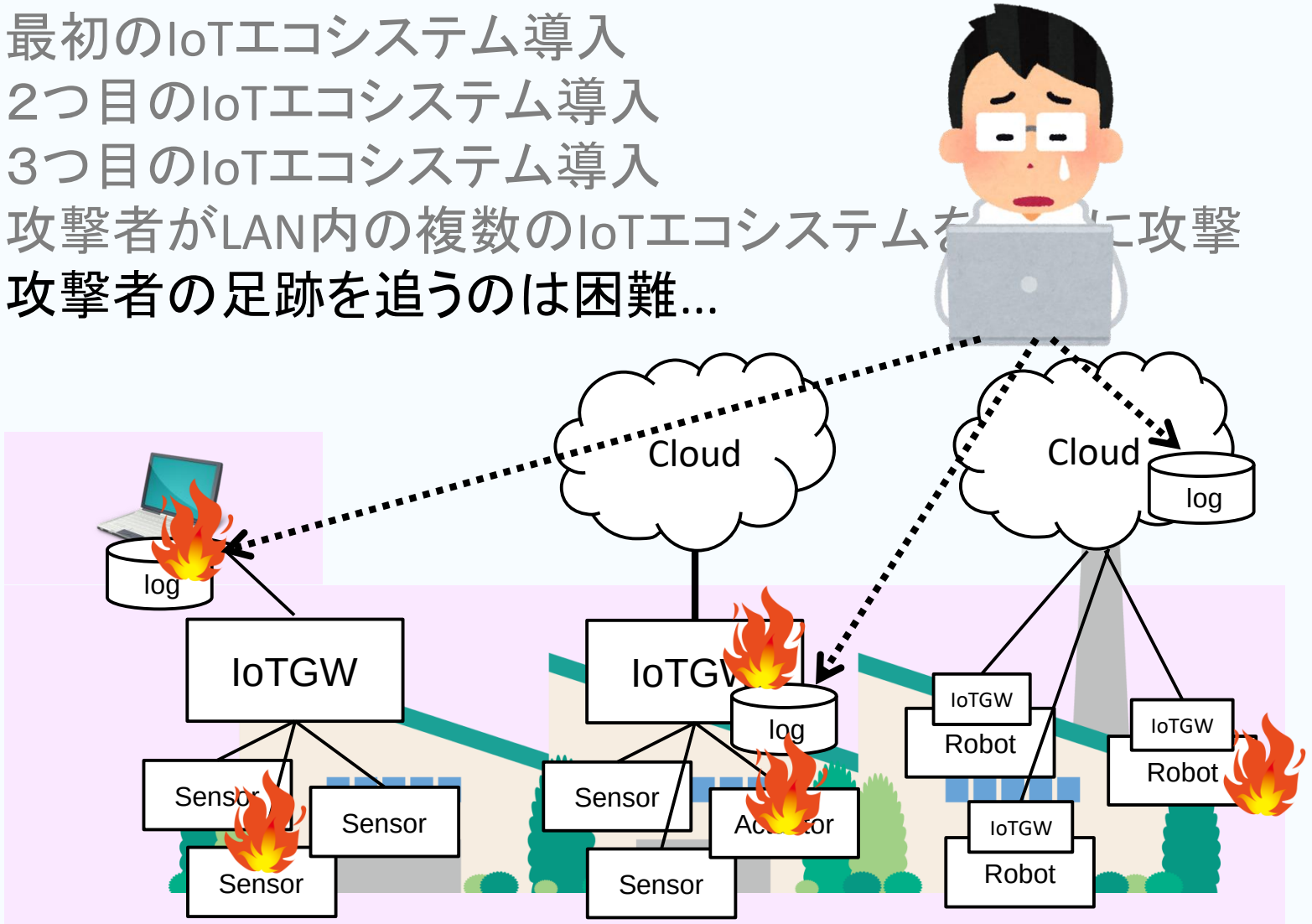
1. 標準エラーログフォーマットでエラーログを受信
2. 標準エラーコードから大雑把な問題を認識

   特定の条件下で以下を実施

3. レポータ、リクエスタ、レスポンダの情報から問題の詳細を把握
4. レポータ、リクエスタの識別子からその他の関連するsyslog情報等の抽出
5. 原因の特定
6. 回避のためのオペレーション
   ※故障の把握にも利用可能

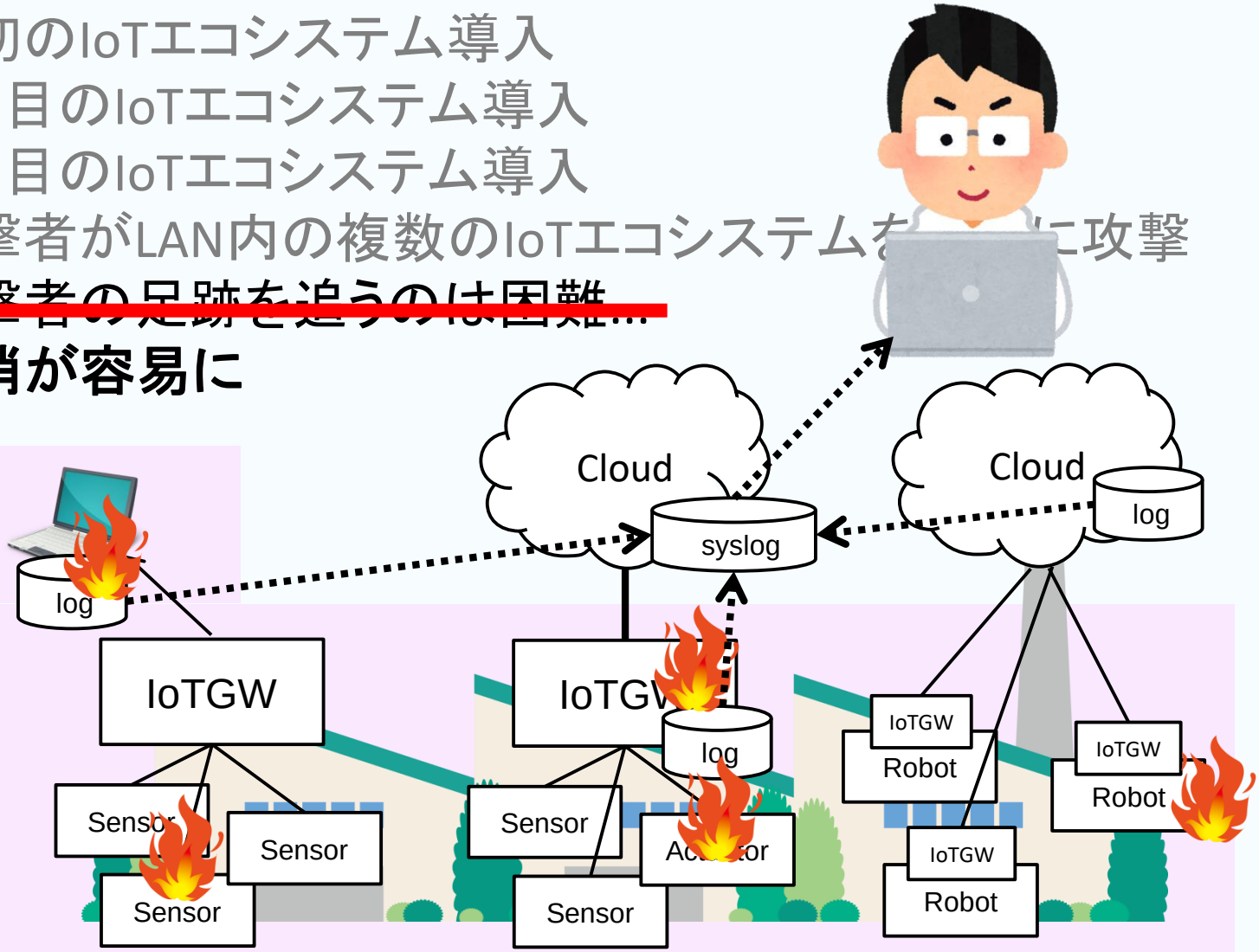
複数のIoTエコシステムが導入される過程

- 最初のIoTエコシステム導入
- 2つ目のIoTエコシステム導入
- 3つ目のIoTエコシステム導入
- 攻撃者がLAN内の複数のIoTエコシステムを[illegible]に攻撃
- 攻撃者の足跡を追うのは困難…

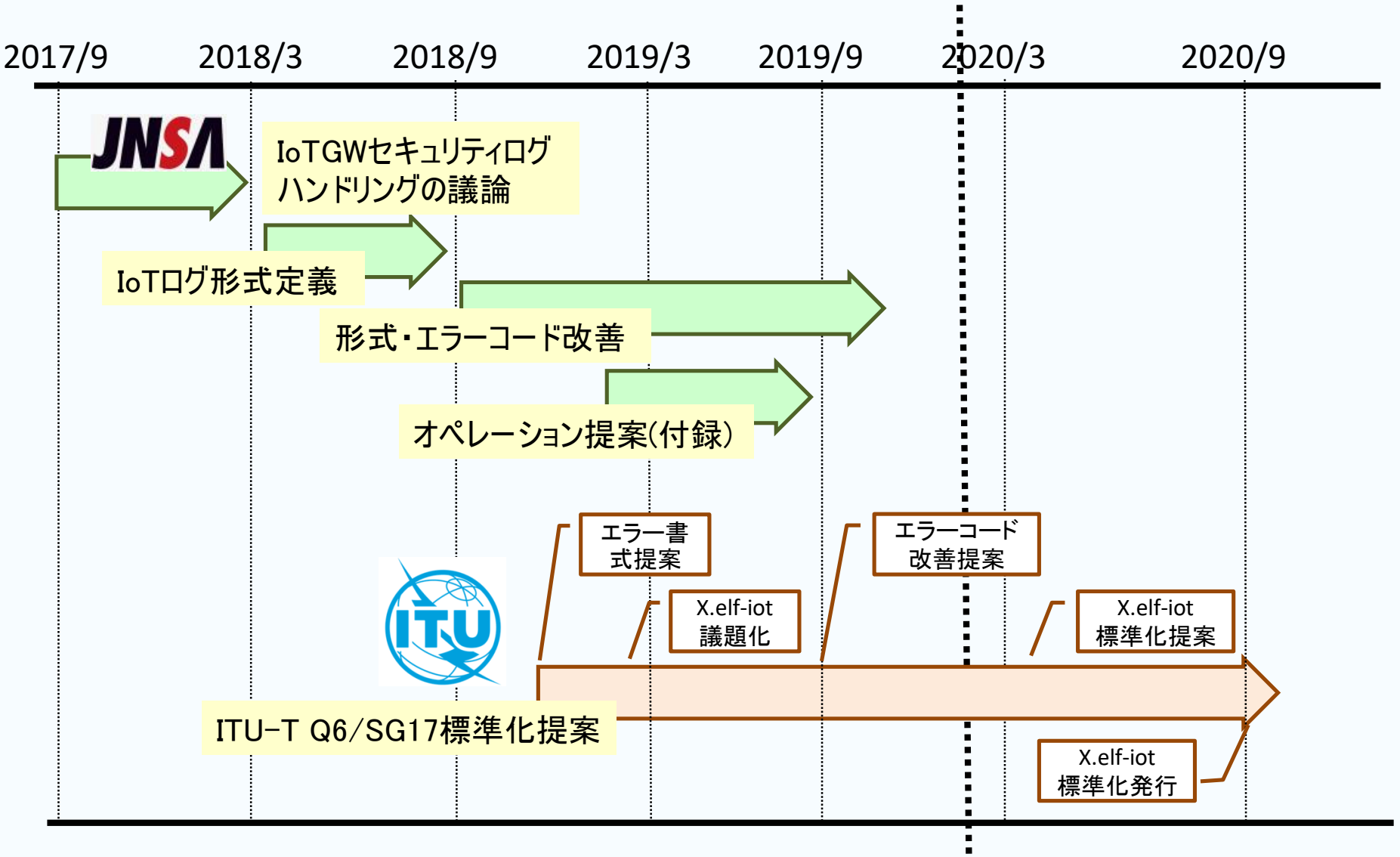


SOCの役割

- 最初のIoTエコシステム導入
- 2つ目のIoTエコシステム導入
- 3つ目のIoTエコシステム導入
- 攻撃者がLAN内の複数のIoTエコシステムを横断的に攻撃
- ~~攻撃者の足跡を追うのは困難…~~
- **火消が容易に**



スケジュール



アウトライン

- 問題の背景
- 想定される攻撃シナリオと対策
- ITU-T標準化の取り組み
- まとめ

まとめ

以下のことを提案している

- 個々のIoTエコシステムから出力されるエラーを標準エラーコードへマッピング
- 個々のIoTエコシステムから出力されるエラーを標準のエラーログフォーマットに合わせること
- 実装が必要になるのは、IoTGWなど少しリッチなリソースを持つデバイス

以下のようにオペレーションが変わる

- エラーログの収集が容易
- 標準エラーコードにより大雑把な問題の把握が容易
- IoTエコシステム以外のログも含めた突合が容易

皆様へのお願い

以下のような方を求めています。

- 実装にご協力いただける方
- 実装されたシステムをお試しいただける方

ご協力いただけると幸いです。

Thank you. Any Questions ?