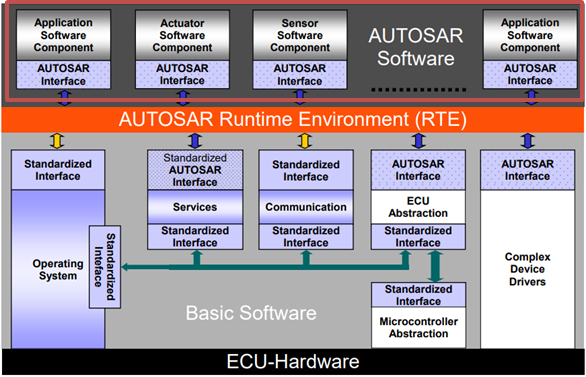
1. BSW 模块 基础软件层
2. CDD(Complex Device Driver or Complex Driver)是复杂设备驱动/复杂驱动的缩写
3. SWC Software [Component](https://so.csdn.net/so/search?q=Component&spm=1001.2101.3001.7020)（软件组件）一般位于AUTOSAR[架构](https://so.csdn.net/so/search?q=%E6%9E%B6%E6%9E%84&spm=1001.2101.3001.7020)的Application层只要了解过AUTOSAR的都知道，AUTOSAR分四层：**Application、RTE、BSW和MCAL**。
4. 
5. SWC 和 CDD 还可以报告在 AUTOSAR 中未标准化的⾃定义安全事件类型 自定义的这块没有BSW
6. 每个 BSW SWS 都列出了由相应模块报告的安全事件类型 没有CDD
7. IdsM缓冲报告的安全事件
8. Security Events (SEv)
9. Qualified Security Events (QSEv 合格安全事件)
10. Intrusion Detection System Reporter (IdsR ⼊侵检测系统报告器)
11. AUTOSAR 不提供 IdsR 的规范
12. IdsM 实例和 IdsR 之间的通信协议在 IdsM 协议规 范[6] 中指定。

传感器

SOC

多个ECU的上报

QSEv)

QSEv)

IdsR（接收多个ECU上报的信息）

与/或序列化QSEv

过滤器

Idsm

安全事件存储器(Sem) 合格安全事件

QSEv)

SEv

SEv

自主定义传输协义

1. 以下⽤例推动了对板载IDS 的要求。

• UC1：收集有关安全事件 (SEv) 的数据 ⽤例 ID 引⽤⾃以下要求。

• UC2：从安全事件数据中过滤合格的板载安全事件 (QSEv)

• UC3：本地存储 QSEv 记录

• UC4：通过SOC连接将QSEv 转发到 ECU

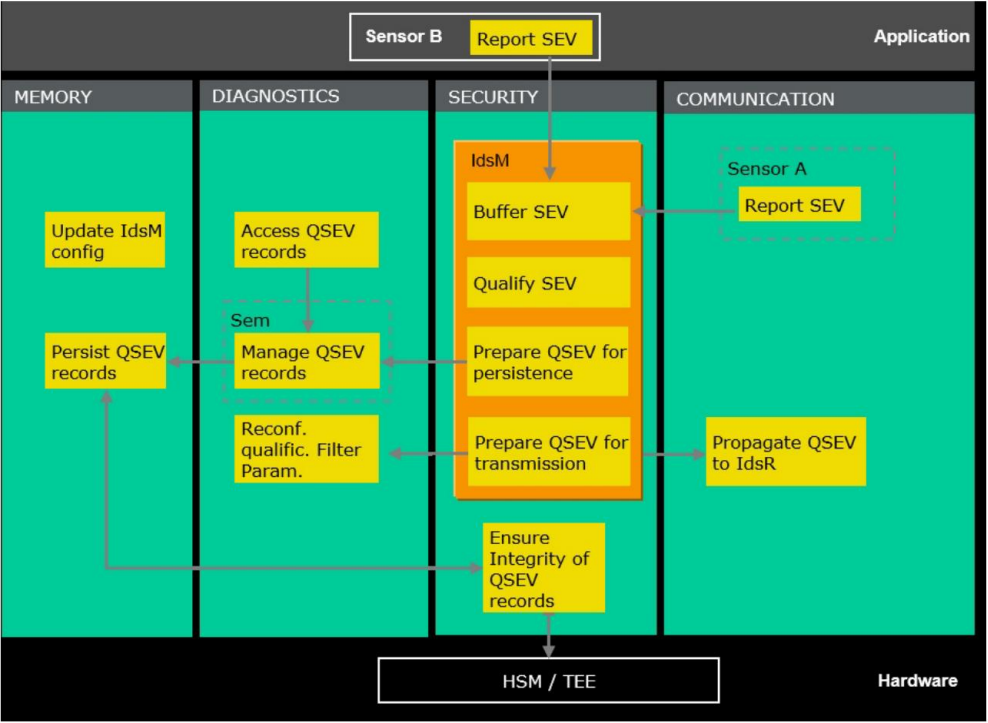
• UC5：提供对本地存储的 QSEv 记录的访问

• UC6：在操作期间重新配置资格参数

• UC7：更新 IdsM 配置 • UC8：保护 IdsM 配置

• UC8：保护 IdsM 配置

• UC9：保护传输中和静⽌



16.

IdsM 的初始化

提供⽤于报告 SEv 的接⼝

IdsM 应为调⽤者缓冲报告的 SEv

配置每个安全事件类型的报告模式和IdsM 实例

为合格的 SEv 提供可配置的过滤器链

提供多个过滤器链

⽀持机器状态过滤器

IdsM 应⽀持 SEv 的抽样

IdsM 应⽀持将多个 SEv 聚合成⼀个 QSEv

IdsM 应仅⽀持转发 SEv，如果它们发⽣得更频繁在可配置的时间间隔内超过可配置的阈值

IdsM 应提供向 SEv 添加时间戳的机制

IdsM 应提供机制让应⽤程序或传感器软件提供时间戳

IdsM 应允许将 QSEv 传输到 IdsR

IdsM 应该能够通过⽤⼾定义的诊断存储器在本地保存 QSEv。⽤⼾定义的诊断存储器应与主诊断存储 器分开，以允许对⽤于存储 QSEv 记录的 NVM 块进行单独的访问控制和保护

（这种⽤⼾定义的诊断 存储器也称为安全事件存储器）应保持QSEv的以下属性：

1. IDS 协议标头，指⽰协议版本和使⽤的协议选项
2. IdsM 实例的标识符
3. 传感器模块实例的标识符
4. 合格的 SEv 的标识符
5. 计数器，指示 SEv 在合格之前报告的频率
6. 时间戳（可选），指⽰ SEv 的时间点是合格的
7. 提供附加信息的上下⽂数据（可选），可由诊断测试仪或任何其他具有访问权限的安全分析实 例在 SOC 中进行评估。
8. 签名（可选），⽀持 IdsM 到SOC的完整性和真实性

将哪个 SEv 报告给 IdsM 应该是可配置的。

IdsM 将哪些资格过滤器应⽤于 SEv 应是可配置的，具体取决于项⽬特定的安全分析，需要应⽤不同 的资格过滤器来验证不同的 SEv 类型

如果 QSEv 应该在本地持久化，它应该是可配置的 根据项⽬特定的安全分

如果将 QSEv 传播到 IdsR，它应该是可配置的

⽀持在运行时重新配置报告模式·

IdsM 应允许替换完整的过滤器链配置

被认为与安全相关的基本软件模块应向 IdsM 报告安全事件。

⽀持检测 QSEv 记录d的操作

IdsM 应⽀持限制传输到 IdsR 的 QSEv 的速率和这些传输消耗的带宽

## 二．AUTOSAR\_RS\_SecurityExtractTemplate

1.安全提取的主要⽬的是将安全事件及其属性定义为⻋辆电⼦设备⼊侵检测系统的输⼊

2. 安全提取可用作交换文件，以在开发过程中以及在车辆已经在现场的维护过程中从多个来源收集安全事件及其属性。

3. 应该可以将与安全相关的事件和相关信息从传感器（实现为硬件或软件）传送到处理这些事件和相 关信息的软件模块。

4. 应该可以将安全事件与安全传感器的数字标识符相关联。

5. 应该可以配置使⽤附加上下⽂数据报告给定的安全事件

6. 应该可以为安全事件链接过滤器，以便多阶段可以实现由⼏种算法组成的过滤

7. 安全提取应⽀持安全限制过滤器的定义事件

8. 作为过滤限定的替代方案，（智能）传感器应能够报告预限定的安全事件，该事件应由 IdsM 直接作为合格的安全事件处理。

8.应该可以配置给定的 ECU 报告给定的安全事件

9. 应该可以配置为给定的通信总线报告给定的安全事件

10. 应该可以配置给定的应⽤程序报告给定的安全事件

11. 安全提取应⽀持给定安全事件的持久性配置。

12.安全提取应⽀持为安全事件配置具有不同详细级别的默认报告模式

13.当安全事件映射到 ECU 或机器时，安全提取应⽀持单独为安全事件定义严重性级别

14. 安全提取应⽀持⼀个或多个 IdsM 实例的可选定义和部分配置，与每个相应的 IdsM 实例 将在⾃适应平台上运行还是在经典平台上运行⽆关

15. 安全提取应⽀持配置 IdsM 实例是否为报告的安全事件提供时间戳信息，如果提供，该时间戳信息是否 为 AUTOSAR 标准化格式

16. 当向报告的安全事件添加时间戳信息时，安全提取应⽀持 IdsM 使⽤的配置格式。时间戳格式可能是 AUTOSAR 标准化的、由其他机构标准化的、公司标准化的或任意定义的

17. 对于 AUTOSAR 经典平台和⾃适应平台，安全提取应⽀持可选配置，即要求 IdsM 实例将⾝份验证信息 （即签名）添加到它发送到⽹络上的所有安全事件消息中。

18. 对于 AUTOSAR 经典平台和⾃适应平台，安全提取模板应描述如何定义 IdsM 实例与其⽹络配置的关 联

19.对于 AUTOSAR 经典平台和⾃适应平台，安全提取模板应描述如何定义 IdsM 实例与其⽹络配置的关 联。

20. IDS 设计通常涉及⻋辆内所有 ECU 的⼦集。该⼦集的每个 ECU 应能够报告单独定义和/或调 整的安全事件。因此，对于 IDS 的开发，Security Extract 需要能够定义属于 IDS 的所有系统部 分以及这些 IDS 系统部分的特定系统级功能。

21. 为了实现 IDS 的分布式开发，多个开发合作伙伴贡献了⼀个描述整个 IDS 设计的安 全提取⽂件。这些贡献者需要能够在 IDS 设计允许的情况下独⽴于其他⼈指定他们 的安全提取部分。

22. 处于系统 (M2) 级别的安全提取必须提供所需的信息，以获取与 ECU 的 IdsM 模块的安全事件相关的 配置参数（M1 级别）。因此，关于 AUTOSAR ⽅法，它同样⽤作诊断提取物或 ECU 提取物。

23. 推导相关ECU-C参数，安全提取应⽀持与 IdsM 模块的安全事件相关的 ECU-C 参数的派⽣。

24. 随着⼊侵检测系统管理器的引⼊，AUTOSAR 还将为现有的 BSW 模块（经典平台）或功能集群（⾃ 适应平台）提供标准化的安全事件。这些标准化的安全事件应使⽤安全提取模板指定，以启⽤⽂档⽣ 成的⾃动处理（单⼀来源原则）。

## 三．AUTOSAR\_SWS\_AdaptiveIntrusionDetectionSystemManager（讲解每个功能点应该如何去做）

### 7.1 功能集群⽣命周期

### 7.2 事件⽣成

### 7.3 上报⽅式

### 7.4 过滤链

#### 7.4.1 机器状态过滤器

如果 IdsM 评估机器状态过滤器并且当前机器状态等于 SecurityEventStateFilter.blockIfStateActiveAp 引用的状态之一，则 IdsM 将丢弃 SEv

#### 7.4.2 采样滤波器

实现通常会为每个 SecurityEventDefinition 维护一个计数器，当采样过滤器评估给定类型的 SEv 时，该计数器将递增。 如果计数器等于 n，则 SEv 不会被丢弃并且计数器重置为 0

IdsM应初始化 SEv的采样过滤器，以便转发每个SecurityEventDefinition接收到的第⼀个SEv, ⽰例： SecurityEventOneEveryNFil ter.n对于某个事件类型设置为 3，然后SEvs 1, 4, 7, ... 将被转发（1 描述重置后报告的第⼀个SEv ）。如果设置的4 那么SEvs 1，5，9

#### 7.4.3 聚合过滤器

1.置的时间间隔内发生的所有给定类型的 SEv 被聚合到一个 SEv 中，并附加一个额外的计数器信息，指示事件在该时间间隔内发生的频率。

2.商应将参数 SecurityEventAggregationFilter.aggregationIntervalLength 配置为应聚合给定类型的 SEv 的间隔的持续时间

3. 在聚合间隔期间，聚合过滤器不应转发（即，到下一个过滤器）任何传入的 SEv。

4. 如果在过去的聚合时间间隔内聚合过滤器没有接收到相同事件类型的 SEv，则不采取任何措施

5. 如果在过去的聚合间隔中聚合过滤器接收到一个或多个相同事件类型的 SEv，则应将 SEv 转发到链中的下一个过滤器

6. SEv 被转发到过滤器链中的下一个过滤器，则 SEv 的计数参数应等于过去时间间隔内聚合过滤器处理的给定事件类型的所有 SEv 的所有计数参数的总和（这个计数个数总和应该怎么计算）

7. SEv 被转发到过滤器链中的下一个过滤器，并且如果 SecurityEventAggregationFilter.contextDataSource 等于 IDSM\_FILTERS\_CTX\_USE\_FIRST，则上下文数据应等于在过去时间间隔内在聚合过滤器处接收到的给定类型的 SEv 的第一个上下文数据 .

8. 如果 SEv 被转发到过滤器链中的下一个过滤器，并且如果 SecurityEventAggregationFilter.contextDataSource 等于 IDSM\_FILTERS\_CTX\_USE\_LAST，则上下文数据应等于在过去时间间隔内在聚合过滤器处接收到的给定类型的 SEv 的最后一个上下文数据 .

9.SEv 被转发到过滤器链中的下一个过滤器，则时间戳应取自上下文数据来自的同一 SEv（通过 SecurityEventAggregationFilter.contextDataSource 配置）。

10. 请注意，如果 SecurityEventAggregationFilter.contextDataSource

等于 IDSM\_FILTERS\_CTX\_USE\_LAST，则报告或存储的 QSEv 将包含在配置的时间间隔内创建的最后一个 SEv 的上下文数据，但在配置的时间间隔内创建的第一个 SEv 的时间戳（时间戳还是要使用第一个SEv创建的么？）

#### 7.4.4 阈值过滤器

1. 如果阈值过滤器在当前阈值间隔内处理的所有给定类型的 SEv 的计数参数之和小于配置的参数 SecurityEventThresholdFilter.thresholdNumber，则阈值过滤器应丢弃给定类型的 SEv

2. 如果阈值过滤器在当前阈值间隔内处理的所有给定类型的 SEv 的计数参数之和等于或大于配置的参数 SecurityEventThresholdFilter.thresholdNumber，则阈值过滤器应转发给定类型的 SEv

#### 7.4.5资格

在 SEv 成功通过过滤器链的最后一个配置过滤器后，它被认为是 QSEv。 根据配置，QSEv 可以传输到 IdsR 和/或在本地保存。

### 7.5 时间戳

1. 时间戳是可选的

2. 堆栈提供的时间戳

3. 通过事件报告接⼝提供的时间戳

4. 通过应⽤软件接口提供的时间戳

5. 如果 IdsmInstance.timestampFormat 不等于“‘AUTOSAR”’，但在没有时间戳参数的情况下调用 ara::idsm::EventReporter::ReportEvent 函数并且没有注册 TimestampProvider，则 IdsM 不应向 QSEv 添加时间戳

6. 如果使用时间戳参数调用 ara::idsm::EventReporter::ReportEvent 函数，则 IdsM 应将该值截断 2 个最高有效位，即仅保留 62 个最低有效位以供进一步使用

7. TimestampProvider SWCL 应使用函数 ara::idsm::RegisterTimestampProvider 注册回调。 回调应返回时间戳

8. 请注意，虽然指定了 TimestampProvider API，但 TimestampProvider 的集成和配置仍然是特定于堆栈供应商的

### 7.6 QSEv 的传播

1. PlatformModuleEthernetEndpointConfiguration 在角色网络接口中的 IdsPlatformInstantiation 聚合，IdsM 应使用 [2] 中定义的 IDS 协议将 QSEvs 传输到通过 PlatformModuleEthernetEndpointConfiguration 配置的端点。

IdsM 应将 IDS 消息分离头的消息 ID 字段设置为全零（0x00000000）头的message id是0？

### 7.7 传输的 QSEv的真实性

1.idsM 可以选择使用加密签名保护传输的 QSEv 的真实性。

2.如果 IdsmSignatureSupportAp 在角色 signatureSupportAp 中的 IdsmInstance 处聚合，则 IdsM 应将加密签名附加到传输到 IdsR 的每个 QSEv 和每个本地持久的 QSEv。[2] 中规定了应根据哪些数据计算签名以及应如何将签名包含在传输到 IdsR 的消息中。 可以使用 IdsmSignatureSupportAp 模型元素配置应使用哪个签名原语和哪个密钥

3. IdsM 应使用在参数 IdsmSignatureSupportAp.cryptoPrimitive 中指定的签名算法和由角色 keySlot 中 IdsmSignatureSupportAp 引用的 CryptoKeySlot 标识的密钥

4. 要使⽤的签名算法的命名⽅案在 SWS Cryp tography [5] 中指定。

### 7.8 速率和流量限制

1.在将 QSEv 发送到 IdsR 之前，IdsM 应应用可能导致丢弃 QSEv 的速率和流量限制

2. 如果 IdsM 的传输会导致在 IdsmRateLimitation.timeInterval 中指定的当前间隔内传输的 QSEv 的数量超过配置为 IdsmRateLimitation.maxEventsInInterval 的最大传输数量，则 IdsM 应从传输中删除 QSEv。

3. 如果 IdsM 的传输会导致在 IdsmTrafficLimitation.timeInterval 中指定的当前间隔内传输的字节数超过配置为 IdsmTrafficLimitation.maxBytesInInterval 的最大字节数，则 IdsM 应从传输中删除 QSEv

4. IdsM 应将流程可以生成的事件类型限制为清单中角色安全事件中流程引用的那些安全事件定义

### 7.9 访问控制

1. 安全事件的产⽣受到访问控制，即可以通过配置来限制特定的SWCL可以产⽣哪些事件类型。访问控制由 IAM 在⾃适应平台上实施。

2. IdsM 应将流程可以生成的事件类型限制为清单中角色安全事件中流程引用的那些安全事件定义

3. TimestampProvider 接⼝也需要受到访问控制，以防⽌恶意或受损的应⽤程序向IdsM提供错误的时间戳为了⽀ 持项⽬特定的 TimestampProvider（例如，基于硬件或驱动程序），对 TimestampProvider 的访问控制是超出 了本规范的范围，并且必须以特定于项⽬的⽅式强制执行。

### 7.10 诊断访问

IdsM 允许诊断访问以支持两个用例：首先，可以通过诊断访问读取持久事件。 其次，可以通过诊断访问重新配置报告模式

#### 7.10.1 访问持久事件

每个安全事件都引用一个诊断事件，而该诊断事件又引用一个 DTC（诊断故障代码）。

如果⼀个事件已成功限定并且该事件被配置为持久（即， Securi tyEventContextProps.persistentStorage == 1），则IdsM应限定事件引⽤的DTC并添加事件数据作为它 的快照记录

#### 7.10.2 重新配置报告模式

IdsM标准化了⼀个DID ，⽤于在运行时读取和更改事件的报告模式。

1. dIdsM应提供诊断服务 GetReportingMode(SecurityEventDefinition.id)，返回查询的SecurityEventDefinition.c(RS\_ -Ids\_00700) 的当前报告模式
2. IdsM应提供诊断服务 SetReportingMode(SecurityEventDefinition.id, ReportingMode) 设置给定SecurityEventDefinition.c (RS\_Ids\_00700)的报告模式

### 7.11 IdsM 提供的 SEv

IdsM本⾝也可以⽤作安全事件传感器。

1. IdsM模块报告的安全事件列在 [SWS\_IdsM\_91015]
2. 请注意，对应于每个安全事件的十六进制值在 SecXT 集中定义
3. IdsM 应确保即使没有可用的缓冲区也可以处理 IdsM 内部事件。
4. IdsM 内部 SEv 不应通过速率和流量限制过滤器进行过滤

Idsm对外接口有哪些呢？

AUTOSAR 决定不对功能集群之间专⻔使⽤的接⼝（仅在平台级别）进行标准化，以允许有效的实现，这 可能取决于例如使⽤的操作系统。

## 四．AUTOSAR\_PRS\_IntrusionDetectionSystem（讲解收到的evet的格式和发出去到idsr的数据格式）

本协议要求规范定义了 AUTOSAR 协议⼊侵检测系统(IDS) 的格式、消息序列和语义。

### 5 协议规范

IDS 协议的主要⽬的是将合格的安全事件(QSEv)从⼊侵检测系统管理器(IdsM)实例传输到⼊ 侵检测系统报告器(IdsR)实例

#### 5.1 IDS消息格式

1.如果除了 IdsR 提供的时间戳之 外还需要更精确的时间戳

2. 传感器或IdsM可以为每个QSEv 添加时间戳。

3. 该选项必须通过协议头中的相应配置位设置

##### 5.1.1 IDS 协议概述

5.1.2 字节序字节顺

##### 5.1.6 上下⽂数据（可以配置）

###### 5.1.6.1 上下⽂数据 – 长尺⼨

###### 5.1.6.2 上下⽂数据 - 短尺⼨

##### 5.1.7 上下⽂数据⻓度编码

长度编码解释了上下文长尺寸和短尺寸的选择问题

##### 5.1.8 签名（可选的）

IDS协议提供了⼀个可选功能，使 QSEv的传输更安全。可以将数字签名添加到IDS 消息中。它可⽤于确保真实性以及证明签名的完整性，来⾃IdsM的消息通过所有通信系统直到到达后端或SOC (End2End-Security)

签名最长是18个字节

#### 5.1.9 IDS消息分离

1.在以太网上，IDS 消息分离头是强制性的。 它用于明确地处理 IDS 消息。 除了通过以太网传输单个 IDS 消息外，还可以在单个以太网帧中收集和发送多个 IDS 消息

2. ⼀个唯⼀的以太⽹端⼝地址应该⽤于IDS通信

3. SOME/IP 和 IDS 消息不应该混合在同一个端口上，因为它们不能被接收者正确区分。

###### 5.1.9.1 IDS 消息分离头

消息整体= ids消息分离头+ids消息

一个例子

Event Frame: 8 Bytes

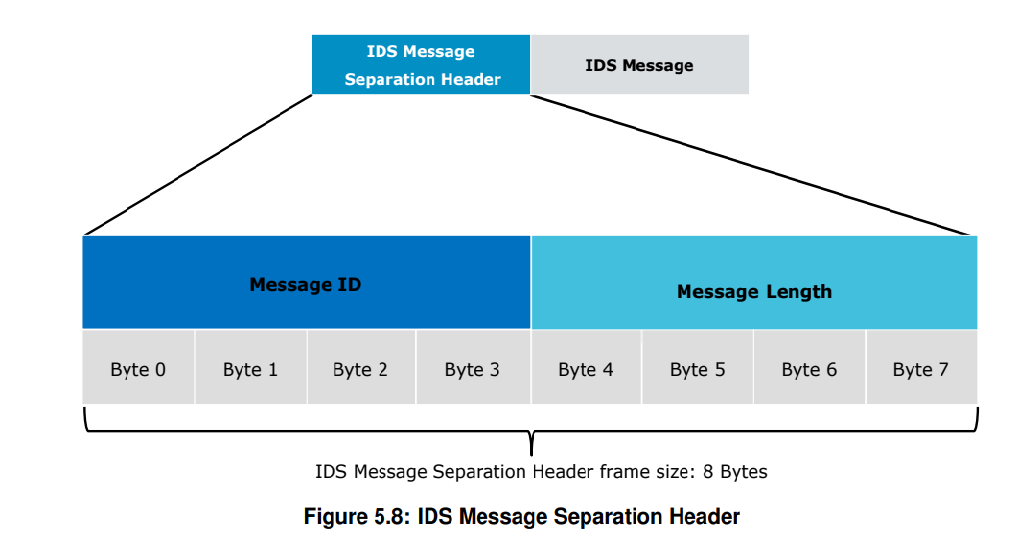
Timestamp: 8 Bytes

Context Data Size Long: 231-1 Bytes = 2.147.483.647 Bytes

Context Data Size Long Length Encoding: 4 Bytes

Signature: 65535 Bytes

Signature Length Encoding: 2 Bytes



5.5 错误信息

IDS 协议不发送特定的错误消息。

如果出现特定于 IdsM 的内部错误合格的安全事件被发送到配置的接收器

1. 安全事件缓冲区溢出：没有更多的事件缓冲区可⽤于处理事件。
2. 上下⽂数据缓冲区溢出：没有更多的上下⽂数据缓冲区可⽤于存储上下⽂数据。
3. 流量限制溢出：当前流量超过配置的限制。

## 五．AUTOSAR\_TPS\_SecurityExtractTemplate

AUTOSAR (M2 level) 或者(M1 level)是什么意思？

### 4 安全提取建模说明

1. SecurityEventFilterChain派⽣⾃IdsCommonElement并定义可应⽤于报告的SecurityEventDefinitions 的各种过滤器的适⽤性和属性
2. IdsmProperties派⽣⾃IdsCommonElement ，并提供⼀个容器，⽤于定义与IdsmInstances相关的功能 属性，可以 IdsDesign通过在⻆⾊元素中被引⽤。以下具体 对定义执行上下⽂的IdsmInstance的定义 它们与CommunicationConnector相关的发⽣上下⽂。 元类派⽣⾃SecurityEventContextMapping： ‒ SecurityEventContextMappingApplication将 SecurityEventDefinitions映射到定义执行的 IdsmInstance 通过各⾃的引⽤以可重⽤的⽅式应⽤。⼀个例⼦是IdsM 实例创建的⽹络带宽限制
3. • IdsmInstance派⽣⾃IdsCommonElement并指定⼀个实例 IdsMapping⽤作各种上下⽂相关映射的基类 它们在功能集群中发⽣的上下⽂。 ‒ SecurityEventContextMappingCommConnector映射Securi IdsM 及其系统级配置参数。

## 4.2 IdsDesign

**IdsDesign的语义**d元类IdsDesign表示一个结构容器，它通过链接在一起（通过角色元素中的所有相关安全性的引用）定义 IDS 设计和实现的范围（以及系统边界）提取元素

## 4.3 安全事件定义

SecurityEventDefinition表示具有预定义属性的安全相关事件的原子单元，由安全传感器报告并由 IdsM 进一步处理

### 4.3.1 安全事件的属性

## 4.4 过滤安全事件 SecurityEventFilterChain

这些条件检查按如下顺序执行：

1. 默认报告模式（见第4.6.1.2章）
2. 过滤器链（见第4.4.1章）
3. 限制过滤器（见第4.5章）
4. 安全事件聚合过滤
5. 安全事件OneEvery过滤器
6. 安全事件状态过滤器
7. 安全事件阈值过滤器

### 4.4.3 SecurityEventOneEveryNFilter

SecurityEventOneEveryNFilter的**语义**定义了功能“Forward Every Nth”的采样过滤器，其中 N 由属性n定义。每第n个安全事件通过此过滤器进一步向下过滤器链

# B 上游映射

本章描述了 ECU 配置参数（M1 模型）

2.3. 元模型体系

完整的AUTOSAR模板元模型体系共有五层：

M0层：AUTOSAR对象

这是对AUTOSAR系统的实现：真实的ECU执行包含雨刷控制软件的软件映像。

M1层：AUTOSAR模型

这一元层的模型是由AUTOSAR终端用户(汽车工程师)构建的。由他们定义名为“雨刷”的软件组件和一系列连接其它软件组件的接口等等。在这一层AUTOSAR系统被细分成可重用的组件和特定实例。

M2层：AUTOSAR元模型

这一层定义之后将被AUTOSAR终端用户使用的“词汇表”，比如，这层定义了在AUTOSAR中有名为“软件组件”的实体和另一个名为“端口”的实体。这些实体之间的联系和语义都属于整个模型的一部分。

M3：AUTOSAR模板的UML profile

M2层的模板是由M3层定义的元模型构建的。正如之前讨论过的，这是UML加上一个特定的UML profile，以更好的支持模板建模工作。严格意义上M2层上的模板仍然是UML的实例，但同时也采用了模板profile。

M4：元对象设施(meta object facility)

为了概念上的完整性，OMG将MOF放在最后一层元层上。因为MOF被定义为是反射式的，所以不再需要进一步的元层。

## 六．AUTOSAR\_TPS\_ManifestSpecification

获取安全事件的自定义时间戳

## 8.5 安全事件的报告

## 9.11 IdsM 部署

**9.11.1 IdsM 实例化**

**9.11.2 获取安全事件的自定义时间戳**

**9.11.3 安全事件部署**