From ECU to VSOC: UDS Security Monitoring Strategies

Ali Recai Yekta*, Nicolas Loza†, Jens Gramm†, Michael Peter Schneider†, Stefan Katzenbeisser‡
*Yekta IT GmbH, Dortmund, Germany
†ETAS GmbH, Stuttgart, Germany
‡University of Passau, Passau, Germany
EMails: ali@yekta-it.de, {nicolas.loza | jens.gramm | michaelpeter.schneider}@etas.com,
stefan.katzenbeisser@uni-passau.de

***Abstract*—Increasing complexity and connectivity of modern vehicles have heightened their vulnerability to cyberattacks. This paper addresses security challenges associated with the Unified Diagnostic Services (UDS) protocol, a critical communication framework for vehicle diagnostics in the automotive industry. We present security monitoring strategies for the UDS protocol that leverage in-vehicle logging and remote analysis through a Vehicle Security Operations Center (VSOC). Our approach involves specifying security event logging requirements, contextual data collection, and the development of detection strategies aimed at identifying UDS attack scenarios. By applying these strategies to a comprehensive taxonomy of UDS attack techniques, we demonstrate that our detection methods cover a wide range of potential attack vectors. Furthermore, we assess the adequacy of current AUTOSAR standardized security events in supporting UDS attack detection, identifying gaps in the current standard. This work enhances the understanding of vehicle security monitoring and provides an example for developing robust cybersecurity measures in automotive communication protocols.**

Keywords-Automotive Networks, Automotive Security, UDS, Security Monitoring, VSOC, UN R155, IDS.

I. Introduction

The growing complexity and interconnectivity of modern vehicles have created notable security challenges. Vehicles are increasingly susceptible to cyberattacks, which poses serious risks to both vehicle integrity and safety. This issue is tackled by the recent UN R155 regulation [1], which emphasizes the urgent requirement for strong cybersecurity management systems and protective measures. One essential layer of defense involves implementing effective security monitoring systems.

Cybersecurity challenges are particularly relevant for the Unified Diagnostic Services (UDS) protocol [2], which is the most commonly used diagnostic protocol in the automotive sector. UDS facilitates communication between vehicle Electronic Control Unit (ECU)s and diagnostic testers —- either external to the vehicle or vehicle-internal units. The services provided by UDS encompass a broad range of fundamental functionalities that the automotive industry utilizes throughout all phases of an ECU's lifecycle, including development, testing, operation, maintenance, and decommissioning. Consequently, these services are of significant interest to attackers, as they enable a high degree of control over the ECU. While the security of the UDS protocol has been explored in various studies [3]–[6], security monitoring for UDS has not been studied systematically before.

This paper presents security monitoring strategies for the UDS protocol, wherein detection is based on in-vehicle logging and on processing log events in a remote Vehicle Security Operations Center (VSOC) [7]. The VSOC collects security events from the vehicle fleet and puts them in context with other data sources, e.g., vehicle records including maintenance plans, and threat intelligence digests. More concretely, firstly, we present log strategies specifying which security events are to be logged in vehicles. Secondly, we describe context data to be logged with security events. Finally, we describe detection strategies to analyze logged vehicle security events, with the goal to detect UDS attack scenarios. Strategies are formulated for the specific case of UDS but have the potential to generalize to the security monitoring of vehicle security events in general.

We underline the relevance of the presented monitoring strategies by applying them to a comprehensive taxonomy of UDS attack techniques [8]. This taxonomy is based on Tactics, Techniques, and Procedures (TTP) and is structured along the automotive-specific 'Vehicle Adversarial Tactics, Techniques, and Expert Knowledge' (VATT&EK) [9] and the more general MITRE 'Adversarial Tactics, Techniques, and Common Knowledge' (ATT&CK) [10] frameworks. Our results show that presented detection strategies cover almost all of the attack techniques in this taxonomy. Among others. Moreover, our results show which attack techniques can be detected already on the vehicle side and which techniques require correlation of data sources in a fleet backend. We also show to which extent the security events standardized by a current industry standard, AUTOSAR, are already suited to support the detection of UDS attack techniques, and we identify corresponding gaps in the standard.

In summary, we give an overview on detection strategies for attack techniques misusing the UDS protocol. In this way, our approach gives an example for developing security monitoring strategies for an automotive communication protocol. While VSOC infrastructures have been established in recent years by vehicle manufacturers, it is still a challenge how to detect the occurrence of higher-level attack techniques based on low-level security events. The presented end-to-end monitoring strategies address this challenge. They can be used to implement UDS security monitoring, by deriving vehicle-side logging requirements and by guiding backend-side log processing in a VSOC.

The paper is structured as follows. Section II provides background and related work. Section III lays down the methodology used in this work and Section IV presents the

evaluation of the results. In Section V, we conclude our discussion and refer to possible future work.

II. Background and Related Work

Background. Cybersecurity attacks have become a highly relevant threat for modern cars. First standards and regulations on security have already been created in the automotive industry. Examples are the ISO/SAE 21434 standard on vehicle cybersecurity [11] and the United Nations (UN) R155 regulation [1] providing cybersecurity provisions for vehicle type approval. The latter requests automotive manufacturers to be able to detect and respond to security attacks in their vehicles. For this goal, automotive manufacturers introduce security monitoring solutions for their vehicle fleets.

UDS. In this work, we specifically consider security monitoring targeting to detect threat scenarios for the UDS protocol. The UDS protocol, standardized in [2], is the most widely used protocol for vehicle diagnostics. It allows diagnostic tools to contact the ECU installed in a vehicle which has UDS services enabled. Diagnostic services cover, among others, testing, calibration, or software updates. Table I provides an overview on UDS services. More details about the services can be found in [2].

UDS Security. There are a number of reported vehicle vulnerabilities based on UDS services, e.g., [12]–[14], which underlines the relevance to study security aspects of UDS. A focus of recent research on UDS security has been on implementation weaknesses of the UDS Security Access Service [15]–[17]. For first systematic evaluations of the attack surface of automotive diagnostic protocols, see [4][6].

TABLE I. UDS Services overview.

SID	Service	Short
0x10	DiagnosticSessionControl	DSC
0x11	ECUReset	ER
0x14	ClearDiagnosticInformation	CDTCI
0x19	ReadDTCInformation	RDTCI
0x22	ReadDataByIdentifier	RDBI
0x23	ReadMemoryByAddress	RMBA
0x24	ReadScalingDataByIdentifier	RSDBI
0x27	SecurityAccess	SA
0x28	CommunicationControl	CC
0x29	Authentication	AUTH
0x2A	ReadDataByPeriodicIdentifier	RDBPI
0x2C	DynamicallyDefineDataIdentifier	DDDID
0x2E	WriteDataByIdentifier	WDBI
0x2F	InputOutputControlByIdentifier	IOCBI
0x31	RoutineControl	RC
0x34	RequestDownload	RD
0x35	RequestUpload	RU
0x36	TransferData	TD
0x37	RequestTransferExit	RTE
0x38	RequestFileTransfer	RFT
0x3D	WriteMemoryByAddress	WMBA
0x3E	TesterPresent	TP
0x83	AccessTimingParameters	ATP
0x84	SecuredDataTransmission	SDT
0x85	ControlDTCSetting	CDTCS
0x86	ResponseOnEvent	ROE
0x87	LinkControl	LC

A comprehensive analysis of attack techniques for UDS has been provided in [8]. The derived taxonomy categorizes 53 UDS attack techniques along 9 tactics of known attack frameworks. Concretely, the used tactics are *Resource Development (RD)*, *Persistence (PS)*, *Privilege Escalation (PE)*, *Defense Evasion (DE)*, *Credential Access (CA)*, *Discovery (DS)*, *Lateral Movement (LM)*, *Collection (CL)*, and *Affect Vehicle Function (AF)*. The attack techniques are used in the evaluation of detection strategies in Section IV (Table III).

Security Monitoring. As part of security monitoring solutions, in-vehicle software sensors are used to monitor automotive systems for security anomalies. Also research has so far focused on these on-board Intrusion Detection Systems (IDSs), for an overview see [18]. Network IDS (NIDS) monitors in-vehicle networks, e.g., Controller Area Network (CAN) busses or Ethernet networks. Host IDS (HIDS) monitors in-vehicle electronic control units, e.g., on the operating system level or on their interfaces. The setup of VSOC, i.e., backend infrastructures for fleet security monitoring, has been described in [7][19][20].

AUTOSAR Security Events. AUTOSAR is a firmware specification that is widely used in the automotive industry. AUTOSAR supports a set of Security Events (SEvs) for different technologies [21], as well as modules to qualify SEvs [22] and distribute them on the network [23]. Within this work, we will compare our results with what has been standardized in AUTOSAR, to determine which functionality can be used off-the-shelf and where extensions are needed.

III. Methodology

In this section, a systematic strategy for UDS security monitoring is developed. First, in Section III-A, a set of logging strategies is defined that allows the generation of appropriate security-related logs in the vehicle components running UDS. Then, in Section III-B, we provide a context data strategy, specifiying context data to be captured with vehicle security logs. Finally, in Section III-C, we define *detection strategies* allowing to identify higher-level attack scenarios with high certainty. In general, detection can be executed both on the vehicle side as well as on the backend side in a VSOC. However, in many cases, detection relies on the VSOC receiving the data from the vehicle and validating it against information only available in offboard systems, in order to differentiate attacks from false positives.

A. Logging Strategies

This section defines the logging strategies that a vehicle and its subcomponents can implement to detect attacks on the UDS protocol. Due to constraints in the vehicle – runtime, storage, connectivity limitations – it is not possible to just record and send all data generated by the vehicle for analysis to a remote VSOC. Therefore, we need to rely on an appropriate logging concept, defining which events are to be logged. In the following, we present a set of three logging strategies.

a) Invalid Request (IR): Logs are generated whenever a UDS request is recognized as invalid due to one of the following reasons.

- A UDS request is observed which does not satisfy input validation checks due to unexpected formats, parameters out of range, or invalid payloads.
- A UDS request is observed under unexpected or non-permitted circumstances, at ECU or vehicle level, e.g., while the vehicle is driving at high speed or without required authorizations.

b) Function Execution (FE): Log the execution of selected SIDs, due to their criticality for the security of the ECU. This can be used by VSOC to validate if the operation makes sense in the context the vehicle is in. Examples are given by memory modifications or the execution of critical routines.

c) Message Flow Inconsistency (MFI): Logs are generated whenever a UDS SID is recognized as inconsistently routed due to one of the following reasons.

- A message is observed with unexpected source.
- A routed message is different from the original message.
- A routed message appears without first seeing the original message.
- Messages are observed in an unexpected sequence, e.g., multiple 0x27 seed requests are observed without a subsequent key response.

These logging strategies can then be activated or deactivated for each single UDS SID, according to the needs of the identified threats. Note that *Invalid Request* and *Function Execution* are both logging mechanisms that can be implemented by a HIDS or a NIDS, whereas implementing *Message Flow Inconsistency* is more feasible as part of a NIDS, since an overview of the different vehicle networks is needed.

B. Log Context Data Strategy

Whenever one of the previously introduced logging strategies is activated, it generates a log. In order to enrich a log, to make it more useful for further analysis, it must be complemented with appropriate *context data*.

For the strategy *Message Flow Inconsistency*, the context data strategy is always the same: the observed UDS SID, the targeted ECU, the observed request origin, and the expected request origin.

For the strategies *Invalid Request* and *Function Execution*, context data depend on the associated UDS SID. Table II specifies context data to be logged for these two logging strategies.

The column *AR support* indicates whether AUTOSAR already provides security events for this UDS service, based on the logging strategies outlined before. The AUTOSAR security events define context data that is very well aligned with the proposal from Table II. The only differences are that AUTOSAR does not provide hashes over data for SIDs WriteDataByIdentifier (WDBI (0x2E)) and WriteMemoryByAddress (WMBA (0x3D)), but it does provide the logical client source address for all UDS security events.

TABLE II. CONTEXT DATA TO BE LOGGED FOR EACH UDS SERVICE WITH STRATEGIES INVALID REQUESTS (IR) AND FUNCTION EXECUTION (FE).

SID	Context data to be logged for logging strategies **Invalid Requests** (1) and **Function Execution** (2)	**AR support**
0x10	$SID^{(1,2)}$, $SF^{(1,2)}$, $NRC^{(1)}$	-
0x11	$SID^{(1,2)}$, $SF^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x14	$SID^{(1,2)}$, $groupOfDTC^{(1,2)}$, $MemorySelection^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x19	$SID^{(1,2)}$, $SF^{(1,2)}$, $NRC^{(1)}$	-
0x22	$SID^{(1,2)}$, $DID1^{(1,2)}$, ..., $DIDn^{(1,2)}$, $NRC^{(1)}$	-
0x23	$SID^{(1,2)}$, $memAddr^{(1,2)}$, $memSize^{(1,2)}$, $NRC^{(1}$	-
0x24	$SID^{(1,2)}$, $DID^{(1,2)}$, $NRC^{(1)}$	-
0x27	$SID^{(1,2)}$, $SF^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x28	$SID^{(1,2)}$, $SF^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x29	$SID^{(1,2)}$, $SF^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x2A	$SID^{(1,2)}$, $transmissionMode^{(1,2)}$, $periodicDID\#1^{(1,2)}$, ..., $periodicDID\#n^{(1,2)}$, $NRC^{(1)}$	-
0x2C	$SID^{(1,2)}$, $SF^{(1,2)}$, $dynamicallyDefinedDID^{(1,2)}$, $sourceDID\#1^{(1,2)}$, ..., $sourceDID\#n^{(1,2)}$, $memAddr^{(1,2)}$, $memSize^{(1,2)}$, $NRC^{(1)}$	-
0x2E	$SID^{(1,2)}$, $DID^{(1,2)}$, hash over $dataRecord^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x2F	$SID^{(1,2)}$, $DID^{(1,2)}$, I/O $controlParameter^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x31	$SID^{(1,2)}$, $SF^{(1,2)}$, $RID^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x34	$SID^{(1,2)}$, $memAddr^{(1,2)}$, $memSize^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x35	$SID^{(1,2)}$, $memAddr^{(1,2)}$, $memSize^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x36	$SID^{(1)}$, $blockSequenceCounter^{(1)}$, $NRC^{(1)}$	-
0x37	$SID^{(1,2)}$, $NRC^{(1)}$, hash over transferred $data^{(2)}$	-
0x38	$SID^{(1,2)}$, $modeOfOperation^{(1,2)}$, $filePathAndName^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x3D	$SID^{(1,2)}$, $memAddr^{(1,2)}$, $memSize^{(1,2)}$, $NRC^{(1)}$, hash over transferred $data^{(2)}$	IR, FE
0x3E	n/a	-
0x84	$SID^{(1,2)}$, $Apar^{(1,2)}$, Signature/Encryption $Calculation^{(1,2)}$, req. $SID^{(1,2)}$, $NRC^{(1)}$	-
0x85	$SID^{(1,2)}$, $SF^{(1,2)}$, $NRC^{(1)}$	IR, FE
0x86	$SID^{(1,2)}$, $SF^{(1,2)}$, SID for $response^{(1,2)}$, $NRC^{(1)}$	-
0x87	$SID^{(1,2)}$, $SF^{(1,2)}$, $NRC^{(1)}$	-

SID = service ID, SF = subfunction, NRC = negative response code, DID = data identifier, other context data fields refer to parameters defined in [2].

The proposed context data from Table IIcombines data from the UDS request and response and provides hence the security-relevant information in a compact form. Using the raw UDS requests/responses as context data is not recommended due to (1) possibly large messages (up to several hundred bytes e.g. for Authentication (Auth29 (0x29)), TransferData (TD (0x36)) or WMBA (0x3D)) which could exhaust the resources of deeply embedded ECUs, (2) risk of information disclosure when sending UDS payload data in clear text to the VSOC and (3) separate SEvs for UDS requests and responses, which would need to be mapped in the VSOC and would prohibit the configuration of IR SEvs without FE SEvs.

Note that the presented logging strategies together with the context data strategy described in this subsection can generate a lot of false positives if applied indiscriminately, e.g., when activating *Function Execution* for ReadDataByIdentifier (RDBI (0x22)) without any additional conditions. Therefore, on top of the logging strategies, additional detection strategies must be defined, to differentiate between true attacks and false positives.

C. Detection Strategies

This section defines *detection strategies* allowing to identify higher-level attack scenarios. Detection strategies are needed for two reasons. Firstly, many of the logs proposed in Section III-A will also be generated under regular vehicle operations. Advanced checks and validations are needed to avoid false positive alerts. Secondly, there are attack scenarios which cannot be detected by vehicle-side logs alone. We introduce three detection strategies as follows.

a) Suspicious Log Patterns (SLP): This detection strategy monitors for the occurrence of suspicious patterns in logs collected in the vehicle. They refer to failed, rejected or inconsistent UDS operations in the vehicle. This strategy includes pattern matching rules with counting. Counting is required to implement checks against thresholds, since, during regular vehicle operation, occasional failed UDS operations are to be expected. Therefore, for each SID service, a threshold defines how many failed operations are to be observed within a time interval before an alert is triggered. Detection of this category can be executed on the vehicle side.

b) Contextualized Log Checks (CLC): This detection strategy assesses the (successful or failed) execution of UDS services in context of additional information. Context information includes the vehicle state, vehicle records with maintenance and service plans, as well as summaries of preceding and succeeding logs. Vehicle records are usually maintained in a backend but not in a single vehicle. Concrete checks to be executed as part of this strategy are given as follows:

- Service calls are inconsistent with the vehicle status, e.g., workshop session, development/production mode.
- Service call uses unexpected permissions.
- Service call is inconsistent with vehicle configurations.
- Service call is inconsistent with other logs, also from backend systems.
- In a service call, memory hashes do not match hashes of authentic software releases.
- In a service call, DIDs or memory ranges rated as sensitive are referenced, e.g., when files or memory are to be read out or modified.

Detection of this category can be executed on the vehicle side only if required context data is available, otherwise it needs to be done in the backend.

c) Product Threat Intelligence (PTI): This detection strategy uses threat intelligence information about the vehicle and its components to identify attack patterns. Sources for this can span from publicly available information, e.g., entries in public vulnerability databases, forums, or research papers, to confidentially disclosed information. Examples for the latter are supplier vulnerability disclosures, responsible vulnerability disclosures by white-hat-hackers, or internal penetration tests. For concrete cases, tags can be defined, including vehicle model, ECU type and attack patterns, to filter information feeds and to link them to concrete attack techniques. Alerts are then triggered whenever, based on this filtering, relevant information is identified.

In the implementation of detection strategies a)-c), a baseline of rules and their configuration is initially derived from the service specification of a vehicle model, and is finetuned based on evaluating false positive logs collected from test vehicles.

IV. Evaluation

This section evaluates the effectiveness of the logging and detection strategies presented in Section III. To this end, we applied the logging and detection strategies to a comprehensive taxonomy of attack techniques [8]. For each attack technique of this taxonomy, we evaluated which strategies can be applied to detect the respective attack technique. The resulting mapping table is presented in Table III. The table lists all attack techniques of this taxonomy, with their ID, name and affected UDS SIDs. Attack techniques are grouped by attack tactics. Columns "Logging Strategies" and "Detection Strategies" specify which strategies from Section III can be used to detect an occurrence of the respective attack technique. Moreover, column "AUTOSAR support" indicates that logging requirements of strategies IR and FE are already covered by the current AUTOSAR standardization.

Our evaluation focuses on three major topics. In Section IV-A, we focus on the logging aspects and compare our proposed logging strategies with the AUTOSAR-provided security events to identify gaps that need to be addressed in implementation projects. In Section IV-B, we discuss how to actually detect UDS attacks based on illustrative examples. Finally, in Section IV-C we draw conclusions and formulate take-away messages based on our analysis.

A. AUTOSAR Logging Coverage

Efficient intrusion detection is relying on standardized logging strategies that are available off-the-shelf and hence easy to deploy and use. AUTOSAR lends itself as a basis for such an approach, due to its good acceptance in the automotive domain and native support for security events.

As shown in Table II and discussed in Section III-B, AUTOSAR defines Security Events for 50% of the UDS services (13 of 26). The coverage analysis for the UDS attacks shown in Table III is a bit more complex, since AUTOSAR does not provide support for all SIDs and can hence log certain attacks only partially. Out of the 53 attacks, AUTOSAR supports full logging for 20 and partial logging for an additional 10 attacks, rendering the overall logging support to 38-56%.

While AUTOSAR provides a good basis for UDS attack logging, it fails at providing complete coverage. It is hence advised to introduce additional security events based on the context data proposal in Table II. This can be done by automotive manufacturers for their respective products, or directly in AUTOSAR by extending the available Security Events.

In addition, please note that AUTOSAR supports only the logging strategies IR and FE. MFI is not supported by AUTOSAR, since it is typically implemented as part of an NIDS. Automotive manufacturers should take care that their NIDS specification supports the MFI security event proposed in Section III-B.

TABLE III. UDS ATTACK TECHNIQUES AND THEIR DETECTION STRATEGIES.

Attack ID	Attack Name	SIDs	Logging Strategies	AUTOSAR Support	Detection Strategies
AT-RD-1	Firmware Reverse-Engineering	-	NA	No	PTI
AT-RD-2	Leak Secrets		NA	No	PTI
AT-PS-1	Download Custom Package	0x34, 0x36, 0x37	IR, FE	Only 0x34	SLP, CLC, PTI
AT-PE-1	Change to Privileged Session	0x10	FE, MFI	No	CLC
AT-PE-2	Valid Credentials	0x27, 0x29	FE	✓	CLC, PTI
AT-PE-3	Replay Attack SA	0x27	IR, FE, MFI	✓	SLP, CLC, PTI
AT-PE-4	Brute-Force SA	0x27	IR, FE	✓	SLP, CLC
AT-PE-5	Weak Auth29 configurations	0x29	IR, FE	✓	CLC
AT-DE-1	Block DTCs Generation	0x85	FE	✓	CLC
AT-DE-2	Remove Attack Traces in DTCs	0x14	FE	✓	CLC
AT-DE-3	Replay Download	0x34, 0x36, 0x37	FE	Only 0x34	CLC
AT-DE-4	Bypass Checks	Multiple	Various	No	CLC, PTI
AT-DE-5	Bypass Read Protections using DDDID	0x2C, 0x22	FE	No	CLC, PTI
AT-CA-1	Extract Secrets	0x22, 0x23, 0x31	FE	Only 0x31	CLC
AT-DS-1	Service Discovery	Multiple	IR, FE	(✓)	SLP, CLC
AT-DS-2	Subfunction Discovery	Multiple	IR, FE	(✓)	SLP, CLC
AT-DS-3	Diagnostic Sessions Discovery	0x10	IR, FE	No	SLP, CLC
AT-DS-4	UDS Fuzzing	Multiple	IR, FE	(✓)	SLP, CLC
AT-DS-5	Check seed entropy in SA	0x27	IR, MFI	No	SLP
AT-DS-6	Reverse-engineer SA algorithm	0x27	FE	✓	CLC, PTI
AT-DS-7	Identify Auth29 configuration	0x29	FE	No	CLC, PTI
AT-DS-8	Enumerate algorithms, Auth29	0x29	FE	No	CLC, PTI
AT-DS-9	Check challenge entropy, Auth29	0x29	IR, MFI	No	SLP
AT-DS-10	Identify Configurations for SDT	0x84	FE	No	CLC, PTI
AT-DS-11	DID Enumeration	0x22	IR, FE	No	CLC, SLP
AT-DS-12	Routine Enumeration	0x31	IR, FE	✓	CLC, SLP
AT-DS-13	File System Discovery	0x38	IR, FE	✓	CLC, SLP
AT-DS-14	Eavesdropping	Multiple	NA	No	NA
AT-LM-1	Man-in-the-Middle	Multiple	IR, FE, MFI	(✓)	SLP, CLC, PTI
AT-CL-1	Event-Based Data Extraction	0x86	IR, FE	No	SLP, CLC
AT-CL-2	Periodic Data Extraction	0x2A	IR, FE	No	SLP, CLC
AT-CL-3	DID Data Extraction	0x22	IR, FE	No	CLC
AT-CL-4	Memory Extraction	0x23, 0x35	IR, FE	Only 0x35	CLC
AT-CL-5	File Extraction	0x38	IR, FE	✓	CLC
AT-CL-6	Read DTCs	0x19	IR, FE	No	SLP, CLC
AT-AF-1	Request Flooding	Multiple	IR, FE	(✓)	SLP, CLC, PTI
AT-AF-2	Request Blocking	Multiple	IR, FE, MFI	(✓)	PTI, SLP
AT-AF-3	Interrupt Operations, DSC	0x10	IR, FE, MFI	No	SLP
AT-AF-4	Impede Usage of SA	0x27	IR	✓	SLP
AT-AF-5.1	Resource Overload via ROE	0x86	IR, FE	No	SLP, CLC
AT-AF-5.2	Resource Overload via RDBPI	0x2A	IR, FE, MFI	No	SLP, CLC
AT-AF-6	Interrupt Periodic Data Readout	0x2A	IR, FE	No	SLP, CLC
AT-AF-7	Change IO Configuration	0x2F	IR, FE, MFI	✓	SLP, CLC
AT-AF-8	Routine Misuse	0x31	FE	✓	CLC
AT-AF-9	Early Transfer Termination	0x37	IR, FE, MFI	No	SLP, CLC
AT-AF-10	Interrupt Routine	0x31	IR, FE, MFI	✓	SLP, CLC
AT-AF-11	Keep Session Open	0x10, 0x3E	FE, MFI	No	CLC
AT-AF-12	I/O Control	0x2F	IR, FE	✓	CLC
AT-AF-13	Disrupt ECU Communication	0x28	IR, FE, MFI	✓	CLC
AT-AF-14	Reset ECU	0x11	IR, FE, MFI	✓	SLP, CLC
AT-AF-15	DID Manipulation	0x2E	IR, FE	✓	SLP, CLC
AT-AF-16	File Manipulation	0x38	IR, FE	✓	SLP, CLC
AT-AF-17	Memory Manipulation	0x3D, 0x34	IR, FE	✓	SLP, CLC

— Attack IDs refer to UDS attack techniques derived in [**Anonymous2025uds**], where IDs have the format AT-<TT>-<NO> where <TT> refers to the attack tactic and <NO> to the number of the attack technique in the respective category.
— (✓) refers to logging for supported SIDs only.

B. UDS attack detection - examples

Detection of UDS attacks is very individual and strongly depending on the actual attack technique. Space restrictions do not allow to describe detection for every attack technique in detail. Instead, we illustrate the detection capabilities of our approach through three representative attack techniques, each demonstrating different aspects of our multi-layered security monitoring approach. Figure 1 shows the general detection process, highlighting the detection possibilities in the vehicle and in the VSOC, while locating the detection of the following examples.

(1) AT-PE-4 Brute-Force SA Attack: In this attack technique, an attacker tries to brute-force all possible response ("key") values for SA (0x27). Applying logging strategy IR, Security Access brute-force attacks can be detected using existing AUTOSAR security events for SID 0x27, namely this is AUTOSAR security event 103 (SEV_UDS_SECURITY_ACCESS_ FAILED) [21]. By application of detection strategy SLP, multiple occurrences of this event within a short timeframe indicate a brute-force attempt against the Security Access service. This demonstrates effective detection using established AUTOSAR events with simple rate-based analysis. Additionally, detection strategy CLC may identify when authorizations are not consistent with the vehicle status.

(2) AT-CL-3 DID Data Extraction: In this attack technique, an attacker uses UDS service RDBI (0x22) to extract the information stored behind the DIDs, which may contain confidential data, e.g., keys. Detection of unauthorized RDBI operations is not possible through existing AUTOSAR security events, as no events are specified for this service. By application of logging strategies IR (logging unsuccessful access attempts) and FE (logging successful access), security events can address this gap by logging all accesses to sensitive DIDs, e.g., accessing cryptographic material. Context-data strategies ensure that DIDs are available as context data, and, for unsuccessful access attempts, the reason for rejection is available as Negative Response Code (NRC). Using strategy CLC, it can be ensured that only critical data identifier access attempts are captured, enabling detection of attacks targeting sensitive ECU information.

(3) AT-PS-1 Download Custom Package: In this attack technique, an attacker uses UDS services RD (0x34, request download), TD (0x36, transfer data), and RTE (0x37, request transfer exit) to download their own data into the ECU. Detection is possible by using logging strategies FE and IR, logging successful and unsuccessful invocation of relevant services (0x34, 0x36, 0x37). Context-data strategies ensure that firmware hashes are captured when completing download operations (0x37). By application of logging strategy CLC, these hashes are transmitted from the vehicle to the VSOC, where they are correlated with authorized firmware databases to detect downgrade attacks and unauthorized firmware installations. Logging strategies SLP and PTI can additionally be used to raise reliability of the detection, e.g., by detecting failed attempts in the operation, or by looking for known exploit patterns to install firmware. This attack cannot be detected by AUTOSAR security events alone, due to two fundamental limitations:

1) Attack detection requires firmware hash validation, which is not included in standard AUTOSAR security events.
2) Determining whether older or modified firmware is being installed requires backend knowledge of authorized firmware versions, which cannot be maintained locally in each vehicle.

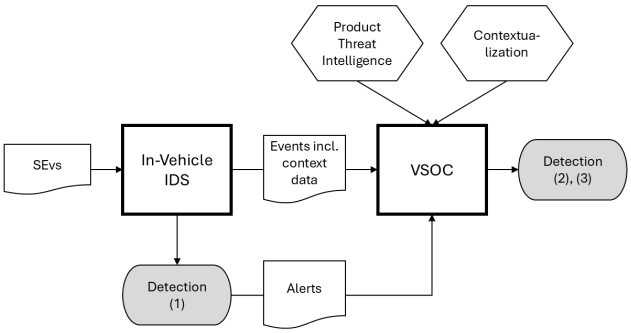

Figure 1. Detection process, including in-vehicle detection and VSOC-based detection. The numbers refer to the examples from Section IV-B

C. UDS attack detection - take-aways

Based on our analysis from Section IV-B, we can compile the following take-away messages for detecting UDS attacks:

Vehicle-side detection can only cover a subset of UDS attack techniques. Some attack techniques can be reliably detected on the vehicle-side. Examples are given by techniques of the Discovery tactic, e.g., service discovery or UDS Fuzzing, which can be detected by observing a large number of certain requests in a short time window. However, for the majority of attacks, the additional information and contextualization possibilities of a VSOC are needed for reliable detection, as described by the following two points.

Product Threat Intelligence is needed as part of a VSOC infrastructure. For attack techniques of the attack tactic Resource Development, detection is possible using strategy PTI (Product Threat Intelligence) alone. Reverse engineering of firmware and leakage of secrets is usually done offline and can neither be detected by sensors in the vehicle nor by consistency analysis of logs in the backend. It can only be detected by observing reports of leakage of ECU firmware or UDS cryptographic material, e.g., in forums or news feeds.

A *combination* of detection strategies as well as backend processing in a VSOC are needed for a maximum coverage and reliable detection of UDS attack techniques. For many attack techniques, single detection strategies alone cannot provide sufficient evidence on the occurrence of an attack technique. However, the combination of detection strategies allows to reach a higher confidence by elimination of false

positives. For example, consider AT-PE-1 "Change to Privileged Session" - an attacker using DSC (0x10) to change to a privileged session. In this case, the vehicle-side can log that DSC was called but needs additional data to distinguish whether this happened in context of a valid scenario, e.g., in context of a planned car service session.

V. Conclusion and Future Work

This paper presents multi-layered detection strategies for UDS-based attack techniques — combining vehicle-level intrusion detection sensors with VSOC-level processing and threat intelligence. It is shown that strategies are suited to cover almost all elements from a comprehensive taxonomy of UDS techniques. Security monitoring strategies presented in this paper can be used as a guide to implement the detection of UDS attack techniques in a VSOC infrastructure:

Logging requirements. Logging and context data strategies can be used as requirements for on-board intrusion detection components. The analysis from Table III also shows in which cases we can refer to AUTOSAR standardized security events.

Automated processing rules. Detection strategies of the Suspicious Log Pattern and Contextualized Log Check categories can be used to define automated processing rules in a processing pipeline for aggregated onboard logs. Depending on system architecture, resources, and availability of context data, log processing may be done on the onboard side as well as on the backend side. Automated processing results in alerts to be handled in an incident management system.

Threat intelligence triggers. Detection strategies of the Product Threat Intelligence category can be used to define trigger criteria for the evaluation of threat intelligence information. Depending on the trigger critria, news feeds will be filtered down towards notifications relevant for the UDS monitoring use cases, and can be linked to alerts.

Playbooks. On a higher level, detection scenarios can be implemented in playbooks, guiding the validation of alerts in an incident management system, also including manual analysis steps. Each UDS security attack technique can be covered by a playbook, while alerts with similar processing steps can be bundled in a joint playbook.

In this way, this paper gives concrete guidelines on building VSOC detection scenarios based on the UDS protocol, and our accepted follow-up describes a VSOC for automotive and rail, specifying formats for vehicle security events and alerts, as well as detection and response capabilities [24].

While this paper gives a qualitative assessment of detection strategies, their experimental evaluation with real vehicles remains a topic for future work.

Acknowledgment

This research is accomplished within the project "FINESSE" (FKZ 16KIS1584K). We acknowledge the financial support for the project by the Federal Ministry of Research, Technology and Space (BMFTR).

References

[1] UNECE, "UN Regulation No. 155 - Cyber security and cyber security management system," United Nations Economic Commission for Europe (UNECE), Geneva, CH, Standard UN R155, 2021. [Online]. Available: https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security (visited on 09/05/2025).

[2] ISO, "Road Vehicles — Unified Diagnostic Services (UDS) Part 1: Application Layer," International Organization for Standardization, Geneva, CH, Standard ISO 14229:2020, 2020. [Online]. Available: https://www.iso.org/standard/72439.html.

[3] S. Kulandaivel, S. Jain, J. Guajardo, and V. Sekar, "CANdid: A stealthy stepping-stone attack to bypass authentication on ECUs," *Journal on Autonomous Transportation Systems*, pp. 1–17, 2024.

[4] T. Lauser and C. Krauß, "Formal security analysis of vehicle diagnostic protocols," in *Proceedings of the 18th International Conference on Availability, Reliability and Security (ARES)*, ACM, 2023, pp. 1–11. DOI: 10.1145/3600160.3600184.

[5] M. Matsubayashi *et al.*, "Attacks against UDS on DoIP by exploiting diagnostic communications and their countermeasures," in *Proceedings of the 93rd IEEE Vehicular Technology Conference (VTC2021-Spring)*, IEEE, 2021, pp. 1–6.

[6] N. Weiss, S. Renner, J. Mottok, and V. Matoušek, "Automated threat evaluation of automotive diagnostic protocols," in *ESCAR USA*, 2021, pp. 1–16.

[7] K. Mayer, T. Volkersdorfer, J. Hofbauer, P. Heinl, and H.-J. Hof, "Vehicle security operations center for cooperative, connected and automated mobility," in *Proceedings of the 18th International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2024)*, IARIA, 2024, pp. 156–164.

[8] A. R. Yekta, N. Loza, J. Gramm, M. P. Schneider, and S. Katzenbeisser, "Uds attack taxonomy: Systematic classification of vehicle diagnostic threats," in *2025 IEEE Conference on Communications and Network Security (CNS)*, 2025, pp. 1–8. DOI: 10.1109/CNS66487.2025.11195020.

[9] A. R. Yekta, D. Spychalski, E. Yekta, C. Yekta, and S. Katzenbeisser, "VATT&EK: Formalization of cyber attacks on intelligent transport systems - a TTP based approach for automotive and rail," in *Proceedings of the 7th ACM Computer Science in Cars Symposium (CSCS)*, ACM, 2023, pp. 1–17. [Online]. Available: https://doi.org/10.1145/3631204.3631867.

[10] T. M. Corporation, *MITRE ATT&CK*, 2024. [Online]. Available: https://attack.mitre.org/ (visited on 09/05/2025).

[11] ISO/SAE, "Road Vehicles — Cybersecurity Engineering," International Organization for Standardization, Geneva, CH, Standard ISO/SAE 21434:2021, 2021. [Online]. Available: https://www.iso.org/standard/70918.html (visited on 09/05/2025).

[12] ASRG, *CVE-2024-6348 - Predictable seed generation in the security access mechanism of UDS in the Blind Spot Protection Sensor ECU in Nissan Altima.* 2024. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2024-6348 (visited on 09/05/2025).

[13] N. Ronge, S. Zari, and A. Yadav, *KIA-SELTOS-Vehicle-Cluster-Vulnerabilities*, 2024. [Online]. Available: https://github.com/nitinronge91/KIA-SELTOS-Cluster-Vulnerabilities (visited on 09/05/2025).

[14] P. Pełechaty and Ł. Konieczny, "Analysis of security vulnerabilities in vehicle on-board diagnostic systems," *Diagnostyka*, vol. 25, no. 3, pp. 1–8, 2024. DOI: 10.29354/diag/192162.

[15] J. Dürrwang, J. Braun, M. Rumez, and R. Kriesten, "Security evaluation of an airbag ECU by reusing threat modeling artefacts," in *2017 International Conference on Computational*

Science and Computational Intelligence (CSCI), IEEE, 2017, pp. 37–43.

[16] J. Van den Herrewegen and F. D. Garcia, "Beneath the bonnet: A breakdown of diagnostic security," in *Proceedings of the 23rd European Symposium on Research in Computer Security (ESORICS 2018)*, Springer, 2018, pp. 305–324. DOI: 10.1007/978-3-319-99073-6_15.

[17] M. Ring, T. Rensen, and R. Kriesten, "Evaluation of vehicle diagnostics security–implementation of a reproducible security access," in *Proceedings of the 8th International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2014)*, IARIA, 2014, pp. 214–219.

[18] B. Lampe and W. Meng, "Intrusion detection in the automotive domain: A comprehensive review," *IEEE Communications Surveys Tutorials*, vol. 25, no. 4, pp. 2356–2426, 2023.

[19] F. Langer, F. Schüppel, and L. Stahlbock, "Establishing an automotive cyber defense center," in *Proceedings of the 17^{th} Embedded Security in Cars Conference (ESCAR Europe 2019)*, 2019, pp. 1–14. DOI: 10.13154/294-6652.

[20] D. Grimm, M. Zink, M. Schindewolf, and E. Sax, "Cyber situational awareness in vehicle security operations: Holistic monitoring and a data model," in *Proceedings of the 20th International Conference on Network and Service Management (CNSM)*, IEEE, 2024, pp. 1–7.

[21] AUTOSAR, "Technical Report on Security Events Specification," AUTOSAR R24-11, 2024.

[22] AUTOSAR, "Specification of Intrusion Detection System Manager," AUTOSAR R24-11, 2024.

[23] AUTOSAR, "Specification of Intrusion Detection System Protocol," AUTOSAR R24-11, 2024.

[24] A. R. Yekta *et al.*, "Towards a holistic and multi-modal vehicle security monitoring," in *Proceedings of the 20th International Conference on Critical Information Infrastructures Security (CRITIS 2025)*, Jönköping, Sweden: Springer, 2025.