車両 SOC における現実の課題とその解決策

千葉 靖伸 [1,*]

概要：車両に対するサイバー攻撃の監視は，車両への攻撃への対策として最も重要な手段の一つである．企業や組織の IT 環境に対するサイバー攻撃の監視は広く実践され，確立されているが，車両の監視には特有の課題があり，依然として困難な領域である．これらの特有の課題については，その可能性や概要が共有されているが，具体的な現実の課題とその解決策は広く知られていない．本稿では，車両に対するサイバー攻撃を監視するセキュリティオペレーションセンター（車両 SOC）の試行と運用を通じて認識した IT 環境と車両の監視の主な差異と具体的な技術的課題について議論する．加えて，多数の車両を適切なコストで高精度に監視するための実践的な方法に焦点を当て，これらの課題の具体的な解決策とその効果を示す．

キーワード：セキュリティオペレーションセンタ，車両 SOC，SIEM

Real-World Challenges in Vehicle SOC and Their Solutions

Yasunobu Chiba[1,*]

Abstract: Monitoring cyber-attacks on vehicles is one of the most critical measures against such attacks. While monitoring cyber-attacks in enterprise Information Technology (IT) environments is widely practiced and well-established, monitoring cyber-attacks on vehicles has unique challenges and remains a challenging area. Although the possibility and overview of these unique challenges are shared within the community, specific real-world issues and their solutions have not been widely disseminated. This paper discusses the major and specific differences between monitoring enterprise IT environments and vehicles, as recognized through our trials and operations of a Security Operations Center (SOC) monitoring cyber-attacks on vehicles. We also discuss concrete solutions to the technical challenges arising from these differences, particularly focusing on practical methods to achieve high detection accuracy while monitoring a large number of vehicles at an appropriate cost.

Keywords: Security Operation Center, Vehicle SOC, SIEM

1. はじめに

車両のコネクテッド化や車両に関わる IT サービスのエコシステムの拡大に伴い，車両がサイバー攻撃の標的となるリスクが高まっている．また，これを受け，車両のライフサイクル全般のサイバーセキュリティに関する法規[1][2]が制定，施行されている．

車両の出荷後，利用段階においては，車両を監視し，車両に対するサイバー攻撃を検知し，検知した攻撃に対して迅速な対処，すなわち，攻撃の遮断や車両の機能の回復を行うことが重要である. 企業や組織の IT 環境に対するサイバー攻撃の監視は広く実践され，確立されているが，車両の監視には特有の課題があり，依然として困難な領域である．これらの特有の課題については，その可能性や概要が共有されている[3]が，具体的な現実の課題とその解決策は広く知られていない．

本稿では，まず，車両に対するサイバー攻撃を監視するセキュリティオペレーションセンター（車両 SOC）の試行と運用を通じて認識した IT 環境と車両の監視における主な違いと具体的な技術的課題について述べる．次に，それら課題の具体的な解決策とその効果を示す．

2. 車両の監視における課題

IT 環境の監視と比較して，車両の監視には以下のような違いが存在する．

- 監視の規模が一般的な IT 環境よりもはるかに大きい．車両の監視では数千万台の車両と数億個の車載セキュリティセンサ（IDS/IPS やファイアウォールなど）を扱う必要がある．
- IT 環境では，監視対象システムと監視を行うセキュリティオペレーションセンター（SOC）間のネットワークが常時接続されており，セキュリティセンサにより生成されたログが即時に SOC へ送信される．一方，車両の監視では，モバイルネットワークの使用やセンサの設計や実装の事由により，ログの送信が不規則に遅延し，ログの発生から SOC での受信までの遅延が可変となる．

これらの違いは以下の課題を引き起こす:

1 NTT セキュリティ・ジャパン株式会社
NTT Security (Japan) KK

* yasunobu.chiba@security.ntt

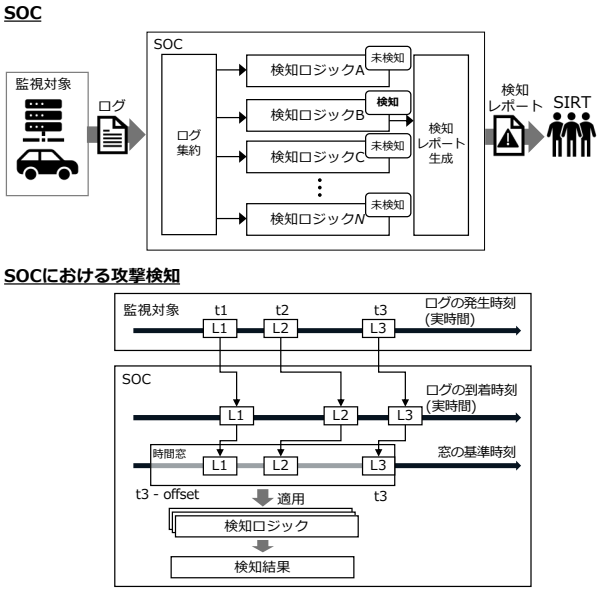

図 1　SOC における攻撃検知の一般的手法

SOC における攻撃検知は, 一般に, 時間窓, 具体的には, Sliding Window [4]を用いて一定期間のログを集約し，それを検知ロジック（複合イベント処理（CEP）[5]ルール，プログラム，機械学習・深層学習モデル[6]など）へ適用することにより実現される（図 1）.

時間窓に基づき一定期間のログを集約するためには，時間窓の生成の基準となる時刻（以降「時間窓の基準時刻」と呼ぶ）を管理する必要がある．監視対象におけるログの発生から SOC における受信までの時間（ログの受領遅延）が均一であれば，時間窓の基準時刻は単一で良いが，車両の監視ではその遅延が車両毎に秒単位から数ヶ月単位まで大きく異なるため，車両毎に異なる時間窓の基準時刻と時間窓を管理する必要がある（図 2）．さもなければ，多くのログが分析対象から漏れ，検知漏れ（偽陰性）が発生する（図 3）.

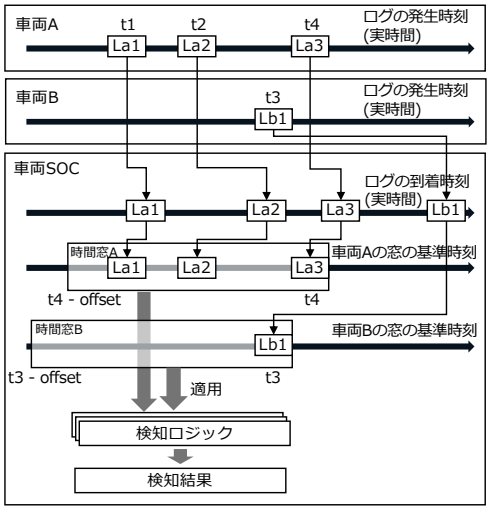

図 2　車両毎の時間窓の基準時刻と時間窓の管理

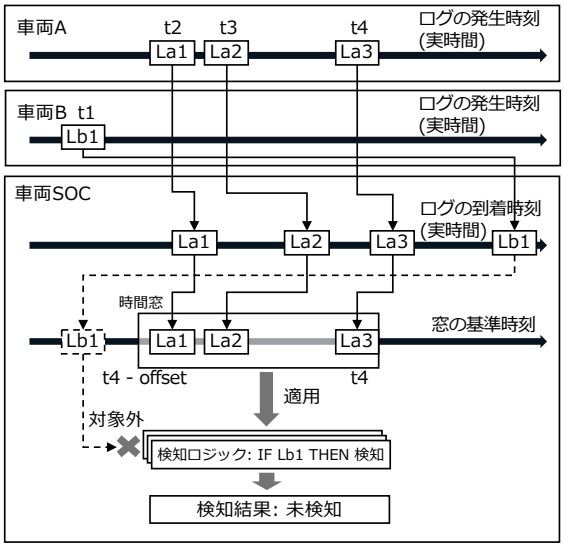

図 3　単一時間窓の使用による検知漏れの発生

結果として，個別の車両に対応する時間窓毎に検知ロジックを実行する必要があり，SOC において計算資源が爆発的に消費される．この計算資源の消費は，検知の失敗，遅延，監視対象車両数の制限，SOC の機能の停止などを引き起こす.

3. 解決策

前述の課題に対処するためには, 以下の解決策が考えられる.

- 検知ロジックの実行抑制
- 検知ロジック実行時のログ取得処理の最適化
- 計算資源が真に不足している場合の SOC の機能の縮退

本稿では，これら 3 つの解決策について議論する.

3.1 検知ロジックの実行抑制

検知ロジックの実行回数が多い理由は，車両毎に個別の時間窓の基準時刻と時間窓を管理する必要があるためである．これを軽減する方法として以下が考えられる.

(1) 時間窓の基準時刻の統合

複数の車両間で同一または近似の時間窓の基準時刻を統合し，時間窓の数を削減する（図 4）．具体的には，車両毎の時間窓の基準時刻を管理する時間窓の基準時刻テーブルを備え，車両からログを受領するごとに，ログに含まれるログの発生時刻に基づき車両毎の時間窓の基準時刻を更新する. その上で定期的に時間窓の基準時刻テーブルを走査し，同一，もしくは，近似の時間窓の基準時刻を統合，それらに基づき時間窓を生成，ログを取得，検知ロジックの実行を行う.

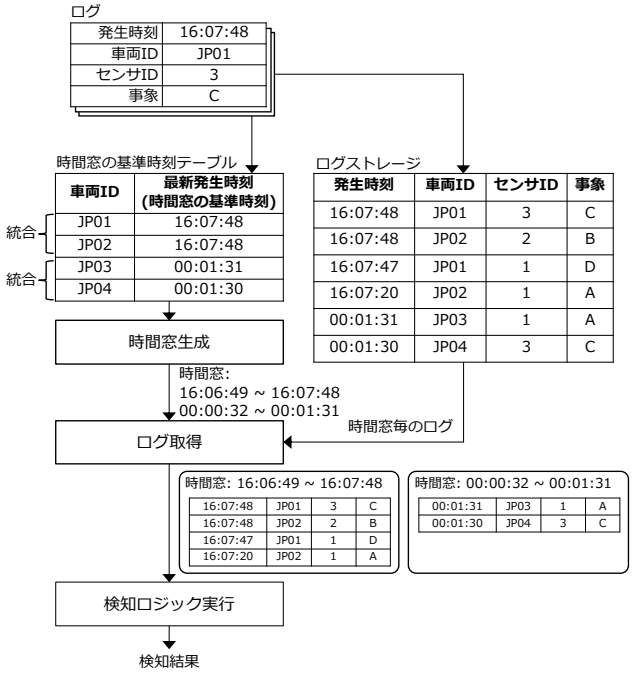

図 4　時間窓の基準時刻の統合

(2) **Hopping Window の使用と時間窓内の新規ログの管理**

Sliding Window の代わりに Hopping Window [4]を使用し，発生し得る時間窓を限定，事前に定義可能とすることで，車両毎の時間窓の必要性を排除する．Sliding Window がログの受領の度にログに含まれる時刻に基づき時間窓を生成するのに対し, Hopping Window は, 予め定めた一定間隔（ホップサイズ）ごとに時間窓を備える方式である（図 5）．加えて，時間窓に対して新たにログが含まれた際にのみ当該時間窓を検知ロジックへ適用することで，検知ロジックに適用する時間窓の数を削減する（図 6）．具体的には，予め定まる時間窓ごとに当該時間窓に含まれるログの最新の受領時刻（最新ログ受領時刻）を管理する時間窓管理テーブルを備える．その上で，車両からログを受領するごとに，ログに含まれるログの発生時刻に基づき当該ログが含まれる時間窓を特定し，当該ログを受領した時刻（実時間）に基づき時間窓ごとの最新ログ受領時刻を更新する．さらに定期的に時間窓管理テーブルを走査し，最新ログ受領時刻が前回の走査以降である時間窓を選択，それらに基づきログを取得，検知ロジックの実行を行う．

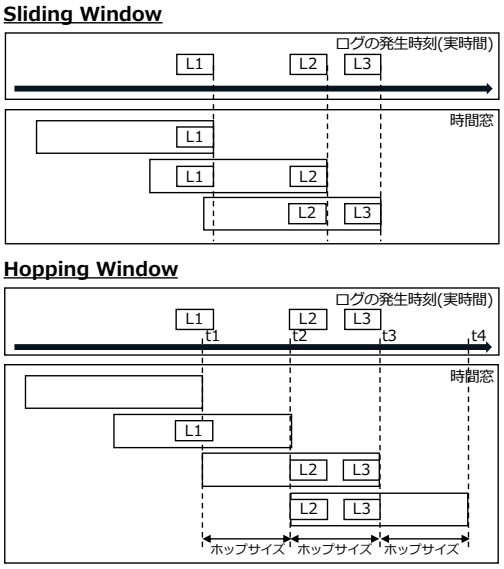

図 5　Sliding Window と Hopping Window

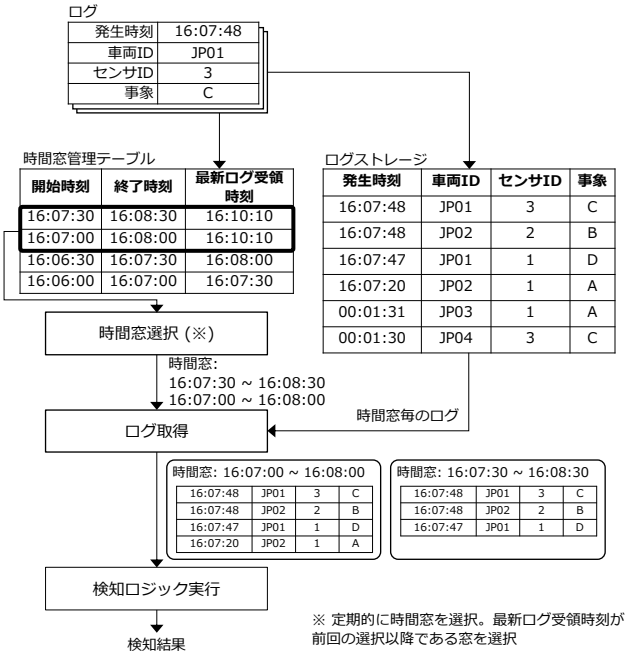

図 6　Hopping Window の使用と時間窓内のログの有無の管理

これらの方法により，SOC での攻撃検知に必要な計算資源，すなわち，CPU とメモリを大幅に削減できる．

3.2 検知ロジック実行時のログ取得処理の最適化

検知ロジックにおける典型的な検知手法として，ログの発生順や時間範囲内での同時発生に基づく検知が挙げられる．その際の処理として，一般に，以下二つの方式が考えられる．

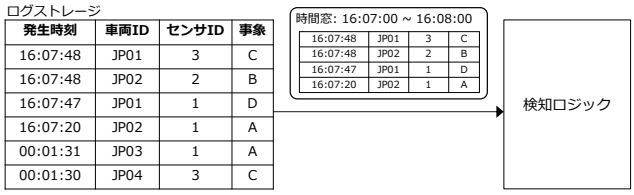

図 7　時間窓内のログの一括取得

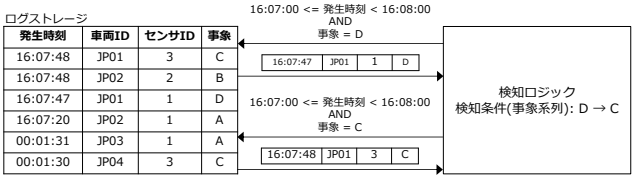

図 8　検知ロジックによるログの逐次取得

(1) 時間窓内のログの一括取得

検知ロジックの実行前に時間窓内のすべてのログを取得し，検知ロジックへ適用する（図 7）.

(2) 検知ロジックによるログの逐次取得

検知ロジックの実行中に参照が必要となったログを逐次取得する（図 8）.

後者の方式とし，それに加え，発生確率の低いログから逐次的に取得することにより，不要なログの取得を抑制でき，ログの取得に伴うディスクやネットワークの I/O 回数，帯域を軽減できる.

3.3 計算資源が真に不足している場合の SOC の機能の縮退

計算資源の消費を抑制したとしても，車両への同時多発的な攻撃により検知ロジックの実行回数が膨大となった際など，必要な資源量が確保可能な資源の上限を超過し，検知ロジックの実行失敗やシステムの停止を招く可能性がある．これを防ぐために，検知ロジックが参照するログの種類と特徴を継続的に監視し，発生頻度が著しく増加したログを参照する検知ロジックの実行を抑制することが考えられる（図 9）．これにより，一部の検知ロジックの実行は停止されるものの，他の検知ロジックの実行を継続できる.

これは SOC の機能を縮退させることになるが，SOC の機能の完全な停止を避けることが可能となる．なお，実行が抑制された検知ロジックは人手により同等の分析を行うことで，SOC の機能の大幅な劣化を回避できる.

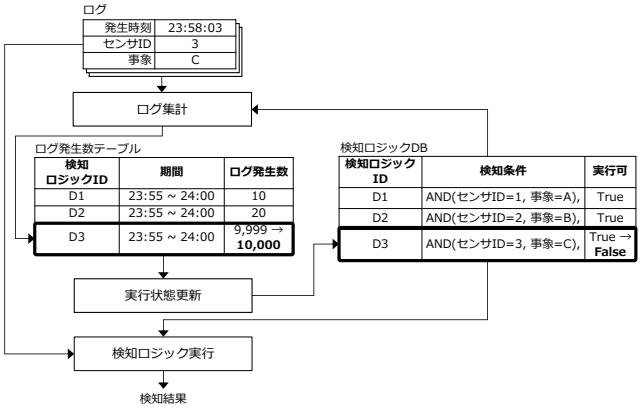

図 9　検知ロジックの実行の抑制

4. 評価

本稿では，前述の解決策のうち，特に，検知ロジックの実行抑制について評価結果を示す.

4.1 評価条件

評価にあたり，模擬車両，および，SOC において攻撃検知を行うシステム（SIEM）を用意した．加えて，車両におけるログの発生頻度，および，ログの受領遅延の条件，SOC における時間窓の条件を定めた.

(1) 模擬車両

車両の構成として，今日の車両の E/E アーキテクチャを抽象化したものを定義した．車両の ECU には各種のセキュリティセンサが搭載されており，車両における異常，または，攻撃の発生を検知し，検知を行った際，その事実をログとして記録，出力するものとした．その構成を図 10 に示す．車両の ECU を車両外と直接接続されている ECU（Electronic Control Unit），Central Gateway/Central Computer，その他の ECU に分類し，車両外と直接接続されている ECU は 2 個，Central Gateway/Central Computer は 1 個，その他の ECU は 20 個存在するものとした．また，単一の車両外と直接接続されている ECU には二つのセキュリティセンサ，Central Gateway/Central Computer には二つのセキュリティセンサ，単一のその他の ECU には単一のセキュリティセンサが搭載されているものとした.

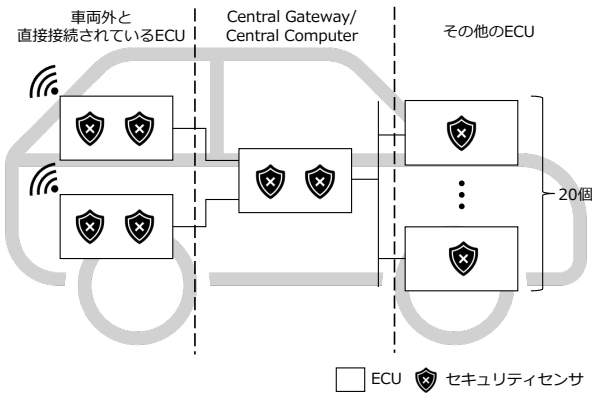

図 10　模擬車両の E/E アーキテクチャ

表 1　車両におけるログの発生頻度

センサ種別	ログの発生頻度
車両外と直接接続されている ECU のセンサ	1.0 ログ/秒
Central Gateway/Central Computer のセンサ	1.0 ログ/時間
その他の ECU のセンサ	1.0 ログ/日

(2) **車両におけるログの発生頻度**

車両におけるログの発生頻度は表 1 の通りとした．これは，車両外と直接接続されている ECU が定常的にインターネットなど外部からの攻撃を受けることを模擬している．また，監視対象となる車両が世界中（多くのタイムゾーン）に分散していることを想定し，車両の稼働率やログの発生に時間的な偏りはないものとした．加えて，車両の稼働率は 10%とした．

(3) **ログの受領遅延**

車両におけるログの発生からそれを SOC で受領するまでの遅延は 1 秒から 7 日が一様に発生することとした．

(4) **SIEM**

SIEM は車両のセキュリティセンサが出力したログを収集し，当該ログを検知ロジックに適用することにより攻撃を検知する．検知ロジックの実行基盤は CEP の一実装である EQL Analytics Library [7]とした．検知ロジックは単一とし，車両外と直接接続されている ECU のセンサのログ，Central Gateway/Central Computer のセンサのログ，その他の ECU のセンサのログが時系列に発生した際に攻撃が発生したものとした．また，その実装は Event Query Language [8]を用いて行なった．検知ロジックへ入力するログの集約範囲，すなわち，時間窓は 7 日間とした．

(5) **時間窓**

時間窓のサイズは 3600 秒とした．また，時間窓の基準時刻の統合の際，時間窓の基準時刻の差が 600 秒以内であるものを統合することとした．Sliding Window の代わりに Hopping Window を使用する際のホップサイズは 60 秒とした．

4.2 評価方法

前述の条件下で，模擬車両により発生させたログと検知ロジックにより攻撃を検知させ，検知ロジックの総実行時間，および，総メモリ使用量を計測した．解決策の効果を測るため，解決策未適用の場合と適用した場合の総実行時間，および，総メモリ使用量を比較した．

表 2　評価結果

条件	正規化 総実行時間	正規化 総メモリ 使用量
解決策未適用	1.0	1.0
時間窓の基準時刻の統合	0.730	0.990
Hopping Window の使用と時間窓内の新規ログの管理	0.063	0.035

4.3 評価結果

評価結果を表 2 に示す．この表の値は，総実行時間と総メモリ使用量を解決策未適用の値を 1 とし正規化している．複数の車両間で同一または近似の時間窓の基準時間を統合し，時間窓の数を削減した場合，解決策未適用の場合と比較して，総実行時間を 27%削減できることが確認できる．また，Sliding Window の代わりに Hopping Window を使用し，かつ，時間窓に対して新たにログが含まれた際にのみ当該時間窓を検知ロジックへ適用することで，検知ロジックに適用する時間窓の数を減らした場合，解決策未適用の場合と比較し，総実行時間を約 1/16，総メモリ使用量を約 1/29 にまで削減できることが確認できる．

5. 結論

車両に対するサイバー攻撃を監視する SOC の試行と運用を通じて認識した IT 環境の監視と車両の監視の主な差異，それに起因する技術的課題への具体的な解決策，および，その効果について示した．本稿で示した解決策により，SOC における攻撃の検知に要する計算資源を抑制可能である．これは，ひいては，多数の車両を適切なコストで高精度に監視することを可能とする．

参考文献

[1] “UN Regulation No. 155 - Cyber security and cyber security management system”. https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security, (参照 2025-07-16).

[2] “ISO/SAE 21434:2021 - Road vehicles - Cybersecurity engineering”. https://www.iso.org/standard/70918.html, (参照 2025-07-16).

[3] Hofbauer, J., Buquerin, K., Hof, H.. From SOC to VSOC - Transferring Key Requirements for Efficient Vehicle Security Operations. 21th escar Europe. 2023.

[4] Verwiebe, J., Grulich, P.M., Traub, J. et al.. Survey of window types for aggregation in stream processing systems. The VLDB Journal 32. 2023, p.985–1011.

[5] Leavit, N.. Complex-Event Processing Poised for Growth. In: Computer. IEEE Computer Society. 2009, vol. 24, no. 4, p. 17-20.

[6] Zhang, J., Pan, L., Han, Q., Chen, C., Wen, S., Xiang, Y.. Deep Learning Based Attack Detection for Cyber-Physical System Cybersecurity: A Survey. IEEE/CAA Journal of Automatica Sinica. 2022, 9 (3), p.377-391.

[7] “EQL Analytics Library”. https://eqllib.readthedocs.io/en/latest/, (参照 2025-07-16).

[8] “Event Query Language”. https://github.com/endgameinc/eql, (参照 2025-07-16).