STPA-STRIDE 기반 구조적 보안 위협 모델링 프레임워크 제안: 차량 통합제어장치 적용 사례

윤민희[1]
[1]고려대학교 SW·AI 융합대학원 소프트웨어보안학과
ymhys@korea.ac.kr

A Structured STPA-STRIDE Based Security Threat Modeling Framework: Application to Vehicle Integrated Control Units

Min-Hee Yun[1]
[1]Dept. of Software Security, Graduate School of SW·AI Convergence, Korea University

요 약

본 논문은 기존 STRIDE 위협 모델링의 한계를 보완하고자 시스템 제어 구조 기반의 STPA 기법을 STRIDE와 통합한 새로운 보안 위협 모델링 방법론을 제안한다. 제안된 방법은 위험한 제어행위(UCA)를 도출하고 이를 STRIDE 위협 범주에 매핑하여 시간적·상태적 조건 및 복합 시나리오를 체계적으로 분석할 수 있도록 한다. 본 연구에서는 차량통합제어장치(ICU)의 기능 중 도어 제어 시스템을 중심으로 절차를 설명하고 이후 ICU 전체 기능에 확장 적용하여 분석한 결과, 제안한 접근법이 기존 방법 대비 92%의 정밀도를 달성함을 확인하였다. 이는 차량 개발 초기 단계에서의 보안 요구사항 도출 및 위협 대응 전략 수립에 효과적인 분석 수단으로서의 가능성을 입증한다.

1. 서론 및 연구 배경

차량 시스템의 복잡화와 커넥티드 카의 확산은 통합제어장치(ICU)와 같은 핵심 구성요소의 사이버 보안 위협을 증가시키고 있으며, 원격 해킹 사례는 차량 보안의 중요성을 부각하고 있다[1]. 한편, 보안 위협 모델링 기법으로 널리 사용되는 STRIDE는 여섯 가지 위협 범주를 기반으로 간결한 분석이 가능하나, 정적이고 자산 중심의 특성으로 인해 동적 상태 변화나 기능 간 상호작용을 충분히 반영하지 못하는 한계가 있다[2][3][4].

한편, STPA(System-Theoretic Process Analysis)는 제어 구조를 기반으로 위험한 제어행위(UCA)를 식별하고 시스템 사고를 분석하는 방법론으로, 기능 간 상호작용과 논리적 결함 분석에 효과적이다[5]. 그러나 STPA를 보안에 적용한 STPA-Sec은 기존 보안 체계와의 정합성 부족 등으로 실무 적용에 제약이 있다[4]. 이에 본 연구는 STPA의 제어 구조 분석 기법과 STRIDE의 위협 분류 체계를 통합하여, 시간·상태 조건 및 시스템 상호작용을 반영한 구조적이고 맥락 기반의 보안 위협 모델링 방법론을 제안한다.

2. 제안하는 방법론

본 연구는 STPA의 제어 구조 분석과 STRIDE 위협 분류를 통합한 보안 위협 모델링 기법을 제안한다. 해당 기법의 절차와 적용 흐름은 도어 제어 시스템을 예시로, 구체적으로 설명한다.

2.1 시스템 이해 및 정의

분석 대상 시스템의 범위와 주요 기능을 설정하고, 보호 대상 자산과 보안 목표를 명확히 식별한다. 이 과정에서 차량 내 외부 시스템과의 인터페이스 및 상호작용 지점 또한 파악하여 위협 분석을 위한 초기 기반을 마련한다.

<표 1> 분석 대상 시스템(도어 제어) 정의 및 범위 설정

기능 분석 및 경계 정의	차량의 도어 락/언락 요청을 받아 릴레이를 제어하여 도어를 잠그거나 여는 시스템.
	입력은 사용자(차량 키, 원격 키 등) 의 요청 신호이며, 출력은 도어 락/언락 릴레이 신호.
	외부 입력(원격 키, CAN 메시지)에서 ICU를 거쳐 도어 릴레이까지의 범위를 분석 대상으로 설정.
보호 대상 자산 정의	도어 제어 릴레이 신호 (무결성)
	사용자 요청 신호(무결성, 가용성)
분석 목표 및 분석 범위 설정	공격자에 의한 부적절한 도어 제어를 방지하여 탑승자의 안전과 차량 보안을 유지
	차량 내부의 ICU 및 연계된 CAN 통신 구간에서 발생할 수 있는 사이버 위협 분석

2.2 제어 구조 모델링

차량용 ICU의 기능과 관련 구성요소를 제어 구조 형태로 모델링 (<그림1>참조)한다.

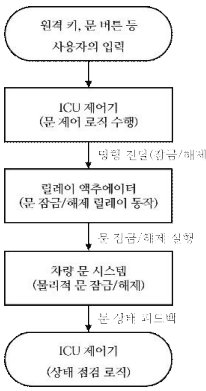


그림 1

모델에는 상위 제어 계층(차량 시스템, 운전자), ICU 내부 제어 로직, 하위 액추에이터 및 센서 간의 제어 신호와 피드백 흐름을 명시한다. STPA 분석을 위해 차량 운행 시 발생 가능한 사고(Loss, <표2>참조) 및 위험(Hazard, <표3>참조)을 정의하고, 모델링 된 제어 구조를 바탕으로 안전과 보안을 통합적으로 분석할 수 있는 기반을 구축한다.

<표 2> 도어 시스템 제어 오류에 따른 사고 가능성 유형

Loss ID	사고 유형	설명
L1	탑승자 생명 위협	사고 발생 시 도어가 열리지 않아 비상 탈출 불가, 탑승자 인명 피해 발생 가능
L2	차량 도난	도어가 비인가 상태에서 열려 차량이 절도 되는 경우
L3	탑승자 부상	주행 중 도어가 오작동으로 갑작스럽게 열려 탑승자 낙상 등 물리적 부상 초래
L4	제어기 오작동으로 인한 시스템 손상	릴레이 또는 액추에이터가 반복 제어 실패로 물리적 고장 발생

<표 3> 사고들이 발생할 수 있는 위험

Hazard ID	위험 상태	관련 설명 및 예시
H1	충돌 직후 되어 잠금 상태 유지	사고 발생 후 되어 잠금 해제 실패 → 승객 탈출 지연 (→ L1로 연결)
H2	주행 중 되어 잠금 해제 명령 실행	잘못된 조건에서 되어 연락 → 주행 중 열림 사고 위험 (→ L3로 연결)
H3	외부 공격/스푸핑에 의한 되어 연락	인증되지 않은 신호로 되어 연락 → 차량 절도 가능성 (→ L2로 연결)
H4	릴레이 제어 신호 반복 실패 또는 타이밍 오류	릴레이가 지속적으로 작동하거나 반응이 지연되어 릴레이 또는 시스템 손상 (→ L4로 연결)
H5	상태 피드백 신뢰성 부족	ICU가 문의 잠김 여부 정확히 인식하지 못함 → 제어 오류로, 사고로 이어질 가능성

2.3 UCA 도출 및 사고 시나리오 연계

모델링 된 제어 구조에서 각 제어 행위(control action)에 대하여 시스템 사고를 유발할 수 있는 위험한 제어 행위(Unsafe Control Action; 이하 UCA)를 도출한다. UCA는 부적절하거나 누락된 제어로 인해 발생하며, 다음의 네 가지 조건으로 식별할 수 있다. ①필요한 제어 행위가 제때 발생하지 않을 경우, ②불필요한 제어 행위가 불필요하게 발생할 경우, ③제어 행위가 잘못된 타이밍이나 순서로 발생할 경우, ④지속성이 필요한 제어가 너무 일찍 중단되거나 지나치게 오래 지속될 경우 [5].

각 조건이 Hazard를 유발하면 해당 제어 행위를 UCA로 정의하고 명시적으로 Hazard와 연계한다.

<표 4> 도어 제어 시스템의 UCA 유형 및 위협 시나리오

제어 행위	UCA 유형	설명(위험 시나리오)
도어 언락 명령	필요 제어 미실행	사고 시 탑승자가 갇혀 탈출 불가능
도어 락 명령	부적절한 제어 행위 실행	운행 중 의도치 않게 문 잠김(탑승자위험)
도어 락/언락 명령	잘못된 타이밍/순서	주행 중 오작동으로 인한 탑승자 이탈 위험 발생
도어 언락 유지 시간	부적절한 지속시간	장시간 문 잠김으로 보안 취약점 발생

예를 들어 도어 언락 명령이 사고 시 미실행되거나, 주행 중 불필요하게 실행되는 경우 이에 해당한다. 이처럼 도출된 UCA <표4>는 이후 보안 위협 시나리오의 핵심 분석 단위가 된다.

2.4 STRIDE 위협 시나리오 매핑

UCA 목록을 도출한 후 STRIDE의 여섯 가지 위협 범주를 적용함으로써, 공격자의 관점에서 해당 UCA가 어떤 방식으로 유발될 수 있는지를 구조적으로 분석할 수 있다. 이를 통해 시스템이 노출될 수 있는 보안 취약점을 명확하게 파악할 수 있다[6].

- 스푸핑: 공격자가 신뢰 관계나 신원을 위조하여 UCA를 유발할 수 있는가?
- 변조: 시스템 내 데이터를 조작하거나 무결성을 해치면 해당 UCA가 발생하는가?
- 부인: 공격 후 자신의 행위를 숨기거나 로그를 조작하여 추적을 어렵게 하는가?
- 정보 노출: 민감 정보가 노출되어 후속 공격으로 이어질 수 있는가?
- 서비스 거부: 자원 고갈 및 통신 방해로 제어 기능이 정상 수행되지 않는가?
- 권한 상승: 낮은 권한 상태에서 시스템의 높은 권한을 탈취함으로써 UCA를 일으킬 수 있는가?

이와 같은 분석을 통해 각 UCA와 STRIDE 위협 간의 연관성을 도출할 수 있으며, <표 5>와 같이 UCA 별로 STRIDE 범주와 대응하는 위협 시나리오를 체계적으로 정리할 수 있다.

특히, 기존 STRIDE 접근법에서는 도출이 어려웠던 시스템 맥락 기반의 위협 요소까지도 식별할 수 있다는 점에서, 본 방법론은 STPA의 시스템 사고 모델(System-Theoretic Accident Model)을 통합한 분석으로써 강점을 갖는다[6].

<표 5> UCA 기반 STRIDE 위협 유형 및 시나리오 매핑

UCA 상황	가능한 STRIDE 위협 유형	구체적 공격 시나리오 예시
필요한 언락 명령 미실행	서비스 거부	공격자가 CAN 메시지 전달 차단하여 사고 시 도어가 열리지 않음
불필요한 락 명령 실행	스푸핑, 변조	공격자가 사용자 요청 메시지를 위조하여 차량 주행 중 도어가 갑자기 잠김
잘못된 타이밍의 락/언락 명령 실행	변조, 서비스 거부	공격자가 ICU의 명령 타이밍을 지연, 조작하여 차량 상태와 무관하게 도어 제어
과도하게 긴 언락 유지	변조, 권한 상승	공격자가 제어 소프트웨어를 변조 후, 언락 상태를 지속 유지하여 차량 도난 가능성 증가

2.5 동적 및 복합 위협 시나리오 도출

최종 단계에서는 시간 흐름에 따른 시스템 상태 변화와 기능 간 상호작용을 반영한 동적 위협 시나리오와 다수의 위협 요인이 동시에 결합 되어 발생하는 복합 위협 시나리오를 도출한다.

예를 들어 차량 제어 기능을 무력화하려는 공격자는 먼저 인증 절차를 우회하는 Spoofing, 이후 제어

<표 6> UCA에 따른 STRIDE 위협 유형 및 구체적 공격 시나리오 매핑

분석 기법	활용 조건	적용 예시	비고
① 시퀀스 다이어그램 (Sequence Diagram)	• 공격 단계가 시간 흐름 순서대로 진행됨 • 행위자 간 메시지 교환 중심 • "A가하고→B가 반응하고→C가 영향받는다" 구조	• 원격 키 스푸핑 (S) • 앱 통신 감청 후 언락(I→S) • 타이밍 오류로 인한 점등 실패(T)	• 행위 기반 분석에 적합 • 사용자 - ICU - 릴레이 간 상호작용 시나리오
② 공격 트리 (Attack Tree)	• 다단계 공격 조건이 AND/OR 논리로 연결됨 • 복합 공격 경로로 단일 목표 달성 • "이것 AND 저것→결과 발생" 구조	• 펌웨어 변조 + DoS 조합 (T + D) • 정보 탈취+인증 우회(I+S) • 센서 조작+명령 차단(S+T)	• 복합 논리적 구조 표현에 적합 • 보안 침투 경로 시각화에 효과적
③ 조건 기반 상태 시나리오 분석	• 특정 시스템 상태나 시간 조건이 위협 발생의 핵심 • "이 상태 + 이 시점→위협 성립" 구조 • 시스템 동작 맥락이 반드시 포함됨	• 사고 감지 후 1분 내 언락 실패(D) • 진단 세션이 주행 중 유지됨(T) • 주행 중 릴레이 과도 지속(T)	• STPA의 UCA "Context"와 매우 유사 • 시간/상태 기반 조건 명시 가능
④ STRIDE 기반 시나리오 구조화 분석	• STRIDE를 단순 분류가 아닌 공격 단계로 사용 • 각 시나리오의 행위/상태 변화에 STRIDE 요소를 대응 • STRIDE가 "A 공격자→B 행위(S/T/…)"로 사용됨	• 스푸핑(S) → 변조(T)로 상태 지속 • 정보 노출(I)→스푸핑(S) 연계 • 변조(T)→DoS(D) 연쇄 공격	• STRIDE를 시나리오 구성단위로 활용 • 기존 STRIDE의 정적 자산 중심의 한계를 보완 • "S+T+D"식 조합으로 시나리오 분석 구조화

<표 7> 도어 시스템에 대한 복합 보안 위협 시나리오 유형별 특성 및 분석 방법 비교

시나리오 분류	시나리오 상세 내용	시나리오 특징	시나리오 분석을 위해 사용된 기법
시나리오 1 (S+T 복합)	원격 키 신호 스푸핑(S)으로 언락 요청 위조 → ICU 펌웨어 변조(T)로 언락 상태 지속	시간순으로 명확한 행위 흐름이 있음 단순 스푸핑→변조 단계로 흐름이 선형적	시퀀스 다이어그램
시나리오 2 (T+D 복합)	ICU 펌웨어 변조(T)로 메시지 우선순위 조작 → 사고 발생 시 DoS(D) 공격으로 도어 제어 실패	다단계 복합 공격구조(AND 조건) 복합 노드 기반 구조적 분석에 적합	공격 트리 분석
시나리오 3 (I+S 복합)	차량 잠금 정보 노출(I) 후 해당 정보를 이용한 원격 신호 스푸핑(S)으로 차량 접근 및 도어 열림	탈취→ 스푸핑→ 연락까지의 시간 흐름 중요 선형적/단계적 행위 전개	시퀀스 다이어그램
시나리오 4 (상태 기반 조건 추가)	사고 감지 직후 특정 시간 조건(사고 후 1분 내)에서 DoS 공격으로 언락 명령 실패 시나리오 추가	시간 조건(사고 직후), 상태 조건(충돌 감지 후) 상태 기반 위협 발생 시점의 조건 정의 필요	조건 기반 상태 시나리오 분석

명령을 변조하는 Tampering, 해당 명령 전달을 지연시키거나 차단하는 DoS 공격을 연쇄적으로 수행할 수 있다. 해당 단계에서는 시스템 제어 구조와 UCA 발생 조건 및 실제 공격 경로 간 인과관계를 통합적으로 고려해 위협 간의 연결 구조를 정교하게 구성하였다.

이 과정에서 각 위협 시나리오의 구조적 흐름과 유형별 특성을 시각화해서 <표 6, 7>에 표현하고 이를 기반으로 정량적 기법(시나리오 정밀도 계산)과 정성적 분석(시나리오 유형 분류 및 시퀀스 기반 평가) 을 병행하여 위협 간 상호작용을 모델링 하였다. 제안한 모델링 기법은 공격 트리(Attack Tree)와 시퀀스 다이어그램(Sequence Diagram)과 같은 시나리오 기반 분석 기법을 활용해서 단일 공격 행위가 아닌 많은 이벤트가 순차적 또는 병렬적으로 결합 되어 발생하는 현실적인 공격 경로식별 효과를 극대화하였다.

본 연구에서 제안한 접근법은 기존의 정적이고 개별적인 STRIDE 기반 분석이 가진 한계를 극복하고, 현실적이고 심각한 위험성을 지닌 복합 위협 시나리오를 포괄적으로 분석할 수 있는 체계적 틀을 제공한다.

3. 결과 및 비교 분석

제안한 분석 방법론은 도어 제어 시스템을 중심으로 설명되었으나 진단 기능, 강제 구동 등 ICU 전체 기능에 동일한 방식으로 수행되었고 이를 바탕으로 기존 STRIDE 분석 기법과의 성능을 비교하였다. 따라서, 정밀도 측정 및 위협 시나리오 분포 분석에는 전체 기능에서 도출한 결과가 반영되었다. 분석 및 평가 기준은 식별된 위협 시나리오의 수, 정확도(Precision), STRIDE 유형별 발생 빈도로 설정하였으며. 이때 정확도는 다음과 같은 방식으로 정의하였다.

$$\text{정밀도} = \frac{\text{실제로 } Hazard\text{와 연결되는 시나리오 수}}{\text{전체 도출된 위협 시나리오 수}} \times 100\%$$

본 논문에서는 시스템 사고(Hazard)와 명시적으로 연결된 위협 시나리오의 비율을 정밀도로 정의하여, 단순 수량이 아닌 실효성 중심의 위협 식별 성과를 측정하였다. 두 분석 결과는 <표 8>에 요약되어 있으며, 각 기법의 실효성을 수치 기반으로 비교하였다.

<표 8> 방법론별 위협 시나리오 도출 수 및 정밀도 비교

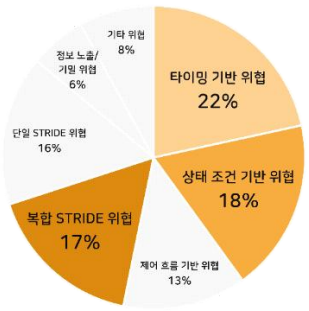
분석 방식	위협 시나리오	Hazard 연결시나리오	정밀도
기존 STRIDE 위협 모델링	228	153	67%
제어 행위 기반 위협 모델링	**250**	**230**	**92%**

기존 STRIDE 기반 기법은 총 228건의 위협 시나리오 중 약 67%만이 시스템 Hazard와 명확히 연결되어 위협 식별의 실효성 측면에서 한계를 보였다. 즉, 위협 시나리오 수는 많았으나 실질적인 보안 설계에 활용하기 위해서는 후속 선별이 필요한 구조였다.

반면, 본 논문에서 제안한 방법론은 총 250건의 위협 시나리오 중 약 92%인 230건 이상이 사전에 정

의된 사고 시나리오(Hazard)와 명확히 연계되어 높은 정밀도를 달성하였다. 단순히 위협을 나열하는 데 그치지 않고 실제 사고 유발 가능성이 높은 시나리오에 집중함으로써 분석의 실효성을 높였다.

위협 시나리오의 유형별 정량적 분포를 분석한 결과 타이밍 기반 위협(21.6%), 상태 조건 기반 위협(18.4%), 복합 STRIDE 위협(16.8%)이 전체의 56.8%를 차지하였다. 이는 정적인 자산 중심 접근에 머무르는 STRIDE 기반 분석과 달리 제안한 방법론은 시스템의 동적 조건 및 복합 제어 흐름에 따른 위협을 효과적으로 식별하고 있음을 알 수 있다. 특히 복합 STRIDE 위협이 전체 시나리오에서 높은 비중을 차지한 점은 단일 위협 유형만을 고려한 기존 방어 전략으로는 복합 위협 시나리오에 대해 대응이 어렵다는 점을 보여준다. 따라서 다양한 위협 요소가 연결된 복합적인 공격 경로를 포괄할 수 있는 통합 보안 설계가 요구된다. 실제로 STRIDE 단독 분석 결과 단일 유형으로 분류된 시나리오는 전체의 약 16%에 불과해 기존 방법이 포착할 수 있는 위협 범위에 구조적인 한계가 있음을 드러낸다. 이러한 결과는 단순히 위협의 양적 식별에 그치지 않고, 위협 발생 조건과 흐름을 구조적으로 파악할 수 있는 분석 기반으로 확장되었음을 의미한다.



제안한 방법론은 각 위협을 시스템 맥락(Context)과 제어 흐름(Flow)에 따라 구조화하여 제어 실패와 사고 간 인과관계를 정량적으로 설명할 수 있도록 한다. 나아가, 보안 대응 전략 측면에서도 개별 위협 대응을 넘어서 공통 원인 기반의 통합적 대응 체계로 확장할 수 있음을 보여준다.

4. 결론

제안된 통합 기법을 실제 시스템 사례에 적용하여 도출된 위협 시나리오의 결과를 분석하고, 기존의 방법을 적용한 결과와 비교하였다. 그 결과 통합 기법은 STRIDE 기법만을 사용할 때보다 더 다양한 위협 시나리오를 식별하는 데 기여하는 것으로 나타났다. 특히, STPA 기반 분석을 통해 시스템 제어 상호작용 과정에서 발생할 수 있는 동적 시나리오를 추가로 발견할 수 있었으며, STRIDE 분류를 적용함으로써 STPA 단독으로는 명확히 드러나지 않던 구체적인 공격 경로들도 체계적으로 파악할 수 있었다. 이러한 비교 분석을 통해 두 기법의 상호 보완적 효과가 입증되었으며, 통합 방법론이 기존 방법론에 비해 위협 간 인과 구조를 토대로 복합 위협 시나리오를 체계적으로 구성하고, 실질적인 공격 경로식별 결과를 제공함을 확인하였다. 이로 인해 보안 요구사항 도출과 대응 전략 수립의 일관성 및 실효성을 크게 향상 시킬 수 있음을 알 수 있다.

특히, 사전 예방 중심의 보안 설계에 효과적인 위협 분석을 제공한다는 점에서 실질적인 가치를 지닌다. 다만, 본 방법론은 전문가의 개입이 필요한 수작업 기반 분석이라는 한계가 있으며, 향후에는 모델링 자동화 및 도구화를 통한 실무 적용 성과 분석 효율성을 제고할 필요가 있다.

참고문헌

[1] K. Koscher, A. Czeskis, F. Roesner, S. Patel, and T. Kohno, Experimental Security Analysis of a Modern Automobile, 2010 IEEE Symposium on Security and Privacy, Oakland (USA), 2010, pp. 447 - 462.

[2] Y. Li, W. Liu, Q. Liu, X. Zheng, K. Sun, and C. Huang, Complying with ISO 26262 and ISO/SAE 21434: A safety and security co-analysis method for intelligent connected vehicle, Sensors, vol. 24, no. 6, p. 1848, 2024.

[3] F. Swiderski and W. Snyder, Threat Modeling: Designing for Security, Redmond, WA, Microsoft Press, 2004.

[4] C. Schmittner, Z. Ma, and P. Puschner, Limitation and improvement of STPA-Sec for safety and security co-analysis, Computer Safety, Reliability, and Security (SAFECOMP 2016), Cham, 2016, pp. 195 - 209.

[5] N. G. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety, Cambridge, MA, MIT Press, 2011.

[6] N. P. de Souza, C. A. C. César, J. de M. Bezerra, and C. M. Hirata, Extending STPA with STRIDE to identify cybersecurity loss scenarios, Journal of Information Security and Applications, vol. 55, p. 102618, 2020.