자율주행 자동차 내부 CAN 네트워크 공격 탐지를 위한 프레임워크 개발

김수빈*, 고예지*, 김세진*, 이예솔*, 석지은*, 임강빈**

*순천향대학교(대학생), **순천향대학교(교수)

Development of a Framework for Detection of CAN Network Attacks inside Autonomous Vehicles

Jieun Seok*, Subin Kim*, Yeji Koh*, Sejin Kim*, Yesol Lee*, Kangbin Yim**

*Soonchunhyang University(Undergraduate student)

**Soonchunhyang University(Professor)

요 약

자율주행 기술은 주변 환경과 통신하여 위험 분석과 운행 경로를 계획하는 기술로 운전자의 안전성과 효율성, 편의성을 향상시킨다. 자율주행 차량의 내부에서는 모듈과 컨트롤러가 CAN 네트워크를 사용하여 제어된다. 이러한 네트워크는 Broadcast 형식으로 통신하여 송신자를 확인하지 않는다는 점에서 메시지 주입과 같은 공격을 야기할 수 있다. 본 논문에서는 실제 자율주행기술이 탑재된 차량을 기반으로 자동차 내부 네트워크인 CAN 프로토콜의 데이터 수집을 통하여 위협 가능성을 분석한다. 또한, 수집한 CAN 프로토콜을 기반으로 실제 주행 상황의 재연과 실시간 모니터링 및 비정상 메시지 탐지가 가능한 동적 분석 프레임워크를 개발한다.

I. 서론

현대의 자동차는 이동수단으로서의 의미뿐만 아닌 다양한 기능이 결합하여 자율주행이라는 새로운 분야로서 발전해가고 있다. 자율주행이란 운전과 관련된 기술적인 간섭 없이 시스템 자체적으로 판단하여 차량이 주행하는 것이다. 차량에 부착된 레이더, 카메라, 초음파 센서 등으로 차간거리 제어, 차선 유지 제어, 충돌 회피기술과 같은 위험 요소를 판단하고 주변 환경을 인식하여 경로를 계획한다.[1]

본 논문에서는 자율주행 시스템이 탑재된 차량을 기반으로 수집된 CAN 메시지 주입 및 재실행을 위한 프레임워크를 구축한다. 구축한 프레임워크를 이용하여 각 모듈에서 추출한 CAN 메시지를 수집 및 분석하는 과정을 진행한다. 이후, 실제 주행을 통해 수집한 CAN 메시지 데이터를 프레임워크에 주입 및 재실행한다. 동시에 프로그램을 통해 아이디의 경우 프레임워크에서 모듈별로 수집하였던 CAN 메시지 아이디와의 비교 분석을 진행하며, 메시지 데이터값의 경우 일정 값 이상의 오차 발생을 감지한다. 감지된 이상 데이터는 웹 페이지를 이용해 시각화하여 CAN(Controller Area Network) 프로토콜의 실시간 모니터링이 가능하도록 하였으며 해당 과정에서 특정 CAN 메시지에 비정상적인 메시지가 탐지될 경우, 그와 관련된 공격 탐지 및 경고 알림이 발생하도록 개발했다.

II. CAN 네트워크 분석

2.1 CAN 네트워크

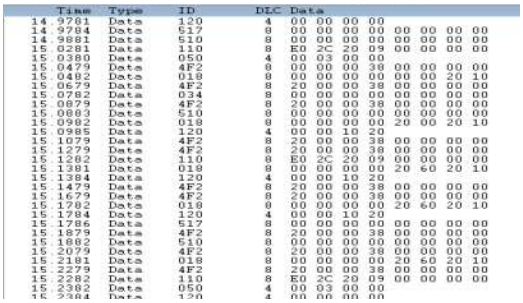
주행 데이터의 송·수신 과정에서 차량의 각 모듈은 차량 내 통신 네트워크 형식인 CAN 프로토콜을 사용하여 통신한다[2]. CAN 버스를 통해, 병렬로 연결된 다수의 ECU 간 데이터를 주고받는[3] CAN 네트워크 종류에는 C-CAN, B-CAN, M-CAN 등이 존재한다.

C-CAN은 속도, RPM, 브레이크 등 전반적

인 차량 운행과 관련된 데이터가 통신하는 네트워크이다. B-CAN은 차량 차체 부분의 제어 메시지가 통신하는 네트워크이며, M-CAN은 내비게이션과 같은 차량 내 멀티미디어 데이터 관련 메시지가 통신하는 네트워크이다[4].

2.2 CAN 메시지 수집

아래 [그림 1]은 실제 자율주행 차량의 ECU에서 수집한 CAN 데이터가 CAN ID, Type, DLC, Data 순서로 시각화된 화면이다. 프로그램을 통해, 미리 수집한 데이터뿐만 아니라 CAN 수집기와 각 모듈을 연결하여 수집한 모듈이 동작할 때 생성되는 CAN 데이터 또한 실시간으로 확인할 수 있다.



Time	Type	ID	DLC	Data
14.9701	Data	120	4	00 00 00 00
14.9704	Data	517	8	00 00 00 00 00 00 00 00
14.9881	Data	510	8	00 00 00 00 00 00 00 00
15.0281	Data	110	8	E0 2C 20 09 00 00 00 00
15.0380	Data	050	4	00 03 00 00
15.0479	Data	4F2	8	00 00 00 38 00 00 00 00
15.0482	Data	018	8	00 00 00 00 00 00 20 10
15.0679	Data	4F2	8	20 00 00 38 00 00 00 00
15.0782	Data	034	8	00 00 00 00 00 00 00 00
15.0879	Data	4F2	8	20 00 00 38 00 00 00 00
15.0883	Data	510	8	00 00 00 00 00 00 00 00
15.0982	Data	018	8	00 00 00 00 20 00 20 10
15.0985	Data	120	4	00 00 10 20
15.1079	Data	4F2	8	20 00 00 38 00 00 00 00
15.1279	Data	4F2	8	20 00 00 38 00 00 00 00
15.1282	Data	110	8	E0 2C 20 09 00 00 00 00
15.1381	Data	018	8	00 00 00 00 20 60 20 10
15.1384	Data	120	4	00 00 10 20
15.1479	Data	4F2	8	20 00 00 38 00 00 00 00
15.1679	Data	4F2	8	20 00 00 38 00 00 00 00
15.1782	Data	018	8	00 00 00 00 20 60 20 10
15.1784	Data	120	4	00 00 10 20
15.1786	Data	517	8	00 00 00 00 00 00 00 00
15.1879	Data	4F2	8	20 00 00 38 00 00 00 00
15.1882	Data	510	8	00 00 00 00 00 00 00 00
15.2079	Data	4F2	8	20 00 00 38 00 00 00 00
15.2181	Data	018	8	00 00 00 00 20 60 20 10
15.2279	Data	4F2	8	20 00 00 38 00 00 00 00
15.2282	Data	110	8	E0 2C 20 09 00 00 00 00
15.2382	Data	050	4	00 03 00 00
15.2384	Data	120	4	00 00 00 00

[그림 1] 수집된 CAN 메시지

실제 자율주행 차량의 ECU에 해당하는 ICU, IBU, ECM, CLUSTER, AVN, MDPS, DATC의 모듈별 단위 테스트를 통해 개별 CAN 데이터를 수집하였으며, ECU를 두 개 이상 연결할 경우 발생하는 CAN 데이터 또한 동일한 방법을 이용하여 수집하였다. 실제 주행 데이터의 경우, 연구 대상인 차량을 일정 거리를 주행하며 실시간으로 발생하는 데이터를 수집하였다. 이처럼 수집한 두 가지의 데이터를 비교하여 중복되는 CAN 데이터를 분석하였다.

2.3 CAN 메시지 분석

전반적인 차량 운행에 관련된 C-CAN 데이터 중 차량의 속도 값과 RPM, 기어, 좌우 방향등, 비상등, 브레이크 및 충돌 경고음을 집중적으로 분석하였다. 다음 [표 1]은 집중적으로 분석이 이뤄진 C-CAN의 메시지 데이터값이다.

분석을 통해, 실제 주행 중 수집한 CAN 데이터값과 분석한 내용을 기반으로 프로그램에 직접 조작하여 주입한 CAN 데이터값의 결과가 일치함을 확인할 수 있었다.

CAN ID	기능	DATA
386h	속도	홀수 바이트의 첫 번째 비트 값에 따라 속도 값의 범위 변화
366h	RPM	3byte에서 (상위 4비트 * 0.065 + 하위 4비트) * 1000 연산 수행 시, RPM 값 예측 가능
367h	기어	4byte에서 x5이면 D/ x0이면 P
541h	좌측 방향 지시등	2byte에서 41이면 On/ 49이면 Off
541h	우측 방향 지시등	7byte에서 4D이면 On/ 0D이면 Off
541h	비상등	2byte와 7byte에서 41이면 On/ 4D이면 Off
4a9h	브레이크	7byte가 브레이크를 밟으면 40/ 밟지 않으면 00
58bh	충돌 경고음	3byte에서 0x이면 Off

[표 1] 분석한 CAN 메시지 데이터

III. CAN 네트워크 공격 탐지 방안

3.1 공격 탐지를 위한 Database 구축

데이터 관리 기능 및 보안성을 위해 수집된 데이터의 Database를 구축하였다. CAN의 종류에 따라 테이블을 생성하고 각각의 테이블에는 CAN ID, 길이, 데이터 필드를 생성한다. CAN 수집기를 통해 수집된 데이터는 파이썬 코드 내의 ReadData 함수에서 Json 포맷 형태로 변환하고, ProcessData 함수 부분에서 16진수로 표현되어있는 데이터 부분을 가공하여 일반 사용자도 쉽게 의미를 알아차릴 수 있도록 한다.

아래의 [그림 4]는 수집된 차량 속도 데이터를 가공하는 코드 중 일부분이다. 수집된 데이터(예: 01 CA 01 05 01 4F 01 C0)를 가지고 데이터를 가공하여 70이라는 가공된 값을 데이터베이스에 전송하고 동시에 이상 데이터 탐지를 수행한다. 가공된 속도 값과 이전의 속도 값의 차이가 일정 값 이상 발생하거나 단위 테스트 과정에서 수집되지 않은 CAN ID가 탐지될 경우 이상 데이터로 간주한다.

```
if msg.ID == 902:
    if Data[3] == "0" or Data[3] == "4" or Data[3] == "8" or Data[3] == "c":
        n = 0
        while (n < 23):
            temp = float(int(Data[4 + n],16))
            value = value + temp * 2
            temp = float(int(Data[6 + n],16))
            value = value + temp * 0.15
            n = n + 6
        Data = str(value)
```

[그림 2] 데이터 가공 코드 - 속도

이와 마찬가지로 수집된 RPM 데이터도 가공 과정을 거쳐 이해하기 쉬운 데이터로 변환되어 데이터베이스에 저장된다. 이 부분에서도 이전 RPM 값과 비교하여 일정 값 이상 차이가 나면 이상 데이터로 간주하는 이상 탐지 기능이 실행된다. 탐지 과정에서 이상 데이터로 간주 될 경우, 속도 및 RPM 값이면 -1, CAN ID 부분이면 0xffffffff이 데이터 필드 값에 저장된다.

가공된 데이터는 SendData 함수 내의 SQL 쿼리문을 통해 약 0.5초마다 데이터베이스에 저장된다. 다음 [그림 5]는 가공된 데이터가 C_CAN 테이블에 저장된 모습이다.

```
mysql> select * from c_can;
+------------+--------+-------------------------+
| ID         | Length | Data                    |
+------------+--------+-------------------------+
| 0x1        |      4 | 00 00 00 04             |
| 0x130      |      8 | d8 7f 7f 80 00 00 0b 81 |
| 0x140      |      8 | 33 7f 00 71 20 00 0a 16 |
| 0x153      |      8 | 20 80 10 ff 00 ff 70 1e |
| 0x162      |      3 | 00 8e 15                |
| 0x164      |      4 | 00 08 16 28             |
| 0x220      |      8 | ff 03 7e 00 00 f0 0f b6 |
| 0x222      |      8 | 60 00 03 00 00 00 00 00 |
```

[그림 3] 저장된 가공된 CAN 데이터

IV. 테스트 프레임워크 개발

3.1 테스트베드 구축

아래 [그림 6]은 실제 차량 내부를 그대로 재현한 프레임워크의 모습이다. 전체적인 전원은 파워를 사용하여 모듈에 전원을 공급하고 아두이노 릴레이를 통해 모듈들의 ON/OFF 제어가 가능하다. 아크릴 박스 최하층 칸에는 B-CAN, C-CAN, M-CAN 별로 CAN 버스를 배치하여 데이터를 수집한다. 프레임워크에 위치한 모니터 중 좌측은 실제 주행 데이터 수집 과정에서 녹화된 영상이 재생되고, 우측에는 실시간으로 주입되는 CAN 메시지가 분석된 결과가 웹 페이지를 통해 실시간으로 시각화된다.

[그림 4] 테스트 프레임워크 전체 모습

3.2 CAN 네트워크 공격 탐지

웹 페이지는 가공된 CAN 데이터가 저장되어 있는 DB로부터 결과를 받아와 보기 쉽게 UI로 차량의 속도, RPM, 방향, 브레이크의 실시간 상태를 시각화하였다.

[그림 5] CAN메시지 조작 탐지 화면

수집된 CAN 데이터의 실시간 모니터링이 가능한 웹 페이지는 [그림 7]과 같다. CAN 메시지 데이터 이상 탐지 시, 웹 페이지 좌측 상단에 메시지 조작 탐지 문구와 함께 알림을 생성하여 공격 탐지 결과를 알려준다.

V. 결론

본 논문에서는 실제 자율주행 시스템이 탑재된 차량을 기반으로 테스트 프레임워크를 구축하였으며 수집한 CAN 메시지 데이터를 분석하여 실시간 공격 탐지 모니터링이 가능한 웹 페이지를 개발하였다.

테스트 프레임워크는 실제 자율주행 차량에서 수집한 데이터와 모듈별 데이터를 이용하여 차량 내의 메시지 조작, 비정상 전달 등의 시스템 공격 탐지 기술개발에 활용할 수 있다.

[참고문헌]

[1] 이병윤, “국내외 자율주행자동차 기술개발 동향과 전망”, 2016.

[2] 이병수, 박민규, 성금길, “CAN을 기본으로한 전기자동차용 차량 네트워크 교육용 시스템 개발”, 2006.

[3] Youngho An, Junyoung Park, Insu Oh, Myoungsu Kim, Kangbin Yim, “Design and Implementation of a Novel Testbed for Automotive Security Analysis”

[1] 커넥티드 카 침해사고 시나리오 모델 구현 및 분석방법 연구(KISA-WP-2018-002)