**arp spoofing (네트워크) 스터디**

<https://www.itfind.or.kr/publication/regular/weeklytrend/pastList/read.do?selectedId=1240&selectedCategory=B_ITA_01&pageSize=10&pageIndex=0>

spoofing공격 시나리오 <http://www.koreascience.or.kr/article/JAKO202111037333685.pdf>

ARP: 네트워크 상에서 IP주소를 물리적 네트워크 주소로 대응(bind) 주는 프로토콜

같은 네트워크에서 해당 IP 주소를 보낼 때 어떤 기기를 보낼지에 대한 설정을 위한 것으로, IP주소와 MAC 주소를 테이블로 연결

해당 호스트가 네트워크에 연결된 모든 호스트에 arp를 request하고, 응답하는 reply 패킷을 보낸 패킷과 연결

**arp spoofing**

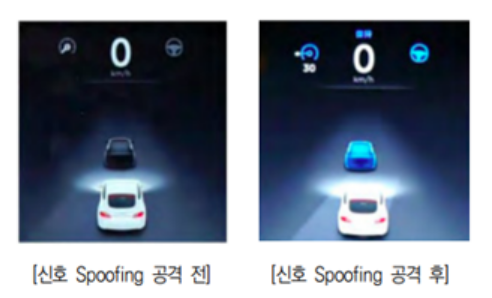
네트워크를 통해 밖으로(WAN으로) 가기 위한 통로(게이트웨이)를 속여서 피해자는 공격자에게 모든 데이터를 보내게하는 방법

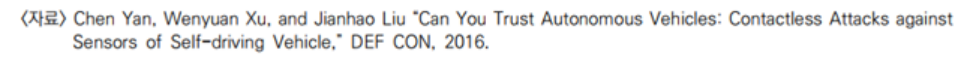
스푸핑은 네트워크나 웹사이트의 데이터를 위조 내지 변조함으로써 정상 시스템인양 일반 사용자를 속이는 해킹 기법

arp 테이블 확인 방법

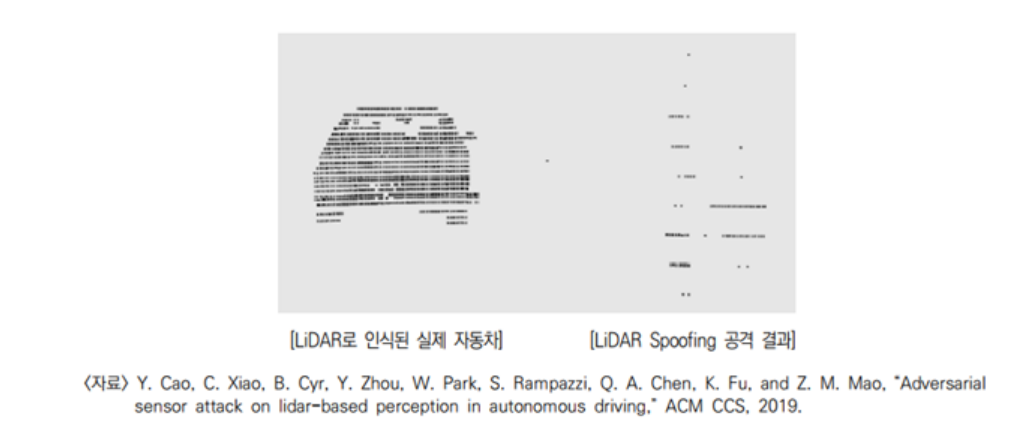
* cmd창에 ‘**arp -a**’ 입력

**레이더 센서**와 **라이더(Lidar) 센서**는 spoofing 공격이 가능





**레이더 센서**는 자동 긴급 브레이크 기능, 그리고 차선을 유지하며 주행할 수 있도록 하는 것이다. 이를 위해 **전자파를 방출하고 수신하는 원리를 통해 이른바 ‘time-of-flight’**, 즉 빛이 피사체에서 반사되어 **돌아오는 시간을 측정**하여 거리를 계산하는 비행시간법을 통해 장애물과의 거리를 측정  
  
네트워크 시스템의 IP 주소나, 도메인 네임 시스템(DNS) 등을 위·변조하여 공격 대상인 시스템이 정상적인 사용자가 접근하는 것으로 오인하게 한다. 실험 결과 스푸핑에 의해 레이더 센서가 주변 장애물과의 측정 거리를 잘못 측정  
실험에선 레이더 센서에서 발생하는 비슷한 무선 신호를 이용하여 스푸핑 공격을 장애물과의 거리를 주기적으로 변경



**라이더(LiDAR) 센서** 역시 레이저를 이용한 스푸핑 공격에 취약한 것으로 나타났다. 이 경우 자율주행차의 라이더 센서에 레이저 신호를 재생하여 스푸핑 공격을 수행함으로써 사물의 위치를 더 멀리 혹은 더 가깝게 있는 것으로 착각하게 할 수 있다

라이더 센서는 고출력의 펄스레이저를 이용해 물체에 반사되어 돌아오는 레이저 빔의 시간을 측정해 거리 정보를 획득한다. 이를 이용해 라이더 객체 인식 시스템을 속이기 위한 적대적 모델 생성 방법으로 가짜 장애물을 통해 자동차의 비상 브레이크를 작동시키는 스푸핑 공격이 가능하다.

**spoofing 공격 시나리오**

LiDAR 센서 구매 링크

[**https://www.icbanq.com/P012835088?utm\_source=naver&utm\_medium=cpc&utm\_campaign=%EC%87%BC%ED%95%91\_%EC%9E%90%EC%82%AC%EB%AA%B0&utm\_id=%EC%87%BC%ED%95%91\_%EC%9E%90%EC%82%AC%EB%AA%B0&utm\_term=notset&utm\_content=notset&n\_media=27758&n\_query=LIDAR%EC%84%BC%EC%84%9C&n\_rank=1&n\_ad\_group=grp-a001-02-000000031495916&n\_ad=nad-a001-02-000000222194228&n\_campaign\_type=2&n\_mall\_id=powercomer&n\_mall\_pid=P012835088&n\_ad\_group\_type=2&NaPm=ct%3Dlcp4kvm0%7Cci%3D0z80003PkpLxn4PS41nO%7Ctr%3Dpla%7Chk%3D3711357d7cf5213ef749ccedc933101b4db6ab11**](https://www.icbanq.com/P012835088?utm_source=naver&utm_medium=cpc&utm_campaign=%EC%87%BC%ED%95%91_%EC%9E%90%EC%82%AC%EB%AA%B0&utm_id=%EC%87%BC%ED%95%91_%EC%9E%90%EC%82%AC%EB%AA%B0&utm_term=notset&utm_content=notset&n_media=27758&n_query=LIDAR%EC%84%BC%EC%84%9C&n_rank=1&n_ad_group=grp-a001-02-000000031495916&n_ad=nad-a001-02-000000222194228&n_campaign_type=2&n_mall_id=powercomer&n_mall_pid=P012835088&n_ad_group_type=2&NaPm=ct%3Dlcp4kvm0%7Cci%3D0z80003PkpLxn4PS41nO%7Ctr%3Dpla%7Chk%3D3711357d7cf5213ef749ccedc933101b4db6ab11)

신호 오류 주입 공격을 위한 신호발생기, 증폭기, 스피커 준비



LiDAR 센서와 아두이노 보드를 연결 후 PC에 연결



아두이노 보드에서 LiDAR 센서의 신호를 실시간으로 읽어오는 프로그램 작성



신호발생기, 증폭기, 스피커를 이용해 LiDAR 센서에 악의적인 신호(노이즈) 주입

* 물체 인식 알고리즘 spoofing



작성한 프로그램을 통해 오류 주입 전과 후의 신호 값 비교

= 신호 값에 따라 거리 정보를 얻기 때문에 사물의 위치 오인 가능

**센서 값 조작 (기기) 스터디**

<https://koreascience.kr/article/JAKO201734964190851.pdf>

아두이노

: 사용하기 쉬운 하드우어 및 소프트웨어를 기반으로 하는 오픈소스 전자 플랫폼

아두이노의 특징

* 저사양의 MCU(Micro Controller Unit)를 탑재
* 센서와 액츄에이터를 이용할 수 있는 여러 개의 디지털 핀과 아날로그 핀 존재

MCU

: 마이크로프로세서와 입/출력 모듈을 하나의 칩으로 만들어져 정해진 기능을 수행하는 컴퓨터

MCU의 특징

* CPU 코어, 메모리, 프로그램 가능한 입/출력 보유
* 임베디드 애플리케이션을 위해 고안됨 => 임베디드 시스템에 널리 사용
* 기능을 설정 후 장치 등에 장착되어 정해진 일을 수행하도록 동작
* PC에 비해 성능이 낮음

디지털 핀, 아날로그 핀의 특징

* 디지털 핀: 자이로 센서 등의 장비에 응용
* 아날로그 핀: 조도, 온도, 습도 등을 측정 가능

아두이노 장치에서 사용하는 MCU는 아트멜(Atmel)에서 제작한 AVR계열로 크게 AVR, Tiny와 MEA AVR, AT90 AVR로 나뉨

주로 MEGA AVR을 이용해 아두이노 보드 제작

아두이노 보드의 메모리 구성

* EEPROM, SRAM, Flash Memory로 구성
* MCU에 따라 메모리의 크기가 다름

SRAM

* 휘발성 메모리: 전원 공급이 끊기면 데이터가 지워짐
* Flash Memory와 데이터 공유
* 아두이노 IDE를 통해 소스 코드의 각종 필요한 변수와 버퍼 등이 생성되는 공간

메모리가 한정적인 AVR 기반의 하드웨어에 크기가 큰 공격 익스플로잇 페이로드 사용 불가

* ROP(Return-oriented programming) 라이브러리 함수를 통해 추가적인 데이터를 RAM으로 주입하는 방법의 공격 사용

ROP공격 기법

* ret(리턴) 명령어로 프로그래밍을 해서 공격
* DEP(Data Execution Prevention)와 ASLR(Address Space Layout Randomization), ASCII ARMOR 와 같은 보호 방법이 적용되어 있어도 공격 가능

AVR 기반 하드웨어

* 레지스터를 통해 함수 인자 저장

= > 레지스터를 수정해 무인자동차의 자이로스코프 센서에 대한 컨트롤 가능

* 인자가 많을 경우 스택에 저장

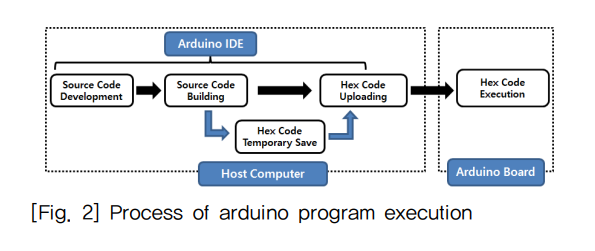
**공격자는 SRAM의 데이터를 변경해 공격!**

* 빌드 후 플래쉬 메모리에서 SRAM으로 데이터가 넘어가는 것을 이용한 코드 재사용 공격 성공 사례 有

AVR 디바이스는 자원이 한정적

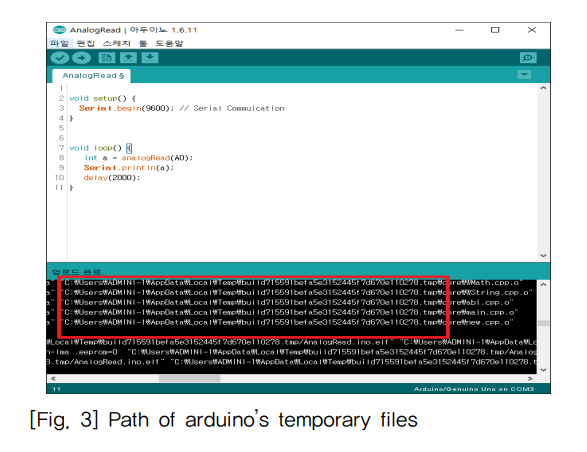
* 큰 오버헤드 초래
* 실행코드가 실행되지 못하게 막는 DEP 방어가 기본적으로 제공됨

아두이노 보드에 있는 센서를 통해 입력받은 정보가 처리되는 방식

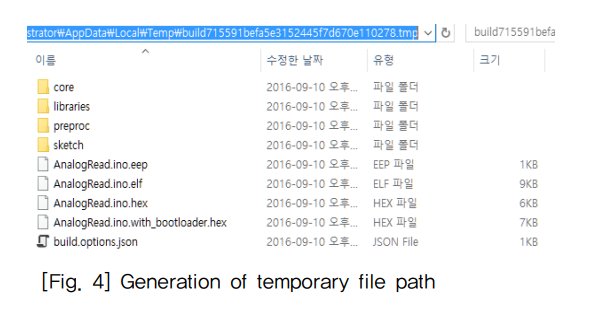


아두이노 프로그램은 위와 같은 과정을 통해 아두이노 보드에서 실행됨

아두이노 프로그램을 개발하기 위한 IDE를 설치하여 소스코드 작성 및 컴파일과 링크 등 빌드과정을 수행한다. 그리고 빌드된 실행코드는 업로드 과정을 통해 아두이노 보드에 로딩되어 실행된다. 소스코드가 호스트 컴퓨터에서 빌드되면서 실행 가능한 Hex 코드가 임시로 저장되며 이를 이용하여 업로딩이 이루어진다. Windows 환경에서 실행된 아두이 노 IDE의 경우 임시파일이 저장되는 경로는 [Fig. 3]과 같다.



Windows 운영체제의 사용자 임시 디렉터리(예: C:\ Users\사용자명\AppData\Local\Temp)에 “프로젝트 명 칭의 알파벳명칭+임시번호.tmp” 라는 이름의 디렉터리 가 임시파일이 저장되는 위치다. 아두이노의 임시파일은 프로젝트 파일 생성 위치에 상관없이 고정된다. 컴파일 시마다 임시파일은 새롭게 재생성, 수정되며 임시파일 디렉터리에 저장된 임시 Hex코드는 ICSP(In Circuit Serial Programming) 업로드 장치를 통해 직접 업로드해 도 실행이 가능하다.[16]. 생성된 임시 디렉터리에는 [Fig. 4]와 같이 core, sketch, preproc 등의 파일이 저장된다.



이와 같이 **저장되는 Hex 코드에 대한 조작을 통하여 공격자가 원하는 내용의 실행코드를 아두이노 보드에 업로드 시킬 수 있는 취약성이** 있다.

**센서 값 조작 공격 시나리오**

IDE에 소스코드 작성 및 컴파일, 링크 등 빌드 과정 수행



빌드된 실행코드가 업로드 과정을 통해 아두이노 보드에 로딩되어 실행됨



소스코드가 호스트 컴퓨터에 빌드되면서 실행 가능한 Hex 코드가 임시 저장됨



임시파일에 저장된 임시 Hex코드 확인



저장된 Hex 코드에 대한 조작

* Hex 코드 조작으로 원하는 내용의 실행코드를 아두이노 보드에 업로드(=취약성!)