< 목 차 >

1. 문제 정의

1.1 문제 정의

노트북이 도난 당했을 경우 대부분의 사람들이 **범인의 인상착의**를 알 수 없다.!!

자신의 노트북이 자신의 의지와 상관없이 팔려가는 것을 **방지**할 수는 없을까?

노트북의 경우 가격도 높고 차칫 잘못 관리할 경우 도난을 당하게 되며, 도난시 범인을 잡거나 노트북을 되찾기란 많이 힘들다는 문제점이 있다. 또한 중요 업무를 수행하는 곳에서 정보의 유출 또한 무시할 수 없는 것 중 하나이다. 본프로젝트에서는 이러한 문제점을 해결하기 위해 노트북에 설치되어있는 운영체제와 관계없이 부팅감지시 운영체제의 커널이 로드되기 전 네트워크 통신을 통해 관리자의 스마트폰으로 부팅여부를 전송하여 관리자로 하여금 부팅여부를 제어하는 것을 목적으로 한다.

1.2 사용자 요구

- Jelly Bean 4.1 버전 이상
- GNU Grub 2.0
- Ubuntu 14.04, CentOS

1.3 응용 분야 및 기여도

- 응용분야

  Booting 레벨단에서 부팅을 하기 전 Grub Mlti-bootloader 이용하여 네트워크 통신을 활용하여 노트북을 부팅전에 제어한다.

  특정 운영체제에 종속되어 동작하지 않으므로 임베디드에서 많이 사용하는 부트로더인 UBOOT 기반하에 GRUB 가 제작되었으므로 UBOOT가 작동하는 모든 임베디드 보드에서 사용이 가능

- 기여도

  기존의 많은 도난방지 프로그램은 운영체제가 Booting 완료후에 구동이 되므로 특정 운영체제에 종속되어 동작하지만 본 프로젝트의 경우 설치된 운영체제에 종속되지 않고 작동된다.

2. 목표 시스템 정의

- 입력 및 출력

2.1 Android

- 시작 버튼 클릭시 팝업 다이얼로그에서 YES 문구 선택시 Boot 신호를 중계서버에 넘김
- 시작 버튼 클릭시 팝업 다이얼로그에서 No 문구 선택시 activity 종료

- 종료 버튼 클릭시 팝업 다이얼로그에서 YES 문구 선택시 Shutdown 신호를 중계서버에 넘김
- 종료 버튼 클릭시 팝업 다이얼로그에서 No 문구 선택시 activity 종료

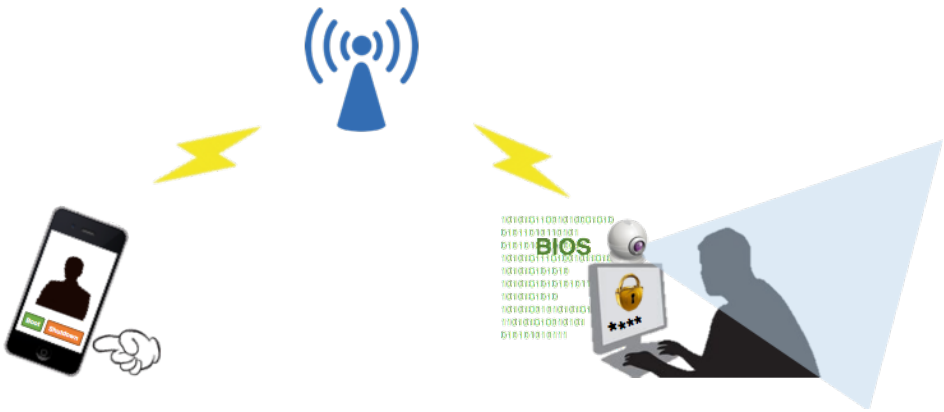
2.2 Grub (PC)

- 중계서버에서 Boot 신호가 오면 Booting 레벨에서 Grub 실행
- 중계서버에서 Shutdownt 신호가 오면 Booting 레벨에서 Grub 종료

2.3 중계서버 (CentOS)

- 클라이언트(GRUB)에서 Boot 신호값을 전달 받음
- Android의 Background Service 로 클라이언트의 부팅 신호값을 전송하여 응답값에 따라 클라이언트에게 부팅관련 제어신호 값을 전송

- 예상 사용자 인터페이스



- 제한 요건 (제약 사항들)

1) Grub에서 웹캠에 대한 모듈을 지원하지 않아 필요시 Linux의 표준 캠 디바이스 드라이버를 분석하여 GRUB에서 작동 되도록 제작을 수행해야함.
(드라이브 구현 다음 시프2때 구현 예정)
(참조 : https://github.com/torvalds/linux/tree/master/drivers/media)

2) 네트워크가 없는 경우 본 시스템을 사용 불가

3. 사용 예정인 기술과 예상 개발 방법

기존 시스템	특징	차이점
켄싱턴 락	노트북 측면이나 뒤에 고리가 있어서 여기에 케이블을 감아 도난방지	추가적으로 별도의 물품을 구입할 필요가 없으므로 비용절감
히든(hidden)	Application 영역에서의 도난방지 기능(원격 제어, 위치추적, 데이터삭제), 특정 운영체제에 종속적임	설치된 운영체제의 커널 로드되기 전에 작업이 수행되므로 특정 운영체제에 종속되어 동작하지 않음

4. 대략적인 기능명세서

- Android 기능
  1. 시작 버튼 클릭시 팝업 다이얼로그를 이용하여 사용자가 YES / NO 선택하여 컴퓨터 Boot에 대한 패킷을 전송하여 제어

  2. 종료 버튼 클리시 팝업 다이얼로그를 이용하여 사용자가 YES / NO 선택하여 컴퓨터 Shutdown에 대한 패킷을 전송하여 제어

- 중계서버 기능
  1. Android 와 Grub간의 1:N 통신 서비스 제공
  2. 자체적인 Protocol 규약을 제공

- GRUB 기능
  1. 중계서버로부터 부팅시 알림 패킷을 전송
  2. 위치를 추적할 수 있는 정보를 중계서버를 통해 Android 사용자에게 제공

5. 사용 예정인 기술과 예상 개발 방법

- Android
  1. 카카오톡과 같이 Android 시스템이 부팅될 경우 자동으로 서비스가 실행
  2. background service를 제작하여 네트워크 통신에 대한 기능을 수행
  3. 중계서버로부터 수신받은 정보를 바탕으로 지도 및 범인에 대한 정보 시각화

- Grub
  1. Grub 2.0 오픈소스 이용하여 Multi-Bootloader System 제공 및 Multi-Bootloader System 동작 구조 분석

2. iPXE 네트워크 부팅을 통한 프로그램 설치 없이 사용가능하도록 설정
3. Grub에서 모듈방식으로 작동이 되도록 별도의 확장 모듈 제작

- Server

1. 중계서버를 통한 Grub Mlti-bootloader와 사용자 스마트 폰 과 1:N 및 1:1 통신 지원
2. map + vector 자료구조를 활용하여 사용자들을 그룹단위로 관리
3. Multi-Thread 방식으로 동작하므로 각 쓰레드 간의 동기화 및 ThreadPool제작
4. 서버의 성능적 이점을 위해 Epoll 방식의 서버 제작

6. 팀 구성 및 작업 분담

구성원	작업
공통업무	문서 작성 DEBUG & Merge
김병욱	EPOLL 기반의 Multi-Thread 서버 아키텍쳐 설계 자체적인 Protocol 규약 설계 중계서버 / Android / Grub에서 통신하기 위한 통신 라이브러리 제작 Grub, iPXE 분석 Grub 네트워크 통신 관련 함수 호출 스택 분석
강호용	Android Background Service 제작 Android 부팅시 제작한 서비스가 자동 실행 Grub Custom 모듈 제작 Grub, iPXE 분석 Grub 메인 함수 호출 스택 분석

7. 개발환경

- 운영체제 : Ubuntu 14.04 , CentOS
- 중계서버 : 언어 C++(G++)/C(GCC)
- Android : Jelly Bean 4.1 버전
- 가상머신 : QEMU(qemu-system-i386)
- Mlti-bootloader : GRUB 2.0
- 개발툴 : Vim, Android Studio, StarUML

8. 진행 일정

	1	2	3	4	5	6	7	8	9	10	11	12	13
요구분석	~9.24												
설계			~10.22										
개발 및 중간 데모					~11.19								
DEBUG & Merge											~12.10		
최종 데모												~12.10	