

박재원 – 포트폴리오



Content list

- 2015 Activity(Page 3)
- 2016 Activity(Page 37)



2015 Activity List

1. 교내 정보보안 동아리 및 전국 대학 보안 연합 동아리 활동(15/03/02 ~ 16/08/01) (Page 4)
2. 한이음 빅 프로젝트 수행 (15/04/04 ~ 15/12/10) (Page 5)
3. 소프트웨어 공학 프로젝트 수행(15/05/11 ~ 15/06/10) (Page 19)
4. 컴퓨터 네트워크 프로젝트 수행(15/09/08 ~ 15/11/18) (Page 20)
5. 현장실습 참여(15/12/21 ~ 16/02/05) (Page 34)
6. 각종 코딩대회 참여 (Page 35)



1. 정보보안 동아리 활동

보안에 관심이 많아 편입한 후 보안동아리에 가입하여 활동하게 되었습니다.

가입 첫 학기는 일반회원으로 활동하며 동아리의 회원들과 친해지려고 노력했습니다. 이후 운영진으로 활동하며, 매년 교내 기술 세미나 자료 검토, HDCON 등 해킹 대회 참여, 전국 연합 동아리 주관 컨퍼런스 기획 등 다양한 활동을 진행하였습니다.

과정 중 가장 재밌었던 활동은 HDCON 해킹 대회 참여입니다. 대회는 24시간 동안 4인 1팀으로 진행하였습니다. 팀원 전원이 동아리 방에 모여 한 문제를 해결하기 위해 머리를 맞대고 고민했습니다. 각자 관심을 갖는 분야가 달랐기에, 관심 분야를 설정하고 관련 문제를 해결하는 것이 도움이 될 것 같다고 의견을 제시하였습니다. 참여한 동아리원이 제 의견에 동의해주었고, 관심분야 위주로 문제를 접근해나갔습니다.

총 3문제를 풀었지만 다음 라운드로 진출하지는 못했습니다. 이 경험을 통해 공동의 목표를 가졌을 때 각자의 관심분야가 문제 해결능력에 도움을 줄 수 있다는 것을 경험하게 되었습니다.



2. 한이음 빅 프로젝트 수행

한이음 ICT 멘토링 프로젝트에 참여한 경험이 있습니다. 처음으로 업무를 계획, 분배, 수행하며 협업을 경험했기 때문에 가장 기억에 남는 프로젝트입니다.

프로젝트 주제는 '스마트 홈 모형을 활용한 맞춤형 실내환경 서비스 구현'였습니다. 프로젝트 참여 인원은 15명이었으며, HW, 서버, SW 각 5명씩 3팀으로 나누었습니다. 저는 서버 팀에서 스마트 홈 시스템에 필요한 요소 선정, 시스템 API 설계 및 구현을 담당했습니다.

시스템 요소 선정에서 이견이 발생하였습니다. 그 이유는 SW팀에서는 보다 많은 요소를 제어 하길 원했으며, HW 팀에서는 많은 요소를 제어해야 한다는 업무 부담감 때문이었습니다. 서로 다른 생각과 업무 부담감을 해결하고자 세 번의 회의를 진행하였습니다. 첫 번째 회의에서는 SW 팀에서 제시한 시스템 요소인 LED 전등, 커튼, 가스 밸브, 콘센트가 시스템에 필요한 이유를 설명하였습니다. 두 번째 회의에서는 HW 팀이 커튼 제어 개발에 있어 어려움이 있는 이유를 설명하였습니다. 마지막으로 세 번째 회의에서 SW팀과 서버팀의 1명이 HW 팀의 커튼 제어를 도와주기로 하였습니다.

이러한 과정들을 통해 프로젝트를 성공적으로 마무리할 수 있었습니다. 이 경험으로 서로의 업무를 이해하고 소통하는 것이 협업에서 가장 중요하다는 것을 깨우칠 수 있었습니다.

한이음 결과 보고서

개요

스마트 홈은 집안의 전등, 물 온도, 가스레인지, 블라인드 등에 대한 기기의 상태를 스마트폰으로 확인할 수 있으며, 또한 이 기기들에 대한 제어를 사용자로 하여금 스마트 폰이라는 모바일 기기를 통해 장소에 상관없이 제어를 할 수 있도록 하는 시스템이다. 스마트 폰 사용자들은 스마트 폰을 통해서 자신의 상태를 한 눈에 볼 수 있으며 손쉽게 집 내부의 기기들에 접근할 수 있다.

스마트 홈 모형은 아두이노를 활용해서 조명, 가스레인지, 물 온도, 도어락과 같은 제어시스템을 형성하며 스마트 홈 시스템을 통해 제어 시스템에 포함된 모든 기기들에 대한 상태를 장소에 관계없이 확인이 가능하다. 즉, 애플리케이션 사용자들은 자신의 집 내부 상황(조명, 온도 등 기기들 상태)를 실시간으로 확인 할 수 있고 이로 인해 상황에 따른 변화에 대처를 가능하게 하여 방범기능이라는 효과를 기대할 수 있다.

스마트 홈의 원리는 집 내부에 존재하는 공유기를 통해 해당 IP를 아두이노에 할당하여 아두이노에 구성되어 있는 모터, 스위치 등의 제어 시스템의 네트워크를 제어하는 것이다. 사용자들이 스마트 홈 모바일 서비스를 통해 집 내부 기기들에 대한 조작을 하는 것은 안드로이드를 사용해서 TCP/IP를 통한 Serial 통신을 이용한 것으로 이것을 통해 제어 기기들 즉, 하드웨어를 제어 하는 방식으로 동작한다. 또한, 기기들의 상태에 대해서 하드웨어의 신호가 데이터베이스 서버에 저장되며 이 상태 값들은 안드로이드와 서버 통신을 통해 사용자에게 보여 진다. 이 어플리케이션을 통해 사용자가 가스, 온도 등을 제어 하며 상태를 변화시키게 되면 그들은 집 내부에 발생할 수 있었던 문제에 대해 미리 예방하는 것이 가능하다.

스마트 홈 애플리케이션을 사용하면 거동이 불편하거나 체력적으로 움직임이 힘든 장애인이나 노인들에게 스위치나 가스레인지와 같은 기기들에 대한 수동적 제어에 대한 어려움을 극복하게 해주며 이로 인해 더 좋은 삶의 충족을 전해 줄 수 있다. 이 프로젝트가 널리 알려진다면 사람들 대부분의 일상생활에 대한 불편이 많이 줄어들 것이라고 확신 한다. 예를 들어, 아이들만 있는 집에 가스가 켜져 있는 경우처럼 사고가 일어날 가능성이 있는 상황에서 가스 상태를 미리 확인하여 미연의 사고를 예방할 수 있을 것이다. 또한, 추가적으로 구현되는 타이머 기능을 통해 기기 작동 시간을 예약해둠으로써 나타나는 장점도 존재한다. 예를 들어, 주부들은 가스레인지로 인한 요리를 하는 상황에서 오래 불을 켜 놓았을 경우 이를 잊을 경우를 대비하여 시간을 맞춰 일정 시간이 되면 기기 작동을 자동으로 중지시켜 후에 일어나는 좋지 못한 상황을 미리 차단할 수 있다. 또 하나의 예로 이 스마트 홈은 타이머 기능을 쾌적한 아침을 맞이할 수 있는 상황을 만드는 데 사용할 수 있다. 기기 동시 제어가 가능하기 때문에 블라인드가 오픈됨과 동시에 스마트 폰에 설정해놓은 멜로디가 나오면서 방의 조명이 켜진다면 사용자가 원하는 시간에 적합한 실내 환경을 조성할 수 있다. 이 애플리케이션은 바쁘거나 혼자 사는 20~30대 현대인들에게도 긍정적인 효과를 가져 올 것이며 또한 이로 인해 분주한 아침 시간에 사람들이 직접 해야 하는 일들을 스마트 폰이 대신 함으로써 시간 절약과 함께 실내 환경을 효율적으로 관리할 수 있다.

작품 정보	
프로젝트명	스마트홈 모형을 활용한 맞춤형 실내환경 서비스 구현
작품명	알제리 (알려주면 제어하는 리모콘)
작품 소개	<p>본 프로젝트는 사용자가 어플리케이션과 지속적으로 커뮤니케이션함으로써 집 내부의 기기들을 장소에 관계없이 제어할 수 있는 시스템 개발을 목표로 한다.</p>
작품의 개발배경 및 필요성	현재 기존의 SmartHome 앱들은 집 안에서 계속해서 작동하고 있는 기기들에 대한 정보를 앱 접속을 통해 직접 확인 후 제어해야하는 방식으로 동작한다. 하지만 이 앱에 접속하는 번거로움을 줄이고 사용자가 인식하지 못한 낭비되는 기계에 대한 정보를 알림 메시지로 제공함으로서 알림에서 간편 제어가 가능하도록 한다.
작품의 특장점	사용자가 애플리케이션으로 메시지를 보내는 단방향 통신이 아닌 애플리케이션도 사용자에게 메시지를 보내는 양방향 통신이 가능하다.
작품 기능	<p>앱에 접속해서 기기를 제어함으로써 느끼는 사용자의 편리함 이전에 앱 접속의 번거로움 없이 사용자가 인식하지 못한 낭비되고 있는 기기에 대한 상태를 사용자에게 알림으로써 간편 제어를 가능하게 한다.</p> <p>또한, 사용자가 외출 시, 기기들을 한번에 off시키는 외출 모드 설정 기능과 기기 작동을 미리 예약함으로써 원하는 시간에 기기를 제어하고 동시 제어가 가능하도록 하는 타이머 설정 기능을 통해 앱의 완성도를 더욱 높인다.</p>
작품의 기대효과 및 활용분야	기존의 SmartHome 어플리케이션에 '대화형'이라는 작은 차이를 만듦으로써 사용자의 높은 앱 사용률과 기기에 대한 효율적인 접근과 제어에 대한 기대해 볼 수 있다.

I. 작품 개요

1. 작품 소개

○ 사용자의 일상에서 집 내부 상태의 확인과 전원 스위치, 가스레인지, 물 온도 설정 등 가정에서의 일들을 제어할 수 없다는 사실과 직접 집 안의 기기들에 접근해서 기기 상태들을 확인하고 제어한다는 점에 불편함을 느끼고 이것을 개선해보려고 생각하였으며, 이로 인해 사용자의 삶의 방식을 변화시키는 것이 본 프로젝트의 목표이다. 스마트 홈은 집안의 전등, 물 온도, 가스레인지, 블라인드 등에 대한 상태를 스마트폰으로 확인 할 수 있으며, 또한 이것들에 대한 조종을 사용자로 하여금 모바일을 통해 장소에 상관없이 제어를 할 수 있도록 하는 시스템이다.

스마트 홈을 목표로 한 이 프로젝트를 기술적인 면에서 한마디로 정의하면, ‘안드로이드를 이용한 아두이노 제어 방식’이라고 말할 수 있다. 즉, 안드로이드에서 스위치나 버튼리스너를 통해 서버에게 메시지를 전송하면 서버는 아두이노에게 메시지에 해당하는 기기를 제어하도록 입력 값을 넘겨준다. 입력 값을 받은 아두이노는 다시 출력 값을 서버로 전송하고, 사용자는 이 값을 안드로이드로 확인 할 수 있다.

정리해서 ‘스마트 홈’이라는 이번 프로젝트는 애플리케이션 사용자로 하여금 어디서나 집 내부의 상황을 인식하고 이를 제어할 수 있도록 함으로써 온도 조절이나 콘센트 조절을 할 수 있다는 점에서 전기 절약이 가능한 착한 시스템이며 가스레인지와 같은 사고 발생 가능성이 있는 기기들에 대한 제어 역시 가능하다는 점에서 예방 차원에서도 장점이 많은 시스템이다.

2. 작품의 개발 배경 및 필요성

○ 동기 : 대부분의 사람들은 전등을 켜고 가스레인지를 켜 놓고 하는 것을 다 수동으로 하는 것이 일반적이다. 하지만 이것들을 전부 스마트폰을 통해 장소에 불문하여 사용할 수 있다면 사람들의 삶의 질이 더욱 좋아 질것이 분명하다. 또한, 장애인이나 노인들처럼 몸이 불편하거나 체력적으로 일상생활이 힘든 경우에는 일반적인 사람들에게는 쉽고 평범한 것도 힘들어 하는 경우가 종종 있다.

실제로 외가댁에 외할머니를 종종 뵙는데 그 때마다 다리가 편찮으셔서 일어나는 것도 힘들고 스위치를 켜는 것도 힘든 경우가 많이 있었다. 이 때, 할머니를 도와드리며 생각 한 것이 만약 스마트 홈과 같이 전등과 가스레인지 등의 스위치를 스마트폰 터치 한번으로 제어가 가능하다면, 장애인, 노인뿐만 아니라 일반 사람들의 삶의 질까지 더

높일 수 있다고 생각하였으며, 필요 시 온도와 가스와 같은 집안의 상태를 외부에서도 확인을 할 수 있게 한다면 집으로 갈 수 없는 상황에서도 집안의 상황을 파악해서 화재와 같은 불의의 사고를 미리 예방이 가능 하다고 생각하여 이 시스템에 대한 필요성을 느꼈고 개발해보기로 하였다.

3. 작품의 특장점

○ 스마트 홈 애플리케이션의 기본적인 기능은 집 안에서 뿐만 아니라 외부에서도 가스레인지, 조명, 온도와 같은 기기 상태를 애플리케이션 사용자가 원할 때 모바일 서비스로 확인이 가능하며 이를 제어할 수 있도록 하는 것이다.

기존에 존재하는 스마트 홈 앱들이 이러한 기본적인 기능을 갖추고 있는 상황에서 현재 개발 중인 스마트 홈은 기능적인 면에서 차별화를 두었다. 바로 ‘타이머 기능’과 ‘알림 메시지 기능’이 그것이다.

먼저 타이머 기능에 대해서 설명하겠다. 사용자들은 자신이 제어할 수 있는 기기들에 대해 타이머 기능을 추가할 수 있고 이를 통해 기기들에 대한 동시 제어가 가능해지고 자신들의 일상 생활의 속도를 높일 수 있다.

다음으로는 알림 메시지 기능이다. 이 기능이 이 프로젝트의 가장 핵심이 되는 기능이다. 기존의 스마트 홈 애플리케이션은 기기 상태에 확인과 제어를 위해 직접 애플리케이션 내에 접속을 해야 한다. 하지만 이것은 앱 사용자들로 하여금 번거로움을 느끼게 하여 어플리케이션에 대한 의미가 없어질 수 있다. 즉, 말 그대로 의미 없이 방치되는 애플리케이션이 될 수 있다는 것이다. 접속을 통한 기기 제어가 딱히 번거로울 것이 없다고 생각할 수 있지만 실제로 주변 지인들이 사용하는 앱들을 보면 앱의 존재를 잊고 방치되고 있는 것들이 많다. 반대로 ‘카카오X’이라는 앱을 예로 들면 이 앱의 사용률은 높고 그 이유는 직접적인 접속 없이도 사용자에게 메시지가 전송됨으로써 앱에 대한 존재성을 일깨우는 것에 있다고 생각한다. 따라서 사용자의 삶의 질을 높이는 어플리케이션이 되기 이전에 메시지를 통해 주인 없는 빈 집에서 낭비되고 있는 기기 상태에 대한 정보를 메시지를 통해 알림으로써 사용자의 접근부터 친숙하게 하는 것이 프로젝트 목표이다. 즉, 사용자가 어플리케이션에게 메시지를 보내 요청하는 것뿐만 아니라 어플리케이션도 사용자에게 메시지를 보내줌으로써 양방향 소통이 되는 획기적인 어플리케이션이 완성 될 것이다.

II. 작품 내용

1. 작품 구성도

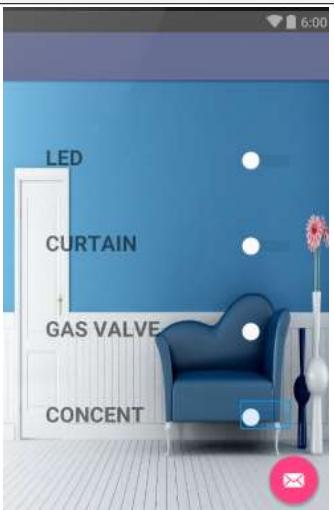


2. 작품 기능

2-1. 전체 기능 목록

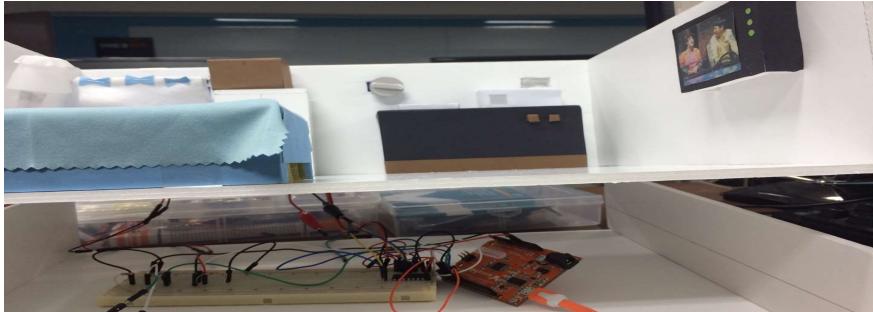
구분	기능	설명	진척도(%)
S/W	기기 제어 기능	안드로이드를 통해 아두이노 기기 제어	100%
	알림기능	사용자가 인식하지 못한 기기의 상태정보 전송	60%
	타이머 기능	사용자가 설정한 시간에 맞추어 기기 동작	70%
H/W	서버로 데이터 전송	기기 상태정보를 서버로 전송	100%
	타이머	시간 정보를 지속적으로 동작시킴으로서	70%
	인체감지 센서	일정시간동안 집안 내부에 사람 움직임이 없다면 켜져있는 기기 정보를 서버로 전송	50%

2-2. S/W 주요 기능

기능	설명	사진(스크린 캡쳐)
기기 제어	안드로이드 화면에서 스위치를 통해 아두이노 기기를 제어할 수 있는 기능이다.	
알림 기능	현재 하드웨어의 정보를 서버에 저장하고 서버에서는 정보를 사용자에게 전달할 수 있는 기능이다.	
타이머 기능	사용자가 기기를 제어할 시간을 설정하는 기능이다. 사용자가 설정한 시간에 맞춰 기기들이 동작하게된다.	

- 안드로이드 화면에서 기기 제어를 위한 스위치뿐만 아니라 외출 시 전체 스위치를 한번에 off시키는 버튼도 추가할 예정

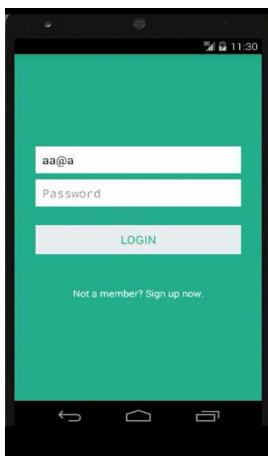
2-3. H/W 주요 기능

기능/부품	설명	사진(실사)
LED	방의 전등을 표현한다. 사용자가 전등을 켜기를 요청하면 LED가 켜지고, 방의 불을 끄기를 요청하면 LED가 꺼진다.	
LED	사용자의 방에 연결된 콘센트를 나타낸다. 사용자가 콘센트 전원을 끄기를 요청하면 LED가 꺼지고, 전원을 켜기를 요청하면 LED가 켜진다.	
서보 모터	사용자 방에 설치된 가스 벨브를 나타낸다. 사용자가 벨브를 잠그기를 요청하면 모터가 시계방향으로 90도 회전하고, 벨브를 켜기를 요청하면 반시계 방향으로 90도 회전한다.	
DC 모터	사용자 방에 설치된 커튼을 나타낸다. 사용자가 커튼을 내리기를 요청하면 DC모터가 정방향으로 회전하고, 사용자가 커튼 올리기를 요청하면 반대방향으로 회전한다.	
		

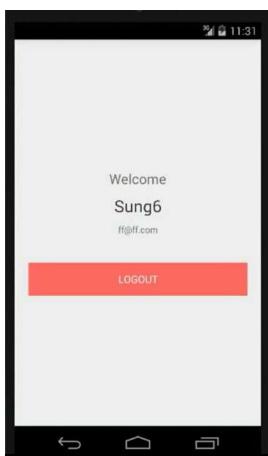
- 인체감지 센서와 RTC 센서는 코드 수정 중에 있습니다.

2-4. 사용자 인터페이스

- 등록되지 않은 사용자는 사용할 ID와 이메일 주소, 비밀번호를 먼저 회원가입을 한다.



- 회원가입을 완료한 사용자는 해당 ID와 비밀번호를 입력하여 로그인 한다.



- 사용자 확인이 되면 해당 앱으로 접속이 승인된다.



- 사용자는 해당 스위치를 통해 집 내부 기기를 제어할 수 있다.

3. 중요 알고리즘 및 적용 기술

- S/W

http 통신 : 안드로이드에서 사용자가 스위치나 버튼을 동작시켰을 시, 선택한 스위치 값들을 WAS 서버로 전송하는 통신 기술. 사용자가 변경한 스위치 값을 기기명과 함께 WAS 서버로 전송한다. 사용자가 앱을 접속하였을 때 서버에서 데이터베이스에 저장된 기기명과 기기상태를 불러와 안드로이드로 전송한다.

- H/W

시리얼 USB 통신 : 아두이노와 WAS 서버 사이의 통신 기술. 서버에서 아두이노에게 시리얼(serial)통신으로 사용자가 변경시키길 원하는 기기명과 기기상태의 값을 전송하면 아두이노는 이 값에 따라 기기 상태를 변경시키고 변경이 완료되면 다시 시리얼 통신으로 서버에게 기기명과 변경된 기기 상태를 전송한다. 서버는 전송 받은 값을 데이터베이스에 저장하여 사용자가 앱에 접속하였을 때 현재 기기 상태 값을 불러 올수 있도록 한다.

4. 작품 개발 환경

구분	항목	적용내역
S/W 개발환경	OS	안드로이드 안드로이드 어플리케이션 개발
	개발환경(IDE)	sdk API 22: Android 5.1 (Lollipop)
	개발도구	안드로이드스튜디오 화면 UI를 구성하고 서버로 데이터 값을 전송하는 데 사용되는 개발 도구이다.
	개발언어	자바 - .xml에 구성 된 스위치나 버튼의 값을 동작시키는 데 사용되는 언어 jsp - 안드로이드에서 받은 데이터 값을 아두이노에 전송할 때 사용하는 언어 xml - 버튼이나 스위치 같은 객체들을 안드로이드 화면상에 구성할 때 사용하는 언어
H/W 구성장비	디바이스	아두이노 컴퓨터에 USB케이블을 연결하여 전원을 공급하면 전자 소자에 입/출력 신호를 주는 하드웨어이다.
	센서	적외선 인체 감지 센서 근거리에 있는 움직임을 감지하여 현재 집 내부에 사용자가 있는지 확인하는 센서이다.
	통신	시리얼 USB 통신 웹 서버와 통신하기 위한 수단이다. 디바이스에서 서버에 기기 상태들을 시리얼 통신으로 전달한다.
	개발언어	C, C++ 아두이노 개발 환경을 C, C++ 언어를 사용하여 소스 코드를 작성한다.

5. H/W 내역

5-1. H/W 부품 내역

부품명	설명	작품에서의 기능
LED	사용자에 요청에 따라 LED를 켜고 끈다.	방 전등
LED	사용자에 요청에 따라 LED를 켜고 끈다.	콘센트
서보 모터	사용자에 요청에 따라 모터를 제어한다.	가스 벨브
DC 모터	사용자에 요청에 따라 모터를 제어한다.	커튼
RTC 모듈	사용자에 요청한 시간과 현재시간이 일치하는지 확인한다.	시간 예약
인체감지 센서	집안 내부에 있는 사람의 존재 여부를 확인한다.	알림 메시지 전송 여부 결정

III. 프로젝트 수행 내용

1. 프로젝트 수행 일정

구분	추진내용	수행기간										
		1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월
계획	소프트웨어 & 서버 & 하드웨어 팀으로 나눠서 팀마다 역할 분담. 전반적으로 할 일, 일정표 등 작성해서 팀원들에게 배포											
분석 & 설계	유저 시나리오 작성											
	하드웨어 필요 장비 목록 작성											
	서버 통신 방법 선정											
개발	안드로이드 UI구성											
	안드로이드 - 서버 통신											
	서버 - 아두이노 통신											
테스트	안드로이드 - 서버 - 아두이노 통신 검사											
종료												

3. 문제점 및 해결방안

3-1. 프로젝트 관리 측면

○ 기획 부분에서 확실하게 어떤 순서로 진행할지 팀원들과 결정하지 않아 기획을 끝내고 개발을 시작하기까지 어려움이 있었다. 각자 떨어져있음에도 서로 적극적으로 의사소통을 하려 하지 않아 프로젝트가 오래 정체되었다. 다행이도 멘토님께서 팀장에게 연락을 취해 팀원들과 소통의장을 만들어주셨고, 다시 한 번 프로젝트 완성에 대한 결의를 다지면서 프로젝트를 수월하게 진행할 수 있었다.

3-2. 작품 개발 측면

○ 프로젝트가 안드로이드 - 웹 서버 - 아두이노를 통신하는 게 중점 인만큼 개발

에 있어서 통신하는 거에 집중하였다. 그 중 안드로이드와 서버를 http 통신을 하는데 있어 개발환경의 SDK가 개발자마다 달라 파일이 호환되지 않는 점에서 어려움이 있었다.

그리고 http통신에서 사용자는 라이브러리가 특정 버전 이후에는 호환이 되지 않아 호환성을 맞추는 데에도 많은 시간이 걸렸다.

오류 메시지를 인터넷을 통해 검색하여 계속해서 수정하며 시도한 끝에 기기(스마트폰)와 호환시켰다. 그 후 팀원들과 호환이 되는 개발환경의 SDK 지정하여 서로 개발한 파일을 주고받았을 때 문제가 되지 않도록 하였다.

4. 프로젝트를 통해 배우거나 느낀점

○ 한이음 Big 프로젝트를 통해서 가장 크게 배웠다고 느끼는 부분은 협업이다. 프로젝트를 진행하기 전에는 ‘많은 사람들과 같이하니깐 개발 중에 어려움을 겪어도 같이 해결해 나갈 수 있겠지.’ ‘각자 맡은 일만 잘하면 문제 없겠지’라는 생각으로 빅 프로젝트를 시작했는데 개발하면서 느끼는 점은 ‘내가 협업과 분업을 구분하지 않고 있었구나.’였다. 협업 안에서 분업을 할 순 있지만 절대 분업만으로 협업을 할 순 없었다. 나 혼자 하는게 아니기 때문에 팀원들과 계속해서 의사소통해야하고, 서로 동상이몽에 빠지지 않게 노력해야했다.

IV. 작품의 기대효과 및 활용분야

○ 바일로 컨트롤 한다는 부분에서 몸이 불편하거나 움직이기 힘든 노약자들도 집 안에서 쉽게 기기들에 접근할 수 있게 하여 일상생활에서의 어려움을 덜어준다. 또한 바쁜 현대인들에게 모바일을 통해 집 내부의 상태를 확인할 수 있도록 하여 방범이나 전력 낭비를 막을 수 있도록 도와준다.

또한 사용자가 집안 내부를 일일이 확인하지 않아도 어플리케이션이 불필요하게 소비되고 있는 기기 정보를 전달함으로써 어플리케이션 사용 도를 더 높였다.

ICT멘토링 수행확인서

<프로젝트명 : [B15-318] 스마트홈 모형을 활용한 맞춤형 실내환경
서비스 구현 - 서버>

<수행 기간 : 2015-04-04~2015-12-10>

<멘토 : 은석훈>

<회사명 : 삼성SDS>

<대학 : 한국산업기술대학교>

<멘티 : 박재원>



미래창조과학부가 주최하고 정보통신기술진흥센터가 주관하는
ICT멘토링 프로젝트에 참여하였음을 확인합니다.

<2016.04.04>

정보통신기술진흥센터장





3. SE 과목 프로젝트 수행

소프트웨어 공학 과목을 청강하며 IoT 기반 화분관리 시스템 프로젝트를 구현한 경험이 있습니다.

소프트웨어공학 교과목에서 폭포수 모델을 기반으로 IoT 기반 집 안의 화분 관리 프로젝트 설계를 맡은 경험이 있습니다.

프로젝트 수행일정에 맞춰 진행 중이었으나, 최종 발표일이 1주일 당겨져서 수행일정을 모두 수정해야 하는 경우가 발생하였습니다. 팀원 간 회의에서 기존 일정대로 진행하게 되면 발표를 할 수 없는 상황이라는 것을 인지시킨 뒤, 진화 모델로 바꿔서 진행하는 것이 일정상 유리하다고 이야기하였습니다. 그러나 저 포함 팀원 모두가 진화 모델을 이용해 프로젝트를 진행해본 적이 없어 진화 모델을 선택하는 데 어려움이 있었습니다. 팀원들에게 차분하게 설득하여 진화모델을 택했습니다. 처음에는 팀원 모두 진행하는 데 어려움을 겪었습니다. 그러나 제가 제안한 만큼 다른 팀원들이 어떤 일을 진행해야 하는지 함께 고민하고, 자료수집이나 조언을 구하며 팀원들을 도와 각자 맡은 일을 일정에 지연 없이 진행되도록 도왔습니다.

결과적으로 프로젝트는 최종 발표에서 만점을 받으며, 성공적으로 마무리할 수 있었습니다. 이번 경험을 통하여 프로젝트를 진행할 때 한 가지 방식으로 편향되어 진행하지 말고 여러 가지 방법을 조사하여 가장 최적의 방법을 채택하는 법을 배울 수 있었습니다.



4. 컴퓨터 네트워크 프로젝트 수행

컴퓨터 네트워크 교과목을 청강하던 중 패킷 스니핑 툴을 개발한 경험이 있습니다.

교과목에서 패킷 구조에 대한 강조를 많이 했고, 구조 분석을 할 수 있는 툴을 만들어보고자하는 목적으로 개발하였습니다.

프로젝트 수행 중 제가 맡은 역할은 프로그램 구현과 환경 구축이었습니다. 프로그램 구현과정에서 Kali Linux에서 패킷 캡쳐 과정에 어려움이 있었습니다. 이 문제점을 해결하고자 pcap lib 설치를 통하여 만들어진 패킷 캡쳐 함수를 사용하여 구현하였습니다.

이러한 과정을 통하여 Linux 네트워크 환경 구축과 명령어를 익힐 수 있었고, Windows보다 커스텀마이징하기 편리하다는 점을 알 수 있었습니다.

2011151011 | 2012154037 | 2013152048 | 2013150044
나 승 환 | 이 진 우 | 박 재 원 | 곽 다 인

Packet Sniffing 구현

목차

- 목표
 - 설계 환경
 - 설계 내용
 - 결과
 - 기대 효과
-

목표

- ✓ Packet Sniffing 프로그램 구현
 - Wireshark에서 캡쳐되는 패킷 내용 중 일부분 출력
 - 각 계층 Header Size
 - 2계층 (출발/목적지 Mac Address)
 - 3계층 (출발/목적지 IP, Type, ID와 Length)
 - 4계층 (송/수신지 포트번호, Seq #, Ack #, Flags)
 - App data와 App data size 정보 출력
- ✓ TCP기반의 패킷 정보 확인 및 TCP IP 3way handshaking 확인하기 (http port 80)
- ✓ UDP기반의 패킷 정보 확인 (Nslookup 사용, Dns port 53)
- ✓ Wireshark와 Packet sniff Program 패킷 비교분석

설계 환경

- ✓ Kali Linux (Debian 계열) + Raw socket
- ✓ Linux Version 3.18.0-kali3
- ✓ Gcc Version 4.7.2(Debian 4.7.2.-5)
- ✓ Debian 3.18.6-1~kali2(2015-03-02)
- ✓ 패킷 캡쳐를 위한 pcap 라이브러리 사용
(`sudo apt-get install libpcap-dev` 명령어로 설치 후 사용)

설계내용

컴파일 화면

The screenshot shows a Kali Linux desktop environment within a VMware Workstation window. The desktop has a dark theme with icons for Home, Computer, Trash, and two folders named 'Packet_property' and 'tools'. A terminal window titled 'root@kali: ~/Desktop/tools' is open, displaying the following command sequence:

```
root@kali:~# ls
data Desktop Documents Downloads Music Pictures Public Templates Videos
root@kali:~# cd Desktop
root@kali:~/Desktop# ls
dump.h~ libnet_test.c~ packet~ Packet_property Syn_Flood.c~ test.c~ tools
root@kali:~/Desktop# cd tools
root@kali:~/Desktop/tools# cd
root@kali:~/Desktop/tools# ls
decode_sniff decode_sniff_update.c password_list
decode_sniff.c decode_sniff_update.c~ temp_header
decode_sniff_update mimikatz wce v1.41beta_universal
root@kali:~/Desktop/tools# gcc decode_sniff_update.c -o sniff_update -lpcap -w
root@kali:~/Desktop/tools#
```

The last command, `gcc decode_sniff_update.c -o sniff_update -lpcap -w`, is highlighted with a red rectangle.

At the bottom of the terminal window, there is a message: "To direct input to this VM, move the mouse pointer inside or press Ctrl+G."

Gcc decode_sniff_update.c -o sniff_update -lpcap -w

설계내용

Host IP로 스니핑하는 법

Kali-Linux-1.1.0c-vm-486 - VMware Workstation

File Edit View VM Tabs Help | Home Kali-Linux-1.1.0c-vm-486 Applications Places 일 11월 29, 9:47 오후 root@kali: ~/Desktop/tools

Code::Blocks IDE
Configurable and extensible IDE

```
root@kali:~/Desktop/tools# clear
root@kali:~/Desktop/tools# ls
decode_sniff decode_sniff_update decode_sniff_update.c~ password_list temp_header
decode_sniff.c decode_sniff_update.c mimikatz sniff_update wce_v1_41beta_universal
root@kali:~/Desktop/tools# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:bd:24:08
          inet addr: 192.168.79.131 Bcast:192.168.79.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:febd:2408/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:62 errors:0 dropped:0 overruns:0 frame:0
            TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:7379 (7.2 KiB) TX bytes:5513 (5.3 KiB)
            Interrupt:19 Base address:0x2000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:12 errors:0 dropped:0 overruns:0 frame:0
            TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:720 (720.0 B) TX bytes:720 (720.0 B)

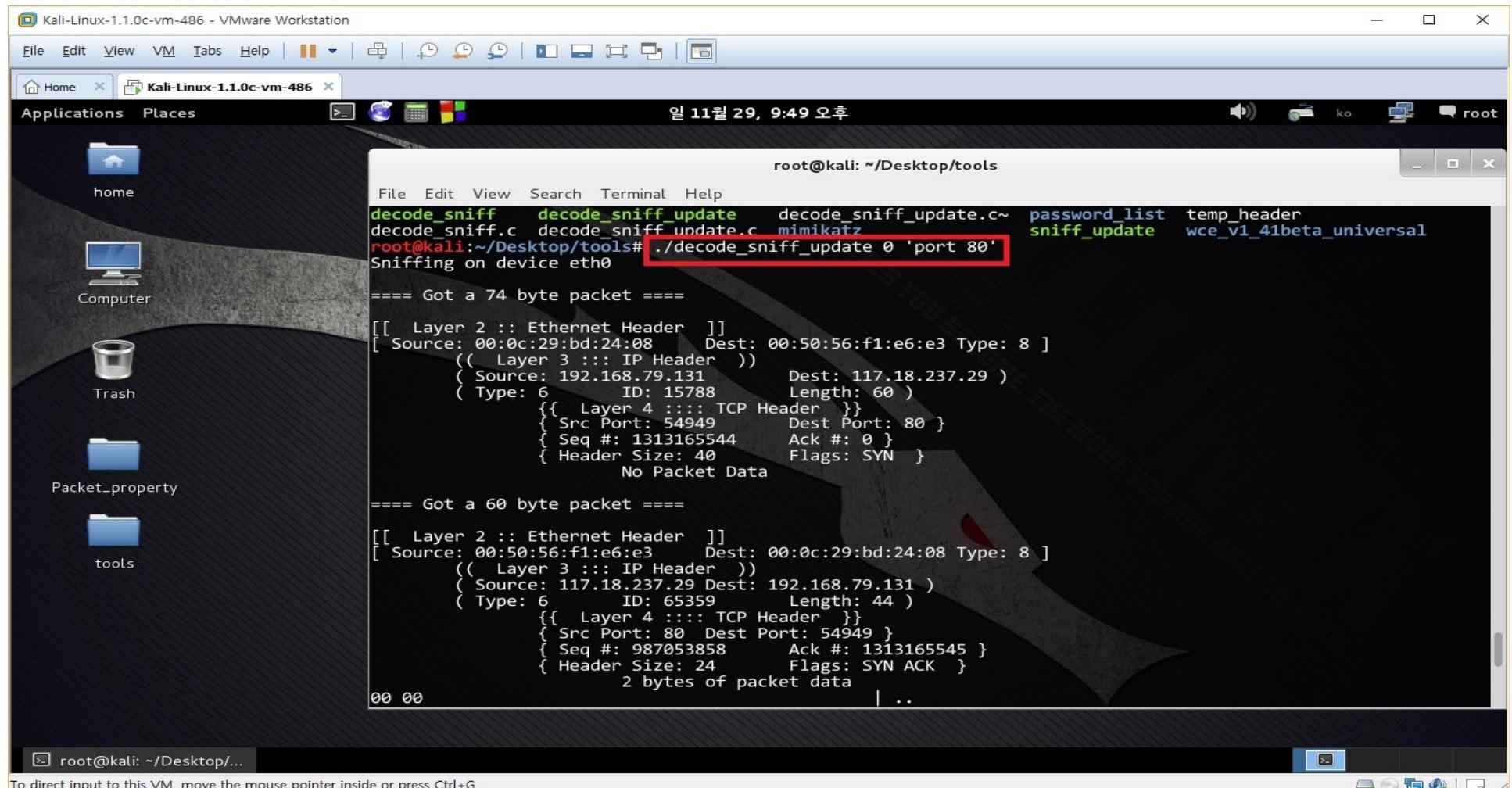
root@kali:~/Desktop/tools# ./decode_sniff_update 0 'host 192.168.79.131'
Sniffing on device eth0
===== Got a 72 byte packet =====
[[ Layer 2 :: Ethernet Header ]]
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

– 호스트 IP를 확인 후 스니핑 대상을 선정

설계내용

Port number로 스니핑 하는 법



The screenshot shows a terminal window on a Kali Linux desktop. The terminal output is as follows:

```
root@kali: ~/Desktop/tools
decode_sniff decode_sniff_update decode_sniff_update.c~ password_list temp_header
decode_sniff.c decode_sniff_update.c mimikatz sniff_update wce_v1_41beta_universal
root@kali:~/Desktop/tools# ./decode_sniff_update 0 'port 80'
Sniffing on device eth0

===== Got a 74 byte packet =====

[[ Layer 2 :: Ethernet Header ]]
[ Source: 00:0c:29:bd:24:08 Dest: 00:50:56:f1:e6:e3 Type: 8 ]
  (( Layer 3 :: IP Header ))
    ( Source: 192.168.79.131 Dest: 117.18.237.29 )
    ( Type: 6 ID: 15788 Length: 60 )
      {{ Layer 4 :::: TCP Header }}
        { Src Port: 54949 Dest Port: 80 }
        { Seq #: 1313165544 Ack #: 0 }
        { Header Size: 40 Flags: SYN }
          No Packet Data

===== Got a 60 byte packet =====

[[ Layer 2 :: Ethernet Header ]]
[ Source: 00:50:56:f1:e6:e3 Dest: 00:0c:29:bd:24:08 Type: 8 ]
  (( Layer 3 :: IP Header ))
    ( Source: 117.18.237.29 Dest: 192.168.79.131 )
    ( Type: 6 ID: 65359 Length: 44 )
      {{ Layer 4 :::: TCP Header }}
        { Src Port: 80 Dest Port: 54949 }
        { Seq #: 987053858 Ack #: 1313165545 }
        { Header Size: 24 Flags: SYN ACK }
          2 bytes of packet data
00 00 | ..
```

The command entered in the terminal is `./decode_sniff_update 0 'port 80'`. The output shows two captured network packets. The first packet is a SYN packet from source port 54949 to destination port 80. The second packet is a SYN-ACK response from source port 80 to destination port 54949.

– 스니핑 대상을 80번 Port로 설정

설계내용

TCP IP 3way handshaking

```
===== Got a 74 byte packet =====
[[ Layer 2 :: Ethernet Header ]]
[ Source: 00:0c:29:bd:24:08 Dest: 00:50:56:f1:e6:e3 Type: 8 ]
  (( Layer 3 :::: IP Header ))
    ( Source: 192.168.79.131 Dest: 117.18.237.29 )
    ( Type: 6 ID: 15788 Length: 60 )
      {{ Layer 4 ::::: TCP Header }}
        { Src Port: 54949 Dest Port: 80 }
        { Seq #: 1313165544 Ack #: 0 }
        { Header Size: 40 Flags: SYN }
          No Packet Data

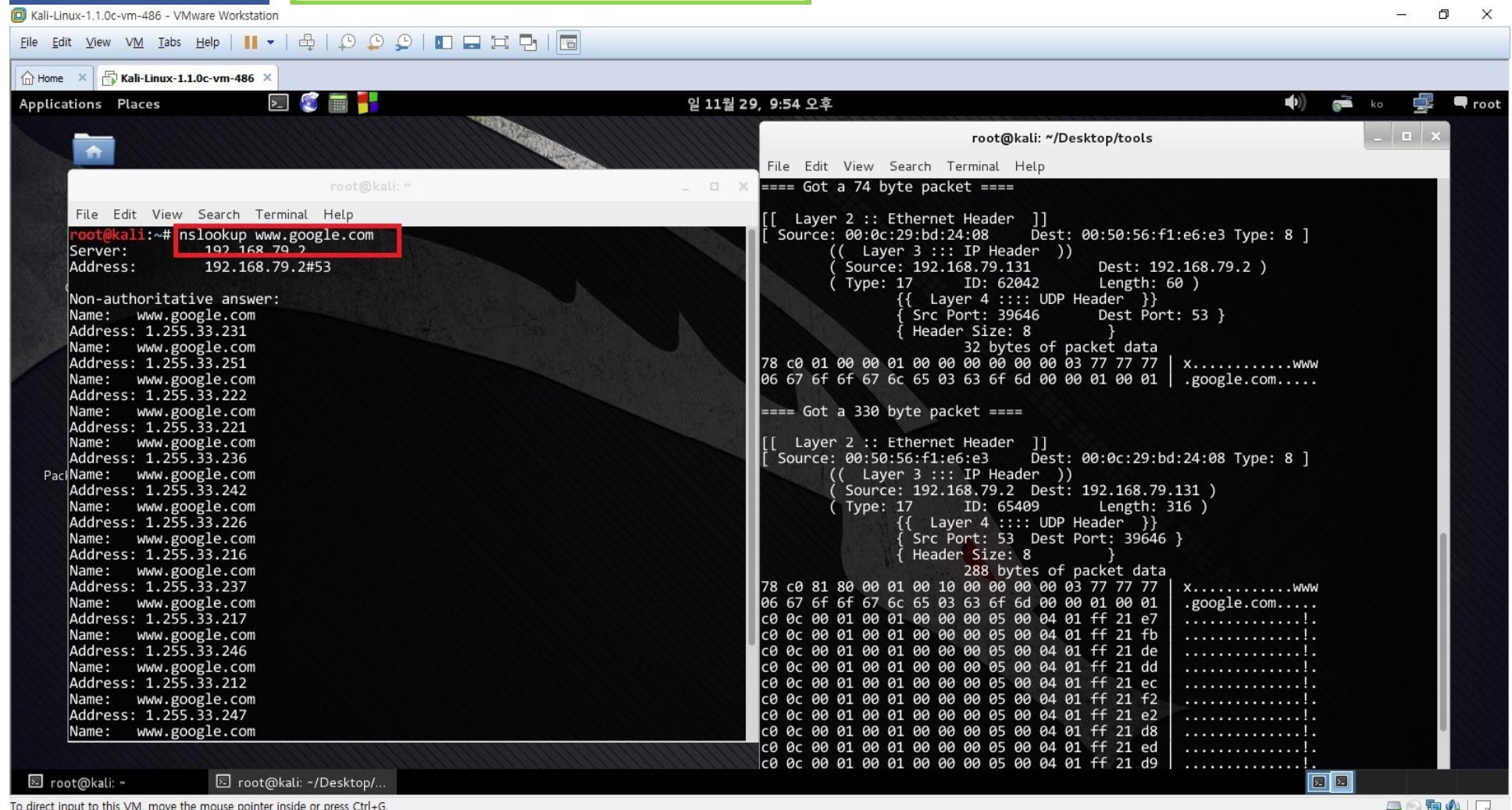
===== Got a 60 byte packet =====
[[ Layer 2 :: Ethernet Header ]]
[ Source: 00:50:56:f1:e6:e3 Dest: 00:0c:29:bd:24:08 Type: 8 ]
  (( Layer 3 :::: IP Header ))
    ( Source: 117.18.237.29 Dest: 192.168.79.131 )
    ( Type: 6 ID: 65359 Length: 44 )
      {{ Layer 4 ::::: TCP Header }}
        { Src Port: 80 Dest Port: 54949 }
        { Seq #: 987053858 Ack #: 1313165545 }
        { Header Size: 24 Flags: SYN ACK }
          2 bytes of packet data | ...
00 00

===== Got a 54 byte packet =====
[[ Layer 2 :: Ethernet Header ]]
[ Source: 00:0c:29:bd:24:08 Dest: 00:50:56:f1:e6:e3 Type: 8 ]
  (( Layer 3 :::: IP Header ))
    ( Source: 192.168.79.131 Dest: 117.18.237.29 )
    ( Type: 6 ID: 15789 Length: 40 )
      {{ Layer 4 ::::: TCP Header }}
        { Src Port: 54949 Dest Port: 80 }
        { Seq #: 1313165545 Ack #: 987053859 }
        { Header Size: 20 Flags: ACK }
          No Packet Data
```

Flags 정보로부터 3-way임을 알 수 있다.

설계내용

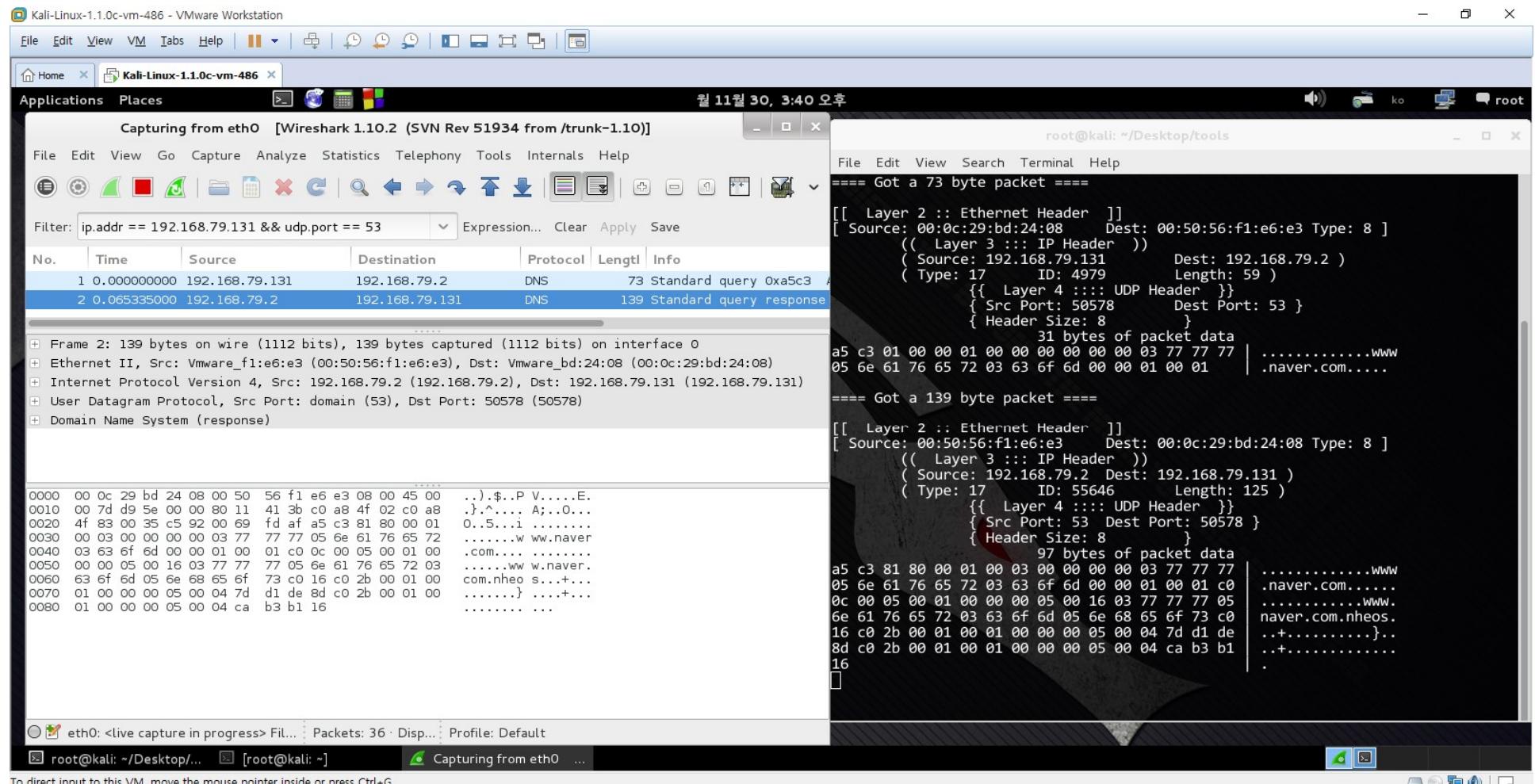
UDP Packet Sniffing



Nslookup에 따른 UDP Packet 캡처

설계내용

WireShark와 Packet sniff Program 캡쳐 정보 비교



- 헤더 정보들을 제외한 나머지 정보들이 같음을 알 수 있다.

결과

- ✓ 각 계층별로 요약된 정보를 확인 할 수 있음
- ✓ UDP와 TCP에서의 패킷 정보를 확인할 수 있음
- ✓ Wireshark와 Packet sniff Program이 스니핑한 패킷 정보가 같다는 것을 알 수 있음.

기대 효과

- ✓ 실시간으로 Packet 캡쳐
- ✓ 스니핑 대상은 현재 실행하는 컴퓨터가 속해있는 네트워크 내 모든 컴퓨터가 대상이 될 수 있음.
- ✓ 특정 호스트 및 특정 포트에 대한 패킷 필터링 가능
- ✓ TCP의 경우 Flags를 바탕으로 Packet 형태, 상태 등을 알 수 있음

감사합니다



5. 현장실습 참여

15년 12월부터 16년 2월까지 기업과 연계하여 현장실습을 진행한 경험이 있습니다.

교내에서 배운 전공을 활용해보고 싶어 기업 매칭 과정에서 전공을 활용할 수 있는 기업을 선호하였고, 그 결과 인터넷 방송 플랫폼 제작, 서비스하는 Emotion Wave에 매칭되었습니다.

기업에서 방송 서비스 사용자들의 트래픽 사용량, 시스템 점유율을 볼 수 있는 모니터링 모듈 개발 업무를 수행하였습니다.

우선 서버의 로그 정보를 통해 전반적인 서비스 사용량을 분석하고자 했습니다. 그러나 기업 측에서 정보유출 문제 때문에 로그 정보를 지원해주지 않았고, 저는 개인 PC에 서버와 유사한 환경을 구축하여 개발을 진행하게 되었습니다. 개인 서버의 HW 성능 문제로 인해 방송 방에 동시 접속자가 10명이 초과하게 되면 네트워크 대역폭에 문제가 생긴다는 점을 알게 되었습니다. 이 문제를 해결하고 싶었지만, HW 성능상 문제였기 때문에 네트워크 카드를 증축하지 않는 이상 해결이 어려웠습니다. 그래서 저는 네트워크 대역폭의 80%이상을 사용하게 되면 모니터링 시스템에 알람을 주는 방식으로 구현하였습니다.

예상치 못한 상황에서 냉정하게 할 수 있는 것들을 선별해 진행할 수 있었습니다. CentOS, Maria DB 등 다양한 지식을 습득할 수 있었고, 특히 서버를 Linux로 구현하다보니 Linux 명령어들을 많이 익힐 수 있었습니다.



6. 코딩 대회 참여

삼성 대학생 프로그래밍 경진대회, LG 코드 챌린저, 교내 프로그래밍 대회에 참여한 경험이 있습니다.

코딩 대회 참가 계기는 학교에서 배운 프로그래밍 과목을 상황에 맞춰 응용할 수 있는지 알아보고 싶었습니다.

삼성 대회의 경우 10개의 문제 중 5개를 풀었고, LG 대회의 경우 봤던 4문제 중 3문제를 풀었습니다. 풀이과정에서 인터넷을 참조할 수 있어, 풀이에 도움이 많이 되었던 것 같습니다. 또한 삼성의 경우 제약조건이 상당히 까다로웠고 LG의 경우 제약조건이 없어 쉽게 풀 수 있었습니다. 결과는 두 대회 모두 2라운드에 진출하지 못했습니다. 이를 통해 저는 아직 주어진 상황에 맞춰 알고리즘을 제작하고 코딩으로 구현하는 실력이 부족하다는 것을 느끼게 되었고, 시간이 생길 때 마다 코딩 관련 서적과 정보 검색을 통하여 공부하고자 노력하고 있습니다.

제 2015-0028 호

장려상

팀명: 프로그램못짜요
성명: 나승환
성명: 박재원

위 학생은 한국산업기술대학교 총장배
제10회 KPU 소프트웨어 경진대회에서 우수한
성적을 거두었으므로 이 상장을 수여함.



2015년 11월 27일

한국산업기술대학교 총장 이재훈





2016 Activity List

1. 가상화 클라우드 환경의 모니터링 시스템 구현(16/01/02 ~ 16/08/20) (Page 38)
2. 2016 Imagine cup 참여(16/01/18 ~ 16/02/15) (Page 51)
3. 일본 기업 탐방(16/02/22 ~ 16/02/25) (Page 67)
4. 네트워크 매니지먼트 교과목 조교 활동(16/03/02 ~ 16/06/20) (Page 68)
5. 개발자 컨퍼런스, 코딩대회 참여(Page 69)

) 1. 가상화 클라우드 모니터링 시스템

지속적인 클라우드 기술 이슈에 맞춰 효율적인 클라우드 시스템 구축, 관리를 위해 가상화 기술을 사용하게 되었고, 모니터링 시스템을 개발하게 되었습니다. 모니터링 시스템의 목표는 클라우드 시스템 내 VM들의 시스템 정보를 Web Dashboard를 통해 원격에서 볼 수 있도록 하는 것입니다. 3명으로 구성된 팀으로써 DB 설계, 구현, Web 설계를 담당했습니다.

프로젝트 진행 과정에서 팀원과 연락이 되지 않았고, 지속적으로 연락을 시도해서 연락할 수 있었습니다. 그러나 팀을 나가겠다는 이야기와 함께 다시 연락이 되지 않았고, 저는 프로젝트 진행 속도에 문제가 생기지 않게 빠르게 역할 분담을 다시 구성하였습니다. 역할 분담 후 업무 양이 많았지만, 많은 회의, 대화를 통해 팀원을 다독이며 진행하였습니다. 진행 과정에서 모니터링 요소 선정 과정에서 사용자가 서비스를 이용하는 샘플 데이터를 만들었고, 이를 토대로 모의 환경에서 테스트를 하며 데이터 분석을 진행하였습니다. 이 과정을 통해 CPU, RAM, 네트워크 대역폭, 서비스 로그 정보가 중요하다는 것을 알게 되었고, Python, SQL을 통하여 각 VM에 데몬을 구현하였습니다.

남은 팀원과 서로 의지하며 프로젝트를 마무리 할 수 있었습니다. 이 프로젝트를 통해 팀원 간 소통이 프로젝트 완성도, 결과에 많은 영향을 끼친다는 것을 느꼈습니다.

가상화 클라우드 모니터링 시스템



컴퓨터공학과 김승연
컴퓨터공학과 박재원

1

개요

연구 배경

- 클라우드 가상화 기술이 비즈니스 효율성을 높이는데 큰 역할을 한다는 의견이 많아지면서 이를 이용한 혁신의 필요성 대두
- 그와 함께 클라우드 시스템 관리에 관한 기술이 이슈화
- 시대적인 분위기에 따라 가상화 환경을 모니터링 하는 시스템의 개발은 필수적임

1

개요

연구 개발 목표

- 클라우드 시스템에서 가상화된 guest들은 시스템의 컴포넌트 기능에 관련된 log들을 선별하여 DB server에 저장 후 분석
- 클라우드 시스템으로 오고 가는 패킷을 캡쳐하고 이를 DB server에 저장
- 관리자는 저장된 정보를 바탕으로 가상화 환경의 자원들의 정보 모니터링

1

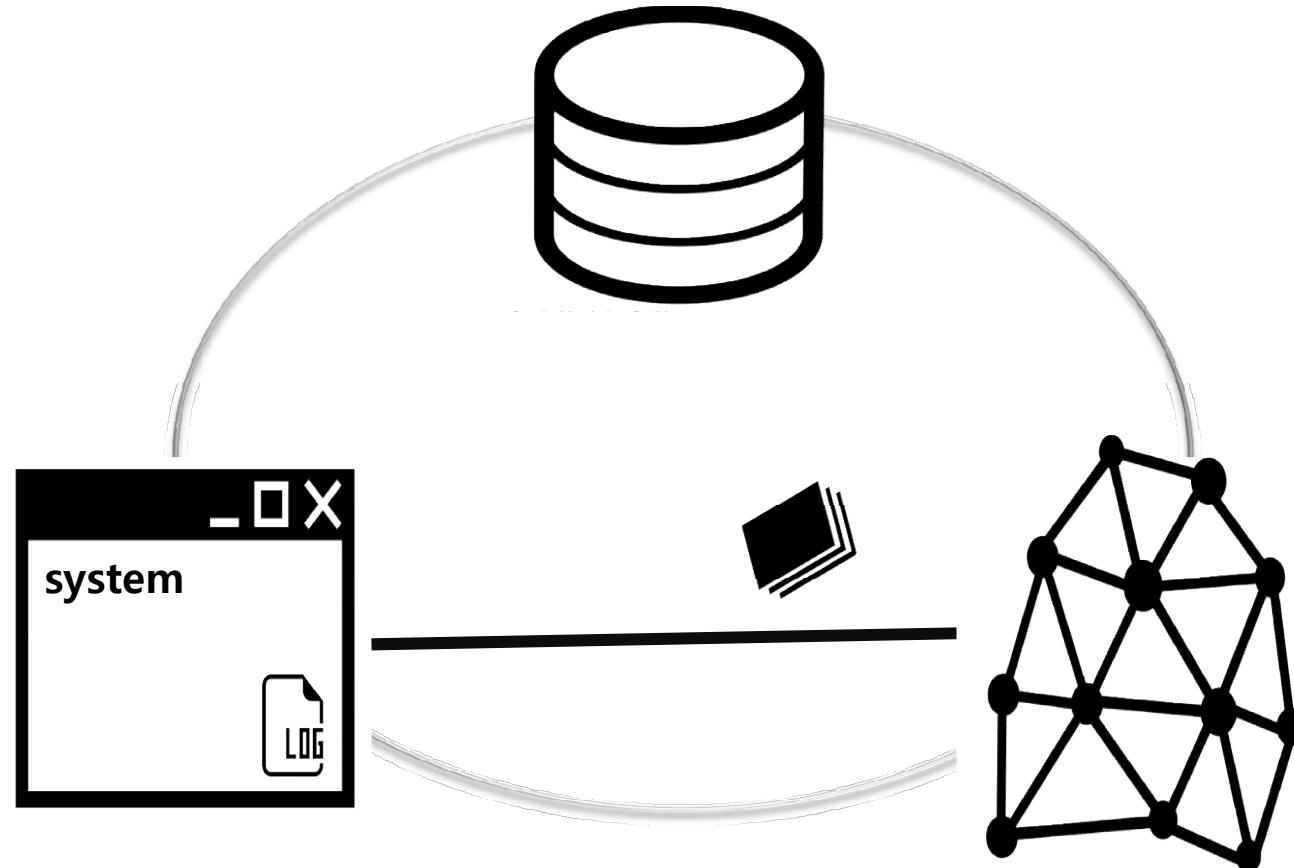
개요

연구 개발 효과

- 네트워크 패킷의 내용을 파악
- 패킷 분석을 통하여 어떠한 사용자가 어떠한 기능을 자주 사용하는지
파악 가능
- 네트워크 패킷을 통해서 트래픽 양을 알 수 있음
- Log 기록으로 일반적인 패턴을 분석 가능

2

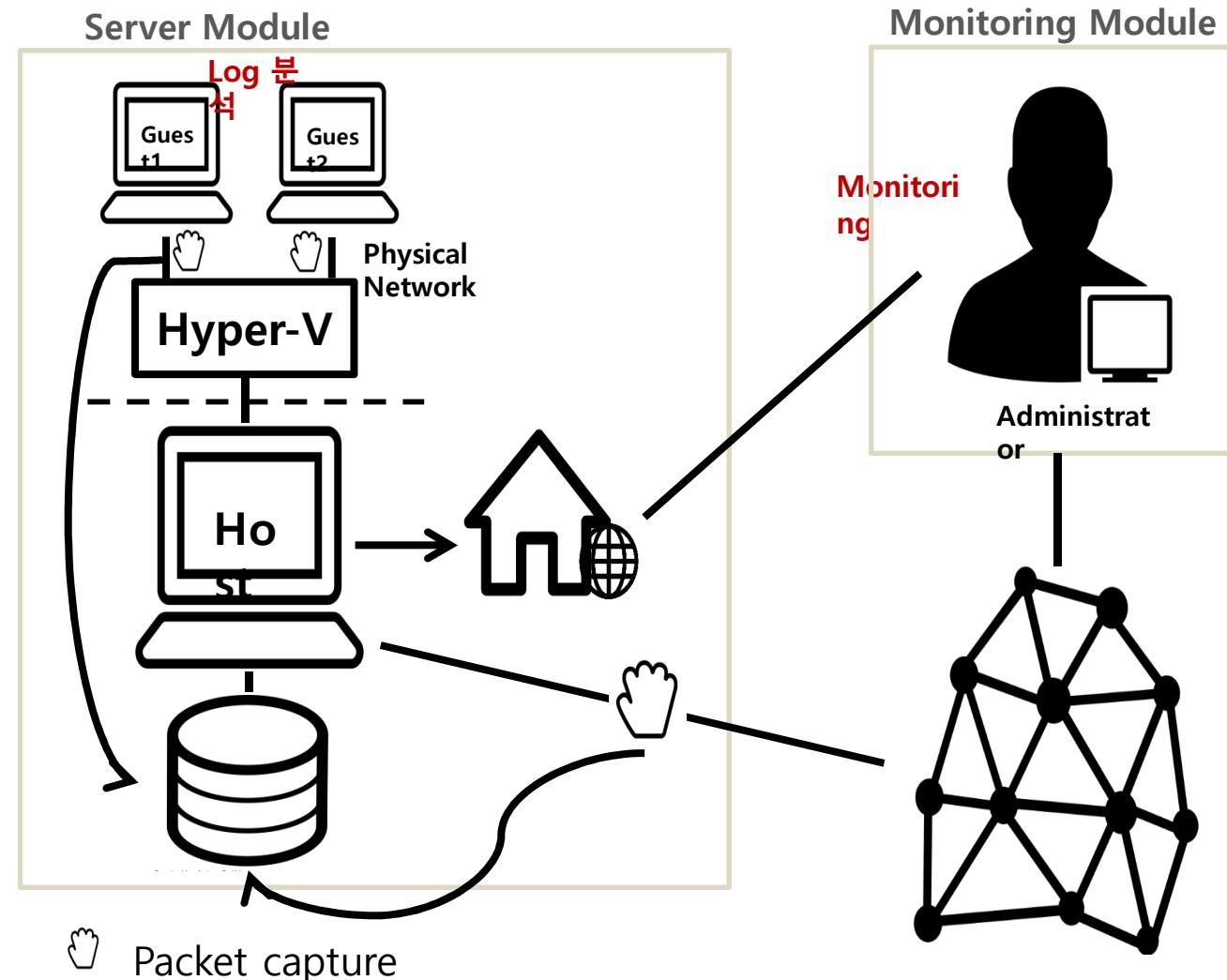
시스템 수행 시나리오



3

시스템 구성도

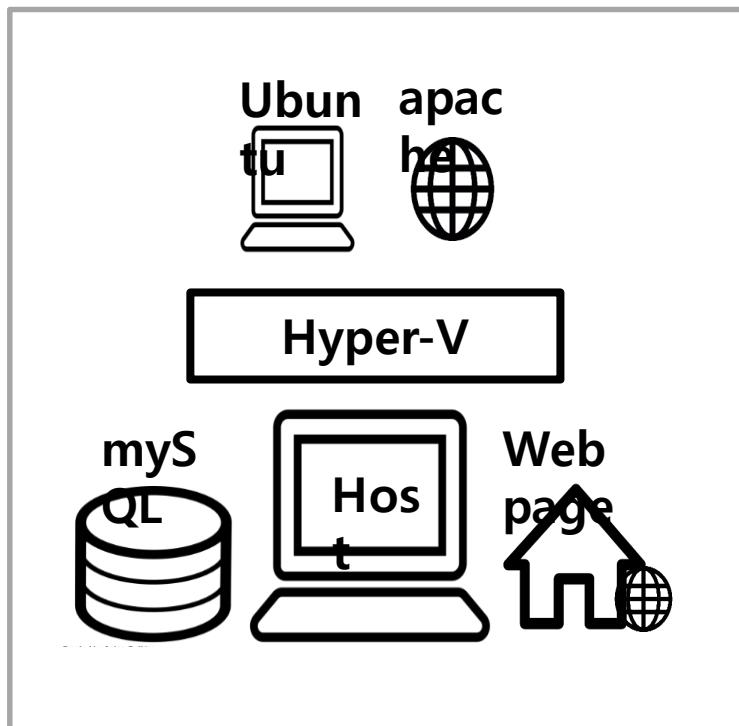
도식화



4

시스템 모듈 상세 설계

Network – VM1(Ubuntu-web server)



Guest 1 (Ubuntu) web server로 사용

해당 기능에 관련한 critical 한 log인 access.log와 error.log 파싱

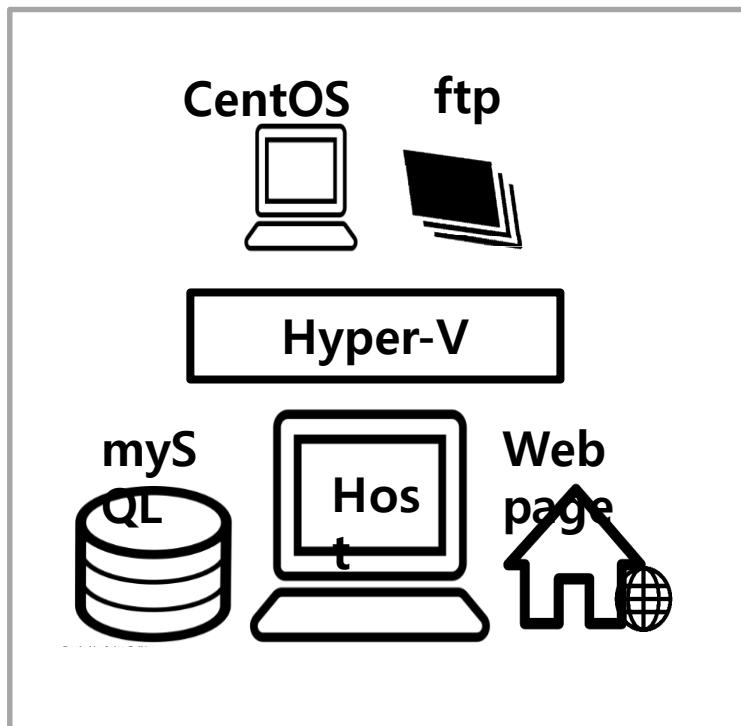
Web Server로 접근하는 Data Packet 파싱

이들을 Host DB에 저장

4

시스템 모듈 상세 설계

Network – VM2(CentOS FTP-Server)



Guest 2 (CentOS) ftp 로 사용

해당 기능에 관련한 critical 한 log인 Vsftpd.log와 syslog를 파싱

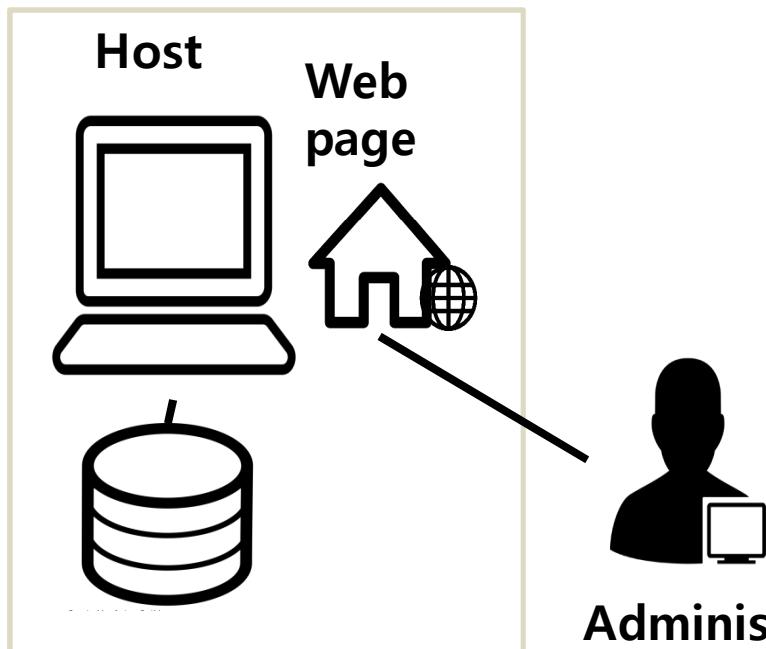
수신되는 Network Data packet 파싱

이를 Host DB에 저장

4

시스템 모듈 상세 설계

Network – Host(Windows server)



Host (windows 8.1 pro) windows server

운용중인 DB와 Web server에 관한 Log 및 SysLog를 파싱

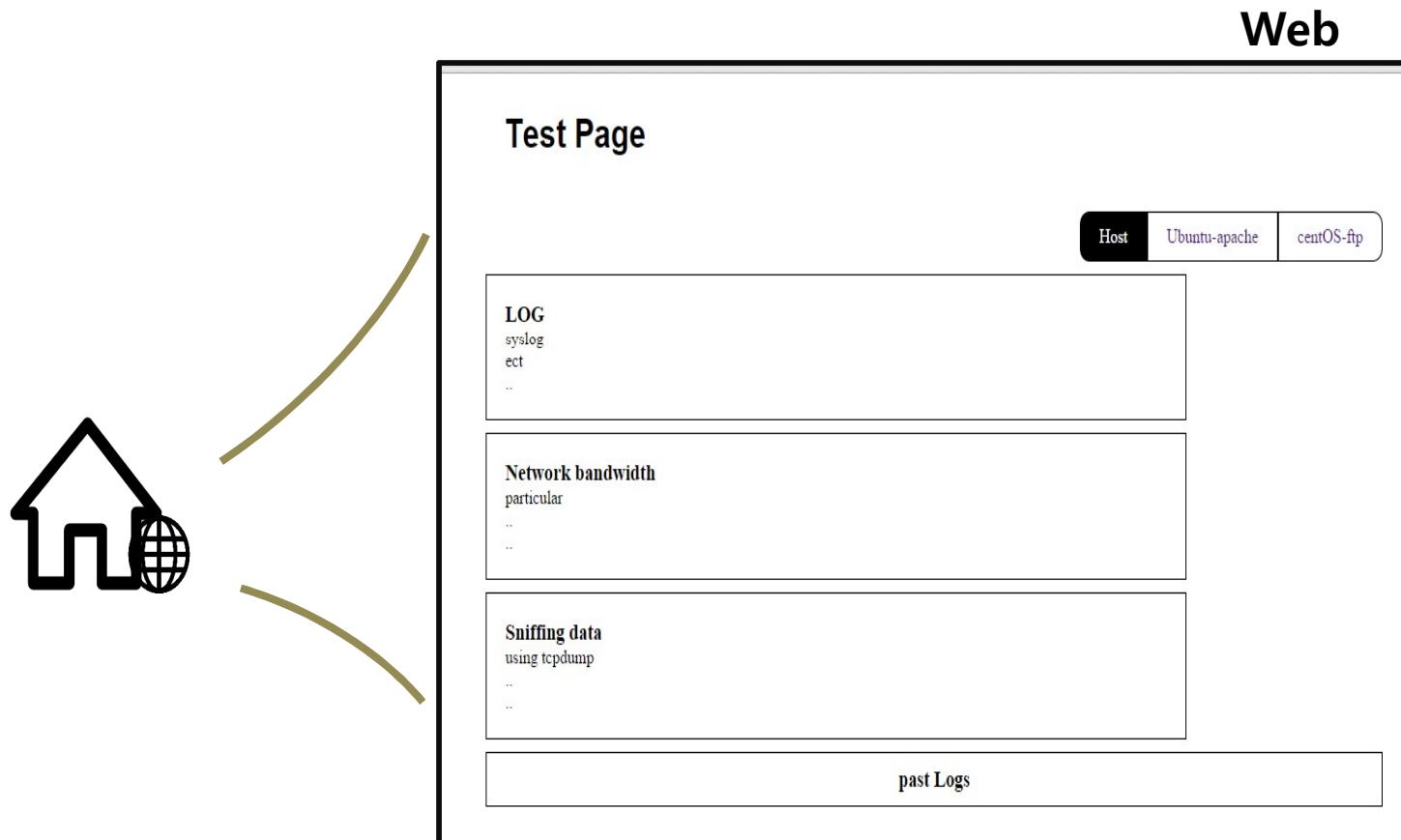
파싱된 데이터를 DB에 저장

Web Page는 향후 Monitoring Page로 사용

4

시스템 모듈 상세 설계

Network – Host(Web View)



5

개발 환경 및 방법

Server Module

Host		Guest1		Guest2	
OS	Windows 8.1 Pro	OS	Ubuntu Linux	OS	CentOS7
HyperViso r	Hyper-V	Server	Apache	Server	FTP
server	apache	Language	PHP, Python	Language	python
DB	Maria DB	NetWork	Bridge Network	NetWork	Bridge Network
Language	PHP, Python	Stroage	host Stroage(Share)	Stroage	host Stroage(Share)
		Design	HTML5	Design	X

6

업무 분담

설계 및 구현에 대한 업무 분담 계획

	박재원	김승연
자료수집	클라우드 시스템 내 주요 기능에 대한 Log 조사	
설계	DB 스키마 설계 시스템 서브 모듈(Host) 상세 설계	Guest 1(Linux) Module 설계 Guest 2(CentOS) Module 설계
구현	DB server 구현 Web, DB 연동, Guest2 ftp구축	Guest1 내의 server 구축
테스트	통합테스트 유지보수	



2. Imagine cup 참여

Imagine cup이 대학생만 참가할 수 있는 대회라는 것을 대회공지를 통해 알게 되었습니다. 4학년인 저는 마지막 기회라 생각하고 저 포함 동아리 4명과 함께 참여하게 되었습니다.

대회 중 제 임무는 아이디어 선정, 프로토 타입 개발이었습니다. 아이디어는 5개 중 가장 팀원의 관심을 끈 물 온도 조절 수도꼭지가 선정되었습니다.

프로토 타입 개발 과정에서 수도꼭지 안에서 물 온도를 조절해야 할 모터가 잦은 고장으로 테스트 하지 못하는 문제점이 발생하였습니다. 저는 모터 지식이 부족하여 모터 고장 원인을 찾는데 상당히 애먹었습니다. 팀원 간 회의에서 모터 고장 원인을 찾는데 도움을 부탁하였고, 팀원들은 흔쾌히 수락하였습니다. 구로, 수원 공구상가를 함께 돌며 모터가 수도꼭지에 맞지 않아 고장이 발생한다는 것을 알 수 있었습니다. 이후 모터를 교체하여 프로젝트를 진행하였습니다.

마감시간에 맞춰 프로토 타입의 시연영상을 제출 할 수 있었습니다. 이번 프로젝트를 통해 문제 해결 방법에 있어 솔직하게 이야기하는 것도 해결책이 된다는 것을 알 수 있었습니다.

온도 조절 수도꼭지

Part - World Citizenship

팀 - 애자일

CONTENTS

- 01 팀 구성원
- 02 프로젝트 개요
- 03 프로젝트 개발 배경
- 04 개발 환경
- 05 프로젝트 설명
- 06 기존 유사품과 차별성
- 07 기대 효과

01 팀 구성원

성 명	소 속	TEL	E-MAIL	비 고
박재원	한국산업기술대	010-9487-2119	pjw9889@naver.com	리 더
김종현	한국산업기술대	010-7399-8002	kimorning77@gmail.com	
이동현	한국산업기술대	010-2239-0401	samuel940401@gmail.com	
최장혁	한국산업기술대	010-2906-4247	gial7586@naver.com	

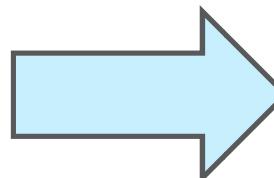
02 프로젝트 개요

문제점

특정 온도의 물을 필요로 할 때

그 온도가 되어지기 까지

물을 낭비하게 된다.



아이디어

수도꼭지 내부에서 기계적으로
온도를 조절하여 물을 흘려버릴
필요없이 해당 온도가 되면
알람이 울려 사용자에게 알려줌

03

프로젝트 개발 배경



- 샤워를 하기 위해 온수가 나올 때까지 물을 흘려 보내는 경우가 대다수
- 이때 온수가 나오는데 걸리는 소요 시간 5~10초

03

프로젝트 개발 배경



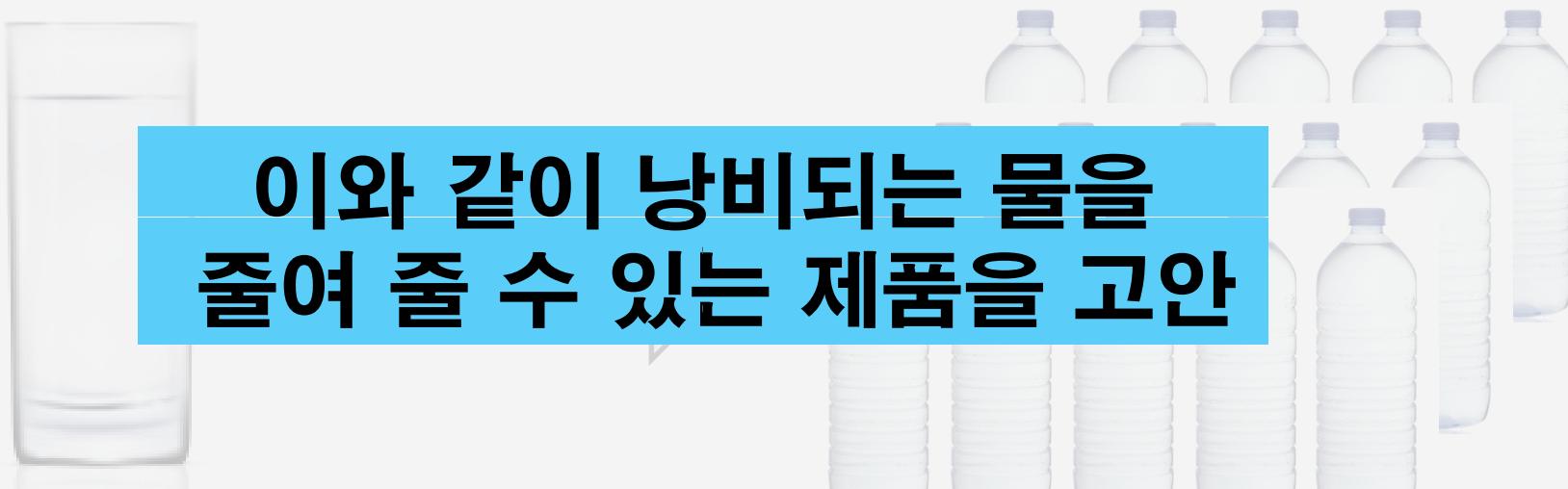
365일 환산



- 1인 1회 물 사용시 적정온도를 맞추기 위해 평균 500 ~ 1000ml를 낭비

- 1년 약 182,500ml 이상 낭비

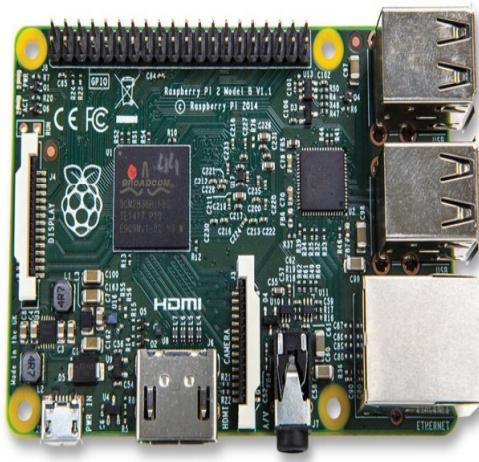
03 프로젝트 개발 배경



- 흘려 버려지는 물의 양
약 500~1000ml
- 1년 약 182,500ml 이상 낭비

04 개발 환경

▶ H/W



Raspberry Pi b+



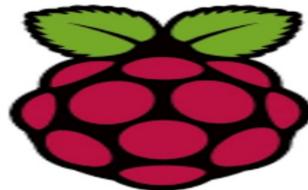
DS18B20



Servo motor SG-90

04 개발 환경

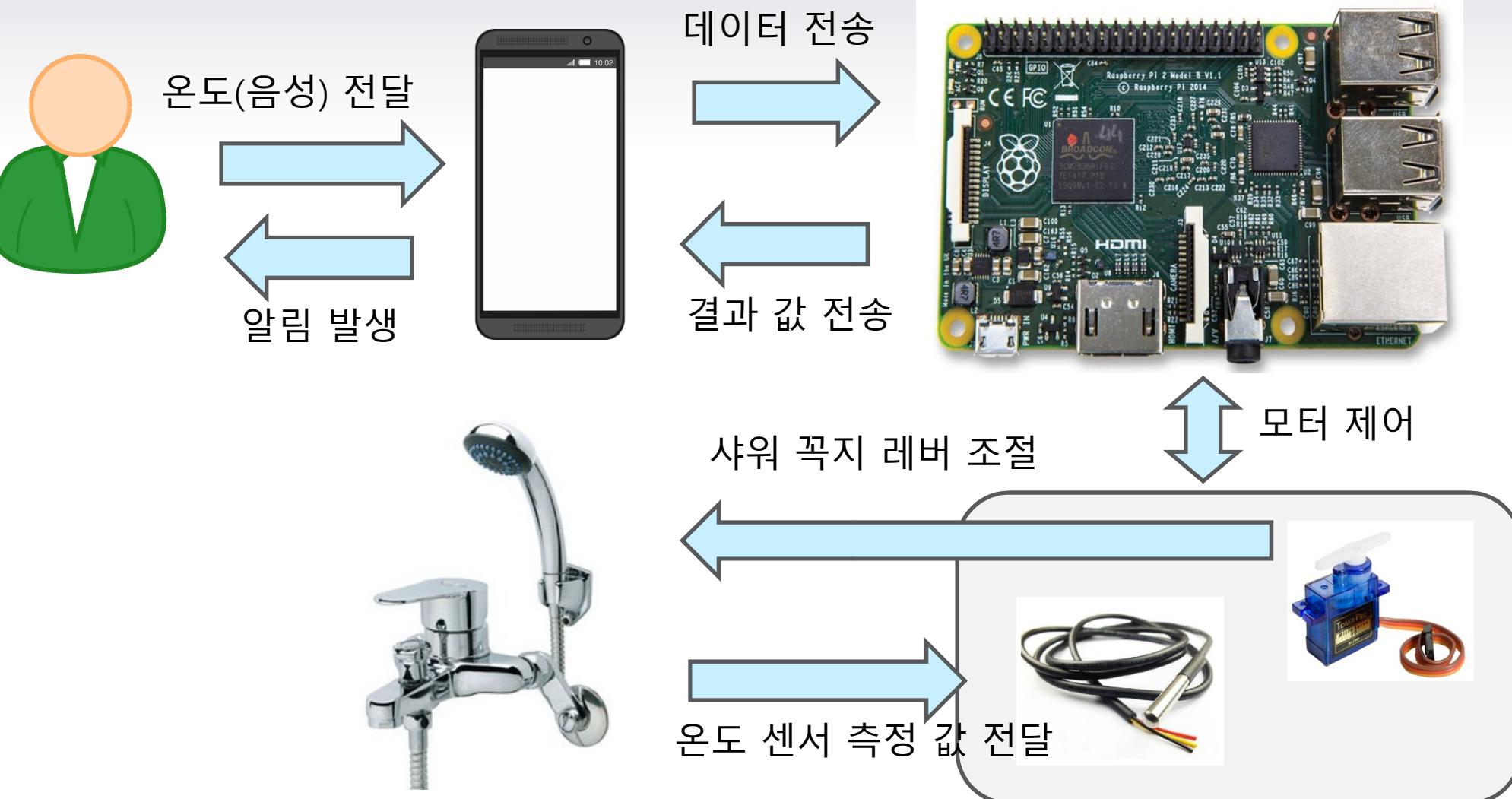
▶ Server

OS	Language
  Rasbian	 python™ Python 2.7.10

▶ Client

OS	Language
 Android 4.0.3	 Java™ Java

05 프로젝트 구조



06 기존 유사품과 차별성



< GROHE사의 자동온도조절 샤워 밸브 >

▶ GROHE사의 자동온도조절 샤워 밸브

- 온도를 조절하기 위해 샤워 밸브까지 접근해야하는 불편함 존재
- 시각 장애인의 접근성 제한

▶ 애자일의 온도조절 수도꼭지

- 스마트폰을 이용한 원격 조정 가능
- 음성 인식이 가능함으로써 시각 장애인의 접근성 향상

기대 효과

- 원하는 온도의 물이 나올 때 까지 물을 흘려버리지 않아도 됨으로써 물 절약 효과
- 시스템 사용 전 미리 스마트폰을 이용하여 원하는 온도를 설정하여 대기 시간을 단축 할 수 있음
- 음성 전달 기능으로 일반인 뿐만 아닌 시각 장애인 등과 같은 특수 군의 사용자도 이용 가능

E N D



Competitor

THIS CERTIFICATE IS PRESENTED TO :

Jae Won Park

Agile

Congratulations and thank you for your hard work dedication as a competitor in Imagine Cup 2016. You are now part of an elite international community of students who have shown remarkable creativity and innovation to push technology forward.

일련번호 제 2016-029호

학습동아리 활동 인증서

이름(학번) : 박재원(2013152048)
학부/학과 : 컴퓨터공학부 컴퓨터공학전공
주 제 : IoT
활동 기간 : 2016년도 1학기

위 학생은 교수학습지원센터에서 주관하는
학습동아리 활동에 성실히 참여하였음을 인증합니다.

2016년 6월 1일

한국산업기술대학교 교수학습지원센터





3. 일본 기업 탐방 활동

현장실습 우수자 선발과정을 통해 일본 규슈 지역에 견학한 경험이 있습니다.

연수 기간 중 5개 기업을 방문하였습니다. 그 중에서 자동차 기업이었던 도요타, 닛산이 가장 기억에 남습니다. 도요타 공장에서는 '장인정신'을 강조하며 공정 과정 중 기술자가 지속적인 검사를 진행하여 품질 개선에 힘쓰는 모습을 보여주었습니다. 닛산의 경우 로봇을 컴퓨터 언어로 구현하여 높은 생산 효율성을 추구하다는 점이 인상 깊었습니다.

저는 두 공장을 방문한 뒤, 미래 자동차뿐 아니라 현재 자동차 생산, 품질 개선 상황에서도 IT 기술이 필요하다는 것을 알게 되었습니다. 이후 언론매체를 통해 자동차에 다양한 전자기기를 탑재하는 것을 보며 IoT와 빅 데이터가 미래 자동차 시장에 중요할 것이라는 생각을 하게 되었습니다.



4. 네트워크 매니지먼트 조교 활동

네트워크 매니지먼트 교과목을 청강하던 중 네트워크 과목에 대한 이해를 높이기 위해 조교를 신청하였습니다. 조교의 주 역할은 수업 실습 장비인 스위치, 라우터 점검과 실습 환경 구축이었습니다.

매주 1시간 씩 실습 전 스위치, 라우터 점검을 하며, 문제가 있는 장비는 A/S 업체와 연락을 하였습니다. 또한 실습에 필요한 라우터와 스위치간 연결 구축, 라우터 간 연결도 수행하였습니다.

조교활동을 통해, 자주 접할 수 없었던 네트워크 장비인 스위치, 라우터를 40대씩 관리하는 특별한 경험을 할 수 있었습니다. 또한 관리, 구축에 필요한 지식들도 습득할 수 있었고, 네트워크 전공 교수님들과 친해질 수 있는 계기가 되었던 것 같습니다.

) 5. 컨퍼런스 및 코딩대회 참여

Openstack 2016 Days 참여(16/02/18)

Blockchain forum 참여(16/03/30)

Nexon Code high 2016 코딩 대회 참여(16/05/28)

2016 Google I/O Extend 참여(16/06/19)

SK 개발자 컨퍼런스 참여(16/06/30)

아시아 태평양 지역 개발자 세미나 참여(16/07/20)

한국정보과학회 학부생 논문 발표회 참여(16/12/23)

클라우드 가상화 환경에서 실시간 모니터링 시스템 구현

김승연 박재원 이보경
한국산업기술대학교 컴퓨터공학과
{seungyeon9312, pjw9889}@kpu.ac.kr

Real-time Monitoring System for Cloud Virtualization Environment

Seung-Yeon Kim Jae-Won Park Bo-Kyung Lee
Dept. of Computer Engineering, Korea Polytechnic University

요약

가상화 환경은 하드웨어를 추가하지 않고도 저장 공간을 보다 효율적으로 사용할 수 있다는 장점을 가지고 있다. 하드웨어를 구입할 비용을 줄일 수 있어 비즈니스 효율성이 획기적으로 좋아진다는 것이다. 따라서 가상화 개념을 클라우드 서비스에 접목하여 한정된 자원을 가지고 보다 나은 품질의 서비스를 제공하려는 시도가 많아지고 있다. 그래서 클라우드 가상화 환경에 대한 관리의 필요성이 높아지고 있는 실정이다. 본 논문에서는 클라우드 가상화 환경의 관리를 위하여 특정 기능을 갖는 클라우드 가상화 환경을 구축하고 실시간으로 데이터를 모니터링하는 시스템을 개발하였다.

1. 서론

최근 몇 년간 IT 서비스 분야에서 클라우드와 가상화 개념은 이슈화되어 자주 등장하고 있다. 클라우드는 다양한 IT 기기에서 접근이 가능하고 포털에서 제공하는 가상 데이터를 사용하기 때문에 사용자는 적은 저장 공간으로 서비스를 이용할 수 있다. 현재 많은 포털 사이트에서 클라우드의 강점을 이용해 저장 공간을 무료로 제공하고 있다. 또 다른 키워드인 가상화는 기존의 물리적인 장치들이 가지는 한계를 보완하는 개념이다. 가상화를 통하여 한정된 자원을 보다 효율적인 자원으로 이용할 수 있도록 한다.[1][2]

또한 클라우드 시장이 점점 확대됨에 따라 서버 내 기존의 저장 공간을 유지하면서 더 좋은 효율을 낼 수 있는 방안을 고민하게 되었다. 이러한 고민을 해결할 수 있는 기술이 가상화 클라우드 개념이다. 그러나 사용자들은 클라우드 가상화 환경에서 어떻게 시스템 자원을 이용하고 있는지 확인하기 원하며 서버 관리자도 역시 클라우드 서버가 어떻게 운영되는지 모니터링하기를 원한다.

본 논문에서는 메인 컴퓨터(Host)에 가상 머신 기술을 적용하여 가상화 환경을 구축하고 이러한 클라우드 가상화 환경에서 실시간으로 데이터를 모니터링하는 시스템을 구현하였다. 본 논문의 구성은 살펴보면 2장에서는 관련기술에 대한 연구, 3장에서는 구현 시스템의 구성과 세부기능, 4장에서는 결과 및 향후 연구과제에 대한 내용을 포함하고 있다.

2. 관련 기술

2.1 클라우드 컴퓨팅

클라우드 컴퓨팅은 각 컴퓨터에서 개별적으로 프로그램을 설치해 데이터를 저장하던 기존 방식에서 벗어나 인터넷 네트워크 상에 모든 컴퓨팅 자원을 저장하여 개

별 컴퓨터에 할당하는 개념을 말한다. 그래서 클라우드 사용자는 필요할 때마다 인터넷을 통해 하드웨어, 소프트웨어 등 IT 자원을 손쉽게 이용할 수 있다.

2.2 가상화 기술과 모니터링 효과 [3]

본 논문에서는 서버 가상화 기술을 적용하여 시스템을 구현하였다. 서버 가상화 기술을 사용함으로써 서버의 한정적인 물리적 자원을 유동적으로 사용할 수 있고, 이는 자원 활용도를 증대 시킬 수 있다. 가상화 환경을 구축하기 위하여 하이퍼바이저라는 중간계층을 두는데 표 1은 다양한 하이퍼바이저에 대한 성능을 비교하고 있다. 본 주제에서는 Windows Hyper-V를 이용하여 가상화 환경을 구축하였다. Hyper-V는 Windows 8.1 Pro 이상의 버전에서는 무료로 사용할 수 있고, GUI를 통해 구성할 수 있는 편리한 인터페이스이므로 누구나 이용할 수 있다는 장점이 있다. 또한 점유하는 리소스 또한 적기 때문에 사용자들로 하여금 소규모 클라우드를 생성하고 관리하는 것을 보다 간편하게 할 수 있도록 한다. 서버 관리 중 서버 상태를 실시간으로 모니터링 하는 것은 예외 상황에 대해 보다 빠르고 유연하게 대처할 수 있다는 효과가 있다.

표 1 상용중인 하이퍼바이저 성능 비교

구분	Openstack	Xen	KVM	Hyper-V
분류	Native	Native	Native	Native
가상화방식	반가상화	전가상화	반가상화	반가상화
가용 OS	Linux	Linux	Linux	Windows
장점	다양한 모듈	점유리소스 많음	커널수정 불필요	점유리소스 적음
단점	복잡성증가	점유리소스 많음	벤더별 구현필요	제약조건이 많음

3. 실시간 모니터링 시스템

3.1 시스템 구성도 및 개발 환경 [4]

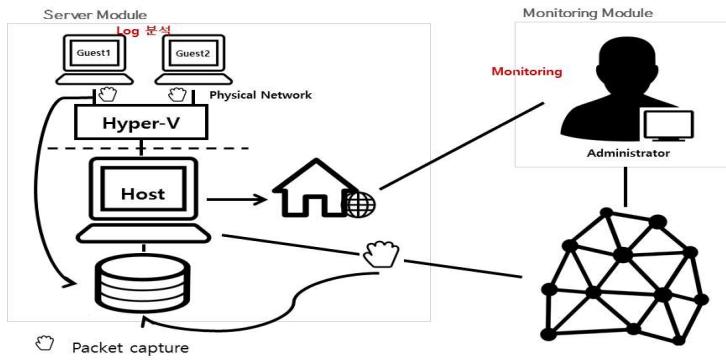


그림 1 시스템 구성도

구현 시스템은 클라우드 가상화 환경을 위한 서버모듈과 데이터를 모니터링하는 모듈로 나된다.

서버모듈은 윈도우즈 기반의 Host와 두 개의 가상머신인 Guest1과 Guest2로 구성된다. Host의 경우에는 데이터베이스를 두어 각 Guest에서 발생하는 기록을 모을 수 있도록 한다. Host에 하이퍼바이저를 사용하여 두 개의 가상화 공간을 생성한다. 본 논문에서는 윈도우즈 환경에서 사용이 용이한 Hyper-V 하이퍼바이저를 이용하여 가상화 공간을 구축하였다.

Guest1과 Guest2는 각기 다른 기능을 갖는 클라우드 서버로의 역할을 수행하도록 설계하였다. 또한 보다 다양한 환경의 가상화를 모니터링하기 위하여 Guest들의 운영체제는 Host와 다르게 리눅스 계열로 설정하였다. 세부적인 개발환경은 표2, 표3, 표4에서 설명하고 있다.

또 다른 모듈은 모니터링 모듈이다. 외부에 존재하는 관리자는 인터넷을 통하여 Host에서 구동되는 웹 서버에 접속하여 Guest들의 현재 상황을 실시간으로 확인할 수 있도록 구현하였다.

표 2 Host 개발 환경

OS	windows 10.1 pro
Hypervisor	Hyper-V
Server	apache 2.4
Database	Maria 10.1.13
Program Language	php, sql
Dashboard Design	HTML, CSS

표 3 Guest1 개발 환경

OS	Ubuntu Linux
Server	Apache
Program Language	Php, Python
Network	Physical Bridge Network
Storage	host storage(shared)
Design	HTML5

표 4 Guest2 개발 환경

OS	CentOSLinux
Server	ftp
Program Language	Php, Python
Network	Physical Bridge Network
Storage	host storage(shared)
Design	X

3.2 세부 기능

가. 서버 모듈 [5][6]

시스템 데이터를 확인하고 Ftp와 Web (Guest1,2)를 지나면서 발생한 데이터의 유무를 확인하고 발생된 데이터는 데이터베이스에 저장한다.

Host의 경우 CPU 사용량, Network Traffic, RAM 정보를 추출하여 데이터베이스에 저장한다.

Guest1은 Ubuntu 운영체제 기반이고 여기에 Apache Web Server를 구축하였다. 해당 웹 서버를 사용하면서 생기는 log들은 Host의 데이터베이스에 계속해서 기록되게 된다. 모든 log를 관리하기 보다는 웹 서버 기능에서 중요하게 다루어지는 error log와 access log 만을 추출하여 기록한다. error log는 웹 서버에 접근하는 과정에서 오류가 생겼을 경우 발생하는 로그이다. 발생 시간과 error 종류, pid, tid, statement, 에러 발생 지점 등을 확인할 수 있다. access log의 경우는 웹 서버에 접근하는 모든 기록이 담겨있다. 마찬가지로 발생 시간, access 종류, pid, tid, statement, 접근 발생 지점에 대한 확인이 가능하다. 또한 가상 공간을 생성하면서 사용자 임의대로 부여한 메모리가 어떻게 사용되고 있는지, 가상화한 Guest의 CPU와 RAM 사용량과 네트워크 대역폭 등에 대한 정보도 데이터베이스에 기록한다.

Guest2는 CentOS 운영체제 기반이고 여기에 Ftp server를 구축하였다. 마찬가지로 파일을 업로드하고 다운로드하는 동안 생기는 기록은 Host의 데이터베이스에 기록된다. Ftp 서버 기능에서 파일 전송에 관련하여 중요하게 다루어지는 log인 error와 warning log를 추출하여 기록한다. error log에는 에러 발생 시간, 발생한 에러 종류 등의 정보를 추출하며 warning log의 경우도 크게 다르지 않다. Guest2 로그의 경우에도 메모리 사용량, CPU 사용량, 네트워크 대역폭 등에 대한 정보 또한 데이터베이스에 기록한다.

```
VJ:1---[29/Apr/2016:16:33:40 +0900] "GET / HTTP/1.1" 200 3594 "-" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:4.0.1) [29/Apr/2016:16:33:40 +0900] "GET /icons/ubuntu-logo.png HTTP/1.1" 200 3688 "http://127.0.0.1/" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:4.0.1) [28/May/2016:17:19:47 +0900] "GET /HTTP/1.1" 200 3594 "-" Mozilla/5.0 (Windows NT 10.0; WOW64; ;NCLIENT50_AAP1A981889C9D) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.109 Safari/537.36 [28/May/2016:17:21:21 +0900] "GET /HTTP/1.1" 200 3594 "-" Mozilla/5.0 (Windows NT 10.0; WOW64; ;NCLIENT50_AAP1A981889C9D) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.109 Safari/537.36 [28/May/2016:17:21:21 +0900] "GET /icons/ubuntu-logo.png HTTP/1.1" 304 179 "http://192.168.0.9/" Mozilla/5.0 (Windows NT 10.0; WOW64; ;NCLIENT50_AAP1A981889C9D) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.109 Safari/537.36 [28/May/2016:17:21:22 +0900] "GET /HTTP/1.1" 200 3593 "-" Mozilla/5.0 (Windows NT 10.0; WOW64; ;NCLIENT50_AAP1A981889C9D) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.109 Safari/537.36 [28/May/2016:17:21:22 +0900] "GET /icons/ubuntu-logo.png HTTP/1.1" 304 179 "http://192.168.0.9/" Mozilla/5.0 (Windows NT 10.0; WOW64; ;NCLIENT50_AAP1A981889C9D) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.109 Safari/537.36 [28/May/2016:17:21:23 +0900] "GET /HTTP/1.1" 200 3593 "-" Mozilla/5.0 (Windows NT 10.0; WOW64; ;NCLIENT50_AAP1A981889C9D) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.109 Safari/537.36 [28/May/2016:17:21:23 +0900] "GET /icons/ubuntu-logo.png HTTP/1.1" 304 179 "http://192.168.0.9/" Mozilla/5.0 (Windows NT 10.0; WOW64; ;NCLIENT50_AAP1A981889C9D) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.109 Safari/537.36
```

그림 2 apache access log 캡쳐링

```
[Thu May 26 14:18:25.987547 2016] [core:notice] [pid 1005:tid 140068685129600] AH00094: Command line: '/usr/sbin/apache2'
[Thu May 26 15:06:27.491533 2016] [mpm_eventnotice] [pid 1005:tid 140068685129600] AH00491: caught SIGTERM, shutting down
[Thu May 26 17:26:15.682994 2016] [mpm_eventnotice] [pid 1157:tid 139834218682240] AH00489: Apache/2.4.7 (Ubuntu) configured -- 
[Thu May 26 17:26:15.699094 2016] [core:notice] [pid 1157:tid 139834218682240] AH00094: Command line: '/usr/sbin/apache2'
[Sat May 28 17:07:34.598590 2016] [mpm_eventnotice] [pid 1041:tid 140370917660544] AH00489: Apache/2.4.7 (Ubuntu) configured -- 
[Sat May 28 17:07:34.637942 2016] [core:notice] [pid 1041:tid 140370917660544] AH00094: Command line: '/usr/sbin/apache2'
[Sat May 28 17:16:26.788465 2016] [mpm_eventnotice] [pid 1041:tid 140370917660544] AH00493: SIGUSR1 received. Doing graceful restart
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive in 
[Sat May 28 17:16:26.789050 2016] [mpm_eventnotice] [pid 1041:tid 140370917660544] AH00489: Apache/2.4.7 (Ubuntu) configured --
```

그림 3 apache error log 캡쳐링

time	use_cpu	nice_cpu	system_cpu	iowait_cpu	steal_cpu	idle_cpu
2016-10-05 13:32:40	17.62	1.57	3.62	6.2	0	70.99
2016-10-05 13:32:41	17.6	1.57	3.63	6.19	0	71.01
2016-10-05 13:32:42	17.59	1.57	3.63	6.18	0	71.04
2016-10-05 13:32:44	17.57	1.57	3.63	6.17	0	71.06
2016-10-05 13:32:45	17.55	1.57	3.63	6.16	0	71.09
2016-10-05 13:32:46	17.54	1.56	3.63	6.16	0	71.11
2016-10-05 13:32:47	17.53	1.56	3.63	6.15	0	71.13
2016-10-05 13:32:48	17.51	1.56	3.63	6.14	0	71.15
2016-10-05 13:32:50	17.5	1.56	3.64	6.13	0	71.17
...

그림 4 cpu 사용 현황 데이터베이스 캡쳐링

나. 모니터링 모듈

가장 먼저 관리자에게 부여된 아이디와 비밀번호로 로그인을 한다. 올바른 아이디와 그에 부합하는 비밀번호 여부를 확인한 후 각 모니터링 요소를 웹 상에 출력하도록 한다. Host, Guest1, Guest2에 대하여 그림 5와 같은 템이 있어 각각의 OS에 대하여 network traffic과 cpu usage, ram usage, log 등을 확인할 수 있다. 클릭하기 전에는 수치화된 형태로 표현되지만 원하는 템을 클릭하면 보다 상세한 정보를 확인할 수 있다. 네트워크의 경우 사용량의 절대치를 보여주기보다는 상대적인 사용량을 확인하기 용이하도록 꺾은선 그래프로 나타내었다. CPU사용량과 RAM사용량의 경우 할당된 최대의 양과 현재 사용량을 원형 그래프에 함께 표현하여 현재 얼마나 사용하고 있는지 한눈에 확인할 수 있도록 하였다. 모든 정보는 5초마다 refresh되며 가시화된 그래프로서 트래픽의 변화를 확인할 수 있다.



그림 5 Dash Board(Network, CPU, RAM, Log)



그림 7 RAM 사용량 실시간 모니터링

4. 결론 및 향후 연구과제

본 논문에서 개발된 시스템은 메인 컴퓨터에 가상 머신을 사용하여 두 개의 가상화 공간을 만드는 것에서 시작된다. 두 가상화 환경은 각기 다른 용도인 Ftp와 Web Server로 사용되며 사용하는 과정에서 생긴 log데이터와 가상 머신의 기본적인 CPU 현황 등을 외부에 둔 웹 서버를 통하여 확인할 수 있도록 하였다.

클라우드 환경을 모니터링하는 기준의 연구도 물론 존재한다. 하지만 대형 플랫폼에 대한 국한적인 모니터링만을 지원하며 개인적이고 규모가 작은 가상화 서버에 대하여는 서비스가 지원되지 않는다. 본 연구가 소규모 가상화 환경을 대상으로 한 모니터링이라는 점에서 그 의의가 있다고 할 수 있다. 사용자는 이 시스템을 통해 자신의 PC를 용도에 맞게 리소스를 분배하고 좀 더 효율적인 가상화 환경으로 작업할 수 있다는 긍정적인 효과를 기대할 수 있다. 또한 본 논문에서 개발한 기능을 바탕으로 데이터 마이닝, AI 등 신 기술을 접목한다면 자동화까지 가능한 모니터링 시스템 구현이 가능하다. 또한 보안에 대한 이슈가 끊임없이 제기되고 있는 클라우드 환경에 대하여 향후 침입탐지 시스템(IDS)을 포함하는 보안 이슈를 해결하기 위한 방안을 고려하여야 한다.

5. 참고문헌

- [1] Huaibin Wang, Haiyun Zhou and Chundong Wang, 「virtual machine based Intrusion Detection System Framework in Cloud Computin Environmnet」 (JOURNAL OF COMPUTERS, VOL.7, NO.10, OCTOBER 2012, 2397)
- [2] S C Rachana, Dr. H S Guruprasad, 「Virtual Machine Introspection」, (COMPUSOFT, Aninternational journal of advanced computer technology, 3(4), April–2014, (Volume–III, Issue–VI))
- [3] 백승주, 『Hyper-V를 다루는 기술』, 길벗, 2014
- [4] 최동욱 외 4명, 『실전 클라우드 가상화 구축기술』 한빛미디어 2012
- [5] 양진혁 외 6명, 「분산 데이터 통합 모니터링 시스템의 설계와 구현 : 국가 과학 기술지식 정보서비스 (NTIS) 사례중심」, 한국컴퓨터종합학술대회 논문집 Vol.37, 2010
- [6] 김환태, 김황남, 「가상화 클라우드 데이터센터에서 가상머신간의 균등한 성능 보장을 위한 제어 알고리즘」, 한국 통신학회 논문지' 13-06 Vol.38C No.06, 2013



그림 6 Network 실시간 모니터링



THANK YOU