

# 보안을 위한 홈 네트워크 원격 모니터링 시스템

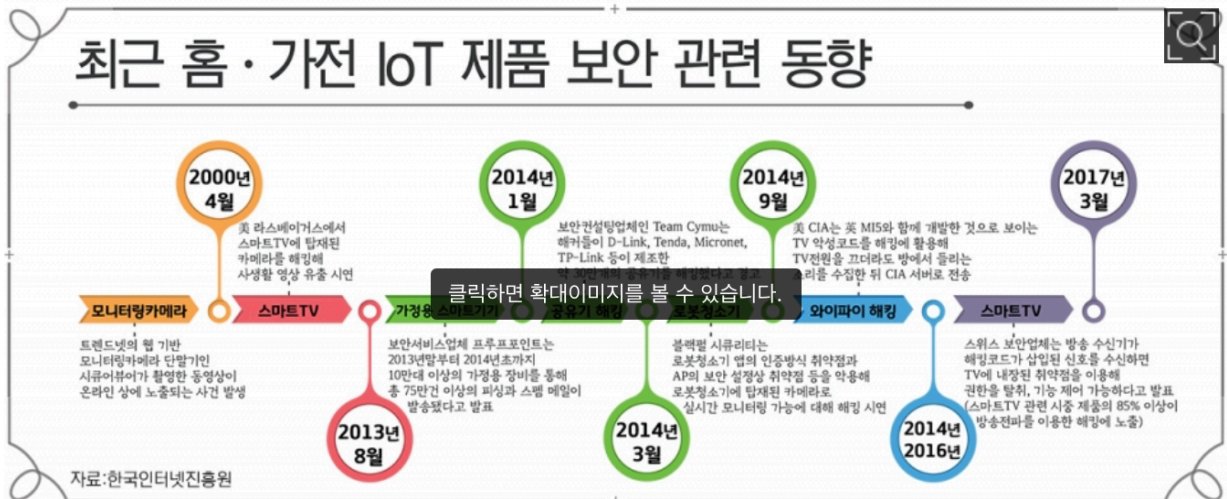
## Contents

1. 문제 정의	
1.1 문제가 발생한 이유	.....
1.2 설계 목표	.....
1.3 제약조건 분석	.....
1.4 문제에 대한 기본 접근법	.....
1.5 Revised Problem Statement	.....
2. 배경 이론 및 기존 방법	
2.1 패킷 스니핑	.....
2.2 패킷 스니퍼	.....
3. 문제 접근 방법	
3.1 하드웨어 설계	.....
3.2 소프트웨어 설계	.....
3.2.1 Raspberry Pi OS 설치	.....
3.2.2 Raspberry Pi Server 설치	.....
3.2.3 Raspberry Pi Cam 설정 및 사용	.....
3.2.4 Raspberry Pi 에서의 패킷 스니핑	.....
3.3 설계 대안 선정	.....
3.4 설계 대안 분석	.....
4. 실험 결과	
4.1 유효성과 효율성 검증	.....
4.2 기존 방법과 비교 분석	.....
5. 결론	.....
5.1 요약 및 향후 과제	.....
6. 참조	.....

## 1. 문제 정의

### 1.1 문제가 발생한 이유

IoT, 즉 Internet of Things 의 시대가 오면서, 온갖 사물이 인터넷에 연결되어 외부의 공격으로부터 취약해지고, 해커가 공격할 수 있는 경로도 더욱 다양해졌다. 하지만, 그에 대한 대비는 미약하게 이루어지고 있어, 실제 IOT 장비 해킹으로 인한 많은 피해 사례가 발생하고 있다.



자료:한국인터넷진흥원

<[그림 1-1] IOT 보안 관련 동향>

가장 피해가 많이 발생하는 곳은 가정집이다. 회사 또는 공공기관에서 사용되는 기기들은 철저하게 검증되고 엄격한 보안 솔루션이 적용되어 공격에 대한 방어가 이루어지고 있지만, 일반 가정집에서 사용되는 스마트 기기들은 보안에 큰 노력을 들이지 않을 뿐만 아니라, 사용자가 공격에 대한 인지를 하지 못하여 피해가 점점 커지고 있다.

가장 피해가 크게 발생할 수 있는 기기는 IP 카메라이다. IP 카메라가 공격 받으면서 어떤 공간보다 사생활이 보호되어야 할 집안 생활이 여과없이 노출되어 진다. 실제로, 해커가 Brute Force 기법으로 IP 카메라를 해킹하여 이용자의 허가 없이 이용되어 지는데, 초기 비밀번호를 사용하는 경우가 많아 쉽게 접근이 가능하며, 비밀번호를 바꾼다 하더라도 많은 시도를 통해 아이디와 비밀번호를 획득하여 해킹을 한다.

위와 같은 사례에서 볼 때, 무분별한 불특정 공격은 유저가 보안 위협에 대하여 인지를 하고 있다면 충분히 방지 또는 방어를 할 수 있는 방법이 있을 것이라고 결론을 얻을 수 있었다.

### 1.2 설계 목표

보안을 위한 홈 네트워크 원격 모니터링 시스템을 설계를 위해 다음의 4 가지 목표를 추구한다.

**시장성:** 보안 시스템을 위한 Hardware 와 Software 설계 비용, 공급자와 수요자의 서비스 유지 비용을 최소한으로 하여 시장 가격을 낮추기 위한 목표이다.

**휴대성:** 홈 네트워크 시스템이라 휴대성을 크게 고려하지 않아도 된다.

**유용성:** 개인의 사생활이 보장되어야 함으로, 많은 위협으로부터 노출되는 IOT 장비 특성 상 개인 정보 보호에 각별한 주의가 필요하다.

**내구성:** 책상 또는 높은 곳에 놓고 사용 할 것이기 때문에, 물리적 내충성이 충분히 보장되어야 한다.

### 1.3 제약조건 분석

모든 형태의 IOT 장비에 대해 처리할 수 있어야 한다.

: 제조사, 네트워크 프로토콜에 관련없이 모든 종류에 장비에 대해 공격을 감지하고, 이용자에게 제공할 수 있어야 한다.

추가 서비스 비용이 발생하여야 하지 않아야 한다.

: 기기 구입, 시스템 업그레이드 이외에 데이터 분석과 같은 서비스에 필요한 서버를 운용하지 않겠다는 의미이다.

### 1.4 문제에 대한 기본 접근법

근본적인 이유를 해결하지 못하고, 임시적인 방편에 의존한다. 유저가 경각심을 가지고 해당 IOT 기기의 초기 비밀번호를 변경하고 주기적으로 비밀번호를 복잡한 형태를 가지고 있는 것으로 바꾸는 것을 기본 접근법으로 하고 있다.

### 1.5 Revised Problem Statement

안전한 홈 네트워킹 환경을 위하여, Raspberry Pi 를 활용한 원격 감시 타워를 설계하라

## 2. 배경이론 및 기존 방법

### 2.1 패킷 스니핑

패킷 스니핑이란 네트워크 상에서 자신이 아닌 다른 디바이스 간 주고 받는 통신을 엿들음으로 데이터를 전송하고 분석하는 행위를 말한다. 패킷 스니핑을 하여, 어떠한 사용자가 네트워크를 사용 중이며, 프로그램 마다 사용하는 네트워크 사용량을 감지, 악의적인 목적으로 사용하는 사용자 감지, 비효율적 사용자 분석 후 최적화 방안 모색 등 다양한 방법으로 사용 될 수 있다. 주로 관리자가 의미 있는 데이터를 추출 후, 원하는 목적에 맞게 사용하려고 패킷 스니핑을 한다.

### 2.2 패킷 스니퍼 (Wireshark)

패킷 스니퍼는 스니핑을 실질적으로 수행하는 소프트웨어이다. 패킷 스니퍼를 사용할 때, 여러 조건들을 고려하여야 한다. 운영체제, 지원하는 환경과 프로토콜, 사용의 편리성, 사용 비용을 고려하여 선정하여야 한다.

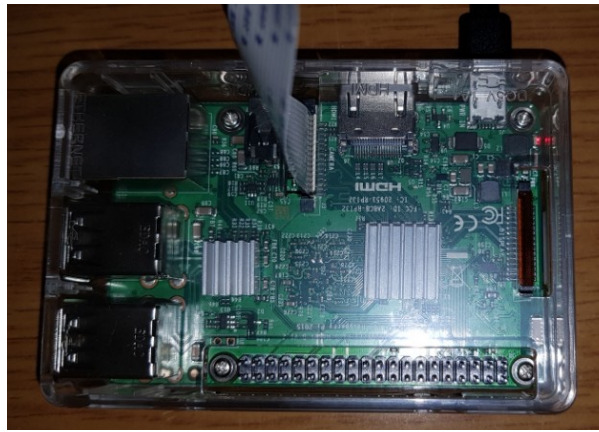
이번 설계에서는 오픈소스 패킷 스니퍼인 와이어샤크를 사용하려고 한다. 윈도우, 맥, 리눅스의 운영체제에서 돌아가며, 850 여 개 이상의 프로토콜을 지원한다. 또한, 오픈소스이기 때문에, 새로운 프로토콜에 대해서 추가적인 업데이트 또한 빠르다. GUI 를 지원하여 사용하기 쉽고, 오픈소스이기 때문에, 사용 비용 또한 없다.

패킷 스니퍼는 시스템의 네트워크 카드를 조작하여 무차별적 모드로 전환하여, 모든 패킷에 대하여 접근 권한을 가지고, 모든 정보를 수집할 수 있도록 해준다.

## 3. 문제 접근 방법

### 3.1 하드웨어 설계

아래 그림과 같이 Raspberry Pi 3 Model B 를 조립하여서, 전원케이블에 연결하면 사용할 수 있도록 조립한다. I/O device (모니터, 키보드, 마우스)를 연결하여 개별적 인터페이스에서 사용가능하고, SSH 를 통한 원격으로 접속하여 Raspberry Pi 를 컨트롤 할 수 있다.

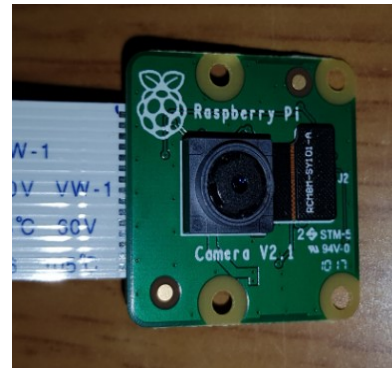


<[그림 3-1] 라즈베리파이 3 모듈>

또한, Pi Cam 을 모듈에 추가적으로 장착하여서, IP Camera 를 보유하고 있는 IOT 장치를 설계하여, 해당 기기에 대하여 공격을 감지할 수 있도록 한다. 위의 절차를 마지막으로, 하드웨어적 설계를 마무리 한다. 아래의 그림은 Pi Cam 을 장착한 Raspberry Pi 의 사진이다.



<[그림 3-2] Raspberry Pi 3 with Pi Cam>



<[그림 3-3] Raspberry Pi Cam>

### 3.2 소프트웨어 설계

먼저, 개발환경으로 맥북을 사용하여, 모든 소프트웨어 설계를 Linux 환경에서 진행하였음을 밝힌다.

#### 3.2.1 Raspberry Pi OS 설치

Raspbian Stretch 의 img 파일을 다운 받아, 메모리칩을 컴퓨터에 연결하여 칩에 OS 를 설치한다. USB 는 fat-32 형태로 포맷이 되어야 하며, unamout 된 상태에서 칩에 이미지 파일을 구워야 한다.GUI 를 지원하는 Raspbian Stretch OS 를 라즈베리파이에 설치하였다.

#### 3.2.2 Raspberry Pi Server 설치

설치를 한 후, ssh 를 통하여 다른 컴퓨터에서도 Raspberry Pi 에 직접적인 I/O device 를 연결하지 않고, 원격으로 접속하여 사용할 수 있다. Raspberry Pi 의 IP 주소를 알고 있다면, "ssh pi@IP\_ADDRESS" 를 사용하여서 접속할 수 있다.

```

baeggeonhoui-MacBook-Pro:~ a21400357$ ssh pi@192.168.0.114
pi@192.168.0.114's password:
Linux raspberrypi 4.14.39-v7+ #1112 SMP Sat May 5 12:01:33 BST 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon May 7 15:22:49 2018
pi@raspberrypi:~ $

```

<[그림 3-4] SSH 를 활용한 원격 접속>

서버로는 AMP (Apache, MySQL, PHP)를 설치하였다. 추가적으로 GUI 환경에서 데이터베이스를 관리하기 위해서 phpMyAdmin 를 설치하였다.

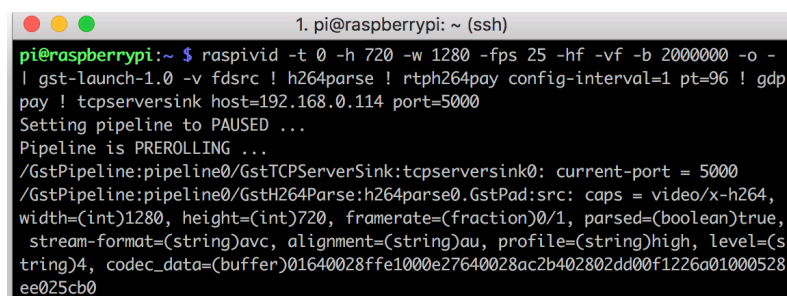
### 3.2.3 Raspberry Pi Cam 설정 및 사용

Pi Cam 을 하드웨어적으로 Raspberry Pi 에 먼저 연결한다. Raspberry Pi3 모델 B 의 Stretch OS 에서는, 인터페이스 옵션 항목에서 파이 카메라 활용을 설정해줄 수 있다.

하드웨어와 소프트웨어적으로 구성을 완료하고, 실제 카메라가 작동하는 지 테스트를 할 수 있다. "Rastill" 명령어를 활용해 카메라, "Raspivid" 명령어를 이용해 비디오 촬영을 할 수 있고, 원하는 경로에 저장을 할 수 있다.

현재, 카메라의 사용목적은 CCTV 의 기능을 하도록 만드는 것이기 때문에, 실시간 스트리밍을 가능하도록 설계하려고 한다. 이를 하기 위하여, gstreamer 라는 스트리밍 툴을 Raspberry Pi 서버에 설치하였다. 다른 스트리밍 도구와 비교하였을 때, 응답속도가 빠르고, Linux 환경에서 활용하기 좋아서, gstreamer 를 선택하였다. "Raspivid" 명령어를 활용해서, 스트리밍하는 IP 주소를 할당해준다면, 해당 IP 에서 현재 Pi Cam 에서 촬영되어지고 있는 상황을 스트리밍 할 수 있다.

실제로 사용되는 CCTV 에서는 해당 IP 주소에 접근하기 위하여, 아이디와 비밀번호가 필요하겠지만, 이번 설계에서 아이디와 비밀번호는 알고 있지 않아도, IP 주소만 알아도 해당 Pi Cam 에 접속하여, 영상을 볼 수 있도록 설계한다.



```

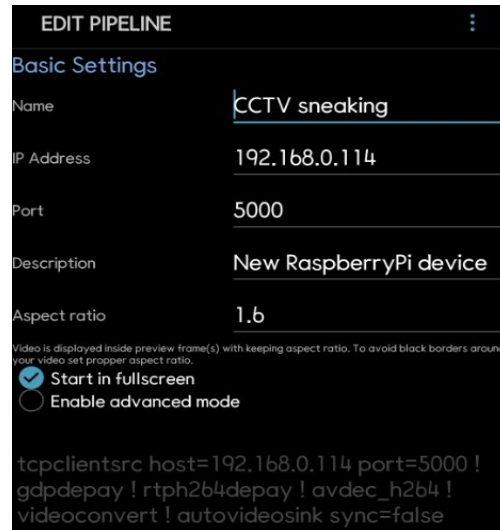
1. pi@raspberrypi: ~ (ssh)
pi@raspberrypi:~ $ raspivid -t 0 -h 720 -w 1280 -fps 25 -hf -vf -b 2000000 -o -
| gst-launch-1.0 -v fdsrc ! h264parse ! rtph264pay config-interval=1 pt=96 ! gdp
pay ! tcpserver sink host=192.168.0.114 port=5000
Setting pipeline to PAUSED ...
Pipeline is PREROLLING ...
/GstPipeline:pipeline0/GstTCPServerSink:tcpserver sink0: current-port = 5000
/GstPipeline:pipeline0/GstH264Parse:h264parse0.GstPad:src: caps = video/x-h264,
width=(int)1280, height=(int)720, framerate=(fraction)0/1, parsed=(boolean)true,
stream-format=(string)avc, alignment=(string)au, profile=(string)high, level=(s
tring)4, codec_data=(buffer)01640028ffe1000e27640028ac2b402802dd00f1226a01000528
ee025cb0

```

<[그림 3-5] Pi Cam 을 이용한 스트리밍>

이번 설계 모델에서, 다른 기기에서 스트리밍 서버에 접속하기 위해서 IP 주소와 해당 포트 번호만 알면 된다. 비단 PC 뿐만 아니라, 스마트폰과 같은 네트워크 스트리밍을 지원하는 프로그램을 설치 할 수 있는 운영체제에서는 모두 다 접속 가능함을 알 수 있다. 예를 들어, 모바일 기기에서 접속하기 위해서, 네트워크 스트리밍을 지원하는 동영상 프로그램으로 쉽게 해당 Pi Cam 에

접속할 수 있다. 다른 기기에서 몰래 영상을 볼 수 있다는 것을 확인하기위해서, ANDROID 기기의 RaspberryPi Camera Viewer (embedding gstreamer)를 다운 받아서 사용했다. 보안을 위한 장벽이 아무것도 없었기 때문에, 단순히 IP 주소를 입력하는 것만으로도, 약 1 초 정도의 딜레이가 생기긴 하였지만, 별도의 제제 없이 카메라에서 촬영되어지고 있는 영상을 수신할 수 있다. 수신을 하여, 그 영상을 별도로 저장 가능하며, 접속자의 입장에서 원하는 대로 사용이 가능하기 때문에, 보안에 각별히 신경을 써야 해야 함을 알 수 있다.



<[그림 3-6] 다른 기기에서 IP 주소를 활용하여 원격 접속 방법>

하지만 이 외의 방법으로, 카메라가 설치 되어 있는 라즈베리파이에서는 스트리밍 서버를 열고, 그 이외의 패킷 스니핑을 진행하는 기기에서 gstreamer 를 설치하여 열려있는 스트리밍 서버에 접속하여 실제 스트리밍 되고 있는 영상을, 가로채서 볼 수 있었다.

### 3.2.4 Raspberry Pi 패킷 스니핑

Raspberry Pi 에 스니퍼를 분석한 결과를 토대로 와이어샤크를 설치하였다 wireshark 는 ssh 를 통한 원격 접속해서 수행을 허용하지 않아, Raspberry Pi 기기에서 직접 실행을 해주어야 했다. 패킷 스니핑을 수행 하였을 때, 현 기기에서 WLAN 과 bluetooth 로만 외부와 통신을 하고 있음으로, 두 항목에 대해서만 신호가 잡혔다. Pi Cam 의 보안성을 높이기 위해, bluetooth 로 정보를 수신하지 않기로 판단, 오직 WLAN 혹은 Ethernet 으로만 외부로부터 정보를 수신한다. 카메라를 사용하지 않을 때, 패킷 스니핑을 한 데이터와 스트리밍을 할 때 패킷 스니핑을 한 결과를 비교하여서, 공격을 감지 할 수 있는 유의미한 방법을 찾는다.

## 4. 실험 결과

### 4.1 유효성과 효율성 검증

먼저, IP Camera 가 작동하고 있지않은 상황을 가정하여, Pi Cam 을 실행시키지 않은 상태에서 네트워크의 상태를 알아보기 위해서 패킷 스니핑을 1~2 분 단위로 다섯 번에 걸쳐 결과를 측정하였다. 그리고 그 반대의 경우를 가정하여, 실제 홈 네트워크 환경과 유사하게 Pi Cam 을 실행시켜, 영상 스트리밍을 하고 있는 환경을 세팅하고, 마찬가지로 패킷 스니핑을 1~2 분 단위로 다섯 번에 걸쳐 측정했다. 패킷이 가지고 있는 정보를 나열해보면 Source, Destination, Protocol, PacketLength,



Information 이 있다. 스니핑을 한 패킷을 CSV 파일의 형태로 추출하여서, PacketLength 에 대한 데이터를 가공하였다. 먼저, 해당 네트워크에서 IP 카메라와 관련 없는 노이즈가 발생하여, 영상을 스트리밍하는 라즈베리파이 기기의 주소를 네트워크 소스로 가지는 패킷에 대하여만 1 차적으로 필터링하여 라즈베리파이에서 송신만 하는 패킷에 대한 데이터를 얻었다.

167	7.31733982	192.168.0.115	192.168.0.104	IPA	80	unknown 0x96
168	7.31745743	192.168.0.115	192.168.0.104	RSL	1514	unknown 0 unknc
169	7.31793149	192.168.0.115	192.168.0.108	IPA	1514	unknown 0xd3
170	7.31805513	192.168.0.115	192.168.0.104	IPA	1514	unknown 0x4f
171	7.31844279	192.168.0.115	192.168.0.108	IPA	1514	unknown 0x70
172	7.31856425	192.168.0.115	192.168.0.104	IPA	1514	unknown 0xa6
173	7.31894904	192.168.0.115	192.168.0.108	IPA	1514	unknown 0x60
174	7.31906596	192.168.0.115	192.168.0.104	IPA	1514	unknown 0xce
175	7.31944825	192.168.0.115	192.168.0.108	IPA	1514	unknown 0xb2

<그림 4-1> 스트리밍 시 카메라에 대한 패킷 패턴>

No.	Time	Source	Destination	Protocol	Length	Info
138	105.514705	192.168.0.11	224.0.0.251	IGMPv2	46	Membership Report group 224.0.0.251

<그림 4-2> 스트리밍 안할 시 카메라에 대한 패킷 패턴>

스트리밍을 하는 상황과 하지 않는 상황에서, 패킷을 살펴보면 각 PacketLength 가 1514 인, 프로토콜을 RSL, IPA 로 가지며, 영상을 수신하는 기기의 아이피 주소가 목적지로 나타났다. 그와 반대 방향의 통신으로, 영상을 수신하는 기기의 아이피 주소를 source 로 하고 IP Camera 를 목적지로 가지는 패킷은 TCP 프로토콜을 사용하며 상호 간의 통신을 하였다.

하나의 기기 뿐만 아니라, 여러 대의 기기에 대해서 영상을 스트리밍을 할 때, 해당하는 기기의 IP 주소가 SOURCE -> IP\_ADDR\_OF\_RECEIVING\_VIDEO\_DEVICE 의 형태로 나타나기 때문에, 패킷이 갈 수 있는 목적지를 사용자가 원하는 device 에 한해서 제약하고, 만약 새로운 IP 주소를 가진 이용자가 나타난다면 사용자에게 공격을 감지하는 알람을 보내주는 전략을 사용할 수 있다.

## 4.2 기존 방법과 비교 분석

지금까지 기술력과 정보에 대한 보안을 중요하게 생각하는 큰 기업이 아닌 이상, 일반적인 가정에서는 IOT 장비에 대한 보안을 전혀 심각하게 고려하지 않았으며, 심지어 초기 비밀번호조차 변경하지 않고 사용되어지는 경우가 많았다. 이와 같은 환경에서, 대중매체가 피해 예방을 하기 위해서 제안한 방법은 사용자가 추가적으로 시간과 노력을 들여야 하는 초기 비밀번호 변경, 특정한 주기마다 비밀번호 변경이다. 또한, 제안된 방법을 사용한다고 하더라도, 홈 네트워크의 IOT 장비가 공격 받고 있는 지 여부조차 알지 못해서, 개인 프라이버시에 매우 큰 위협이 가해지고 있다.

## 5. 결론

### 5.1 요약 및 향후 과제

네트워크 트래픽의 관찰 결과를 사용자가 이해하기 쉽게 제공하고, 홈 네트워크 시스템에 공격을 당했을 때와 유사한 환경을 분석하여, 그 상황이 재발한다면 사용자에게 경고를 해주는 시스템을 설계하였다. Raspberry Pi 를 사용하여 별도로 큰 비용을 들이지 않고, 시스템을 구현하였다. 그렇다면, 사용자는 개인의 프라이버시가 침해되는 지 여부를 실시간으로 확인하여, IOT 장비를 사용함으로써



얻을 수 있는 편리함을 보안의 불안함에 희생하지 않아도 된다.

향후 과제로는, 홈 네트워크에 대한 공격 감지로 끝나는 것이 아니라 실제 공격을 네트워크 라우터 단계에서 막아 줄 수 있는 보안 방화벽을 설계 하려한다.

## 6. 출처

"[기획]스마트홈 구멍 '송송'... 해킹 위협받는 기기들," *정보통신신문*, 2018 년 1 월 16 일 수정, 2018 년 3 월 31 일 접속, <http://www.koit.co.kr/news/articleView.html?idxno=72540>

"[사건추적]거실 IP 카메라 속 은밀한 사생활 동영상 해킹 유포," *중앙일보*, 2017 년 9 월 20 일 수정, 2018 년 3 월 28 일 접속, <http://news.joins.com/article/21950524>.

"사생활이 사라진다...IP 카메라·웹캠에 블랙박스까지 해킹," *노컷뉴스*, n.d. 수정, 2018 년 3 월 31 일 접속, <http://www.nocutnews.co.kr/news/4860086>.

"국내 IP 카메라 해킹한 해커, 페이스북에 해킹 정보와 화면 노출," *보안뉴스*, n.d. 수정, 2018 년 3 월 29 일 접속,

<http://www.nocutnews.co.kr/news/4860086http://www.boannews.com/media/view.asp?idx=57912>.

"프로토콜 종류," *Tistory*, last modified Jan 10, 2012, accessed March 18,2018,

<http://wryul12.tistory.com/entry/%ED%94%84%EB%A1%9C%ED%86%A0%EC%BD%9C%EC%9D%98-%EC%A2%85%EB%A5%98>.

"허브, 스위치 및 라우터 차이," *Naver Blog*, last modified June 24, 2016, accessed March 18,2018,

<https://m.blog.naver.com/PostView.nhn?blogId=chlalsdud61&logNo=220744988172&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F>.

"패킷 가로채기," *Wikipedia*, last modified Nov 18, 2017, accessed March 18,2018,

[https://ko.wikipedia.org/wiki/%ED%8C%A8%ED%82%B7\\_%EA%B0%80%EB%A1%9C%EC%B1%84%EA%B8%B0](https://ko.wikipedia.org/wiki/%ED%8C%A8%ED%82%B7_%EA%B0%80%EB%A1%9C%EC%B1%84%EA%B8%B0).

"패킷 스니핑의 원리," *Tistory*, last modified June 24, 2009, accessed March 18,2018,

<http://archiblue.tistory.com/41>.

"TCP Header 구조," *Tistory*, last modified Sep 14, 2013, accessed March 18,2018,

<http://mindnet.tistory.com/entry/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-%EC%89%BD%EA%B2%8C-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0-19%ED%8E%B8-TCP-Header-4%EA%B3%84%EC%B8%B5-TCP-%ED%97%A4%EB%8D%94-%EA%B5%AC%EC%A1%B0?category=702276>.

<http://mindnet.tistory.com/entry/%EB%84%A4%ED%8A%EC%9B%8C%ED%81%AC-%EC%89%BD%EA%B2%8C-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0-19%ED%8E%B8-TCP-Header-4%EA%B3%84%EC%B8%B5-TCP-%ED%97%A4%EB%8D%94-%EA%B5%AC%EC%A1%B0?category=702276>.

"TCP/IP 프로토콜에서 데이터 통신을 처리하는 방법," *Oracle*, last modified 2013, accessed March 19,2018,

<http://mindnet.tistory.com/entry/%EB%84%A4%ED%8A%EC%9B%8C%ED%81%AC-%EC%89%BD%EA%B2%8C-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0-19%ED%8E%B8-TCP-Header-4%EA%B3%84%EC%B8%B5-TCP-%ED%97%A4%EB%8D%94-%EA%B5%AC%EC%A1%B0?category=702276>.

<http://mindnet.tistory.com/entry/%EB%84%A4%ED%8A%EC%9B%8C%ED%81%AC-%EC%89%BD%EA%B2%8C-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0-19%ED%8E%B8-TCP-Header-4%EA%B3%84%EC%B8%B5-TCP-%ED%97%A4%EB%8D%94-%EA%B5%AC%EC%A1%B0?category=702276>.