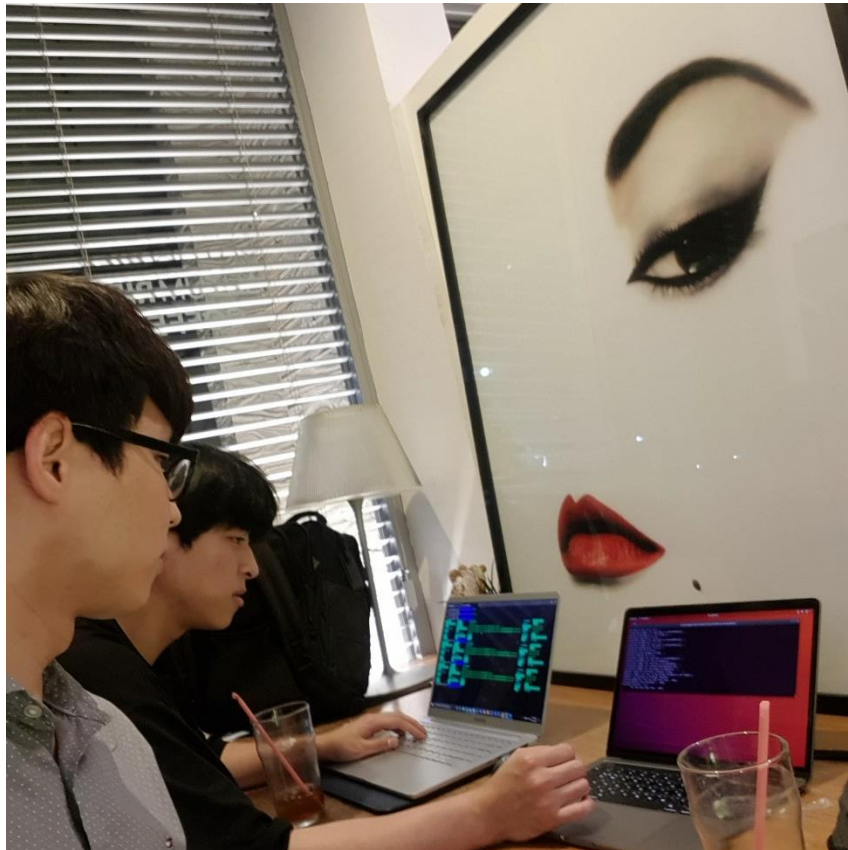


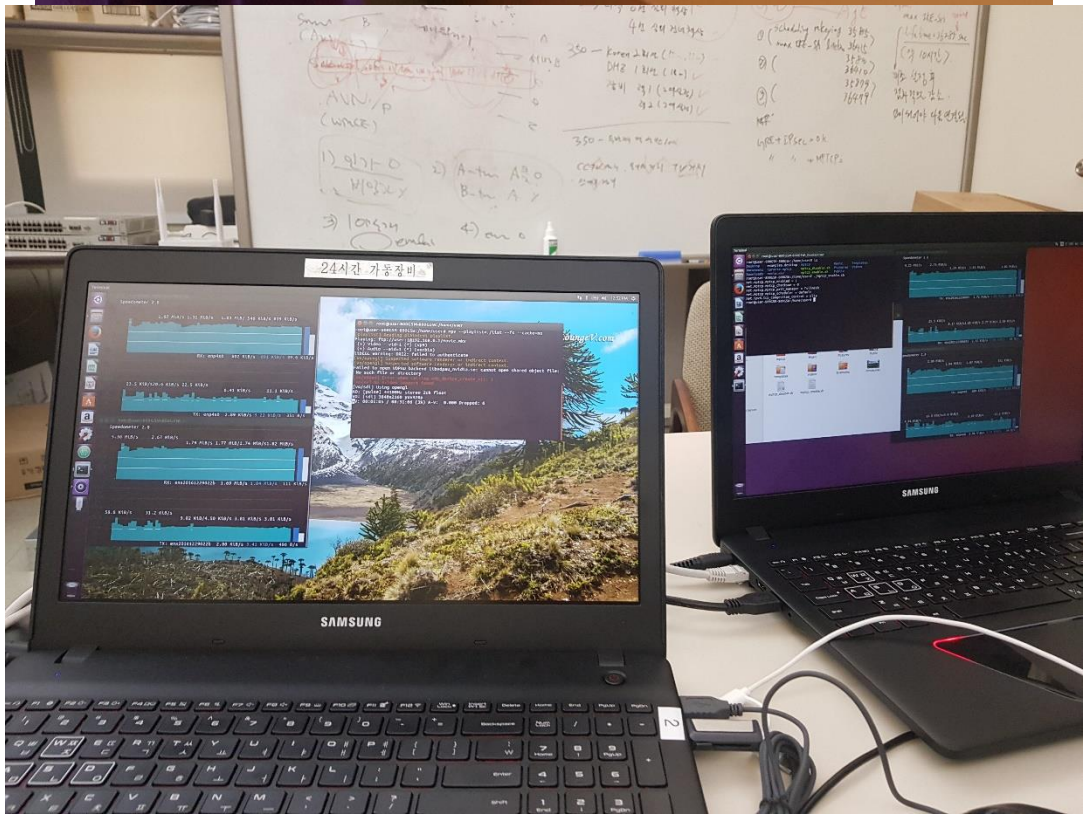
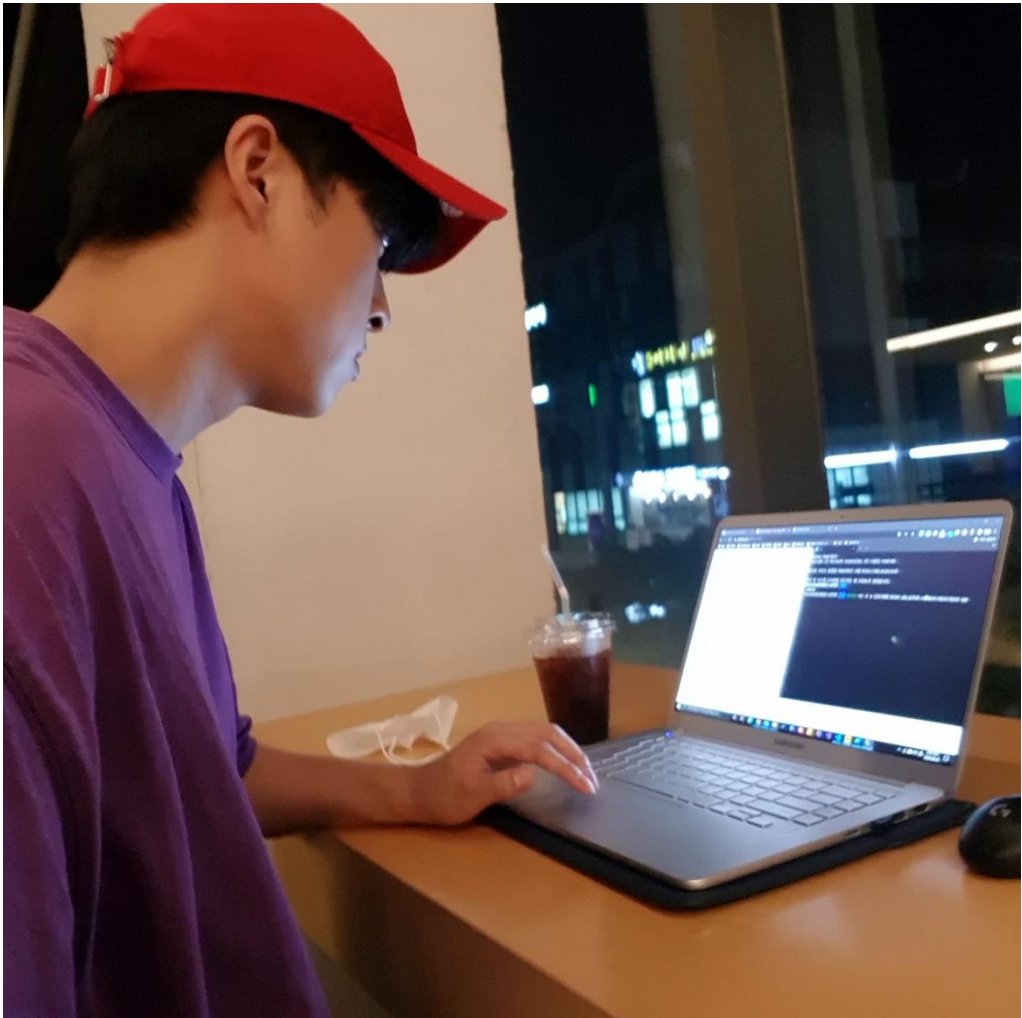
# 2020 공개SW 개발자대회 결과보고서

## □ 참가팀 개요

구 분		세부내용		
팀 명		MPSec		총 인원 ( 1 명)
팀 구 성		성명	소속	부서/학과
팀 장		김준희	MIDAS IT	기술 연구소
참가 지원	부문	<input type="checkbox"/> 학생		<input checked="" type="checkbox"/> 일반
	자유과제	<input type="checkbox"/> 인공지능	<input type="checkbox"/> 사물인터넷	<input type="checkbox"/> 빅데이터
	지정과제	<input type="checkbox"/> 블록체인	<input type="checkbox"/> 보안	<input type="checkbox"/> 모바일
프로젝트 활용방향		<input checked="" type="checkbox"/> 대회참가 <input type="checkbox"/> 졸업작품 <input type="checkbox"/> 비즈니스화 (창업 등) <input checked="" type="checkbox"/> 직접입력           ( 개인 오픈소스 관리 )		

## □ 참가팀 활동 사진

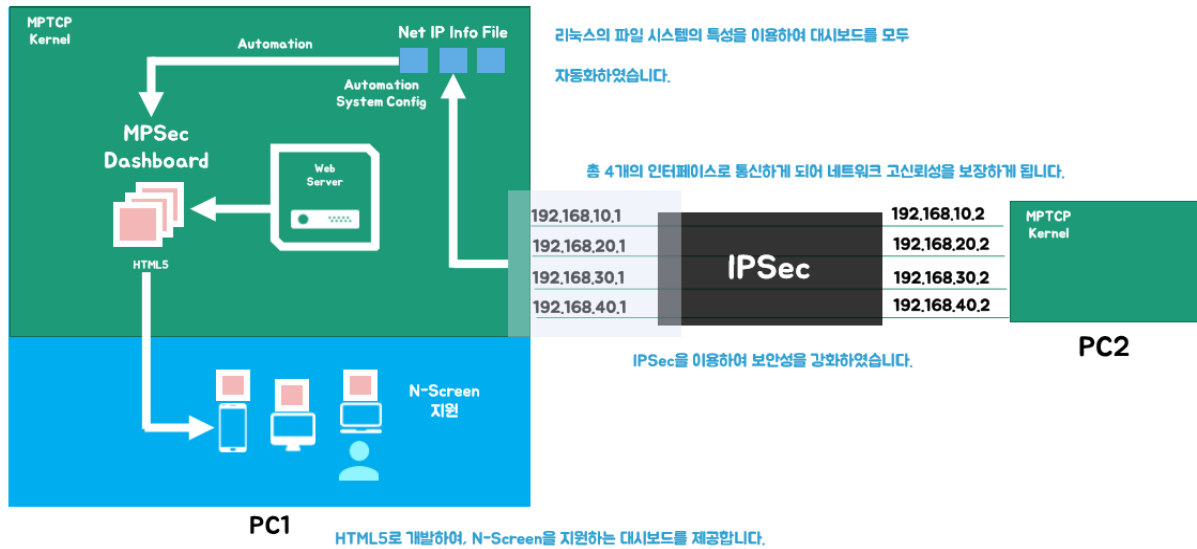




## □ 출품작 결과 보고서

출품작명	MPSec
프로그램 등록URL	Github : <a href="https://github.com/MPSec/MPSec">https://github.com/MPSec/MPSec</a> Youtube : <a href="https://youtu.be/gumSyz-ub-o">https://youtu.be/gumSyz-ub-o</a>
출품작 소개 (요약)	One Stop MPTCP Service (단 한번의 클릭으로 다중 전송 경로 기술을 활용한 안정적인 네트워킹 환경 제공)
작품 세부 내용	
<h3>1. 개발배경 및 목적</h3> <p>MPSec (Multipath Security)은 다중 경로 전송 기술인 MPTCP 프로토콜을 사용하여 고신뢰, 고생존성을 지원하여 네트워크 장애 상황에 효과적으로 대처할 수 있는 매우 안정적인 네트워킹을 제공합니다. 다중 경로 전송 기술은 사용처나 활용방안이 명확하며, 잘 활용할 수만 있다면 정말 다양한 기술과 연관되어 사용할 수 있다고 생각했습니다. 그러나 커널 컴파일 및 Linux System의 세부 설정, 매우 적은 양의 레퍼런스 눈에 보이지 않는 네트워크 인터페이스의 대역폭 측정, 실제로 연결되었는지 확인하는 방법 모두 네트워크 엔지니어가 아니라면 활용하기 매우 힘든 상태였습니다. 실제로 다양한 대기업에서는 다중 경로 전송 기술을 활용하여 이미 여러 기술들을 더 강력하게 사용하고 있지만, 일반 개발자 개개인에게는 정말 생소하고 접근하기 어려운 기술이었습니다. 본 프로젝트는 오픈소스 철학에 맞게 강력한 기술을 누구나 쉽게 사용할 수 있도록 구성하여 해당 원천 기술을 바탕으로 다양한 프로젝트들을 만들고자 하였습니다.</p> <h3>2. 개발환경 및 개발언어</h3> <h4>2.1 개발환경</h4> <ul style="list-style-type: none"><li>- Linux Ubuntu 16.04, MPTCP Protocol</li><li>- Docker</li><li>- Tomcat 8.5</li><li>- FFServer, FFMpeg, IPSec</li></ul> <h4>2.2 개발언어</h4> <ul style="list-style-type: none"><li>- C, C++ (Kernel, System Query API)</li><li>- HTML/CSS/JS, Java (Web Application Front-End, Back-End)</li><li>- Roff, Markdown (Documentation)</li><li>- Shell Script, DSLDomain Specific Language (DevOps)</li></ul> <h3>3. 시스템 구성 및 아키텍처</h3> <h4>3.1 Testbed 구축 환경</h4>	

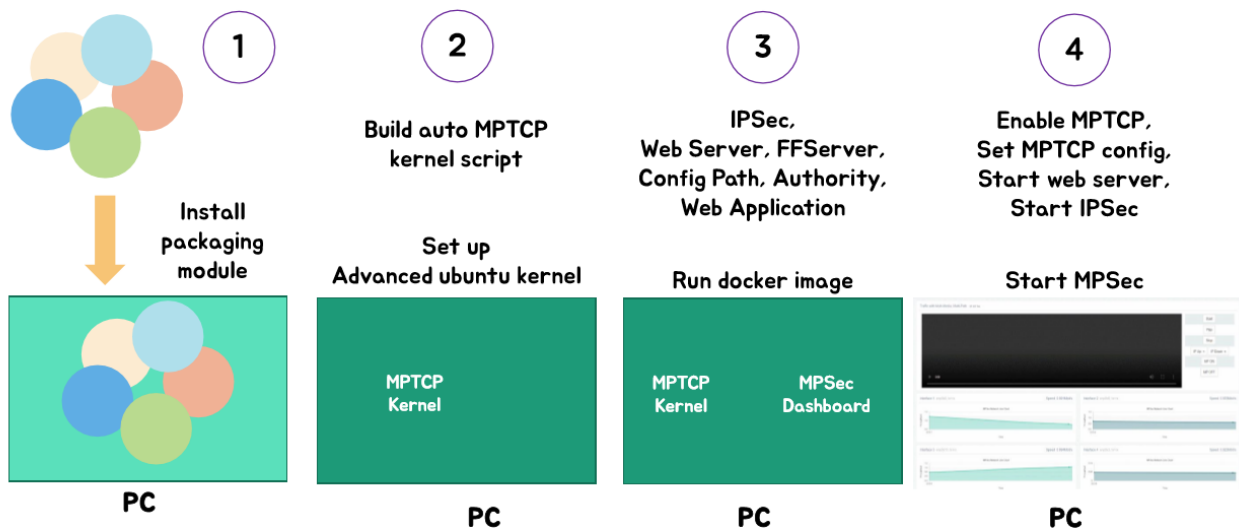
## • Example Testbed Structure



PC1과 PC2에 MPsec Auto Kernel Patch Script를 이용하여 MPTCP Kernel을 부팅합니다. 먼저 Multi Path 동작 확인의 경우, PC2에서 동영상 파일을 PC1으로 가져와 FFMpeg와 FFServer를 통해 WebM 코덱으로 인코딩하여 브라우저에 실시간으로 스트리밍 합니다. 1초마다 PC1의 각 인터페이스 대역폭을 측정합니다. 측정된 데이터를 Chart JS를 이용해 그래프로 그려줍니다. Single Path일 때와 Multi Path일 때의 대역폭 현황을 실시간으로 확인할 수 있습니다. 두 번째 IPsec 동작 확인의 경우, PC1에서 PC2의 해당 파일을 FTP 프로토콜을 이용해 가져옵니다. 구현한 Packet Capture Program과 Shell Script 코드를 동작시켜 Packet의 내용을 캡처하여 브라우저에 보여줍니다. IPsec이 동작된 경우 암호화된 Packet을 보여주며, 아닐 경우 해당 데이터의 내용이 그대로 나오는 것을 확인할 수 있습니다.

## 3.2 DevOps 자동화 스크립트 동작 과정

### • DevOps 인프라 동작 과정



1. MPsec 프로젝트를 Github에서 Clone합니다.
2. MPsec Auto Kernel Patch Script를 이용하여 Advanced Ubuntu Kernel을 Patch 합니다.
3. Github에서 확인할 수 있는 Docker 명령어를 통해 Docker Image를 다운받고 실행합니다.
4. 각종 Package 및 Kernel 설정이 완료된 이미지를 통해 Dashboard를 localhost에서 사용할 수 있습니다.



## 4. 프로젝트 주요기능

구체적인 핵심 기능은 아래와 같이 5가지가 있습니다.

- MPsec의 Auto 커널 패치 스크립트를 통해 어려운 MPTCP 커널 환경을 단 한번의 클릭으로 구성할 수 있습니다.
- 실시간 모니터링 어플리케이션 및 인터페이스 설정과 같은 환경 설정을 Docker를 통해 매우 간단히 사용할 수 있으며 관리됩니다.
- 실시간 모니터링을 Dashboard 통해 사용자가 효과적으로 관리할 수 있습니다.
- 패킷은 IPsec을 통해 보안됩니다.
- 유사한 대역폭으로 다중 인터페이스를 통해 통신할 때 네트워크 속도가 향상됩니다.
- TCP 프로토콜로 통신이 가능하기 때문에 호환성이 좋습니다.

## 5. 기대효과 및 활용분야

### 5.1 기대효과

누구나 쉽게 고신뢰, 고생존성을 지원하여 네트워크 장애 상황에 효과적으로 대처할 수 있는 매우 안정적인 네트워킹 환경을 단 한번의 클릭으로 구현할 수 있습니다. 그로인한 네트워크 개발 영역의 러닝 커브가 낮아짐으로써 좀 더 비즈니스 로직 및 어플리케이션 영역에서 활발한 연구가 이루어질 수 있습니다.

### 5.2 활용분야

#### 5.2.1 Army

군대와 같이 중요한 서버가 있으며 네트워크가 끊겨서는 안 되는 장소에서 다중 경로 전송을 통해, 만약 하나의 인터페이스가 끊겨도 다른 인터페이스로 데이터를 전송함으로써, 네트워크의 생존성을 증가시킬 수 있습니다. 또한 IPsec을 통해 보안이 중요한 군시설에 Packet의 내용을 볼 수 없도록 구성할 수 있습니다.

#### 5.2.2 High-definition streaming

고화질 영상의 경우 빠르고 안정적인 데이터 전송을 유지해야 합니다. MPTCP를 이용해 여러 개의 인터페이스를 연결하여 빠르고 끊기지 않게 영상을 전송할 수 있습니다.

## 6. 기타 (출품작에 대한 추가 설명 및 PT 자료 등 첨부 가능)

### 6.1 MPTCP Structure

MPTCP는 전송계층 프로토콜의 한 종류로 사용자와 서버 사이에 2가지 이상의 네트워킹 인터페이스를 활용하여 하나의 TCP 스트림을 다중 경로로 전송하는 전송 계층의 프로토콜을 의미합니다. 이때 각각의 네트워킹 인터페이스로 나뉘어진 TCP 스트림은 subflow라 부릅니다.

하나의 어플리케이션은 MPTCP를 활용하여 여러 개의 TCP subflow를 생성하고 생성된 subflow들로 동시에 전송하여 데이터 전송률을 높이거나 번갈아가며 사용하여 인터페이스의 부하를 줄일 수 있습니다. 또한 MPTCP는 기존 TCP Header의 옵션 필드를 활용하여 사용자-서버 간 MPTCP 제어 데이터를 송수신하기 때문에, MPTCP가 설치되지 않은 단말이더라도 TCP로 인식하여 처리하기 때문에 호환성이 뛰어나다고 할 수 있습니다.

## 6.2 MPTCP Features

### 6.2.1 Path Manager

MPTCP Path-Manager는 사용자-서버 간의 다중 경로 생성에 있어 어떠한 방식으로 다중 경로를 생성할지를 결정하는 기능으로 크게 3가지의 설정 방법을 제공합니다.

- **default - default** 방식은 MPTCP를 사용하지 않는 일반적인 단일 경로 전송 방식으로 단말이 자기가 가진 인터페이스 정보와 이를 통해 만들 수 있는 subflow 정보를 전달하지 않습니다.
- **fullmesh - fullmesh** 방식은 MPTCP에서 가장 많이 사용되는 방식으로 사용자가 가진 인터페이스의 개수와 서버가 가진 인터페이스의 개수를 조합하여 모든 경우의 수로 가는 다중 경로를 생성하는 방식입니다.
- **ndiffports - ndiffports** 방식은 기존의 하나의 인터페이스만을 사용하여 통신하는 환경에서 송신측 단말의 TCP 포트를 여러 개 생성하여 가상으로 다중 경로를 생성하는 방식입니다.

### 6.2.1 Scheduler

MPTCP Scheduler는 Path-Manager를 통해 사용자-서버 사이에 생성된 다중 경로 중 어떠한 경로를 선택하여 데이터를 전송할지 결정하는 기능으로 크게 3 가지의 설정 방법을 제공합니다.

- **default - default** 스케줄러는 MPTCP에서 가장 많이 사용되는 스케줄러로 여러 가지 subflow 중 가장 RTT가 적은 subflow를 선택하여 해당 subflow의 congestion window가 가득 찰 때까지 데이터를 전송합니다. 만약 선택된 subflow의 congestion window가 가득 찬 경우 그 다음 적은 RTT를 가지는 subflow를 선택하여 데이터를 전송합니다.
- **roundrobin - roundrobin** 스케줄러는 존재하는 모든 subflow에 설정된 TCP segment 수에 따라 순차적으로 데이터를 전송하는 방식으로 인터페이스 별 부하분산 기능을 극대화하여 사용할 때 주로 사용합니다.
- **redundant - redundant** 스케줄러는 생성된 모든 subflow에 동일한 데이터를 중복적으로 송신하는 방식으로 이를 수신한 단말에서는 수신한 각 패킷의 sequence number와 time stamp 값을 비교하여 동일 데이터 중 먼저 수신한 데이터를 가지고 이후에 들어오는 데이터는 버리는 방식으로 동작하며 가용 가능한 네트워크 대역폭을 희생하여 신뢰성을 보장하는 서비스를 사용하고자 할 때 주로 사용합니다.

### 6.2.1 Congestion Control

MPTCP Congestion Control는 기존 TCP의 Congestion Control과 유사한 기능을 담당합니다. 하지만 기존 TCP Congestion Control은 하나의 경로에 대한 congestion window size를 계산하는 Uncoupled 방식의 알고리즘으로 여러 subflow를 가지는 MPTCP 환경에서 사용하기 적절하지 않습니다. 따라서 MPTCP Congestion Control은 MPTCP Scheduler의 결정을 보다 효율적으로 하기 위해 congestion window size 계산 시 가용 가능한 모든 subflow의 상황을 보며 window size를 결정하는 Coupled 방식의 알고리즘을 제공합니다. 제공하는 알고리즘은 크게 4가지가 있습니다.

- LIA (Linked Increase Algorithm)
- OLIA (Opportunistic Linked Increase Algorithm)

- wVegas (Delay-based Congestion Control for MPTCP)
- BALIA (Balanced Linked Adaptation Congestion Control Algorithm)

### 6.2.1 Reordering

MPTCP Reordering 기능은 여러 subflow로 전송한 데이터를 수신한 단말이 이를 원상복구 시키기 위해 데이터 재조립을 사용합니다.