

[네트워크프로그래밍] 프로젝트 정의서

소속 : IT대학 – 소프트웨어학부

학번 : 20192851, 20192829, 20192847

이름 : 윤성준, 박민수, 양조은

프로젝트명 : Cut-Thorough(컷또로)

1. 프로젝트 목표

본 프로젝트의 목표는 HLS / MPEG-DASH 패킷 분석을 통해 HLS / MPEG-DASH 기반 동영상 콘텐츠를 차단하는 백그라운드 프로그램을 개발하는 것이다. 기본적으로 사용자에게 패킷 차단 On/Off를 제공하며, Off를 할시에 HTTP 패킷을 막고 HTTPS 패킷을 재요청하여 http기반으로 설계된 두 프로토콜의 보안성을 높인다. 이를 위해 네트워크 트래픽을 모니터링하고, 프로토콜 유형을 분석하는 시스템이 설계한다. 결과적으로, 이 백그라운드 시스템은 프로토콜 차단을 통해 효과적으로 동영상을 차단하고 안전한 콘텐츠 접근을 보장한다. 이러한 솔루션은 다양한 환경에서 적용될 수 있으며, 기업과 같은 조직 내부에서 차단 프로그램으로 큰 도움이 될 것이다.

2. 필요성

(1) 동영상 차단

기업과 같은 조직 내부에선 보안 강화 및 문제 예방, 네트워크 리소스 관리 등 다음의 이유로 특정 콘텐츠에 대한 접근을 규제하는 시스템을 필요로 한다.

- 1) 콘텐츠 규제: 기업은 콘텐츠 규제 정책을 시행하고 직원들이 업무 시간 동안 부적절하거나 업무와 관련이 없는 비디오 콘텐츠에 액세스하지 못하도록 막을 수 있다. 이를 통해 직원의 생산성을 높이고 방해 요소를 줄일 수 있다. 또한 교육 기관이나 의료 서비스 제공업체와 같은 특정 산업 또는 조직에는 스트리밍 서비스 사용과 관련하여 특정 규정 준수 또는 규제 요건이 있을 수 있다.
- 2) 보안 문제 예방 및 강화: 비디오 콘텐츠에 대한 무단 액세스를 방지하고, 스트리밍을 통해 전달되는 멀웨어 또는 악성 코드의 위험을 줄이며, 특정 스트리밍 서비스와 관련된 잠재적인 보안 취약성에 대한 노출을 줄일 수 있다. 기업은 이러한 시스템을 활용하여 네트워크 보안을 강화하고 자산을 보호할 수 있다.
- 3) 대역폭 관리 및 비용 절감: 조직 네트워크 망에서 동영상 접근을 차단함으로써 네트워크 대역폭 소비를 제어하여 비용을 절감하고 네트워크 리소스를 더 효과적으로 관리할 수 있다. 이는 시간이 지남에 따라 상당한 비용 절감 효과를 가져올 수 있다.

본 프로젝트에서는 위와 같은 요구 사항에 대하여, 동영상 스트리밍 패킷을 차단하는 백그라운드 실행 서비스를 제공한다. 위 기능을 위하여 URL을 필터링하는 방식이 사용되기도 하는데, 기존 URL 필터링 방식과의 비교는 아래 3. 특징점에 자세히 기술하였다.

(2) HTTP 패킷 처리

일반적인 HTTP 프로토콜은 데이터를 암호화하지 않고 전송하기 때문에, 중간에서 데이터를 탈취하거나 변조될 위험이 있다. 이에 반해 HTTPS는 데이터를 SSL/TLS 암호화 프로토콜을 사용해 암호화하고, 인증서를 통해 서버의 신원을 검증한다. 본 서비스의 사용자는 필요에 의해 해당 기능을 Off함으로써 일시적으로 동영상 콘텐츠를 재생할 수 있는데, 이때 HTTP 패킷은 Drop하여 보안성을 높인다.

3. 특징점

기존의 URL 필터링 방식은 다양한 웹사이트와 콘텐츠 유형에 대응할 수 있지만, HLS / MPEG-DASH 프로토콜 차단은 특정 프로토콜에 초점을 맞추어 스트리밍 콘텐츠 제어에 효과적이다.

특정 프로토콜 차단 방식은 유지 보수가 더 적게 필요하고, 우회하기 어렵다는 장점이 있다.

- 1) 스트리밍 콘텐츠 제어에 더 집중할 수 있다: URL 필터링을 사용하면 사이트 전체를 차단하지 않고 사이트 내의 특정 페이지나 콘텐츠에 대한 액세스를 차단하기가 어려울 수 있다. 반면 본 프로젝트의 특정 프로토콜 차단 방식은 관련된 특정 트래픽을 타겟팅하여 해당 트래픽만 차단할 수 있으므로 사용자는 동일한 사이트의 다른 콘텐츠에 액세스할 수 있다.
- 2) 유지보수 감소: 기존 URL 필터링의 경우 차단 목록을 지속적으로 업데이트해야 하는 반면, 특정 프로토콜을 차단 방식은 그러한 유지 보수의 번거로움을 절감할 수 있다.
- 3) 우회 방지: DASH 프로토콜 차단은 기본 스트리밍 기술을 대상으로 하기 때문에 사용자가 우회하기가 더 어려울 수 있다. 프록시 서버, VPN 또는 기타 방법을 사용하여 URL 필터링을 우회할 수 있지만, DASH 프로토콜 차단을 우회하는 것은 더 어렵다.

4. 개발환경

- a. Python 3.11
- b. PyQt5 5.12
- c. Pyshark 0.5.3
- d. OpenSSL 1.1.1

5. 관련 기술 (사용한 오픈 소스 소개 등)

- a. OpenSSL 1.1.1
 - OpenSSL은 TLS 및 SSL 프로토콜 구현을 위한 오픈 소스 라이브러리로, 암호화 및 인증 기능을 제공한다. 이 프로젝트에서는 HTTPS 통신 및 프로토콜 감지를 위해 OpenSSL을 사용한다. OpenSSL은 Apache License 2.0을 따른다.
- b. Pyshark 0.5.3
 - PyShark는 Wireshark를 기반으로 한 Python 라이브러리로, 네트워크 패킷 캡처 및 분석을 돕는다. PyShark는 오픈 소스 프로젝트로, MIT 라이선스를 따른다.
- c. PyQt 5.12
 - PyQt는 Qt 프레임워크를 Python으로 사용할 수 있게 해주는 바인딩 라이브러리이다. PyQt는 Python 프로그래밍 언어를 사용하여 Qt 프레임워크의 기능과 사용자 인터페이스를 쉽게 구현할 수 있도록 돕는다. 이 프로젝트에서는 GPL 라이선스에 해당되는 PyQt5 기능을 사용한다.
- d. MPEG-DASH 프로토콜
 - MPEG-DASH는 ISO/IEC 표준의 최초의 적응 비트레이트 HTTP 기반 스트리밍 프로토콜이다. 전통적인 HTTP 웹 서버로부터 전달되는, 인터넷을 경유하는 미디어의 고품질 스트리밍을 가능케 하는 적응 비트레이트 스트리밍 기술이다. 애플의 HTTP Live Streaming(HLS) 솔루션과 비슷하게 MPEG-DASH는 내용을 일련의 작은 크기의 HTTP 기반 파일 세그먼트들로 분리시킴으로써 동작하며, 각 세그먼트는 영화나 스포츠 이벤트 생방송 등 잠재적으로 수시간에 걸친 내용물의 재생 시간의 짧은 간격(interval)을 포함하고 있다. MPEG-DASH가 사용하는 전송 프로토콜은 TCP이다.
- e. HLS 프로토콜
 - HLS 프로토콜은 Apple에서 개발한 HTTP Live Streaming의 약자로, 동적 비디오

오 스트리밍을 위한 프로토콜이다. HLS는 콘텐츠를 작은 세그먼트로 나누어 HTTP를 통해 전송하며, 클라이언트는 필요한 세그먼트를 요청하고 전송 속도에 따라 재생한다. HLS는 여러 가지 비디오 코덱과 함께 사용할 수 있으며, 다양한 장치와 플랫폼에서 호환성이 좋다.

f. TCP

- TCP(Transmission Control Protocol)는 인터넷 프로토콜 스위트의 핵심 프로토콜 중 하나로, 데이터를 신뢰성 있게 전송하기 위한 프로토콜이다. TCP는 인터넷 상에서 데이터를 분할하여 전송하고, 이를 수신측에서 다시 합치는 역할을 한다. 이를 통해 데이터 전송 과정에서 발생할 수 있는 오류나 손실을 최소화하고, 데이터 전송의 안정성을 보장한다.

6. 수업 연관성

프로젝트에서 네트워크 패킷을 분석하기 위해, 강의에서 습득한 소켓 프로그래밍 기술과 소켓 API 활용에 대한 기본 개념이 필요하다. 이를 바탕으로 네트워크 패킷을 효과적으로 분석할 수 있다. 또한, 강의에서 배운 TCP/UDP 소켓과 서버 클라이언트 모델에 대한 이해를 활용하여 프로젝트에서 HLS 및 MPEG-DASH 패킷을 차단할 수 있다. 프로젝트에서 패킷 캡처 및 분석을 위해 Packet Socket을 사용하며, 에러 처리에 대한 이해가 필요한데, 이에 대한 내용은 10주차 강의에서 배워 활용할 수 있다. 프로젝트에서 HTTP 패킷을 막고 HTTPS 패킷 재요청을 구현하기 위해서는 HTTP 및 웹 서버에 대한 기본 지식이 필요하다. 12주차 강의에서는 간단한 웹 서버와 HTTP 기본 개념에 대한 응용 프로그램 구현 및 테스트를 함으로서 관련 지식을 습득할 수 있다. 마지막으로, 프로젝트에서 보안성을 높이기 위해 HTTP 패킷을 차단하고 HTTPS 패킷을 사용하는 방식을 구현하는데 13주차 강의에서는 OpenSSL을 사용하여 보안 통신을 구현하는 방법을 습득하고 HTTPS에 대한 응용 프로그램 구현 및 테스트를 진행한다.

이처럼 각각의 강의 내용이 프로젝트의 다양한 측면에서 중요한 역할을 한다. 프로젝트를 진행 함으로써 소켓, TCP 통신, HTTP 등등 네트워크 프로그래밍의 전반적인 이해도가 늘 것으로 기대가 된다.

7. 프로젝트 Architecture

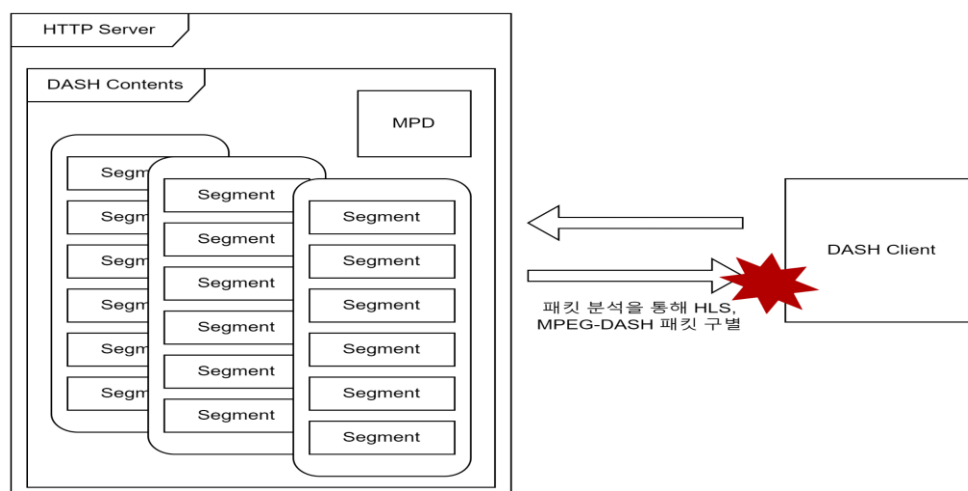


그림1. Cut-Thorough 프로그램 blocking position

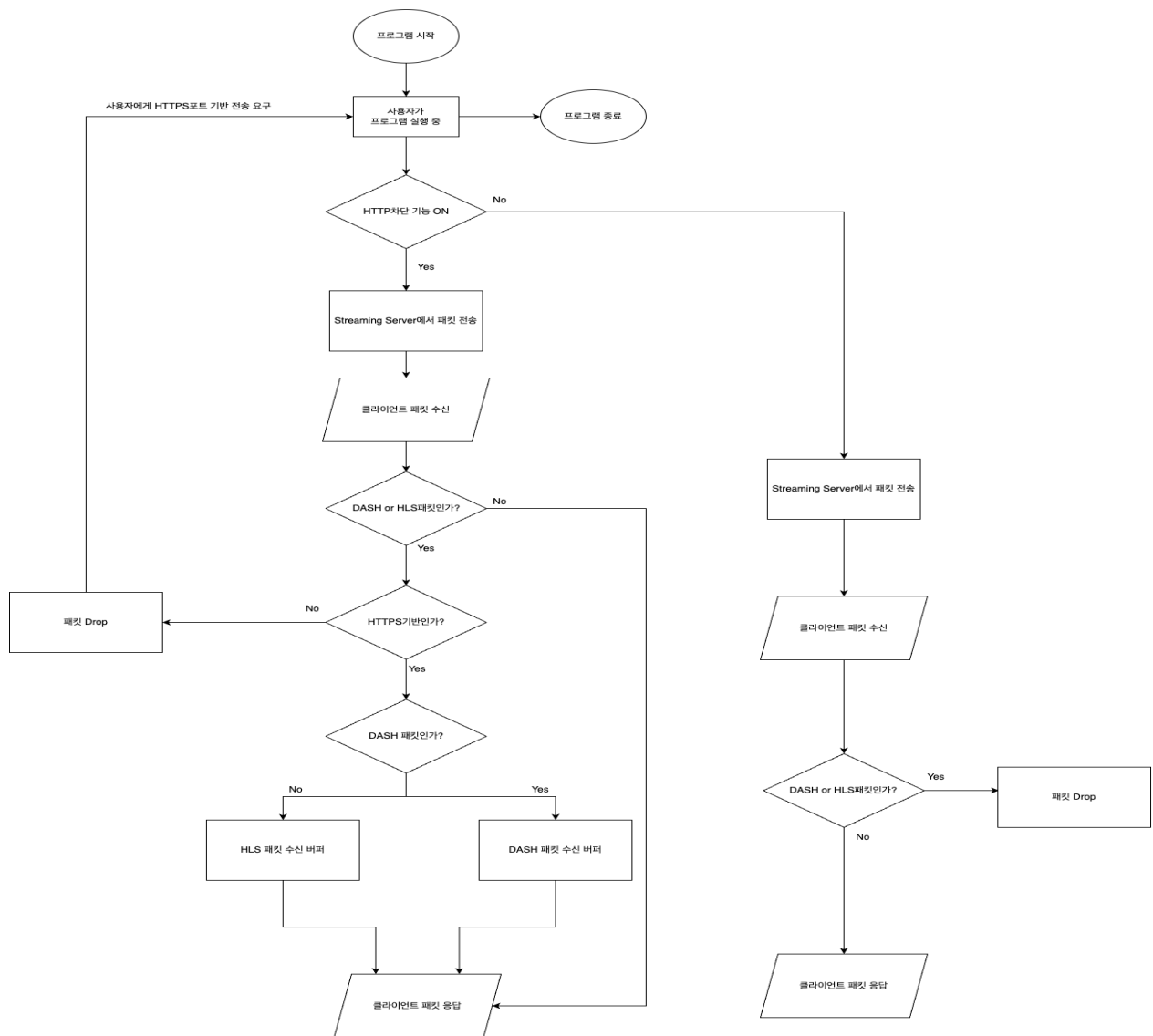


그림2. Cut-Thorough Flow Chart

8. 참고 문헌

- (1) 정상호, 2011, HTTP상에서 동적 적응적 스트리밍 소프트웨어 개발
- (2) 한국전자통신연구원, 2011, HTTP상에서 동적 적응적 스트리밍 소프트웨어 개발