SAMSUNG SW ACADEMY FOR YOUTH

對外秘

프로젝트 명세서

네트워크 보안 with Wireshark

對外秘

목차

1. 프로젝트 개요

본 과제는 Wireshark 를 사용해 네트워크 보안의 필요성에 대해 알아보는 과제입니다.

보안(security)은 일반적인 애플리케이션 개발자의 영역이라고 보기 어렵습니다. 그보다는 보안 전문가가 다루는 것이 맞을 것입니다. 하지만 애플리케이션 개발자에게도 보안은 간과할 수 없는 분야입니다. 잘못된 설계와 코드, 환경 설정은 보안 결함을 제공하기 때문입니다.

그 중 네트워크는 보안에 있어 가장 주의해야 할 부분입니다. 네트워크는 기본적으로 공개되어 있기 때문입니다. 네트워크에서 이동하는 패킷(Packet)은 누구나 열어볼 수 있습니다. 예를 들어 암호화를 제공하지 않는 HTTP 를 사용해 사용자 ID 와 비밀번호를 전송한다면, 서버까지 가는 네트워크 경로상의 컴퓨터에서는 해당 정보를 볼 수 있으며 사용자 정보를 탈취할 수 있을 것입니다

Wireshark 는 패킷 분석 도구로 네트워크를 오가는 패킷을 읽고 분석해줍니다. 해킹, 크래킹 등 잘못된 용도로 사용되는 경우도 있으나, 올바른 용도로 사용한다면 보안 취약점 발견이나 애플리케이션의 디버깅에도 도움을 줄 수 있습니다. 또한 네트워크를 깊이 있게 학습하는데 필수적인 도구이기도 합니다.

본 과제를 통해 패킷 스니핑(Packet Sniffing)을 실습해보고 보안의 필요성에 대해 체감해 보시기 바랍니다.

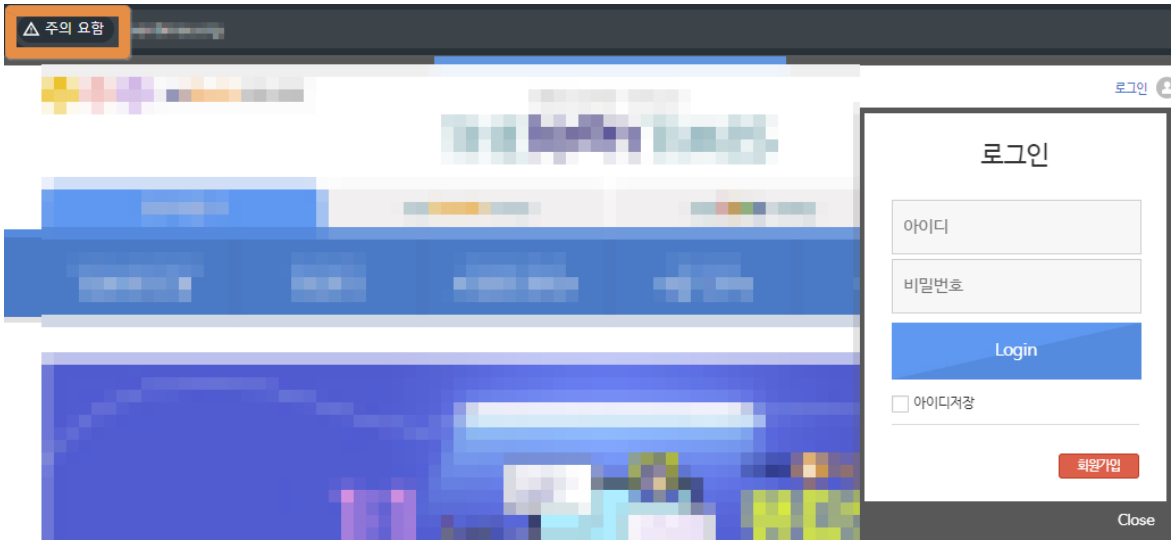
2. 웹 보안 기초

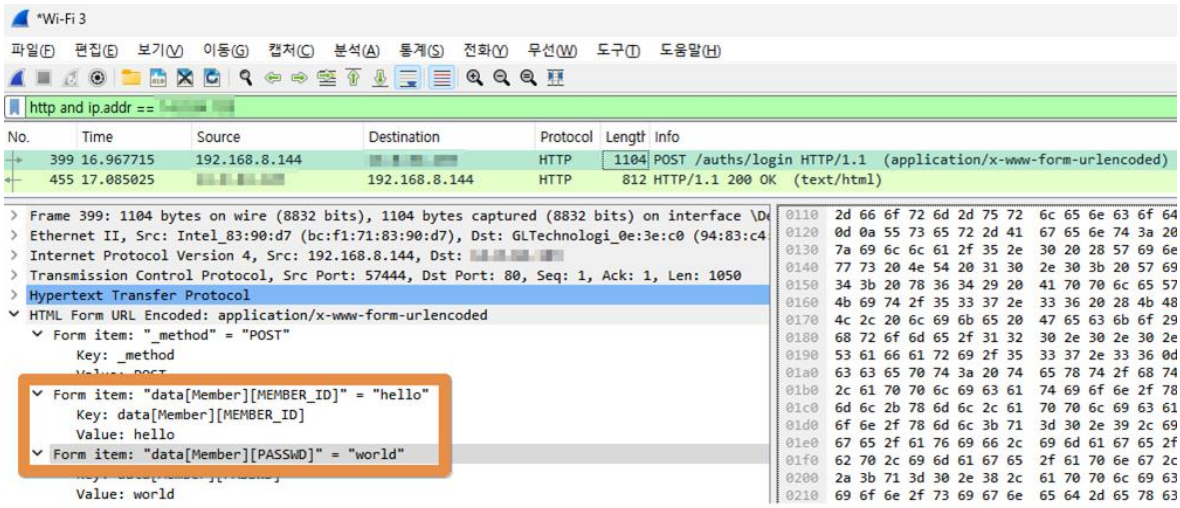
웹 애플리케이션의 보안은 광범위한 주제이며 애플리케이션 개발자와 보안 전문가의 협업이 필요한 영역입니다. 웹 애플리케이션에는 SQL/Code Injection, XSS(Cross Site Scripting), CSRF(Cross Site Request Forgery), 세션 탈취, 인증 우회 등의 보안 위협이 있으며 이외에도 네트워크 레벨에서 생길 수 있는 보안 위협에 노출됩니다. 다음은 각 보안 위협에 대한 간략한 설명입니다.

- SQL/Code Injection : SQL 또는 코드를 애플리케이션에 주입하여 의도치 않은 동작을 하게 만드는 공격 방식입니다.
- XSS : 악의적 스크립트를 사용자 브라우저에서 실행되도록 하여 정보를 탈취하는 공격 방법입니다.
- CSRF : 사용자가 의도치 않은 요청(request)을 발생하게 만드는 공격 방식입니다.
- 세션 탈취 : 쿠키 또는 인증 토큰을 탈취하여 해당 사용자로 위장하는 공격 방식입니다.
- 인증 우회 : 애플리케이션의 결함을 이용하여 허가 받지 않은 API 등을 사용하는 공격 방식 입니다.

이러한 여러가지 보안 위협에 대해 최신의 브라우저와 웹 개발 프레임워크는 기본적인 방어를 제공합니다. 대표적으로 SOP(Same Origin Policy), CSP(Content Security Policy)가 있습니다.

여러 보안 위협 중 네트워크 레벨에서의 패킷 감청으로 인한 정보 유출도 흔히 생길 수 있는 위협입니다. 본 과제에서는 이 부분에 집중합니다. 암호화되지 않은 통신은 네트워크 상의 누구나 손쉽게 내용을 볼 수 있습니다. 다음은 HTTP 를 사용하고 있는 웹 사이트의 로그인 요청을 Wireshark 로 스니핑한 것입니다.





로그인 form 의 내용이 그대로 네트워크상에 노출됩니다. 이 정보는 패킷이 지나가는 네트워크 안에 속한 누구나 볼 수 있습니다.

따라서 웹 사이트의 최소한의 보안을 위해 암호화를 제공하는 HTTPS 를 사용하는 것이 좋습니다.

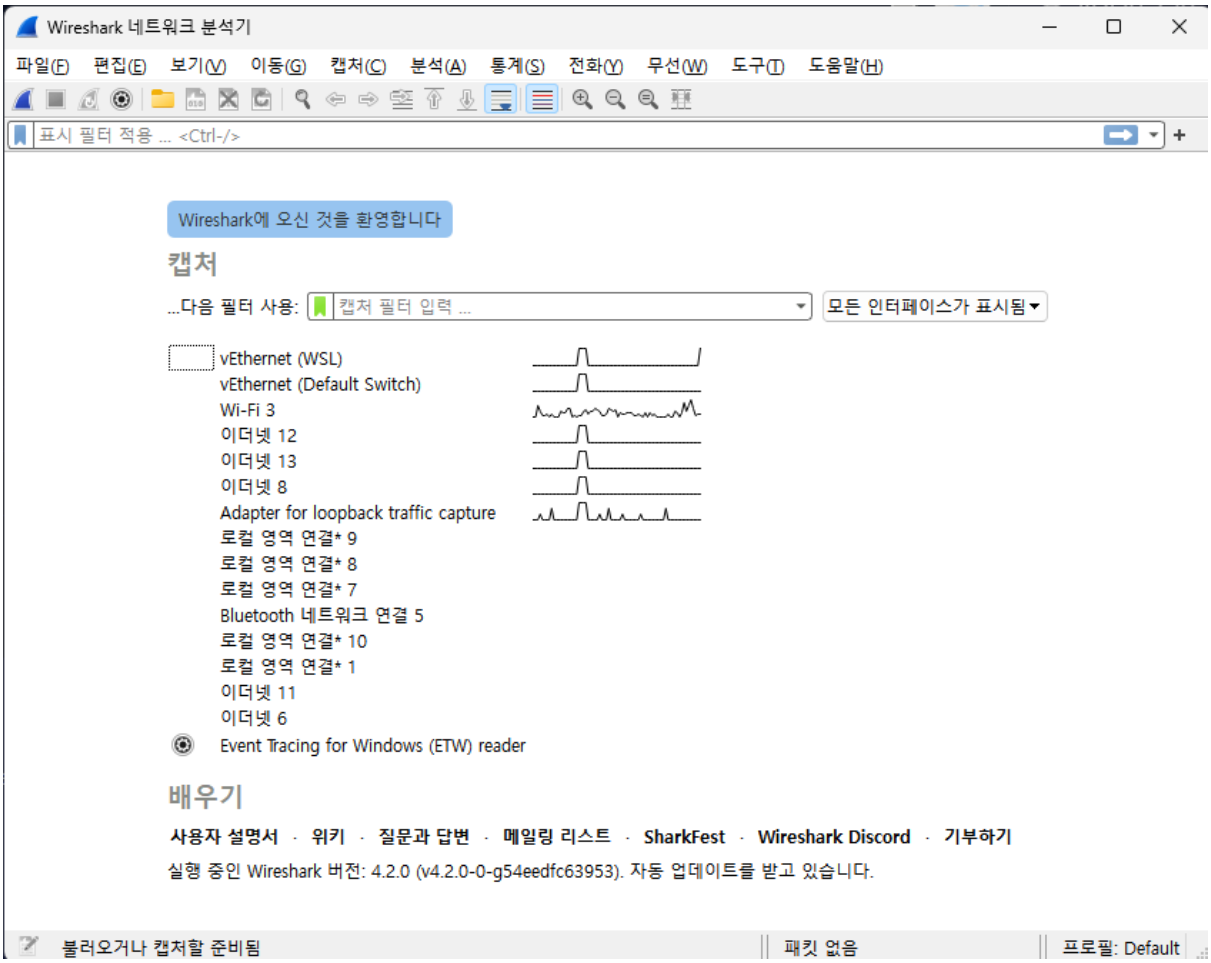
3. Wireshark 로 해보는 패킷 스니핑

Wireshark 는 GUI 기반의 패킷 분석 도구입니다. 윈도우와 맥 OS, 리눅스 환경에서 사용할 수 있습니다. Wireshark 와 유사한 도구로는 tcpdump, tcpflow, netsh trace 등이

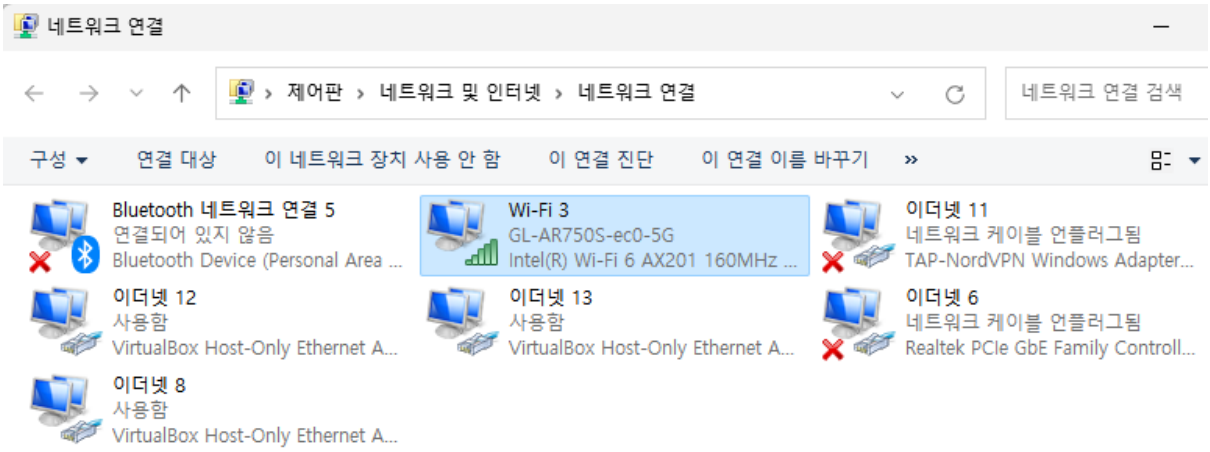
있습니다. 아래는 설치와 간단한 패킷 스니핑 안내입니다.

1. Wireshark 설치 : https://www.wireshark.org/download.html 로 이동하여 본인 환경에 맞는 Wireshark 를 다운로드 받고 설치합니다.

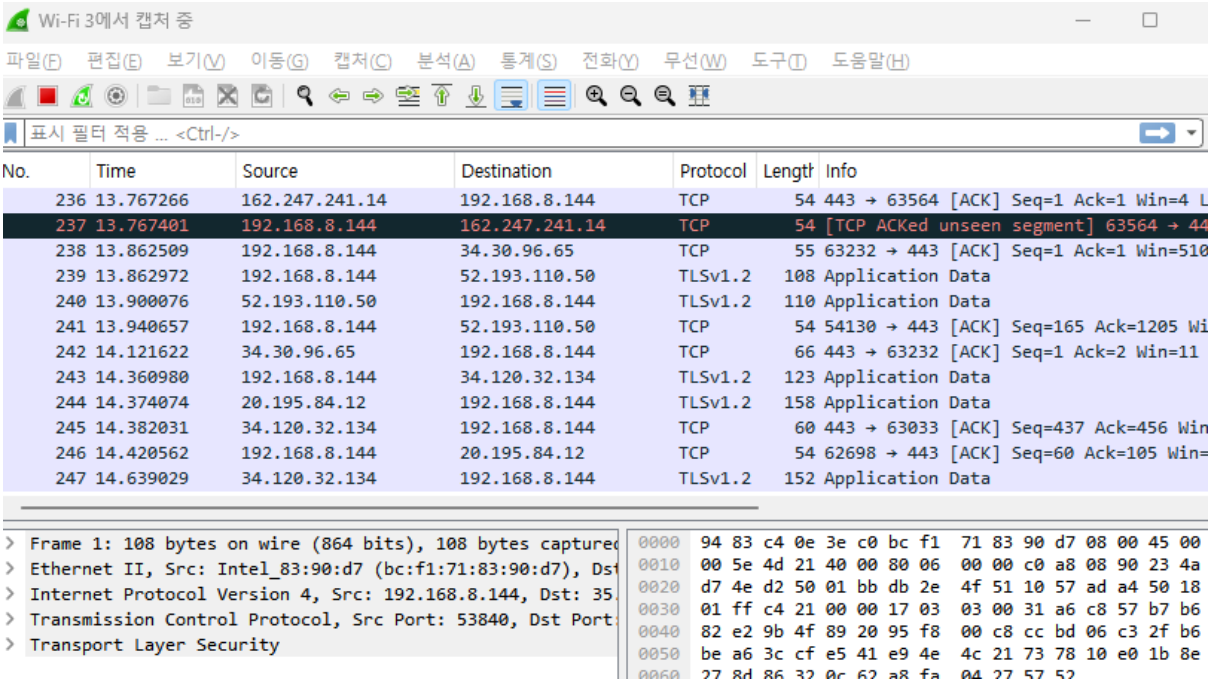
2. Wireshark 실행 : Wireshark 를 실행하면 다음과 같은 화면이 나타납니다.



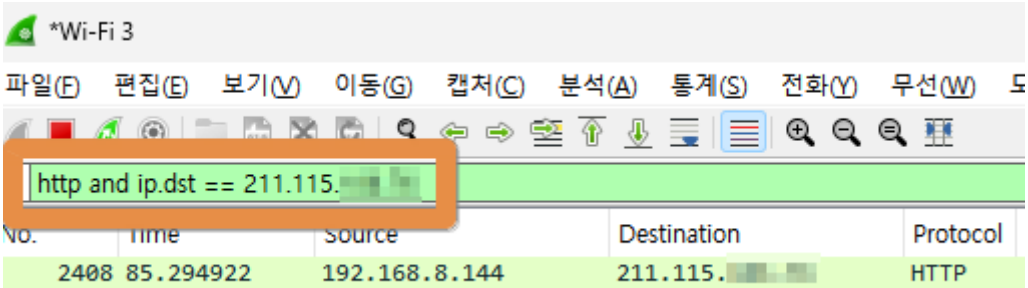
3. 캡처 장치 선택 : 현재 사용하고 있는 네트워크 장치를 선택합니다. 사용자에 따라 다르게 나타날 수 있습니다. '네트워크 연결' 이나 윈도우 명령 프롬프트 창에서 ipconfig 명령어를 통해 현재 사용 중인 네트워크를 확인할 수 있습니다.



4. 장치를 선택하면 Wireshark 에서 캡쳐를 시작합니다. 다음 창이 열리고 뭔가 알 수 없는 것들이 올라갑니다.



5. 필터를 통해 표시할 패킷을 선택할 수 있습니다. 다음은 HTTP 프로토콜을 사용하며 목적지 주소가 211.115.xxx.xxx 인 패킷만을 보기 위한 필터입니다.



4. 참고 자료

- [네트워크 해킹] Sniffing
  https://quio314.tistory.com/58

- HTTP vs HTTPS 차이, 알면 사이트의 레벨이 보인다.
  https://yozm.wishket.com/magazine/detail/130/

- Wireshark 패킷 분석을 통해 http 와 https 의 차이점 알아보기
  https://velog.io/@hyeewooon/http-https

- Wireshark 를 이용한 네트워크 계층별 공격 확인 방법
  https://www.igloo.co.kr/security-information/wireshark%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%9C-%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-%EA%B3%84%EC%B8%B5%EB%B3%84-%EA%B3%B5%EA%B2%A9-%ED%99%95%EC%9D%B8-%EB%B0%A9%EB%B2%95/

5. 과제

본 과제는 Wireshark 로 패킷 스니핑을 해보고 암호화되지 않은 통신이 어떤 문제를 일으키는지, 어떻게 문제를 보완할 수 있을지 조사해보는 두 부분으로 이루어져 있습니다.

1. 패킷 스니핑 실습

◆ 다음은 기본적으로 암호화를 제공하지 않아 패킷 감청에 취약한 프로토콜입니다.

- DNS(Donmain Name System) : 도메인 이름으로 IP 를 검색하는 프로토콜 입니다. 패킷 감청을 통해 어느 도메인에 접속했는지 알 수 있습니다.

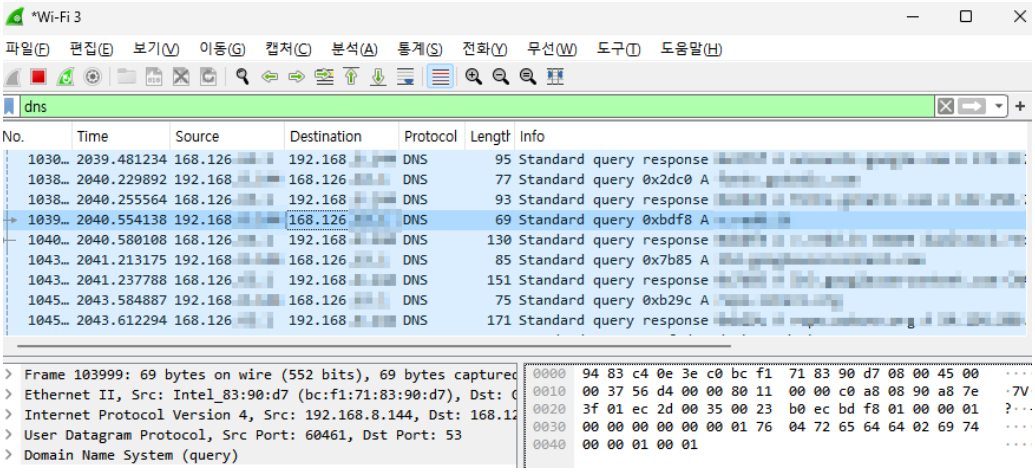
- FTP(File Transfer Protocol) : 파일 전송을 위해 많이 사용하는 프로토콜 입니다. 패킷 감청을 통해 사용자 계정과 비밀번호, 어떤 파일에 접근했는지 등을 알 수 있습니다.

- Telnet : SSH(Secure Shell) 이전에 원격 접속에 많이 사용하던 프로토콜 입니다. 감청시 시스템 계정/비밀번호/전송하는 명령어/응답 등 많은 정보를 알 수 있습니다.

- HTTP : 앞의 '웹 보안 기초'에서 다루었듯 HTTP 는 기본적으로 암호화를 제공하지 않습니다. 감청시 요청(Request)/응답(Response) 되는 모든 내용을 알 수 있습니다.

- MySQL : TCP 기반으로 작동하는 MySQL 프로토콜은 기본적으로 암호화를 제공하지 않습니다. 따라서 네트워크를 통해 전송되는 모든 SQL 문은 평문으로 보이게 됩니다.

◆ Wireshark 에서 위의 프로토콜이 전송되는 것을 스니핑 해보고 결과를 캡쳐 합니다. 다음 예시는 DNS 를 대상으로 해본 것입니다(프라이버시를 위해 도메인과 IP 는 모자이크 합시다).



2. 보완 방법 조사

◆ 각각의 프로토콜에서 암호화를 제공하기 위한 방법을 조사합니다. 예를 들어, HTTP 라면 HTTPS 프로토콜을 적용하기 위한 방법을 조사합니다. 이를 위해서는 인증서를 발급받아야 하며 let's encrypt 의 certbot 을 사용하면 쉽고 편리하게 이를 적용할 수 있습니다.

3. 결과 문서 작성

◆ 패킷 스니핑 캡쳐와 함께 조사한 보완 방법을 문서로 작성해 제출합니다.

6. 산출물 제출

---

https://lab.ssafy.com/s10-study/self-project/ 의 “산출물 제출 가이드.docx” 참조