팀 프로젝트 소개

악성코드 소스코드 분석

Jaewoo Shim

Mar. 14. 2018

목차

- ❖ 팀프로젝트 소개
- ❖ 수행 내용
  - 소스코드 정적 분석
  - 컴파일 후 실행파일 동적 분석
- ❖ 제안서 작성방식
- ❖ 보고서 작성방식
- ❖ 채점 기준 및 총 배점

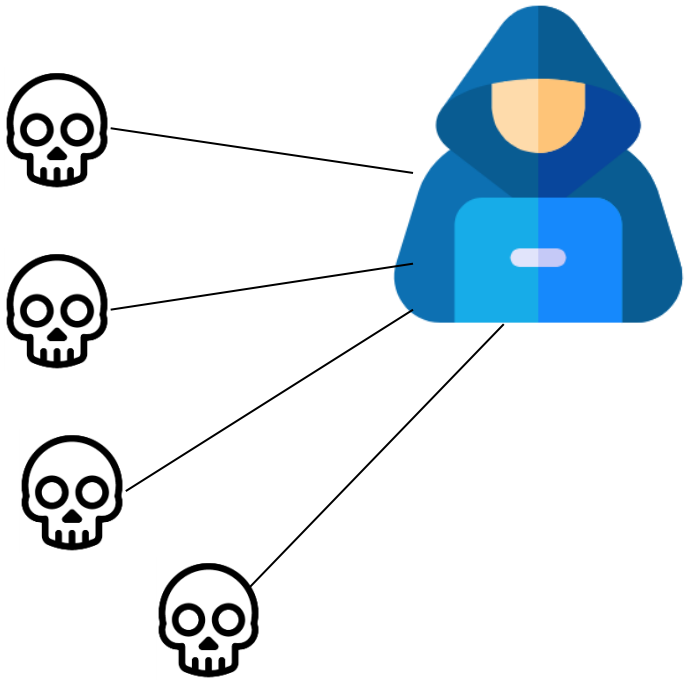
팀프로젝트 소개

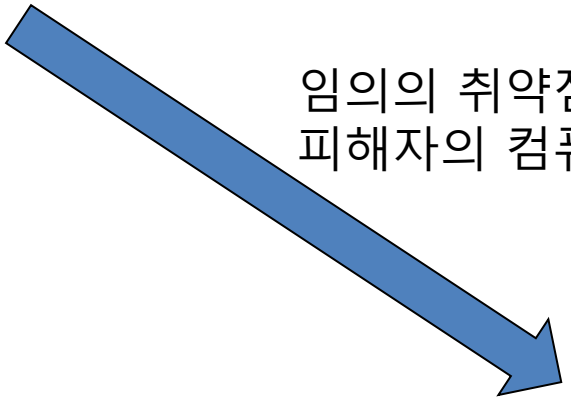
- **악성코드 분석 – lightAidra**
  - 소스코드가 없는 바이너리를 분석할 수 있는 역량을 익히기에는 시간상 현실적으로 어려움이 있음
  - 소스코드가 있는 악성코드를 대상으로 분석을 수행 – 2000줄 정도

- **소스코드 다운로드 가능 : https://github.com/eurialo/lightaidra**

- **3 ~ 4명의 팀으로 수행하는 것을 권장**
- **팀원 구성 후 조교에게 전달 : 3월 14일(수) 까지**
- **제안서 제출 : 4월 4일(수) 까지**
- **4주간 수행 후 결과 보고서 제출 : 5월 2일(수) 까지**

팀프로젝트 소개

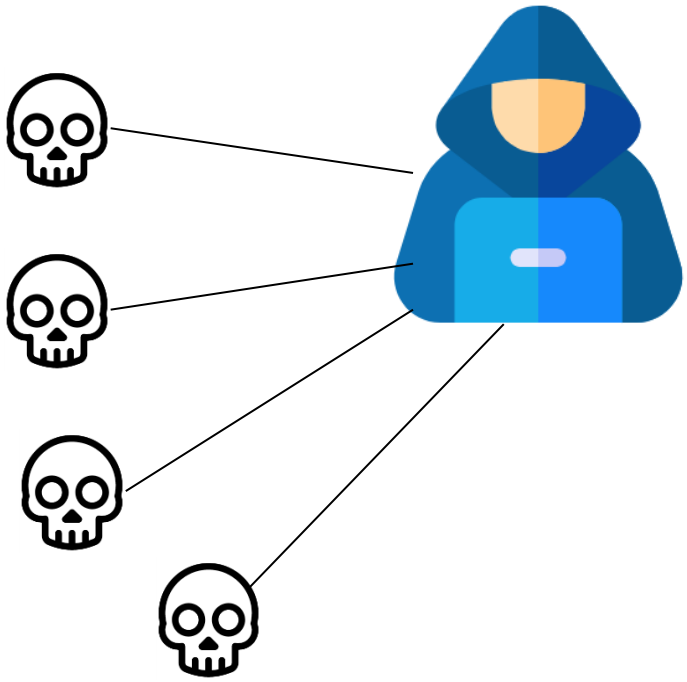




임의의 취약점 등을 이용해
피해자의 컴퓨터에 **악성코드**를 실행

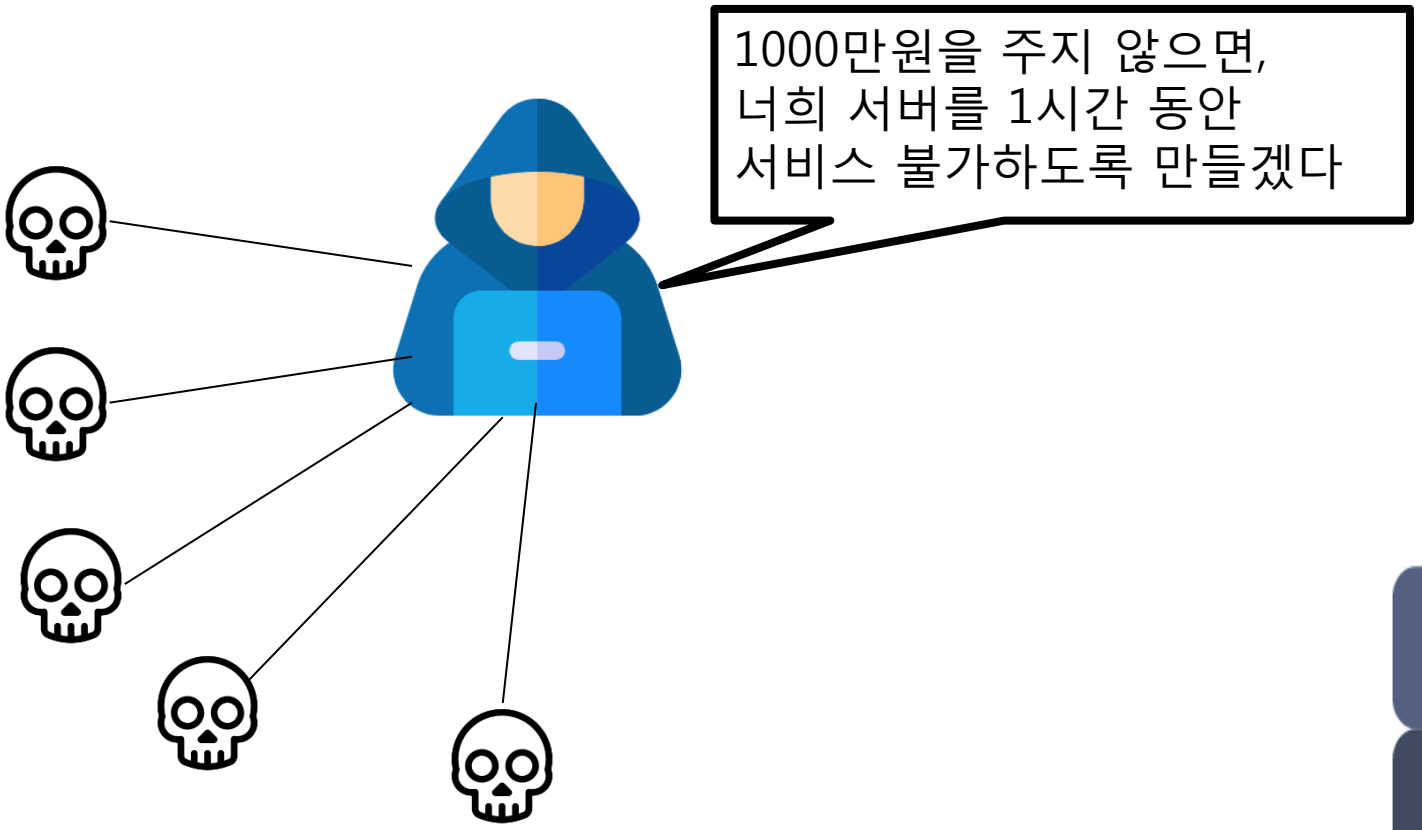
lightaidra

팀프로젝트 소개



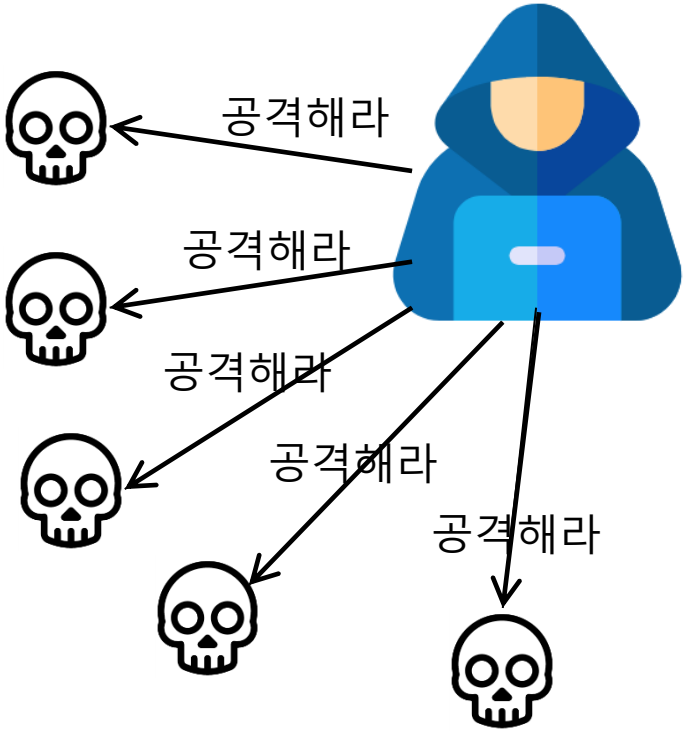
해커의 명령에 따라 동작
"너에대한 정보를 알려줘"
"너의 PC에 저장된 파일을 줘"
…

팀프로젝트 소개



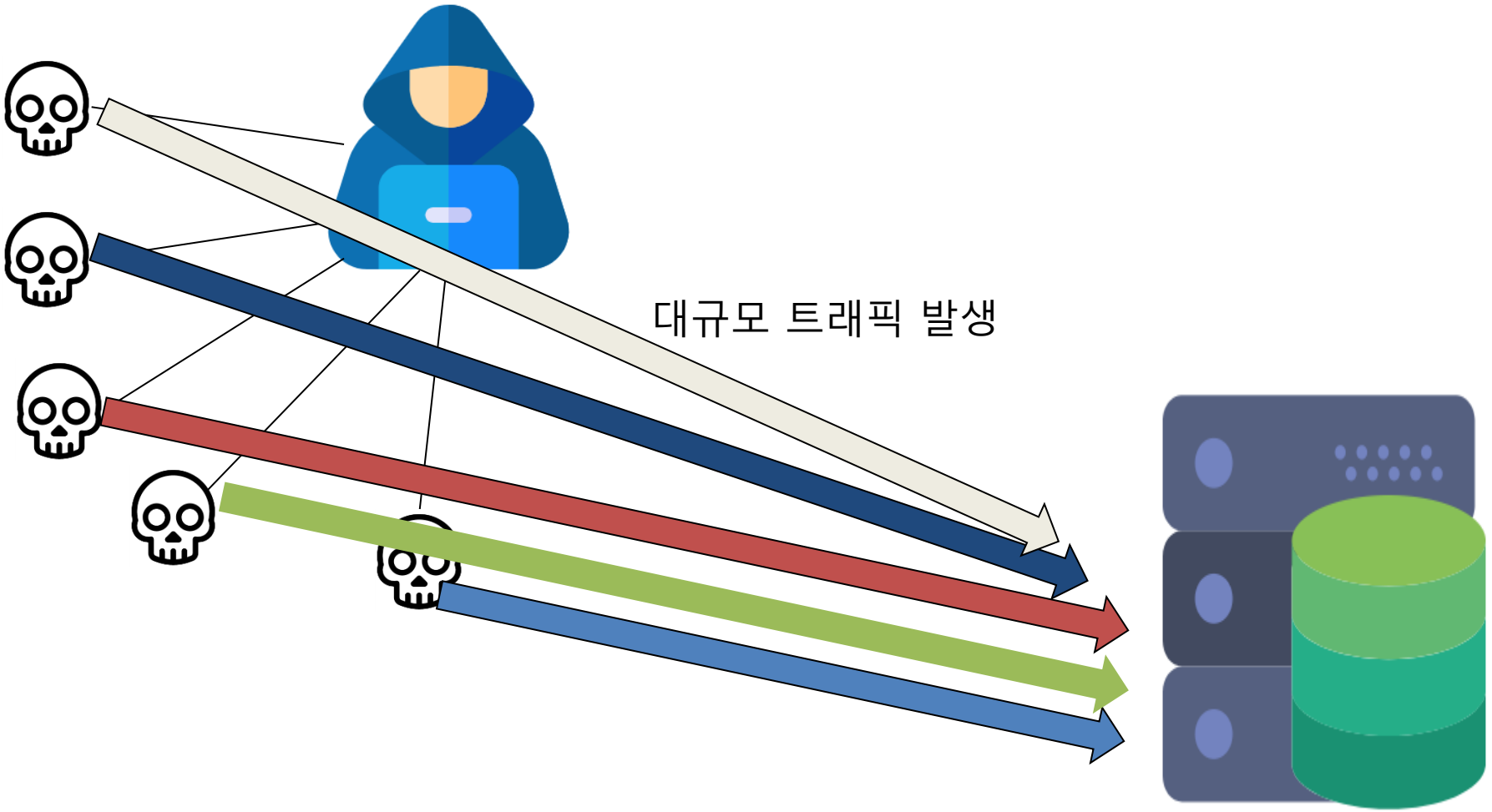
기업 또는 공공기관

팀프로젝트 소개



기업 또는 공공기관

팀프로젝트 소개



수행 내용 – 정적 분석

❖ **소스코드 정적 분석**

- ~/include 폴더에는 소스코드에서 사용하는 헤더파일(~~~.h)이 포함
- ~/source 폴더에는 소스코드(~~~.c) 파일이 포함
- README.md, Makefile, getbinaries.sh, docs 폴더 내 파일은 분석 X

❖ **소스코드를 분석해서 어떠한 기능을 수행하는 프로그램인지 파악**

TIP

소스코드에 함수마다 도움이 될 수 있는 주석이 있습니다.

수행 내용 – 정적 분석

❖ 예시

```
int main(int argc, char *argv[]) {
    if (argv[1] == 0 || argv[2] == 0) {
        printf("./lighthide [-encode|-decode] [string]₩n");
        return(1);
    }
    else if (!strncmp(argv[1], "-encode", 7)) {
        encode(argv[2]);
        decode(encoded);
        printf("encoded[%s]:₩n%s₩n", decoded, encoded);
    }
    else if (!strncmp(argv[1], "-decode", 7)) {
        decode(argv[2]);
        encode(decoded);
        printf("decoded[%s]:₩n%s₩n", argv[2], decoded);
    }
    return(0);
}
```

수행 내용 – 정적 분석

❖ 예시 – 나쁜예

```
int main(int argc, char *argv[]) {      // main함수 선언
    if (argv[1] == 0 || argv[2] == 0) {          // 만약 argv[1]이 0이거나 …
        printf("./lighthide [-encode|-decode] [string]₩n");          // 출력
        return(1);    // 종료
    } // 조건문 종료
    else if (!strncmp(argv[1], "-encode", 7)) { // 만약 argv[1]이 “-encode” 라면
        encode(argv[2]);       // encode()함수 실행
        decode(encoded);    // decode()함수 실행
        printf("encoded[%s]:₩n%s₩n", decoded, encoded);       // 결과값 출력
    } // 조건문 종료
    else if (!strncmp(argv[1], "-decode", 7)) { // 만약 argv[2]이 “-decode”라면
        decode(argv[2]);       // decode()함수 실행
        encode(decoded);    // encode()함수 실행
        printf("decoded[%s]:₩n%s₩n", argv[2], decoded); // 출력
    } //조건문 종료
    return(0); //종료
}
```

수행 내용 – 정적 분석

❖ 예시 – 좋은예

```
int main(int argc, char *argv[]) {
    if (argv[1] == 0 || argv[2] == 0) {
        printf("./lighthide [-encode|-decode] [string]₩n");
        return(1);
    }
    else if (!strncmp(argv[1], "-encode", 7)) {
        encode(argv[2]);
        decode(encoded);
        printf("encoded[%s]:₩n%s₩n", decoded, encoded);
    }
    else if (!strncmp(argv[1], "-decode", 7)) {
        decode(argv[2]);
        encode(decoded);
        printf("decoded[%s]:₩n%s₩n", argv[2], decoded);
    }
    return(0);
}
```

main 함수는 사용자에게 옵션을 입력받아 encode, 또는 decode 역할을 수행하여 결과값을 출력해주는 함수이다.

수행 내용 – 동적 분석

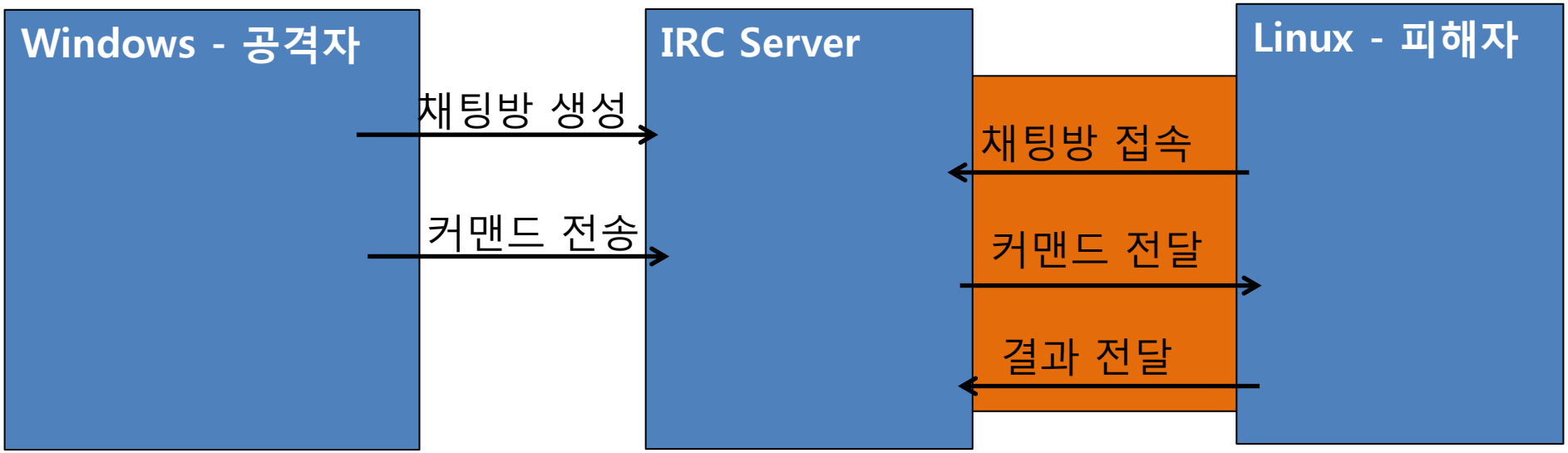
❖ **컴파일 후 악성코드를 직접 사용해보고, 동작을 분석**

- Lightaidra는 해커가 여러 기능을 수행할 수 있도록 만들어진 악성코드
- 어떤 기능을 수행하는 악성코드인지?
- 어떤 식으로 동작하는지?(커맨드, 환경구성 등)

수행 내용 – 동적 분석

❖ 환경구축 - 예시

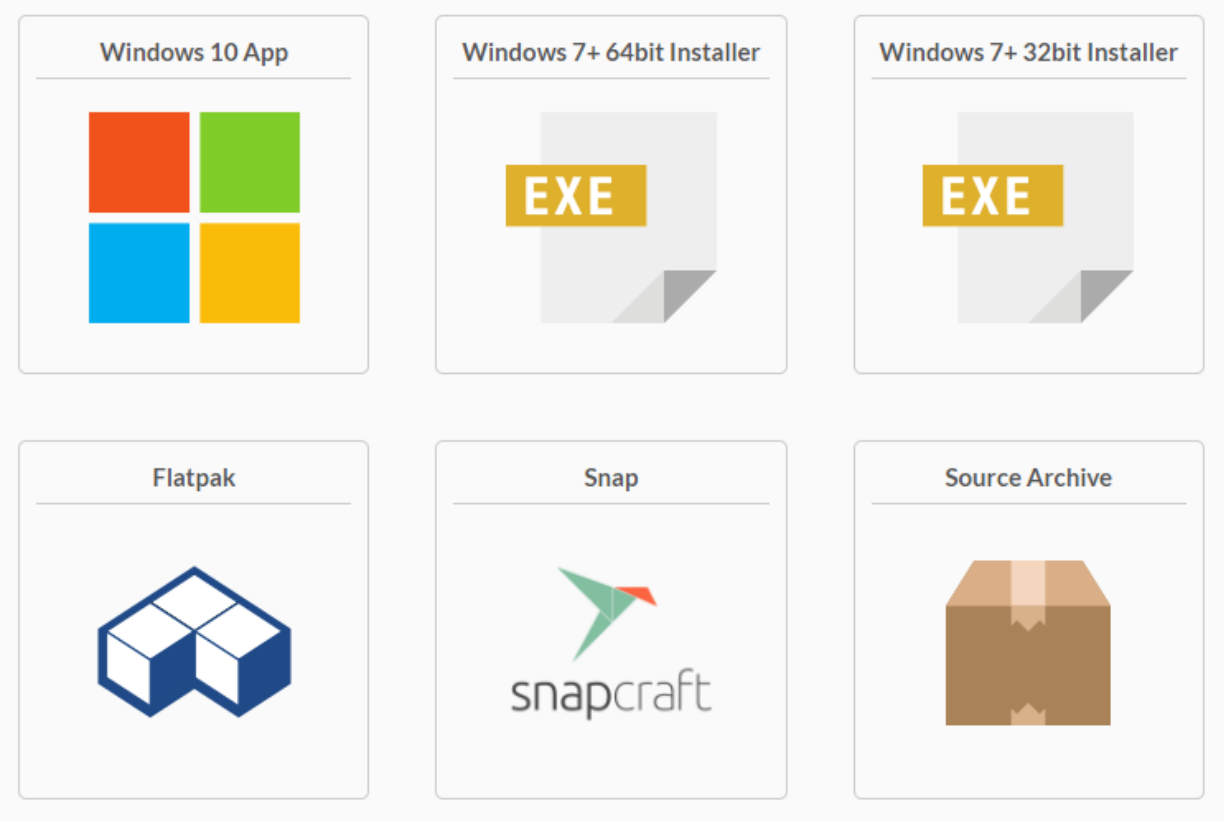
- Windows – 공격자
- Virtual box
  - Ubuntu linux(16.04) - 피해자



수행 내용 – 동적 분석

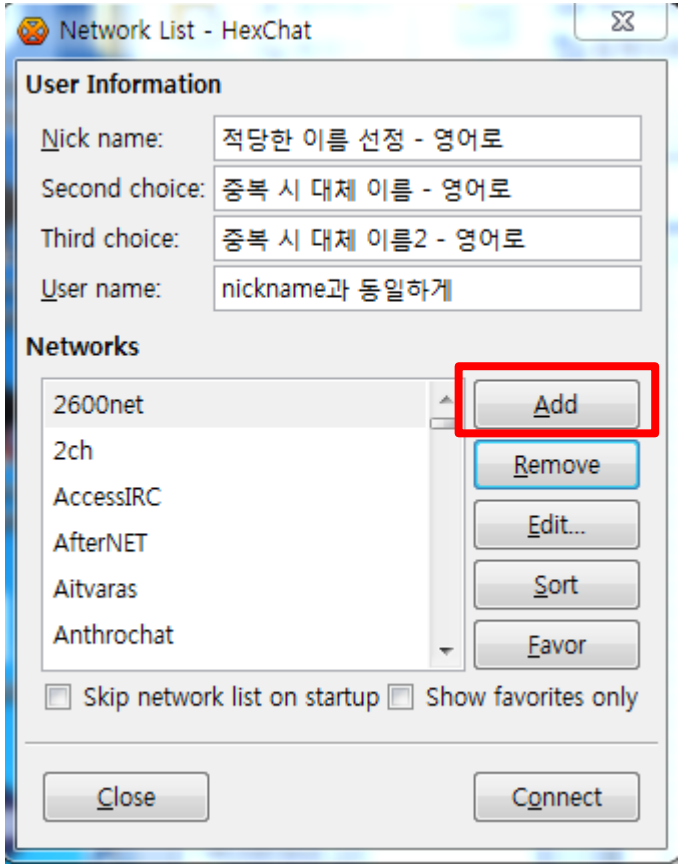
❖ 공격자 측 클라이언트 설치 : hexchat

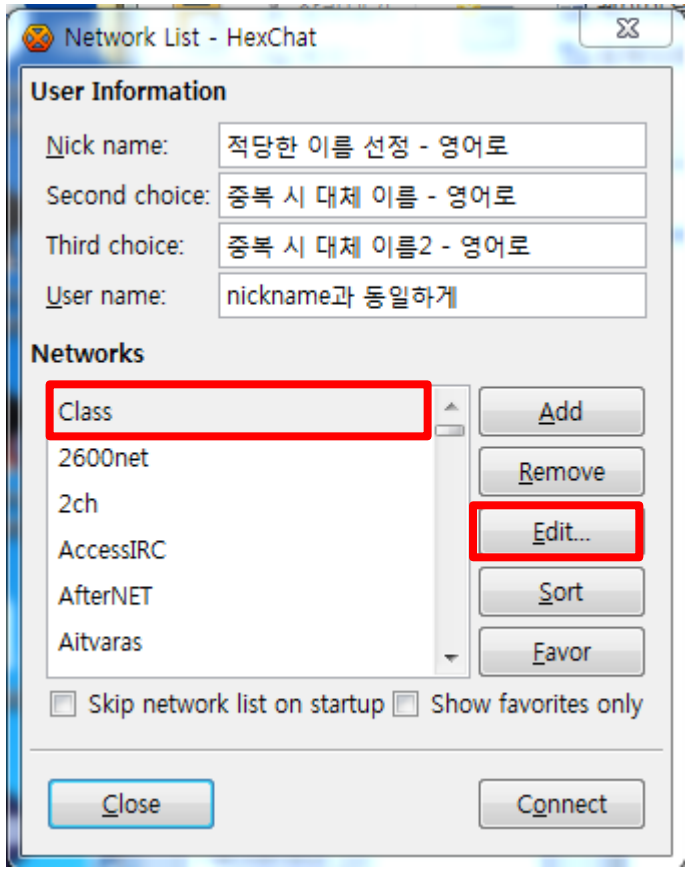
- https://hexchat.github.io/downloads.html



수행 내용 – 동적 분석

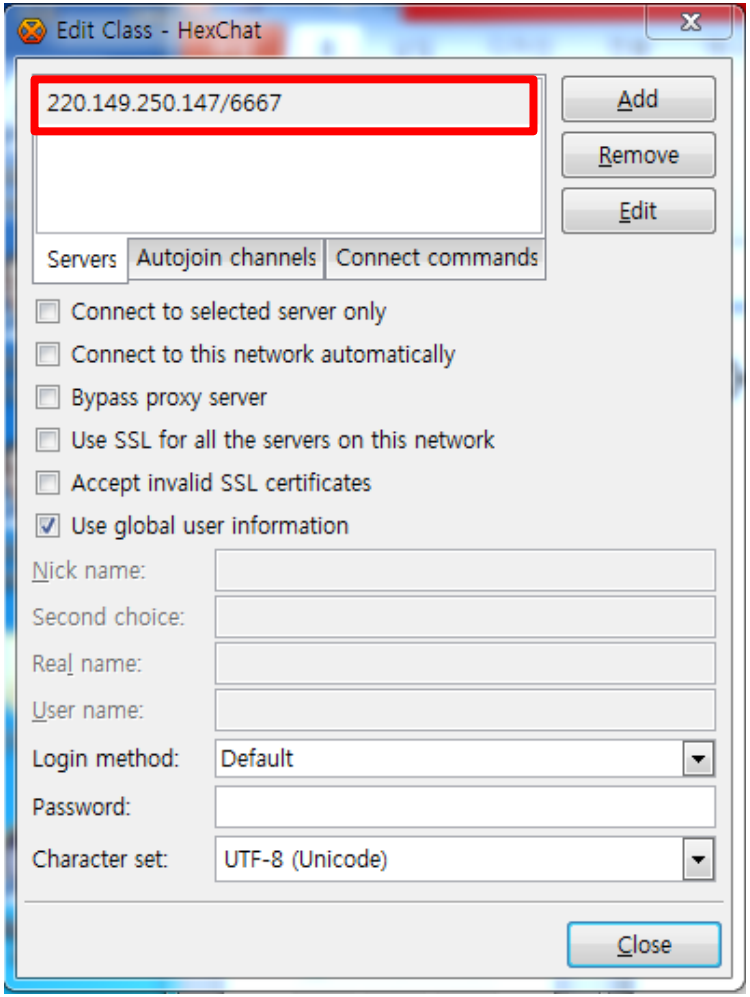
❖ 공격자 측 클라이언트 설치 : hexchat





수행 내용 – 동적 분석

❖ 공격자 측 클라이언트 설치 : hexchat



더블클릭하여 수정 후 엔터

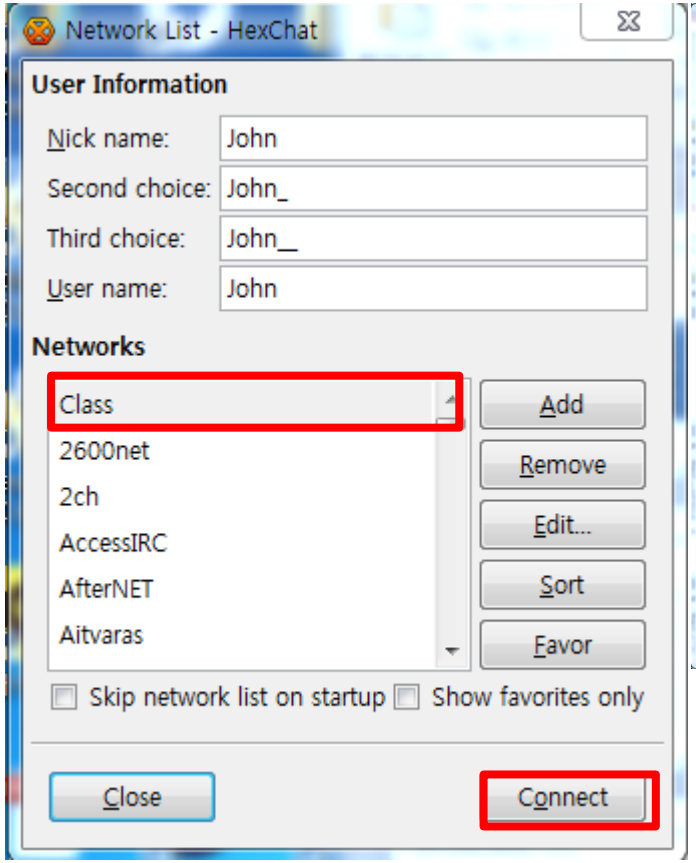
아이피주소 : 220.149.250.147/6667
연결이 안될 시 조교에게 연락

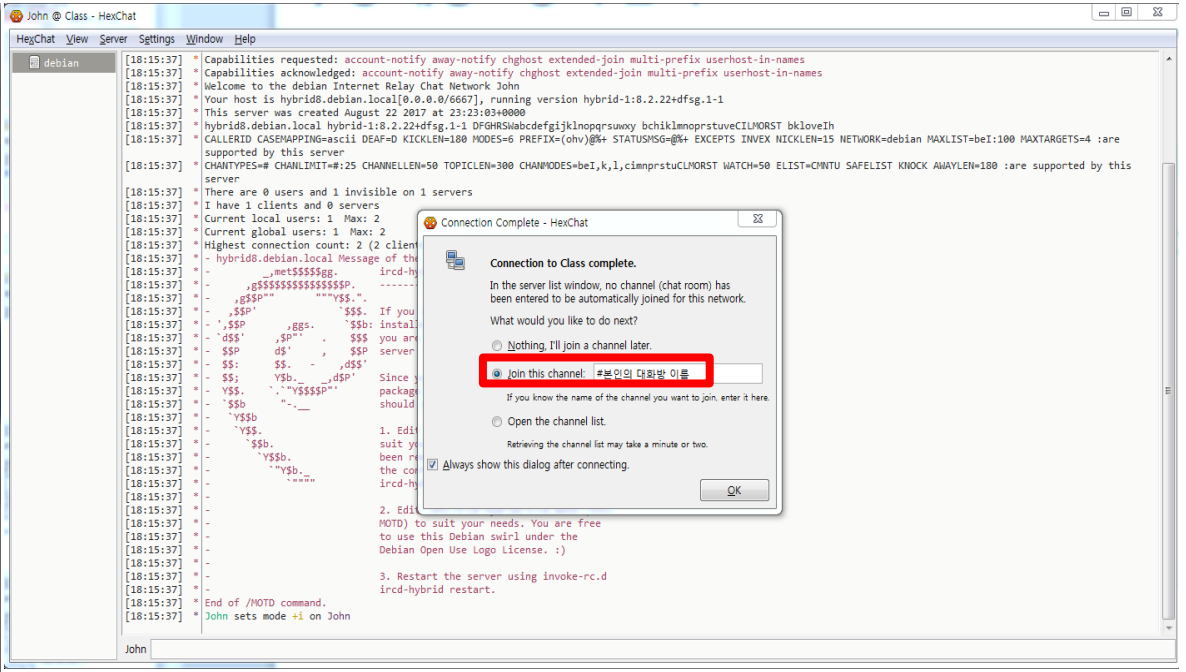
수정 완료 후 close

수행 내용 – 동적 분석

❖ **공격자 측 클라이언트 설치 : hexchat**

- 오른쪽 과같이 접속이 수행되는 것을 확인



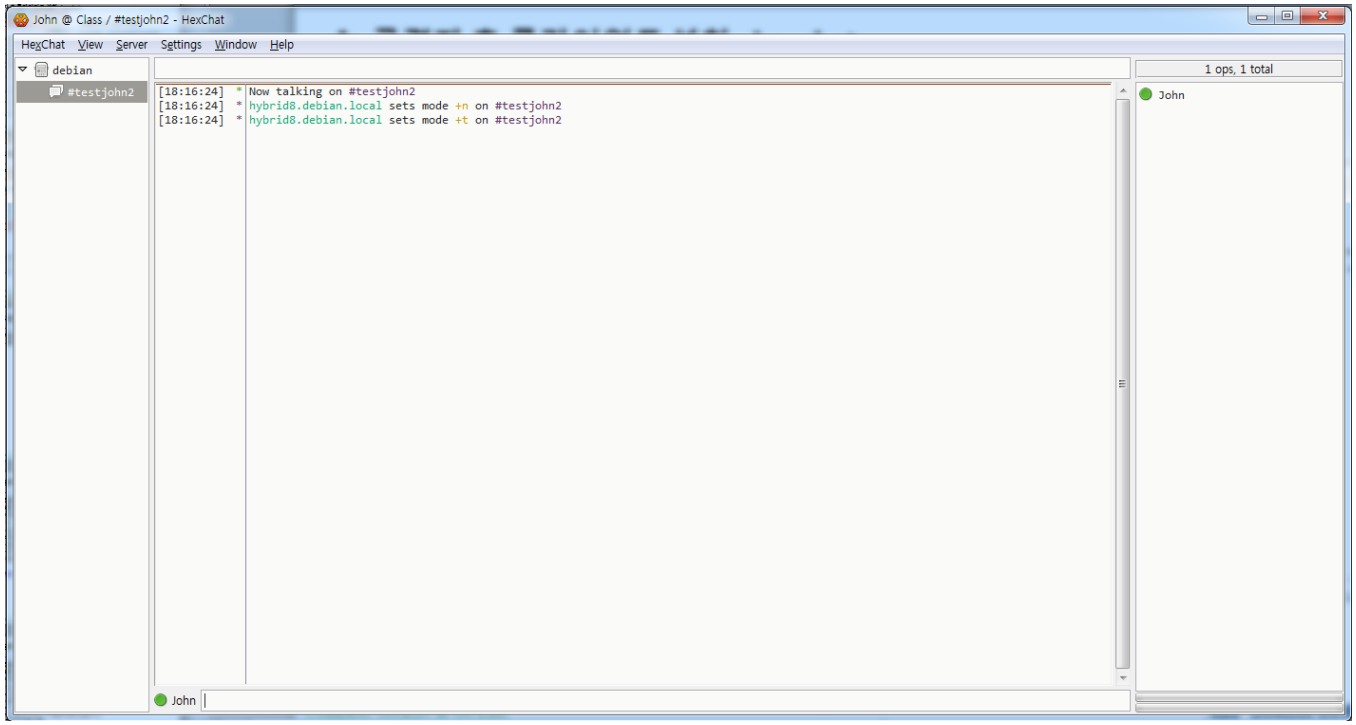


두번째 join the channel 에
본인이 생성할 대화방 이름 입력

수행 내용 – 동적 분석

❖ **공격자 측 클라이언트 설치 : hexchat**

- 아래와 같이 대화방이 생성되면, 이후 피해자가 접속될 대화방 생성 완료



- 대화창에 다음과 같이 입력 : **/topic #자기대화방이름 .join #자기대화방이름**

수행 내용 – 동적 분석

- **피해자는 lightaidra의 config.h파일을 분석해서 변경 후**
  - $ make x86_64
  - $ ./x86_64

- **대화방에 이상한 계정이 접속되었다면 악성코드 동작이 성공**
  - 이후 정적으로 분석을 수행시 파악된 명령어를 사용해보면서 분석 수행

팀프로젝트 소개 - 제안서

- **양식 및 분량 – 자유**
- **팀 별로 1부 제출**
- **제안서에 반드시 포함되어야 하는 내용**
  - 팀명
  - 팀원 이름/학번/이메일 - 팀장이 누구인지 표기
  - IRC 통신에 대한 개략적 설명
  - 분석을 수행할 환경
  - 분석에 사용할 도구
  - 일정(간트차트)

- **4월 4일 수요일까지**

팀프로젝트 소개 - 보고서

- **양식 및 분량 – 자유**
- **팀 별로 1부 제출**
- **보고서에 반드시 포함되어야 하는 내용**
  - 제안서에 기입했던 내용들
    - 환경과 도구는 분석 시 다르게 이용했을 경우 실제 사용한 내용을 작성
  - 소스코드 분석 내용(소스코드 주석달아 제출하지 마세요 - **감점**)
    - 각 *.c 파일의 역할 – 무슨 기능을 하는지?
    - 각 함수의 역할 – "~기능을 수행하는 함수이다."
  - 컴파일 후 실행 내용
    - 어떤 기능을 수행하는 악성코드인지?
    - 어떤 식으로 동작하는지?(자세하게)
      - 어떤 분석환경을 어떻게 갖추어 어떻게 실행시켜 확인했는지?
  - 결론
    - 이 악성코드는 어떤 환경에서 무슨 기능을 수행하는 악성코드다.
    - 내가 보안 담당자라면 어떻게 해당 악성코드를 탐지 및 차단할 것인지?

- **5월 2일(수) 까지**

채점 기준 및 총 배점

- **과제물 반영비율 내 팀프로젝트 비중 : 10%**
- **과제물 반영비율 내 BoF 과제 비중 : 5%**
- **과제물 반영비율 내 SQLi 과제 비중 : 3%**
- **과제물 반영비율 내 패킷과제 비중 : 2%**
- **----------**
- **SW보안개론 과제물 반영 비율 : 20%**

- **팀프로젝트 채점 기준(100점 만점이며 50점부터 시작)**
  - 이전 슬라이드 내 반드시 포함되어야 하는 내용 미기재 시 **감점** – 개당 -5
  - 주요 키워드가 들어가있지 않을 시 **감점**(10개 선정) – 개당 -5
    - 키워드는 채점이 완료된 이후에 공개 예정
      - 키워드 만으로 보고서를 작성하는 경우 방지
    - Hint) IRC, 봇, Synflood, config.h 포함 10가지

QnA

- 조교 이름 : 심재우
- 이메일 : tlawodn94@gmail.com
- 연구실 : 미디어센터 504호
- 서버 연결이 안될 시 : 010-3377-5148

Thank You !