1.1 ~ 1.5

아래의 그림은 HKIT의 정보시스템 자산별 중요도 현황을 나타낸 것으로 위험 평가 방법론으로 '복합접근법'을 활용하고 위험 처리 방법론으로는 '위험 감소'를 선택할 예정이다. 이 때 복합접근법과 위험 감소에 대해 서술하고 웹서버(데이터베이스)에 대해 정보 수집 및 취약점을 진단하시오.

자산 번호	자산 유형	모델	용도	담당자	사용자	기밀성	무결성	가용성	합계	중요 등급
HW-1	H/W	T440	웹서버	홍길동	ALL	3	3	3	9	매우 높음
NW-1	H/W	C-3890	L3스위치	박문수	김수로	1	1	3	5	낮음
SE-1	H/W	S-8890	방화벽	박문수	전산부	3	3	1	7	높음
DB-1	S/W	ORA111	데이터베이스	성춘향	운영자	3	3	3	9	매우 높음

<복합접근법>

복합 접근법은 정보 자산의 중요도 및 위험 수준에 따라 서로 다른 위험 평가 방법을 병행 적용하는 방법론이다. 이 방법은 자원의 효율적 배분과 보안 전략의 신속한 수립에 효과적이다.
중요하거나 고위험의 자산)에 대해서는 상세 위험 분석(Detail Risk Analysis)을 적용하여 자산 가치, 위협, 취약성을 정량적·정성적으로 평가한다. 중요도가 낮은 자산에는 기준선 접근법(Baseline Approach)을 적용하여 보안 기본 수준만을 충족하도록 한다.

<위험 감소>

위험 감소란, 식별된 위험에 대해 보안 대책을 적용하여 위험 수준을 낮추는 방법이다.
대책 적용 전과 후의 연간 기대 손실(ALE)을 비교하여 보호 대책의 경제성을 평가한다. 대책의 효과가 ALE 감소 효과보다 크다면, 해당 대책을 도입한다.

<정보 수집 및 취약점 진단>

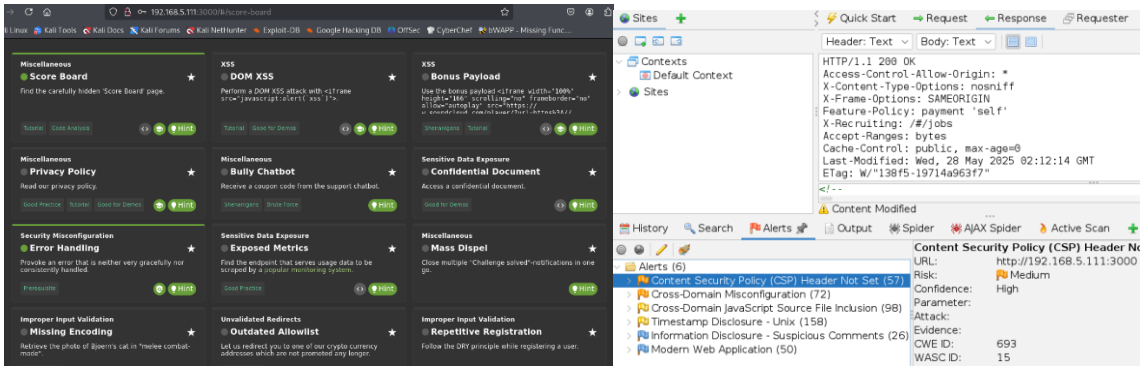
[JuiceShop 정보 수집]

```
┌──(root㉿kali-kim)-[~]
└─# nmap 192.168.5.111
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-27 22:22 EDT
Nmap scan report for 192.168.5.111
Host is up (0.00014s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
443/tcp  open  https
3000/tcp open  ppp
MAC Address: 08:00:27:D1:79:2C (Oracle VirtualBox virtual NIC)
```

```
┌──(root㉿kali-kim)-[~]
└─# dirb http://192.168.5.111:3000 -w /usr/share/wordlists/dirb/big.txt

_____________

DIRB v2.22
By The Dark Raver
_____________

START_TIME: Tue May 27 22:23:24 2025
URL_BASE: http://192.168.5.111:3000/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Not Stopping on warning messages

_____________


GENERATED WORDS: 4612

---- Scanning URL: http://192.168.5.111:3000/ ----

+ http://192.168.5.111:3000/assets (CODE:301|SIZE:156)
+ http://192.168.5.111:3000/ftp (CODE:200|SIZE:11071)
+ http://192.168.5.111:3000/profile (CODE:500|SIZE:1049)
+ http://192.168.5.111:3000/promotion (CODE:200|SIZE:6586)
+ http://192.168.5.111:3000/redirect (CODE:500|SIZE:3119)
+ http://192.168.5.111:3000/robots.txt (CODE:200|SIZE:28)
+ http://192.168.5.111:3000/snippets (CODE:200|SIZE:792)
+ http://192.168.5.111:3000/video (CODE:200|SIZE:10075518)
+ http://192.168.5.111:3000/Video (CODE:200|SIZE:10075518)
```

nmap과 dirb 명령어를 이용하여 포트 스캐닝 및 폴더/파일 확인

[JuiceShop 취약점 진단]



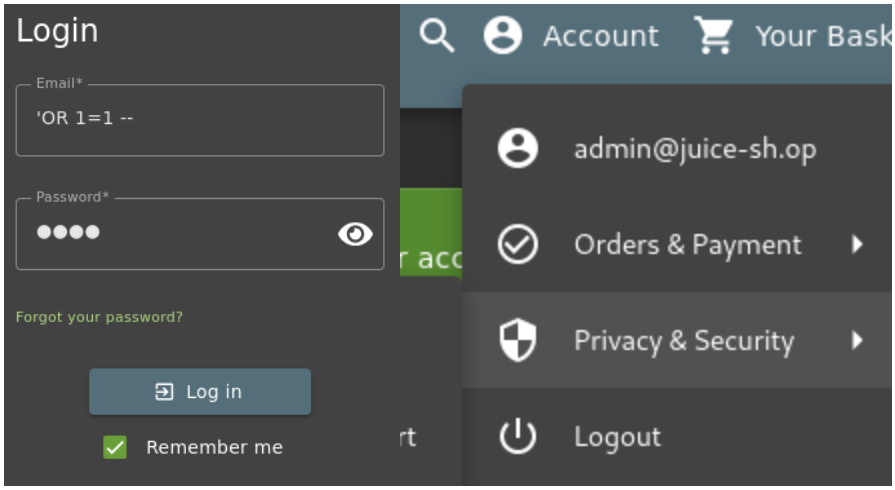
정보 수집을 통해 찾아낸 JuiceShop의 숨겨진 페이지에서 JuiceShop의 취약점 확인 / zaproxy를 이용한 취약점 검사

2.1 ~ 2.3

앞서 확인한 취약점들에 대해 수동 및 자동 또는 면담을 활용하여 실제 공격이 가능한지 진단을 수행하여 점검을 하고 면담을 통해 진단 시 효율적인 면담을 위한 사항 2가지 이상을 서술하시오.

<취약점 진단>

웹서버(JuiceShop)에서 SQL Injection의 취약점이 있는지 확인
진단 방식: 수동
진단 절차: 로그인 페이지에서 SQL Injection 명령어가 실행되는지 확인
공격 가능성: 일반 사용자가 간단한 명령어를 통해 admin으로 접근할 수 있음
결과:



로그인 페이지에 SQL Injection 구문 입력 -> 관리자 계정으로 로그인 확인

<효율적인 면담을 위한 사항>

사전 질문지 및 체크리스트 준비 – 자산별 핵심 점검 항목에 대해 사전 질문 구성

면담 대상자의 역할과 책임 명확화 – 실제 시스템 운영 또는 보안을 담당하는 사람과 면담해야 정확한 정보 확보 가능

운영 담당자/보안 책임자 동시 면담 – 실무자는 기술적인 설정 사항을, 보안 책임자는 정책 수립 및 예외사항 설명

3.1 ~ 3.4

확인한 취약점 2가지에 대한 보호 대책(대응 방안)에 대해 서술하고 추가적으로 IDS(Snort/Suricata) 시스템에서 SQL Injection을 탐지하기 위한 Rule을 작성하시오.

<보호 대책>

웹서버(JuiceShop)에서 SQL Injection의 취약점 해결방안
- 모든 사용자 입력에 대해 문자열 길이, 타입, 형식 등을 검증
- 쿼리를 파라미터화하여 SQL 명령어로 인식되지 않도록 함
- ORM(Object Relational Mapping) 프레임워크 같은 강한 구조를 사용함
- WAP 사용

<Rule 작성>

[Suricata 룰]

1) alert tcp any any -> any 80 (msg:"SQL Injection attempt (1=1)"; flow:to_server,established; content:"1=1"; nocase; http_uri; classtype:web-application-attack; sid:1000001; rev:1;)

2) alert tcp any any -> any 80 (msg:"SQL Injection attempt (UNION SELECT)"; flow:to_server,established; content:"UNION SELECT"; nocase; http_uri; classtype:web-application-attack; sid:1000002; rev:1;)

- flow:to_server,established: 서버로 가는 HTTP 트래픽에서 1=1같은 흔한 SQL Injection 페이로드를 URL 내에서 탐지함

[Snort 룰]

alert tcp any any -> any 80 (msg:"SQL Injection Attempt Detected";
flow:to_server,established;
content:"select"; nocase;
content:"from"; nocase;
content:"where"; nocase;
http_uri;
sid:1000001;
rev:1;)

- TCP 80번 포트를 통해 접근 시 SELECT, FROM, WHERE 키워드가 동시에 존재 시 경고 발생