**정보 수집**

정보 수집 단계는 모의해킹의 대상을 정하고 나면 가장 처음 수행하게 되는 단계

수집할 정보:

1. 웹 애플리케이션을 서비스하기 위한 호스트 환경 (운영체제, 웹 서버, 웹 프레임워크)

-> 알려진 취약점을 쉽게 찾아 공격

2. 웹 애플리케이션 자체에 대한 정보

-> 웹 애플리케이션 매핑을 통해 정보 수집 후 구조 파악하여 공격 지점 찾기

**1. 배너를 통한 정보 수집**

- 웹 서버와 관련된 정보를 가장 간단하게 확인할 수 있는 방법 -> 응답 메시지의 서버 헤더

- 서버 응답을 통한 정보 수집 방법 -> 배너 그래빙(banner grabbing)

- 서버 헤더를 통해 아파치 웹 서버 버전 정보, 운영체제, PHP 버전 정보, OpenSSL 등 정보 수집

- 서버 헤더 이외 X-Powered-By, X-ASPNET-VERSION과 같은 헤더들이 호스트 환경의 정보 노출

- 이와 같은 응답 헤더의 정보는 웹 브라우저의 개발자 도구에서 확인 가능

- Burp Suite의 Proxy 히스토리 기능으로도 확인 가능

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

**2. 기본 설치 파일을 통한 시스템 정보 수집**

- 웹 애플리케이션 운영 위한 웹 서버와 웹 프레임워크, 기타 구성 요소 설치시, 기본 설치 파일로 인해 호스트 환경에 대한 정보 노출

(ex. PHP의 경우, phpinfo.php라는 기본 설치 파일을 통해 PHP 환경에 대한 정보, 환경 변수, 그 밖의 다양한 호스트 관련 정보가 노출될 수 있음.)

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**3. 웹 취약점 스캐닝**

- 자동화된 프로그램을 이용하여 웹사이트의 여러 가지 정보 수집 및 취약점 파악

- 많은 트래픽 유발로 인해 모의 해킹할 때, 담당자의 허가 후에 진행

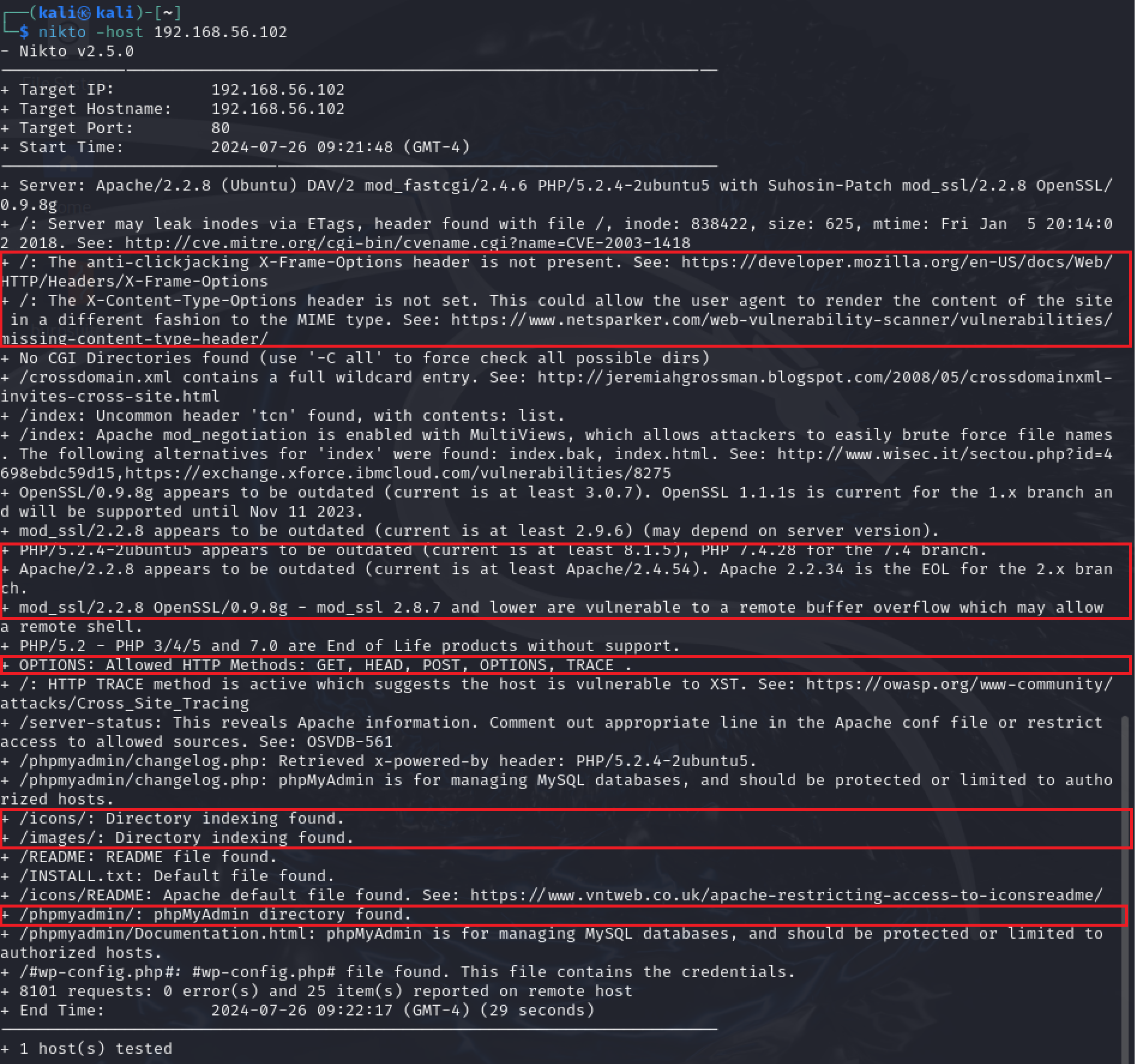
- 칼리 리눅스에 기본으로 설치된 웹 스캐닝 프로그램인 “nikto”로 정보 쉽게 수집 가능

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

<터미널에서 nikto의 옵션을 확인한 모습>

- “-host” 옵션 뒤에 스캐닝할 IP 주소만 입력해주면 스캐닝을 실행하여 많은 정보 수집



<실습용 가상 머신의 IP 주소를 입력하여 실행한 화면>

- 서버 헤더 및 PHP, 아파치, mod\_ssl 등의 버전이 오래되었다.

- 보안 관련 헤더가 설정되어 있지 않다.

- 허용된 메소드 목록 정보

- /icons/, /images/ 경로에서 디렉토리 인덱싱이 발견되었다.

- phpMyAdmin 디렉토리가 발견되었다.

- 이와 같은 정보는 모의해킹을 수행하는 과정에서 공격에 중요한 단서가 될 수 있다.

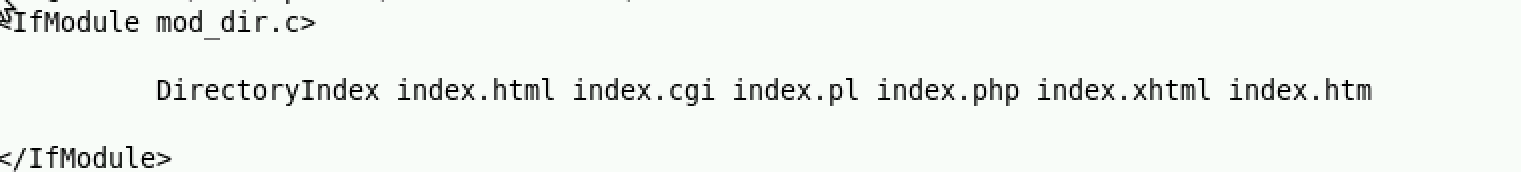
- 오래된 버전을 사용하는 소프트웨어에는 이미 알려진 취약점들이 존재할 수 있어, 이 취약점을 활용하여 쉽게 공격 성공할 수 있다.

**4. 디렉토리 인덱싱**

- 웹 서버의 잘못된 설정으로 웹 서버 디렉토리의 파일들이 노출되는 취약점

- 디렉토리 리스팅 취약점이라고도 부르며, 과거 국가정보원 8대 홈페이지 취약점으로도 선정되었다.

- 디렉토리 목록 출력은 원래 웹 서버가 지원하는 기능



<디렉토리의 기본 출력파일을 설정하는 mod\_dir 아파치 모듈>

- 사용자가 웹 브라우저의 주소로 디렉토리 경로(ex. /images/) 요청시, 해당 디렉토리에 위 사진과 같은 파일이 존재하면 해당 파일 출력

- 지정된 파일들이 모두 디렉토리에 존재하지 않으면 해당 디렉토리의 모든 파일 출력 -> 문제

- 아파치와 같은 일부 웹 서버의 경우, 소프트웨어 설치시 기본으로 디렉토리 인덱싱 설정

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

- 이전 스텝에서 “nikto”로 /icons/에 디렉토리 인덱싱 발견을 토대로 웹 브라우저에 검색하면 디렉토리 파일 목록 출력

- web.confg.bak, wp-config.bak과 같은 백업 파일을 통해서도 중요한 정보 노출 가능

**5. 수동 매핑**

- 직접 웹 애플리케이션에 접속하여 각 메뉴 확인

- 웹 애플리케이션의 새로운 메뉴를 누를 때마다 사이트 맵 트리에 새로운 정보 업데이트

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

Ex. Burp Suite 실행 후 인터셉트 기능 off, 웹 브라우저 Proxy 기능 활성화하고 bWapp이나, DVWA 접속 -> 로그인 시도 -> 사이트 맵에 URL와 파라미터들이 트리형태로 표시

**6. Crawling**

- 웹 애플리케이션 매핑 과정 자동 수행

- 웹 페이지의 링크 분석 -> 새로운 웹 페이지 찾아내는 과정

- 크롤링 수행 프로그램: 크롤러

- 수행 과정

1) 크롤러가 처음 지정된 URL 요청 (URL은 크롤러가 처음 시작할 때, 사용자가 지정, 웹 로봇 등과 같은 크롤러는 robots.txt 파일 읽고 크롤링 시작)

2) 처음 요청에 의해 전송받은 응답 메시지를 분석하고, 응답에 포함된 링크 각각 추가 요청

3) 링크 요청에 의해 전송받은 응답 메시지 다시 분석, 링크가 다시 포함되어 있으면 또다시 해당 링크 추가 요청

4) 더 이상 링크 찾기 불가 또는 404나 500 등과 같은 에러메시지 응답시까지 과정 반복

**7. DirBuster**

- URL 목록 파일 사용하여 각 URL을 자동으로 입력해보는 방식으로 웹 애플리케이션 구조 파악 (브루트 포스)

- 숨겨진 페이지도 찾을 수 있음

- 주로 사용자의 패스워드를 찾을 때 사용하는 기법 (URL 찾을 때도 사용 가능)

스크린샷, 소프트웨어, 텍스트, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

- DirBuster는 칼리 리눅스에 기본적으로 설치되어 있다.

- Target URL을 입력한 후, Browse 버튼을 통해 URL 목록 파일 선택, Start 누르면 실행

- DirBuster로 찾은 URL로 직접 접속하며 수동 매핑과 크롤링 추가로 연계 가능

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**8. robots.txt**

텍스트, 소프트웨어, 멀티미디어 소프트웨어, 그래픽 소프트웨어이(가) 표시된 사진

자동 생성된 설명

- 검색 엔진은 대부분 웹 로봇을 이용하여 웹 사이트 정보 수집 (웹 로봇 = 크롤러 일종)

- robots.txt 파일을 웹 사이트의 가장 상위 디렉토리에 위치시켜, 웹 로봇에게 해당 웹 사이트의 정보 수집 허용 또는 불허할 수 있음

- “User-agent” 키워드를 이용하여 지정된 “User-agent” 요청 헤더를 전송하는 특정 로봇에게 명령 내림 (“\*” 와일드카드로 모든 로봇에게 명령 내리기 가능)

- “Disallow” 키워드를 이용하여 설정된 경로에 대한 수집 차단 명령 내림

- “Allow” 키워드를 이용하여 설정된 경로에 대한 수집 허용 명령 내림

- robots.txt를 만들어도 그 내용을 따를지는 웹 로봇에게 있음 (악의적으로 제작된 웹 로봇은 robots.txt를 무시하고 Disallow 키워드로 차단한 내용을 정보 수집에 활용)

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

<”Disallow” 키워드로 된 admin에 접속한 모습>

**9. 정보 수집 대응 방법**

**1) 불필요한 정보 노출 삭제**

- 반드시 필요한 경우가 아니라면, 서버 헤더에 제공되는 정보 삭제 후 전송

- 웹 서버 및 웹 프레임워크 중 자체적인 헤더 삭제 기능 제공하는 경우 있음

- 웹 방화벽 등 보안 장비를 이용하여 일괄적으로 헤더 삭제 가능

- 불필요한 기본 설치 파일, 백업 파일, 테스트 파일은 모두 실제 서비스가 운용되는 환경에는 존재하지 않도록 해야 함

- 필요에 의해 사용해야 하는 설정 파일도 웹을 통해 외부에 공개되지 않도록 주의

**2) 스캐너/크롤러 차단**

- 스캐너 또는 크롤러와 같은 자동화된 프로그램으로부터 완전하게 방어하는 것은 불가능

- 적절한 로깅와 모니터링을 통해 어느 정도 공격 시도 탐지 및 차단 가능

- 탐지 방법 예시

[ cat /var/log/apache2/access\* | awk -F\” ‘{print $6}’ | sort | uniq -c | sort -n ---생략---]

위 명령어와 같이 실행하면 아파치 웹 서버 접근 로그에 기록된 요청 헤더 정보와 접속 횟수를 확인할 수 있음

하지만, 정상적인 접속 내용도 포함되어 있어 신중히 구분할 필요 있음

- 공격이 의심되는 경우 IP 주소를 확인하여 차단 또는 “User-agent” 헤더의 값을 이용하여 차단

- IP 주소는 해당 접근 로그로 확인

- 침임 탐지 시스템(IDS)/침입 방지 시스템(IPS) 또는 웹 방화벽 등의 보안 장비를 사용해서 자동화된 공격 탐지 및 차단

**3) 디렉토리 인덱싱 설정 제거**

- 웹 서버의 설정을 변경하여 대응

- 아차피의 경우, 아파치 설정 파일의 “Indexes” 옵션을 제거하여 디렉토리 인덱싱으로부터 보호

- 칼리 리눅스의 아파치 웹 서버는 “/etc/apache2/apache2.conf” 파일의 내용 확인

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

자동 생성된 설명

- 위 사진의 Indexes 옵션을 제거하면 디렉토리 인덱싱 설정 해제

- IIS 서버의 경우, “디렉토리 검색” 옵션을 비활성화하면 디렉토리 인덱싱 설정 해제