1. 감사 로그(Audit Log) (3)

- 감사 로그는 내부 보안 위협을 사전에 파악, 시스템 운영 및 사용을 추적하기 위해 OS(운영 체제), 애플리케이션 또는 장치에 대한 모든 변경 사항을 시간순으로 자세히 기록한 것이다.

- 감사 로그를 사용하면 조직 관리자가 조직 구성원이 수행한 작업을 신속하게 검토할 수 있다. 여기에는 작업을 수행한 사람, 작업 유형, 작업이 수행된 시기 등이 기록된다.

- Net명령(System32 폴더에 명령이 있음)이 수행이 안되면 PATH 설정을 해야한다.
- PATH에 'C:\Windows\System32'를 추가
  (https://ssimplay.tistory.com/223)

2. 관리자 권한(6)

관리자 권한으로 실행하기를 사용하면 다음과 같은 작업을 수행할 수 있다.

- 시스템 설정 변경: 관리자 권한으로 실행할 경우, 사용자는 시스템 설정을 변경할 수 있다. 예를 들어, 디스크 파티션을 만들거나 삭제하거나, 사용자 계정을 관리하거나, 시스템 시간을 변경할 수 있다.

- 프로그램 및 파일 수정: 일부 프로그램이나 파일은 일반 사용자 계정으로는 수정할 수 없다. 그러나 관리자 권한으로 실행하면 이러한 프로그램이나 파일에 액세스하여 수정할 수 있다.

- 시스템 서비스 관리: 관리자 권한을 가진 사용자는 시스템 서비스를 시작, 중지 또는 재시작할 수 있다. 이는 시스템의 안정성과 성능을 관리하고 문제를 해결하는 데 도움이 된다.

- 관리자 권한이 필요한 프로그램 실행: 일부 프로그램은 관리자 권한이 필요한 작업을 수행해야 한다. 이러한 프로그램을 일반 사용자 계정으로 실행하면 작동하지 않을 수 있지만, 관리자 권한으로 실행하면 정상적으로 작동할 수 있다.

- 일시적: 마우스 오른쪽 버튼 → 관리자 권한으로 실행
- 영구적: 실행 프로그램을 오른쪽 마우스 클릭하여 속성을 선택 → 고급 → 관리자 권한으로 실행

3. 유닉스 계정 관리(9)

- "wheel:x:10:wishfree"의 구성 요소
  ① wheel: 그룹 이름. 일반적으로 시스템 관리 권한을 가진 사용자 그룹을 의미
  ② x: 암호 필드. 이 필드는 보통 'x'로 표시되며, 실제 암호는 /etc/shadow 파일에 저장
  ③ 10: 그룹 ID (GID). 이 숫자는 시스템에서 해당 그룹을 식별하는 데 사용
  ④ wishfree: 이 그룹에 속한 사용자 목록. 여러 사용자가 있을 경우, 쉼표로 구분하여 나열, wheel:x:10:root,admin,testuser와 같은 형식으로, wheel 그룹에 root, admin, testuser가 포함되어 있다.

- bin:x:1의 bin은 사용자 그룹의 이름을 나타낸다. 시스템에서 실행되는 바이너리 파일과 관련된 사용자 계정들을 포함하는 그룹이다. 구체적으로, bin 그룹은 다음과 같은 역할을 한다.
  ① 파일 권한 관리: bin 그룹에 속한 사용자들은 시스템의 중요한 바이너리 파일에 대한 접근 권한을 가질 수 있다. 이는 시스템의 기본적인 명령어와 프로그램들이 포함된 디렉토리(예: /bin, /usr/bin)에 대한 접근을 포함한다.
  ② 사용자 관리: bin 그룹에 속한 사용자들은 일반적으로 시스템 관리나 특정 작업을 수행하는 데 필요한 권한을 부여받는다.
  ③ 보안: 특정 파일이나 디렉토리에 대한 접근을 제한하여 시스템의 보안을 강화하는 데 기여한다.

- adm:x:4는 리눅스 시스템에서 adm이라는 이름의 그룹을 나타내며, 이 그룹은 시스템 관리와 관련된 권한을 가진 사용자들이 속해 있다. 이 정보는 시스템의 보안과 관리에 중요한 역할을 한다.

- Sync: 리눅스에서 sync 명령어는 파일 시스템의 데이터를 디스크에 동기화하는 역할이다. 구체적으로, 이 명령어는 메모리에 있는 모든 수정된 파일 시스템 데이터를 디스크에 기록하여, 데이터 손실을 방지하고 파일 시스템의 일관성을 유지하는 역할을 수행한다.

- root: Unix/Linux 시스템에서 'root'는 시스템의 관리자 계정을 의미한다. 이 계정은 모든 파일과 명령에 대한 완전한 접근 권한을 가지고 있으며, 시스템 설정을 변경하거나 다른 사용자 계정을 관리할 수 있다.
- shutdown: 'shutdown' 명령은 시스템을 안전하게 종료하는 데 사용된다. 이 명령을 사용하면 현재 실행 중인 프로세스가 종료되고, 파일 시스템이 정리된 후 시스템이 꺼진다. 다양한 옵션을 통해 즉시 종료하거나 일정 시간 후에 종료할 수 있다.
- halt: 'halt' 명령은 시스템을 즉시 중지시키는 명령이다. 이 명령은 시스템을 안전하게 종료하지 않고 전원을 끄는 것과 유사한 방식으로 작동한다.
- operator: 'operator'는 일반적으로 시스템 관리자나 운영자를 의미한다. 이 용어는 시스템의 유지 관리, 모니터링 및 문제 해결을 담당하는 사람을 지칭한다. 특정 시스템에서는 'operator'라는 사용자 그룹이 있을 수 있으며, 이 그룹은 특정 권한을 가질 수 있다.

4. 응용 프로그램 해킹(10)

취약한 응용 프로그램을 통해 공격자가 운영 체제에 접근하여 민감한 정보를 습득하고 이를 이용해 운영 체제를 공격하는 사례는 여러 가지가 있다.

- **SQL 인젝션**: 공격자가 취약한 웹 애플리케이션의 데이터베이스 쿼리에 악의적인 SQL 코드를 삽입하여, 데이터베이스에 저장된 민감한 정보를 추출할 수 있다. 이 정보는 이후 운영체제의 사용자 계정이나 비밀번호를 알아내는 데 사용될 수 있다.
- **버퍼 오버플로우**: 취약한 응용 프로그램에서 입력값의 길이를 검증하지 않아 발생하는 버퍼 오버플로우 공격을 통해, 공격자는 악성 코드를 실행하거나 운영체제의 권한을 상승시킬 수 있다. 이를 통해 시스템에 대한 완전한 제어를 획득할 수 있다.
- **크로스 사이트 스크립팅 (XSS)**: 공격자가 취약한 웹 애플리케이션에 악성 스크립트를 삽입하여, 사용자의 브라우저에서 실행되도록 유도한다. 이를 통해 세션 쿠키나 민감한 정보를 탈취하고, 이후 이를 이용해 운영체제에 접근할 수 있다.
- **파일 업로드 취약점**: 웹 애플리케이션에서 사용자가 파일을 업로드할 수 있는 기능이 있을 때 발생할 수 있는 보안 취약점이다. 이 취약점은 공격자가 악의적인 파일을 서버에 업로드하여 시스템을 손상시키거나, 데이터에 접근하거나, 원격 코드 실행을 할 수 있는 기회를 제공한다.
- **원격 코드 실행 (RCE)**: 취약점은 공격자가 원격에서 악의적인 코드를 실행할 수 있는 보안 취약점이다. 이 취약점이 존재하는 시스템에서는 공격자가 네트워크를 통해 악성 코드를 전송하고, 이를 실행하여 시스템에 대한 제어를 획득 이를 통해 운영체제에 대한 접근 권한을 얻고, 민감한 정보를 수집하거나 시스템을 제어할 수 있다.

5. TACACS+ (10)

- 네트워크 장비의 관리자를 인증하고 권한을 부여하는 데 사용되는 프로토콜이다. 이 프로토콜은 TCP 기반으로 설계되어 있으며, 주로 라우터와 스위치와 같은 네트워크 장비의 보안을 강화하는 데 사용한다. TACACS+의 주요 기능인증 기능은 아래와 같다.

  ① 사용자가 네트워크 장비에 접근하기 전에 신원을 확인하는 과정을 제공한다. 이를 통해 불법적인 접근을 방지할 수 있다.
  ② 권한 부여: 인증이 완료된 후, 사용자가 어떤 작업을 수행할 수 있는지를 결정한다. 이를 통해 각 사용자에게 적절한 권한을 부여한다.
  ③ 어카운팅: 사용자의 활동을 기록하여 나중에 감사할 수 있도록 한다. 이는 보안 사고 발생 시 중요한 정보를 제공한다.

6. 지속적인 인증(12)

- 지속적인 인증(Continuous Authentication)은 웹 서비스나 애플리케이션에서 사용자의 신원을 지속하여 보안 프로세스를 의미한다. 이는 사용자가 처음 로그인한 후에도 세션 동안 지속하여 신원을 검증하여, 비정상적인 행동이나 보안 위협을 감지하고 대응하는 방식이다.

- 지속적인 인증의 주요 특징은 다음과 같다.
  ① 세션 중 지속적인 검증: 사용자가 로그인한 후에도, 시스템은 사용자의 행동 패턴, 위치, 장치 정보 등을 모니터링하여 사용자가 여전히 인증된 사용자임을 확인한다.
  ② 행동 기반 인증: 사용자의 행동(예: 마우스 움직임, 키 입력 패턴, 터치스크린 사용 방식 등)을 분석하여 비정상적인 행동이 감지되면 추가 인증을 요구한다.
  ③ 위치 기반 인증: 사용자의 위치 정보를 활용하여, 사용자가 예상치 못한 장소에서 로그인하거나 활동할 경우 추가적인 인증 절차를 요구한다.
  ④ 위험 기반 인증: 사용자의 행동이나 환경이 위험하다고 판단될 경우, 시스템은 추가 인증을 요구하거나 세션을 종료한다.
  ⑤ 다단계 인증 통합: 지속적인 인증은 다단계 인증(Multi-Factor Authentication, MFA)과 통합되어, 사용자가 특정 행동을 수행할 때 추가적인 인증 수단(예: SMS 코드, 생체 인식 등)을 요구한다.

7. 접근 제어(Access Control) (13)

- 운영체제의 접근 제어는 시스템 자원에 대한 접근을 관리하고 제한하는 메커니즘을 의미한다. 이는 사용자가 시스템 자원(파일, 프로세스, 네트워크 등)에 접근할 수 있는 권한을 설정하고, 이를 통해 보안을 유지하는 데 중요한 역할 수행한다. 접근 제어는 다음과 같은 주요 요소로 구성된다.

  ① 인증(Authentication): 사용자가 시스템에 접근하기 전에 자신의 신원을 증명하는 과정이다. 일반적으로 사용자 이름과 비밀번호, 생체 인식, 스마트카드 등을 사용하여 인증을 수행한다.
  ② 권한 부여(Authorization): 인증된 사용자가 어떤 자원에 접근할 수 있는지를 결정하는 과정이다. 이는 사용자의 역할(Role)이나 그룹(Group)에 따라 다르게 설정한다.
  ③ 접근 제어 목록(Access Control List, ACL): 특정 자원에 대한 접근 권한을 정의하는 목록이다. 각 자원에 대해 어떤 사용자나 그룹이 어떤 작업(읽기, 쓰기, 실행 등)을 수행할 수 있는지를 명시한다.
  ④ 역할 기반 접근 제어(Role-Based Access Control, RBAC): 사용자의 역할에 따라 접근 권한을 부여하는 방식이다. 사용자는 특정 역할에 할당되며, 그 역할에 따라 자원에 대한 접근 권한이 결정한다.
  ⑤ 정책(Policy): 접근 제어를 위한 규칙이나 기준을 정의하는 문서. 이는 조직의 보안 요구 사항에 따라 다르게 설정한다.

8. IP 주소 접근 권한 제어 (14)

- 시스템에 대한 접근 제어 정책이 기본적으로 IP(Internet Protocol)를 통해 수행된다는 의미는, 네트워크 환경에서 특정 IP 주소를 기반으로 사용자의 접근 권한을 관리하고 제한한다는 것이다. 이는 주로 다음과 같은 방식으로 이루어진다.

  ① IP 주소 기반 필터링: 시스템은 특정 IP 주소 또는 IP 주소 범위에 대해 접근을 허용하거나 차단하는 규칙을 설정할 수 있다. 예를 들어, 특정 네트워크(예: 회사 내부 네트워크)에서만 접근을 허용하고, 외부의 IP 주소에서의 접근은 차단하는 방식이다.
  ② 방화벽(Firewall): 방화벽은 네트워크 트래픽을 모니터링하고, 사전에 정의된 규칙에 따라 트래픽을 허용하거나 차단하는 장치이다. IP 주소를 기반으로 한 접근 제어 정책은 방화벽에서 자주 사용되며, 특정 IP 주소에서 오는 요청을 차단하거나 허용하는 규칙을 설정할 수 있다.
  ③ VPN(가상 사설망): VPN을 사용하면 외부에서 내부 네트워크에 안전하게 접근할 수 있다. 이 경우, VPN 서버는 사용자의 IP 주소를 확인하고, 허용된 IP 주소 목록에 있는 경우에만 접근을 허용한다.
  ④ 접근 제어 목록(ACL): 네트워크 장비(예: 라우터, 스위치)에서 ACL을 사용하여 특정 IP 주소에 대한 접근 권한을 설정할 수 있다. 이를 통해 특정 IP 주소에서의 트래픽을 허용하거나 차단할 수 있다.

- 이러한 IP 기반 접근 제어 정책은 네트워크 보안을 강화하고, 무단 접근을 방지하는 데 중요한 역할을 한다. 그러나 IP 주소는 변할 수 있기 때문에, IP 기반 접근 제어만으로는 완벽한 보안을 제공하기 어려운 경우도 있다. 따라서 다른 인증 및 권한 부여 메커니즘과 함께 사용되는 것이 일반적이다.

- TCP Wrapper는 네트워크 서비스에 대한 접근 제어를 제공하는 소프트웨어이다. 주로 리눅스 및 유닉스 시스템에서 사용되며, 특정 IP 주소나 호스트 이름에 따라 서비스에 대한 접근을 허용하거나 차단한다. TCP Wrapper는 주로 다음과 같은 기능을 제공한다.

  ① 접근 제어: /etc/hosts.allow와 /etc/hosts.deny 파일을 사용하여 특정 호스트나 네트워크에 대한 접근을 제어한다. 이를 통해 서비스에 대한 보안을 강화한다.
  ② 로깅: TCP Wrapper는 접근 시도를 기록할 수 있어, 보안 감사 및 문제 해결에 유용한다.
  ③ 서비스 보호: TCP Wrapper를 지원하는 서비스에 대해 접근 제어를 적용할 수 있으며, 이를 통해 서비스가 불법적인 접근으로부터 보호한다.

- TCP Wrapper는 SSH, FTP, Telnet 등 다양한 네트워크 서비스와 함께 사용할 수 있으며, 시스템 관리자가 네트워크 보안 강화에 도움을 준다.

9. IP 주소 접근 권한 제어 (15)

- inetd(Internet Demon)는 /etc/inetd.conf 파일을 통해 관리되며, 이 파일에서 어떤 서비스가 어떤 포트에서 대기할지를 설정한다.

- 서비스가 요청되면 inetd가 해당 서비스를 실행하고, 요청을 처리한다. TCP Wrapper는 /etc/hosts.allow와 /etc/hosts.deny 파일을 통해 접근 제어를 설정한다. 서비스 요청이 들어오면 TCP Wrapper(tcpd 데몬)가 먼저 접근 제어를 수행하고, 허용된 경우에만 해당 서비스로 요청을 전달한다.

10. 데이터베이스의 접근 제어 (16)

- NAMES.DIRECTORY_PATH = (TNSNAMES, EZCONNECT)는 Oracle 데이터베이스 클라이언트의 sqlnet.ora 파일에서 사용되는 매개변수로, 네임 리졸루션 방법의 우선순위를 설정한다. 이 매개변수는 클라이언트가 데이터베이스에 연결할 때 사용할 수 있는 네임 리졸루션 메커니즘을 정의한다.

- TNSNAMES(Transparent Network Substrate NAMES): TNSNAMES는 tnsnames.ora 파일을 통해 데이터베이스 연결 정보를 관리하는 방법이다. 이 파일에는 데이터베이스의 호스트, 포트, 서비스 이름 등의 정보가 포함된다. **클라이언트는 이 파일을 참조하여 데이터베이스에 연결한다.**

TNS_ALIAS: 데이터베이스에 연결하기 위한 별칭. 클라이언트가 이 이름을 사용하여 데이터베이스에 연결(=MYDB)
DESCRIPTION: 연결에 대한 설명을 포함하는 블록.
ADDRESS: 데이터베이스 서버의 주소를 정의.
PROTOCOL: 사용되는 프로토콜 (예: TCP).
HOST: 데이터베이스 서버의 호스트 이름 또는 IP 주소.
PORT: 데이터베이스가 수신 대기하는 포트 번호.
CONNECT_DATA: 데이터베이스에 연결하는 데 필요한 추가 정보
SERVICE_NAME: 연결할 데이터베이스 서비스의 이름.

이 예에서 MYDB라는 별칭을 사용하여 db.example.com의 1521 포트에서 mydbservice라는 서비스에 연결 - Oracle 데이터베이스에 연결하기 위한 TNS (Transparent Network Substrate) 구성한다.

```
MYDB =
    (DESCRIPTION =
        (ADDRESS = (PROTOCOL = TCP)(HOST = db.example.com)(PORT = 1521))
        (CONNECT_DATA = (SERVICE_NAME = mydbservice) )
)
```

- EZCONNECT: Oracle의 간단한 연결 방법으로, 별도의 TNS 이름 정의 없이도 데이터베이스에 연결할 수 있게 해준다. 사용자는 데이터베이스의 호스트 이름, 포트 번호, 서비스 이름을 직접 지정하여 연결할 수 있다. 예를 들어, hostname:port/service_name 형식으로 연결할 수 있다.

- 이 설정은 클라이언트가 데이터베이스에 연결할 때 어떤 방법을 우선적으로 사용할지를 결정한다. 위의 설정에서는 클라이언트가 먼저 tnsnames.ora 파일을 참조하여 TNS 이름으로 연결을 시도하고, 만약 해당 이름이 없거나 연결에 실패하면 EZCONNECT 방법을 사용하여 직접 연결을 시도한다.

- GRANT ALL PRIVILEGES ON mydatabase.* TO 'myuser'@'192.168.1.100' IDENTIFIED BY 'mypassword'

  ① ALL PRIVILEGES: 사용자가 가질 수 있는 모든 권한을 의미. 특정 권한만 부여하고 싶다면 SELECT, INSERT, UPDATE 등으로 변경할 수 있다.
  ② mydatabase.*: mydatabase 데이터베이스의 모든 테이블에 대한 권한을 의미. 특정 테이블에만 권한을 부여하고 싶다면 mydatabase.mytable과 같이 지정할 수 있다.
  ③ myuser'@'192.168.1.100': myuser라는 사용자가 192.168.1.100 IP 주소에서 접속할 수 있도록 설정한다.
  ④ IDENTIFIED BY 'mypassword': 사용자의 비밀번호를 설정. 이미 존재하는 사용자에게 권한을 부여할 경우 이 부분은 생략할 수 있다

11. 유닉스의 권한 관리 (19)

- 주어진 출력 drw-r-xr-x 117 root root 12288 Jul 28 06:42 etc는 유닉스 또는 리눅스 시스템에서 ls -l 명령어를 사용하여 디렉토리의 상세 정보를 보여주는 형식이다.
- 이 출력은 root 사용자와 root 그룹에 속하는 etc라는 이름의 디렉토리의 권한 및 속성을 보여준다. 각 부분의 의미는 다음과 같다.

① d: 첫 번째 문자는 파일의 유형을 나타낸다. 여기서 d는 이 항목이 디렉토리임을 의미한다.
② rw-r-xr-x: 다음 9자는 권한을 나타낸다.
③ rw-: 소유자(root)의 권한. 읽기(r)와 쓰기(w) 권한이 있으며, 실행(x) 권한은 없다.
④ r-x: 그룹(root)의 권한. 읽기(r)와 실행(x) 권한이 있으며, 쓰기(w) 권한은 없다.
⑤ r-x: 기타 사용자(others)의 권한. 읽기(r)와 실행(x) 권한이 있으며, 쓰기(w) 권한은 없다.
⑥ 117: 링크 수(link count). 이 디렉토리에 연결된 하위 디렉토리의 수를 나타낸다.
⑦ root: 소유자(user) 이름. 이 디렉토리의 소유자는 root이다.
⑧ root: 그룹(group) 이름. 이 디렉토리가 속한 그룹도 root이다.
⑨ 12288: 파일 크기. 이 디렉토리의 크기는 12,288 바이트이다.
⑩ Jul 28 06:42: 마지막 수정 날짜와 시간. 이 디렉토리는 7월 28일 06:42에 마지막으로 수정되었다.
⑪ etc: 디렉토리의 이름. 이 항목은 etc라는 이름의 디렉토리를 나타낸다.

12. 권한 제어 (22)

- 유닉스 및 유닉스 계열 운영 체제에서 nobody 계정은 일반적으로 제한된 권한을 가진 사용자 계정이다. 이 계정은 주로 시스템에서 특정 프로세스나 서비스가 최소한의 권한으로 실행하도록 사용된다. 이를 통해 시스템의 중요한 자원에 대한 접근을 제한하고, 잠재적인 보안 위협을 줄일 수 있다.nobody 계정의 주요 목적은 보안과 관련이 있다.

  ① nobody 계정의 특징 최소 권한 원칙: nobody 계정은 시스템의 다른 사용자나 프로세스에 대한 접근 권한이 거의 없거나 전혀 없는 상태로 설정되어 있다. 이를 통해 시스템의 보안을 강화할 수 있다.
  ② 서비스 실행: 웹 서버, FTP 서버, 또는 기타 네트워크 서비스와 같은 특정 서비스가 nobody 계정으로 실행될 수 있다. 이렇게 하면 해당 서비스가 시스템의 중요한 파일이나 데이터에 접근할 수 없도록 제한할 수 있다.
  ③ 파일 소유권: nobody 계정으로 생성된 파일은 일반적으로 다른 사용자와 그룹이 접근할 수 없는 상태로 설정된다. 이는 파일의 보안을 유지하는 데 도움이 된다.
  ④ 익명 사용자: nobody 계정은 종종 익명 사용자와 관련이 있다. 예를 들어, FTP 서버에서 익명 로그인 기능을 제공할 때 nobody 계정을 사용할 수 있다.

- "자신을 실행한 계정의 권한을 물려받는다"는 의미는, **특정 응용 프로그램이나 프로세스가 실행될 때, 그 프로세스가 실행되는 사용자 계정의 권한을 그대로 상속받는다는 것이다.** 즉, 응용 프로그램이 실행되는 환경에서 해당 계정이 가지고 있는 모든 권한과 접근 권한을 갖게 된다는 것을 의미한다. 이는 보안 취약점이 있는 응용 프로그램의 경우, 해당 계정의 권한이 악용될 수 있는 위험을 내포하고 있다. 이 개념의 중요성은 아래와 같다.

① 권한 상속: 응용 프로그램이 특정 사용자 계정으로 실행되면, 그 계정이 가진 권한(예: 파일 시스템 접근, 네트워크 접근, 시스템 리소스 접근 등)을 그대로 물려받는다. 따라서, 해당 응용 프로그램은 그 계정이 허용하는 모든 작업을 수행할 수 있다.

② 보안 취약점: 만약 응용 프로그램이 보안상 취약하다면, 악의적인 사용자가 그 응용 프로그램을 통해 해당 계정의 권한을 악용할 수 있다. 예를 들어, 취약한 웹 애플리케이션이 공격을 받으면, 공격자는 그 애플리케이션을 실행한 사용자 계정의 권한을 이용해 시스템에 대한 접근을 시도할 수 있다.

③ 최소 권한 원칙: 보안 관점에서, 응용 프로그램은 필요한 최소한의 권한만을 가져야 한다. 이를 통해 만약 응용 프로그램이 공격받더라도, 공격자가 얻을 수 있는 권한을 제한할 수 있다. 따라서, 응용 프로그램을 실행할 때는 별도의 사용자 계정을 설정하고, 그 계정에 최소한의 권한만 부여하는 것이 중요하다.

13. 로그 관리 – secure 파일 (29)

- 제시된 로그 항목 Jun 6 05:14:58 wishfreegdm-welcome] [847]: 이 로그 항목은 6월 6일 05:14:58에 "wishfree"라는 시스템에서 GDM 관련 프로세스가 실행되었음을 나타내며, 프로세스 ID는 847이다.

  ① Jun 6 05:14:58: 로그가 기록된 날짜와 시간을 나타낸다. 여기서 "Jun"은 6월을 의미하고, "6"은 날짜, "05:14:58"은 시간(시:분:초)을 나타낸다.
  ② wishfree: 로그를 생성한 호스트 이름 또는 시스템의 이름을 나타낸다. 즉, "wishfree"라는 이름의 서버 또는 컴퓨터에서 로그가 기록되었다는 것을 의미한다.
  ③ gdm-welcome: 로그 메시지의 출처 또는 관련된 프로세스를 나타낸다. "gdm"은 GNOME Display Manager의 약자로, 리눅스 및 유닉스 계열 운영 체제에서 그래픽 로그인 화면을 제공하는 프로그램이다. "gdm-welcome"은 GDM의 환영 메시지 또는 관련된 이벤트를 나타낼 수 있다.
  ④ [847]: 로그 메시지와 관련된 프로세스 ID(PID)를 나타낸다. 즉, 847이라는 숫자는 이 로그 메시지를 생성한 프로세스의 고유 식별자이다.

- 로그 항목 Jun 6 05:14:58 wishfreepolkitd(authority=local): 6월 6일 05:14:58에 "wishfree"라는 시스템에서 PolicyKit 데몬이 로컬 권한을 사용하여 어떤 작업을 수행했음(사용자가 시스템에 등록된 후 그 사용자의 신원을 확인)을 나타낸다.

  ① Jun 6 05:14:58: 로그가 기록된 날짜와 시간을 나타낸다. "Jun"은 6월을 의미하고, "6"은 날짜, "05:14:58"은 시간(시:분:초)을 나타낸다.
  ② wishfree: 로그를 생성한 호스트 이름 또는 시스템의 이름을 나타냅니다. 즉, "wishfree"라는 이름의 서버 또는 컴퓨터에서 로그가 기록되었다는 것을 의미한다.
  ③ polkitd: 로그 메시지의 출처를 나타낸다. "polkitd"는 PolicyKit데몬을

의미하며, 이는 리눅스 시스템에서 권한 관리를 담당하는 서비스이다. PolicyKit은 사용자와 시스템 간의 권한을 관리하고, 특정 작업을 수행하기 위해 필요한 권한을 부여하는 역할을 수행한다.

④ (authority=local): "polkitd"가 사용하는 권한의 출처를 나타냅니다. "authority=local"은 로컬 권한을 의미하며, 이는 해당 작업이 로컬 사용자에 의해 요청되었음을 나타낸다.

14. 응용 프로그램의 로그 관리(38)

- GET/XSS/GetCookie.asp?cookie=ASPSESSIONIDQQCAQDDA80는 웹 애플리케이션에서 특정 리소스를 요청하는 HTTP GET 요청을 나타낸다.

  ① GET: HTTP 메서드 중 하나로, 서버에 리소스를 요청하는 데 사용된다. 이 경우, GetCookie.asp라는 ASP 파일에서 특정 정보를 요청하고 있다.
  ② /XSS/GetCookie.asp: 요청하고자 하는 리소스의 경로이다. GetCookie.asp는 ASP(Active Server Pages)로 작성된 서버 측 스크립트 파일로, 쿠키와 관련된 정보를 처리하는 기능을 있을 가능성이 크다.
  ③ ?cookie=ASPSESSIONIDQQCAQDDA80: URL의 쿼리 문자열 부분으로, cookie라는 파라미터에 ASPSESSIONIDQQCAQDDA80라는 값이 전달되고 있다. 이 값은 ASP 세션 ID를 나타내며, 사용자의 세션을 식별하는 데 사용된다.

- 이 URL은 XSS 공격과 관련된 예시로 보일 수 있으며, 공격자가 세션 ID를 포함한 요청을 통해 사용자의 세션을 탈취하려고 시도할 수 있다. 따라서, 이러한 요청을 처리하는 서버 측 스크립트는 적절한 보안 조치를 취해야 하며, 입력값을 검증하고 인코딩하여 XSS 공격을 방지해야 한다.

15. 403 Forbidden (40)

- HTTP 상태 코드 403 Forbidden은 클라이언트가 요청한 리소스에 접근할 권한이 없음을 의미한다. 즉, 서버는 요청을 이해했지만, 클라이언트가 해당 리소스에 접근할 수 있는 권한이 없으므로 요청을 거부한 것이다. 이 오류는 여러 가지 이유로 발생할 수 있다.

  ① 인증 문제: 사용자가 로그인하지 않았거나, 필요한 권한이 없는 경우.
  ② IP 차단: 특정 IP 주소나 IP 범위가 차단된 경우.
  ③ 파일 권한: 서버의 파일이나 디렉토리에 대한 접근 권한이 설정되어 있지 않은 경우.
  ④ 리소스 제한: 특정 리소스에 대한 접근이 제한된 경우 (예: 관리자 전용 페이지).

16. SIEM, MRTG, NMS

- SIEM(Security Information and Event Management; 빅데이터 기반 보안관제 시스템)은 보안 정보와 이벤트를 실시간으로 수집, 분석, 저장 및 보고하는 시스템이다. 다양한 소스에서 발생하는 로그 데이터를 중앙에서 관리하여 보안 위협을 탐지(실시간 모니터링)하고 대응한다. 주요 기능은 다음과 같다.
  ① 데이터의 수집: 다양한 소스(서버, 네트워크 장비, 애플리케이션 등)에서 로그와 이벤트 데이터를 수집한다.
  ② 데이터 저장: 수집된 데이터를 중앙 집중식으로 저장하여 나중에 분석할 수 있도록 한다.
  ③ 실시간 모니터링: 실시간으로 보안 이벤트를 모니터링하여 이상 징후를 탐지한다.
  ④ 분석 및 상관관계: 수집된 데이터를 분석하고, 서로 다른 이벤트 간의 상관관계를 찾아내어 보안 위협을 식별한다.
  ⑤ 경고 및 대응: 위협이 탐지되면 경고를 생성하고, 필요한 경우 자동으로 대응 조처할 수 있다.

- MRTG(Multi Router Traffic Grapher)는 네트워크 트래픽을 모니터링하고 시각화하는 데 사용되는 오픈 소스 소프트웨어이다. MRTG는 주로 라우터와 스위치와 같은 네트워크 장비의 트래픽을 측정하고, 이를 그래프 형태로 표시하여 네트워크 성능을 분석한다.
  ① 트래픽 모니터링: 네트워크 인터페이스의 입력 및 출력 트래픽을 지속적으로 모니터링한다.
  ② 그래프 생성: 수집된 데이터를 기반으로 시간에 따른 트래픽 변화를 시각적으로 표현하는 그래프를 생성한다.
  ③ 웹 인터페이스: 생성된 그래프는 웹 페이지를 통해 쉽게 접근할 수 있으며, 이를 통해 네트워크 상태를 실시간으로 확인할 수 있다.
  ④ 다양한 프로토콜 지원: SNMP(Simple Network Management Protocol)를 사용하여 다양한 네트워크 장비에서 데이터를 수집할 수

있다.

- NMS(Network Management System)는 네트워크를 관리하고 모니터링에 사용되는 소프트웨어 시스템을 의미한다. NMS는 네트워크 장비(라우터, 스위치, 방화벽 등)의 성능, 가용성, 보안 및 구성 상태를 관리하는 데 도움을 준다. NMS 솔루션으로는 SolarWinds, Nagios, Zabbix, PRTG Network Monitor 등이다.
  ① 모니터링: 네트워크 장비와 트래픽을 실시간으로 모니터링하여 성능을 분석한다.
  ② 장비 관리: 네트워크 장비의 상태를 점검하고, 문제 발생 시 경고를 제공한다.
  ③ 구성 관리: 네트워크 장비의 설정을 관리하고, 변경 사항을 기록한다.
  ④ 성능 분석: 네트워크 성능을 분석하여 병목 현상이나 문제를 식별한다.
  ⑤ 보고서 생성: 네트워크 상태와 성능에 대한 보고서를 생성하여 관리자가 쉽게 이해할 수 있도록 한다.

17. SNMP

- SNMP는 트워크 장비를 관리하고 모니터링하기 위한 프로토콜이다. SNMP는 주로 라우터, 스위치, 서버, 프린터 등 다양한 네트워크 장비의 상태와 성능을 관리하는 데 사용된다. SNMP의 주요 구성 요소는 다음과 같다.
  ① 관리자(Manager): 네트워크를 관리하는 시스템으로, SNMP를 통해 네트워크 장비에서 정보를 수집하고 제어한다.
  ② 에이전트(Agent): 네트워크 장비에 설치된 소프트웨어로, 장비의 상태와 성능 데이터를 수집하고 이를 관리자에게 전달한다.
  ③ 관리 정보 베이스(MIB): SNMP에서 관리되는 데이터의 구조를 정의하는 데이터베이스이다. MIB는 네트워크 장비의 다양한 매개변수(예: 트래픽, CPU 사용량 등)를 설명하는 객체를 포함하고 있다.

- SNMP는 네트워크 관리의 표준 프로토콜로 널리 사용되며, 다양한 네트워크 관리 도구와 함께 사용되어 네트워크의 효율성과 안정성 향상에 기여한다.
  ① 모니터링: 네트워크 장비의 상태와 성능을 실시간으로 모니터링할 수 있다.
  ② 경고 및 알림: 특정 조건이 발생했을 때 관리자에게 경고를 보낼 수 있다.
  ③ 구성 관리: 네트워크 장비의 설정을 변경하거나 업데이트할 수 있다.

- SNMP와 TCP/IP와의 관계

  ① SNMP는 TCP/IP의 응용 계층 프로토콜로, SNMP는 네트워크 장비를 모니터링하고 관리하기 위한 프로토콜이다.
  ② SNMP 메시지는 UDP(User Datagram Protocol)를 통해 전송된다, UDP는 비연결형 프로토콜로, 빠른 데이터 전송을 가능하게 하지만 신뢰성은 보장하지 않는다.
  ③ SNMP는 TCP/IP 네트워크에서 장비의 상태를 모니터링하고, 성능 데이터를 수집하며, 장비의 설정을 변경하는 데 사용된다.

18. Sendmail 데몬

- Sendmail 데몬은 이메일 전송을 처리하는 소프트웨어로, Unix 및 Unix-like 운영 체제에서 널리 사용. Sendmail은 이메일 메시지를 전송하고 수신하는 데 필요한 다양한 기능을 제공하며, SMTP(간단한 메일 전송 프로토콜, 포트번호 25번)를 사용하여 이메일을 전송한다.

- vrfy : 특정 이메일 주소가 유효한지를 확인하는 명령어

- vrfy wishfree: wishfree계정이 존재하는가? sendmail데몬에 문의함을 의미