3주차 3차시 취약점관리

【학습목표】

1. 취약점 관리의 개념을 설명할 수 있다.
2. 응용 프로그램별 고유 위험 관리를 설명할 수 있다.

학습내용1 : 취약점관리의 개념 및 패치 관리

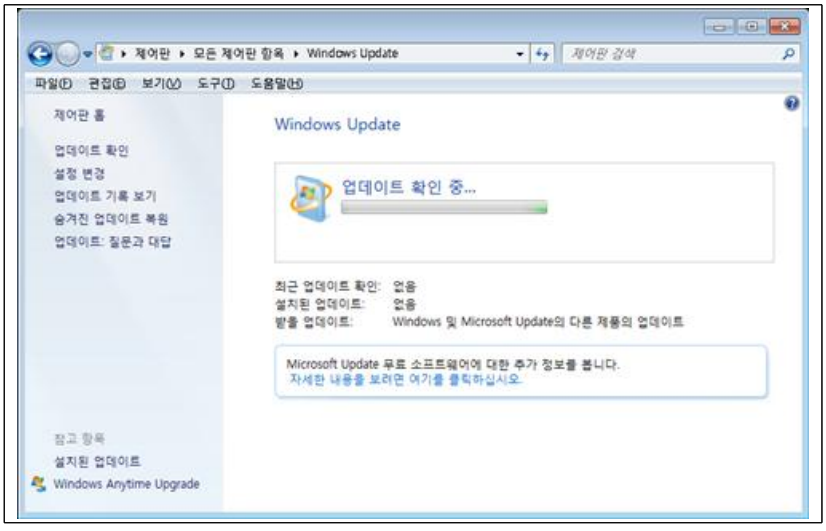
1. 취약점 관리의 개념

① 패치를 확인하고, 패치를 적용하는 패치관리 : 패치의 배포자의 신뢰성, 배포처
② 응용프로그램 별 고유 위험관리 : SQL, XP_CmdShell 등 어플이 가진 고유한 위험
③ 응용프로그램을 통한 정보수집 제한 : 어플이 OS의 정보유출X

2. 패치 관리

<윈도우 업데이트를 통해 자동으로 보안 패치를 확인하고, 패치를 적용할 수 있음>

1) [그림 2-34] 윈도우 업데이트 항목 확인

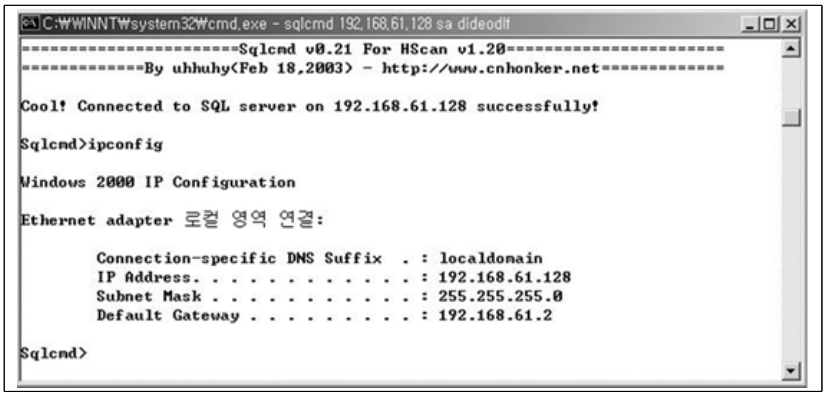


학습내용2 : 취약점관리의 응용

1. 응용 프로그램별 고유 위험 관리

- 응용 프로그램 중에는 해당 응용 프로그램을 통해 운영체제의 파일이나 명령을 실행시킬 수 있는 것이 있음
- MS-SQL의 xp_cmdshell은 데이터베이스를 통해 운영체제의 명령을 실행하고, 파일 등에 접근할 수 있음
- 응용 프로그램의 동작과 관련하여 운영체제에 접근할 수 있는 함수나 기능이 있으면 그 적절성을 검토해야 함

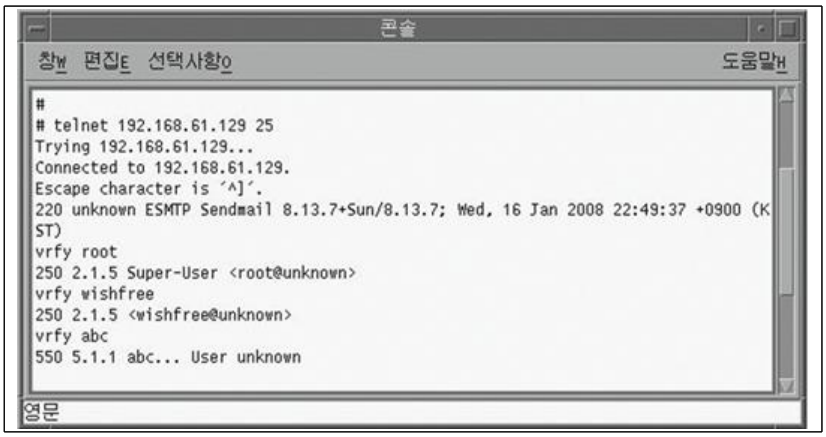
1) [그림 2-35] MS-SQL 2000에서 xp_cmdshell 툴을 이용한 명령창 획득



2. 응용 프로그램을 통한 정보 수집 제한

- 응용 프로그램이 운영체제에 직접적인 영향을 미치지 않아도 응용 프로그램의 특정 기능이 운영체제의 정보를 노출시키기도 함
- 유닉스에서는 이메일을 보낼 때, 수신자가 있는 시스템의 sendmail 데몬에 해당 계정이 존재하는지 확인하기 위해 일반 계정은 vrfy(verify) 명령을, 그룹은 xpn(expansion) 명령을 시스템 내부적으로 사용
- 일반 사용자는 다음과 같이 Telnet을 이용해 시스템에 존재하는 계정의 목록을 어느 정도 파악할 수 있음
- 이러한 응용 프로그램의 기능은 제한하는 것이 바람직함

1) [그림 2-36] sendmail 데몬에 접속하여 vrfy 명령을 실행한 결과



【학습정리】

1. 패치관리는 패치배포자의 신뢰성, 배포처가 가장 중요한 요소이다.

2. 응용프로그램 별 고유 위험관리는 XP_CmdShell 등 어플이 가진 위험성이 예이다.