5.7 신뢰 모델(Trust Model)

개요

사용자가 인터넷에서 소프트웨어를 다운로드할 때, 그 소프트웨어에 악성 코드가 없다는 커다란 신뢰가 필요합니다. 컴퓨터에 다운로드하는 소프트웨어가 여러분 컴퓨터에 있는 모든 것을 삭제할 수도 있기 때문입니다. 하지만, 우린 여전히 우리가 다운로드하는 소프트웨어가 안전하다고 믿습니다.
이것이 **신뢰 모델**의 기본입니다.

핵심개념

* 신뢰 모델
* 백도어

백도어(Back Door)

```
1  if ((strcmp(username, "rob") == 0 &&
2         strcmp(password, "thisiscs50") == 0) ||
3      (strcmp(username, "tommy") == 0 &&
4         strcmp(password, "i<3javascript") == 0))
5  {
6      printf("Success!! You now have access.\n");
7  }
8  else if (strcmp(username, "hacker") == 0 &&
9         strcmp(password, "LOLihackedyou") == 0)
10 {
11     printf("Hacked!! You now have access.\n");
12 }
13 else
14 {
15     printf("Invalid login.\n");
16 }
```

▲ 〈코드 1〉

〈코드 1〉은 사용자 이름과 비밀번호를 확인하여 사용자 계정 자격이 유효한지 확인하는 가상의 로그인 프로그램입니다. 로그인 프로그램들은 1행부터 7행과 비슷한 방법으로 사용자 이름과 비밀번호를 데이터베이스에 들어있는 것과 비교합니다. strcmp는 두 인자(문자열)가 동일한지 비교하는 함수입니다.

사용자 이름과 비밀번호가 맞는지 확인한 후에 시스템에 대한 액세스 권한을 부여하는 또다른 방법이 쓰여진 코드가 8행부터 12행에 있습니다.
사용자들이 원래 시스템에 접속하는 **방법과는 다른 비정상적인 수단으로 시스템에 접속하는 방법을 백도어(Backdoor)**라고 부릅니다.

이 로그인 프로그램의 코드를 읽는 사용자는 시스템에 백도어가 있다는 것을 알아차릴 수 있지만 대부분의 사용자는 프로그램이 컴파일 되기 전의 프로그램 코드를 볼 수 없습니다.

컴파일러 내부의 익스플로잇(Exploit)

프로그램을 다운로드하기 전에 프로그램의 코드를 볼 수 있고, 그 코드에 악성 코드나 백도어가 없어 보인다고 해도 프로그램 자체가 안전한 것은 아닙니다. 소스 코드를 오브젝트 코드로 변환시키는 프로그램인 컴파일러가 **익스플로잇(exploit)**의 원천일 가능성이 있습니다. 예를 들어 컴파일러는 백도어가 없는 로그인 프로그램에 백도어를 만드는 코드를 주입하도록 프로그램될 수 있습니다. 로그인 프로그램의 코드를 보더라도 백도어의 흔적을 찾을 수 없을 것입니다. 만약 소스 코드가 악성 컴파일러에 의해 컴파일 되었다면, 결과적으로 프로그램에는 백도어가 있을 것입니다.

물론 컴파일러의 코드를 보게 된다면 로그인 프로그램에 악성 코드를 주입하는 코드가 컴파일러에 있다는 것을 알 수 있습니다. 하지만 컴파일러에 악성 코드를 주입하는 컴파일러를 만들 수 있다면, 이론상으로 해커는 악의가 없는 소스로 된 컴파일러를 악성 컴파일러로 만들 수 있습니다.

결론적으로, 컴파일러의 소스 파일과 로그인 프로그램의 소스 파일이 악성코드나 백도어를 포함하지 않을 지라도, 소스 파일을 컴파일 하는 과정에서 악성 코드가 주입될 수 있습니다.