Daniel Garcia

ECE 548

Homework 2

3.7 Because of the known risks of the UNIX password system, the SunOS-4.0 documentation recommends that the password file be removed and replaced with a publicly readable file called /etc/publickey. An entry in the file for user A consists of a user’s identifier *IDA*, the user’s public key, *PUa*, and the corresponding private key *PRa*. This private key is encrypted using DES with a key derived from the user’s login password *Pa*. When A logs in, the system decrypts E(*Pa*, *PRa*) to obtain *PRa*.

1. The system then verifies that *Pa* was correctly supplied. How?
2. How can an opponent attack this system?

**Answer**

a. Public and private keys have an important relationship. When something is encrypted with one key, and can be decrypted by the other. If for example, the decrypted public key was used to encrypt the ID of the user and then the ciphertetxt was decrypted by the public key, then the resulting plaintext would be the ID of the user if and only if the public key was decrypted properly by the correct password.

b. The biggest flaw in the system is the use of the DES algorithm. Though fast, the DES algorithm is very simple to decrypt and in today’s security landscape is considered very vulnerable. In addition, by cracking DES, not only does the attacker gain access to any communications that the user makes, but additionally obtains the password to the account which can be used to escalate privileges and even possibly gain access to the root account in the system.

3.8 It was stated that the inclusion of the salt in the UNIX password scheme increases the difficulty of guessing by a factor of 4096. But the salt is stored in the plaintext in the same entry as the corresponding ciphertext password. Therefore, those two characters are known to the attacker and need not be guessed. Why is it asserted that the salt increases security?

**Answer**

Yes. Added a salt to a password even in plaintext in the same file still adds extra security that does not need to be protected. The reason for this is that often times attackers will hash common passwords and look them up in a database. If there is no way for common passwords to be found, the attacker must resort to cracking the hashes manually. Even with the salts, attackers still need to guess the entire password correctly and waste time hashing the data in order to guess it.