Why does it take longer to reject an invalid password than to accept a valid one?

devblogs.microsoft.com/oldnewthing/20100323-00

March 23, 2010

Raymond Chen

You may have noticed that it takes longer to reject an invalid password than to accept a valid one. There are a few reasons for this. First of all, it simply takes longer to confirm that a password is invalid. Your local computer retains a password cache. This password cache improves performance of local authentication operations, such as unlocking a workstation. If you unlock the workstation with the same password you used to log on, then the password is assumed to be good. This allows the workstation to unlock quickly. Without the password cache, unlocking the workstation would require going back to the domain controller to validate the password, which for slow network connections, can take a very long time. In fact, you might not have any network connection at all (an extreme case of *slow*), say because you've taken your laptop on the road and are not connected to the corporate network any more. In that case, without the password cache, it would be impossible for you to unlock your workstation at all! Note that you can disable these password caches if they offend you. The algorithm for testing if a password is valid goes like this:

1. If password is in cache and matches: Return *valid*.
2. Else contact domain controller for password validation.

If you pass a valid password, then the validation succeeds at step 1. Notice that step 1 can be performed entirely on the local machine. It doesn't need to contact any other computers to get an answer. On the other hand, if you pass an invalid password, then we go on to step 2, which attempts to contact the domain controller to validate the password. Obviously you have to do this for passwords not in the cache, because you have no information about those passwords. But why do you also have to do this for passwords that are in the cache and don't match? Why don't you just say *invalid* without contacting the domain controller? Because your cache itself may be invalid. If the user recently changed the password on another machine, then the password in your machine's cache is not valid. If the user tries to use the new password, your computer's cache says, "Nope, that's not the right password." If you returned *invalid* immediately instead of contacting the domain controller, then users whose passwords have changed would not be able to use that password to access any computer which had cached the old password! They would have to sit around and wait for the old

password to fall out of the cache, so that the computer would continue to step 2 and get the new password from the domain controller. You can imagine the bizarro technical support calls that would have resulted. “Yes, I know you changed your password, but you have to keep using your old password until the system starts rejecting it, and then you switch to the new password. And the rejection time will vary from computer to computer, depending on how many other people use the computer also. Oh, and if you’re the only person who uses the computer, then it will *never* accept your new password. But once you log onto the computer with the old password, you might need to give your *new* password when connecting from that machine to *other* machines, because those other machines might have received your new password.” Okay, so one reason why invalid passwords take longer to reject is that the computer has to try more things before finally deciding to reject it. Another reason why invalid passwords take longer to reject is to reduce the effectiveness of dictionary attacks. If invalid passwords were rejected just as quickly as valid passwords were accepted, then a bad guy could just churn through a dictionary trying out invalid passwords at high speed. Adding a delay of a few seconds before rejecting invalid passwords introduces a minor inconvenience to users who mistyped their passwords, but makes a huge dent in stopping dictionary attacks. For example (and these numbers are completely made up), suppose you have a 75,000 word password dictionary, and passwords are accepted or rejected in 100ms. It would take a little over three hours to attempt every password in the dictionary. Introducing even a simple 5-second delay into the rejection of invalid passwords increases the time to perform a dictionary search to over four days.

The invalid password rejection time in some places can get quite high, especially if the delay escalates each time you get the password wrong. For example, after you type the third (fourth?) incorrect password to the Windows logon screen, it displays the *incorrect password* error for something like 30 seconds before letting you try again.

Raymond Chen

Follow