1. **Include a list of all passwords that you cracked, where each row should have the username, the password hash, and the cleartext password.**

win-pwd.txt:

nancydrew:DISENGAGES::0B6D06B34A0B437D3BE64BC57F89E16C:::

sherlock:5LYYPM::7F6010574E24B264D81F0225164AEE6C:::

winpass.txt:

Administrator:???????FR$#::8D00A0D9A090799F7F82A7100C91D3D2:::

andrea:COFF33::15E837FB855C92A70A1D75B91226E924:::

andrew:S0CCR1::2A6BB0E4AAA8E933661977EC5EFD02F6:::

carlos:DENTIST::4A546A8FD69A1FE0121C00EB7404B773:::

michael:ABACUS5::4BE619B5D33FD71EB222AD1DF041B8BC:::

richard:Z1AYPM::A3D5EAE39100B5276794C11DA366B071:::

robert:PANACHE::D6B3D2E9E6D4EE5443996E01BC448CBC:::

ruiyang:???????&%#:1018:942704C68B36CA0F95AB190ABB79B9D2:::

smith:DC72B4:1017:3CF77AE0DFA519BCDD7CFCD64B3FAE16:::

lin-pwd.txt:

magellan:subjugated7:1oMRtGdiY$CdaII9KyvKWqKsYBwJwVq0::::

marcopolo:grimaces:1OUnMA/nW$XdvYJVquDCgce0cCrBuiJ1::::

linux_passwords.txt:

SHA512 CANNOT BE CRACKED...

2. **Name at least three reasons why LM hashes are easier to crack than salted SHA-1 hashes.**

A. Passwords are encoded based on DES (which is unsafe today) and are limited to a maximum of only 14 characters, given a tractable key space to crack.

B. Passwords longer than 7 characters are divided into two pieces and each of them is hashed separately. This narrows the key space substantially and allows the brutal-force attack being more efficiently.
C. Without the salt, LM is vulnerable to pre-computed dictionary attacks (i.e. rainbow table attack)

3. **LM hashes are disabled by default in Windows Server 2012. However, many administrators enable it on their servers. In previous versions, it was enabled by default. Why are LM hashes still required?**

Because some old systems are still running with LM, LM is enabled to let the system be compatible with the old ones.

4. **Suppose a user selects a random 8 character password from the set of characters `[A-Za-z0-9]`. The password is stored as an unsalted SHA-1 hash. If an attacker wishes to precompute all possible 8 character password hashes for this character set and store the pairs in a simple list, how many megabytes of disk space would this require at a minimum?**

   Assume that the passwords are stored as 8-bit ASCII characters and that the pairs of password/hash are separated by a single ASCII character.

   Each SHA1 hash is 20 bytes long, and each password is 8 bytes, plus the separator 1 byte, we have 29 bytes for each record.

   Also notice that there are 62 possibilities for each position in the 8 character password, hence the total number of possibilities is 62^8.

   So, the total number of space needed to store all the possibilities is:

   62^8 * 29 = 6331863061961984 bytes = 6331863062 megabytes