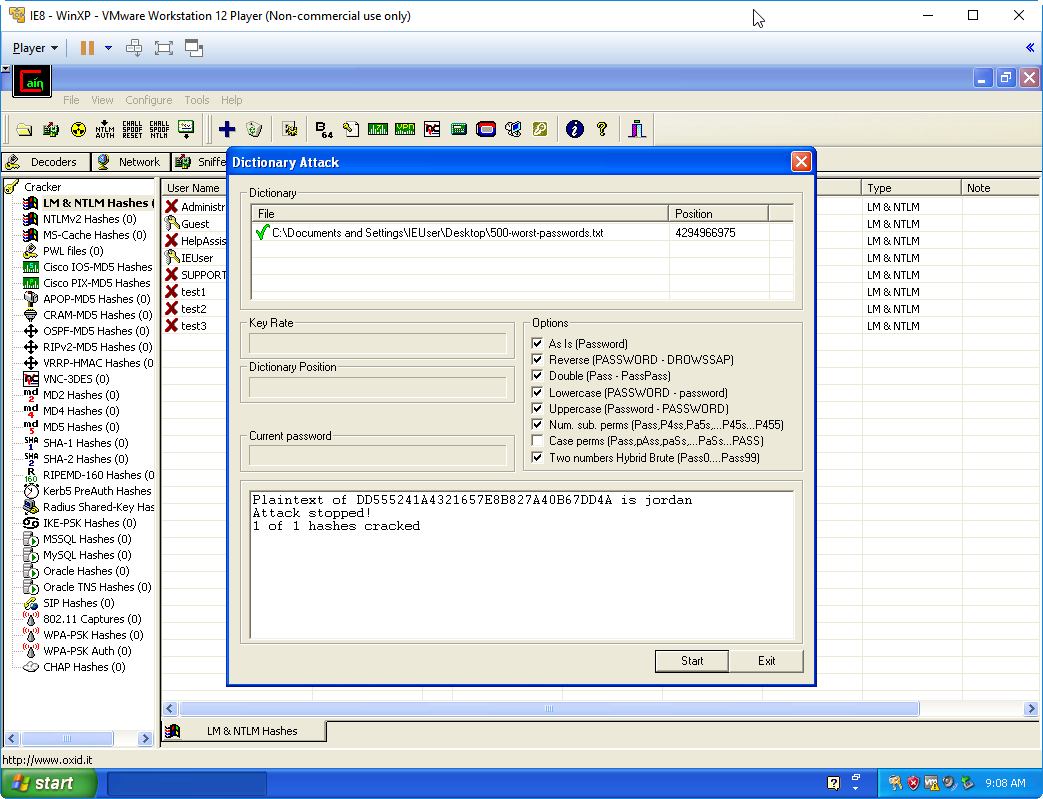
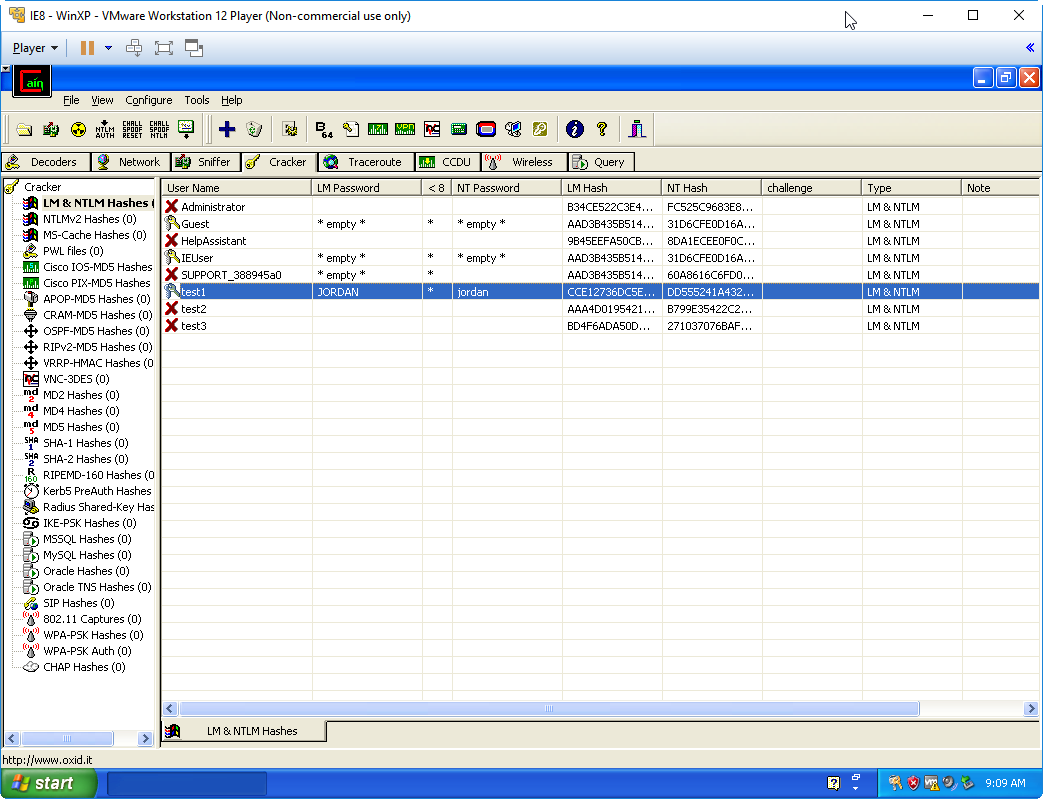
1)Password discovered from Task 1

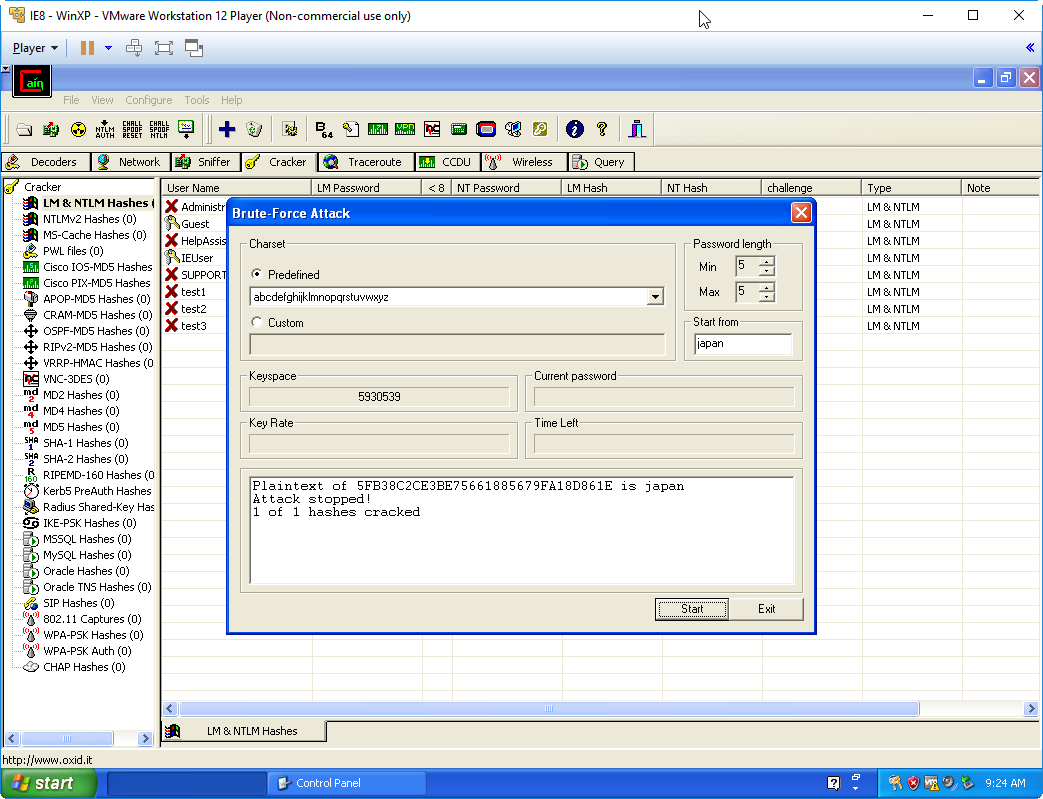
The password discovered for **test1** user account is “jordan”

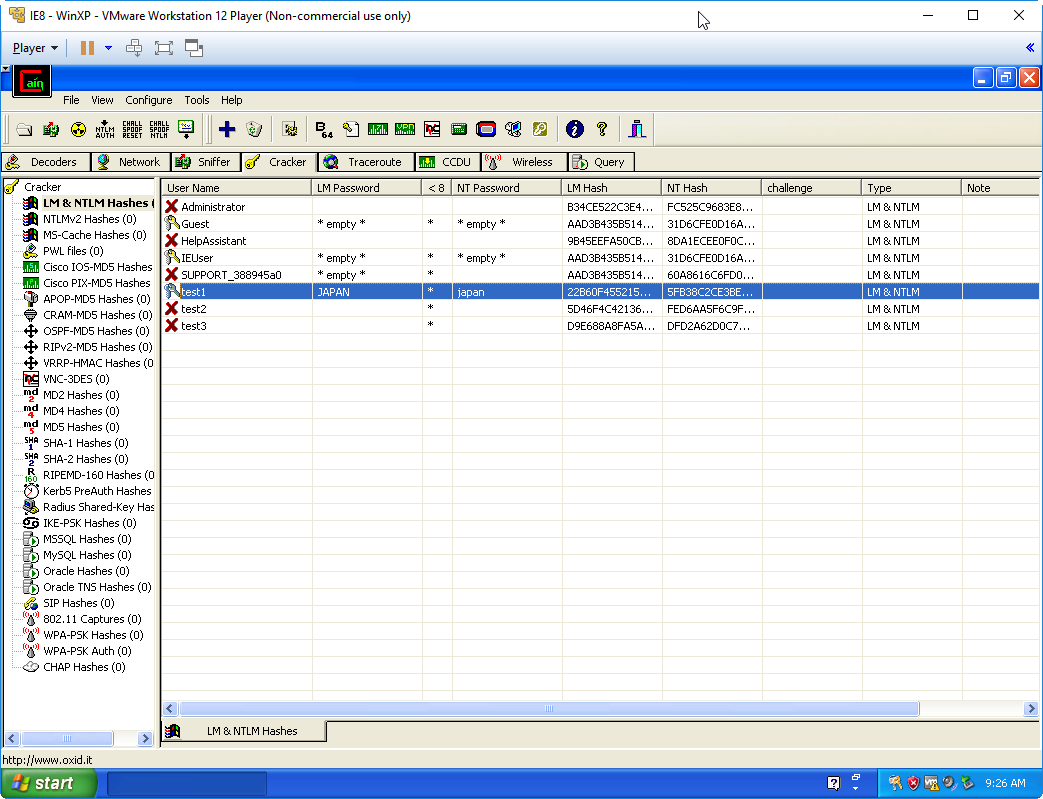


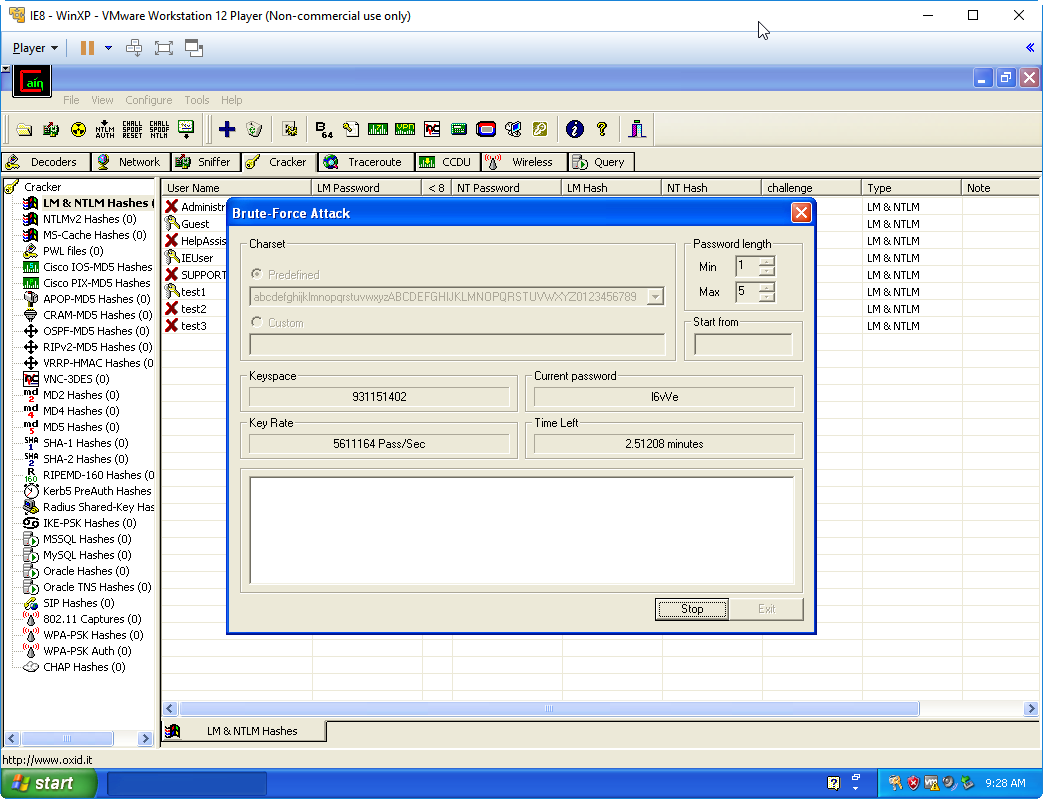


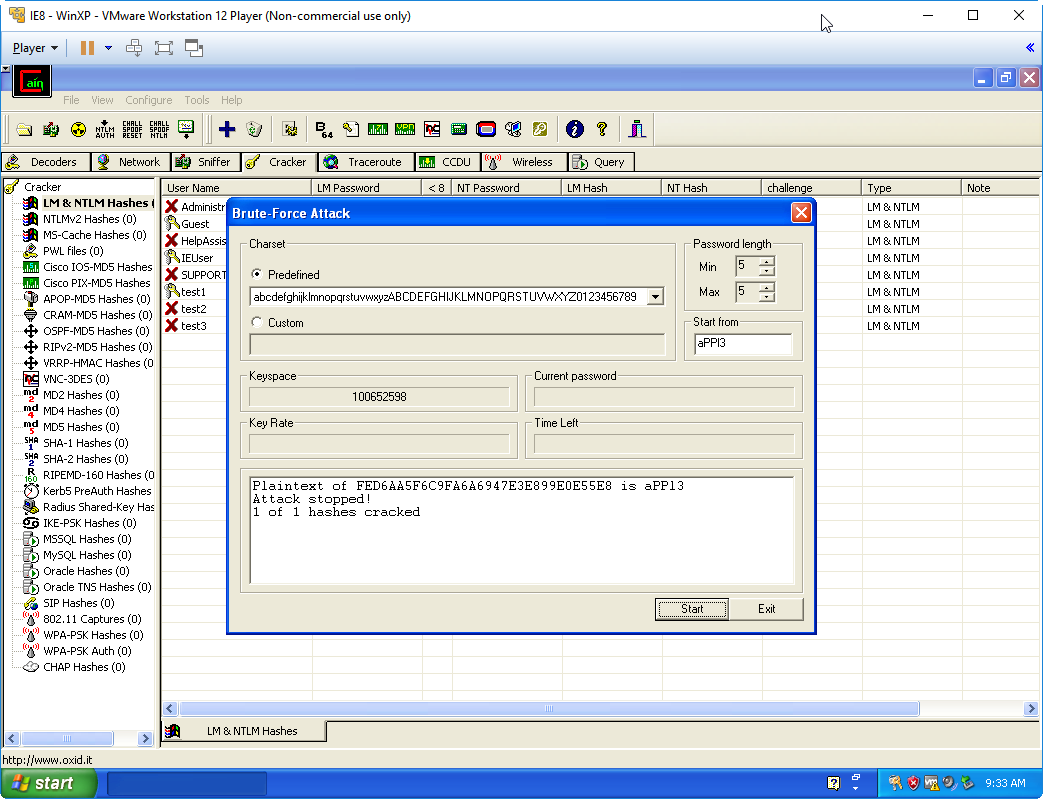
2) Filled table from Task 2

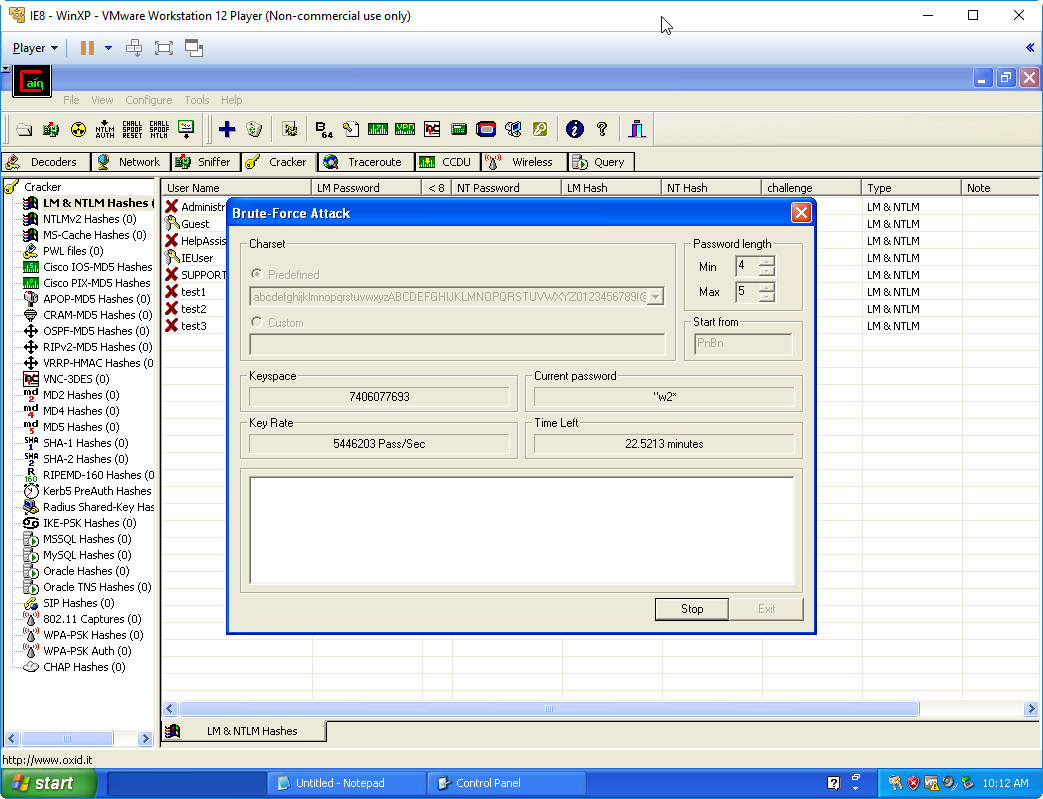
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Password Description** | **Chosen password** | **Charset** | **Time taken** |
| 1 | Lowercase letters only (length 5) | japan | abcdefghijklmnopqrstuvwxyz | 2 seconds |
| 2 | Lowercase, uppercase letters and numbers from 0 to 9 (length 5) | aPPl3 | abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 | 3 min |
| 3 | Lowercase, uppercase letters, numbers from 0 to 9 and symbols (length 5) | @PPl3 | abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&\*() \_+|}{“:?><( | 22 min |

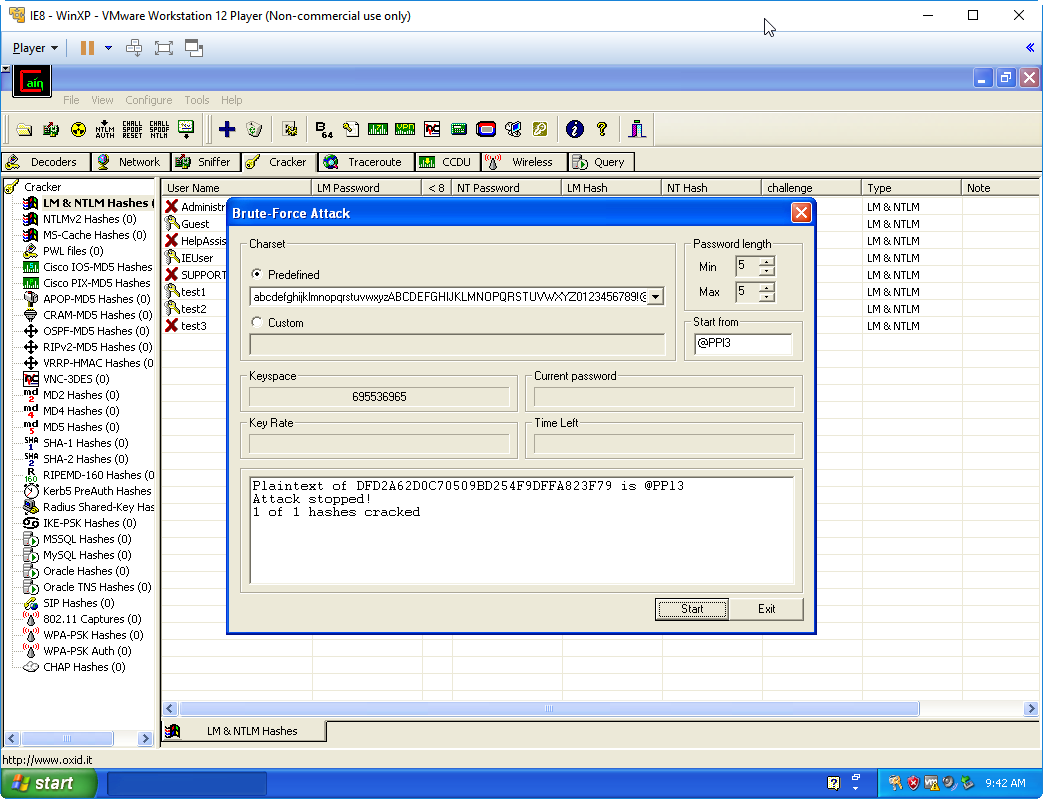
**Test1**:  


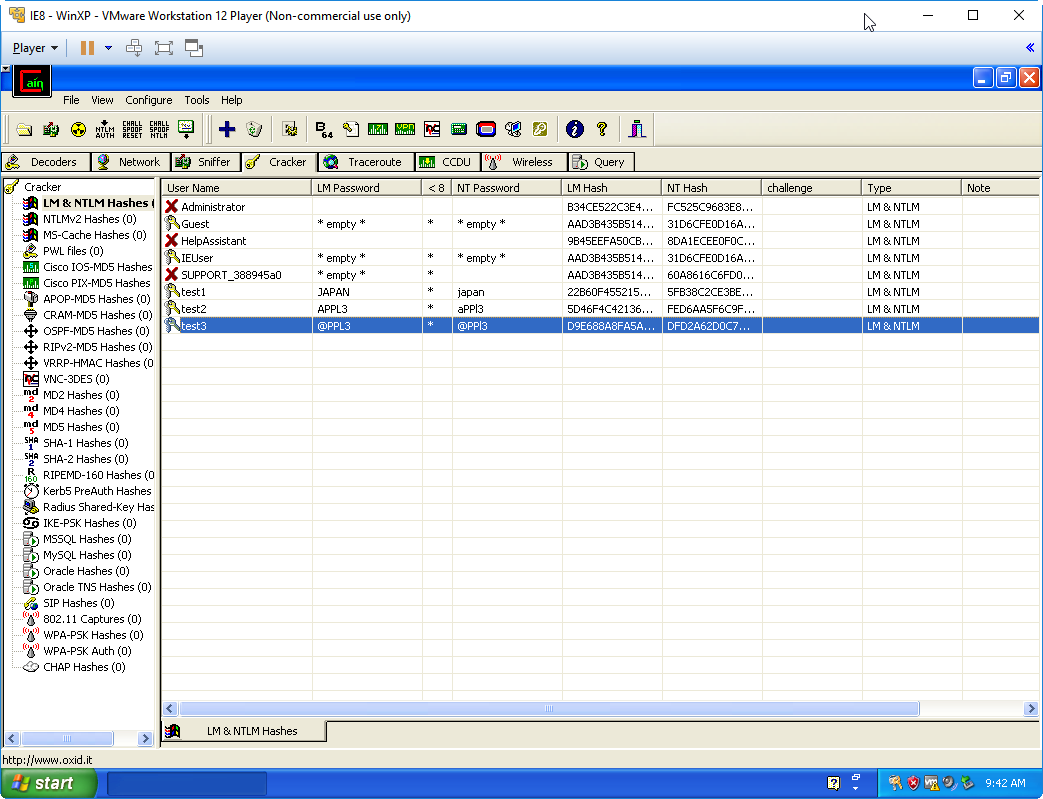


**Test2**:  




**Test3**:  
  






When you created passwords for the brute force attack, would Cain & Abel have finished faster if your password didn’t include all the character types in the password description? So, for example if the description said “lower and uppercase letters”, and if your chosen password was “aaa”, would Cain and Abel have discovered it faster than if you had chosen “aBC”? Remember that in real scenarios, if you were trying to recover a password using a tool like Cain & Abel, you would not know what the password was, only what the password space was!

1. Cain and Abel would have finished the brute force attack with less character types.The keyspace size increases if the number of character types increases (screenshots have the keyspace sizes).Thus **aaa** will take less time to crack than **aBC** .