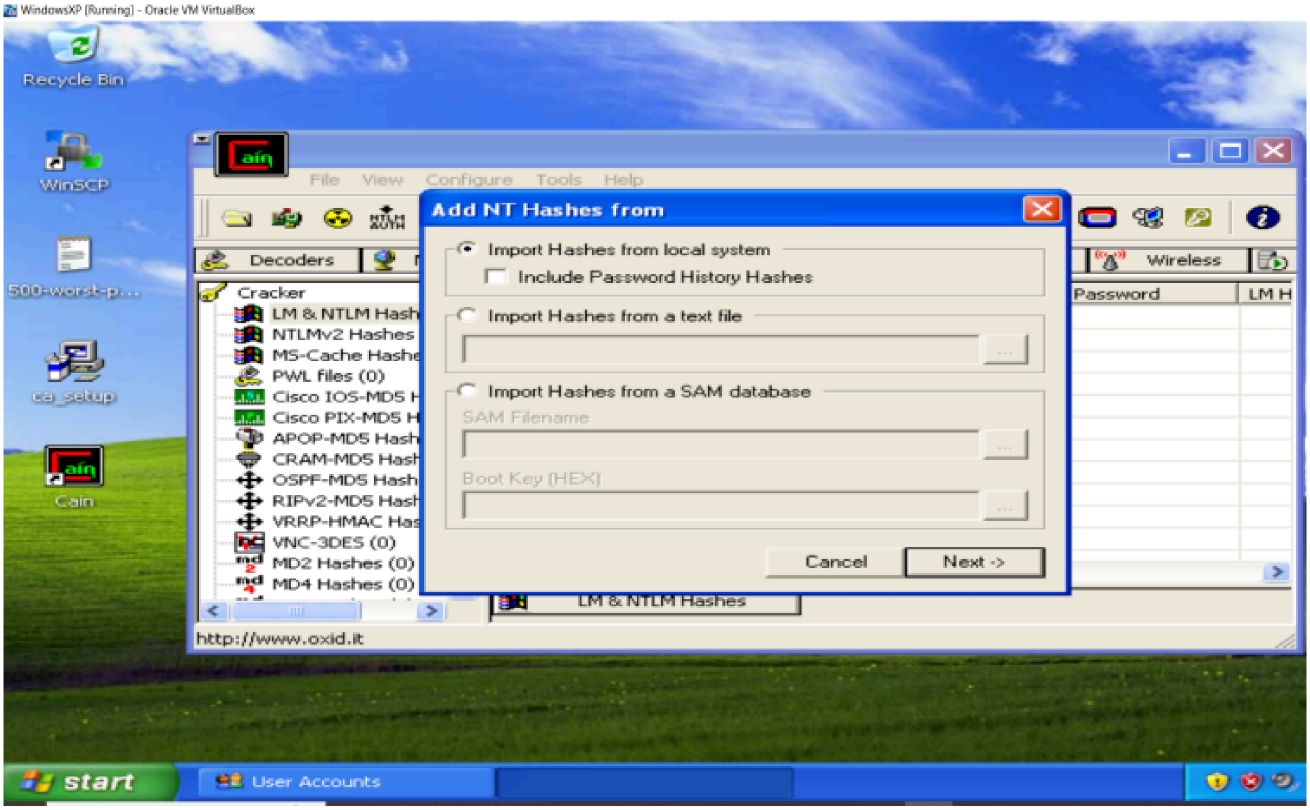
ITIS6200(PISP) PROJECT: 02

Amaan syed (asyed15@uncc.edu)

Task 1: Dictionary Attack

Open Cain & Abel, go under the “Cracker” tab, and select ‘LM & NTLM Hashes’ from the left column. (The two red squares in the figure below.) Now click on the plus sign from the taskbar to add NT hashes. Select ‘Import hashes from local system’ and click next



Add NT Hashes from

● Import Hashes from local system

□ Include Password History Hashes

○ Import Hashes from a text file

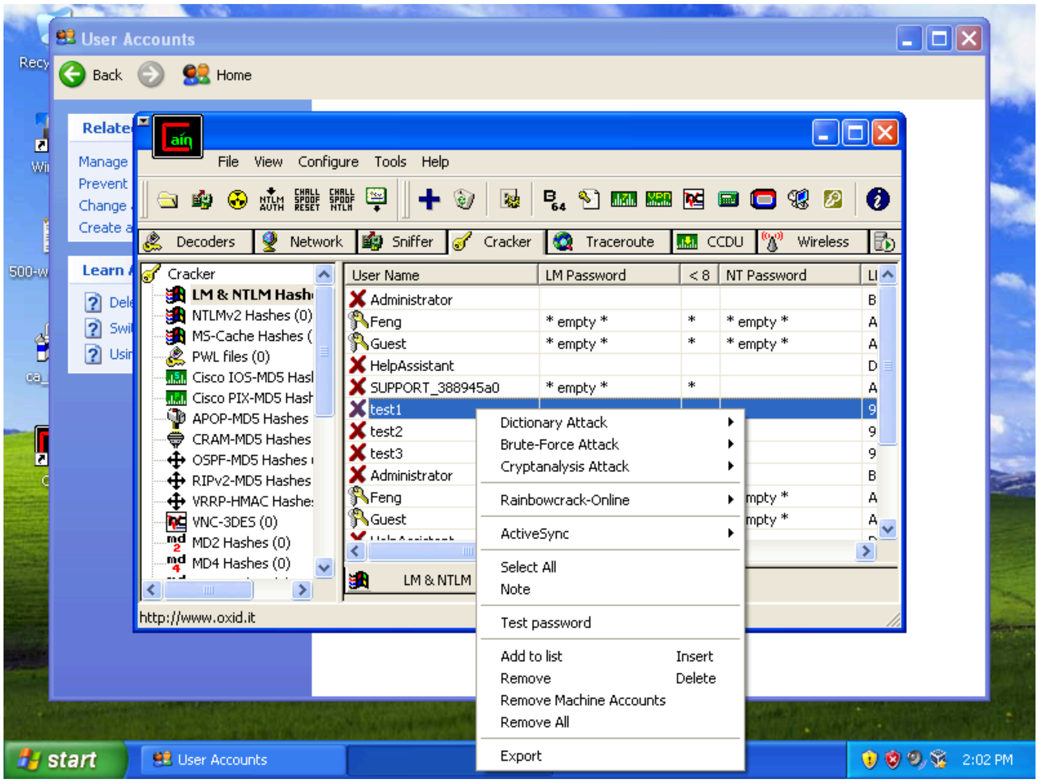
○ Import Hashes from a SAM database

SAM Filename

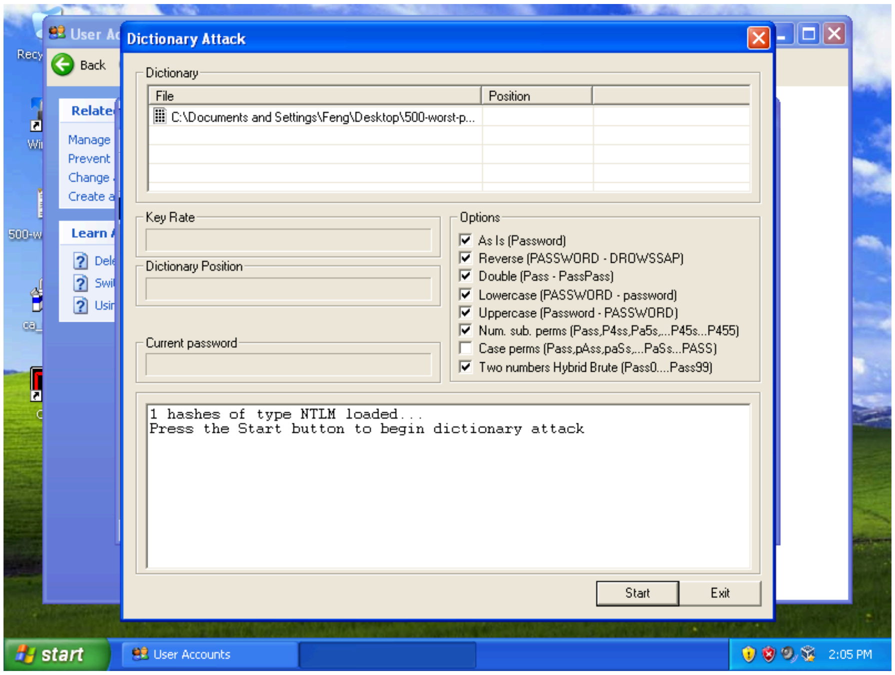
Boot Key (HEX)

Cancel | Next ->

Right click on ‘test1’ account and select ‘Dictionary Attack’. Select ‘NTLM hashes’ from the sub list. Now right click in the dictionary section and select ‘Add to list’ to add dictionaries. Navigate to the Desktop and select 500-worst-passwords.txt.



User Name	LM Password	< 8	NT Password	LI
Administrator				B
Feng	* empty *	*	* empty *	A
Guest	* empty *	*	* empty *	A
HelpAssistant				D
SUPPORT_388945a0	* empty *	*		A
test1				9
test2				9
test3				9
Administrator				B
Feng			mpty *	A
Guest			mpty *	A



Dictionary

File	Position
C:\Documents and Settings\Feng\Desktop\500-worst-p...	

Key Rate

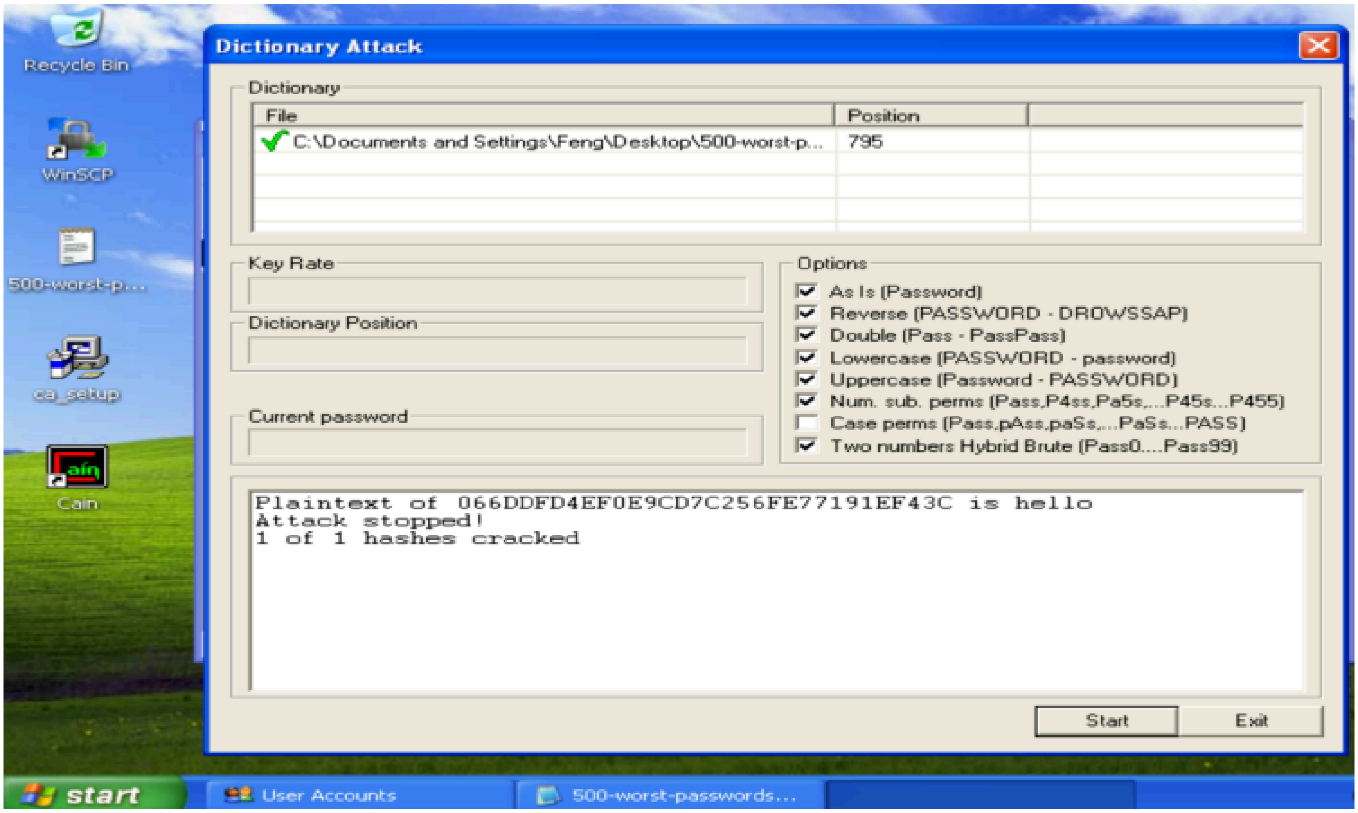
Dictionary Position

Current password

Options

☑ As Is (Password)
☑ Reverse (PASSWORD - DROWSSAP)
☑ Double (Pass - PassPass)
☑ Lowercase (PASSWORD - password)
☑ Uppercase (Password - PASSWORD)
☑ Num. sub. perms (Pass,P4ss,Pa5s,...P45s...P455)
□ Case perms (Pass,pAss,paSs,...PaSs...PASS)
☑ Two numbers Hybrid Brute (Pass0....Pass99)

```
1 hashes of type NTLM loaded...
Press the Start button to begin dictionary attack
```



Dictionary

File	Position	
✓ C:\Documents and Settings\Feng\Desktop\500-worst-p...	795	

Key Rate

Dictionary Position

Current password

Options

☑ As Is (Password)
☑ Reverse (PASSWORD - DROWSSAP)
☑ Double (Pass - PassPass)
☑ Lowercase (PASSWORD - password)
☑ Uppercase (Password - PASSWORD)
☑ Num. sub. perms (Pass,P4ss,Pa5s,...P45s...P455)
☐ Case perms (Pass,pAss,paSs,...PaSs...PASS)
☑ Two numbers Hybrid Brute (Pass0....Pass99)

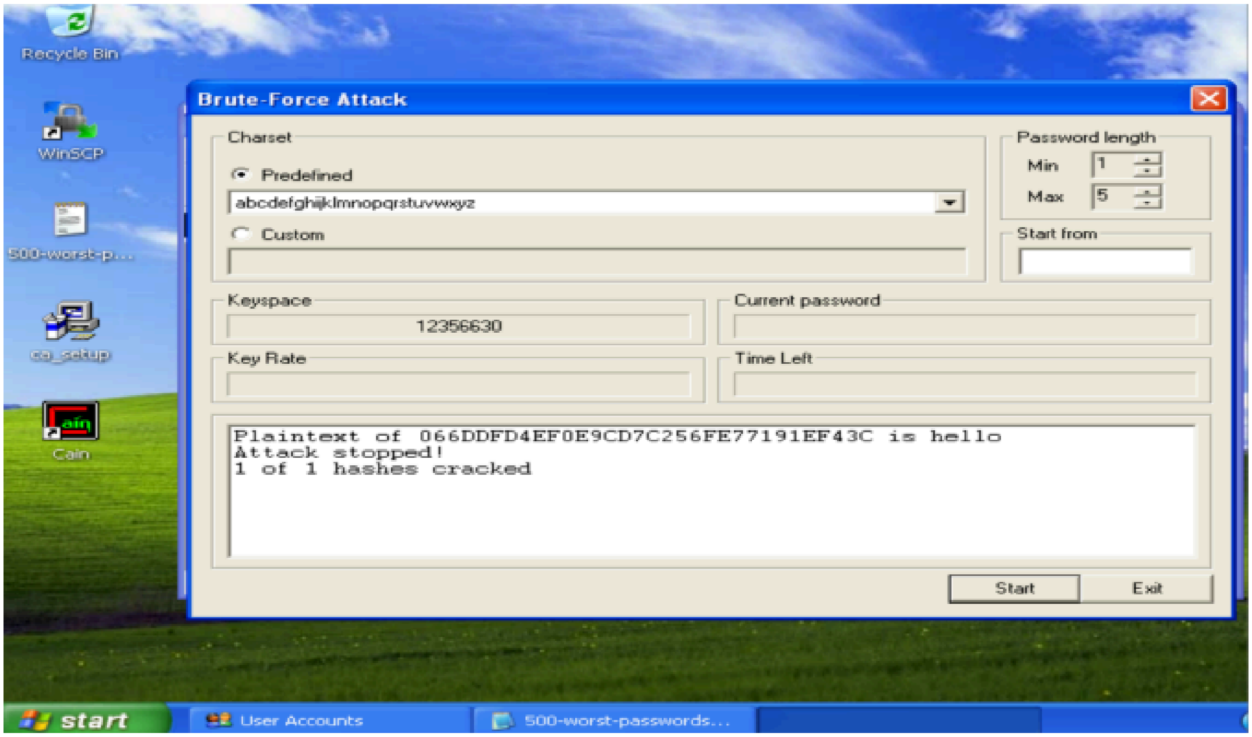
```
Plaintext of 066DDFD4EF0E9CD7C256FE77191EF43C is hello
Attack stopped!
1 of 1 hashes cracked
```

Task 2: Brute-Force Attack

You will need test1, test2, and test3 for this. Create for each account, one password from each type below (in the table). Note: follow exact specifications for the password as specified in the table below. Note your chosen password for each type in the table below. Right click on the appropriate account, for e.g., ‘test1’ and select ‘Brute-force Attack’. Select ‘NTLM hashes’ from the sub list. Make sure that you adjust the password length correspondingly. Otherwise, it will take days to finish. Adjust password length. Choose the appropriate charset. Perform the activity with the three passwords. Fill the following table with the details based on your activity.

	Password Description	Chosen Password	Charset	Time Taken
1	Lowercase letters only (length 5)	test1: hello test2: money test3: tiger	abcdefghijklmnopqrstuvwxyz	< 3 sec
2	Lowercase, uppercase letters and numbers from 0-9 (length 5)	test1: Test1 test2: Test2 test3: Test3	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789	4.49 min 3.54 min 4.27 min
3	Lowercase, uppercase letters, numbers from 0-9 and symbols (length 5)	test1: Tes@1 test2: Te$t2 test3: Tes!3	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789~!@#$%^&*()_+{}:”>?<,./;’[]=-\`	8.32 min 7.51min 7.51 min

Case 1 - test1



Charset
● Predefined
abcdefghijklmnopqrstuvwxyz
○ Custom

Password length
Min 1
Max 5

Start from

Keyspace 12356630

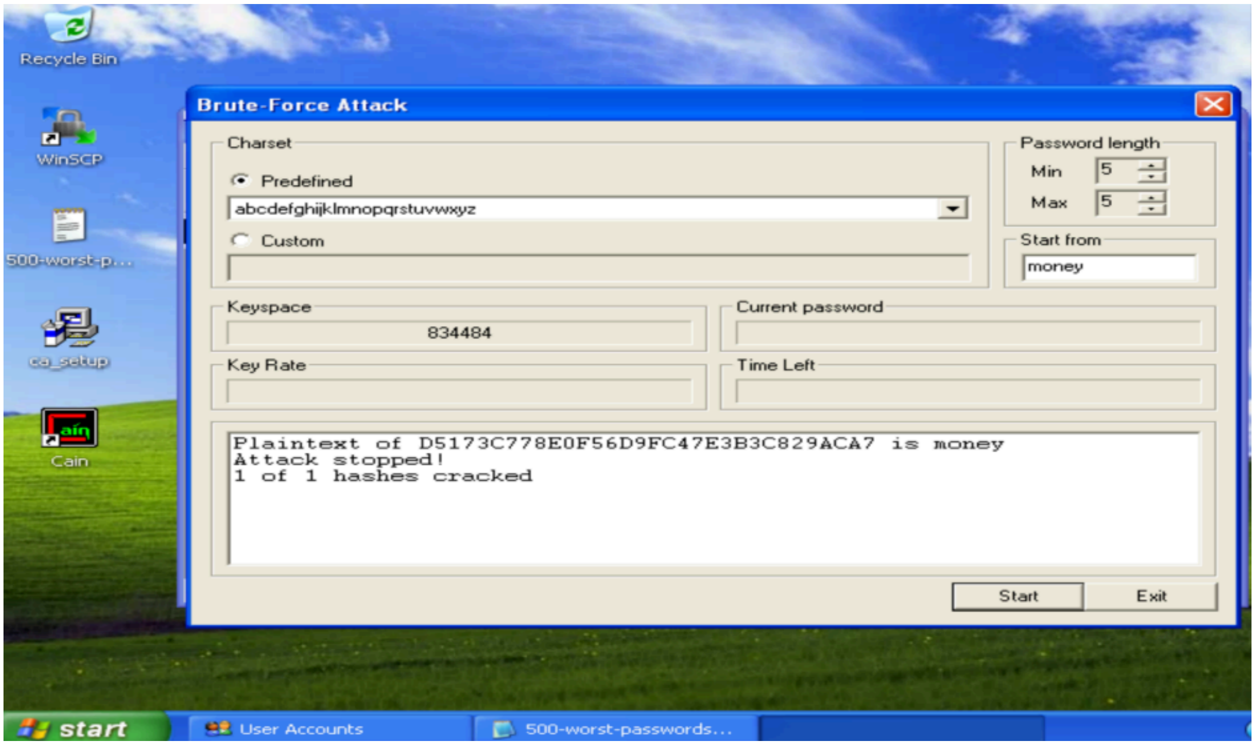
Current password

Key Rate

Time Left

```
Plaintext of 066DDFD4EF0E9CD7C256FE77191EF43C is hello
Attack stopped!
1 of 1 hashes cracked
```

Case 1 - test2



Charset
● Predefined
abcdefghijklmnopqrstuvwxyz
○ Custom

Password length
Min 5
Max 5

Start from money

Keyspace 834484

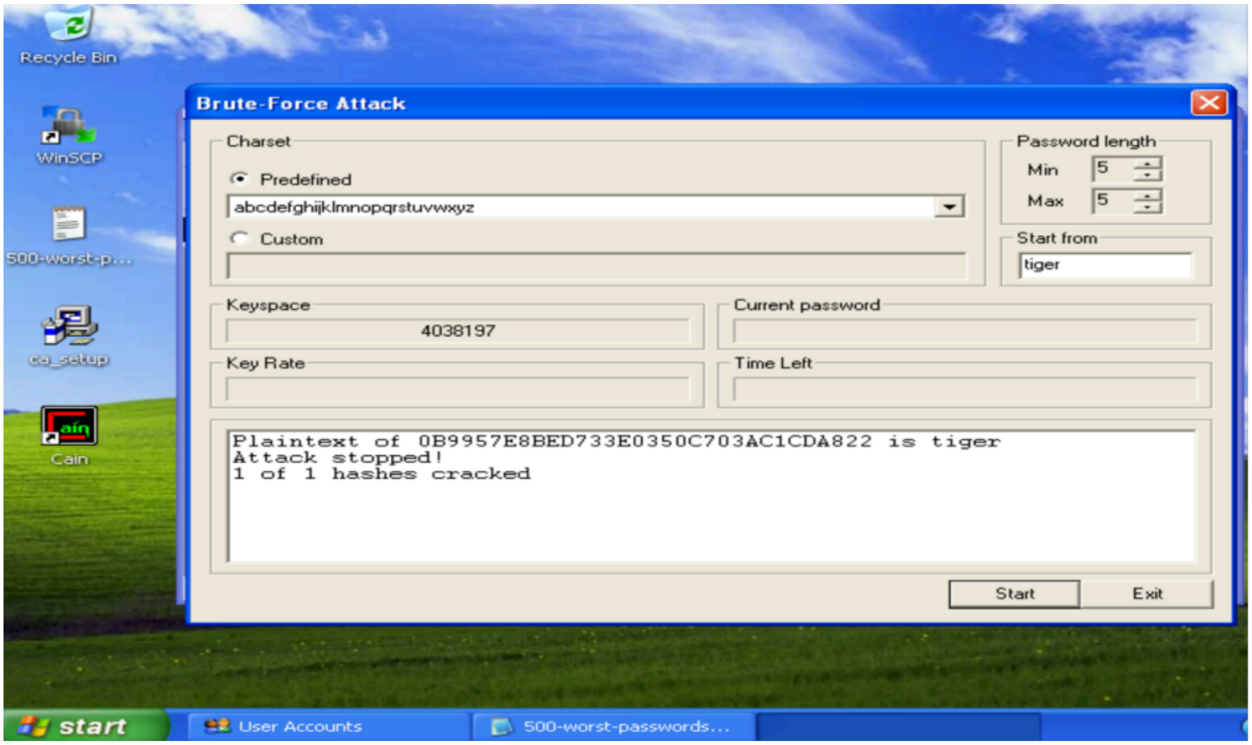
Current password

Key Rate

Time Left

```
Plaintext of D5173C778E0F56D9FC47E3B3C829ACA7 is money
Attack stopped!
1 of 1 hashes cracked
```

Case 1 - test3



Brute-Force Attack

Charset

● Predefined

abcdefghijklmnopqrstuvwxyz

○ Custom

Password length

Min 5

Max 5

Start from

tiger

Keyspace

4038197

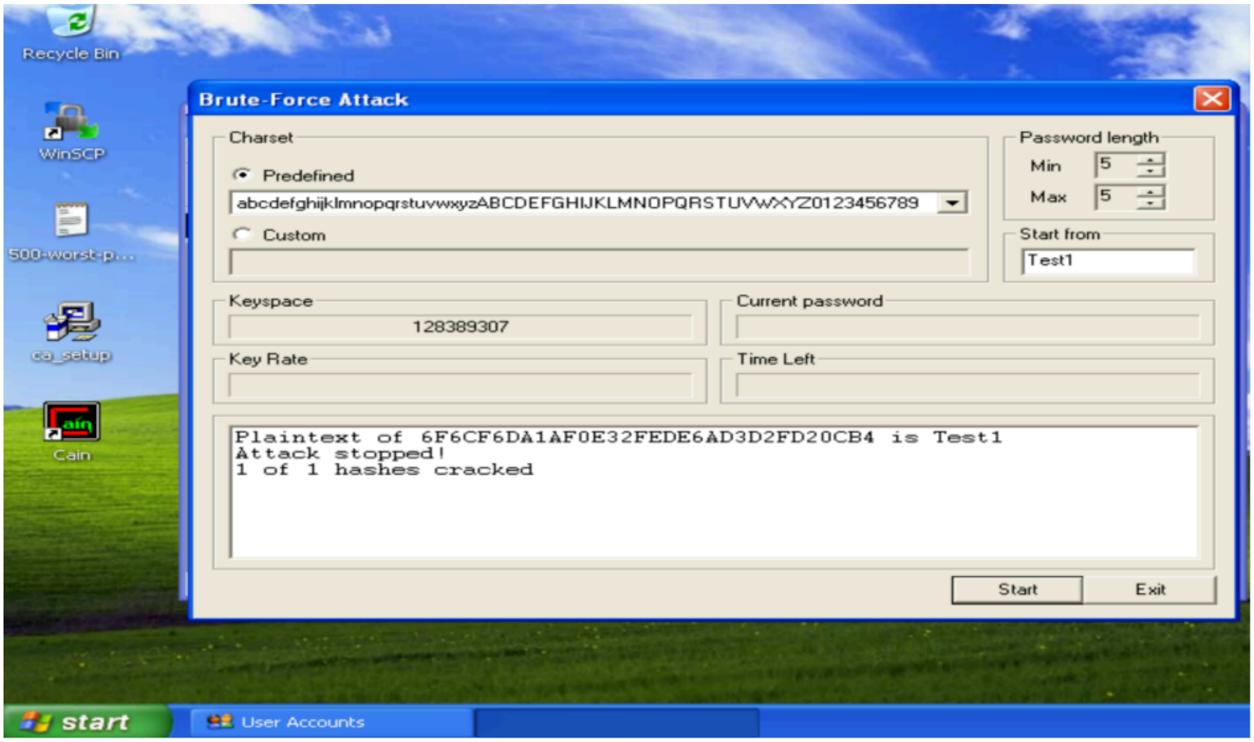
Current password

Key Rate

Time Left

```
Plaintext of 0B9957E8BED733E0350C703AC1CDA822 is tiger
Attack stopped!
1 of 1 hashes cracked
```

Case 2 - test1



Brute-Force Attack

Charset

● Predefined

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

○ Custom

Password length: Min 5, Max 5

Start from: Test1

Keyspace: 128389307

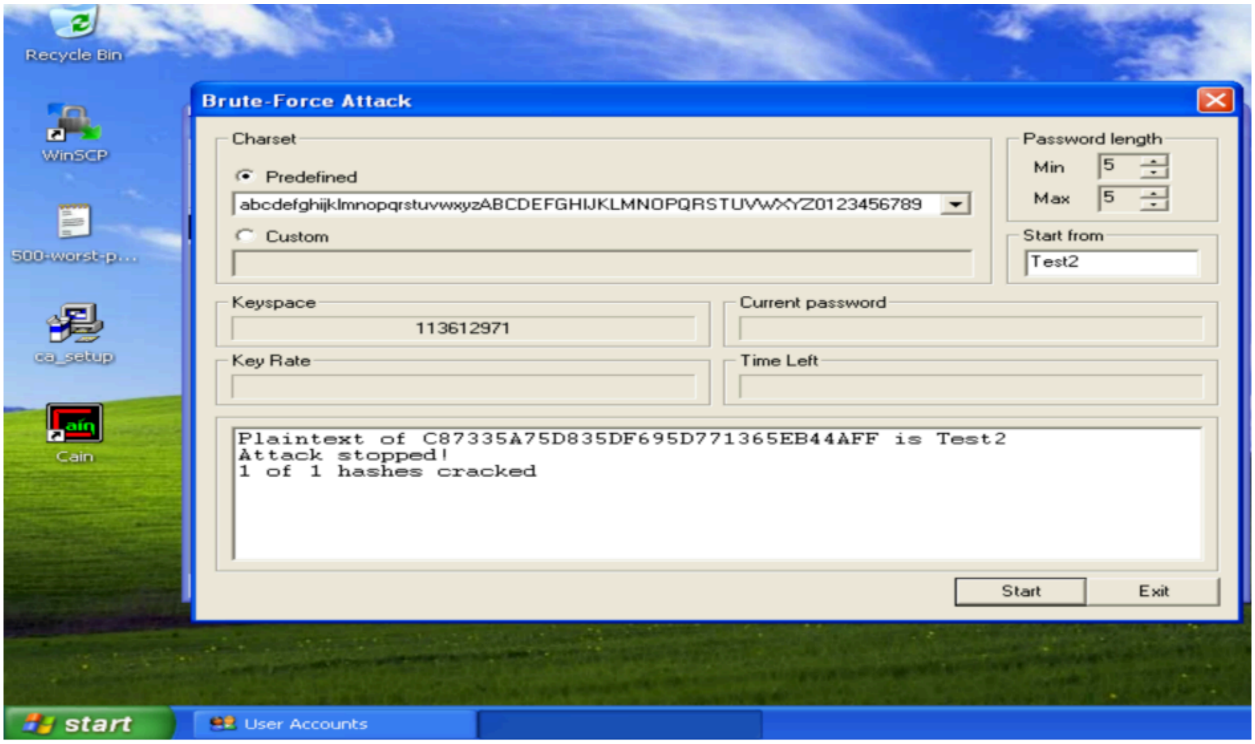
Current password:

Key Rate:

Time Left:

```
Plaintext of 6F6CF6DA1AF0E32FEDE6AD3D2FD20CB4 is Test1
Attack stopped!
1 of 1 hashes cracked
```

Case 2 - test2



Brute-Force Attack

Charset

● Predefined

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

○ Custom

Password length: Min 5, Max 5

Start from: Test2

Keyspace: 113612971

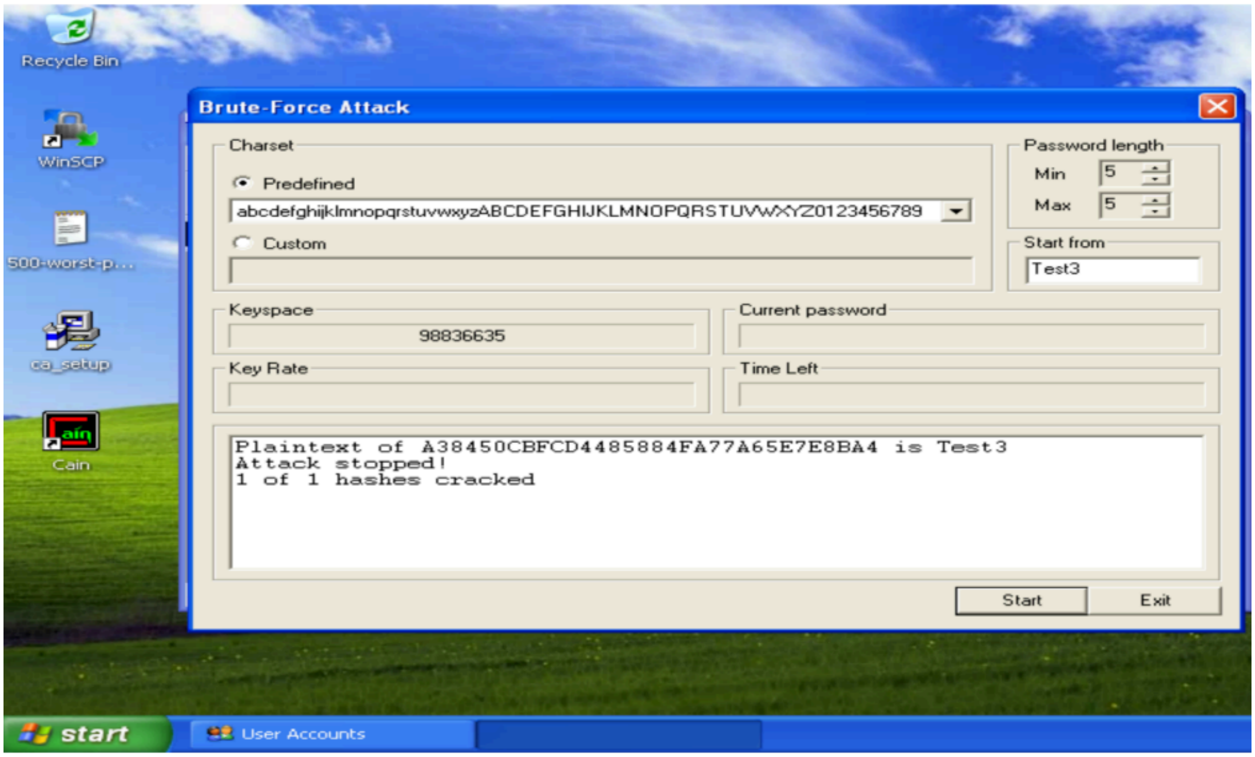
Current password:

Key Rate:

Time Left:

```
Plaintext of C87335A75D835DF695D771365EB44AFF is Test2
Attack stopped!
1 of 1 hashes cracked
```

Case 2 - test3



Brute-Force Attack

Charset

● Predefined

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

○ Custom

Password length

Min 5

Max 5

Start from

Test3

Keyspace

98836635

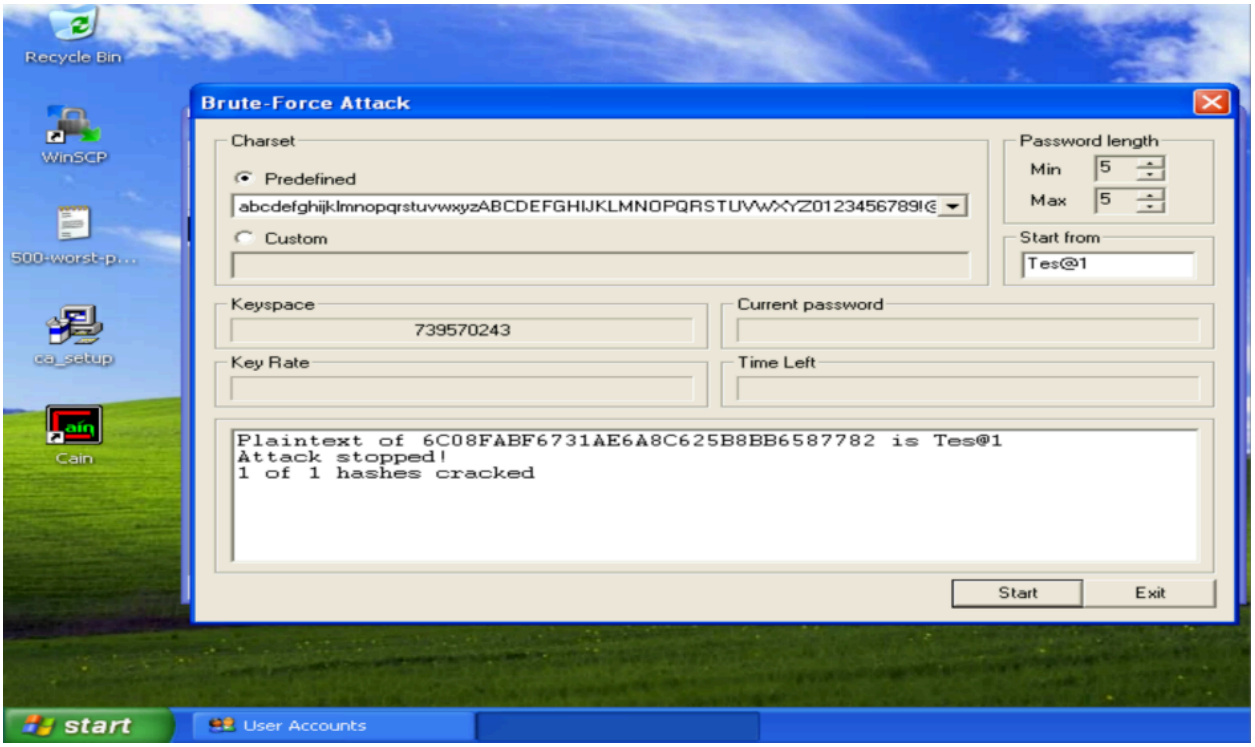
Current password

Key Rate

Time Left

```
Plaintext of A38450CBFCD4485884FA77A65E7E8BA4 is Test3
Attack stopped!
1 of 1 hashes cracked
```

Case 3 - test1



Brute-Force Attack

Charset

● Predefined

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@

○ Custom

Password length

Min 5

Max 5

Start from

Tes@1

Keyspace

739570243

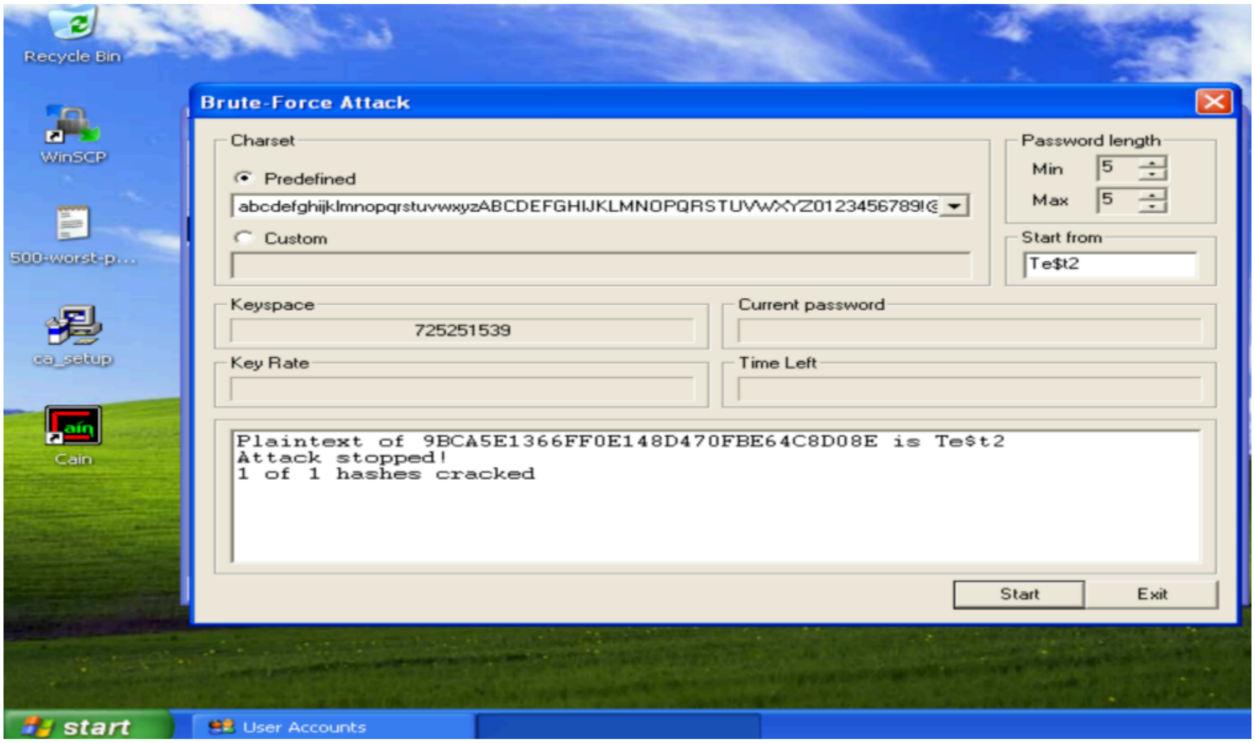
Current password

Key Rate

Time Left

```
Plaintext of 6C08FABF6731AE6A8C625B8BB6587782 is Tes@1
Attack stopped!
1 of 1 hashes cracked
```

Case 3 - test2



Charset

● Predefined

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@

○ Custom

Password length

Min 5

Max 5

Start from: Te$t2

Keyspace: 725251539

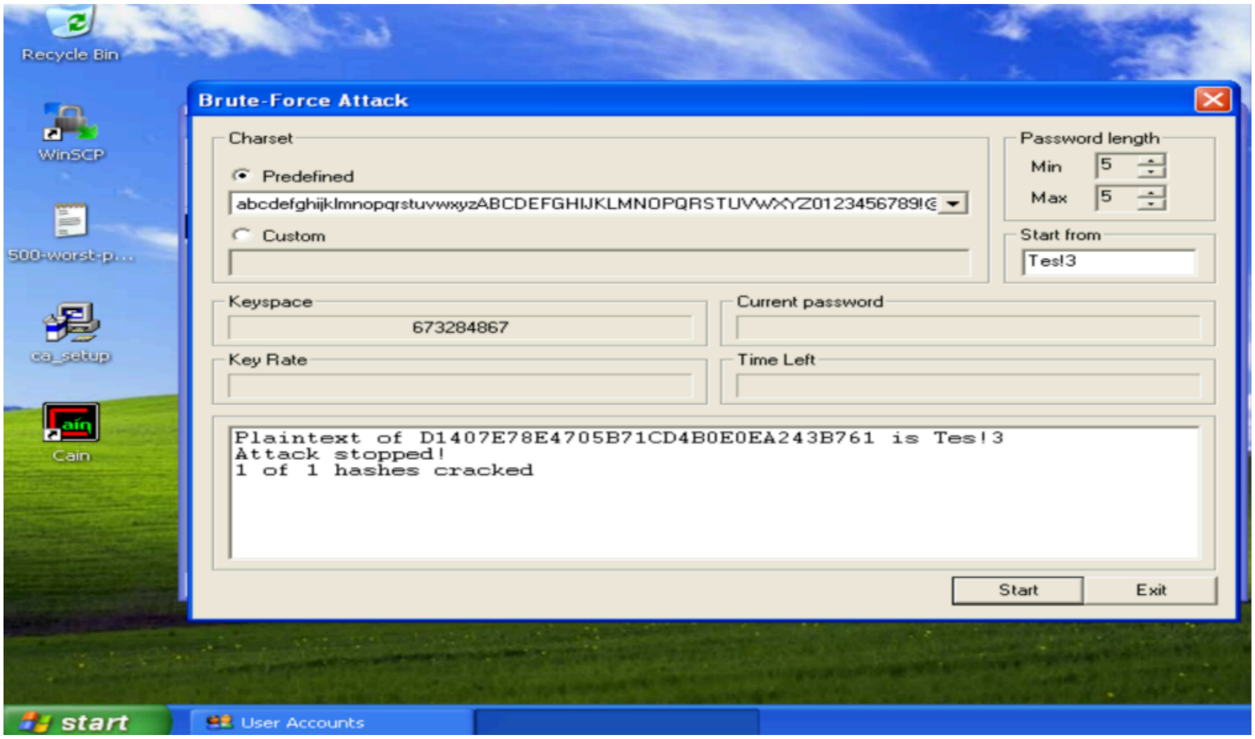
Current password:

Key Rate:

Time Left:

```
Plaintext of 9BCA5E1366FF0E148D470FBE64C8D08E is Te$t2
Attack stopped!
1 of 1 hashes cracked
```

Case 3 - test3



Charset

● Predefined

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@

○ Custom

Password length

Min 5

Max 5

Start from: Tes!3

Keyspace: 673284867

Current password:

Key Rate:

Time Left:

```
Plaintext of D1407E78E4705B71CD4B0E0EA243B761 is Tes!3
Attack stopped!
1 of 1 hashes cracked
```

Answer the question: 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 rcover a password using a tool like Cain & Abel, you would not know what the password was, only what the password space was!

The Cain & Abel application predicts that the first case will take less than a second, second case will take around 2.5 minutes, and third case will take around 5-6 minutes based on how my system is configured. A password with a highly complex charset took longer to crack, as can be seen in the table with custom passwords. It's possible that the application tried up to 670,000,000 different passwords in a second. Nevertheless, the program will run around the same length of time for each charset due to the brute force nature of this technique. The program must try each character combination inside the given bounds in order to find a hash value and be able to recognize a match. A shorter password would be easier to crack than a longer one, considering the power of computers and their ability to try many different passwords in a single second. Therefore, it can take the same amount of time to crack the passwords "aaa" and "aBC" even if the charset is greater, such ABCDEFGHIKLMNOPQRSTUVWXYZ. When I tried to verify this, it took my machine less than a second for aaa and aBC to break. Therefore, it will be harder to crack and take longer if we only know the password space and not its length. Since more charset combinations are possible with longer passwords, the likelihood of finding the right one quickly decreases. In addition, our passwords become harder and take longer to figure out when we add symbols to them. Consequently, having a wider charset and a longer length is always recommended. Because of this, most contemporary websites recommend utilizing a minimum of 8 characters, including digits, symbols, lowercase and uppercase letters, and numbers