

Forum:[Hack Forums](#)

Thread: [\[C++ SOURCE CODE\] Brute force code](#)

malo1992 - June 28, 2011, 12:55 pm

Hey fellas!

This is an example for a simple brute force algorithm in C++ ?

I commented it with my weird english skills as good as I could, and I hope that the C++ beginners, to whom this is addressed, are able to understand what I wanted to say. Maybe it's even something like a lil tutorial on brute forcers ?

As you see, it's open source, so feel free to use it for whatever you like(transferring it to an md5hash "decrypter", for example, shouldn't be too hard...).

The syntax ain't perfect, I'm no C++ pro ? Anyway, I like the "mise en abyme" procedure in brute force codes, and although there are way faster codes out there(multithreading etc... going to learn this shit ^^), this one gives a cute little insight into the functionality of brute forcers, I think... ?

Blablabla, here it is:

```
Code // *** SIMPLE C++ BRUTE FORCE EXAMPLE CODE ***
```

```
// written by malo1992
```

```
//
```

```
// Completely open source, do what you want with it.
```

```
// Have fun!
```

```
//
```

```
// Germany, June 2011
```

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
/* CHARACTER LISTS */
```

```
char lower[27]={ 'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','\0'};
```

```
char upper[27]={ 'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z','\0'};
```

```
char numbs[11]={ '0','1','2','3','4','5','6','7','8','9','\0'};
```

```
char symbols[30]={ '!', '?', ':', ';', ',', '-', '_', '#', '$', '%', '&', "'", '"', '(', ')', '[', ']', '@', '=', '<', '>', '|', '^', '\0'};
```

```
char alphabet[95]; // This will contain the brute force alphabet
```

```
/* VARIABLES */
```

```
string password = "";
```

```
// We will brute-force this
```

```
string bruteword = "";
```

```
// The string we want to be =password
```

```
string pw = "";
```

```
// Just a help variable; contains the matching bruteword
```

```
string letters = "";
```

```
// Letters that will be added to the current character(alphabet[X]) in order to
```

```
create the bruteword
```

```
bool pw_found = false;
```

```
// Password ain't found yet.
```

```
int maxChars = 15;
```

```
// Maximum length of the bruteword (can be changed by the user)
```

```
int minChars = 1;
```

```
// Minimum length o. t. BW ( " )
```

```
int alph_length = 0;
```

```
// Length of the char alphabet
```

```
/* VOIDS */
```

```

void asdf();    // Nonsense, but funny ;P
void createAlph();    // Creates brute alphabet
void check(string bruteword);    // Checks if bruteword = password
void brute (int charsInBW, int instance, string letters);    // BRUTE BRUTE BRUTE

/* PROGRAM CODE */
int main()
{
    // START ROUTINE //
    cout << endl;
    cout << "Enter a password to attack: ";
    cin >> password;

    cout << endl;
    cout << "Minimum length of the bruteword(default 1): ";
    cin >> minChars;
    cout << "Maximum length of the bruteword(default 15): ";
    cin >> maxChars;

    // CREATING THE ALPHABET ARRAY //
    createAlph();

    // Right before we start...
    cout << endl;
    cout << "***" << endl;
    cout << "Password string that will be attacked: " << password << "" << endl;
    alph_length = strlen(alphabet);
    cout << "Brute force alphabet: " << alphabet << endl;
    cout << "Characters: " << alph_length << endl;
    asdf();    // Some useless extra feature, I was bored
    cout << "***" << endl << endl;
    cout << "Press ENTER to start brute forcing..." << endl;
    cin.get();cin.get();    // Forces the application to wait for user input

    // LET'S GO! //
    for( int i=minChars; i<=maxChars ; i++)    // Bruteword needs [i] characters.
    {    // ( e.g. the brutewords with 1 character don't match? Then use 2 characters etc etc ... )
        brute(i,0,letters);
    }    // Read the comments in void brute() for further information.

    // FOR()-LOOP STOPPED, PASSWORD FOUND OR NOT?
    if ( pw_found )
    { // match! :D
        cout << "Bruteforcing stopped." << endl << endl;
        cout << "PASSWORD MATCH!!! -> " << pw << endl << endl;
        cout << "Press ESCAPE to close the application...";
        cin.get();    // Forces the application to wait for user input
    }
    else
    { // no match ... :(
        cout << "Bruteforcing stopped." << endl << endl;
        cout << "*** No matching password has been found." << endl << endl;
        cout << "Press ESCAPE to close the application...";
    }
}

```

```

    cin.get();    // Forces the application to wait for user input
}

return 0;
}

void brute(int charsInBW, int instance, string letters)
{
    for(int b=0; b<alph_length ; b++)    // There are [alph_length] characters in char array
    alphabet( e.g.'a'=position 0 and 'z'=position 25).
    {    // Now, alphabet[b] can display every single character in the chosen alphabet.
        if( !pw_found )    // Password still unknown?
        {
            if ( instance+1 != charsInBW )    // How many characters needs the bruteword(this is told
            by charsInBW)? If >1, then use this ( comparable to "picture in picture" or "mise en abyme" ) ...
            {    // It creates [charsInBW-1] running brute voids(and each one of them defines exactly 1
            character in the bruteword) that run within a brute void that already started(e.g. the first instance
            started by int main).
                brute(charsInBW, instance+1, alphabet[b]+letters);    // "instance+1": One needed
            brute-instance less, because a new one ...
            }    // ... with alphabet[b]+(letters of other instances) in the "letters"-string has been
            started.
            else check(alphabet[b]+letters);    // This is the last instance which checks ...
        }    // ... if (current character)+(letters of all other instances) is =password.
        else    // A matching password has been found!
        {
            b=alph_length;    // This stops the for()-loops in void brute...
            maxChars = pw.size();    // ... and in int main().
        }
    }
}

void check(string bruteword)
{
    cout << bruteword << endl;
    if ( bruteword == password ) // Bruteword matches?
    {
        pw_found = true;    // Stops the creation of instances in void brute.
        pw = bruteword;    // Writes a global string for displaying the matching bruteword in int main.
    }
}

void createAlph()    // Should be self explanatory ...
{
    cout << endl;
    string answ = "";    // This will store the user's answers during alphabet construction

    cout << "Attack with lower case letters(Y/N)? ";
    cin >> answ;
    if ( answ == "y" || answ == "Y" ) strcat_s(alphabet,sizeof(alphabet),lower,sizeof(lower));
    else if ( answ == "n" || answ == "N" ) cout << "**** No lower case letters." << endl;
    else cout << "**** Wrong input." << endl << "Lower case letters have NOT been added to the
    brute force alphabet." << endl;
}

```

```

    answ = "";    // Reset for possible rewrite

    cout << "Attack with numbers(Y/N)? ";
    cin >> answ;
    if ( answ == "y" || answ == "Y" ) strcat_s(alphabet,sizeof(alphabet),numbs,sizeof(numbs));
    else if ( answ == "n" || answ == "N" ) cout << "**** No numbers." << endl;
    else cout << "**** Wrong input." << endl << "Numbers have NOT been added to the brute force
alphabet." << endl;
    answ = "";

    cout << "Attack with upper case letters(Y/N)? ";
    cin >> answ;
    if ( answ == "y" || answ == "Y" ) strcat_s(alphabet,sizeof(alphabet),upper,sizeof(upper));
    else if ( answ == "n" || answ == "N" ) cout << "**** No upper case letters." << endl;
    else cout << "**** Wrong input." << endl << "Upper case letters have NOT been added to the
brute force alphabet." << endl;
    answ = "";

    cout << "Attack with symbols(Y/N)? ";
    cin >> answ;
    if ( answ == "y" || answ == "Y" ) strcat_s(alphabet,sizeof(alphabet),symls,sizeof(symls));
    else if ( answ == "n" || answ == "N" ) cout << "**** No symbols." << endl;
    else cout << "**** Wrong input." << endl << "Symbols have NOT been added to the brute force
alphabet." << endl;
    answ = "";
}

void asdf()
{
    string first = "";
    string last = "";
    char cFirst=alphabet[0];
    char cLast=alphabet[alph_length-1];

    for(int i=1; i<=minChars;i++)
    {
        first=first+cFirst;
    }

    for(int i=1; i<=maxChars;i++)
    {
        last=last+cLast;
    }

    cout << "Going to attack from " << first << " to " << last << "!" << endl;
}
//
// END OF FILE

```

btw, I worked with Visual Studio C++ 2010 Express on a windows XP 32bit system

EDIT: why does the forum cut every textline that contains more than 2 equal letters? ? wtf?

EDIT2: fixed some stupid grammar mistakes and added a pw_found-request at the end of in main(),

behind the for()-loop

EDIT 3: THE NEW MULTITHREADED CODE

Okay dudes, now THIS is fucking awesome...

You may know that the key of a good bruteforcer is its SPEED.

I google'd a lot on multithreading and found this: <http://www.boost.org/> ?

This really drives me crazy. Some godlike nerd called Anthony Williams created a multithread-library(simply called "thread") for this great code collection. I was immediately like: ? and ? !

It was easy to install and apply to my project.

AND NOW COMES THE CLUE!

With only two parallel-running bruteforce engines(this is just the beginning... if you want, you can use 123123123 billion, i think ?), my new code rapes the string "asdf" in LESS THAN A SECOND!!! I'm still so fucking shocked. To compare, my old code took 18 seconds(!!!!) to bust this fuckin simple 4 character string ?

Accordingly, my new multithreaded brute force attack is about 1800 % faster :D

WHAT ABOUT THAT?!?!?! MWAHAHAHA

You'll need the boost-library thing to compile the source code, and its damn big, thus I uploaded the prog right here(I swear that it contains NO MALWARE! would be <3ly if someone could proof this with an virustotal scan report.)

<http://www.mediafire.com/?9wcv8xb0zk5rn27>

heres the code, I didnt give a fuck on new // comments this time. was too excited. furthermore, I had to cut the bruteword outputs out, cuz cout sux on multithreaded applications and I was too lazy for printf'ing this(i don't even know if printf works correctly).... who cares. you'll still love it.

MESSAGE WAS TOO LONG; LOOK <https://hackforums.net/showthread.php?tid=1426822> TOO SEE THE NEW SOURCECODE!!!

Judas - June 28, 2011, 1:09 pm

Good job! :) Will rep whenever I become l33t ^_^

jthehacker - June 28, 2011, 5:39 pm

nice brute forcer i love it

malo1992 - June 29, 2011, 2:40 am

YEHA!! ?

Thanks a lot mates, your feedback makes me happy ? !

Navigator - June 29, 2011, 6:02 am

Here's my version that I coded a while back: <https://hackforums.net/showthread.php?tid=1155672>.

I really think that creating the character set should be 100% dynamic since the character codes are consecutive anyway.

malo1992 - June 29, 2011, 6:24 am

Really nice one, Navigator, amazingly fast! O_O

I'm just trying to make my code checking up to 12 brutewords at once... I'll update my first post if it works well :)

EDIT, half an hour later: AIIII this is hard xD may take a while.
but i wont give up! =D

blackbird0 - January 1, 2013, 7:22 am

hi guys,

can anyone make a fast 3des bruteforcer??

GanGBanGeR - June 29, 2014, 3:31 am

and how do i use the bruteforce like this? <https://www.youtube.com/watch?v=EpRIhAsIFVo>

My only idea was to simulate a keyboard and let it press (keys ip.ki.wVk = 0x41)

Niely - June 29, 2014, 3:40 am

Nice one OP! :)

I learnt something from it ! :)

But why two times cin.get()?

You have:

Code cin.get();cin.get(); somewhere.

0xDEAD10CC - June 29, 2014, 3:54 pm

This is horrible code, and wrong. You don't understand null terminators, and getline() should be used for C++ strings. This won't really do anything optimal either.

FreezeComplex - March 28, 2015, 10:12 pm

Thanks for the tutorial. I find stuff like this very interesting.

TomKeen - March 29, 2015, 4:32 am

Thanks for the share , Hope you don't mind if I modify the code a bit...

Meeting - March 29, 2015, 7:06 am

Not the best code ever , on the other hand that could help some people , thanks for sharing.

CreativeHD - March 29, 2015, 7:14 am

Hey, why don't you use string, instead of all those char array ?

I'm just wondering if char arrays are faster than strings ?

And, i don't think you need to initialize tour strings,

You did like that : string bruteword = "";

But, i think this is a better way: string bruteword; //Perhaps it don't change anything, but could make your code prettier :)

8086 - March 29, 2015, 8:14 am

Do you all realise the date the thread was created?

CreativeHD - March 29, 2015, 8:23 am

Houp, hep, was a long time i go, didn't saw it, thx :)

But, i will publish this code in a new thread to make it better and faster :)

Of course, i will mention its creator ;) (malo1992)

sharkyz - March 29, 2015, 11:38 am

Will have to look through the code. But seems people are finding it interesting.

Wozniak - March 29, 2015, 3:33 pm

My personal opinion is to never use boost. I hate it.

On the other hand, I have never used any libs that dont come with VS.

I've never downloaded anything to use in my programs.

I feel like its cheating. I mean if you want to use threading, then spare the user of the 3 boost dll's that they will have to deal with (in the bin) and just learn to thread it yourself.

But again, completely my own opinion.

(Mar 29, 2015 04:32 AM)TomKeen Wrote: [\[CITE\]](#) Thanks for the share , Hope you don't mind if I modify the code a bit...

Omg. I <3 your name.

Assuming its a reference to my fav show The Blacklist

Awesome!

0xDEAD10CC - March 29, 2015, 4:15 pm

(Mar 29, 2015 03:33 PM)_Wozniak_ Wrote: [\[CITE\]](#) My personal opinion is to never use boost. I hate it.

On the other hand, I have never used any libs that dont come with VS.

I've never downloaded anything to use in my programs.

I feel like its cheating. I mean if you want to use threading, then spare the user of the 3 boost dll's that they will have to deal with (in the bin) and just learn to thread it yourself.

But again, completely my own opinion.

Omg. I <3 your name.

Assuming its a reference to my fav show The Blacklist

Awesome!

How can you have an opinion about BOOST if you don't use third party libraries? If you want to waste your time writing something that may be better written in some third party library then go ahead.. There's nothing cheating about avoiding having to reinvent the wheel, and unless you are really concerned about preserving the output binary size, there's no need to be concerned about using libraries, whether you compile with their source, or link to them statically or dynamically in the case of a shared library.

malo1992 - April 13, 2015, 5:30 am

(Mar 29, 2015 04:32 AM)TomKeen Wrote: [\[CITE\]](#) Thanks for the share , Hope you don't mind if I modify the code a bit...Of course not.

You won't even have to mention my "name" anywhere, or whatever. Just do your thing, mate

(Mar 29, 2015 07:06 AM)Meeting Wrote: [\[CITE\]](#) Not the best code everOf course not.

(Mar 29, 2015 07:14 AM)CreativeHD Wrote: [\[CITE\]](#) I'm just wondering if char arrays are faster than strings ?Of course not.

(Mar 29, 2015 07:14 AM)CreativeHD Wrote: [\[CITE\]](#) Hey, why don't you use string, instead of all those char array ? [...]

And, i don't think you need to initialize tour strings,

You did like that : string bruteword = "";

But, i think this is a better way: string bruteword;You're absolutely right. Today, I would probably do it right that way. Mostly.

(Mar 29, 2015 08:23 AM)CreativeHD Wrote: [\[CITE\]](#) But, i will publish this code in a new thread to make it better and faster :) Nice, I am curious! PM me on release if you want, so I can check it out ? I still may learn from it, for sure ?

(Jun 29, 2014 03:40 AM)Niely Wrote: [\[CITE\]](#) But why two times cin.get()? Well, that one is a bit embarrassing ? used it to force ESC as keyboard input. But it worked so I didn't care further, anyway it's probably preferable to have higher standards than this ?

(Mar 29, 2015 03:33 PM)_Wozniak_ Wrote: [\[CITE\]](#) never use boost. I hate it.

I have never used any libs that dont come with VS.
I've never downloaded anything to use in my programs.
I feel like its cheating.

completely my own opinion. Last statement gets the point.
But seriously, "cheating"?? Oh boy. You then would have to rewrite iostream, stdlib and stuff as well in order to avoid plagiarizing and being a legit coder. I mean, where's the ethical difference between including boost or deflibs, actually?

Anyway, under those circumstances you mention, I'd choose cheating without hesitation. ? _____

(Jun 28, 2011 12:55 PM)malo1992 Wrote: [\[CITE\]](#) MESSAGE WAS TOO LONG; LOOK <https://hackforums.net/showthread.php?tid=1426822> TOO SEE THE NEW SOURCECODE!!! Nah, don't look. It only re-directs to this thread, due to me being stupid or whatever.
Here's the actual link: <https://hackforums.net/showthread.php?tid=1430580>

Thanks for any other feedback so far! ?

Cheers

malo1992 - April 14, 2015, 10:04 am

(Apr 13, 2015 05:30 AM)malo1992 Wrote: [\[CITE\]](#) Quote: why two times cin.get();? used it to force ESC as keyboard input. But it worked so I didn't care further, anyway it's probably preferable to have higher standards than this ? Completion: I didn't clear my cin steam, therefore the 1st cin.get(); have not needed an "ENTER"/newline ("ESC" was a typo, ofc), as it was supposed to (program pause, you know). But the second one then did what I wanted - sorry, I know this code was disgusting haha
cin.clear(), *.sync(), and preferably *.ignore() instead of *.get() solves this, as you may already know.

Godacity - April 16, 2015, 10:49 am

Good info everyone! Will take these things into consideration!