

Forum:[Hack Forums](#)

Thread: [C++ Simple Algorithm To Crack CRC32 Hashes](#)

Zer0Memo0ry - December 29, 2016, 4:58 am

This algorithm bruteforces the entire keyspace of input strings, hashing them with CRC32 algorithm and comparing it to the target hash. If target hash matches the created hash, a collision is logged.

Now this algorithm is really just a proof of concept, and won't be useful in most cases. But it can be useful to nail down possible password hashes, or something like that.

Project on Github: <https://github.com/Zer0Memory/CrackCRC32>

```
Code #include <iostream>
#include <string>
```

```
#include <Windows.h>
```

```
#include "CRC.h"
```

```
using namespace std;
```

```
//
string numerals = "0123456789";
string alphabetLower = "abcdefghijklmnopqrstuvwxyz";
string alphabetUpper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
string specialAnsi = "!\"#$%&'()*=?^`*';:_,.";
```

```
void CrackHash(string source, unsigned int hash, string seed, size_t length)
{
    // this is used to measure time
    static const int startTick = GetTickCount64();

    static int iterations; // store the count of iterations
    static int collisions; // store the count of collisions/hash

    iterations++; // increment iteration count each time function is executed.

    if (length == 0) // if the length is reached
    {
        if (CRC32(seed.c_str(), seed.length()) == hash) // if the hash created from seed matches the
input
        {
            // calculate the time the operation took
            double endTick = GetTickCount64() - startTick;
            collisions++; // increment collisions

            cout << "Collision #" << collisions << ", Source: " << seed << ", Hash: 0x" <<
StrCRC32(seed)
                << ", Iterations: " << iterations << endl << "Ticks: " << endTick << ", Seconds: "
```

```

        << endTick / 1000 << ", Minutes: " << endTick / 1000 / 60 << ", Hours: " << endTick /
1000 / 60 / 60;
        cout << endl << endl;
    }
    return;
}

```

```

// loop through the provided characters
for (size_t i = 0; i < source.length(); i++)
{
    string appended = seed + source[i];

    CrackHash(source, hash, appended, length - 1);
}
}

```

```

int main()
{
    // our target hash is 'test' hashed
    unsigned int targetHash = 0xd87f7e0c;
    //cout << "CRC32 hash of 'test' is: " << StrCRC32("test") << endl;
    //cin.get();

    // now let's try to crack the hash of test
    while (1)
    {
        // the length of string that the algorithm starts from
        static unsigned int strlen = 1;
        CrackHash(alphabetLower + alphabetUpper + numerals + specialAnsi, targetHash, "", strlen);
        strlen++;
    }
}

```