

Forum:[Hack Forums](#)

Thread: [\[SOURCE\] Data to bitmap and back using pixel color values](#)

hgdfhsdgdgdfh - March 27, 2012, 7:52 am

Again not maybe the best quality code.

Converts any byte array to a square bitmap and back.

GOT THE IDEA FROM:

<https://hackforums.net/showthread.php?tid=2339901>

Special thing: bitmap looks just the same as original bytes/file when opened in text editor (pixel values correspond to original bytes) :P

Works only if bitmap is also saved as bitmap file and not compressed etc.

My 0xViewer.exe converted:

[img]<http://img685.imageshack.us/img685/4841/tmp5a88.png>[/img]

BMP opened in notepad, you see the PE header:

[img]<http://img684.imageshack.us/img684/5541/tmp1742.png>[/img]

Maybe gonna add also encryption capability: have other picture with same size used as mask: xor all pixel values. Original data only can be restored using the mask picture :P

NOTE: Avira detects the BITMAP file as DR/FakePic or something like that if exe file is converted to bmp xD

Code /*

* Convert binary data to bitmap and vice versa

* Copyright (C) 2012 0xDEADDEAD

* Removal of this copyright is prohibited

*/

using System;

using System.Collections.Generic;

using System.Drawing.Imaging;

using System.Text;

using System.Drawing;

namespace TestApp

{

public static class BitmapDataStorage

{

public static Bitmap CreateBitmapFromData(byte[] binaryData)

{

// calc padding amount

int paddedSize = binaryData.Length + (3 - binaryData.Length%3) + 6;

int pixelCount = paddedSize/3;

```

int countPerRow = (int) Math.Ceiling(Math.Sqrt(pixelCount));

Bitmap bmp = new Bitmap(countPerRow, countPerRow, PixelFormat.Format24bppRgb);

byte[] paddedData = new byte[paddedSize];
Buffer.BlockCopy(BitConverter.GetBytes(binaryData.Length), 0, paddedData, 0, 4);
Buffer.BlockCopy(binaryData, 0, paddedData, 4, binaryData.Length);

int columnIndex = 0;
int rowNumber = bmp.Height - 1;
for (int i = 0; i < pixelCount; i++)
{
    if (columnIndex == countPerRow)
    {
        columnIndex = 0;
        rowNumber--;
    }

    Color pixelColor = Color.FromArgb(
        paddedData[i*3 + 2],
        paddedData[i*3 + 1],
        paddedData[i*3]);
    bmp.SetPixel(columnIndex, rowNumber, pixelColor);
    columnIndex++;
}
return bmp;
}

public static byte[] ReadDataFromBitmap(Bitmap bitmap)
{
    byte[] buffer = new byte[bitmap.Width*bitmap.Height*3];

    int i = 0;
    for (int y = bitmap.Height - 1; y >= 0; y--)
    {
        for (int x = 0; x < bitmap.Width; x++)
        {
            Color pixelColor = bitmap.GetPixel(x, y);
            buffer[i * 3 + 2] = pixelColor.R;
            buffer[i * 3 + 1] = pixelColor.G;
            buffer[i * 3] = pixelColor.B;
            i++;
        }
    }

    byte[] data = new byte[BitConverter.ToInt32(buffer, 0)];
    Buffer.BlockCopy(buffer, 4, data, 0, data.Length);
    return data;
}
}
}

```

Small sample code I used for test

```
Code byte[] buf = System.IO.File.ReadAllBytes("0xViewer.exe");
BitmapDataStorage.CreateBitmapFromData(buf).Save("bmp.bmp", ImageFormat.Bmp);

Bitmap loaded = (Bitmap)Bitmap.FromFile("bmp.bmp");

System.IO.File.WriteAllBytes("result.exe", BitmapDataStorage.ReadDataFromBitmap(loaded));
```

SIM-1033745 - March 27, 2012, 1:40 pm

That's fucking awesome, nice job!--

N3oNight - March 27, 2012, 3:17 pm

Yap Its Cool..

But any One Can post a VB.Net Version Over here..

hgdfhsdgdgdfh - March 27, 2012, 10:01 pm

Thanks. And for VB.Net you can use some online converter (but im gonna post vb.net version also later :))

CyNecx - March 28, 2012, 6:51 am

<https://hackforums.net/showthread.php?tid=2284748>

nice ^^

Edit : updated version 6mb/sec

-removed not one ite....-

BaussHacker - March 28, 2012, 11:26 am

This is really nice. It seems very interesting.

Kill?Muvz - March 29, 2012, 4:03 am

(Mar 27, 2012 03:17 PM)N3oNight Wrote: [\[CITE\]](#) Yap Its Cool..

But any One Can post a VB.Net Version Over here..

Can't believe you asked for that.

Anyway, it looks freaking awesome, I shall play with it!

Super Scumbag - March 29, 2012, 7:37 pm

Might be able to avoid the avira detection by adding some junk code, or clone of part of a bmp.

I'll play with it once I get VS2010U installed again in a little bit.

hgdfhsdgdgdfh - March 30, 2012, 4:11 am

Avira detection can be avoided by simply reversing the order of image rows.

In BMP file pixels are saved row by row starting from LAST row. So atm I'm starting from last row and going upwards until first. This makes bytes in BMP file be just the same as in original data so thats why I used it :P

0xDEAD10CC - March 31, 2012, 11:33 am

The concept is cool, because uncompressed BMP's are just RGBRGBRGB,,, basically lol. Each pixel color takes up 3 bytes I believe and you would see the 3 parts of that color in #hex form if using a hex editor..

Nice idea though. :)

ForceJew - August 31, 2016, 7:31 am

Looks great!
Really nice!

Oliver Sykes - *September 3, 2016, 11:50 am*

(Aug 31, 2016 07:31 AM)FORC3FI3LD Wrote: [\[CITE\]](#) Looks great!
Really nice!
And well over four years after the last post...