Vinself now with steganography

VinSelf is a known RAT malware already explained on other blogs . It's a family that has been long used in APT attacks. VinSelf can be recognized in two ways:

- the network patterns used;
- the strings obfuscation in the binary.

The VinSelf obfuscation algorithm is quite simple, but specific enough to state that samples using it are from the same family:

```
def vinself_cipher(x, key):
    output = ""
    lkey = ord(x[0])
    for i in xrange(len(x)-1):
        output += chr( ( ( ord(x[i+1]) ^ ord(key[i]) ) - lkey) & 0xff)
        lkey = ord(x[i+1])
    return output
```

Recently, we came accross an interesting sample that, instead of connecting to a malicious C&C, was grabbing a file ("colors.bmp") from Google Docs. Due to the presence of the aforementioned algorithm, the sample had been categorized as VinSelf, so such a behavior was unexpected and confusing.

Starting point

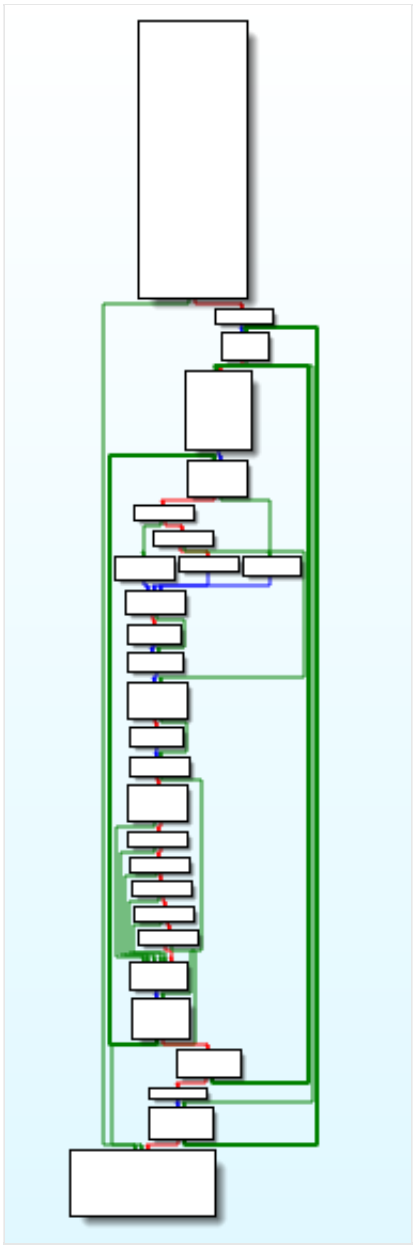
While the image is a valid Bitmap and can actually be displayed, it may be something more than a simple Bitmap.

We have seen pieces of malware appending data at the end of legitimate/innofensive files being retrieved. For example, VinSelf itself sends encrypted data to its C&C prepended by a "GIF89a" header. Foxy also receives encrypted commands from its C&C in what seems a legitimate JPEG image, and Shady RAT is concealing commands in encrypted HTML commentaries, or inside images using real steganography.

Let's look at the code following the retrieval of this file from Google Docs if there's something interesting.

Steganography

The function following the retrieval of the "colors.bmp" file is quite interesting.



```
lea     eax, [ebp+pv]
push    eax             ; pv
push    18h             ; c
push    esi             ; h
call    ds:GetObjectA
mov     ecx, [ebp+yMax]
xor     edx, edx
mov     [ebp+yCur], edx
test    ecx, ecx
jle     loc_10001ECF
```

```
mov     eax, [ebp+xMax]
mov     [ebp+var_68], 3
```

```
loc_10001DF7:
xor     esi, esi
test    eax, eax
jle     loc_10001EC3
```

```
loc_10001E01:           ; y
push    edx
push    esi             ; x
push    [ebp+hdc]       ; hdc
call    ds:GetPixel
movzx   ecx, al
mov     [ebp+var_60], ecx
```

As you can see, it is scanning the image pixel per pixel. The outermost loop is incrementing the row counter, the next one is incrementing the column counter while the innermost one is a loop among the three primary colors.

The function is grabbing the LSB (Least Significant Bit) of each color of each pixel, thus generating three bits of data per pixel of the image. Once all those LSBs have been grabbed, each byte of the bitstream is reversed.

Unciphering

Now that we have extracted the hidden data, it must be unciphered: this is done in four steps:

- the first step is the use of the VinSelf custom obfuscation algorithm with an hard-coded key in the binary;
- the second step is another use of the VinSelf custom obfuscation algorithm with the key decoded at the previous step on the next 32 bytes of the data;
- the third step is a decryption algorithm that was, at first, unknown to us. Thanks to the specific bitwise manipulations employed by this code (shifts and rotations) and to the quick and efficient research of our cryptoteam, it was successfully identified as HC-128, a stream cipher that is not used that much;
- finally, the fourth and last step is once again the use of the VinSelf custom obfuscation algorithm with the key used in the second step on the HC-128 decrypted data.

End point

Ultimately we end up with a C&C configuration that looks like:

```
192.168.1.101:2.2.2.2:3.3.3.3:4.4.4.4
```

As a matter of fact, we changed the real content in the source image to not disclose the real C&C.

So instead of having just one layer of obfuscation (the custom VinSelf algorithm), we end up with several layers:

- the custom VinSelf algorithm encrypting the Google Docs URL;
- an LSB-extraction steganography;
- two instances of the VinSelf algorithm;
- an HC-128 encryption;
- a final VinSelf encryption.

As usual, a script to extract this information from a VinSelf BMP file has been released on our Bitbucket repository.

Steganography is not just for hipsters, it is still being used nowadays.