
Subject: Moving on with supporting old things...
Posted by [mirek](#) on Sun, 31 Jan 2016 10:20:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

I think it is time for finally dropping some old platforms. Major opensource projects like Chrome are now dropping Windows XP support, therefore I think it is time to remove last bits of pre-WinXP support too (I believe that theide could now actually still run in Win2K...)

It would our life much easier if we moved this to at least Windows Vista - Vista API has some really good improvements in multithreading. Do we still need to support WinXP?

In related news, I plan to fully embrace C++11 and drop support for pre-C++11 (I mean, I want to start using C++11 constructs all over the place and stop worrying with old C++ variants).

Mirek

Subject: Re: Moving on with supporting old things...
Posted by [unodgs](#) on Sun, 31 Jan 2016 10:24:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

One of my client still uses XP And it's hard to convince him to drop it since it's working ok... But I can simply use the current uppsrc - not a big problem.

Subject: Re: Moving on with supporting old things...
Posted by [mirek](#) on Sun, 31 Jan 2016 10:51:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well, WinXP vs Win7 is less critical.

About the only ugly thing is that in WinXP TLS variables do not work for LoadLibrary loaded .dlls, in Win7 they do. IMO not a big deal for U++.

Subject: Re: Moving on with supporting old things...
Posted by [mr_ped](#) on Sun, 31 Jan 2016 21:22:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

I don't care about windows at all (I still wonder the people are using it? Actually I'm now for short period time outsourced to other company, and they gave me Windows PC for Android development??? Wonders never cease...).

About C++11 ... yes please. I already use it in all my projects (target platform usually android+iOS+winphone8+, linux for myself during development), and so far only Windows Phone 8/10 is problematic (hint: MS C Compiler).

I daily see such poor C++ code from legacy libraries, and so on... I wish everybody would write at U++ quality level, and imagine that plus C++11/14 ... that would be phenomenal.

Subject: Re: Moving on with supporting old things...
Posted by [Novo](#) on Mon, 01 Feb 2016 05:35:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Sun, 31 January 2016 05:20
In related news, I plan to fully embrace C++11 and drop support for pre-C++11 (I mean, I want to start using C++11 constructs all over the place and stop worrying with old C++ variants).
Mirek

IMHO, it is safe to start using C++14. Even MSVC 2015 upd. 1 supports everything (except of expression SFINAE).

Subject: Re: Moving on with supporting old things...
Posted by [Tom1](#) on Mon, 01 Feb 2016 12:05:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Mirek,

Is this the death of Bazaar::Protect package, which only works on MSC9, or is there a workaround?

Best regards,

Tom

Subject: Re: Moving on with supporting old things...
Posted by [mirek](#) on Mon, 01 Feb 2016 15:09:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Tom1 wrote on Mon, 01 February 2016 13:05Hi Mirek,

Is this the death of Bazaar::Protect package, which only works on MSC9, or is there a workaround?

Best regards,

Tom

I keep this in mind, but I have no time and energy to fix it I have to apologize, but I do not want to hold changes just because Protect...

Why do not you try? It should be quite simple, all you need is to find or implement "length of x86 instruction" algorithm and then replace increments by 1 in 'code scrambler' by increments by length...

Mirek

Subject: Re: Moving on with supporting old things...
Posted by [Tom1](#) on Mon, 08 Feb 2016 09:16:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

Thanks for looking into this. I need to think this through carefully, since there is a continuously increasing need for going 64-bit anyway and this would not solve that part. So, for now, I'll stick with MSC9 and current U++, and return to this if the other solutions are even more difficult.

Best regards,

Tom

Subject: Re: Moving on with supporting old things...
Posted by [mdelfede](#) on Thu, 18 Feb 2016 14:56:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

sorry for my lack of time to fix protect.... we'd need indeed some other way to implement it, in particular because of lacking of inline assembler in 64 bit code.

The "scramble just the opcode" way would be a nice workaround, but only for 32 bit windows.

I think that it would be possible to implement a pure C solution, with help of a couple of external assembly routines, but it needs much work and some time which I don't have right now. I'm just dropping here my idea: we could put a call in front of to-protect code :

```
.....  
DecodeMe(0xaa, 0x23, 0x55, 0x44.....some-unique-byte-pattern);  
.... some code to encrypt  
DecodeEnd(0xaa, 0x23, 0x55, 0x44.....some-unique-byte-pattern);
```

DecodeMe function should be written in pure assembly, and should use return address to locate the code to change.

The DecodeEnd should be a dummy function in order to have an "end pattern" to know where code ends.

In assembly such calls should contain be a sequence of PUSH number PUSH number....., so quite easy to locate, both from external encoder and to internal decoder code.

DecodeMe() function should of course decode just the op-codes as Mirek suggestion.

If I'll find some spare time I'll try to implement it.

Ciao

Massimo
