

BUG: timeSetEvent() wraps after interval of 429496

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.mmedia/2005-06/msg00024.>

- *From:* "Chris P." <msdn@xxxxxxxxxxxxx>
 - *Date:* Fri, 3 Jun 2005 16:26:14 -0400
-

There is a bug in timeSetEvent(), which is used to schedule a callback after a given period, in that if the period is greater than 429496 milliseconds the actual time period will wrap back to x-429496. So for example, setting a period of 430000 which generate a callback in 504ms. This is despite the capabilities reported by timeGetDevCaps() reporting the maximum supported value is 1000000.

The actual exports for these time functions are in winmm.dll, and this bug seems to have been there for quite a while. I've tested on XP SP2 back to Win2K, but don't have anything older on hand. I don't see how it could be used as a security compromise in any way, but the behavior is definately unexpected.

Below is complete repro code. Thanks BilalD for reporting this.

<code>

```
// timerbug.cpp
//
// This code reproduces a bug in timeSetEvent() where if a periodic timing
// event is set for
// a interval larger than 429496 milliseconds, a wrap occurs and the timing
// interval is
// actually x-429496 milliseconds. This also occurs at the mulitple,
// 858992 where the
// result will be x-858992 milliseconds.
// This bug only effects PERIODIC timers, it does not seem to effect
// ONESHOT timers.
```

```
#pragma comment ( lib, "winmm.lib" )
```

```
#include "stdafx.h"
#include <windows.h>
#include <stdio.h>
#include <conio.h>
#include <mmsystem.h>
```

BUG: timeSetEvent() wraps after interval of 429496

```
HANDLE g_timerEvent;
DWORD g_Start;

#define TIMER_INTERVAL 430000
#define MILLIPERSEC 1000.0

void CALLBACK TimerProc(UINT uID, UINT uMsg, DWORD dwUser, DWORD dw1, DWORD
dw2);

int _tmain(int argc, _TCHAR* argv[])
{
printf("Press any key to start timer for %0.3f seconds\n",
TIMER_INTERVAL/MILLIPERSEC);
getch();

TIMECAPS caps;
timeGetDevCaps(&caps, sizeof(TIMECAPS));
printf("Timer caps min:%i max:%i\n", caps.wPeriodMin, caps.wPeriodMax);

g_timerEvent = CreateEvent(NULL, 0, FALSE, NULL);
g_Start = GetTickCount();

MMRESULT result = timeSetEvent(TIMER_INTERVAL, 1000, TimerProc, 0,
TIME_PERIODIC);
// wait for the timerproc to receive timing event
WaitForSingleObject(g_timerEvent, INFINITE);

printf("Test complete\n");
DWORD duration = GetTickCount()-g_Start;
printf("Our timer lasted for %0.3f seconds\n", duration/MILLIPERSEC);
int diff = (int)duration - TIMER_INTERVAL;
diff = abs(diff);
if (diff > 1000)
printf("The timer is broken. Expected %0.1fs, got %0.1fs\n",
TIMER_INTERVAL/MILLIPERSEC, duration/MILLIPERSEC);
else
printf("The timer is working OK.\n");
CloseHandle(g_timerEvent);
getch();
return 0;
}

void CALLBACK TimerProc(UINT uID, UINT uMsg, DWORD dwUser, DWORD dw1, DWORD
dw2)
{
timeKillEvent(uID);
SetEvent(g_timerEvent);
}
```

</code>

-
- *Follow-Ups:*
 - ◆ *Re: timeSetEvent() wraps after interval of 429496*
 - ◇ *From:* Tobia Quantrill
 - ◆ *Re: timeSetEvent() wraps after interval of 429496*
 - ◇ *From:* Alexander Grigoriev
 - Prev by Date: *Re: Images From HTTP server*
 - Next by Date: *Re: can't play wav file snippet*
 - Previous by thread: *can't play wav file snippet*
 - Next by thread: *Re: timeSetEvent() wraps after interval of 429496*
 - Index(es):
 - ◆ *Date*
 - ◆ *Thread*