Subject: Re: NaN Magic Bingo!
Posted by Peter Mason on Thu, 15 Nov 2007 21:16:05 GMT

Mort Canty wrote:

<...>
> It was the CISCO VPN client! Thank you Peter. That was the best four
> cents I ever got! If you're ever anywhere near the town of Juelich,
> Germany, let me know. I owe you a big, cold, German beer.
>
> Mort
>
> PS. CISCO -> The Cisco Kid (50's radio) -> Mexico -> Coyote ... David
> was right, as usual.

Thanks Mort, I'll keep that beer in mind.   I've never actually been to
Germany, even though my mother grew up next door (Switzerland).
I'm glad it worked out.   It's hard to believe that something as mainstream
as VPN Client 4.x could have such a glitch in it but there you go.   Seems
none of us software developers are safe from bugs.
In case anyone's curious, here's a simple little test routine that uses
brute force to expose the bug.   (Sorry it's in C.)

Cheers
Peter

```c
// Little FP integrity checker.   MUST be called with VAL=1.

// Returns 0 if okay and 1 if not.

#pragma optimize( "", off )    //a smart compiler might see me coming and
avoid some of my calcs

static int FPIntegrity(HWND hw, int val)

{

register double a=val, b=val, c=val, d=val, e=val, f=val;

register int i, j, k;

for( j=1, k=0; j && (k<1000); ++k ) {

for( i=0x00400000; i&&j; --i ) { //do some simple calcs that shouldn't
change B, C, D, E or F (they should all remain at 1.0)

b *= a;
```

```
c += a-b;

d += b-a;

e *= b+c-d;

f -= (a-b+c-d)*e;

if( a-b+c-d+e-f ) {

char tx[256];

j=0;

sprintf( tx, "Floating-point Integrity broke on iteration %d.\nThe following
numbers should all have been 0:\n%f %f %f\n%f %f %f",
k,a-1,b-1,c-1,d-1,e-1,f-1 );

MessageBox( hw, tx, "Warning", MB_ICONWARNING|MB_OK|MB_TOPMOST );

} }

}

return j ? 0 : 1;

}

#pragma optimize( "", on )
```

Subject: Re: NaN Magic Bingo!
Posted by R.Bauer on Mon, 19 Nov 2007 09:19:41 GMT
View Forum Message <> Reply to Message

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Peter Mason schrieb:
> Mort Canty wrote:
> <...>
>> It was the CISCO VPN client! Thank you Peter. That was the best four
>> cents I ever got! If you're ever anywhere near the town of Juelich,
>> Germany, let me know. I owe you a big, cold, German beer.
>>
>> Mort
>>
>> PS. CISCO -> The Cisco Kid (50's radio) -> Mexico -> Coyote ... David

>> was right, as usual.
>
> Thanks Mort, I'll keep that beer in mind.   I've never actually been to
> Germany, even though my mother grew up next door (Switzerland).
> I'm glad it worked out.   It's hard to believe that something as mainstream
> as VPN Client 4.x could have such a glitch in it but there you go.   Seems
> none of us software developers are safe from bugs.
> In case anyone's curious, here's a simple little test routine that uses
> brute force to expose the bug.   (Sorry it's in C.)
>
> Cheers
> Peter

Dear Peter

thanks for pointing this out. Do you have knowledge on which platforms
this trouble occure. Does the cisco code affect all platforms.

cheers
Reimar

>
> // Little FP integrity checker.   MUST be called with VAL=1.
>
> // Returns 0 if okay and 1 if not.
>
> #pragma optimize("", off)   //a smart compiler might see me coming and
> avoid some of my calcs
>
> static int FPIntegrity(HWND hw, int val)
>
> {
>
> register double a=val, b=val, c=val, d=val, e=val, f=val;
>
> register int i, j, k;
>
> for(j=1, k=0; j && (k<1000); ++k) {
>
> for(i=0x00400000; i&&j; --i) { //do some simple calcs that shouldn't
> change B, C, D, E or F (they should all remain at 1.0)
>
> b *= a;
>
> c += a-b;
>

```
> d += b-a;
>
> e *= b+c-d;
>
> f -= (a-b+c-d)*e;
>
> if( a-b+c-d+e-f ) {
>
> char tx[256];
>
> j=0;
>
> sprintf( tx, "Floating-point Integrity broke on iteration %d.\nThe following
> numbers should all have been 0:\n%f %f %f\n%f %f %f",
> k,a-1,b-1,c-1,d-1,e-1,f-1 );
>
> MessageBox( hw, tx, "Warning", MB_ICONWARNING|MB_OK|MB_TOPMOST );
>
> } }
>
> }
>
> return j ? 0 : 1;
>
> }
>
> #pragma optimize( "", on )
>
>
```

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.5 (GNU/Linux)
Comment: Using GnuPG with SUSE - http://enigmail.mozdev.org

 iD8DBQFHQVUh5aOc3Q9hk/kRAo0hAJ92tUkGMIIsZ5BNhMeIVP07yK13MQCf bkh6
E6wlxjZoMzpQC6dJVy26bYY=
=dnTg
-----END PGP SIGNATURE-----