From:	David A. Cooper
To:	Bassham, Lawrence E. (Fed)
Subject:	Re: Constant time memcmp
Date:	Tuesday, June 23, 2020 4:34:14 PM

It certainly looks like it doesn't. Are there cases where one needs a constant time memcmp where the caller needs to know more than just whether the two inputs are the same?

On 6/23/20 8:22 AM, Bassham, Lawrence E. (Fed) wrote:

> I didn’t want to bother everyone with this, but will the OpenSSL version of the code still provide the “less than”/“greater than’/“equal” notion that traditional memcmp has? Daniel’s version won’t.
>
> Larry
>
> ---
>
> On: 22 June 2020 15:20, "David A. Cooper" <david.cooper@nist.gov> wrote:
>
> > I'm not an expert on this, but my guess would be that even this version isn't constant time, especially since a compiler may optimize out the "else" part.
> >
> > Here is how it was done in OpenSSL:
> >
> > ```
> > /*
> >  * The volatile is used to to ensure that the
> > compiler generates code that reads
> >  * all values from the array and doesn't try to
> > optimize this away. The standard
> >  * doesn't actually require this behavior if
> > the original data pointed to is
> >  * not volatile, but compilers do this in
> > practice anyway.
> >  *
> >  * There are also assembler versions of this
> > function.
> >  */
> > # undef CRYPTO_memcmp
> > int CRYPTO_memcmp(const void * in_a, const void
> > * in_b, size_t len)
> > {
> >     size_t i;
> >     const volatile unsigned char *a = in_a;
> >     const volatile unsigned char *b = in_b;
> >     unsigned char x = 0;
> >
> >   for (i = 0; i < len; i++)
> >   x |= a[i] ^ b[i];
> >
> >   return x;
> > }
> > ```