

[Home](#) [Services](#) [Products & Freebies](#)



[Case Studies](#) [Contact Us](#)

Posted on [2020-02-05](#)

[← Previous](#) [Next →](#)

Stay positive Lolbins... not!

Update

After I posted this, a number of comments on Twitter followed; there was an interesting development from [@sixtyvividtails](#):

[illegible]

As such, a very well-known lolbin:

- `rundll32 advpack.dll, RegisterOCX calc.exe`

(which in turns loads the `advpack.dll`, resolves its `RegisterOCXW` function before calling it with a `'calc.exe'` argument) can take a form of:

- `rundll32 advpack.dll, #12 calc.exe`

I checked on Windows XP, Windows 7, and Windows 10 and the ordinal is always the same. Using the same approach one can find similar syntax for other lolbins. Simple, but possibly evasive.

Oh wait... but this is not the end.

The `Rundll32` takes ordinal numbers by using the following syntax:

`#<number>`

The `<number>` is converted from a string to an integer using a `wtoi` function. This API in turn accepts both positive and negative numbers.

Aha...

Knowing how positive and negative numbers are represented in memory, we can easily come up with a negative number that will be converted by `wtoi` to... a positive 12:

Try this:

- `rundll32 advpack.dll, #-4294967284 calc.exe`

Btw. if you are wondering, `'+'` prefix works too:

- `rundll32 advpack.dll, #+12 calc.exe`

After playing with it a bit more, you can also add some additional stuff after the digits e.g.:

- rundll32 advpack.dll, #-4294967284-foobar calc.exe

And yes, there is one moar... for 64-bit rundll you can run:

- rundll32 advpack.dll, #-1152921504606846964 calc.exe

Happy hunting!

This entry was posted in [Anti-Forensics](#), [Living off the land](#), [LOLBins](#) by [adam](#). Bookmark the [permalink](#).

[Privacy Policy](#) | Proudly powered by [WordPress](#)