Basic commands for Windbg – breakpoints Part 5 – Conditional Breakpoints

By Anand George

SourceLens.Com.Au - Training, Consulting and Mentoring -Sydney, NSW, Australia
All Rights Reserved

Conditional Breakpoints

- Use to filter out the unnecessary breaks of bp, bm, bu, ba variant.
- A small "program" is given with the breakpoint command which decide to break or not.
- We can even use the program to print the stack on break and just "go"
- Its always better to use gc than g to go from a conditional break.

SourceLens.Com.Au - Training, Consulting and Mentoring -Sydney, NSW, Australia
All Rights Reserved

Conditional Breakpoints

```
bp HelloWorld!MyTestFunc ".if ( poi(testVar)>0n1500) {} .else {gc} "
```

SourceLens.Com.Au - Training, Consulting and Mentoring -Sydney, NSW, Australia
All Rights Reserved

Demo

SourceLens.Com.Au - Training, Consulting and Mentoring -Sydney, NSW, Australia
All Rights Reserved

SourceLens
sourcelens.com.au/Training
sourcelens.com.au/Mentoring
sourcelens.com.au/Consult

Can be a bit more complicated.

```
bp kernel32!CreateEventW "$$<c:\\commands.txt“

.if (@r9 != 0)
{
  as /mu ${/v:EventName} @r9
}
.else
{
  ad /q ${/v:EventName}
}
.if ($spat(@"${EventName}", "Global*") == 0)
{
  gc
}
.else
{
  .echo EventName
}
```

SourceLens.Com.Au - Training, Consulting and Mentoring -Sydney, NSW, Australia
All Rights Reserved

Try avoid complicated conditions

- These scripts are more or less difficult to – debug.
- They are slow.
- Last and worst you wont find enough sample scripts anywhere to modify and accomplish the task.

SourceLens.Com.Au - Training, Consulting and Mentoring -Sydney, NSW, Australia
All Rights Reserved

SourceLens
sourcelens.com.au/Training
sourcelens.com.au/Mentoring
sourcelens.com.au/Consult

Another better approach towards complicated debug script.

- Just print everything and grep (search) using your favorite text editor. Mine is visual studio ☺
- Example –
  - Find out if a called to nt! IoGetDmaAdapter is from a particular binary/code (suspected malware where we don't have the symbol but some suspicious address ranges in which any where eip can be present – may not be even a binary).
    - It would be nice to have conditional breakpoint but it can be complicated and slow.
    - Rather I recommended just print all the stacks and just grep (search) and it worked.

SourceLens.Com.Au - Training, Consulting and Mentoring -Sydney, NSW, Australia
All Rights Reserved

Other ways to filter the output (kernel)

- **/1** Creates a "one-shot" breakpoint. After this breakpoint is triggered, the breakpoint is permanently removed from the breakpoint list.
- Give process context if possible.
  /p *EProcess* (Kernel mode only) Specifies a process that is associated with this breakpoint. *EProcess* should be the actual address of the EPROCESS structure, not the PID. The breakpoint is triggered only if it is encountered in the context of this process.
- Give thread context of interest if possible.
  /t *EThread* (Kernel mode only) Specifies a thread that is associated with this breakpoint. *EThread* should be the actual address of the ETHREAD structure, not the thread ID. The breakpoint is triggered only if it is encountered in the context of this thread. If you use **/p** *EProcess* and **/t** *EThread* , you can enter them in either order.
- **/c** *MaxCallStackDepth* Causes the breakpoint to be active only when the call stack depth is less than *MaxCallStackDepth*. You cannot combine this option together with **/C**.
- **/C** *MinCallStackDepth* Causes the breakpoint to be active only when the call stack depth is larger than *MinCallStackDepth*. You cannot combine this option together with **/c**.

SourceLens.Com.Au - Training, Consulting and Mentoring -Sydney, NSW, Australia
All Rights Reserved

Summary

- Conditional Breakpoints
- Try grep(search) in the non filtered output or partially filtered output than giving very complicated conditional breakpoint.
- Some other ways to filter the number of breaks.

SourceLens.Com.Au - Training, Consulting and Mentoring -Sydney, NSW, Australia
All Rights Reserved

SourceLens
sourcelens.com.au/Training
sourcelens.com.au/Mentoring
sourcelens.com.au/Consult

Thank you

SourceLens.Com.Au - Training, Consulting and Mentoring -Sydney, NSW, Australia
All Rights Reserved