

This is a fully working example of using Intel's GD (General Detection) bit, to invoke debug exceptions upon any access to a debug register.

Currently, the provided source will lock down any drx access to only NTICE, a hardcoded base for my NTICE is in the src, you may need to modify this for your own, search the source for the keyword ACCESS_RIGHTS.

Any attempt of a MOV REG, DRX will be 'faked' by placing a default value into the reg to fool the calling app into thinking no BPMs are set. Any attempt of a MOV DRX, REG will be totally ignored, or emulated if NTICE is the caller.

All output is given via debug msgs which have been formatted to be read by sysinternals debugview (included) with force linefeed on.

Example data below:

```
-- I SET MY BPM TO CATCH CODE WRITTING --

00000548    1.99602408  Alert: MOV DR3, ESI ADDR: B666917B
00000549    1.99604000  004010DC    < - contents of esi
00000550    1.99604894  DRX Updated

00000551    1.99606207  Alert: MOV DR7, EBX ADDR: B666917E
00000552    1.99607911  100004C0
00000553    1.99608777  DRX Updated

--- ASPROTECT SEH TRIES TO OVERWRITE, KERNEL IS DENIED ----

00000557    6.80938862  Alert: MOV DR0, ESI ADDR: 80465362
00000558    6.80940343  00000000
00000559    6.80941293  [o] Ignored  MOV DRX, REG

00000560    6.80943025  Alert: MOV DR1, EDI ADDR: 80465368
00000561    6.80944506  00000000
00000562    6.80945427  [o] Ignored  MOV DRX, REG

00000563    6.80947159  Alert: MOV DR2, EBX ADDR: 8046536B
00000564    6.80948640  00000000
00000565    6.80949590  [o] Ignored  MOV DRX, REG

00000566    6.80951182  Alert: MOV DR3, ESI ADDR: 80465377
00000567    6.80952775  00000000
00000568    6.80953725  [o] Ignored  MOV DRX, REG

00000569    6.80955317  Alert: MOV DR6, EDI ADDR: 8046537A
00000570    6.80956965  0000A005
```



```
00000571    6.80957915  [o] Ignored  MOV DRX, REG
00000572    6.80959535  Alert: MOV DR7, EBX ADDR: 8046537D
00000573    6.80961156  00000155
00000574    6.80962133  [o] Ignored  MOV DRX, REG
```

--- Running a SafeDisc EXE -----

Me setting a BPM, NTICE in action

```
00000566    3.91877124  Alert: MOV DR3, ESI ADDR: B666917B
00000567    3.91878856  00405273
00000568    3.91879750  DRX Updated

00000569    3.91881063  Alert: MOV DR7, EBX ADDR: B666917E
00000570    3.91882655  100004C0
00000571    3.91883522  DRX Updated
```

some memory doing checks, turns out to be SECDRV safediscs driver
drx requests are faked.

```
00000932    9.08994774  Alert: MOV EAX, DR1 ADDR: B707593A
00000933    9.08998239  00000000
00000934    9.08999216  [o] Faked MOV REG, DRX

00000935    9.09002122  Alert: MOV EAX, DR7 ADDR: B7075950
00000936    9.09003658  00000400    <- fake value placed in reg
00000937    9.09004608  [o] Faked MOV REG, DRX

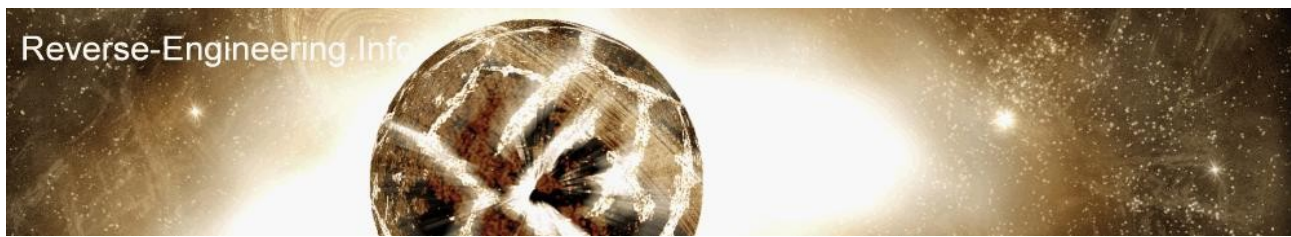
00001154    9.60005839  Alert: MOV EAX, DR7 ADDR: B70759DA
00001155    9.60010811  00000400
00001156    9.60011817  [o] Faked MOV REG, DRX

00001157    9.60336942  Alert: MOV EAX, DR7 ADDR: B70759DA
00001158    9.60341161  00000400
00001159    9.60342138  [o] Faked MOV REG, DRX

00001160    9.61938878  Alert: MOV EAX, DR7 ADDR: B70759DA
00001161    9.61943180  00000400
00001162    9.61944716  [o] Faked MOV REG, DRX

00001163    9.62103089  Alert: MOV EAX, DR7 ADDR: B70759DA
00001164    9.62106245  00000400
00001165    9.62107279  [o] Faked MOV REG, DRX
```

--



and for interest heres ntice operating my bc* request

```
00000575    8.94788583  Alert: MOV ECX, DR7 ADDR: B66691C4
00000576    8.94790343  100004C0
00000577    8.94791321  [o] Emulated MOV REG, DRX

00000578    8.94792914  Alert: MOV DR7, ECX ADDR: B66691C9
00000579    8.94794478  10000400
00000580    8.94795344  DRX Updated
```

comments, bug reports to ryates@reverse-engineering.info