

Forum:[Hack Forums](#)

Thread: [.NET Reverse Engineering Tutorial Series - by debug](#)

debug - April 5, 2016, 4:37 pm

.NET Reverse Engineering Tutorial Series

Preface

I've been wanting to do this for a long time as I enjoy reverse engineering and also securing my own applications.

Tutorials

Ep. 1: The Basics of MSIL - <https://www.youtube.com/watch?v=aYZmP7CEO6w>

Ep. 2: The Essential Tools and Basic CrackMe - <https://www.youtube.com/watch?v=sFqcXDHtMeI>

Feedback

Please leave me some comments, questions, or let me know if I have explained something incorrectly or insufficiently. I'll be happy to correct mistakes and answer your questions, so let me know!

Credits

LaPanthere - My partner in crime.

0xd4d - de4dot / dnSpy

Plugs

Visit my website: <https://debug.im> (Will post more stuff like this in the future!)

SIM-1725686 - April 5, 2016, 4:45 pm

Just watched the first video, it's good. I like the way you explained things. And it was definitely a good idea to start your series by explaining MSIL, as that will make it much easier to understand DREDD-ing as you go on.

10/10 for episode 1.

debug - April 5, 2016, 4:49 pm

(Apr 5, 2016 04:45 PM)regiostech Wrote: [\[CITE\]](#) Just watched the first video, it's good. I like the way you explained things. And it was definitely a good idea to start your series by explaining MSIL, as that will make it much easier to understand DREDD-ing as you go on.

10/10 for episode 1.Thanks dude! Working on episode two as we speak, should be up shortly. ?

Unknown - April 5, 2016, 5:06 pm

Pretty good tutorial man, Looking forward to seeing more of these types of tutorials.

debug - April 5, 2016, 5:08 pm

(Apr 5, 2016 05:06 PM)Cynn. Wrote: [\[CITE\]](#) Pretty good tutorial man, Looking forward to seeing more of these types of tutorials.Appreciate ya!

Second tutorial has been posted - <https://www.youtube.com/watch?v=sFqcXDHtMeI>

Wardow - April 5, 2016, 5:10 pm

"LaPanthere - My partner in crime."
Interesting.

???? - April 5, 2016, 5:13 pm

I'm really enjoying this :)

Keep it going!

Btw you forgot to post the crackme dl.

debug - April 5, 2016, 5:34 pm

(Apr 5, 2016 05:13 PM)ScorpiOn Wrote: [\[CITE\]](#) I'm really enjoying this :)

Keep it going!

Btw you forgot to post the crackme dl.Updated the video description with the link. Enjoy!

VmX5 - April 5, 2016, 6:37 pm

Nice tutorials! I'd love to see this become a series, I learnt a lot from just the 2 episodes! Great work!

Only3 - April 6, 2016, 12:07 am

patching if else is not really reversing but good to see this kind of threads

Iterator - April 6, 2016, 1:10 am

Great tutorials, thanks

Waiting for more

Cecilia - April 6, 2016, 5:57 am

Thanks for the reference :)

Good tutorials for beginners, I've told you my feedback already.

Great to see someone who also cares for MSIL.

P.S you can also right click on the branch instruction and hit 'invert branch'.

Aristocles - April 6, 2016, 10:55 am

w8 m8 its not the good place to post this thread, move it to DREDD

debug - April 6, 2016, 12:43 pm

(Apr 6, 2016 10:55 AM).NETGuard Wrote: [\[CITE\]](#) w8 m8 its not the good place to post this thread, move it to DREDDREDD's dead, baby.

Aristocles - April 7, 2016, 5:03 am

(Apr 6, 2016 12:43 PM)debug Wrote: [\[CITE\]](#) DREDD's dead, baby.

«DREDD's dead» So poetic

debug - April 10, 2016, 2:46 pm

Thanks for all the feedback guys. Feel free to leave suggestions for the next episodes.

\$lick - April 11, 2016, 6:45 am

Debug, teaching children to steal because they are poor since 2002.

Vouch.

debug - April 11, 2016, 3:21 pm

(Apr 11, 2016 06:45 AM)Nexus™ ?? Wrote: [\[CITE\]](#) Debug, teaching children to steal because they are poor since 2002.

Vouch.You know it, bb.

VisualSoft - April 23, 2016, 7:18 am

was fun watching this Videos.

i would request you to post some nice tutorials to Protect App from being Cracked from skid Crackers.

I know you can not make it impossible to get cracked, but can make it hard.

???? - April 23, 2016, 8:19 pm

You should do an episode 3. :)

Corios - April 23, 2016, 8:56 pm

Only watched the first video so far, but one thing I will note:

You shouldn't just nop instructions instead of deleting them. Deleting an operation isn't going to mess with the call stack over noping it, and actually has a higher chance of not messing the source up in the case you're attempting to modify msil at run time, as it has already aligned its boundary. And for compile time modification, I mean it's a very slim chance, but if something is dependent on another thread, you creating a nop sled might dis align these times (which this would be a bad programming practice on the authors part in the first place, but still a possibility).

In general, disregarding the small time a nop instruction takes to hit, it's better to remove than to nop.

(Apr 6, 2016 12:07 AM)Only3 Wrote: [\[CITE\]](#) patching if else is not really reversing but good to see this kind of threads
no... by definition, it is.

DisguiserNET - April 24, 2016, 5:27 am

(Apr 23, 2016 08:56 PM)Corios Wrote: [\[CITE\]](#) Only watched the first video so far, but one thing I will note:

You shouldn't just nop instructions instead of deleting them. Deleting an operation isn't going to mess with the call stack over noping it, and actually has a higher chance of not messing the source up in the case you're attempting to modify msil at run time, as it has already aligned its boundary. And for compile time modification, I mean it's a very slim chance, but if something is dependent on another thread, you creating a nop sled might dis align these times (which this would be a bad programming practice on the authors part in the first place, but still a possibility).

In general, disregarding the small time a nop instruction takes to hit, it's better to remove than to nop.

no... by definition, it is.This is wrong. Nopping an instruction insted of deleting them preserves branches while deleting an instruction might cause problems. And i dont get why this should cause problems with the call stack, since nop actually is "no operation"

DonD - April 24, 2016, 5:28 am

Very much appreciate this little starter thread for people ionterested in debugging.
Thanks alot :D

Dennamanarjag - April 24, 2016, 8:28 am

(Apr 23, 2016 08:56 PM)Corios Wrote: [\[CITE\]](#) In general, disregarding the small time a nop instruction takes to hit, it's better to remove than to nop.

As someone else said, removing instructions may mess with branching offsets which would need to be updated, replacing with NOP does not.

Additionally I think you're all (including OP) confusing the call stack with the evaluation stack. They're two separate things and not really related. You have to be careful with just NOPing instructions willy-nilly. As an example; a conditional branch pops a value off the eval stack and if you simply NOP it it'll most likely fuck the rest of the code up since the object at the top of the stack that's meant to be popped by the branch will linger.

Corios - April 24, 2016, 11:33 pm

(Apr 24, 2016 05:27 AM)DisguiserNET Wrote: [\[CITE\]](#) This is wrong. Nopping an instruction insted of deleting them preserves branches while deleting an instruction might cause problems. And i dont get why this should cause problems with the call stack, since nop actually is "no operation"

(Apr 24, 2016 08:28 AM)Dennamanarjag Wrote: [\[CITE\]](#) As someone else said, removing instructions may mess with branching offsets which would need to be updated, replacing with NOP does not.

Additionally I think you're all (including OP) confusing the call stack with the evaluation stack. They're two separate things and not really related. You have to be careful with just NOPing instructions willy-nilly. As an example; a conditional branch pops a value off the eval stack and if you simply NOP it it'll most likely fuck the rest of the code up since the object at the top of the stack that's meant to be popped by the branch will linger.

I must not read it the same way you guys did, I was agreeing with the part of "Deleting an operation isn't going to mess with the call stack over noping it". (The OP said it will mess with call stack, I'm stating it won't)

And to address the other part: So you would rather take the chance of a branch falling through than experiencing a compilation error? This is leading to unsafe operations in a safe language. Deleting a line instead of noping it will cause the method to be unable to be written so you can fix the error on the spot, noping it is going to cause undefined behavior especially if a large change is made. You shouldn't nop things.

DisguiserNET - April 24, 2016, 11:55 pm

(Apr 24, 2016 11:33 PM)Corios Wrote: [\[CITE\]](#) And to address the other part: So you would rather take the chance of a branch falling through than experiencing a compilation error? This is leading to unsafe operations in a safe language. Deleting a line instead of noping it will cause the method to be unable to be written so you can fix the error on the spot, noping it is going to cause undefined behavior especially if a large change is made. You shouldn't nop things.

You shouldnt talk about the call stack but the evaluation stack insted. And of course removing an instruction might mess up the stack and exceed the maxstack. You gotta fix that in both possibilities. However, you dont have to care about branches when you nop insted of deleting. And this is the reason why unexperienced reverse engineers should nop :D

Moreover, you can always create unsafe code with .net language and the part with "unable to be written" is just a thing you favourite decompiler will tell you by default, its possible nonetheless

Dennamanarjag - April 25, 2016, 1:06 am

(Apr 24, 2016 11:33 PM)Corios Wrote: [\[CITE\]](#) And to address the other part: So you would rather

take the chance of a branch falling through than experiencing a compilation error? This is leading to unsafe operations in a safe language.

A compilation error? Are you talking about when the JIT compiles it? I don't see how this leads "unsafe operations" any more than removing an instruction.

(Apr 24, 2016 11:33 PM)Corios Wrote: [\[CITE\]](#) Deleting a line instead of noping it will cause the method to be unable to be written so you can fix the error on the spot

Are we talking about patching built assemblies here? Because removing an operation is definitely not guaranteed to make the assembly unable to be written any more than a NOP would.

(Apr 24, 2016 11:33 PM)Corios Wrote: [\[CITE\]](#) noping it is going to cause undefined behavior especially if a large change is made.

Implying simply removing a branch won't cause undefined behavior? You talked about creating a NOP sled earlier. Removing a branch instead of NOPing is more likely to cause behavior similar to a NOP sled e.g misaligned relative branches.

Not to mention if the application is distributed in Release mode as it should NOPs won't even be compiled so you get the both of best worlds.

Cecilia - April 25, 2016, 5:21 am

I see no reason to not nop, except the rare chance something will fuck up, which is easy to detect and fix by deletion.

As others have said, there will always be issues with your exception handlers and other branches that rely on those offsets when you delete an instruction.

Modern libraries do adjust for this and will not be a problem in most decompilers/deobfuscators/assembly modification libraries.

I'd rather nop than delete because it is easier, its such a small detail guys, there's no reason to go into a debate about 'nop vs delete' it is just something people do or do not do, depending on the situation.

debug - August 28, 2016, 7:01 pm

Thinking about reviving this series if anyone has any interest.

baxterbigboy - August 29, 2016, 9:54 pm

I'm very interested @debug

-SomeE. - August 31, 2016, 9:00 am

(Aug 28, 2016 07:01 PM)debug Wrote: [\[CITE\]](#) Thinking about reviving this series if anyone has any interest.

You should totally, really soothing voice

ForceJew - August 31, 2016, 9:32 am

Looks great!

Looking forward for more~!

debug - August 31, 2016, 9:36 am

(Aug 31, 2016 09:00 AM)-SomeE. Wrote: [\[CITE\]](#) You should totally, really soothing voiceHaha, thanks.
?

If anyone has suggestions for what they'd like to see next, let me know. I'm thinking of doing some more complicated, unobfuscated examples first. Then after that, moving on how to deobfuscate.

baxterbigboy - *September 1, 2016, 7:27 am*

I would like to see unpacking, deobfuscation etc.

I hope I'm not asking too much lol

debug - *September 1, 2016, 7:32 am*

(Sep 1, 2016 07:27 AM)baxterbigboy Wrote: [\[CITE\]](#) I would like to see unpacking, deobfuscation etc.

I hope I'm not asking too much lol I'll do that next. Thanks

Infinity. - *November 20, 2016, 6:58 pm*

botbotRDP made my day

Thanks

DopplerAF - *November 21, 2016, 12:23 am*

Pretty good, I have to tell this is really helping me.

Keep it up.

lcoderz - *November 21, 2016, 1:33 am*

Nice tutorials! thanksfor share gay !