

Forum:[Hack Forums](#)

Thread: [Want to get started in reverse engineering? Look here.](#)

Phizo - January 11, 2017, 8:40 pm

Matasano, a bunch of awesome people in the infosec business, have made progressive challenges for those new to reverse engineering. You may know them through their cryptography challenges (<http://cryptopals.com>). The setting of the reverse engineering challenges are similar to the cryptography challenges, and being an advocate of the challenges for learning how to break crypto, you can be rest assured this is a road to reverse engineering worth driving down.

Don't know Assembly or how to use a low-level debugger? Great, this is for you. You will be working with a simple instruction set; namely, MSP430 (a low-powered 16-bit MCU). This will be of great benefit to you when you go out into the wild and work with IA-32 or x86-64. The debugger, as well as everything else you need, is built into a single web-page. It's beautifully designed.

Here are screenshots of the beginning of one of the challenges, showing the interface:

<https://i.imgur.com/Fiq8NYU.png>

<https://i.imgur.com/ZWjY45V.png>

Everything you need to know about the challenges is here: <https://microcorruption.com/about>
Register and login from here: <https://microcorruption.com/login>

Here is the extremely small user manual: <https://microcorruption.com/manual.pdf>

Everything you need to know about MSP430, including the Assembly language, can be found here: https://en.wikipedia.org/wiki/TI_MSP430

I recommend keeping these both open while you work through the challenges.

Definetly check out the crypto challenges at cryptopals if you're interested. You won't regret it. I've done a fair amount, so if you need help, let me know.

Stoels - January 11, 2017, 9:32 pm

Thanks for the tip! Will check this out when I have the chance.

Twistd - January 12, 2017, 4:33 am

Why would you learn people to use this webcrap when they miss out learning the real deal? ollydbg/ida. Doing this properly through olly would be a way better way to learn. This however seems like a waste of time.

I really dont see the purpose of learning through some web ui thats missing a lot of crucial features you'd be using while reversing in olly, also the fact that its 16 bit in stead of just going for the real deal 32 bit.

Phizo - January 12, 2017, 4:43 am

(Jan 12, 2017 05:33 AM)Twistd Wrote: [\[CITE\]](#) Why would you learn people to use this webcrap when they miss out learning the real deal? ollydbg/ida. Doing this properly through olly would be a way better way to learn. This however seems like a waste of time.

I really dont see the purpose of learning through some web ui thats missing a lot of crucial features you'd be using while reversing in olly, also the fact that its 16 bit in stead of just going for the real deal 32 bit.

If you read the thread, you would see that the idea is to provide a softer introduction to those who

are very new (no experience with Assembly or a low-level debugger) so that they may make the transition to more complex architectures like IA-32 or x86-64. The fact that it is 16-bit holds no relevance. Do you even have any idea what you're talking about? This is not some fake, web-based shit. It runs on an actual CPU mounted on a microcontroller. It all works as it would in a real environment. It's a real (but simple) debugger, a real CPU, with real live memory inspection and I/O. What's more, you're given vulnerable binaries to reverse as challenges. How can you not see how awesome this is?

Twistd - January 12, 2017, 7:01 am

(Jan 12, 2017 05:43 AM)Phizo Wrote: [\[CITE\]](#) If you read the thread, you would see that the idea is to provide a softer introduction to those who are very new (no experience with Assembly or a low-level debugger) so that they may make the transition to more complex architectures like IA-32 or x86-64. The fact that it is 16-bit holds no relevance. Do you even have any idea what you're talking about? This is not some fake, web-based shit. It runs on an actual CPU mounted on a microcontroller. It all works as it would in a real environment. It's a real (but simple) debugger, a real CPU, with real live memory inspection and I/O. What's more, you're given vulnerable binaries to reverse as challenges. How can you not see how awesome this is?

Yes i know perfectly well what im talking about:)

I just dont see the point of someone using a web UI which is way less advanced than just learning olly straight up.

It's not like ollydbg is super hard so you need to use this to manage the transition to ollydbg later.

I view it differently.. Ollydbg has quite a lot of options and lots of them help you on your way to achieving different goals.

By using this simple UI without actually using the software youre going to be using in the future.. youre just stalling time.

That's how i see it though, pretty much waste of time when you can learn perfectly fine using ollydbg.

I get the idea, you think its cool and all that... I can also see you've put some work in to it... It's pretty well done.

I can understand that your debugger looks less frightening than looking at ollydbg for the first time... But when people start looking at it.. its not really that hard...

Maybe im just different than some people.. i like to just jump in to the worst shit straight up to get done with it.. in stead of using stepping stones or going the long way around :)

Phizo - January 12, 2017, 8:15 am

(Jan 12, 2017 08:01 AM)Twistd Wrote: [\[CITE\]](#) Yes i know perfectly well what im talking about:)

I just dont see the point of someone using a web UI which is way less advanced than just learning olly straight up.

It's not like ollydbg is super hard so you need to use this to manage the transition to ollydbg later.

I view it differently.. Ollydbg has quite a lot of options and lots of them help you on your way to achieving different goals.

By using this simple UI without actually using the software youre going to be using in the future.. youre just stalling time.

That's how i see it though, pretty much waste of time when you can learn perfectly fine using ollydbg.

I get the idea, you think its cool and all that... I can also see you've put some work in to it... It's pretty well done.

I can understand that your debugger looks less frightening than looking at ollydbg for the first time...

But when people start looking at it.. its not really that hard...

Maybe im just different than some people.. i like to just jump in to the worst shit straight up to get done with it.. in stead of using stepping stones or going the long way around :)

I think you misunderstand what this is, my friend. You seem to be implying that I created this: I didn't. This is made by Matasano, who are a team of very skilled people who do security consulting among other services for a living. Their cryptography challenges are well-renowned also.

I understand your point. It's good to learn how to work in a real environment. I'm not denying that OllyDbg or IDA Pro are much better debuggers, but it's beside the point. This is a CTF (Capture The Flag). Have you played wargames before, such as Smash The Stack, Over The Wire, etc.? This is all this is. But in addition, I think it can be a good starting point for people who are new to reverse engineering, learning by working through the challenges, which are progressively difficult. Even if you're experienced, you may benefit from these challenges once they get to a point by which you find them to be difficult.

In doing so, a newbie becomes familiar with a simple debugger and a simple instruction set, which can give them preliminary knowledge and practice that will help with the transition to more complicated debuggers and instruction sets. Remember it's not all about the debugger, which is expendable when you consider it can change depending on the operating system you're working on: it's becoming proficient in assembly, understanding memory and overall computer architecture, file formats, memory corruption bugs, etc. This is the sort of thing this CTF helps to introduce.

I hope that clears it up. :)

EDIT: Also, I never said what the debugger is capable of. It's very primitive in comparison to OllyDbg, but it has the necessities. It's great for a CTF.

Code > help

Valid commands:

- Help - show this message
- Solve - solve the level on the real lock
- Reset - reset the state of the debugger
- (C)ontinue - run until next breakpoint
- (S)tep [count] - step [count] instructions
- step Over / (N)ext - step until out or pc is next instruction
- step Out / (F)inish - step until the function returns
- (B)reak [expr] - set a breakpoint at address
- (U)nbreak [expr] - remove a breakpoint
- (R)ead [expr] [c] - read [c] bytes starting at [expr]
- track [reg] - track the given register in memory
- untrack [reg] - removes the tracking of the given register
- (L)et [reg]/[addr] = [expr] - write to register or memory
- Breakpoints - show a list of breakpoints
- Insncount - count number of CPU cycles executed
- Manual - show the manual for this page

Scripting commands:

- #define name [commands] - alias "name" to run [commands].
- command;command - run first command, then second comamnd.

List of types:

- [reg] := 'r' followed by a number 0-15

[addr] := base-16 integer or label name (e.g., 'main')
[expr] := [reg] or [addr] or
[expr]+[expr] or [expr]-[expr]

WingsOfAltair - January 13, 2017, 12:31 pm

Thanks for the share. This will help me and others immensely.

Kitlith - January 13, 2017, 5:33 pm

Oh, microcorruption! That was a fun one. I haven't finished it, got to Chernobyl and got stuck, and haven't gotten around to working on it since then. Would highly recommend people to try it.

Lucrid - January 13, 2017, 5:46 pm

I'll have to give this a shot, thanks for the share!

Rada - January 14, 2017, 3:19 pm

Thanks, im going to try this out, as for those talking about ollydbg - do I have to download or are there challenges online like this one.