

Forum:[Hack Forums](#)

Thread: [Well I created a polymorphic .NET crypter.. What now?](#)

PhillyStyle - November 2, 2025, 11:44 pm

I created a polymorphic .NET crypter.

It uses dnlib to dynamically create the exe, adding stack neutral random code where it can.

If you don't include an option like install to APPDATA, or Add to Registry, then it completely omits the code for those methods.

It has 3 different payload embed methods (my favorite is Lots of Little Strings), and I assume it is pretty FUD right now. I haven't uploaded to VirusTotal cuz I don't want it detected, but I have ran it against at least 5 AVs and it is undetected on all including Windows Defender, which is most important in my opinion.

Anyways, the question now is what to do with it. Do I release it for free and allow it to get detected, or do I sell it, or do a crypting service? Give away randomly? IDK. What do you guys think I should do? Do you believe it is truly polymorphic? Is there something I should add before I release it? Let me know what you think.

BTW it injects into it's self, that is why it is .NET only.

jmark7 - November 3, 2025, 12:54 am

Its very simple, If you want to give something to the community, you can release for free.

but

If you like to earn a few bucks from it you can sell it, but selling will give you hassle of providing satisfactory service, if you fail you might find unhappy buyers and open dispute.

Or

You can release it for free and sell crypting service. it will increase satisfaction level in buyers if the crypter is good.

If you need any help selling it, you can PM me. I can help you out.

PhillyStyle - November 3, 2025, 1:31 am

(Nov 3, 2025 01:54 AM)jmark7 Wrote: [\[CITE\]](#) Its very simple, If you want to give something to the community, you can release for free.

but

If you like to earn a few bucks from it you can sell it, but selling will give you hassle of providing satisfactory service, if you fail you might find unhappy buyers and open dispute.

Or

You can release it for free and sell crypting service. it will increase satisfaction level in buyers if the crypter is good.

If you need any help selling it, you can PM me. I can help you out.

Yeah I really wanted to release it for free all along, I just feel bad doing that because I know it will get picked up by AV fairly fast that way. I really don't want to deal with buyers, I don't know if it would be worth the amount of money I would make off it.

jmark7 - November 3, 2025, 3:41 am

(Nov 3, 2025 02:31 AM)PhillyStyle Wrote: [\[CITE\]](#) Yeah I really wanted to release it for free all along, I just feel bad doing that because I know it will get picked up by AV fairly fast that way. I really don't want to deal with buyers, I don't know if it would be worth the amount of money I would make off it.

You can release it to selected Trusted users who will use it wisely. mainly dont scan on VT.

PhillyStyle - November 3, 2025, 3:59 am

(Nov 3, 2025 04:41 AM)jmark7 Wrote: [\[CITE\]](#) You can release it to selected Trusted users who will use it wisely. mainly dont scan on VT.

Yeah I considered that, and was close to doing that, but then I decided I would have to contend with people calling it a virus and stuff. I think I will just release it on github and call it a day. Then I can move on to other projects without dealing with this one forever.

jmark7 - November 3, 2025, 4:03 am

(Nov 3, 2025 04:59 AM)PhillyStyle Wrote: [\[CITE\]](#) Yeah I considered that, and was close to doing that, but then I decided I would have to contend with people calling it a virus and stuff. I think I will just release it on github and call it a day. Then I can move on to other projects without dealing with this one forever.

making it purely public will increase the risk of high detection. I dont think people will tell it virus if you share only source and no executable in it.

DramaKing - November 3, 2025, 4:53 am

I'm not sure where I read it now, but I remember a post saying that a lot of so-called 'polymorphic' crypters really aren't, so it will be interesting seeing if yours achieves true polymorphism. I think that metamorphic code is the one that is really rare, though.

You could look at this as a good portfolio piece.

PhillyStyle - November 3, 2025, 5:13 am

OK so as it stands right now, I think I will be giving it out source code only to up to 10 people with good standing in the community. That means you have to have good popularity and at least 50 posts to be considered. I still reserve the right to release it publicly at some point in time, but some of you will get first crack at it at least. It would probably be at least a month to 6 months before I might consider releasing it public.

DramaKing - November 3, 2025, 7:17 am

(Nov 3, 2025 06:13 AM)PhillyStyle Wrote: [\[CITE\]](#) OK so as it stands right now, I think I will be giving it out source code only to up to 10 people with good standing in the community. That means you have to have good popularity and at least 50 posts to be considered. I still reserve the right to release it publicly at some point in time, but some of you will get first crack at it at least. It would probably be at least a month to 6 months before I might consider releasing it public.

I'll take a slot.

possessed-client - November 3, 2025, 7:38 am

Does it actually work with detected payloads? What techniques other than obfuscation does it employ? Okay, I assume that it disables AMSI so the payload at least is guaranteed to pass static scans and run, what about bypassing behavioural detection, how does it convince AVs to not scan its process or ignore it due to somehow looking trusted?

PhillyStyle - November 3, 2025, 8:14 am

(Nov 3, 2025 08:38 AM)possessed-client Wrote: [\[CITE\]](#) Does it actually work with detected payloads? What techniques other than obfuscation does it employ? Okay, I assume that it disables AMSI so the payload at least is guaranteed to pass static scans and run, what about bypassing behavioural detection, how does it convince AVs to not scan its process or ignore it due to somehow looking trusted?

It doesn't employ any trickery other than polymorphism/obfuscation. It is really just simple process injection under the hood. It runs without being detected by Windows Defender at least. I just ran it with Xeno-rat to confirm it works. I don't know how it will perform with all AVs. Some might detect it in memory once ran, but I don't know how to prevent that.

ts1000 - November 12, 2025, 12:58 pm

can you describe shortly in what way it is polymorphic?

???????????? - November 12, 2025, 1:09 pm

(Nov 3, 2025 09:14 AM)PhillyStyle Wrote: [\[CITE\]](#) It doesn't employ any trickery other than polymorphism/obfuscation. It is really just simple process injection under the hood. It runs without being detected by Windows Defender at least. I just ran it with Xeno-rat to confirm it works. I don't know how it will perform with all AVs. Some might detect it in memory once ran, but I don't know how to prevent that.

If you think it is good and profitable and without issues then you should sell it. Make sure to have good support for all buyers to build reputation.

PhillyStyle - November 12, 2025, 6:45 pm

(Nov 12, 2025 01:58 PM)ts1000 Wrote: [\[CITE\]](#) can you describe shortly in what way it is polymorphic?

Here is an excerpt from the readme that comes with it...

What Makes It Polymorphic?

Well it is polymorphic, meaning it is programmatically different every time you build it. How it works, this program utilizes dnlib which is a library that can write .NET code on the opcode level. When we write the code to the output exe, there are places in the code that are stack neutral (there isn't really anything on the stack). In those places we can insert stack neutral code (randomized code that doesn't affect the stack). Adding the random code in these places makes it harder to get a signature on the program.

We can also do other things since we are writing the output with dnlib. We make it modular. Modular meaning we don't include things that don't need to be included. For instance.. if you are using the one managed resource payload embed method, the code for the other 2 embed methods is not included in the output executable.

We also do things such as randomizing the names of methods and delegates. We also encrypt strings where applicable. This all adds up to having the most polymorphic/obfuscated output that we can possibly create.

EliteCipher21 - November 13, 2025, 4:19 am

Please release it with commented source code so we can learn how to make one like this =)

possessed-client - November 13, 2025, 7:51 am

(Nov 3, 2025 09:14 AM)PhillyStyle Wrote: [\[CITE\]](#) It doesn't employ any trickery other than polymorphism/obfuscation. It is really just simple process injection under the hood. It runs without being detected by Windows Defender at least. I just ran it with Xeno-rat to confirm it works. I don't know how it will perform with all AVs. Some might detect it in memory once ran, but I don't know

how to prevent that.

Oh yeah, have you tried your crypter and the common RATs against free Avast? I can get them past Avast scantime and runtime, but only briefly, after a few connections or minutes Avast usually detects them with Behaviour IDP detections, which I'm assuming are largely networking signatures + general heuristic weighing.

I was suggested by benio2 to run the process with flags to block non-Microsoft DLL loading, haven't tried yet, but I have my doubts, given that I've tried indirect syscalls with local and remote APC injection, ETW patching and a couple unhooking techniques, which should technically deal with userland DLL injections, but no luck, so might be that the data is largely collected at kernel level and then heuristically generating the behavioural detections based on it.

ts1000 - November 14, 2025, 10:26 am

If its that good you should start a service with it. There is a lot of people that need this kind of service

DramaKing - November 16, 2025, 4:02 pm

(Nov 3, 2025 08:38 AM)possessed-client Wrote: [\[CITE\]](#) Does it actually work with detected payloads? What techniques other than obfuscation does it employ? Okay, I assume that it disables AMSI so the payload at least is guaranteed to pass static scans and run, what about bypassing behavioural detection, how does it convince AVs to not scan its process or ignore it due to somehow looking trusted?

I just had something to test it on. I tested it with a Pulsar payload. It's detected by WD as soon as it touches the disk.