

Forum:[Hack Forums](#)

Thread: [where to learn cryptography](#)

arav - August 22, 2024, 11:32 am

Does anyone know a good source (youtube channel, books etc... to learn how to encrypt programs, perhaps to be able to create a personal crypter?

__main__ - August 22, 2024, 2:00 pm

(Aug 22, 2024 11:32 AM)arav Wrote: [\[CITE\]](#) Does anyone know a good source (youtube channel, books etc... to learn how to encrypt programs, perhaps to be able to create a personal crypter? Cryptography isn't exactly the same as thing as crypting exes.

But anyway, here is my advice. Despite having lots of experience in C programming and the Windows API, it still took quite some time to finish my own runtime crypter. Since the field of "crypting" is somewhat esoteric and not covered in mainstream books or courses, you will have to rely heavily on trial and error to discover what works best.

For memory mapping techniques, I recommend taking a look at the source code for UPX. It doesn't matter that UPX is a file packer and not a crypter, as the same mapping techniques apply. By reading the source code, you can, for example, learn how to memory map the executable and how to use .BSS segments to dynamically auto-allocate space for different parts of the payload (just to name one example). UPX is a beast when it comes to running virtually anything in memory.

arav - August 23, 2024, 4:50 am

(Aug 22, 2024 02:00 PM)__main__ Wrote: [\[CITE\]](#) Cryptography isn't exactly the same as thing as crypting exes.

But anyway, here is my advice. Despite having lots of experience in C programming and the Windows API, it still took quite some time to finish my own runtime crypter. Since the field of "crypting" is somewhat esoteric and not covered in mainstream books or courses, you will have to rely heavily on trial and error to discover what works best.

For memory mapping techniques, I recommend taking a look at the source code for UPX. It doesn't matter that UPX is a file packer and not a crypter, as the same mapping techniques apply. By reading the source code, you can, for example, learn how to memory map the executable and how to use .BSS segments to dynamically auto-allocate space for different parts of the payload (just to name one example). UPX is a beast when it comes to running virtually anything in memory.
thanks you very much

b0r_ealis - August 24, 2024, 7:48 am

(Aug 22, 2024 02:00 PM)__main__ Wrote: [\[CITE\]](#) Cryptography isn't exactly the same as thing as crypting exes.

But anyway, here is my advice. Despite having lots of experience in C programming and the Windows API, it still took quite some time to finish my own runtime crypter. Since the field of "crypting" is somewhat esoteric and not covered in mainstream books or courses, you will have to rely heavily on trial and error to discover what works best.

For memory mapping techniques, I recommend taking a look at the source code for UPX. It doesn't

matter that UPX is a file packer and not a crypter, as the same mapping techniques apply. By reading the source code, you can, for example, learn how to memory map the executable and how to use .BSS segments to dynamically auto-allocate space for different parts of the payload (just to name one example). UPX is a beast when it comes to running virtually anything in memory.

On a personal level, vouching for trial and error / personal experience is probably my best bet on going about this. What I have in mind has been said, so as was said, get into it yourself. Experience is key.

DramaKing - August 24, 2024, 8:27 am

Encrypting a PE and decoder stub is easy. The runtime evasion techniques that still work take real work.

arav - August 24, 2024, 3:49 pm

(Aug 24, 2024 08:27 AM)DramaKing Wrote: [\[CITE\]](#) Encrypting a PE and decoder stub is easy. The runtime evasion techniques that still work take real work.

everything is easy if you know where to learn... that's the problem

PureCoder - August 26, 2024, 7:29 am

Assuming you already have coding experience, this website (<https://maldevacademy.com>) will provide detailed insights into crypting techniques, along with practical code examples.