

# Forum:[Hack Forums](#)

Thread: [nodes-crypter | Javascript C++ Crypter | tmp file child process run method | npm pkg](#)

Hackers - May 26, 2024, 6:50 pm

nodes-crypter | npm package | C++ Javascript Crypter

Today you will learn how to take a C/C++ program, encrypt it with XOR and then execute it, all using Javascript with this easy-to-use npm package. Take your static or dynamic compiled program and protect its runtime by utilizing a tmp file and run it in a child process, all from a JS file! You heard that right - your program's encrypted binary will be stored inside of a final script "run.js", which you can then run with "node run.js" to inject the embedded C/C++ program and run it as normal.

Get it now

Code npm install -g nodes-crypter

Tested on Linux & Windows

run command: nodes-crypter ./original-cpp-program secretKeyPhraseHere

nodes-crypter will generate a file: run.js

run command: node run.js

your original C++ program will be decrypted (inside of run.js) and run as a program

<https://www.npmjs.com/package/nodes-crypter/v/1.0.6>

This is an example of how you can use JS in unique ways due to npm & node. This method of encrypting and then decrypting, pushing to a tmp file & running as a child process is a proof of concept. You are free to learn from it, use it for personal use, and enhance the codebase - within limits of its license & npm TOS.

The main file is node-crypter's only bin script, which encrypts your original program using XOR.

```
Code const xorEncrypt = (input, key) => {  
  const keyBuffer = Buffer.from(key);  
  const output = Buffer.alloc(input.length);  
  for (let i = 0; i < input.length; i++) {  
    output[i] = input[i] ^ keyBuffer[i % keyBuffer.length];  
  }  
  return output;  
};
```

Similarly, the final run.js file has a decrypt function:

```
Code const xorDecrypt = (input, key) => {  
  const keyBuffer = Buffer.from(key);  
  const output = Buffer.alloc(input.length);  
  for (let i = 0; i < input.length; i++) {  
    output[i] = input[i] ^ keyBuffer[i % keyBuffer.length];  
  }  
  return output;  
};
```

The run.js file will decrypt the program (encrypted binary held inside of run.js) & push it to a tmp file:  
Code `const tempPath = path.join('/tmp', 'decrypted_program');`

```
fs.writeFileSync(tempPath, decryptedData);  
fs.chmodSync(tempPath, '755');
```

Then, it runs the tmp file using the `exec()` function.

This is a basic method, mostly meant to highlight the capabilities of using an npm package to handle the "build process", and using node to handle the runtime. You can always use other runtime methods instead, and add other build steps too.

I hope you enjoyed this & if you learned anything, please say a thank you below. If you have any questions or suggestions, feel free to reply!!

7 - May 26, 2024, 7:03 pm

Beautiful mate.

Great job, I really need to get back into this area of field.

Will test this out in the near future and add some exclusions and features

**Klonopin** - May 27, 2024, 6:41 am

Awesome release brother, keep up with the high-quality posts.

You should provide a before and after of the crypt.

**Hackers** - May 27, 2024, 7:09 am

nodes-crypter v1.0.6 is released

Added support for Windows (.exe extension & temp folder pathing), tested with a 64bit program compiled with g++

(May 26, 2024 07:03 PM)NulldB Wrote: [\[CITE\]](#) Beautiful mate.

Great job, I really need to get back into this area of field.

Will test this out in the near future and add some exclusions and features

Thanks null, much appreciated!

It's fun to play with these things and see different methods for running programs

(May 27, 2024 06:41 AM)Klonopin Wrote: [\[CITE\]](#) Awesome release brother, keep up with the high-quality posts.

You should provide a before and after of the crypt.

Thanks! :)

Here are some stats:

Embedded in run.js file: 0/64 on VT

Embedded in run.js: 0/40 on kleenscan

Keep in mind that this is just based on those tests & actual runtime can be detected due to the decrypted program being pushed to a tmp file and executed. Further work would be needed, but for protecting your personal programs in scantime this method works 100% of the time.

versus:

Regular / unencrypted: 42/72 on VT

Regular (scantime): 12/40 on Kleenscan

**Hackers** - May 27, 2024, 5:45 pm

(May 27, 2024 12:46 PM)blyat1337 Wrote: [\[CITE\]](#) This does not bypass any antivirus in runtime so I am not sure if a crypter is a suitable name, maybe detected dropper?  
Definitely, that's a better one for sure

**Tsar** - May 28, 2024, 12:46 pm

Wow very hq and informative as always, well done

**Absence** - May 28, 2024, 1:07 pm

Love seeing the work you put in, [Hackers](#) it is awesome!

Keep it up.

**Hackers** - May 28, 2024, 1:47 pm

(May 28, 2024 12:46 PM)Rapid Wrote: [\[CITE\]](#) Wow very hq and informative as always, well done thanks Rapid, just figured I'd give it a go & see.

(May 28, 2024 01:07 PM)Absence Wrote: [\[CITE\]](#) Love seeing the work you put in, [Hackers](#) it is awesome!

Keep it up.

Thank you Absence, much appreciated!

**secret\_code** - May 30, 2024, 8:48 am

nice work, the method is good, but it is ineffective in real world scenarios, it will fail for 2 reasons:

- 1) JavaScript is easy to de-obfuscate and reverse-engineer because it remains in source code form, even when minified or obfuscated.
- 2) c++ will be quickly detected if uses low level system calls as those are the most monitored by AVs

using xor for your crypter is HIGHLY discouraged for a LOT of reasons, one being "easy detection", xor produces very predictable patterns, during static analysis the av detection system will easily detect those patterns, one other reason is, if the decryption key is short and easy, it will be a matter of 1000 nanoseconds for the av to reverse the encryption, in other words, xor by itself is very weak, is just walk in the park for av tools to decrypt, instead, use more secure and strong encryption algorithms like AES or chacha20 which is also a good option for both security and performance, finally, LotL is a VERY good approach also...

Your approach is GOOD, you got the point, but unfortunately it will not work IRL scenarios, you seem to be learning AV detection right? how long have you been into it?

**Hackers** - May 30, 2024, 8:54 am

(May 30, 2024 08:48 AM)secret\_code Wrote: [\[CITE\]](#) nice work, the method is good, but it is ineffective in real world scenarios, it will fail for 2 reasons:

- 1) JavaScript is easy to de-obfuscate and reverse-engineer because it remains in source code form, even when minified or obfuscated.

2) c++ will be quickly detected if uses low level system calls as those are the most monitored by AVs

using xor for your crypter is HIGHLY discouraged for a LOT of reasons, one being "easy detection", xor produces very predictable patterns, during static analysis the av detection system will easily detect those patterns, one other reason is, if the decryption key is short and easy, it will be a matter of 1000 nanoseconds for the av to reverse the encryption, in other words, xor by itself is very weak, is just walk in the park for av tools to decrypt, instead, use more secure and strong encryption algorithms like AES or chacha20 which is also a good option for both security and performance, finally, LotL is a VERY good approach also...

Your approach is GOOD, you got the point, but unfortunately it will not work IRL scenarios, you seem to be learning AV detection right? how long have you been into it?

yes, agreed on all points. AES 256 > XOR, runtime injection > tmp file drop & fun

To be honest the point of this was just to show how it could be done. I mostly just wanted to have a node module calling a child process and injecting encrypted binary. Once I finished that I closed it up, published, shared here and then came back to support Windows. I'm not really trying to get FUD or anything here, just show that it's possible to have a JS based build tool, and a JS based run method. It started as a Linux only tool and I really just wanted to create a POC.

**secret\_code** - May 30, 2024, 9:00 am

(May 30, 2024 08:54 AM)nodes Wrote: [\[CITE\]](#) yes, agreed on all points. AES 256 > XOR, runtime injection > tmp file drop & fun

To be honest the point of this was just to show how it could be done. I mostly just wanted to have a node module calling a child process and injecting encrypted binary. Once I finished that I closed it up, published, shared here and then came back to support Windows. I'm not really trying to get FUD or anything here, just show that it's possible to have a JS based build tool, and a JS based run method. It started as a Linux only tool and I really just wanted to create a POC.

your method was good at some point, but as i said, it will never work IRL, but im sure you're on the right way, just remember never to use xor for anything, it is VERY predictable and easy to reverse, but that was a nice post to read anyway, so good job :)

**Hackers** - May 30, 2024, 9:19 am

(May 30, 2024 09:00 AM)secret\_code Wrote: [\[CITE\]](#) your method was good at some point, but as i said, it will never work IRL, but im sure you're on the right way, just remember never to use xor for anything, it is VERY predictable and easy to reverse, but that was a nice post to read anyway, so good job :)

Yeah, I understand. This is not my only project like this, but also not really working on crypters anymore. I had a random urge to try this out but overall I have no plans to continue researching this topic. There's not enough money in it to justify the level of effort and risk, plus I am whitehat.

**secret\_code** - May 30, 2024, 9:30 am

(May 30, 2024 09:19 AM)nodes Wrote: [\[CITE\]](#) Yeah, I understand. This is not my only project like this, but also not really working on crypters anymore. I had a random urge to try this out but overall I have no plans to continue researching this topic. There's not enough money in it to justify the level of effort and risk, plus I am whitehat.

well there's no need for extra money to do that, just serious dedication to it

**Hackers** - May 30, 2024, 11:17 am

(May 30, 2024 09:30 AM)secret\_code Wrote: [\[CITE\]](#) well there's no need for extra money to do that, just serious dedication to it

true, I did all of my research into it for \$0. I mean more like the payoff for having a FUD crypter is

worth far less than any other use of my time. It's not all about money, but I've lost interest in crypters in general so the only other incentive remaining is \$ and it just doesn't make sense to dedicate so much time to something that won't have that much of a positive benefit for me.

**secret\_code** - May 31, 2024, 6:47 am

(May 30, 2024 11:17 AM)nodes Wrote: [\[CITE\]](#) true, I did all of my research into it for \$0. I mean more like the payoff for having a FUD crypter is worth far less than any other use of my time. It's not all about money, but I've lost interest in crypters in general so the only other incentive remaining is \$ and it just doesn't make sense to dedicate so much time to something that won't have that much of a positive benefit for me.

nice job anyway my man, crypters are very big concept, im not surprised you've lost interest in it, what is your background anyway if i may ask? im new to the forum

**Hackers** - May 31, 2024, 6:54 am

(May 31, 2024 06:47 AM)secret\_code Wrote: [\[CITE\]](#) nice job anyway my man, crypters are very big concept, im not surprised you've lost interest in it, what is your background anyway if i may ask? im new to the forum

I make websites & apps & software. I got interested in this topic when seeing what others were doing, learning more about it, and then trying to make some of my own. I am still learning C++ and started building them with that. I've still never actually used a crypter, not even my own! It was all just proof of concept / whitehat program encryption & decryption for programming education.

**secret\_code** - May 31, 2024, 7:03 am

(May 31, 2024 06:54 AM)nodes Wrote: [\[CITE\]](#) I make websites & apps & software. I got interested in this topic when seeing what others were doing, learning more about it, and then trying to make some of my own. I am still learning C++ and started building them with that. I've still never actually used a crypter, not even my own! It was all just proof of concept / whitehat program encryption & decryption for programming education.

you r learning c++? why?

**Hackers** - May 31, 2024, 7:13 am

(May 31, 2024 07:03 AM)secret\_code Wrote: [\[CITE\]](#) you r learning c++? why?  
for fun and also to extend my capabilities.

I like how fast it is, cross-platform, and can do a LOT with C++

**secret\_code** - May 31, 2024, 7:28 am

(May 31, 2024 07:13 AM)nodes Wrote: [\[CITE\]](#) for fun and also to extend my capabilities.

I like how fast it is, cross-platform, and can do a LOT with C++

keep going brother, if you want a tip, i can tell you that the fastest way to learn c++ write a simple jabber, at the end, you will see how you learned both c++ core libraries, networking(sockets), multi-threading, cryptography(for secure communication) and everything else... try it out if you still want to learn c++ and you're not comfortable with the above concepts yet...

**pinguoin** - June 5, 2024, 11:16 am

Waww, still amazing work that amazing.

Thanks lots for your work in here.

Hope the best for your goal what ever it can be.

**Hackers** - June 5, 2024, 11:48 am

(Jun 5, 2024 11:16 AM)pinguoin Wrote: [\[CITE\]](#) Waww, still amazing work that amazing.

Thanks lots for your work in here.

Hope the best for your goal what ever it can be.

thanks, appreciate it!

Lemme know if you have any questions - feel free to take the source and spin it off into something else. You just install w/ npm & then go to node\_modules and you can grab source, or you can get it on npmjs

goal: learn more, share knowledge :)

**SIM-5457022** - June 8, 2024, 11:02 am

(May 30, 2024 08:54 AM)nodes Wrote: [\[CITE\]](#) yes, agreed on all points. AES 256 > XOR, runtime injection > tmp file drop & fun

To be honest the point of this was just to show how it could be done. I mostly just wanted to have a node module calling a child process and injecting encrypted binary. Once I finished that I closed it up, published, shared here and then came back to support Windows. I'm not really trying to get FUD or anything here, just show that it's possible to have a JS based build tool, and a JS based run method. It started as a Linux only tool and I really just wanted to create a POC.

I give you a hug bro... keep on learning. The key to a good software engineer is wanting to keep learning more on how to do things In a better way.

No software engineer or programmer ever stops learning new things. I jus think your projects are interesting. Keep up the good work.

Eventually you will get to a point where you can make it FUD.

**Hackers** - June 8, 2024, 12:56 pm

(Jun 8, 2024 11:02 AM)SIM-5457022 Wrote: [\[CITE\]](#) I give you a hug bro... keep on learning. The key to a good software engineer is wanting to keep learning more on how to do things In a better way.

No software engineer or programmer ever stops learning new things. I jus think your projects are interesting. Keep up the good work.

Eventually you will get to a point where you can make it FUD.

Yeah, exactly, I hope these threads are helpful to people.

Appreciate it!

I probably will not go for FUD ever - I am not trying to make anything fud, this was just for fun on a whim

I'm completely whitehat so don't really have a need for it, but the encryption, decryption, memory management, processes, etc are useful for other non-crypter projects too so it is worth it to play around with.

**SIM-5457022** - June 9, 2024, 8:52 am

(Jun 8, 2024 12:56 PM)nodes Wrote: [\[CITE\]](#) Yeah, exactly, I hope these threads are helpful to people.

Appreciate it!

I probably will not go for FUD ever - I am not trying to make anything fud, this was just for fun on a whim

I'm completely whitehat so don't really have a need for it, but the encryption, decryption, memory management, processes, etc are useful for other non-crypter projects too so it is worth it to play around with.

There's no need for FUD for whitehat but if you can do it you can use the FUD to make it harder for people to steal your source code.

It can help protect software from being pirated and can be used for other things like making communications more secure.

I just am interested on seeing what others are doing and what they learned. Have you seen the youtuber micheal reeves?

I like watching him because he builds stuff even though the stuff he makes is not practical to be used. He builds it just to learn and to see if it can be built. Most innovations happens by mistakenly finding them but doing random projects.

graphene was found out by mistake. 2 russian guys were drunk and they used graphite and ran electrical charges thru them. This was back in 2002 and yet now we are working on trying to get it in commercial use for eletronics. If we can mass produce it cheaply. We will make a major change in eletronics and computers.

No need for fans or anything like that because the electrical currents flows as a wave which means no resistance which means no heat production. It all started by people just putting a charge to graphite. They later made it a paste and saw potential.

This applies with any thing you do wheatear it's electronics or any field. Just messing around on your free time you can learn more than what a book can teach.

This is the art of hacking and hacking isn't always for bad or evil or dark intentions. Hacking is the art of taking something that is used for one purpose an finding how you cane change it or modify it to be used for other purposes.

I really do like your posts and have been watching your projects. Just interesting to see what someone can do with creative thinking.

**Uolcan** - June 9, 2024, 10:08 am

(May 26, 2024 06:50 PM)nodes Wrote: [\[CITE\]](#) nodes-crypter | npm package | C++ Javascript Crypter

Today you will learn how to take a C/C++ program, encrypt it with XOR and then execute it, all using Javascript with this easy-to-use npm package. Take your static or dynamic compiled program and protect its runtime by utilizing a tmp file and run it in a child process, all from a JS file! You heard that right - your program's encrypted binary will be stored inside of a final script "run.js", which you can then run with "node run.js" to inject the embedded C/C++ program and run it as normal.

Get it now

Code npm install -g nodes-crypter

Tested on Linux & Windows

run command: nodes-crypter ./original-cpp-program secretKeyPhraseHere

nodes-crypter will generate a file: run.js

run command: node run.js

your original C++ program will be decrypted (inside of run.js) and run as a program

<https://www.npmjs.com/package/nodes-crypter/v/1.0.6>

This is an example of how you can use JS in unique ways due to npm & node. This method of encrypting and then decrypting, pushing to a tmp file & running as a child process is a proof of concept. You are free to learn from it, use it for personal use, and enhance the codebase - within limits of its license & npm TOS.

The main file is node-crypter's only bin script, which encrypts your original program using XOR.

```
Code const xorEncrypt = (input, key) => {  
  const keyBuffer = Buffer.from(key);  
  const output = Buffer.alloc(input.length);  
  for (let i = 0; i < input.length; i++) {  
    output[i] = input[i] ^ keyBuffer[i % keyBuffer.length];  
  }  
  return output;  
};
```

Similarly, the final run.js file has a decrypt function:

```
Code const xorDecrypt = (input, key) => {  
  const keyBuffer = Buffer.from(key);  
  const output = Buffer.alloc(input.length);  
  for (let i = 0; i < input.length; i++) {  
    output[i] = input[i] ^ keyBuffer[i % keyBuffer.length];  
  }  
  return output;  
};
```

The run.js file will decrypt the program (encrypted binary held inside of run.js) & push it to a tmp file:

```
Code const tempPath = path.join('/tmp', 'decrypted_program');
```

```
fs.writeFileSync(tempPath, decryptedData);  
fs.chmodSync(tempPath, '755');
```

Then, it runs the tmp file using the exec() function.

This is a basic method, mostly meant to highlight the capabilities of using an npm package to handle the "build process", and using node to handle the runtime. You can always use other runtime methods instead, and add other build steps too.

I hope you enjoyed this & if you learned anything, please say a thank you below. If you have any questions or suggestions, feel free to reply!!

wow man this is so cool you can even make the binary code run inside memory rather than run on disk

nice share

**Hackers** - June 9, 2024, 10:38 am

(Jun 9, 2024 08:52 AM)SIM-5457022 Wrote: [\[CITE\]](#) There's no need for FUD for whitehat but if you can do it you can use the FUD to make it harder for people to steal your source code.

It can help protect software from being pirated and can be used for other things like making communications more secure.

I just am interested on seeing what others are doing and what they learned. Have you seen the youtuber micheal reeves?

I like watching him because he builds stuff even though the stuff he makes is not practical to be used. He builds it just to learn

and to see if it can be built. Most innovations happens by mistakenly finding them but doing random projects.

graphene was found out by mistake. 2 russian guys were drunk and they used graphite and ran electrical charges thru them. This was back in 2002 and yet now we are working on trying to get it in commercial use for eletronics. If we can mass produce it cheaply. We will make a major change in

electronics and computers.

No need for fans or anything like that because the electrical currents flows as a wave which means no resistance which means no heat production. It all started by people just putting a charge to graphite. They later made it a paste and saw potential.

This applies with any thing you do wheatear it's electronics or any field. Just messing around on your free time you can learn more than what a book can teach.

This is the art of hacking and hacking isn't always for bad or evil or dark intentions. Hacking is the art of taking something that is used for one purpose an finding how you cane change it or modify it to be used for other purposes.

I really do like your posts and have been watching your projects. Just interesting to see what someone can do with creative thinking.

This is true, it is a useful skill for cybersecurity and understanding the systems better.

I haven't heard of that YT but I don't really follow many - I'll check it out.

Definitely like the sound of it though - it's fun to just build to learn, explore, etc. That's usually what I'm doing but I try to knock out certain features, and then build on old programs or bring them into future projects.

(Jun 9, 2024 10:08 AM)Uolcan Wrote: [\[CITE\]](#) wow man this is so cool you can even make the binary code run inside memory rather than run on disk

nice share

Appreciate that, thank you!

I also made a Python package that does the same as here, using 'pip install snake-crypter' and then generating a new .py script that will run your original binary :)

<https://hackforums.net/showthread.php?tid=6272902>

**secret\_code** - June 15, 2024, 7:16 am

still trying to figure out why you're using root mode?

**Hackers** - June 15, 2024, 10:00 am

(Jun 15, 2024 07:16 AM)secret\_code Wrote: [\[CITE\]](#) still trying to figure out why you're using root mode?

I have a few systems and VMs that aren't connected to anything and I just use them for testing

**Hackers** - January 31, 2025, 7:24 am

If anyone has any questions just LMK!

**Stanley** - March 16, 2026, 8:26 pm

I'm Stanley HF's helper bot. The OP (original poster) of this thread wants everyone to see this page and has elected to bump it to the top of the showthread listing. Since HF doesn't want a ton of LQP (Low Quality Posts) this is the best method of bumping your content. There is a small ?yte fee for each bump. Any OP can bump any of their threads on HF from the linked icon on the top of their first post.

If you haven't already read the first post of this thread we encourage you to do so.

Thanks,  
Stanley

(May 26, 2024 06:50 PM)Hackers Wrote: [\[CITE\]](#) nodes-crypter | npm package | C++ Javascript Crypter

Today you will learn how to take a C/C++ program, encrypt it with XOR and then execute it, all using Javascript with this easy-to-use npm package. Take your static or dynamic compiled program and protect its runtime by utilizing a tmp file and run it in a child process, all from a JS file! You heard that right - your program's encrypted binary will be stored inside of a final script "run.js", which you can then run with "node run.js" to inject the embedded C/C++ program and run it as normal.

Get it now

Code npm install -g nodes-crypter

Tested on Linux & Windows

run command: nodes-crypter ./original-cpp-program secretKeyPhraseHere

nodes-crypter will generate a file: run.js

run command: node run.js

your original C++ program will be decrypted (inside of run.js) and run as a program

<https://www.npmjs.com/package/nodes-crypter/v/1.0.6>

This is an example of how you can use JS in unique ways due to npm & node. This method of encrypting and then decrypting, pushing to a tmp file & running as a child process is a proof of concept. You are free to learn from it, use it for personal use, and enhance the codebase - within limits of its license & npm TOS.

The main file is node-crypter's only bin script, which encrypts your original program using XOR.

```
Code const xorEncrypt = (input, key) => {  
  const keyBuffer = Buffer.from(key);  
  const output = Buffer.alloc(input.length);  
  for (let i = 0; i < input.length; i++) {  
    output[i] = input[i] ^ keyBuffer[i % keyBuffer.length];  
  }  
  return output;  
};
```

Similarly, the final run.js file has a decrypt function:

```
Code const xorDecrypt = (input, key) => {  
  const keyBuffer = Buffer.from(key);  
  const output = Buffer.alloc(input.length);  
  for (let i = 0; i < input.length; i++) {  
    output[i] = input[i] ^ keyBuffer[i % keyBuffer.length];  
  }  
  return output;  
};
```

The run.js file will decrypt the program (encrypted binary held inside of run.js) & push it to a tmp file:

```
Code const tempPath = path.join('/tmp', 'decrypted_program');
```

```
fs.writeFileSync(tempPath, decryptedData);  
fs.chmodSync(tempPath, '755');
```

Then, it runs the tmp file using the exec() function.

This is a basic method, mostly meant to highlight the capabilities of using an npm package to handle the "build process", and using node to handle the runtime. You can always use other runtime methods instead, and add other build steps too.

I hope you enjoyed this & if you learned anything, please say a thank you below. If you have any questions or suggestions, feel free to reply!!

Windows Defender is detecting a threat; it's not passing through the smart app. Fix it now.

**Hackers** - March 18, 2026, 12:37 pm

(Mar 18, 2026 12:18 PM)kriminaltetic Wrote: [\[CITE\]](#) Windows Defender is detecting a threat; it's not passing through the smart app. Fix it now.

This is a starter kit that is not intended to bypass anything, it's a loader with encrypted program being run.

Read my note from original post:

"This is a basic method, mostly meant to highlight the capabilities of using an npm package to handle the "build process", and using node to handle the runtime. You can always use other runtime methods instead, and add other build steps too."

**DramaKing** - March 19, 2026, 5:24 am

(Mar 18, 2026 12:18 PM)kriminaltetic Wrote: [\[CITE\]](#) Windows Defender is detecting a threat; it's not passing through the smart app. Fix it now.

You're being really presumptuous of a free proof-of-concept method that isn't meant to be FUD.

**kriminaltetic** - March 19, 2026, 6:59 am

(Mar 19, 2026 05:24 AM)DramaKing Wrote: [\[CITE\]](#) You're being really presumptuous of a free proof-of-concept method that isn't meant to be FUD.

Dude, are you serious? Why did I get a -3? And I'm going to report this situation. I don't understand why it led to me getting a -3.

**DramaKing** - March 19, 2026, 7:13 am

(Mar 19, 2026 06:59 AM)kriminaltetic Wrote: [\[CITE\]](#) Dude, are you serious? Why did I get a -3? And I'm going to report this situation. I don't understand why it led to me getting a -3.

You do that.

**Hackers** - March 19, 2026, 7:29 am

(Mar 19, 2026 06:59 AM)kriminaltetic Wrote: [\[CITE\]](#) Dude, are you serious? Why did I get a -3? And I'm going to report this situation. I don't understand why it led to me getting a -3.

Shut the fuck up, enjoy your -12 now for -15 total. I had even given you +1 for a valid bug report, but now you are just being obnoxious.

Feel free to report me for giving you negative popularity. You could even make a community alert and see what people think.

**kriminaltetic** - March 19, 2026, 8:13 am

(Mar 19, 2026 07:29 AM)Hackers Wrote: [\[CITE\]](#) Shut the fuck up, enjoy your -12 now for -15 total. I had even given you +1 for a valid bug report, but now you are just being obnoxious.

Feel free to report me for giving you negative popularity. You could even make a community alert and see what people think.

Hey friend, if you could check the time interval? If you check when I replied to the thread, you'll understand; it's not a problem, I won't take -15 points.

**Hackers** - April 14, 2026, 8:31 am

Happy to help people understand how this all works. This was fun because at the time nobody was really taking JS-based crypts seriously and I wanted to prove how it could be done using a node runtime.

**endless1337** - May 6, 2026, 2:41 pm

Thank you brother

**NomosopH** - May 13, 2026, 5:15 am

good job