

Forum:[Hack Forums](#)

Thread: [snake-crypter | Python Builder / Stub / Run | Easy to use | pip install snake-crypter](#)

Hackers - June 8, 2024, 5:58 pm

snake-crypter | pip package | C/C++ to Python Builder & Process Run Method

Today you will learn how to take your original, compiled program, encrypt it with XOR and then execute it, using Python with this easy-to-use pip package. Take your static or dynamic compiled program and protect it in a Python subprocess! You simply install the package, use it to encrypt your original program, which will create a python file with your program embedded inside. You can then run this script/program on any system or CLI that supports Python.

Spoiler ([Click to View](#))
old Fallen easter egg ;)

Get it now
Code pip install snake-crypter

Tested on Windows
run command: snake-crypter your_executable_file your_key

snake-crypter will generate a file: tmp.py

run command: python tmp.py , or run it however you wish.

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

<https://pypi.org/project/snake-crypter>

The main file is the builder's encryptor.py script that will encrypt the program with XOR, and then save the tmp file with a stub template, putting the encrypted data into it alongside the decryption key. This next part is very important: This is a demo codebase. For true security, you should clone the project and modify the template in encryptor.py ([download the source / view files & hashes](#))

```
Code import os
import sys
from tempfile import NamedTemporaryFile
import subprocess
```

```
def xor_encrypt(input_bytes, key):
    key_bytes = key.encode()
```

```

output_bytes = bytearray(input_bytes)
for i in range(len(output_bytes)):
    output_bytes[i] ^= key_bytes[i % len(key_bytes)]
return bytes(output_bytes)

def generate_run_script(encrypted_data, key):
    from tempfile import NamedTemporaryFile, mkstemp
    import os
    import subprocess

    # Convert encrypted data to hexadecimal for embedding in the Python script
    encrypted_hex = "".join(format(x, '02x') for x in encrypted_data)

    # Define the content of the Python script that will decrypt and execute the data
    script_content = f"""import os, subprocess
from tempfile import mkstemp

# Key for decryption
key = '{key}'

# Encrypted data in hexadecimal
encrypted_hex = '{encrypted_hex}'

# Convert hex to bytes
encrypted_data = bytes.fromhex(encrypted_hex)

# Decrypt data
decrypted_data = bytearray(encrypted_data)
for i in range(len(decrypted_data)):
    decrypted_data[i] ^= ord(key[i % len(key)])

# Create a new temporary file for the decrypted executable
fd, path = mkstemp(suffix='.out')
os.close(fd) # Close the file descriptor immediately

# Write the decrypted data to the file
with open(path, 'wb') as f:
    f.write(decrypted_data)
os.chmod(path, 0o755)

# Execute the decrypted file
subprocess.run([path])

# Clean up the temporary file after execution
os.unlink(path)
"""

    # Write the script content to a named temporary Python file
    with NamedTemporaryFile(delete=False, suffix='.py', mode='w') as tmp_file:
        tmp_file.write(script_content)
        os.chmod(tmp_file.name, 0o755)

    # Return the path of the temporary Python script

```

```
return tmp_file.name
```

```
def encrypt_file(input_path, key):  
    with open(input_path, 'rb') as f:  
        input_data = f.read()  
        encrypted_data = xor_encrypt(input_data, key)  
        run_script_path = generate_run_script(encrypted_data, key)  
        print(f'Run script generated at {run_script_path}')
```

```
if __name__ == "__main__":  
    if len(sys.argv) != 3:  
        print("Usage: python encryptor.py <input_file> <key>", file=sys.stderr)  
        sys.exit(1)  
    encrypt_file(sys.argv[1], sys.argv[2])
```

This is a basic method, mostly meant to highlight the capabilities of using a pip package to handle the "build process", and using Python to handle the runtime. You can always use other methods instead, and add other build steps too if you wish to advance this further. Best of luck & reply in thread if you have any questions. I do not offer any sort of private crypts or support.

Ren And Stimpy - June 10, 2024, 4:41 pm

Nice share mate! Always giving back to the community!

Hackers - June 11, 2024, 3:44 am

(Jun 10, 2024 04:41 PM)Ren And Stimpy Wrote: [\[CITE\]](#) Nice share mate! Always giving back to the community!

thanks, that's the goal!

Hope it helps someone learn about different solutions. This is a great way for example to roll out new features & updates or to package in their other programs in general.

7 - June 11, 2024, 4:20 am

very beautiful share mate!

Great job! Will be testing this out

Hackers - June 11, 2024, 4:50 am

(Jun 11, 2024 04:20 AM)NulldB Wrote: [\[CITE\]](#) very beautiful share mate!

Great job! Will be testing this out

Thanks I appreciate it!

Just keep in mind this is basically just an example package, but source can be easily seen & modified to create a custom package.

Py_Dev - June 13, 2024, 3:53 pm

(Jun 8, 2024 05:58 PM)nodes Wrote: [\[CITE\]](#) snake-crypter | pip package | C/C++ to Python Builder & Process Run Method

Today you will learn how to take your original, compiled program, encrypt it with XOR and then execute it, using Python with this easy-to-use pip package. Take your static or dynamic compiled program and protect it in a Python subprocess! You simply install the package, use it to encrypt your original program, which will create a python file with your program embedded inside. You can then run this script/program on any system or CLI that supports Python.

Get it now

Code pip install snake-crypter

Tested on Windows

run command: snake-crypter your_executable_file your_key

snake-crypter will generate a file: tmp.py

run command: python tmp.py , or run it however you wish.

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

<https://pypi.org/project/snake-crypter>

The main file is the builder's encryptor.py script that will encrypt the program with XOR, and then save the tmp file with a stub template, putting the encrypted data into it alongside the decryption key. This next part is very important: This is a demo codebase. For true security, you should clone the project and modify the template in encryptor.py ([download the source / view files & hashes](#))

Code import os

import sys

from tempfile import NamedTemporaryFile

import subprocess

```
def xor_encrypt(input_bytes, key):
    key_bytes = key.encode()
    output_bytes = bytearray(input_bytes)
    for i in range(len(output_bytes)):
        output_bytes[i] ^= key_bytes[i % len(key_bytes)]
    return bytes(output_bytes)
```

```
def generate_run_script(encrypted_data, key):
    from tempfile import NamedTemporaryFile, mkstemp
    import os
    import subprocess
```

```
    # Convert encrypted data to hexadecimal for embedding in the Python script
    encrypted_hex = "".join(format(x, '02x') for x in encrypted_data)
```

```
    # Define the content of the Python script that will decrypt and execute the data
    script_content = f"""import os, subprocess
from tempfile import mkstemp
```

```
# Key for decryption
```

```
key = '{key}'
```

```
# Encrypted data in hexadecimal
```

```
encrypted_hex = '{encrypted_hex}'
```

```

# Convert hex to bytes
encrypted_data = bytes.fromhex(encrypted_hex)

# Decrypt data
decrypted_data = bytearray(encrypted_data)
for i in range(len(decrypted_data)):
    decrypted_data[i] ^= ord(key[i % len(key)])

# Create a new temporary file for the decrypted executable
fd, path = mkstemp(suffix='.out')
os.close(fd) # Close the file descriptor immediately

# Write the decrypted data to the file
with open(path, 'wb') as f:
    f.write(decrypted_data)
os.chmod(path, 0o755)

# Execute the decrypted file
subprocess.run([path])

# Clean up the temporary file after execution
os.unlink(path)
"""

# Write the script content to a named temporary Python file
with NamedTemporaryFile(delete=False, suffix='.py', mode='w') as tmp_file:
    tmp_file.write(script_content)
    os.chmod(tmp_file.name, 0o755)

# Return the path of the temporary Python script
return tmp_file.name

```

```

def encrypt_file(input_path, key):
    with open(input_path, 'rb') as f:
        input_data = f.read()
    encrypted_data = xor_encrypt(input_data, key)
    run_script_path = generate_run_script(encrypted_data, key)
    print(f'Run script generated at {run_script_path}')

```

```

if __name__ == "__main__":
    if len(sys.argv) != 3:
        print("Usage: python encryptor.py <input_file> <key>", file=sys.stderr)
        sys.exit(1)
    encrypt_file(sys.argv[1], sys.argv[2])

```

This is a basic method, mostly meant to highlight the capabilities of using a pip package to handle the "build process", and using Python to handle the runtime. You can always use other methods instead, and add other build steps too if you wish to advance this further. Best of luck & reply in thread if you have any questions. I do not offer any sort of private crypts or support.

Nice work. For people looking for more encryption options try: Code pip Install Cryptodome

Wow! This is impressive. A crypter within the pip packages, many kudos to you OP.

Hackers - June 15, 2024, 6:51 pm

(Jun 15, 2024 06:25 PM)happygonutty Wrote: [\[CITE\]](#) Wow! This is impressive. A crypter within the pip packages, many kudos to you OP.

heh, thanks! I did the same on npm as well for 'npm install nodes-crypter' with same features basically :)

happygonutty - June 15, 2024, 7:14 pm

(Jun 15, 2024 06:51 PM)nodes Wrote: [\[CITE\]](#) heh, thanks! I did the same on npm as well for 'npm install nodes-crypter' with same features basically :)

Lol, thanks for the share. You should have made a separate thread for the npm so that others could notice. Have you posted that here already? ?

SpyNet - June 16, 2024, 12:11 pm

Never done one in Python , i am saving this for future stuff. Thanksss!

Hackers - June 16, 2024, 4:11 pm

(Jun 16, 2024 12:11 PM)SpyNet Wrote: [\[CITE\]](#) Never done one in Python , i am saving this for future stuff. Thanksss!

No problem, this is just a super simple example with XOR. Obviously use something stronger like AES 256. My goal with this is really to give people a unique idea for creating a builder in the Python ecosystem as a pip package, which makes it easy to roll out updates. You can still have private files, for example by having a stub source file that can change, with the builder using a templating language to insert the bytes data. You can also obviously create a pip package for your final program, if your end user is going to use it / has access to pip to install. You can also take the source of it and use the scripts by manually calling them in the command line, rather than relying on the pip public registry package. All in all, I wanted to create something open source that is a good prototype for encryption on programs for personal use.

SpyNet - June 17, 2024, 4:27 am

(Jun 16, 2024 04:11 PM)nodes Wrote: [\[CITE\]](#) No problem, this is just a super simple example with XOR. Obviously use something stronger like AES 256. My goal with this is really to give people a unique idea for creating a builder in the Python ecosystem as a pip package, which makes it easy to roll out updates. You can still have private files, for example by having a stub source file that can change, with the builder using a templating language to insert the bytes data. You can also obviously create a pip package for your final program, if your end user is going to use it / has access to pip to install. You can also take the source of it and use the scripts by manually calling them in the command line, rather than relying on the pip public registry package. All in all, I wanted to create something open source that is a good prototype for encryption on programs for personal use.

Yes i figured that, but yes constant updates is good with this one.

Good idea, some knowledge for future stuff!

Hackers - June 17, 2024, 5:57 am

(Jun 17, 2024 04:27 AM)SpyNet Wrote: [\[CITE\]](#) Yes i figured that, but yes constant updates is good with this one.

Good idea, some knowledge for future stuff!

Yeah definitely - there are a lot of members who have posted open source on HF and on github, but using these types of package managers (pip, npm, yarn, apt repos, etc) can be super helpful

SpyNet - June 17, 2024, 6:12 am

(Jun 17, 2024 05:57 AM)nodes Wrote: [\[CITE\]](#) Yeah definitely - there are a lot of members who have posted open source on HF and on github, but using these types of package managers (pip, npm, yarn, apt repos, etc) can be super helpful
Yes auto updates and so on, much easier

I am a big Python developer but focusing much more on my own rat, not my own rat, im on the developer team ,we are realasing soon. Really happy with the results but this type of python stuff for other means could be good aswell

Hackers - November 19, 2024, 12:53 pm

(Jun 17, 2024 06:12 AM)SpyNet Wrote: [\[CITE\]](#) Yes auto updates and so on, much easier

I am a big Python developer but focusing much more on my own rat, not my own rat, im on the developer team ,we are realasing soon. Really happy with the results but this type of python stuff for other means could be good aswell

Looks like you got closed :O

Definitely love the fact that I could come and make a 2.0 and release via the registry on this tool...
It's nice to see that it is still available there. I wish more people here were willing to go open source.
Oh well!

If anyone has any questions, please just msg me or reply here :)
Happy to help

Stanley - March 16, 2026, 8:30 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

kriminaltetic - March 18, 2026, 12:19 pm

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

Hackers - March 18, 2026, 12:35 pm

(Mar 18, 2026 12:19 PM)kriminaltetic Wrote: [\[CITE\]](#) Windows Defender is detecting a threat; it's not passing through the smart app. Fix it now.

Re-read the thread please, I never said I would update this forever, and I'm whitehat. If you are using this loader with something detectable, that's unrelated to this project.

kriminaltetic - March 18, 2026, 1:07 pm

(Mar 18, 2026 12:35 PM)Hackers Wrote: [\[CITE\]](#) Re-read the thread please, I never said I would update this forever, and I'm whitehat. If you are using this loader with something detectable, that's unrelated to this project.

Hey, don't get me wrong—I didn't say I was updating it. If anyone's already working on it, I'll definitely use it. I just wrote this for informational purposes for those who are new to it or trying to compile it. If you want to release a new crypter, I'm already following the thread.

Hackers - March 18, 2026, 1:22 pm

(Mar 18, 2026 01:07 PM)kriminaltetic Wrote: [\[CITE\]](#) Hey, don't get me wrong—I didn't say I was updating it. If anyone's already working on it, I'll definitely use it. I just wrote this for informational purposes for those who are new to it or trying to compile it. If you want to release a new crypter, I'm already following the thread.

Just curious, is the detection on the NPM package itself or is it detecting the payload in runtime?

I just want to make sure snake-crypter is not being falsely flagged as a threat just because it uses encryption & a run method.

DramaKing - March 19, 2026, 7:19 am

(Mar 18, 2026 01:22 PM)Hackers Wrote: [\[CITE\]](#) Just curious, is the detection on the NPM package itself or is it detecting the payload in runtime?

I just want to make sure snake-crypter is not being falsely flagged as a threat just because it uses encryption & a run method.

The crypter is a few years old. I don't know a lot about AV detection routines, but it seems to me that WD could have been updated with a decryption function. The alert message should give a clue.

Hackers - March 19, 2026, 7:29 am

(Mar 19, 2026 07:19 AM)DramaKing Wrote: [\[CITE\]](#) The crypter is a few years old. I don't know a lot about AV detection routines, but it seems to me that WD could have been updated with a decryption function. The alert message should give a clue.

I don't use Windows anymore so it's hard for me to know. I don't mind if it's detected, it will keep the world safer and this is really just a proof of concept for a unique way to load a program.