Introduction to CyberChef

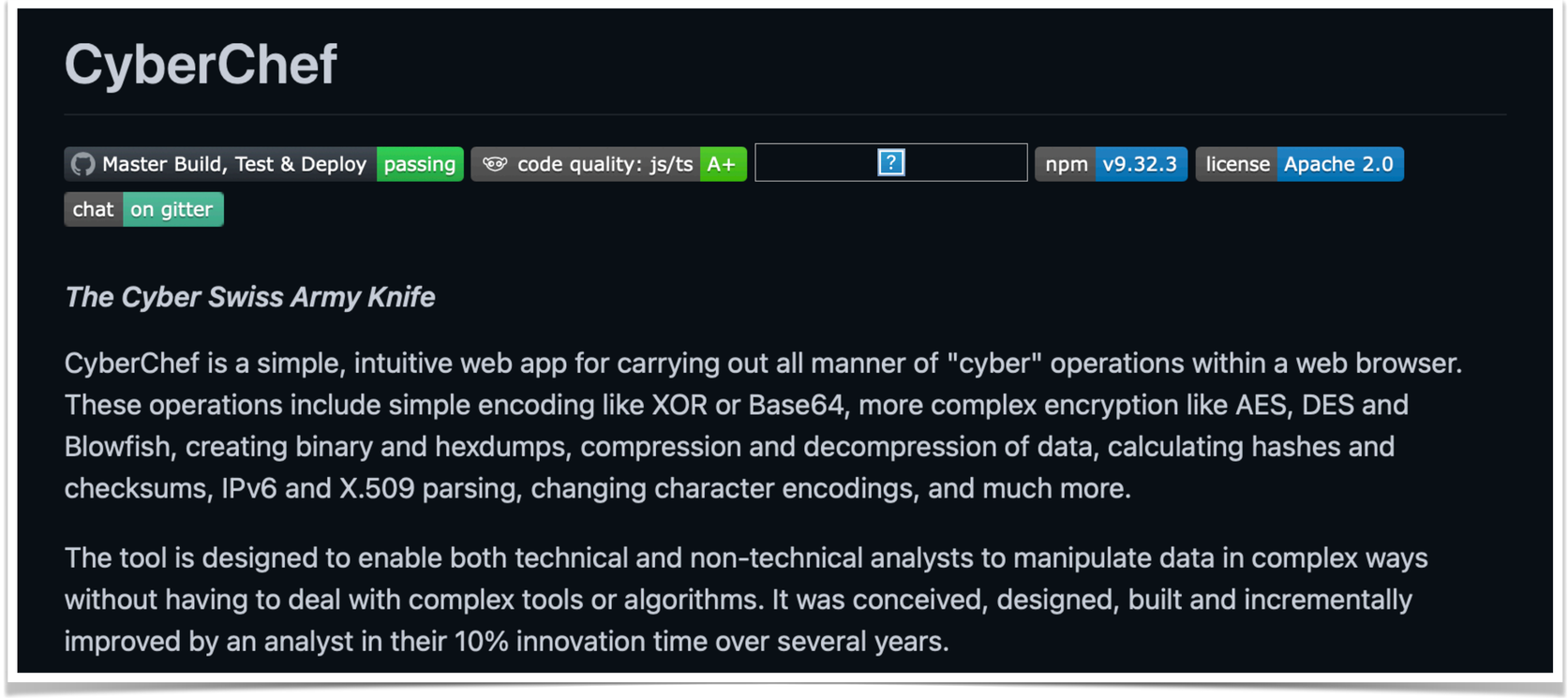
(For Malware Analysis)

Xavier Mertens | PTS22 | July 2022
[TLP:White]

Who's Talking?

- Xavier Mertens (@xme)
- Freelance consultant based in Belgium
- Hunting bad guys
- SANS ISC Senior Handler
- SANS Instructor (FOR610)
- BruCON Co-Organizer

Introduction to CyberChef



Introduction to CyberChef

Developed by GCHQ

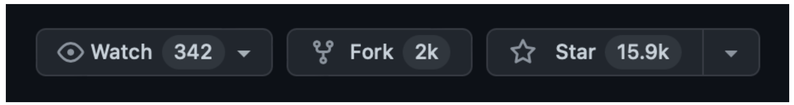
Regularly updated

Easy to use (if you can use a browser, you can use it)

Client side application (important from a data management point of view)

Very popular

Based on “recipes”



Interesting Features

Load & save recipes (read: build your lib and/or share with peers)

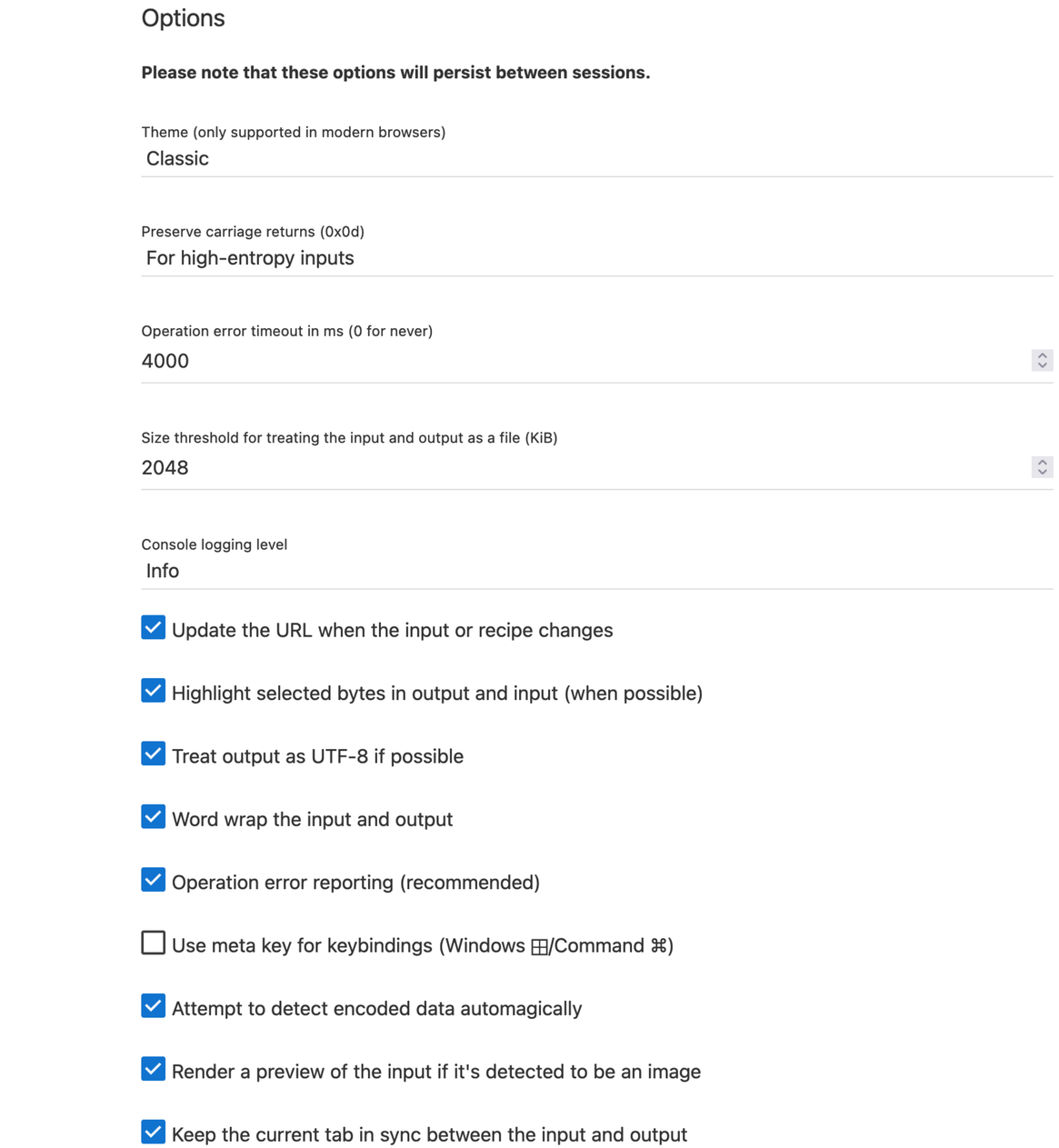
Load & save files (can add cyberchef in a process flow)

Search, highlight text

Customizable…

Customizable

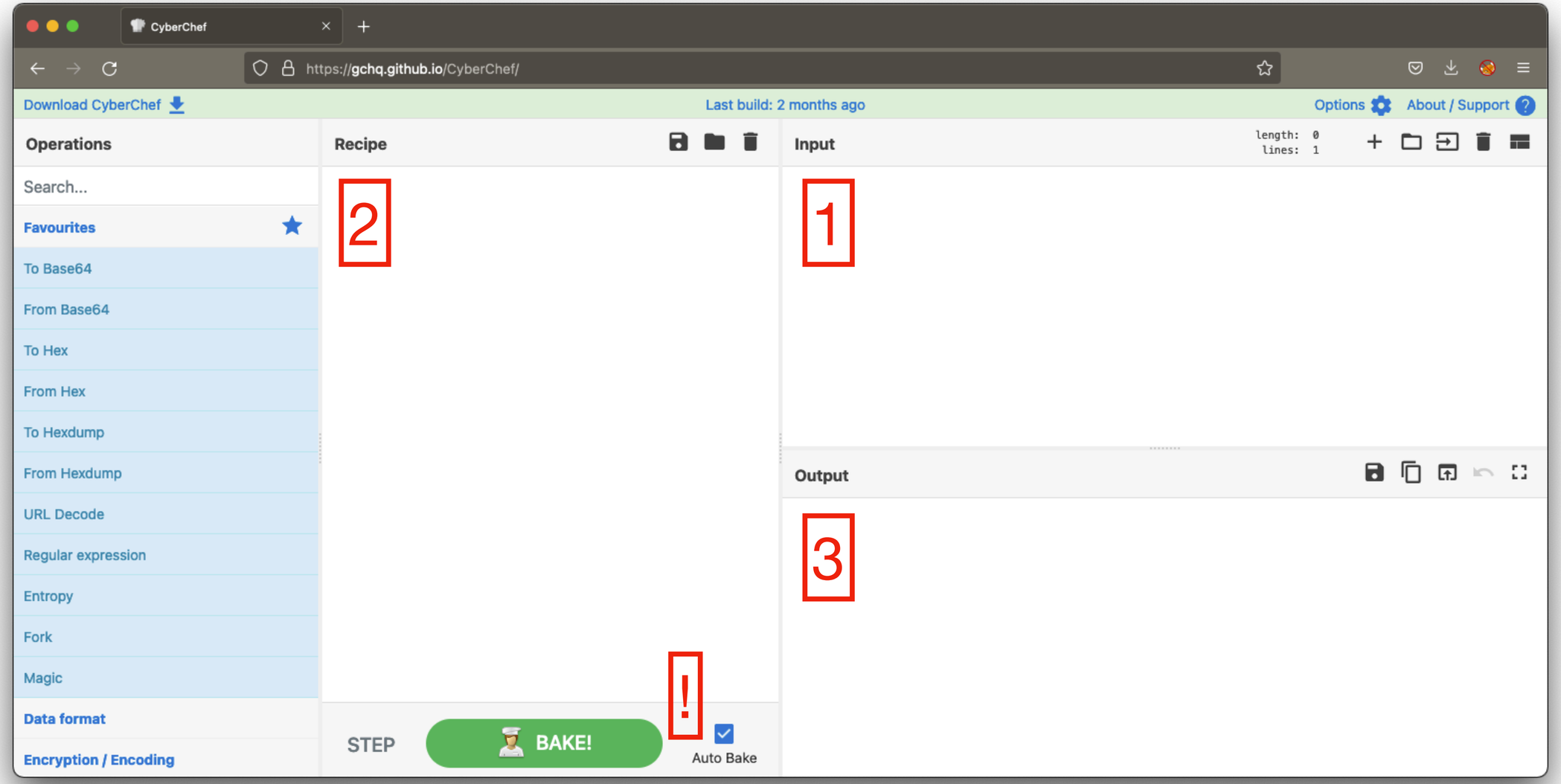
Feel like $HOME!



Quick Access

Online	https://gchq.github.io/CyberChef/
Offline	$ firefox /usr/local/cyberchef/CyberChef_v9.28.0.html &
Docker	$ docker pull mpepping/cyberchef

Simple as 1, 2, 3...



Operations

Data Format

Encryption/Encoding/Keys

Arithmetic/Logic

Networking

Language

Utils

Date/Time

Extractors

Compression

Hashes

Code Tidy

Forensics

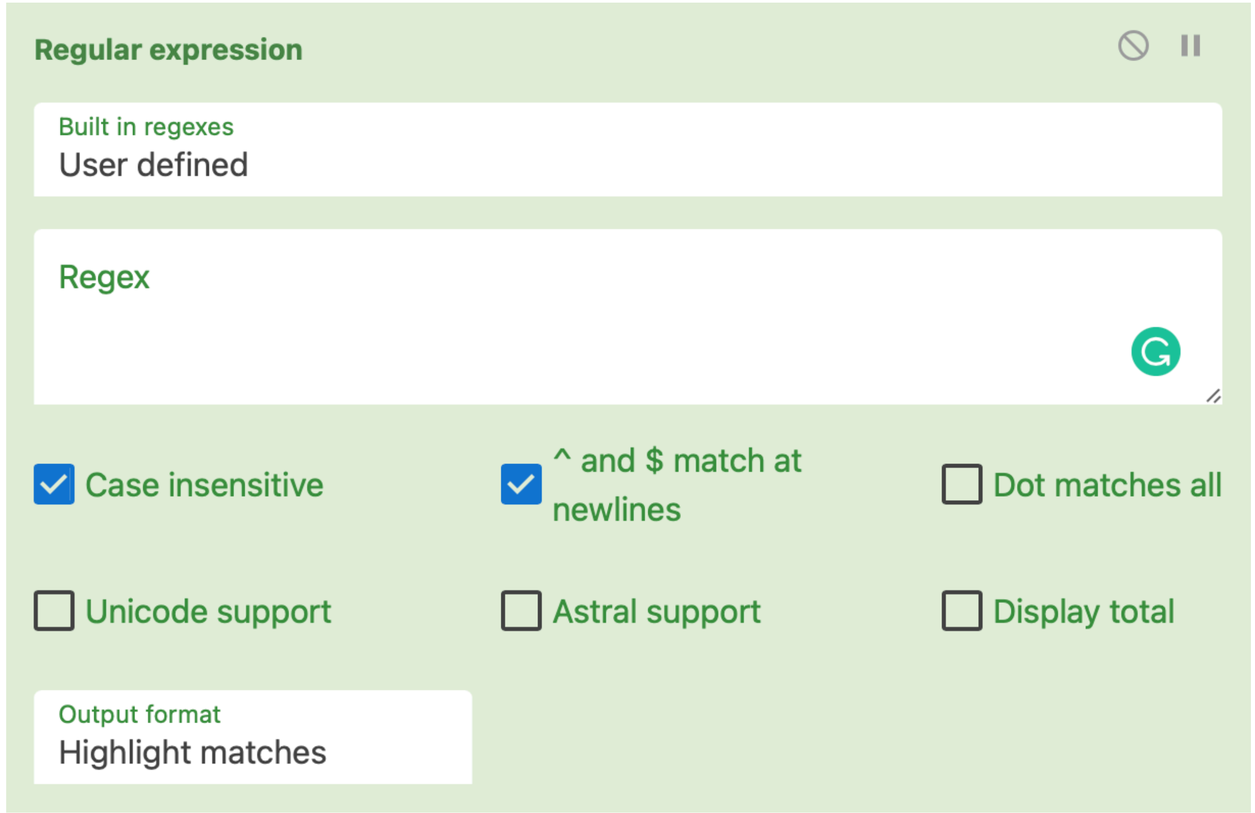
Multimedia

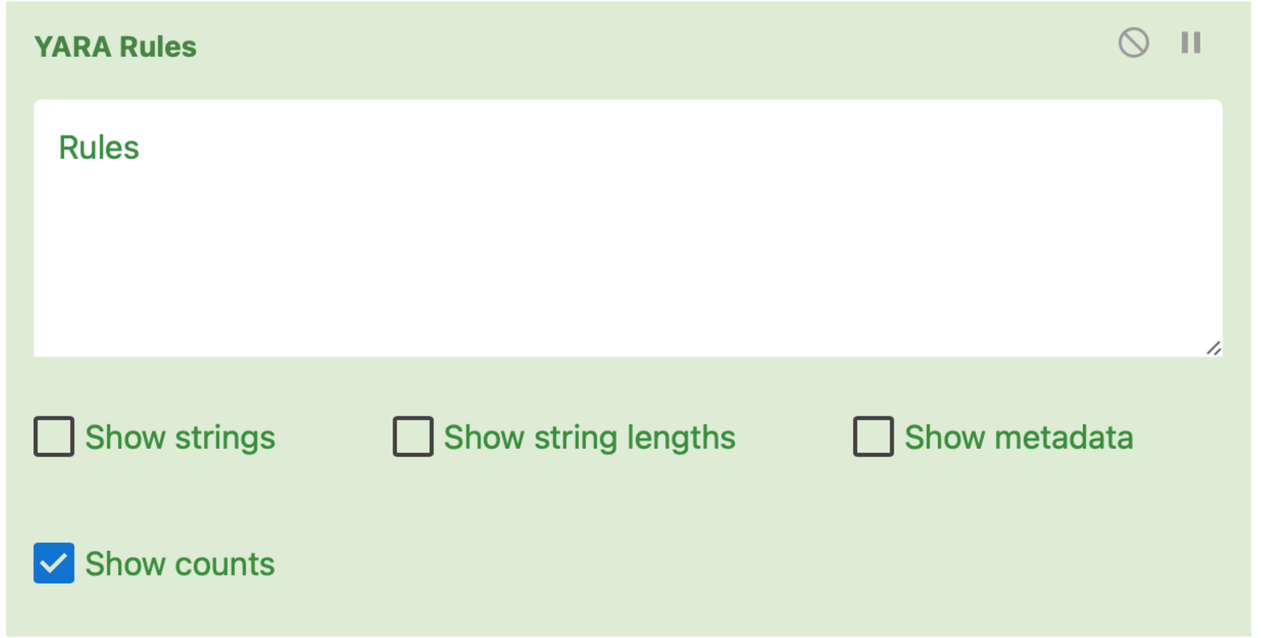
Flow Control

….

Regex

Expand Capabilities





Friendly Helper

Cyberchef will try to suggest a simple recipe…

Extracting Base64

Base64 is everywhere

Native PowerShell encoding

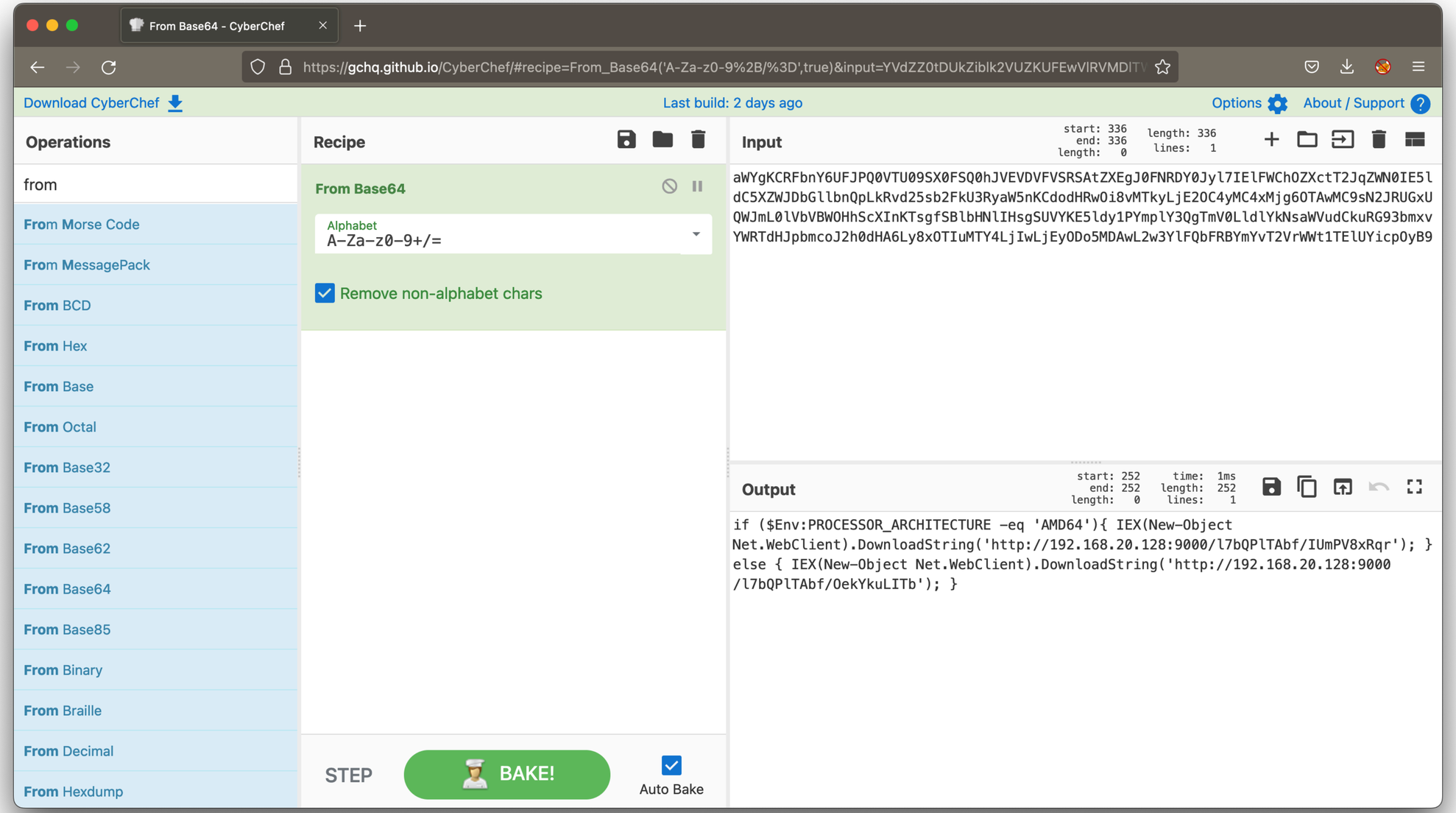
Macros

Python

…

Demo1

Extracting Base64

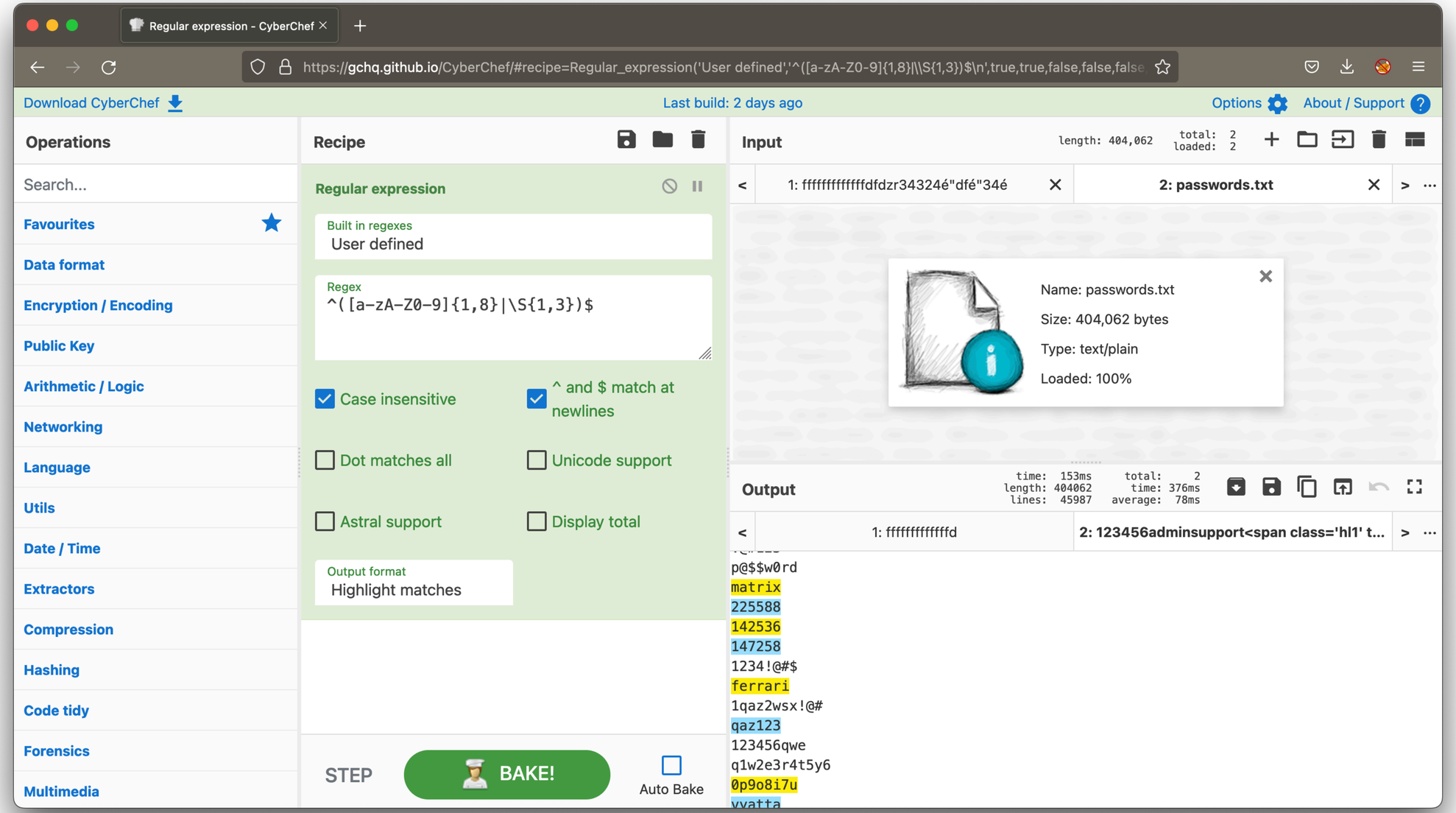


Simple Regexes

Let’s try to find weakest passwords from a list…

Demo2

Simple Regexes

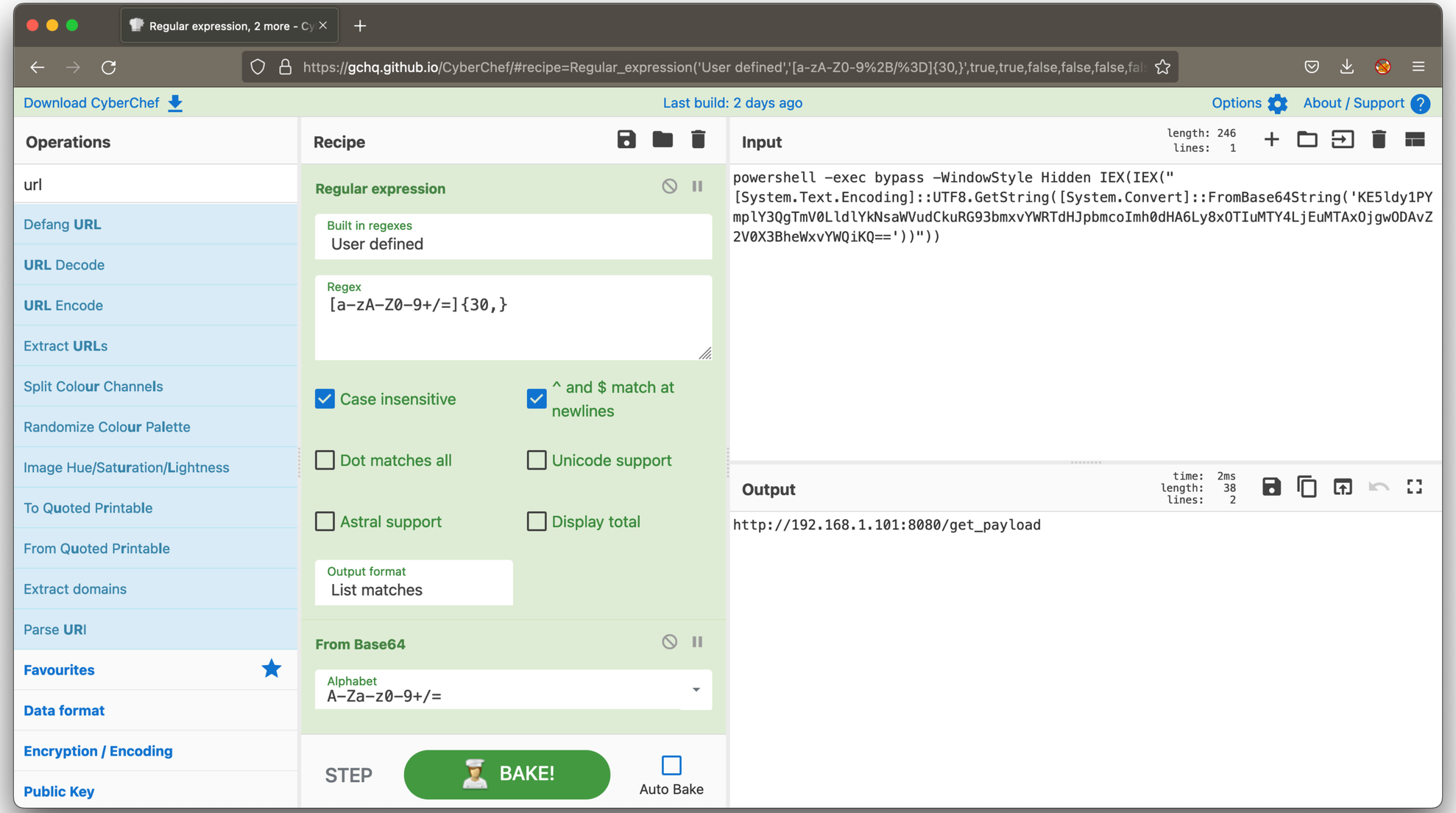


Chaining Operations

How can we deobuscate a simple script?

Demo3

Chaining Operations

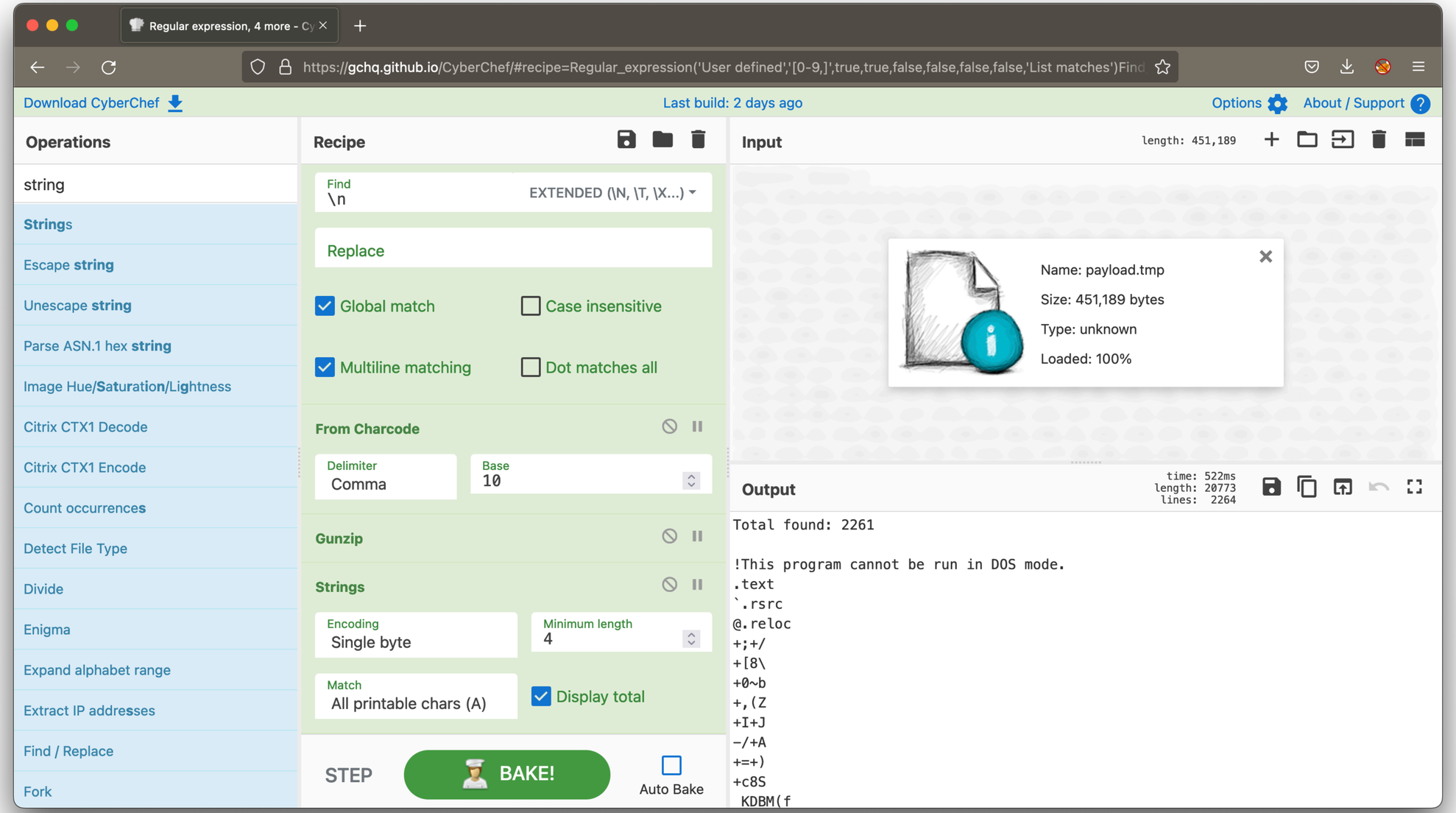


Deobfuscating Payload

In many malicious scripts, payload are encode/compress/… multiple times.

Demo4

Deobfuscating Payload



Extract Shellcode from Maldoc

Maldocs are very popular for a long time

Cyberchef does not support the OLE2 format (legacy format)

Need to extract the initial payload with other tools

Demo5

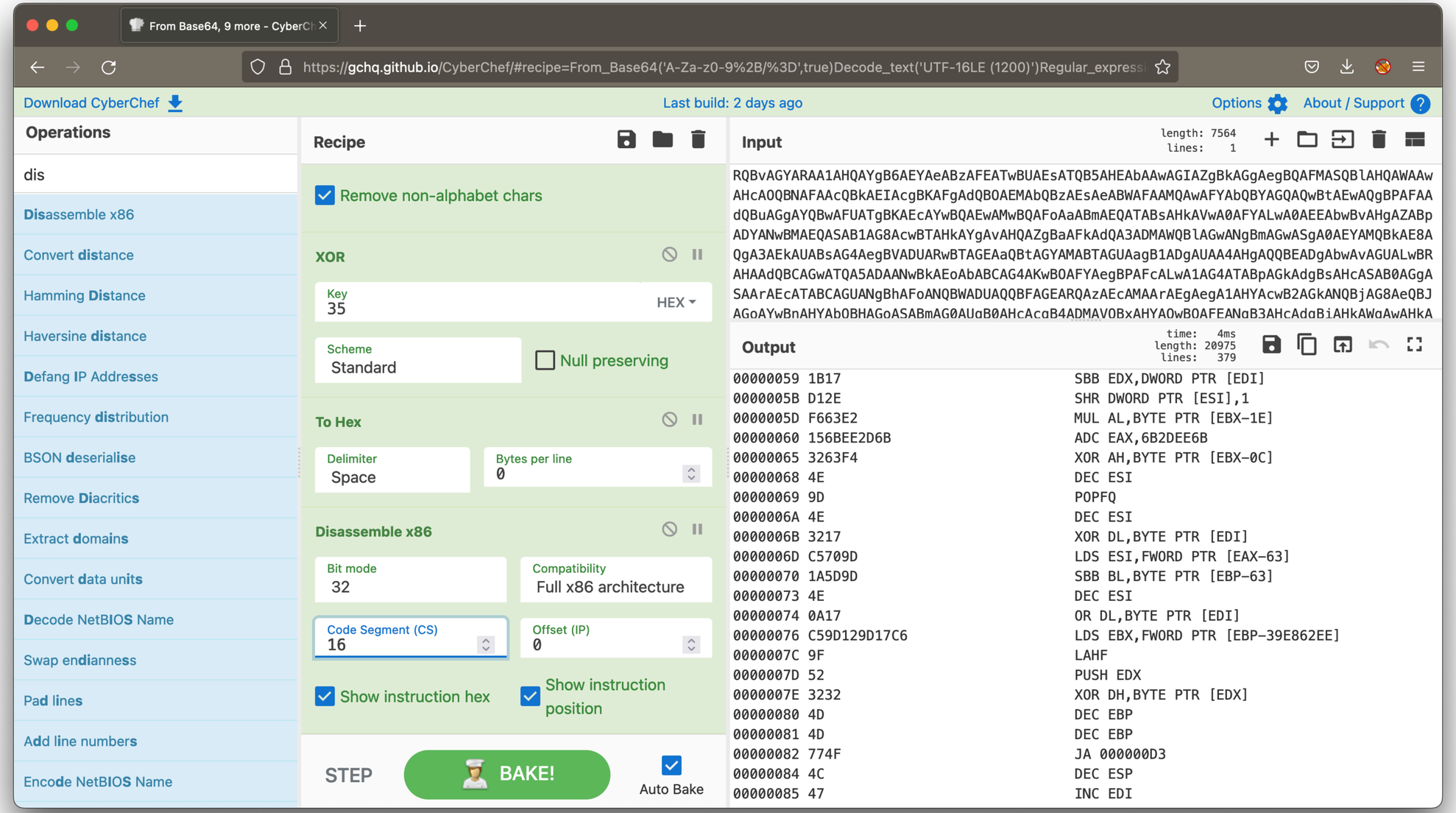
Extract Shellcode from Maldoc

```
remnux@remnux:~/malwarezoo/$ oledump.py checkbox.doc
  1:        114 '\x01CompObj'
  2:       7097 '1Table'
  3:        520 'Macros/PROJECT'
  4:         95 'Macros/PROJECTwm'
  5:         97 'Macros/UserForm1/\x01CompObj'
  6:        292 'Macros/UserForm1/\x03VBFrame'
  7:       7662 'Macros/UserForm1/f'
  8:         84 'Macros/UserForm1/o'
  9: M      163 'Macros/VBA/Module1'
 10: m      157 'Macros/VBA/ThisDocument'
 11: m      231 'Macros/VBA/UserForm1'
 12:          7 'Macros/VBA/_VBA_PROJECT'
 13:        812 'Macros/VBA/dir'
 14:       4096 'WordDocument'
```

Extract Shellcode from Maldoc

```
remnux@remnux:~/malwarezoo/$ oledump.py checkbox.doc -s 7 -S
CheckBox1
JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdABy
AGUAYQBtACgALABbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYA
NABTAHQAcgBpAG4AZwAoACIASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEAQQBLADEAWAA2
ADMATwBpAHkAQgBiAC8ASABQADgASwBQAHEAUgBLAEwAVQAyAEMAagB4AGkAZABByAFYA
UwBOAEQAeABBAFEAMABJAGkAdgBtAEUAMgBsAEcAbQBnAFIAUgBjAEMAbQBBAFgARgBu
AC8AdgBjADkAbwBHAFkAegBPADUAbAA3AHAAKwBwAGUAcQB5AGkANwBtAC8AUAA4AG4A
VQBjAGYATgBFAHgAdgBOAEUAcABzAGcAeQBxAGUAaQBaAG0AYgBHAFMAYQBCADcAYgBs
AE0ATgBaAGUANwA3AG4AawBpAFoAUgA2AFoAcgAvAG4AYwBLAG4AUQBOAG0AaAA2AG4A
aQB6AGMATAAwAHoAZQBmAGUATQBZAGIATQBrADIAQwBnADQARAA1AEsAMwBjADEAUQBn
AFQAdABtAE0ASgAxAGgATQBqAGIAegBqAE4ARABCADUAZQBaAGIASgBNAFMAWQBqAE0A
awB1AEgAaAAxAGwAYgB2AEsAagBrAEkAMwBRAEMAdgA4ADUAaQBKAHEAUgAvAGgAdABo
ACsAbgBhAE0AdwBOAFEAVgBIAGgAcAArADMANwBQADIAeQBIAGIAZgBmADMAeQBwAFIA
cwBTAGcAbAAxADYAMgB0AC8AMgBNAFcAMABIAEEAZAA3AHAAagBvADIARABRAHAASAA1
AHgAcwB6AFgAbQBPAEMAYgBvAGIANwBCAEIAbQBYACsAWQBxADcAZgBiAHYAdQ
```

Extract Shellcode from Maldoc



Alternative to CyberChef?

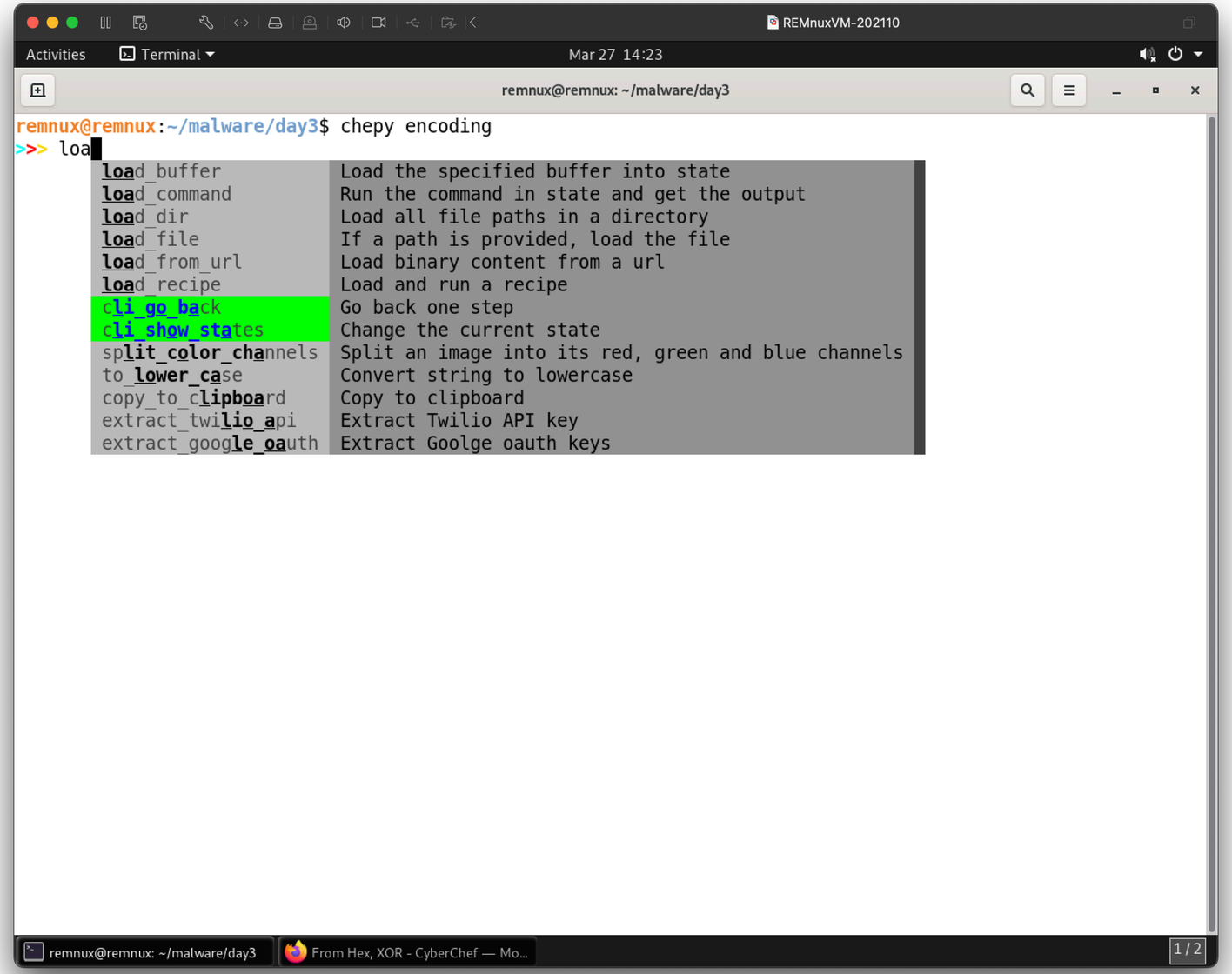
CyberChef requires a browser to play with data.

What if you need to process data on a headless system?

Let’s try “Chepy”!

https://github.com/securisec/chepy

Chepy


REMnuxVM-202110
Activities
Terminal
Mar 27 14:23
remnux@remnux: ~/malware/day3
remnux@remnux:~/malware/day3$ chepy encoding
>>> loa
load_buffer Load the specified buffer into state
load_command Run the command in state and get the output
load_dir Load all file paths in a directory
load_file If a path is provided, load the file
load_from_url Load binary content from a url
load_recipe Load and run a recipe
cli_go_back Go back one step
cli_show_states Change the current state
split_color_channels Split an image into its red, green and blue channels
to_lower_case Convert string to lowercase
copy_to_clipboard Copy to clipboard
extract_twilio_api Extract Twilio API key
extract_google_oauth Extract Goolge oauth keys
remnux@remnux: ~/malware/day3
From Hex, XOR - CyberChef — Mo...
1 / 2

Full Automation

Chepy is written in Python

Available as a command line but also as a module!

Easy as “pip3 install chepy”

```
from chepy import Chepy

file_path = “/tmp/payload.txt"

print(
    Chepy(file_path)
    .load_file()
    .reverse()
    .rot_13()
    .base64_decode()
    .base32_decode()
    .hexdump_to_str()
    .o
)
```

Thank You!

And Happy Hunting!

! or ?