



Sign in

nccgroup / redsnarf Public

Notifications

Fork 239

Star 1.2k

<> Code Issues 4 Pull requests 1 Actions Projects Security Insights

redsnarf / redsnarf.py



5403 lines (4416 loc) · 247 KB

Code Blame

Raw



```
1  #! /usr/bin/python
2  # Released as open source by NCC Group Plc - https://www.nccgroup.trust/uk/
3  # https://github.com/nccgroup/redsnarf
4  # Released under Apache V2 see LICENCE for more information
5
6  from __future__ import print_function
7  import os, argparse, signal, sys, re, binascii, subprocess, string, SimpleHTTPServer, multiprocessing
8  import socket, fcntl, struct, time, base64, logging, urllib
9
10 import time
11 import xml.etree.ElementTree as ET
12
13 try:
14     from nmb.NetBIOS import NetBIOS
15 except ImportError:
16     print("You need to install pysmb")
17     print("pip install pysmb")
18     logging.error("pysmb missing")
19     exit(1)
20
21 try:
22     from docopt import docopt
23 except ImportError:
24     print("You need to install docopt")
25     print("pip install docopt")
26     logging.error("docopt missing")
```

```
27         exit(1)
28
29     try:
30         from pykeyboard import PyKeyboard
31     except ImportError:
32         print("You need to install pyuserinput")
33         print("pip install pyuserinput")
34         logging.error("pyuserinput missing")
35         exit(1)
36
37     try:
38         from pymouse import PyMouseEvent
39     except ImportError:
40         print("You need to install pyuserinput")
41         print("pip install pyuserinput")
42         logging.error("pyuserinput missing")
43         exit(1)
44
45     try:
46         import wget
47     except ImportError:
48         print("You need to install wget")
49         print("pip install wget")
50         logging.error("wget missing")
51         exit(1)
52
53     try:
54         from libnmap.process import NmapProcess
55     except ImportError:
56         print("You need to install python-libnmap")
57         print(" git clone https://github.com/savon-noir/python-libnmap.git")
58         print(" cd python-libnmap")
59         print(" python setup.py install")
60         logging.error("NmapProcess missing")
61         exit(1)
62
63     try:
64         from libnmap.parser import NmapParser
65     except ImportError:
66         print("You need to install python-libnmap")
67         print(" git clone https://github.com/savon-noir/python-libnmap.git")
68         print(" cd python-libnmap")
69         print(" python setup.py install")
70         logging.error("NmapProcess missing")
71         exit(1)
72
```

```
73     from random import shuffle
74
75     # Logging definitions
76     logging.basicConfig(level=logging.DEBUG, format='%(asctime)s %(levelname)s %(message)s', filename='
77     logging.debug("Command parameters run: %s", sys.argv[1:])
78
79     try:
80         import ldap
81     except ImportError:
82         print("Try installing these modules to help with this error")
83         print("run 'pip install python-ldap' to install ldap module.")
84         print("apt-get install libpq-dev python-dev libxml2-dev libxslt1-dev libldap2-dev libsasl2-
85         print("apt-get install python2.7-dev")
86         logging.error("ldap dependencies missing")
87         exit(1)
88
89     try:
90         from IPy import IP
91     except ImportError:
92         print("You need to install IPy module: apt-get install python-ipy")
93         logging.error("IPy missing")
94         exit(1)
95
96     try:
97         from netaddr import IPNetwork
98     except ImportError:
99         print('Netaddr appears to be missing - try: pip install netaddr')
100         logging.error("Netaddr missing")
101         exit(1)
102
103     try:
104         from termcolor import colored
105     except ImportError:
106         print('termcolor appears to be missing - try: pip install termcolor')
107         logging.error("termcolor missing")
108         exit(1)
109
110     from Crypto.Cipher import DES3
111     from Crypto.Hash import SHA
112     from Crypto.Cipher import AES
113     from base64 import b64decode
114     from socket import *
115     from threading import Thread
116     from impacket.smbconnection import *
117     from random import randint
118     from base64 import b64encode
```

110 from urllib import urlopen


```
5330
5331                                     #Get DC ip from domain name
5332                                     dcip=socket.gethostbyname(domain_name)
5333
5334                                     #Query RPC for user details
5335                                     proc = subprocess.Popen("rpcclient "+dcip+'
5336                                     stdout_value = proc.communicate()[0]
5337
5338                                     #Cycle through output
5339                                     for line in stdout_value.splitlines():
```

```
5340                                     #If we hit a user_rid line grab info
5341                                     if "user_rid" in line:
5342                                         proc = subprocess.Popen("rpcclient "+dcip+" -U "+user+"%"+passw+" -c \\"queryuser "+
5343                                         stdout_value = proc.communicate()[0]
5344                                         #Cycle output
5345                                         for grid in stdout_value.splitlines():
5346                                             #If we hit group rid, grab info and querygroup to get full
5347                                             if "group rid" in grid:
5348                                                 proc = subprocess.Popen("rpcclient "+dcip+" -U "+user+"%"+passw+" -c \\"querygroup "+
5349                                                 stdout_value = proc.communicate()[0]
5350                                                 #This (should) print Group Memberships to screen
5351                                                 print(stdout_value)
5352
5353                                     except IOError as e:
5354                                         print("I/O error({}): {}".format(e.errno, e.strerror))
5355
5356                                     else:
5357                                         print(colored("[-]No service accounts found: "+targets[0],'red'))
5358                                         logging.info("[-]No service accounts found: "+targets[0])
5359
5360                                     else:
5361                                         #Undocumented
5362                                         #Get usernames and query domain for memberships
5363                                         print(colored("[+]Attempting to get account details ", 'green')+service_accounts)
5364
5365                                         #Get Group Membership using RPC to enumerate details
5366                                         print(colored("[+]Attempting to retrieve Group Membership via RPC", 'green'))
5367
5368                                         #Get DC ip from domain name
5369                                         dcip=socket.gethostbyname(domain_name)
5370
5371                                         #Query RPC for user details
5372                                         proc = subprocess.Popen("rpcclient "+dcip+" -U "+user+"%"+passw+" -c \\"queryuser "+
5373                                         stdout_value = proc.communicate()[0]
5374
5375                                         #Cycle through output
5376                                         for line in stdout_value.splitlines():
5377                                             #If we hit a user_rid line grab info and queryusergroups
5378                                             if "user_rid" in line:
5379                                                 proc = subprocess.Popen("rpcclient "+dcip+" -U "+user+"%"+passw+" -c \\"queryuser "+
5380                                                 stdout_value = proc.communicate()[0]
5381                                                 #Cycle output
5382                                                 for grid in stdout_value.splitlines():
5383                                                     #If we hit group rid, grab info and querygroup to get full
5384                                                     if "group rid" in grid:
5385                                                         proc = subprocess.Popen("rpcclient "+dcip+" -U "+user+"%"+passw+" -c \\"querygroup "+
5386                                                         stdout_value = proc.communicate()[0]
5387                                                         #This (should) print Group Memberships to screen
```

```
5385             # This (should) print group memberships to screen
5386             print(stdout_value)
5387
5388         sys.exit()
5389
5390     if targets is None:
5391         print(colored('[+]You have not entered a target!, Try --help for a list of parameters','red'))
5392         sys.exit()
5393
5394     syschecks()
5395
5396     if __name__ == '__main__':
5397         signal.signal(signal.SIGINT, signal_handler)
5398         main()
5399         now = time.strftime("%c")
5400
5401         print(colored("[+]Scan Stop " + time.strftime("%c"),'blue'))
5402         print(colored("[+]end - check redsnarf.log for log related information",'green'))
5403         logging.info("[+]end")
```