redsnarf / redsnarf.py

5403 lines (4416 loc) · 247 KB

Code Blame Raw

```
#! /usr/bin/python
# Released as open source by NCC Group Plc - https://www.nccgroup.trust/uk/
# https://github.com/nccgroup/redsnarf
# Released under Apache V2 see LICENCE for more information

from __future__ import print_function
import os, argparse, signal, sys, re, binascii, subprocess, string, SimpleHTTPServer, multiprocessi
import socket, fcntl, struct, time, base64, logging, urllib

import time
import xml.etree.ElementTree as ET

try:
        from nmb.NetBIOS import NetBIOS
except ImportError:
        print("You need to install pysmb")
        print("pip install pysmb")
        logging.error("pysmb missing")
        exit(1)

try:
        from docopt import docopt
except ImportError:
        print("You need to install docopt")
        print("pip install docopt")
        logging.error("docopt missing")
```

```
        exit(1)

try:
        from pykeyboard import PyKeyboard
except ImportError:
        print("You need to install pyuserinput")
        print("pip install pyuserinput")
        logging.error("pyuserinput missing")
        exit(1)

try:
        from pymouse import PyMouseEvent
except ImportError:
        print("You need to install pyuserinput")
        print("pip install pyuserinput")
        logging.error("pyuserinput missing")
        exit(1)

try:
        import wget
except ImportError:
        print("You need to install wget")
        print("pip install wget")
        logging.error("wget missing")
        exit(1)

try:
        from libnmap.process import NmapProcess
except ImportError:
        print("You need to install python-libnmap")
        print(" git clone https://github.com/savon-noir/python-libnmap.git")
        print(" cd python-libnmap")
        print(" python setup.py install")
        logging.error("NmapProcess missing")
        exit(1)

try:
        from libnmap.parser import NmapParser
except ImportError:
        print("You need to install python-libnmap")
        print(" git clone https://github.com/savon-noir/python-libnmap.git")
        print(" cd python-libnmap")
        print(" python setup.py install")
        logging.error("NmapProcess missing")
        exit(1)

```

```
from random import shuffle

# Logging definitions
logging.basicConfig(level=logging.DEBUG, format='%(asctime)s %(levelname)s %(message)s', filename='
logging.debug("Command parameters run: %s", sys.argv[1:])

try:
        import ldap
except ImportError:
        print("Try installing these modules to help with this error")
        print("run 'pip install python-ldap' to install ldap module.")
        print("apt-get install libpq-dev python-dev libxml2-dev libxslt1-dev libldap2-dev libsasl2-
        print("apt-get install python2.7-dev")
        logging.error("ldap dependencies missing")
        exit(1)

try:
        from IPy import IP
except ImportError:
        print("You need to install IPy module: apt-get install python-ipy")
        logging.error("IPy missing")
        exit(1)

try:
        from netaddr import IPNetwork
except ImportError:
        print('Netaddr appears to be missing - try: pip install netaddr')
        logging.error("Netaddr missing")
        exit(1)

try:
        from termcolor import colored
except ImportError:
        print('termcolor appears to be missing - try: pip install termcolor')
        logging.error("termcolor missing")
        exit(1)

from Crypto.Cipher import DES3
from Crypto.Hash import SHA
from Crypto.Cipher import AES
from base64 import b64decode
from socket import *
from threading import Thread
from impacket.smbconnection import *
from random import randint
from base64 import b64encode
```

```
from base64 import b64encode
```

```

                                #Get DC ip from domain name
                                dcip=socket.gethostbyname(domain_name)

                                #Query RPC for user details
                                proc = subprocess.Popen("rpcclient "+dcip+"
                                stdout_value = proc.communicate()[0]

                                #Cycle through output
                                for line in stdout_value.splitlines():
```

```
                                    #If we hit a user_rid line grab inf
                                    if "user_rid" in line:
                                        proc = subprocess.Popen("rp
                                        stdout_value = proc.communi
                                        #Cycle output
                                        for grid in stdout_value.sp
                                            #If we hit group ri
                                            if "group rid" in g
                                                proc = subp
                                                stdout_valu
                                                #This (shou
                                                print(stdou

                except IOError as e:
                    print("I/O error({0}): {1}".format(e.errno, e.strerror))
            else:
                print(colored("[-]No service accounts found: "+targets[0],'red'))
                logging.info("[-]No service accounts found: "+targets[0])
        else:
            #Undocumented
            #Get usernames and query domain for memberships
            print(colored("[+]Attempting to get account details ",'green')+service_accounts)

            #Get Group Membership using RPC to enumerate details
            print(colored("[+]Attempting to retrieve Group Membership via RPC",'green'))

            #Get DC ip from domain name
            dcip=socket.gethostbyname(domain_name)

            #Query RPC for user details
            proc = subprocess.Popen("rpcclient "+dcip+" -U "+user+"%"+passw+" -c \"queryuser "+
            stdout_value = proc.communicate()[0]

            #Cycle through output
            for line in stdout_value.splitlines():
                #If we hit a user_rid line grab info and queryusergroups
                if "user_rid" in line:
                    proc = subprocess.Popen("rpcclient "+dcip+" -U "+user+"%"+passw+" -
                    stdout_value = proc.communicate()[0]
                    #Cycle output
                    for grid in stdout_value.splitlines():
                        #If we hit group rid, grab info and querygroup to get full
                        if "group rid" in grid:
                            proc = subprocess.Popen("rpcclient "+dcip+" -U "+us
                            stdout_value = proc.communicate()[0]
                            #This (should) print Group Memberships to screen
```

```
                                                #This (should) print Group Memberships to screen
                                                print(stdout_value)

        sys.exit()

if targets is None:
        print(colored('[-]You have not entered a target!, Try --help for a list of parameters','red
        sys.exit()

syschecks()

if __name__ == '__main__':
        signal.signal(signal.SIGINT, signal_handler)
        main()
        now = time.strftime("%c")

        print(colored("[+]Scan Stop " + time.strftime("%c"),'blue'))
        print(colored("[+]end - check redsnarf.log for log related information",'green'))
        logging.info("[+]end")
```