Product Solutions Resources Open Source Enterprise Pricing Sign in Sign up

nccgroup / redsnarf Public

Notifications Fork 239 Star 1.2k

Code Issues 4 Pull requests 1 Actions Projects Security Insights

Files

35949b3

Go to file

- bits_and_bobs
- wupdate
- Invoke-Vnc.ps1
- LICENSE.md
- SessionGopher.ps1
- a
- b
- readme.md
- redsnarf.py
- redsnarf.rc
- rsc.exe
- setup.py
- snarf_client.exe
- snarf_client.py
- snarf_client_32bit.exe

redsnarf / redsnarf.py

cclauss Add missing closing parenthesis) 7301742 · 7 years ago History

Code Blame 5403 lines (4416 loc) · 247 KB 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, m
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(" 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'
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-d
        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
```

```

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

                                                        #Query RPC for user details
                                                        proc = subprocess.Popen("rpccli
                                                        stdout_value = proc.communicate

                                                        #Cycle through output
                                                        for line in stdout_value.splitl
                                                                #If we hit a user_rid l
                                                                if "user_rid" in line:
                                                                        proc = subproce
                                                                        stdout_value =
                                                                        #Cycle output
                                                                        for grid in std
                                                                                #If we
                                                                                if "gro





                        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_

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

                #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 \"
                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+"
                                stdout_value = proc.communicate()[0]
                                #Cycle output
                                for grid in stdout_value.splitlines():
                                        #If we hit group rid, grab info and querygroup
                                        if "group rid" in grid:
                                                proc = subprocess.Popen("rpcclient "+dc
                                                stdout_value = proc.communicate()[0]
                                                #This (should) print Group Memberships
                                                print(stdout_value)

        sys.exit()

if targets is None:
        print(colored('[-]You have not entered a target!, Try --help for a list of para
        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")
```