

Notifications

Fork14k

Star34.1k

Code

Issues410

Pull requests43

Discussions


Actions


Projects1


Wiki


Security


Insights


Files


1416b57





Go to file


> .github


> app


> config


> data


> db


> docker


> docs


> documentation


> external


> kubernetes


> lib


> anemone


> metasploit


> msf


> msfdb_helpers


> net


> postgres


> rabal


> rbmysql


> rex


> crypto


> google


> io


> logging


> parser


> payloads


> post


> proto


> acpp


> adb


> dcerpc

> dhcp

> dns

> drda

> ftp

> http

metasploit-framework / lib / rex / proto / smb / client.rb





adfoster-r7Prefix time module with namespace resolution operatoraf33087 · 3 years agoHistory

Code

Blame

2103 lines (1631 loc) · 60.9 KB

Raw







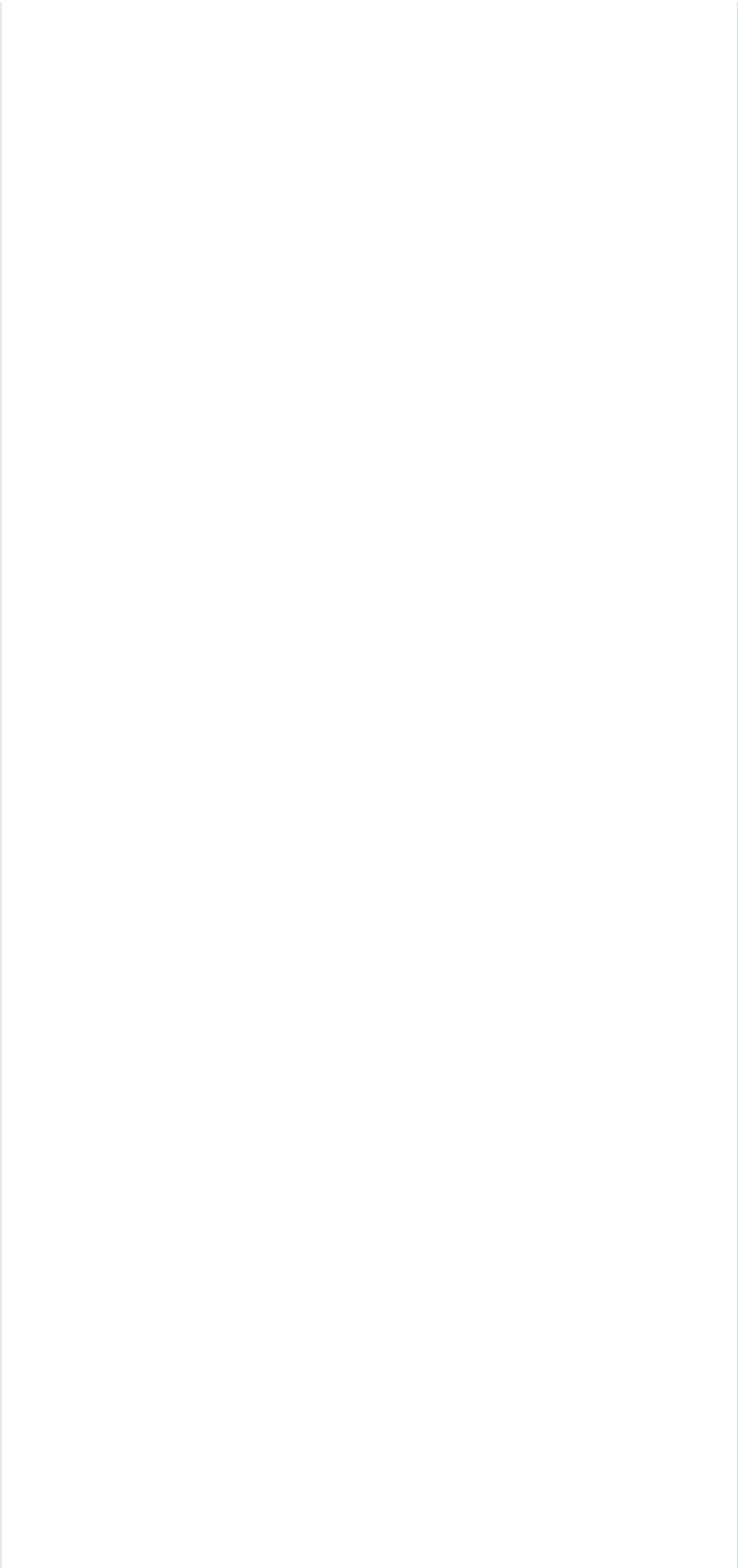
```
1  # -*- coding: binary -*-
2  module Rex
3  module Proto
4  module SMB
5  class Client
6
7      require 'net/ntlm'
8
9      require 'rex/text'
10     require 'rex/struct2'
11
12
13     # Some short-hand class aliases
14     CONST = Rex::Proto::SMB::Constants
15     CRYPT = Rex::Proto::SMB::Crypt
16     UTILS = Rex::Proto::SMB::Utils
17     XCEPT = Rex::Proto::SMB::Exceptions
18     EVADE = Rex::Proto::SMB::Evasions
19     NTLM_CRYPT = Rex::Proto::NTLM::Crypt
20     NTLM_CONST = Rex::Proto::NTLM::Constants
21     NTLM_UTILS = Rex::Proto::NTLM::Utils
22
23     def initialize(socket)
24         self.socket = socket
25         self.native_os = 'Windows 2000 2195'
26         self.native_lm = 'Windows 2000 5.0'
27         self.encrypt_passwords = true
28         self.extended_security = false
29         self.multiplex_id = rand(0xffff)
30         self.process_id = rand(0xffff)
31         self.read_timeout = 10
32         self.evasion_opts = {
33
34             # Padding is performed between packet headers and data
35             'pad_data' => EVADE::EVASION_NONE,
36
37             # File path padding is performed on all open/create calls
38             'pad_file' => EVADE::EVASION_NONE,
39
40             # Modify the \PIPE\ string in trans_named_pipe calls
41             'obscure_trans_pipe' => EVADE::EVASION_NONE,
42         }
43
44         self.verify_signature = false
45         self.use_ntlmv2 = false
46         self.usentlm2_session = true
47         self.send_lm = true
48         self.use_lanman_key = false
49         self.send_ntlm = true
50
51         # Signing
52         self.sequence_counter = 0
53         self.signing_key = ''
54         self.require_signing = false
55         self.peer_require_signing = false
56
57         # Mimic
```

- > iax2
- > ipmi
- > kademlia
- > kerberos
- > ldap
- > mms

```
37         #misc
58         self.spnopt = {}
59         self.default_max_buffer_size = 0xffdf
60     end
61
62     # Read a SMB packet from the socket
63     def smb_recv
64
65         data = socket.timed_read(4, self.read_timeout)
66         if (data.nil? or data.length < 4)
67             raise XCEPT::NoReply
68         end
69
70         recv_len = data[2,2].unpack('n')[0]
71         if (recv_len == 0)
72             return data
73         end
74
75         recv_len += 4
76
77         while (data.length != recv_len)
78             buff = ''
79
80             begin
81                 buff << self.socket.timed_read(recv_len - data.length, self.read_timeout)
82             rescue Timeout::Error
83             rescue
84                 raise XCEPT::ReadPacket
85             end
86
87             if (buff.nil? or buff.length == 0)
88                 raise XCEPT::ReadPacket
89             end
90
91             data << buff
92         end
93
94         #signing
95         if self.require_signing && self.signing_key != ''
96             if self.verify_signature
97                 raise XCEPT::IncorrectSigningError if not CRYPT::is_signature_correct?(self.sig
98             end
99             self.sequence_counter += 1
100         end
101
102         return data
103
104     end
105
106     # Send a SMB packet down the socket
107     def smb_send(data, evasion_level=0)
108         # evasion_level is ignored, since real evasion happens
109         # in the actual socket layer
110
111
112         size = 0
113         wait = 0
114
115         #signing
116         if self.require_signing && self.signing_key != ''
117             data = CRYPT::sign_smb_packet(self.signing_key, self.sequence_counter, data)
118             self.sequence_counter += 1
```












```
2030         end
2031
2032         $stderr.puts [e, e.get_error(e.error_code), search_path]
2033
2034         raise e
2035     end
2036 end
2037
2038 end
2039
2040     files.uniq
2041 end
2042
2043 # Creates a new directory on the mounted tree
2044 ✓ def create_directory(name)
2045     parm = [0].pack('V') + name + "\x00"
2046     resp = trans2(CONST::TRANS2_CREATE_DIRECTORY, parm, '')
2047     resp
2048 end
2049
2050
2051 # Send SMB echo request
2052 ✓ def echo(do_recv = true)
2053
2054     pkt = CONST::SMB_ECHO_RES_PKT.make_struct
2055     self.smb_defaults(pkt['Payload']['SMB'])
2056
2057     pkt['Payload']['SMB'].v['Command'] = CONST::SMB_COM_ECHO
2058     pkt['Payload']['SMB'].v['Flags1'] = 18
2059     if self.require_signing
2060         pkt['Payload']['SMB'].v['Flags2'] = 0x2807
2061     else
2062         pkt['Payload']['SMB'].v['Flags2'] = 0x2801
2063     end
2064
2065     pkt['Payload']['SMB'].v['TreeID'] = self.last_tree_id || 0xffff
2066     pkt['Payload']['SMB'].v['WordCount'] = 1
2067     pkt['Payload'].v['EchoCount'] = 1
2068
2069     ret = self.smb_send(pkt.to_s)
2070     return ret if not do_recv
2071     return self.smb_recv parse(CONST::SMB_COM_ECHO)
```

```
2072         end
2073
2074
2075     # public read/write methods
2076     attr_accessor :native_os, :native_lm, :encrypt_passwords, :extended_security, :read_t
2077     attr_accessor :verify_signature, :use_ntlmv2, :usentlm2_session, :send_lm, :use_lanma
2078     attr_accessor :system_time, :system_zone
2079     attr_accessor :spnopt
2080     attr_accessor :default_max_buffer_size
2081
2082     # public read methods
2083     attr_reader :dialect, :session_id, :challenge_key, :peer_native_lm, :peer_n
2084     attr_reader :default_domain, :default_name, :auth_user, :auth_user_id
2085     attr_reader :multiplex_id, :last_tree_id, :last_file_id, :process_id, :last
2086     attr_reader :dns_host_name, :dns_domain_name
2087     attr_reader :security_mode, :server_guid
2088     attr_reader :sequence_counter, :signing_key, :require_signing, :peer_require
2089
2090     # private write methods
2091     attr_writer :dialect, :session_id, :challenge_key, :peer_native_lm, :peer_n
2092     attr_writer :default_domain, :default_name, :auth_user, :auth_user_id
2093     attr_writer :dns_host_name, :dns_domain_name
2094     attr_writer :multiplex_id, :last_tree_id, :last_file_id, :process_id, :last
2095     attr_writer :security_mode, :server_guid
2096     attr_writer :sequence_counter, :signing_key, :require_signing, :peer_require
2097
2098     attr_accessor :socket
2099
2100     end
2101   end
2102 end
2103 end
```